Development of computational tools for the *in silico* design and optimization of bioactive peptides

Inaugural Dissertation

for

the doctoral degree of

Dr. rer. nat.

from the Faculty of Biology

University of Duisburg-Essen

Germany

Submitted by

MSc. Sandra Romero Molina

Born in Santa Clara, Cuba

December, 2022

The experiments underlying the present work were conducted at the Department of Computational Biochemistry at the University of Duisburg-Essen.

1. Examiner: Prof. Dr. Elsa Sanchez Garcia

2. Examiner: Prof. Dr. Michael Ehrmann

3. Examiner: Prof. Dr. Jörg Behler

Chair of the Board of Examiners: Prof. Dr. Daniel Hoffmann

Date of the oral examination: 21.04.2023

DOI: 10.17185/duepublico/78311
URN: urn:nbn:de:hbz:465-20230508-090012-5

Table of Contents

1 Preface 5
2 Zussamenfasung 6
3 Summary 9
4 Introduction 12
4.1 The drug discovery process 12
4.1.1 The stages of the drug development process 12
4.1.2 Targeting protein-protein interactions 12
4.1.3 Development of drugs 13
4.2 Peptides as therapeutic compounds 14
4.2.1 Bioactive peptides 14
4.2.2 Therapeutic potential of bioactive peptides 15
4.2.3 *In silico* development of bioactive peptides 16
4.2.3.1 Virtual screening 16
4.3 Machine learning 17
4.3.1 Types of learning 17
4.3.1.1 Supervised learning 18
4.3.1.2 Unsupervised learning 20
4.3.1.3 Ensemble learning 20
4.3.2 Components of the learning task 21
4.3.2.1 Data collection and curation 21
4.3.2.2 Features generation 22
4.3.2.3 Data pre-processing 24
4.3.2.4 Feature selection 25
4.3.2.5 Learning algorithms 29
4.3.2.6 Optimization of hyperparameters 33
4.3.2.7 Evaluation approaches 34
4.3.2.8 Performance measures 35
4.3.2.9 Definition of the applicability domain 38
4.4 Machine learning in drug discovery 38
4.4.1 The principles of the *Organization for Economic Co-operation and Development* for the development of *Quantitative Structure-Activity Relationships* (QSAR) models 39
4.4.2 Standard procedure to develop machine-learning-based QSAR models 39
4.4.2.1 Implementation of web servers 41
4.4.3 Applications and importance of machine learning in drug discovery 41

4.4.4 Machine learning for the identification and study of bioactive peptides 43
4.4.4.1 State-of-the-art methods 43
4.4.4.2 Challenges of state-of-the-art methods 44
5 Objectives 46
6 Publications 47
6.1 Publication I 48
6.2 Publication II 77
6.3 Publication III 140
6.4 Publication IV 178
7 Discussion 205
7.1 The models 205
7.2 Modelling process 206
7.3 Evaluation of the models 209
7.4 Evaluation on other experimental data 211
7.5 Implementation of web servers 212
7.6 General considerations concerning the models 213
7.7 Conclusions 214
8 References 217
9 Appendix 229
9.1 List of Abbreviations 229
9.2 List of Figures 232
9.3 List of Tables 233
9.4 Curriculum Vitae 234

1 Preface

This thesis comprises four original publications. The work presented in this thesis was performed between September 2018 and July 2022 under the supervision of Prof. Dr. Elsa Sanchez Garcia at the Department of Computational Biochemistry, University of Duisburg-Essen, Germany.

2 Zussamenfasung

Wegen ihrer Biokompatibilität, biologischen Abbaubarkeit und Selektivität sind Peptide wichtige therapeutische Moleküle. Aufgrund ihrer Biochemie eignen sich Peptide unter anderem zur Imitation der Bindungsstellen von Proteinen, zur Inhibition krankheitsrelevanter Protein-Protein-Interaktionen und um das Problem der Multiresistenz zu studieren. Deshalb wurde in den letzten Jahren der Entwicklung und Optimierung bioaktiver Peptide viel Aufmerksamkeit gewidmet. Die Entdeckung neuer Arzneimittel beginnt oft mit der Analyse großer Peptidbibliotheken. Das experimentelle Screening solcher Bibliotheken ist jedoch teuer und zeitaufwändig. In-silico-Methoden, die die Zahl der Kandidaten mit verbesserten Eigenschaften reduzieren können, sind für das moderne Drogendesign unerlässlich.

In den letzten Jahrzehnten wurden mehrere Methoden für Protein-Protein-Interaktionen entwickelt, die auf *machine learning* (ML) basieren. Anhand der verfügbaren Informationen wurden diese Methoden trainiert, um beispielsweise Proteininteraktionen zu erkennen (Klassifizierungsproblem) oder um die Bindungsaffinität (BA) als Regressionsproblem vorherzusagen. Unabhängig von der vorhergesagten Variable leiden die meisten der bisher vorgestellten Methoden jedoch unter einer geringen Generalisierungsfähigkeit, da sie eine hohe Varianz bei der Vorhersage von neuen Daten aufweisen. Darüber hinaus werden Peptide im Bereich der Protein-Protein- und Protein-Ligand-Interaktionen von den meisten Methoden auf die gleiche Weise behandelt wie Proteine oder kleine organische Liganden. Diese Überlegung unterschätzt die Spezifität kurzer Peptidsequenzen und reduziert die Leistung bei der Vorhersage von Protein-Peptid-Interaktionen. Ähnlich wurden ML-basierende Methoden auch zur Identifizierung therapeutischer Moleküle, wie z. B. antimikrobieller Peptide (AMPs), eingeführt. Viele Methoden sind jedoch nicht darauf ausgelegt, eine bestimmte Funktion für mutmaßliche AMPs vorherzusagen, wie z. B. antibakterielle Aktivität. Bei der Suche nach bioaktiven Peptiden zur Bekämpfung der Multiresistenz von Bakterien zeigen die modernen Methoden eine eingeschränkte Genauigkeit bei der Vorhersage der antibakteriellen Aktivität und es fehlen häufig weitere Informationen über die Art der möglichen Ziele. Um das akkurate De-novo-Design bioaktiver Peptide zu ermöglichen, ist die Entwicklung neuartiger computergestützter Werkzeuge notwendig. Diese Dissertation beschreibt, wie ML-Techniken dazu beitragen können, Methoden zu erstellen, mit deren Hilfe komplexe Fragestellungen im Peptidesign gelöst werden können (**Table 1**). Schwerpunkte der Dissertation sind:

(1) Ein sequenzbasiertes Werkzeug für Protein-Protein- und Protein-Peptid-Interaktionen, der zur Identifizierung von Leitstrukturen durch extensives *in silico* Screening von Protein-Peptid-Wechselwirkungen eingesetzt werden kann. Das Werkzeug basiert auf einem ML-

basierten Klassifikator, der die Wahrscheinlichkeit von Interaktion vorhersagt. Das Ausgabemodell wurde durch die Nutzung von Informationen aus mehreren öffentlich zugänglichen Datenbanken und durch die Verwendung von Support Vector Machines (SVM) erstellt. Die Methode wurde als Web-Werkzeug namens **PPI-Detect** implementiert. Die ML-Studie nutzte die gleichen molekularen Deskriptoren, die bereits in ProtDCal implementiert sind. ProtDCal ist ein Programm für die numerische Kodierung von Proteinen, das in mehreren Studien validiert wurde. ProtDCal wurde anfangs dazu entwickelt, einzelne Proteine zu kodieren. Daher erforderte die Modellierung des sequenzbasierten Modelles die Einführung eines neuartigen Verfahrens zur Kodierung zweier individueller Aminosäuresequenzen in eindeutigen numerischen Deskriptoren. Dieses Verfahren wurde in ProtDCal implementiert um künftigen datenbasierden Studien zur Analyse von Proteinpaaren zur Verfügung zu stehen.

(2) Werkzeuge um die Bindungsaffinitäten (BA) von Protein-Protein- und Protein-Peptid-Bindungen für 3D-Strukturen zu schätzen, mit Anwendung in Mutagenese-Experimenten und Protein-Engineering. Die ML-Modelle verwendeten Informationen, die in öffentlichen Datenbanken gelistet sind. Beide Modellierungen wurden mit SVM durchgeführt, und die Ausgabemodelle wurden als Web-Werkzeug namens **PPI-Affinity** implementiert. Neben der BA-Schätzung ermöglicht die PPI-Affinity auch die Optimierung von Peptidsequenzen, für die 3D-Komplexstrukturen bestimmt wurden. Die implementierten Funktionalitäten ermöglichen die Erzeugung von Tausenden von Peptidderivaten durch Substitutionen und/oder Auslöschungen an den Aminosäureresten, die sich an der Kontaktfläche des Protein-Peptid-Komplexes befinden.

(3) Ein Werkzeug zur Identifizierung antibakterieller Peptide (ABPs) und des Gram-Färbungstyps der Zielbakterien, das zur Identifizierung von Leitpeptiden mit dem Potenzial zur Bekämpfung der Multidrogenresistenz dient. Die Methode für ABPs, genannt **ABP-Finder**, wurde von mir als Webserver implementiert. Das Programm ermöglicht die Zerlegung von Proteinsequenzen in kurze Peptidfragmente vor der Modellvorhersage. Diese Funktionalität findet Anwendung bei der Suche nach Proteindomänen mit antibakterieller Aktivität.

(4) Die in (1) – (3) erwähnten ML-Werkzeuge nutzten die molekularen Deskriptoren von ProtDCal. Dieses Programm wurde zuerst als Standalone-Program implementiert. Diese Arbeit zielt darauf ab, die Benutzerfreundlichkeit des Programms zu erweitern, und die Nutzung der entwickelten ML-Modelle, die mit dem ProtDCal-Kodieransatz erstellt wurden, zu erleichtern. Dieses Ziel wurde durch folgende Maßnahmen erreicht:

Implementierung einer Web-Plattform, die die Berechnung von ProtDCal-Moleküldeskriptoren für Data-Mining-Studien und die Nutzung der ProtDCal-basierten Werkzeugen für das virtuelle Screening in den ersten Schritten der Peptidentdeckung ermöglicht. Der Webserver namens **ProtDCal-Suite** bietet Zugang zu den in dieser Arbeit vorgestellten ML-basierten Methoden und zu anderen veröffentlichten Werkzeugen, die die funktionelle Analyse von Proteinen und Peptiden erleichtern. Darüber hinaus enthält die Online-Schnittstelle von ProtDCal eine Zusatzfunktion, mit der die Moleküldeskriptoren nach den Shannon-Entropiewerten der eingegebenen Proteine geordnet und gefiltert werden können. Die entwickelten Werkzeuge bieten die Möglichkeit für das virtuelle Screening von Peptiden in den frühen Phasen des Drug-Designs-Prozesses von peptidbasierten Arzneimitteln. ProtDCal-Suite ist frei zugänglich unter https://protdcal.zmb.uni-due.de.

Table 1. Liste der entwickelten Programme im Rahmen dieser Arbeit.

	Zweck
PPI-Detect	Eine sequenzbasierte Methode zur Vorhersage von für Protein-Protein- und Protein-Peptid-Interaktionen.
PPI-Affinity	Werkzeuge zur Vorhersage der Protein-Protein- und Protein-Peptid-Bindungsaffinitäten für 3D-Strukturen.
ABP-Finder	Ein Werkzeug zur Identifizierung antibakterieller Peptide und des Gram-Färbungstyps der Zielbakterien.
ProtDCal-Suite	Eine Web-Plattform, die (i) die Berechnung von ProtDCal-Moleküldeskriptoren und (ii) die Nutzung der ProtDCal-basierten Methoden für das virtuelle Screening von Peptidebiblioteken ermöglicht.

Die Generalisierungsfähigkeit der trainierten Modelle wurde an mehreren externen Testsets, die experimentelle Daten enthielten, validiert. **PPI-Detect** wurde verwendet, um Derivate von EPI-X4, einem endogenen Peptidinhibitor des Chemokinrezeptors CXCR4, zu untersuchen. Diese Analyse führte zur Identifizierung eines kürzeren und aktiveren Derivats von EPI-X4. **PPI-Affinity** wurde bei der Bewertung von EPI-X4-Mutanten, die an CXCR4 gekoppelt sind, und von Peptiden, die Komplexe mit den Serinproteasen HTRA1 und HTRA3 bilden, überprüft. Die Auswertung der PPI-Affinity in den diversen Testsets zeigte, dass die Protein-Protein-BA-Methode zur Spitze der modernsten BA-Methoden gehört. Außerdem war die Protein-Peptid-BA-Methode die erste, die auf Daten trainiert wurde, die aus diversen Protein-Peptid-Strukturen bestanden. **ABP-Finder** steht an der Spitze der modernsten ML-Methoden für ABPs, insbesondere im Bereich der Genauigkeit. ABP-Finder wurde für das Screening einer großen Peptidbibliothek aus dem Peptidom des menschlichen Urins verwendet. Auf der Basis dieser virtuelle Screening Studie wurde ein neuartiges antibakterielles Peptid experimentell identifiziert.

3 Summary

Peptides are important therapeutic molecules due to their biocompatibility, biodegradability, and selectivity. Their biochemistry makes peptides suitable for mimicking the binding site of proteins, for the inhibition of disease-relevant protein-protein interactions, and to address the problem of multi-drug resistance, among other applications. Therefore, much attention has been devoted in recent years to the design and optimization of bioactive peptides. Frequently, the discovery of new drugs starts with the analysis of large peptide libraries. However, the experimental screening of such libraries is expensive and time-consuming. *In silico* approaches that potentially reduce the list of candidates for further improvement are essential for modern drug design.

Several machine-learning-based predictors of protein-protein interactions have emerged in the last decades. Based on the available information, these predictors have been trained, for instance, to detect protein interactions or the lack of them (classification problem), or to predict binding affinity (BA) as a regression problem. However, regardless of the output variable, most models introduced so far suffer from low generalization capabilities, displaying high variance when predicting unseen data. Additionally, within the context of protein-protein and protein-ligand interactions, most methods contemplate peptides in the same way as proteins or small organic ligands. This consideration underestimates the specificity of short peptide sequences and results in poor performance in predicting protein-peptide interactions. Similarly, machine-learning-based methods aiming to identify therapeutic molecules, such as antimicrobial peptides (AMPs), have been introduced. However, many of these methods are not able to predict a specific function for putative AMPs, such as antibacterial activity. Consequently, in the search for bioactive peptides to address multi-drug resistance in bacteria, state-of-the-art tools display limited precision in predicting antibacterial activity and generally lack further information about the possible targets. Thus, novel computational methods to accurately aid the *de novo* design of bioactive peptides are needed. In this work, my aim was to leverage machine learning (ML) techniques to create tools to study bioactive peptides (**Table 1**). My work focused on:

(1) A sequence-based predictor of protein-protein and protein-peptide interactions applicable to the identification of lead compounds from extensive *in silico* screening of protein-peptide interactions. The model is a classifier that predicts the likelihood of interaction. It was created by exploiting information annotated on various public databases and by using Support Vector Machines (SVM). The output model was implemented as a web tool named **PPI-Detect**. The ML study utilized the molecular

descriptors implemented in ProtDCal, a tool for the numerical codification of proteins, which was validated in diverse studies. ProtDCal was initially intended to encode individual proteins. Thus, the modeling of the sequence-based predictor required introducing a novel procedure to encode the information of two individual amino acid sequences into unique numerical descriptors. This procedure was implemented in ProtDCal and made available for future data-driven studies encompassing the analysis of protein pairs.

(2) Predictors of protein-protein and protein-peptide binding affinities for 3D structures, with applications for mutagenesis experiments and protein engineering. The ML models utilized information annotated on various public databases. Both modeling processes were conducted using SVM and the output models were implemented as a web tool named **PPI-Affinity**. The web server allows, in addition to the BA estimation, the optimization of a putative peptide sequence for which a 3D complex structure has been resolved. In addition, the implemented functionalities permit the generation of thousands of peptide derivatives by performing substitutions and/or deletions on the peptide residues located at the interface of contact of the protein-peptide complex.

(3) A tool to identify antibacterial peptides (ABPs) and the Gram-staining type of targeted bacteria, with applications for the identification of lead peptides with the potential to tackle multi-drug resistance. The predictor of ABPs, named **ABP-Finder**, was implemented by me as a web server. Before the step of prediction by the model takes place, the server permits the breakdown of protein sequences into short peptide fragments. Such functionality finds application in the discovery of protein domains with antibacterial activity.

(4) The ML tools mentioned in (1) – (3) utilized the molecular descriptors implemented in ProtDCal. Originally, ProtDCal was implemented as a standalone application. In this work, I aimed to extend the applicability of ProtDCal and to facilitate the use of models created using the ProtDCal codification approach. To this end, my aims were: To implement a web platform to permit (i) the generation of ProtDCal molecular descriptors for data-mining purposes and (ii) the application of ProtDCal-based tools for virtual screening in the early steps of peptide discovery. The resulting web server, named **ProtDCal-Suite**, provides access to the ML-based methods introduced in this work and to other tools previously published, facilitating the functional analysis of proteins and peptides. Additionally, the online interface of the ProtDCal software includes a post-processing optional functionality to rank and filter the molecular descriptors according to

the Shannon entropy values of the input set of proteins. The developed tools allow for the virtual screening of peptides at the early stages of the drug design process involving peptide-based pharmaceuticals. ProtDCal-Suite is freely accessible at https://protdcal.zmb.uni-due.de.

Table 1. List of tools developed within this work.

	Purpose
PPI-Detect	A sequence-based predictor of protein-protein and protein-peptide interactions.
PPI-Affinity	A tool to predict and optimize the binding affinity of protein-protein and protein-peptide complexes.
ABP-Finder	A tool to identify antibacterial peptides and the Gram-staining type of targeted bacteria.
ProtDCal-Suite	A web platform to facilitate (i) the generation of ProtDCal molecular descriptors and (ii) the application of ProtDCal-based tools for the virtual screening of peptide libraries.

The generalization capability of the models trained by me was validated by assessing the models' performance on several external test sets that included experimental data. **PPI-Detect** was used to study derivatives of EPI-X4, an endogenous peptide inhibitor of the chemokine receptor CXCR4. This analysis resulted in the identification of a shorter and more active derivative of EPI-X4. **PPI-Affinity** was evaluated in the ranking of mutants of EPI-X4 coupled to CXCR4, and peptides forming complexes with the serine proteases HTRA1 and HTRA3. The evaluation for PPI-Affinity on the different test sets evidenced that the protein-protein BA predictor ranks among the top state-of-the-art BA predictors to date. Moreover, to the best of my knowledge, our protein-peptide BA predictor was the first tool trained on data comprised exclusively of diverse protein-peptide structures. **ABP-Finder**, on the other hand, ranked on top of the state-of-the-art predictors of antibacterial peptides, particularly in terms of precision. ABP-Finder was used to screen a large peptide library from the human urine peptidome. Based on this virtual screening study, a novel antibacterial peptide was experimentally established.

4 Introduction

4.1 The drug discovery process

4.1.1 The stages of the drug development process

Drug discovery is an arduous process comprising several stages (**Figure 1**). Often, it begins either with the discovery and validation or with the use of already known target biomolecules that, in association with certain compounds, might have therapeutic purposes. Next, compounds with activity against a validated target, as well as with suitable properties for further screening are designed. The most promising hits are then optimized to improve their activity against the target and their absorption, distribution, metabolism, excretion (ADME), and toxicity (T) profiles are analyzed. Lastly, preclinical and clinical studies are conducted to determine the efficacy and safety of the developed drug in patients, as well as to decide the method of administration, and dosage, among other specificities[1]. Overall, the aforementioned pipeline is a high-risk investment process whose cost generally fluctuates between $161 million and $4.54 billion (2019 US$), with the highest expenditures for anticancer drugs[2]. Such expense is mainly due to the high failure rates associated with the identification of suitable candidates, which still account for more than 90% of the failure of clinical trials[3].

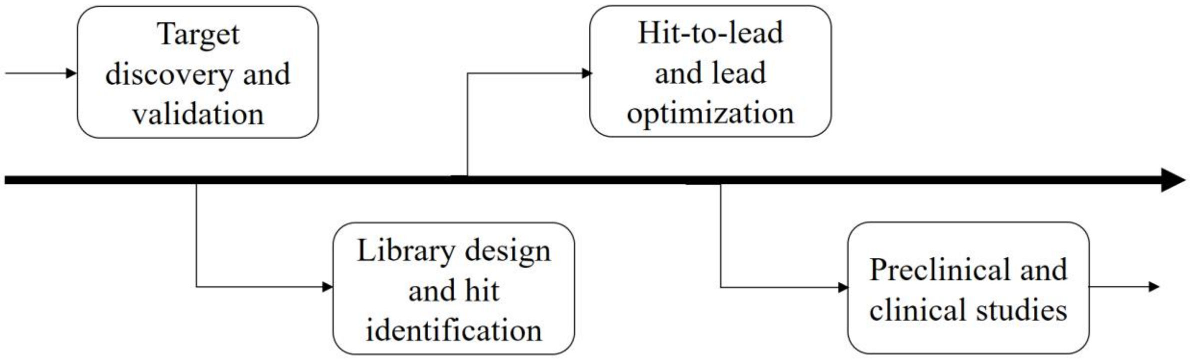

Target discovery and validation
Hit-to-lead and lead optimization
Library design and hit identification
Preclinical and clinical studies

Figure 1. Stages of the drug design process[1]

4.1.2 Targeting protein-protein interactions

Proteins are biomolecules that can bind to other molecules and exert relevant biological functions. For example, protein-protein interactions (PPIs) participate in almost all processes occurring in living cells[4-6]. Relevant PPIs functions include the modification of properties of enzymes, activation or inhibition of other proteins, transport of molecules, immune recognition to infections, and catalysis of metabolic reactions[7, 8]. The loss of key interactions, conformational changes of protein complexes, as well as anomalous aggregation, can alter the normal functioning of PPIs[9]. Some disruptions might not cause significant damage, but others are related to severe diseases[10]. For instance, the aggregation of certain proteins can be related to the development of degenerative pathologies, such as Parkinson’s and Alzheimer’s[11].

Likewise, some host-pathogen protein associations can lead to bacterial infections[12]. In cancer cells, alterations of cell signaling and regulatory pathways are triggered by mutations occurring in some proteins[13].

Some ligands (small molecule or other macromolecules) bind specifically to a protein receptor and compete with the original cognate partner of the protein, leading to an agonist or antagonist interaction that interferes with the function of the PPI[9]. These molecules can interfere with PPIs through either orthosteric (binding to the active site of the PPI) or allosteric (binding to other parts of the non-interacting protein surface) mechanisms (**Figure 2**). Both binding modes can lead to the modulation (inhibition or stabilization) of the PPI, and likewise, such modulation can result in either the inhibition or activation of the biological function[14]. Therefore, in the last few decades, significant efforts have been aimed to understand and predict PPIs, and to discover therapeutic ligands that can be used to modulate PPIs involved in disease[15].

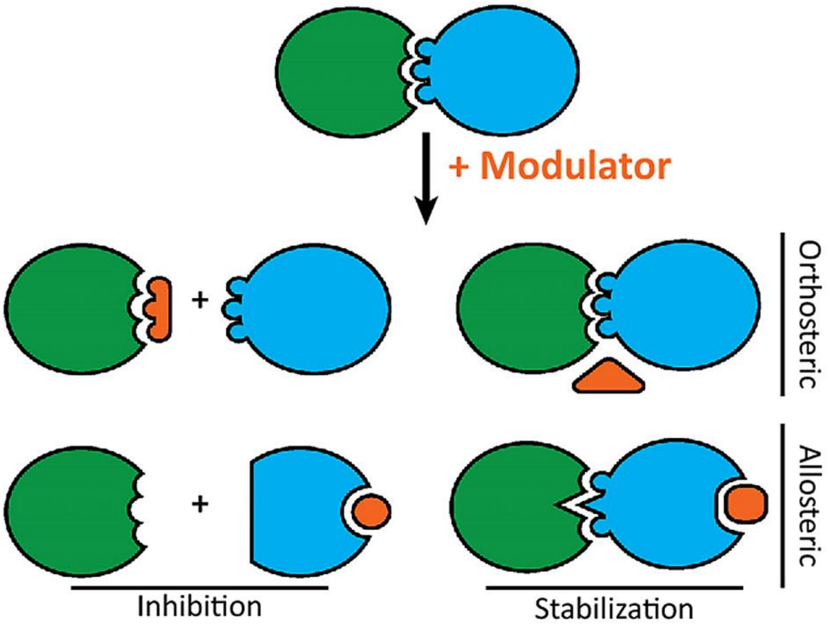


Figure 2. Mechanisms of action (orthosteric or allosteric) of modulators leading to the inhibition or stabilization of PPIs (from Modell, A. E. et al.[14])[Note1]

4.1.3 Development of drugs

Ligands can be either small molecules or larger macromolecules[16]. However, most compound libraries comprise mainly small molecules (molecular weight < 900 Da), including fatty acids, glucose, amino acids, cholesterol, lipids, glycosides, and alkaloids, among others. Consequently, small molecules represent 90% of the pharmaceutical market[17]. Small molecules are more explored due to their ability to penetrate the cell membrane and attach to deep folding

[1] Reprinted with permission from Elsevier and Copyright Clearance Center. License Number: 5402960059011. License date: Oct 06, 2022

pockets of the target protein with sufficient strength to alter the biological function of the target. However, small molecules-based strategies developed to interfere with intracellular PPIs face some drawbacks. The binding surfaces involved in PPIs are generally large (1500-3000 Å^2), driven by many polar and hydrophobic interactions, as well as flat and deprived of a well-defined binding pocket for efficient drug-candidate binding[18, 19]. Such limitations may be tackled by peptides (molecular weight between 500 and 5000 Da), which are located at an intermediate place between small molecules and short proteins, but with distinctive characteristics[20].

4.2 Peptides as therapeutic compounds

Peptides, like proteins, are amino acid sequences joined by peptide bonds. Peptide sequences typically range from 2 to 50 amino acids, while proteins usually comprise more than 50 amino acids[21]. However, these boundaries are flexible, as some polypeptides might also be considered proteins, e.g., the protein *crambin* containing 46 amino acids. Thus, according to the length of the sequence, peptides can be broadly classified as oligopeptides (maximum 20 amino acids) or polypeptides (between 20 and 50 amino acids). Nevertheless, while oligopeptides are classified as peptides, some polypeptides can be identified as small proteins as well[22] depending on their functions.

4.2.1 Bioactive peptides

In nature, peptides are encrypted in native protein sequences. *In vivo*, short peptide fragments can be released by digestive gastrointestinal enzymes (e.g., *trypsin, pancreatin, peptidase, pepsin, lipase*) or by microbial enzymes. *In vitro*, peptides can be also produced by proteolytic enzymes or by fermentation using microorganisms (e.g., *Lactobacillus helveticus*)[23, 24]. Inside the parent protein, these peptides are inactive, however, once released from the protein, they can display different properties. Such peptide fragments with the potential to affect biological functions and influence health are known as bioactive peptides[25].

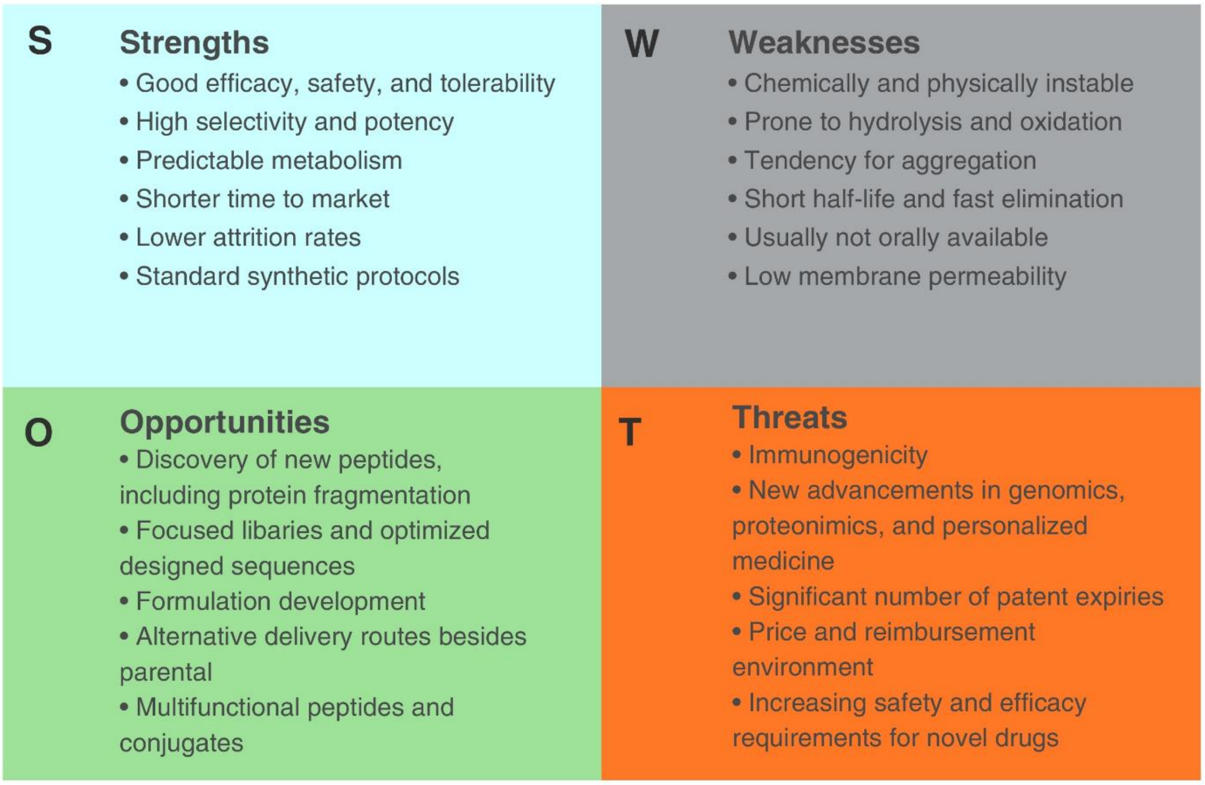
Bioactive peptides are considered excellent therapeutic molecules. They can be classified according to their therapeutic function as antimicrobial, anticancer, antidiabetic, antioxidant, and immunomodulatory peptides[26-28]. For instance, antimicrobial peptides (AMPs) are oligopeptides with a broad spectrum of inhibitory effects against infections caused by several organisms. In nature, AMPs can be found in various microorganisms, such as bacteria, as well as in eukaryotic species such as fungi, plants, and animals. In animals, AMPs are considered the first line of immune defense due to their ability to destroy viruses, bacteria, and fungi. Based on specific activities, AMPs may also be sub-classified as antiparasitic, antifungal, antiviral, anticancer (antitumor), and antibacterial peptides. Antibacterial peptides account for about 60% of AMPs[29].

4.2.2 Therapeutic potential of bioactive peptides

Peptides can bind to target protein receptors with high affinity and specificity[30]. Moreover, due to their amino acid composition, peptides are biodegradable and present low toxicity. Additionally, they exhibit a low risk of drug-drug interactions. Thus, therapeutically, peptides are usually safe, tolerable, and effective in humans. Such strengths are the main fundamentals of why peptide discovery has become an increasing field of research in the last decades[31]. As a result, it was estimated that in the current pharmaceutical market, the success rate of peptide-based drugs is twice the rate of small-molecule-based drugs with 60 approved peptides and around 20 peptide-based drugs entering the clinical trials annually[32].

Peptide drugs have been used in different areas such as cancer, diabetes, and human immunodeficiency virus type I (HIV-1) treatment as well as in hormone therapy, among others[33]. However, the development of bioactive peptides is a very challenging task. Among other weaknesses, peptides have limited stability, short half-live, and poor oral bioavailability (**Figure 3**). Usually, such limitations are addressed by using various strategies aimed at improving the physicochemical properties of promising lead compounds. For instance, proteins might be screened to find fragments with high affinity to a target receptor. Then, promising leads are used as scaffolds in an optimization process in which other techniques, such as sequence length, side chain, or peptide backbone modifications, as well as C-terminal amidation and N-terminal acetylation, are applied[22].

Peptide drug candidates are obtained through an intensive search and optimization process in which large libraries of peptide compounds are exploited to find the most promising ones to fulfill a desired therapeutic function[34]. However, the combination of the 20 naturally occurring amino acids to generate peptides of different lengths, together with the identification of peptide leads result in almost an unlimited search. Initial leads can be detected through experimental protein-peptide recognition techniques, but the processes involved are expensive and laborious. Thereby, advances in science and technology are constantly exploited to create methods aiming to assist scientists at all stages of the search for peptides modulating PPIs. These developments respond to the paradigm of rational drug design, in which several interdisciplinary fields, such as molecular biology, computational chemistry, and information technology, work together in the design of pharmaceutical compounds for which a target of interest has been identified and validated[35]. Thus, *in silico* methods that aim to improve the tasks involved in the drug design process are under constant development.



S **Strengths**
- Good efficacy, safety, and tolerability
- High selectivity and potency
- Predictable metabolism
- Shorter time to market
- Lower attrition rates
- Standard synthetic protocols

W **Weaknesses**
- Chemically and physically instable
- Prone to hydrolysis and oxidation
- Tendency for aggregation
- Short half-life and fast elimination
- Usually not orally available
- Low membrane permeability

O **Opportunities**
- Discovery of new peptides, including protein fragmentation
- Focused libaries and optimized designed sequences
- Formulation development
- Alternative delivery routes besides parental
- Multifunctional peptides and conjugates

T **Threats**
- Immunogenicity
- New advancements in genomics, proteonimics, and personalized medicine
- Significant number of patent expiries
- Price and reimbursement environment
- Increasing safety and efficacy requirements for novel drugs

Drug Discovery Today

Figure 3. Analysis of the strengths, weaknesses, opportunities, and threats (SWOT) of peptides (from Fosgerau and Hoffmann[31])[Note2]

4.2.3 *In silico* development of bioactive peptides

Drug design, as discussed above, encompasses predicting whether a specific molecule is likely to bind a target receptor and if so, the strength of such interaction[36]. To this end, advancements in computational technologies have favored the development of theoretical and computational methods enabling what is known as computer-aided drug design (CADD)[37, 38]. The use of CADD methods for virtual screening at the early stages of the drug design process can reduce time and cost by focusing experimental efforts on only a short list of promising compounds.

4.2.3.1 Virtual screening

The identification of hits involves the screening of large libraries of compounds. For this, traditional *in vitro* techniques such as High-Throughput Screening (HTS) may be used. However, a valuable approach, complementary and alternative to HTS is virtual screening (VS), consisting of the screening of large libraries using *in silico* approaches. The application of CADD methods for VS can lead to the cost-effective identification of hit compounds, which may also be derived from non-physical libraries[1]. Usually, VS techniques are classified as

[2] Published under the terms of the Creative Commons Attribution License (CC BY-NC-ND 3.0), which permits the copy and redistribution of the material in any medium or format.

ligand-based and structure-based. Ligand-based VS attempts to find new active compounds based on molecular similarity, employing as scaffolds known active and inactive molecules. In contrast, structure-based approaches assess the likelihood of a ligand binding a target receptor for which a three-dimensional structure is known[39].

One approach followed in both ligand- and structure-based VS is the use of data-driven models. Such is the case of Quantitative Structure-Activity Relationships (QSAR) models, whose utility in drug design and optimization has been well established[40]. In QSAR development, the structure and activity of compounds are correlated to create a model able to accurately predict the activity profile of untested compounds[41]. Introduced more than 50 years ago[42], QSAR has evolved from simple regression analysis to the use of ML techniques, capable of analyzing large datasets of biological systems[40]. Although initially used for single compounds, QSAR studies can incorporate information about the target, for instance, by introducing the amino acids sequence of the protein receptor. The developed models can be used in lead discovery and optimization to identify peptides with high activity and selectivity against a target PPI, among other applications[35].

4.3 Machine learning

Machine learning (ML) is the discipline of computer science that allows computers to have the ability to “learn” without being specifically programmed for the task[43]. It belongs to the broader field of artificial intelligence, which focuses on developing intelligent machines. In the early 1950s, Arthur Samuel popularized the “machine learning” term in the computer games domain[44]. As a research field, ML is an area of continuous evolution, with significant growth in the last three decades, reflected in the wide variety of services and software applications that nowadays use ML models.

The basic premise of learning is *to use a set of available observations to uncover an underlying process*[45]. Based on this, three criteria may motivate the application of ML techniques: the existence of a pattern, the difficulty for humans to mathematically define it, and the presence of data representative of the phenome. The outcome of the learning process is a mathematical model (or rule system) that can make accurate predictions on unseen data[46].

4.3.1 Types of learning

Machine learning involves three main learning paradigms: supervised, unsupervised, and reinforcement learning. *Supervised learning* (SL) aims at finding relationships between a set of input characteristics (independent variables) and an output variable (dependent variable). In *Unsupervised learning* (UL), as opposed to SL, the output variable is not explicitly specified, and the only available information is the input variables. The objective is to find relations between the variables that can lead to a higher representation level of the input data. Ideally, the identified relations are sufficiently relevant to form clusters or categories in the analyzed data. A third learning paradigm is *Reinforcement learning,* which is concerned with the problem of learning intelligent behavior in complex dynamic situations[47]. In this learning paradigm, the classification of the samples (output variable) is initially missing. The learning task is to discover the optimal outputs in a trial/error process with a reward/penalization system[48].

Additionally, there is a hybrid learning paradigm known as *Semi-supervised learning*, in which the data has labeled and unlabeled samples. The learning task is to leverage both data sources to train the model. Other approaches are recognized based on the strategy used to improve the learning process. *Multi-Task learning* seeks to improve generalization performance by learning several output variables measured on the same training samples. *Active learning* collects the training samples used to build a model by actively querying a system for the label of new instances. *Transfer learning* uses existing models as starting point to fit novel models. *Ensemble learning* consists of the development of several models on the same data. For the prediction step, a unique prediction value for an observation results from aggregating the outputs of individual models. *Deep learning* groups algorithms that improve the supervised learning technique Neural Networks to learn large and complex data representations with multiple levels of abstraction. Deep learning is considered a subset of ML, currently very successful due to the improvements achieved in highly complex tasks such as speech and visual object recognition[49].

In this work, supervised, unsupervised, and ensemble learning are the types of learning leveraged to create the ML models. Therefore, the following sections explain each of them in more detail.

4.3.1.1 Supervised learning

Supervised learning aims to find relationships between an input set of characteristics, $x = \{V_1, V_2, \ldots, V_d\}$ where d is the number of independent variables, and an output variable (dependent variable). For this, a collection of N samples is used $(x_i, y_i), \ldots, (x_N, y_N)$ from $\mathbb{R}^d$x $\mathbb{R}$, in which the i^{th} instance is a pair consisting of an input x_i object (d-dimensional vector) and an output y_i (e.g., class). Such sets of samples or observations are generally assumed as generated from a probability distribution P on X.

Learning stems from the assumption that there is an unknown target function $f: \mathbb{R}^d \rightarrow \mathbb{R}$ (e.g., the ideal formula to estimate whether a protein-peptide pair interacts) such that $y_i = f(x_i)$ for $i = 1, \ldots, N$. The learning task is to find a function $g: \mathbb{R}^d \rightarrow \mathbb{R}$ that approximates f based on the training samples. The function $g \approx f$ is a hypothesis selected by a learning algorithm from a set of candidate formulas H (hypothesis set). An example of H may be the set of all linear equations. Then $g \in H$ may be the best linear fit to the data. The final goal is to use the inferred function g to obtain the output variable of new independent data points (sample data).

SL tasks solve either classification or regression problems[50]. Classification tasks imply that the target variable is discrete, while regression alludes to numerical (continuous) values.

Model selection is usually based on a compromise between the ability of a model to fit the data and the complexity of the model needed to achieve this purpose[51]. Different levels of complexity can be applied to each hypothesis set, e.g., the number of degrees of a polynomial regressor. If the complexity is too low, all g in H may tend to underfit the training data (high bias), resulting in large training and test errors. In contrast, if the complexity of H is too high, all g in H may find spurious patterns and thus overfit the training data (high variance). This leads to a large gap between training and test errors (**Figure 4**). This relation is known as the bias-variance trade-off, used to control the complexity of H and balance underfitting/overfitting effects on the training process[52]. Thus, finding an intermediate spot between both concerns usually guides the selection process aiming to find generalizable models.

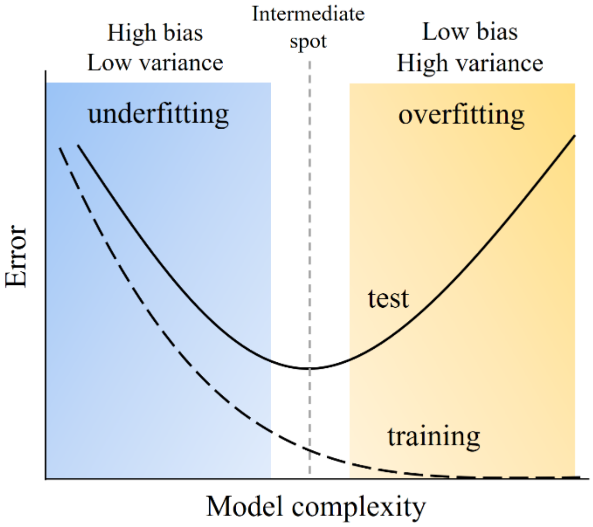


Figure 4. The learning curve arising from bias-variance trade-off (adapted from Beyeler et al.[53])[Note3]

The predictor g is commonly chosen from the hypothesis set by minimizing a regularized empirical risk function (ERM):

$$\text{ERM} = \frac{1}{N}\sum_{i=1}^{N} e(g(x_i), y_i) + \lambda \cdot r(g) \tag{1}$$

where $e: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ is an error or loss function that accounts for the quality of the fit, and $r: \mathrm{L} \rightarrow \mathbb{R}$ is a regularization function that penalizes the complexity of the function g to prevent overfitting. The amount of penalization is balanced by the λ parameter. Most ML methods

[3] Published under the terms of the Creative Commons Attribution License (CC BY), which permits the unrestricted use and reproduction of the image, on condition that the original author and source are cited.

apply different empirical risk function in terms of error and regularization functions, e.g., the absolute error $e(g(x_i), y_i) = |g(x_i) - y_i|$ in regression, or 0-1 loss $e(g(x_i), y_i) = sign(g(x_i) \,!= y_i)$ in classification[52].

4.3.1.2 Unsupervised learning

There are datasets comprised of a set of independent variables that lack an explicit definition of an outcome classification for the observations. Unsupervised learning corresponds to a group of techniques used to extract knowledge from this data. The goal of this type of problem, as described by Bishop (2006)[54], may be:

"to discover groups of similar examples within the data, where it is called clustering, or to determine the distribution of data within the input space, known as density estimation, or to project the data from a high-dimensional space down to two or three dimensions for the purpose of visualization".

UL techniques find applications to identify meaningful trends and structures in the data, uncover groups of samples, extract valuable features, and understand the data via visualization[55]. In this work, I used unsupervised learning for feature selection. I applied several techniques to remove highly correlated features and reduce the dimensionality of the data (see Section 4.3.2.4).

4.3.1.3 Ensemble learning

In Ensemble learning, several models are generated and aggregated in a unique final hypothesis that outputs a prediction. This type of learning assumes that the combination of weak models to deliver a consensus prediction improves the performance of a single model. Ensemble learning involves deciding how to build the base models and which criteria to use to combine the prediction of individual models on the final ensemble. Popular ensemble methods include Bagging[56], Boosting[57], and Stacking[58].

Bagging methods train several classifiers using different subsamples of the data. The selection process, called bootstrapping, picks randomly the samples (with replacement) from the complete dataset. For a test sample, each model outputs a prediction value. The outcomes of the different models are then combined using an average or consensus criterion to output a single final decision. Several rules are applied to combine the predictions of the base learners, including majority vote in classification and average, minimum or maximum predictions in regression.

Boosting methods create an ensemble model by training the base learners sequentially. In this process, each model in the ensemble is built using the same dataset but giving preference (weights) to the instances misclassified by the previously created model. For a test sample, each model outputs a prediction value. A weighted majority vote (or sum) of individual predictions outputs the final prediction.

Stacking methods consist of training a model to combine the predictions of several other models. For this, base models are built using typically different learning algorithms and the same available data. Next, a final meta-model balances the predictions outputted by the base models.

4.3.2 Components of the learning task

A ML task may not be a linear process. However, it generally consists of several stages involving taking decisions and carrying out steps that define the final model.

4.3.2.1 Data collection and curation

The first stage in the creation of machine learning models is data collection. The continuous advance of techniques and instruments for proteomics has led to a steady increase in the amount of data generated[59]. From experiments to publications, constant efforts aim to publish experimental records in centralized online databases. Thus, the produced data is usable beyond the specific project that initially generated it. As a result, several public databases relating protein structures to biological activities or properties have been published, with information determined by either human experts or experimental measurements.

Information on structurally resolved protein-protein/peptide interactions appears in diverse databases. For instance, the databases 3did[60], iPfam[61], and Negatome[62] collect thousands of interaction profiles of pairs of domain sequences whose three-dimensional complexes are in the Protein Data Bank (PDB)[63]. Likewise, the PDBbind database[64] reports the experimental binding affinities (BA) of protein-protein/ligand complexes stored in the PDB. Information on mutagenesis experiments is also available. For instance, the SKEMPI[65] database contains thousands of BA upon mutation of 350 structurally resolved protein-protein complexes. Repositories of bioactive peptides and their activity are also available. In this context, the database starPep[66] provides access to around 45,000 peptides with reported antimicrobial activity. These datasets are regularly updated to include new observations. For instance, PDBbind is updated annually with ~10% growth between the last two releases[67]. The PDBbind database (v. 2020) reports information on the binding affinities of different biomolecular systems, such as protein-ligand (19,443) and protein-protein (2,852) complexes. All these databases offer the opportunity to conduct ML studies to leverage available information in the analysis of peptides properties and functions.

Medicinal chemistry publications and bioactivity databases are known to contain high error rates. Thus, regardless of the database, data points must navigate through a rigorous data curation process to remove or correct those with questionable characteristics. In this pre-cleaning phase, samples with unreported activities or errors in the structure need to be detected to avoid affecting the predictivity of the model due to erroneous data[68]. This step may include removing some data points. However, samples with structural problems may be corrected, i.e., by adding hydrogen or other missing atoms with the aid of available computational methods.

4.3.2.2 Features generation

The adequate representation of input samples is critical for pattern recognition[69]. In ML, a collection of features represents the samples. A feature can be a characteristic or attribute describing the observation as a whole, or a part of it. Each feature represents a dimension of the space in which the sample is represented. The sample is then a data point in this space, which is associated with a specific vector. The feature vectors, corresponding to the multiple samples, are used as input by a learning algorithm(s) to model an endpoint. Most ML algorithms work on numerical values. Therefore, the transformation of input molecules into useful numerical features has been a standard procedure in ML modeling[70]. Many successful applications have addressed such transformations with the use of molecular descriptors[69]. Todeshini et al.[71] defined a molecular descriptor as:

> *"the final result of a logic and mathematical procedure which transforms chemical information encoded within a symbolic representation of a molecule into a useful number or the result of some standardized experiment".*

Based on this definition, molecular descriptors derive from (1) experimental measurements, e.g., measured physicochemical properties, and (2) theoretical definitions involving, among others, principles of information theory, graph theory, and computational chemistry[71]. In proteomics, diverse aspects of protein structure are extracted to quantitatively describe physicochemical properties of amino acids, as well as topological and structural features.

A wide variety of molecular descriptors has been gradually introduced[40]. From them, three families of descriptors are widely applied and validated in the analysis of protein function and properties. They are: 1) sequence-composition-based descriptors (0D), representing different physicochemical and structural aspects of the amino acid sequence, 2) linear-topology-based descriptors (1D), reflecting sequence-order information and its effect on the properties of individual residues, and 3) 3D-structure descriptors (3D), encoding information that characterizes the conformational structure of proteins[72]. Several applications implement information-rich molecular descriptors for proteins. Relevant examples are PseAAC[73] (extended to PSe-in-one 2.0[70, 74]), PROFEAT[75], ProtDCal[76], and more recently PyBioMed[77], Mordred[78], and BioMedR[79]. Notably, when ProtDCal's descriptors were introduced, it was shown how they captured, at that time, more data variance than the other tools available for protein codification[76].

ProtDCal molecular descriptors

ProtDCal, the acronym for PROTein Descriptors' CALculation, produces protein profiles based on a large diversity of descriptive statistical parameters (e.g., variance, mean, kurtosis, quantiles, and Shannon entropy) applied to different groups of residues extracted from the protein. The methodology of ProtDCal allows the calculation of tens of thousands of molecular descriptors per protein for both protein sequence (0D, 1D) and structure (3D). Ruiz-Blanco et al.[76] assessed ProtDCal's descriptors in a non-redundant dataset of 874 proteins. The evaluation considered:

1. The redundancy of the information contained within ProtDCal descriptors.
2. The variability of the molecular descriptors implemented in ProtDCal and those implemented in state-of-the-art generators of molecular descriptors[75, 80].
3. The diversity of ProtDCal's features compared to the molecular descriptors delivered by other programs[75, 80].

Comparisons with other packages leveraged only sequence-based features, as the assessed applications lacked an encoding approach for 3D structures. The evaluation demonstrated that ProtDCal generates a more extended and informative list of sequence-based features. Furthermore, the introduced 3D-structure-based features provide additional and complementary information to that offered for protein sequences, extending the protein characterization to a broader range of available data. These analyses showed the potential of the molecular descriptors implemented in ProtDCal to develop ML-based QSAR models.

The molecular descriptors implemented in ProtDCal have been employed in various studies. Sequence-based descriptors were applied to the prediction of antibacterial peptides[81, 82], antihypertensive activity and hemotoxicity of peptides[83], the development of monoclonal antibodies[84, 85], the identification of N-glycosylation sites[86] and methylation sites[87], as well as the prediction of protein stability[88, 89], residues critical for protein function[90], enzyme-substrate scope[91], and enzymatic function[72]. Likewise, the 3D-based descriptors of ProtDCal have found applications to model enzymatic function[72] and enzyme-substrate scope[91]. Such a diverse list of applications validates the suitability of ProtDCal descriptors to build data-driven models for the analysis of proteins. However, as the originally implemented codification approach only conceived the generation of molecular descriptors for individual proteins, other relevant problems, such as the modeling of protein-protein and protein-peptide interactions, could not be addressed using ProtDCal descriptors. Problems involving protein pairs require an encoding procedure that considers synergy between the two proteins and does not encode each one individually. As part of the work presented here, I implemented a novel encoding approach for protein pairs to extend the applicability of ProtDCal descriptors to a broader set of protein studies.

4.3.2.3 Data pre-processing

Measurements can be obtained from both experiments and theoretical models[92]. Such heterogeneous origin may cause a certain level of noise in the data. In broad terms, noise is considered anything that prevents a learning algorithm from identifying a reliable model. Noise also includes the different numerical scales that certain measures or molecular descriptors can present. ML algorithms applied to noisy data can lead to wrong pattern recognition and poor performance. In addition to the noise of available data, the feature generation step may calculate a large set of molecular descriptors, where only a few may correlate with the endpoint. Thus, data pre-processing is needed to enhance data quality for model development. General techniques include the scaling and normalization of numerical values, the treatment of missing values, and outlier detection.

Normalization

The descriptors are scaled up or down to transform them into a uniform range. This is done to avoid that descriptors with larger magnitude have a greater influence on the model over those with shorter range values. Several forms of normalization can be used:

Min-max normalization scales numerical features to the range (0,1) using the min and max values of the descriptors column in the training dataset. The formula to apply the transformation is:

$$d_{scaled} = (d - d_{min})/(d_{max} - d_{min}) \tag{2}$$

Where d is the original value of the descriptor, d_{min} and d_{max} correspond to the minimum and maximum descriptor values in the dimension, respectively.

Z-Score normalization scales the numerical features using the mean (μ) and the standard deviation (σ) values of the descriptor column in the training dataset, according to the following expression:

$$d_{scaled} = (d - \mu)/\sigma \tag{3}$$

Handling missing values

A dataset may have missing values for some instances or features, e.g., ProtDCal assigns a constant (-9999) to a descriptor which calculation is unviable for a certain protein. This output constitutes a missing value for the ML algorithms. In classification techniques, some occurrences of missing values may not damage the performance of the model. However, in regression, many algorithms cannot work on a dataset with missing data. Several approaches are applied to treat missing values, such as manually filling in the missing values or replacing those fields with the mean of the dimension. Besides, removing the instance or dimension is often a suitable option depending on how this action affects the further analysis.

Outlier detection

An outlier can be generally considered as a data point that is notably different from the other data points or that does not imitate the expected normal behavior of the other data points[93]. Many situations can give rise to the appearance of outliers, i.e., the heterogeneity in the source of the data.

Techniques for the detection of outliers can be categorized into statistical-, distance-, density-, clustering-, graph-, ensemble-, and learning-based methods. *Statistical-based* methods identify outliers by considering their relationship with the distribution model. *Distance-based* methods compute the distance between data points to detect those distant from the closest neighbors. *Density-based* techniques identify as outliers those points appearing in low-density regions. *Clustering-based* methods use classical clustering algorithms to detect observations in small clusters. *Graph-based* methods use graph techniques to analyze interdependencies in the data point and thus flag outliers. *Ensemble-based* approaches can help to detect outliers as they explore different models based on different subsets of data. *Learning-based* methods train models to detect outliers. A survey of the different outlier detection methods can be found elsewhere[93].

The question of how to handle outliers is problem dependent. Some techniques such as ensemble, allow to keep the outliers, while others require their removal in order to train accurate models. Approaches can analyze data points in multivariate or univariate space. The univariate technique identifies data points that contain extreme values on a single variable.

Data pre-processing also involves feature selection, explained in the next section.

4.3.2.4 Feature selection

The feature extraction problem was defined by Devijver and Kittler (1982) as[94]:

> *"... that of extracting from the raw data the information which is most relevant for classification purposes, in the sense of minimizing the within-class pattern variability while enhancing the between class pattern variability".*

Strategies for feature selection aim to identify the features relevant to uncover the pattern in the dataset. Thus, with their application, the dimensionality of the data is usually reduced[95]. This step is almost always essential before modeling, especially for the case in which a large set of molecular descriptors is initially available. A smaller set of features decreases the computational cost of the training phase and the complexity of the final model. Furthermore, it improves the accuracy of the model and reduces the chances of overfitting. For a certain number of training samples, for any classifier, including more dimensions to the feature space improves model performance. However, after a certain threshold in the number of features, performance only deteriorates[96].

Several methodologies and techniques can be used for feature selection. In the work presented here, the techniques applied are based on filter and wrapper methods.

Filter by correlation with the class

The selection of features involves evaluating the worth of each attribute according to its correlation with the output variable. Correlation measures quantify the relationship between two variables. Therefore, this filtering strategy applies statistical tests rather than machine learning algorithms. In this work, as an initial step in feature selection, Pearson's correlation coefficient and Information Gain are used on regression and classification problems, respectively, to select top correlating variables.

Pearson's correlation coefficient (R) quantifies the linear dependence between two continuous variables. The measure outputs a score for the relationship between the variables. R indicates the strength and direction of such an association. The score varies from -1 to +1. The value –1 indicates that changes in one variable trigger proportional changes in the other variable but in the opposite direction. The value 0 reveals a lack of correlation between the variables. A score of 1 indicates perfect correlation, showing that the two variables change in the same direction. Given two variables x and y, and N samples, the equation to calculate Pearson's Correlation is as follows:

$$R = \frac{\sum_{i=1}^{N}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i-\bar{x})^2 \sum_{i=1}^{N}(y_i-\bar{y})^2}} \qquad (4)$$

Where x_i and y_i are the value of x and y for the i^{th} sample in the dataset, and $\bar{x}$ and $\bar{y}$ are the mean of variables x and y for the entire dataset, respectively. The ranking of the descriptors by their correlation with the class permits to identify and select top-ranked descriptors. Such selection implies using a threshold value. The identified descriptors can be fed to other feature selection techniques for further analysis.

Information Gain (IG)[97] is an entropy-based method that measures the information content provided by a variable Y to describe the information of a variable X. In other words, IG calculates the difference between the entropy of the variable X and the conditional entropy of X given a second variable Y. In feature selection algorithms, X corresponds to the outcome variable (e.g., class) and Y to a feature (e.g., molecular descriptor). IG is calculated as follows:

$$IG_c(X|Y) = H(X) - H_c(X|Y) \tag{5}$$

The term H(X) in eq. 5 measures the total information necessary to describe the distribution of the variable X and it is formulated as:

$$H(X) = -\sum_i P(x_i)\log_2(P(x_i)) \;\; i = 1,2 \tag{6}$$

where $P(x_i)$ is the probability of the class i, calculated as the fraction of the number of instances with value x_i by the total number of samples of X.

The term H_c(X|Y) in eq. 5 is the conditional entropy of variable X. It represents the amount of uncertainty remaining in variable X after introducing variable Y. This term is formulated as follows:

$$H_c(X|Y) = -\sum_j P_c(y_j)\sum_i P_c(x_i|y_j)\log_2(P_c(x_i|y_j)) \tag{7}$$

Where $P_c(y_j)$ corresponds to the probability of the set of cases c with values y_j for the variable Y. This is obtained as the ratio between the number of cases in the selected subset and the number of cases in the entire data set. $P_c(x_i|y_j)$ is the conditional probability of class x_i given the values y_j of variable Y. This is obtained as the fraction of the number of cases of class x_i in the selected subset and the total numbers of cases in the same subset.

The IG is always a value larger than or equal to zero. A value of zero indicates that the two variables are independent. Then, the larger the IG value, the larger the dependence between the variables. In the present work, I used the normalized Information Gain (NIG) calculated using the information content (entropy) of the class variable. Since NIG is a relative measure of the information provided by the feature to describe the dependent variable (eq. 8), it is a more intuitive magnitude than the absolute IG value.

$$NIG_c(X|Y) = \frac{IG_c(X|Y)}{H(X)} \tag{8}$$

In this way, descriptors are selected whose IG values exceed a certain percentage of the total information content of the class variable.

Filter by redundancy among features

The filters described above eliminate features with low or no correlation with the class. However, filter-based methods do not analyze multicollinearity among descriptors. Redundant descriptors do not provide novelty to the endpoint[98]. Moreover, they may significantly increase the computational cost of the learning process. A popular approach to remove redundant dimensions is the use of clustering methods. Clustering is the process of forming groups of features so that the distance between features in the same cluster is minimized, while the distance of those in different clusters is maximized. Various clustering algorithms can be used to identify highly similar descriptors. In this work, single-linkage clustering was employed for this purpose.

Single-linkage clustering is a technique that follows an agglomerative hierarchical methodology. Hierarchical clustering orders the samples based on a notion of similarity, to facilitate finding correlations in the data. The method starts by considering each observation as a separate cluster. Then, the closest points (based on a measure of similarity) are clustered together in each iteration. The algorithm ends when the data has formed a unique cluster.

This technique can be used as an unsupervised learning method for feature selection, to analyze the redundancy among the features in the dataset. The criterion for forming clusters with this algorithm is that members of different clusters cannot be found below a certain cut-off value according to the measure of diversity used (the opposite applies if a similarity measure is used). Different similarity measures can be used to define the relation between points. Here, the Spearman's rank correlation coefficient is employed for quantifying the association between every two variables in order to detect and remove redundant dimensions.

The *Spearman's rank correlation coefficient* is a nonparametric correlation coefficient that, comparable to Pearson's correlation coefficient, quantifies the strength and direction of the association between two variables. However, Spearman's correlation determines monotonic relationships, while the Pearson's correlation coefficient determines linear relationships between the variables. In a monotonic relationship, the values of both variables increase in the same direction, or as the value of one variable increases, the value of the other decreases.

To calculate the coefficient, each variable is ranked independently. Then, for each instance, the differences between the rank values are calculated and subsequently squared. The Spearman's rank correlation coefficient (R_s) is calculated as follows:

$$R_s = 1 - \left(\frac{6\sum d^2}{n^3 - n}\right) \tag{9}$$

where d is the difference between the ranks of two observations and n is the number of observations in the dataset. The value for R_s can change in the range from -1 to +1, indicating negative or positive associations of ranks, respectively. A R_s value of zero indicates no association between the ranks.

Wrapper-based methods

The Wrapper method[99] evaluates different subsets of features by applying a learning scheme. Such a scheme combines a specific learning algorithm, a search method, and an evaluation criterion to assess each selected subset. It follows a greedy search approach to analyze all the possible combinations of features. Cross-validation (see Section 4.3.2.7) is used as the test mode to avoid overfitting. The merit of subsets is determined using the evaluation measure of interest to assess model performance, for instance, Pearson’s correlation coefficient for regression or accuracy for classification (see Section 4.3.2.8). The method outputs the combination of features that delivered the best performance for the specified learning algorithm. The search problem is solved by using heuristic methods such as stepwise forward selection, backward elimination, or a combination of both methods.

Forward selection is an iterative process that initiates from an empty subset of features. Initially, the method evaluates the relevance of each feature (using the same evaluation measure as for evaluating the subsets) and adds the one with the highest merit to the set. Then, at each iteration, it evaluates each feature in combination with those already selected and adds to the subset the feature that best improves model performance. The algorithm repeats the process until adding additional features does not improve the performance of the model.

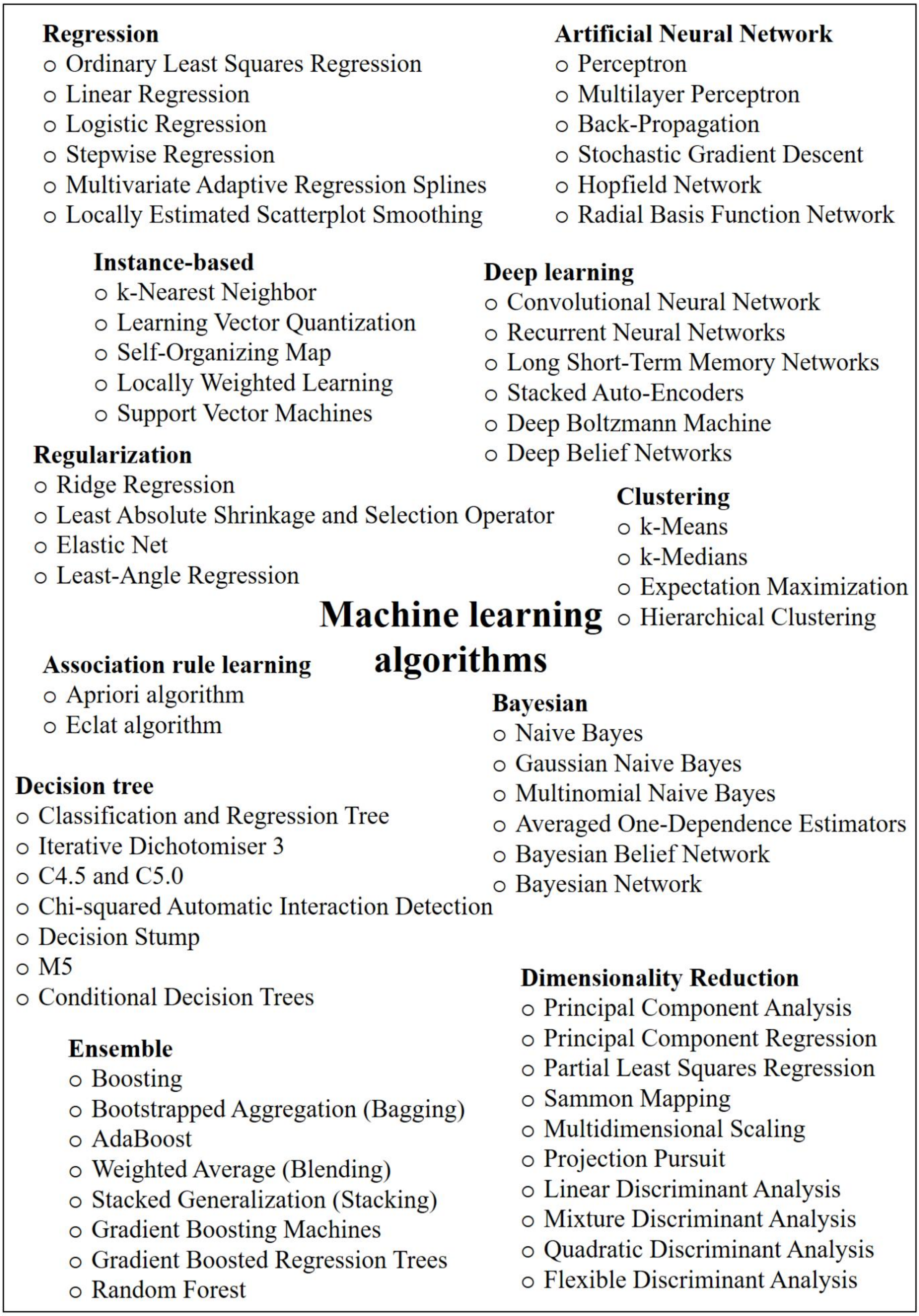
Backward elimination does the opposite to Forward selection. The algorithm starts with the set of all features. Then, at each iteration, the least significant feature (evaluated using the same evaluation measure as for evaluating the subsets) is removed. This process is repeated until the removal of additional features does not improve model performance.

Bi-directional selection combines forward selection and backward elimination to find the optimal subset of features. It applies forward selection to add a new feature to the subset. However, once the method adds a new variable to the set, it checks the significance of the already separated features. Then, in backward elimination, insignificant features are removed. This process is repeated until the optimal subset of features is found.

Wrapper methods are computationally intensive, especially for a highly dimensional dataset. However, the interaction with the classifier permits the identification of model features dependencies and to improve model performance.

4.3.2.5 Learning algorithms

Numerous ML algorithms exist, which may be grouped by similarity according to the mode of operation (**Figure 5**). Algorithms successfully applied in drug discovery are Naive Bayes, Support Vector Machines, the tree-based model Random Forest, and Artificial Neural Networks[100]. Deep learning revolutionized the field of drug discovery in recent years due to its ability to solve complex problems. However, such a strength relies on the availability of large volumes of data[92].



