PLOS ONE

RESEARCH ARTICLE

Efficient multi-task chemogenomics for drug specificity prediction

Benoit Playe[1,2,3]*, **Chloé-Agathe Azencott**[1,2,3], **Véronique Stoven**[1,2,3]

1 Center for Computational Biology, Mines ParisTech, PSL Research University, Paris, France, **2** Institut Curie F-75248, Paris, France, **3** INSERM U900, F-75248, Paris, France

* benoit.playe@mines-paristech.fr

OPEN ACCESS

Citation: Playe B, Azencott C-A, Stoven V (2018) Efficient multi-task chemogenomics for drug specificity prediction. PLoS ONE 13(10): e0204999. https://doi.org/10.1371/journal.pone.0204999

Editor: Alexandre G. de Brevern, UMR-S1134, INSERM, Université Paris Diderot, INTS, FRANCE

Received: January 23, 2018

Accepted: September 18, 2018

Published: October 4, 2018

Data Availability Statement: Code is available at https://github.com/bplaye/efficient_MultiTask_SVM_for_chemogenomics. All data are available at http://members.cbio.mines-paristech.fr/~bplaye/efficient_MT_chemo.tar.gz.

Funding: This work is founded by the French ministry of Industry (https://www.economie.gouv.fr/) as founding our institution (http://www.mines-paristech.eu/). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Abstract

Adverse drug reactions, also called side effects, range from mild to fatal clinical events and significantly affect the quality of care. Among other causes, 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 the application of chemogenomics approaches. We formulate the study of drug specificity as a problem of predicting interactions between drugs and proteins at the proteome scale. We build several benchmark datasets, and propose *NN-MT*, a multi-task Support Vector Machine (SVM) algorithm that is trained on a limited number of data points, in order to solve the computational issues or proteome-wide SVM for chemogenomics. We compare *NN-MT* to different state-of-the-art methods, and show that its prediction performances are similar or better, at an efficient calculation cost. Compared to its competitors, the proposed method is particularly efficient to predict (protein, ligand) interactions in the difficult double-orphan case, i.e. when no interactions are previously known for the protein nor for the ligand. The *NN-MT* algorithm appears to be a good default method providing state-of-the-art or better performances, in a wide range of prediction scenario that are considered in the present study: proteome-wide prediction, protein family prediction, test (protein, ligand) pairs dissimilar to pairs in the train set, and orphan cases.

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 in order to alter disease progression. Once 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].

Competing interests: The authors have declared that no competing interests exist.

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 2216000 patients hospitalized in the US suffered from a serious ADR, and approximately 106000 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]. In that case, the molecular mechanisms at the source of the therapeutic effect and of the unwanted side effects are of similar nature: they both involve interactions between the drug and a protein. However, the complete study of drug specificity at early stages of drug development is experimentally out of reach, since it would require the evaluation of potential interactions between the hit molecule and the entire human proteome. Therefore, there is a strong incentive to develop *in silico* methods that predict specificity. 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.

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, i.e. proteins that present pockets into which drugs can bind. The approaches that have been developed to predict interactions between a protein and a small molecule can be separated into three categories.

First, *ligand-based approaches* such as Quantitative Structure Activity Relationship (QSAR) (refer to [6] for a recent review on QSAR) build a model function that predicts the affinity of any molecule for a given target, based on the affinities of known ligands for this 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 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* (refer to [7] for a recent review on docking), 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.

Finally, *chemogenomic approaches* [8] 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 [9]. In this context, drug specificity prediction is formulated as a classification problem, where the goal is to distinguish protein-ligand pairs that bind from those that do not: the aim is to predict "interacting" or "not interacting" labels for all pairs, but not to predict the strength of the interaction, which would correspond to a regression problem. Chemogenomics 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.

Various chemogenomics methods have been proposed in the last decade [10–24]. They all rely on the assumption that "similar" proteins are expected to bind "similar" ligands. They differ by (i) the descriptors used to encode proteins and ligands, (ii) how similarities between these objects are measured, (iii) the ML algorithm that is used to learn the model and make the predictions.

Predecessors of our approach include Support Vector Machine (SVM) [11], kernel Ridge Linear Regression (kernelRLS) [10, 12, 18–20], and matrix factorization (MF) [22–24].

MF approaches decompose the interaction matrix that lives in the (protein, molecule) space into the product of matrices of lower rank, living in the two latent spaces of proteins and of molecules. The most recent and efficient MF based approach by [24] consider more specifically Logistic Matrix Factorization [25]. They display good performances and are also computationally efficient. [24] also generalized their approach to orphan molecules and proteins by computing latent representations of orphan molecules and proteins as a weighted sum of the latent representation of their neighbors.

BLM make prediction for a (protein, molecule) pair based first on the prediction of target proteins for the considered molecule, and then on the prediction of ligand molecules for the considered protein. The predictor used is the kernelRLS. This gives two independent predictions for each putative drug-target interaction, which are combined into a final prediction.

Finally, kernel methods using the Kronecker product of the molecule and protein space (presented in the next section) can handle orphan cases, but are more computationally expensive. Among them, although the *KronRLS* method (a Kernel Regularized Least Square classifier) succeeded to dramatically reduce the computational complexity of its exact solution when used on the Kronecker product of the molecule and protein space, and is hence applicable to large scale chemogenomics studies, SVM-based methods are still computationally inappropriate at such scale. The present study propose a SVM-based approach and aims at addressing this issue.

In most cases, previous studies have been implemented to predict interactions of molecules with proteins belonging to the same family, such as kinases or GPCRs [10, 18–24, 26, 27]. A few studies have been devoted to larger scales in the protein space, such as [17] which however does not focus on settings relevant to the prediction of drug specificity. Some rely on the 3D structure of the binding pocket [28, 29], which limits the number of proteins that can be considered, others on coarse protein descriptors based on the presence of structural or functional domains [30]. In the present paper, we propose a computationally efficient approach to study the applicability of these ML techniques to the entire druggable proteome.

1.3 Single-task and multi-task algorithms

In the context of the present paper, a *Single-Task* method consists in predicting protein targets for a given molecule m. In this setting, the specificity of m is studied by learning a model function $f_m(p)$ that predicts whether molecule m interacts with protein p, based on known protein targets for m. This means that a new model function is learned for each 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 protein p interacts with molecule m, based on all ligands known for protein p. We call this the *target-based ST* setting.

In contrast, *Multi-Task* methods predict whether p and m interact by training a model based on all known interactions, including those involving neither m nor p. In other words, the task of predicting ligands for protein p is solved not only based on the data available for this task (i.e. known ligands for this protein), but also based on the data available for the other

tasks (i.e. known ligands any other protein). The main issue is to define how the data available in all tasks can be used to make the predictions made for a given task.

More generally, the main idea behind multi-task learning is that, when solving several related tasks for which the data available for each task are scarce, a multi-task framework defines how to share information across tasks, which can improve the performance of the final prediction models that can be built.

In our case, the tasks are predictions of ligands for proteins whose sequences can be compared, which explains how the tasks are related. Such approaches are of particular interest when few interactions are known for a given ligand or protein, as is often the case when looking for secondary targets at the size of the human proteome. Even within the multi-task framework, orphan proteins (for which no ligand is known) and proteins for which the only known ligands are very dissimilar to the tested molecule, are those for which predictions are the most difficult. Therefore, our study will focus more particularly on these cases.

Various multi-task methods have been proposed in a number of applications that do not refer to bioinformatics [31–34]. Among them, multi-tasks methods have been proposed to solve the problem using kernel with Support Vector Methods (SVM, their basic principles are recalled in Supplementary Materials). Such multi-task kernel methods have widely been used in bioinformatics, including for chemogenomics applications [8]. Our contributions in the present paper belong to this category of methods that we briefly review in the next section.

1.4 Kernel methods for chemogenomics

In this study, we formalize and solve the problem of drug-target interaction prediction with Support Vector Machines (SVM), an algorithm for learning a classification or a regression rule from labeled examples [35].

Intuitively, SVMs seeks to find the optimal hyperplane separating two classes of data points. In addition to the choice of kernels, SVMs requires an important regularization parameter classically called C. This parameter controls the trade-off between maximizing the margin (i.e. the distance separating the hyperplane and the classes distributions) and classification errors on the training points. A deeper intuition on how SVM works and how this regularization parameters plays its role is given in Supplementary Materials S1. Additionally, as recalled in Supplementary Materials S1, although SVMs can be solved from vector representations of the data, they can also be solved using the "kernel trick", based only on the definition of a kernel function K which gives the similarity value $K(x, x')$ between all pairs of data points x and x', without needing an explicit representation of the data. Many kernels have been proposed for molecules and for proteins, and an overview of such kernels is presented in the Material and Methods section.

In chemogenomics, our goal can be viewed as finding the optimal hyperplane that separates the pairs (m, p) of molecules and proteins that interact from those that do not interact. This classification task can be solved using an SVM with a kernel K_{pair} defined on (ligand, protein) pairs. Given N example pairs, solving the SVM in the space of (m, p) pairs using the K_{pair} kernel corresponds to finding the optimal α_i coefficients such that (see Supplementary Materials S1):

$$\alpha^* = \underset{\alpha}{\operatorname{argmin}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K_{pair}((m_i, p_i), (m_j, p_j)) \tag{1a}$$

$$\text{subject to} \quad \alpha_i \geq 0, \forall i = 1, \ldots, N \tag{1b}$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{1c}$$

A general method to build a kernel on such pairs is to use the Kronecker product of the molecule and protein kernels [36]. Given a molecule kernel $K_{molecule}$ and a protein kernel $K_{protein}$, the Kronecker kernel K_{pair} is defined by:

$$K_{pair}((m,p),(m',p')) = K_{molecule}(m,m') \times K_{protein}(p,p') \tag{2}$$

Thus, the Kronecker kernel K_{pair} captures interactions between features of the molecule and features of the protein that govern their interactions (see Supplementary Materials S2 for an explicit definition of the Kronecker product of two matrices). If $K_{molecule}$ is a $n \times n$ matrix and $K_{protein}$ is a $p \times p$ matrix, their Kronecker product K_{pair} has size np #x00D7; np. In the context of chemogenomics, this can correspond to a very large size, leading to intractable computations. However, one interesting property of the Kronecker kernel is that calculating its values on a data set of (m, p) pairs does not require storing this entire matrix since it is sufficient to store $K_{molecule}(m, m')$ and $K_{protein}(p, p')$.

Therefore, solving the SVM (Eq 1) only requires calculation of the $K_{molecule}$ and $K_{protein}$ kernels according to Eq 2.

Once the α_i coefficients have been determined, the ability of a given (m, p) pair to interact is predicted based on:

$$f((m,p)) = \text{sign}\left(\sum_{i=1}^{np} \alpha_i y_i K_{molecule}(m,m_i).K_{protein}(p,p_i) + b\right) \tag{3}$$

This equation illustrates why the use of such a kernel can be viewed as a multi-task method. Indeed, in a single-task approach where one task corresponds to the prediction of ligands for a given protein p, the ability of molecule m to bind protein p would be estimated by:

$$f(m) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i y_i K_{molecule}(m,m_i) + b\right) \tag{4}$$

where the m_i molecules are ligand and non-ligand molecules known for protein p.

In the multi-task setting, $f((m, p))$ evaluates the ability of m to bind to protein p using m_i molecules that are all ligand or non-ligand molecules known for all proteins p_i. However, the contribution of the labels y_i of ligands for p_i proteins that are different from p to calculate $f((m, p))$ is weighted by $K_{protein}(p, p_i)$. In other words, the more similar two tasks (i.e. the corresponding proteins) are, the more known instances for one of the task will be taken into account to make predictions for the other task.

Such kernel-based multi-task approaches have been successfully applied to biological problems, including the prediction of protein-ligand interactions [8, 11, 37, 38].

1.5 Contribution

The goal of this paper is to investigate the application of multi-task Support Vector Machines (SVM) to the prediction of drug specificity, by predicting interactions between the drug of interest and the entire druggable proteome. To this end, we evaluate our multi-task SVMs based methods and state-of-the-art approaches in several key scenario that explore the impact of the similarity between the query (protein, molecule) pair and the training data on the

prediction performance, a point that is rarely discussed in the literature. We also explore their applicability to orphan settings, a situation often encountered large scale studies, and where single-task methods are not applicable.

We first 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 the *NN-MT* algorithm, 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 defined by pairs (p, m) with either $p = p^*$ or $m = m^*$; (2) a limited number of *extra-task* (protein, ligand) pairs, defined by pairs (p, m) with $p \neq p^*$ and $m \neq m^*$, chosen based on the similarity of p and m to p^* and m^*, respectively; 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 evaluate the performance on various assembled datasets in which the protein and/or the ligand are orphan.

