Master DMKM Report

Towards the Improvement of the Protein-Ligand Binding Affinity Prediction, using Machine Learning Techniques

Gabriela HERNÁNDEZ LARIOS

defended the 07/06/2016

Supervision : Víctor Guallar[1], Jorge Estrada[2], Ricard Gavaldà[3]

Location : Barcelona Supercomputing Center - Centro Nacional de Supercomputación

[1]BSC-CNS Advisor. Nexus II Building. C. Jordi Girona, 29, 08034. Barcelona, Spain
[2]BSC-CNS Advisor.Nexus II Building. C. Jordi Girona, 29, 08034. Barcelona, Spain
[3]UPC Advisor. Campus Diagonal Nord. C. Jordi Girona, 1-3, 08034. Barcelona, Spain

Abstract:

Predicting protein-ligand binding affinity constitutes a key computational method in the early stages of the drug discovery process. Molecular docking programs attempt to predict them by using mathematical approximations, namely, scoring functions. Constantly, new experiments and techniques are carried out in order to find out crucial descriptors that better characterize the protein-ligand interaction. In this work, we investigate and apply different machine learning and statistical techniques to create a novel framework capable of combining different descriptors as well as scoring functions in order to both estimate and improve the overall binding affinity. This strategy also filters the strongest descriptors and scoring functions while permitting more complex interpretations by examining non-linearities and interactions. This approach consists of two steps. First,several descriptors and scoring functions are separately combined and assessed through models based on penalized linear regression methods with embedded feature selection, such as LASSO and Elastic Net. In order to avoid strong parametric assumptions, alternative intelligible non-parametric techniques are exploited such as Generalized Additive Models and Kernel-based Regularized Least Squares. Secondly, stacking methods are employed to further boost the binding affinity-prediction of different scoring functions, by adding new models with the descriptors. We apply this methodology to well-studied datasets of high-quality protein-ligand complexes, based on the 2007 and 2013 PDBbind Benchmarks, achieving a significant improvement in overall prediction of binding affinity. Furthermore, by analyzing the results of the models, we acquire important insights on the role of the descriptors and scoring functions to estimate the binding free energy.

Résumé :

Prédire l'affinité de liaison ligand-protéine constitue une méthode informatique clé dans les premiers stades du processus de découverte des médicaments. Des programmes d'amarrage moléculaire tentent de les prévoir en utilisant des approximations mathématiques, à savoir, des fonctions de score. Constamment, des nouvelles expériences et des techniques sont réalisées dans le but de trouver des descripteurs essentiels qui mieux caractérisent l'interaction protéine-ligand. Dans ce travail, nous étudions et appliquons des différentes techniques d'apprentissage automatiqueet des statistiques pour créer un nouveau cadre capable de combiner des descripteurs différents ainsi que des fonctions de score pour estimer et améliorer l'affinité de liaison globale. Cette stratégie filtre également les descripteurs les plus forts et les fonctions de score, tout en permettant des interprétations plus complexes, en examinant les non-linéarités et lesinteractions. Cette approche consiste en deux étapes. Premièrement, plusieurs descripteurs et fonctions de score sont combinés et évalués séparément, en utilisant des modèles basés sur des méthodes de régression pénalisé, avec une sélection de variables intégrée comme LASSO et Elastic Net. Afin d'éviter des fortes hypothèses paramètriques, des techniques non-paramètriqueset intelligibles sont exploitèes, tels que Modèles Additifs Généralisés et Régression par Moindres Carrés Régularisés à base des Noyaux. Deuxièmement, des méthodes d'ensemble sont utilisées pour stimuler davantage l'affinité de liaison de prédiction des différentes fonctions de score, en ajoutant des descripteurs. Nous appliquons cette méthodologie à données avec des complexes ligand-récepteurs de haute qualité, sur la base du PDBbind Benchmarks 2007 et 2013, et ainsi nous obtenons une amélioration importante dans la prévision globale de l'affinité de liaison. En outre, en analysant les réesultats des modèles, nous acquérons des renseignements utiles sur le rôle des descripteurs et des fonctions de score pour estimer l'énergie libre de liaison.

Contents

1 Hosting institution

BSC is a unique fusion of a classic national scientific support structure and a cutting-edge research institute, endowing it to develop the next generation of Supercomputing technologies and apply multidisciplinary research teams to bridge the gap between high performance computing (HPC) hardware and the user-friendly applications required by scientists. BSC currently totals some 380 staff from around 40 different countries, hosts around 50-70 visitors, and trains in the order of 15-20 new PhDs every year.

BSC is not only a research institute, it is also the National Supercomputing Centre and a PRACE node, offering Spanish, European and international users access to one of the six most powerful supercomputers in Europe. BSC largest computer, MareNostrum 3, has a compute capacity of 1.1 PFlops, based on 48896 Intel cores and 84 Xeon Phi. In December 2015, it was approved by the Spanish Government a 34 million euros budget for building the new MareNostrum 4, to be implemented at the end of 2016.

BSC's areas of research include both HPC hardware and software, as well as multidisciplinary teams developing cutting edge applications in Life and Earth sciences and Engineering. This combination of depth in computer science combined with broad coverage of key fields, enables BSC to play a pivotal role in many national and international collaborative initiatives.

The Life Sciences Department has 5 principal researchers and approximately 70 people including students, postdocs and senior researchers. In addition, BSC has signed an agreement with the Biomedical Institute of Research (IRB), and the Center for Genomic Regulation (CRG), to form a trijoint center covering most of aspects in computational Biology.

The master project has been developed at the "Electronic and Atomic Protein Modeling" (EAPM) group, led by Professor Victor Guallar. The the group is currently formed (in addition to the PI) by 1 senior researcher, 4 postdoctoral researchers, 9 PhD students, 2 master students and 2 technicians, with approximately 50% of the group members being foreigners. The EAPM is devoted to develop and apply computational algorithms for understanding and predicting biochemistry and biophysics at a molecular level. For this purpose, they focus on two different sets of modeling techniques: classical force field simulations (mainly using the in-house Monte Carlo algorithm PELE), and mixed quantum mechanics/molecular mechanics (QM/MM) techniques. The lab places emphasis in methods development and in their implementation to applied research areas, primarily involving drug design and enzyme engineering. One of the main objectives of the lab is to couple their state of the art sampling technique, PELE, with robust scoring functions developed through cognitive computing. For this, they have partner with Ricard Gavalda, from the Laboratory for Relational Algorithmics, Complexity and Learning (LARCA) group at UPC, an international research group working on data mining, machine learning, data analysis, and mathematical linguistics. The first realization of this joint effort is this current master project, co-directed by Prof. Ricard Gavald à (UPC), and Dr Jorge Estrada and Prof. Victor Guallar (from the BSC).

2 Acknowledgements

Firstly, I would like to express my deepest thanks to my tutors Prof. Víctor Guallar, Dr. Jorge Estrada and Prof. Ricard Gavaldà, for their wisely advices and suggestions, for their corrections and recommendations for this report and for all the support and guidance to accomplish the objectives of this work.

I would like to express my sincere gratitude to Suwipa Saen-oon, Daniel Lecina and Jelisa Iglesias, for preparing the datasets and for helping me to understand biochemistry concepts that where far away from my scope.

I would also like to thank to all my fellow colleagues, for the exchange of knowledge and experiences and for their support along this Master.

Gabriela

3 Introduction

The amount of proteins and molecules with publicly-accessible 3D structures is rapidly growing [17]. In parallel, there has been a significant advance in molecular target therapies, where specific receptors, mostly proteins, are identified as possible molecular targets for therapy [33]. As a consequence, Structure-Based Drug Design (SBDD) is becoming increasingly popular to discover new potential drugs. In this process, the protein-ligand binding plays a fundamental role. For a protein of interest, putative ligand drug candidates are discovered or designed in order to bind the target protein and modulate its activity. The strength of these docked molecules is referred as binding affinity [2]. In *vitro* determination of binding affinity is highly expensive and time consuming. In order to address this issue, *in silico* molecular docking techniques have emerged, using scoring functions (SFs) to estimate the binding affinity of each protein-ligand complex [1].

In general, SFs can be broadly classified into four categories: 1) force-field based, 2) knowledge-based, 3) descriptor-based and 4) empirical scoring functions [26]. They have been under continuous development since the 80s, encompassing different terms, from electrostatic forces to protein-ligand interaction fingerprints and beyond. However, it has been noticed that usually each individual SF cannot be generalized for every protein-ligand dataset and its predictive power is arguable.

Moreover, new experiments and techniques are constantly conducted in order to determine further specific descriptors that can provide more precisely information about the protein-ligand complex, nonetheless, yet there is no framework that provides guidelines on how to combine them with either other descriptors or scoring functions, without loosing predictive power and intelligibility.

Recently, many SFs have been attempting to be developed by using machine learning techniques [1, 2, 21]. However, they have been limited trained to predict the binding affinity of only true complexes [6]. In the pharmaceutical industry, such a scenario is typically not the normal working conditions.

All this suggest that a more general procedure, capable of combining SFs and descriptors in a more general manner, would be of high value to the academic and industrial drug design field. Based on the former premise, in this work, we aim to overcome the limitations of the actual strategies by providing a novel framework, based mainly on machine learning and statistical techniques. With this new methodology, we seek to address two important aspects. First, provide different methods for combining sets of SFs and/or descriptors, to estimate and improve the overall binding affnity prediction. Secondly, bring specific guidelines for combining either a single or a set of SFs with new descriptors; relying on the fact that it might not be necessary, for some situations, create again new SFs when new descriptors emerge. Both aspects were intended to be developed in an accurately yet intelligible and flexible manner.

The rest of this report is structured as follows. Section 4 briefly introduces the literature regarding the use of Machine Learning (ML) algorithms to predict the protein-ligand Binding Affinity; along this section some comments about the previous studies are also presented. Section 5 fully describes the protein-ligand datasets, the methodology behind their computation and the construction of the case-study dataset used throughout this study. Section 6 presents the main machine learning and statistical foundations applied in this work jointly with the proposed framework; furthermore, explanation of the models employed and their treatment is also provided. Section 7, exhibits the main results of the designed protein-ligand case-study using the proposed methodology. Section 8 discusses the results presented in the former section, highlighting the importance of this study and pointing out the future work to improve our model. Finally, Section 9 recapitulates the main findings of the work and outlines final remarks.

4 Related Work

Despite the efforts in develop SFs conformed by different kind of terms and underlying different principles to accurately predict the binding free energy, it has been shown in different studies [8, 31, 6], their limited predictive power and generalization. Nevertheless, it also has been noticed that it is very unlikely that a set of SFs will be in error at the same time for a protein-ligand system. Based on this idea, exhaustive studies have been realized to create a most robust scoring function by using the best combination of a set of individual scoring functions in different fashions. Some attempts were performed in [29, 31], where the authors sought to create consensus scoring functions based on conventional approaches such as rank-based, percent-based, range-based and vote-based strategies in order to combine sets of SFs to predict the binding free energy, however their results are based on a strong assumption which entails that all the individual scoring functions contribute equally for each configuration being scored [6].

In other studies, the authors proposed protocols to rescue poor docking results from different SFs by combining conventional approaches such as rank-based with a classifier in order to only discriminate good and bad binders for some target proteins with a set of ligands, without predicting the binding free energy [23, 22].

With the emergency of new descriptors and freely available databases of protein-ligand complexes, many SFs have been attempting to be developed by using machine learning techniques [20, 35, 1, 4, 2]. However, all these studies are very skewed either for a small scale experiments in which just a specific set of protein-ligand complexes are used for training and validation, or for a specific machine learning technique, with a lack of flexibility to be used with datasets encompassing other different terms.

In [3] the authors sought to add new descriptors to their already developed RFScore SF. They point out that adding more precise chemical descriptor does not lead to a more accurate prediction of the binding affinity. Nonetheless, their justifications are mainly based on the chemistry behind the descriptions and not in their model. Moreover, they do not provide any solution to tackle this problem.

To the best of our knowledge, no study has fully investigated and exploited different machine learning and statistical techniques in order to better understand how to improve the overall binding affinity prediction of a protein-ligand complex, not only for combining different SFs and descriptors, separately, but also for including new descriptors to already developed SFs. Leaving room for improvements.

5 Protein-Ligand Complex Datasets

Protein-Ligand Complex structures are stored as pdb format files in the Protein Data Bank (PDB), freely accessible, which is an archive of experimentally determined three-dimensional structures of biological macromolecules. These files include atomic coordinates, crystallographic structure factors and Nuclear Magnetic Resonance spectroscopy, experimental data. Aside from coordinates, each deposition also includes the names of molecules, primary and secondary structure information, sequence database references, ligand and biological assembly information [5].

There exists databases in charge of gather, process, filter and store archives for numerous high quality Protein-Ligand complexes and other molecules deposited in PDB, such as the PDBbind database [4]. The PDBbind database is a comprehensive collection of experimentally measured binding affinity data for the protein-ligand complexes deposited in PDB. It thus provides a link between energetic and structural information of protein-ligand complexes. The PDBbind database is updated on an annual basis to keep up with the growth of the Protein Data Bank, therefore, the datasets, are organized by years, starting from 2007 until 2015. In this study, we worked with the datasets corresponding to the 2007 and 2013 years. From these collections of protein-ligand systems, various scoring functions (SFs) can be calculated and descriptors can be extracted.

In this section, we present the most important information regarding the construction and methodologies behind the datasets containing SFs and descriptors for various protein-ligand complexes. In Section 5.1, we explain the underlying theory of the SFs as well as their classification and we briefly describe the SFs we are using in this study. In Section 5.2 an overview of different possible descriptors is provided, and we detailed the ones we are using. Lastly, in Section 5.3, we explain the preparation and selection of the datasets constructed for the case study.

5.1 Scoring Functions (SFs)

From a collection of several protein-ligand complexes in pdb format, several Scoring Functions (SFs), underlying different principles, can be calculated and descriptors derived. SFs are methods especially developed for evaluating protein-ligand interactions in structure-based drug design. General speaking, the purpose of SFs is to maximize the correlation with the binding affinity for a set of known binders.

The binding affinity measures the strength of the binding between the protein and a ligand (smaller molecule), basically allowing to know, under equilibrium conditions, which percentage of the protein molecules will be bound. Binding is measured by using the dissociation constant K_d, which gives the relation of bound and unbound concentrations of a ligand binding a protein. Alternatively, the binding affinity can be expressed in energy terms as the free energy of binding which is related to K_d through the logarithmic function $\Delta G = RTln(k_d)$, where R is the proportional constant of 0.002, and T the temperature measured in Kelvins. In this work, all the K_d were assumed to be at room temperature (T = 300 Kelvin).

SFs can be classified according to the techniques and principles used to create them, in order to avoid ambiguities, in [26], the authors propose the following up-to-date classification scheme of the current scoring functions:

1. *Category I: "Force-Field based" or "Physics-Based".* Relies on available force fields to compute the direct interaction between protein and ligand. Force field energy functions are designed to compute potential energy in the gas phase, which is only one component of the free energy change in a protein-ligand binding process. Initially, the noncovalent energy terms and electrostatic terms in a force field and hydrogen bonding were taking into consideration, and later these functions were augmented by using solvation energy terms such as Poisson-Boltzmann (PB) or Generalized Born (GB) continuum solvation models.

[4] http://www.pdbbind.org.cn

2. *Category II: "Empirical" or "Regression-Based".* An empirical SF computes the fitness of protein-ligand binding by summing up the contributions of a number of individual terms, each representing an important energetic factor in protein-ligand binding. Since multiple terms with different implications are combined to give the final binding score, an empirical scoring function normally relies on multivariate linear regression (MLR) or partial least-squares (PLS) analysis to derive the weight factor of each term.

3. *Category III: "Knowledge-based".* Despite their background knowledge and technical aspects, in general these functions sum pairwise statistical potentials between a protein and a ligand,

$$A = \sum_{i}^{lig} \sum_{j}^{prot} \omega_{ij}(r) \tag{1}$$

where $\omega_{ij}(r)$ is the distance-dependent potential between atom pair $i-j$, and it is derived from an inverse Boltzmann analysis as:

$$\omega_{ij}(r) = -k_B T ln[g_{ij}(r)] = -k_B T ln\left[\frac{\rho_{ij}(r)}{\rho^*_{ij}}\right] \tag{2}$$

In equation 2, $\rho_{ij}(r)$ is the numeric density of atom pair $i-j$ at distance r and ρ^*_{ij} is the numeric density of the same atom pair in a reference state where inter-atomic interactions are assumed to be zero. With this approach, the occurrence frequency of a pairwise contact is assumed to be a measure of its energetic contribution to protein-ligand binding. Hence, if a specific pairwise contact occurs more frequently than that in the reference state, it indicates an energetically favorable interaction between the given atom pair, if it occurs less frequently, then it indicates an unfavorable interaction.