Machine learning algorithms

Regression
- Ordinary Least Squares Regression
- Linear Regression
- Logistic Regression
- Stepwise Regression
- Multivariate Adaptive Regression Splines
- Locally Estimated Scatterplot Smoothing

Artificial Neural Network
- Perceptron
- Multilayer Perceptron
- Back-Propagation
- Stochastic Gradient Descent
- Hopfield Network
- Radial Basis Function Network

Instance-based
- k-Nearest Neighbor
- Learning Vector Quantization
- Self-Organizing Map
- Locally Weighted Learning
- Support Vector Machines

Deep learning
- Convolutional Neural Network
- Recurrent Neural Networks
- Long Short-Term Memory Networks
- Stacked Auto-Encoders
- Deep Boltzmann Machine
- Deep Belief Networks

Regularization
- Ridge Regression
- Least Absolute Shrinkage and Selection Operator
- Elastic Net
- Least-Angle Regression

Clustering
- k-Means
- k-Medians
- Expectation Maximization
- Hierarchical Clustering

Association rule learning
- Apriori algorithm
- Eclat algorithm

Bayesian
- Naive Bayes
- Gaussian Naive Bayes
- Multinomial Naive Bayes
- Averaged One-Dependence Estimators
- Bayesian Belief Network
- Bayesian Network

Decision tree
- Classification and Regression Tree
- Iterative Dichotomiser 3
- C4.5 and C5.0
- Chi-squared Automatic Interaction Detection
- Decision Stump
- M5
- Conditional Decision Trees

Dimensionality Reduction
- Principal Component Analysis
- Principal Component Regression
- Partial Least Squares Regression
- Sammon Mapping
- Multidimensional Scaling
- Projection Pursuit
- Linear Discriminant Analysis
- Mixture Discriminant Analysis
- Quadratic Discriminant Analysis
- Flexible Discriminant Analysis

Ensemble
- Boosting
- Bootstrapped Aggregation (Bagging)
- AdaBoost
- Weighted Average (Blending)
- Stacked Generalization (Stacking)
- Gradient Boosting Machines
- Gradient Boosted Regression Trees
- Random Forest

Figure 5. Machine learning algorithms

The ML models developed in this work leveraged Support Vector Machines and Random Forest algorithms. Therefore, a description of these two algorithms appears next in this section.

Support Vector Machines

Support Vector Machines (SVM)[101] is a supervised learning algorithm that has proven robust in bioinformatics studies. Given a set of observations of the form $(x_i, y_i), \ldots, (x_N, y_N)$ from $\mathbb{R}^d$x $\mathbb{R}$, where $x \in \mathbb{R}_d$, and $y \in \{-1,1\}$ for a binary classification problem, SVM aims to find the optimal hyperplane separating the observations of the two classes. The hyperplane can be represented as follows:

$$W^T X + b = 0 \tag{10}$$

Where W is the normal vector of weights to the hyperplane, and b is the offset of the hyperplane.

If the training data is linearly separable, it is possible to select two parallel hyperplanes that divide the data into two classes (**Figure 6**). The hyperplanes are selected by increasing the distance (margin) between the classes as much as possible. The hyperplane with the maximum margin is the one located halfway between both hyperplanes. The margin between the hyperplane and the classes is $\frac{2}{\|w\|}$. Thus, minimizing w converges to the maximum margin.

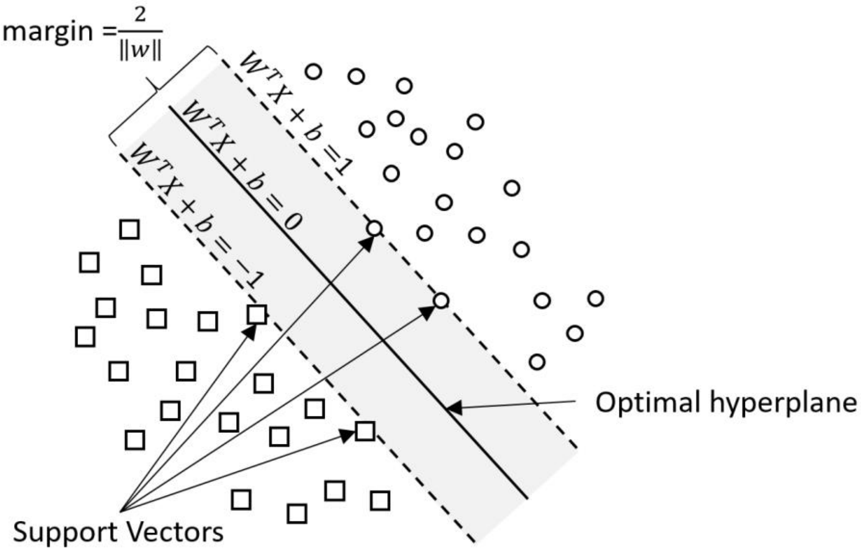


Figure 6. Hyperplane with the maximum margin for a SVM trained on a data set with two classes (adapted from Pino et al.[102])[Note4]

The optimization is performed under the following two constraints to allow the separation of the two classes to the corresponding side of the hyperplane:

$$g(x_i) = \begin{cases} (w^T x_n + b) \leq -1 \; if \; y_i = \; -1 \\ (w^T x_n + b) \geq 1 \; if \; y_i = 1 \end{cases} \tag{11}$$

[4] Published under the terms of the Creative Commons Attribution License (CC BY 4.0), which permits the unrestricted use and reproduction of the image, on condition that the original author and source are cited.

The resolution of w is solved as an optimization problem usually presented as a minimization:

$$\begin{cases} \min\limits_{\mathrm{w,\,b}} \frac{1}{2} w^T w \\ \text{such that } y_n(w^T x_n + b) \geq 1, for\ n = 1, \ldots, N \end{cases} \tag{12}$$

Equation 12 solves a "hard margin" classifier and is used when the data are linearly separable. For slightly non-linearly separable data, the optimization is modified to penalize those points violating the margin. Thus, a term to balance maximizing the margin and minimizing the error is added to the optimization problem:

$$\begin{cases} \min\limits_{\xi_n \in \mathrm{R}^+,\mathrm{w,\,b}} \frac{1}{2} w^T w + C \sum_{n=1}^{N} \xi_n \\ \text{such that } y_n(w^T x_n + b) \geq 1 - \xi_n, for\ n = 1, \ldots, N\ and\ \xi_n\ \geq 0\ for\ n = 1, \ldots, N \end{cases} \tag{13}$$

Where ξ measures the slack of the violation (distance of the data point to the margin) and C is the degree to which the violations are allowed. Large C values imply higher complexity that may lead to overfitting, as violating the margin will not be allowed. Small C values, on the other hand, lead to lower complexity but allow violating the margin very frequently. The classifier learned by solving this problem is called a "soft margin" support vector classifier.

Strong non-linearity relations are modeled using a kernel function $K(x_i, x_j)$, which transforms the input space X into a higher-dimensional Z space where the data can be linearly separable. The peculiarity of kernel functions is that they calculate high-dimensional relationships between each pair of observations as dot products without computing the coordinates of the data in the Z space. This operation is called the kernel trick, and it is computationally more effective than explicitly computing the coordinates. Most SVM models are trained using a combination of both, soft margin (tolerance to misclassifications) and kernel trick, to find the best hypothesis in nonlinear data. The final model is found as an optimization problem using the Lagrange multipliers and quadratic programming. The output of the optimization step is the identification of some points called support vectors, which are those achieving the margin and thus used to define the final hyperplane. SVM can be used for regression in addition to classification. The regression method is called Support Vector Regression (SVR).

A simple methodology to train a model using SVM is as follows:

1. Normalize the attributes.
2. Select the most appropriate parameters.
    a. Parameter C (cost, complexity).
    b. Define the Kernel to be used. Popular Kernel functions are:
        i. Linear: $K(x_i, x_j) = {x_i}^T x_j$

ii. Polynomial: $K(x_i, x_j) = ({x_i}^T x_j + c)^d$, where d is the degree of the polynomial.

iii. Radial Basis Function: $K(x_i, x_j) = e^{(-\gamma \|x_i - x_j\|^2)}$, where γ is Gamma, a parameter to define the influence of individual observations.

3. Build the model with the best parameters and the training set.

Random Forest

Random Forest (RF)[103] is an ensemble technique that combines the output of several independently trained models to provide a unique final prediction. Each model corresponds to a decision tree, trained on a dataset extracted from the main dataset using the bootstrapping (Bagging) technique. Bootstrapping selects randomly a set of features and instances (with replacement) from the original data set. The learning model produces a different g using each dataset and the output prediction is obtained as a consensus by applying majority voting rules. Each model has a different perspective of the modeled dataset. Thus, the combination of weak models is expected to improve the generalization capabilities of the ensemble model.

Given a data set of N instances and M attributes, a *Random Forest* classifier of *k* trees is built according to the following algorithm:

1. A data set of *N* instances is randomly selected, keeping a similar class distribution. The remaining instances are used as test cases.
2. A subset of *m* attributes is randomly selected from the total (*M*) in the initial dataset.
3. The best partition between the *m* attributes is identified and two new nodes are formed.
4. The tree is completed by repeating steps 2 and 3 until each node reaches the maximum level of purity.
5. The steps from 1-4 are repeated *k* times.

For prediction, a new case is evaluated by applying the rules of each tree independently. The case then obtains a classification per tree corresponding to the class of the reached terminal node. The final classification is obtained by the majority vote of the constructed *k* trees.

4.3.2.6 Optimization of hyperparameters

Each algorithm has one or several parameters that control hypothesis generation, e.g., the cost or complexity (C) in SVM. These parameters are known as hyperparameters as they differ from other parameters, such as those learned, e.g., weights of linear function. Different hyperparameter setups work differently on distinct datasets[104]. Thus, several values are usually assessed in a tuning process to find the most suitable configuration for the modeled endpoint. This optimization approach generally improves model performance over the default setting of algorithms supplied in ML libraries[105]. Hyperparameter optimization or tuning follows feature selection (if performed) and is part of the training process.

Hsu et al.[106] described an approach for adjusting the hyperparameters of SVMs. This strategy consists of a grid search that varies the hyperparameters' values to cover different possible values.

For instance, the parameter C may change from a small (e.g., 2^{-5}) to a larger quantity (e.g., 2^5), with an increase determined by a stepwise value (e.g., increasing the exponent by 0.5). Each hyperparameter configuration generates a model. The generalization power of each model is assessed and stored (e.g., using cross-validation and development sets) for further analysis. Finally, the hyperparameter configuration that produced the best-performing model is selected. This grid-search strategy was followed in this work to develop the different ML models.

4.3.2.7 Evaluation approaches

Internal and external validation approaches (**Figure 7**) may increase the chances of training a robust and reliable model. In both schemes, instance selection and data partitioning are conducted to create various subsets of the data that serve different purposes. Subsets formation may include strategies such as stratified partitioning, repeatable random sampling, and over- or under-sampling of the minority or majority class, respectively.

Internal validation: Popular approaches for internal validation are k-fold cross-validation (k-fold CV) and the use of a development set. In k-fold CV, the dataset is distributed randomly into k disjoint (relatively equal size) folds. Then, every fold is used once as a test set to evaluate the performance of a model fitted on a training set formed by the other k-1 folds. Finally, the average performance on the test sets of the k-generated models is calculated. The variability of the accuracy estimations obtained from the random division of samples provides an estimate of the generalization power of the model[107]. A development set is a separate set of samples used as an external test set for the selection of the model during the training steps, e.g., in hyperparameters optimization.

External validation: One or more test sets can be separated from the initial data for external validation. These sets aim to assess the generalization capabilities of the model by predicting the output of unseen data[108]. External validation provides a way to estimate how close the in-sample error is to the out-sample error, and it can be conducted retrospectively by using validated test set(s). External validation can also be performed prospectively by predicting the class of new compounds, to be later verified experimentally[109].

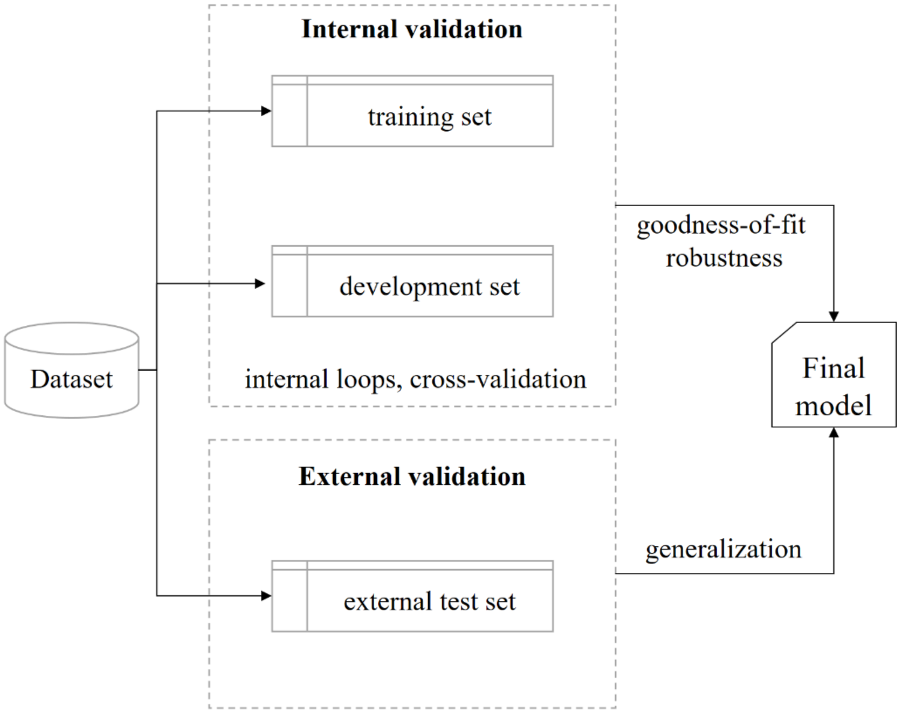


Figure 7. Internal and external validation.

4.3.2.8 Performance measures

Performance measures that evaluate the quality of a model are part of every ML pipeline. Their objective is to numerically expose the success of the learning task and compare different models. There are several performance measures to work with, and the first criterion for the selection relies on whether the learning task corresponds to a classification or regression problem. The performance measures used in this work are summarized below.

Classification measures

In classification, the error is measured as the binary difference between the predicted and true output values. Then, *true positive* (TP), *true negative* (TN), *false positive* (FP), and *false negative* (FN) predictions are counted and introduced in a confusion matrix (**Table 2**).

Table 2. Confusion matrix for a binary classification problem.

		Predicted	
		P	N
Actual	P	TP	FN
	N	FP	TN

The confusion matrix permits to determine the overall quality of the model based on several measures (**Table 3**).

Table 3. List of performance measures used in the classification problems addressed in this work.

Measure / Formula	**Description**
Accuracy (Acc) $Acc = (TP + TN)/(P + N)$	The ratio of correctly identified predictions and the total number of instances.
Sensitivity, aka Recall (Sn) $Sn = TP/(TP + FN)$	The ratio of samples correctly identified as positive and the total number of positive samples.
Precision (Pr) $Pr = TP/(TP + FP)$	The ratio of samples correctly identified as positive and the total number of samples predicted as positive.
Specificity (Sp) $Sp = TN/(TN + FP)$	The ratio of samples correctly identified as negative and the total number of negative samples.
Matthews Correlation Coefficient (MCC) $MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$	Association between two variables. (Similar interpretation as the Pearson's correlation coefficient for regression).
F1-Score $\mathrm{F1} = 2 * \frac{Sn * Pr}{Sn + Pr} = \frac{TP}{TP + {}^{1}/_{2}\,(FP + FN)}$	The harmonic mean of precision and recall.
Prevalence-corrected precision (PCPr) $PCPr = \frac{Sn}{Sn + r(1 - Sp)}$	Precision values normalized for comparing different test sets, i.e. with different sizes of positive and negative samples in the data. The value r is the new ratio of negative and positive data samples.

Precision and Recall measures are valuable for evaluating the performance of a model aimed at virtual screening. For instance, for predicting the likelihood of interaction between a list of peptides and a protein receptor, it is relevant to avoid false positives when selecting top-ranked peptide candidates for experimental validation. However, there is a trade-off between these two measures, and model selection based purely on one of them usually limits the performance of the other. The precision-recall curve (PRC) allows visualizing this trade-off for different thresholds (score separating the classes) values. A high area under the generated curve indicates high precision (low false positive rate) and high recall (low false negative rate) values. An accurate classifier has a high precision value, while a classifier with a high value of recall correctly identifies most positive instances. Thus, an ideal classifier balances both measure values with relatively high scores.

Regression metrics

In regression, the error accounts for the numerical difference between the predicted and actual outcome values. Then, several measures may determine the error rate for the model (**Table 4**). Additionally, correlation measures (Pearson, Spearman, and Kendall) permit the calculation of the overall relationship between the predicted and the actual output variables.

Table 4. List of performance measures used in the regression problems addressed in this work.

Measure - Formula	Description
Mean Absolute Error (MAE) $MAE = \frac{1}{N}\sum_{i=1}^{N}\lvert y_i - \hat{y_i}\rvert$	Average of the difference between the actual $\hat{y_i}$ and predicted y_i values. N is the total number of instances in the set.
Pearson's correlation coefficient (R) $R = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2 \sum_{i=1}^{N}(y_i - \bar{y})^2}}$	Linear dependence between two continuous variables x and y. $\bar{x}$ and $\bar{y}$ are the mean of variables x and y for the entire dataset, respectively.
Kendall's tau correlation coefficient (τ) $\tau = \frac{N_c - N_d}{\sqrt{(N_c + N_d + N_t) * (N_c + N_d + N_u)}}$	N_c and N_d are the number of concordant and discordant pairs, respectively. N_t and N_u are the number of ties in the order of each variable, respectively.
Enrichment Factor (EF) $EF_I = \frac{\left[N_{positives}^{top_i} / I\right]}{\left[N_{positives}/N_{total}\right]}$	I is a number of top-ranked instances as predicted by the model. $N_{positives}^{top_i}$ is the number of positive predictions in the top I. $N_{positive}$ is the number of positive samples in the dataset, and N_{total} is the total number of instances in the dataset.

The evaluation of the model in regression usually relies on a measure of correlation (e.g., Pearson's) and a measure of distance (e.g., MAE) between predicted and actual values. The former estimates the dependence of both predicted and output variables. This can be of use, for instance, to assess the ranking power of the model. The measure of distance calculates, on average, how close the prediction is to the actual value. Furthermore, other metrics may be a good choice according to the information available. For instance, τ permits assessing the ranking power of a model on a test set of few samples. Likewise, EF may be used to evaluate a classifier on a test set when the predicted and the actual values have different magnitudes, i.e., a model predicts BA as binding free energy, but the test samples have BA indicated as IC_{50} values.

4.3.2.9 Definition of the applicability domain

The definition of the applicability domain (AD) aims to deliver information on the coverage or similarity of a query sample among the samples used to train the model. It is an informative tool needed to express the scope and limitations of a predictor, which can help to judge the reliability of the model's predictions.

As Netzeva[110] expressed, "*this need is based on the fact that (Q)SARs are reductionist models, which are inevitably associated with limitations in terms of the types of chemical structures, physicochemical properties and mechanisms of action for which they can generate reliable predictions".*

Several approaches can be used in a multivariate space to estimate the AD of a model. Popular methods use ranges, geometry, distances, and probability density distribution functions[110]. In this work, the range approach estimates the projection of query samples into the AD of the models. Range-based methods analyze the projection of a data point into the training data by using the range values. That is, by checking that each descriptor value is within the range of values of the same descriptor in the dataset used to train the model. If the value of a descriptor exceeds the determined values range, a warning message indicates that the test sample is out of the AD. This approach is straightforward, although it cannot detect holes in the training data, namely regions with scarce data representation. However, range-based methods are easy to apply and computationally efficient, and thus a suitable choice for virtual screening studies.

4.4 Machine learning in drug discovery

The field of ML has grown in the last decades, and it has proven successful in different areas, such as image, video, finance, robotics, autonomous driving, and so forth. Such success, and the need to apply hybrid techniques to cope with the high complexity of drug design and development, have increased the interest in ML-based methods as tools to support drug discovery in recent decades as well.

One of the applications of ML techniques in bioinformatics is QSAR development. Hansch et al. published the first work of QSAR modeling in 1962[42]. Since then, around 20,000 papers on QSAR applications for computer-aided drug discovery have been published, as reported by Muratov et al. in 2020[111], with the highest growth in the number of publications taking place in recent years. From its first usage, QSAR development has evolved from applying simple regression approaches to using ML techniques capable of analyzing nonlinear data[68]. Consequently, ML-based QSAR models have gained relevance in CADD as methods to support the process of drug design and optimization.

4.4.1 The principles of the *Organization for Economic Co-operation and Development* for the development of *Quantitative Structure-Activity Relationships* (QSAR) models

Almost two decades ago, the Organization for Economic Co-operation and Development (OECD) established five principles[112] to harmonize and safely guide the development of QSAR models:

"To facilitate the consideration of a (Q)SAR model for regulatory purposes, it should be associated with the following information:

1) a defined endpoint

2) an unambiguous algorithm

3) a defined domain of applicability

4) appropriate measures of goodness-of-fit, robustness and predictivity

5) a mechanistic interpretation, if possible".

The OECD principles describe a decision-making process, and each choice will ultimately determine the robustness and reliability of the developed model. The first principle refers to having a well-defined endpoint and the information required to fit and use a model. An endpoint may be any physicochemical, biological, or environmental effect that can be measured and modeled. The second principle is related to the need for transparency in the description of the modeling algorithm to be reproducible. The third principle refers to the importance of establishing the model limitations through an AD definition. Thus, it is possible to specify for which input samples the model produces reliable predictions. The fourth principle is concerned with using different measures to evaluate the internal performance (as represented by goodness-of-fit and robustness) and external performance (as determined by external validation) of the model. Lastly, the fifth principle refers to the desire for mechanistic interpretations of the association between descriptors and the modeled endpoint. Yet, the OECD recognizes that it is not always possible to fulfill this principle, for what other considerations applied by principles third and fourth may be sufficient to accept a production model.

4.4.2 Standard procedure to develop machine-learning-based QSAR models

Many ML-based QSAR models have been introduced in the last decades to address biological problems. The standard methodology (**Figure 8**) used to build ML-based QSAR models includes the following steps[113]:

1. Pre-processing of the data, i.e., data curation or standardization.
2. Split the data into training, development (optional), and test sets.
3. Features generation (e.g., molecular descriptors).
4. Selection of the learning algorithm.

5. Model training, including the application of internal validation approaches.
6. Evaluation of the performance of the model using appropriate metrics.
7. External validation of the selected model based on the internal validation step.

There are several aspects to contemplate when creating ML-based QSAR models. The performance of a model will be as good as the data used to train it. Thus, data curation involves strategies before and after feature generation, e.g., for outlier detection. The splitting of the data into several subsets is usually random, but the cases in each set shall be as distant as possible to each other. Descriptors used as features must be informative enough to capture the characteristics of the data that correlates with the endpoint. Thus, feature generation involves determining which encoding approach is suitable to represent the information to be modeled, i.e., molecular descriptors for protein sequences or 3D structures. Feature selection techniques may incorporate information from the class. Those techniques involving the outcome variable and the interaction with the learning algorithm are part of model training. Furthermore, model training usually includes the optimization of the hyperparameters. The learning algorithm may be selected after assessing several techniques on the data set to identify the most appropriate algorithm or by evidence of good performance for the modelled endpoint. The evaluation of the model should include several validation strategies, e.g., performance in cross-validation or development/tuning set. The final model must be performant on external test set(s).

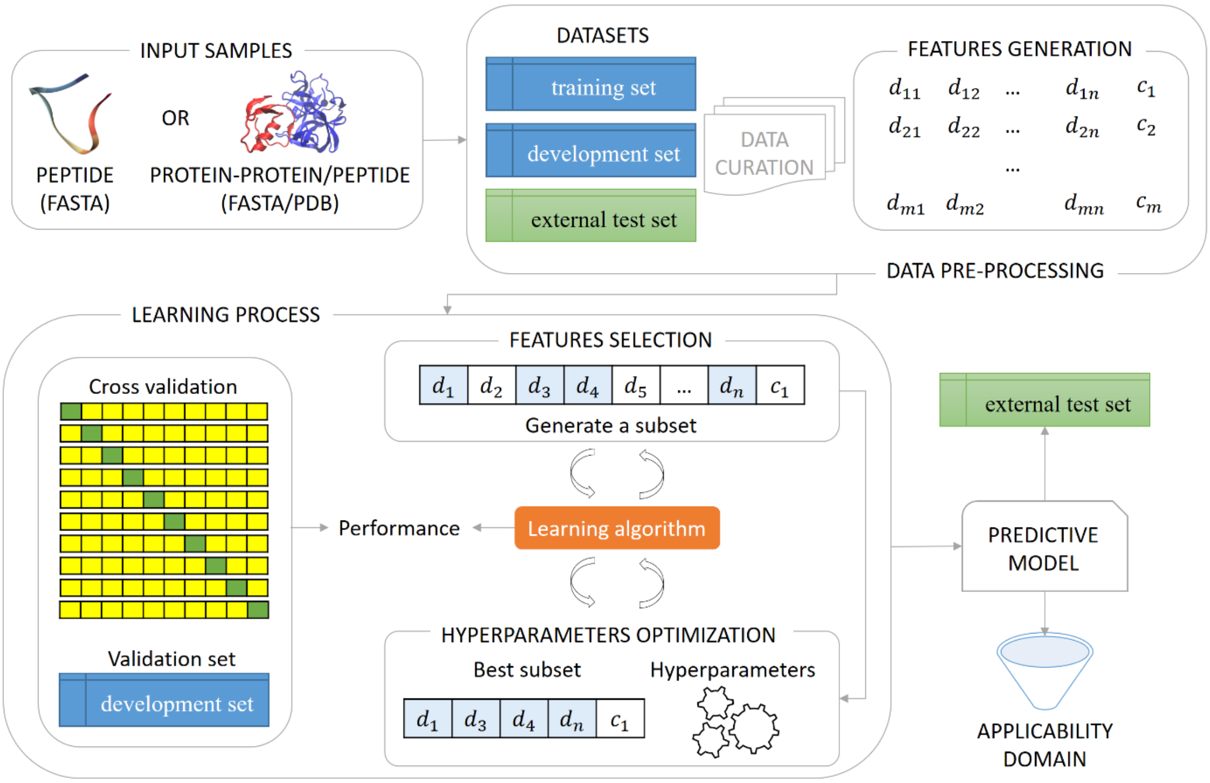


Figure 8. Standard process to develop ML-based QSAR models

Each time a model is assessed, the relation between the error in training (in-sample error) and error in test (out-sample error) is analyzed to examine how the in-sample error tracks the out-sample error. As shown in **Figure 4**, the more complex the model, the most likely the in-sample error will be low, but the chances of overfitting increase (out-sample error). By contrast, if the model is too simple, both errors will probably be close. Yet, with such high error values, the model will lack practical utility. As a rule of thumb, the difference between the training and test errors should be below 15% of the training error. The acceptance of the model is dictated by how it compares with respect to other state-of-the-art models after assessment on external, preferably benchmark, test set(s) using performance measures.

4.4.2.1 Implementation of web servers

Once a model is created and validated, its applicability will depend on its availability and ease of use. Thus, any proposed ML model should be available as a web server or as at least as a standalone application. Although several ML models are publicly available via a web server, open-access tools increase the usability of the model in academic and industrial sceneries. Yet, only a limited amount of publications in the field of drug development offer open-access websites[100].

4.4.3 Applications and importance of machine learning in drug discovery

The applications of ML techniques to create predictive models for drug discovery can be broadly grouped, according to the problem addressed, into drug mechanisms, drug properties, and drug repurposing[92].

Drug mechanisms

ML models may be used to predict the likelihood of interaction (interaction/non-interaction) between the drug and the target receptor, but also the binding affinities of such interaction. Targets can be enzymes, ion channels, nuclear receptors, and G protein-coupled receptors, being those protein-related the most studied targets. Likewise, several types of molecule can interact with receptors, such as small organic molecules, peptides, and other proteins[114].

The identification of drug-target interactions is frequently the first step of the drug discovery process, aiming to reduce the initial number of candidates. Thus, ML methods have been developed to predict interactions between protein and prospective drugs and to screen new drug candidates effectively and efficiently[115-122]. Most state-of-the-art methods predict protein-protein interactions or protein-ligand interactions, where ligands are small molecules or peptides. Such studies can contribute to understanding the mechanism of action of the drug, the pathology of the disease, and possible side effects of the drug. Moreover, the identification of

PPIs allows detecting protein complexes, identifying domain interactions, identifying proteins involved in disease pathways, and developing effective strategies in drug design[123].

Drug properties

The pharmacokinetic properties of a compound are essential to regulate its usage. Thus, one of the most relevant biological problems involves analyzing ADME, and toxicity properties. Computational attempts have aimed to create predictive models for human oral and intestinal absorption, Caco-2 permeability, carcinogenicity, clearance, identification of P-glycoprotein substrates, phospholipidosis, blood-brain barrier permeability, cytochrome P450 activity (CYP450), and mutagenicity. Priya et al.[124] published a review article that summarized ML-based studies aiming to predict such properties. These studies leveraged random forest, artificial neural networks, SVM, and deep-learning algorithms.

Antimicrobial compounds against bacteria, viruses, parasites, or fungi are well studied[125, 126]. Antibiotics are the main treatment against bacterial infections[127]. However, the overuse of antibiotics has increased bacterial resistance to those antibiotics available on the market. Such increase is, however, in notable disagreement with the low number of new antibiotics introduced in recent years. Therefore, it is necessary to discover novel compounds tackling multi-resistant organisms[128]. Multi-drug resistance also involves parasites, such as the protozoan parasite *plasmodium falciparum* producing malaria. Thus, antimicrobial ML models to study parasites have also been developed[129]. Furthermore, viruses causing severe diseases such as the acquired immunodeficiency syndrome (AIDS), Ebola or COVID-19 have been studied using ML techniques[130].

Many efforts aim to develop treatments for cancer, a public health problem causing the death of millions of people annually. Several therapeutic targets are the focus of cancer treatments. ML techniques find applications to create methods to predict the activity of drugs on known cancer-related targets, such as G-protein-coupled receptors (GPCRs), which are involved in cell signaling mechanisms and whose alteration may lead to cancer progression[100]. Thus, ML methods can be applied for the identification and development of anticancer agents[131-134].

Drug repurposing

The use of existing drugs for therapeutic purposes others than those already established is known as drug repurposing. The aim of this strategy is to explore the possible usage of known drugs as an alternative to overcome the expensive and time-consuming process of discovering novel compounds. For this, data related to the drug, such as its chemical structure, target, and side effects are leveraged, among others. ML techniques can be used to build models for drug repurposing, for instance, to predict the class of therapeutic drugs, and for cancer cell line response to drug treatment. Park published a comprehensive review article on computational methods for drug repurposing[135].

Drug repurposing is important to face epidemics[136]. In 2019, with the novel coronavirus variant (SARS-CoV-2) outbreak, several approaches immediately aimed to find treatments to tackle the disease. Because of the urgency for treatment and the lack of knowledge of the disease, drug repurposing was one of them[137]. Beck et al.[138] used a previously introduced ML model to screen datasets of known antiviral drugs. The screening aimed to find compounds that potentially disrupt molecular elements of SARS-CoV-2 (e.g., proteinase, RNA-dependent RNA polymerase). Beck's study exemplifies how predicting the interactions between target and drug finds applications in drug repurposing. Drug repositioning with ML techniques also focused on antimicrobial compounds to address COVID-19[139].

The complexity associated with human diseases requires methods able to explore a broader chemical space to facilitate the identification of novel molecules to be synthesized. With the increase of available data and the development of information and computational technologies, the application of ML has become a valuable tool for drug design and development. The impact of ML in drug discovery is directly associated with the speed that predictive models can offer to accelerate the research process and decrease the cost and risk of clinical trials. QSAR is a promising technique in drug development because it allows processing large compound datasets fast and without losing much precision[40]. QSAR models allow proposing drugs with specific biological properties[140]. Estimates indicate that, by using ML, the introduction of novel drugs requires less than 1/3 of the time and cost of traditional drug development[141]. Drug discovery covers more than 35% of the Artificial Intelligence/Machine Learning market. With an annual growth rate of 53%, the market was estimated to reach US$8 billion in 2022[40]. Overall, the use of ML brings automation and sophistication to the drug development process. Nevertheless, the definition of thresholds to select the best candidates is subjective. It is domain-specific, and it requires the expertise of humans.

4.4.4 Machine learning for the identification and study of bioactive peptides

4.4.4.1 State-of-the-art methods

ML techniques have already found applications in the analysis of bioactive peptides. In 2019, Basith et al.[142] published a review article citing a comprehensive list of ML models trained on peptide databases. The described methods were used for screening function-specific therapeutic peptides, such as anticancer[143-157], antihypertensive[158-160], antitubercular[161-164], anti-inflammatory[155, 165-167], quorum-sensing[155, 168, 169], and cell-penetrating peptides[155, 170-180]. Most of these approaches used publicly available datasets and various ML algorithms, mainly SVM and RF.

A body of research also addressed predicting antimicrobial peptides[181-183] and putative functional peptides in this class, e.g., antibacterial peptides[184-186], and to discriminate among several function-related classifications such as anti-inflammatory, antiviral[130, 187], and antifungal[188] peptides [189-196]. Furthermore, ML models have found applications to study antiangiogenic[197-200] and immunosuppressive peptides[201, 202]. To fight the AIDS disease, anti-HIV-1 peptides were investigated using ML techniques[203-205].

These studies show how most state-of-the-art models predict function endpoints (drug properties) and focus on exploring the structures of various peptide sequences. Interestingly, the interactions between peptides and their potential targets have been less investigated by means of ML[206]. State-of-the-art work suggests that a next generation of ML-based methods is needed to develop peptide-based pharmaceutics[142].

Currently, the study of protein-peptide interactions based on databases of only protein-peptide pairs is mostly performed in specific contexts, for instance, to predict peptide binding to major histocompatibility complexes (MHC) classes I, II[207, 208]. Otherwise, protein-peptide interactions are part of datasets used to train predictors of protein-protein or protein-ligand interactions, in which ligands include mostly small molecules. In the context of the prediction of the BA of protein-peptide complexes and to the best of my knowledge, prior our work there was no publicly available ML model trained on only protein-peptide BA data. Due to the specific characteristics of peptides, such as low systemic stability, poor membrane permeability, poor oral bioavailability, low solubility, and fast clearance, ML methods trained on curated protein-peptide datasets are necessary.

4.4.4.2 Challenges of state-of-the-art methods

ML-based QSAR models for proteins and peptides analysis is a field of ample research in CADD. One approach is the prediction of PPIs as a classification problem (interaction/non-interaction). This modeling strategy delivered several PPI classifiers. However, some drawbacks were reported for those classifiers. For instance, Park[209] analyzed existing sequence-based PPIs predictors and identified some issues affecting their robustness, as listed below:

1. The approach adopted for encoding the protein pairs.
2. The effectiveness of the features representation (for those ML-based methods).
3. The approach followed to assess performance, including the measures used for the evaluation of the model and the lack of comparisons with previously introduced methods.

Moreover, the lack of verified non-PPI samples is another factor affecting PPI classification. The scarcity of negative samples requires strategies to create the negative class and improve the performance of the model.

The prediction of the BA of protein-protein and protein-ligand complexes using regression to calculate the strength of the interaction is another widely studied area[118-122, 210, 211]. In recent years, the availability of structures increased, from X-ray, NMR, electron microscopy, homology modeling, and ML methods[141]. Thus, most ML models leverage structural (3D) datasets for BA prediction. However, most models are based on protein-protein or protein-ligand datasets and have low representation of protein-peptide interactions. For those methods, the performance of models on protein-peptide complexes remains unstudied, and their applicability to the investigation of protein-peptide interactions is thus limited. Therefore, ML-based QSAR models specifically trained on protein-peptide BA databases are in demand.

Likewise, the creation of AMPs predictors receives much attention. However, most methods deliver limited precision when predicting the specific function of putative AMPs, such as antibacterial peptides.

Frequently, the protocols used to create many state-of-the-art ML-based methods for the mentioned endpoints are insufficiently validated, one of the problems noted by Park[209]. The evaluation of the models on additional external test sets, including novel experimental data, is encouraged to avoid high variance in future predictions. Moreover, most methods miss the definition of an applicability domain to use the model. In the absence of an applicability domain, it is difficult to discard unreliable predictions in VS. Additionally, several methods lack a web server implementation for their usage. Sometimes, those tools with a web server do not offer the possibility to perform VS for optimizing the primary structure of putative peptides interacting with a target protein. Consequently, identifying and optimizing promising peptides targeting PPIs in a cost-effective and time-efficient manner remains an area with room for improvement.

5 Objectives

Peptide development, although challenging, represents a promising opportunity for modern drug design. In this endeavor, the application of *in silico* methods for the virtual screening of peptide libraries can be an alternative to reduce the time and costs of this process. One approach to conduct virtual screening is the use of ML-based QSAR predictors for various peptide-related endpoints. In the scope of protein-peptide associations, several predictors of protein-protein and protein-ligand interactions have emerged in the last decades. However, these models suffer from low generalization capabilities, with high variance when predicting unseen data. Moreover, most methods contemplate peptides the same way as proteins or small organic ligands, underestimating the specificity of short peptide sequences.

Similarly, due to the urgent need of novel therapies to address multi-drug resistance, predictors of AMPs have been introduced. However, most methods are not intended to predict specific functions, such as antibacterial activity, and those that can consistently address this task have limited precision and lack information on potential targets. Consequently, novel computational methods aimed to accurately identify and optimize bioactive peptides are needed. The aims of my work were to leverage machine learning techniques to develop novel tools for the analysis of bioactive peptides by:

i. Developing a sequence-based predictor of protein-protein and protein-peptide interactions applicable to the identification of lead compounds from extensive *in silico* screening of protein-peptide and protein-protein interactions.

ii. Developing predictors of protein-protein and protein-peptide binding free energies based on 3D structures, with application to mutagenesis experiments and protein engineering.

iii. Contributing to the development of predictors of antibacterial peptides and the Gram-staining type of targeted bacteria. This tool allows the identification of function-specific lead peptides and derivatives thereof.

iv. Implementing a web platform permitting (*i*) the generation of ProtDCal molecular descriptors for data-driven studies, and (*ii*) the application of the developed ML-based tools for the virtual screening in the early steps of peptide discovery.

6 Publications

I. ***PPI-Detect: A support vector machine model for sequence-based prediction of protein–protein interactions***

Romero-Molina, S.; Ruiz-Blanco, Y. B.; Harms, M.; Münch, J.; Sanchez-Garcia, E., PPI-Detect: A support vector machine model for sequence-based prediction of protein–protein interactions. *Journal of Computational Chemistry* 2019; 40(11): 1233-1242.

https://doi.org/10.1002/jcc.25780

II. ***PPI-Affinity: A Web Tool for the Prediction and Optimization of Protein–Peptide and Protein–Protein Binding Affinity***

Romero-Molina, S.; Ruiz-Blanco, Y. B.; Mieres-Perez, J.; Harms, M.; Münch, J.; Ehrmann, M.; Sanchez-Garcia, E. PPI-Affinity: A Web Tool for the Prediction and Optimization of Protein–Peptide and Protein-Protein Binding Affinity. *Journal of Proteome Research* 2022; 21(8): 1829–1841.

https://doi.org/10.1021/acs.jproteome.2c00020

III. ***ABP-Finder: A Tool to Identify Antibacterial Peptides and the Gram-Staining Type of Targeted Bacteria***

Ruiz-Blanco, Y.B.; Agüero-Chapin, G.; Romero-Molina, S.; Antunes, A.; Olari, L.-R.; Spellerberg, B.; Münch, J.; Sanchez-Garcia, E. ABP-Finder: A Tool to Identify Antibacterial Peptides and the Gram-Staining Type of Targeted Bacteria. *Antibiotics* 2022; 11(12): 1708.

https://doi.org/10.3390/antibiotics11121708

IV. ***ProtDCal-Suite: A web server for the numerical codification and functional analysis of proteins***

Romero-Molina, S.; Ruiz-Blanco, Y.B.; Green, J.R.; Sanchez-Garcia, E. ProtDCal-Suite: A web server for the numerical codification and functional analysis of proteins. *Protein Science* 2019; 28(9): 1734-1743.

https://doi.org/10.1002/pro.3673

6.1 Publication I

Author contributions

PPI-Detect: A support vector machine model for sequence-based prediction of protein–protein interactions

Sandra Romero-Molina, Yasser B. Ruiz-Blanco, Mirja Harms, Jan Münch, Elsa Sanchez-Garcia

Research Article in:	**Journal of Computational Chemistry**
Impact factor:	3.672 (2021)
Published online:	February 15th, 2019
DOI:	10.1002/jcc.25780
Citations	44 (Dec 6th, 2022)

Complete citation from source:

Romero-Molina, S.; Ruiz-Blanco, Y. B.; Harms, M.; Münch, J.; Sanchez-Garcia, E., PPI-Detect: A support vector machine model for sequence-based prediction of protein–protein interactions. *Journal of Computational Chemistry* 2019; 40(11): 1233-1242. https://doi.org/10.1002/jcc.25780

Contributions:

Conception:	30%
Model creation:	80%
Model validation:	85%
Experimental assays:	0%
Code implementations:	100%
Web(tool) validation and deployment:	100%
Web(tool) maintenance:	100%
Manuscript writing:	40%
Manuscript revision:	20%

PPI-Detect: A Support Vector Machine Model for Sequence-Based Prediction of Protein–Protein Interactions

Sandra Romero-Molina,[a] Yasser B. Ruiz-Blanco, *[a] Mirja Harms,[b] Jan Münch,[b,c] and Elsa Sanchez-Garcia *[a]

The prediction of peptide–protein or protein–protein interactions (PPI) is a challenging task, especially if amino acid sequences are the only information available. Machine learning methods allow us to exploit the information content in PPI datasets. However, the numerical codification of these datasets often influences the performance of data mining approaches. Here, we introduce a procedure for the general-purpose numerical codification of polypeptides. This procedure transforms pairs of amino acid sequences into a machine learning-friendly vector, whose elements represent numerical descriptors of residues in proteins. We used this numerical encoding procedure for the development of a support vector machine model (PPI-Detect), which allows predicting whether two proteins will interact or not. PPI-Detect (https://ppi-detect.zmb.uni-due.de/) outperforms state of the art sequence-based predictors of PPI. We employed PPI-Detect for the analysis of derivatives of EPI-X4, an endogenous peptide inhibitor of CXCR4, a G-protein-coupled receptor. There, we identified with high accuracy those peptides which bind better than EPI-X4 to the receptor. Also using PPI-Detect, we designed a novel peptide and then experimentally established its anti-CXCR4 activity. © 2019 Wiley Periodicals, Inc.

DOI:10.1002/jcc.25780

Introduction

The effective representation of the information content in datasets is essential for the development of machine learning approaches. Based on a novel numerical codification procedure for polypeptide sequences, we created a computational tool for the prediction of protein–protein interactions (PPI), with key implications for drug design.

PPI are intrinsically related to protein function[1,2] and a crucial aspect in pharmaceutical and biomedical applications.[3] The growing interest of the scientific community on PPI has gradually transformed paradigms, from the elucidation of genomes, to proteomes and to interactomes.[4–6] With the constantly increasing amount of PPI data,[7,8] improvements on the ability of data mining methods to extract valuable insights from the corresponding databases are necessary. Several computational studies modeling PPI-related functions from amino acid sequences have been described.[9–13] These works evidence the applicability of primary structure data to predict complex proteomic properties. However, despite important advances in this area, shortcomings in the precision of state-of-the-art sequence-based PPI prediction methods[14–17] have been highlighted[18] and robust benchmarking datasets of PPI with reliable noninteracting data are needed.[19]

As mentioned above, a key step of any learning scheme is an adequate representation of the input data. Thus, our first aim is to provide a general framework for the numerical encoding of proteins. For small organic molecules, a large amount (~10^6) of numerical descriptors are defined.[20] However, peptides and proteins have not received the same attention. Noteworthy in the area of numerical descriptors for proteins is the work of Chou and collaborators, who introduced pseudo-amino acid composition descriptors for the first time.[21] Web servers such as PROFEAT[22] and Pse-in-one 2.0,[23] containing Chou's descriptors, were the only comprehensive tools for the general-purpose codification of protein sequences. Broadly expressed, the pseudo-amino acid composition concept encompass any procedure that, starting from the primary structure, delivers a numerical feature vector with information of the amino acid composition and the sequence ordering.[24] In this context, Ruiz-Blanco et al. also presented numerical descriptors for individual proteins (ProtDCal).[25] ProtDCal found applications in several studies involving posttranslational modifications, antibacterial peptides, and protein function.[25–32] Notably, these descriptors showed low correlation with those in PROFEAT, highlighting the need for further codification approaches.[25]

Here, we tackle precision-related shortcomings of state of the art sequence-based PPI prediction methods by introducing an approach that allows ProtDCal to encode protein pairs. With

[a] *S. Romero-Molina, Y. B. Ruiz-Blanco, E. Sanchez-Garcia*
Center of Medical Biotechnology, University of Duisburg-Essen, Duisburg, Germany
E-mail: yasser.ruizblanco@uni-due.de or elsa.sanchez-garcia@uni-due.de

[b] *M. Harms, J. Münch*
Institute of Molecular Virology, Ulm University Medical Center, Ulm, Germany

[c] *J. Münch*
Core Facility Functional Peptidomics, Ulm University Medical Center, Ulm, Germany

S. Romero-Molina and Y. B. Ruiz-Blanco contributed equally to this work.

Contract Grant sponsor: Boehringer Ingelheim Stiftung; Contract Grant number: Plus-3; Contract Grant sponsor: Deutsche Forschungsgemeinschaft; Contract Grant numbers: SFB1279, EXC 2033

this pairwise codification scheme, we extend the applicability of ProtDCal to the study of proteome properties associated with more than one protein (which include PPI as well as protein functional relations, identification of remote homology and co-evolution events in proteins, among others). Subsequently, based on this pairwise codification scheme, we created a novel PPI sequence-based predictor, named PPI-Detect.

We found that our obtained model outranks the estimated accuracy of state-of-the-art predictors such as PIPE,[14,18] SPPS,[33] and Pred-PPI,[16] thus highlighting the applicability of the new pairwise codification strategy for modeling PPI data. The performance of PPI-Detect was further evaluated in the identification of active mutants from a dataset of EPI-X4 derivatives which are inhibitors of the G-protein coupled receptor CXCR4, a well-known drug target.[34]

Materials and Methods

Chou established a five-step rule[24] for the development of a useful sequence-based statistical predictor for a biological system[35–41]: (1) to select a valid benchmark dataset to train and test the predictor; (2) to formulate the biological sequence samples with a mathematical expression that reflects their intrinsic correlation with the target application; (3) to introduce a powerful algorithm or engine to operate the prediction; (4) to properly perform cross-validation (CV) tests to evaluate the accuracy of the predictor; and (5) to make the predictor accessible to the public via a web server. We follow these guidelines in the present work.

Data collection and construction of the benchmarking set

First, we gathered a nonredundant benchmarking dataset of PPI from three comprehensive, curated and publicly available databases. These databases contain information about pairs of protein domains with proven interactions (3did[42] and iPfam[43]), and domain pairs with very little chances of being involved in an interaction (Negatome 2.0[44]).

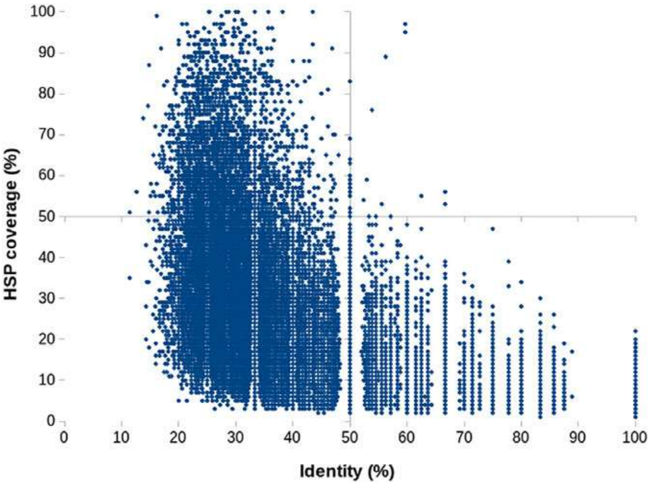


Figure 1. High scoring pairs (HSP) coverage versus alignment identity for all domains in the knowledge database. [Color figure can be viewed at wileyonlinelibrary.com]

Thus, the following amount of domains pairs was obtained: 9326 pairs (3did[42]), 9516 pairs (iPFam[43]) and 2666 pairs (Negatome 2.0[44]). We removed 261 pairs from the Negatome input because they were also found in the dataset of interacting domains. Negatome is built by regular application of filters involving the crossing of its entries with known PPI.[44] Then, the occurrence of some redundant pairs with 3did and iPFam is probably related to an outdated cleaning of the database.

Domains exclusively found in the interacting (positive) or noninteracting (negative) datasets can bias the performance of a predictor. Therefore, we kept only those domain pairs whose individual members were present in both positive and negative data in the final dataset. This analysis resulted in 1922 interacting pairs and 2405 noninteracting pairs of domains. The final dataset (comprising 4327 pairs) thus exhibits high reliability for the evaluation of the possibility of an interaction between pairs (see Supporting Information file-1).

Protein domains are often expressed in several proteins, the Pfam[45] database contains the multiple sequences associated with a unique domain. Thus, upon collecting the interdomain interaction data, the amino acid sequences of the individual domains were obtained from the Pfam database. Then, the CD-HIT program[46] was used to eliminate redundant domain sequences using a cut-off of 40% of identity. Next, the most representative sequence was extracted for each domain based on the levels of identity with respect to the remaining protein sequences.

BLASTp[47] was used to run pairwise local alignments among all the domains using a permissive e-value = 100, in order to increase the number of hits between a given domain and the rest of the data. The similarity between all the high scoring pairs of domains (interacting and noninteracting) was represented by two parameters, the coverage index (fraction of the whole sequence covered for the local alignment), and the alignment identity (fraction of identical residues in the aligned fragment (Fig. 1). The first quadrant includes domain pairs with high identity and coverage, the third quadrant includes domain pairs with very little identity and coverage. Most of the domain pairs (with some distinguishable similarity [e-value <100]), show levels of alignment identity and coverage below 50% (Fig. 1). The lack of alignment of domain pairs with both, identity and coverage measures over 60%, evidences the low sequence similarity among the individual domains in the data. Thus, even if two pairs share a domain, that is A-B and A-C, the non-common partners diverge enough for an effective differentiation of the two pairs.

Subsequently, we split the dataset into training and test sets. Importantly, all the test instances were excluded from the training dataset. Here, we refer to the interacting domains as the positive cases and the noninteracting domains as the negative cases. We used 836 pairs of domains (309 positive and 527 negative) for testing and 3491 pairs (1613 positive and 1878 negative) for training. To estimate the performance of the final model, we grouped the test data by degrees of difficulty:

Very hard subset. It gathers pairs of individual domains not present in the training data. If the pair A-B is included in this subset, neither A nor B are found in any pair of domains of the training data. This subset contains 103 domain pairs (57 positive and 46 negative).

Mid-hard subset. It comprises domain pairs where only one of the domains is present in the training data. If the pair A-B is part of this subset either A or B, but not both, is found in the training data. This subset contains 307 domain pairs (102 positive and 205 negative).

Easy subset. This subset comprises pairs where both domains are present in the training data, and found only once in this test subset. If a pair A-B is included in this subset, then both A and B are part of other domain pairs in the training set. In addition, no other pair containing A or B is found in the test subset. This subset includes 426 domain pairs (150 positive and 276 negative).

ProtDCal: A tool for the general-purpose encoding of proteins

A challenging issue lies, given the increasing amount of available protein data, on expressing an amino acid sequence with a discrete model while preserving considerable sequence-order information. This is because the majority of machine-learning algorithms that have found applications in bioinformatics (such as k-nearest neighbor,[48] support vector machine,[49] and random forest[50]) mostly handle numerical features vectors.[51] A pioneering way to generate effective numerical vectors from primary structures, without losing the sequence-pattern information, was the pseudo amino acid composition[52] or PseAAC.[21,53] According to Chou's remarks on pseudo-amino acid composition, this concept was extended to include all discrete set of features capable of describing the amino acid content of proteins while also capturing the sequence ordering information.[24] Within this context, we previously introduced a novel protein features generation algorithm in the software ProtDCal.[25] The complementary nature of both protein codification approaches was established given the significant absence of correlation between the numerical features produced by PseAAC and those generated with ProtDCal.[25]

The ProtDCal program is a flexible and potent tool for general-purpose numerical encoding of individual protein sequences and structures.[25,27] Here, we extended the protocol of ProtDCal to also encode pairs of amino acid sequences. The algorithm of the program follows a pipeline of five steps (Fig. 2):

Steps 1 to 4 (previously implemented in ProtDCal) permit to encode individual proteins. Next, we briefly describe these four steps of the program's workflow, and in the section "Definition of novel-pairwise descriptors for amino acid sequences" we present the implementation introduced in this work, which corresponds to step 5 in Figure 2.

Step 1: Numeric codification of residues. ProtDCal uses a set of structural and chemical-physical properties of amino acids (molar weight, hydrophobicity, isoelectric point, among others), taken from the AAindex database.[54] Using these properties, a *residue features matrix* is created. The rows of this matrix correspond to the residues in the protein while the columns of the matrix correspond to the properties describing the amino acids. The residue features matrix is then fed into the second step of the pipeline (Fig. 2).

Step 2: Modification by vicinity. In this step, a *vicinity-modified residue features matrix* is obtained by applying a vicinity-modification operator to update the elements of the matrix using information from the ordering of the residues in the amino acid sequence. For encoding individual proteins, four local vicinity operators are implemented in ProtDCal. One of them is the Moreau-Broto autocorrelation (AC_l)[55,56] function:

$$D_{\mathrm{AC}i} = D_i(D_{i+l} + D_{i-l}) \tag{1}$$

where D_i represents the value of an amino acid property D for residue i, $D_{\mathrm{AC}i}$ corresponds to the vicinity-modified property after applying the AC operator with the neighbor residues ($i + l$ and $i - l$) separated by a topological distance l from the residue i. This way, the use of the AC operator allows us to incorporate in the description of each residue a contribution from neighbors at a user-defined topological distance (l).

Global vicinity operators, like the electro-topological state (E-State) operator,[57] are also implemented. For each residue, the E-State operator takes into account all other residues during the calculation:

$$D_{\mathrm{ES}i} = D_i - \sum_{j \neq i}^{N} \frac{D_j - D_i}{(j-i)^2} \tag{2}$$

where D_i represents the value of an amino acid index D for residue i, and $D_{\mathrm{ES}i}$ corresponds to the vicinity-modified index after applying the E-State operator. Currently, for the extension of ProtDCal to pairwise descriptors (see next section), the E-State is the only used vicinity operator.

Step 3: Grouping. ProtDCal follows a *divide-and-conquer* approach based on splitting the sequence of a protein into multiple groups of related residues, each of these groups is then used to generate separate numeric descriptors. At this stage, the *vicinity-modified residue features matrix* is split into many group-based matrices composed by rows associated to selected residues in the complete matrix. Grouping operators are defined as groups of residues sharing similar properties, for example, all aromatic, polar, nonpolar, acid, and basic residues. These group-based matrices can be as large as the entire matrix (i.e., a group formed by the whole protein), or as small as a single row (i.e., a group formed by just one residue).

Step 4: Invariant aggregation. This step comprises descriptive statistics (variance, standard deviation, skewness, among others) and Shannon entropy-related measures[58] that are used to transform the distribution of values for each of the properties (steps 1 and 2) within a group of residues, into a single numerical value. The operators applied in this step are referred to as invariant, given that the final descriptor values are independent of the ordering of the residues within the group. These operators

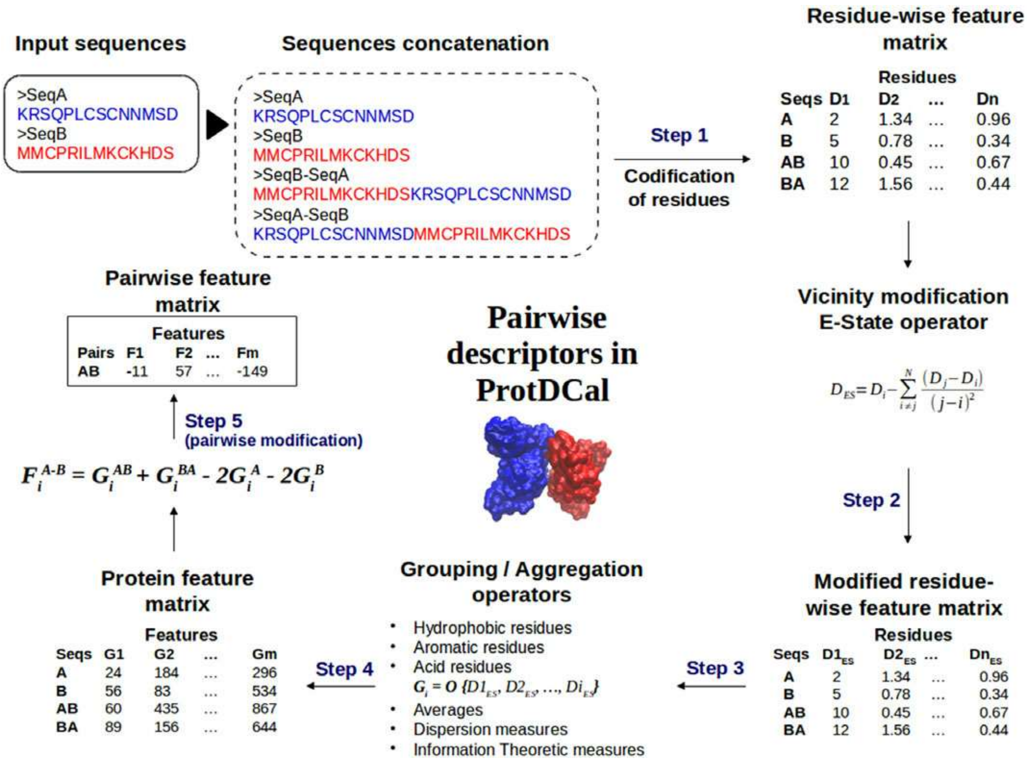


Figure 2. General workflow of the ProtDCal program from left to right, clockwise: is *D* represents an amino acid property (e.g. hydrophobicity, molar weight, etc.), *G* is the intermediate group-based descriptor and *F* represents the final pairwise descriptors. [Color figure can be viewed at wileyonlinelibrary.com]

act over the columns of the group-based matrices, transforming each column into a final numerical feature that accounts for the distribution of values of a given property within the group (note that each column of the matrices corresponds to a single amino acid property). Finally, the features obtained from all the intermediate sub-matrices are arranged into a single vector that characterizes the entire input protein. Therefore, once all proteins are processed, this step returns a *protein features matrix* with dimension $l \times m$, where l is the number of proteins in the dataset and m is the total number of features obtained from the aggregations in all the sub-matrices. This number corresponds to $m = P \times G \times A$, where P, G, and A are the number of selected amino acid properties, grouping criteria, and aggregation operators, respectively.

ProtDCal delivers a large features vector that encodes the input protein, given its combinatorial algorithm and the available number of choices for amino acid properties, vicinity, grouping and aggregation operators. All elements of this vector are univocally defined by a specific combination of the mentioned choices. For instance, an individual feature (***Mw_ACk_PCR_V***) could be generated by the *variance* (***V***), among the *positively charged residues* (*PCR*), of the modified *molar weights* (*M_w of the PCR residues). M_w* values are in turn obtained using the *autocorrelation operator* of order *k* (*ACk*) acting on the original *molar weights* of all residues in the protein.

Modeling protocol

Our protocol for obtaining pairwise protein descriptors within ProtDCal is presented in the section "Definition of novel-pairwise descriptors for amino acid sequences." This new codification allows generating up to 13248 features for each pair of proteins in the dataset (Supporting Information Section SM1). We use this set of features to initiate our modeling procedure. To reduce the dimensionality of the features vector, we apply the following attribute-selection steps:

1. The information gain[59] scoring method implemented in Weka 3.7.11[60] is used to rank all features according to the information content relevant to describe the class distribution. We set a threshold value equivalent to 5% of information content (Shannon entropy) of the class distribution. This allowed us to eliminate those features without information relevant to distinguish interacting from noninteracting domains in the training set. In this manner, the initial set was reduced to 326 features.
2. We further reduced the redundancy among the extracted subset by using the DCluster tool of ProtDCal, especially designed for this purpose. DCluster is a Perl script that, by implementing a single-linkage clustering method, extracts the most representative element of each cluster of features to build a reduced and nonredundant dataset. The algorithm uses the Spearman correlation coefficient as similarity measure between two features. We defined a threshold of 0.95 to add new members to a growing cluster. After redundancy screening, only four attributes were removed leading to 322 features in the nonredundant dataset. This fact evidences the low redundancy among the numeric features introduced in the present study.
3. The third step of the selection process comprises a supervised selection of the best subset of attributes to be used in the final model. Here, the WrapperSubsetEval method, implemented in Weka 3.7.11, was coupled to a genetic search algorithm[61] to

select and evaluate multiple subsets of attributes. We used a population of 20 individuals with mutation and crossover probabilities of 0.033 and 0.6, respectively, and extended the search for 100 generations. The evaluation of each subset along the search was performed with fivefold cross validations, by scoring each subset with the value of the accuracy of the obtained classifier. The technique used for building the models was a support vector machine (SVM) with linear kernel. This selection step led to a final subset of 19 features (Supporting Information Table SM2) which are the final variables included in our model.

Learning technique

Previous approaches for the prediction of PPI take advantage of the strength of machine learning algorithms such as SVM.[16,33,62,63] Given the robustness of SVM, and to test the value of our novel descriptors within the same learning scheme of previous methods, we also used SVM as the learning algorithm to train our predictor.

Once a suitable subset of attributes was extracted, the models were trained using the SVM package SMO.[64] The descriptors are normalized before the optimization of the model and logistic regression is used to calibrate the outcome of the SMO algorithm for a better estimation of the outcome probabilities. The optimal setup for SMO and its kernel function was estimated by means of a grid search along the parameters space. We explored the RBF and polynomial kernels following the grid-search approach described by Hsu et al.[65] (Supporting Information Section SM3 and Table SM4). The final model was selected with a linear kernel and a cost (*C*) for misclassified cases *C* = 11.3.

Performance measures

Here, we use classical performance measures: precision (*Pr*) is defined as the fraction of *correctly* predicted interactions (TP) out of the *total* number of predicted interactions (TP + FP) where FP is the number of *false-positive* predictions.

$$\mathrm{Pr} = \mathrm{TP}/(\mathrm{TP}+\mathrm{FP}) \tag{3}$$

Our goal is to obtain a predictor that effectively balances precision and sensitivity (Sn). Sn is defined as the fraction of *correctly* predicted interactions out of the number of *known* interactions (TP + FN) where FN is the number *of false-negative* predictions.

$$\mathrm{Sn} = \mathrm{TP}/(\mathrm{TP}+\mathrm{FN}) \tag{4}$$

Prevalence-corrected precision[66] (PCPr) is used to unify the precision values of the different test sets at a unique level of prevalence, for a proper comparison between test sets. This way, we avoid bias in the precision values because of the different ratios of positive and negative data (i.e., class imbalance) in each test set.

$$\mathrm{PCPr} = \frac{\mathrm{Sn}}{\mathrm{Sn}+r(1-\mathrm{Sp})} \tag{5}$$

The prevalence value *r* represents the new ratio of negative to positive protein pairs used to re-estimate the precision. Sn and Sp are the sensitivity and specificity of the predictor, respectively.

$$\mathrm{Sp} = \mathrm{TN}/(\mathrm{TN}+\mathrm{FP}) \tag{6}$$

Accuracy (Acc) was used to evaluate the overall performance of our model:

$$\mathrm{Acc} = (\mathrm{TP}+\mathrm{TN})/(\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}) \tag{7}$$

The Mathew correlation coefficient (MCC) is also used to assess the overall quality of the model distinguishing between the two classes (interacting and noninteracting pairs).

$$\mathrm{MCC} = \frac{\mathrm{TP}\times\mathrm{TN}-\mathrm{FP}\times\mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})}} \tag{8}$$

We note that, although the classical confusion matrix elements are traditionally used for formulating performance measures of classifiers,[67] Chou's terminology[68] presents an intuitive alternative to the classical terms. Here, for the sake of consistency with previous work reported by Ruiz-Blanco et al., we employ the classical nomenclature.

Experimental details

Peptides were synthesized on a 0.10 mM scale using standard Fmoc solid-phase peptide synthesis techniques with the microwave synthesizer (Liberty blue; CEM). Afterwards peptides were purified using reverse phase preparative high-performance liquid chromatography. After lyophilization, the peptide mass was verified by liquid chromatography mass spectroscopy. In order to verify peptide binding to the CXCR4 receptor, a CXCR4-tropic HIV-1 inhibition assay was performed as previously described.[34] Briefly, TZM-bl cells (a HIV-1 reporter cell line harboring a β-galactosidase construct under the control of the HIV-1 promoter) were seeded in flat 96-well cell culture plates. The next day, growth medium was replaced to DMEM containing 2.5% fetal calf serum and peptides were added at increasing concentrations. After 10 min incubation at 37°C, CXCR4-tropic NL4-3 HIV-1 was added. Three days postinoculation, infection rates were determined by quantifying cell-associated β-galactosidase activity in a luminescence-based assay. Half-maximal inhibitory concentrations were calculated by GraphPad Prism.

Results and Discussion

Definition of novel-pairwise descriptors for amino acid sequences

Originally, ProtDCal was only able to encode single amino acid sequences or 3D structures. However, for modeling pairwise protein data, such as PPI, we need to define descriptors for *pairs* of amino acid sequences. Here, we introduce an approach to generate new pairwise descriptors intended to encode information associated to forming pairs of sequences. We define the pairwise descriptor as a function of the single-chain descriptors of products and reagents of a block co-dimerization reaction. Given two amino acid sequences A and B and the reaction:

$$2A + 2B = > AB + BA$$

AB and BA are block copolymers formed by the concatenation of A and B.