Finally, compare the *NN-MT* algorithm to state-of-the-art approaches in drug-target interaction prediction [18–24]. We used and updated the PyDTI package [24], adding an implementation of *NN-MT* together with key cross-validation schemes and a DrugBank-based dataset built in the present study. In addition to all other experiments performed in the present study, this benchmark study concludes that *NN-MT* is a good default method providing state-of-the-art or better performances, in a wide range of prediction scenario that can be encountered in real-life studies: proteome-wide prediction, protein family prediction, test (protein, ligand) pairs dissimilar to pairs in the train set, and orphan cases.

All codes and all the datasets that were built are available at https://github.com/bplaye/efficient_MultiTask_SVM_for_chemogenomics/. The updated PyDTI package is available at https://github.com/bplaye/PyDTI/.

2 Materials and methods

2.1 Protein kernels

We used sequence-based kernels since they are suitable for proteome-wide approaches, unlike kernels relying on the 3D structure of the proteins or on binding pocket descriptions. Numerous studies have already been devoted defining descriptors of proteins based on amino-acid sequence [39–43]. We considered three sequence-based kernels: the *Profile kernel* [43], the *SWkernel*, and the *LAkernel*.

The *Profile kernel* uses as protein descriptors the set of all possible subsequences of amino acids of a fixed length k, and considers their position-dependent mutation probability. 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 Smith-Waterman (SW) alignment score between two proteins [44] 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*) [42] 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 [42]. 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 2.3).

In the last part of the study, we also considered kernels on proteins based on their family hierarchy. Indeed, the most important classes of drug targets have been organized into hierarchies established on the sequence and the function of the proteins within these families (GPCR [45], kinases [46] and ion channels [47]). As in [11], the hierarchy kernel is built based on the number of common ancestors shared by two proteins in the 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.

2.2 Small molecule kernels

Many descriptors have been proposed for molecules, based on physico-chemical and structural properties [48–51]. 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 [50], calculates the similarity between two molecules based on the infinite sets of random walks over their molecular graphs.

The second kernel, called the Tanimoto kernel, uses a description of molecules by vectors whose elements count the number of fragments of a given length. The similarity between molecules is based on the Tanimoto metric [48].

Kernel hyperparameters values. 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 2.3). The Tanimoto kernel has one hyperparameter: the length d of the paths, which we considered in {2, 4, 6, 8, 10, 12, 14}. All molecule kernels were centered and normalized.

2.3 Evaluation of prediction performance

Prediction performance is commonly evaluated with a cross-validation (CV) scheme [52]: 1) the dataset is randomly split into K folds 2) the model is run K times, each run using the union of (K-1) folds as the training set, and measuring the performance on the remaining fold. Prediction performance are averaged over all folds. When hyperparameters had to be selected, we used a nested cross validation (*Nested-CV*) scheme [53]. It consists in a (K-1) folds cross validation (*inner-CV*) nested in a K folds cross validation (*outer-CV*). At each step of the outer-CV, the inner-CV is repeated for all considered values of the hyperparameters. The values leading to the best prediction performance are retained as optimal. We used K = 5, a classical value in CV.

We also considered leave-one-out cross-validation (*LOO-CV*), for which the number of folds is the number of available points in the dataset. The *LOO-CV* scheme is particularly useful when the number of samples is small. It was used in the present paper when the size of the considered dataset was too small to perform *5-fold-CV*.

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] (ROC-AUC). The ROC curve plots true positive rate as a function of false positive rate, for all possible thresholds on the prediction score. Intuitively, the ROC-AUC score of a classifier represents the probability that if a positive and a negative interaction are each picked at random from the dataset, the positive one will have a higher positive score than the negative one. The second one is the area under the Precision-Recall curve [55] (AUPR). It indicates how far the prediction scores of true positive interactions are from false positive interactions, on average. 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 our SVM-based machine learning algorithms and we used the PyDTI package [24] for the other recent algorithms.

2.4 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 all the datasets used in the present study from the DrugBank database v4.3, because it contains FDA-approved drugs, or drug candidate molecules. This allowed optimize and test our models on drug-like molecules, on which they intend be applied. In addition, we assumed that the list of human proteins appearing as targets for molecules of DrugBank can represent a relevant "druggable" human proteome on which we could train models that predicting the specificity of drug-like molecules.

We built a first learning dataset called S, based on Version 4.3 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 3980 molecules targeting 1821 proteins, and including 9536 protein-ligand interactions that correspond to the positive training 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 as negative examples, allowing the predictor to re-classify some of these pairs as positive examples.

We built several other datasets using exactly the same training pairs as those in S, but 5-folded in various ways. Datasets S_1, S_2, S_3, and S_4 are folded so as to correspond to random, orphan protein, orphan ligand, and double orphan prediction situations. The construction of these four datasets is detailed in Section 3.2, where they are used. Datasets S'_1, S'_2, S'_3, and S'_4 are also folded to mimic the same situations, but with the additional constraint that proteins and ligands were clustered based on their similarities, and each fold contains only one cluster of proteins and of ligands. The goal is to test the performance of the method in situations similar to those of S_1, S_2, S_3, and S_4, but with the added difficulty that the test set (one fold) and the train set (the 4 other folds) contain pairs that have low similarities. This setting is relevant

Table 1. Dataset statistics.

	S	S_0	GPCR	IC	Kinases
number of interactions	9536	5908	1735	1603	847
number of proteins	1821	788	85	140	143
number of molecules	3980	1180	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

https://doi.org/10.1371/journal.pone.0204999.t001

when considering proteome-wide predictions: many of the proteins to consider may not have close neighbors among the proteins for which the most information (i.e. ligands) are known. The construction of these four datasets is detailed in Section 3.3, where they are used.

We also built a dataset called S_0 by keeping only molecules and proteins in *S* which are involved at least in two interactions, in order to compare the prediction performance of the proposed methods with those of ligand-based and target-based approaches. Indeed, these two single-task approaches require at least two data points, one used as train, and one as test. Consequently, when a *LOO-CV* scheme is used, no ligand and no protein are orphans in S_0. S_0 contains 5908 positive interactions and was used in Sections 3.4 and 3.5. In addition, we randomly generated four sets of 5908 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. Then, we assessed performance by computing the mean and standard deviation of the AUPR scores over test sets including the positive interactions set and one of the negative interactions sets.

Finally, we built three protein family datasets by extracting from S_0 all protein-ligand interactions involving respectively only G-Protein Coupled Receptors (GPCR set), ion channels (IC set), and kinases (Kinases set). These datasets were used to evaluate performance of our method within a family of proteins, and compare it to those of single-task approaches. They were extracted from S_0 (and not from the larger dataset *S*) since again, these comparisons used the *LOO-CV* scheme, which requires at least two data points per protein and per molecule.

Table 1 gives some statistics about the datasets, including the distribution of targets per drug and the distribution of ligands per protein.

3 Results

3.1 Kernel selection and parametrization

The goal of this section is to choose 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 proteins such as dataset *S* would be good default kernels for the prediction of drug candidates specificity. Therefore, we optimized kernels on dataset *S* (the largest dataset built in the present study), and used the best-performing couple of kernels in the remainder of the paper.

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

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

Table 2. Best *nested 5-fold-CV* AUPR for each kernel combination, together with optimal hyperparameters.

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

https://doi.org/10.1371/journal.pone.0204999.t002

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 performance. Therefore, in what follows, we only consider these two kernels, and call *MT* the Multi-Task SVM that uses their Kronecker product.

We also considered one-class SVM using the same kernels [61]. However, the performance of one-class SVM were clearly lower than those of KronSVM. The AUPR scores of one-class SVM were in the range of 0.6 for all considered kernels when those of KronSVM were in the range of 0.9. Therefore, we did not further consider one-class SVM.

It is worth noting that the SW-kernel gave the worst performance, although it is used in many studies [10, 15, 18, 20]. Overall, the good performance of the six multi-task methods observed on *S* is consistent with previously reported results [17, 20]. 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 in real-case prediction of drug specificity is expected to be lower than that obtained on *S*, since at the proteome scale, some of the test (protein, ligand) pairs will be far from all pairs of the train set. This will be particularly true in the case of new drugs and therapeutic targets, as already pointed by [26].

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. Therefore, in the following, we study the evolution of *MT*'s performance in more realistic settings where the protein, the molecule, or both, are orphan, or where the tested (protein, ligand) pair has low similarity with the pairs belonging to the train set. Before pursuing, we bring to the reader's attention that, for all experiments that follow, the numerical values of the performance measures and their standard deviations can be found in the Supporting Information.

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 contain proteins and/or molecules that are *not* in the training set, as proposed by [26]. For that purpose, all the pairs of dataset *S* were used and 5-folded as follows in order to create four cross-validation data sets:

- S_1: randomly and balanced in positive and negative pairs;

- S_2 (corresponding to the "orphan ligand" case): (protein, molecule) pairs in one fold only contain molecules that are absent from all other folds; prediction on each test set (each fold) is performed using train sets (the four other folds) in which no the ligands of the test set are absent.
- S_3 (corresponding to the "orphan protein" case): (protein, molecule) pairs in one fold only contain proteins that are absent from all other folds; prediction on each test set is performed using train sets in which no the proteins of the test set are absent.
- S_4 (corresponding to the "double orphan" case): (protein, molecule) pairs in one fold only contain proteins *and* molecules that are both absent from all other folds. Prediction on each test set is performed using train sets in which no the proteins and the ligands of the test set are absent. The folds of S_4 were built by intersecting those of S_2 and S_3 and S_4. Thus, S_4 contains 25 folds and not 5.

Fig 1 shows the nested-CV AUC and AUPR scores obtained by the *MT* method on the S_1-S_4 datasets. As expected, the best scores are obtained for S_1, and the worst for S_4, since in S_4, no pairs of the train set contain the protein or the ligand of the tested pair to guide the predictions. 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 dataset 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 made on less diverse datasets [26]. It is important to point that single-task approaches would not be able to provide any prediction on the S_4 dataset.

The scores obtained in the S_2 and S_3 datasets are intermediate between those observed on S_1 and S_4. This was to be expected, as in these datasets, the algorithm can rely on training pairs

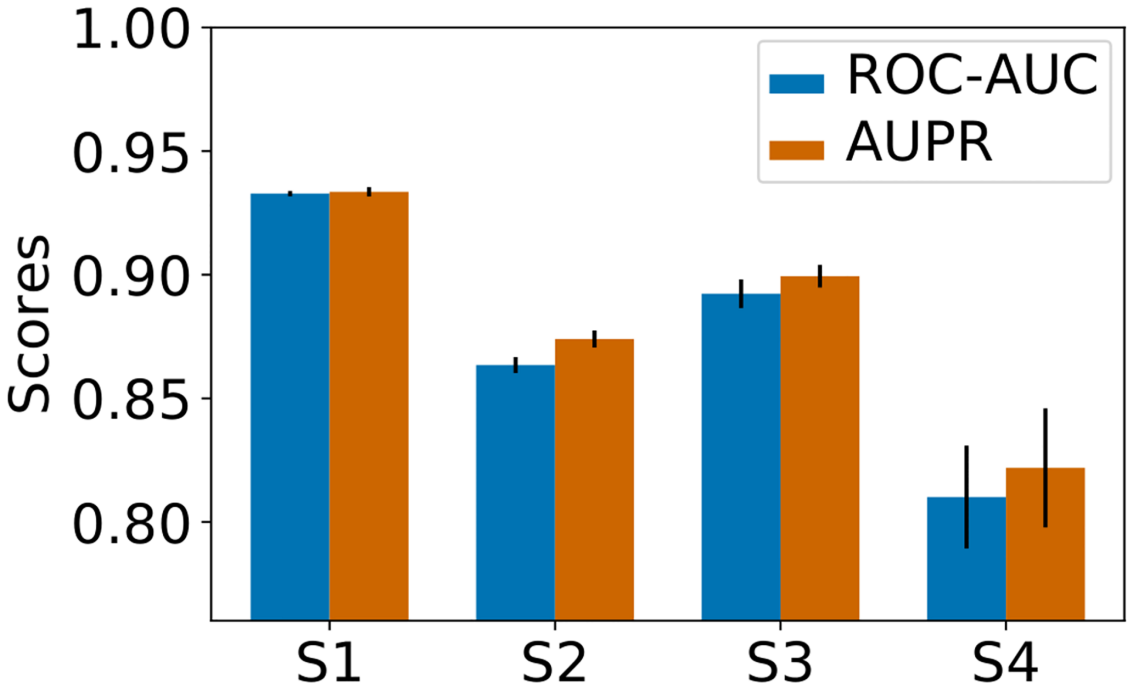

Fig 1. *Nested 5-fold-CV* performance of the *MT* method on the S_1 – S_4 datasets. Numerical values can be found in Supporting Information S1 Table.

https://doi.org/10.1371/journal.pone.0204999.g001

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 predicting ligand for new protein targets is easier than predicting targets for new compounds, as already noticed in [26]. We also observed similar behaviors when replacing the SVM with a kernel ridge regression (see Supporting Information S1 Fig) and hence did not further consider this algorithm.