4. *Category IV: "Descriptor-based" or "Machine Learning based".* These methods introduce modern Quantitative Structure-Activity Relationship (QSAR) analysis into protein-ligand interaction evaluation. These functions are based on the fact that there can be certain patterns in the protein-ligand interactions that can be coded with some descriptors such as structural interaction fingerprints, electrostatic interactions, hydrogen bonds, aromatic stacking, geometrical descriptors (shape or surface properties) and conventional ligand-based descriptors (molecular weight, number of rotable single bonds, etc.). Thereafter, with the descriptors, machine learning techniques are employed to derive statistical models that compute protein-ligand binding scores.

For this study, we used 10 SFs, having at least one from each category. For avoiding ambiguities, we are going to briefly describe them by categories.

Force-Field Based:

1. MM-GBSA

   The MM-GBSA (Molecular Mechanical-Generalized Born Surface Area) approach aims at estimating the free energy difference between two states which most often represent the bound and unbound state of two solvated molecules. In this method, ΔG_{bind} is estimated from the free energies of the "reactants" and "product" of the binding "reaction" equation

$$\Delta G_{bind} = \langle G_{PL} \rangle - \langle G_P \rangle - \langle G_L \rangle \tag{3}$$

   In equation 3, G (which is usually averaged among different conformations, as indicated by the brackets) represents the free energy of a state, that is P, L or PL, and it is estimated as follows:

$$G = E_{bnd} + E_{el} + E_{vdW} + G_{pol} + G_{np} - TS \tag{4}$$

where E_{bnd}, E_{el}, E_{vdW} are the bonded, electrostatic and van der Waals energy terms, respectively whereas G_{pol} and G_{np} are the polar a non-polar contributions to the solvation free energies. In the last term TS, T stands for the absolute temperature and S is the entropy.

Genheden et al. reviewed the use of this method in [16]. The review concludes that the results strongly depend on certain details such as the charges used for the receptor and the ligand, the dielectric constant used for the protein, the sampling method and the entropies. It has also been shown that this method often overestimate differences between sets of ligands. Another downside of this method is that it involves strong assumptions in approximating the entropy and lack of information about the number and free energy of water molecules in the binding site. Therefore, this method is useful to improve the results of docking and virtual screening or to understand observed affinities, but it should not be used for later states of predictive drug design.

In this work, the MM-GBSA implementation of Prime [5] has been used, which considers a single conformation for the bound (PL) state and, after separating protein and ligand, uses the same conformations, minimized, for the protein (P) and ligand (L). The entropic term (TS) is ignored for the calculation of the free energy δG, and the rest of terms are calculated using the VSGB 2.0 energy model [24].

Empirical:

X-Score
X-Score is a scoring function proposed by Wang et al.(2002) [38] and is based on the combination of three individual empirical functions that include terms accounting for van der Waals interaction, hydrogen bonding, deformation penalty, and hydrophobic effect. From these, four different algorithms have been implemented:

2. XScore::HSScore, to calculate the hydrophobic effect term, using the buried hydrophobic surface of the ligand.

3. XScore::HPScore, to calculate the hydrophobic effect term, using the the hydrophobic atomic contacts.

4. XScore::HMScore, to calculate the hydrophobic effect term, using the hydrophobic matching between ligand and protein.

5. XScore::Average, employs a calibrated multivariate regression analysis with HSScore, HPScore and HMScore of a set of 200 protein-ligand complexes, reproducing the binding free energies of the entire training dataset.

   In this work, X-Score v1.2.1 has been used.

Knowledge-based:

6. DSX:

   DSX, proposed by Wang et al. (2011) [30], is a knowledge-based scoring function that consists of distance-dependent pair potentials, novel torsion angle potentials, and newly defined solvent accessible surface-dependent potentials. In this work, DSX version 0.90 has been used.

7. Glide-SP: GlideScore-Standard Precision, is a "softer" scoring function, proposed by R. A. Friesner et al. (2004) [14], to be used with the Glide software [6]. This scoring is a more forgiving function than Glide-XP designed for identifying ligands that have a reasonable propensity to bind, even in cases in which the Glide pose has significant imperfections. This version seeks to minimize false negatives. This function can be seen as an extension of the empirically based ChemScore function [12].

[5] Schrödinger Release 2016-1: Prime, version 4.3, Schrödinger, LLC, New York, NY, 2016
[6] Small-Molecule Drug Discovery Suite 2016-1: Glide, version 7.0, Schrödinger, LLC, New York, NY, 2016.

8. Glide-XP:

   Grid-based Ligand Docking with Energetics- Extra Precision (Glide-XP), proposed by R. A. Friesner et al. (2006) [15], is a scoring function that takes into account unique water desolvation energy terms, the presence of hydrophobic enclosure motifs where groups of lipophilic ligand atoms are enclosed on opposite faces by lipophilic protein atoms, neutral-neutral single or correlated hydrogen bonds in a hydrophobically enclosed environment, and five categories of charged-charged hydrogen bonds. The XP scoring function and docking protocol have been developed to reproduce experimental binding affinities for a set of 198 complexes. The key novel features characterizing XP Glide scoring are: the application of large desolvation penalties to both ligand and protein polar and charged groups in appropriate cases; and the identification of specific structural motifs that provide exceptionally large contributions to enhanced binding affinity.

9. Autodock Vina The original AutoDock SF [28] is derived from the AMBER [9] force field and includes five components for atom-atom interactions as well as an estimate of the conformational entropy loss upon binding. The atom-atom interactions used in this function are the following: a dispersion repulsion term, a directional Hbond term, the Coulomb potential to estimate electrostatic interactions and the desolvation potential energy. [36]. The AutoDock Vina SF [37], is a modification of the Autodock SF. This new version is based on pairwise interactions between atoms which are defined by the following five terms: a repulsion, hydrophobic, hydrogen bonding, and two terms based on the surface distance between two atoms.

Descriptor-based (Machine Learning based):

10. RF-Score: RF-Score [3] is a scoring function that avoids assuming any linear form and, instead, was created using a Random Forest machine learning model. As input, only the number of contacts, at a 12 Angstrom distance, between pairs of types of ligand (C,N,O,F,P,S,Cl,Br,I) and protein atoms (C, O, N, S), are considered, where hydrogen atoms are ignored. The scoring function was originally trained with PDBBind 2007, but to avoid bias in the model developed in this work, the RF-Score used here has been trained with PDBBind 2013 after removing those structures present in the 2007 version (used in data sets 1 and 2 in this work), as well as the complexes in data set 3.

5.2 Descriptors

There can be several characteristics extracted from protein-ligand interactions, from the perspective of geometry, interaction, structure, surface, energy and beyond. Frequently, exhaustive studies are carried out to find more descriptors in order to better achieve an understanding of the interaction protein-ligand. Hence, the necessity to find tools to better understand their behavior in big sets of protein-ligand complexes.

In this work, besides the values obtained with the different scoring functions, a set of descriptors for the protein-ligand complexes are also used as input. These descriptors characterize the interactions between the protein and ligand both from geometric and energetic points of view. Examples of the first type are the number of contacts between ligand and protein atoms, or the number of hydrogen bonds, while examples of the second are the van der Waals interaction energy or the binding GB (Generalized Born) solvation energy. Descriptors like these and others, are the ones used when building the different types of scoring functions, and they have also been used, not only to analyze and characterize the binding between a protein and ligand, but also to derive or improve machine learning scoring functions, as seen in [11].

The descriptors used in this work have been obtained with the software BINANA v1.2.0 [10], for the structural-interaction descriptors, and with the Prime [7] tool prime_mmgbsa, which calculates the MM-GBSA scoring function used in this work as well as the individual terms (see the VSGB 2.0 model

[7] Schrödinger Release 2016-1: Prime, version 4.3, Schrödinger, LLC, New York, NY, 2016

[24], for a detailed description of each term) which we use as descriptors.

The structural-interaction descritors are:

1. 2.5_contacts: Number of pairs of protein-ligand atoms which are at a distance of 2.5 A or less in the complex structure.
2. 4.0_contacts: Number of pairs of protein-ligand atoms which are at a distance of 4.0 A or less in the complex structure.
3. backbone_alpha: Number of protein backbone atoms (named CA, C, O or N) within 4.0 A of any ligand atom, and corresponding to a residue belonging to an alpha helix.
4. backbone_beta: Same as backbone_alpha, but the residue belongs to a beta sheet.
5. backbone_other: Same as backbone_alpha, but the residue does not belong neither to an alpha helix nor to a beta sheet.
6. sidechain_alpha: Number of nonbackbone protein atoms (those heavy atoms with names different from CA, C, O or N) within 4.0 A of any ligand atom, and corresponding to a residue belonging to an alpha helix.
7. sidechain_beta: Same as sidechain_alpha, but the residue belongs to a beta sheet.
8. sidechain_other: Same as sidechain_alpha, but the residue does not belong neither to an alpha helix nor to a beta sheet.
9. ligand_HBD: Number of hydrogen bonds where the donor is a ligand atom.
10. receptor_HBD: Number of hydrogen bonds where the donor is a receptor (protein) atom.
11. hydrophobic: Number of hydrophobic contacts between the protein and the ligand. These are the subset of 4.0_contacts where both the ligand and the protein atoms are carbon atoms.
12. pi-cation: Number of cation-pi interactions between the ligand and the protein, that is, interactions between positively charged functional groups and aromatic rings.
13. pi-pi: Number of pi-pi interactions between the ligand and the protein, that is, interactions between ligand and protein aromatic rings where these rings adopt a stacked conformation.
14. pi-t: Number of T-stacking interactions between the ligand and the protein, that is, edge-face interactions between ligand and protein aromatic rings.
15. salt_bridge: Number of salt bridges formed between the ligand and receptor.
16. rotatable bonds (TORSDOF): Number of ligand rotatable bonds.

The Prime MM-GBSA energy terms (all values are in $kcal/mol$) are:

1. MMGBSA_dG_Bind_Coulomb (Coulomb): The contribution from the Coulomb term to MMGBSA_dG_Bind.
2. MMGBSA_dG_Bind_Covalent (Covalent): The contribution from the covalent (that is, bonded) term to MMGBSA_dG_Bind.
3. MMGBSA_dG_Bind_Hbond (Hbond): The contribution from the hydrogen-bonding term to MMGBSA_dG_Bind.
4. MMGBSA_dG_Bind_Lipo (Lipo): The contribution from the lipo term to MMGBSA_dG_Bind.

5. MMGBSA_dG_Bind_Packing (Packing): The contribution from the pi-pi packing energy term to MMGBSA_dG_Bind.

6. MMGBSA_dG_Bind_SelfCont (SelfCont): The contribution from the self-contact term (a special case of hydrogen-bonding term) to MMGBSA_dG_Bind.

7. MMGBSA_dG_Bind_Solv_GB (GB): The contribution from the Generalized Born electrostatic solvation term to MMGBSA_dG_Bind.

8. MMGBSA_dG_Bind_vdW (vdW): The contribution form the van der Waals term to MMGBSA_dG_Bind.

9. Ligand_Strain Energy (Ligand_Strain): The strain energy of the ligand due to its change in conformation from its minimized structure in solvation to the actual bound structure in the complex.

5.3 Case-study datasets

In order to design our case-study and assess the methodologies proposed, we used the PDBbind databases corresponding to the versions 2007 and 2013 of PDBbind.

Cheng et al. obtained refined sets containing high-quality 3D structures of 1300 protein-ligand complexes from the 2007 and 2013 version of PDBbind [7, 25]. From these sets, the curators of PDBbind built two new test sets with 195 unique protein-ligand complexes, mainly intended to be used for benchmarking docking and scoring systems. These sets are called core sets. From all these sets we obtained three datasets to use as a case-study, in this work.

From the version 2007, we derived two datasets:

1. *Dataset 1*: Based on the refined set of 2007, with 828 protein-ligand complexes. Used in this study as training set.

2. *Dataset 2*: Based on the core set of 2007, with 180 unique protein-ligand complexes. It is important to mention that the complexes used for this dataset are completely disjointed from the *Dataset 1*, although both contain proteins with similar characteristics. Hence, is mainly used for benchmarking our models.

In order to further assess our models, from the version 2013, we obtained another dataset:

3. *Dataset 3*: Based on the core set of 2013, with 60 representative complexes. This set contains proteins with slightly different characteristics from the *Dataset 1* and *Dataset 2*. And it is used further asses the framework under different circumstances.

The protein-ligand complexes for each dataset, were prepared using Schrödinger's Protein Preparation Wizard (PrepWizard) [8], by adding hydrogens, assigning disfulfide bonds, removing waters further than 5 Armstrong from the ligand, adjusting charges, adding missing side-chains and optimizing hydrogen bond clusters.

Once the protein-ligand complexes were prepared, for the three datasets we calculated for each protein-ligand complex, the already described ten different SFs (MM-Gbsa, Autodock VINA, XScore::HMS, XScore::HSS, XScore::HPS, XScore::Average, RFScore, DSX, Glide SP and Glide XP). Additionaly, for each protein-ligand complex, we extracted 16 structural-interactions descriptors, via the software BINANA v1.2.0 and 9 Energy-based descriptors, via prime_mmgbsa tool, as previously explained in section 5.2.

In this manner, for each dataset, we obtained three new datasets for structural-interaction based descriptors, energy-based descriptors and SFs, as explained below:

[8] Schrödinger Suite 2016-1 Protein preparation wizard

1. *Dataset 1 with SFs*: Conformed by 828 different protein-ligand complexes, each one described by a 10-dimensional vector of SFs.

2. *Dataset 1 with structural- interaction based descriptors*: Conformed by 828 different protein-ligand complexes, each one described by a 16-dimensional vector of structural-interaction based descriptors.

3. *Dataset 1 with energy based descriptors*: Conformed by 828 different protein-ligand complexes, each one described by a 9-dimensional vector of energy based descriptors.

4. *Dataset 2 with SFs*: Conformed by 180 different protein-ligand complexes, each one described by a 10-dimensional vector of SFs.

5. *Dataset 2 with structural- interaction based descriptors*: Conformed by 180 different protein-ligand complexes, each one described by a 16-dimensional vector of structural-interaction based descriptors.

6. *Dataset 2 with energy based descriptors*: Conformed by 180 different protein-ligand complexes, each one described by a 9-dimensional vector of energy based descriptors.

7. *Dataset 3 with SFs*: Conformed by 60 different protein-ligand complexes, each one described by a 10-dimensional vector of SFs.

8. *Dataset 3 with structural- interaction based descriptors*: Conformed by 60 different protein-ligand complexes, each one described by a 16-dimensional vector of structural-interaction based descriptors.

9. *Dataset 3 with energy based descriptors*: Conformed by 60 different protein-complexes, each one described by a 9-dimensional vector of energy-based descriptors.

For the sake of simplicity, we will refer to them as *Dataset 1*, *Dataset 2* and *Dataset 3* along this report, and additionally we will mention the kind of features they encompass.

6 Methodology

Characterization of protein-ligand interactions can involve different experimental and computational methods, that result in different sets of descriptors and SFs. In this work, we are interested in creating a framework to obtain the best combination of descriptors as well as SFs in order to not only improve the overall binding affinity, but also provide methods to understand and interpret the behavior of their interactions, in such a way that, in a consistent manner, we can filter the best descriptors and SFs. This scheme is also planned to enhance a single or a set of different SFs with a set of new descriptors.

In this section, we present the machine learning and statistical methods used along this work, likewise our main motivations. Section 6.1 describes the theory behind the penalized linear regression methods, and our implementation to discover potential sets of features with LASSO and Elastic Net, and to perform ridge regression for stacking purposes. In order to add flexibility to our framework, we investigated semi-parametric and non-parametric techniques, detailed in Section 6.2. With regard to the improvement of SFs by adding new descriptors, we present our stacking procedure in Section 6.3. Finally, Section 6.4 presents the framework constructed, involving all the previous exposed methods.