The pairwise descriptor $D_{(A\text{-}B)}$ is calculated as:

$$D_{(A\text{-}B)} = D_{(AB)} + D_{(BA)} - 2D_{(A)} - 2D_{(B)} \tag{9}$$

where $D_{(X)}$ corresponds to the value of the single-chain descriptor for a given sequence X (A, B, AB, or BA in this example). The net value of $D_{(A\text{-}B)}$ is related to the augmented topology upon the concatenation process, and thus it corresponds to a numerical representation of the *relation* between the independent sequences and no to a simple sum of their individual features. The electro-topological state[57] (E-State) operator is then used to leverage the topological information of the original and of the combined sequences during the vicinity-modification step. We added this step (step 5, Fig. 2) to the pipeline of ProtDCal. As implemented here, the calculation of pairwise descriptors with ProtDCal only requires an additional input file containing the definition of the protein pairs. The program performs the sequence concatenation and automatically computes the pairwise features.

Performance of PPI-Detect during training and prediction tests

Here, we followed the modeling protocol described in the Methods section. To start with, we used all descriptors (obtained by the pairwise transformation discussed above) from the modified ProtDCal's pipeline.

For analyzing our results, we used precision-recall curves (PRCs) (see Supporting Information Table SM-4 for additional data on other performance measures). The PRCs of the training data show high precision values (~90%) with a sensitivity in the range of 30%, even in the region of high decision threshold (beginning of the curve in Fig. 3). At the mid points of the performance measures, 50% of precision is obtained with ~90% of sensitivity and 50% of sensitivity is reached with ~78% of precision.

To assess the robustness of this fitting, we analyzed the PRC curves of the 10-fold CV and of the training fitting. Large deviations from the training performance due to the perturbations of the CV would indicate that the fitted predictor is too dependent of training instances and thus it is overfitted. Remarkably, the PRC curve of the CV closely resembles the one obtained during the training. We can thus confirm that our model is robust and its performance is the result of the generalization of relevant information in the training data.

Next, we tested the generalization of the model using three test subsets (see section "Data collection and construction of the benchmarking set") with scaled levels of similarity with the training data. In this way, potential users can estimate the performance of our method by crossed similarity analysis between their proteins of interest and those in our training data. The PRCs for the entire test set and subsets (Fig. 4A) show the levels of precision and sensitivity in each case. Figure 4B depicts analogous PRCs including a prevalence (imbalance) correction to the precision. Prevalence was fixed to r = 1, which transforms the precision values to those expected if the number of negative and positive data is the same. This procedure allows standardizing the prevalence among the three datasets, and thus an unbiased comparison between the different tests is possible.

For the *easy* test, a precision of 90% is reached with a sensitivity of ~70%, while the *mid-hard* test shows a precision of 90% for a sensitivity of ~45%. The decay in sensitivity between these subsets is a consequence of the significantly lower similarity of the *mid-hard* test set with the training data (half of the individual domains are not present in the training data). Remarkably, the *very-hard* test yields a precision of 90% with also a sensitivity of 45%, which further validates the generalization of the model.

Provided the low redundancy among domain sequences in the entire knowledge base (Fig. 1), our results evidence, not only the generalization, but also the prediction power of the developed model. This predictor is implemented in a user-friendly tool named PPI-Detect (Supporting Information Section SM5). PPI-Detect is freely accessible via web interface at: https://ppi-detect.zmb.uni-due.de/. Additionally, the compiled version is available upon request.

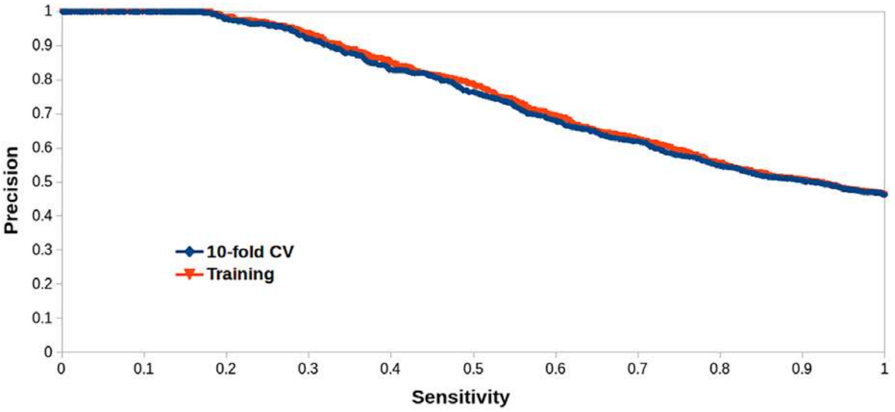


Figure 3. Precision-recall curves of 10-fold cross-validation and training procedures. CV, cross validation. [Color figure can be viewed at wileyonlinelibrary.com]

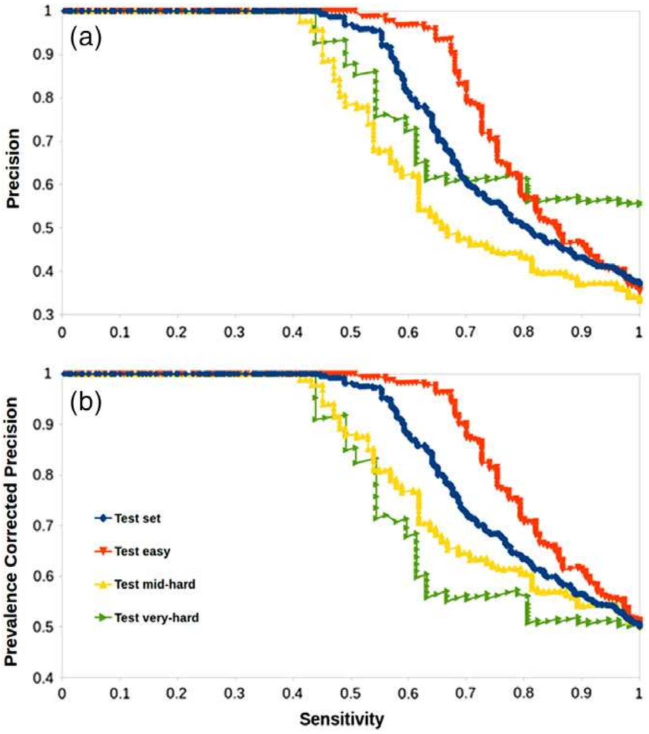


Figure 4. Precision-recall curves (PRC) obtained for the complete test set and the three subsets with varying levels of difficulty. Panel A (top) shows raw values of precision and sensitivity along the threshold range. Panel B (bottom) shows the plots using prevalence-corrected precision with a prevalence value $r = 1$. [Color figure can be viewed at wileyonlinelibrary.com]

To define the applicability domain (AD) of the model we use the range of the descriptors' values in our training dataset (Supporting Information Table SM6). This approach is based on the intuitive definition of AD as the subspace delimited by the range of the variables (descriptors) in the model.[69] As part of the PPI-Detect workflow, we implemented a tool named ADcheck, which allows the users to map the descriptor values of their new data into the AD. Specifically, the ranges implemented in ADcheck are defined between the 5th and 95th percentiles of each descriptor. Any new case whose descriptors' values are outside any of the ranges is considered as a potential outlier. The input for ADcheck is the data with all the descriptors' values of the predicted protein pairs, which is one of the outputs files of PPI-Detect. With this information, ADcheck identifies the cases that are outside the range of descriptors' values in the training data.

Additionally, by inspecting the amino acid composition and length of the new sequences, users can assess qualitatively whether new data are within the AD. Only the 20 standard amino acids are present in our training data, also no chemical modification or d-type residues were used. Regarding the length of the individual sequences, it varies from 16 amino acids (aa) to 544 aa. The average sequence length in the training data is 109 aa (standard deviation of 69 aa). For the pairs of sequences, a minimum sum of 40 aa and a maximum of 1088 aa is desirable. The average sum of the sequence sizes of the pairs in the training data is 198 aa (standard deviation of 87 aa).

Comparison of the performance of PPI-Detect with respect to other predictors

A combined subset, containing the instances found in the *mid-hard* and *very-hard* subsets, was used to compare the performance of our method to that of other sequence-based predictors of PPI: PIPE,[14,18] Pred-PPI,[16] and SPPS.[33] PIPE is a sequence alignment-based predictor, which performs massive assessments of interaction probability between pairs of fragments, considering all known interactions of several species, for example, human, yeast, *Escherichia coli*. Pred-PPI and SPPS are SVM predictors and their different behavior with respect to ours can be partly associated to the attributes used to represent the protein–protein pairs.

Remarkably, our model largely surpasses the performance of the other state of the art predictors. For instance, at 50% of sensitivity, the other methods yield about 50% of precision, while our model reaches 80% (Fig. 5).

This way, the precision–recall curves, which show the performance along the entire range of the outcome score, evidence the superior performance of PPI-Detect for all the precision (or sensitivity) values (Fig. 5). This is also indicated by the global metrics (such as accuracy and MCC) that display for PPI-Detect concomitant precision and sensitivity values above 50% (Table 1).

An important factor that contributes to the better performance of our method is the number and curated nature of the training data. We use the three largest repositories of curated protein domain–domain interactions. This wealth of learning information is relevant to achieve more generalization in a predictor. PIPE predictions are also derived from large databases such as DIP and MIPS.[7,70] However, PIPE does not use data of noninteracting sequences in its prediction pipeline. By including such data as negative control during the development of our

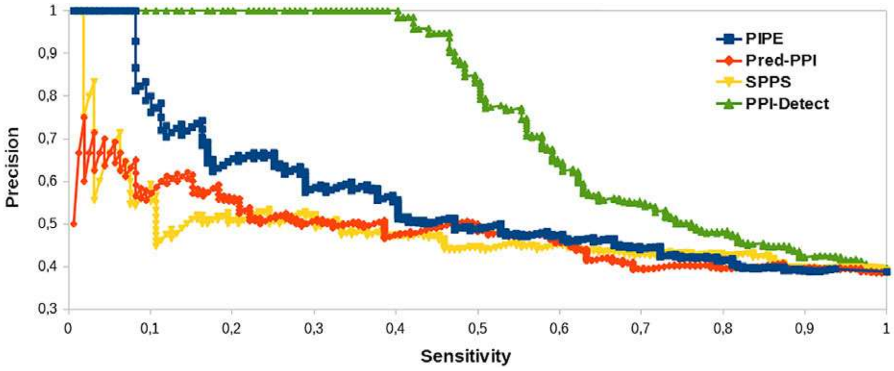


Figure 5. Precision-recall curves (PRC) of the different predictors of PPI and of PPI-Detect (*mid-hard* + *very-hard* subsets). [Color figure can be viewed at wileyonlinelibrary.com]

Table 1. Comparison of performance measures for PPI-Detect and other methods in the hard test set (*mid-hard* + *very-hard* subsets)

	Precision	Sensitivity	Accuracy	MCC
PIPE	0.762	0.101	0.639	0.178
Pred-PPI	0.396	0.880	0.435	0.049
SPPS	0.514	0.239	0.617	0.121
PPI-Detect[a]	0.554	0.648	0.661	0.310

[a] The threshold is fixed at an outcome probability of 0.5.

supervised classifier, we improve the final precision avoiding over-producing positive predictions.

Application of PPI-Detect to EPI-X4 derivatives

The CXC chemokine receptor 4 (CXCR4) is a G-protein-coupled receptor that is expressed in multiple cells.[34,71] Activation of CXCR4 by its chemokine ligand, stromal-cell-derived factor-1 (SDF-1 or CXCL12),[72,73] governs important physiological processes.[74,75] Deregulation of CXCR4-CXCL12 signaling in humans is involved in multiple disorders, such as cancers, inflammation, and cardiovascular diseases. Zirafi et al. reported the discovery of a novel endogenous antagonistic ligand of CXCR4 named EPI-X4, which is an evolutionary conserved fragment of human serum albumin.[34] Binding of EPI-X4 to CXCR4 is highly specific and suppresses both, basal and CXCL12-induced signaling. Furthermore, this endogenous CXCR4 antagonist blocked CXCL12-mediated receptor internalization and suppressed the migration and invasion of cancer cells toward a CXCL12 gradient, suggesting that EPI-X4 may have anti-metastatic activity.[34]

The amino acid sequence of EPI-X4 is LVRYTKKVPQVSTPTL. Zirafi et al.[34] evaluated a number of EPI-X4 derivatives in order to enhance its activity. From such derivatives, we extracted a total of 35 (monomeric and non-chemically modified) peptides to assess the predictability of our method in terms of peptide activity. First, we classified the set of 35 peptides (Supporting Information Table SM7) in two groups (21 low-active and 14 highly active peptides) using the IC_{50} value of EPI-X4 in HIV-1 inhibition assays as the reference for this classification. The calculated precision (success rate) of the experimental study was 40% (14 out of 35 variants were experimentally found to be more active than EPI-X4).

Next, we divided the exposed region of the membrane protein CXCR4 into four fragments (Fig. 6).

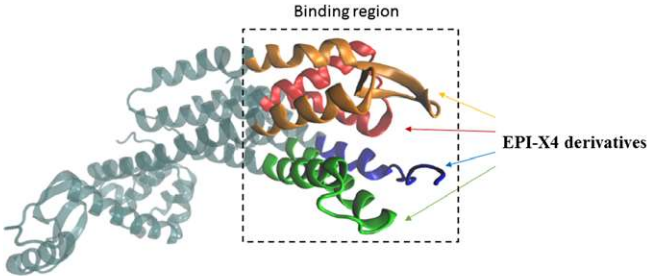
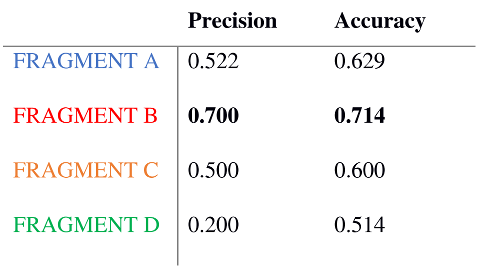


Figure 6. CXCR4 three-dimensional structure (PDB entry 3OE0[76]). Fragment A (residues 25–45, blue), fragment B (residues 87–121, red), fragment C (residues 164–205, orange), and fragment D (residues 252–292, green) are highlighted. [Color figure can be viewed at wileyonlinelibrary.com]

Table 2. Summary of the success rate (precision and accuracy) in identifying active EPI-X4 derivatives based on the predicted interaction with CXCR4 fragments. The color code is the same as in Figure 6



	Precision	Accuracy
FRAGMENT A	0.522	0.629
FRAGMENT B	**0.700**	**0.714**
FRAGMENT C	0.500	0.600
FRAGMENT D	0.200	0.514

We then predicted the interactions of each CXCR4 region with the 35 peptides using PPI-Detect. The outcome for each pair was classified as *low*-likely or *high*-likely interacting, if their predicted scores were lower or higher than the scores for the interactions between EPI-X4 and the four fragments (Supporting Information Table SM7). We note that the outcome scores of the model are not directly related to interaction strength, but to likelihood of interaction. Thus, we are assuming that more likely interactions imply more active candidates, which is not necessarily the case. Table 2 shows the fragment-based performance of PPI-Detect for the prediction of the activity class.

Interestingly, three of the fragment-based estimates lead to precision values equal or higher than 50% when the calculated precision (success rate) for the study was only 40%. In particular, the analysis of fragment B shows high precision (70%). This suggests that the EPI-X4 derivatives may establish specific interactions with CXCR4 that strongly influence the binding affinity of the peptides. On the contrary, the interaction with fragment D does not show any apparent relation with the derivatives' activity. Notably, fragments A, B, and C encompass the minor pocket of this receptor, while C and D comprise the major pocket. Thus, EPI-X4 may bind more favorably to the minor pocket of CXCR4.

Although several factors regulate the biological activity of a given compound, PPI between receptor and ligand provide important hints in the search of improved ligands. In this context, we used PPI-Detect to design a shorter derivative of EPI-X4, based on the previously reported sequence of WSC02 (IVRWSKKVPCVS), a very active EPI-X4 derivative.[34] Ten thousand initial sequences were generated by applying conservative mutations on randomly selected residues of WSC02, residue deletions were implemented

Table 3. Sequences of the experimentally reported and designed peptides

	Sequence
EPI-X4	LVRYTKKVPQVSTPTL
WSC02	IVRWSKKVPCVS
JM130	IVRWSPPCVS
JM133	IVRWSKYVS
JM135	IVRSSRKVVS

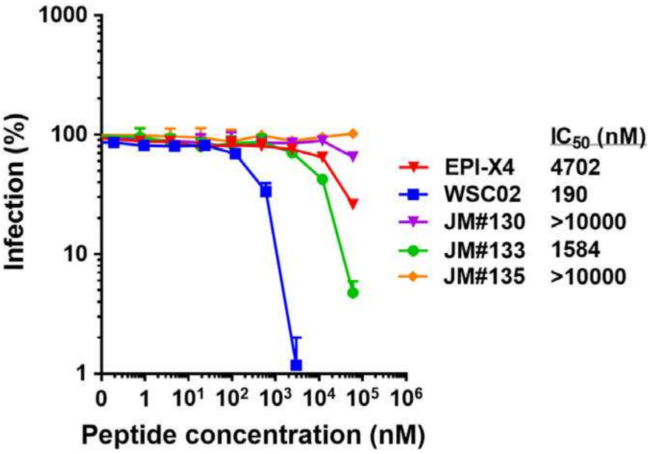


Figure 7. Inhibition of CXCR4-tropic HIV-1 infection by EPI-X4 and derivatives thereof. TZM-bl reporter cells containing the indicated concentrations of peptides were inoculated with CXCR4-tropic HIV-1. Infection rates were determined 3 days later by the β-galactosidase assay. Shown are data from three individual experiments performed in triplicates ± SD. [Color figure can be viewed at wileyonlinelibrary.com]

with a probability of 0.05. For each candidate the modification process was stopped when the molar weight was below 1200 Da.

Then, the peptide library was virtually screened using PPI-Detect to predict the interaction of each of the peptides with the fragments A, B, and C of CXCR4 (Fig. 6). From this screening, three candidates (**JM130**, **JM133**, and **JM135**, Table 3) were identified with the highest possible prediction score (score = 3, obtained as the sum of the interaction scores with the CXCR4 fragments A, B, and C).

The anti-CXCR4 activity of the peptides was evaluated with the HIV-1 inhibition assay. We found that, while **JM130** and **JM135** had no effect on HIV-1 infection, **JM133** blocked viral infection in a dose-dependent manner with a mean half-maximal inhibitory concentration (IC_{50}) of ~1.6 μM, which is approximately three times more active than the endogenous peptide EPI-X4 (4.7 μM), albeit it was less active than WSC02 (Fig. 7). This behavior can be rationalized by analyzing the nature of the modifications on these peptides. Zirafi et al. reported that the introduction of Ile in position 1, Trp in position 4, and Ser in position 5 results in an increased activity of the peptides.[34] This explains why **JM133** is more active than EPI-X4. **JM135** lacks the aromatic residue in position 4, which could lead to its absence of activity. In **JM130**, the lysine residues in position 6 and 7 are removed. Since the CXCR4 binding pocket is rich in negatively charged residues,[76] the positively charged amino acids of the ligand should play a key role in binding and their double deletion abolishes the activity of **JM130**.[34] The elimination of Lys7 in **JM133** could also provide a rationale to the fact that this peptide cannot reach the activity levels of WSC02.

On the contrary, we note that the short lengths (required for experimental reasons) of **JM130**, **JM133**, and **JM135** are below the shortest sequences in the training dataset. This places these derivatives outside the AD and limits the reliability of the predictions. Further work will focus on extending the lower boundary of the current AD for a more general training of the PPI-Detect model. Atomistic simulations are also needed to provide a rationale to the relative accuracy estimates of CXCR4 fragments and the activity of the peptides. Even so, this study illustrates the potential applicability of PPI-Detect for the design of bioactive peptides. PPI-Detect is a fast computational tool that allows, without the need of structural data from the peptide or the target, the primary screening of peptide and protein libraries.

Conclusions

Here, we present a numeric codification approach for pairs of amino acid sequences. This approach provides a platform for many and diverse bioinformatics applications, such as the study of PPI networks and functional relations. Other potential field of application is the identification of remote evolutionary relations among proteins using alignment-free algorithms. There, a suitable classification problem is to distinguish pairs of orthologues from paralogues proteins between proteomes.

We also applied our new pairwise codification methodology to the development and validation of PPI-Detect, a novel and accurate PPI predictor. We employed PPI-Detect to study the interactions between derivatives of EPI-X4 and its receptor CXCR4, highlighting the potentialities of the predictor for tailored peptide design. In this context, **JM133**, a shorter and more active derivative of EPI-X4 was identified solely based on PPI-Detect predictions. PPI-Detect relies, uniquely, on the primary structure of proteins and it is thus applicable to the massive screening of putative PPI on the entire proteome scale. In this manner, our protein encoding approach and PPI-Detect are useful tools for the discovery and extension of the interactomes of multiple species.

ACKNOWLEDGMENTS

This work was supported by the German Research Foundation, DFG – Project number 316249678 – SFB 1279 (E.S.-G., S.R.-M. and J.M.). E.S.-G., was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germanyś Excellence Strategy - EXC-2033 - Project number 390677874. E.S-G. acknowledges the support of the Boehringer Ingelheim Foundation (Plus-3 grant).

Keywords: PPI-Detect · ProtDCal · G-protein-coupled receptor · CXCR4 · protein–protein interactions · protein descriptor · EPI-X4

How to cite this article: S. Romero-Molina, Y. B. Ruiz-Blanco, M. Harms, J. Münch, E. Sanchez-Garcia. *J. Comput. Chem.* **2019**, *40*, 1233–1242. DOI: 10.1002/jcc.25780

Additional Supporting Information may be found in the online version of this article.

[1] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, G. Sherlock, *Nat. Genet.* **2000**, *25*, 25.
[2] A. W. Rives, T. Galitski, *Proc. Natl. Acad. Sci. U. S. A.* **2003**, *100*, 1128.
[3] X. Shen, L. Yi, X. Jiang, Y. Zhao, X. Hu, T. He, J. Yang, *Methods* **2016**, *110*, 90.
[4] F. Crick, *Nature* **1970**, *227*, 561.

[5] M. A. Ghadie, J. Coulombe-Huntington, Y. Xia, *Curr. Opin. Struct. Biol.* **2017**, *50*, 42.

[6] D. Szklarczyk, A. Franceschini, S. Wyder, K. Forslund, D. Heller, J. Huerta-Cepas, M. Simonovic, A. Roth, A. Santos, K. P. Tsafou, M. Kuhn, P. Bork, L. J. Jensen, C. von Mering, *Nucleic Acids Res.* **2015**, *43*, D447.

[7] L. Salwinski, C. S. Miller, A. J. Smith, F. K. Pettit, J. U. Bowie, D. Eisenberg, *Nucleic Acids Res.* **2004**, *32*, D449.

[8] D. Szklarczyk, J. H. Morris, H. Cook, M. Kuhn, S. Wyder, M. Simonovic, A. Santos, N. T. Doncheva, A. Roth, P. Bork, L. J. Jensen, C. von Mering, *Nucleic Acids Res.* **2017**, *45*, D362.

[9] J. Jia, Z. Liu, X. Xiao, B. Liu, K.-C. Chou, *J. Biomol. Struct. Dyn.* **2016**, *34*, 1946.

[10] J. Jia, Z. Liu, X. Xiao, B. Liu, K. C. Chou, *J. Theor. Biol.* **2015**, *377*, 47.

[11] K. C. Chou, Y. D. Cai, *J. Proteome Res.* **2006**, *5*, 316.

[12] J. Jia, Z. Liu, X. Xiao, B. Liu, K. C. Chou, *Molecules* **2016**, *21*, E95.

[13] X. W. Zhao, Z. Q. Ma, M. H. Yin, *Protein Pept. Lett.* **2012**, *19*, 492.

[14] S. Pitre, F. Dehne, A. Chan, J. Cheetham, A. Duong, A. Emili, M. Gebbia, J. Greenblatt, M. Jessulat, N. Krogan, X. Luo, A. Golshani, *BMC Bioinf.* **2006**, *7*, 365.

[15] S. Martin, D. Roe, J. L. Faulon, *Bioinformatics* **2005**, *21*, 218.

[16] Y. Guo, L. Yu, Z. Wen, M. Li, *Nucleic Acids Res.* **2008**, *36*, 3025.

[17] J. Shen, J. Zhang, X. Luo, W. Zhu, K. Yu, K. Chen, Y. Li, H. Jiang, *Proc. Natl. Acad. Sci. U. S. A.* **2007**, *104*, 4337.

[18] Y. Park, *BMC Bioinf.* **2009**, *10*, 419.

[19] Y. Park, E. M. Marcotte, *Bioinformatics* **2011**, *27*, 3024.

[20] R. Todeschini, V. Consonni, Frontmatter. In Handbook of Molecular Descriptors, Wiley-VCH Verlag GmbH, **2008**, p. i.

[21] H. B. Shen, K. C. Chou, *Anal. Biochem.* **2008**, *373*, 386.

[22] H. B. Rao, F. Zhu, G. B. Yang, Z. R. Li, Y. Z. Chen, *Nucleic Acids Res.* **2011**, *39*, W385.

[23] B. Liu, H. Wu, K.-C. Chou, *Nat. Sci.* **2017**, *09*, 23.

[24] K. C. Chou, *J. Theor. Biol.* **2011**, *273*, 236.

[25] Y. B. Ruiz-Blanco, W. Paz, J. Green, Y. Marrero-Ponce, *BMC Bioinf.* **2015**, *16*, 162.

[26] Y. B. Ruiz-Blanco, Y. Marrero-Ponce, E. Garcia-Hernandez, J. Green, *Amino Acids* **2017**, *49*, 317.

[27] Y. B. Ruiz-Blanco, G. Agüero-Chapin, E. García-Hernández, O. Álvarez, A. Antunes, J. Green, *BMC Bioinf.* **2017**, *18*, 349.

[28] A. Speck-Planche, V. V. Kleandrova, J. M. Ruso, M. N. Cordeiro, *J. Chem. Inf. Model.* **2016**, *56*, 588.

[29] A. Speck-Planche, M. N. D. S. Cordeiro, Speeding Up the Virtual Design and Screening of Therapeutic Peptides. In Multi-Scale Approaches in Drug Discovery, Elsevier, **2017**, p. 127.

[30] R. Corral-Corral, J. A. Beltran, C. A. Brizuela, G. Del Rio, *Molecules* **2017**, *22*.

[31] Y. Yang, S. Urolagin, A. Niroula, X. Ding, B. Shen, M. Vihinen, *Int. J. Mol. Sci.* **2018**, *19*, 1009.

[32] V. V. Kleandrova, J. M. Ruso, A. Speck-Planche, M. N. Dias Soeiro Cordeiro, *ACS Comb. Sci.* **2016**, *18*, 490.

[33] X. Liu, B. Liu, Z. Huang, T. Shi, Y. Chen, J. Zhang, *PLoS One* **2012**, *7*, e30938.

[34] O. Zirafi, K. A. Kim, L. Standker, K. B. Mohr, D. Sauter, A. Heigele, S. F. Kluge, E. Wiercinska, D. Chudziak, R. Richter, B. Moepps, P. Gierschik, V. Vas, H. Geiger, M. Lamla, T. Weil, T. Burster, A. Zgraja, F. Daubeuf, N. Frossard, M. Hachet-Haas, F. Heunisch, C. Reichetzeder, J. L. Galzi, J. Perez-Castells, A. Canales-Mayordomo, J. Jimenez-Barbero, G. Gimenez-Gallego, M. Schneider, J. Shorter, A. Telenti, B. Hocher, W. G. Forssmann, H. Bonig, F. Kirchhoff, J. Munch, *Cell Rep.* **2015**, *11*, 737.

[35] X. Cheng, X. Xiao, K. C. Chou, *Gene* **2017**, *628*, 315.

[36] P. Feng, H. Yang, H. Ding, H. Lin, W. Chen, K. C. Chou, *Genomics* **2018**.

[37] B. Liu, K. Li, D. S. Huang, K. C. Chou, *Bioinformatics* **2018**.

[38] X. Cheng, W. Z. Lin, X. Xiao, K. C. Chou, *Bioinformatics* **2018**.

[39] X. Cheng, X. Xiao, K. C. Chou, *J. Theor. Biol.* **2018**, *458*, 92.

[40] K. C. Chou, X. Cheng, X. Xiao, *Genomics* **2018**.

[41] X. Xiao, X. Cheng, G. Chen, Q. Mao, K. C. Chou, *Genomics* **2018**.

[42] R. Mosca, A. Ceol, A. Stein, R. Olivella, P. Aloy, *Nucleic Acids Res.* **2014**, *42*, D374.

[43] R. D. Finn, B. L. Miller, J. Clements, A. Bateman, *Nucleic Acids Res.* **2014**, *42*, D364.

[44] P. Blohm, G. Frishman, P. Smialowski, F. Goebels, B. Wachinger, A. Ruepp, D. Frishman, *Nucleic Acids Res.* **2014**, *42*, D396.

[45] R. D. Finn, P. Coggill, R. Y. Eberhardt, S. R. Eddy, J. Mistry, A. L. Mitchell, S. C. Potter, M. Punta, M. Qureshi, A. Sangrador-Vegas, G. A. Salazar, J. Tate, A. Bateman, *Nucleic Acids Res.* **2016**, *44*, D279.

[46] W. Li, A. Godzik, *Bioinformatics* **2006**, *22*, 1658.

[47] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, D. J. Lipman, *Nucleic Acids Res.* **1997**, *25*, 3389.

[48] K. C. Chou, Y. D. Cai, *Biochem. Biophys. Res. Commun.* **2006**, *339*, 1015.

[49] P. M. Feng, W. Chen, H. Lin, K. C. Chou, *Anal. Biochem.* **2013**, *442*, 118.

[50] J. Jia, Z. Liu, X. Xiao, B. Liu, K. C. Chou, *J. Theor. Biol.* **2016**, *394*, 223.

[51] K. C. Chou, *Med. Chem.* **2015**, *11*, 218.

[52] K. C. Chou, *Proteins* **2001**, *43*, 246.

[53] K. C. Chou, *Bioinformatics* **2005**, *21*, 10.

[54] S. Kawashima, M. Kanehisa, *Nucleic Acids Res.* **2000**, *28*, 374.

[55] B. Hollas, *J. Math. Chem.* **2003**, *33*, 91.

[56] P. Broto, G. Moreau, C. Vandycke, *Eur. J. Med. Chem.* **1984**, *19*, 66.

[57] L. H. Hall, L. B. Kier, *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 1039.

[58] C. E. Shannon, *Bell Syst. Tech. J.* **1948**, *27*, 379.

[59] J. T. Kent, *Biometrika* **1983**, *70*, 163.

[60] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, *SIGKDD Explor. Newsl.* **2009**, *11*, 10.

[61] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Longman Publishing Co., Inc., **1989**.

[62] C. Cortes, V. Vapnik, *Mach. Learn.* **1995**, *20*, 273.

[63] O. Ivanciuc, Applications of Support Vector Machines in Chemistry. In Reviews in Computational Chemistry, John Wiley & Sons, Inc., **2007**, p. 291.

[64] Platt, J., **1998**, p 21.

[65] C. W. Hsu, C. C. Chang, C. J. Lin, A Practical Guide to Support Vector Classification, **2003**.

[66] R. J. Peace, K. K. Biggar, K. B. Storey, J. R. Green, *Nucleic Acids Res.* **2015**, *43*, e138.

[67] J. Han, M. Kamber, J. Pei, Classification. In Data Mining, 3rd ed.; J. Han, M. Kamber, J. Pei, Eds., Morgan Kaufmann, Boston, **2012**, p. 327.

[68] K.-C. Chou, *Peptides* **2001**, *22*, 1973.

[69] T. I. Netzeva, A. Worth, T. Aldenberg, R. Benigni, M. T. Cronin, P. Gramatica, J. S. Jaworska, S. Kahn, G. Klopman, C. A. Marchant, G. Myatt, N. Nikolova-Jeliazkova, G. Y. Patlewicz, R. Perkins, D. Roberts, T. Schultz, D. W. Stanton, J. J. van de Sandt, W. Tong, G. Veith, C. Yang, *Altern. Lab. Anim.* **2005**, *33*, 155.

[70] H. W. Mewes, D. Frishman, U. Guldener, G. Mannhaupt, K. Mayer, M. Mokrejs, B. Morgenstern, M. Munsterkotter, S. Rudd, B. Weil, *Nucleic Acids Res.* **2002**, *30*, 31.

[71] Y. R. Zou, A. H. Kottmann, M. Kuroda, I. Taniuchi, D. R. Littman, *Nature* **1998**, *393*, 595.

[72] E. Oberlin, A. Amara, F. Bachelerie, C. Bessia, J. L. Virelizier, F. Arenzana-Seisdedos, O. Schwartz, J. M. Heard, I. Clark-Lewis, D. F. Legler, M. Loetscher, M. Baggiolini, B. Moser, *Nature* **1996**, *382*, 833.

[73] C. C. Bleul, M. Farzan, H. Choe, C. Parolin, I. Clark-Lewis, J. Sodroski, T. A. Springer, *Nature* **1996**, *382*, 829.

[74] I. Petit, M. Szyper-Kravitz, A. Nagler, M. Lahav, A. Peled, L. Habler, T. Ponomaryov, R. S. Taichman, F. Arenzana-Seisdedos, N. Fujii, J. Sandbank, D. Zipori, T. Lapidot, *Nat. Immunol.* **2002**, *3*, 687.

[75] Y. Nie, J. Waite, F. Brewer, M. J. Sunshine, D. R. Littman, Y. R. Zou, *J. Exp. Med.* **2004**, *200*, 1145.

[76] B. Wu, E. Y. Chien, C. D. Mol, G. Fenalti, W. Liu, V. Katritch, R. Abagyan, A. Brooun, P. Wells, F. C. Bi, D. J. Hamel, P. Kuhn, T. M. Handel, V. Cherezov, R. C. Stevens, *Science* **2010**, *330*, 1066.

Received: 24 September 2018
Revised: 29 November 2018
Accepted: 29 December 2018
Published online on 15 February 2019

Supplementary Material:

PPI-Detect: A Support Vector Machine Model for Sequence-Based Prediction of Protein - Protein Interactions

Sandra Romero-Molina[1]†, *Yasser B. Ruiz-Blanco*[1]†*, *Mirja Harms*[2], *Jan Münch*[2,3], *Elsa Sanchez-Garcia*[1]*

[1] Computational Biochemistry, University of Duisburg-Essen, Germany

[2] Institute of Molecular Virology, Ulm University Medical Center, Germany

[3] Core Facility Functional Peptidomics, Ulm University Medical Center, Germany

† These authors contributed equally. *Corresponding authors

Supplementary File-1 (separated). Summary of the 1922 interacting and 2405 non-interacting pairs of domains, conforming the entire knowledge base.

Section SM1. Annotated configuration file of ProtDCal to compute all pairwise protein descriptors. 2

Table SM2. Summary of names and structural information associated with the 19 descriptors in the model 3

Section SM3. Details of the optimization process for parameters of the SVM method.............................. 4

Table SM4. Summary of performance measures of the model in the training and test sets, as well as in the 10-fold cross validation... 13

Section SM5. PPI-Detect: a simple and user-friendly tool.. 14

Table SM6. Summary of values' range for all descriptors used to train the model................................. 15

Table SM7. Summary of EPI-X4 derivatives, their activities from the study of Zirafi *et al*. and predicted scores using PPI-Detect (score values higher than the one for EPI-X4 indicate higher activity and vice versa).. 16

Section SM1. Annotated configuration file of ProtDCal to compute all pairwise protein descriptors.

```
directory:                                          # Mandatory line marking a section

Datasets/Fasta_Protein_Format                       # Path to directory with the input FASTA file

indices:                                            # Mandatory line marking a section

Gw(U),Gs(U),W(U),Mw,HP,IP,ECI,L1-9,DHf,Z1,Z2,Z3,ISA,Xi,Ap,Pa,Pb,Pt,  # List of used amino-acid properties

groups:                                             # Mandatory line marking a section

ALA,ARG,ASN,ASP,CYS,GLU,GLN,GLY,HIS,ILE,LEU,LYS,MET,PHE,PRO,SER,THR,TRP,TYR,VAL,RTR,BSR,AHR,ALR,NPR,A
RM,PLR,PCR,NCR,UCR,UFR,PRT,                         # List of grouping operators

invariants:                                         # Mandatory line marking a section

N1,N2,N3,Ar,P2,P3,M,G,V,CV,Q3,S,RA,MN,K,Q1,MX,DE,Q2,I50,SI,MI,TI,   # List of aggregation operators

parameters(t_cont,s_cont,A%,HydGroup,n,bins,K,SubG):  # Mandatory line marking a section

4.0,8.0,5.0,9.4,3.0,50,5,3                          # Fixed parameters for internal options of the program

options(decimals,harmonicMeanType,geometricMeanType,windexID,datasetType,outputOrder):  # Mandatory line

3,0,0,4,fasta,true                                  # Fixed parameters for internal options of the program
```

The command line options to calculate pairwise descriptors are described in the README file distributed with ProtDCal-v4.

Table SM2. Summary of names and structural information associated with the 19 descriptors in the model

DESCRIPTOR	RELATED STRUCTURAL INFORMATION
GS(U)_ES_PRT_TI50	Global hydrophobicity
PA_ES_NPR_TI50	Presence of non-polar residues (NPR) weighted with their propensity to form alpha helices (PA)
ISA_ES_NPR_TI50	Presence of non-polar residues (NPR) weighted with their isotropic surface area (ISA, a measure of the non-polar area)
ECI_ES_PRT_TI50	Global polarity
Z3_ES_BSR_TI50	Presence of residues promoting beta sheets (BSR)
GW(U)_ES_PRT_TI50	Global hydrophilicity
IP_ES_GLY_TI50	Presence of glycine residues
MW_ES_ILE_TI50	Presence of Isoleucine residues
W(U)_ES_ALA_TI50	Presence of Alanine residues
PT_ES_ARG_SI50	Presence of Arginine residues weighted with their propensity to form beta turns (PT)
W(U)_ES_PHE_SI50	Presence of phenylalanine residues
PA_ES_PHE_SI50	Presence of phenylalanine residues weighted by their propensity to form alpha helices (PA)
PT_ES_PCR_SI50	Presence of basic residues (ARG, LYS, HIS) weighted with their propensity to form beta turns (PT)
IP_ES_GLY_SI50	Presence of glycine residues
AP_ES_PCR_TI50	Presence of basic residues (ARG, LYS, HIS) weighted with their polar area (AP)
IP_ES_AHR_TI50	Presence of residues promoting alpha helices (AHR)
W(U)_ES_CYS_SI50	Presence of Cysteine residues
MW_ES_ALR_MI50	Presence of aliphatic resides (ALR)
PB_ES_MET_SI50	Presence of Methionine residues weighted with their propensity to form beta sheet (PB)

Glossary:

Gs(U)	An amino acid descriptor obtained as the product of the hydrophobicity and the surface area	[1,2]
Pa	Propensity index to form alpha helices	[3]
Pb	Propensity index to form beta sheets	[3]
Pt	Propensity index to for beta turns	[3]
ISA	isotropic surface area (non-polar area)	[4]
ECI	Electronic charge index (sum of absolute values of the charges of all the residue atoms)	[4]
IP	Isoelectric point	
Mw	Molar weight	
W(U)	Estimated number of water molecules that can coordinate the residue and its adjacent neighbors (a measure of hydrophilicity)	[1,4]
Z3	Coefficients of the third principal component extracted from PCA of multiple physical-chemical properties of amino acids.	[4]
TI50	Total information content with 50 bins (aggregation operator)	[1,5]
SI50	Standardized information content with 50 bins (aggregation operator)	[1,5]
ES	Electro-topological state index adapted for amino acids (vicinity operator)	[1,6]

*Additional information on values of the amino acid properties and the mathematical formalism of the operator can be found in the documentation of ProtDCal (http://bioinf.sce.carleton.ca/ProtDCal/) and in the Supplementary Materials of [1]

Section SM3. Details of the optimization process for parameters of the SVM method.

A grid search was performed along a search space defined by the following range of values for the parameters of the SVM and its kernel function:

- Cost (C) parameter of the SVM: 2^{-5}, $2^{-4.5}$, 2^{-4}, $2^{-3.5}$, 2^{-3}, …, 2^{1}, $2^{1.5}$, 2^{2}, $2^{2.5}$, 2^{3}, $2^{3.5}$ and 2^{4}
- Exponent (Exp) degree of the polynomial function kernel: 1, 2 and 3
- Gamma (G) parameter of the radial basis function kernel: 2^{-15}, 2^{-14}, 2^{-13}, …, 2^{-1}, 2^{0}, 2^{1}, 2^{2} and 2^{3}

The gridding approach and the range of values for each parameter were set in accordance to established guidelines [7]. The evaluation of the intermediate models was carried out by means of 10-fold cross-validation. The performance was monitored by keeping record of the false positive rate (FPR), precision (Pr), and sensitivity (Sn) for the positive class, *i.e.* the interacting proteins. A combined score defined as: Score = Pr + Sn – FPR, was used to extract the optimum model.

After an exploration with both kernel functions, the optimum model was extracted based on the combined score. This approach led to the identification of a model with C = 11.31371 and the linear kernel (SMO_PolyK_C11.31371_Exp1) as the optimum classifier for our training data. Tables 4.1 and 4.2 summarize the values of the performance measures for each of the intermediate models.

Table S3.1. List of intermediate models and performance measures using the polynomial function kernel.

MODEL	FPR	PRECISION	SENSITIVITY	SCORE
SMO_POLYK_C0.03125_EXP1	0,166	0,722	0,501	1,057
SMO_POLYK_C0.03125_EXP2	0,171	0,730	0,540	1,099
SMO_POLYK_C0.03125_EXP3	0,185	0,717	0,544	1,076
SMO_POLYK_C0.04419_EXP1	0,165	0,722	0,500	1,057
SMO_POLYK_C0.04419_EXP2	0,177	0,724	0,543	1,090
SMO_POLYK_C0.04419_EXP3	0,185	0,714	0,539	1,068
SMO_POLYK_C0.0625_EXP1	0,166	0,725	0,509	1,068
SMO_POLYK_C0.0625_EXP2	0,187	0,714	0,543	1,070
SMO_POLYK_C0.0625_EXP3	0,184	0,715	0,537	1,068
SMO_POLYK_C0.08839_EXP1	0,167	0,724	0,512	1,069
SMO_POLYK_C0.08839_EXP2	0,189	0,713	0,548	1,072
SMO_POLYK_C0.08839_EXP3	0,183	0,715	0,536	1,068
SMO_POLYK_C0.125_EXP1	0,166	0,728	0,519	1,081
SMO_POLYK_C0.125_EXP2	0,190	0,712	0,545	1,067
SMO_POLYK_C0.125_EXP3	0,180	0,716	0,529	1,065
SMO_POLYK_C0.17678_EXP1	0,168	0,729	0,526	1,087
SMO_POLYK_C0.17678_EXP2	0,190	0,710	0,543	1,063
SMO_POLYK_C0.17678_EXP3	0,183	0,714	0,533	1,064
SMO_POLYK_C0.25_EXP1	0,174	0,724	0,532	1,082
SMO_POLYK_C0.25_EXP2	0,190	0,711	0,544	1,065
SMO_POLYK_C0.25_EXP3	0,185	0,713	0,534	1,062
SMO_POLYK_C0.35355_EXP1	0,176	0,723	0,535	1,082
SMO_POLYK_C0.35355_EXP2	0,188	0,714	0,549	1,075
SMO_POLYK_C0.35355_EXP3	0,191	0,707	0,537	1,053

SMO_POLYK_C0.5_EXP1	0,175	0,725	0,537	1,087
SMO_POLYK_C0.5_EXP2	0,185	0,717	0,547	1,079
SMO_POLYK_C0.5_EXP3	0,192	0,706	0,538	1,052
SMO_POLYK_C0.70711_EXP1	0,171	0,731	0,542	1,102
SMO_POLYK_C0.70711_EXP2	0,182	0,721	0,547	1,086
SMO_POLYK_C0.70711_EXP3	0,196	0,704	0,543	1,051
SMO_POLYK_C1_EXP1	0,170	0,731	0,540	1,101
SMO_POLYK_C1_EXP2	0,182	0,72	0,543	1,081
SMO_POLYK_C1_EXP3	0,198	0,705	0,550	1,057
SMO_POLYK_C1.41421_EXP1	0,174	0,727	0,539	1,092
SMO_POLYK_C1.41421_EXP2	0,182	0,719	0,540	1,077
SMO_POLYK_C1.41421_EXP3	0,201	0,702	0,553	1,054
SMO_POLYK_C2_EXP1	0,173	0,729	0,542	1,098
SMO_POLYK_C2_EXP2	0,178	0,721	0,537	1,080
SMO_POLYK_C2_EXP3	0,209	0,696	0,556	1,043
SMO_POLYK_C2.82843_EXP1	0,173	0,729	0,542	1,098
SMO_POLYK_C2.82843_EXP2	0,175	0,726	0,539	1,090
SMO_POLYK_C2.82843_EXP3	0,213	0,694	0,563	1,044
SMO_POLYK_C4_EXP1	0,172	0,730	0,542	1,100
SMO_POLYK_C4_EXP2	0,174	0,727	0,537	1,090
SMO_POLYK_C4_EXP3	0,224	0,687	0,572	1,035
SMO_POLYK_C5.65685_EXP1	0,173	0,730	0,542	1,099
SMO_POLYK_C5.65685_EXP2	0,171	0,728	0,536	1,093
SMO_POLYK_C5.65685_EXP3	0,228	0,683	0,572	1,027
SMO_POLYK_C8_EXP1	0,172	0,731	0,544	1,103
SMO_POLYK_C8_EXP2	0,171	0,730	0,539	1,098
SMO_POLYK_C11.31371_EXP1	**0,170**	**0,733**	**0,545**	**1,108**
SMO_POLYK_C11.31371_EXP2	0,169	0,732	0,538	1,101
SMO_POLYK_C16_EXP1	0,171	0,732	0,545	1,106
SMO_POLYK_C16_EXP2	0,174	0,727	0,538	1,091

Table S3.2. List of intermediate models and performance measures using the RBF kernel.

MODEL	FPR	PRECISION	SENSITIVITY	SCORE
SMO_RBF_C0.03125_G0.0001220703125	0,243	0,649	0,522	0,928
SMO_RBF_C0.03125_G0.000244140625	0,226	0,662	0,517	0,953
SMO_RBF_C0.03125_G0.00048828125	0,218	0,667	0,509	0,958
SMO_RBF_C0.03125_G0.0009765625	0,214	0,671	0,509	0,966
SMO_RBF_C0.03125_G0.001953125	0,214	0,671	0,509	0,966
SMO_RBF_C0.03125_G0.00390625	0,215	0,67	0,51	0,965
SMO_RBF_C0.03125_G0.0078125	0,217	0,669	0,509	0,961
SMO_RBF_C0.03125_G0.015625	0,218	0,667	0,509	0,958
SMO_RBF_C0.03125_G0.03125	0,22	0,666	0,512	0,958
SMO_RBF_C0.03125_G0.0625	0,224	0,662	0,512	0,95
SMO_RBF_C0.03125_G0.125	0,206	0,675	0,497	0,966
SMO_RBF_C0.03125_G0.25	0,206	0,674	0,496	0,964
SMO_RBF_C0.03125_G0.5	0,204	0,67	0,484	0,95
SMO_RBF_C0.03125_G1	0,188	0,679	0,464	0,955
SMO_RBF_C0.03125_G2	0,171	0,677	0,419	0,925
SMO_RBF_C0.03125_G3.0517578125E-05	0,248	0,639	0,51	0,901

SMO_RBF_C0.03125_G4	0,161	0,692	0,423	0,954
SMO_RBF_C0.03125_G6.103515625E-05	0,243	0,644	0,512	0,913
SMO_RBF_C0.03125_G8	0,188	0,698	0,506	1,016
SMO_RBF_C0.04419_G0.0001220703125	0,237	0,653	0,52	0,936
SMO_RBF_C0.04419_G0.000244140625	0,221	0,666	0,512	0,957
SMO_RBF_C0.04419_G0.00048828125	0,216	0,669	0,51	0,963
SMO_RBF_C0.04419_G0.0009765625	0,214	0,671	0,507	0,964
SMO_RBF_C0.04419_G0.001953125	0,214	0,671	0,509	0,966
SMO_RBF_C0.04419_G0.00390625	0,215	0,671	0,509	0,965
SMO_RBF_C0.04419_G0.0078125	0,217	0,668	0,509	0,96
SMO_RBF_C0.04419_G0.015625	0,218	0,667	0,509	0,958
SMO_RBF_C0.04419_G0.03125	0,22	0,665	0,511	0,956
SMO_RBF_C0.04419_G0.0625	0,207	0,673	0,495	0,961
SMO_RBF_C0.04419_G0.125	0,184	0,694	0,484	0,994
SMO_RBF_C0.04419_G0.25	0,207	0,675	0,502	0,97
SMO_RBF_C0.04419_G0.5	0,198	0,681	0,491	0,974
SMO_RBF_C0.04419_G1	0,19	0,678	0,467	0,955
SMO_RBF_C0.04419_G2	0,176	0,679	0,435	0,938
SMO_RBF_C0.04419_G3.0517578125E-05	0,248	0,639	0,51	0,901
SMO_RBF_C0.04419_G4	0,163	0,697	0,438	0,972
SMO_RBF_C0.04419_G6.103515625E-05	0,242	0,648	0,519	0,925
SMO_RBF_C0.04419_G8	0,186	0,699	0,504	1,017
SMO_RBF_C0.0625_G0.0001220703125	0,226	0,662	0,517	0,953
SMO_RBF_C0.0625_G0.000244140625	0,219	0,667	0,51	0,958
SMO_RBF_C0.0625_G0.00048828125	0,214	0,672	0,509	0,967
SMO_RBF_C0.0625_G0.0009765625	0,214	0,671	0,509	0,966
SMO_RBF_C0.0625_G0.001953125	0,214	0,671	0,508	0,965
SMO_RBF_C0.0625_G0.00390625	0,216	0,669	0,509	0,962
SMO_RBF_C0.0625_G0.0078125	0,217	0,669	0,509	0,961
SMO_RBF_C0.0625_G0.015625	0,22	0,666	0,512	0,958
SMO_RBF_C0.0625_G0.03125	0,219	0,667	0,512	0,96
SMO_RBF_C0.0625_G0.0625	0,182	0,692	0,475	0,985
SMO_RBF_C0.0625_G0.125	0,189	0,69	0,49	0,991
SMO_RBF_C0.0625_G0.25	0,19	0,693	0,498	1,001
SMO_RBF_C0.0625_G0.5	0,183	0,698	0,491	1,006
SMO_RBF_C0.0625_G1	0,188	0,685	0,475	0,972
SMO_RBF_C0.0625_G2	0,177	0,687	0,452	0,962
SMO_RBF_C0.0625_G3.0517578125E-05	0,243	0,644	0,512	0,913
SMO_RBF_C0.0625_G4	0,164	0,705	0,457	0,998
SMO_RBF_C0.0625_G6.103515625E-05	0,243	0,649	0,522	0,928
SMO_RBF_C0.0625_G8	0,184	0,7	0,502	1,018
SMO_RBF_C0.08839_G0.0001220703125	0,221	0,666	0,512	0,957
SMO_RBF_C0.08839_G0.000244140625	0,214	0,672	0,509	0,967
SMO_RBF_C0.08839_G0.00048828125	0,214	0,67	0,507	0,963
SMO_RBF_C0.08839_G0.0009765625	0,214	0,671	0,509	0,966
SMO_RBF_C0.08839_G0.001953125	0,214	0,671	0,507	0,964
SMO_RBF_C0.08839_G0.00390625	0,215	0,671	0,509	0,965
SMO_RBF_C0.08839_G0.0078125	0,216	0,67	0,511	0,965
SMO_RBF_C0.08839_G0.015625	0,219	0,667	0,512	0,96
SMO_RBF_C0.08839_G0.03125	0,19	0,686	0,484	0,98

SMO_RBF_C0.08839_G0.0625	0,175	0,698	0,47	0,993
SMO_RBF_C0.08839_G0.125	0,188	0,695	0,498	1,005
SMO_RBF_C0.08839_G0.25	0,179	0,703	0,494	1,018
SMO_RBF_C0.08839_G0.5	0,171	0,712	0,494	1,035
SMO_RBF_C0.08839_G1	0,182	0,697	0,488	1,003
SMO_RBF_C0.08839_G2	0,174	0,7	0,471	0,997
SMO_RBF_C0.08839_G3.0517578125E-05	0,242	0,649	0,52	0,927
SMO_RBF_C0.08839_G4	0,167	0,709	0,473	1,015
SMO_RBF_C0.08839_G6.103515625E-05	0,236	0,654	0,52	0,938
SMO_RBF_C0.08839_G8	0,185	0,702	0,507	1,024
SMO_RBF_C0.125_G0.0001220703125	0,218	0,667	0,509	0,958
SMO_RBF_C0.125_G0.000244140625	0,213	0,673	0,51	0,97
SMO_RBF_C0.125_G0.00048828125	0,214	0,67	0,507	0,963
SMO_RBF_C0.125_G0.0009765625	0,215	0,671	0,51	0,966
SMO_RBF_C0.125_G0.001953125	0,214	0,671	0,509	0,966
SMO_RBF_C0.125_G0.00390625	0,216	0,67	0,511	0,965
SMO_RBF_C0.125_G0.0078125	0,217	0,668	0,509	0,96
SMO_RBF_C0.125_G0.015625	0,215	0,67	0,51	0,965
SMO_RBF_C0.125_G0.03125	0,169	0,7	0,461	0,992
SMO_RBF_C0.125_G0.0625	0,181	0,697	0,486	1,002
SMO_RBF_C0.125_G0.125	0,176	0,709	0,499	1,032
SMO_RBF_C0.125_G0.25	0,171	0,714	0,498	1,041
SMO_RBF_C0.125_G0.5	0,17	0,715	0,498	1,043
SMO_RBF_C0.125_G1	0,178	0,702	0,49	1,014
SMO_RBF_C0.125_G2	0,172	0,706	0,481	1,015
SMO_RBF_C0.125_G3.0517578125E-05	0,243	0,649	0,522	0,928
SMO_RBF_C0.125_G4	0,169	0,709	0,481	1,021
SMO_RBF_C0.125_G6.103515625E-05	0,227	0,662	0,517	0,952
SMO_RBF_C0.125_G8	0,19	0,697	0,507	1,014
SMO_RBF_C0.17678_G0.0001220703125	0,214	0,671	0,509	0,966
SMO_RBF_C0.17678_G0.000244140625	0,214	0,672	0,511	0,969
SMO_RBF_C0.17678_G0.00048828125	0,214	0,671	0,509	0,966
SMO_RBF_C0.17678_G0.0009765625	0,214	0,671	0,508	0,965
SMO_RBF_C0.17678_G0.001953125	0,216	0,669	0,509	0,962
SMO_RBF_C0.17678_G0.00390625	0,215	0,671	0,511	0,967
SMO_RBF_C0.17678_G0.0078125	0,217	0,669	0,511	0,963
SMO_RBF_C0.17678_G0.015625	0,181	0,695	0,48	0,994
SMO_RBF_C0.17678_G0.03125	0,169	0,701	0,462	0,994
SMO_RBF_C0.17678_G0.0625	0,181	0,704	0,499	1,022
SMO_RBF_C0.17678_G0.125	0,17	0,717	0,502	1,049
SMO_RBF_C0.17678_G0.25	0,167	0,722	0,503	1,058
SMO_RBF_C0.17678_G0.5	0,169	0,718	0,502	1,051
SMO_RBF_C0.17678_G1	0,173	0,713	0,499	1,039
SMO_RBF_C0.17678_G2	0,177	0,704	0,491	1,018
SMO_RBF_C0.17678_G3.0517578125E-05	0,236	0,654	0,52	0,938
SMO_RBF_C0.17678_G4	0,179	0,703	0,494	1,018
SMO_RBF_C0.17678_G6.103515625E-05	0,222	0,664	0,512	0,954
SMO_RBF_C0.17678_G8	0,197	0,692	0,516	1,011
SMO_RBF_C0.25_G0.0001220703125	0,214	0,672	0,509	0,967
SMO_RBF_C0.25_G0.000244140625	0,215	0,67	0,508	0,963

SMO_RBF_C0.25_G0.00048828125	0,214	0,671	0,509	0,966
SMO_RBF_C0.25_G0.0009765625	0,215	0,669	0,507	0,961
SMO_RBF_C0.25_G0.001953125	0,214	0,67	0,507	0,963
SMO_RBF_C0.25_G0.00390625	0,215	0,67	0,509	0,964
SMO_RBF_C0.25_G0.0078125	0,213	0,672	0,507	0,966
SMO_RBF_C0.25_G0.015625	0,166	0,702	0,456	0,992
SMO_RBF_C0.25_G0.03125	0,175	0,703	0,483	1,011
SMO_RBF_C0.25_G0.0625	0,173	0,712	0,499	1,038
SMO_RBF_C0.25_G0.125	0,166	0,723	0,506	1,063
SMO_RBF_C0.25_G0.25	0,163	0,728	0,509	1,074
SMO_RBF_C0.25_G0.5	0,161	0,728	0,502	1,069
SMO_RBF_C0.25_G1	0,168	0,721	0,504	1,057
SMO_RBF_C0.25_G2	0,178	0,707	0,5	1,029
SMO_RBF_C0.25_G3.0517578125E-05	0,226	0,663	0,517	0,954
SMO_RBF_C0.25_G4	0,187	0,699	0,505	1,017
SMO_RBF_C0.25_G6.103515625E-05	0,218	0,667	0,51	0,959
SMO_RBF_C0.25_G8	0,208	0,687	0,53	1,009
SMO_RBF_C0.35355_G0.0001220703125	0,214	0,67	0,507	0,963
SMO_RBF_C0.35355_G0.000244140625	0,215	0,67	0,509	0,964
SMO_RBF_C0.35355_G0.00048828125	0,214	0,671	0,509	0,966
SMO_RBF_C0.35355_G0.0009765625	0,215	0,67	0,51	0,965
SMO_RBF_C0.35355_G0.001953125	0,218	0,667	0,508	0,957
SMO_RBF_C0.35355_G0.00390625	0,217	0,669	0,511	0,963
SMO_RBF_C0.35355_G0.0078125	0,179	0,694	0,475	0,99
SMO_RBF_C0.35355_G0.015625	0,167	0,704	0,461	0,998
SMO_RBF_C0.35355_G0.03125	0,177	0,709	0,502	1,034
SMO_RBF_C0.35355_G0.0625	0,168	0,72	0,504	1,056
SMO_RBF_C0.35355_G0.125	0,163	0,729	0,512	1,078
SMO_RBF_C0.35355_G0.25	0,158	0,735	0,511	1,088
SMO_RBF_C0.35355_G0.5	0,159	0,734	0,509	1,084
SMO_RBF_C0.35355_G1	0,167	0,723	0,508	1,064
SMO_RBF_C0.35355_G2	0,183	0,705	0,507	1,029
SMO_RBF_C0.35355_G3.0517578125E-05	0,222	0,665	0,512	0,955
SMO_RBF_C0.35355_G4	0,202	0,691	0,525	1,014
SMO_RBF_C0.35355_G6.103515625E-05	0,215	0,67	0,509	0,964
SMO_RBF_C0.35355_G8	0,211	0,688	0,543	1,02
SMO_RBF_C0.5_G0.0001220703125	0,214	0,67	0,507	0,963
SMO_RBF_C0.5_G0.000244140625	0,214	0,671	0,508	0,965
SMO_RBF_C0.5_G0.00048828125	0,215	0,671	0,509	0,965
SMO_RBF_C0.5_G0.0009765625	0,215	0,67	0,509	0,964
SMO_RBF_C0.5_G0.001953125	0,214	0,671	0,509	0,966
SMO_RBF_C0.5_G0.00390625	0,211	0,672	0,504	0,965
SMO_RBF_C0.5_G0.0078125	0,164	0,703	0,452	0,991
SMO_RBF_C0.5_G0.015625	0,175	0,705	0,485	1,015
SMO_RBF_C0.5_G0.03125	0,17	0,718	0,504	1,052
SMO_RBF_C0.5_G0.0625	0,167	0,726	0,514	1,073
SMO_RBF_C0.5_G0.125	0,158	0,737	0,514	1,093
SMO_RBF_C0.5_G0.25	0,16	0,733	0,514	1,087
SMO_RBF_C0.5_G0.5	0,165	0,73	0,517	1,082
SMO_RBF_C0.5_G1	0,168	0,724	0,516	1,072

SMO_RBF_C0.5_G2	0,183	0,707	0,514	1,038
SMO_RBF_C0.5_G3.0517578125E-05	0,219	0,666	0,509	0,956
SMO_RBF_C0.5_G4	0,203	0,693	0,535	1,025
SMO_RBF_C0.5_G6.103515625E-05	0,214	0,671	0,509	0,966
SMO_RBF_C0.5_G8	0,218	0,686	0,555	1,023
SMO_RBF_C0.70711_G0.0001220703125	0,215	0,671	0,509	0,965
SMO_RBF_C0.70711_G0.000244140625	0,214	0,67	0,507	0,963
SMO_RBF_C0.70711_G0.00048828125	0,215	0,671	0,51	0,966
SMO_RBF_C0.70711_G0.0009765625	0,216	0,669	0,509	0,962
SMO_RBF_C0.70711_G0.001953125	0,215	0,671	0,511	0,967
SMO_RBF_C0.70711_G0.00390625	0,178	0,696	0,475	0,993
SMO_RBF_C0.70711_G0.0078125	0,164	0,706	0,46	1,002
SMO_RBF_C0.70711_G0.015625	0,171	0,715	0,501	1,045
SMO_RBF_C0.70711_G0.03125	0,172	0,715	0,503	1,046
SMO_RBF_C0.70711_G0.0625	0,163	0,731	0,516	1,084
SMO_RBF_C0.70711_G0.125	0,159	0,737	0,517	1,095
SMO_RBF_C0.70711_G0.25	0,158	0,739	0,519	1,1
SMO_RBF_C0.70711_G0.5	0,166	0,728	0,519	1,081
SMO_RBF_C0.70711_G1	0,174	0,72	0,52	1,066
SMO_RBF_C0.70711_G2	0,191	0,703	0,528	1,04
SMO_RBF_C0.70711_G3.0517578125E-05	0,215	0,67	0,509	0,964
SMO_RBF_C0.70711_G4	0,207	0,691	0,54	1,024
SMO_RBF_C0.70711_G6.103515625E-05	0,214	0,671	0,507	0,964
SMO_RBF_C0.70711_G8	0,234	0,672	0,558	0,996
SMO_RBF_C1_G0.0001220703125	0,215	0,671	0,51	0,966
SMO_RBF_C1_G0.000244140625	0,215	0,67	0,509	0,964
SMO_RBF_C1_G0.00048828125	0,215	0,67	0,509	0,964
SMO_RBF_C1_G0.0009765625	0,214	0,672	0,509	0,967
SMO_RBF_C1_G0.001953125	0,212	0,671	0,504	0,963
SMO_RBF_C1_G0.00390625	0,16	0,707	0,451	0,998
SMO_RBF_C1_G0.0078125	0,173	0,706	0,485	1,018
SMO_RBF_C1_G0.015625	0,17	0,717	0,503	1,05
SMO_RBF_C1_G0.03125	0,169	0,722	0,512	1,065
SMO_RBF_C1_G0.0625	0,164	0,731	0,519	1,086
SMO_RBF_C1_G0.125	0,157	0,74	0,519	1,102
SMO_RBF_C1_G0.25	0,159	0,738	0,522	1,101
SMO_RBF_C1_G0.5	0,168	0,728	0,525	1,085
SMO_RBF_C1_G1	0,182	0,713	0,525	1,056
SMO_RBF_C1_G2	0,194	0,702	0,533	1,041
SMO_RBF_C1_G3.0517578125E-05	0,214	0,671	0,509	0,966
SMO_RBF_C1_G4	0,214	0,686	0,545	1,017
SMO_RBF_C1_G6.103515625E-05	0,214	0,67	0,507	0,963
SMO_RBF_C1_G8	0,249	0,659	0,561	0,971
SMO_RBF_C1.41421_G0.0001220703125	0,215	0,67	0,508	0,963
SMO_RBF_C1.41421_G0.000244140625	0,215	0,671	0,511	0,967
SMO_RBF_C1.41421_G0.00048828125	0,217	0,668	0,509	0,96
SMO_RBF_C1.41421_G0.0009765625	0,217	0,67	0,512	0,965
SMO_RBF_C1.41421_G0.001953125	0,178	0,696	0,473	0,991
SMO_RBF_C1.41421_G0.00390625	0,163	0,708	0,46	1,005
SMO_RBF_C1.41421_G0.0078125	0,17	0,716	0,501	1,047