Overall, 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 prediction performance of this multi-task algorithm.

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* following the "clustered cross-validation" approach [62]. More precisely, we clustered proteins (resp. ligands) into 5 clusters by hierarchical clustering [63]. We then built four cross-validation datasets, $S'_1 - S'_4$, generated based on folds 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. Therefore, test pairs are more dissimilar from train pairs than in the $S_1 - S_4$ datasets, which makes the problem more difficult.

Overall, all the pairs of dataset *S* were 5-folded as follows in order to create four cross-validation data sets:

- S'_1: randomly and balanced in positive and negative pairs, each fold containing proteins and ligands belonging to the same cluster;
- S'_2 (corresponding to the "orphan ligand" case): (protein, molecule) pairs in one fold only contain molecules that are absent from all other folds; prediction on each test set (each fold) is performed using train sets (the four other folds) in which no the ligands of the test set are absent, with the additional constraint that each fold contains proteins and ligands belonging to the same cluster.
- S'_3 (corresponding to the "orphan protein" case): (protein, molecule) pairs in one fold only contain proteins that are absent from all other folds; prediction on each test set is performed using train sets in which no the proteins of the test set are absent, with the additional constraint that each fold contains proteins and ligands belonging to the same cluster.
- S'_4 (corresponding to the "double orphan" case): (protein, molecule) pairs in one fold only contain proteins *and* molecules that are both absent from all other folds. Prediction on each test set is performed using train sets in which no the proteins and the ligands of the test set are absent, with the additional constraint that each fold contains proteins and ligands belonging to the same cluster. The folds of S_4 were built by intersecting those of S_2 and S_3 and S_4. Thus, S_4 contains 25 folds and not 5.

We used the same kernels as for the *MT* method. Fig 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 $S_1 - S_4$ datasets. This suggests that good performance on a query pair (p^*, m^*) is driven by the presence in the training set of pairs made *both* of proteins similar to p^* and of molecules similar to m^*, even if the query pair (p^*, m^*) is a double orphan, as in S_4. Our results also suggest that it is more important to train on pairs similar to the double orphan query pair (p^*, m^*), as in S_4, than on data

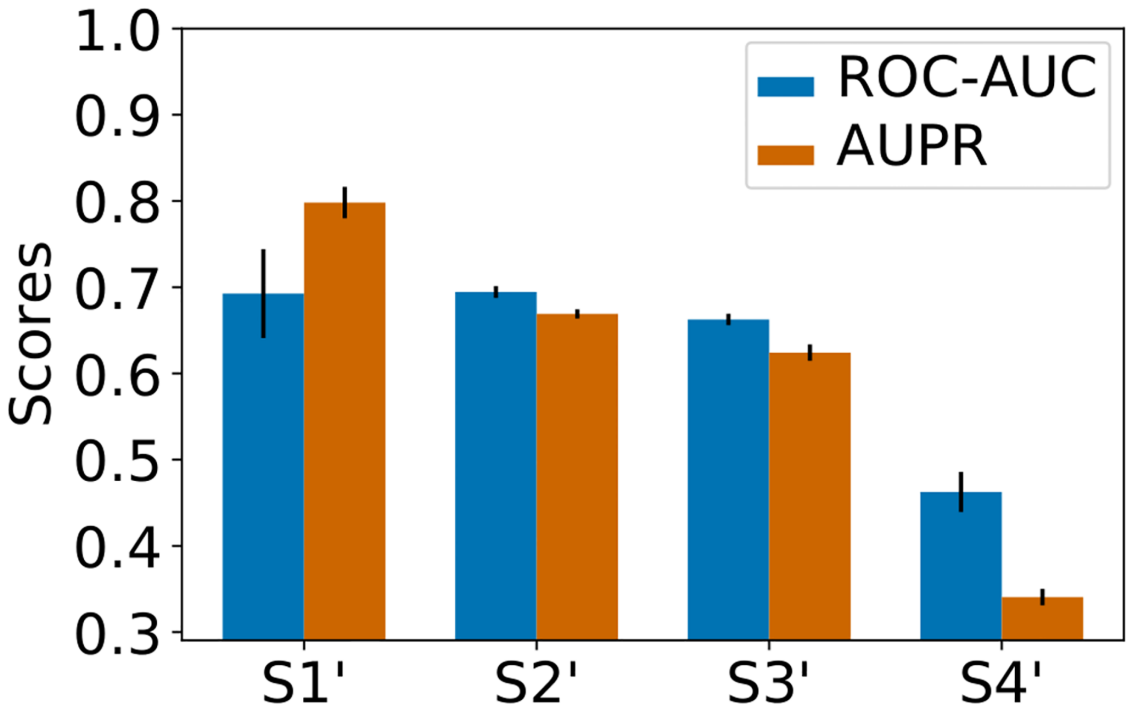

Fig 2. *Nested 5-fold-CV* performance of the *MT* method on the $S'_1 - S'_4$ datasets. Numerical values can be found in Supporting Information S2 Table.

https://doi.org/10.1371/journal.pone.0204999.g002

containing, for example, p^* itself, but paired only with molecules quite dissimilar to m^*, as in S'_2.

In the most difficult case of S'_4, the performance is even lower than that of a random predictor, which would display an AUC of 0.5. This is somewhat intriguing. One explanation could be that, when the data points are randomly dispatched in the folds used to build the train and test sets, this value is of 0.5 can be considered as a baseline. In the case of S'4, the folds are built using clustered protein-ligand pairs, so that the data distribution in the test set (one of the clusters/folds) may be different from the data distribution in the train set (the remaining samples). This explains why a machine-learning algorithm like *MT* is unable to learn a model that is relevant for the test set. In other words, the baseline performance expected when systematically learning on differently distributed data might be worse than that observed by a random predictor.

These results suggest that pairs in the training set that are very dissimilar to the query pair do not help making more accurate predictions. In other words, although the kernels used in multi-task approaches modulates how information available in one task is shared for training other tasks (the further the tasks are, the less information is shared), using information from distant tasks seems to degrade performance. This insight is interesting since the *MT* algorithm requires calculating the Kronecker kernel on all (protein, molecule) pairs, which is computationally demanding. Therefore, the next two sections evaluate whether removing distant pairs from the training set can improve computational efficiency without degrading performance.

3.4 Multi-task approaches on reduced training sets

Based on the insight that *MT* prediction is driven by training examples that are close to the query data, we propose to build training tests of reduced sizes by removing training examples

distant from the query. The goal of this section is also 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, since there would be no point in using the more complex *MT* method if a single-task method performs better. 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, n^+ (resp. n^-) will refer to the number of positive (resp. negative) examples in the train set.

In all the following experiments, we used the *LOO-CV* scheme because intra-task and extra-task pairs can only be defined for each pair separately, which prevents from using *K-fold-CV* schemes. 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-CV* scheme did not trigger a bias, as sometimes observed [26] (see Supporting Information S2 Fig and S3 Table).

Because prediction of a given (protein, ligand) interaction can only be made by single-task if the pair partners are present in at least another pair of the train set, in the following experiments, we used the S_0 dataset in which all ligands and all proteins are involved in at least two known interactions, as explained in Section 2.4.

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) to those of single-task methods. Since *ligand-based ST* can only use intra-task positive examples, the only positive training pairs we use for the *MT* method are the intra-task pairs as well. Note that *MT* still gets more training examples than *ligand-based ST*, since pairs formed with the query protein and a different ligand are also included. By reducing the training set size, the computational times required by the *MT* method are now 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 $100 \times n^+$.

Fig 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 in the train set pairs displaying low similarity to the tested pair degrades both the computational time and the quality of the prediction of *MT*.

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*
- 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.

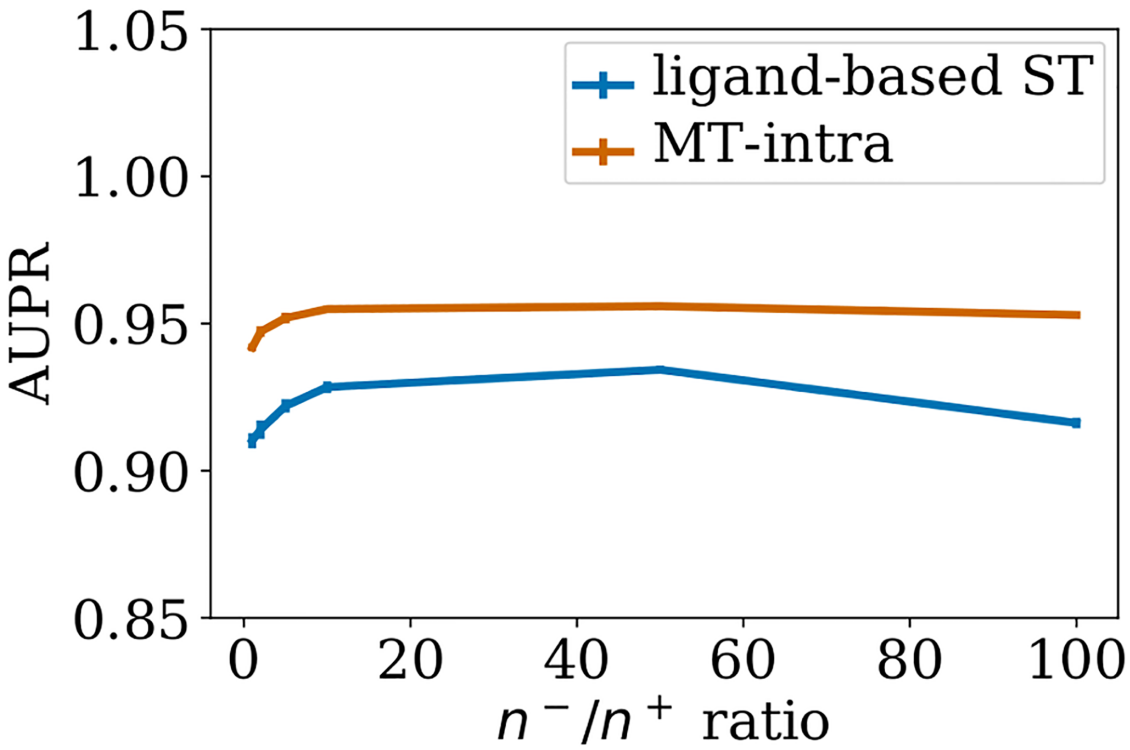

Fig 3. Performance of single-task and *MT-intra* as a function of the n^-/n^+ ratio. Numerical values can be found in Supporting Information S4 Table.

https://doi.org/10.1371/journal.pone.0204999.g003

We call *NN-MT* (for Nearest Neighbor *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* (for Random Neighbor *MT*).

We report the *LOO-CV* performance of *NN-MT* and *RN-MT* on Fig 4. Fig 4(a) 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. Fig 4(b) 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

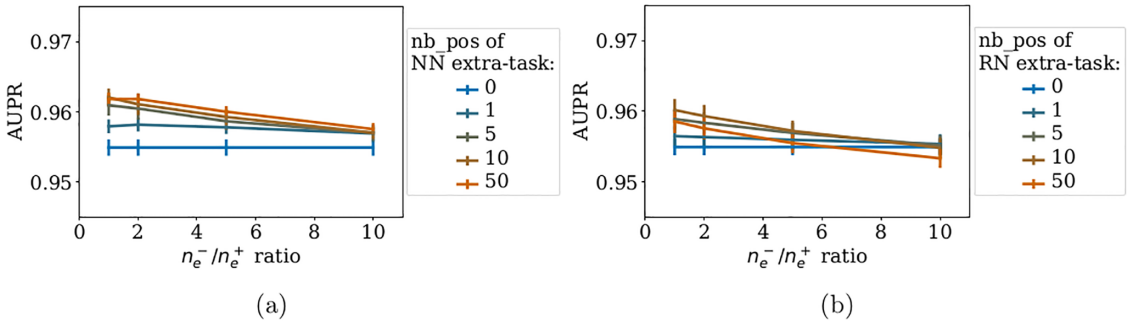

Fig 4. AUPR as a function of the n_e^-/n_e^+ ratio for increasing numbers random extra-task points in the train set. (a): *NN-MT*. (b): *RN-MT*. The blue horizontal line corresponds to *MT-intra* (which is trained only on intra-task pairs). Numerical values can be found in Supporting Information S5 and S6 Tables.

https://doi.org/10.1371/journal.pone.0204999.g004

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.