In order to allow a better understanding on the methods explained in this section, we carry out a running example, using the *Dataset 1* with the SFs.

For the purposes of coherence, in this section, we refer to $\mathbf{X}$ as the n x p matrix of predictors (independent variables) with n observations and p variables. The variables are expressed as X_j, $j = 1, ..., p$ whereas the observations as x_i, $i = 1, ..., n$. The dependent variable is represented as Y, with $y_1, y_2, ...y_n$ values.

6.1 Penalized Linear Regression Methods

Datasets involving different descriptors and/or Scoring Functions tend to suffer from multicollinearity, therefore in this study we look forward to find models with the best trade-off among accuracy, variable selection and interpretability. Penalized linear regression methods fit linear regression models, at the same time that they can select the strongest subset of variables, providing more understandable models with better predictions.

Considering the following usual linear regression model with a predictor matrix $\mathbf{X}$ with n observations and p variables, a dependent variable $Y \in \mathbb{R}^n$ and i.i.d. errors of the form $\boldsymbol{\epsilon} = \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$:

$$Y = E(Y|\mathbf{X}) = \beta_0 + \sum_{j=1}^{p} X_j\beta_j + \epsilon_j \tag{5}$$

The common linear regression model assumes that the conditional expectation $E(Y|\mathbf{X})$ is linear and therefore, we only seek to estimate the parameters β from the data by minimizing the Residual Sum of Squares (RSS):

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 \tag{6}$$

Penalized regression for linear models, instead, solve the constrained minimization problem expressed in equation 7, for estimating the parameters β.

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 + \lambda R(\beta) \quad \text{where } R(\beta) \text{ is the penalization term} \tag{7}$$

The penalization term $R(\beta)$ in the above equation can be either the $L_1 = \sum_{j=1}^{p} |\beta_j|$ norm for LASSO regression, the $L_2 = \sum_{j=1}^{p} \beta_j^2$ norm for Ridge regression and both of them for Elastic Net. Although

Ridge regression presents a stable method which shrinks coefficients, it does not set coefficients to zero and hence it does not allow to obtain a smaller subset of variables. Consequently, in this work, for filtering the best subset of descriptors and SFs we only consider penalized techniques that allow selection of variables, i.e. LASSO and Elastic Net, and for performing the stacking procedure of SFs with a new model of descriptors, we use Ridge regression.

6.1.1 Ridge Regression

Based on the premise that each SF brings something different for each complex, and that when we aim at stacking SFs with new potential descriptors, the SFs have similar performance, but they involve different theory and principles, and they do not already encompass the descriptors, we seek to take advantage of this knowledge, in a consistent, yet simple manner, according to their performance. For this situation, and in order to not add more complexity to our scheme, Ridge regression represents a stable and robust model that penalizes the size of the regression coefficients. Specifically, the ridge regression estimates the parameters β by solving the following constrained optimization problem with the residuals sum of squares:

$$\hat{\beta}_R = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{8}$$

In Equation 8, λ represents the shrinkage coefficient, in other words, how much we seek to shorten the coefficients. In order to avoid bias, we selected this term by means of 10-Fold Cross Validation. It is important to mention that we did not choose the lambda that minimizes the Cross Validation (CV) error, but rather we used the value of λ that minimizes the CV error plus one standard error ($\lambda 1.se$). We often use the one-standarderror rule when selecting the best model; this acknowledges the fact that the risk curves are estimated with error, so it errs on the side of parsimony [13]. It means that the best model (with the minimum λ) may be too complex and slightly overfitted. Therefore, the simplest model, which has comparable error to the best model, is within one standard error of the minimum. To exemplify this, in Figure 1, we take as an example *Dataset 1* with the SFs. On the left part, it exhibits the selection of the parameter λ, via 10-Fold cross validation, showing two dotted lines, corresponding to the complex model chosen with the lambda that minimizes the CV error and the simpler model selected with the λ that minimizes the CV error plus one standard error. On the right part, we can appreciate how the coefficients were shrink with the $\lambda 1.se$.

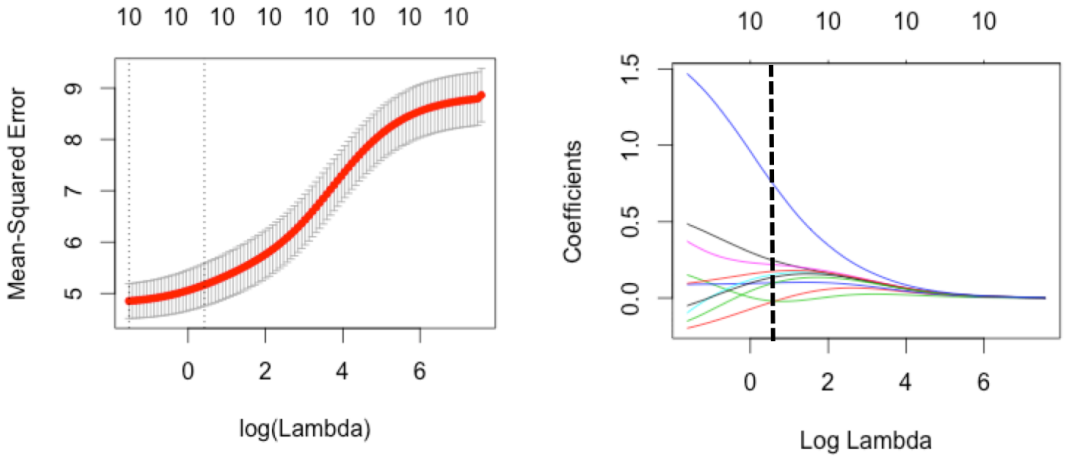

Figure 1: From left to right. a) Selection of the λ parameter, via 10-Fold cross validation. The dotted lines represent the value of $\log(\lambda)$ that minimizes the CV error, and the value of $\log(\lambda.1se)$ that minimizes the CV error plus one standard error. At the top it exhibits, the number of variables selected with respect to $\log(\lambda)$ (in this case all of them will always be selected) and, in the vertical axis, its corresponding Mean Squared Error. b) Ridge shrinkage of coefficients over the SF variables in *Dataset 1*. Each curve represents the regularization path of a variable as a function of the log scaled parameter λ. The dotted line represents the model selected for $\log(\hat{\lambda}.1se) = -0.719$ ($\hat{\lambda}.1se = 0.487$)

6.1.2 Least Absolute Shrinkage and Selection Operator (LASSO)

We aim to also combine, separately, different descriptors and SFs for protein-ligand systems, and select subsets of strong variables. These can result in datasets with several redundant and/or not relevant features. Additionally, many descriptors can be involved as well as SFs, and using wrapping methods for selecting features, can lead to highly computational costs. Therefore, we seek to perform variable selection as a part of the learning procedure, yet keeping intelligibility. To this end, we employed LASSO, which performs penalized least squares procedure minizing the RSS subject to the non-differentiable constrain expressed in terms of the L_1 norm of the coefficients. Because of the nature of this constraint it tends to produce some zero coefficients and hence give both a subset of features and an intelligible stable model. The objective function to find the most adequate β parameters is, therefore, modified as follows:

$$\hat{\beta}_L = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{9}$$

In equation 9, $\lambda \geq 0$ is the parameter that controls the shrinkage of the coefficients, i.e. the amount of regularization that is applied to the estimates. If this parameter is set to 0, then the regular Ordinary Least Squares is performed with no penalization. In order to avoid overfitting, we adjusted the regularization parameter λ via 10-Fold CV, choosing the λ that minimizes the CV error plus one standard error, as explained in the Ridge regression method. Figure 2, shows this implementation applied to *Dataset 1* with SF variables, and its shrinking effects.

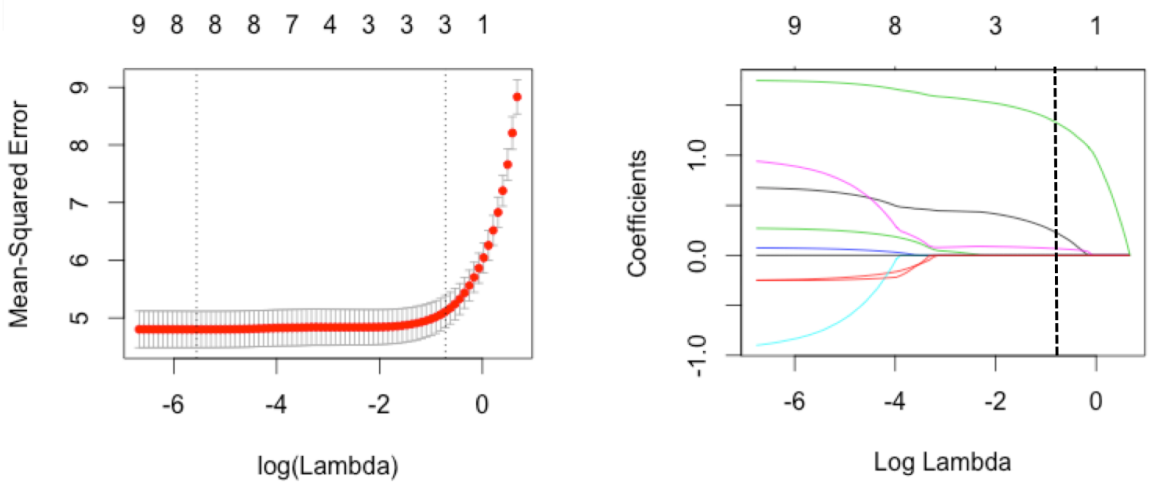

Figure 2: From left to right. a) Selection of the λ parameter, via 10-Fold cross validation. The dotted lines represent the value of $\log(\lambda)$ that minimizes the CV MSE, and the value of $log(\lambda.1se)$ that minimizes the CV MSE plus one standard error. At the top it exhibits, the number of variables selected with respect to the values of $\log(\lambda)$, and in the vertical axis its corresponding MSE. b) LASSO shrinkage of coefficients over the SF variables in *Dataset 1*. Each curve represents the regularization path of a variable as a function of the log scaled parameter λ. The values at the top exhibit the number of variables selected with respect to the value of $\log(\lambda)$. The dotted line represents the model selected for $\log(\hat{\lambda}.1se) = -0.211$ ($\hat{\lambda}.1se = 0.809$)

6.1.3 Elastic Net

Protein-ligand complexes can be characterized by specific groups of descriptors encompassing interactions, structures, geometries and energy terms, yielding to groups of variables with highly pairwise correlations. The same applies for sets with only Scoring Functions as features. In these situations, LASSO tends to select only one variable from the group without taking into consideration which one is the best. Elastic Net addresses this shortcoming by penalizing with both the L_1 and L_2 norms. This has the effect to effectively shrink the coefficients (as in ridge regression) and setting some coefficients to zero (as in LASSO) [41]. The objective function to estimate the parameters β via Elastic Net can be, therefore, expressed as:

$$\hat{\beta}_{EN} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^{p} \frac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j| \tag{10}$$

In order to make a fairly trade-off between both regularization terms, we set up $\alpha = 0.5$ and by means of 10-Fold Cross Validation we estimated the best value for $\lambda.1se$, as explained for the above methods. Following the running example, Figure 3 illustrates this procedure, applied to *Dataset 1* with only SFs as variables.

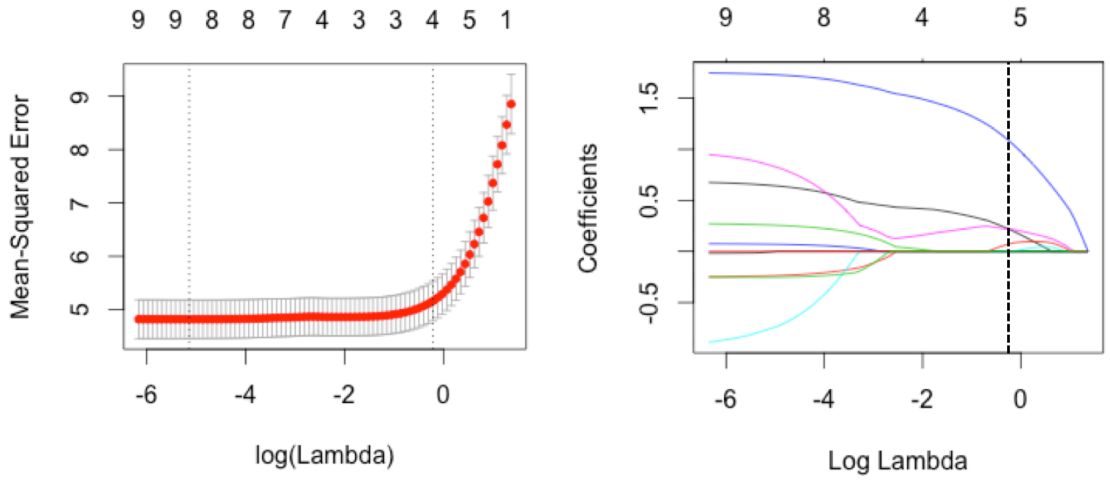

Figure 3: From left to right. a) Selection of the parameter λ, when $\alpha = 0.5$, via 10-Fold cross validation. The dotted lines represent the value of $log(\lambda)$ that minimizes the CV MSE, and the value of $log(\lambda.1se)$ that minimizes the CV MSE plus one standard error. At the top it exhibits, the number of variables selected with respect to the value of $log(\lambda)$, and, in the vertical axis, its corresponding Mean Squared Error. b) Elastic Net shrinkage of coefficients over the SF variables in *Dataset 1*. Each curve represents the regularization path of a variable as a function of the log scaled parameter λ. The values at the top exhibit the number of variables selected with respect to the value of $log(\lambda)$. The dotted line represents the model selected for $log(\lambda.\hat{1}se) = -0.211$ ($\lambda.\hat{1}se = 0.809$)

6.2 Non Parametric and semi-parametric Regression Methods

In the former regularization methods we were assuming a linear relationship among the predictor variables and the response one. Even though this assumption might be adequate for combining SFs with a good trade-off between results and computational costs, the same assumption may not be suitable for modeling the descriptors. What is more, it can be the case in which a SF performs bad for certain intervals of the binding affinity but for others outperforms to any other SF. This behavior cannot be modeled by assuming uniquely linear relationship. For instance, in figure 4, we can see the scatterplot of the estimated binding affinity of the XScore::HPS SF with the experimental binding affinity. A simple linear model fits a straight line through the set of points. We can realize, that with this fit we are missing important information. Therefore, in this section, we investigate more flexible nonparametric and semi parametric techniques that avoid strong parametric assumptions, yet allow interpretation, such as General Additive Models with cubic smoothed splines and Kernel-based Regularized Least Squares.

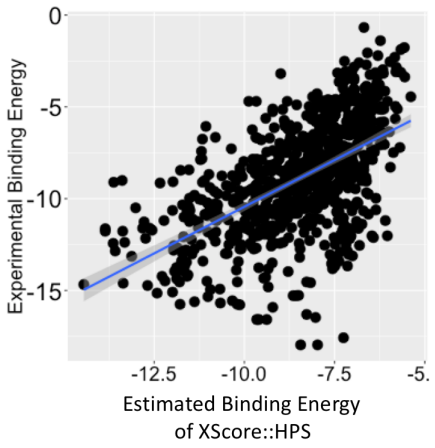

Figure 4: Scatterplot of the estimated binding affinity of the XScore::HPS SF with the experimental binding affinity. A simple linear model fits a straight line through the set of points.

6.2.1 Generalized Additive Models (GAM)

Standard linear regression models assume a linear relationship between the covariates $X_1, X_2, ..., X_p$ and the dependent variable Y. This premise can lead to a poorly performance when the underlying relationship is not strictly linear in the whole observations, but rather with some variations. The idea behind the Generalized Additive Models (GAM) is to replace the linear part $\sum_{j=1}^{p} \beta_j X_j$ from equation (5) by a sum of smoothing functions $\sum_{j=1}^{p} f_j(X_j)$ where the nonparametric function $f_j(X_j)$ can be estimated in a flexible manner using a cubic spline smoother [19]. The general form of this method, can be written as

$$g(E(\mathbf{Y}|\mathbf{X})) = \beta_0 + \sum_{j=1}^{p} f_i(X_i) \tag{11}$$

In statistics, y_i is usually measured with noise, and it is generally more useful to smooth the x_i, y_i points, rather than interpolating them [40]. A spline is a function that is piecewise-defined by polynomial functions and have high degree of smoothness at the places where the polynomial pieces connect, known as knots. A cubic spline, is a spline constructed of piecewise third-order polynomials which pass through a set of m control points (knots). This produce an interpolated function that is continuous through to the second derivative [32]. Splines tend to be stabler than fitting a polynomial through a set of n points, with less possibility of wild oscillations between the set of points.