SMO_RBF_C1.41421_G0.015625	0,168	0,719	0,5	1,051
SMO_RBF_C1.41421_G0.03125	0,169	0,724	0,517	1,072
SMO_RBF_C1.41421_G0.0625	0,163	0,733	0,522	1,092
SMO_RBF_C1.41421_G0.125	0,158	0,74	0,523	1,105
SMO_RBF_C1.41421_G0.25	0,165	0,735	0,532	1,102
SMO_RBF_C1.41421_G0.5	0,177	0,721	0,533	1,077
SMO_RBF_C1.41421_G1	0,187	0,711	0,535	1,059
SMO_RBF_C1.41421_G2	0,198	0,699	0,537	1,038
SMO_RBF_C1.41421_G3.0517578125E-05	0,214	0,671	0,507	0,964
SMO_RBF_C1.41421_G4	0,225	0,677	0,551	1,003
SMO_RBF_C1.41421_G6.103515625E-05	0,215	0,67	0,508	0,963
SMO_RBF_C1.41421_G8	0,261	0,652	0,571	0,962
SMO_RBF_C2_G0.0001220703125	0,215	0,67	0,509	0,964
SMO_RBF_C2_G0.000244140625	0,215	0,67	0,509	0,964
SMO_RBF_C2_G0.00048828125	0,215	0,669	0,507	0,961
SMO_RBF_C2_G0.0009765625	0,211	0,672	0,503	0,964
SMO_RBF_C2_G0.001953125	0,16	0,707	0,451	0,998
SMO_RBF_C2_G0.00390625	0,173	0,707	0,485	1,019
SMO_RBF_C2_G0.0078125	0,168	0,72	0,502	1,054
SMO_RBF_C2_G0.015625	0,167	0,723	0,507	1,063
SMO_RBF_C2_G0.03125	0,167	0,728	0,521	1,082
SMO_RBF_C2_G0.0625	0,162	0,734	0,52	1,092
SMO_RBF_C2_G0.125	0,164	0,734	0,526	1,096
SMO_RBF_C2_G0.25	0,175	0,723	0,532	1,08
SMO_RBF_C2_G0.5	0,183	0,715	0,535	1,067
SMO_RBF_C2_G1	0,194	0,704	0,537	1,047
SMO_RBF_C2_G2	0,21	0,691	0,546	1,027
SMO_RBF_C2_G3.0517578125E-05	0,214	0,67	0,507	0,963
SMO_RBF_C2_G4	0,238	0,666	0,554	0,982
SMO_RBF_C2_G6.103515625E-05	0,214	0,671	0,509	0,966
SMO_RBF_C2_G8	0,274	0,645	0,581	0,952
SMO_RBF_C2.82843_G0.0001220703125	0,215	0,67	0,509	0,964
SMO_RBF_C2.82843_G0.000244140625	0,216	0,67	0,51	0,964
SMO_RBF_C2.82843_G0.00048828125	0,216	0,669	0,509	0,962
SMO_RBF_C2.82843_G0.0009765625	0,178	0,695	0,471	0,988
SMO_RBF_C2.82843_G0.001953125	0,163	0,708	0,461	1,006
SMO_RBF_C2.82843_G0.00390625	0,17	0,717	0,502	1,049
SMO_RBF_C2.82843_G0.0078125	0,168	0,719	0,502	1,053
SMO_RBF_C2.82843_G0.015625	0,168	0,726	0,517	1,075
SMO_RBF_C2.82843_G0.03125	0,168	0,728	0,524	1,084
SMO_RBF_C2.82843_G0.0625	0,159	0,739	0,524	1,104
SMO_RBF_C2.82843_G0.125	0,167	0,732	0,53	1,095
SMO_RBF_C2.82843_G0.25	0,179	0,719	0,534	1,074
SMO_RBF_C2.82843_G0.5	0,184	0,714	0,533	1,063
SMO_RBF_C2.82843_G1	0,196	0,701	0,538	1,043
SMO_RBF_C2.82843_G2	0,218	0,685	0,553	1,02
SMO_RBF_C2.82843_G3.0517578125E-05	0,215	0,67	0,509	0,964
SMO_RBF_C2.82843_G4	0,248	0,659	0,559	0,97
SMO_RBF_C2.82843_G6.103515625E-05	0,214	0,671	0,508	0,965
SMO_RBF_C2.82843_G8	0,281	0,641	0,586	0,946

SMO_RBF_C4_G0.0001220703125	0,216	0,669	0,507	0,96
SMO_RBF_C4_G0.000244140625	0,214	0,671	0,509	0,966
SMO_RBF_C4_G0.00048828125	0,211	0,672	0,502	0,963
SMO_RBF_C4_G0.0009765625	0,16	0,708	0,451	0,999
SMO_RBF_C4_G0.001953125	0,173	0,706	0,483	1,016
SMO_RBF_C4_G0.00390625	0,168	0,719	0,502	1,053
SMO_RBF_C4_G0.0078125	0,167	0,724	0,509	1,066
SMO_RBF_C4_G0.015625	0,171	0,724	0,524	1,077
SMO_RBF_C4_G0.03125	0,163	0,735	0,525	1,097
SMO_RBF_C4_G0.0625	0,163	0,737	0,531	1,105
SMO_RBF_C4_G0.125	0,171	0,729	0,535	1,093
SMO_RBF_C4_G0.25	0,178	0,721	0,537	1,08
SMO_RBF_C4_G0.5	0,184	0,713	0,53	1,059
SMO_RBF_C4_G1	0,196	0,703	0,539	1,046
SMO_RBF_C4_G2	0,223	0,68	0,551	1,008
SMO_RBF_C4_G3.0517578125E-05	0,214	0,671	0,508	0,965
SMO_RBF_C4_G4	0,257	0,655	0,568	0,966
SMO_RBF_C4_G6.103515625E-05	0,215	0,67	0,509	0,964
SMO_RBF_C4_G8	0,296	0,633	0,596	0,933
SMO_RBF_C5.65685_G0.0001220703125	0,216	0,669	0,509	0,962
SMO_RBF_C5.65685_G0.000244140625	0,214	0,671	0,509	0,966
SMO_RBF_C5.65685_G0.00048828125	0,178	0,695	0,471	0,988
SMO_RBF_C5.65685_G0.0009765625	0,163	0,707	0,46	1,004
SMO_RBF_C5.65685_G0.001953125	0,169	0,719	0,502	1,052
SMO_RBF_C5.65685_G0.00390625	0,166	0,721	0,501	1,056
SMO_RBF_C5.65685_G0.0078125	0,167	0,726	0,515	1,074
SMO_RBF_C5.65685_G0.015625	0,168	0,729	0,528	1,089
SMO_RBF_C5.65685_G0.03125	0,165	0,733	0,528	1,096
SMO_RBF_C5.65685_G0.0625	0,169	0,73	0,533	1,094
SMO_RBF_C5.65685_G0.125	0,174	0,727	0,54	1,093
SMO_RBF_C5.65685_G0.25	0,179	0,717	0,529	1,067
SMO_RBF_C5.65685_G0.5	0,187	0,709	0,532	1,054
SMO_RBF_C5.65685_G1	0,204	0,696	0,543	1,035
SMO_RBF_C5.65685_G2	0,237	0,668	0,556	0,987
SMO_RBF_C5.65685_G3.0517578125E-05	0,215	0,67	0,507	0,962
SMO_RBF_C5.65685_G4	0,27	0,647	0,578	0,955
SMO_RBF_C5.65685_G6.103515625E-05	0,215	0,671	0,509	0,965
SMO_RBF_C5.65685_G8	0,312	0,625	0,606	0,919
SMO_RBF_C8_G0.0001220703125	0,214	0,671	0,509	0,966
SMO_RBF_C8_G0.000244140625	0,211	0,672	0,503	0,964
SMO_RBF_C8_G0.00048828125	0,16	0,708	0,451	0,999
SMO_RBF_C8_G0.0009765625	0,172	0,707	0,483	1,018
SMO_RBF_C8_G0.001953125	0,167	0,721	0,502	1,056
SMO_RBF_C8_G0.00390625	0,166	0,726	0,511	1,071
SMO_RBF_C8_G0.0078125	0,172	0,723	0,524	1,075
SMO_RBF_C8_G0.015625	0,169	0,73	0,53	1,091
SMO_RBF_C8_G0.03125	0,164	0,736	0,533	1,105
SMO_RBF_C8_G0.0625	0,173	0,726	0,533	1,086
SMO_RBF_C8_G0.125	0,18	0,72	0,538	1,078
SMO_RBF_C8_G0.25	0,177	0,721	0,533	1,077

SMO_RBF_C8_G0.5	0,185	0,711	0,53	1,056
SMO_RBF_C8_G1	0,209	0,692	0,549	1,032
SMO_RBF_C8_G2	0,242	0,667	0,564	0,989
SMO_RBF_C8_G3.0517578125E-05	0,215	0,67	0,509	0,964
SMO_RBF_C8_G4	0,284	0,637	0,581	0,934
SMO_RBF_C8_G6.103515625E-05	0,217	0,667	0,507	0,957
SMO_RBF_C8_G8	0,318	0,622	0,609	0,913
SMO_RBF_C11.31371_G0.0001220703125	0,215	0,67	0,508	0,963
SMO_RBF_C11.31371_G0.000244140625	0,177	0,695	0,471	0,989
SMO_RBF_C11.31371_G0.00048828125	0,162	0,709	0,46	1,007
SMO_RBF_C11.31371_G0.0009765625	0,169	0,719	0,502	1,052
SMO_RBF_C11.31371_G0.001953125	0,165	0,723	0,5	1,058
SMO_RBF_C11.31371_G0.00390625	0,165	0,727	0,512	1,074
SMO_RBF_C11.31371_G0.0078125	0,173	0,725	0,53	1,082
SMO_RBF_C11.31371_G0.015625	0,168	0,731	0,532	1,095
SMO_RBF_C11.31371_G0.03125	0,167	0,733	0,535	1,101
SMO_RBF_C11.31371_G0.0625	0,177	0,722	0,536	1,081
SMO_RBF_C11.31371_G0.125	0,179	0,719	0,536	1,076
SMO_RBF_C11.31371_G0.25	0,177	0,72	0,529	1,072
SMO_RBF_C11.31371_G0.5	0,187	0,711	0,538	1,062
SMO_RBF_C11.31371_G1	0,223	0,68	0,553	1,01
SMO_RBF_C11.31371_G2	0,253	0,659	0,571	0,977
SMO_RBF_C11.31371_G3.0517578125E-05	0,215	0,67	0,509	0,964
SMO_RBF_C11.31371_G4	0,293	0,633	0,589	0,929
SMO_RBF_C11.31371_G6.103515625E-05	0,216	0,669	0,508	0,961
SMO_RBF_C11.31371_G8	0,328	0,616	0,614	0,902
SMO_RBF_C16_G0.0001220703125	0,211	0,672	0,502	0,963
SMO_RBF_C16_G0.000244140625	0,16	0,708	0,452	1
SMO_RBF_C16_G0.00048828125	0,173	0,706	0,483	1,016
SMO_RBF_C16_G0.0009765625	0,166	0,721	0,501	1,056
SMO_RBF_C16_G0.001953125	0,168	0,723	0,51	1,065
SMO_RBF_C16_G0.00390625	0,167	0,729	0,521	1,083
SMO_RBF_C16_G0.0078125	0,171	0,727	0,532	1,088
SMO_RBF_C16_G0.015625	0,167	0,735	0,537	1,105
SMO_RBF_C16_G0.03125	0,17	0,73	0,536	1,096
SMO_RBF_C16_G0.0625	0,176	0,724	0,538	1,086
SMO_RBF_C16_G0.125	0,176	0,723	0,537	1,084
SMO_RBF_C16_G0.25	0,169	0,73	0,532	1,093
SMO_RBF_C16_G0.5	0,195	0,704	0,542	1,051
SMO_RBF_C16_G1	0,231	0,676	0,56	1,005
SMO_RBF_C16_G2	0,267	0,651	0,58	0,964
SMO_RBF_C16_G3.0517578125E-05	0,216	0,669	0,508	0,961
SMO_RBF_C16_G4	0,305	0,625	0,591	0,911
SMO_RBF_C16_G6.103515625E-05	0,216	0,669	0,509	0,962
SMO_RBF_C16_G8	0,333	0,613	0,614	0,894

Table SM4. Summary of performance measures of the model in the training and test sets, as well as in the 10-fold cross validation.

	FPR	PRECISION	SENSITIVITY	GLOBAL ACCURACY
TRAINING	0.165	0.741	0.548	0.703
10-FOLD CV	0.170	0.733	0.544	0. 698
TEST EASY	0.268	0.617	0.793	0. 754
TEST MID-HARD	0.283	0.528	0.637	0. 691
TEST VERY-HARD	0.543	0.603	0.667	0. 573

*Performance measures are calculated for the positive class (interacting proteins). FPR: false positive rate, FPR = 1 – specificity.

Section SM5. PPI-Detect: a simple and user-friendly tool

PPI-Detect is implemented in Java (JDK version 1.7). The compiled version is available upon request. We also provide a user-friendly web application which requires the following input files:

File 1: A file with extension ".ppi" containing pairs of sequence labels separated by a semicolon (;). Given four sequences labeled A, B, C, D and grouped in two pairs, this file should contain the following information:

```
A;B
C;D
```

File 2: A FASTA file containing the sequences for proteins labeled A, B, C and D:

```
>A
SEQUENCE OF A...
>B
SEQUENCE OF B...
>C
SEQUENCE OF C...

>D
SEQUENCE OF D...
```

The outcome is provided in a text file with the instances listed in the same order as the list of pairs in the file *.ppi. The program can be executed from a command line interface, which facilitates the incorporation of the method within other codes implementing additional filters for the prediction of PPI.

Table SM6. Summary of values' range for all descriptors used to train the model.

DESCRIPTOR	MINIMUM	MAXIMUM	5^{TH} PERCENTILE	95^{TH} PERCENTILE
GS(U)_ES_PRT_TI50	-62.966	218.652	13.01	109.69
PA_ES_NPR_TI50	-58.474 .00	145.976	8.47 .47	85.84
ISA_ES_NPR_TI50	-53.282 .00	137.308	14.49 .28	88.39
ECI_ES_PRT_TI50	-55.647 .25	228.649	8.8	109.52
Z3_ES_BSR_TI50 *	-38.07	104.01	12.25	70.59
GW(U)_ES_PRT_TI50	-59.358	353.195	7.24	117.84
IP_ES_GLY_TI50 *	-9.659 .00	57.245	4.00	34.49
MW_ES_ILE_TI50 *	-5.51 .00	64.585	4.00	34.45
W(U)_ES_ALA_TI50 *	-9.059 .00	73.101	6.49	36.46
PT_ES_ARG_SI50 *	-3 .58	-0.931	-2.37	-1.75
W(U)_ES_PHE_SI50 *	-3 .58	-0.931	-2.48	-1.74
PA_ES_PHE_SI50 *	-3 .58	-0.931	-2.46	-1.83
PT_ES_PCR_SI50 *	-3	-1.47	-2.22	-1.76
IP_ES_GLY_SI50 *	-3	-1.333	-2.37	-1.76
AP_ES_PCR_TI50 *	-30.187	98.169	8.98	55.97
IP_ES_AHR_TI50	-83.319	135.287	4.05	72.52
W(U)_ES_CYS_SI50 *	-3	-1.242	-2.52	-1.80
MW_ES_ALR_MI50	-10.476	-2.95	-9.20	-5.62
PB_ES_MET_SI50 *	-3	-0.898	-2.52	-1.50

*The descriptor can contain missing values which are labeled as -9999 in the outcome file of ProtDCal. Missing values indicate that certain descriptor can not be evaluated, for instance the descriptor Pa_ES_PHE_SI50 requires that both sequences in the pair contain phenylalanine (Phe) residues.

Table SM7. Summary of EPI-X4 derivatives, their activities from the study of Zirafi *et al.* and predicted scores using PPI-Detect (score values higher than the one for EPI-X4 indicate higher activity and vice versa)

Derivative	*Activity**	*Class ID*	*F1 Score*	*F1 Class ID*	*F2 Score*	*F2 Class ID*	*F3 Score*	*F3 Class ID*	*F4 Score*	*F4 Class ID*
EPI-X4-408-423	**-**	**-**	**0.273**	**-**	**0.36**	**-**	**0.309**	**-**	**0.354**	**-**
415-423	low	0	0.380	1	0.435	1	0.452	1	0.420	1
414-423	low	0	0.343	1	0.372	1	0.417	1	0.405	1
413-423	low	0	0.244	0	0.272	0	0.323	1	0.287	0
412-423	low	0	0.245	0	0.234	0	0.299	0	0.280	0
411-423	low	0	0.253	0	0.260	0	0.269	0	0.256	0
410-423	low	0	0.251	0	0.288	0	0.276	0	0.259	0
409-423	low	0	0.259	0	0.34	0	0.293	0	0.290	0
408-422	low	0	0.271	0	0.306	0	0.311	1	0.343	0
408-421	low	0	0.285	1	0.331	0	0.341	1	0.354	0
408-420	low	0	0.276	1	0.319	0	0.323	1	0.315	0
408-419	high	1	0.256	0	0.339	0	0.317	1	0.292	0
408-418	low	0	0.288	1	0.372	1	0.316	1	0.364	1
408-417	low	0	0.296	1	0.327	0	0.258	0	0.331	0
408-415	low	0	0.236	0	0.273	0	0.305	0	0.307	0
408-414	low	0	0.223	0	0.246	0	0.237	0	0.258	0
408-413	low	0	0.218	0	0.29	0	0.279	0	0.285	0
408I-419	high	1	0.275	1	0.376	1	0.352	1	0.312	0
408F-419	low	0	0.316	1	0.285	0	0.264	0	0.263	0
408A-419	low	0	0.223	0	0.297	0	0.228	0	0.209	0
408G-419	low	0	0.304	1	0.360	0	0.315	1	0.330	0
408I-419R410H	low	0	0.361	1	0.359	0	0.381	1	0.359	1
408I-419R410K	low	0	0.344	1	0.344	0	0.317	1	0.321	0
408I-419Y411F	high	1	0.288	1	0.31	0	0.324	1	0.311	0
408I-419Y411S	low	0	0.283	1	0.356	0	0.333	1	0.335	0
408I-419Y411W	high	1	0.299	1	0.384	1	0.373	1	0.351	0
408I-419T412S	high	1	0.296	1	0.367	1	0.348	1	0.335	0
408I-419K414C	high	1	0.296	1	0.347	0	0.348	1	0.386	1

*The notation used in the first column is the same used by [8]. Numerical values of activity (IC50 values) are found in the Supplementary materials of [8].

Table SM7 (Cont.). Summary of EPI-X4 derivatives, their activities from the study of Zirafi *et al.* and predicted scores using PPI-Detect (score values higher than the one of EPI-X4 indicate higher activity and vice versa).

Derivative	*Activity **	*Class ID*	*F1 Score*	*F1 Class ID*	*F2 Score*	*F2 Class ID*	*F3 Score*	*F3 Class ID*	*F4 Score*	*F4 Class ID*
408I-419V418C	high	1	0.271	0	0.349	0	0.273	0	0.317	0
408I-418SC	high	1	0.300	1	0.389	1	0.330	1	0.315	0
408I-419WC01	high	1	0.305	1	0.412	1	0.386	1	0.330	0
408I-419WC02	high	1	0.299	1	0.342	0	0.326	1	0.32	0
408I-419WC03	high	1	0.296	1	0.361	1	0.364	1	0.308	0
408I-419WSC01	high	1	0.295	1	0.357	0	0.298	0	0.334	0
408I-419WSC02	high	1	0.310	1	0.378	1	0.350	1	0.308	0
408I-419WSC03	high	1	0.282	1	0.342	0	0.274	0	0.347	0

*The notation used in the first column is the same used by [8]. Numerical values of activity (IC50 values) are found in the Supplementary materials of [8].

To compute the performance measures presented in Table 2 of the main manuscript, the elements of the confusion matrix are defined according to Table S7.1.

Table S7.1. Definition of elements in the confusion matrix.

Element	**Definition**
TP	EPI-X4 derivatives with higher scores than the wild type EPI-X4, which also have higher affinity for CXCR4
TN	EPI-X4 derivatives with lower scores than the wild type EPI-X4, which also have lower affinity for CXCR4
FP	EPI-X4 derivatives with higher scores than the wild type EPI-X4, which have lower affinity for CXCR4
FN	EPI-X4 derivatives with lower scores than the wild type EPI-X4, which have higher affinity for CXCR4

Precision = TP / (TP + FP); Sensitivity = TP / (TP + FN); Specificity = TN / (TN + FP); Accuracy = (TP + TN) / (TP + FP + FN + TN).

References

1. Ruiz-Blanco, Y. B.; Paz, W.; Green, J.; Marrero-Ponce, Y. BMC Bioinformatics 2015, 16(1), 162.
2. Kyte, J.; Doolittle, R. F. Journal of molecular biology 1982, 157(1), 105-132.
3. Levitt, M. Biochemistry 1978, 17(20), 4277-4285.
4. Collantes, E. R.; Dunn, W. J., 3rd. Journal of medicinal chemistry 1995, 38(14), 2705-2713.
5. Shannon, C. E. Bell System Technical Journal 1948, 27(3), 379-423.
6. Hall, L. H.; Kier, L. B. Journal of Chemical Information and Computer Sciences 1995, 35(6), 1039-1045.
7. Hsu, C. W.; Chang, C. C.; Lin, C. J. A practical guide to support vector classification, 2003.
8. Zirafi, O.; Kim, K. A.; Standker, L.; Mohr, K. B.; Sauter, D.; Heigele, A.; Kluge, S. F.; Wiercinska, E.; Chudziak, D.; Richter, R.; Moepps, B.; Gierschik, P.; Vas, V.; Geiger, H.; Lamla, M.; Weil, T.; Burster, T.; Zgraja, A.; Daubeuf, F.; Frossard, N.; Hachet-Haas, M.; Heunisch, F.; Reichetzeder, C.; Galzi, J. L.; Perez-Castells, J.; Canales-Mayordomo, A.; Jimenez-Barbero, J.; Gimenez-Gallego, G.; Schneider, M.; Shorter, J.; Telenti, A.; Hocher, B.; Forssmann, W. G.; Bonig, H.; Kirchhoff, F.; Munch, J. Cell reports 2015, 11(5), 737-747.

6.2 Publication II

Author contributions

PPI-Affinity: A Web Tool for the Prediction and Optimization of Protein–Peptide and Protein–Protein Binding Affinity

Sandra Romero-Molina, Yasser B. Ruiz-Blanco, Joel Mieres-Perez, Mirja Harms, Jan Münch, Michael Ehrmann, Elsa Sanchez-Garcia

Research Article in:	**Journal of Proteome Research**
Impact factor:	5.37 (2021)
Published online:	June 2nd, 2022
DOI:	10.1021/acs.jproteome.2c00020
Citations:	3 (Dec 6th, 2022)

Complete citation from source:

Romero-Molina, S.; Ruiz-Blanco, Y. B.; Mieres-Perez, J.; Harms, M.; Münch, J.; Ehrmann, M.; Sanchez-Garcia, E. PPI-Affinity: A Web Tool for the Prediction and Optimization of Protein–Peptide and Protein-Protein Binding Affinity. *Journal of Proteome Research* 2022; 21(8): 1829–1841. https://doi.org/10.1021/acs.jproteome.2c00020

Contributions:

Conception	30%
Model creation	85%
Model validation	85%
Experimental assays	0%
Code implementations	95%
Web(tool) validation and deployment:	100%
Web(tool) maintenance:	100%
Manuscript writing	70%
Manuscript revision	60%

Journal of proteome research

pubs.acs.org/jpr

Article

PPI-Affinity: A Web Tool for the Prediction and Optimization of Protein−Peptide and Protein−Protein Binding Affinity

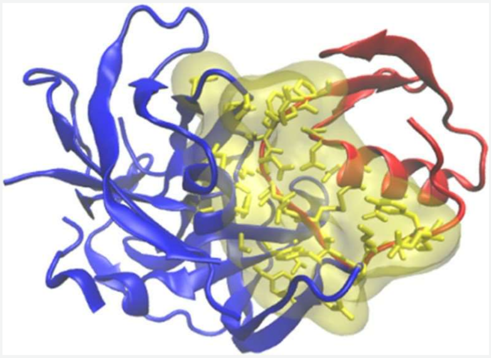
Sandra Romero-Molina, Yasser B. Ruiz-Blanco, Joel Mieres-Perez, Mirja Harms, Jan Münch, Michael Ehrmann, and Elsa Sanchez-Garcia*

Cite This: *J. Proteome Res.* 2022, 21, 1829−1841

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Virtual screening of protein−protein and protein−peptide interactions is a challenging task that directly impacts the processes of hit identification and hit-to-lead optimization in drug design projects involving peptide-based pharmaceuticals. Although several screening tools designed to predict the binding affinity of protein−protein complexes have been proposed, methods specifically developed to predict protein−peptide binding affinity are comparatively scarce. Frequently, predictors trained to score the affinity of small molecules are used for peptides indistinctively, despite the larger complexity and heterogeneity of interactions rendered by peptide binders. To address this issue, we introduce PPI-Affinity, a tool that leverages support vector machine (SVM) predictors of binding affinity to screen datasets of protein−protein and protein−peptide complexes, as well as to generate and rank mutants of a given structure. The performance of the SVM models was assessed on four benchmark datasets, which include protein−protein and protein−peptide binding affinity data. In addition, we evaluated our model on a set of mutants of EPI-X4, an endogenous peptide inhibitor of the chemokine receptor CXCR4, and on complexes of the serine proteases HTRA1 and HTRA3 with peptides. PPI-Affinity is freely accessible at https://protdcal.zmb.uni-due.de/PPIAffinity.



KEYWORDS: *machine learning, mutation, dissociation constant, peptide design, protein−protein interaction, binding free energy*

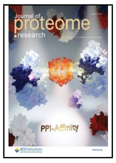
■ INTRODUCTION

Protein−protein interactions (PPIs) are fundamental to most biological processes.[1] Prominent disorders, such as cancer and degenerative diseases, are related to aberrant PPIs.[2] In therapy, optimized PPIs are also critical for the strong binding of antibodies to their protein antigens.[3] Therefore, the characterization of PPIs in terms of their binding affinity (BA) is highly relevant to the design of new biologics and therapeutic compounds.[4] Notably, peptides are a promising class of bioactive compounds, which often have higher specificity and reduced side effects compared to small-molecule pharmaceuticals.[5] Currently, there are more than 60 approved peptide drugs, and hundreds of peptidic compounds undergo clinical or preclinical trials.[6] However, the design of peptide drugs remains a challenging task due to their flexible structures and diversity of binding sites.[7]

Virtual screening approaches, based on BA predictions, reduce the time of the drug development pipelines.[3] Thus, over the past decades, several screening tools based on the BA of protein−protein complexes have been introduced.[8−26] Relevant examples of these methods are DFIRE,[9] CP_PIE,[16] ISLAND,[24] and the web server PRODIGY,[21,27] (which tailored and improved the original model introduced by Kastritis et al.).[20] Protein−peptide complexes are usually scored with functions derived from binding affinity data of small molecules. Examples of such scoring methods are RF-Score[28] and Kdeep,[29] which used a random forest algorithm[30] and a convolutional neural network,[31,32] respectively, to train their models. Supporting Information Table SI-1 summarizes the above-mentioned BA predictors, as well as other state-of-the-art methods that contribute to the broad field of PPI prediction tools.

Noteworthy, to the best of our knowledge, there is no publicly available web tool **specifically** designed to predict and optimize the BA of diverse protein−peptide complexes, by considering as a peptide an amino acid sequence with less than 30 residues. Several works have approached this aim specifically for the identification of binders of the major histocompatibility complexes MHC-I, II.[33,34] However, given that their training is restricted to MHC data, these models are not applicable to predict the binding free energy of other protein−peptide complexes. Besides, the available tools typically do not leverage the possibility of optimizing the

Received: January 11, 2022
Published: June 2, 2022



ACS Publications

https://doi.org/10.1021/acs.jproteome.2c00020

primary structure of the peptide to improve the affinity of the complex.

We evaluated the performance of the above-mentioned screening tools for estimating the BA of protein−peptide complexes by testing 100 randomly selected protein−peptide complexes from the Biolip[35] database. This test set contained complexes with peptides ranging from 4 to 29 amino acids, coupled to receptors with sizes between 51 and 496 amino acids. The binding free energy of the complexes covered the range between −12.6 and −4.6 kcal/mol. The highest correlation was delivered by Kdeep (R = 0.32), while the correlation with all of the other methods was in the range R = [0.13, 0.24] (Supporting Information Table SI-2). This comparison evidences the rather low dependability of state-of-the-art screening tools for the prediction of the BA of protein−peptide complexes.

Given the remarkable scaffold that peptides represent for drug development, we addressed this issue by developing machine-learning-based predictors of BA that are specific for protein−peptide complexes. In addition, we present a predictor of protein−protein BA that rivals the performance of the state-of-the-art screening tools built for such systems. Both predictors are integrated into a novel tool named PPI-Affinity, which is a web server designed to score protein−protein and protein−peptide complexes based on their predicted BA, as well as to optimize the affinity of a complex by mutating and screening selected residues. With such functionalities, PPI-Affinity can be employed at early steps of drug design processes, which are focused on the screening and optimization of protein/peptide binders for a given protein target.

MATERIALS AND METHODS

In this section, we describe the dataset and the modeling procedure used to develop both predictors. In both scenarios, protein−protein and protein−peptide systems, the performance of the models is evaluated with cross-validation and hold-out test sets.

Data Collection: Protein−Protein Complexes

We initially retrieved 2 852 protein−protein complexes with known BA data deposited in the PDBbind (v.2020)[36] database. Subsequently, we curated these data by extracting only dimeric complexes and removed those in common with the benchmark used by Vangone and Bonvin[21] to assess their model. We excluded those cases in which the binding affinity values were reported with measures other than K_d, K_i, or ΔG_{bind}. Likewise, those instances with imprecise BA values (*i.e.*, reporting ranges of K_d values instead of precise values) were removed. Cases with binding free energy values outside the range [−18.1, −3.1] kcal/mol were also excluded, as these instances are sparingly represented and the difference between their BA value and the rest of the distribution was $\Delta G \geq 0.5$. The application of such filters rendered a dataset of 833 protein−protein complexes (Supporting Information Figure SI-1 depicts the distribution of binding free energy values for this dataset). All of the structures were preprocessed by adding hydrogens and other missing atoms with MODELLER v9.23[37−40] and PDB2PQR.[41]

Features Generation

We employed the 3D-structure descriptors implemented in the protein codification package ProtDCal.[42] This program is accessible via the web server ProtDCal-Suite,[43] which also provides access to other models developed by us using this codification approach. ProtDCal has a workflow of four automated steps, which are: (i) residue-wise codification, (ii) modification based on the vicinity, (iii) residue grouping based on amino acid types, and (iv) numerical aggregation using descriptive statistics. The options selected at each step are used in a combinatorial scheme to generate an array of numerical descriptors that characterize the input structure. Each descriptor in the array is the result of the combination of one residue property (e.g., reside-wise contact order, RWCO),[44] a vicinity operator (e.g., autocorrelation, AC), a grouping criterion (e.g., nonpolar residues, NPR), and an aggregation operator (e.g., variance, V); thus, every element in the array is identified by a unique label (e.g., RWCO_AC_NPR_V). In the Supporting Information Section SI-1, we provide the configuration file used to compute the descriptors employed in this study. This configuration generates 23 040 descriptors for each protein−protein complex. In ProtDCal, interchain residue contacts are determined by a maximum spatial distance (d) measured between the α carbons of the residues. We set up the calculation of such contacts with a spatial cutoff d = 10 Å. ProtDCal has been successfully applied by us and other authors to model post-translational modifications,[42,45,46] protein−protein interaction,[47] enzyme-like amino acid sequences,[48] critical residues for protein function,[49] and antibacterial peptides.[50,51]

Modeling Protocol for Protein−Protein Affinity Data

The learning process was carried out with Weka 3.8.4.[52] Support Vector Machine (SVM) was selected as the learning algorithm after performing a preliminary study that showed that it is the simplest and best-performing solution for developing the models (Supporting Information Section SI-2). SVM is a successful approach widely validated in drug discovery.[53,54] This technique has delivered a worthy performance on small and mid-sized datasets,[55] where deeper machine learning (ML) approaches, such as various neural network architectures, tend to overfit.[56,57] The implementation of SVM for regression in the package SMOreg[58] was employed to develop the models. We randomly split the collected data into three datasets: a training set with 653 complexes, a development set with 90 complexes, and a test set with 90 complexes (Script SI-1). The purpose of the development set was to monitor the generalization of multiple configurations of the hyperparameters during the training process. The hold-out test set was exclusively used to compare the final model with external predictors.

Initially, every instance in the training set was represented as a vector of 23 040 molecular descriptors generated with ProtDCal. The steps of feature selection included the removal of invariant descriptors and those carrying missing values. Next, an optimal subset of attributes was searched with a supervised exploration guided by the error in the prediction of the binding affinity for cases in the training set. The resulting optimal subset comprised 26 descriptors, which were used to train the final models (Supporting Information Section SI-3). Supporting Information File SI-1 provides a configuration file for ProtDCal to compute this specific list of descriptors.

We adopted an ensemble learning approach by creating four partially overlapping subsets of the training data with a distribution of the output variable flatter than the complete training set (Supporting Information Figure SI-2). Thus, we

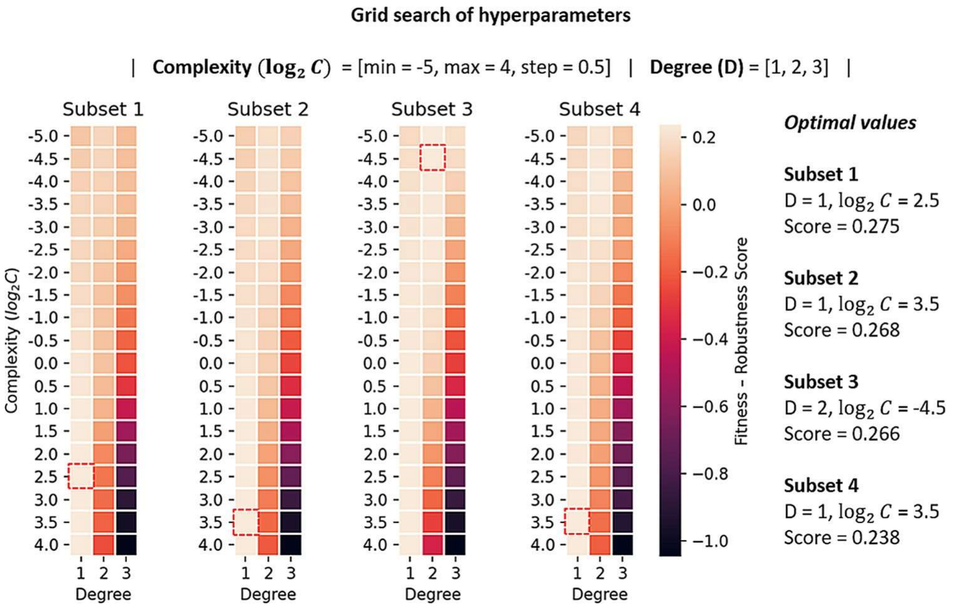


Figure 1. Results of the optimization of hyperparameters. The selected model per training subset is marked with a red square and corresponds to the combination of degree (D) and complexity ($\log_2 C$) values that maximized the fitness-robustness score defined in eq 2.

trained independent models with each subset and combined their outcomes using the Vote method implemented in Weka 3.8.4,[52,59,60] with combination schemes based on the average, maximum, and minimum predicted values among selected models.

With each training subset, the hyperparameters of the SVM were adjusted using a grid search.[61] The search space was defined by the following range of hyperparameter values:

- Complexity (C): 2^{-5}, $2^{-4.5}$, 2^{-4}, $2^{-3.5}$, 2^{-3},..., 2^{1}, $2^{1.5}$, 2^{2}, $2^{2.5}$, 2^{3}, $2^{3.5}$ and 2^{4}
- Degree (D) of the polynomial function kernel: 1, 2, and 3

The models generated during the search were assessed using the Pearson's correlation coefficient (R) (eq 1) of the estimations in the training data via resubstitution and 10-fold cross-validation (10-fold CV), as well as in the development set.

$$R = \frac{\sum_{i=1}^{n} (y_i - \bar{y})(y_i^{\text{pred}} - \bar{y}^{\text{pred}})}{\sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2 \sum_{i=1}^{n} (y_i^{\text{pred}} - \bar{y}^{\text{pred}})^2}} \quad (1)$$

The terms y_i and $\bar{y}$ are the actual affinity values and their mean in the datasets. Analogously, the terms y_i^{pred} and $\bar{y}^{\text{pred}}$ correspond to the same type of values but as predicted by the model.

To identify the optimum values of the hyperparameters, we formulated an *ad hoc* fitness-robustness score (FRS) (eq 2) as a function of the correlation coefficient, which combines in a single measure the performance of a model in terms of goodness of fit and robustness. The FRS function has an optimum maximum value at FRS = 1; thus, we selected the configuration that maximized the FRS function.

$$\text{FRS} = (\bar{R})^2 - ((R_{\text{CV}} - R_{\text{TS}})^2 + (R_{\text{DEV}} - R_{\text{TS}})^2) \quad (2)$$

The terms R_{TS}, R_{CV}, and R_{DEV} are the correlation coefficients obtained on the training set, in 10-fold CV, and development set, respectively. $\bar{R}$ is the arithmetic mean of the performance on these three tests. The first term of the function aims to combine the performance on the training set (goodness of fit), with the performance in cross-validation and development set (generalization). The next two terms reduce the deviations between the performance on the training set and the performance when evaluating in cross-validation and development set. These weighting terms improve the robustness of the selected model by considering the generalization power of the predictor in addition to the goodness of fit. Overall, such a unified quantity is a highly informative measure to guide the optimization of hyperparameters during the modeling. We intend to continue challenging this formulation in further studies and applications. Figure 1 summarizes the results of the optimization of the hyperparameter values in each training subset. A complete list of all of the intermediate models and performance measures is provided in Supporting Information Table SI-3.

All possible combinations with at least two models were evaluated in the development set to determine the best ensemble model. We summarized all of the performance measures for the independent models and the distinct ensembles in Supporting Information Table SI-4.

Data Collection: Protein−Peptide Complexes

The model was developed with data extracted from the Biolip database, which also incorporates data from the PDBbind Protein-Ligand[36] database. We downloaded the nonredundant dataset of Biolip, containing 105 152 entries. Only protein−peptide complexes with less than 90% of identity between the binding site's residues and the full receptor sequence are included in these data. We extracted the complexes containing single-chain receptor only and a peptide formed by standard residues with a minimum length of three residues. Instances reported with post-translational modifications or fusion constructs (peptide/nonpeptide) were discarded. Subsequently, we selected the cases for which their BA values are reported in terms of the dissociation (K_d) or inhibition (K_i) constants only. We excluded the complexes with ambiguous K_d or K_i values (*i.e.*, reporting a range of values), and those with

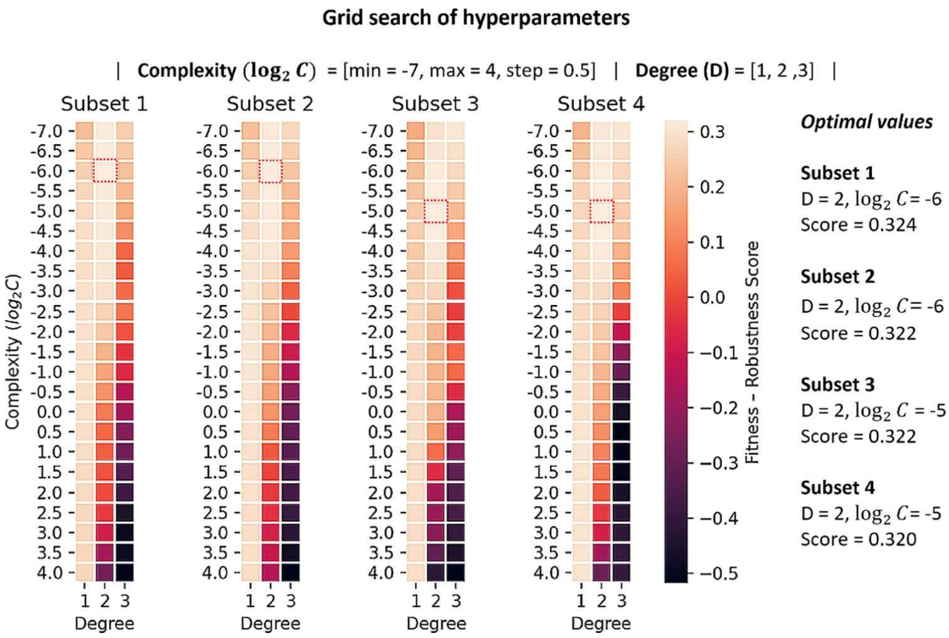


Figure 2. Results of the optimization of hyperparameters. The selected model per training subset is marked with a red square and corresponds to the combination of degree (D) and complexity ($\log_2 C$) values that maximized the FRS.

binding free energies outside the range from −14.4 to −3.6 kcal/mol, as these instances were poorly represented and the difference between their BA value and the rest of the distribution was $\Delta G \geq 0.5$. The curated dataset contained 1149 complexes, with peptides of length ranging between 3 and 29 amino acids and receptors of sizes between 31 and 957 amino acids (Supporting Information Figure SI-3). Hydrogen atoms and other missing atoms were added to the structures using MODELLER v9.23[37−40] and PDB2PQR.[41]

Modeling Protocol for the Protein−Peptide BA Data

We used SVM, implemented in the package SMOreg[58] in Weka 3.8.4, following a protocol equivalent to that employed for the protein-protein model described above. In summary, we randomly divided the dataset into three subsets: training dataset (949 instances), development dataset (100 instances), and test dataset (100 instances) (Script SI-1). Subsequently, we extracted numerical descriptors from the complexes' structure using ProtDCal[42,43] with the configuration file summarized in Supporting Information Section SI-1. This step generated 23 040 molecular descriptors for each instance in the dataset. Next, we reduced this large multidimensional space to 37 descriptors through a features selection process (Supporting Information Section SI-4; a list of the extracted set of descriptors can be found in the file SI-2).

The training scheme used to develop the final predictor followed an ensemble approach. Four individual models were built with partially overlapping subsets of the training set. The subsets are random samples of the training data with a flatter distribution of the BA values (Supporting Information Figure SI-4) compared to the entire training set. This is achieved by undersampling the region next to the mode value of the distribution, while keeping the tails. Such transformation in the distribution of the data allows a balanced error weight along the entire range of the response variable. Subsequently, we optimized the hyperparameters of each model independently and combined their predictions with the Vote method implemented in Weka 3.8.4, according to the average, maximum, and minimum ensemble rules.

To create the models, the hyperparameters of the estimator were adjusted in a grid search following the same methodology as for the protein−protein model. During the optimization of the hyperparameters, the performance of the models was monitored in the training data, in 10-fold CV, and the development set. The optimum set of hyperparameter values for each training dataset was selected according to the FRS defined in eq 2. Figure 2 shows the results of the hyperparameters tuning process for each training subset. A complete list of all of the intermediate models and performance measures can be found in Supporting Information Table SI-5.

Next, the best ensemble model was selected by evaluating all possible combinations of models on the development set (Supporting Information Table SI-6). The selected ensemble model was evaluated on the hold-out test set of 100 complexes and compared with other available state-of-the-art protein−protein and protein−ligand BA predictors. Finally, the ranking power of the model was assessed on a set of mutants of the peptide EPI-X4, an endogenous inhibitor of CXCR4 whose activity was experimentally determined and on a set of complexes between peptides and the serine proteases HTRA1 or HTRA3.

■ RESULTS AND DISCUSSION

In this section, we analyze the performance of the developed models and compare them with other state-of-the-art predictors.

Performance of the Protein−Protein Model

The best ensemble model contains two out of the four training subsets, whose individual correlations are $R_2 = R_3 = 0.50$. The ensemble model improved the individual ones achieving a correlation of $R = 0.53$ on the development set. The rule of minimum predicted value was used to build the ensemble predictor.

Recently, Vangone and Bonvin[21] developed a model that predicts the BA based on two structural descriptors: the network of inter-residue contacts (ICs) and the noninteracting surface (NIS). The model, named ICs/NIS-based predictor was implemented in the web server PRODIGY.[27] This tool delivered a much better performance (R = 0.74 and MAE = 1.4) than other state-of-the-art methods on a benchmark set of 79 protein−protein complexes. Thus, we employed this benchmark set to evaluate our method (Table 1, Test set 1) against the ICs/NIS-based model, the other two top-ranked and currently available tools in that assessment, and the ISLAND method. We estimated the performance of our ensemble model on this benchmark set and obtained a correlation coefficient R = 0.62 and MAE = 1.8 kcal/mol (Test set 1) that ranks our method second, after PRODIGY.

Table 1. Summary of the Evaluation of PPI-Affinity and State-of-the-Art BA Predictors on Two Sets of Protein−Protein Affinity Data[a]

	test set 1		test set 2	
method	R	MAE (kcal/mol)	R	MAE (kcal/mol)
PRODIGY	0.74	1.4	0.31	2.5
DFIRE	0.60	4.6	0.10	25.4
CP_PIE	−0.52	8.8	−0.10	11.0
ISLAND	0.38	2.1	0.27	2.2
PPI-Affinity	0.62	1.8	0.50	1.8

[a] The performance is expressed as the Pearson's correlation coefficient (R) between experimental and predicted BA. The test set 1 corresponds to the benchmark employed by Vangone and Bonvin,[21] while the test set 2 corresponds to the hold-out set of 90 data points taken from PDBbind (v.2020). The performance for the other methods on test set 1 was reported by Vangone and Bonvin.[21] The negative values of the correlation coefficients indicate that the corresponding method predicts unbinding free energy.

Next, we challenged the predictors with a larger hold-out test set of 90 complexes, which was initially extracted from the data collected from the PDBbind (v.2020) protein−protein dataset (Table 1, test set 2). In this test, our model renders a correlation coefficient of R = 0.50, with an error MAE = 1.8 kcal/mol (Test set 2), performance which is only marginally inferior to that obtained in the benchmark set of Vangone and Bonvin[21] (R = 0.62; MAE = 1.8 kcal/mol). The performance of ISLAND was diminished with respect to our method. Nevertheless, ISLAND delivered consistent results, in general superior to those delivered by other methods, in both test sets. The other predictors (PRODIGY, DFIRE, CP_PIE) show a large decrease in their performance with respect to their results in the test set 1 (Table 1, test set 1), with PRODIGY being the second best with a correlation coefficient R = 0.31 and MAE = 2.5 kcal/mol. Such a dramatic decay in the predictions suggests the presence of overfitting toward the previous benchmark set, specifically in the case of PRODIGY. Nonetheless, the analysis of the performance of the other predictors is hindered by the lack of applicability domain (AD) definition for using the methods. The absence of a defined AD limits the analysis of the errors of the predictions, as it is not possible to examine whether test samples are simply outside the scope of these predictors or there is a specific structural issue that affects the quality of the prediction.

In short, the data evidence the generalization achieved by our predictor, which performs consistently well in different external test sets. Supporting Information Figure SI-5 displays plots of experimental versus predicted BA values for PPI-Affinity in both test sets.

Next, we evaluated the performance of PPI-Affinity on a set of mutants taken from the SKEMPI v2.0[62] dataset (Figure 3).

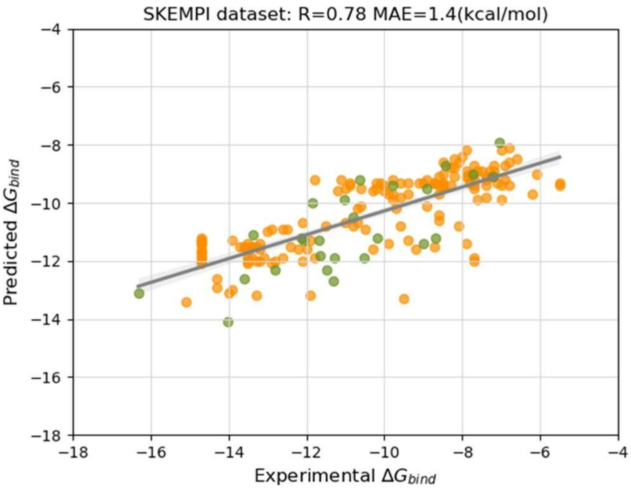


Figure 3. Performance of PPI-Affinity on the test set of 26 wild-type complexes and 151 mutants of protein−protein affinity data points taken from the SKEMPI dataset. The performance is reported as the Pearson's correlation coefficient (R) between experimental and predicted BA. The green points correspond to BA values of the wild-type complexes, and the orange points correspond to the BA values of the mutants.

This database reports the binding affinity changes of 7085 mutations of 345 protein−protein interactions for which the structure of the complex has been resolved. We selected a subset from this dataset by applying the following filtering steps: (I) we extracted the dimeric complexes having at least 30 amino acids in each protein sequence; (II) we removed the complexes overlapping between the selected data, the PDBbind (v.2020) dataset, and the benchmark set of Vangone and Bonvin;[21] and (III) we removed the wild-type systems with more than one binding affinity value reported, as well as all mutants with ambiguous or unreported binding affinities. The output of the filtering steps reduced the dataset to 34 wild-type complexes and 182 mutants. We fed the conformed test set to PPI-Affinity. Eight wild-type complexes and their related mutants were found outside the applicability domain of the model. Thus, the final test set contained 26 wild-type structures and 151 mutants. The assessed mutants featured between one and six mutations per protein sequence, with 80% of the structures accounting for only one mutation. The binding free energy of all of the complexes was in the range of −16.3 to −5.5 kcal/mol.

The performance of PPI-Affinity (R = 0.78 and MAE = 1.4 kcal/mol) on the SKEMPI dataset is superior to that obtained in the Vangone and Bonvin benchmark[21] (Table 1, Test set 1) and in the 90 protein−protein complexes taken from the PDBbind (v.2020) database (Table 1, Test set 2). When considering only the wild-type complexes, PPI-Affinity delivered a performance of R = 0.77 and MAE = 1.1 kcal/mol. These results evidence the robustness of our protein−protein model in a third external dataset, at the time that show the ability of PPI-Affinity to characterize the changes upon mutations of protein−protein complexes.

Additionally, we assessed the performance of PPI-Affinity against the LUPIA[26] classifier. LUPIA uses a threshold value to classify as “high” or “low” the binding affinity of protein−protein complexes. This evaluation (Supporting Information Section SI-5) evidenced the shortcomings of discretizing the BA values to train classifiers rather than regressors models. Taken together, the performance of our protein−protein model on several tests ranks our ensemble model on the top of state-of-the-art fast protein−protein BA predictors.

Performance of the Protein−Peptide Model

The best ensemble model was obtained by two out of the four training subsets, improving their performances ($R_1 = 0.53$, $R_3 = 0.54$) to $R = 0.56$ on the development set. The rule of maximum value was used to build the ensemble model. The model was assessed on the test set of 100 protein−peptide complexes initially hold-out from the data extracted from the Biolip database (Supporting Information Figure SI-6). Here, we compared the results of our method with state-of-the-art protein−protein and protein−ligand BA predictors. The considered tools include Kdeep and RF-Score for protein−ligand complexes, as well as the above-presented PRODIGY, DFIRE, and CP_PIE methods. ISLAND requires a minimum size of 20 amino acids for each protein sequence. For this reason, the tool was applied to the assessment of only the protein−protein model.

Our protein−peptide affinity model outperformed all the other tools, showing a correlation coefficient of $R = 0.55$ with MAE = 1.1 kcal/mol (Table 2). The low correlations values delivered by other state-of-the-art tools can be related to the fact that the functions of Kdeep and RF-Score are fitted primarily using small organic ligands.[63,64] This suggests that fitting with mostly small ligand data cannot capture the differences imposed by the larger size of the peptides, as well as the diversity of peptide interactions with the target and the solvent.

Table 2. Correlation Coefficient (R) of Protein−Protein and Protein−Ligand BA Predictors on the Test Set of 100 Protein−Peptide Complexes[a]

method	R	MAE (kcal/mol)
PRODIGY	0.13	1.9
DFIRE	0.29	8.7
CP_PIE	−0.28	9.0
Kdeep*	0.32	10.7
RF-Score*	0.23	1.8
PPI-Affinity	0.55	1.1

[a] Protein−ligand methods are marked with a star. The negative values of the correlation coefficients indicate that the corresponding method predicts unbinding free energy.

Case Study I: Ranking the Affinity of Mutants of EPI-X4

EPI-X4 (Endogenous Peptide Inhibitor of CXCR4) is a fragment of albumin identified as an endogenous antagonist of the CXC chemokine receptor 4 (CXCR4) by Zirafi et al.[65] Given the implication of CXCR4 in viral (HIV) infection, inflammation and cancer,[66,67] this peptide represents a highly promising scaffold to develop therapeutic drugs targeting the CXCR4 receptor.

Recently, mutants of EPI-X4 have been screened to identify derivatives with enhanced stability and affinity. For this, the affinity values (in terms of IC_{50} nanomolar, nM) of EPI-X4 and 56 derivatives to CXCR4 were estimated using an antibody competition assay.[67] The scheme of this assay is based on the competitive binding of a fluorescently labeled anti-CXCR4 antibody (clone 12G5) with CXCR4 ligands (Supporting Information Section SI-6).[67] These derivatives are peptides with size in the range of 6−16 amino acids. From the experiments, 26 mutants out of 30 active (IC_{50} < 10 000nM) derivatives were found to be more active compared to EPI-X4. Here, we employed these data to evaluate the ranking power of our protein−peptide affinity model. For doing this, we use the enrichment factor (EF_I),[68] defined as (eq 3):

$$EF_I = \frac{[N_{\text{active}}^{\text{top}_i} / I]}{[N_{\text{active}}/N_{\text{total}}]} \tag{3}$$

where I represents a fixed number of top-ranked instances based on predicted values; $N_{\text{active}}^{\text{top}_i}$ is the number of active peptides, within the top I instances of the dataset; N_{active} is the number of *active* peptides; and N_{total} is the total number of peptides in the whole dataset. Peptides with IC_{50} < 10 000 nM were defined as active for an initial test. In a second and more stringent evaluation, the *active* peptides were considered as those more active than EPI-X4.

The structures of the complexes, formed by CXCR4 and each peptide mutant, were generated via homology modeling

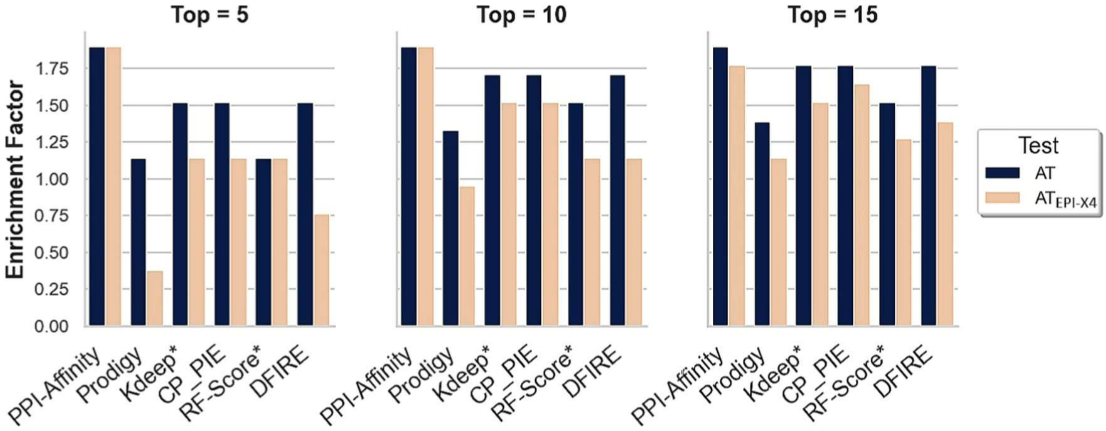


Figure 4. Performance of protein−protein and protein−ligand BA predictors on the set of 56 derivatives of EPI-X4. The performance of the methods is based on the enrichment factor (EF) obtained among the top 5, 10, and 15 ranked candidates. Two results are shown per tool, one corresponding to the activity test (AT), and the second corresponding to the peptides with an affinity higher than EPI-X4 (AT_{EPI-X4}).

using as a template the structural model of the complex CXCR4/EPI-X4, reported by Sokkar et al.[69,70] We estimated the BA of each complex with different predictors, as well as the associated enrichment factors.

Figure 4 shows the results of the tests corresponding to (1) the conventional activity test using the activity threshold determined by the competition assay (IC_{50} = 10 000 nM) and (2) the peptides with an affinity higher than EPI-X4. PPI-Affinity achieved the maximum enrichment $EF_5 = EF_{10} = EF_{15}$ = 1.9 for this test, *i.e.*, the 15 top-ranked instances according to the PPI-Affinity estimation are active. In the case of the stringent test, taking only those with higher affinity than EPI-X4 as active, the maximum EF was obtained within the top 10 peptides, i.e., $EF_5 = EF_{10}$ = 1.9, while the enrichment within the top 15 derivatives also reached a high value EF_{15} = 1.8. The high enrichment factor within the top 5, 10, and 15 peptides, representing more than 25% of the dataset, evidenced the remarkable ranking capabilities of PPI-Affinity. The other tools show moderate to high enrichment values although below the level reached by PPI-Affinity (Figure 4). Noteworthy are CP_PIE and Kdeep, which deliver a steady high performance in the different tests.

Case Study II: Ranking the Affinity of Peptides for the PDZ Domains of HtrAs

High-temperature requirement serine proteases (HtrAs) are involved in many physiological processes and neurodegenerative diseases such as Alzheimer's disease and CARASIL.[71] These proteases are largely regulated by an allosteric mechanism whose initial step is the interaction of polypeptide chains with the peripheric PDZ domain. Here, we challenged PPI-Affinity with a series of peptides bound to PDZ domains of two human HtrAs: HTRA1 and HTRA3.

For these sets of protein−peptide interactions, we compared the relative ranking based on the binding affinity predicted by PPI-Affinity to the ranking based on the experimentally determined IC_{50} values[72] (Tables 3 and 4).

The prediction of PPI-Affinity suggests that three of the peptides have more affinity for the PDZ domain of HTRA1 than the experimentally determined best binder DSRIWWV (in bold, Table 3). However, the calculated binding affinities of those peptides differ from that of DSRIWWV by only 0.1, 0.4, and 0.7 kcal/mol (for WDKIWHV, GWKTWIL, and DSAIWWV, respectively). These differences are very small and within the MAE of PPI-Affinity (Table 2).

Table 3. Ranking of BA of Protein−Peptide Interactions in HTRA1 as Predicted by PPI-Affinity and Based on the Experimental IC_{50} Values[c]

PPI-Affinity		experimental	
ranking	ΔG (kcal/mol)	ranking	(IC50 (μM))[a]
(1) DSAIWWV	−8.8	(1) **DSRIWWV**	0.9 ± 0.1
(2) GWKTWIL	−8.5	(2) DARIWWV	1.3 ± 0.1
(3) WDKIWHV	−8.2	(3) DSAIWWV	2.5 ± 0.4
(4) **DSRIWWV**	−8.1	(4) WDKIWHV	2.8 ± 0.3
(5) DARIWWV	−8.0	(5) ASRIWWV	2.8 ± 0.3
(6) ASRIWWV	−8.0	(6) DSRIWWA	3.5 ± 0.9
(7) DIETWLL	−7.8	(7) DSRIWAV	6 ± 1
(8) DSRIWWA	−7.3	(8) GWKTWIL	7.7 ± 0.6
(9) DSRAWWV	−7.3	(9) DSRAWWV	13 ± 1
(10) DSRIWAV	−7.2	(10) DIGPVCFL	16 ± 3
(11) DIGPVCFL[b]	−7.1	(11) DIETWLL	23 ± 3
(12) EVKIMVV[b]	−7.0	(12) EVKIMVV	24 ± 8
(13) DSRIAWV	−6.9	(13) DSRIAWV	40 ± 5

[a]Values taken from ref 72. [b]Protein−peptide structures that are at the border of the applicability domain of PPI-Affinity. [c]No protein−peptide structure is outside the applicability domain.

Table 4. Ranking of BA of Protein−Peptide Interactions in HTRA3 as Predicted by PPI-Affinity and Based on the Experimental IC_{50} Values

PPI-Affinity		experimental	
ranking	ΔG (kcal/mol)	ranking	(IC50 (μM))[a]
(1) FGAWV[c]	−7.7	(1) **FGRWV**	0.6 ± 0.1
(2) **FGRWV**[b]	−7.5	(2) RSWWV	0.6 ± 0.1
(3) FGRWI[c]	−7.5	(3) FGAWV[b]	0.9 ± 0.1
(4) RSWWV	−7.4	(4) FGRWI[b]	1.0 ± 0.1
(5) FGRWF[c]	−7.3	(5) GRWV	1.0 ± 0.1
(6) GRWV[b]	−7.2	(6) FARWV[b]	1.1 ± 0.2
(7) FGRWA[b]	−7.1	(7) RWV	1.3 ± 0.1
(8) FGRAV[b]	−6.9	(8) FGRWL	2.9 ± 0.3
(9) FGRWL[b]	−6.6	(9) FGRWA	3.5 ± 0.3
(10) WV[c]	−6.6	(10) WV[b]	4.7 ± 0.4
(11) WA[c]	−6.5	(11) FGRWF[b]	7.7 ± 0.8
(12) FARWV[c]	−6.4	(12) WA[b]	14 ± 1
(13) WG[c]	−6.2	(13) WG[b]	22 ± 3
(14) RWV[b]	−5.2	(14) FGRAV	270 ± 110

[a]Values from ref 72. [b]Protein−peptide structures are at the border of the applicability domain of PPI-Affinity. [c]Protein−peptide structures that are outside the applicability domain of PPI-Affinity.

The rest of the peptides are correctly predicted by the method as weaker binders than DSRIWWV to the PDZ domain of HTRA1. We note, however, that the binding energy differences for most of these peptides also fall within the MAE reported for our method based on the test set of protein−peptide affinity (Table 3).

Interestingly, even the two peptides (DIGPVCFL and EVKIMVV) at the border of the applicability domain of our method are correctly predicted as weaker binders to the PDZ domain of HTRA1 compared to DSRIWWV. Nevertheless, the predicted values for peptides that are outside the applicability domain of PPI-Affinity should be considered with care. The K_d of the complex between HTRA1 and DSRIWWV is experimentally reported as 1.3 ± 0.2 μM, which corresponds to $\Delta G = -8.2 \pm 0.1$ kcal/mol at the experimental temperature of 301.15 K. This value is in excellent agreement with the value of −8.1 kcal/mol predicted by PPI-Affinity (Table 3).

We also tested our method on peptides binding to the PDZ domain of HTRA3 (Table 4). In this set of protein−peptide complexes, eight systems are outside the applicability domain of PPI-Affinity.

As shown in Table 4, only one peptide (FGAWV) is incorrectly predicted to have better BA for HTRA3 than the best experimental binder FGRWV. We note that FGRWV lies outside the applicability domain of the model, and this prediction should be taken with caution.

In addition, we measured the overall ranking power of PPI-Affinity by calculating Kendall's correlation coefficient[73] on both test sets, HTRA1 and HTRA3 protein−peptide complexes (Table 5). The results obtained with PRODIGY, Kdeep, RF-Score, DFIRE, and CP_PIE are also shown for comparison. The binding affinity values delivered by each tool are listed in Supporting Information Section SI-7.

Table 5. Correlation of IC$_{50}$ Values with the Estimations from PPI-Affinity and Other State-of-the-Art BA Predictors on the Sets of HTRA1's and HTRA3's PDZ Binders

	R		*τ*	
method	HTRA1	HTRA3	HTRA1	HTRA3
PRODIGY	0.25	**0.29**	0.18	0.01
DFIRE	0.59	0.24	0.38	0.15
CP_PIE	−0.11	−0.03	−0.25	−0.08
Kdeep*	0.38	0.11	0.16	0.41
RF-Score*	0.56	0.13	0.56	0.27
PPI-Affinity	**0.63**	0.02	**0.59**	**0.42**

Kendall's tau coefficient is a robust nonparametric measure highly suitable to discriminate correlated from uncorrelated variables. This measure involves two main magnitudes: the number of concordant and discordant pairs. Two observations (x_i, y_i) and (x_j, y_j) are classified as a concordant pair if $x_i > x_j$, and $y_i > y_j$, or vice versa $x_i < x_j$ and $y_i < y_j$. If none of these conditions are true, the pair is known as discordant.

The Kendall's tau correlation coefficient (τ) is defined as (eq 4)

$$\tau = \frac{N_c - N_d}{\sqrt{(N_c + N_d + N_t)*(N_c + N_d + N_u)}} \quad (4)$$

where N_c and N_d are the number of concordant and discordant pairs, respectively; N_t is the number of ties in the order of the binding affinity determined by PPI-Affinity; and N_u is the number of ties in the experimental binding affinity. The Kendall's correlation coefficient is equal to 1 if the calculated ranking of the peptides completely agrees with the ranking determined experimentally.

Importantly, here we correlate IC$_{50}$ values with predicted binding free energy changes; these magnitudes are expected to be linked by a nonlinear relation, which then violates the linearity requirement for the proper interpretation of Pearson's correlation coefficient. Besides, the small size of the data induces violations of the bivariate normality requisite of this coefficient. Consequently, the use of a robust nonparametric measure such as Kendall's tau coefficient is a requirement to achieve a correct interpretation of the correlation tests.

The Pearson's correlation coefficient obtained by PPI-Affinity ($R = 0.02$) in the HTRA3 test set evidences the lack of correlation with the experimentally measured IC$_{50}$ values. Beyond the previously noted shortcomings of Pearson's coefficient to assess these data, this result can be also a consequence of half of the peptides in the test set being outside the applicability domain of PPI-Affinity. For HTRA1 binders, where all cases are within the applicability domain, PPI-Affinity delivers the highest correlation coefficient value ($R = 0.63$).

In terms of Kendall's tau coefficient, PPI-Affinity produces the highest values for both targets ($\tau = 0.59$ and $\tau = 0.42$ for HTRA1 and HTRA3, respectively), evidencing its better ranking power compared to other state-of-the-art predictors. Interestingly, RF-Score gives the second-best performance on the HTRA1 test set with $\tau = 0.56$, which is diminished by about half on the HTRA3 test set ($\tau = 0.27$). Conversely, Kdeep's performance is close to PPI-Affinity's on the HTRA3 test set ($\tau = 0.41$), but Kdeep delivers the lowest positive Kendall's tau value on the HTRA1 test set ($\tau = 0.16$). The weak performance of PRODIGY on both test sets might be related to the limitations of its protein−protein affinity predictor to evaluate smaller protein−peptide complexes. An important advantage of PPI-Affinity is that, thanks to two tailored models for protein−protein and protein−peptide complexes, PPI-Affinity can deliver comparable performance levels for both types of biomolecular systems.