Taken together, these results show that *NN-MT* outperforms not only *MT-intra*, but also the more computationally demanding *MT* method trained in the *LOO-CV* setting in Section 3.2. AUPR scores for (protein, ligand) pairs involving non-orphan ligands and non-orphan proteins are 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 *MT-intra* and *NN-MT* proposed methods, and to compare them to those of the *ligand-based ST* method, 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 each test pair (p^*, m^*), the training set only included 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, since, unlike the positive pairs, they are available in large numbers and at all distances from the tested pairs.

Fig 5 reports the *LOO-CV* AUPR scores of *ligand-based ST* and *MT-intra* for varying values of θ (20^{th}, 30^{th}, 50^{th}, and 80^{th} percentiles) and of the n^-/n^+ ratio (from 1 to 50).

Fig 5(a) 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, see Section 3.4.1). Fig 5(a) also suggests that a n^-/n^+ ratio of 10 is again an appropriate choice, as when all intra-task positive example are used (see Section 3.4.1). We therefore set n^-/n^+ to this value for the remaining experiments of this section.

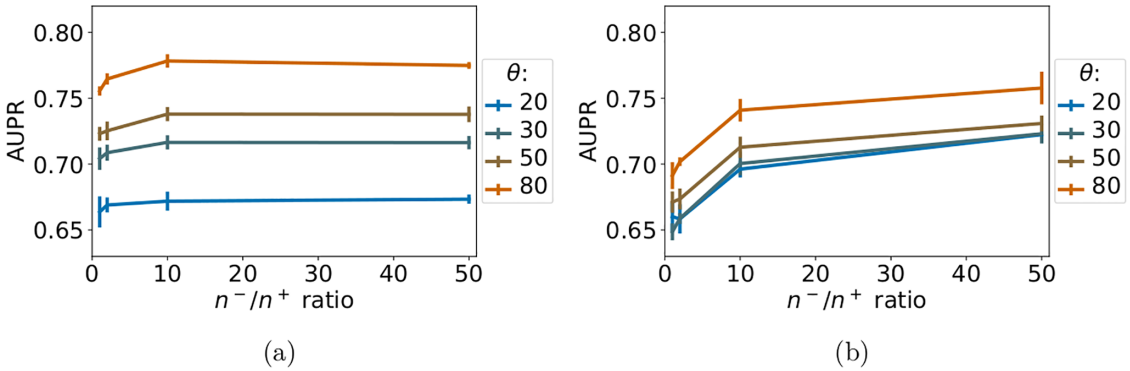

Fig 5. AUPR scores as a function of the n^-/n^+ ratio, for percentile-based threshold θ ranging from 20% to 80%. (a): *MT-intra* method. (b): *ligand-based ST* method. Numerical values can be found in Supporting Information, respectively S7 and S8 Tables.

https://doi.org/10.1371/journal.pone.0204999.g005

Fig 5(b) 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 above those 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.

Applying the same percentile-based similarity constraint to the intra-task positive pairs, we compared the performance of *NN-MT* and *RN-MT* when respectively adding n_e^+ nearest neighbors or random extra-task positive to 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 scarce or far from the tested pair, as illustrated by the poor performance of *ligand-based ST* in the previous section. A number n_e^- of nearest neighbors (respectively random) negative extra-task pairs were also added for *NN-MT* (respectively *RN-MT*).

Fig 6(a) and 6(b) report the *LOO-CV* AUPR of *NN-MT*, as a function of n_e^-/n_e^+ (ratio of negative over positive extra-task pairs) and for a number of extra-task positive pairs varying from 0 to 50, respectively for percentile similarity constraints θ of 20 and 80. The blue horizontal line (for $n_e^+ = 0$) corresponds to the performance of the *MT-intra* methods. Fig 6(a) and 6(b) show that adding extra-task pairs to the train set dramatically improves performance. The

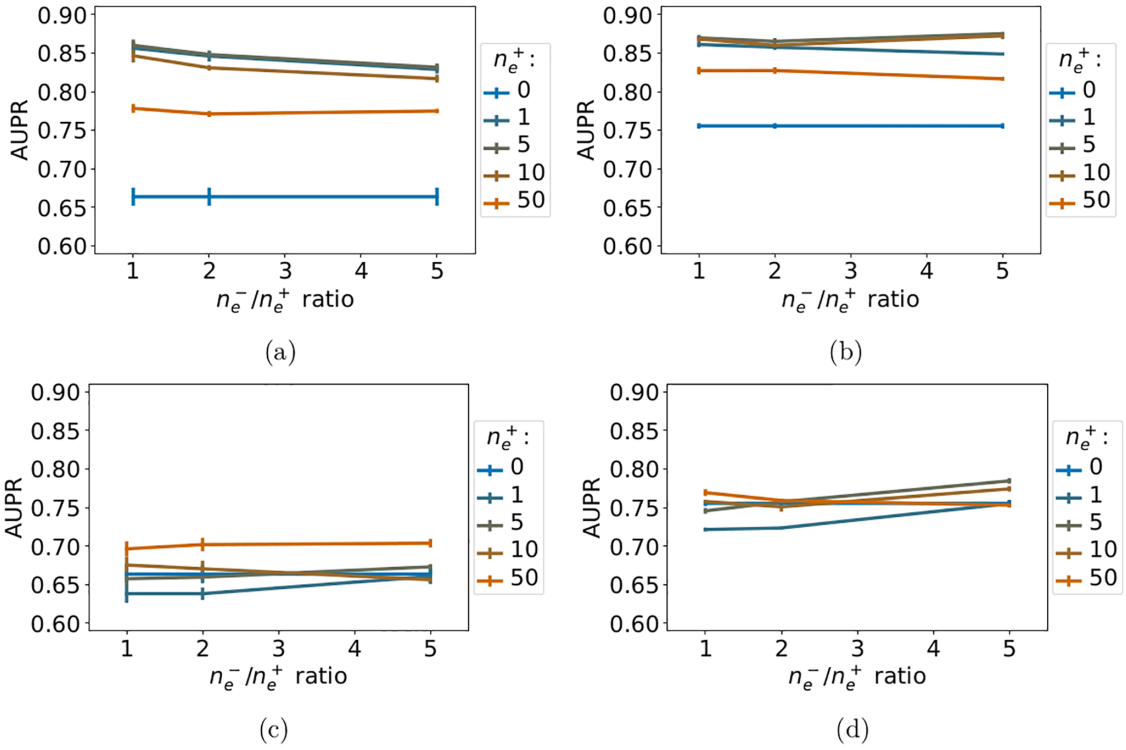

Fig 6. AUPR score of *NN-MT* and *RN-MT* as a function of the n_e^-/n_e^+ ratio, for a number of extra-task positive pairs n_e^+ varying from 0 to 50, and for percentile-based similarity threshold θ of 20 and 80 applied to the intra-task positive pairs. (a): *NN-MT*, $\theta = 0.20$. (b): *NN-MT*, $\theta = 0.80$. (c): *RN-MT*, $\theta = 0.20$. (d): *RN-MT*, $\theta = 0.80$. Numerical values can be found in Supporting Information, respectively S9–S12 Tables.

https://doi.org/10.1371/journal.pone.0204999.g006

AUPR score reaches values above 0.85, independently of θ, suggesting that when no close intra-task pairs are available, performance is driven mainly by extra-task training pairs, confirming our observations in Section 3.3. Moreover, when the number of extra-task pairs increases, the performance of *NN-MT* increases, then tends towards that of *RN-MT*, and then degrades at larger values of n_e^+ because 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. Adding the same number of negative extra-task pairs ($n_e^-/n_e^+ = 1$) provides the best AUPR, which again limits the size of the required training set. Unsurprisingly, the best AUPR in the absence of the closest 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 percentile, 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, Fig 6(c) and 6(d) show that the performance does not improve when the extra-task training pairs are chosen at random, and therefore, are on average further from the test pair. It might even degrade when the number of extra-task pairs becomes large. Finally, Fig 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 the best prediction performance are obtained using the *NN-MT* method trained with 10 times more negative intra-task pairs than positive ones, 1 to 5 extra-task nearest neighbor 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.

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

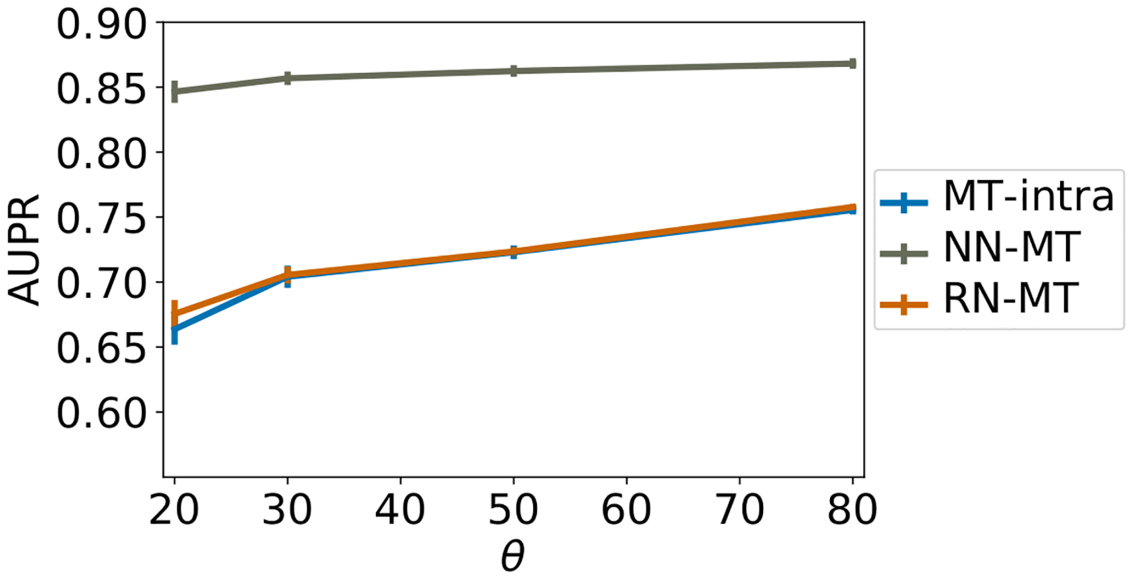

Fig 7. AUPR score as a function of percentile-based similarity θ, for $n^-/n^+ = 10$, a number of extra-task positive pairs $n_e^+ = 10$ and a ratio of $n_e^-/n_e^+ = 1$ for extra-task pairs. Numerical values can be found in Supporting Information S13 Table.

https://doi.org/10.1371/journal.pone.0204999.g007

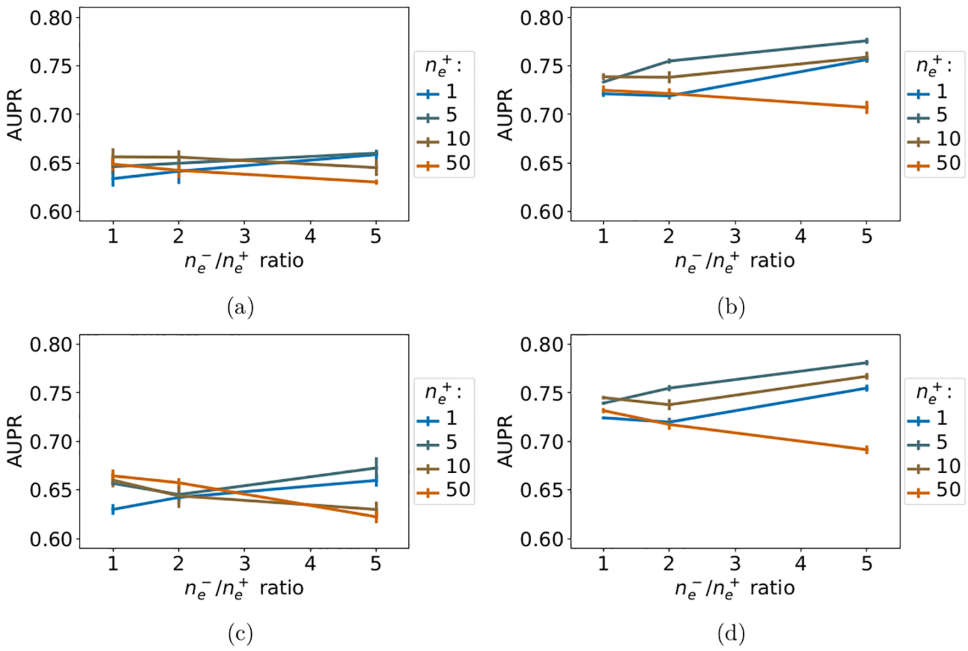

Fig 8. AUPR scores of the *NN-MT* and *RN-MT* multi-task methods as a function of the n_e^-/n_e^+ ratio, for a number of extra-task positive pairs n_e^+ varying from 1 to 50. (a): *NN-MT*, $\theta = 0.20$. (b): *NN-MT*, $\theta = 0.80$. (c): *RN-MT*, $\theta = 0.20$. (d): *RN-MT*, $\theta = 0.80$. The two methods are trained with intra-task and extra-task examples that are both dissimilar to the tested pair (percentile-based similarity thresholds θ of 20 and 80). Exact values can be found in Supporting Information respectively in S14–S17 Tables.

https://doi.org/10.1371/journal.pone.0204999.g008

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-CV* AUPR on Fig 8.