The flexibility of this model, allows to explore and visualize the relationship between each predictor X_i and the response variable Y by means of the scatterplot smoothers. This is an important aspect in this work for two main reasons. First, it allows to visually detect irregularities in a SF when it does not have a good performance for some ranges of the binding affinity; secondly it permits to better understand the role of each descriptor with respect to the free binding energy.

Using nonparametric functions arises the concern of increasing bias. To address possible overfitting issues, we used penalized cubic regression splines. The penalization refers to the fact that the spline has its penalty modified to shrink towards zero at high enough smoothing parameters. This means that the smoothing parameter estimation, that is part of fitting, can completely remove terms from the model [27], allowing in some sense to get a subset of important features. The parameters of each spline are estimated via Generalized Cross-Validation (GCV).

Additionally to the penalty already used, a naïve approach is considered to further improve the variable selection, in which we obtain the approximate significance of the smooth terms and we remove those from which p-value is less than 0.05, afterwards we re-run the model excluding those variables.

Following our running example, figure 5 exhibits the scatterplot smoothers for each SF variable of the *Dataset 1*.

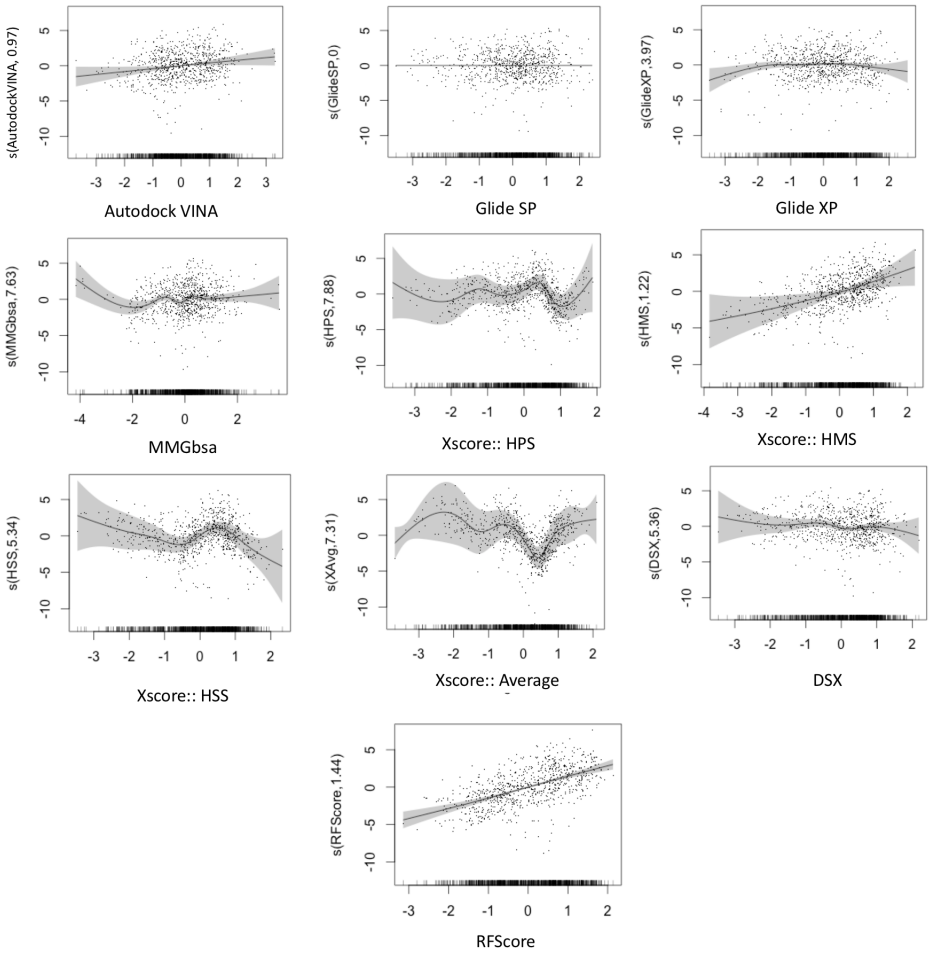

Figure 5: GAM predicted cubic smooth splines of the Experimental binding affinity as a function of the 10 SF variables of the *Dataset 1*: Autodock VINA, Glide SP, Glide XP, MMGbsa, XScore::HPS, XScore::HMS, XScore::HSS, XScore::Average, DSX and RFScore. The degrees of freedom are in the parenthesis on the y-axis. The gray areas represent the confidence intervals of the smooth splines. The thick marks in the x-axis indicate the distribution of the observations.

6.2.2 Kernel-based Regularized Least Squares

The Kernel-based Regularized Least Squares (KRLS) method was proposed by Hainmueller et al. [18] as a flexible scheme to model social science inquiries. The idea behind this model is to fit multidimensional functions $Y = f(\mathbf{X})$ without relying on linearity or additive assumptions. KRLS finds the best fitting function by minimizing the squared loss of a Tikhonov regularization problem (L_2 norm), using Gaussian kernels as radial basis functions.

The KRLS searches over a space of functions H and chooses the best fitting function f by minimizing the squared loss adding the L_2regularization term:

$$\underset{f \in H}{\operatorname{argmin}} \sum_{i}^{n} (y_i - f(x_i))^2 + \lambda ||f||_H^2 \tag{12}$$

where $\sum_i^N (y_i - f(x_i))^2$ is the loss function and $||f||_H^2$ measures the complexity of the function according to the L_2 norm, $||f||^2 = \int f(X)^2 dX$. λ is the regularization parameter that controls the trade-off between model fit and complexity. The function that minimizes the regularized loss is defined by means of a positive semidefinite Gaussian kernel function $k(x_i, x_l) = \exp \frac{-||x_i - x_l||^2}{\sigma^2}$ for measuring the distance

between two observations, and c_i is the 'weight' for each observation i:

$$f(x_l) = \sum_{i}^{n} c_i k(x_i, x_l) \tag{13}$$

The essential idea behind this approach is that it does not model y_i as a function of x_i. Instead, it takes advantage of the information about the similarity between observations. If we consider, for instance, that we want to evaluate this function for a new test point x^*, given that the model has already been trained and the parameters have been fixed, the predicted value is given by

$$\begin{aligned} f(x^*) &= c_1 k(x*, x_1) + ... + c_N k(x^*, x_n) \\ &= c_1(\text{similarity of x* to } x_1) + ... + c_n(\text{similarity of x* to } x_n) \end{aligned} \tag{14}$$

Expressly, the outcome is linear in the similarities of the test point to each observation, in such a way that the closer x^* is from another x_i point, the greater the impact of x_i on the predicted $f(x^*)$.

The regularization parameter λ is chosen by using Leave-One-Out Cross Validation (LOOCV), minimizing the sum of the squared leave-one-out errors, to avoid overfitting. In this procedure, the role of the kernel is principally as a measurement decision incorporated in the kernel to ensure that it is carrying useful information from the data $\mathbf{X}$. This contrast with the common Kernel-regression approaches in which the Kernel bandwidth is basically the only smoothing parameter. Therefore, the kernel bandwidth is set by default to $\sigma^2 = dim(\mathbf{X})$ to not further add more computational costs.

One of the main advantages of this scheme is that it allows to estimate the expectation of Y conditional on $\mathbf{X} = x$ via plots. For each plot, the predictor of interest varies from its 1st to its 3rd quartile values, while holding the other predictors fixed. These kind of interventions are very convenient to discover further possible interactions among the predictors and the target variable. To exemplify this, figure 6, shows these plots for our running example of the SFs.

Our main motivation of using this method is for including further statistical parameters to our model, in order to detect depth details and changes, while combining SFs and descriptors, separately.

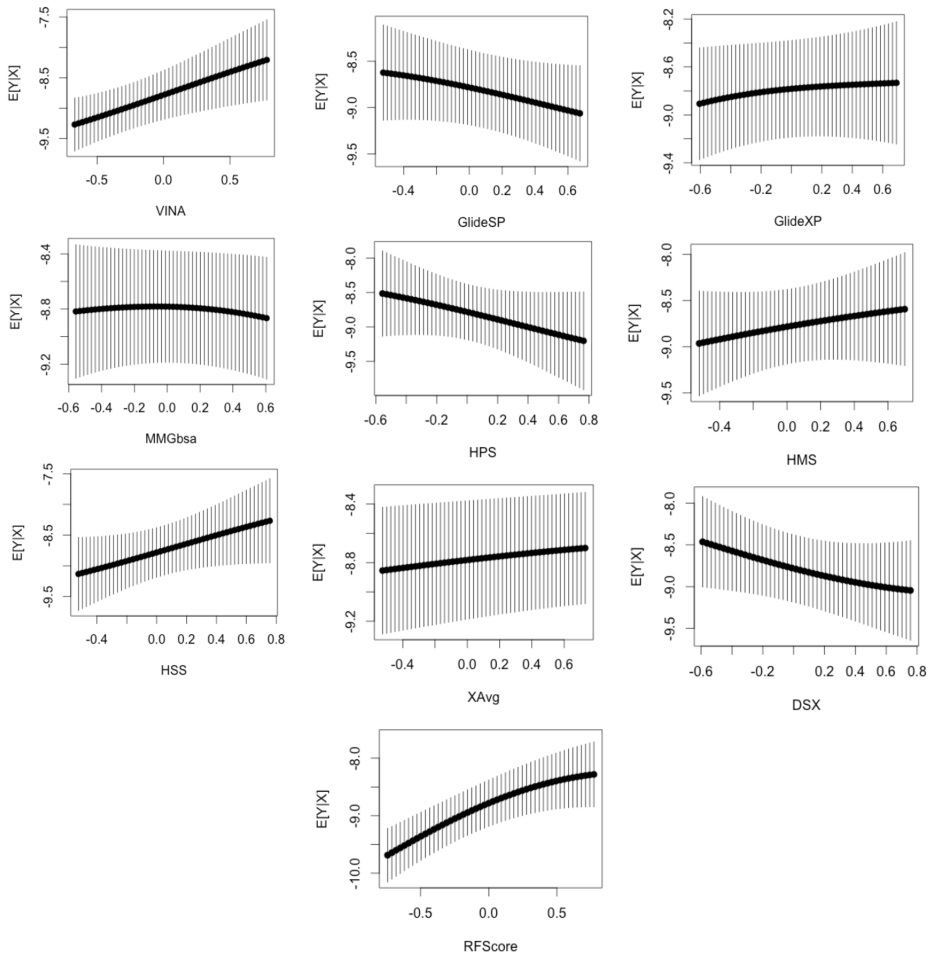

Figure 6: Estimates of the conditional expectation functions of $E[Y|X]$ for every SF in the *Dataset 1*. Horizontal axis shows the values of the predictor from its 1st to its 3rd quartile values, whereas the vertical axis shows its correspondent expected value. Vertical lines over the expectation line, represent the confidence band.

6.3 Stacking Methods

Frequently, new descriptors and terms emerge to characterize protein-ligand systems, hence new SFs need to be calculated to include these terms. We believe that we can take advantage of previous SFs developed, encompassing other different characteristics, by adding new emerging descriptors. Based on this premise, we capitalize on stacking techniques, with the purpose of enrich a single or a set of SFs with new descriptors.

Stacking is a technique in which the predictors of a collection of models are given as inputs to a second-level learning algorithm. This second-level algorithm is trained to combine the model predictions optimally to form a final set of predictors [34]. Stacked generalization is a technique introduced by Wolpert [39] to combine estimators, deducing the biases of the models by using LOOCV. The idea is to estimate the parameters $\hat{\beta}$ of m different models h_j, $j = 1, ..., m$, by solving the following optimization task

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{m} \beta_j h_j^{(-i)}(x_{i,j}) \right)^2$$
$$\text{where } h_j^{(-i)} \text{ is the leave-one-out estimate} \tag{15}$$

In this study, we address the two following different situations:

1. Combine a set of SFs with a set of new potential descriptors
2. Combine a single SF with a set of new potential descriptors

Following the scheme presented in figure 7, for the first case (1), we define each SF_j, $j = 1, ..., p$, as a variable. In order to take as much advantage as possible from their underlying knowledge, we combined them by using the methods LASSO, Elastic Net, GAM and KRLS and then we choose the one with the best performance to obtain the 1^{st} level predictors for the SFs, namely, $h_1(X)$. With the same rationale, we combine each descriptor D_j, $j = 1, ..., p$, to obtain the 1^{st} level predictors for the descriptors, $h_2(X)$. Finally, we learned the 2^{nd} level algorithm by blending $h_1(X)$ and $h_2(X)$, to obtain the final set of predictors.

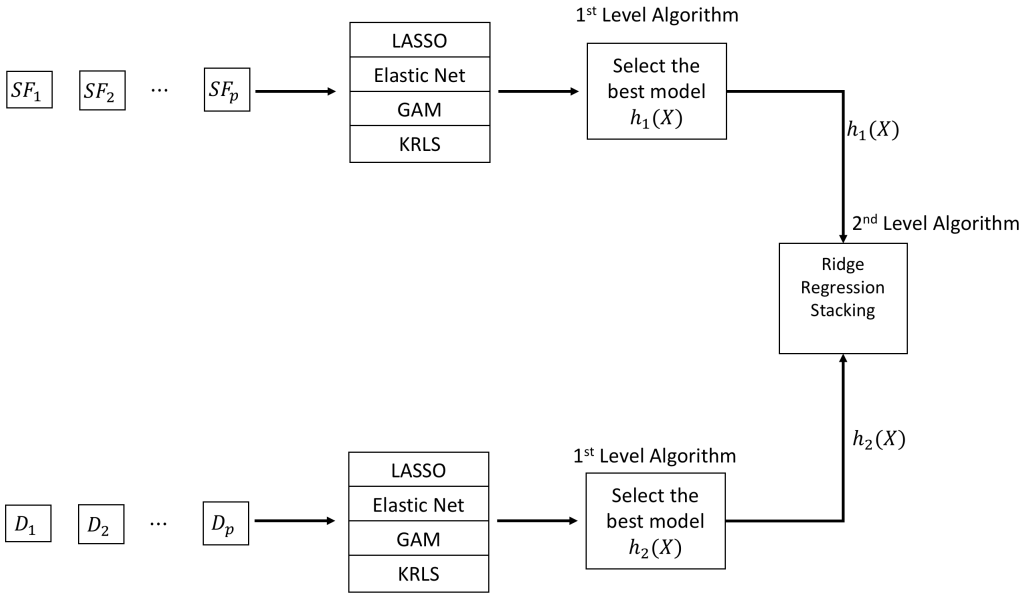

Figure 7: Ridge Regression Stacking scheme, with SF_j, $j = 1, ..., p$ different SFs and D_j, $j = 1, ..., p$ different descriptors.

Based on the premise that we are already covering distinctive methods to obtain the best combination of SFs and descriptors, separately. It only suffices to use Ridge regression for stacking purposes, instead of the stacked generalization. Hence, we modified the objective function stated in equation 15, for stacking, as follows:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{m} \beta_j h_j(x_{i,j}) \right)^2 + \lambda \sum_{j=1}^{m} \beta_j^2$$

where the parameter λ is estimated via 10 Fold CV and m=2 (16)

As established in the scheme presented in figure 8, for the second situation (2), we define the single SF as the 1^{st} level of predictors, $h_1(X)$. For the descriptors D_j, $j = 1, .., p$, we proceed as in the former case. Finally, we blend each new learned model ($h_1(X)$ and $h_2(X)$) by means of ridge regression stacking.