In short, the evaluation of BA predictors on different tests and test sets evidenced that state-of-the-art BA predictors, either intended for protein−protein or protein−ligand complexes, deliver low accuracy in the prediction of the BA of protein−peptide complexes. Although it improves the accuracy of the state-of-the-art approaches only slightly, PPI-Affinity is a solution to that issue since it delivers a significantly better and robust performance among different tests. This fact

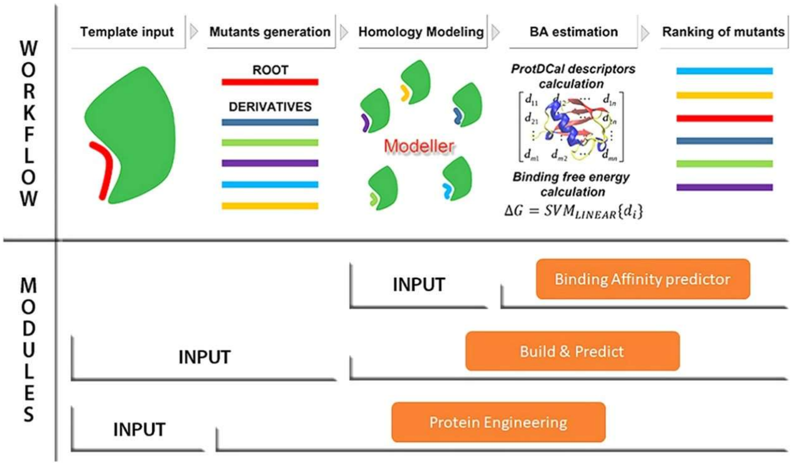


Figure 5. Workflow of the three modules included in PPI-Affinity. For each module, the required input data and associated steps are indicated. The main difference among them lies in the input data: the *Binding Affinity predictor* module receives as input a set of protein−protein/peptide complexes (in PDB format); the *Build & Predict* module generates the complexes from a template file and a list of amino acid sequences (in FASTA format) provided by the user; and the *Protein Engineering* module receives as input only a template model, generates a list of derivatives for one of the protein/peptide contained in the PDB file, and calculates homology models for all created mutants. Regardless of the module, once the PDB files have been prepared, PPI-Affinity computes the structural descriptors with ProtDCal, estimates the BA values with the machine learning models, and returns the list of derivatives ranked by their BA values.

tackles the inherent issues in generalization and overfitting that apparently affect other predictors. The shortcomings in the accuracy of predictions of absolute binding free energy values are to a large extent a consequence of the inherent deviations in the experimental data available for training, which arise from uneven standards under the experimental conditions and measurement procedures. Nonetheless, within such a noisy scenario, the support vector machine models developed by us allow making predictions that improve to some extent the state of the art at the same time of showing steady performance in both absolute binding affinity prediction and ranking assessments.

Web Server Implementation

We implemented PPI-Affinity as a web server to facilitate the use of our models. PPI-Affinity is a suite organized in three modules, which are summarized below by order of increasing complexity of their functionality.

Binding Affinity Predictor. This method is the direct application of the prediction models in a set of PDB files with protein−protein and protein−peptide complexes that should be provided by the users. The module has the functionality to characterize diverse complexes, whose structures were obtained from external sources, based on their binding affinity.

Build & Predict. This module follows the same purpose as the previous one. However, instead of receiving coordinate files with the complexes of interest, it only requires a template file (in PDB format) and a list of amino acid sequences (in FASTA format). Then, the server builds five structural homology models for each sequence in the list, using MODELLER, and selects the structure with the lowest DOPE as the best candidate to represent the complex. Subsequently, it scores all the complexes using the predictor of binding affinity. The Build & Predict module is particularly suitable for screening structurally similar complexes for which no structure has been elucidated.

Protein Engineering. This third and most comprehensive module allows for the automatic generation and scoring of mutations at the interface of the complexes, which aims to optimize the affinity of these complexes. Figure 5 depicts the workflow of the module, which encompasses the next steps:

Step 1: Template Input. The user provides an input structure of the complex (in PDB format) and optionally the amino acid sequences of the chains in FASTA format. In addition, the user must specify which chain will be optimized.

Step 2: Mutants Generation. Next, a maximum of 10 000 derivatives is constructed for the specified chain. The generation of mutants is controlled by the following parameters: the maximum number of modifications to the reference sequence, deletion and mutation probabilities, maximum molecular weight, type of mutations, and a mutability vector whose elements take value 0 for residues that remain unmodified and a value between 0 and 1 representing the probability of modifying each position. The types of mutations are defined by groups of amino acids; the types are "Conservative" formed by the groups Polar (NCQHSTG), Acid (DE), Basic (KR), Non-polar (AILMPV), and Aromatic (WYF) residues; "Conservative extended" for Polar extended (NCQHSTGDEKR) and Non-polar extended (AILMPVWYF) residues; and "Unrestrained" allows the residues to be changed by any other type of residue.

Step 3: Homology Modeling. In this step, five structural models per derivative are generated with MODELLER. Among them, the structure with the lowest DOPE value is selected as the best model for the corresponding mutant.

Step 4: BA Estimation. The binding free energy is calculated using either the protein−protein or the protein−peptide affinity predictor. The selection of the estimator depends on the lengths of the chains in the structure; if a chain contains less than 30 residues, the protein−peptide predictor is used, otherwise the protein−protein estimator is applied.

Step 5: Ranking of Mutants. Finally, the mutants will be arranged in decreasing or increasing order of affinity, and either all or a selection of top candidates are returned to the user. The two sorting schemes allow the use of this module not only as an optimizer of the complex but also to spot mutations that can largely destabilize the complex of interest.

Applicability Domain

Defining the applicability domain (AD) of a model is an important step before the deployment of a predictor as it allows to provide insights into the reliability of the estimations in new systems.[74] Here, the AD is the subspace defined by the value range of the variables of the models (structural descriptors) in our training dataset (Supporting Information Tables SI-7 and SI-8). Thus, the descriptors' value of a new complex is checked to determine whether this structure is within the AD of the model. The result of this analysis is provided to the users alongside the predicted binding affinity value. Estimations associated with instances outside the AD should be interpreted with special care and probably corroborated by other methods. Additionally, Supporting Information Table SI-9 summarizes the sizes of the peptides and receptors used to train and test the protein−peptide model. Supporting Information Tables SI-10 and SI-11 present the descriptive statistics of the training, development, and test sets used in the modeling.

Implementation Details

The frontend of PPI-Affinity was implemented with the Python Framework Django v3.1.1 for uploading and validating the user data. The backend, which constitutes the core of the program, was programmed in Python 3. Internally, PPI-Affinity uses MODELLER[37−40] to create the new homology structures. ProtDCal[42] is used to calculate the structural descriptors required by the SVM models, and Weka 3.8.4[52] is the interpreter of these predictors to predict the binding free energy of the generated structures. Finally, a queuing system is employed for the management of the jobs sent by the users.

■ CONCLUSIONS

We developed PPI-Affinity, a binding free energy predictor targeting protein−protein and protein−peptide complexes specifically. This fast-screening tool moderately outperformed the predictions and ranking power of similar empirical predictors. The performance of the models was evaluated on various test sets, which include a largely used benchmark set for empirical binding free energy predictors and scoring functions, as well as new augmented datasets gathered in this work from BioLip and PDBbind. We also evaluated the ranking power of the protein−peptide model on a set of EPI-X4 derivatives and HtrAs peptide binders. Altogether, these tests highlight PPI-Affinity, not only as a top-ranked predictor but also as the most robust tool with respect to performance in different tests.

Furthermore, we implemented our models in a freely available web server that incorporates diverse functionalities that allow the screening of protein complexes as well as the engineering of the amino acid composition at the interface of the complex, to enhance the binding affinity or to spotlight critical mutants that may destabilize the interaction. The PPI-Affinity web server is thus a versatile tool with a direct impact on the design of peptide binders as well as in protein engineering and design.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jproteome.2c00020.

Setup of parameters used for the calculation of the structural descriptors (Supporting Information Section SI-1); preliminary study of Machine Learning (ML) techniques (Supporting Information Section SI-2); feature selection process for the protein−protein BA modeling (Supporting Information Section SI-3); feature selection process for the protein−peptide BA modeling (Supporting Information Section SI-4); evaluation of PPI-Affinity and a state-of-the-art classifier (Supporting Information Section SI-5); assays used to determine the BA of EPI-X4 derivatives against the CRCX4 receptor (Supporting Information Section SI-6); generation and ranking of peptide binders to the PDZ domains of HTRA1 and HTRA3 (Supporting Information Section SI-7); tabular description of PPI-Affinity and other state-of-the-art PPI prediction tools (Supporting Information Table SI-1); correlation coefficient (R) of BA predictors on the test set of 100 protein−peptide complexes (Supporting Information Table SI-2); performance of intermediate models for the protein−protein BA modeling during the hyper-parameters tuning process (Supporting Information Table SI-3); performance of individual and ensemble models for the protein−protein complexes (Supporting Information Table SI-4); performance of the intermediate models for the protein−peptide modeling during the hyperparameters tuning process (Supporting Information Table SI-5); performance of individual and ensemble models for the protein−peptide complexes (Supporting Information Table SI-6); applicability domain of the protein−protein model (Supporting Information Table SI-7); applicability domain of the protein−peptide model (Supporting Information Table SI-8); minimum and maximum values of the sequences' length of peptides and proteins (Supporting Information Table SI-9); descriptive statistics of the protein−protein data sets (Supporting Information Table SI-10); descriptive statistics of the protein−peptide data sets (Supporting Information Table SI-11); distribution of ΔG bind values in the dataset of protein−protein complexes (Supporting Information Figure SI-1); distribution of ΔG bind values in the datasets of the protein−protein ensemble model (Supporting Information Figure SI-2); characterization of the dataset of protein−peptide complexes (Supporting Information Figure SI-3); distribution of ΔG bind values in the datasets of the protein−peptide ensemble model (Supporting Information Figure SI-4); plots of experimental vs predicted on the test sets of protein−protein BA data (Supporting Information Figure SI-5); plot of experimental vs predicted on the test set of protein−peptide BA data (Supporting Information Figure SI-6) (PDF)

Python script used to randomly split the datasets into training, development, and test subsets (Script SI-1); configuration file for ProtDCal to compute the descriptors of the protein−protein model (File SI-1); configuration file for ProtDCal to compute the descriptors of the protein−peptide model (File SI-2); binding affinity predictions of PPI-Affinity and state-of-the-art tools on the benchmark of 79 protein−protein complexes employed by Vangone and Bonvin[21] (File SI-3); binding affinity predictions of PPI-Affinity and state-of-the-art tools on the hold-out set of 90 protein−protein complexes taken from PDBbind (v.2020) (File SI-4); binding affinity predictions of PPI-Affinity and state-of-the-art tools on the hold-out set of 177 (26 wild-type and 151 mutants) protein−protein complexes taken from the SKEMPI v2.0 dataset (File SI-5); binding affinity predictions of PPI-Affinity and state-of-the-art tools on the hold-out set of 100 protein−peptide complexes taken from the Biolip database (File SI-6); binding affinity predictions of PPI-Affinity and other state-of-the-art tools on the test set of protein−peptide complexes containing EPI-X4 and 56 derivatives coupled to the CXCR4 receptor (File SI-7); and summary of the protein−protein and protein−peptide complexes used in the training, validation, and test of the PPI-Affinity models (File SI-8) (ZIP)

■ AUTHOR INFORMATION

Corresponding Author

Elsa Sanchez-Garcia − *Computational Biochemistry, Center of Medical Biotechnology, University of Duisburg-Essen, Essen 45141, Germany;* orcid.org/0000-0002-9211-5803; Email: elsa.sanchez-garcia@uni-due.de

Authors

Sandra Romero-Molina − *Computational Biochemistry, Center of Medical Biotechnology, University of Duisburg-Essen, Essen 45141, Germany*

Yasser B. Ruiz-Blanco − *Computational Biochemistry, Center of Medical Biotechnology, University of Duisburg-Essen, Essen 45141, Germany*

Joel Mieres-Perez − *Computational Biochemistry, Center of Medical Biotechnology, University of Duisburg-Essen, Essen 45141, Germany*

Mirja Harms − *Institute of Molecular Virology, Ulm University Medical Center, Ulm 89081, Germany*

Jan Münch − *Institute of Molecular Virology, Ulm University Medical Center, Ulm 89081, Germany; Core Facility Functional Peptidomics, Ulm University Medical Center, Ulm 89081, Germany;* orcid.org/0000-0001-7316-7141

Michael Ehrmann − *Faculty of Biology, Center of Medical Biotechnology, University of Duisburg-Essen, Essen 45141, Germany;* orcid.org/0000-0002-1927-260X

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jproteome.2c00020

Author Contributions

S.R.-M., Y.B.R.-B., and E.S.-G. designed the project. S.R.-M. implemented the codes, performed the machine learning analyses and part of the validation tests, and built the webserver. J.M.-P. carried out the HtrAs validation tests. Y.B.R.-B. and E.S.-G. supervised the modeling and implementation of the method. S.R.-M., J.M.-P., Y.B.R.-B., and E.S.-G. wrote the manuscript. M.H. carried out the antibody competition assays. J.M. supervised the experiments and the synthesis of the peptides, and both, J.M. and M.H., contributed to the analysis of the experimental data. M.E. contributed HtrAs expertise. All authors contributed to revising and editing the manuscript. All authors have given approval to the final version of the manuscript.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)-Project ID: 436586093 and the SFB 1430 “Molecular Mechanisms of Cell State Transitions”, subprojects B04 and A06. E.S.-G. was also supported by the DFG under Germany’s Excellence Strategy-EXC-2033-Project number 390677874. E.S.-G., S.R.-M., M.J., and J.M. acknowledge the SFB 1279 “Exploiting the Human Peptidome for Novel Antimicrobial and Anticancer Agents”, subproject A06, funded by the DFG. E.S.-G. and J.M.-P. thank Dr. Melisa Merdanovic for useful discussions. J.M. also acknowledges funding by the DFG (3115/11−1). M.E. was supported by the Deutsche Forschungsgemeinschaft EH 100/18−1. M.H. received project start-up funding via the “Bausteinprogramm” of the Medical Faculty and via the Gender Equality Program for female scientists of Ulm University.

■ REFERENCES

(1) Lu, H.; Zhou, Q.; He, J.; Jiang, Z.; Peng, C.; Tong, R.; Shi, J. Recent advances in the development of protein−protein interactions modulators: mechanisms and clinical trials. *Signal Transduction Targeted Ther.* **2020**, *5*, 213.

(2) Blazer, L. L.; Neubig, R. R. Small Molecule Protein−Protein Interaction Inhibitors as CNS Therapeutic Agents: Current Progress and Future Hurdles. *Neuropsychopharmacology* **2009**, *34*, 126−141.

(3) Al Qaraghuli, M. M.; Kubiak-Ossowska, K.; Ferro, V. A.; Mulheran, P. A. Antibody-protein binding and conformational changes: identifying allosteric signalling pathways to engineer a better effector response. *Sci. Rep.* **2020**, *10*, No. 13696.

(4) Hill, T. A.; Shepherd, N. E.; Diness, F.; Fairlie, D. P. Constraining Cyclic Peptides To Mimic Protein Structure Motifs. *Angew. Chem., Int. Ed.* **2014**, *53*, 13020−13041.

(5) Lau, J. L.; Dunn, M. K. Therapeutic peptides: Historical perspectives, current development trends, and future directions. *Bioorg. Med. Chem.* **2018**, *26*, 2700−2707.

(6) Henninot, A.; Collins, J. C.; Nuss, J. M. The Current State of Peptide Drug Discovery: Back to the Future? *J. Med. Chem.* **2018**, *61*, 1382−1414.

(7) Wang, S. H.; Yu, J. Structure-based design for binding peptides in anti-cancer therapy. *Biomaterials* **2018**, *156*, 1−15.

(8) Horton, N.; Lewis, M. Calculation of the free energy of association for protein complexes. *Protein Sci.* **1992**, *1*, 169−181.

(9) Liu, S.; Zhang, C.; Zhou, H.; Zhou, Y. A physical reference state unifies the structure-derived potential of mean force for protein folding and binding. *Proteins: Struct., Funct., Bioinf.* **2004**, *56*, 93−101.

(10) Cheng, T. M.-K.; Blundell, T. L.; Fernandez-Recio, J. pyDock: Electrostatics and desolvation for effective scoring of rigid-body protein−protein docking. *Proteins: Struct., Funct., Bioinf.* **2007**, *68*, 503−515.

(11) Pierce, B.; Weng, Z. ZRANK: Reranking protein docking predictions with an optimized energy function. *Proteins: Struct., Funct., Bioinf.* **2007**, *67*, 1078−1086.

(12) Andrusier, N.; Nussinov, R.; Wolfson, H. J. FireDock: Fast interaction refinement in molecular docking. *Proteins: Struct., Funct., Bioinf.* **2007**, *69*, 139−159.

(13) Pierce, B.; Weng, Z. A combination of rescoring and refinement significantly improves protein docking performance. *Proteins: Struct., Funct., Bioinf.* **2008**, *72*, 270−279.

(14) Su, Y.; Zhou, A.; Xia, X.; Li, W.; Sun, Z. Quantitative prediction of protein−protein binding affinity with a potential of mean force considering volume correction. *Protein Sci.* **2009**, *18*, 2550−2558.

(15) Chaudhury, S.; Lyskov, S.; Gray, J. J. PyRosetta: a script-based interface for implementing molecular modeling algorithms using Rosetta. *Bioinformatics* **2010**, *26*, 689−691.

(16) Ravikant, D. V. S.; Elber, R. PIE-efficient filters and coarse grained potentials for unbound protein-protein docking. *Proteins* **2010**, *78*, 400−419.

(17) Moal, I. H.; Agius, R.; Bates, P. A. Protein−protein binding affinity prediction on a diverse set of structures. *Bioinformatics* **2011**, *27*, 3002−3009.

(18) Pons, C.; Talavera, D.; de la Cruz, X.; Orozco, M.; Fernandez-Recio, J. Scoring by Intermolecular Pairwise Propensities of Exposed Residues (SIPPER): A New Efficient Potential for Protein−Protein Docking. *J. Chem. Inf. Model.* **2011**, *51*, 370−377.

(19) Viswanath, S.; Ravikant, D. V. S.; Elber, R. Improving ranking of models for protein complexes with side chain modeling and atomic potentials. *Proteins: Struct., Funct., Bioinf.* **2013**, *81*, 592−606.

(20) Kastritis, P. L.; Rodrigues, J. P. G. L. M.; Folkers, G. E.; Boelens, R.; Bonvin, A. M. J. J. Proteins Feel More Than They See: Fine-Tuning of Binding Affinity by Properties of the Non-Interacting Surface. *J. Mol. Biol.* **2014**, *426*, 2632−2652.

(21) Vangone, A.; Bonvin, A. M. Contacts-based prediction of binding affinity in protein-protein complexes. *eLife* **2015**, *4*, No. e07454.

(22) Huang, X.; Zheng, W.; Pearce, R.; Zhang, Y. SSIPe: accurately estimating protein−protein binding affinity change upon mutations using evolutionary profiles in combination with an optimized physical energy function. *Bioinformatics* **2020**, *36*, 2429−2437.

(23) Wang, B.; Su, Z.; Wu, Y. Computational Assessment of Protein−Protein Binding Affinity by Reverse Engineering the Energetics in Protein Complexes *Genomics, Proteomics Bioinf.* 2021, DOI: 10.1016/j.gpb.2021.03.004.

(24) Abbasi, W. A.; Yaseen, A.; Hassan, F. U.; Andleeb, S.; Minhas, F. U. A. A. ISLAND: in-silico proteins binding affinity prediction using sequence information. *BioData Mining* **2020**, *13*, 20.

(25) Yugandhar, K.; Gromiha, M. M. Protein−protein binding affinity prediction from amino acid sequence. *Bioinformatics* **2014**, *30*, 3583−3589.

(26) Abbasi, W. A.; Asif, A.; Ben-Hur, A.; Minhas, F. uA. A. Learning protein binding affinity using privileged information. *BMC Bioinf.* **2018**, *19*, 425.

(27) Xue, L. C.; Rodrigues, J. P.; Kastritis, P. L.; Bonvin, A. M.; Vangone, A. PRODIGY: a web server for predicting the binding affinity of protein−protein complexes. *Bioinformatics* **2016**, *32*, No. btw514.

(28) Ballester, P. J.; Mitchell, J. B. O. A machine learning approach to predicting protein-ligand binding affinity with applications to molecular docking. *Bioinformatics* **2010**, *26*, 1169−1175.

(29) Jiménez, J.; Škalič, M.; Martínez-Rosell, G.; De Fabritiis, G. KDEEP: Protein−Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks. *J. Chem. Inf. Model.* **2018**, *58*, 287−296.

(30) Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5−32.

(31) LeCun, Y.; Boser, B.; Denker, J.; Henderson, D.; Howard, R.; Hubbard, W.; Jackel, L. *Proceedings of Advances in Neural Information Processing Systems*, 1989; 396−404.

(32) LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278−2324.

(33) Hu, J.; Liu, Z. DeepMHC: Deep Convolutional Neural Networks for High-performance peptide-MHC Binding Affinity Prediction. *bioRxiv Preprint* **2017**, 239236.

(34) Jurtz, V.; Paul, S.; Andreatta, M.; Marcatili, P.; Peters, B.; Nielsen, M. NetMHCpan-4.0: Improved Peptide-MHC Class I Interaction Predictions Integrating Eluted Ligand and Peptide Binding Affinity Data. *J. Immunol.* **2017**, *199*, 3360−3368.

(35) Yang, J.; Roy, A.; Zhang, Y. BioLiP: a semi-manually curated database for biologically relevant ligand-protein interactions. *Nucleic Acids Res.* **2012**, *41*, D1096−D1103.

(36) Wang, R.; Fang, X.; Lu, Y.; Wang, S. The PDBbind Database: Collection of Binding Affinities for Protein−Ligand Complexes with Known Three-Dimensional Structures. *J. Med. Chem.* **2004**, *47*, 2977−2980.

(37) Šali, A.; Blundell, T. L. Comparative Protein Modelling by Satisfaction of Spatial Restraints. *J. Mol. Biol.* **1993**, *234*, 779−815.

(38) Fiser, A.; Do, R. K.; Sali, A. Modeling of loops in protein structures. *Protein Sci.* **2000**, *9*, 1753−1773.

(39) Martí-Renom, M. A.; Stuart, A. C.; Fiser, A.; Sánchez, R.; Melo, F.; Šali, A. Comparative Protein Structure Modeling of Genes and Genomes. *Annu. Rev. Biophys. Biomol. Struct.* **2000**, *29*, 291−325.

(40) Webb, B.; Sali, A. Comparative Protein Structure Modeling Using MODELLER. *Curr. Protoc. Bioinform.* **2016**, *54*, 5.6.1−5.6.37.

(41) Dolinsky, T. J.; Nielsen, J. E.; McCammon, J. A.; Baker, N. A. PDB2PQR: an automated pipeline for the setup of Poisson-Boltzmann electrostatics calculations. *Nucleic Acids Res.* **2004**, *32*, W665−W667.

(42) Ruiz-Blanco, Y. B.; Paz, W.; Green, J.; Marrero-Ponce, Y. ProtDCal: A program to compute general-purpose-numerical descriptors for sequences and 3D-structures of proteins. *BMC Bioinf.* **2015**, *16*, 162.

(43) Romero-Molina, S.; Ruiz-Blanco, Y. B.; Green, J. R.; Sanchez-Garcia, E. ProtDCal-Suite: A web server for the numerical codification and functional analysis of proteins. *Protein Sci.* **2019**, *28*, 1734−1743.

(44) Kinjo, A. R.; Nishikawa, K. Recoverable one-dimensional encoding of three-dimensional protein structures. *Bioinformatics* **2005**, *21*, 2167−2170.

(45) Biggar, K. K.; Charih, F.; Liu, H.; Ruiz-Blanco, Y. B.; Stalker, L.; Chopra, A.; Connolly, J.; Adhikary, H.; Frensemier, K.; Hoekstra, M.; Galka, M.; Fang, Q.; Wynder, C.; Stanford, W. L.; Green, J. R.; Li, S. S. C. Proteome-wide Prediction of Lysine Methylation Leads to Identification of H2BK43 Methylation and Outlines the Potential Methyllysine Proteome. *Cell Rep.* **2020**, *32*, 107896.

(46) Ruiz-Blanco, Y. B.; Marrero-Ponce, Y.; García-Hernández, E.; Green, J. Novel "extended sequons" of human N-glycosylation sites improve the precision of qualitative predictions: an alignment-free study of pattern recognition using ProtDCal protein features. *Amino Acids* **2017**, *49*, 317−325.

(47) Romero-Molina, S.; Ruiz-Blanco, Y. B.; Harms, M.; Münch, J.; Sanchez-Garcia, E. PPI-Detect: A support vector machine model for sequence-based prediction of protein−protein interactions. *J. Comput. Chem.* **2019**, *40*, 1233−1242.

(48) Ruiz-Blanco, Y. B.; Agüero-Chapin, G.; García-Hernández, E.; Álvarez, O.; Antunes, A.; Green, J. Exploring general-purpose protein features for distinguishing enzymes and non-enzymes within the twilight zone. *BMC Bioinf.* **2017**, *18*, 349.

(49) Corral-Corral, R.; Beltrán, J. A.; Brizuela, C. A.; Del Rio, G. Systematic Identification of Machine-Learning Models Aimed to Classify Critical Residues for Protein Function from Protein Structure. *Molecules* **2017**, *22*, 1673.

(50) Kleandrova, V. V.; Ruso, J. M.; Speck-Planche, A.; Dias Soeiro Cordeiro, M. N. Enabling the Discovery and Virtual Screening of Potent and Safe Antimicrobial Peptides. Simultaneous Prediction of Antibacterial Activity and Cytotoxicity. *ACS Comb. Sci.* **2016**, *18*, 490−498.

(51) Speck-Planche, A.; Kleandrova, V. V.; Ruso, J. M.; D S Cordeiro, M. N. First Multitarget Chemo-Bioinformatic Model To Enable the Discovery of Antibacterial Peptides against Multiple Gram-Positive Pathogens. *J. Chem. Inf. Model.* **2016**, *56*, 588−598.

(52) Frank, E.; Hall, M. A.; Witten, I. H. *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques"*, 2016.

(53) Patel, L.; Shukla, T.; Huang, X.; Ussery, D. W.; Wang, S. Machine Learning Methods in Drug Discovery. *Molecules* **2020**, *25*, 5277.

(54) Yang, X.; Wang, Y.; Byrne, R.; Schneider, G.; Yang, S. Concepts of Artificial Intelligence for Computer-Assisted Drug Discovery. *Chem. Rev.* **2019**, *119*, 10520−10594.

(55) Rodríguez-Pérez, R.; Vogt, M.; Bajorath, J. Influence of Varying Training Set Composition and Size on Support Vector Machine-Based Prediction of Active Compounds. *J. Chem. Inf. Model.* **2017**, *57*, 710−716.

(56) Nguyen, A.; Yosinski, J.; Clune, J. In *Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images*, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015; pp 427−436.

(57) Nalepa, J.; Kawulok, M. Selecting training sets for support vector machines: a review. *Artif. Intell. Rev.* **2019**, *52*, 857−900.

(58) Shevade, S. K.; Keerthi, S. S.; Bhattacharyya, C.; Murthy, K. R. K. Improvements to the SMO algorithm for SVM regression. *IEEE Trans. Neural Networks* **2000**, *11*, 1188−1193.

(59) Kuncheva, L. I., *Combining Pattern Classifiers: Methods and Algorithms*; John Wiley & Sons, Inc., 2004.

(60) Kittler, J.; Hatef, M.; Duin, R. P. W.; Matas, J. On combining classifiers. *IEEE Trans. Pattern Analysis Machine Intelligence* **1998**, *20*, 226−239.

(61) Chih-Wei, H.; Chih-Chung, C.; Chih-Jen, L. *A Practical Guide to Support Vector Classification*, 2003.

(62) Jankauskaitė, J.; Jiménez-García, B.; Dapkūnas, J.; Fernández-Recio, J.; Moal, I. H. SKEMPI 2.0: an updated benchmark of changes in protein−protein binding energy, kinetics and thermodynamics upon mutation. *Bioinformatics* **2019**, *35*, 462−469.

(63) Wang, R.; Fang, X.; Lu, Y.; Yang, C.-Y.; Wang, S. The PDBbind Database: Methodologies and Updates. *J. Med. Chem.* **2005**, *48*, 4111−4119.

(64) Liu, Z.; Su, M.; Han, L.; Liu, J.; Yang, Q.; Li, Y.; Wang, R. Forging the Basis for Developing Protein−Ligand Interaction Scoring Functions. *Acc. Chem. Res.* **2017**, *50*, 302−309.

(65) Zirafi, O.; Kim, K.-A.; Ständker, L.; Mohr, K. B.; Sauter, D.; Heigele, A.; Kluge; Silvia, F.; Wiercinska, E.; Chudziak, D.; Richter, R.; Moepps, B.; Gierschik, P.; Vas, V.; Geiger, H.; Lamla, M.; Weil, T.; Burster, T.; Zgraja, A.; Daubeuf, F.; Frossard, N.; Hachet-Haas, M.; Heunisch, F.; Reichetzeder, C.; Galzi, J.-L.; Pérez-Castells, J.; Canales-Mayordomo, A.; Jiménez-Barbero, J.; Giménez-Gallego, G.; Schneider, M.; Shorter, J.; Telenti, A.; Hocher, B.; Forssmann, W.-G.; Bonig, H.; Kirchhoff, F.; Münch, J. Discovery and Characterization of an Endogenous CXCR4 Antagonist. *Cell Rep.* **2015**, *11*, 737−747.

(66) Pawig, L.; Klasen, C.; Weber, C.; Bernhagen, J.; Noels, H. Diversity and Inter-Connections in the CXCR4 Chemokine Receptor/Ligand Family: Molecular Perspectives. *Front. Immunol.* **2015**, *6*, No. 429.

(67) Harms, M.; Gilg, A.; Ständker, L.; Beer, A. J.; Mayer, B.; Rasche, V.; Gruber, C. W.; Münch, J. Microtiter plate-based antibody-competition assay to determine binding affinities and plasma/blood stability of CXCR4 ligands. *Sci. Rep.* **2020**, *10*, No. 16036.

(68) Liang, S.; Meroueh, S. O.; Wang, G.; Qiu, C.; Zhou, Y. Consensus scoring for enriching near-native structures from protein-protein docking decoys. *Proteins* **2009**, *75*, 397−403.

(69) Harms, M.; Habib, M. M. W.; Nemska, S.; Nicolò, A.; Gilg, A.; Preising, N.; Sokkar, P.; Carmignani, S.; Raasholm, M.; Weidinger, G.; Kizilsavas, G.; Wagner, M.; Ständker, L.; Abadi, A. H.; Jumaa, H.; Kirchhoff, F.; Frossard, N.; Sanchez-Garcia, E.; Münch, J. An optimized derivative of an endogenous CXCR4 antagonist prevents

atopic dermatitis and airway inflammation. *Acta Pharm. Sin. B* **2021**, *11*, 2694−2708.

(70) Sokkar, P.; Harms, M.; Stürzel, C.; Gilg, A.; Kizilsavas, G.; Raasholm, M.; Preising, N.; Wagner, M.; Kirchhoff, F.; Ständker, L.; Weidinger, G.; Mayer, B.; Münch, J.; Sanchez-Garcia, E. Computational modeling and experimental validation of the EPI-X4/CXCR4 complex allows rational design of small peptide antagonists. *Commun. Biol.* **2021**, *4*, 1113.

(71) Clausen, T.; Southan, C.; Ehrmann, M. The HtrA Family of Proteases: Implications for Protein Composition and Cell Fate. *Mol. Cell* **2002**, *10*, 443−455.

(72) Runyon, S. T.; Zhang, Y.; Appleton, B. A.; Sazinsky, S. L.; Wu, P.; Pan, B.; Wiesmann, C.; Skelton, N. J.; Sidhu, S. S. Structural and functional analysis of the PDZ domains of human HtrA1 and HtrA3. *Protein Sci.* **2007**, *16*, 2454−2471.

(73) Kendall, M. G. A New Measure of Rank Correlation. *Biometrika* **1938**, *30*, 81−93.

(74) Sheridan, R. P. The Relative Importance of Domain Applicability Metrics for Estimating Prediction Errors in QSAR Varies with Training Set Diversity. *J. Chem. Inf. Model.* **2015**, *55*, 1098−107.

Supporting Information

PPI-Affinity: A web tool for the prediction and optimization of protein – peptide and protein – protein binding affinity

*Sandra Romero-Molina[1], Yasser B. Ruiz-Blanco [1], Joel Mieres-Perez[1], Mirja Harms[2], Jan Münch[2,3], Michael Ehrmann[4] and Elsa Sanchez-Garcia[1]**

[1]Computational Biochemistry, Center of Medical Biotechnology, University of Duisburg-Essen, Essen, Germany

[2]Institute of Molecular Virology, Ulm University Medical Center, Ulm, Germany

[3]Core Facility Functional Peptidomics, Ulm University Medical Center, Ulm, Germany

[4]Faculty of Biology, Center of Medical Biotechnology, University of Duisburg-Essen, Essen, Germany

*Email: elsa.sanchez-garcia@uni-due.de

Table of Contents

Table SI-1. Description of PPI-Affinity and state-of-the-art PPI prediction tools. S4

Table SI-2. Correlation coefficient (R) of protein-protein and protein-ligand BA predictors on the test set of 100 protein-peptide complexes. S10

Figure SI-1. Distribution of $\Delta Gbind$ values in the dataset of 833 protein-protein complexes. S11

Section SI-1. Parameter setup for the calculation of the structural descriptors in ProtDCal.[7, 8] S12

Section SI-2 Preliminary study conducted to the identify the Machine Learning (ML) technique to use in the development of the models. S13

Section SI-3. Description of the feature selection process for the protein-protein BA modeling. S15

Table SI-3.1. Descriptors of the protein-protein model. S15

File SI-1 (separated). Configuration file for ProtDCal to compute the 26 structural descriptors. S17

Figure SI-2. Distribution of $\Delta Gbind$ values in the four subsets of the training data for the protein-protein ensemble learning protocol. S18

Table SI-3. Summary of the intermediate models and performance measures for the protein-protein BA modeling during the hyperparameters tuning process. S19

Table SI-4. Summary of the performance of the individual and the ensemble models for protein-protein models. S21

Figure SI-3. Characterization of the dataset of protein-peptide complexes used in this work. S22

Section SI-4. Description of the process of feature selection for creating the protein-peptide model................S23

Table SI-4.1. Descriptors of the protein-peptide model................S23

File SI-2 (separated). Configuration file for ProtDCal to compute the 37 structural descriptors. S25

Figure SI-4. Distribution of $\Delta Gbind$ values in the four subsets of the training data for the protein-peptide ensemble learning protocol................S26

Table SI-5. Summary of the intermediate models and performance measures for the protein-peptide modeling during the hyperparameters tuning process. S27

Table SI-6. Summary of the performance of the individual and the ensemble models for protein-peptide models................S30

Figure SI-5. Plots of experimental vs. predicted BA values of PPI-Affinity on the test sets of protein – protein affinity data. S31

Section SI-5. Performance of PPI-Affinity vs. a state-of-the-art biding affinity classifier...S32

Table SI-5.1. Summary of the evaluation of PPI-Affinity and the LUPIA[2] classifier on two sets of protein – protein affinity data. S33

Figure SI-6. Plot of experimental vs. predicted by PPI-Affinity BA values on the test set of protein – peptide affinity data. S34

Section SI-6. Description of the assays used to determine the binding affinities of EPI-X4 derivatives against the CRCX4 receptor**.**...S35

Table SI-6.1 Summary of the binding affinities of EPI-X4 derivatives against CRCX4. S35

Section SI-7. Description of the models used for the generation of the data related to peptide binders to the PDZ domain of HTRA1 or HTRA3 (Tables 3 and 4)....................................S37

Table SI-7.1. Ranking of BA of HTRA1-peptide complexes as predicted by state-of-the-art models...S38

Table SI-7.2. Ranking of BA of HTRA3-peptide complexes as predicted by state-of-the-art models...S38

Table SI-7. Summary of values that define the applicability domain of the protein-protein model...S40

Table SI-8. Summary of values that define the applicability domain of the protein-peptide model...S42

Table SI-9. Summary of the minimum and maximum values of the sequences' length of the peptides and proteins in each dataset. S44

Table SI-10. Descriptive statistics of the different data sets used in the modeling and test of the protein-protein BA predictor..S45

Table SI-11. Descriptive statistics of the training, development and test sets used in the modeling of the protein-peptide BA predictor. S46

Table SI-1. Description of PPI-Affinity and state-of-the-art PPI prediction tools.

Method	***Training size***	***Database***	***Type of problem***	***Validation technique***	***Domain of application***	***Input information***	***Type of molecular descriptors***	***ML algorithms***	***Availability***
Prodigy[1]	81	Benchmark dataset (Kastritis et al., 2011)[2]	Regression	4-fold cross validation (repeated 10 times)	Protein-Protein	3D	Descriptors based in the network of inter-residue contacts (ICs) and the non-interacting surface (NIS)	Linear Regression	Webserver
DFIRE[3]	-	-	-	Test set	Protein-Protein/Peptide	3D	-	-	Webserver / Standalone program
CP_PIE[4]	540	PDB[5, 6]	Regression	Test set	Protein-Protein	3D	Descriptors based on residue contacts and the overlapping area	Linear programming formulation	Standalone program
Kdeep[7]	3767	PDBbind (v.2016)[8]	Regression	Test set	Protein-Ligand	3D	Pharmacophoric-like properties in the	Convolutional Neural Networks	Webserver

							binding site		
RF-Score[9]	1105	PDBbind (v.2007)[8]	Regression	Out-Of-Bag data/Test set/y-scrambling	Protein-Ligand	3D	Occurrence counts of atom type pairs within a distance range	Random Forest	Standalone program
Trypano-PPI[10]	5872	PDB[5, 6] / dataset reported by Dobson and Doig[11]	Classification	Validation set	Protein-Protein interactions in Trypanosome	3D	Markov Chain numerical descriptors based on average electrostatic potentials (MARCH-INSIDE 2.0 package)	Linear Neural Network	Webserver
Plasmod-PPI[12]	5257	PDB[5, 6] / dataset reported by Dobson and Doig[11]	Classification	Validation set	Protein-Protein interactions in Plasmodium	3D	Markov Chain numerical descriptors based on electrostatic entropies calculations (MARCH-INSIDE 2.0 package)	Classifier Trees (CT) / Linear discriminant analysis (LDA) for feature selection	Webserver

***GO-PseAA*[13]**	6323	STRING[14]	Classification	Jack-knife test (LOOCV)	Protein-Protein	1D	Pseudo-amino acid composition (PseAA)/ Gene Ontology Consortium (GO)	Intimate Sorting (ISort) Classifier	-
***ALT-IN*[15]**	Dataset 1: 1831 Dataset 2: 5460	Dataset 1:[16] Dataset 2:[16-20]	Classification	Nested leave-group-out cross validation (CV) /Case of study	Protein-Protein (predicting disruptions in PPIs induced by alternative splicing)	1D	Biochemical features of the reference isoform and its interaction partner/ Domain interaction knowledge-based statistical potentials/ Selected characteristics of alternative splicing events	Random Forest-driven supervised and semi-supervised learning	Standalone program
***iPPBS-Opt*[21]**	13771 surface-residue / 27442	Dataset of 99 proteins[22]	Classification	10-fold CV	Protein-Protein/Peptide binding sites	1D	Pseudo Amino Acid Composition (PseAAC)	Random Forest	Webserver

	all-residue								
***PPA-Pred*[23]**	135	Benchmark dataset (Kastritis *et al.*, 2011)[2]	Regression	LOOCV	Protein-Protein	1D	Descriptors based in sequence and structure properties	Multiple regression technique	Webserver
***ISLAND*[24]**	135/39 test	Benchmark dataset (Kastritis *et al.* , 2011)[2]	Regression	LOOCV	Protein-Protein	1D	Descriptors based on biophysical amino acids properties and structural properties derived from the sequence (Propy package)[25]	Support Vector Machine (SVM) for regression	Webserver
***SSIPe*[26]**	1470 training/ 734 test/ 888 test/ 190/152	NIL[27] /STRING[14]/ SKEMPI 2.0[28]/CAPRI[29]	Regression	5-fold CV/Test set	Protein-Protein	3D	Descriptors based in sequence and structure	Monte Carlo procedure/ Linear regression	Webserver/ Standalone

							interface evolutionary profiles		
Wang et al.[30]	158 wild-type protein complexes and 3205 mutants	SKEMPI 1.0[31]	Regression	LOOCV	Protein-Protein	3D	Numerical descriptors based on energetic contributions and calculated with the Monte-Carlo algorithm	Knowledge-based potential (Monte-Carlo simulations for weights optimization)	-
LUPIA[32]	128 (training set) /39 protein-(validation set)	Benchmark dataset (Kastritis *et al.*, 2011)[2]/ Chen et al.[33]	Classification	LOOCV/ validation set	Protein-Protein	1D	k-mer composition, BLOSUM-62 features,[3] [4] number of interacting residue pairs, Moal[35] and Dias	Learning Using Privileged Information (LUPI) /SVM	Webserver / Standalone program

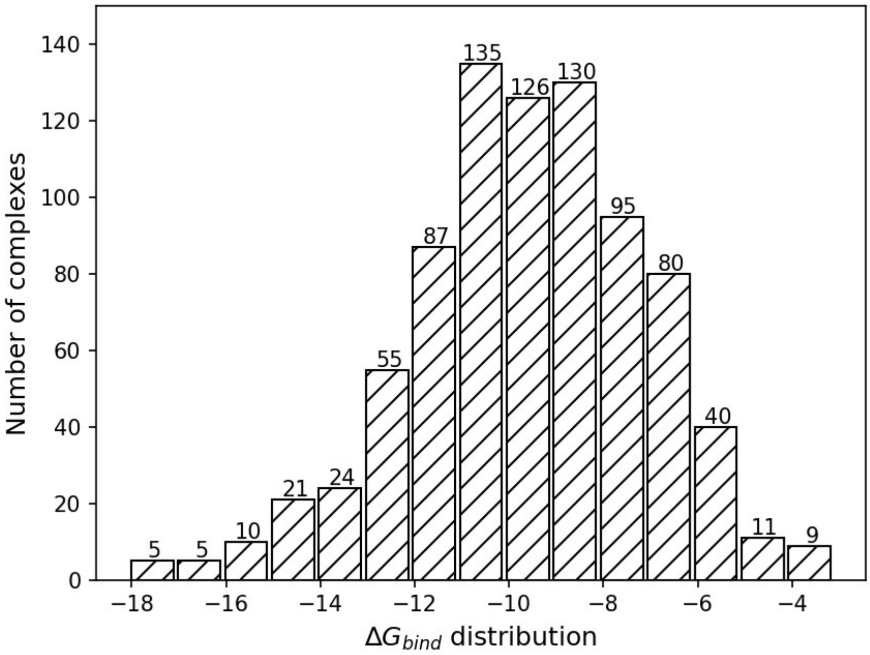
							descriptors[36]		
PPI-Affinity	648 protein-protein / 922 protein-peptide	PDBbind (v.2020)[8] / Biolip[37]	Regression	10-fold CV/Validation set/Test set	Protein-Protein/Peptide	3D	Structural numerical descriptors calculated with ProtDCal software	Ensemble method / SVM	Webserver

Table SI-2. Correlation coefficient (R) of protein-protein and protein-ligand BA predictors on the test set of 100 protein-peptide complexes.

Method	*R*	MAE (kcal/mol)
Prodigy[1, 38]	0.13	1.9
DFIRE[3]	0.29	8.7
PIE[4]	-0.28	9.0
Kdeep*[7]	0.32	10.7
RF-Score*[39]	0.23	1.8

Protein-ligand methods are marked with *

Figure SI-1. Distribution of ΔG_{bind} values in the dataset of 833 protein-protein complexes.



Section SI-1. Parameter setup for the calculation of the structural descriptors in ProtDCal.[40, 41]

Sections	Description
directory:	
Datasets/PDB_Protein_Format	Path to directory with the input PDB files
indices:	
wNc, wFLC, wNLC, wCO, wLCO, wRWCO, wCTP, wCLQ	List of used Topographic Indices of Folded Protein States
wCO: ECI, IP, ISA, Z1, Z2, Z3 wRWCO: ECI, IP, ISA, Z1, Z2, Z3 wNc: ECI, IP, ISA, Z1, Z2, Z3 wCLQ: ECI, IP, ISA, Z1, Z2, Z3 wNLC: ECI, IP, ISA, Z1, Z2, Z3 wLCO: ECI, IP, ISA, Z1, Z2, Z3 wCTP: ECI, IP, ISA, Z1, Z2, Z3 wFLC: ECI, IP, ISA, Z1, Z2, Z3	List of used weighting coefficients for each Topographic Indices of Folded Protein States
groups:	
ALA, ARG, ASN, ASP, CYS, GLU, GLN, GLY, HIS, ILE, LEU, LYS, MET, PHE, PRO, SER, THR, TRP, TYR, VAL, RTR, BSR, AHR, ALR, NPR, ARM, PLR, PCR, NCR, UCR, UFR, PRT	List of grouping operators
invariants:	
N1, N2, Ar, P2, G, V, CV, S, RA, K, DE, I50, SI, MI, TI	List of vicinity operators
parameters(t_cont,s_cont,A%,HydGroup,n,bins,K,SubG):	
4000.0, 10.0, 5.0, 9.4, 3.0, 30, 5, 3	Parameters values for internal options of the program
options(decimals,harmonicMeanType,geometricMeanType,windexID,datasetType,outputOrder):	
3, 0, 0, -1, pdb, true	Parameters values for internal options of the program

Section SI-2 Preliminary study conducted to the identify the Machine Learning (ML) technique to use in the development of the models.

First, we randomly divided the 833 protein-protein complexes into datasets of 743 and 90 structures to train and test the models respectively. Initially, ProtDCal generated 23 040 molecular descriptors for each protein-protein complex. We reduced this high number of dimensions through an attribute selection process. First, we applied a filter to remove descriptors with non or little variation in the training set (*RemoveUseless* filter of Weka). A descriptor that cannot be calculated for an instance represents a missing value in the final vector. For a regression problem, this means that missing values need to be replaced by a number (mean, median, ...). Since this could add more noise to the dataset, we handled this issue by deleting all the attributes that contained at least one instance with a missing value. These steps reduced the dataset to 1 477 attributes. Next, we ordered the attributes by their Pearson's correlation coefficient with the class (*CorrelationAttributeEval* implementation of Weka). The highest correlation of an attribute with the class was 0.24, while the minimum was -0.32. We selected those with correlation values between 0.1 <= R <= -0.1, reducing the data to 476 descriptors.

Next, we applied the *WrapperSubsetEval* technique implemented in Weka for obtaining the best subset of attributes to predict the class value. This is a supervised technique that evaluates subsets of features by training and evaluating, directly employing the classifier. The selection of the subsets is held by a search method. Here, we employed a genetic algorithm with a population of 20 individuals, crossover and mutation probabilities of 0.6 and 0.033 respectively and 20 generations maximum. Each subset was evaluated in 5-fold cross-validation, and the selection of the best subset was carried out attending to the correlation coefficient achieved by the classifier. This process was performed by each method individually (Table SI-2.1). We employed the Weka implementations of Linear Regression, Multilayer Perceptron: with zero (Linear Neural Network, LNN), one (ANN-1H) and two (ANN-2H) hidden layers, Random Forest, SVM for regression with the polynomial of degree one (SVM-PK-D1), degree two (SVM-PK-D2), and radial basis function (SVM-RBF) kernels. All the classifiers were executed with the default parameters values provided by Weka. In the case of Neural Networks, the amount of nodes (a) per hidden layer is defined by default in Weka as: a = (d + c) / 2, where d is the number of descriptors and c is the number of classes (c=1 for regression). For SVM-PK-D2, the Wrapper method was not applied, and the evaluation was performed on the attributes selected by SVM-PK-D1.

Table SI-2.1. Summary of the performance of different classifiers after the application of feature selection steps on the protein-protein BA data. The column "Descriptors" contains the number of attributes selected by each method with the Wrapper technique. The performance is expressed as the Pearson's Correlation coefficient (R) between experimental and predicted BA. Each model was evaluated on the training set, in 10-fold cross-validation (10-fold CV), and on the test set of 90 data points taken from PDBbind (v.2020).[8] The methods tested were Linear Regression, Multilayer Perceptron with zero (Linear Neural Network, LNN), one (ANN-1H) and two (ANN-2H) hidden layers, Random Forest, Support Vector Machine for regression with the polynomial (SVM-PK-D1, SVM-PK-D2) and the Radial Basis Function (SVM-RBF) kernels.

	Descriptors	Training set		10-fold CV		Test set	
		R	MAE	R	MAE	R	MAE
Linear Regression	74	0.68	1.4	0.60	1.6	0.39	2.0
LNN	12	-0.05	3.4	0.20	2.3	-0.01	3.1
ANN-1H	11	0.58	1.6	0.40	1.9	0.41	1.8
ANN-2H	10	0.55	1.7	0.39	1.9	0.40	1.9
Random Forest	27	0.98	0.6	0.67	1.5	0.61	1.6
SVM-PK-D1	36	0.57	1.5	0.54	1.6	0.40	1.9
SVM-PK-D2	36	0.77	1.0	0.42	2.0	0.38	2.0
SVM-RBF	171	0.63	1.4	0.58	1.6	0.45	1.8

The smallest subset of descriptors was obtained by the Neural Networks methods. However, under these architectures the models suffered from both: under-fitting in the case of LNN, with non or small correlation values among the model evaluations, as well as over-fitting in the case of ANN-1H and ANN-2H, with a difference greater than the 20% between the correlation on the training set, and 10-fold CV, and test set. In the case of Random Forest, the results on the training set (R = 0.98, MAE = 0.6), along with the performance in CV (R = 0.67, MAE = 1.5) and on the test set (R = 0.61, MAE = 1.6) clearly denoted high over-fitting.

Linear Regression showed close performance on the training set (R = 0.68, MAE = 1.4) and in 10-fold CV (R = 0.6, MAE = 1.6). However, the number of descriptors equal to 74 could harm the generalization power of the model, which may explain the fall in performance on the test set.

SVM exhibited slightly better results with the RBF kernel than with the polynomial kernel of degree one (SVM-PK-D1). However, this performance was achieved with almost five times more descriptors. In addition, we must consider that the polynomial kernel of first order is a linear equation. Thus, we can conclude that SVM with the polynomial kernel of first order provides the simplest and best-performing solutions. Therefore, we selected this framework to develop our models.

Section SI-3. Description of the feature selection process for the protein-protein BA modeling.

Initially, each instance in the dataset was represented as a vector of 23 040 molecular descriptors generated with ProtDCal, creating an initial matrix of 653 instances x 23.040 descriptors. For identifying the subset of attributes that best approximates the binding free energy value, we applied unsupervised and supervised features selection techniques.

First, we reduced the matrix dimensionality to 9 004 descriptors by applying the *RemoveUseless* filter that eliminates the attributes with none or too much variation for all the instances in the training set. A descriptor that cannot be calculated for an instance represents a missing value in the final vector. For a regression problem, this means that missing values need to be replaced by a number (mean, median, ...). Since this could add more noise to the dataset, we handled this issue by deleting all the attributes that contained at least one instance with a missing value, reducing the dataset to 1 477 attributes.

Then, we made use of the supervised method *CorrelationAttributeEval*, which orders all the attributes by their Pearson's correlation coefficient with the class. The highest correlation of an attribute with the class was 0.25, while the minimum was -0.32. We selected those attributes with correlation values between 0.1 <= *R* <= -0.1, which were 447 attributes.

Next, we applied the filter *InterquartileRange* implemented in Weka to identify instances with extreme values. This method flags a descriptor of an instance as extreme if its value is greater than the 75th quartile or if it is minor than the 25th quartile, by the product of an extreme value factor and the interquartile range. We kept the default value of the filter, extremeValuesFactor = 6, and we removed those instances with more than the 5% of the amount of attributes flagged as extreme cases. This way we reduced the training set to 648 instances with 447 attributes.

Finally, for obtaining the best subset of attributes to train the final model, we applied the *WrapperSubsetEval* technique for the selection of attributes. This supervised method evaluates subsets of attributes by training a classifier and assessing its performance in cross-validation. The classifier we used was Support Vector Machine for regression, SMOreg package of Weka, with the polynomial kernel. The selection of the subsets is held by a search method. Here, we employed a genetic algorithm with a population of 20 individuals, crossover and mutation probabilities of 0.6 and 0.033 respectively and 20 generations maximum. Each subset was evaluated in 5-folds cross-validation, and the selection of the best one was carried out attending to the correlation coefficient achieved by the classifier. This step reduced the dataset to 26 structural features to train and test our model. The list of final descriptors can be found in the Supporting Information ppro_project.idl file. The file can be directly uploaded in ProtDCal-Suite[41] to calculate the descriptors of protein-protein complexes.

Table SI-3.1. Descriptors of the protein-protein model.

The evaluation measures are the Pearson's correlation coefficient (*R*), the Spearman's rank correlation coefficient (R_S), and the Kendall's (R_K) rank correlation coefficient between each descriptor and the binding affinity values.

Descriptor	Description	*R*	R_S	R_K
wNc(ECI)_NO_AHR_G	Geometric mean (G) of the weighted number of contacts (wNc) of the common residues in Alfa Helix structure (AHR)	-0.20	-0.18	-0.12

wNc(ECI)_NO_ALR_G	Geometric mean (G) of the weighted number of contacts (wNc) of the aliphatic residues (ALR)	-0.17	-0.16	-0.11
wNc(Z2)_NO_PLR_V	Variance (V) of the weighted number of contacts (wNc) of the polar residues (PLR)	-0.11	-0.05	-0.03
wNc(Z2)_NO_PCR_G	Geometric mean (G) of the weighted number of contacts (wNc) of the positive charged residues (PCR)	-0.11	-0.11	-0.07
wNc(Z3)_NO_GLU_V	Variance (V) of the weighted number of contacts (wNc) of the glutamic acid residues (GLU)	0.12	0.13	0.09
wNc(Z3)_NO_PLR_P2	Potential mean (P2) of the weighted number of contacts (wNc) of the polar residues (PLR)	0.15	0.19	0.13
wNc(Z3)_NO_PRT_P2	Potential mean (P2) of the weighted number of contacts (wNc) of the whole protein (PRT)	0.16	0.20	0.14
wFLC(ECI)_NO_ILE_P2	Potential mean (P2) of the weighted fraction of local contacts (wFLC) of the isoleucine residues (ILE)	0.16	0.19	0.14
wFLC(IP)_NO_PCR_Ar	Arithmetic mean (Ar) of the weighted fraction of local contacts (wFLC) of the positive charged residues (PCR)	0.18	0.22	0.15
wFLC(IP)_NO_PCR_V	Variance (V) of the weighted fraction of local contacts (wFLC) of the positive charged residues (PCR)	0.12	0.14	0.11
wFLC(ISA)_NO_PLR_Ar	Arithmetic mean (Ar) of the weighted fraction of local contacts (wFLC) of the polar residues (PLR)	0.18	0.22	0.17
wNLC(ECI)_NO_AHR_V	Variance (V) of the weighted number of local contacts (wNLC) of the common residues in Alfa Helix structure (AHR)	0.15	0.16	0.11
wNLC(ECI)_NO_NPR_N1	Manhattan distance (N1) of the weighted number of local contacts (wNLC) of the nonpolar residues (NPR)	-0.15	-0.20	-0.13
wNLC(ECI)_NO_NPR_DE	Standard deviation (DE) of the weighted number of local contacts (wNLC) of the nonpolar residues (NPR)	-0.15	-0.16	-0.10
wNLC(IP)_NO_BSR_N1	Manhattan distance (N1) of the weighted number of local contacts (wNLC) of the common residues in Beta Sheet structure (BSR)	-0.14	-0.20	-0.13
wNLC(IP)_NO_PLR_N1	Manhattan distance (N1) of the weighted number of local contacts (wNLC) of the polar residues (PLR)	-0.12	-0.18	-0.12
wNLC(ISA)_NO_BSR_N2	Euclidean distance (N2) of the weighted number of local contacts (wNLC) of the common residues in Beta Sheet structure (BSR)	-0.13	-0.15	-0.10
wNLC(ISA)_NO_PLR_G	Geometric mean (G) of the weighted number of local contacts (wNLC) of the polar residues (PLR)	0.15	0.16	0.11
wNLC(Z1)_NO_AHR_DE	Standard deviation (DE) of the weighted number of local contacts (wNLC) of the common residues in Alfa Helix structure (AHR)	0.13	0.15	0.10

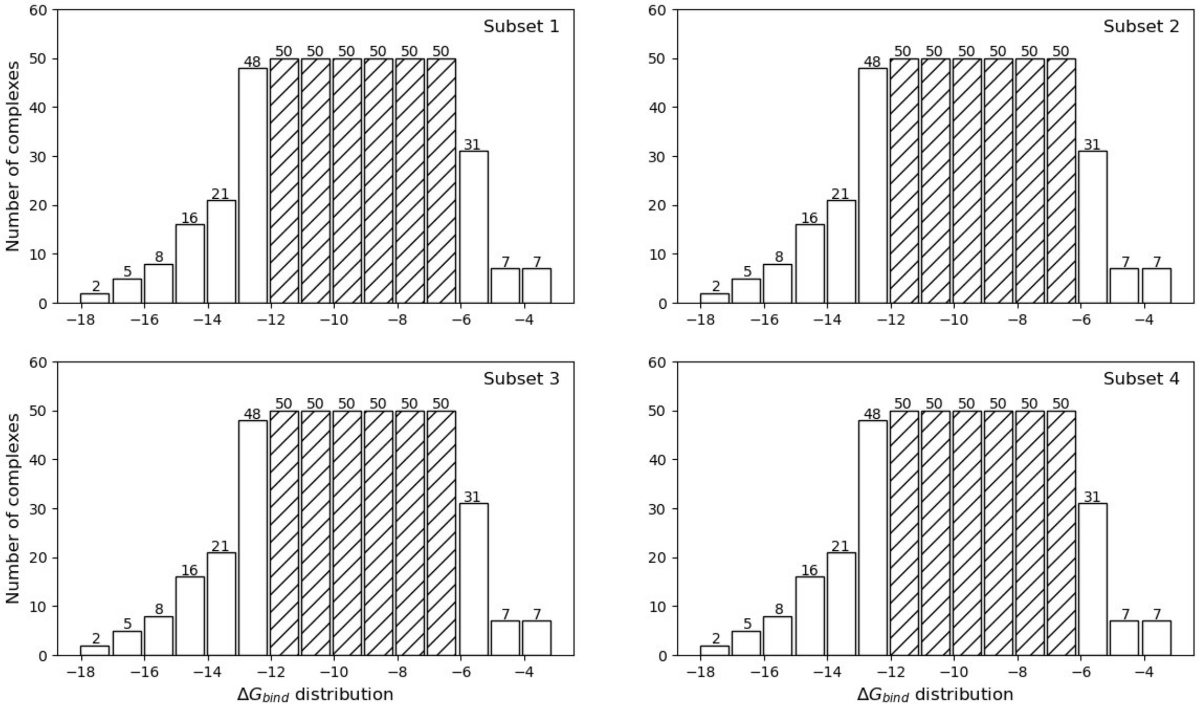
wNLC(Z1)_NO_ALR_N2	Euclidean distance (N2) of the weighted number of local contacts (wNLC) of the aliphatic residues (ALR)	-0.13	-0.13	-0.09
wNLC(Z1)_NO_NCR_P2	Potential mean (P2) of the weighted number of local contacts (wNLC) of the negative charged residues (NCR)	-0.17	-0.15	-0.10
wNLC(Z1)_NO_PRT_Ar	Arithmetic mean (Ar) of the weighted number of local contacts (wNLC) of the whole protein (PRT)	-0.16	-0.15	-0.10
wNLC(Z2)_NO_BSR_G	Geometric mean (G) of the weighted number of local contacts (wNLC) of the common residues in Beta Sheet structure (BSR)	-0.13	-0.12	-0.08
wNLC(Z2)_NO_ALR_N1	Manhattan distance (N1) of the weighted number of local contacts (wNLC) of the aliphatic residues (ALR)	-0.32	-0.29	-0.20
wNLC(Z3)_NO_AHR_Ar	Arithmetic mean (Ar) of the weighted number of local contacts (wNLC) of the common residues in Alfa Helix structure (AHR)	0.25	0.25	0.16
wNLC(Z3)_NO_ALR_Ar	Arithmetic mean (Ar) of the weighted number of local contacts (wNLC) of the aliphatic residues (ALR)	0.14	0.13	0.09

The weights of the descriptors are:

- IP: Isoelectric Point
- ECI: Electronic Charge Index
- ISA: Isotropic Surface Area
- Z1: Combined measure of hydrophobicity related properties
- Z2: Combined measure of bulkiness related properties
- Z3: Combined measure of electron related properties

File SI-1 (separated). Configuration file for ProtDCal to compute the 26 structural descriptors.

Figure SI-2. Distribution of ΔG_{bind} values in the four subsets of the training data for the protein-protein ensemble learning protocol.



We defined 15 intervals in the scale of ΔG_{bind} values, according to the range in the entire data. We filled the intervals with a maximum of 50 instances by sampling (without replacement) the entire dataset. We iterated this procedure to create four subsets, each one containing 445 complexes distributed along the complete range of affinity values. The white bars denote the intervals that were repeated among the four datasets, while the stripped bars denote those intervals where the sampling was performed.

Table SI-3. Summary of the intermediate models and performance measures for the protein-protein BA modeling during the hyperparameters tuning process.