We observe that the performance of *NN-MT* remains overall 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 neither close intra-task nor close extra-task positive pairs are available, no method can provide performance good enough for the purpose of drug sensitivity prediction. These interactions would have to be experimentally tested if they are critical in the context of a drug's development program. These observations were expected given that adding random extra-task training pairs, possibly far from the tested pair, did not improve performance of the *MT-intra* method (see Section 3.5.2).

Taken together, our results show that the proposed *NN-MT* method 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, in a context where *ligand-based ST* could not make any prediction.

The results we have presented so far address the issue of using kernel methods with SVM in the context of proteome-wide specificity prediction, at a tractable computational cost thanks

to the choice of a reduced learning dataset, without loss in prediction performance. Reducing datasets to the most informative data points is also the underlying idea of active-learning methods [64]. In the latter, the goal is to guide step by step which data points are needed (i.e. have to be observed and labeled) to best improve the prediction performance. These methods are used when acquiring data is the limiting factor. By essence, their goal is also to reduce the size of the data sets that are used.

However, another key issue corresponds to study the specificity of a molecule within a family of related proteins. Indeed, when a new drug candidate is identified against a given therapeutic target, proteins belonging to the same family are important off-target candidates. This corresponds to the setting where similar training pairs are available, since proteins of the same family are similar in terms of sequence.

In the next section, we therefore assess whether the proposed *NN-MT* method, initially dedicated and tuned in proteome-wide prediction problems, also 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. All the (protein, molecule) pairs involving GPCRs, ICs, or kinases that were present in the dataset *S* described in Section 2.4 were 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 2.1), because a sequence-based kernel may not be optimal to study the specificity of the molecule within a family of proteins [11, 27]. The corresponding methods are called *MT-intra-family*, *NN-MT-family*, and *RN-MT-family*.

As in the above section, each (protein, ligand) test pair is considered in turn in a *LOO-CV* scheme. We used a learning dataset containing: all positive intra-task positive pairs, ten times more random negative intra-task pairs (this value was found adequate in previous sections), a varying number of positive extra-task pairs (nearest neighbors for *NN-MT* or *NN-MT-family*, random 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. Fig 9 shows 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 remains above those reached when adding positive random extra-task pairs, as observed in the larger scale studies presented above. Overall, 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.

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 of *NN-MT*, one should note that the family GPCR kernel is based on a GPCR hierarchy that was

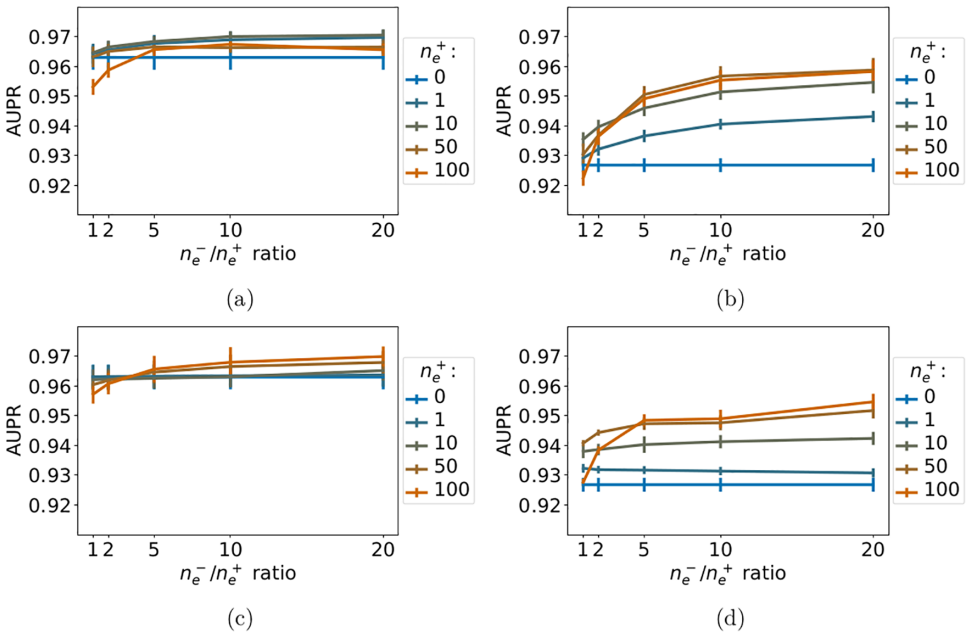

Fig 9. AUPR score of the considered multi-task methods on the GPCR family as a function of the n_e^-/n_e^+ ratio, for a varying number n_e^+ of extra-task positive pairs. (a): *NN-MT-family* (family hierarchy kernel). (b): *NN-MT* (sequence kernel). (c): *RN-MT-family* (family hierarchy kernel). (d): *RN-MT* (sequence kernel). The blue horizontal line corresponds to the *MT-intra* method trained only on intra-task pairs. Numerical values can be found in Supporting Information, respectively S18–S21 Tables.

https://doi.org/10.1371/journal.pone.0204999.g009

established using known GPCR ligands. Therefore, the results obtained by the *NN-MT-family* might be biased, which is not the case for those obtained by the *NN-MT*.

3.6.2 Ion Channel family. The conclusions obtained above in the GPCR family also hold in the IC family, as shown in Fig 10. Again, all methods perform very well, reaching AUPR scores above 0.97. As for the GPCR family, 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.

3.6.3 Kinase family. In the kinase family, the results are somewhat different from those obtained on 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 Fig 11(b) and 11(d). 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 hierarchy 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 Fig 11(a) and 11(c). In addition, their best AUPR scores reaches 0.90, which is lower than those of the *NN-MT* and *RN-MT* methods which are in the range of 0.93. These observations may reflect the fact that the kinase family gathers proteins that are relatively more diverse than GPCRs and IC. For example, one can distinguish Tyrosine kinases and Serine/Threonine kinases, or globular protein kinases and receptor protein kinases. This diversity is illustrated by the organization of the kinome in some 50 distinct sub-families [46]. In this context, the sequence kernel that was optimized in proteome-wide studies might better capture the degree of similarity between two kinases than the hierarchy kernel does.

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

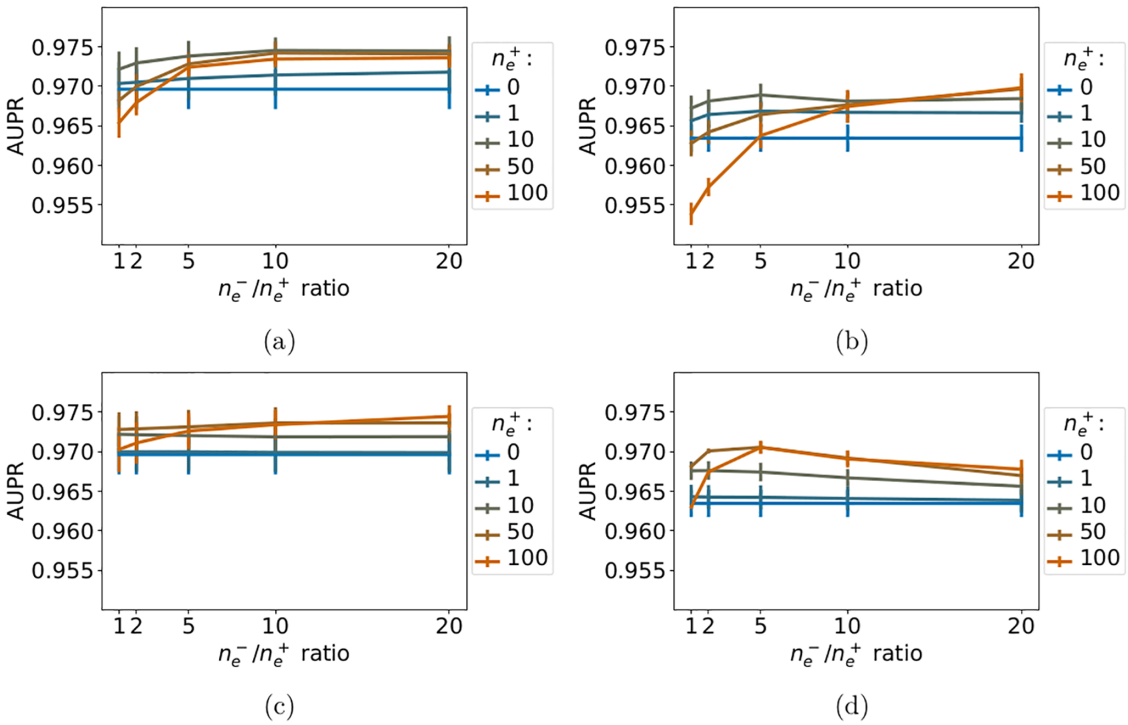

Fig 10. AUPR of the multi-task methods on the IC family. (a): *NN-MT-family* (family hierarchy kernel). (b): *NN-MT* (sequence kernel). (c): *RN-MT-family* (family hierarchy kernel). (d): *RN-MT* (sequence kernel). The blue horizontal line corresponds to the *MT-intra* method trained only on intra-task pairs. Numerical values can be found in Supporting Information, respectively S22–S25 Tables.

https://doi.org/10.1371/journal.pone.0204999.g010

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 [11, 20], they remain in the same order of magnitude.

4 Discussion: Comparison to other methods

As mentioned in Section 1.2, a few methods have been proposed to predict interactions between proteins and ligands. We compared the prediction performances of the proposed *NN-MT* method to those of two state-of-the art methods: a recent Matrix Factorization method called Neighborhood Regularized Logistic Matrix Factorization (*NRLMF*) [24], and the Kronecker (kernel) Regularized Least Square regression method *KronRLS* (a kernel-based method, as *NN-MT*) [18, 19].

The *KronRLS* and *NRLMF* methods were published based on their prediction performance on four protein family datasets, Nuclear Receptors (NR), GPCR, Ion channels (IC), and Enzymes (E) that contained respectively 90, 636, 1476 and 2926 interactions [10].