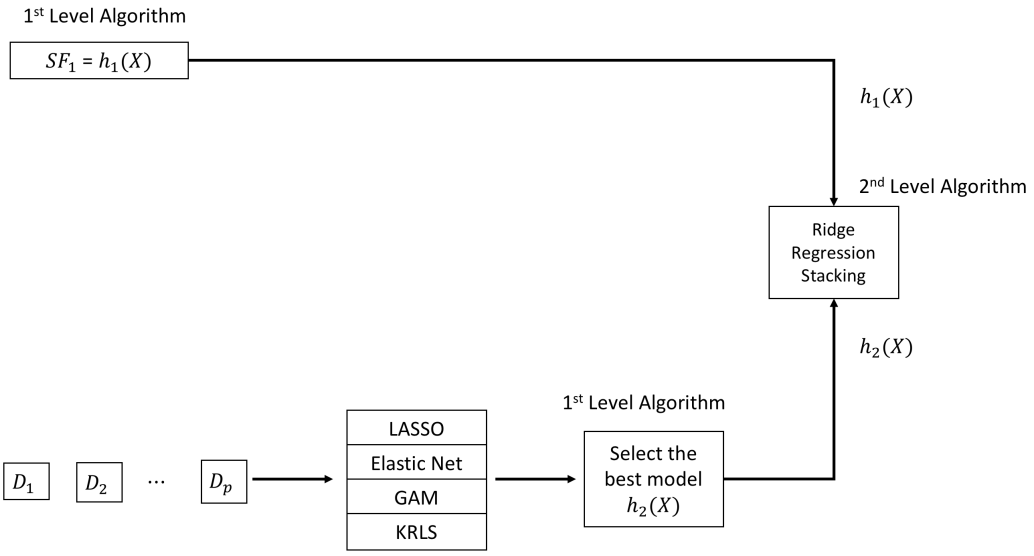

Figure 8: Stacked Regression with a single SF and D_j, $j = 1, ..., p$ different descriptors.

It is necessary to remark that this methodology is based upon two important assumptions:

1. The set of SFs has been chosen adequately, with SFs underlying different theories and principles.
2. The set of SFs used do not already encompass the new set of descriptors.

6.4 Framework construction and specifications

Heretofore, we have presented the theory and motivations behind all the methods used in this work. In this section, we explain how we implement them in order to achieve our goals and to provide a flexible framework capable of being used for future studies in this field.

For simplicity reasons, the framework is outlined in terms of the two main goals of this study:

1. Exploit different machine learning and statistical techniques to assess the combination of different SFs and descriptors, separately. Provide tools to detect the possible subset of most significant variables, and analyze the independent interaction of each descriptor and SF with the protein-ligand binding affinity from different perspectives.
2. Enrich a single or a set of SFs with a set of new descriptors.

With the aid of the diagram shown in figure 9, we are going to describe the workflow for the first objective, as follows:

1. We retrieve the SFs and Descriptors datasets separately. There can be three sets specified for SFs and descriptors: training, validation and new data. Validation and new data are optional. If not validation dataset is provided, we split the training dataset into: 70% for training and 30% for validation, using stratified sampling.
2. Z-score standardization $\left(X_j^z = \frac{X_j - \mu(X_j)}{\sigma(X_j)}\right)$ is applied to each variable of the dataframes. Some of the methods we are using such as LASSO and Elastic Net require the variables to be standardized to make the computation faster. For the other methods, no harm is done with the standardized variables.
3. With the training dataset, the models are learned. For LASSO and Elastic Net, 10-Fold CV is performed to estimate the parameters, for GAM, GCV is performed, and for KRLS, LOOCV, as specified in the methodology section.

4. With the final models, we evaluate their performance in the validation set. In this part we report the features selected by LASSO and Elastic Net, the scatterplot smoothers obtained from GAM and the Estimates of the conditional expectations functions for each variable from KRLS. Additionally, the performance results for each model are provided in terms of the Pearson correlation R_p and RMSE.

5. If a new data is provided, we give in csv format the predicted values.

Following the diagram in figure 10, we explain the workflow for the second objective, as following:

1. We retrieve the SFs and Descriptors datasets separately. There can be three sets specified for SFs and descriptors: training, validation and new data. Validation and new data are optional. If not validation dataset is provided, we split the training set into: 70% for training and 30% for validation, using stratified sampling. In this case, is allowed to have only one SF in the SFs datasets.

2. Z-score standardization $\left(X_j^z = \frac{X_j - \mu(X_j)}{\sigma(X_j)}\right)$ is applied to each variable of the dataframes.

3. With the training set, the models are learned. For LASSO and Elastic Net, 10-Fold CV is performed for estimate the parameters, ffor GAM, GCV is performed, and for KRLS, LOOCV, as specified in the methodology section.

4. We select the best model for the SFs and the best model for the descriptors via the MSE of each model. In the case of one SF, we pass this as the best model for SFs to the ridge regression stacking.

5. With the two best models for SFs and descriptors, we learn the ridge regression stacking model, via 10-Fold CV.

6. With the final model from stacking, we evaluate its performance in the validation set. In this part, we report the features selected for descriptors and SFs, in the later case only if there is more than one SF. The scatterplot smoothers obtained from GAM and the estimates of the conditional expectation functions from KRLS. Additionally, the performance of each model, including the ridge regression stacking, are provided in terms of the Pearson correlation R_p and RMSE.

7. If a new data is provided, we give in csv format the predicted values.

Finally, on the top of these two schemes, it is allowed to specify the procedure that is desired to execute.

We implemented this framework in R version 3.3.0, using the libraries: glmnet [9] version 1.8-12, caret [10] version 6.0-70, mgcv [11] version 1.8-12 and KRLS [12] version 0.3-7.

[9] https://cran.r-project.org/web/packages/mgcv/
[10] https://cran.r-project.org/web/packages/caret/index.html
[11] https://cran.r-project.org/web/packages/mgcv/mgcv.pdf
[12] https://cran.r-project.org/web/packages/KRLS/index.html

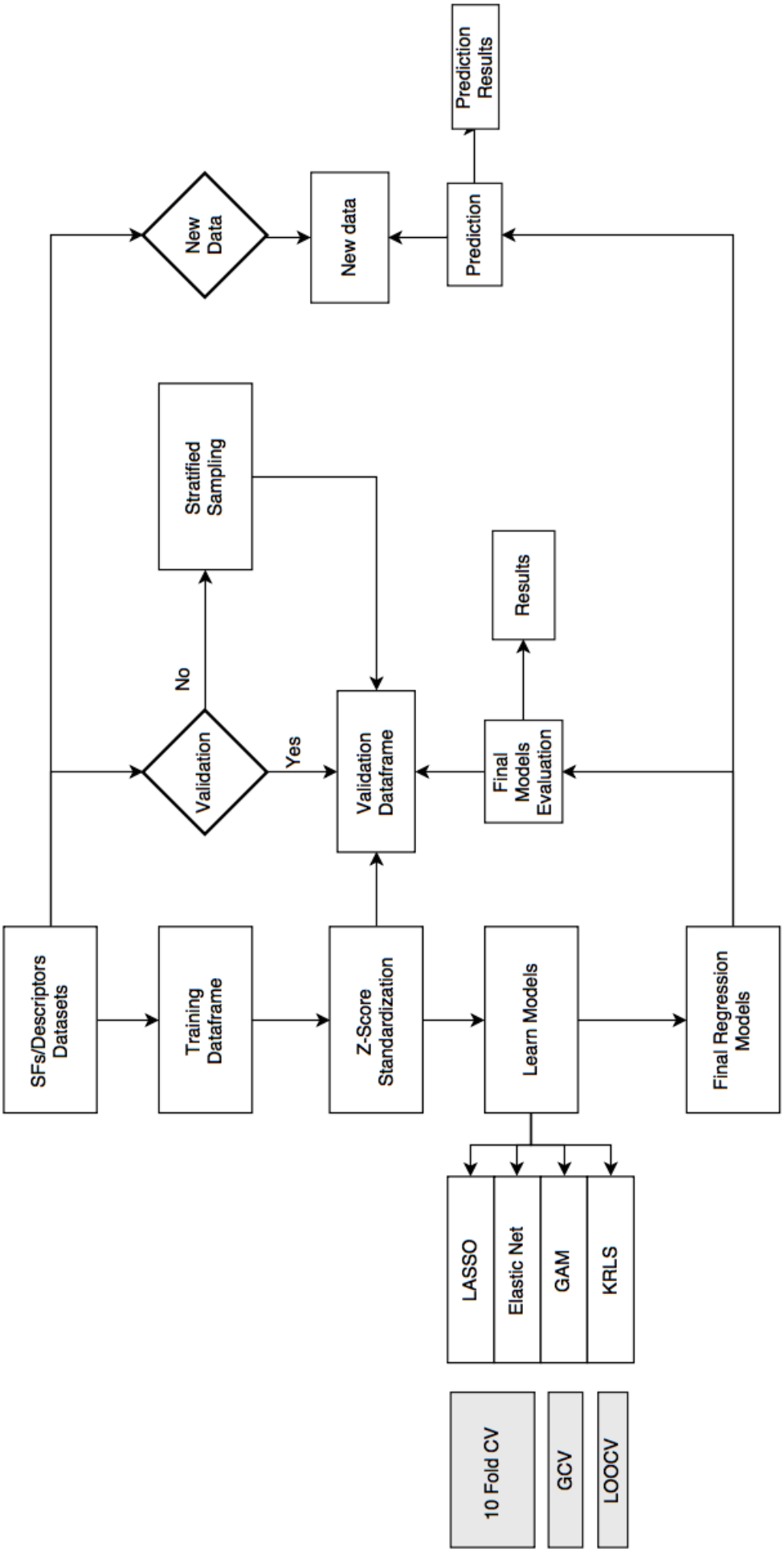

Figure 9: Scheme for combining either different sets of SFs or different sets of descriptors.

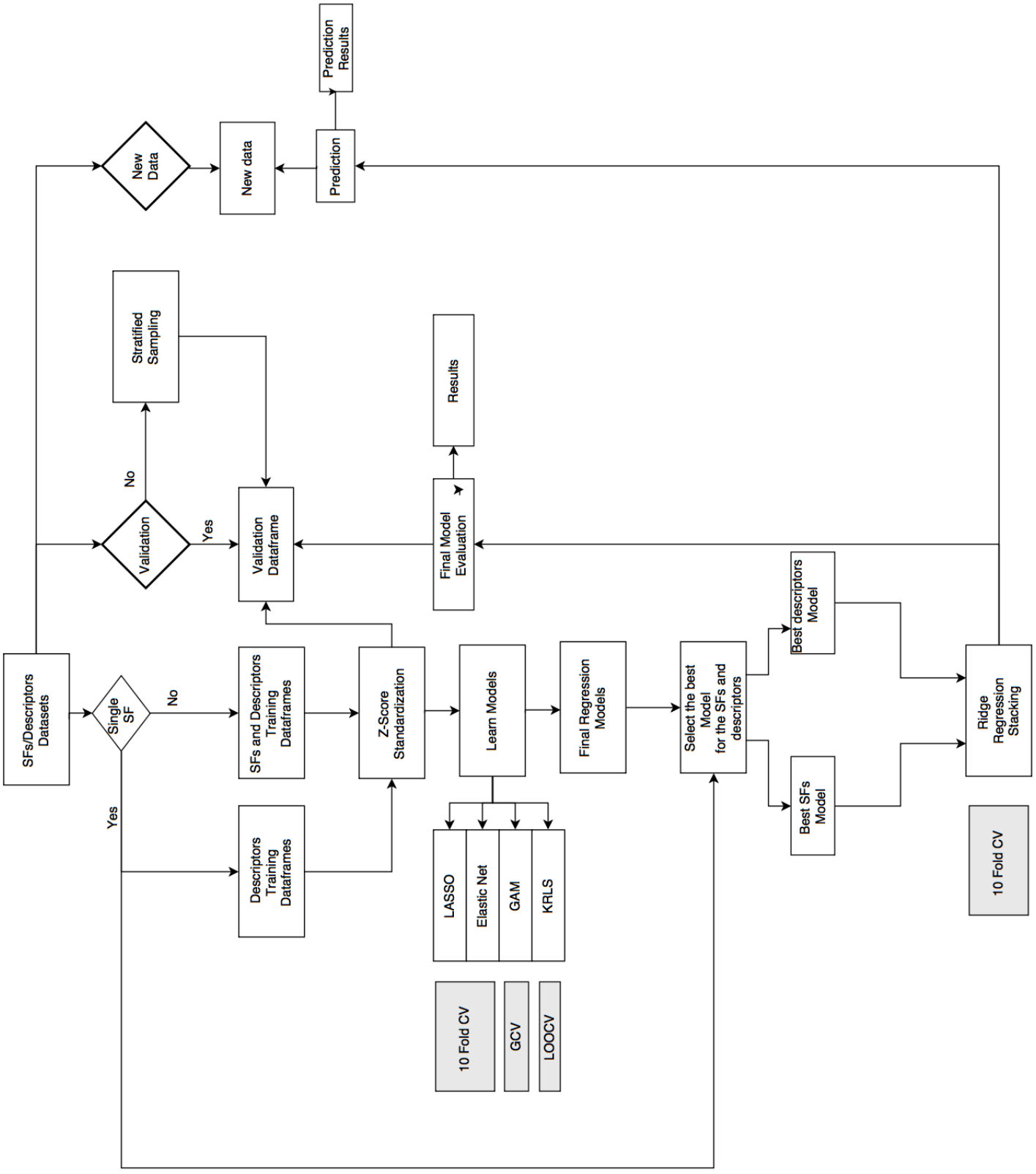

Figure 10: Scheme for enrich either a single or a set of SFs with new descriptors

7 Results

We applied the proposed methodology to the well-studied datasets from the PDBbind database, fully explained in section 5. We recall that *Dataset 1*, from the 2007 version of PDBbind, with 828 different complexes described by 10 different SFs, 16 structural-interaction descriptors and 9 energy-based descriptors, is used as training in this case-study. *Dataset 2*, with 181 different complexes, drawn from the same version, with similar protein families, is used as validation dataset. In order to further examine the overall performance and possible drawbacks while using a set coming from a different year and with different protein families, we also used the *Dataset 3*, with 60 protein-ligand systems, from the 2013 version as another validation dataset. Along this section, we will refer to them as "Dataset 1", "Dataset 2" and "Dataset 3", respectively, mention the variables used.

In order to analyze and compare our results, table 1 presents the assessment of the SFs used during this study, in terms of the Pearson correlation (R_p) and the Root Mean Squared Error (RMSE). For consistency, we are going to evaluate our results with the same metrics. For some SFs, we could not calculate their RMSE, because their outputs do not correspond to the real experimental values of the binding affinity, they only aim at maximizing the correlation with this experimental energy.

Table 1: Evaluation of the SFs employed in this study, in terms of the Pearson Correlation R_p and the Root Mean Squared Error (RMSE)

Scoring Function	**Dataset 1**		**Dataset 2**		**Dataset 3**	
	R_p	RMSE	R_p	RMSE	R_p	RMSE
Autodock VINA	0.516	2.927	0.499	3.292	0.427	3.177
Glide SP	0.392	3.296	0.448	3.232	0.494	3.113
Glide XP	0.219	4.681	0.448	3.921	0.397	3.944
MMGbsa	0.411	-	0.479	-	0.461	-
XScore::HPS	0.564	2.491	0.606	2.672	0.593	2.633
XScore::HMS	0.572	2.454	0.645	2.526	0.605	2.535
XScore::HSS	0.554	2.527	0.622	2.688	0.606	2.656
XScore::Average	0.569	2.458	0.635	2.596	0.609	2.580
DSX	0.526	-	0.584	-	0.551	-
RFScore	0.661	2.260	0.571	2.705	0.636	2.518

7.1 Results of the Models

Formerly, we have described the main machine learning and statistical methods used to combine a diverse kind of descriptors and SFs, in order to estimate the binding affinity and to improve it. In this section, we present the results obtained in our designed case-study, with the different studied models.

Following the workflow of the framework presented in Section 6.4 and the main objectives of this study. First we show the results of the models for descriptors and SFs separetely (Section 7.1.1 and Section 7.1.2). Afterwards, in Section 7.1.3 we exhibit the results of stacking the best model of the de-

scriptors with SFs that do not involve them in their computation. Finally, in Section 7.1.4, we present a particular case, in which we stack the best model of descriptors with the RFScore SF. Additional results are provided in order to emphasize the motivations of using this framework.

7.1.1 Results of the models for the descriptors

Regarding the descriptors, in this work, we used two different categories: 1) structural-interaction descriptors and 2) energy-based descriptors. We assessed and analyzed their individual and jointly performance and contribution to estimate the binding free energy. In table 2, we present the results for the structural-interaction descriptors, whereas table 3 presents the corresponding results for the energy-based descriptors and table 4 the results when all the descriptors are incorporated together.