	Training subset 1				***Training subset 2***				***Training subset 3***				***Training subset 4***			
Model	**TS**	**CV**	**DEV**	**Score**	**TS**	**CV**	**DEV**	**Score**	**TS**	**CV**	**DEV**	**Score**	**TS**	**CV**	**DEV**	**Score**
PK_C0.03125_D1	0,514	0,476	0,343	0,167	0,494	0,441	0,331	0,148	0,517	0,479	0,406	0,205	0,498	0,434	0,372	0,169
PK_C0.03125_D2	0,582	0,476	0,429	0,211	0,586	0,455	0,468	0,222	0,588	0,509	0,493	0,265	0,578	0,461	0,459	0,222
PK_C0.03125_D3	0,702	0,460	0,426	0,145	0,696	0,418	0,484	0,161	0,709	0,490	0,497	0,227	0,684	0,385	0,464	0,124
PK_C0.04419_D1	0,525	0,486	0,369	0,186	0,503	0,446	0,342	0,156	0,527	0,493	0,426	0,221	0,508	0,450	0,395	0,187
PK_C0.04419_D2	0,597	0,483	0,427	0,211	0,599	0,464	0,486	0,235	**0,602**	**0,508**	**0,495**	**0,266**	0,588	0,468	0,470	0,230
PK_C0.04419_D3	0,720	0,461	0,440	0,146	0,713	0,409	0,479	0,138	0,725	0,486	0,492	0,211	0,699	0,374	0,446	0,087
PK_C0.0625_D1	0,534	0,495	0,385	0,198	0,512	0,452	0,372	0,175	0,534	0,501	0,433	0,228	0,514	0,462	0,400	0,195
PK_C0.0625_D2	0,611	0,490	0,419	0,206	0,614	0,463	0,492	0,236	0,618	0,507	0,488	0,260	0,598	0,470	0,476	0,233
PK_C0.0625_D3	0,739	0,462	0,455	0,147	0,731	0,400	0,471	0,108	0,745	0,480	0,469	0,173	0,715	0,365	0,442	0,060
PK_C0.08838_D1	0,542	0,492	0,411	0,212	0,522	0,458	0,392	0,188	0,542	0,505	0,442	0,235	0,520	0,470	0,405	0,201
PK_C0.08838_D2	0,627	0,494	0,419	0,202	0,632	0,457	0,501	0,233	0,633	0,514	0,481	0,257	0,611	0,461	0,478	0,227
PK_C0.08838_D3	0,759	0,465	0,462	0,142	0,751	0,390	0,473	0,082	0,764	0,470	0,453	0,133	0,732	0,358	0,445	0,041
PK_C0.125_D1	0,551	0,493	0,417	0,216	0,531	0,463	0,409	0,199	0,547	0,508	0,442	0,237	0,524	0,474	0,406	0,202
PK_C0.125_D2	0,645	0,494	0,416	0,193	0,649	0,455	0,495	0,223	0,649	0,516	0,482	0,256	0,627	0,453	0,479	0,217
PK_C0.125_D3	0,778	0,466	0,464	0,129	0,774	0,383	0,462	0,041	0,784	0,461	0,428	0,080	0,750	0,354	0,444	0,016
PK_C0.17677_D1	0,556	0,501	0,410	0,215	0,539	0,473	0,417	0,207	0,551	0,511	0,448	0,241	0,530	0,470	0,423	0,210
PK_C0.17677_D2	0,665	0,494	0,411	0,180	0,661	0,458	0,493	0,219	0,664	0,514	0,486	0,254	0,641	0,444	0,471	0,201
PK_C0.17677_D3	0,796	0,462	0,452	0,094	0,794	0,377	0,457	0,007	0,805	0,450	0,409	0,026	0,770	0,342	0,450	-0,015
PK_C0.25_D1	0,562	0,506	0,409	0,216	0,547	0,476	0,422	0,212	0,555	0,512	0,455	0,246	0,533	0,468	0,438	0,217
PK_C0.25_D2	0,682	0,496	0,408	0,169	0,672	0,458	0,483	0,208	0,675	0,512	0,485	0,248	0,655	0,435	0,465	0,185
PK_C0.25_D3	0,814	0,454	0,442	0,057	0,815	0,357	0,452	-0,049	0,823	0,434	0,398	-0,028	0,792	0,327	0,435	-0,075
PK_C0.35355_D1	0,565	0,513	0,427	0,230	0,554	0,487	0,432	0,222	0,561	0,514	0,468	0,253	0,536	0,465	0,452	0,222
PK_C0.35355_D2	0,697	0,495	0,411	0,163	0,684	0,450	0,475	0,190	0,686	0,505	0,483	0,237	0,663	0,415	0,459	0,159
PK_C0.35355_D3	0,833	0,444	0,438	0,020	0,834	0,337	0,439	-0,115	0,842	0,411	0,394	-0,085	0,817	0,318	0,430	-0,126
PK_C0.5_D1	0,570	0,521	0,438	0,240	0,560	0,493	0,454	0,236	0,567	0,519	0,470	0,257	0,540	0,462	0,461	0,226
PK_C0.5_D2	0,710	0,490	0,424	0,163	0,695	0,436	0,475	0,171	0,698	0,501	0,483	0,229	0,672	0,394	0,452	0,130

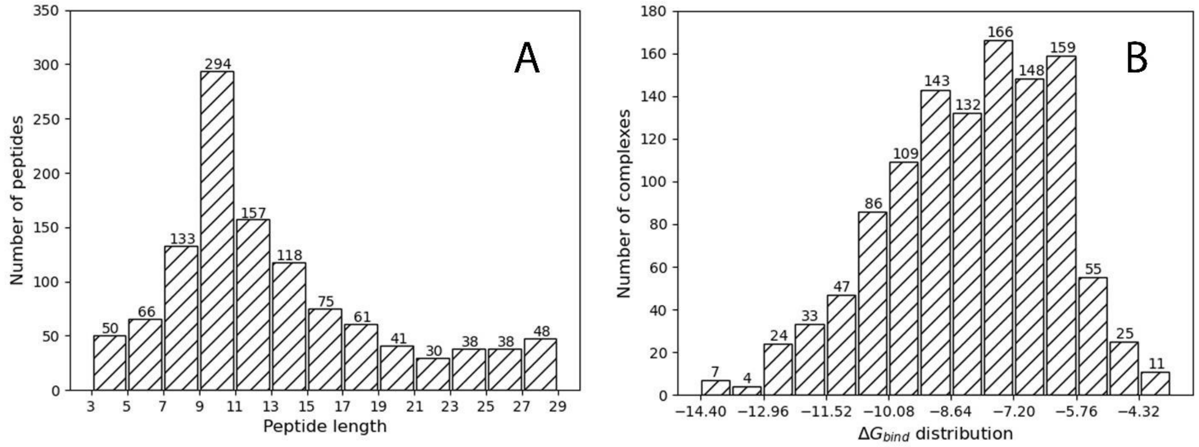
PK_C0.5_D3	0,852	0,439	0,422	-0,031	0,854	0,326	0,420	-0,182	0,859	0,393	0,376	-0,156	0,843	0,313	0,424	-0,179
PK_C0.7071_D1	0,574	0,525	0,449	0,248	0,566	0,494	0,456	0,238	0,573	0,522	0,471	0,259	0,544	0,463	0,451	0,221
PK_C0.7071_D2	0,723	0,481	0,437	0,158	0,705	0,424	0,487	0,163	0,710	0,495	0,467	0,205	0,683	0,374	0,450	0,102
PK_C0.7071_D3	0,870	0,434	0,407	-0,079	0,872	0,316	0,394	-0,260	0,877	0,377	0,369	-0,216	0,866	0,293	0,415	-0,256
PK_C1.41421_D1	0,581	0,526	0,471	0,262	0,574	0,505	0,475	0,254	0,580	0,524	0,471	0,261	0,553	0,469	0,467	0,232
PK_C1.41421_D2	0,751	0,472	0,448	0,140	0,733	0,394	0,493	0,119	0,739	0,476	0,416	0,122	0,702	0,354	0,427	0,048
PK_C1.41421_D3	0,901	0,412	0,384	-0,188	0,906	0,297	0,352	-0,409	0,912	0,357	0,338	-0,352	0,901	0,255	0,370	-0,441
PK_C1_D1	0,577	0,527	0,466	0,259	0,570	0,500	0,460	0,243	0,577	0,524	0,472	0,261	0,550	0,466	0,462	0,228
PK_C1_D2	0,737	0,476	0,448	0,155	0,719	0,408	0,501	0,150	0,725	0,488	0,442	0,168	0,691	0,363	0,436	0,075
PK_C1_D3	0,886	0,429	0,399	-0,120	0,889	0,310	0,373	-0,326	0,894	0,371	0,357	-0,270	0,885	0,275	0,394	-0,346
PK_C11.3137_D1	0,588	0,532	0,486	0,273	**0,581**	**0,507**	**0,500**	**0,268**	0,586	0,523	0,472	0,261	**0,560**	**0,466**	**0,483**	**0,238**
PK_C11.3137_D2	0,810	0,403	0,365	-0,087	0,811	0,268	0,406	-0,214	0,804	0,387	0,240	-0,266	0,788	0,298	0,396	-0,149
PK_C11.3137_D3	0,981	0,309	0,224	-0,770	0,984	0,235	0,091	-1,167	0,991	0,238	0,184	-0,995	0,981	0,180	0,271	-0,917
PK_C16_D1	0,589	0,529	0,491	0,274	0,581	0,505	0,500	0,267	0,585	0,524	0,472	0,261	0,560	0,464	0,483	0,237
PK_C16_D2	0,818	0,382	0,360	-0,130	0,820	0,256	0,391	-0,264	0,812	0,363	0,199	-0,368	0,796	0,286	0,381	-0,195
PK_C16_D3	0,989	0,296	0,218	-0,826	0,991	0,225	0,059	-1,274	0,996	0,224	0,159	-1,085	0,989	0,163	0,231	-1,045
PK_C2.82842_D1	0,586	0,528	0,487	0,271	0,579	0,511	0,489	0,264	0,587	0,524	0,475	0,263	0,558	0,467	0,473	0,234
PK_C2.82842_D2	0,772	0,451	0,410	0,062	0,755	0,367	0,454	0,035	0,764	0,450	0,375	0,030	0,729	0,336	0,443	0,016
PK_C2.82842_D3	0,933	0,368	0,330	-0,387	0,938	0,272	0,295	-0,604	0,941	0,335	0,283	-0,532	0,932	0,215	0,346	-0,609
PK_C2_D1	0,584	0,529	0,486	0,271	0,574	0,509	0,480	0,258	0,586	0,525	0,473	0,262	0,555	0,469	0,467	0,232
PK_C2_D2	0,762	0,466	0,435	0,113	0,745	0,380	0,469	0,073	0,752	0,463	0,390	0,072	0,715	0,347	0,432	0,032
PK_C2_D3	0,918	0,387	0,361	-0,283	0,923	0,282	0,326	-0,507	0,929	0,343	0,305	-0,457	0,916	0,235	0,357	-0,524
PK_C4_D1	0,587	0,533	0,488	0,275	0,582	0,511	0,492	0,266	0,586	0,527	0,470	0,262	0,559	0,465	0,472	0,232
PK_C4_D2	0,784	0,440	0,382	0,007	0,767	0,344	0,442	-0,017	0,776	0,436	0,349	-0,027	0,746	0,330	0,443	-0,009
PK_C4_D3	0,946	0,350	0,305	-0,481	0,951	0,258	0,246	-0,742	0,955	0,315	0,266	-0,622	0,946	0,208	0,339	-0,665
PK_C5.65685_D1	**0,588**	**0,533**	**0,488**	**0,275**	0,580	0,506	0,495	0,265	0,588	0,525	0,474	0,263	0,560	0,468	0,474	0,235
PK_C5.65685_D2	0,796	0,428	0,364	-0,041	0,781	0,313	0,428	-0,087	0,787	0,421	0,323	-0,089	0,761	0,321	0,439	-0,040
PK_C5.65685_D3	0,958	0,342	0,268	-0,583	0,964	0,249	0,199	-0,875	0,968	0,291	0,238	-0,743	0,959	0,198	0,329	-0,731
PK_C8_D1	0,588	0,533	0,486	0,274	0,581	0,507	0,495	0,266	0,587	0,525	0,478	0,265	0,560	0,468	0,480	0,238
PK_C8_D2	0,803	0,414	0,368	-0,061	0,798	0,284	0,424	-0,152	0,796	0,403	0,288	-0,168	0,774	0,309	0,419	-0,091
PK_C8_D3	0,970	0,327	0,237	-0,692	0,974	0,242	0,138	-1,032	0,981	0,260	0,214	-0,874	0,970	0,192	0,308	-0,804

Table SI-4. Summary of the performance of the individual and the ensemble models for protein-protein models.

The models M1, M2, M3 and M4 correspond to the best predictors obtained from the training subsets 1, 2, 3 and 4 respectively. Then, the correlation coefficients (R) of the estimations in the development set were calculated for each model (R_IND), as well as all possible combinations of the models. The combination rules were the average (V_AVG), maximum (V_MAX), and minimum, (V_MIN) predictions. The optimal ensemble model corresponds to the model that outputs the binding affinity based on the minimum predicted value between the models obtained from the training subsets 2 and 3.

Ensemble	*Model*	*R_IND*	*V_AVG*	*V_MAX*	*V_MIN*
1	M1	0,488	0,501	0,508	0,489
	M2	0,500			
2	M1	0,488	0,505	0,479	0,520
	M3	0,495			
3	M1	0,488	0,491	0,482	0,497
	M4	0,482			
4	M2	0,500	0,508	0,480	**0,526**
	M3	0,495			
5	M2	0,500	0,496	0,496	0,493
	M4	0,482			
6	M3	0,495	0,500	0,483	0,508
	M4	0,482			
7	M1	0,488	0,508	0,482	0,517
	M2	0,500			
	M3	0,495			
8	M1	0,488	0,498	0,496	0,492
	M2	0,500			
	M4	0,482			
9	M1	0,488	0,502	0,479	
	M3	0,495			
	M4	0,482			
10	M2	0,500	0,505	0,473	0,507
	M3	0,495			
	M4	0,482			
11	M1	0,488	0,505	0,479	0,508
	M2	0,500			
	M3	0,495			
	M4	0,482			

Figure SI-3. Characterization of the dataset of protein-peptide complexes used in this work.



The distribution of the peptide lengths (number of residues) in the dataset is presented in panel A, while panel B shows the distribution of ΔG values across all the protein-peptide complexes contained in the dataset.

Section SI-4. Description of the process of feature selection for creating the protein-peptide model.

Here, we followed the pipeline described in section SI-2 for selecting the attributes of the final model. Starting with a matrix of 949 instances x 23 040 descriptors generated with ProtDCal, we reduced the matrix dimensionality to 8 999 attributes by applying the filter *RemoveUseless*. Then, we deleted all the attributes that contained at least one instance with a missing value, reducing the dataset to 2 358 attributes. In a third step, we made use of the supervised method *CorrelationAttributeEval,* for correlating each attribute with the class. The highest correlation obtained was 0.24 and the minimum -0.23. We selected those attributes with correlation values between 0.1 <= *R* <= -0.1 that were 631 attributes.

At this point, after decreasing the dimensionality of the problem by more than 95%, we applied the filter *InterquartileRange* implemented in Weka to identify instances with extreme values. We kept the default value of the filter, which is the extremeValuesFactor = 6 and we removed those instances with more than the 5% of number of attributes flagged as extreme cases. This way we reduced the training set to 922 instances with 631 attributes.

Finally, for obtaining the best subset of attributes to train the final model we applied the *WrapperSubsetEval* attribute selection technique. The classifier we used was support vector machine for regression with the lineal kernel and the search method a genetic algorithm with a population of 20 individuals, crossover and mutation probabilities of 0.6 and 0.033 respectively and 20 generations maximum. Each subset was evaluated in 5-folds cross-validation, and the selection of the best subset was attending to the correlation coefficient achieved by the classifier. This step reduced the dataset to 37 structural features to train and test the models.

Table SI-4.1. Descriptors of the protein-peptide model.

The evaluation measures are the Pearson's correlation coefficient (*R*), the Spearman's rank correlation coefficient (R_S), and the Kendall's (R_K) rank correlation coefficient between each descriptor and the binding affinity values.

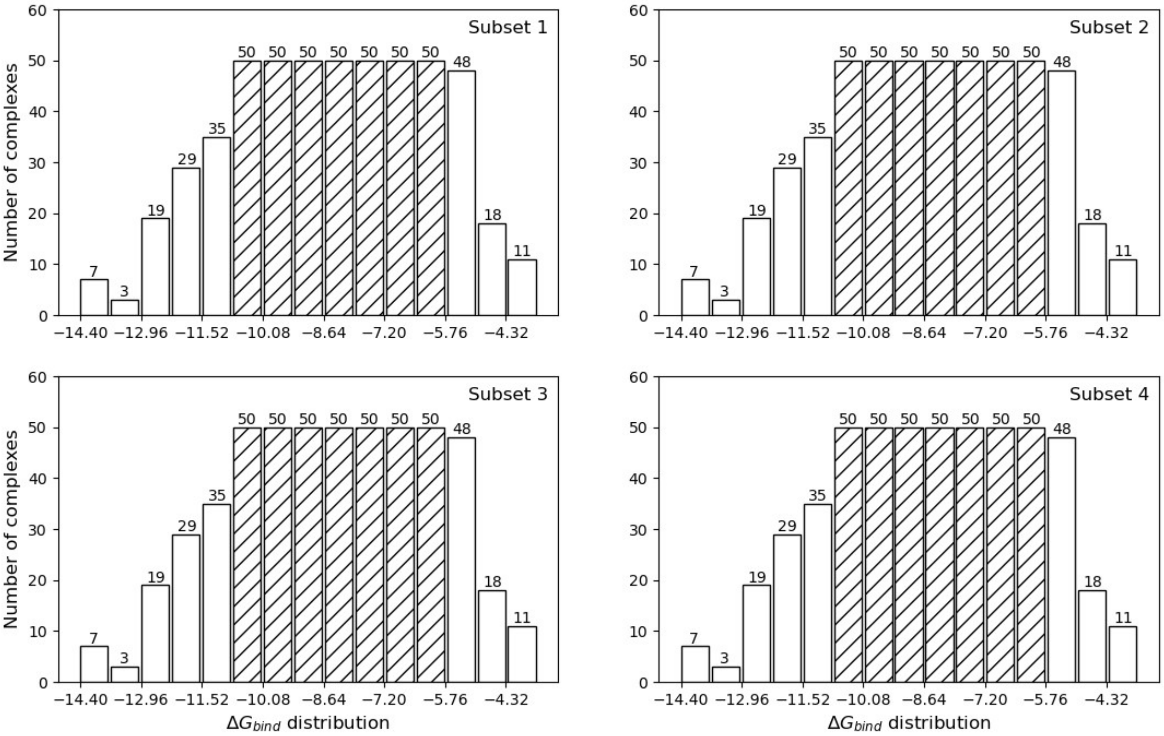
Descriptor	Description	*R*	R_S	R_K
wNc(ECI)_NO_AHR_N1	Manhattan distance (N1) of the weighted number of contacts (wNc) of the common residues in Alfa Helix structure (AHR)	-0.20	-0.17	-0.11
wNc(ECI)_NO_ALR_N1	Manhattan distance (N1) of the weighted number of contacts (wNc) of the aliphatic residues (ALR)	-0.13	-0.05	-0.04
wNc(IP)_NO_PLR_N1	Manhattan distance (N1) of the weighted number of contacts (wNc) of the polar residues (PLR)	-0.13	-0.12	-0.08
wNc(ISA)_NO_PLR_V	Variance (V) of the weighted number of contacts (wNc) of the polar residues (PLR)	0.10	0.10	0.06
wNc(Z1)_NO_NPR_N2	Euclidean distance (N2) of the weighted number of contacts (wNc) of the nonpolar residues (NPR)	-0.19	-0.17	-0.12

wNc(Z1)_NO_ARM_V	Variance (V) of the weighted number of contacts (wNc) of the aromatic residues (ARM)	-0.11	-0.05	-0.04
wNc(Z1)_NO_PRT_N1	Manhattan distance (N1) of the weighted number of contacts (wNc) of the whole protein (PRT)	-0.12	-0.11	-0.07
wNc(Z3)_NO_AHR_N1	Manhattan distance (N1) of the weighted number of contacts (wNc) of the common residues in Alfa Helix structure (AHR)	-0.17	-0.11	-0.07
wNc(Z3)_NO_UCR_N1	Manhattan distance (N1) of the weighted number of contacts (wNc) of the uncharged residues (UCR)	-0.13	-0.07	-0.04
wFLC(ECI)_NO_AHR_SI30	Standardized Information Content (SI30) of the weighted fraction of local contacts (wFLC) of the common residues in Alfa Helix structure (AHR)	0.16	0.15	0.10
wFLC(ECI)_NO_PCR_V	Variance (V) of the weighted fraction of local contacts (wFLC) of the positive charged residues (PCR)	0.11	0.09	0.07
wFLC(IP)_NO_RTR_MI30	Mean Information Content (MI30) of the weighted fraction of local contacts (wFLC) of the common residues in reverse turn (RTR)	0.12	0.07	0.05
wFLC(IP)_NO_BSR_Ar	Arithmetic mean (Ar) of the weighted fraction of local contacts (wFLC) of the common residues in Beta Sheet structure (BSR)	0.11	0.17	0.12
wFLC(IP)_NO_UCR_RA	Range (RA) of the weighted fraction of local contacts (wFLC) of the uncharged residues (UCR)	0.13	0.12	0.09
wFLC(IP)_NO_UCR_K	Kurtosis (K) of the weighted fraction of local contacts (wFLC) of the uncharged residues (UCR)	-0.17	-0.21	-0.14
wFLC(IP)_NO_PRT_TI30	Total content (TI30) of the weighted fraction of local contacts (wFLC) of the whole protein (PRT)	-0.20	-0.22	-0.15
wFLC(ISA)_NO_NPR_P2	Potential mean (P2) of the weighted fraction of local contacts (wFLC) of the nonpolar residues (NPR)	0.24	0.26	0.18
wFLC(ISA)_NO_NPR_DE	Standard deviation (DE) of the weighted fraction of local contacts (wFLC) of the nonpolar residues (NPR)	0.24	0.27	0.18
wFLC(ISA)_NO_PRT_TI30	Total content (TI30) of the weighted fraction of local contacts (wFLC) of the whole protein (PRT)	-0.19	-0.20	-0.13
wFLC(Z1)_NO_BSR_SI30	Standardized Information Content (SI30) of the weighted fraction of local contacts (wFLC) of the common residues in Beta Sheet structure (BSR)	0.20	0.22	0.14
wFLC(Z2)_NO_PLR_K	Kurtosis (K) of the weighted fraction of local contacts (wFLC) of the polar residues (PLR)	-0.12	-0.15	-0.10
wNLC(ECI)_NO_BSR_I50	Interquartile range (I50) of the weighted number of local contacts (wNLC) of the	-0.10	-0.09	-0.06

	common residues in Beta Sheet structure (BSR)			
wNLC(ECI)_NO_PLR_K	Kurtosis (K) of the weighted number of local contacts (wNLC) of the polar residues (PLR)	-0.12	-0.17	-0.11
wNLC(ECI)_NO_UCR_S	Skewness (S) of the weighted number of local contacts (wNLC) of the uncharged residues (UCR)	-0.14	-0.13	-0.09
wNLC(IP)_NO_RTR_V	Variance (V) of the weighted number of local contacts (wNLC) of the common residues in reverse turn (RTR)	0.11	0.09	0.06
wNLC(IP)_NO_UCR_I50	Interquartile range (I50) of the weighted number of local contacts (wNLC) of the uncharged residues (UCR)	-0.12	-0.09	-0.06
wNLC(ISA)_NO_UCR_S	Skewness (S) of the weighted number of local contacts (wNLC) of the uncharged residues (UCR)	-0.20	-0.22	-0.14
wNLC(Z1)_NO_RTR_SI30	Standardized Information Content (SI30) of the weighted number of local contacts (wNLC) of the common residues in reverse turn (RTR)	0.12	0.13	0.09
wNLC(Z1)_NO_BSR_K	Kurtosis (K) of the weighted number of local contacts (wNLC) of the common residues in Beta Sheet structure (BSR)	-0.10	-0.17	-0.11
wNLC(Z1)_NO_UCR_K	Kurtosis (K) of the weighted number of local contacts (wNLC) of the uncharged residues (UCR)	-0.13	-0.17	-0.11
wNLC(Z2)_NO_AHR_P2	Potential mean (P2) of the weighted number of local contacts (wNLC) of the common residues in Alfa Helix structure (AHR)	-0.17	-0.19	-0.13
wNLC(Z2)_NO_PCR_N2	Euclidean distance (N2) of the weighted number of local contacts (wNLC) of the positive charged residues (PCR)	-0.15	-0.15	-0.10
wNLC(Z2)_NO_NCR_P2	Potential mean (P2) of the weighted number of local contacts (wNLC) of the negative charged residues (NCR)	-0.11	-0.09	-0.06
wNLC(Z2)_NO_UCR_K	Kurtosis (K) of the weighted number of local contacts (wNLC) of the uncharged residues (UCR)	-0.11	-0.17	-0.12
wNLC(Z3)_NO_PLR_SI30	Standardized Information Content (SI30) of the weighted number of local contacts (wNLC) of the polar residues (PLR)	0.12	0.13	0.08
wNLC(Z3)_NO_NCR_RA	Range (RA) of the weighted number of local contacts (wNLC) of the negative charged residues (NCR)	-0.13	-0.15	-0.10
wNLC(Z3)_NO_UCR_TI30	Total content (TI) of the weighted number of local contacts (wNLC) of the uncharged residues (UCR)	-0.19	-0.20	-0.14

File SI-2 (separated). Configuration file for ProtDCal to compute the 37 structural descriptors.

Figure SI-4. Distribution of ΔG_{bind} values in the four subsets of the training data for the protein-peptide ensemble learning protocol.



We defined 15 intervals in the scale of ΔG_{bind} values, according to the rage in the entire data. Then, we filled the intervals with a maximum of 50 instances by sampling (without replacement) the entire dataset. We iterated this procedure to create four subsets, each one containing 520 complexes distributed along the complete range of affinity values. The white bars denote the intervals that were repeated among the four datasets, while the stripped bars denote those intervals where the sampling was performed.

Table SI-5. Summary of the intermediate models and performance measures for the protein-peptide modeling during the hyperparameters tuning process.

	Training 1				***Training 2***				***Training 3***				***Training 4***			
Model	**TS**	**CV**	**DEV**	**Score**	**TS**	**CV**	**DEV**	**Score**	**TS**	**CV**	**DEV**	**Score**	**TS**	**CV**	**DEV**	**Score**
PK_C0.0078125_D1	0,503	0,463	0,481	0,230	0,480	0,441	0,406	0,189	0,494	0,448	0,430	0,203	0,494	0,464	0,412	0,201
PK_C0.0078125_D2	0,636	0,574	0,529	0,321	0,625	0,549	0,539	0,313	0,630	0,550	0,519	0,302	0,639	0,577	0,505	0,307
PK_C0.0078125_D3	0,735	0,586	0,465	0,259	0,728	0,535	0,506	0,261	0,726	0,549	0,556	0,313	0,722	0,539	0,555	0,305
PK_C0.0110485_D1	0,529	0,486	0,495	0,250	0,507	0,468	0,437	0,215	0,522	0,468	0,449	0,222	0,517	0,484	0,440	0,223
PK_C0.0110485_D2	0,644	0,578	0,527	0,322	0,637	0,557	0,546	0,322	0,642	0,557	0,525	0,309	0,648	0,578	0,518	0,316
PK_C0.0110485_D3	0,755	0,586	0,467	0,251	0,746	0,528	0,490	0,233	0,748	0,538	0,558	0,297	0,740	0,529	0,550	0,287
PK_C0.015625_D1	0,555	0,510	0,499	0,267	0,531	0,489	0,463	0,238	0,549	0,492	0,452	0,235	0,545	0,502	0,462	0,244
PK_C0.015625_D2	**0,654**	**0,581**	**0,527**	**0,324**	**0,650**	**0,564**	**0,539**	**0,322**	0,653	0,562	0,533	0,317	0,657	0,578	0,524	0,320
PK_C0.015625_D3	0,777	0,581	0,468	0,237	0,764	0,524	0,481	0,209	0,770	0,526	0,565	0,283	0,760	0,521	0,550	0,271
PK_C0.022097_D1	0,575	0,532	0,503	0,281	0,555	0,504	0,485	0,257	0,573	0,509	0,453	0,243	0,569	0,523	0,479	0,264
PK_C0.022097_D2	0,664	0,583	0,521	0,320	0,661	0,564	0,537	0,320	0,666	0,563	0,538	0,320	0,665	0,576	0,526	0,320
PK_C0.022097_D3	0,798	0,571	0,472	0,218	0,786	0,521	0,467	0,178	0,792	0,512	0,563	0,256	0,782	0,513	0,565	0,264
PK_C0.03125_D1	0,592	0,542	0,507	0,290	0,574	0,523	0,492	0,271	0,586	0,526	0,467	0,259	0,587	0,539	0,478	0,272
PK_C0.03125_D2	0,678	0,585	0,521	0,320	0,671	0,561	0,535	0,316	**0,678**	**0,562**	**0,543**	**0,322**	**0,675**	**0,570**	**0,532**	**0,320**
PK_C0.03125_D3	0,818	0,562	0,470	0,193	0,807	0,517	0,465	0,156	0,813	0,497	0,565	0,229	0,802	0,502	0,575	0,250
PK_C0.0441941_D1	0,601	0,551	0,504	0,292	0,584	0,535	0,502	0,282	0,596	0,537	0,465	0,263	0,595	0,548	0,477	0,275
PK_C0.0441941_D2	0,694	0,588	0,517	0,317	0,685	0,559	0,540	0,317	0,692	0,561	0,546	0,321	0,685	0,563	0,537	0,317
PK_C0.0441941_D3	0,837	0,554	0,448	0,144	0,830	0,512	0,475	0,139	0,835	0,492	0,551	0,194	0,821	0,487	0,587	0,234
PK_C0.0625_D1	0,605	0,556	0,500	0,293	0,592	0,543	0,515	0,294	0,600	0,543	0,455	0,259	0,602	0,551	0,490	0,285
PK_C0.0625_D2	0,713	0,591	0,516	0,314	0,701	0,555	0,536	0,308	0,707	0,555	0,543	0,312	0,698	0,553	0,546	0,315
PK_C0.0625_D3	0,856	0,542	0,423	0,082	0,853	0,507	0,466	0,102	0,854	0,493	0,544	0,170	0,840	0,471	0,585	0,199
PK_C0.0883883_D1	0,607	0,558	0,505	0,297	0,597	0,547	0,515	0,297	0,604	0,545	0,460	0,263	0,612	0,556	0,484	0,284
PK_C0.0883883_D2	0,732	0,598	0,511	0,310	0,718	0,547	0,505	0,274	0,725	0,547	0,537	0,297	0,714	0,543	0,550	0,306
PK_C0.0883883_D3	0,874	0,525	0,436	0,060	0,874	0,498	0,439	0,033	0,873	0,493	0,508	0,112	0,858	0,460	0,579	0,163
PK_C0.125_D1	0,610	0,561	0,505	0,299	0,605	0,557	0,504	0,296	0,607	0,545	0,476	0,274	0,617	0,563	0,471	0,279
PK_C0.125_D2	0,749	0,602	0,502	0,299	0,734	0,540	0,472	0,232	0,742	0,541	0,526	0,277	0,730	0,528	0,537	0,280

PK_C0.125_D3	0,893	0,512	0,488	0,088	0,895	0,486	0,413	-0,041	0,891	0,492	0,479	0,056	0,877	0,456	0,553	0,113
PK_C0.1767766_D1	0,613	0,564	0,508	0,302	0,610	0,562	0,500	0,296	0,614	0,548	0,489	0,282	0,621	0,575	0,465	0,280
PK_C0.1767766_D2	0,767	0,596	0,488	0,273	0,751	0,537	0,450	0,199	0,759	0,535	0,503	0,244	0,746	0,515	0,537	0,263
PK_C0.1767766_D3	0,911	0,494	0,537	0,106	0,914	0,473	0,404	-0,098	0,907	0,484	0,480	0,028	0,895	0,454	0,466	-0,013
PK_C0.25_D1	0,615	0,562	0,508	0,301	0,613	0,565	0,500	0,297	0,617	0,550	0,487	0,283	0,623	0,577	0,467	0,282
PK_C0.25_D2	0,786	0,590	0,480	0,251	0,766	0,535	0,435	0,171	0,778	0,524	0,501	0,221	0,763	0,501	0,540	0,244
PK_C0.25_D3	0,926	0,477	0,525	0,049	0,931	0,460	0,382	-0,174	0,925	0,469	0,524	0,039	0,911	0,446	0,413	-0,116
PK_C0.3535533_D1	0,616	0,561	0,503	0,298	0,613	0,565	0,505	0,301	0,620	0,554	0,488	0,285	0,624	0,573	0,472	0,284
PK_C0.3535533_D2	0,805	0,582	0,464	0,214	0,784	0,527	0,432	0,148	0,800	0,511	0,525	0,216	0,781	0,485	0,554	0,229
PK_C0.3535533_D3	0,940	0,463	0,513	-0,002	0,944	0,454	0,366	-0,229	0,943	0,447	0,618	0,095	0,927	0,437	0,374	-0,210
PK_C0.5_D1	0,619	0,564	0,501	0,298	0,612	0,562	0,501	0,297	0,622	0,554	0,493	0,289	0,624	0,569	0,476	0,285
PK_C0.5_D2	0,821	0,571	0,459	0,187	0,803	0,518	0,439	0,130	0,817	0,499	0,556	0,220	0,798	0,469	0,557	0,204
PK_C0.5_D3	0,951	0,451	0,526	-0,018	0,954	0,453	0,340	-0,290	0,956	0,428	0,660	0,098	0,941	0,428	0,350	-0,285
PK_C0.7071067_D1	0,620	0,569	0,498	0,298	0,613	0,560	0,502	0,297	0,623	0,554	0,494	0,289	0,626	0,566	0,490	0,292
PK_C0.7071067_D2	0,835	0,562	0,447	0,154	0,822	0,506	0,447	0,108	0,833	0,493	0,579	0,223	0,813	0,451	0,567	0,181
PK_C0.7071067_D3	0,962	0,434	0,500	-0,093	0,963	0,447	0,298	-0,385	0,967	0,408	0,612	-0,001	0,955	0,410	0,322	-0,381
PK_C1.4142135_D1	0,622	0,568	0,497	0,297	0,614	0,554	0,496	0,290	0,625	0,549	0,503	0,292	0,630	0,563	0,505	0,300
PK_C1.4142135_D2	0,864	0,532	0,441	0,086	0,858	0,487	0,421	0,018	0,860	0,486	0,559	0,174	0,843	0,418	0,577	0,125
PK_C1.4142135_D3	0,980	0,390	0,507	-0,179	0,974	0,430	0,261	-0,498	0,983	0,370	0,586	-0,115	0,973	0,372	0,298	-0,517
PK_C1_D1	0,622	0,569	0,498	0,299	0,612	0,557	0,501	0,295	0,624	0,549	0,503	0,292	0,628	0,564	0,497	0,296
PK_C1_D2	0,850	0,549	0,436	0,113	0,841	0,497	0,445	0,078	0,847	0,494	0,580	0,215	0,827	0,430	0,580	0,157
PK_C1_D3	0,972	0,413	0,516	-0,119	0,969	0,444	0,270	-0,450	0,976	0,386	0,578	-0,088	0,965	0,387	0,305	-0,463
PK_C11.3137084_D1	0,623	0,559	0,480	0,282	0,615	0,552	0,481	0,280	0,624	0,551	0,502	0,292	0,633	0,563	0,503	0,299
PK_C11.3137084_D2	0,935	0,382	0,504	-0,124	0,944	0,391	0,417	-0,243	0,939	0,372	0,453	-0,214	0,931	0,361	0,482	-0,178
PK_C11.3137084_D3	0,994	0,336	0,431	-0,406	0,991	0,336	0,236	-0,728	0,999	0,288	0,542	-0,341	0,992	0,297	0,480	-0,397
PK_C16_D1	0,624	0,558	0,477	0,280	0,615	0,551	0,482	0,280	0,624	0,552	0,505	0,294	0,633	0,563	0,503	0,299
PK_C16_D2	0,947	0,349	0,493	-0,208	0,953	0,387	0,402	-0,286	0,951	0,331	0,441	-0,315	0,944	0,343	0,440	-0,284
PK_C16_D3	0,997	0,329	0,423	-0,436	0,994	0,310	0,235	-0,781	1,000	0,277	0,514	-0,402	0,994	0,289	0,491	-0,401
PK_C2.8284271_D1	0,623	0,564	0,485	0,288	0,615	0,554	0,487	0,285	0,624	0,548	0,502	0,291	0,634	0,562	0,504	0,299
PK_C2.8284271_D2	0,890	0,491	0,475	0,050	0,892	0,445	0,398	-0,109	0,890	0,460	0,466	0,002	0,874	0,410	0,578	0,082
PK_C2.8284271_D3	0,986	0,373	0,474	-0,265	0,982	0,408	0,251	-0,564	0,991	0,337	0,560	-0,217	0,984	0,341	0,347	-0,508
PK_C2_D1	0,623	0,567	0,489	0,292	0,614	0,555	0,495	0,290	0,625	0,547	0,500	0,289	0,633	0,562	0,508	0,301

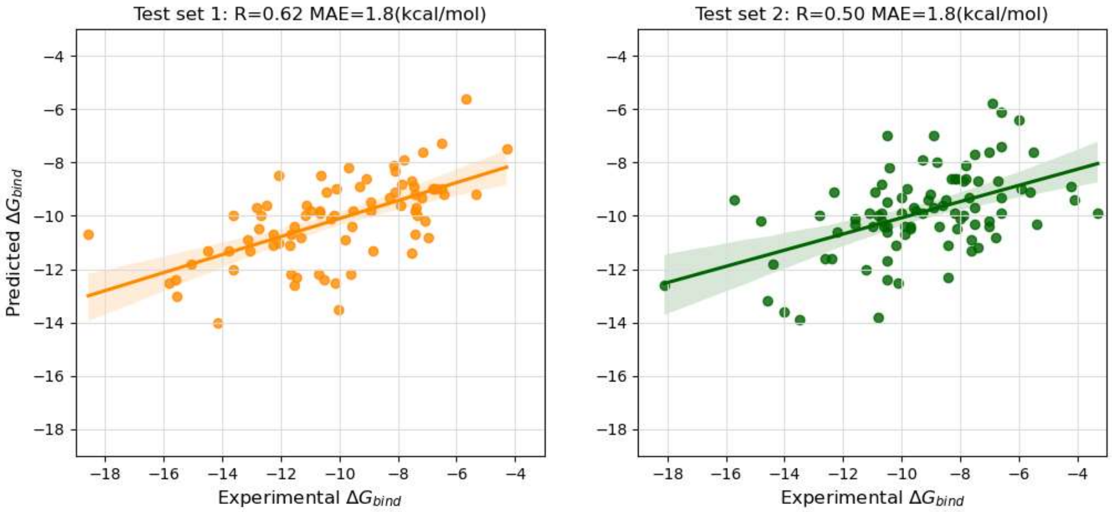
PK_C2_D2	0,877	0,513	0,455	0,067	0,876	0,466	0,397	-0,062	0,875	0,474	0,518	0,100	0,859	0,414	0,576	0,102
PK_C2_D3	0,983	0,382	0,495	-0,216	0,979	0,418	0,257	-0,531	0,988	0,355	0,592	-0,141	0,979	0,358	0,318	-0,517
PK_C4_D1	0,624	0,563	0,483	0,286	0,616	0,553	0,486	0,284	0,624	0,551	0,507	0,295	0,634	0,561	0,503	0,298
PK_C4_D2	0,902	0,468	0,499	0,037	0,907	0,432	0,410	-0,134	0,903	0,438	0,424	-0,100	0,890	0,396	0,569	0,035
PK_C4_D3	0,988	0,361	0,461	-0,306	0,984	0,400	0,247	-0,589	0,994	0,326	0,555	-0,247	0,986	0,328	0,396	-0,456
PK_C5.6568542_D1	0,624	0,562	0,483	0,285	0,615	0,554	0,484	0,283	0,623	0,549	0,505	0,293	0,633	0,562	0,502	0,298
PK_C5.6568542_D2	0,914	0,444	0,518	0,013	0,922	0,421	0,423	-0,155	0,917	0,425	0,435	-0,125	0,905	0,383	0,547	-0,027
PK_C5.6568542_D3	0,990	0,353	0,426	-0,376	0,986	0,386	0,249	-0,610	0,996	0,314	0,560	-0,265	0,988	0,316	0,436	-0,421
PK_C8_D1	0,624	0,560	0,481	0,283	0,614	0,553	0,481	0,280	0,623	0,549	0,505	0,293	0,634	0,562	0,504	0,299
PK_C8_D2	0,925	0,417	0,513	-0,046	0,933	0,403	0,417	-0,206	0,928	0,405	0,453	-0,145	0,919	0,375	0,524	-0,084
PK_C8_D3	0,992	0,344	0,414	-0,414	0,988	0,363	0,245	-0,660	0,997	0,303	0,551	-0,300	0,990	0,305	0,464	-0,403

Table SI-6. Summary of the performance of the individual and the ensemble models for protein-peptide models.

The models M1, M2, M3 and M4 correspond to the best predictors obtained from the training subsets 1, 2, 3 and 4 respectively. Then, the correlation coefficients (R) of the estimations in the development set were estimated for each model (R_IND), as well as for each possible combination of models. The combination rules were average (V_AVG), maximum (V_MAX), and minimum, (V_MIN) probabilities. The optimal ensemble model corresponds, from the group of ensembles two, to the model that outputs the binding affinity based on the maximum predicted value between the models obtained from the training subsets 1 and 3.

Ensembles	***Model***	***R_IND***	***V_AVG***	***V_MAX***	***V_MIN***
1	M1	0,527	0,537	0,552	0,519
	M2	0,539			
2	M1	0,527	0,540	**0,556**	0,519
	M3	0,543			
3	M1	0,527	0,536	0,532	0,533
	M4	0,532			
4	M2	0,539	0,545	0,548	0,539
	M3	0,543			
5	M2	0,539	0,542	0,532	0,546
	M4	0,532			
6	M3	0,543	0,544	0,543	0,540
	M4				
7	M1	0,527	0,542	0,559	0,520
	M2	0,539			
	M3	0,543			
8	M1	0,527	0,540	0,538	0,530
	M2	0,539			
	M4	0,532			
9	M1	0,527	0,542	0,548	0,530
	M3	0,543			
	M4	0,532			
10	M2	0,539	0,545	0,546	0,547
	M3	0,543			
	M4	0,532			
11	M1	0,527	0,543	0,553	0,529
	M2	0,539			
	M3	0,543			
	M4	0,532			

Figure SI-5. Plots of experimental vs. predicted BA values of PPI-Affinity on the test sets of protein – protein affinity data.



The performance is reported as the Pearson's Correlation coefficient (R) and the Mean Absolute Error (MAE) between experimental and predicted BA. Test set 1 corresponds to the benchmark of 79 complexes taken from Vangone and Bonvin.[1] Test set 2 corresponds to the hold-out set of 90 data points extracted from PDBbind (v.2020).[8]

Section SI-5. Performance of PPI-Affinity vs. a state-of-the-art biding affinity classifier.

Recently, Abbasi, W. A. et al.[32] developed a method using *Learning Using Privileged Information* (LUPI) paradigm and Support Vector Machines, that classifies as "high" or "low" the binding affinity of protein-protein complexes. The predictor, called LUPIA,[32] was trained using sequence and structure information. Nevertheless, in production, only the sequence information of protein pairs is required. To build the model, the authors discretized the BA values into two classes. For this, they used as threshold -10.86, which is the median value of the BA of the training dataset.

Here we used this value to discretize the output of PPI-Affinity and compare our protein-protein model with LUPIA.[32] The assessment was performed in two test sets (Table SI-5.1). Test set 1 corresponds to the hold-out set of 90 data points taken from PDBbind (v.2020).[8] We removed four cases that are in common with the training set of LUPIA,[32] and the evaluation was performed on the remaining 86 data points. The test set has protein-protein complexes with BA values between -18.1 and -3.3 kcal/mol. After discretizing such values, 15 and 71 data points were classified as "high" and "low" BA, respectively.

The Test set 2 corresponds to 26 wild-type and 151 mutants of protein-protein complexes taken from the SKEMPI v2.0[28] database. This data was employed to assess the performance of PPI-Affinity (R = 0.78 and MAE = 1.4 kcal/mol). When only considering the wild-types, the performance of the model was R = 0.77 and MAE = 1.1 kcal/mol.

Here, we employed the 26 wild-type complexes to compare our method to the LUPIA[32] classifier. The experimental binding affinity values of the protein pairs ranged between -16.3 and -7.0 kcal/mol. We used as reference the threshold value -10.86 and divided the data into 14 cases classified as "high" and 12 classified as complexes with "low" binding affinity.

The benchmark of 79 protein-protein complexes employed by Vangone and Bonvin[1] was not used in this comparison, as most of the cases were found in to be common with the training set of the LUPIA[32] predictor.

The performance measures used to evaluate the models were Sensitivity (Sn) and Specificity (Sp), formulated as:

$Sn = TP\ /\ (TP + FN)$

$Sp = TN/(TN + FP)$

where:

TP: number of protein-protein complexes correctly predicted as presenting "high" BA,

TN: number of protein-protein complexes correctly predicted as presenting "low" BA,

FP: number of protein-protein complexes incorrectly predicted as presenting "high" BA,

FN: number of protein-protein complexes incorrectly predicted as presenting "low" BA,

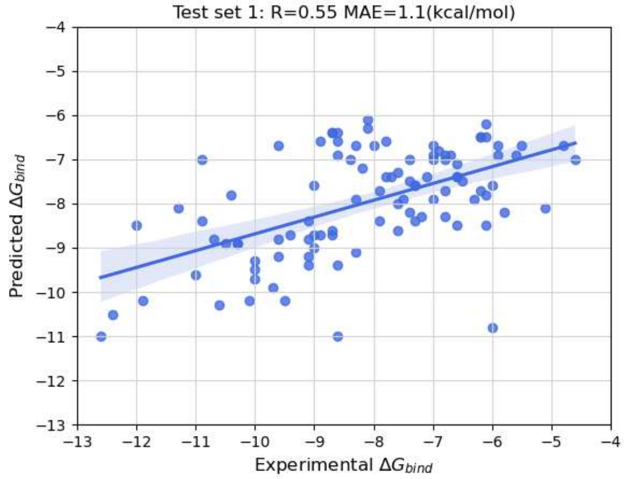
Table SI-5.1. Summary of the evaluation of PPI-Affinity and the LUPIA[32] classifier on two sets of protein – protein affinity data.

	Test set 1		Test set 2	
	LUPIA[32]	PPI-Affinity	LUPIA[32]	PPI-Affinity
Sn	**0.87**	0.47	**1.0**	0.86
Sp	0.32	**0.86**	0.17	**0.67**

The performance measures are Sensitivity (Sn) and Specificity (Sp). Test set 1 corresponds to a test set of 86 data points taken from PDBbind (v.2020),[8] while Test set 2 corresponds to 26 wild-type structures taken from the SKEMPI 2.0 dataset.[28]

The sensitivity of PPI-Affinity in Test set 1 decreased (Sn = 0.47). Nevertheless, of the 15 cases labeled as "High" in this data set, the eight misclassified differ by less than 1.7 kcal/mol from the threshold used, which is within the margin of error of the PPI-Affinity model (MAE = 1.8 kcal/mol). From them, six BA values differ by less than 1 kcal/mol. This reflects the downside of using a threshold to classify BA values. Furthermore, in both test sets, it can be seen that LUPIA[32] suffered from been too optimistic, as the method ranked most of the cases as with "High" affinity.

Figure SI-6. Plot of experimental vs. predicted by PPI-Affinity BA values on the test set of protein – peptide affinity data.



The performance is reported as the Pearson's Correlation coefficient (R) and the Mean Absolute Error (MAE) between experimental and predicted BA. The test set 1 corresponds to the hold-out set of 100 data points extracted from Biolip.[37]

Section SI-6. Description of the assays used to determine the binding affinities of EPI-X4 derivatives against the CRCX4 receptor**.**

To determine the affinity of the peptides for CXCR4, an antibody competition assay was used. The assay is based on the competitive binding of a fluorescently labelled anti-CXCR4 antibody (clone 12G5) with CXCR4 ligands.[42] For this 50,000 SupT1 cells/well were seeded in 96-well V-bottom microtiter plates in PBS supplemented with 1% FCS. The buffer was removed by centrifugation and cells precooled at 4°C for 15 min. The compound was serially diluted in PBS and added to the cells together with a 12G5 antibody at a constant concentration. Cells were incubated at 4°C for 2 hours before the unbound antibody and compounds were removed by 2 washing steps followed by fixation in 2% PFA. Mean fluorescence (MFI) of cells was determined by flow cytometry using FACS CytoFLEX. The isotype control was subtracted and values normalized to the 12G5-APC stained PBS control. IC_{50} values were determined by non-linear regression using GraphPad Prism.

Table SI-6.1 Summary of the binding affinities of EPI-X4 derivatives against CRCX4.

Derivative	Sequence	IC_{50}(nM)
JM#21	ILRWSRKLPCVS	77
JM#122	ILRWSRKLPSVS	130
JM#151	IVRWSKKVPSVS	131
JM#23	ILRWSRKVPSVS	173
JM#19	ILRWSRKMPCFS	262
WSC02	IVRWSKKVPCVS	268
JM#13	ILRWSRKMPCVS	271
JM#20	ILRWSRKMPCMS	272,5
JM#10	IFRWSRKVPCVS	315
JM#18	ILRWSRKMPCLS	385
408-414	LVRYTKK	482
JM#4	IVRWSHKVPCVS	535
408-415	LVRYTKKV	562
JM#146	IYRWSRKMPCLS	584
JM#9	ILRWSHKVPCVS	615
JM#1	ILRWSKKVPCVS	732
408-421	LVRYTKKVPQVSTP	825
JM#105	IIRWSKKVPCVS	966
JM#113	IIRWSRKLPCVS	1099
408-420	LVRYTKKVPQVST	1102
408-417	LVRYTKKVPQ	1103
408-418	LVRYTKKVPQV	1364
408-422	LVRYTKKVPQVSTPT	1407
JM#133	IVRWSKYVS	1584

JM#111	FLRWSRKLPCVS	3353
JM#112	PLRWSRKLPCVS	3371
EPI-X4	**LVRYTKKVPQVSTPTL**	**3709**
JM#94	LIRYTKKVPQVSTPTL	6623
JM#103	FVRWSKKVPCVS	8748
408-413	LVRYTK	9295
JM#123	ILKSSKLPCLS	>10000
JM#125	ILRHSRGPS	>10000
JM#126	IPKWSRGVS	>10000
JM#127	ILKQSRKAPL	>10000
JM#128	ILRTSRFISS	>10000
JM#129	IVRSRKGGTVS	>10000
JM#130	IVRWSPPCVS	>10000
JM#131	IVKSKKAPCVS	>10000
JM#132	IVRKKVPCPS	>10000
JM#134	IVKSHKAPCVS	>10000
JM#135	IVRSSRKVVS	>10000
JM#136	IARSKRGPCAN	>10000
JM#137	IVKNQRKVPV	>10000
JM#138	VVRNSKAAFH	>10000
JM#152	CLKLPGGSCM	>10000
JM#153	CLRLPGGSC	>10000
JM#154	NIRVGGTGMF	>10000
JM#155	QKVVAGVANAL	>10000
JM#156	SRVLNLGPI	>10000
JM#157	MRRAPAFLSA	>10000
JM#158	AGRGKLIAV	>10000
JM#159	NEKRFYLK	>10000
JM#160	SRDKALLRL	>10000
JM#161	GKHVPRAVFV	>10000
JM#162	SKSGRLLLAGY	>10000
JM#92	FVRYTKKVPQVSTPTL	>10000
JM#93	PVRYTKKVPQVSTPTL	>10000

Section SI-7. Description of the models used for the generation of the data related to peptide binders to the PDZ domain of HTRA1 or HTRA3 (Tables 3 and 4).

The template used for the predictions of the binding affinity of peptides with HTRA1 was based on the high-resolution structure corresponding to the PDB code 2JOA.[43] The first model of the ensemble was used for the predictions. In the case of the HTRA3 protein-peptide complexes, the structure with the PDB code 2PW3 was employed as template.[43] Chain A from the protein dimeric structure was used since it lacks only two residues in contrast to chain B which lacks 8 residues. Conformer A of Ser 446 was used in our model. Residues missing in the model were given to the server by using the sequence option.

Table SI-7.1. Ranking of BA of HTRA1-peptide complexes as predicted by state-of-the-art models.

Kdeep		DFIRE		CP_PIE		RF-Score		Prodigy	
Ranking	BA	Ranking	BA	Ranking	BA	Ranking	BA	Ranking	BA
1) WDKIWHV	-13,5	1) ASRIWWV	-16,6	1) DSAIWWV	1,25491	1) DSRAWWV	7,11	1) GWKTWIL	-8,1
2) ASRIWWV	-13,4	2) DIETWLL	-16,5	2) DIETWLL	1,22589	**2) DSRIWWV**	**7,03**	2) DIETWLL	-7,4
3) DIETWLL	-12,6	3) DARIWWV	-16,2	3) DARIWWV	1,16295	3) DARIWWV	7,01	3) WDKIWHV	-7,2
4) DARIWWV	-12,5	3) DSAIWWV	-16,2	4) ASRIWWV	1,11223	4) WDKIWHV	7	4) ASRIWWV	-6,9
4) DSRAWWV	-12,5	**4) DSRIWWV**	**-16,1**	5) DSRIWWA	1,06328	5) ASRIWWV	6,96	5) DARIWWV	-6,7
4) GWKTWIL	-12,5	4) GWKTWIL	-16,1	**6) DSRIWWV**	**1,05627**	6) DSRIWAV	6,95	5) DSAIWWV	-6,7
5) DSRIWWA	-12,0	5) DSRAWWV	-15,7	7) DSRIAWV	1,04117	7) DSAIWWV	6,91	6) DSRAWWV	-6,5
6) DSAIWWV	-11,9	6) WDKIWHV	-15,6	8) DIGPVCFL[a]	1,00710	7) GWKTWIL	6,91	7) DSRIWWA	-6,2
7) DSRIWWV	**-11,7**	7) DIGPVCFL[a]	-15,3	9) DSRAWWV	1,00222	8) DIETWLL	6,9	**7) DSRIWWV**	**-6,2**
8) DSRIAWV	-11,1	8) DSRIWWA	-14,9	10) WDKIWHV	0,91622	8) DSRIWWA	6,9	8) DSRIAWV	-6,1
9) DIGPVCFL[a]	-11,0	9) EVKIMVV[a]	-14,1	11) GWKTWIL	0,88501	9) DSRIAWV	6,79	8) DSRIWAV	-6,1
10 EVKIMVV[a]	-10,7	10) DSRIWAV	-14,0	12) DSRIWAV	0,83207	10) DIGPVCFL[a]	6,64	9) EVKIMVV[a]	-5,8
11) DSRIWAV	-10,1	11) DSRIAWV	-13,4	13) EVKIMVV[a]	0,82242	11) EVKIMVV[a]	6,47	10) DIGPVCFL[a]	-5,6

[a]These protein-peptide structures are at the border of the applicability domain of PPI-Affinity

Table SI-7.2. Ranking of BA of HTRA3-peptide complexes as predicted by state-of-the-art models.

Kdeep		DFIRE		CP_PIE		RF-Score		Prodigy	
Ranking	BA	Ranking	BA	Ranking	BA	Ranking	BA	Ranking	BA
1) FGRWF[b]	-10,0	1) FGRWA[a]	-6,6	1) FGRWA[a]	0,357316	1) FARWV[b]	4,24	1) RSWWV	-4,9
2) FGRWV[a]	**-8,9**	2) FARWV[b]	-6,4	**2) FGRWV[a]**	**0,356035**	**2) FGRWV[a]**	**4,08**	2) RWV[a]	-3,7
3) FGRWI[b]	-8,8	2) FGRWL[a]	-6,4	3) FGRWF[b]	0,35465	3) FGRWI[b]	3,97	3) WG[b]	-3,6
4) RSWWV	-8,6	3) FGAWV[b]	-6,2	4) FGRWL[a]	0,352168	4) FGRWA[a]	3,9	4) FARWV[b]	-3,5
5) FARWV[b]	-8,5	3) FGRWI[b]	-6,2	5) FGAWV[b]	0,348097	5) FGRWL[a]	3,88	4) WA[b]	-3,5
6) FGRWL[a]	-8,2	4) FGRWF[b]	-6,1	6) FGRWI[b]	0,345033	6) FGRWF[b]	3,84	5) FGRWA[a]	-3,3
7) FGRWA[a]	-7,9	**5) FGRWV[a]**	**-6,0**	7) FARWV[b]	0,333353	7) FGAWV[b]	3,81	5) FGRWF[b]	-3,3
8) FGAWV[b]	-7,8	6) RSWWV	-5,3	8) FGRAV[a]	0,248552	7) RSWWV	3,81	**5) FGRWV[a]**	**-3,3**
9) FGRAV[a]	-7,0	7) FGRAV[a]	-4,1	9) WG[b]	0,163576	8) FGRAV[a]	3,66	5) GRWV[a]	-3,3

10) GRWV[a]	-6,8	8) RWV[a]	-3,9	10) RSWWV	0,159256	9) WG[b]	3,47	5) WV[b]	-3,3
11) RWV[a]	-5,7	9) WV[b]	-3,7	11) WA[b]	0,150201	10) WV[b]	3,45	6) FGAWV[b]	-3,2
12) WA[b]	-5,3	10) WA[b]	-3,5	12) RWV[a]	0,14853	11) GRWV[a]	3,44	6) FGRWI[b]	-3,2
13) WV[b]	-5,1	11) WG[b]	-3,3	13) GRWV[a]	0,101262	12) RWV[a]	3,42	6) FGRWL[a]	-3,2
14) WG[b]	-4,2	12) GRWV[a]	-2,9	14) WV[b]	0,0838	13) WA[b]	3,4	7) FGRAV[a]	-3

[a] These protein-peptide structures are at the border of the applicability domain of PPI-Affinity. [b] These protein-peptide structures are outside the applicability domain of PPI-Affinity

Table SI-7. Summary of values that define the applicability domain of the protein-protein model.

MODEL 2

DESCRIPTOR	MINIMUM	MAXIMUM	1st PERCENTILE	99TH PERCENTILE
WNC(ECI)_NO_AHR_G	0	1	0,13	1
WNC(ECI)_NO_ALR_G	0	1	0,05	1
WNC(Z2)_NO_PLR_V	0	46,59	0	27,418
WNC(Z2)_NO_PCR_G	-3,128	2,512	-2,237	1,857
WNC(Z3)_NO_GLU_V	0	6,545	0	5,892
WNC(Z3)_NO_PLR_P2	0	5,317	0	5,195
WNC(Z3)_NO_PRT_P2	0	4,133	0	3,833
WFLC(ECI)_NO_ILE_P2	0	0,014	0	0,006
WFLC(IP)_NO_PCR_AR	0	0,035	0	0,027
WFLC(IP)_NO_PCR_V	0	0,002	0	0,001
WFLC(ISA)_NO_PLR_AR	0	0,011	0	0,008
WNLC(ECI)_NO_AHR_V	0	2,31	0,005	2,15
WNLC(ECI)_NO_NPR_N1	0	81,424	0,407	67,557
WNLC(ECI)_NO_NPR_DE	0	0,599	0,016	0,568
WNLC(IP)_NO_BSR_N1	0	28896,816	304,609	25512,865
WNLC(IP)_NO_PLR_N1	0	56383,04	695,219	51326,103
WNLC(ISA)_NO_BSR_N2	0	607045,942	46153,136	568135,465
WNLC(ISA)_NO_PLR_G	1	10458,877	1,671	9535,444
WNLC(Z1)_NO_AHR_DE	0	12,626	1,257	12,044
WNLC(Z1)_NO_ALR_N2	0	269,091	17,584	212,189
WNLC(Z1)_NO_NCR_P2	0	15,07	1,402	12,439
WNLC(Z1)_NO_PRT_AR	-2,423	3,379	-2,075	2,05
WNLC(Z2)_NO_BSR_G	-4,237	4,778	-4,157	4,315
WNLC(Z2)_NO_ALR_N1	-112,241	2145,601	-51,79	1292,54
WNLC(Z3)_NO_AHR_AR	-0,777	3,884	-0,666	2,621
WNLC(Z3)_NO_ALR_AR	-0,581	1,985	-0,318	1,324

MODEL 3

DESCRIPTOR	MINIMUM	MAXIMUM	1st PERCENTILE	99TH PERCENTILE
WNC(ECI)_NO_AHR_G	0	1	0,175	1
WNC(ECI)_NO_ALR_G	0	1	0,05	1
WNC(Z2)_NO_PLR_V	0	46,59	0	27,418
WNC(Z2)_NO_PCR_G	-2,014	2,512	-1,888	1,615
WNC(Z3)_NO_GLU_V	0	6,457	0	5,216
WNC(Z3)_NO_PLR_P2	0	5,413	0	5,28
WNC(Z3)_NO_PRT_P2	0	4,133	0	3,904
WFLC(ECI)_NO_ILE_P2	0	0,014	0	0,007
WFLC(IP)_NO_PCR_AR	0	0,035	0,001	0,02
WFLC(IP)_NO_PCR_V	0	0,002	0	0,001
WFLC(ISA)_NO_PLR_AR	0	0,009	0	0,008
WNLC(ECI)_NO_AHR_V	0	2,31	0,142	2,177
WNLC(ECI)_NO_NPR_N1	0	75,842	3,807	66,742
WNLC(ECI)_NO_NPR_DE	0	0,599	0,064	0,582
WNLC(IP)_NO_BSR_N1	0	25889,035	458,054	22440,66
WNLC(IP)_NO_PLR_N1	0	52027,76	790,424	47683,177
WNLC(ISA)_NO_BSR_N2	0	569128,79	77888,907	497878,815
WNLC(ISA)_NO_PLR_G	1	10502,058	4,237	9636,287
WNLC(Z1)_NO_AHR_DE	0	12,628	3,082	12,373
WNLC(Z1)_NO_ALR_N2	0	269,091	27,865	186,411
WNLC(Z1)_NO_NCR_P2	0	12,857	3,593	11,926
WNLC(Z1)_NO_PRT_AR	-4,954	2,927	-2,312	2,048
WNLC(Z2)_NO_BSR_G	-4,237	4,778	-4,157	4,315
WNLC(Z2)_NO_ALR_N1	-112,241	2145,601	-51,79	1292,54
WNLC(Z3)_NO_AHR_AR	-1,035	2,876	-0,706	2,59
WNLC(Z3)_NO_ALR_AR	-0,581	1,65	-0,385	1,288

Table SI-8. Summary of values that define the applicability domain of the protein-peptide model.

MODEL 1

DESCRIPTOR	MINIMUM	MAXIMUM	1st PERCENTILE	99TH PERCENTILE
WNC(ECI)_NO_AHR_N1	0	44,171	0,086	28,189
WNC(ECI)_NO_ALR_N1	0	6,367	0,037	4,032
WNC(IP)_NO_PLR_N1	0	5391,296	96,607	3672,786
WNC(ISA)_NO_PLR_V	0	318853829,8	543962,387	250318617,3
WNC(Z1)_NO_NPR_N2	0	132,965	4,695	111,105
WNC(Z1)_NO_ARM_V	0	692,98	0	323,717
WNC(Z1)_NO_PRT_N1	-241,412	418,803	-169,452	331,683
WNC(Z3)_NO_AHR_N1	-57,396	149,198	-37,179	58,761
WNC(Z3)_NO_UCR_N1	-34,347	33,82	-32,638	25,764
WFLC(ECI)_NO_AHR_SI30	0,007	1	0,011	0,764
WFLC(ECI)_NO_PCR_V	0	0,026	0	0,015
WFLC(IP)_NO_RTR_MI30	0,062	4,74	0,074	4,523
WFLC(IP)_NO_BSR_AR	0,001	0,037	0,001	0,024
WFLC(IP)_NO_UCR_RA	0,001	0,5	0,001	0,314
WFLC(IP)_NO_UCR_K	46,226	12840,122	70,881	11654,217
WFLC(IP)_NO_PRT_TI30	14,202	573,443	18,237	454,996
WFLC(ISA)_NO_NPR_P2	0,006	0,094	0,007	0,082
WFLC(ISA)_NO_NPR_DE	0,005	0,089	0,007	0,071
WFLC(ISA)_NO_PRT_TI30	14,202	582,373	18,237	461,122
WFLC(Z1)_NO_BSR_SI30	0,009	1	0,02	0,914
WFLC(Z2)_NO_PLR_K	113,123	49106,737	364,597	40834,695
WNLC(ECI)_NO_BSR_I50	0,124	1,531	0,13	1,323
WNLC(ECI)_NO_PLR_K	110,951	25195,103	240,618	18518,305
WNLC(ECI)_NO_UCR_S	-0,109	2,746	0,071	2,166
WNLC(IP)_NO_RTR_V	339,368	1437,317	375,712	1305,349
WNLC(IP)_NO_UCR_I50	11,749	62,265	18,864	56,295
WNLC(ISA)_NO_UCR_S	-0,009	4,109	0,53	3,201
WNLC(Z1)_NO_RTR_SI30	0,547	1	0,562	0,95
WNLC(Z1)_NO_BSR_K	80,492	12661,731	113,741	10133,441
WNLC(Z1)_NO_UCR_K	38,097	8637,703	57,65	7834,548
WNLC(Z2)_NO_AHR_P2	1,46	4,799	1,647	4,391
WNLC(Z2)_NO_PCR_N2	4,845	56,314	6,671	54,94
WNLC(Z2)_NO_NCR_P2	0,714	4,332	0,905	3,913
WNLC(Z2)_NO_UCR_K	56,544	7155,916	69,33	6747,321
WNLC(Z3)_NO_PLR_SI30	0,419	0,827	0,423	0,775
WNLC(Z3)_NO_NCR_RA	4,361	27,671	5,534	22,597
WNLC(Z3)_NO_UCR_TI30	20,529	504,745	29,219	462,806

MODEL 3

DESCRIPTOR	MINIMUM	MAXIMUM	1st PERCENTILE	99TH PERCENTILE
WNC(ECI)_NO_AHR_N1	0	44,171	0,056	27,473
WNC(ECI)_NO_ALR_N1	0	6,367	0,023	4,239
WNC(IP)_NO_PLR_N1	0	5554,6	66,773	4292,97
WNC(ISA)_NO_PLR_V	0	290472538,7	543962,387	236857692,4
WNC(Z1)_NO_NPR_N2	0	132,965	4,695	115,217
WNC(Z1)_NO_ARM_V	0	692,98	0	323,717
WNC(Z1)_NO_PRT_N1	-241,412	418,803	-167,234	331,683
WNC(Z3)_NO_AHR_N1	-63,649	149,198	-40,292	58,582
WNC(Z3)_NO_UCR_N1	-35,227	33,82	-32,638	25,745
WFLC(ECI)_NO_AHR_SI30	0,01	1	0,013	0,764
WFLC(ECI)_NO_PCR_V	0	0,026	0	0,014
WFLC(IP)_NO_RTR_MI30	0,071	4,657	0,079	4,501
WFLC(IP)_NO_BSR_AR	0,001	0,035	0,001	0,024
WFLC(IP)_NO_UCR_RA	0,001	0,5	0,001	0,322
WFLC(IP)_NO_UCR_K	23,224	20147,154	79,781	11654,217
WFLC(IP)_NO_PRT_TI30	13,456	566,217	17,456	453,162
WFLC(ISA)_NO_NPR_P2	0,006	0,094	0,007	0,081
WFLC(ISA)_NO_NPR_DE	0,004	0,089	0,005	0,071
WFLC(ISA)_NO_PRT_TI30	13,456	577,568	17,456	456,221
WFLC(Z1)_NO_BSR_SI30	0,009	0,969	0,02	0,906
WFLC(Z2)_NO_PLR_K	113,123	47098,384	407,834	35890,325
WNLC(ECI)_NO_BSR_I50	0,118	1,396	0,129	1,253
WNLC(ECI)_NO_PLR_K	110,951	22599,466	244,172	18485,829
WNLC(ECI)_NO_UCR_S	-0,109	2,746	0,106	2,346
WNLC(IP)_NO_RTR_V	305,401	1463,52	375,712	1200,763
WNLC(IP)_NO_UCR_I50	13,728	62,265	19,17	57,883
WNLC(ISA)_NO_UCR_S	0,398	4,109	0,64	2,989
WNLC(Z1)_NO_RTR_SI30	0,552	1	0,569	0,932
WNLC(Z1)_NO_BSR_K	83,132	12260,904	113,637	10124,995
WNLC(Z1)_NO_UCR_K	38,097	10241,145	63,865	7305,526
WNLC(Z2)_NO_AHR_P2	1,575	4,856	1,723	4,396
WNLC(Z2)_NO_PCR_N2	4,845	56,314	6,671	54,92
WNLC(Z2)_NO_NCR_P2	0,714	4,332	0,905	3,968
WNLC(Z2)_NO_UCR_K	58,983	7155,916	72,516	6550,209
WNLC(Z3)_NO_PLR_SI30	0,419	0,833	0,423	0,76
WNLC(Z3)_NO_NCR_RA	4,361	27,671	5,36	22,538
WNLC(Z3)_NO_UCR_TI30	16	504,745	31,299	462,593

Table SI-9. Summary of the minimum and maximum values of the sequences' length of the peptides and proteins in each dataset.

		Protein size (aa)			
		Peptide		Receptor	
	Data size	Min	Max	Min	Max
Training	922	3	29	31	957
Development	100	4	29	51	559
Test	100	4	29	51	496
EPI-X4	57	6	16	319	319
HTRA1	13	7	8	105	105
HTRA3	14	2	5	105	105

The described subsets are the training, development and test sets with data points taken from the Biolip[37] database, as well as those used to test the protein-peptide model on experimentally measured BA data: EPI-X4, HTRA1 and HTRA3.

Table SI-10. Descriptive statistics of the different data sets used in the modeling and test of the protein-protein BA predictor.

		Binding Affinity (ΔG)			
	Data size	Min	Max	Mean	StdDev
Training set	648	-18.1	-3.1	-9.7	2.5
Development set	90	-18.1	-4.3	-9.5	2.6
Test set 1[1]	79	-18.6	-4.3	-10.1	2.8
Test set 2	90	-18.1	-3.3	-9.3	2.6
Test set 3[28]	177	-16.3	-5.5	-10.4	2.6

The reported statistics are the minimum (Min), maximum (Max), mean and standard deviation (StdDev) of the binding free energy (ΔG) values in each dataset. The described subsets are: the training, development, and test (Test set 2) sets with data taken from the PDBbind (v.2020)[8] dataset, Test set 1 corresponding to the benchmark employed by Vangone and Bonvin,[1] and Test set 3 corresponding to the set of 26 wild-types and 151 mutants taken from the SKEMPI[28] dataset. All the training protein-protein complexes contained two protein sequences with individual sequence length ranging from 20 to 958 amino acids.

Table SI-11. Descriptive statistics of the training, development and test sets used in the modeling of the protein-peptide BA predictor.

		Binding Affinity (ΔG)			
	Data size	Min	Max	Mean	StdDev
Training set	922	-14.4	-3.6	-8.2	2.1
Development set	100	-13.6	-4.8	-8.5	2.0
Test set	100	-12.6	-4.6	-8.1	1.7

The reported statistics are the minimum (Min), maximum (Max), mean and standard deviation (StdDev) of the binding free energy (ΔG) values in each dataset.