The *KronRLS* method uses a kernel $K_{molecule}$ for molecules that is defined by:

$$K_{molecule} = \frac{1}{2}(K_{\text{SIMCOMP}} + K_{\text{GIP},m}) \tag{5}$$

where K_{SIMCOMP} is a structure similarity kernel [65], and where $K_{\text{GIP},m}$ is a Gaussian kernel that compares the interaction profiles of molecules against the proteins of the dataset [18]. For

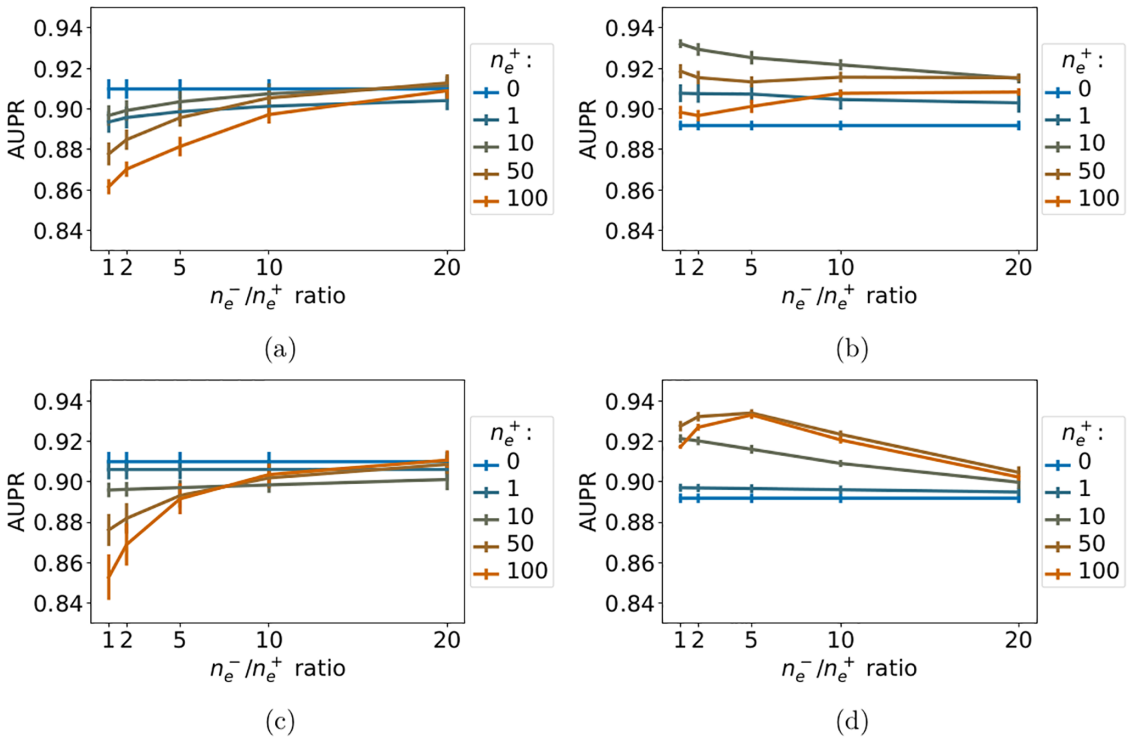

Fig 11. AUPR score of the multi-task methods within the kinase family. (a): *NN-MT-family* (family hierarchy kernel). (b): *NN-MT* (sequence kernel). (c): *RN-MT-family* (family hierarchy kernel). (d): *RN-MT* (sequence kernel). The blue horizontal line corresponds to the *MT-intra* method trained only on intra-task pairs. Numerical values can be found in Supporting Information, respectively S26–S29 Tables.

https://doi.org/10.1371/journal.pone.0204999.g011

proteins, the kernel $K_{protein}$ is defined by:

$$K_{protein} = \frac{1}{2}(K_{\text{sequence}} + K_{\text{GIP},p}) \tag{6}$$

where K_{sequence} is a protein sequence similarity kernel also based on the Smith-Waterman score [44], and $K_{\text{GIP},p}$ is a Gaussian kernel that compares the interaction profile of proteins against the molecules of the dataset.

A specific feature of the *NRLMF* method is that it integrates a neighborhood regularized method which allows to take into account only the K nearest neighbors to predict a given (protein, ligand) interaction (in practice, the authors used K = 5).

We performed benchmark experiments on these family datasets using the PyDTI package. This package initially contained the *KronRLS*, a variant of it called *KronRLS-WNN*, and the *NRLMF* methods, and the kernel matrices $K_{protein}$ and $K_{molecule}$ calculated for the four family datasets. In all experiments, we compare the intrinsic performances of the algorithms: the similarity measures used are the same for all methods. More precisely, the three methods used the kernels available in PyDTI: the structure-based K_{SIMCOMP} for molecules, and a kernel K_{sequence} based on the Smith-Waterman score. In addition, *KronRLS* also used the $K_{\text{GIP},m}$ kernel, leading to the $K_{molecule}$ kernel for molecules defined in Eq 5, and the $K_{\text{GIP},p}$ kernel, and to the $K_{protein}$ kernel for proteins defined in Eq 6. The two other methods do not use the K_{GIP} kernels because they do not take into account information about interaction profiles.

We also performed benchmark experiments on a dataset gathering more diverse protein and ligands. To this end, we used the DrugBank-based dataset S_0 described in Section 2.4)

Table 3. AUPR scores and standard deviations in *10-fold-CV*, test sets balanced in positive and randomly chosen negative samples.

Method / Dataset	NR	GPCR	IC	E
KronRLS	0.75 ± 0.14	0.9 ± 0.03	0.96 ± 0.01	0.96 ± 0.01
NN-MT	0.89 ± 0.09	0.95 ± 0.02	0.97 ± 0.01	0.97 ± 0.01
NRLMF	**0.96 ± 0.04**	**0.96 ± 0.02**	**0.98 ± 0.01**	**0.98 ± 0.0**

https://doi.org/10.1371/journal.pone.0204999.t003

containing 5 908 interactions. Because *KronRLS* and *NRLMF* could not make predictions on S_0 at a manageable computational cost in the *LOO-CV* scheme, we randomly sampled 2 000 of the 5 908 interactions of this data set to create a smaller test data set called $S_{0,2000}$. We still used all of S_0 (minus the test example) for training.

We calculated the Tanimoto and Profile kernels optimized in the present study (see Section 3.1), and these matrices were uploaded in PyDTI so that the three considered methods could used them. In addition, since *KronRLS* also use the $K_{GIP,m}$ and the $K_{GIP,p}$ kernels, we described all molecules and proteins in S_0 by their interaction profile. We calculated the $K_{GIP,m}$ and $K_{GIP,p}$ kernels on S_0 and uploaded these kernels in PyDTI. Only *KronRLS* used these additional kernels. All cross-validation experiments were performed building test sets from $S_{0,2000}$ and using all remaining data points in S_0 for training.

Table 3 presents the performance of the three considered methods on the protein family datasets.

Globally, the performance of all methods are high and close, with AUPR scores above 0.9 in most of the cases. On average, the *NRLMF* and *NN-MT* methods are on par and lead to the best results. These results are consistent with those reported in [24].

Table 4 confirms the tendencies observed in Table 3. Although the performances are slightly lower on this more diverse $S_{0,2000}$ dataset than on the family datasets, they remain high, with *NRLMF* and *NN-MT* keeping the best AUPR scores.

As discussed in Sections 3.1 and 3.2, various prediction methods lead to such high performances because, in the protein family or $S_{0,2000}$ datasets, predictions are averaged over test pairs in which the protein and/or the molecule might be orphan, or not. These averaged results hide less favorable situations, typically double orphan samples. Because these cases are common and important when predicting specificity of a new drug candidate at the proteome scale, we would like to stress that comparing methods in orphan cases is a more stringent and relevant test. In such cases, the performance are expected to be more modest and the methods might not rank in the same order.

Therefore, we ran the three methods using a *LOO-CV* scheme on double orphan (protein, molecule) pairs on the same datasets. We considered the *LOO-CV* scheme as it is particularly convenient to test double orphan drug-target interactions. In these experiments, for each tested (p, m) pair, interactions involving the considered protein or the molecule are ignored in the train set. The *LOO-CV* schemes were balanced in positive and randomly chosen negative pairs.

Table 4. AUPR scores and standard deviations in *10-fold-CV*, test sets balanced in positive and randomly chosen negative samples.

Method / Dataset	$S_{0,2000}$
KronRLS	0.91 ± 0.02
NN-MT	0.95 ± 0.01
NRLMF	**0.96 ± 0.01**

https://doi.org/10.1371/journal.pone.0204999.t004

Table 5. AUPR scores and standard deviations on double orphan *LOO-CV*, balanced number of positive and randomly chosen negative test samples.

Method / Dataset	NR	GPCR	IC	E
KronRLS-WNN	0.78 ± 0.03	0.83 ± 0.01	0.78 ± 0.01	0.84 ± 0.0
KronRLS	0.55 ± 0.01	0.62 ± 0.01	0.64 ± 0.0	0.56 ± 0.0
NRLMF	0.19 ± 0.03	0.15 ± 0.0	0.26 ± 0.01	0.23 ± 0.0
NN-MT	0.72 ± 0.04	0.76 ± 0.01	0.72 ± 0.01	0.66 ± 0.0
NN-MT-WNN	**0.77 ± 0.05**	**0.85 ± 0.0**	**0.79 ± 0.01**	**0.84 ± 0.0**

https://doi.org/10.1371/journal.pone.0204999.t005

In order to better explore the performance of the considered methods in this double-orphan setting, we used two versions of the two kernel-based methods, initially introduced for *KronRLS*. More precisely, in [19], the authors proposed an approach called WNN (weighted nearest neighbor) that, for each orphan molecule *m* (resp. protein), an interaction profile is computed by summing the weighted profiles of non orphan molecules in the dataset. The weighting depends on the similarity between the orphan molecule and all other non orphan molecules. This predicted profile is used in the training to predict labels to all (protein, *m*) pairs of the dataset. Thus, in the first version of *KronRLS* [18], all the labels of (protein, *m*) pairs involving the orphan molecule *m* were set to 0. Based on this WNN procedure some of these non interactions might be re-qualified as true interaction before training the predictor. In other words, the WNN algorithm can be viewed as a mean to de-orphanize molecules or proteins in order to help the predictions on such cases. In the following, we will call *KronRLS-WNN* the *KronRLS* method ran with the WNN algorithm. Using the PyDTI package, we also considered a version of *NN-MT* in which the WNN algorithm is the implemented, and call it *NN-MT-WNN* in the following.

Table 5 presents the results of the double-orphan benchmark on the family datasets. Surprisingly, in these double-orphan experiments, the *NRLMF* method has very modest results and does not perform as well as the other methods. The results of *NN-MT* remain well above the random performance of 0.5, but not the *KronRLS* method. The WNN algorithm dramatically improves the performance of *KronRLS*, and to a lesser extent those of *NN-MT*, and overall, the *NN-MT-WNN* algorithm leads to the best performance in most cases.

Table 6 presents the results of the double-orphan benchmark $S_{0,2000}$ dataset. We did not run the *NRLMF* method in this experiment, because it was computationally too intensive in this *LOO-CV*, and because it already gave very poor results on the easier family dataset. Moreover we shortened the train set of *KronRLS* and *KronRLS-WNN* methods by considering only the thousand molecules (resp. proteins) closest to the molecule (resp. protein) of the test sample. Thus, the computation time was reduced to some hours instead of months which made those methods computationally reasonable.

Overall, the scores are lower on this dataset than on the family datasets because $S_{0,2000}$ is a more diverse dataset on which predictions are more difficult, in general, than on the family

Table 6. AUPR scores and standard deviations on double orphan *LOO-CV*, balanced number of positive and randomly chosen negative test samples.

Method / Dataset	$S_{0,2000}$
NRLMF	None
KronRLS-WNN	75.6 ± 0.43
KronRLS	0.4979 ± 0.0071
NN-MT	0.60 ± 0.01
NN-MT-WNN	**0.85 ± 0.01**

https://doi.org/10.1371/journal.pone.0204999.t006

datasets. However, the same tendency is observed: *NN-MT* performs better than *KronRLS*, and when the WNN algorithm is used, *NN-MT-WNN* performs better than *KronRLS-WNN*.

Overall, the results of these benchmarks show that the *NN-MT* method present state-of-the-art or better results on the protein family datasets and the more diverse DrugBank-based dataset. In the general case, it appears to be a good default method in terms of performance, number of parameters and computational efficiency, which are important issues for non expert users.

In the specific double-orphan case, only the two kernel-based methods *NN-MT* and *KronRLS* lead to performance well above those of a random predictor. The WNN algorithm, proposed in [19] improves the performance of *KronRLS* and of *NN-MT*, but resulting *NN-MT-WNN* method lead to the best performance.

Finally, it is interesting to compare the computational complexities of the methods as a function of the number of hyper-parameters that they contain. Indeed, these hyper-parameters need to be optimized by cross-validation, leading to heavy computational issues in the case of the large-scale datasets used in proteome-wide chemogenomics. As can be seen in the PyDTI package, *NRLMF* has 5 regularization parameters, *KronRLS* has 2 hyper-parameters (decay parameter T and the weight parameter used to combine kernels; the regularization parameter and the bandwidth of the GIP kernel are fixed), and *NN-MT* has 1 hyper-parameter (regularization parameter *C* for SVM). In practice, the optimization of *NRLMF* in the *LOO-CV* scheme was out of reach, requiring several days of calculation while the other methods required hours, which represents a limitation of this method. This could explain in part the very low performances displayed by *NRLMF* in the double-orphan experiment. In addition, the double-orphan case was not considered in [24], and therefore, we do not have any performance reference for this case. In this setting, latent vectors describing proteins and ligands are both estimated from non-orphan neighbors, and interaction prediction may fail in this situation, and contribute to the poor performance observed. However, we did cross-validate *NRLMF* parameters in the double-orphan setting in the case of the family NR dataset (the smallest dataset used in this section). This allowed a modest increase in AUPR score from 0.14 to 0.19 (reported in Table 5). Therefore, even if the *NRLMF* method had been optimized on the other datasets, we do not expect that this would have changed the overall conclusion that this method is not suitable for handling orphan cases.