Table 2: Results of the Models: LASSO, Elastic Net, GAM and KRLS for the structural-Interaction descriptors

Scoring Function	**Dataset 2**		**Dataset 3**		**Features Selected**
	R_p	RMSE	R_p	RMSE	
LASSO	0.428	2.967	0.406	2.890	4.0 Contacts, Sidechain beta, Hydrophobic
Elastic Net	0.419	2.965	0.429	2.851	4.0 Contacts, Backbone alpha, Sidechain beta, Receptor HBD, Hydrophobic, $\pi - \pi$, $\pi - t$, Salt Bridges
GAM	0.429	2.95	0.407	2.94	4.0 Contacts, Sidechain beta, Receptor HBD, Hydrophobic
KRLS	0.523	2.786	0.439	2.90	ALL

Table 3: Results of the Models: LASSO, Elastic Net, GAM and KRLS for the Energy-based descriptors

Model	**Dataset 2**		**Dataset 3**		**Features Selected**
	R_p	**RMSE**	R_p	**RMSE**	
LASSO	0.600	2.79	0.481	2.791	Lipo, VdW
Elastic Net	0.603	2.749	0.489	2.791	Lipo, VdW
GAM	0.584	2.650	0.494	2.774	ALL
KRLS	0.605	2.603	0.500	2.782	ALL
MMGbsa	0.479	-	0.461	-	ALL

From these results, we can deduce that the best descriptors, in this case-study, for modulating the protein-ligand interaction, are those based on their energies.

It is worth mentioning that the MMGbsa SF uses exactly the same energy terms as us, however we can notice that with our models, we achieve better results. In the case of the *Dataset 2*, we improved the prediction 26.3% whereas for the *Dataset 3* the improvement was 8.4%. In order to facilitate this comparison, the outcome can be seen in figure 11. It is important to emphasize that in the *Dataset 3*, the improvement is less significant, this can be explained, because in our training, we are using different protein-families.

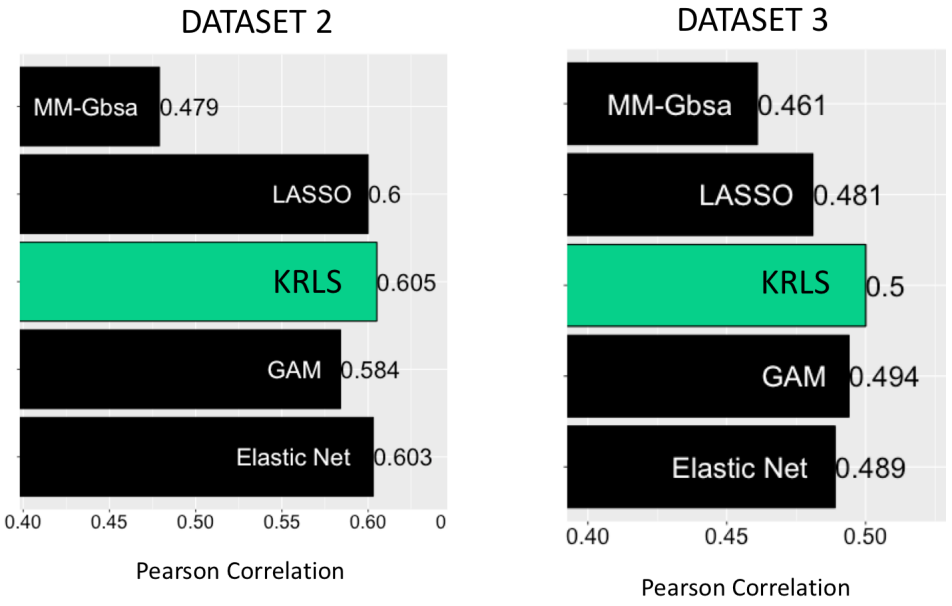

Figure 11: Comparison between the results of the models for the energy=based descriptors with MMGbsa, in Pearson correlation terms. Left figure shows the results for the *Dataset 2* whereas the right shows the results for the *Dataset 3*. In green is highlighted the best result achieved.

When we incorporate all the descriptors together, we improved the prediction in RMSE terms for all the models, except GAM, as shown in table 4.

Table 4: Results of the Models: LASSO, Elastic Net, GAM and KRLS when all the descriptors are incorporated

Model	**Dataset 2**		**Dataset 3**		**Feature Selected**
	R_p	**RMSE**	R_p	**RMSE**	
LASSO	0.585	2.691	0.500	2.738	4.0 Contacts, Sidechain Beta, Hydrophobic, $\pi - t$, Lipo, VdW.
Elastic Net	0.589	2.699	0.509	2.728	4.0 Contacts, Sidechain Beta, Hydrophobic, $\pi - t$, Lipo, VdW.
GAM	0.573	2.681	0.469	2.918	4.0 Contacts, Sidechain other, backbone other, Coulomb, Bond, Lipo, Packing, Solv_GB, Ligand_Strain, VdW
KRLS	0.642	2.502	0.602	2.578	ALL

From the above table, we can notice that when we use the KRLS method to combine them, it shows an outstanding result, showing its ability to combine different groups of descriptors, by using a similarity approach. The reason why we found this method can achieve better results, is because this technique does not encompass any parametric assumption, similar to GAM, however what makes it more suitable for combining different kind of descriptors is that it has a penalization term and allows to shrink the coefficients, yielding to a better performance. This can be further explained, if we observe the smoothed splines of the filtered descriptors obtained from GAM, figures 12 and 13, we can see that although we made an effort to filter features with this method, there still remain some of them, that apparently do not have a substantial relation with the free energy. On the other hand, the estimations of the conditional expectations retrieved from KRLS, figures 14 and 15, give us a better intuition about the expected values and why by using similarity approaches plus a regularization term, the results can be improved.

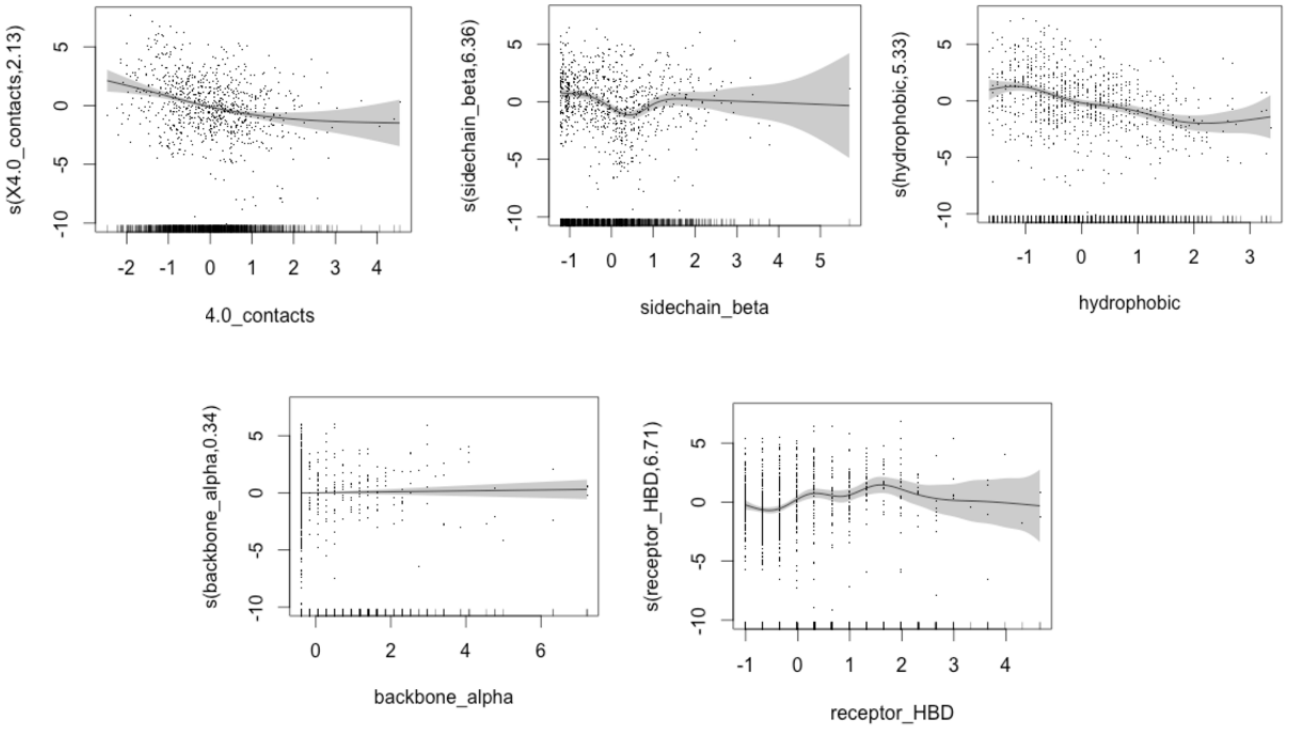

Figure 12: GAM predicted cubic smooth splines of Experimental binding affinity as a function of the structural-interaction descriptors of the *Dataset 1*, selected from this method: 4.0 contatcs, Sidechain Beta, hydrophobic, backbone alpha and receptor HBD . The degrees of freedom are in the parenthesis on the y-axis. The gray areas represent the confidence intervals of the smooth splines. The thick marks in the x-axis indicate the distribution of the observations.

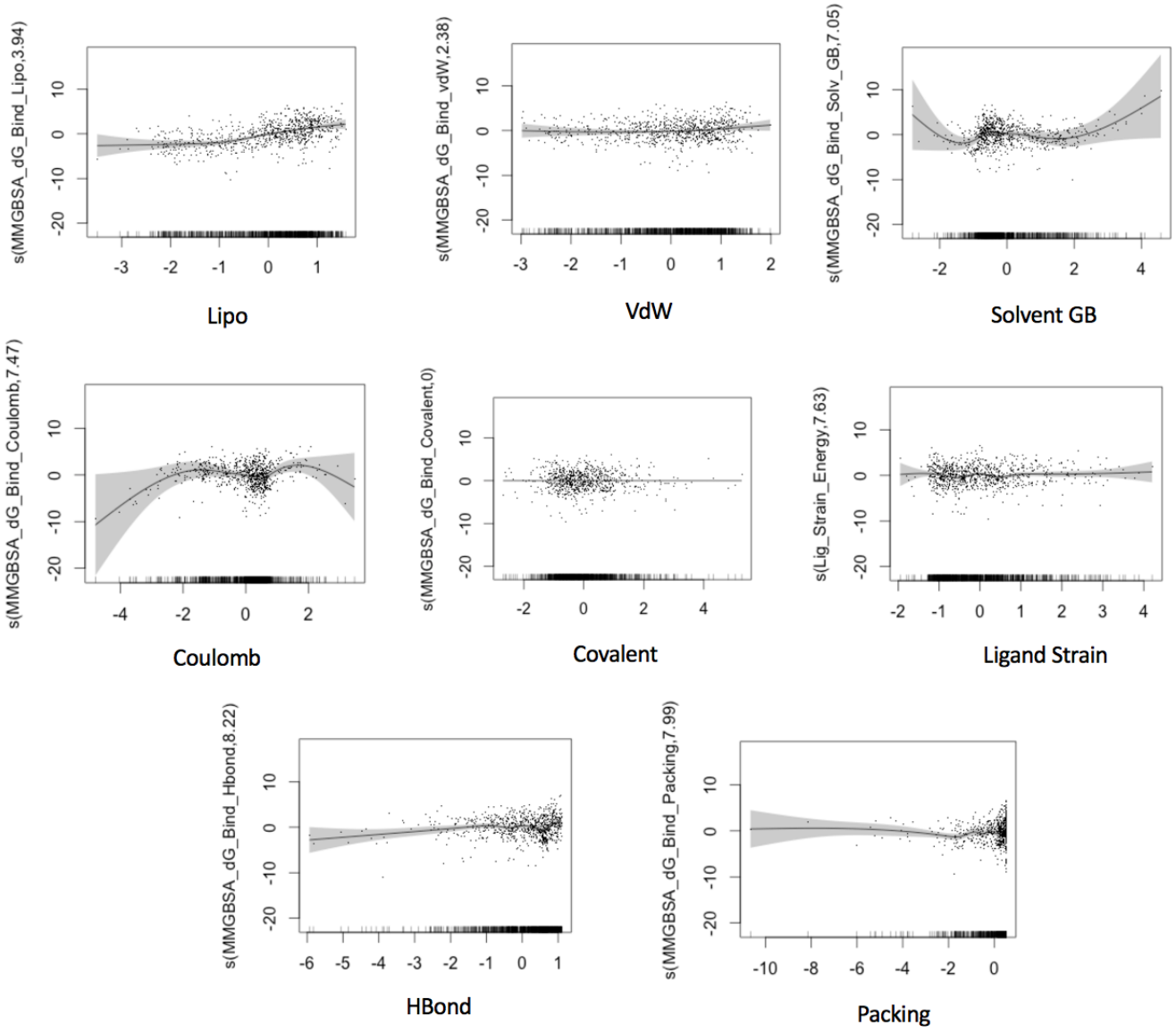

Figure 13: GAM predicted cubic smooth splines of Experimental binding affinity as a function of the energy-based descriptors, selected from this method, of the *Dataset 1*:Coulomb, Covalent, Lipo, VdW, Solvent GB, Ligand Strain, Packing, HBond. The degrees of freedom are in the parenthesis on the yaxis. The gray areas represent the confidence intervals of the smooth splines. The thick marks in the xaxis indicate the distribution of the observations

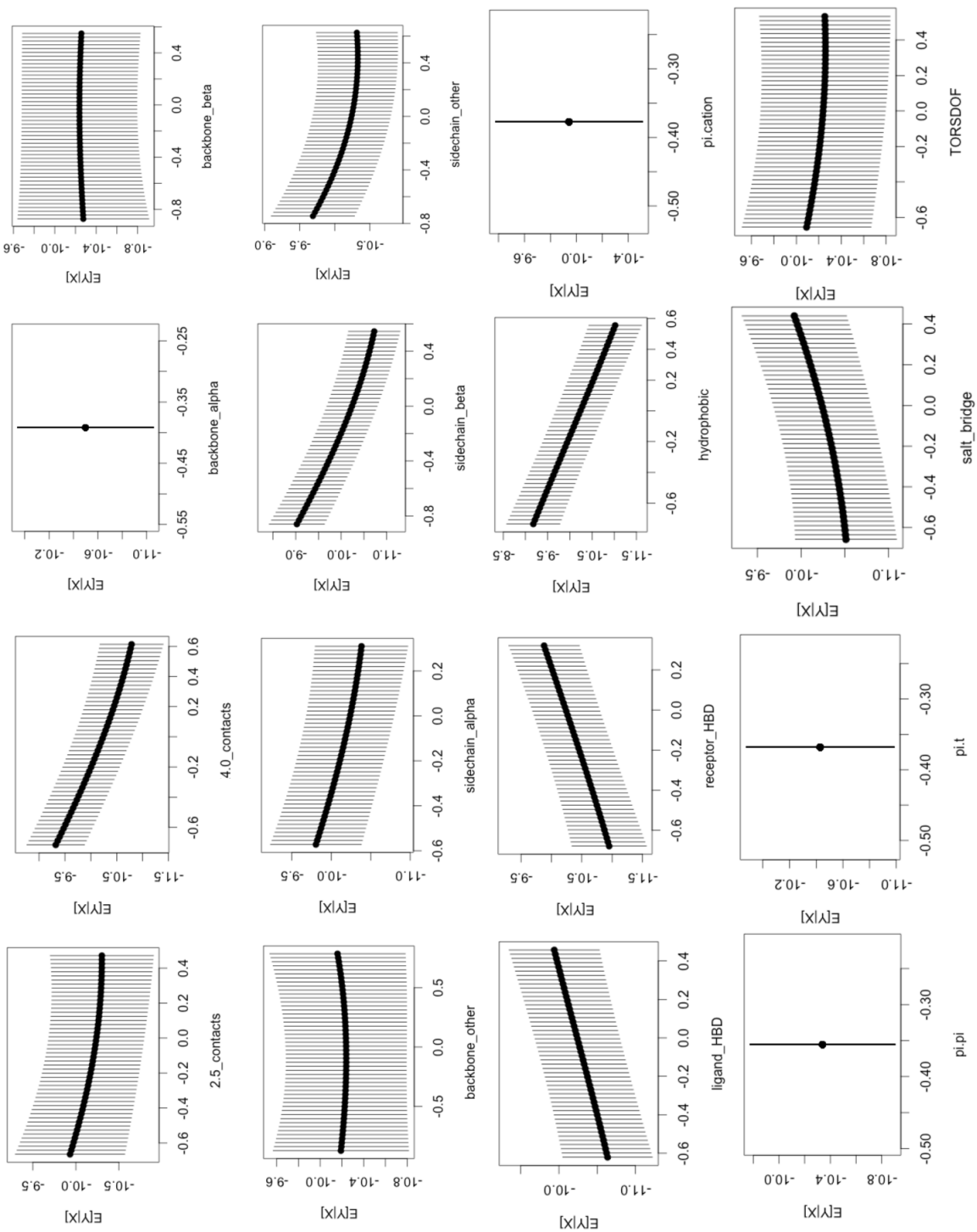

Figure 14: Estimates of the conditional expectations functions $E[Y|X]$ for every structural-interaction predictor of the *Dataset 1*, obtained from the model KRLS. Horizontal axis shows the values of the predictor from its 1st to its 3rd quartile values, whereas the vertical axis exhibits its correspondent expected value. Vertical lines over the expectation line, represent the confidence band.