References

1. Vangone, A.; Bonvin, A. M., Contacts-based prediction of binding affinity in protein-protein complexes. *Elife* **2015**, 4, e07454-e07454.

2. Kastritis, P. L.; Moal, I. H.; Hwang, H.; Weng, Z.; Bates, P. A.; Bonvin, A. M. J. J.; Janin, J., A structure-based benchmark for protein–protein binding affinity. *Protein Science* **2011**, 20, 482-491.

3. Liu, S.; Zhang, C.; Zhou, H.; Zhou, Y., A physical reference state unifies the structure-derived potential of mean force for protein folding and binding. *Proteins: Structure, Function, and Bioinformatics* **2004**, 56, 93-101.

4. Ravikant, D. V. S.; Elber, R., PIE-efficient filters and coarse grained potentials for unbound protein-protein docking. *Proteins* **2010**, 78, 400-419.

5. Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E., The Protein Data Bank. *Nucleic Acids Research* **2000**, 28, 235-242.

6. RCSB PDB. https://www.rcsb.org/

7. Jiménez, J.; Škalič, M.; Martínez-Rosell, G.; De Fabritiis, G., KDEEP: Protein–Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks. *Journal of Chemical Information and Modeling* **2018**, 58, 287-296.

8. Wang, R.; Fang, X.; Lu, Y.; Wang, S., The PDBbind Database: Collection of Binding Affinities for Protein−Ligand Complexes with Known Three-Dimensional Structures. *Journal of Medicinal Chemistry* **2004**, 47, 2977-2980.

9. Ballester, P. J.; Mitchell, J. B. O., A machine learning approach to predicting protein-ligand binding affinity with applications to molecular docking. *Bioinformatics* **2010**, 26, 1169-1175.

10. Rodriguez-Soca, Y.; Munteanu, C. R.; Dorado, J.; Pazos, A.; Prado-Prado, F. J.; González-Díaz, H., Trypano-PPI: A Web Server for Prediction of Unique Targets in Trypanosome Proteome by using Electrostatic Parameters of Protein−protein Interactions. *Journal of Proteome Research* **2010**, 9, 1182-1190.

11. Dobson, P. D.; Doig, A. J., Distinguishing Enzyme Structures from Non-enzymes Without Alignments. *Journal of Molecular Biology* **2003**, 330, 771-783.

12. Rodriguez-Soca, Y.; Munteanu, C. R.; Dorado, J.; Rabuñal, J.; Pazos, A.; González-Díaz, H., Plasmod-PPI: A web-server predicting complex biopolymer targets in plasmodium with entropy measures of protein–protein interactions. *Polymer* **2010**, 51, 264-273.

13. Chou, K.-C.; Cai, Y.-D., Predicting Protein−Protein Interactions from Sequences in a Hybridization Space. *Journal of Proteome Research* **2006**, 5, 316-322.

14. von Mering, C.; Jensen, L. J.; Snel, B.; Hooper, S. D.; Krupp, M.; Foglierini, M.; Jouffre, N.; Huynen, M. A.; Bork, P., STRING: known and predicted protein–protein associations, integrated and transferred across organisms. *Nucleic Acids Research* **2005**, 33, D433-D437.

15. Narykov, O.; Johnson, N. T.; Korkin, D., Predicting protein interaction network perturbation by alternative splicing with semi-supervised learning. *Cell Reports* **2021**, 37.

16. Yang, X.; Coulombe-Huntington, J.; Kang, S.; Sheynkman, Gloria M.; Hao, T.; Richardson, A.; Sun, S.; Yang, F.; Shen, Yun A.; Murray, Ryan R.; Spirohn, K.; Begg, Bridget E.; Duran-Frigola, M.; MacWilliams, A.; Pevzner, Samuel J.; Zhong, Q.; Trigg, Shelly A.; Tam, S.; Ghamsari, L.; Sahni, N.; Yi, S.; Rodriguez, Maria D.; Balcha, D.; Tan, G.; Costanzo, M.; Andrews, B.; Boone, C.; Zhou, Xianghong J.; Salehi-Ashtiani, K.; Charloteaux, B.; Chen, Alyce A.; Calderwood, Michael A.; Aloy, P.; Roth, Frederick P.; Hill, David E.; Iakoucheva, Lilia M.; Xia, Y.; Vidal, M., Widespread Expansion of Protein Interaction Capabilities by Alternative Splicing. *Cell* **2016**, 164, 805-817.

17. Rolland, T.; Taşan, M.; Charloteaux, B.; Pevzner, Samuel J.; Zhong, Q.; Sahni, N.; Yi, S.; Lemmens, I.; Fontanillo, C.; Mosca, R.; Kamburov, A.; Ghiassian, Susan D.; Yang, X.; Ghamsari, L.; Balcha, D.; Begg, Bridget E.; Braun, P.; Brehme, M.; Broly, Martin P.; Carvunis, A.-R.; Convery-Zupan, D.; Corominas, R.; Coulombe-Huntington, J.; Dann, E.; Dreze, M.; Dricot, A.; Fan, C.; Franzosa, E.; Gebreab, F.; Gutierrez,

Bryan J.; Hardy, Madeleine F.; Jin, M.; Kang, S.; Kiros, R.; Lin, Guan N.; Luck, K.; MacWilliams, A.; Menche, J.; Murray, Ryan R.; Palagi, A.; Poulin, Matthew M.; Rambout, X.; Rasla, J.; Reichert, P.; Romero, V.; Ruyssinck, E.; Sahalie, Julie M.; Scholz, A.; Shah, Akash A.; Sharma, A.; Shen, Y.; Spirohn, K.; Tam, S.; Tejeda, Alexander O.; Trigg, Shelly A.; Twizere, J.-C.; Vega, K.; Walsh, J.; Cusick, Michael E.; Xia, Y.; Barabási, A.-L.; Iakoucheva, Lilia M.; Aloy, P.; De Las Rivas, J.; Tavernier, J.; Calderwood, Michael A.; Hill, David E.; Hao, T.; Roth, Frederick P.; Vidal, M., A Proteome-Scale Map of the Human Interactome Network. *Cell* **2014**, 159, 1212-1226.

18. Rual, J.-F.; Venkatesan, K.; Hao, T.; Hirozane-Kishikawa, T.; Dricot, A.; Li, N.; Berriz, G. F.; Gibbons, F. D.; Dreze, M.; Ayivi-Guedehoussou, N.; Klitgord, N.; Simon, C.; Boxem, M.; Milstein, S.; Rosenberg, J.; Goldberg, D. S.; Zhang, L. V.; Wong, S. L.; Franklin, G.; Li, S.; Albala, J. S.; Lim, J.; Fraughton, C.; Llamosas, E.; Cevik, S.; Bex, C.; Lamesch, P.; Sikorski, R. S.; Vandenhaute, J.; Zoghbi, H. Y.; Smolyar, A.; Bosak, S.; Sequerra, R.; Doucette-Stamm, L.; Cusick, M. E.; Hill, D. E.; Roth, F. P.; Vidal, M., Towards a proteome-scale map of the human protein–protein interaction network. *Nature* **2005**, 437, 1173-1178.

19. Venkatesan, K.; Rual, J.-F.; Vazquez, A.; Stelzl, U.; Lemmens, I.; Hirozane-Kishikawa, T.; Hao, T.; Zenkner, M.; Xin, X.; Goh, K.-I.; Yildirim, M. A.; Simonis, N.; Heinzmann, K.; Gebreab, F.; Sahalie, J. M.; Cevik, S.; Simon, C.; de Smet, A.-S.; Dann, E.; Smolyar, A.; Vinayagam, A.; Yu, H.; Szeto, D.; Borick, H.; Dricot, A.; Klitgord, N.; Murray, R. R.; Lin, C.; Lalowski, M.; Timm, J.; Rau, K.; Boone, C.; Braun, P.; Cusick, M. E.; Roth, F. P.; Hill, D. E.; Tavernier, J.; Wanker, E. E.; Barabási, A.-L.; Vidal, M., An empirical framework for binary interactome mapping. *Nature Methods* **2009**, 6, 83-90.

20. Yu, H.; Tardivo, L.; Tam, S.; Weiner, E.; Gebreab, F.; Fan, C.; Svrzikapa, N.; Hirozane-Kishikawa, T.; Rietman, E.; Yang, X.; Sahalie, J.; Salehi-Ashtiani, K.; Hao, T.; Cusick, M. E.; Hill, D. E.; Roth, F. P.; Braun, P.; Vidal, M., Next-generation sequencing to generate interactome datasets. *Nature Methods* **2011**, 8, 478-480.

21. Jia, J.; Liu, Z.; Xiao, X.; Liu, B.; Chou, K.-C., iPPBS-Opt: A Sequence-Based Ensemble Classifier for Identifying Protein-Protein Binding Sites by Optimizing Imbalanced Training Datasets. *Molecules (Basel, Switzerland)* **2016**, 21, E95-E95.

22. Deng, L.; Guan, J.; Dong, Q.; Zhou, S., Prediction of protein-protein interaction sites using an ensemble method. *BMC Bioinformatics* **2009**, 10, 426.

23. Yugandhar, K.; Gromiha, M. M., Protein–protein binding affinity prediction from amino acid sequence. *Bioinformatics* **2014**, 30, 3583-3589.

24. Abbasi, W. A.; Yaseen, A.; Hassan, F. U.; Andleeb, S.; Minhas, F. U. A. A., ISLAND: in-silico proteins binding affinity prediction using sequence information. *BioData Mining* **2020**, 13, 20.

25. Cao, D.-S.; Xu, Q.-S.; Liang, Y.-Z., propy: a tool to generate various modes of Chou's PseAAC. *Bioinformatics* **2013**, 29, 960-962.

26. Huang, X.; Zheng, W.; Pearce, R.; Zhang, Y., SSIPe: accurately estimating protein–protein binding affinity change upon mutations using evolutionary profiles in combination with an optimized physical energy function. *Bioinformatics* **2020**, 36, 2429-2437.

27. Xiong, P.; Zhang, C.; Zheng, W.; Zhang, Y., BindProfX: Assessing Mutation-Induced Binding Affinity Change by Protein Interface Profiles with Pseudo-Counts. *Journal of Molecular Biology* **2017**, 429, 426-434.

28. Jankauskaitė, J.; Jiménez-García, B.; Dapkūnas, J.; Fernández-Recio, J.; Moal, I. H., SKEMPI 2.0: an updated benchmark of changes in protein–protein binding energy, kinetics and thermodynamics upon mutation. *Bioinformatics* **2019**, 35, 462-469.

29. Janin, J.; Henrick, K.; Moult, J.; Eyck, L. T.; Sternberg, M. J. E.; Vajda, S.; Vakser, I.; Wodak, S. J., CAPRI: A Critical Assessment of PRedicted Interactions. *Proteins: Structure, Function, and Bioinformatics* **2003**, 52, 2-9.

30. Wang, B.; Su, Z.; Wu, Y., Computational Assessment of Protein–Protein Binding Affinity by Reverse Engineering the Energetics in Protein Complexes. *Genomics, Proteomics & Bioinformatics* **2021**.

31. Moal, I. H.; Fernández-Recio, J., SKEMPI: a Structural Kinetic and Energetic database of Mutant Protein Interactions and its use in empirical models. *Bioinformatics* **2012**, 28, 2600-2607.

32. Abbasi, W. A.; Asif, A.; Ben-Hur, A.; Minhas, F. u. A. A., Learning protein binding affinity using privileged information. *BMC Bioinformatics* **2018**, 19, 425.

33. Chen, J.; Sawyer, N.; Regan, L., Protein–protein interactions: General trends in the relationship between binding affinity and interfacial buried surface area. *Protein Science* **2013**, 22, 510-515.

34. Eddy, S. R., Where did the BLOSUM62 alignment score matrix come from? *Nature Biotechnology* **2004**, 22, 1035-1036.

35. Moal, I. H.; Agius, R.; Bates, P. A., Protein–protein binding affinity prediction on a diverse set of structures. *Bioinformatics* **2011**, 27, 3002-3009.

36. Dias, R.; Kolaczkowski, B., Improving the accuracy of high-throughput protein-protein affinity prediction may require better training data. *BMC Bioinformatics* **2017**, 18, 102.

37. Yang, J.; Roy, A.; Zhang, Y., BioLiP: a semi-manually curated database for biologically relevant ligand-protein interactions. *Nucleic acids research* **2013**, 41, D1096-103.

38. Xue, L. C.; Rodrigues, J. P.; Kastritis, P. L.; Bonvin, A. M.; Vangone, A., PRODIGY: a web server for predicting the binding affinity of protein–protein complexes. *Bioinformatics* **2016**, 32, 3676-3678.

39. Ballester, P. J.; Mitchell, J. B. O., A machine learning approach to predicting protein-ligand binding affinity with applications to molecular docking. *Bioinformatics (Oxford, England)* **2010**, 26, 1169-1175.

40. Ruiz-Blanco, Y. B.; Paz, W.; Green, J.; Marrero-Ponce, Y., ProtDCal: A program to compute general-purpose-numerical descriptors for sequences and 3D-structures of proteins. *BMC Bioinformatics* **2015**, 16, 162.

41. Romero-Molina, S.; Ruiz-Blanco, Y. B.; Green, J. R.; Sanchez-Garcia, E., ProtDCal-Suite: A web server for the numerical codification and functional analysis of proteins. *Protein Science* **2019**, 28, 1734-1743.

42. Harms, M.; Gilg, A.; Ständker, L.; Beer, A. J.; Mayer, B.; Rasche, V.; Gruber, C. W.; Münch, J., Microtiter plate-based antibody-competition assay to determine binding affinities and plasma/blood stability of CXCR4 ligands. *Scientific Reports* **2020**, 10, 16036.

43. Runyon, S. T.; Zhang, Y.; Appleton, B. A.; Sazinsky, S. L.; Wu, P.; Pan, B.; Wiesmann, C.; Skelton, N. J.; Sidhu, S. S., Structural and functional analysis of the PDZ domains of human HtrA1 and HtrA3. *Protein Science* **2007**, 16, 2454-2471.

6.3 Publication III

Author contributions

ABP-Finder: A Tool to Identify Antibacterial Peptides and the Gram-Staining Type of Targeted Bacteria

Yasser B. Ruiz-Blanco, Guillermin Agüero-Chapin, **Sandra Romero-Molina**, Agostinho Antunes, Lia-Raluca Olari, Barbara Spellerberg, Jan Münch, Elsa Sanchez-Garcia

Research Article in:	**Antibiotics**
Impact factor:	5.222 (2021)
Published online:	November 26, 2022
DOI:	10.3390/antibiotics11121708
Citations:	-

Complete citation from source:

Ruiz-Blanco, Y.B.; Agüero-Chapin, G.; Romero-Molina, S.; Antunes, A.; Olari, L.-R.; Spellerberg, B.; Münch, J.; Sanchez-Garcia, E. ABP-Finder: A Tool to Identify Antibacterial Peptides and the Gram-Staining Type of Targeted Bacteria. *Antibiotics* 2022; 11(12): 1708. https://doi.org/10.3390/antibiotics11121708

Contributions:

Conception:	0%
Model creation:	0%
Model validation:	0%
Experimental assays:	0%
Code implementations:	100%
Web(tool) validation and deployment:	100%
Web(tool) maintenance:	100%
Manuscript writing:	30%
Manuscript revision:	20%

antibiotics

Article

ABP-Finder: A Tool to Identify Antibacterial Peptides and the Gram-Staining Type of Targeted Bacteria

Yasser B. Ruiz-Blanco [1,*], Guillermin Agüero-Chapin [2,3], Sandra Romero-Molina [1], Agostinho Antunes [2,3], Lia-Raluca Olari [4], Barbara Spellerberg [5], Jan Münch [4] and Elsa Sanchez-Garcia [1,*]

[1] Computational Biochemistry, Center of Medical Biotechnology, University of Duisburg-Essen, 45141 Essen, Germany
[2] CIIMAR—Centro Interdisciplinar de Investigação Marinha e Ambiental, Universidade do Porto, Terminal de Cruzeiros do Porto de Leixões, Av. General Norton de Matos, s/n, 4450-208 Porto, Portugal
[3] Departamento de Biologia, Faculdade de Ciências, Universidade do Porto, Rua do Campo Alegre, 4169-007 Porto, Portugal
[4] Institute of Molecular Virology, University Hospital Ulm, 89081 Ulm, Germany
[5] Institute of Medical Microbiology and Hygiene, University Hospital Ulm, 89081 Ulm, Germany
* Correspondence: ybruizblanco@gmail.com (Y.B.R.-B.); elsa.sanchez-garcia@uni-due.de (E.S.-G.)

Citation: Ruiz-Blanco, Y.B.; Agüero-Chapin, G.; Romero-Molina, S.; Antunes, A.; Olari, L.-R.; Spellerberg, B.; Münch, J.; Sanchez-Garcia, E. ABP-Finder: A Tool to Identify Antibacterial Peptides and the Gram-Staining Type of Targeted Bacteria. *Antibiotics* **2022**, *11*, 1708. https://doi.org/10.3390/antibiotics11121708

Academic Editor: Jean-Marc Sabatier

Received: 20 October 2022
Accepted: 17 November 2022
Published: 26 November 2022

Abstract: Multi-drug resistance in bacteria is a major health problem worldwide. To overcome this issue, new approaches allowing for the identification and development of antibacterial agents are urgently needed. Peptides, due to their binding specificity and low expected side effects, are promising candidates for a new generation of antibiotics. For over two decades, a large diversity of antimicrobial peptides (AMPs) has been discovered and annotated in public databases. The AMP family encompasses nearly 20 biological functions, thus representing a potentially valuable resource for data mining analyses. Nonetheless, despite the availability of machine learning-based approaches focused on AMPs, these tools lack evidence of successful application for AMPs' discovery, and many are not designed to predict a specific function for putative AMPs, such as antibacterial activity. Consequently, among the apparent variety of data mining methods to screen peptide sequences for antibacterial activity, only few tools can deal with such task consistently, although with limited precision and generally no information about the possible targets. Here, we addressed this gap by introducing a tool specifically designed to identify antibacterial peptides (ABPs) with an estimation of which type of bacteria is susceptible to the action of these peptides, according to their response to the Gram-staining assay. Our tool is freely available via a web server named ABP-Finder. This new method ranks within the top state-of-the-art ABP predictors, particularly in terms of precision. Importantly, we showed the successful application of ABP-Finder for the screening of a large peptide library from the human urine peptidome and the identification of an antibacterial peptide.

Keywords: antibacterial peptide; machine learning; AMPs database; StarPep; Gram staining-based target; peptide library screening; human peptidome

1. Introduction

Antibiotic resistance is a life-threatening health problem worldwide, and one of the main causes of death in developing countries [1,2]. The potential capability of peptides to overcome resistance [3] has motivated the development of new antibiotics from antimicrobial peptides (AMPs) to combat multi-drug resistant pathogens and the threats of Gram-negative infections [4,5].

AMPs are oligopeptides produced by a great variety of organisms, from prokaryotes to eukaryotes, including humans. Due to their various functions, AMPs are considered a part of the innate immune system of higher eukaryotes. The structural diversity of AMPs allows them to display a broad range of antimicrobial activity against pathogenic agents, including viruses, Gram-positive and Gram-negative bacteria, as well as fungi. Besides, the bacterial selectivity of AMPs over eukaryotic cells and their different action modes make peptides excellent antibiotic candidates [3,4,6]. A widespread mechanism of antibacterial peptides (ABPs) is the destabilization and destruction of bacterial membranes. However, these peptides can also interfere with intracellular processes such as nucleic acid and protein synthesis, enzymatic modulation, and protein degradation [7–9], which is an advantage over traditional antibiotics [3,10].

Most AMPs are naturally occurring peptides that represent promising candidates for optimization in advanced steps of the drug design process [11]. AMP-based drugs have been clinically approved to treat both topical and systemic infections. For instance, polymyxins and gramicidin S were formulated for the prevention of topical infections caused by *Pseudomonas aeruginosa* and *Acinetobacter baumannii*. Colistin, a polymyxin derivative, is currently used for the systemic treatment of lung infections, especially those caused by *Pseudomonas aeruginosa* [12]. Due to its problematic resistance profile, *Pseudomonas aeruginosa* is often difficult to treat by antibiotics [13]. However, it can be targeted by a variety of different AMPs [13–15] that may be further developed into innovative therapeutics.

The specificity of peptides toward certain targets is usually highlighted as an important benefit for therapeutic intervention. Nonetheless, a downside of this feature is the associated challenge for the drug design process, given that small structural modifications can significantly influence both the activity and pharmacokinetic properties of the peptides. Consequently, optimizing the precision of tools for the screening of large datasets of peptides is of utmost relevance to improve efficiency at the early steps of drug design processes.

For over a decade, growth in the publicly available data of AMPs has been witnessed, with the subsequent development of several machine learning (ML)-based predictors integrated with AMP databases such as DAMP [16], APD3 [17], CAMP [18], CAMP$_{R3}$ [19], LAMP [20], DRAMP [21], ADAM [22], and DBAASP [23]. However, most of these prediction tools only discriminate between AMPs and non-AMPs. This is a highly ambiguous outcome given the broad scope of antimicrobial activity, which typically refers to more than 20 biological functions, such as the annotations in APD3 [17].

A group of predictors addressed this issue by applying a hierarchical classification scheme where first the peptides are classified as AMPs or not, and the positive cases are then sub-divided into a couple of classes based on selected AMP functions (e.g., antibacterial, antiviral, and antifungal peptides). Examples of such predictors, which include the antibacterial function are AntiBP2 [24], ClassAMP [25], MLAMP [26], *i*AMPpred [27], AMAP [28], AMP Scanner [29,30], and AMPDiscover [31]. However, of them, only AMP Scanner vr.1 predicts a type of bacterial target (*E. coli* or *S. aureus*) for the identified ABP [29].

In this context, we implemented a two-level predictor focused on antibacterial peptides (ABPs), named ABP-Finder, whose inner classifier estimates the Gram staining type of the putative targets. This tool leverages random forest (RF) classifiers trained with peptide data extracted from StarPep, the largest up to date public database of AMPs [32]. ABP-Finder categorizes ABPs and non-ABPs in the first classification level. Subsequently, the peptides identified as ABPs are sub-classified according to the Gram staining type of the potential targets i.e., exclusively Gram-positive, exclusively Gram-negative bacteria, or broad-spectrum peptides with expected activity against both types of bacteria. The ABPs used to develop this predictor show activity against at least one of nine representative bacterial targets (see Dataset section), among which are species with known multi-drug resistance such as *Acinetobacter baumannii*, *Enterococcus faecium*, *Klebsiella pneumonia*, *Pseudomonas aeruginosa*, and *Staphylococcus aureus*. With ABP-Finder, we weigh precision as the main performance feature of the prediction. In this way, we boost the efficiency of the screening step at the early stages of the drug design process aiming at the development of peptide-based antibiotics. Remarkably, we prove the efficacy of ABP-Finder for such screenings with the identification of a peptide from the human urine peptidome, displaying antimicrobial activity against *Pseudomonas aeruginosa*.

2. Materials and Methods

2.1. Data Collection and Pre-Processing

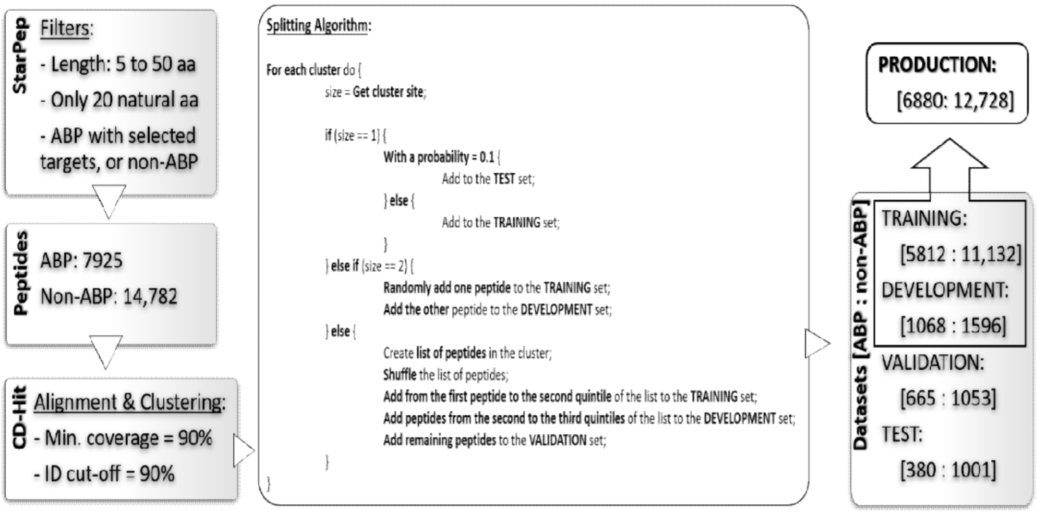
The models developed in this study were derived from the StarPep database [32,33]. This resource, as described by the authors, is a non-redundant compendium from 40 publicly available data sources, which encompasses annotations of more than 20 functions in approximately 45,000 AMPs, with nearly 8000 entries labelled as antibacterial peptides.

Before describing the construction of our training and test sets, we point out a shortcoming of several AMP-based predictors found in the literature [16–22], whose models do not obey the first principle dictated by the Organisation for Economic Co-operation and Development (OECD) to build reliable Quantitative Structure–Activity Relationship (QSAR)/ML-based models [34] (https://doi.org/10.1787/9789264085442-en (accessed on 16 November 2022)). This principle is stated as "a defined endpoint". Commonly, AMPs are annotated as such regardless of the target, mechanism, source, the method used to study the activity, to name some characteristics. The lack of such detailed information makes the discrimination between AMPs and non-AMPs a largely ambiguous endpoint for data analysis. In consequence, several criteria must be introduced to better define the modelled data and thus bring reliability to the predicted outcome. Notably, the most recent AMP predictors [24–29,31] have designed their modeling approaches to break down the AMP annotation into three classes (typically antibacterial, antifungal, and antiviral peptides). This strategy is a suitable approach to fulfil the need for a defined endpoint.

Our work focused on the identification of ABPs. To this end, we extracted peptides from the StarPep database ranging between 5 and 50 residues, and whose composition contains only the 20 standard amino acids. To further refine the selection of ABPs, we only extracted those peptides annotated as active against at least one of the following targets: *Acinetobacter baumannii*, *Bacillus subtilis*, *Enterococcus faecium*, *Escherichia coli*, *Klebsiella pneumonia*, *Listeria monocytogenes*, *Pseudomonas aeruginosa*, *Streptococcus agalactiae*, and *Staphylococcus aureus*. In this way, we discarded entries that are annotated as ABPs without information of their targets, and those exclusively reported with activity against underrepresented targets in the entire database. The selected species cover a set of both Gram-positive and Gram-negative bacteria and are examples of relevant targets for therapeutic applications. The peptides labeled as non-ABP for our learning process are not annotated as antibacterial, against any target, in StarPep, but with a different function such as antifungal or anticancer, among others. This approach clearly carries the risk of mislabeling non-ABP in our dataset, due to insufficient annotation of the peptide in the original source. The pseudo-negative cases in the training data lead to a more stringent prediction of positive cases, and consequently lower false-positive rate and higher precision. The downside is the expected lower recall as the true positives can be also diminished. Nonetheless, the favourable precision is aligned with our stated goal of boosting the precision of the classifier instead of its recall or a combined metric such as accuracy or AUC.

Hence, we extracted a total of 22,707 peptides to design our training and testing schemes. This collection was partitioned into four datasets: training, development, validation, and test sets. The two first are intended for the learning process, while the others are meant for testing the models with hold-out data. The development (Dev) set was used to monitor the generalization of the models built during the optimization of the hyperparameters in the learning algorithm. Usually, the terms development and validation set are applied indistinctively to a dataset used for the above-mentioned purpose. In this work, we made a distinction between these nomenclatures and reserved the term validation for a hold-out set, i.e., peptides that are not used in any step of the learning process. The difference between the validation and the strict test set is that we built the validation set in a way that its peptides share high similarity ($\geq$90% identity) with at least one peptide in the training set (excluding identical matches). In turn, the test set was built in a way that its peptides share less than 90% identity among them, and with any peptide in the training data. Consequently, the test set comprises non-redundant peptides that are also not closely represented in our training. Challenging a peptide predictor in both scenarios, one that closely resembles the training conditions (without strict superposition), and another more distant setup, is important to assess the biasing effect on the generalization of the model due to the characteristics of the training data.

Finally, a production dataset was generated by combining the training and the development sets. The purpose of this set is to perform a final re-training of the model with an augmented dataset, while keeping the selection of descriptors and configuration of hyper-parameters as optimized with the training and development sets. Figure 1 depicts the workflow followed to obtain the four datasets.



```
For each cluster do {
        size = Get cluster site;

        if (size == 1) {
                With a probability = 0.1 {
                        Add to the TEST set;
                } else {
                        Add to the TRAINING set;
                }
        } else if (size == 2) {
                Randomly add one peptide to the TRAINING set;
                Add the other peptide to the DEVELOPMENT set;
        } else {
                Create list of peptides in the cluster;
                Shuffle the list of peptides;
                Add from the first peptide to the second quintile of the list to the TRAINING set;
                Add peptides from the second to the third quintiles of the list to the DEVELOPMENT set;
                Add remaining peptides to the VALIDATION set;
        }
}
```

Figure 1. Workflow for the preparation of the datasets. The peptides extracted from StarPep were clustered with CD-Hit and subsequently distributed among the four sets used for training and testing the predictor. The final panel of the pipeline contains information about the number of peptides in every subset as well as their classification according to StarPep.

Together with the peptide sequences and their classification as ABP or non-ABP, we also extracted, from StarPep, the information about the Gram staining type of their known targets. Accordingly, we further categorized the ABPs into three activity classes: exclusively against Gram-positive targets (Gram+), exclusively against Gram-negative targets (Gram-), and broad-spectrum peptides. The four datasets resulting from the previous splitting were also used to train and assess the secondary classifier based on the Gram staining type of the targets. For this purpose, the non-ABP peptides were removed from such datasets. Table 1 summarizes the number of peptides per type of Gram staining class in the four datasets.

Table 1. Number of peptides per type of Gram staining class in the training, development, validation, and test datasets.

	Gram+	**Gram−**	**Broad Spectrum**
Training	351	478	4983
Development	52	105	911
Validation	37	82	546
Test	27	38	315

2.2. Performance Measures

In this section, we summarize the formulations of the performance measures used to assess the different models described here. The measures are sensitivity (Sn), precision (Pr), accuracy (Acc), F1 score, and the Mathew Correlation Coefficient (MCC) [35]. All of them are formulated in terms of the elements of a binary confusion matrix: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

$$\mathrm{Sn} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

$$\mathrm{Pr} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

$$\mathrm{F1} = 2\frac{\mathrm{Sn} * \mathrm{Pr}}{\mathrm{Sn} + \mathrm{Pr}} = \frac{\mathrm{TP}}{\mathrm{TP} + \frac{1}{2}(\mathrm{FP} + \mathrm{FN})}$$

$$\mathrm{MCC} = \frac{\mathrm{TP} * \mathrm{TN} - \mathrm{FP} * \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}}$$

Besides, we define an ad-hoc measure named Fitness–Robustness Score (FRS) that is specifically used as a scoring function to tune the values of the hyper-parameters of the learning technique.

$$\mathrm{FRS} = \left(\frac{\mathrm{R_T} + \mathrm{R_{CV}} + \mathrm{R_D}}{3}\right)^2 - (\mathrm{R_T} - \mathrm{R_{CV}})^2 - (\mathrm{R_T} - \mathrm{R_D})^2$$

The FRS is a quality measure that provides a consolidated value for the performance of a particular model considering its goodness-of-fit, generalization, and robustness. The first term corresponds to the average performance in the following assessments: re-substitution (RT, fitting the training data), 10-fold cross-validation (RCV, within the training data), and generalization (RD, using the development set). The other two terms weigh the robustness of the model by measuring the deviations from the performance in training samples when the model is evaluated in hold-out data (cross-validation and development set). We formulated this ad-hoc measure as a function of another base quality measure, labelled as R, which should be evaluated in the different assessment schemes. For this study, we selected the MCC as the base measure to evaluate our fitness-robustness score. In the case of the multi-classifier trained to distinguish between the Gram staining classes, the average MCC value among the three classes was used as the base measure. The average was weighted according to the number of peptides in each class.

The FRS, when computed as a function of the MCC, has an optimum maximum value of one. We leveraged this score to identify optimum values for the hyper-parameters of the random forest [36] algorithm used to develop our models.

2.3. Machine Learning Approach and Software

The classifiers developed in this work were random forest (RF) [36] predictors, based on the implementation of this technique in the WEKA environment [37]. RF belongs to the family of ensemble methods [38] with base classifiers formed by decision trees. Recently, RF has been compared with deep learning approaches showing comparable performance for modeling AMP datasets [39]. There, the authors conclude that no definitive evidence was found to support using deep-learning approaches for this problem, knowing the increased algorithmic complexity and computational cost of these methods.

Within RF, all the trees provide a prediction for every instance entering the forest, and the unified outcome is obtained as the majority vote among all the predictions. The hyper-parameters optimized during the learning process were the number of trees, the maximum number of descriptors used to build a tree (these descriptors are taken at the beginning of the training process from the global pool of attributes), and the maximum depth of the trees. In addition, the minimum number of instances in the final leaves of the trees was fixed to 10 in the case of the main classifier (ABPnon-ABP), and to five for the multi-classifier (Gram+/Gram−/broad spectrum).

The peptide descriptors fed to the learning algorithm were computed with the ProtDCal-Suite [40] using the configuration files enclosed in the Supplementary Material. The Prot-

DCal module [41] is intended for the calculation of general-purpose and alignment-free descriptors of amino acid sequences and protein structures. These features are descriptive statistics (such as the variance, average, maximum, minimum, percentiles, etc.) of the distribution of amino acid properties (such as hydrophobicity, isoelectric point, molar weight, among others), in multiple groups of residues extracted from a given protein or peptide. The program possesses additional procedures that modify the intrinsic properties of a residue according to its vicinity in the sequence, thus adding connectivity information in the descriptors. The features derived from ProtDCal have been used by us and other authors to develop machine-learning-based predictors of posttranslational modifications [42,43], protein–protein interaction [44], enzyme-like amino acid sequences [45], residues critical for protein functions [46], and antibacterial peptides [47,48], although with smaller databases. The project files enclosed in the Supplementary Material contain the setup used to compute all the descriptors employed in this work.

2.4. Web Servers Available for ABPs Predictions

In this section, we briefly describe the most relevant state-of-the-art ABP predictors that are available via web server tools. ClassAMP was among the first methods that broke down the AMP family thus allowing the prediction of ABPs specifically [25]. This tool was trained with peptides from the CAMP database [18] and used RF and support vector machine (SVM) [49] algorithms to identify antibacterial, antifungal, and antiviral peptides.

MLAMP, a multi-label classifier of AMPs was developed using a variant of Chou's pseudo amino acid composition (PseACC) features [50] to build an RF-based classifier that firstly distinguishes AMP from non-AMPs, and then subdivides the biological activity into antibacterial, anticancer, antifungal, antiviral, and anti-HIV [26].

Similarly, the *i*AMPpred predictor combines compositional, physicochemical, and structural features into Chou's general PseACC as input variables for an SVM multi-classifier [27]. This work reunited peptides from the databases CAMPR3 [19], APD3 [17], and AntiBP2 [24]. The multi-classifier uses three categories in the outcome variable: antibacterial, antifungal, and antiviral peptides [27].

The Antimicrobial Activity Predictor (AMAP) [28], with a hierarchical multi-label classification scheme, was trained with AMPs annotated with 14 biological activities in the APD3 database and a designed subset of non-AMP. The models used amino acid composition features to feed SVM and XGboost tree [51] algorithms.

The introduction of the AMP-Scanner webserver represented a significant improvement with respect to other predictors. AMP-Scanner vr.1 consists of two RF classifiers, trained with peptides selected from multiple sources [18,52,53]. The first output of the classifier is the identification of ABPs. The second is a classifier trained to distinguish between peptides with Gram-positive or Gram-negative targets, using data of *S. aureus* and *E. coli* as reference targets. The authors refer that peptides predicted with scores within the range [0.4–0.6] for both classes should be considered as active against both types of targets (broad-spectrum peptides) [29]. On the other hand, AMP-Scanner vr.2 is based on a Deep Neural Networks (DNN) classifier fed with ABP data only, obtained from the updated version of the ADP3 database [19,30].

Very recently, AMPDiscover [31] was developed by mining AMP data from StarPep [33]. AMPDiscover encompasses several binary (active/non-active) predictors of functions such as antibacterial, antiviral, antifungal, and antiparasitic peptides. The authors analyzed the performance of RF to model the antibacterial peptides data, which agrees with our choice of this learning scheme for our models.

2.5. Experimental Determination of Antibacterial Activity

Two batches of chemically synthetized peptides from different providers (KE Biochem and the U-PEP facility at Ulm University) were used to assess antimicrobial effects. Antibacterial activity was evaluated by agar diffusion as previously described [54]. Bacteria were cultured in liquid broth at 37 °C overnight, pelleted by centrifugation, and washed in

10 mM sodium phosphate buffer. Following resuspension, optical density was determined at 600 nm and 2 × 10^7 bacteria were seeded into a Petri dish in 1% agarose. After cooling at 4 °C for 30 min, 3–5 mm holes were placed into the 1% agarose. Peptides adjusted to the desired concentration in 10 µL of buffer were filled into the agar-holes. Following incubation at 37 °C in ambient air for 3 h, plates were overlaid with 1% agarose, tryptic soy solved in 10 mM phosphate buffer. Inhibition zones in cm were determined after 16–18 h incubation time at 37 °C in 5% CO_2. LL37 at a concentration of 100 µg/mL served as positive control. Antimicrobial activity was tested on the following bacterial strains: *Bacillus subtilis*, *Streptococcus agalactiae* ATCC 12403, *Staphylococcus aureus* MRSA ATCC 43300, *Klebsiella pneumoniae* Extended Spectrum β-Lactamase (ESBL) ATCC 700603, *Pseudomonas aeruginosa* (ATCC 27853) and *Listeria monocytogenes* (ATCC BAA-679/EGD-e).

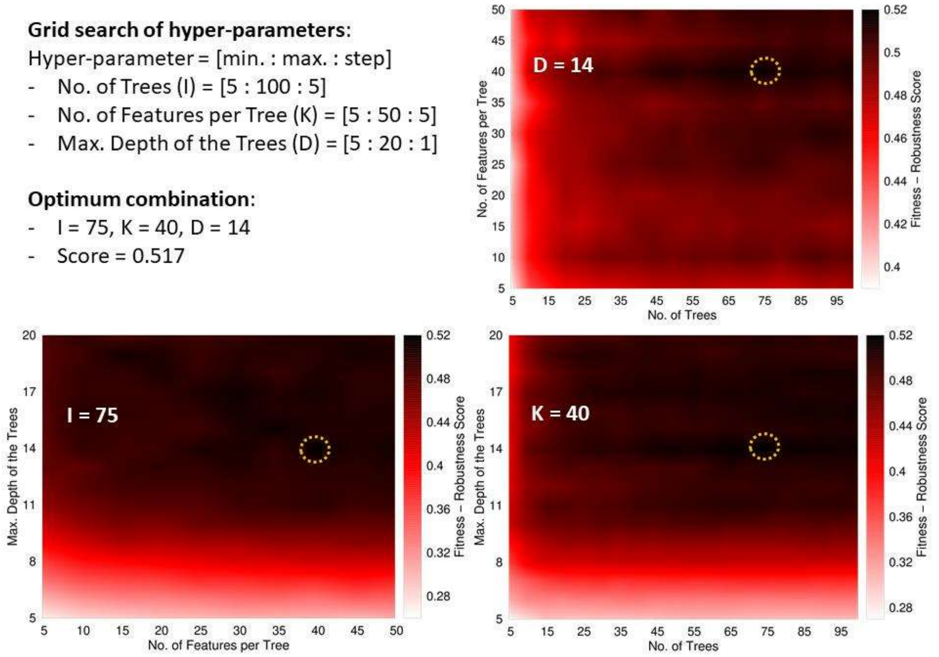
3. Results and Discussion

Below, we summarize the characteristics of the ML-based models developed in this work, as well as their performance relative to the available state-of-the-art ABP predictors. We also introduce a web server, ABP-Finder, which permits the free and user-friendly screening of large peptide libraries. Finally, we present the application of ABP-Finder for the screening of peptides obtained from the human urine peptide. Notably, ABP-Finder permitted to screen and propose a reduced set of eight ABP candidates out of an initial pool of 4696 peptides. From them, one active hit was experimentally validated with activity against *Pseudomonas aeruginosa*.

3.1. Modeling Antibacterial Peptide Data

Feature selection: The feature selection process comprises three steps. (*i*) First, the Information Gain (IG) [55,56] of all the descriptors was calculated with WEKA, retaining only those descriptors whose IG is >5% of the information content of the class variable. This procedure reduced an initial set of 11,298 descriptors to 2746, whose information contents are the most closely related to our end point variable. (*ii*) Secondly, the redundancy in this subset of features was removed, by clustering the descriptors using a quality-threshold-based [57] clustering algorithm, which employs the Spearman correlation coefficient [58] as the similarity measure to group the descriptors. A correlation cut-off of 0.9 was used to form the clusters. The outcome of these steps is thus a non-redundant and smaller dataset that contains only the central attributes of the formed clusters. This step rendered 1242 attributes. (*iii*) Given the still large set of features, a last selection step was used by employing the Wrapper Evaluator and the Classifier Subset Evaluators of WEKA coupled with a genetic search algorithm [59]. The Wrapper Evaluator used five-fold cross-validation on the training data to assess the models obtained from diverse subsets of descriptors. Such models were built with an RF whose number of trees was limited to 15. Next, the Classifier Subset Evaluator used the performance with the development set to identify the most suitable pool of descriptors to train the RF. For both evaluators, the F1 measure was used to score all the assessed subsets of attributes. The genetic search employed to explore the space of all possible combinations of attributes was configured with 20 chromosomes (subsets of attributes) per population, 500 generations, and probabilities of cross-over and mutation of 0.6 and 0.1 respectively. The optimal subset resulting from these selection steps comprised 281 descriptors. A project file type IDL (Individual Descriptor Labels) is enclosed in the Supplementary Material; this project file can be uploaded directly to ProtDCal-Suite to compute the selected 281 descriptors in new peptide datasets.

Tuning hyperparameters: The hyperparameters of the RF were explored using a grid search according to ranges and binning schemes summarized in the top-left panel of Figure 2. The ad hoc FRS function was used to determine the optimum combination of hyperparameters' values, which was obtained with 75 trees each one built from a pool of 40 descriptors and a maximum depth of 14 splits. Such combinations of values rendered the maximum FRS at 0.517.



Grid search of hyper-parameters:
Hyper-parameter = [min. : max. : step]
- No. of Trees (I) = [5 : 100 : 5]
- No. of Features per Tree (K) = [5 : 50 : 5]
- Max. Depth of the Trees (D) = [5 : 20 : 1]

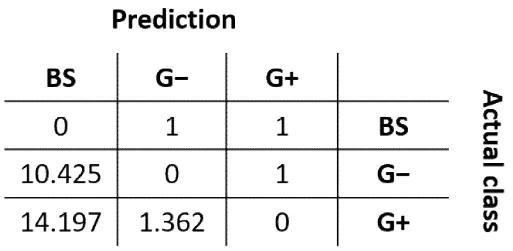
Optimum combination:
- I = 75, K = 40, D = 14
- Score = 0.517

Figure 2. Tuning scheme of the RF's hyperparameters. The top-left panel summarizes the boundaries and binning of the grid search with the three hyper-parameters. This panel also shows the optimum value found for the FRS function and the values of the hyper-parameters in the corresponding solution. The remaining panels show surfaces plotted as heat maps keeping one of the hyper-parameters fixed at its optimum value. The dark regions indicate the best solutions. The optimum regions are highlighted with a dashed circle. The plots highlight that the most critical parameter is the depth of the trees, while high-scored models can be obtained with almost any value of the other hyper-parameters; solutions with a depth below 10 are poorly scored.

3.2. Modeling Data of Gram-Staining Types

This model was trained with the same set of 281 descriptors obtained from the feature selection procedure to discriminate between ABPs and non-ABPs. The training, development, validation, and test sets used for this model were obtained from the splitting described in the Methods section, by removing the non-ABP present in these datasets. The ABPs were then subdivided according to the Gram-staining type of their known targets.

Due to the imbalance in the number of instances from each class, the cost-sensitive RF multi-classifier was trained by applying a cost matrix in the training process with distinct weights for the different types of misclassified cases. The cost matrix takes the form shown in Figure 3.



Prediction

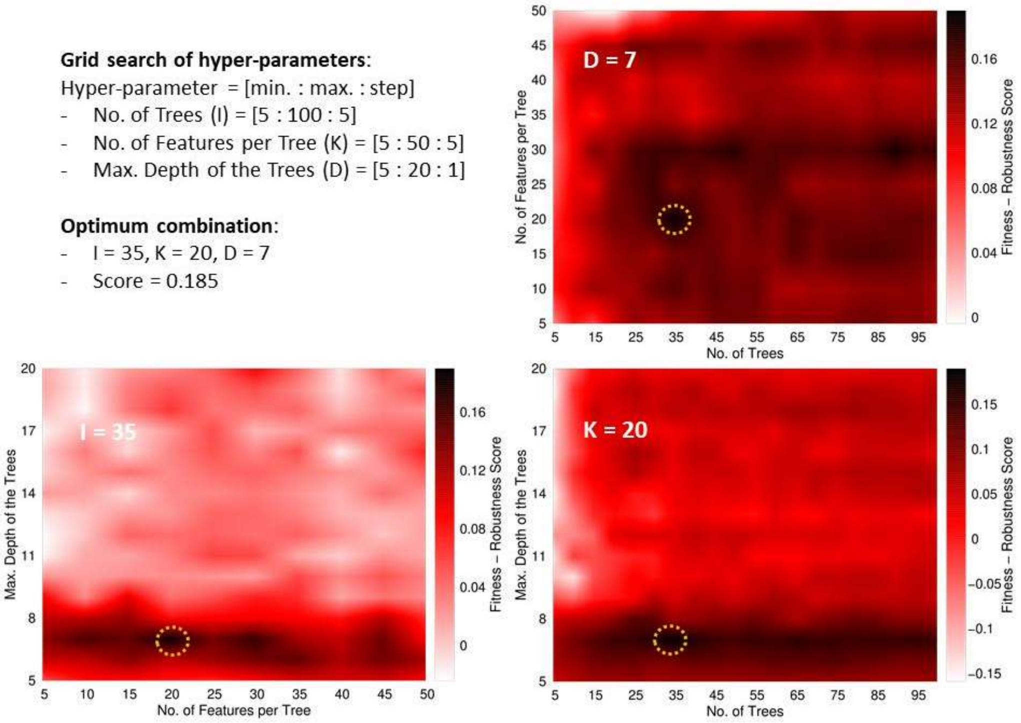
BS	G−	G+	**Actual class**
0	1	1	**BS**
10.425	0	1	**G−**
14.197	1.362	0	**G+**

Figure 3. Cost matrix applied during the training process of the multi-classifier based on the Gram-staining types of the targets.

The multi-class classifier was built with a cost-sensitive learning scheme, which aims to balance the effective error between pairs of classes considering their different prevalence in the training data. The costs were defined as the inverse ratio of the imbalance between the two classes involved in the matrix element, i.e., given the imbalance between Gram+ and broad-spectrum (BS) peptides in the training data is [1:14.197], then the cost of a Gram+ peptide classified as BS was fixed at 14.197 and the cost of a BS peptide classified as Gram+ remained at 1. This approach diminishes the trend towards BS predictions that originates due to the highest representation of this class in the training data.

The costs affect the training process by re-weighting the training samples in the calculation of the different misclassification errors during the training. No re-weighting is applied to the instances in the test datasets.

Tuning hyperparameters: Analogous to the previous model, the hyper-parameters of the RF were explored using a grid search with the ranges and binning schemes summarized in the top-left panel of Figure 4. The FRS function rendered a maximum value for a solution with 35 trees, 20 descriptors per tree, and a maximum depth of 7 splits. Such combinations of values rendered the maximum FRS at 0.185. The lower value of the optimum FRS value, compared with the ABP/non-ABP model, indicates the larger difficulty of discriminating between the three classes of Gram-staining types. Such difficulty is a natural consequence of the overlap between the classes, given that the peptides in the broad-spectrum category should gather intrinsic features of the other two classes.



Grid search of hyper-parameters:
Hyper-parameter = [min. : max. : step]
- No. of Trees (I) = [5 : 100 : 5]
- No. of Features per Tree (K) = [5 : 50 : 5]
- Max. Depth of the Trees (D) = [5 : 20 : 1]

Optimum combination:
- I = 35, K = 20, D = 7
- Score = 0.185

Figure 4. Tuning scheme of the RF's hyper-parameters. The top-left panel summarizes the boundaries and binning of the grid search with the three hyper-parameters. This panel also shows the optimum value found for the FRS function and the hyper-parameters' values of the corresponding solution. The remaining panels show surfaces plotted as heat maps keeping one of the hyper-parameters fixed at its optimum value. The dark regions indicate the best solutions. The optimum regions are highlighted with a dashed circle. As in the exploration for the model ABPs/non-ABPs, the plots show that the most critical parameter is the depth of the trees. Nonetheless, the opposite trend is observed because high-scored models are only obtained with low (<8) depth values. The smaller size of the dataset for this model, as compared with the previous one, leads to the occurrence of overfitting when deep trees are trained.

3.3. Applicability Domain

Following the regulatory principles for QSAR models established by the OECD, we discuss the applicability domain (AD) of our models. Both of our models were built using peptides with lengths between 5 and 50 residues and containing exclusively the 20 standard amino acids. Thus, these length and composition boundaries constitute soft limits of our applicability domain. A quantitative approach for the AD is provided via the range of the descriptors' values in the training or production dataset. In the Supplementary Material, we provide the minimum and maximum values of the descriptors in these datasets. As part of the implementation of these predictors, we automatically evaluate whether any new peptide is found within these ranges or not. If any of the descriptor values of a new peptide falls outside the training ranges, this peptide is labelled as an outlier and the corresponding information is given in the outcome of the program.

3.4. Performance of ABP-Finder in the Context of the State-of-the-Art

Predictors of antibacterial peptides: We compare the performance of our models to five ML-based ABP predictors by employing the hold-out validation and test sets, respectively (Tables 2 and 3). In addition, we employ an external test set originally used by Veltri et al. [30] to assess the performance of AMP-Scanner vr2 (Table 4). We present the performance of our models obtained with the training data only, and with the production dataset. Additionally, we show the performance of our tool considering only those instances that are within the AD of our models.

Table 2. Comparison with external predictors in the validation set. The values in bold denote the best performance for a given measure.

Webserver	Algorithm	Pr.	Sn.	Acc.
ClassAMP	SVM	0.46	0.33	0.59
MLAMP	RF	0.48	0.82	0.59
iAMPred	SVM	0.48	0.90	0.58
AMPScanner_v1 #	RF	0.50	0.98	0.61
AMPScanner_v2 *	DNN	0.48	0.97	0.58
AMPDiscover	RF	0.50	**0.99**	0.61
ABP-Finder (Training)	RF	0.72	0.95	0.84
ABP-Finder (Training, AD)	RF	0.70	0.95	0.83
ABP-Finder (Production)	RF	**0.75**	0.95	**0.85**
ABP-Finder (Production, AD)	RF	**0.75**	0.95	**0.85**

AD: only instances within our applicability domain are considered as valid predictions. # AMPScanner_v1 only considers peptides $\geq$ 10 AA for the predictions. * The method was updated on 20.02.2020.

Table 3. Comparison with external predictors in the test set. The values in bold denote the best performance for a given measure.

Webserver	Algorithm	Pr.	Sn.	Acc.
ClassAMP	SVM	0.34	0.41	0.61
MLAMP	RF	0.38	0.77	0.59
*i*AMPred	SVM	0.36	0.81	0.56
AMPScanner vr.1 #	RF	0.50	0.80	0.68
AMPScanner vr.2 *	DNN	0.37	0.84	0.57
AMPDiscover	RF	0.42	**0.94**	0.62
ABP-Finder (Training)	RF	0.77	0.68	0.86
ABP-Finder (Training, AD)	RF	0.78	0.67	0.86
ABP-Finder (Production)	RF	**0.80**	0.71	**0.87**
ABP-Finder (Production, AD)	RF	**0.80**	0.70	**0.87**

AD: only instances within our applicability domain are considered valid predictions. # AMPScanner_v1 only considers peptides $\geq$ 10 AA for the predictions. * The method was updated on 20 February 2020.

Table 4. Comparison with external predictors in the test set built by Veltri et al. [30]. Redundant instances with our training set were removed. The values in bold denote the best performance for a given measure.

Webserver	Algorithm	Pr.	Sn.	Acc.
ClassAMP	SVM	0.36	0.27	0.66
MLAMP	RF	0.51	0.65	0.72
*i*AMPred	SVM	0.74	**0.90**	0.88
AMPScanner vr.1 #	RF	0.64	0.77	0.81
AMPScanner vr.2 *	DNN	0.82	0.89	**0.91**
AMPDiscover	RF	0.83	0.84	**0.91**
ABP-Finder (Training)	RF	0.83	0.43	0.81
ABP-Finder (Training, AD)	RF	0.83	0.51	0.86
ABP-Finder (Production)	RF	**0.84**	0.48	0.83
ABP-Finder (Production, AD)	RF	**0.84**	0.57	0.86

AD: only instances within our applicability domain are considered valid predictions. # AMPScanner_v1 only considers peptides $\geq$ 10 AA for the prediction. * Performance based on the model from the original training in Veltri et al. [30], where the cases in this test set are held out of the training process.

Tables 2 and 3 show that our models achieved the best precision and global accuracy in the test and validation sets. Particularly, the precision was significantly higher with ABP-Finder with respect to the other methods. This is a key feature to be leveraged when filtering large peptide libraries because the main aim during the screenings for new hits is to avoid false-positive predictions.

We also challenged our models with an external test set designed by Veltri et al. [30] (Table 4) to further assess the robustness of our predictions. This dataset is qualitatively different from our test set since it is not derived from the StarPep database as all our data, and therefore it was not subjected to any of the curation procedures carried out by the StarPep's developers.

These comparisons confirm that our RF-based models render the most precise predictions, although the sensitivity (and consequently the global accuracy) decays in this case compared with other ABP predictors. Nevertheless, we note the importance of a low false-positive rate in virtual screening analyses, which highlights the higher practical value of our predictors.

Predictors of Gram-staining types: Our antibacterial predictor was designed to provide an estimation of against which type of bacteria are the peptides active. Therefore, we tested how our multi-classifier performs for the Gram+, Gram−, and Broad-Spectrum classes compared to AMP-Scanner vr.1. Tables 5 and 6 summarize the comparison with respect to precision and sensitivity of our models and AMP-Scanner vr.1 on the validation and test sets, respectively. The performance measures were computed for the three classes (Gram+, Gram−, and Broad Spectrum).

Table 5. Comparison of ABP-Finder with AMP-Scanner_v1 in the discrimination between Gram-staining classes within the validation set. The values in bold denote the best performance for a given measure.

Method	Gram+		Gram−		Broad Spectrum	
	Pr	Sn	Pr	Sn	Pr	Sn
AMPScanner vr.1 #	0.04	0.19	0.16	0.42	0.81	0.27
ABP-Finder (Training *)	**0.63**	**0.73**	**0.91**	0.38	0.90	**0.97**
ABP-Finder (Production *)	0.62	0.70	0.85	**0.48**	**0.91**	0.96

AD: only instances within our applicability domain are considered valid predictions. # AMPScanner_v1 only considers peptides $\geq$ 10 AA for the predictions. * There are no instances outside the AD of the model.

Table 6. Comparison of ABP-Finder with AMP-Scanner_v1 in the discrimination between Gram-staining classes within the test set. The values in bold denote the best performance for a given measure.

Method	Gram+		Gram−		Broad Spectrum	
	Pr	Sn	Pr	Sn	Pr	Sn
AMPScanner vr.1 #	0.08	**0.42**	0.13	**0.33**	**0.88**	0.23
ABP-Finder (Training)	**0.44**	0.41	**0.90**	0.24	0.87	**0.96**
ABP-Finder (Training, AD)	**0.44**	0.39	**0.90**	0.24	0.87	**0.96**
ABP-Finder (Production)	**0.44**	0.41	0.82	0.24	**0.88**	**0.96**
ABP-Finder (Production, AD)	**0.44**	0.39	0.82	0.24	**0.88**	**0.96**

AMPScanner_v1 only considers peptides ≥ 10 AA for the predictions.

Our models largely outperformed AMP-Scanner vr.1, particularly in terms of precision when detecting the specific types of Gram-staining types (Gram+ and Gram−). Regarding the prediction of broad-spectrum peptides, both methodologies delivered the same precision. However, in this case we greatly surpassed the sensitivity of AMP-Scanner vr.1, thus making more accurate predictions overall. Notably, our multi-classifier showed the best performance for the three classes of Gram-staining types, thus providing a valuable complement to the identification of antibacterial peptides.

The comparison with the state-of-the-art tools showed that, together with ABP-Finder, the top-ranked methods in our tests were *i*AMPred, AMP-Scanner vr2, and AMPDiscover. These approaches were thus confirmed as suitable tools for ABP identification. Nonetheless, ABP-Finder outperformed these predictors, particularly in terms of precision. Importantly, as a distinctive feature, we complement our outcome with an estimation of the Gram-staining type of the putative targets, which can be further pinned down to specific bacterial species by considering that our models were trained with data from nine representative targets (see Dataset section). Furthermore, unlike previously published tools [24–30], we provide an estimation of our applicability domain, which delivers reliability to the predicted outcome.

3.5. ABP-Finder Web Server

Our emphasis in the application of regulatory principles to the development of ML-based predictors relies on our commitment to offer a freely accessible and well-maintained tool to reliably screen peptide libraries. To this end, we implemented our models in a user-friendly web server named ABP-Finder (https://protdcal.zmb.uni-due.de/ABP-Finder/ (accessed on 16 November 2022)). This tool allows screening seamlessly thousands of peptides with a single submission job. The ABP-Finder server delivers for each entry a prediction of the antibacterial function, as well as whether each specific peptide is or not within the AD of our models. ABP predictions are also accompanied by a Gram-staining-based estimation of the putative targets of the antibacterial peptides. Furthermore, the web server offers the functionality of screening regions within a long amino acid sequence to identify promising antibacterial fragments. This application of ABP-Finder's models was recently leveraged by us for the identification of antibacterial motifs within β2-microglobulin [60].

3.6. Virtual Screening of the Human Urine Peptidome

In this section, we describe the successful application of ABP-Finder to screen a peptide library obtained from the human urine peptidome. The library contains 4696 endogenous peptide fragments, detected in the Core Facility Functional Peptidomics at the University Hospital in Ulm, Germany. The peptide library was screened for antibacterial activity following the workflow depicted in Figure 5.

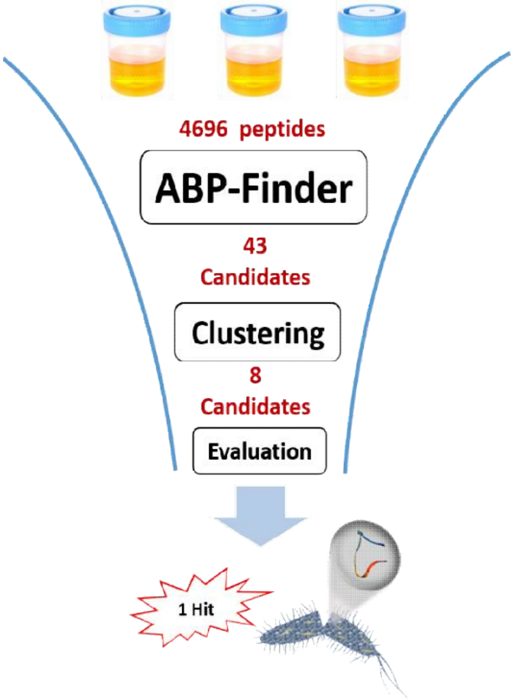


Figure 5. Schematic representation of the virtual screening process carried out on a library of peptides from the human urine peptidome.

ABP-Finder was used to score the original 4696 peptides of the library, obtaining 43 candidates with a probability score larger than 0.6, and within the applicability domain of the model. Subsequently, Blastp [61] was used to cross-align these peptides with known ABPs of our training samples. From there, we excluded two hits that showed 100% identity and coverage in the alignment with previously reported ABPs and therefore did not have value as newly identified peptides. Afterward, we clustered the peptide sequences using CD-Hit [62] with a cut-off of 90% of identity, and minimum coverage of the shortest sequence in the alignment of 90%. From this analysis, eleven clusters were obtained, from which we extracted the shortest sequence as representative of each cluster. Three polyproline peptides, containing none or only one residue other than proline were finally discarded because we considered them unsuitable as candidates for possible lead compounds due to synthetic unfeasibility and the highly homogenous character of their sequences. The final eight candidates (Table 7) were experimentally evaluated using an agar diffusion assay, leading to one active hit, Urine-3462, against *Pseudomonas aeruginosa.*

Table 7. The resulting eight ABP candidates from the virtual human urine peptidome screening and some of its global sequence descriptors. Global peptide descriptors were calculated using the Peptide Design and Analysis Under Galaxy (PDAUG) package [63].

Peptide	Sequence	Length	pI	Total Charge #	Global Hydrophobicity *	GRAVY Index &
U2162	KKVLGAFSDGLAHLDNLKGT	20	10.42	1.09	0.08	−0.12
U687	DKTNVKAAWGKVGAHAGEYGAE	22	9.53	0.10	0.01	−0.73
U4507	WLKEGVLGLVHEF	13	7.70	−0.90	0.39	0.52
U3462	RVDPVNFKLLSHCLLVT	17	10.03	1.03	0.18	0.67
U2125	KAVGKVIPELNGKLTGM	17	10.99	1.99	0.15	0.12
U1930	IAGVGAEILNVAKGIRSF	18	11.40	0.99	0.35	0.92
U1982	IFVKTLTGKTI	11	13.0	1.99	0.32	0.86
U2273	KVVAGVANALAHK	13	13.0	2.09	0.24	0.67

Total Molecular Charge given at pH = 7. * Eisenberg scale. & GRAVY (Grand Average of Hydropathy) is calculated as the sum of hydropathy values of all the amino acids, divided by the number of residues in the sequence [64]. Positive GRAVY values indicate hydrophobic; negative values mean hydrophilic.

3.7. Experimental Evaluation of the Reduced Set of Peptides from the Human Urine Peptidome

To test the antimicrobial potential of the eight candidate peptides identified with ABP-Finder, a radial diffusion assay was carried out, allowing the sensitive detection of antibacterial activity. Activity was determined against various Gram-positive and Gram-negative bacteria species, including *Bacillus subtilis*, *Streptococcus agalactiae*, *Staphylococcus aureus* (MRSA), *Escherichia coli*, *Pseudomonas aeruginosa*, *Klebsiella pneumoniae* (ESBL). While the peptide Urine-3462 was active against *Pseudomonas aeruginosa*, no relevant antibacterial activity could be detected at concentrations of 100 µg/mL and 1 mg/mL of the other peptides. Urine-3462 exhibited a dose-dependent growth of inhibition of *Pseudomonas aeruginosa*, comparable to the inhibitory activity observed for the well described antimicrobial peptide LL37 [54,65], which served as a positive control (Figure 6).

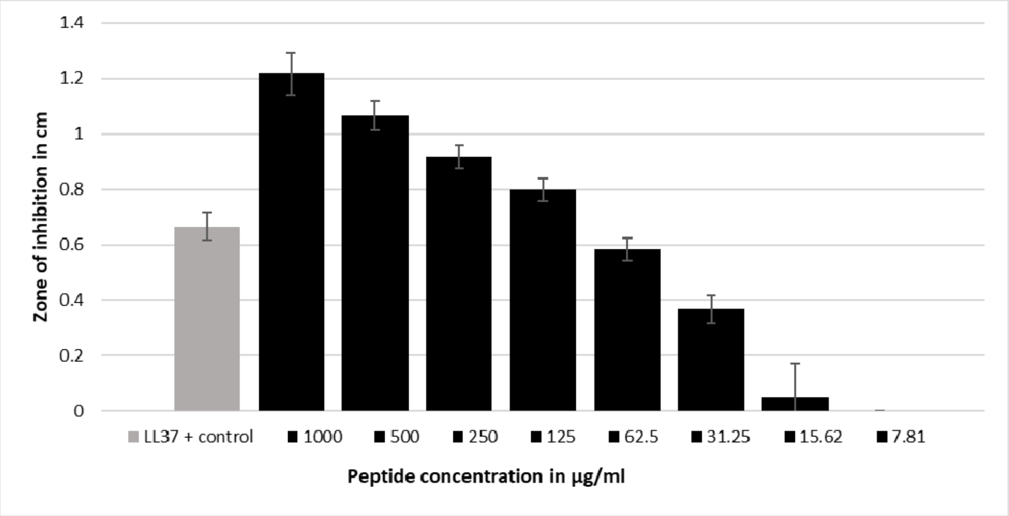


Figure 6. A radial diffusion assay indicated that the peptide Urine-3462 is active against the *Pseudomonas aeruginosa* strain ATCC 27853. Inhibition zones are quantified in cm. The mean values and standard deviations of six independent experiments are shown. LL37 at 100 µg/mL was used as positive control (see Table S3 for exact values).

4. Conclusions

Antibacterial peptides are promising candidates for a new generation of antibiotics designed to address the challenging problem of drug resistance in bacteria. With ABP-Finder we provide a tool that delivers top-ranked predictions as established by several comparisons with prominent examples of the state-of-the-art ABP predictors. Remarkably, ABP-Finder produces the most precise predictions in validation tests with known data. Furthermore, unlike other tools of the state-of-the-art that were used for comparison in this work, we present a successful application of the method in a real-life scenario dealing with the massive screening of unlabeled peptides from the human urine peptidome.

We implemented this RF-based predictor in the user-friendly and freely accessible web server ABP-Finder, which was also leveraged in the identification of the new ABP hit from a large library of peptides derived from the human peptidome.