5 Conclusion

The present study tackles prediction of ligand specificity on large scale in the space of proteins. More precisely, our goal was to propose a method to explore the specificity of molecules with state-of-the-art or better performance over a wide range of prediction situations: at the proteome or protein family scales, on average or in specific situations such as tested pairs far from the train set, or such as orphan proteins and ligands. In other words, the aim was to propose a robust default method, applicable to many types of studies, thus avoiding development of *ad hoc* complex and specific methods to non expert users. We chose to formulated it as a problem of predicting (protein, ligand) interactions within a multi-task framework based on SVM and Kronecker products of kernels on proteins and molecules. Within the kernel-based SVM methods tested in the Results section, we showed that the *NN-MT* method fulfills these requirements. In particular, *NN-MT* outperforms both the multi-task *MT* method and the corresponding single-task kernel-based methods, while it also keeps a computational cost close to that of single-task approaches. The *NN-MT* algorithm 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 default parameters to use the *NN-MT* method in a straightforward manner for users that are not familiar with machine learning approaches.

We used the DrugBank database to build several datasets that illustrate various prediction contexts and that we made available online to the community for future benchmarking studies. We also updated the PyDTI package [24] with an implementation of *NN-MT* together with several cross-validation schemes and our DrugBank-based dataset. This allowed us to compare the *NN-MT* method to recent approaches developed on drug target interaction prediction [18–24]. In the context of wide-scale prediction of molecule specificity, the DrugBank-based dataset is more relevant that the family datasets that have been widely used. Indeed, it contains a set of proteins that can be viewed as a relevant druggable proteome to train and test computational models for drug specificity prediction.

The benchmark study comparing *NN-MT* to the matrix factorization *NRLMF* and the kernel-based *KronRLS* algorithms on family and DrugBank-based datasets showed that, all methods displayed high performances, *NRLMF* and *NN-MT* leading to the best results. However, on the more demanding double-orphan tests performed on the same datasets, *NRLMF* performed much poorer than the kernel-based *NN-MT* and *KronRLS* algorithms. In this orphan case, the WNN algorithm makes it possible not only to significantly improve the performance of *KronRLS*, but also that of *NN-MT*, the *NN-MT-WNN* algorithm leading to the best results.

We formalized (protein, ligand) interaction prediction as a classification problem because they can be solved at a reasonable computational cost on large datasets. 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 [26]. 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 [28, 29].

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 [66]. 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 [67], 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.

To conclude, this study suggests that including only similar samples to the tested sample forces the algorithm to adjust the hyperplane to segregate positive and negative sample distributions in a better way. In the future, it might be relevant to consider a weighted-sample SVM in order to integrate such behavior in a more continuous way, by decreasing the error importance on train samples depending on their distance to the tested sample.

Supporting information

S1 Fig. (A) Scores of *MT* Kernel Ridge regression on datasets S1/2/3 with a *nested 5-fold-CV* scheme. (B) Scores of *MT* Kernel Ridge regression on S1 depending on the CV scheme.
(TIFF)

S2 Fig. Scores of the *MT* method on S1 depending on CV scheme. Overall, all CV schemes provide high prediction performance on this dataset, in the range of 0.93-0.94 in AUC and AUPR. The *nested 5-fold-CV* leads to performance very close to those of *5-fold-CV*, showing that on the S1 dataset, *5-fold-CV* did not suffer from overestimation of the performance due to data over-fitting. *LOO-CV* leads to slightly better results, although very close to those of the other CV schemes. In general, the *LOO-CV* scheme is expected to provide better results because the model is trained on more data points than in *5-fold-CV*. Again, this problem seems to be limited here, since the performance of *LOO-CV* does not differ much from that of *nested 5-fold-CV*.
(TIFF)

S1 Table. Scores of *MT* SVM in *nested 5-fold-CV* scheme on S1–S4 datasets.
(TIFF)

S2 Table. Scores of *MT* SVM with *nested 5-fold-CV* scheme on S1'–S4' datasets.
(TIFF)

S3 Table. Scores of *MT* SVM on S1 dataset, depending on the CV scheme.
(TIFF)

S4 Table. Scores of *ligand-based ST* and *MT-intra* methods with the *LOO-CV* scheme on S1 dataset.
(TIFF)

S5 Table. Scores of *NN-MT* in *LOO-CV* scheme on S1 dataset.
(TIFF)

S6 Table. Scores of *RN-MT* in *LOO-CV* scheme on S1 dataset.
(TIFF)

S7 Table. Scores of *MT-intra* in *LOO-CV* on S1 dataset with similarity constraint on intra-task pairs.
(TIFF)

S8 Table. Scores of *ligand-based ST* in *LOO-CV* scheme on S1 with similarity constraint on intra-task pairs.
(TIFF)

S9 Table. Scores of *NN-MT* in *LOO-CV* scheme on S1 dataset, with similarity constraint on intra-task pairs ($\theta = 20$).
(TIFF)

S10 Table. Scores of *NN-MT* in *LOO-CV* scheme on S1 dataset, with similarity constraint on intra-task pairs ($\theta = 80$).
(TIFF)

S11 Table. Scores of *RN-MT* in *LOO-CV* on S1 with similarity constraint on intra-task pairs ($\theta = 20$).
(TIFF)

S12 Table. Scores of *RN-MT* in *LOO-CV* on S1 with similarity constraint on intra-task pairs ($\theta = 80$).
(TIFF)

S13 Table. Scores of *MT-intra*, *NN-MT*, *RN-MT* in *LOO-CV* on S1 dataset, with similarity constraint on intra-task pairs.
(TIFF)

S14 Table. Scores of *NN-MT* in *LOO-CV* on S1, with similarity constraint on intra- and extra-task pairs ($\theta = 20$).
(TIFF)

S15 Table. Scores of *NN-MT* in *LOO-CV* on S1, with similarity constraint on intra- and extra-task pairs ($\theta = 80$).
(TIFF)

S16 Table. Scores of *RN-MT* in *LOO-CV* on S1 dataset, with similarity constraint on intra- and extra-task pairs ($\theta = 20$).
(TIFF)

S17 Table. Scores of *RN-MT* in *LOO-CV* on S1 dataset, with similarity constraint on intra- and extra-task pairs ($\theta = 80$).
(TIFF)

S18 Table. GPCR dataset: Scores of *NN-MT* in *LOO-CV* scheme with family's hierarchy based kernel.
(TIFF)

S19 Table. GPCR dataset: Scores of *NN-MT* in *LOO-CV* scheme with sequence based kernel.
(TIFF)

S20 Table. GPCR dataset: Scores of *RN-MT* in *LOO-CV* scheme with family's hierarchy based kernel.
(TIFF)

S21 Table. GPCR dataset: Scores of *RN-MT* in *LOO-CV* scheme with sequence based kernel.
(TIFF)

S22 Table. Ion Channel dataset: Scores of *NN-MT* in *LOO-CV* scheme with family's hierarchy based kernel.
(TIFF)

S23 Table. Ion Channel dataset: Scores of *NN-MT* in *LOO-CV* scheme with sequence based kernel.
(TIFF)

S24 Table. Ion Channel dataset: Scores of *RN-MT* in *LOO-CV* scheme with family's hierarchy based kernel.
(TIFF)

S25 Table. Ion Channel dataset: Scores of *RN-MT* in *LOO-CV* scheme with sequence based kernel.
(TIFF)

S26 Table. Kinase dataset: Scores of *NN-MT* in *LOO-CV* scheme with family's hierarchy based kernel.
(TIFF)

S27 Table. Kinase dataset: Scores of *NN-MT* in *LOO-CV* scheme with sequence based kernel.
(TIFF)

S28 Table. Kinase dataset: Scores of *NN-MT* in *LOO-CV* scheme with family's hierarchy based kernel.
(TIFF)

S29 Table. Kinase dataset: Scores of *NN-MT* in *LOO-CV* scheme with sequence based kernel.
(TIFF)

S1 Appendix. Basic principles of SVM.
(PDF)

S2 Appendix. Definition of the Kronecker product of two matrices A and B.
(PDF)

Author Contributions

Conceptualization: Benoit Playe.

Data curation: Benoit Playe.

Formal analysis: Benoit Playe, Véronique Stoven.

Funding acquisition: Véronique Stoven.

Investigation: Benoit Playe.

Methodology: Benoit Playe, Véronique Stoven.

Supervision: Chloé-Agathe Azencott, Véronique Stoven.

Validation: Benoit Playe, Chloé-Agathe Azencott, Véronique Stoven.

Visualization: Benoit Playe.

Writing – original draft: Benoit Playe, Véronique Stoven.

Writing – review & editing: Benoit Playe, Chloé-Agathe Azencott, Véronique Stoven.

References

1. DiMasi JA, Grabowski HG, Hansen RW. Innovation in the pharmaceutical industry: new estimates of R&D costs. Journal of health economics. 2016; 47:20–33. https://doi.org/10.1016/j.jhealeco.2016.01.012 PMID: 26928437

2. Miguel A, Azevedo LF, Araújo M, Pereira AC. Frequency of adverse drug reactions in hospitalized patients: a systematic review and meta-analysis. Pharmacoepidemiology and drug safety. 2012; 21(11):1139–1154. https://doi.org/10.1002/pds.3309 PMID: 22761169
3. Lazarou J, Pomeranz BH, Corey PN. Incidence of adverse drug reactions in hospitalized patients: a meta-analysis of prospective studies. Jama. 1998; 279(15):1200–1205. https://doi.org/10.1001/jama.279.15.1200 PMID: 9555760
4. Onakpoya IJ, Heneghan CJ, Aronson JK. Post-marketing withdrawal of 462 medicinal products because of adverse drug reactions: a systematic review of the world literature. BMC medicine. 2016; 14(1):10. https://doi.org/10.1186/s12916-016-0553-2 PMID: 26843061
5. Scheiber J, Chen B, Milik M, Sukuru SCK, Bender A, Mikhailov D, et al. Gaining insight into off-target mediated effects of drug candidates with a comprehensive systems chemical biology analysis. Journal of chemical information and modeling. 2009; 49(2):308–317. https://doi.org/10.1021/ci800344p PMID: 19434832
6. Martinez-Lopez Y, Caballero Y, J Barigye S, Marrero-Ponce Y, Millan-Cabrera R, Madera J, et al. State of the Art Review and Report of New Tool for Drug Discovery. Current topics in medicinal chemistry. 2017; 17(26):2957–2976. https://doi.org/10.2174/1568026617666170821123856 PMID: 28828995
7. Xu X, Huang M, Zou X. Docking-based inverse virtual screening: methods, applications, and challenges. Biophysics reports. 2018; p. 1–16. https://doi.org/10.1007/s41048-017-0045-8 PMID: 29577065
8. Vert JP, Jacob L. Machine learning for in silico virtual screening and chemical genomics: new strategies. Combinatorial chemistry & high throughput screening. 2008; 11(8):677–685. https://doi.org/10.2174/138620708785739899
9. Bolton EE, Wang Y, Thiessen PA, Bryant SH. PubChem: integrated platform of small molecules and biological activities. Annual reports in computational chemistry. 2008; 4:217–241. https://doi.org/10.1016/S1574-1400(08)00012-1
10. Yamanishi Y, Araki M, Gutteridge A, Honda W, Kanehisa M. Prediction of drug–target interaction networks from the integration of chemical and genomic spaces. Bioinformatics. 2008; 24(13):i232–i240. https://doi.org/10.1093/bioinformatics/btn162 PMID: 18586719
11. Jacob L, Vert JP. Protein-ligand interaction prediction: an improved chemogenomics approach. Bioinformatics. 2008; 24(19):2149–2156. https://doi.org/10.1093/bioinformatics/btn409 PMID: 18676415
12. Bleakley K, Yamanishi Y. Supervised prediction of drug–target interactions using bipartite local models. Bioinformatics. 2009; 25(18):2397–2403. https://doi.org/10.1093/bioinformatics/btp433 PMID: 19605421
13. Yamanishi Y, Kotera M, Kanehisa M, Goto S. Drug-target interaction prediction from chemical, genomic and pharmacological data in an integrated framework. Bioinformatics. 2010; 26(12):i246–i254. https://doi.org/10.1093/bioinformatics/btq176 PMID: 20529913
14. Hizukuri Y, Sawada R, Yamanishi Y. Predicting target proteins for drug candidate compounds based on drug-induced gene expression data in a chemical structure-independent manner. BMC medical genomics. 2015; 8(1):1. https://doi.org/10.1186/s12920-015-0158-1
15. Takarabe M, Kotera M, Nishimura Y, Goto S, Yamanishi Y. Drug target prediction using adverse event report systems: a pharmacogenomic approach. Bioinformatics. 2012; 28(18):i611–i618. https://doi.org/10.1093/bioinformatics/bts413 PMID: 22962489
16. Yamanishi Y. Inferring Chemogenomic Features from Drug-Target Interaction Networks. Molecular Informatics. 2013; 32(11-12):991–999. https://doi.org/10.1002/minf.201300079 PMID: 27481144
17. Yuan Q, Gao J, Wu D, Zhang S, Mamitsuka H, Zhu S. DrugE-Rank: improving drug–target interaction prediction of new candidate drugs or targets by ensemble learning to rank. Bioinformatics. 2016; 32(12):i18–i27. https://doi.org/10.1093/bioinformatics/btw244 PMID: 27307615
18. van Laarhoven T, Nabuurs SB, Marchiori E. Gaussian interaction profile kernels for predicting drug–target interaction. Bioinformatics. 2011; 27(21):3036–3043. https://doi.org/10.1093/bioinformatics/btr500 PMID: 21893517
19. van Laarhoven T, Marchiori E. Predicting drug-target interactions for new drug compounds using a weighted nearest neighbor profile. PloS one. 2013; 8(6):e66952. https://doi.org/10.1371/journal.pone.0066952 PMID: 23840562
20. Mei JP, Kwoh CK, Yang P, Li XL, Zheng J. Drug–target interaction prediction by learning from local information and neighbors. Bioinformatics. 2013; 29(2):238–245. https://doi.org/10.1093/bioinformatics/bts670 PMID: 23162055
21. Xia Z, Wu LY, Zhou X, Wong ST. Semi-supervised drug-protein interaction prediction from heterogeneous biological spaces. BMC systems biology. 2010; 4(Suppl 2):S6. https://doi.org/10.1186/1752-0509-4-S2-S6 PMID: 20840733