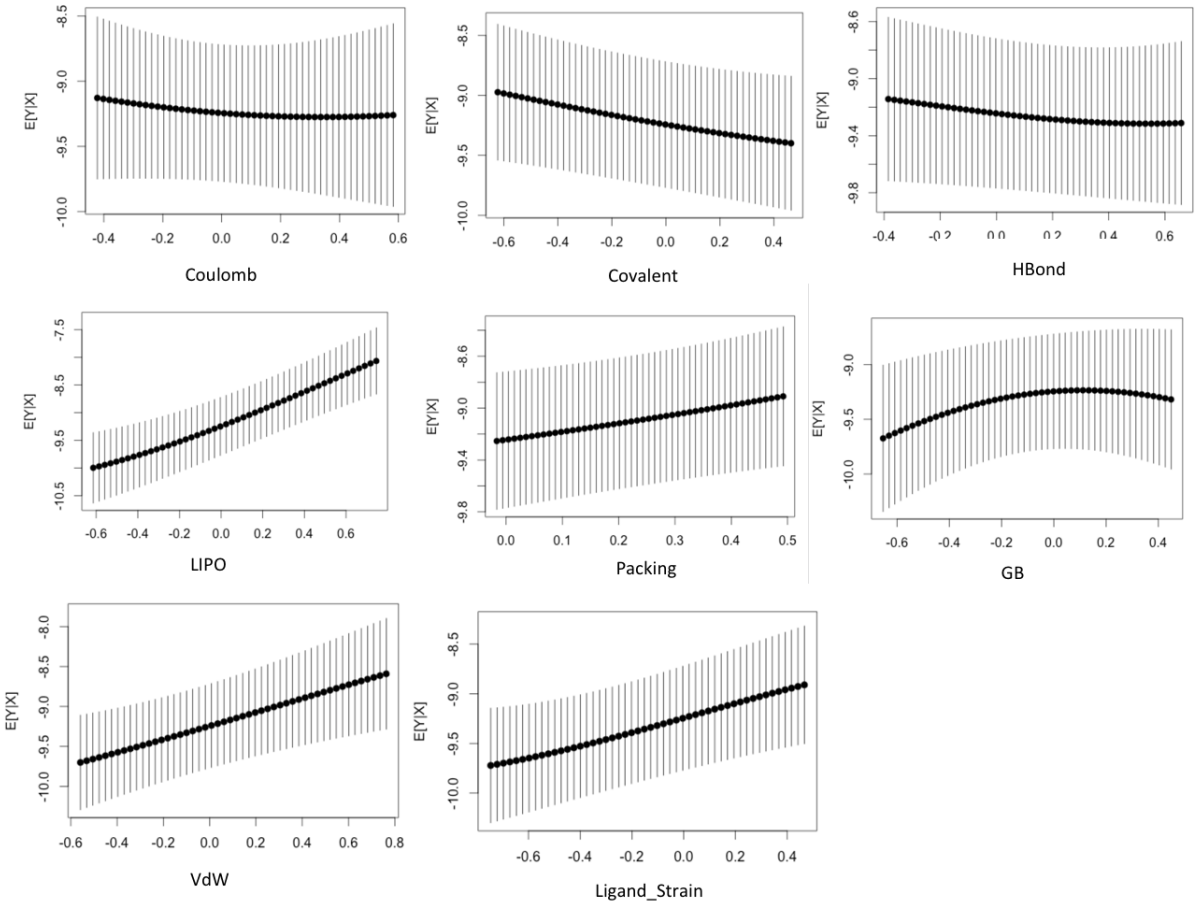

Figure 15: Estimates of the conditional expectations functions $E[Y|X]$ for every energy predictor of the *Dataset 1*, obtained from the model KRLS. Horizontal axis shows the values of the predictor from its 1st to its 3rd quartile values, whereas the vertical axis exhibits its correspondent expected value. Vertical lines over the expectation line, represent the confidence band.

7.1.2 Results of the models for the SFs

The framework proposed in this work also allows to combine different kind of SFs, even in the case when no descriptors are provided. Thereby, we assessed the combination of all the SFs, used in this case-study, with the different proposed methods.

In order to better analyze the improvement achieved by combining the different SFs in each dataset, table 5 shows the best SF for each dataset, based on their performance evaluation. Whereas table 6, shows the results obtained from each model, when we combine all the SFs.

Table 5: Best SF performance in each dataset.

Dataset	**Best Scoring Function**	R_p	RMSE
Dataset 2	XScore::HMS	0.645	2.526
Dataset 3	RFScore	0.636	2.518

Table 6: Results of the Models: LASSO, Elastic Net, GAM and KRLS for combining SFs

Model	**Dataset 2**		**Dataset 3**		**Feature Selected**
	R_p	**RMSE**	R_p	**RMSE**	
LASSO	0.608	2.63	0.650	2.45	Autodock Vina, XScore::HMS, RFScore
Elastic Net	0.636	2.60	0.667	2.45	Autodock Vina, XScore::HMS, XScore::Average, RFScore
GAM	0.667	2.44	0.657	2.379	Autodock Vina, XScore::HMS, XScore::HSS, XScore::HPS, XScore::Average, RFScore
KRLS	0.695	2.365	0.675	2.33	ALL

In the above table, we can see that with our best model (KRLS) we have accomplished a modest improvement in the overall performance. For the *Dataset 2*, the improvement achieved was of 7.7% with respect to the best SF in this dataset, i.e. XScore::HMS. Whereas for *Dataset 3*, the improvement accomplished was of 6.1% with respect to the best SF in this dataset, namely, RFScore. For illustration purposes, figure 16 exhibits these results.

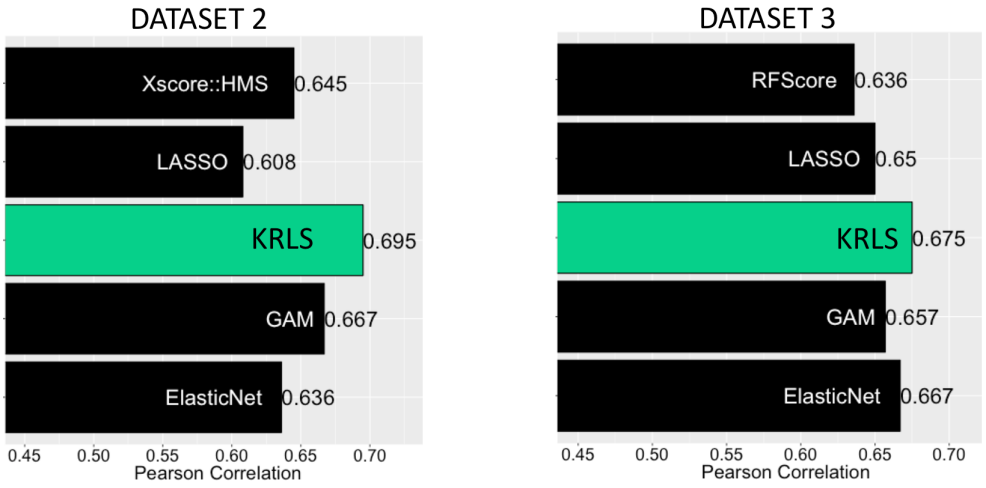

Figure 16: Comparison of the model results with the best SF for each dataset. Left figure, shows the comparison of all the models w.r.t. the XScore::HMS SF, that is the one with the best performance in *Dataset 2*. Right figure, shows the comparison of all the models w.r.t. the RFScore, that is the one with the best performance in *Dataset 3*. In green, the best results achieved are highlighted.

By analyzing the different results obtained from the models, we can realize that the outcomes of the KRLS model stood out. We believe that this method presents very good results for this purpose, because it combines the benefits of using a regularization term, like Ridge regression, without implying any parametric assumption, allowing to shrink coefficients and also detect intervals for which each SF performs better.

7.1.3 Results of the models used for Stacking SFs with descriptors

In view of the need to incorporate new descriptors to already existed SFs, to boost their predictive power, we proposed a methodology based on stacking with ridge regression. We strongly suggest to use this methodology to SFs that do not already encompass the descriptors.

Having said that, in our case-study, we have 10 different SFs, however some of them already circumscribe the descriptors used in this study. Therefore, we filter the SFs that do not contain already these descriptors, to find out how by adding new descriptors, we can improve the overall binding free energy. To this end, we have selected DSX, RFScore, Glide SP and Glide XP, which do not contain neither the interaction-structural descriptors nor the energy terms we are using.

Instead of using the descriptors alone, we created different models with them, we chose the best one and later we blend it with the other SFs. As exposed in Section 7.1.1, the best model for combining all the descriptors was KRLS.

Table 7 shows the results obtained when we combine the SFs (Glide SP, Glide XP, RFScore and DSX) with the best model that characterize the descriptors, i.e. KRLS. For comparison purposes, we added in the results other well known regression techniques for stacking, such as Elastic Net, LASSO and regular Linear Regression in order to highlight that Ridge Regression has the best results, for this situation. Hence, we used this method in our scheme. Additionally, we also added the performance measures of the SFs used.

Table 7: Results of the models: LASSO, Elastic Net, Ridge and Linear Regression when we stack DSX, Glide SP, Glide XP, RFScore with the best model of descriptors, in this case, the KRLS mode (KRLS-DEscriptors-Model).

Model	**Dataset 2**		**Dataset 3**		**Feature Selected**
	R_p	**RMSE**	R_p	**RMSE**	
LASSO	0.695	2.36	0.669	2.385	DSX, RFScore, KRLS-Descriptors-Model
Elastic Net	0.694	2.366	0.667	2.386	DSX, RFScore, KRLS-Descriptors-Model
Ridge Regression Stacking	0.701	2.31	0.681	2.33	ALL (DSX, RFScore, Glide SP, Glide XP, KRLS-Descriptors-Model)
Linear Regression	0.682	2.421	0.648	2.47	ALL (DSX, RFScore, Glide SP, Glide XP, KRLS-DescriptorsModel)
Glide SP	0.448	3.232	0.494	3.113	NA
Glide XP	0.448	3.921	0.397	3.944	NA
DSX	0.584	-	0.551	-	NA
RFScore	0.571	2.705	0.636	2.518	NA

With the aim of better interpret the results disclosed in the above table, in Figure 17, we compare for each dataset the performance of each SF used with the results obtained from stacking them with the best model of descriptors, via Ridge Regression.

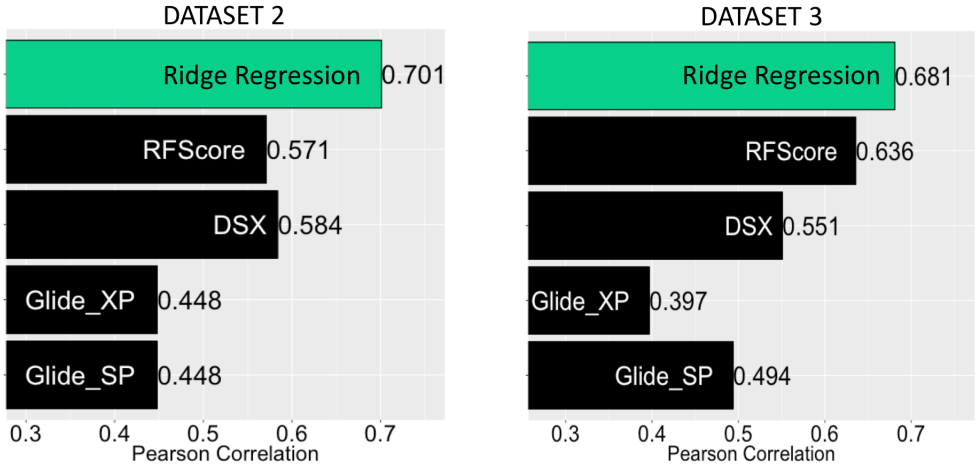

Figure 17: From left to right. a) Comparison of the performance of DSX, RFScore, Glide SP and Glide XP and the results obtained from the ridge stacking with the best model of descriptors.

From the above results, we can realize that in the *Dataset 2*, we achieved an outstanding overall

improvement of 20% with respect to the best SF used in this example, i.e. DSX. Whereas for the *Dataset 3*, the accomplished overall improvement was of only 7%. For the nature of the methods we are using, we cannot expect, in this part, outstanding results in datasets that does not contain similar proteins as in the training, as in the case of *dataset 3*.

As a proof that modeling individually the descriptors, and then add their best model to other set of SFs works better than just adding the descriptors, in Table 8 we show the results when we stack the SFs with single descriptors.

Table 8: Results of the Models: LASSO, Elastic Net, Ridge and Linear Regression for blending DSX, RFScore, Glide SP and Glide XP, with the single descriptors.

Model	**Dataset 2**		**Dataset 3**		**Features Selected**
	R_p	**RMSE**	R_p	**RMSE**	
LASSO	0.599	2.62	0.643	2.43	RFScore, Sidechain beta, 2.5 contacts, 4.0 Contacts, Sidechain Other, Hydrophobic, Lipo
Elastic Net	0.607	2.61	0.639	2.457	RFScore, DSX, Sidechain beta, 2.5 contacts, 4.0 Contacts, Sidechain Other, Hydrophobic, Lipo
Ridge	0.618	2.586	0.595	2.536	ALL
Linear Regression	0.589	2.649	0.600	2.582	ALL

The above results can be explained based on the premise that when we use stacking techniques, we seek to only blend models, which are in essence different, moreover descriptors do not necessarily have a linear relationship with the binding energy, so it is a better idea to first model them and then combine them with other model of SFs.

It is also worth mentioning that when we combine only all the SFs together (table 6) without descriptors, we have gotten similar results as when we combine DSX, RFScore, GlideSP and GlideXP with the best model of descriptors, this is because Autodock VINA and XScore are already encompassing the descriptors we are using.

7.1.4 Results for the case of combining RFScore with more descriptors

In the literature there is a controversy of whether adding different descriptors to RFScore helps to improve its predictive power [4]. We wanted to assess this situation with our framework. We can see in table 9, that in fact, when we use Ridge Regression for stacking RFScore with the best model of descriptors, KRLS, as seen in Section 7.1.1, we were able to greatly improve the overall binding affinity prediction by 21.8% for the *Dataset 2*. Although, we were able to also improve the prediction of RFScore in the *Dataset 3*, the increment was of 7%, which is not as outstanding as it is for the *Dataset 2*. We can explain this situation, as before, because in the *Dataset 3* we don't have proteins with a lot of similarities with the ones in the *Dataset 1*, used as training.

Table 9: Results of the stacking models for blending RFScore with the best model of descriptors in this case KRLS (KRLS-Descriptors-Model)

Model	**Dataset 2**		**Dataset 3**		**Features Selected**
	R_p	**RMSE**	R_p	**RMSE**	
LASSO	0.695	2.365	0.667	2.39	RFScore, KRLS-Descriptors-Model
Elastic Net	0.695	2.364	0.666	2.392	RFScore, KRLS-Descriptors-Model
Ridge	0.696	2.343	0.676	2.332	RFScore,,KRLS-Descriptors-Model
Linear Regression	0.688	2.474	0.649	2.474	RFScore, KRLS-Descriptors-Model

For illustration and explanatory purposes, the above outcomes can be observe more precisely in Figure 18.