22. Zheng X, Ding H, Mamitsuka H, Zhu S. Collaborative matrix factorization with multiple similarities for predicting drug-target interactions. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM; 2013. p. 1025–1033.
23. Gönen M. Predicting drug–target interactions from chemical and genomic kernels using Bayesian matrix factorization. Bioinformatics. 2012; 28(18):2304–2310. https://doi.org/10.1093/bioinformatics/bts360 PMID: 22730431
24. Liu Y, Wu M, Miao C, Zhao P, Li XL. Neighborhood regularized logistic matrix factorization for drug-target interaction prediction. PLoS computational biology. 2016; 12(2):e1004760. https://doi.org/10.1371/journal.pcbi.1004760 PMID: 26872142
25. Johnson CC. Logistic matrix factorization for implicit feedback data. Advances in Neural Information Processing Systems. 2014; 27.
26. Pahikkala T, Airola A, Pietilä S, Shakyawar S, Szwajda A, Tang J, et al. Toward more realistic drug–target interaction predictions. Briefings in bioinformatics. 2014; p. bbu010. https://doi.org/10.1093/bib/bbu010 PMID: 24723570
27. Jacob L, Hoffmann B, Stoven V, Vert JP. Virtual screening of GPCRs: an in silico chemogenomics approach. BMC bioinformatics. 2008; 9(1):363. https://doi.org/10.1186/1471-2105-9-363 PMID: 18775075
28. Paul N, Kellenberger E, Bret G, Müller P, Rognan D. Recovering the true targets of specific ligands by virtual screening of the protein data bank. Proteins: Structure, Function, and Bioinformatics. 2004; 54(4):671–680. https://doi.org/10.1002/prot.10625
29. Kellenberger E, Foata N, Rognan D. Ranking targets in structure-based virtual screening of three-dimensional protein libraries: methods and problems. Journal of chemical information and modeling. 2008; 48(5):1014–1025. https://doi.org/10.1021/ci800023x PMID: 18412328
30. Hue M, Riffle M, Vert JP, Noble WS. Large-scale prediction of protein-protein interactions from structures. BMC bioinformatics. 2010; 11(1):144. https://doi.org/10.1186/1471-2105-11-144 PMID: 20298601
31. Caruana R. Multitask learning. In: Learning to learn. Springer; 1998. p. 95–133.
32. Bakker B, Heskes T. Task clustering and gating for bayesian multitask learning. Journal of Machine Learning Research. 2003; 4(May):83–99.
33. Arora N, Allenby GM, Ginter JL. A hierarchical Bayes model of primary and secondary demand. Marketing Science. 1998; 17(1):29–44. https://doi.org/10.1287/mksc.17.1.29
34. Allenby GM, Rossi PE. Marketing models of consumer heterogeneity. Journal of econometrics. 1998; 89(1-2):57–78. https://doi.org/10.1016/S0304-4076(98)00055-4
35. Cortes C, Vapnik V. Support-vector networks. Machine learning. 1995; 20(3):273–297. https://doi.org/10.1007/BF00994018
36. Erhan D, L'Heureux PJ, Yue SY, Bengio Y. Collaborative filtering on a family of biological targets. Journal of chemical information and modeling. 2006; 46(2):626–635. https://doi.org/10.1021/ci050367t PMID: 16562992
37. Faulon JL, Misra M, Martin S, Sale K, Sapra R. Genome scale enzyme–metabolite and drug–target interaction predictions using the signature molecular descriptor. Bioinformatics. 2008; 24(2):225–233. https://doi.org/10.1093/bioinformatics/btm580 PMID: 18037612
38. Napolitano F, Zhao Y, Moreira VM, Tagliaferri R, Kere J, D'Amato M, et al. Drug repositioning: a machine-learning approach through data integration. J Cheminformatics. 2013; 5:30. https://doi.org/10.1186/1758-2946-5-30
39. Jaakkola T, Diekhans M, Haussler D. A discriminative framework for detecting remote protein homologies. Journal of computational biology. 2000; 7(1-2):95–114. https://doi.org/10.1089/10665270050081405 PMID: 10890390
40. Leslie CS, Eskin E, Noble WS. The spectrum kernel: A string kernel for SVM protein classification. In: Pacific symposium on biocomputing. vol. 7; 2002. p. 566–575.
41. Eskin E, Weston J, Noble WS, Leslie CS. Mismatch string kernels for SVM protein classification. In: Advances in neural information processing systems; 2002. p. 1417–1424.
42. Saigo H, Vert JP, Ueda N, Akutsu T. Protein homology detection using string alignment kernels. Bioinformatics. 2004; 20(11):1682–1689. https://doi.org/10.1093/bioinformatics/bth141 PMID: 14988126
43. Kuang R, Ie E, Wang K, Wang K, Siddiqi M, Freund Y, et al. Profile-based string kernels for remote homology detection and motif extraction. Journal of bioinformatics and computational biology. 2005; 3(03):527–550. https://doi.org/10.1142/S021972000500120X PMID: 16108083

44. Smith TF, Waterman MS. Identification of common molecular subsequences. Journal of molecular biology. 1981; 147(1):195–197. https://doi.org/10.1016/0022-2836(77)90031-6 PMID: 7265238

45. Okuno Y, Tamon A, Yabuuchi H, Niijima S, Minowa Y, Tonomura K, et al. GLIDA: GPCR—ligand database for chemical genomics drug discovery—database and tools update. Nucleic acids research. 2007; 36(suppl_1):D907–D912. https://doi.org/10.1093/nar/gkm948 PMID: 17986454

46. Manning G, Whyte DB, Martinez R, Hunter T, Sudarsanam S. The protein kinase complement of the human genome. Science. 2002; 298(5600):1912–1934. https://doi.org/10.1126/science.1075762 PMID: 12471243

47. Kanehisa M, Araki M, Goto S, Hattori M, Hirakawa M, Itoh M, et al. KEGG for linking genomes to life and the environment. Nucleic acids research. 2007; 36(suppl_1):D480–D484. https://doi.org/10.1093/nar/gkm882 PMID: 18077471

48. Swamidass SJ, Chen J, Bruand J, Phung P, Ralaivola L, Baldi P. Kernels for small molecules and the prediction of mutagenicity, toxicity and anti-cancer activity. Bioinformatics. 2005; 21(suppl 1):i359–i368. https://doi.org/10.1093/bioinformatics/bti1055 PMID: 15961479

49. Kashima H, Tsuda K, Inokuchi A. Marginalized kernels between labeled graphs. In: ICML. vol. 3; 2003. p. 321–328.

50. Mahé P, Ueda N, Akutsu T, Perret JL, Vert JP. Graph kernels for molecular structure-activity relationship analysis with support vector machines. Journal of chemical information and modeling. 2005; 45(4):939–951. https://doi.org/10.1021/ci050039t PMID: 16045288

51. Azencott CA, Ksikes A, Swamidass SJ, Chen JH, Ralaivola L, Baldi P. One-to four-dimensional kernels for virtual screening and the prediction of physical, chemical, and biological properties. Journal of chemical information and modeling. 2007; 47(3):965–974. https://doi.org/10.1021/ci600397p PMID: 17338509

52. Golub GH, Heath M, Wahba G. Generalized cross-validation as a method for choosing a good ridge parameter. Technometrics. 1979; 21(2):215–223. https://doi.org/10.1080/00401706.1979.10489751

53. Friedman J, Hastie T, Tibshirani R. The elements of statistical learning. vol. 1. Springer series in statistics New York; 2001.

54. Hanley JA, McNeil BJ. The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology. 1982; 143(1):29–36.

55. Raghavan V, Bollmann P, Jung GS. A critical investigation of recall and precision as measures of retrieval system performance. ACM Transactions on Information Systems (TOIS). 1989; 7(3):205–229. https://doi.org/10.1145/65943.65945

56. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 2011; 12:2825–2830.

57. Kanehisa M, Goto S, Sato Y, Furumichi M, Tanabe M. KEGG for integration and interpretation of large-scale molecular data sets. Nucleic acids research. 2011; p. gkr988.

58. Law V, Knox C, Djoumbou Y, Jewison T, Guo AC, Liu Y, et al. DrugBank 4.0: shedding new light on drug metabolism. Nucleic acids research. 2013; 42(D1):D1091–D1097. https://doi.org/10.1093/nar/gkt1068 PMID: 24203711

59. Gaulton A, Bellis LJ, Bento AP, Chambers J, Davies M, Hersey A, et al. ChEMBL: a large-scale bioactivity database for drug discovery. Nucleic acids research. 2012; 40(D1):D1100–D1107. https://doi.org/10.1093/nar/gkr777 PMID: 21948594

60. Lipinski CA, Lombardo F, Dominy BW, Feeney PJ. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. Advanced drug delivery reviews. 1997; 23(1-3):3–25. https://doi.org/10.1016/S0169-409X(96)00423-1

61. Schölkopf B, Platt JC, Shawe-Taylor J, Smola AJ, Williamson RC. Estimating the support of a high-dimensional distribution. Neural computation. 2001; 13(7):1443–1471. https://doi.org/10.1162/089976601750264965 PMID: 11440593

62. Kramer C, Gedeck P. Leave-cluster-out cross-validation is appropriate for scoring functions derived from diverse protein data sets. Journal of chemical information and modeling. 2010; 50(11):1961–1969. https://doi.org/10.1021/ci100264e PMID: 20936880

63. Johnson SC. Hierarchical clustering schemes. Psychometrika. 1967; 32(3):241–254. https://doi.org/10.1007/BF02289588 PMID: 5234703

64. Warmuth MK, Liao J, Rätsch G, Mathieson M, Putta S, Lemmen C. Active learning with support vector machines in the drug discovery process. Journal of chemical information and computer sciences. 2003; 43(2):667–673. https://doi.org/10.1021/ci025620t PMID: 12653536

65. Hattori M, Okuno Y, Goto S, Kanehisa M. Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways. Journal of the

American Chemical Society. 2003; 125(39):11853–11865. https://doi.org/10.1021/ja036030u PMID: 14505407

66. Rakotomamonjy A, Bach FR, Canu S, Grandvalet Y. SimpleMKL. Journal of Machine Learning Research. 2008; 9(Nov):2491–2521.

67. Nascimento AC, Prudêncio RB, Costa IG. A multiple kernel learning algorithm for drug-target interaction prediction. BMC bioinformatics. 2016; 17(1):1. https://doi.org/10.1186/s12859-016-0890-3