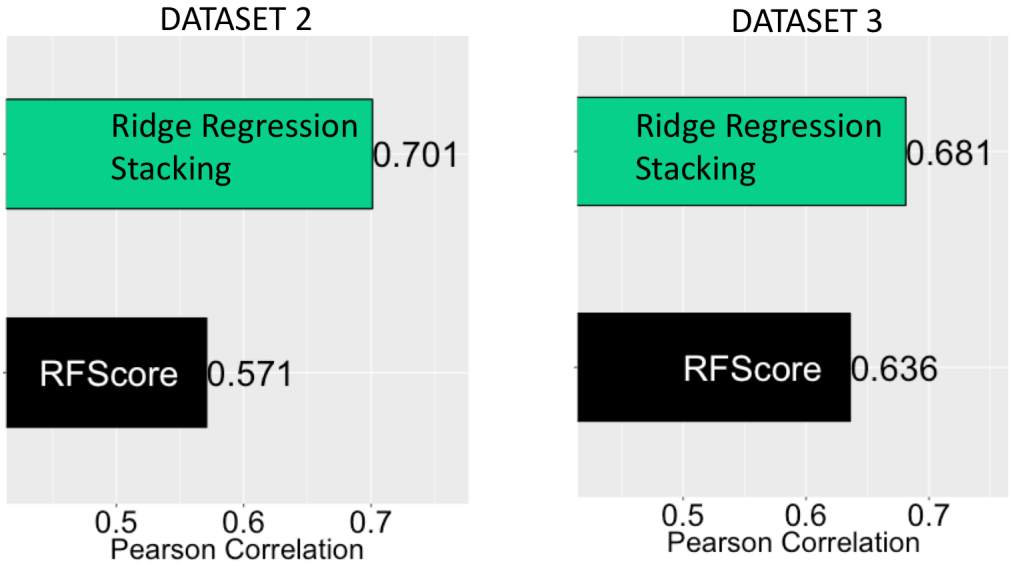

Figure 18: From left to right. a) Comparison between the performance of RFscore and the Ridge stacking result of RFScore with the best model of descriptors, for the *Dataset 2*. b) Comparison between the performance of RFscore and the Ridge stacking result of RFScore with the best model of descriptors, for the *Dataset 3*. In green is highlighted the best result obtained.

.

8 Discussion and Future Work

In this study, we have exploited several machine learning and statistical techniques in order to understand the best manner to combine a set of different descriptors and SFs separately, to estimate the binding energy of a protein and a ligand. Moreover, we have also investigated methods to further enrich the results of already built SFs by means of adding descriptors. We applied this methodology to well known datasets from the PDBbind database.

From the results outlined in Section 7, we have discovered that the nonparametric models present a more robust and stable technique for combining descriptors from different categories, as well as for combining SFs underlying different principles, because they are able to detect significant changes that make a difference while combining them. Previous studies [3, 2, 1] have already capitalized on the use of nonparametric techniques to model descriptors of protein-ligand complexes, using techniques such as Random Forests, Bagging, Boosting and Neural Networks. The reason why we did not use these nonparametric methods, is that we believe that in this field, we are not only interested in improving the overall binding affinity, but also we seek to learn how this improvement is achieved, understanding the processes and their outcomes, in such a way that, in the future, we can make more analyzed decisions for discovering potential drugs.

The downside of the nonparametric methods we used, is that although they give intelligible manners to analyze and interpret the outcomes, the interactions and the non-linearities, they do not explicitly allow to get a subset of the most important variables, i.e. we cannot perform embedded feature selection with them, and other methods must be used on the top if we want this to be achieved. In our case, we used penalized regression techniques with regularization terms that allow to shrink and make zero some coefficients. These methods offer reasonable good results, with a good trade-off between performance and computational cost, to provide helpful insights of the best variables to use.

We believe that the GAM method gives very useful insights, though we could not take much advantage of them for predicting the binding affinity because of the poor performance of the feature selection procedure. Hence, as a future work we would like to modify this and instead of using the standard GAM $g(E[X]) = \beta + \sum_{j=1}^{p} f(X_j)$, where $f(X_j)$ represents the cubic spline for the X_j variable, we would like to estimate cubic splines for each variable and then apply to these splines a linear (in parameters) regression technique with penalization, i.e. $g(E[X]) = \beta_0 + \sum_{j=1}^{p} \beta_j f(X_j) + \lambda R(\beta)$, where $R(\beta)$ is the penalization/ regularization term.

In the same manner, we would like to enhance the KRLS method, by adding an embedded feature selection process. This is aimed to be achieved by modifying the penalization term and instead of using the Tikhonov L_2 norm, change it for a more flexible penalization such as the one used in Elastic Net, so that, we can have more flexibility and a possible good manner to shrink coefficients while selecting a subset of features.

From this study, we learned that stacking techniques should be taken carefully. These techniques have a good performance when we have a set of substantially different models with similar accuracy. We encourage to use this technique for SFs that does not already circumscribe the new descriptors we want to add. With the proposed framework, we can firstly run the models for the SFs, and then make a decision of which SFs to use for adding new descriptors.

Furthermore, to have better achievements when combining SFs with descriptor models, we strongly recommend to cautiously select the training set, trying to select complexes that encompass similar descriptors and characteristics as the ones from which we want to predict the binding free energy. As an example, we can take the case from the results section, in which we combined the Glide SP, Glide XP, RFScore and DSX SFs with the best model of descriptors. In *Dataset 2*, which encloses similar complexes as in the training set (*Dataset 1*), we achieved an improvement of approximately 25%, whereas for the

Dataset 3, that does not contain very similar proteins as in the training, the improvement was only of 7%.

To tackle the above situation, as a future work we would like to make an effort to create a dataset with different descriptors, trying to enclose all the possible ranges for each one, and then create a probabilistic network, such as a Bayesian or Markovian Network, with them, to later generate synthetic samples from this network. In that case, instead of using a traditional bootstrapping technique, which performs sampling with replacement, we would use a new method to generate samples from a Bayesian or Markovian network of the descriptors. This method has to be further planned, to not end up with an excessive computational cost.

One of the main difficulties, in this work, has been the preparation of the datasets and the treatment of outliers, which requires a lot of expertise in the chemistry field. In datasets composed by hundreds or thousands of protein-ligand complexes, it is impossible to check whether the structures of all the systems are correct. The descriptors as well as the SFs are automatically retrieved, so it is a time-consuming task to check complex by complex whether this was correctly computed or not. Therefore as a future work we would like to make use of visualization and other techniques to allow the user identify these kind of problematic systems to better prepare training datasets and decide which kind of complexes would be the most appropriate to use for a specific situation.

9 Conclusions

In this work, we have made an exhaustive study to understand how to improve the prediction of protein-ligand binding affinity. We exploited several machine learning and statistical techniques to learn valuable aspects of the protein-ligand interactions and their role to predict the binding free energy. We have also developed a methodology to improve the performance of a single or a set of SFs by addining new descriptors. We applied all these techniques to a designed case study, obtaining promising improvements. By analyzing the results from different perspectives, such as the features selected from the penalized techniques, the estimation of splines obtained from GAM and the estimates of the conditional expectation functions $E[Y|X]$ from KRLS, we could also get decisive insights of the role of the descriptors to estimate the binding free energy and the performance of the SFs.

We believe that this work can be further extended to be used in other studies of importance for the pharmaceutical industry, such as virtual screening, scoring of induced fit poses and lead optimization on a given target.

Finally, we would like to emphasize the fact that, towards a more reliable prediction of the protein-ligand binding affinity, it does not only suffice to improve the overall results, but also to understand the role of all the elements that have led to this achievement, to determine the conditions for using these techniques and the possible drawbacks and limitations.

References

[1] Marcelino Arciniega and Oliver F Lange. Improvement of virtual screening results by docking data feature analysis. *Journal of chemical information and modeling*, 54(5):1401–1411, 2014.

[2] Hossam M Ashtawy and Nihar R Mahapatra. Bgn-score and bsn-score: Bagging and boosting based ensemble neural networks scoring functions for accurate binding affinity prediction of protein-ligand complexes. *BMC bioinformatics*, 16(4):1, 2015.

[3] Pedro J Ballester and John BO Mitchell. A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. *Bioinformatics*, 26(9):1169–1175, 2010.

[4] Pedro J Ballester, Adrian Schreyer, and Tom L Blundell. Does a more precise chemical description of protein–ligand complexes lead to more accurate prediction of binding affinity? *Journal of chemical information and modeling*, 54(3):944–955, 2014.

[5] Helen Berman, Kim Henrick, Haruki Nakamura, and John L Markley. The worldwide protein data bank (wwpdb): ensuring a single, uniform archive of pdb data. *Nucleic acids research*, 35(suppl 1):D301–D303, 2007.

[6] Yu-Chian Chen. Beware of docking! *Trends in pharmacological sciences*, 36(2):78–95, 2015.

[7] Tiejun Cheng, Xun Li, Yan Li, Zhihai Liu, and Renxiao Wang. Comparative assessment of scoring functions on a diverse test set. *Journal of chemical information and modeling*, 49(4):1079–1093, 2009.

[8] Robert D Clark, Alexander Strizhev, Joseph M Leonard, James F Blake, and James B Matthew. Consensus scoring for ligand/protein interactions. *Journal of Molecular Graphics and Modelling*, 20(4):281–295, 2002.

[9] Wendy D Cornell, Piotr Cieplak, Christopher I Bayly, Ian R Gould, Kenneth M Merz, David M Ferguson, David C Spellmeyer, Thomas Fox, James W Caldwell, and Peter A Kollman. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *Journal of the American Chemical Society*, 117(19):5179–5197, 1995.

[10] Jacob D Durrant and J Andrew McCammon. Binana: a novel algorithm for ligand-binding characterization. *Journal of Molecular Graphics and Modelling*, 29(6):888–893, 2011.

[11] Jacob D Durrant and J Andrew McCammon. Nnscore 2.0: a neural-network receptor–ligand scoring function. *Journal of chemical information and modeling*, 51(11):2897–2903, 2011.

[12] Matthew D Eldridge, Christopher W Murray, Timothy R Auton, Gaia V Paolini, and Roger P Mee. Empirical scoring functions: I. the development of a fast empirical scoring function to estimate the binding affinity of ligands in receptor complexes. *Journal of computer-aided molecular design*, 11(5):425–445, 1997.

[13] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[14] Richard A Friesner, Jay L Banks, Robert B Murphy, Thomas A Halgren, Jasna J Klicic, Daniel T Mainz, Matthew P Repasky, Eric H Knoll, Mee Shelley, Jason K Perry, et al. Glide: a new approach for rapid, accurate docking and scoring. 1. method and assessment of docking accuracy. *Journal of medicinal chemistry*, 47(7):1739–1749, 2004.

[15] Richard A Friesner, Robert B Murphy, Matthew P Repasky, Leah L Frye, Jeremy R Greenwood, Thomas A Halgren, Paul C Sanschagrin, and Daniel T Mainz. Extra precision glide: docking and scoring incorporating a model of hydrophobic enclosure for protein-ligand complexes. *Journal of medicinal chemistry*, 49(21):6177–6196, 2006.

[16] Samuel Genheden and Ulf Ryde. The mm/pbsa and mm/gbsa methods to estimate ligand-binding affinities. *Expert opinion on drug discovery*, 10(5):449–461, 2015.

[17] Holger Gohlke and Gerhard Klebe. Statistical potentials and scoring functions applied to protein–ligand binding. *Current opinion in structural biology*, 11(2):231–235, 2001.

[18] Jens Hainmueller and Chad Hazlett. Kernel regularized least squares: Reducing misspecification bias with a flexible and interpretable machine learning approach. *Political Analysis*, page mpt019, 2013.

[19] Trevor Hastie and Robert Tibshirani. Generalized additive models. *Statistical science*, pages 297–310, 1986.

[20] Richard D Head, Mark L Smythe, Tudor I Oprea, Chris L Waller, Stuart M Green, and Garland R Marshall. Validate: A new method for the receptor-based prediction of binding affinities of novel ligands. *Journal of the American Chemical Society*, 118(16):3959–3969, 1996.

[21] Jérôme Hert, Peter Willett, David J Wilton, Pierre Acklin, Kamal Azzaoui, Edgar Jacoby, and Ansgar Schuffenhauer. New methods for ligand-based virtual screening: use of data fusion and machine learning to enhance the effectiveness of similarity searching. *Journal of chemical information and modeling*, 46(2):462–470, 2006.

[22] Micael Jacobsson, Per Lidén, Eva Stjernschantz, Henrik Boström, and Ulf Norinder. Improving structure-based virtual screening by multivariate analysis of scoring data. *Journal of medicinal chemistry*, 46(26):5781–5789, 2003.

[23] Anthony E Klon, Meir Glick, and John W Davies. Combination of a naive bayes classifier with consensus scoring improves enrichment of high-throughput docking results. *Journal of medicinal chemistry*, 47(18):4356–4359, 2004.

[24] Jianing Li, Robert Abel, Kai Zhu, Yixiang Cao, Suwen Zhao, and Richard A Friesner. The vsgb 2.0 model: a next generation energy model for high resolution protein structure modeling. *Proteins: Structure, Function, and Bioinformatics*, 79(10):2794–2812, 2011.

[25] Yan Li, Zhihai Liu, Jie Li, Li Han, Jie Liu, Zhixiong Zhao, and Renxiao Wang. Comparative assessment of scoring functions on an updated benchmark: 1. compilation of the test set. *Journal of chemical information and modeling*, 54(6):1700–1716, 2014.

[26] Jie Liu and Renxiao Wang. Classification of current scoring functions. *Journal of chemical information and modeling*, 55(3):475–482, 2015.

[27] Giampiero Marra and Simon N Wood. Practical variable selection for generalized additive models. *Computational Statistics & Data Analysis*, 55(7):2372–2387, 2011.

[28] Garrett M Morris, Ruth Huey, William Lindstrom, Michel F Sanner, Richard K Belew, David S Goodsell, and Arthur J Olson. Autodock4 and autodocktools4: Automated docking with selective receptor flexibility. *Journal of computational chemistry*, 30(16):2785–2791, 2009.

[29] Janmenjoy Nayak, Bighnaraj Naik, and H Behera. A comprehensive survey on support vector machine in data mining tasks: applications & challenges. *International Journal of Database Theory and Application*, 8(1):169–186, 2015.

[30] Gerd Neudert and Gerhard Klebe. Dsx: a knowledge-based scoring function for the assessment of protein–ligand complexes. *Journal of chemical information and modeling*, 51(10):2731–2745, 2011.

[31] Akifumi Oda, Keiichi Tsuchida, Tadakazu Takakura, Noriyuki Yamaotsu, and Shuichi Hirono. Comparison of consensus scoring strategies for evaluating computational models of protein-ligand complexes. *Journal of chemical information and modeling*, 46(1):380–391, 2006.

[32] Dale J Poirier. Piecewise regression using cubic splines. *Journal of the American Statistical Association*, 68(343):515–524, 1973.

[33] Daniela D Rosa, Gustavo Ismael, Lissandra Dal Lago, and Ahmad Awada. Molecular-targeted therapies: lessons from years of clinical development. *Cancer treatment reviews*, 34(1):61–80, 2008.

[34] Joseph Sill, Gábor Takács, Lester Mackey, and David Lin. Feature-weighted linear stacking. *arXiv preprint arXiv:0911.0460*, 2009.

[35] Sung-Sau So and Martin Karplus. A comparative study of ligand-receptor complex binding affinity prediction methods based on glycogen phosphorylase inhibitors. *Journal of computer-aided molecular design*, 13(3):243–258, 1999.

[36] Vsevolod Yu Tanchuk, Volodymyr O Tanin, Andriy I Vovk, and Gennady Poda. A new, improved hybrid scoring function for molecular docking and scoring based on autodock and autodock vina. *Chemical biology & drug design*, 2015.

[37] Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

[38] Renxiao Wang, Luhua Lai, and Shaomeng Wang. Further development and validation of empirical scoring functions for structure-based binding affinity prediction. *Journal of computer-aided molecular design*, 16(1):11–26, 2002.

[39] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[40] Simon Wood. *Generalized additive models: an introduction with R*. CRC press, 2006.

[41] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.