In this way, the combination of in silico screening and experiments confirmed the applicability of ABP-Finder as a screening tool for the early steps of the design of peptide-based antibiotics. To the best of our knowledge, no other publicly available ABP predictor has delivered a similar study leading to the successful identification of an active hit from tens of thousands of unlabeled peptides. Further developments of our predictor will include its combination with target-specific models. This will allow improving the design of broad-spectrum candidates, as well as to orient the selection of targets in massive screenings of bioactive peptides.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/antibiotics11121708/s1, The Supporting Information is available free of charge and includes the project files containing the setup used to compute all the descriptors employed in this work, and the AD of the datasets.

Author Contributions: Y.B.R.-B. worked on the conceptualization, data curation, formal analysis, methodology, project administration, validation, visualization and writing of the manuscript. G.A.-C. and A.A. worked mainly on the conceptualization, formal analysis, funding acquisition, supervision, validation, writing and reviewing the manuscript. S.R.-M. carried out the software development. L.-R.O., B.S. and J.M. were responsible for experimental investigation, resources, supervision, validation and revision of the manuscript. E.S.-G. was responsible for the conceptualization, funding acquisition, resources, supervision, writing and reviewing the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the German Research Foundation (DFG) through the CRC 1279-Project number 316249678 to E.S-G., J.M. and B.-S., E.S.-G. was also supported by the DFG under Germany´s Excellence Strategy—EXC 2033—390677874—RESOLV, by the DFG—Project-ID: 436586093 and by the CRC 1430—Project-ID: 424228829. GACh and AA were supported by the Strategic Funding UIDB/04423/2020 and UIDP/04423/2020 through national funds provided by the Portuguese Foundation for Science and Technology (Fundação para a Ciência e a Tecnologia—FCT).

Data Availability Statement: The web server presented in this manuscript, which evaluates the described models for ABP prediction and Gram staining type classification, is freely accessible at: https://protdcal.zmb.uni-due.de/ABP-Finder/index.php (accessed on 16 November 2022). The StarPep database, which was the source for all the in-silico data used to train and validate our models is accessible at: http://mobiosd-hub.com/starpep (accessed on 4 February 2020). WEKA was the machine-learning framework used for feature selection, hyper-parameters optimization, model training and validation steps. This program can be downloaded and installed following the guidelines at: https://waikato.github.io/weka-wiki/downloading_weka/ (accessed on 16 November 2022). ProtDCal descriptors, used to encode the peptide sequences into numeric vectors, can be computed directly from the web server: https://protdcal.zmb.uni-due.de/pages/form.php (accessed on 16 November 2022), using the project files given as Supplementary Material of this manuscript. The project files gather all the configuration of parameters used to obtain the initial set of descriptors screened in this work. The training, development, validation, and test datasets, as well as the boundaries of the applicability domains for the training and production models are included as part of the Supplementary Material of this work. The assessment of the applicability domain is also a feature implemented in our web server (ABP-Finder), therefore it is automatically done and reported by the server for any evaluated peptide.

Acknowledgments: Y-B.R.-B. acknowledges the CRC 1279 of the University Hospital Ulm for a Creative Young Researcher Award for the development of algorithms to identify bioactive peptides. L.-R.O. is part of and would like to acknowledge the International Graduate School in Molecular Medicine Ulm (IGradU).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Talebi Bezmin Abadi, A.; Rizvanov, A.A.; Haertlé, T.; Blatt, N.L. World Health Organization Report: Current Crisis of Antibiotic Resistance. *BioNanoScience* **2019**, *9*, 778–788. [CrossRef]
2. Antimicrobial Resistance, C. Global burden of bacterial antimicrobial resistance in 2019: A systematic analysis. *Lancet* **2022**, *399*, 629–655. [CrossRef]
3. Yeaman, M.R.; Yount, N.Y. Mechanisms of antimicrobial peptide action and resistance. *Pharmacol. Rev.* **2003**, *55*, 27–55. [CrossRef] [PubMed]
4. Guevara Agudelo, A.; Muñoz Molina, M.; Navarrete Ospina, J.; Salazar Pulido, L.; Castro-Cardozo, B. New Horizons to Survive in a Post-Antibiotics Era. *J. Trop. Med. Health* **2018**, *10*, JTMH-130. [CrossRef]
5. Breijyeh, Z.; Jubeh, B.; Karaman, R. Resistance of Gram-Negative Bacteria to Current Antibacterial Agents and Approaches to Resolve It. *Molecules* **2020**, *25*, 1340. [CrossRef] [PubMed]
6. Bahar, A.; Ren, D. Antimicrobial peptides. *Pharmaceuticals* **2013**, *6*, 1543–1575. [CrossRef]
7. Benfield, A.H.; Henriques, S.T. Mode-of-Action of Antimicrobial Peptides: Membrane Disruption vs. Intracellular Mechanisms. *Front. Med. Technol.* **2020**, *2*, 610997. [CrossRef]

8. Le, C.F.; Fang, C.M.; Sekaran, S.D. Intracellular Targeting Mechanisms by Antimicrobial Peptides. *Antimicrob. Agents Chemother.* **2017**, *61*, e02340-16. [CrossRef]
9. Cudic, M.; Otvos, L., Jr. Intracellular targets of antibacterial peptides. *Curr. Drug Targets* **2002**, *3*, 101–106. [CrossRef]
10. Cruz, J.; Ortiz, C.; Guzman, F.; Fernandez-Lafuente, R.; Torres, R. Antimicrobial peptides: Promising compounds against pathogenic microorganisms. *Curr. Med. Chem.* **2014**, *21*, 2299–2321. [CrossRef]
11. Henninot, A.; Collins, J.C.; Nuss, J.M. The Current State of Peptide Drug Discovery: Back to the Future? *J. Med. Chem.* **2018**, *61*, 1382–1414. [CrossRef]
12. Marr, A.K.; Gooderham, W.J.; Hancock, R.E. Antibacterial peptides for therapeutic use: Obstacles and realistic outlook. *Curr. Opin. Pharmacol.* **2006**, *6*, 468–472. [CrossRef]
13. Horcajada, J.P.; Montero, M.; Oliver, A.; Sorli, L.; Luque, S.; Gomez-Zorrilla, S.; Benito, N.; Grau, S. Epidemiology and Treatment of Multidrug-Resistant and Extensively Drug-Resistant Pseudomonas aeruginosa Infections. *Clin. Microbiol. Rev.* **2019**, *32*, e00031-19. [CrossRef]
14. Gonzalez-Garcia, M.; Morales-Vicente, F.; Pico, E.D.; Garay, H.; Rivera, D.G.; Grieshober, M.; Raluca Olari, L.; Gross, R.; Conzelmann, C.; Kruger, F.; et al. Antimicrobial Activity of Cyclic-Monomeric and Dimeric Derivatives of the Snail-Derived Peptide Cm-p5 against Viral and Multidrug-Resistant Bacterial Strains. *Biomolecules* **2021**, *11*, 745. [CrossRef]
15. Mahlapuu, M.; Bjorn, C.; Ekblom, J. Antimicrobial peptides as therapeutic agents: Opportunities and challenges. *Crit. Rev. Biotechnol.* **2020**, *40*, 978–992. [CrossRef]
16. Seshadri Sundararajan, V.; Gabere, M.N.; Pretorius, A.; Adam, S.; Christoffels, A.; Lehvaslaiho, M.; Archer, J.A.; Bajic, V.B. DAMPD: A manually curated antimicrobial peptide database. *Nucleic Acids Res.* **2012**, *40*, D1108–D1112. [CrossRef]
17. Wang, G.; Li, X.; Wang, Z. APD3: The antimicrobial peptide database as a tool for research and education. *Nucleic Acids Res.* **2016**, *44*, D1087–D1093. [CrossRef]
18. Thomas, S.; Karnik, S.; Barai, R.S.; Jayaraman, V.K.; Idicula-Thomas, S. CAMP: A useful resource for research on antimicrobial peptides. *Nucleic Acids Res.* **2010**, *38*, D774–D780. [CrossRef]
19. Waghu, F.H.; Barai, R.S.; Gurung, P.; Idicula-Thomas, S. CAMPR3: A database on sequences, structures and signatures of antimicrobial peptides. *Nucleic Acids Res.* **2016**, *44*, D1094–D1097. [CrossRef]
20. Zhao, X.; Wu, H.; Lu, H.; Li, G.; Huang, Q. LAMP: A Database Linking Antimicrobial Peptides. *PLoS ONE* **2013**, *8*, e66557. [CrossRef]
21. Fan, L.; Sun, J.; Zhou, M.; Zhou, J.; Lao, X.; Zheng, H.; Xu, H. DRAMP: A comprehensive data repository of antimicrobial peptides. *Sci. Rep.* **2016**, *6*, 24482. [CrossRef] [PubMed]
22. Lee, H.T.; Lee, C.C.; Yang, J.R.; Lai, J.Z.; Chang, K.Y. A large-scale structural classification of antimicrobial peptides. *BioMed Res. Int.* **2015**, *2015*, 475062. [CrossRef] [PubMed]
23. Pirtskhalava, M.; Amstrong, A.A.; Grigolava, M.; Chubinidze, M.; Alimbarashvili, E.; Vishnepolsky, B.; Gabrielian, A.; Rosenthal, A.; Hurt, D.E.; Tartakovsky, M. DBAASP v3: Database of antimicrobial/cytotoxic activity and structure of peptides as a resource for development of new therapeutics. *Nucleic Acids Res.* **2021**, *49*, D288–D297. [CrossRef]
24. Lata, S.; Mishra, N.K.; Raghava, G.P. AntiBP2: Improved version of antibacterial peptide prediction. *BMC Bioinform.* **2010**, *11* (Suppl. 1), S19. [CrossRef] [PubMed]
25. Joseph, S.; Karnik, S.; Nilawe, P.; Jayaraman, V.K.; Idicula-Thomas, S. ClassAMP: A prediction tool for classification of antimicrobial peptides. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2012**, *9*, 1535–1538. [CrossRef]
26. Lin, W.; Xu, D. Imbalanced multi-label learning for identifying antimicrobial peptides and their functional types. *Bioinformatics* **2016**, *32*, 3745–3752. [CrossRef]
27. Meher, P.K.; Sahu, T.K.; Saini, V.; Rao, A.R. Predicting antimicrobial peptides with improved accuracy by incorporating the compositional, physico-chemical and structural features into Chou's general PseAAC. *Sci. Rep.* **2017**, *7*, 42362. [CrossRef]
28. Gull, S.; Shamim, N.; Minhas, F. AMAP: Hierarchical multi-label prediction of biologically active and antimicrobial peptides. *Comput. Biol. Med.* **2019**, *107*, 172–181. [CrossRef]
29. Veltri, D.P. *A Computational and Statistical Framework for Screening Novel Antimicrobial Peptides*; George Mason University: Fairfax, VA, USA, 2015.
30. Veltri, D.; Kamath, U.; Shehu, A. Deep learning improves antimicrobial peptide recognition. *Bioinformatics* **2018**, *34*, 2740–2747. [CrossRef]
31. Pinacho-Castellanos, S.A.; García-Jacas, C.R.; Gilson, M.K.; Brizuela, C.A. Alignment-Free Antimicrobial Peptide Predictors: Improving Performance by a Thorough Analysis of the Largest Available Data Set. *J. Chem. Inf. Model.* **2021**, *61*, 3141–3157. [CrossRef]
32. Aguilera-Mendoza, L.; Marrero-Ponce, Y.; Garcia-Jacas, C.R.; Chavez, E.; Beltran, J.A.; Guillen-Ramirez, H.A.; Brizuela, C.A. Automatic construction of molecular similarity networks for visual graph mining in chemical space of bioactive peptides: An unsupervised learning approach. *Sci. Rep.* **2020**, *10*, 18074. [CrossRef]
33. Aguilera-Mendoza, L.; Marrero-Ponce, Y.; Beltran, J.A.; Tellez Ibarra, R.; Guillen-Ramirez, H.A.; Brizuela, C.A. Graph-based data integration from bioactive peptide databases of pharmaceutical interest: Toward an organized collection enabling visual network analysis. *Bioinformatics* **2019**, *35*, 4739–4747. [CrossRef]

34. OECD. *Guidance Document on the Validation of (Quantitative) Structure-Activity Relationship [(Q)SAR] Models*; OECD: Paris, France, 2014.
35. Chicco, D.; Jurman, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genom.* **2020**, *21*, 6. [CrossRef]
36. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
37. Witten, I.H.; Frank, E.; Hall, M.A.; Pal, C.J. *Data Mining, Fourth Edition: Practical Machine Learning Tools and Techniques*, 4th ed.; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2016.
38. Dietterich, T.G. *Ensemble Methods in Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 1–15.
39. Garcia-Jacas, C.R.; Pinacho-Castellanos, S.A.; Garcia-Gonzalez, L.A.; Brizuela, C.A. Do deep learning models make a difference in the identification of antimicrobial peptides? *Brief. Bioinform.* **2022**, *23*, bbac094. [CrossRef]
40. Romero-Molina, S.; Ruiz-Blanco, Y.B.; Green, J.R.; Sanchez-Garcia, E. ProtDCal-Suite: A web server for the numerical codification and functional analysis of proteins. *Protein Sci.* **2019**, *28*, 1734–1743. [CrossRef]
41. Ruiz-Blanco, Y.B.; Paz, W.; Green, J.; Marrero-Ponce, Y. ProtDCal: A program to compute general-purpose-numerical descriptors for sequences and 3D-structures of proteins. *BMC Bioinform.* **2015**, *16*, 162. [CrossRef]
42. Biggar, K.K.; Charih, F.; Liu, H.; Ruiz-Blanco, Y.B.; Stalker, L.; Chopra, A.; Connolly, J.; Adhikary, H.; Frensemier, K.; Hoekstra, M.; et al. Proteome-wide Prediction of Lysine Methylation Leads to Identification of H2BK43 Methylation and Outlines the Potential Methyllysine Proteome. *Cell Rep.* **2020**, *32*, 107896. [CrossRef] [PubMed]
43. Ruiz-Blanco, Y.B.; Marrero-Ponce, Y.; García-Hernández, E.; Green, J. Novel "extended sequons" of human N-glycosylation sites improve the precision of qualitative predictions: An alignment-free study of pattern recognition using ProtDCal protein features. *Amino Acids* **2017**, *49*, 317–325. [CrossRef]
44. Romero-Molina, S.; Ruiz-Blanco, Y.B.; Harms, M.; Münch, J.; Sanchez-Garcia, E. PPI-Detect: A support vector machine model for sequence-based prediction of protein–protein interactions. *J. Comput. Chem.* **2019**, *40*, 1233–1242. [CrossRef]
45. Ruiz-Blanco, Y.B.; Agüero-Chapin, G.; García-Hernández, E.; Álvarez, O.; Antunes, A.; Green, J. Exploring general-purpose protein features for distinguishing enzymes and non-enzymes within the twilight zone. *BMC Bioinform.* **2017**, *18*, 349. [CrossRef] [PubMed]
46. Corral-Corral, R.; Beltrán, J.A.; Brizuela, C.A.; Del Rio, G. Systematic Identification of Machine-Learning Models Aimed to Classify Critical Residues for Protein Function from Protein Structure. *Molecules* **2017**, *22*, 1673. [CrossRef]
47. Kleandrova, V.V.; Ruso, J.M.; Speck-Planche, A.; Cordeiro, M.N.D.S. Enabling the Discovery and Virtual Screening of Potent and Safe Antimicrobial Peptides. Simultaneous Prediction of Antibacterial Activity and Cytotoxicity. *ACS Comb. Sci.* **2016**, *18*, 490–498. [CrossRef] [PubMed]
48. Speck-Planche, A.; Kleandrova, V.V.; Ruso, J.M.; Cordeiro, M.N.D.S. First Multitarget Chemo-Bioinformatic Model to Enable the Discovery of Antibacterial Peptides against Multiple Gram-Positive Pathogens. *J. Chem. Inf. Model.* **2016**, *56*, 588–598. [CrossRef]
49. Hearst, M.A. Support Vector Machines. *IEEE Intell. Syst.* **1998**, *13*, 18–28. [CrossRef]
50. Chou, K.C. Prediction of protein cellular attributes using pseudo-amino acid composition. *Proteins Struct. Funct. Bioinform.* **2001**, *43*, 246–255. [CrossRef]
51. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
52. Xiao, X.; Wang, P.; Lin, W.Z.; Jia, J.H.; Chou, K.C. iAMP-2L: A two-level multi-label classifier for identifying antimicrobial peptides and their functional types. *Anal. Biochem.* **2013**, *436*, 168–177. [CrossRef]
53. Fernandes, F.C.; Rigden, D.J.; Franco, O.L. Prediction of antimicrobial peptides based on the adaptive neuro-fuzzy inference system application. *Biopolymers* **2012**, *98*, 280–287. [CrossRef]
54. Vicente, F.E.M.; Gonzalez-Garcia, M.; Diaz Pico, E.; Moreno-Castillo, E.; Garay, H.E.; Rosi, P.E.; Jimenez, A.M.; Campos-Delgado, J.A.; Rivera, D.G.; Chinea, G.; et al. Design of a Helical-Stabilized, Cyclic, and Nontoxic Analogue of the Peptide Cm-p5 with Improved Antifungal Activity. *ACS Omega* **2019**, *4*, 19081–19095. [CrossRef]
55. Kent, J.T. Information gain and a general measure of correlation. *Biometrika* **1983**, *70*, 163–173. [CrossRef]
56. Lee, C.; Lee, G.G. Information gain and divergence-based feature selection for machine learning-based text categorization. *Inf. Process. Manag.* **2006**, *42*, 155–165. [CrossRef]
57. Heyer, L.J.; Kruglyak, S.; Yooseph, S. Exploring Expression Data: Identification and Analysis of Coexpressed Genes. *Genome Res.* **1999**, *9*, 1106–1115. [CrossRef]
58. Spearman Rank Correlation Coefficient. In *The Concise Encyclopedia of Statistics*; Springer: New York, NY, USA, 2008; pp. 502–505.
59. Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed.; Addison-Wesley Longman Publishing Co. Inc.: Boston, MA, USA, 1989.
60. Holch, A.; Bauer, R.; Olari, L.-R.; Rodriguez, A.A.; Ständker, L.; Preising, N.; Karacan, M.; Wiese, S.; Walther, P.; Ruiz-Blanco, Y.B.; et al. Respiratory β-2-Microglobulin exerts pH dependent antimicrobial activity. *Virulence* **2020**, *11*, 1402–1414. [CrossRef]
61. Altschul, S.F.; Gish, W.; Miller, W.; Myers, E.W.; Lipman, D.J. Basic local alignment search tool. *J. Mol. Biol.* **1990**, *215*, 403–410. [CrossRef]
62. Li, W.; Godzik, A. Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* **2006**, *22*, 1658–1659. [CrossRef]

63. Joshi, J.; Blankenberg, D. PDAUG: A Galaxy based toolset for peptide library analysis, visualization, and machine learning modeling. *BMC Bioinform.* **2022**, *23*, 197. [CrossRef]
64. Kyte, J.; Doolittle, R.F. A simple method for displaying the hydropathic character of a protein. *J. Mol. Biol.* **1982**, *157*, 105–132. [CrossRef]
65. Overhage, J.; Campisano, A.; Bains, M.; Torfs, E.C.; Rehm, B.H.; Hancock, R.E. Human host defense peptide LL-37 prevents bacterial biofilm formation. *Infect. Immun.* **2008**, *76*, 4176–4182. [CrossRef]

Supporting Information

ABP-Finder: A tool to identify antibacterial peptides and the Gram-staining type of targeted bacteria

Yasser B. Ruiz-Blanco[1*], Guillermin Agüero-Chapin[2,3], Sandra Romero-Molina[1], Agostinho Antunes[2,3], Lia-Raluca Olari[4], Barbara Spellerberg[5], Jan Münch[4], and Elsa Sanchez-Garcia[1*]

[1] Computational Biochemistry, Center of Medical Biotechnology, University of Duisburg-Essen, Essen, Germany.

[2] CIIMAR – Centro Interdisciplinar de Investigação Marinha e Ambiental, Universidade do Porto, Terminal de Cruzeiros do Porto de Leixões, Av. General Norton de Matos, s/n, 4450-208, Portugal,

[3] Departamento de Biologia, Faculdade de Ciências, Universidade do Porto, Rua do Campo Alegre, 4169-007 Porto, Portugal

[4] Institute of Molecular Virology, University Hospital Ulm, Ulm, Germany

[5] Institute of Medical Microbiology and Hygiene, University Hospital Ulm, Ulm, Germany

* Correspondence: elsa.sanchez-garcia@uni-due.de (E.S.-G.); ybruizblanco@gmail.com (Y.B.R.-B.)

Table of contents

Table S1. Number of peptides per type of Gram staining class in the training, development, validation, and test datasets.

Section S1. ProtDCal's configuration for vicinity operator Autocorrelation of grade 1 (AC1).

Section S2. ProtDCal's configuration for vicinity operator Autocorrelation of grade 2 (AC2).

Section S3. ProtDCal's configuration for vicinity operator Electrotopological State (EC).

Section S4. ProtDCal's configuration for no vicinity operator.

Section S5. List (FASTA format) of all the peptides in the training dataset.

Section S6. List (FASTA format) of all the peptides in the development dataset.

Section S7. List (FASTA format) of all the peptides in the validation dataset.

Section S8. List (FASTA format) of all the peptides in the test dataset.

Table S2. Minimum (Min) and maximum (Max) boundaries of the applicability domains defined for the models derived from the training and the production datasets. The first model distinguishes antibacterial peptides from non-antibacterial peptides (ABP) and the second model categorizes antibacterial peptides as anti-Gram+, anti-Gram- or anti- both types of bacteria (Gram).

Table S3. Values resulting from the radial diffusion assay applied to the peptide Urine-3462 against the *Pseudomonas aeruginosa* strain ATCC 27853.

Table S1. Number of peptides per type of Gram staining class in the training, development, validation and test datasets.

	Gram+	*Gram-*	*Broad Spectrum*
Training	351	478	4983
Development	52	105	911
Validation	37	82	546
Test	27	38	315

Section S1. ProtDCal's configuration for vicinity operator Autocorrelation of grade 1 (AC1).

```
directory:
Datasets/Fasta_Protein_Format
indices:
Gs(U),Gw(U),W(U),Mw,HP,Z1,IP,Z2,ECI,ISA,Z3
groups:
AHR,PCR,ARM,NPR,BSR,NCR,ALR,PLR,RTR,UCR,UFR,PRT
invariants:
N1,N2,N3,Ar,P2,P3,K,CV,Q1,RA,DE,Q2,S,MN,Q3,V,MX,I50,SI,MI,TI
parameters(t_cont,s_cont,A%,HydGroup,n,bins,K,SubG):
4.0,8.0,5.0,9.4,3.0,5,1,3
options(decimals,harmonicMeanType,geometricMeanType,windexID,datasetType,outputOrd
er):
2,0,0,0,fasta,true
```

Section S2. ProtDCal's configuration for vicinity operator Autocorrelation of grade 2 (AC2).

```
directory:
Datasets/Fasta_Protein_Format
indices:
Gs(U),Gw(U),W(U),Mw,HP,Z1,IP,Z2,ECI,ISA,Z3
groups:
AHR,PCR,ARM,NPR,BSR,NCR,ALR,PLR,RTR,UCR,UFR,PRT
invariants:
N1,N2,N3,Ar,P2,P3,K,CV,Q1,RA,DE,Q2,S,MN,Q3,V,MX,I50,SI,MI,TI
parameters(t_cont,s_cont,A%,HydGroup,n,bins,K,SubG):
4.0,8.0,5.0,9.4,3.0,5,2,3
options(decimals,harmonicMeanType,geometricMeanType,windexID,datasetType,outputOrder):
2,0,0,0,fasta,true
```

Section S3. ProtDCal's configuration for vicinity operator Electrotopological State (EC).

```
directory:
Datasets/Fasta_Protein_Format
indices:
Gs(U),Gw(U),W(U),Mw,HP,Z1,IP,Z2,ECI,ISA,Z3
groups:
AHR,PCR,ARM,NPR,BSR,NCR,ALR,PLR,RTR,UCR,UFR,PRT
invariants:
N1,N2,N3,Ar,P2,P3,K,CV,Q1,RA,DE,Q2,S,MN,Q3,V,MX,I50,SI,MI,TI
parameters(t_cont,s_cont,A%,HydGroup,n,bins,K,SubG):
4.0,8.0,5.0,9.4,3.0,5,5,3
options(decimals,harmonicMeanType,geometricMeanType,windexID,datasetType,outputOrder):
2,0,0,4,fasta,true
```

Section S4. ProtDCal’s configuration for no vicinity operator

```
directory:
Datasets/Fasta_Protein_Format
indices:
Gs(U),Gw(U),W(U),Mw,HP,Z1,IP,Z2,ECI,ISA,Z3
groups:
AHR,PCR,ARM,NPR,BSR,NCR,ALR,PLR,RTR,UCR,UFR,PRT
invariants:
N1,N2,N3,Ar,P2,P3,K,CV,Q1,RA,DE,Q2,S,MN,Q3,V,MX,I50,SI,MI,TI
parameters(t_cont,s_cont,A%,HydGroup,n,bins,K,SubG):
4.0,8.0,5.0,9.4,3.0,5,5,3
options(decimals,harmonicMeanType,geometricMeanType,windexID,datasetType,outputOrder):
2,0,0,-1,fasta,true
```

Section S5. List (FASTA format) of all the peptides in the training dataset.

Data available on the online version of the Supporting Information:

https://doi.org/10.3390/antibiotics11121708

Section S6. List (FASTA format) of all the peptides in the development dataset.

Data available on the online version of the Supporting Information:

https://doi.org/10.3390/antibiotics11121708

Section S7. List (FASTA format) of all the peptides in the validation dataset.

Data available on the online version of the Supporting Information:

https://doi.org/10.3390/antibiotics11121708

Section S8. List (FASTA format) of all the peptides in the test dataset.

Data available on the online version of the Supporting Information:

https://doi.org/10.3390/antibiotics11121708

Table S2. Minimum (Min) and maximum (Max) boundaries of the applicability domains defined for the models derived from the training and the production datasets. The first model distinguishes antibacterial peptides from non-antibacterial peptides (ABP) and the second model categorizes antibacterial peptides as anti-Gram+, anti-Gram- or anti- both types of bacteria (Gram).

	Training (ABP)		*Training (Gram)*		*Production (ABP)*		*Production (Gram)*	
Descriptor	Min	Max	Min	Max	Min	Max	Min	Max
ISA_NO_PCR_N2	52,98	581,41	52,98	511,043	52,98	581,41	52,98	511,043
Z3_NO_PLR_I50	-3,44	7,57	-3,44	7,57	-3,44	7,57	-3,44	7,57
Gs(U)_NO_PLR_I50	-1015,71	1338,19	-1015,709	1338,189	-1015,71	1338,19	-1015,709	1338,189
IP_NO_PRT_I50	0	7,99	0	6,52	0	7,99	0	6,52
IP_NO_PLR_Q2	2,77	10,76	2,77	10,76	2,77	10,76	2,77	10,76
Gs(U)_NO_PRT_I50	0	1793,24	0	1793,239	0	1793,24	0	1793,239
IP_NO_AHR_I50	0	9,74	0	9,74	0	9,74	0	9,74
Z3_NO_PRT_N1	-116,18	50,54	-113,52	46,65	-116,18	50,54	-113,52	46,65
IP_NO_PRT_Q1	2,77	10,76	2,77	10,76	2,77	10,76	2,77	10,76
Z2_NO_PLR_I50	-2,09	4,61	-2,09	4,61	-2,09	4,61	-2,09	4,61
Z3_NO_PRT_CV	-6,38346E+16	9,45044E+16	-3,61425E+16	1,96182E+16	-6,38346E+16	9,45044E+16	-3,61425E+16	1,96182E+16
Z3_NO_PLR_Q2	-3,44	4,13	-3,44	4,13	-3,44	4,13	-3,44	4,13
Z2_NO_AHR_I50	-1,73	3,47	-1,73	3,47	-1,73	3,47	-1,73	3,47
IP_NO_PRT_RA	0	7,99	0	7,99	0	7,99	0	7,99
ISA_NO_PLR_Q3	17,87	132,16	17,87	132,16	17,87	132,16	17,87	132,16
ECI_NO_AHR_I50	0	1,36	0	1,36	0	1,36	0	1,36
Z3_NO_AHR_I50	-3,14	7,27	-3,14	7,27	-3,14	7,27	-3,14	7,27
Gw(U)_NO_PCR_SI5	0	1	0	1	0	1	0	1
IP_NO_AHR_RA	0	6,52	0	6,52	0	6,52	0	6,52
Z1_NO_PCR_V	0	0,11	0	0,11	0	0,11	0	0,11
Z3_NO_ALR_N3	-3,72	6,5	-3,721	6,004	-3,72	6,5	-3,721	6,004
Z1_NO_NPR_N1	-100,56	55,75	-90,82	55,75	-100,56	55,75	-90,82	55,75

Gw(U)_NO_PCR_MX	-35,07	-1,72	-34,315	-1,718	-35,072	-1,718	-34,315	-1,718
Z3_NO_AHR_RA	0	7,27	0	7,27	0	7,27	0	7,27
ISA_NO_PRT_Q1	17,87	189,42	17,87	189,42	17,87	189,42	17,87	189,42
IP_NO_PLR_N3	2,77	33,38	2,77	31,037	2,77	33,38	2,77	31,037
Mw_NO_AHR_I50	0	137	0	137	0	137	0	137
ECI_NO_PLR_MN	0,15	1,69	0,15	1,69	0,15	1,69	0,15	1,69
ECI_NO_PLR_TI5	0	75,86	0	74,98	0	79,338	0	79,338
IP_NO_AHR_MX	3,22	9,74	3,22	9,74	3,22	9,74	3,22	9,74
Z3_NO_PLR_SI5	0	1	0	1	0	1	0	1
Gw(U)_NO_AHR_I50	-28,91	30,86	-28,359	28,912	-28,91	30,86	-28,359	28,912
ECI_NO_PCR_MN	0,53	1,69	0,53	1,69	0,53	1,69	0,53	1,69
IP_NO_AHR_DE	0	4,61	0	4,61	0	4,61	0	4,61
Gs(U)_NO_AHR_P2	185,74	818,65	185,74	818,649	185,74	818,65	185,74	818,649
Gs(U)_NO_BSR_N1	-4052,65	14716,69	-2472,433	14716,693	-4052,65	14716,69	-2472,433	14716,693
Mw_NO_PLR_Q2	87	163	87	163	87	163	87	163
Z2_NO_RTR_P3	-5,36	1,45	-5,36	1,45	-5,36	1,45	-5,36	1,45
Z2_NO_NPR_SI5	0	1	0	1	0	1	0	1
Mw_NO_RTR_P3	57	115	57	115	57	115	57	115
ECI_NO_PLR_Q3	0,15	1,69	0,53	1,69	0,15	1,69	0,53	1,69
Z2_NO_PLR_MI5	0	2,58	0	2,55	0	2,58	0	2,561
ISA_NO_PLR_N2	17,87	581,41	17,87	513,006	17,87	581,41	17,87	513,006
Z2_NO_PRT_RA	0	9,01	0	9,01	0	9,01	0	9,01
Z3_NO_NPR_I50	-1,29	3,52	-1,29	3,52	-1,29	3,52	-1,29	3,52
ECI_NO_PLR_CV	0	1,44	0	1,263	0	1,44	0	1,263
Z2_NO_RTR_I50	-5,36	6,81	-5,36	6,81	-5,36	6,81	-5,36	6,81
Mw_NO_PRT_I50	0	115	0	99	0	115	0	99
HP_NO_PLR_Ar	-4,5	2,5	-4,5	1,3	-4,5	2,5	-4,5	1,3
ECI_NO_RTR_I50	0	1,31	0	1,31	0	1,31	0	1,31
Gw(U)_NO_PCR_P3	-37,77	-1,72	-37,77	-1,718	-37,77	-1,718	-37,77	-1,718
Z1_NO_PRT_MX	-4,92	3,64	-4,92	3,64	-4,92	3,64	-4,92	3,64

Z2_NO_AHR_P2	0,27	1,74	0,27	1,74	0,27	1,74	0,27	1,74
Mw_NO_PRT_RA	0	129	0	129	0	129	0	129
ISA_NO_AHR_DE	0	95,33	0	95,332	0	95,332	0	95,332
Mw_NO_PRT_Q2	57	186	57	186	57	186	57	186
Z3_NO_PLR_P2	0,01	4,13	0,01	3,869	0,01	4,13	0,01	3,879
Gs(U)_NO_PCR_Q1	-1015,71	-583,48	-1015,709	-583,484	-1015,71	-583,48	-1015,709	-583,484
ISA_NO_PLR_CV	0	1,24	0	1,208	0	1,24	0	1,208
Z3_NO_PLR_N2	0,01	19,46	0,01	16,852	0,01	19,46	0,01	16,852
Z3_NO_RTR_I50	0	2,36	0	2,36	0	2,36	0	2,36
Z1_NO_PRT_SI5	0	1	0	1	0	1	0	1
Z1_NO_NPR_I50	-4,92	7,15	-4,92	7,15	-4,92	7,15	-4,92	7,15
Z1_NO_PRT_P2	0,9	4,92	0,901	4,92	0,9	4,92	0,901	4,92
Z1_NO_PLR_N1	-4,17	108,88	-1,39	97,02	-4,17	108,88	-1,39	102,74
ECI_NO_ALR_P3	0,01	0,34	0,01	0,34	0,01	0,34	0,01	0,34
ISA_NO_NPR_SI5	0	1	0	1	0	1	0	1
Mw_NO_RTR_RA	0	58	0	58	0	58	0	58
Z2_NO_PCR_Q3	1,41	2,52	1,41	2,52	1,41	2,52	1,41	2,52
ISA_NO_PLR_V	0	6531,1	0	5198,94	0	6531,1	0	5198,94
ISA_NO_NPR_I50	0	189,42	0	189,42	0	189,42	0	189,42
Z1_NO_AHR_Q2	-4,19	3,08	-4,19	3,08	-4,19	3,08	-4,19	3,08
Z1_NO_PRT_Q1	-4,92	3,64	-4,92	3,64	-4,92	3,64	-4,92	3,64
Mw_NO_ALR_SI5	0	1	0	1	0	1	0	1
IP_AC1_PRT_Q2	15,35	231,56	15,346	231,555	15,35	231,56	15,346	231,555
IP_AC1_AHR_Q1	8,92	209,6	10,368	209,605	8,919	209,6	10,368	209,605
Z1_AC1_PCR_Q3	-28,34	20,97	-27,946	19,757	-28,34	20,97	-27,946	19,757
Z1_AC1_PCR_MN	-28,34	20,97	-28,339	19,354	-28,34	20,97	-28,339	19,354
ECI_AC1_PCR_Q3	0,01	5,71	0,005	5,712	0,005	5,712	0,005	5,712
HP_AC1_PCR_MN	-40,5	36	-40,5	31,5	-40,5	36	-40,5	31,5
Z2_AC1_PCR_MI5	0	2,58	0	2,585	0	2,585	0	2,585
Z1_AC1_PCR_CV	-6194,93	4647,27	-2936,387	2488,821	-32640,381	4647,27	-3155,779	2674,221
IP_AC1_RTR_Q3	14,04	139,45	14,044	139,45	14,04	139,45	14,044	139,45

Mw_AC1_PCR_Q2	7296	58032	7296	58032	7296	58032	7296	58032
Mw_AC1_NPR_MX	4959	69192	7171	69192	4959	69192	7171	69192
IP_AC1_RTR_MX	14,04	139,45	14,044	139,45	14,04	139,45	14,044	139,45
IP_AC1_RTR_Q2	14,04	139,45	14,044	139,45	14,04	139,45	14,044	139,45
Mw_AC1_PCR_SI5	0	1	0	1	0	1	0	1
Mw_AC1_PCR_Q1	7296	58032	7296	58032	7296	58032	7296	58032
IP_AC1_ALR_Q2	16,54	139,45	16,537	139,45	16,54	139,45	16,537	139,45
ISA_AC1_PCR_SI5	0	1	0	1	0	1	0	1
Z2_AC1_PCR_MN	-27,01	18,4	-27,014	18,396	-27,014	18,4	-27,014	18,396
Z2_AC1_PCR_Q2	-27,01	18,4	-27,014	18,396	-27,01	18,4	-27,014	18,396
Gs(U)_AC1_BSR_MN	-1579487,69	1057312,79	-1579487,692	893700,369	-1579487,692	1133208,399	-1579487,692	893700,369
ISA_AC1_ALR_MX	356,15	58473,95	367,908	58473,954	356,15	58473,954	367,908	58473,954
ISA_AC1_PCR_P3	946,75	38937,18	1034,699	38937,175	946,75	38937,18	1034,699	38937,175
Gs(U)_AC1_PLR_Q2	-1579487,69	2063330,66	-1579487,692	2063330,664	-1579487,69	2063330,664	-1579487,692	2063330,664
ISA_AC1_PRT_M	470,47	55813,23	511,157	55813,234	470,47	55813,23	511,157	55813,234
IP_AC1_ARM_Q1	15,18	163,34	15,18	163,337	15,18	163,34	15,18	163,337
Z3_AC1_RTR_MN	-16,24	11,69	-16,237	9,747	-16,24	11,69	-16,237	9,747
IP_AC1_BSR_N3	15,18	296,86	15,678	296,859	15,18	300,115	15,678	296,859
Gs(U)_AC1_PCR_Q1	-1579487,69	2063330,66	-1579487,692	2063330,664	-1579487,692	2063330,664	-1579487,692	2063330,664
Z1_AC1_ALR_MN	-32,32	43,69	-32,323	43,69	-32,32	43,69	-32,323	43,69
Gs(U)_AC1_PCR_CV	-4468,31	33259,35	-4468,312	8490,981	-4468,31	33259,35	-4468,312	8490,981
Z3_AC1_PCR_Ar	-28,41	22,19	-28,414	21,977	-28,41	22,19	-28,414	21,977
Z2_AC1_PCR_CV	-4,1852E+16	4,74163E+15	-4,1852E+16	997,75	-4,1852E+16	4,74163E+15	-4,1852E+16	997,75
Z2_AC1_UFR_Q2	-39,13	57,46	-39,128	57,459	-39,13	57,46	-39,128	57,459
Z2_AC1_PCR_MX	-27,01	18,4	-19,883	18,396	-27,01	18,4	-19,883	18,396

IP_AC1_NPR_MN	15,18	139,45	15,18	129,55	15,18	139,45	15,18	129,55
HP_AC1_AHR_MN	-35,1	35,1	-35,1	35,1	-35,1	35,1	-35,1	35,1
IP_AC1_BSR_MN	15,18	129,55	15,18	129,55	15,18	129,55	15,18	129,55
Z2_AC1_PRT_MX	-13,51	57,46	-13,507	57,459	-13,51	57,46	-13,507	57,459
Gw(U)_AC1_PCR_S	-3,68	3,2	-2,584	3,201	-3,68	3,2	-2,584	3,201
Z3_AC1_UCR_N3	-41,54	54,4	-41,541	54,398	-41,54	54,4	-41,541	54,398
HP_AC1_UFR_Q3	-14,4	14,4	-13,92	14,4	-14,4	14,4	-14,4	14,4
IP_AC1_UFR_MN	16,54	139,45	16,537	139,45	16,537	139,45	16,537	139,45
Z3_AC1_PCR_N2	0,01	109,83	0,022	86,609	0,01	109,83	0,022	86,609
Z3_AC1_UFR_N3	-35,87	22,93	-25,566	22,927	-35,87	22,93	-29,715	22,927
Mw_AC1_AHR_Q2	4047	50964	4047	50964	4047	50964	4047	50964
ISA_AC1_BSR_Q2	1062,19	71759,87	2231,999	71759,873	1062,19	71759,87	2231,999	71759,873
ISA_AC1_UFR_Q3	356,15	46351,07	356,149	46351,074	356,149	46351,074	356,149	46351,074
Mw_AC1_ALR_N3	4959	82987,4	5871	76407,394	4959	82987,4	5871	76407,394
ECI_AC1_PRT_MX	0	5,71	0,002	5,712	0	5,712	0,002	5,712
ISA_AC1_UCR_MX	329,88	50067,49	393,618	50067,494	329,88	50067,49	393,618	50067,494
Z3_AC1_RTR_CV	-1,43686E+16	77823,08	-479,559	1577,997	-1,43686E+16	81603,356	-479,559	1577,997
Gs(U)_AC1_NCR_Q3	-882104,01	1229493,56	-882104,009	1229493,558	-882104,01	1229493,56	-882104,009	1229493,558
ISA_AC1_ARM_Q1	1561,48	65116,91	1561,481	65116,913	1561,48	65116,91	1561,481	65116,913
Gw(U)_AC1_PCR_G	0	2621,92	0	2621,918	0	2621,92	0	2621,918
Z2_AC1_UCR_Q1	-24,87	22,4	-24,87	17,474	-24,87	22,4	-24,87	17,474
Mw_AC1_ARM_Q3	7809	69192	7809	69192	7809	69192	7809	69192
Z2_AC1_NPR_MN	-39,13	28,73	-39,128	22,52	-39,13	28,73	-39,128	22,52
Z2_AC1_PCR_N3	-46,08	33,66	-43,325	32,454	-46,08	33,66	-43,325	32,454
Z3_AC1_ALR_MN	-15,34	11,69	-15,342	9,232	-15,342	11,69	-15,342	9,232
IP_AC1_ARM_Q3	15,18	163,34	15,18	163,337	15,18	163,34	15,18	163,337
Gw(U)_AC1_PCR_MN	0	2226,07	0	1884,606	0	2226,07	0	1884,606

Z1_AC1_BSR_Q3	-33,06	48,41	-28,31	48,413	-33,06	48,41	-28,31	48,413
Mw_AC1_ALR_Q3	4959	48732	5871	44802	4959	48732	5871	44802
HP_AC1_ALR_Q2	-40,5	40,5	-40,5	40,5	-40,5	40,5	-40,5	40,5
Gs(U)_AC1_PCR_I50	-1579487,69	2999111,86	-1480343,057	2999111,859	-1579487,692	2999111,86	-1480343,057	2999111,859
Z3_AC1_UCR_Q1	-28,41	26,27	-28,414	17,057	-28,414	26,27	-28,414	17,057
ECI_AC1_AHR_Q3	0	4,6	0	4,597	0	4,6	0	4,597
Mw_AC1_UCR_Q3	4959	60636	4959	60636	4959	60636	4959	60636
Z1_AC1_AHR_Q3	-30,5	41,23	-29,26	40,517	-30,5	41,23	-29,26	40,517
IP_AC1_ALR_RA	0	122,91	0	122,913	0	122,913	0	122,913
Gw(U)_AC1_AHR_MX	0	2326,26	0	2326,259	0	2326,26	0	2326,259
ISA_AC1_NPR_Q3	393,62	71759,87	787,235	71759,873	393,62	71759,87	787,235	71759,873
Z2_AC1_PLR_Q3	-27,01	22,4	-27,014	18,396	-27,01	22,4	-27,014	18,396
Z1_AC1_PCR_N2	0,17	114,76	0,199	78,499	0,17	114,76	0,199	78,499
Mw_AC1_NCR_MN	6555	47988	6555	47988	6555	47988	6555	47988
W(U)_AC1_UCR_MN	0	1104	0	880	0	1104	0	880
ISA_AC1_UCR_Q3	329,88	50067,49	393,618	47355,571	329,88	50067,49	393,618	47355,571
Z1_AC1_UCR_N3	-40,97	39,48	-40,971	32,283	-40,97	39,48	-40,971	32,676
Mw_AC1_PLR_Q2	4959	58032	5757	58032	4959	58032	5757	58032
Z3_AC1_BSR_Q1	-11,56	9,63	-11,564	9,212	-11,564	9,63	-11,564	9,212
Z1_AC1_BSR_MX	-33,06	48,41	-28,31	48,413	-33,06	48,413	-28,31	48,413
Z2_AC1_ARM_Q1	-39,13	22,52	-39,128	22,52	-39,13	22,52	-39,128	22,52
IP_AC2_PRT_Q1	7,67	231,56	7,673	128,69	7,67	231,56	7,673	128,69
Z1_AC2_PCR_MN	-28,34	20,97	-28,339	16,813	-28,34	20,97	-28,339	19,482
IP_AC2_AHR_MX	15,65	209,6	15,651	209,605	15,65	209,605	15,651	209,605
Gs(U)_AC2_AHR_MN	-1381198,42	1663019,09	-1381198,421	1663019,093	-1381198,421	1663019,09	-1381198,421	1663019,093

Z1_AC2_PCR_N2	0,17	98,8	0,169	78,554	0,17	98,8	0,169	78,554
Gs(U)_AC2_PLR_Q1	-1579487,69	2063330,66	-1579487,692	1340372,899	-1579487,692	2063330,66	-1579487,692	1340372,899
Z3_AC2_PCR_Q3	-28,41	23,67	-28,414	23,667	-28,414	23,67	-28,414	23,667
HP_AC2_PLR_MN	-40,5	21,35	-40,5	20,25	-40,5	21,35	-40,5	20,25
IP_AC2_NPR_P3	15,18	136,22	16,537	128,69	15,18	136,22	16,537	128,69
HP_AC2_PLR_MX	-40,5	40,5	-40,5	40,5	-40,5	40,5	-40,5	40,5
IP_AC2_NPR_N3	15,18	348,41	16,537	348,408	15,18	348,41	16,537	348,408
IP_AC2_PCR_Q1	21,02	231,56	21,024	189,735	21,02	231,56	21,024	189,735
IP_AC2_ALR_P3	15,9	139,45	16,537	139,45	15,9	139,45	16,537	139,45
Gs(U)_AC2_PCR_Q1	-1579487,69	2063330,66	-1579487,692	1646412,111	-1579487,692	2063330,66	-1579487,692	1646412,111
IP_AC2_RTR_MN	7,67	139,45	7,673	139,45	7,67	139,45	7,673	139,45
Z1_AC2_PCR_Q3	-28,34	20,97	-28,339	20,966	-28,34	20,97	-28,339	20,966
IP_AC2_RTR_Q2	7,67	139,45	7,673	139,45	7,67	139,45	7,673	139,45
Z3_AC2_PCR_MI5	0	2,58	0	2,585	0	2,585	0	2,585
IP_AC2_ALR_Q2	15,9	139,45	16,537	139,45	15,9	139,45	16,537	139,45
Z1_AC2_PCR_P3	-28,34	20,97	-28,339	17,014	-28,34	20,97	-28,339	19,482
ECI_AC2_PCR_SI5	0	1	0	1	0	1	0	1
Gs(U)_AC2_AHR_N3	-3233254,1	4144046,87	-2790877,168	3312445,968	-3233254,1	4144046,87	-2790877,168	3312445,968
Z3_AC2_PCR_S	-3,09	3,68	-2,475	2,041	-3,09	3,68	-2,475	2,041
Gs(U)_AC2_PCR_CV	-124046,25	5815,45	-2605,349	3167,065	-124046,25	5815,45	-2605,349	3167,065
IP_AC2_UCR_Q2	8,92	126,32	15,734	126,322	8,92	126,322	15,734	126,322
Z3_AC2_UCR_MX	-25,94	34,11	-25,936	34,114	-25,94	34,114	-25,936	34,114
Gs(U)_AC2_PRT_P2	2826,9	1859861,13	4434,144	1828947,155	2826,9	1859861,13	4434,144	1828947,155

Gs(U)_AC2_PLR_Q3	-1579487,69	2063330,66	-1579487,692	2063330,664	-1579487,69	2063330,664	-1579487,692	2063330,664
IP_AC2_ARM_Q2	15,18	163,34	15,18	163,337	15,18	163,34	15,18	163,337
Z2_AC2_PCR_Q1	-27,01	18,4	-27,014	18,396	-27,014	18,4	-27,014	18,396
HP_AC2_PCR_Q3	-40,5	40,5	-40,5	40,5	-40,5	40,5	-40,5	40,5
IP_AC2_BSR_P3	15,18	129,55	16,537	126,753	15,18	129,55	16,537	126,753
IP_AC2_AHR_MN	8,92	209,6	8,919	209,605	8,919	209,6	8,919	209,605
Z2_AC2_ALR_MN	-39,13	37,47	-39,128	28,73	-39,13	37,47	-39,128	28,73
IP_AC2_PCR_V	0	7459,83	0	6571,732	0	7459,83	0	6571,732
Z1_AC2_PRT_N2	0,29	183,29	0,286	126,991	0,29	183,29	0,286	126,991
Gs(U)_AC2_UFR_MX	-304785,56	398150,24	-304785,562	398150,235	-304785,562	398150,24	-304785,562	398150,235
Z1_AC2_PRT_MN	-35,82	24,21	-35,818	24,206	-35,82	24,21	-35,818	24,206
HP_AC2_UCR_Q3	-31,5	31,5	-31,5	31,5	-31,5	31,5	-31,5	31,5
IP_AC2_BSR_Q1	15,18	129,55	15,18	128,474	15,18	129,55	15,18	128,474
HP_AC2_AHR_Q1	-35,1	35,1	-35,1	35,1	-35,1	35,1	-35,1	35,1
ISA_AC2_ARM_Q1	1561,48	71759,87	1561,481	58931,099	1561,48	71759,87	1561,481	59055,473
Mw_AC2_PLR_Q2	4959	58032	4959	58032	4959	58032	4959	58032
Gs(U)_AC2_NCR_Q3	-941182,13	1229493,56	-941182,126	1229493,558	-941182,13	1229493,56	-941182,126	1229493,558
ISA_AC2_RTR_Q1	319,34	46351,07	319,337	41499,896	319,337	46351,07	319,337	41499,896
Z1_AC2_AHR_Q2	-29,78	41,23	-29,26	39,805	-29,78	41,23	-29,26	39,805
HP_AC2_UCR_Q2	-31,5	31,5	-31,5	31,5	-31,5	31,5	-31,5	31,5
ISA_AC2_RTR_Q3	319,34	46351,07	349,001	46351,074	319,34	46351,074	349,001	46351,074
Mw_AC2_RTR_MN	3249	42780	3249	42780	3249	42780	3249	42780
IP_AC2_UFR_MN	16,54	139,45	16,537	139,45	16,537	139,45	16,537	139,45
ISA_AC2_UCR_MN	319,34	50067,49	319,337	50067,494	319,337	50067,49	319,337	50067,494
Z3_AC2_UFR_Q3	-15,34	18,42	-15,342	18,42	-15,34	18,42	-15,342	18,42
IP_AC2_RTR_MX	7,67	139,45	15,346	139,45	7,67	139,45	15,346	139,45
IP_AC2_ALR_RA	0	123,55	0	116,303	0	123,55	0	116,303

IP_AC2_BSR_MN	15,18	129,55	15,18	126,753	15,18	129,55	15,18	126,753
Z2_AC2_PCR_N2	0,13	68,95	0,212	68,951	0,127	68,95	0,212	68,951
Z3_AC2_UFR_Q1	-15,34	18,42	-15,342	18,42	-15,342	18,42	-15,342	18,42
Z3_AC2_PCR_Ar	-28,41	21,52	-28,414	20,286	-28,414	21,52	-28,414	20,286
Z3_AC2_RTR_N3	-31,15	25,63	-31,148	24,654	-31,15	25,63	-31,148	24,654
Z1_AC2_ARM_Q2	-35,82	48,41	-35,818	48,413	-35,82	48,41	-35,818	48,413
ECI_AC2_ARM_MX	0	3,65	0,001	3,65	0	3,65	0,001	3,65
HP_AC2_NCR_P2	0	31,5	0	31,5	0	31,5	0	31,5
Z3_AC2_PLR_MN	-28,41	15,2	-28,414	15,205	-28,414	15,2	-28,414	15,205
ECI_AC2_PLR_Q2	0	5,71	0,005	5,712	0	5,712	0,005	5,712
HP_AC2_ALR_Q1	-40,5	28,88	-40,5	25,2	-40,5	28,88	-40,5	25,2
Z1_AC2_UFR_MN	-21,94	16,23	-21,943	14,54	-21,943	16,23	-21,943	14,54
Z1_AC2_AHR_Q3	-29,78	41,23	-29,26	41,23	-29,78	41,23	-29,26	41,23
HP_AC2_ALR_N3	-97,43	81,23	-97,433	65,602	-97,43	81,23	-97,433	65,602
HP_AC2_RTR_Q1	-31,5	31,5	-31,5	31,5	-31,5	31,5	-31,5	31,5
Gs(U)_AC2_UCR_Q2	-1001308,09	1308037,85	-1001308,086	1308037,846	-1001308,09	1308037,85	-1001308,086	1308037,846
ECI_AC2_PLR_Q3	0	5,71	0,005	5,712	0	5,712	0,005	5,712
Gs(U)_AC2_NCR_I50	-941182,13	1787105,08	-941182,126	1767297,892	-941182,13	1787105,08	-941182,126	1767297,892
Z1_AC2_AHR_N2	0,01	183,29	0,013	99,909	0,01	183,29	0,013	99,909
Mw_AC2_AHR_Q3	4047	50964	4047	50964	4047	50964	4047	50964
Z2_AC2_AHR_N2	0,01	53,23	0,01	50,224	0,01	53,23	0,01	50,224
Mw_AC2_UCR_MX	4959	60636	4959	60636	4959	60636	4959	60636
Z1_AC2_UCR_Ar	-30,72	20,99	-30,722	18,547	-30,722	20,99	-30,722	18,547
Z1_AC2_NPR_N3	-67,75	111,6	-62,551	84,284	-67,75	111,6	-62,551	84,284
Z1_AC2_RTR_Q2	-35,2	26,5	-34,58	23,587	-35,2	26,5	-34,58	23,587
ECI_AC2_RTR_RA	0	4,43	0	4,427	0	4,43	0	4,427
ECI_AC2_BSR_MX	0	3,65	0,001	3,65	0	3,65	0,001	3,65

Gs(U)_AC2_BSR_Q1	-1579487,69	1057312,79	-1579487,692	1057312,793	-1579487,692	1057312,79	-1579487,692	1057312,793
Gs(U)_AC2_ARM_Q3	-1138695,5	1185300,96	-1138695,501	1185300,962	-1138695,501	1185300,962	-1138695,501	1185300,962
ISA_AC2_PRT_CV	0,19	1,87	0,196	1,829	0,19	1,87	0,196	1,829
Z3_AC2_ALR_N3	-31,31	26,03	-31,312	25,089	-31,31	26,03	-31,312	25,089
Gw(U)_AC2_PCR_CV	0	3,46	0	3,464	0	3,46	0	3,464
Z2_AC2_RTR_MN	-39,13	33,93	-39,128	28,73	-39,13	34,25	-39,128	34,25
Z3_ES_PCR_TI5	0	69,5	0	44,29	0	69,5	0	44,29
Z3_ES_PCR_S	-3,06	3,86	-2,406	3,37	-3,06	3,86	-2,406	3,37
Gs(U)_ES_PRT_P2	65,44	2009,9	195,996	1890,971	65,44	2009,9	195,996	1890,971
Gs(U)_ES_PCR_CV	-0,7	0	-0,617	0	-0,7	0	-0,617	0
ISA_ES_PCR_CV	-10838,16	354,9	-10838,162	354,895	-10838,16	354,9	-10838,162	354,895
HP_ES_PCR_S	-2,68	3,32	-2,406	2,238	-2,68	3,32	-2,406	2,238
IP_ES_PCR_SI5	0	1	0	1	0	1	0	1
ISA_ES_PCR_S	-3,61	3,32	-3,607	2,556	-3,61	3,32	-3,607	2,556
HP_ES_PCR_SI5	0	1	0	1	0	1	0	1
HP_ES_PLR_Q1	-12,14	4,06	-12,01	2,271	-12,14	4,06	-12,01	2,271
Z2_ES_PCR_SI5	0	1	0	1	0	1	0	1
IP_ES_PCR_P3	8,01	17,79	8,401	17,204	8,01	17,79	8,401	17,204
IP_ES_BSR_Ar	1,46	7,67	1,55	7,349	1,46	7,67	1,55	7,349
IP_ES_BSR_MN	0,55	7,51	0,547	7,349	0,55	7,51	0,547	7,349
Z3_ES_AHR_MN	-10,04	10,19	-8,839	10,193	-10,04	10,19	-9,157	10,193
Z1_ES_PCR_CV	0	0,62	0	0,577	0	0,635	0	0,577
ISA_ES_RTR_Q2	-114,28	204,1	-114,276	204,102	-114,28	204,247	-114,276	204,247
Z3_ES_UCR_MX	-3,89	11,54	-3,235	11,54	-3,89	11,577	-3,235	11,577
IP_ES_UCR_Q3	-4,32	7,71	-2,376	7,176	-4,32	7,71	-2,561	7,176
Gs(U)_ES_PLR_Q3	-2314,66	1098,63	-2253,505	1098,627	-2474,666	1098,63	-2474,666	1098,627

Z3_ES_RTR_Q2	-1,3	7,69	-0,819	7,519	-1,322	7,69	-0,819	7,519
Z3_ES_ARM_Ar	-2,96	5,61	-2,228	4,908	-2,96	5,61	-2,228	4,908
ECI_ES_AHR_I50	-1,42	3,29	-1,25	3,148	-1,42	3,29	-1,25	3,148
Z1_ES_NPR_N3	-30,9	7,49	-24,947	7,49	-30,9	7,49	-24,959	7,49
ECI_ES_RTR_MX	-1,34	2,78	-1,34	2,785	-1,34	2,78	-1,34	2,785
IP_ES_UCR_P3	-4,51	7,15	-3,085	6,776	-4,51	7,15	-3,085	6,776
Z3_ES_RTR_Q1	-2,42	7,69	-2,422	7,509	-2,42	7,69	-2,422	7,509
ISA_ES_PCR_MN	-60,58	181,29	-60,582	171,078	-60,58	181,29	-60,582	172,277
IP_ES_ARM_Q2	1,07	11,1	1,071	10,402	1,07	11,1	1,071	10,402
Gs(U)_ES_PRT_Q3	-1052,65	2023,94	-1046,549	1982,117	-1052,65	2023,94	-1046,549	1982,117
Z3_ES_ALR_Q3	-4,91	7,69	-4,908	7,39	-4,91	7,69	-4,908	7,39
ISA_ES_RTR_MN	-132,12	195,41	-132,116	195,413	-132,12	195,41	-132,116	195,413
Z1_ES_PCR_Q1	2,04	9,89	2,125	9,89	2,04	9,89	2,125	9,89
Z3_ES_ARM_N1	-6,91	61,37	-4,621	32,974	-6,91	61,37	-4,621	32,974
ISA_ES_PCR_Q1	-59,37	181,29	-59,373	171,078	-59,37	181,29	-59,373	172,277
Z2_ES_PLR_Ar	-4,01	8,28	-3,257	8,21	-4,01	8,28	-3,257	8,21
Z1_ES_AHR_P2	0,01	10,82	0,111	9,966	0,01	10,82	0,111	10,016
Gs(U)_ES_ALR_MX	-661,06	2504,44	-516,742	2446,231	-661,06	2504,44	-516,742	2446,231

Table S3. Values resulting from the radial diffusion assay applied to the peptide Urine-3462 against the *Pseudomonas aeruginosa* strain ATCC 27853.

EXPERIMENTS	1000	500	250	125	62.5	31.25	15.6	7.8	LL 37
1	1.2	1.1	0.9	0.8	0.6	0.4	0	0	0.7
2	1.3	1.1	0.9	0.7	0.6	0.4	0	0	0.7
3	1.1	1	0.9	0.8	0.5	0.3	0	0	0.6
4	1.3	1	0.9	0.8	0.6	0.3	0	0	0.6
5	1.2	1.1	1	0.8	0.6	0.4	0.3	0	0.7
6	1.2	1.1	0.9	0.8	0.6	0.4	0	0	0.7

6.4 Publication IV

Author contributions

ProtDCal-Suite: A web server for the numerical codification and functional analysis of proteins

Sandra Romero-Molina, Yasser B. Ruiz-Blanco, James R. Green, Elsa Sanchez-Garcia

Research Article in:	**Protein Science**
Impact factor:	6.993 (2021)
Published online:	July 4th, 2019
DOI:	10.1002/pro.3673
Citations	16 (Dec 6th, 2022)

Complete citation from source:

Romero-Molina, S.; Ruiz-Blanco, Y.B.; Green, J.R.; Sanchez-Garcia, E. ProtDCal-Suite: A web server for the numerical codification and functional analysis of proteins. *Protein Science* 2019; 28(9): 1734-1743. https://doi.org/10.1002/pro.3673

Contributions:

Conception:	20%
Statistical analysis:	0%
Code implementations:	100%
Web(tool) validation and deployment:	100%
Web(tool) maintenance:	100%
Manuscript writing:	50%
Manuscript revision:	10%

Received: 11 April 2019 | Revised: 21 June 2019 | Accepted: 24 June 2019

DOI: 10.1002/pro.3673

TOOLS FOR PROTEIN SCIENCE

THE PROTEIN SOCIETY WILEY

ProtDCal-Suite: A web server for the numerical codification and functional analysis of proteins

Sandra Romero-Molina[1] | **Yasser B. Ruiz-Blanco**[1] | **James R. Green**[2] | **Elsa Sanchez-Garcia**[1]

[1]Computational Biochemistry, Center of Medical Biotechnology, University of Duisburg-Essen, Essen, Germany

[2]Systems and Computer Engineering, Carleton University, Ottawa, Ontario, Canada

Correspondence
Yasser B. Ruiz-Blanco and Elsa Sanchez-Garcia, Computational Biochemistry, Center of Medical Biotechnology, University of Duisburg-Essen, Essen, Germany.
Email: yasser.ruizblanco@uni-due.de (Y.B.R.-B.) and elsa.sanchez-garcia@uni-due.de (E. S.-G.)

Funding information
Boehringer Ingelheim Stiftung, Grant/Award Number: Plus 3; Deutsche Forschungsgemeinschaft, Grant/Award Numbers: EXC 2033/390677874, SFB 1279/316249678

Abstract

Computational tools for the analysis of protein data and the prediction of biological properties are essential in life sciences and biomedical research. Here, we introduce *ProtDCal-Suite*, a web server comprising a set of machine learning-based methods for studying proteins. The main module of *ProtDCal-Suite* is the ProtDCal software. ProtDCal translates the structural information of proteins into numerical descriptors that serve as input to machine-learning techniques. The *ProtDCal-Suite* server also incorporates a post-processing optional stage that allows ranking and filtering the obtained descriptors by computing their Shannon entropy values across the input set of proteins. ProtDCal's codification was used in the development of models for the prediction of specific protein properties. Thus, the other modules of *ProtDCal-Suite* are protein analysis tools implemented using ProtDCal's descriptors. Among them are *PPI-Detect*, for predicting the interaction likelihood of protein–protein and protein–peptide pairs, *Enzyme Identifier*, for identifying enzymes from amino acid sequences or 3D structures, and *Pred-NGlyco*, for predicting N-glycosylation sites. *ProtDCal-Suite* is freely accessible at https://protdcal.zmb.uni-due.de.

KEYWORDS

descriptor, enzymes, machine-learning, N-glycosylation, protein–protein interactions, web server

1 | INTRODUCTION

The analysis of protein data and the prediction of protein properties are of fundamental importance in modern Molecular Biology. Subjects such as the elucidation of protein–protein interaction networks, protein function prediction, and computational drug design, all benefit from massive computational analysis of the known protein data to extrapolate new knowledge of biological function.[1–4] The numerical encoding of raw protein sequences or structural data plays an important role for the development of robust prediction tools based on machine-learning techniques.

In this context, ProtDCal is a software package that transforms protein sequences or 3D-structures into general-purpose numerical descriptors, accounting for both global and local information.[5] Due to its complementary performance with respect to other well-established tools in the field like PROFEAT[6] and PseAcc[7] (later extended to Pse-in-one[8]), ProtDCal has been used in a number of studies.[9–19] Notable among them are the modeling of posttranslational modifications,[14] the prediction of protein enzymatic function,[15] the prediction of antimicrobial activity in peptides,[16] the determination of residues critical for protein function,[17] and the prediction of stability changes upon mutations.[18] Very recently, ProtDCal was enhanced with a procedure for encoding protein pairs, which allows targeting the protein–protein interaction identification problem.[19]

Here, we present *ProtDCal-Suite*, a versatile platform for granting web access to the wealth of encoding approaches implemented within ProtDCal, as well as to several protein analysis tools developed using ProtDCal's descriptors. Currently, *ProtDCal-Suite* allows predicting the enzyme-like character of proteins (Enzyme Identifier)[15] and N-glycosylation (Pred-NGlyco) sites[5,14] as well as evaluating the likelihood of protein–protein interactions (PPI-Detect).[19] Recently, a tool for the prediction of methylation sites (MethylSight)[20] was also incorporated by us in *ProtDCal-Suite*. These applications of ProtDCal are useful on their own right, but also illustrate the capabilities of ProtDCal-derived features for novel and diverse protein analysis tasks.

2 | RESULTS

ProtDCal-Suite consists of a main module (ProtDCal) and a set of secondary modules that provide access to machine learning-based tools. These applications are used to predict specific protein functions and were created using ProtDCal descriptors. Next, we describe the generalities of the suite and the available tools.

2.1 | The *ProtDCal-Suite*

The graphical design of *ProtDCal-Suite* is highly intuitive (Figure 1). Each tool has its own interface but shares a similar layout for quick familiarization by users. We documented all individual tools with help content and usage examples. Extended documentation and a tutorial, explaining the protein-encoding features of ProtDCal, are also available. Template python scripts allow remotely accessing the web services and parsing the output data. This way, users can also submit jobs without using the web interface. This feature is valuable for remotely invoking the server services or for integrating the calculation of descriptors into custom third-party workflows.

2.1.1 | *ProtDCal-Suite* input

All the predictive tools implemented in *ProtDCal-Suite* accept input files containing the sequence information of proteins in FASTA format (Enzyme Identifier, PPI-Detect, MethylSight and Pred-NGlyco) and/or structural information in PDB format (Enzyme Identifier). In the main module (ProtDCal), the user can also specify options for the calculation of protein descriptors via the web interface. In the documentation of the interfaces for the different tools within *ProtDCal-Suite* we provide information about the input formats and offer examples for the submission of jobs. Besides the input data, the user enters a job name and (optionally) an email address to receive information about the progress of the job. Using the identification code (ID) assigned to the job, the user can follow its status in the computing queue and subsequently retrieve the results of the calculations.

2.1.2 | *ProtDCal-Suite* output

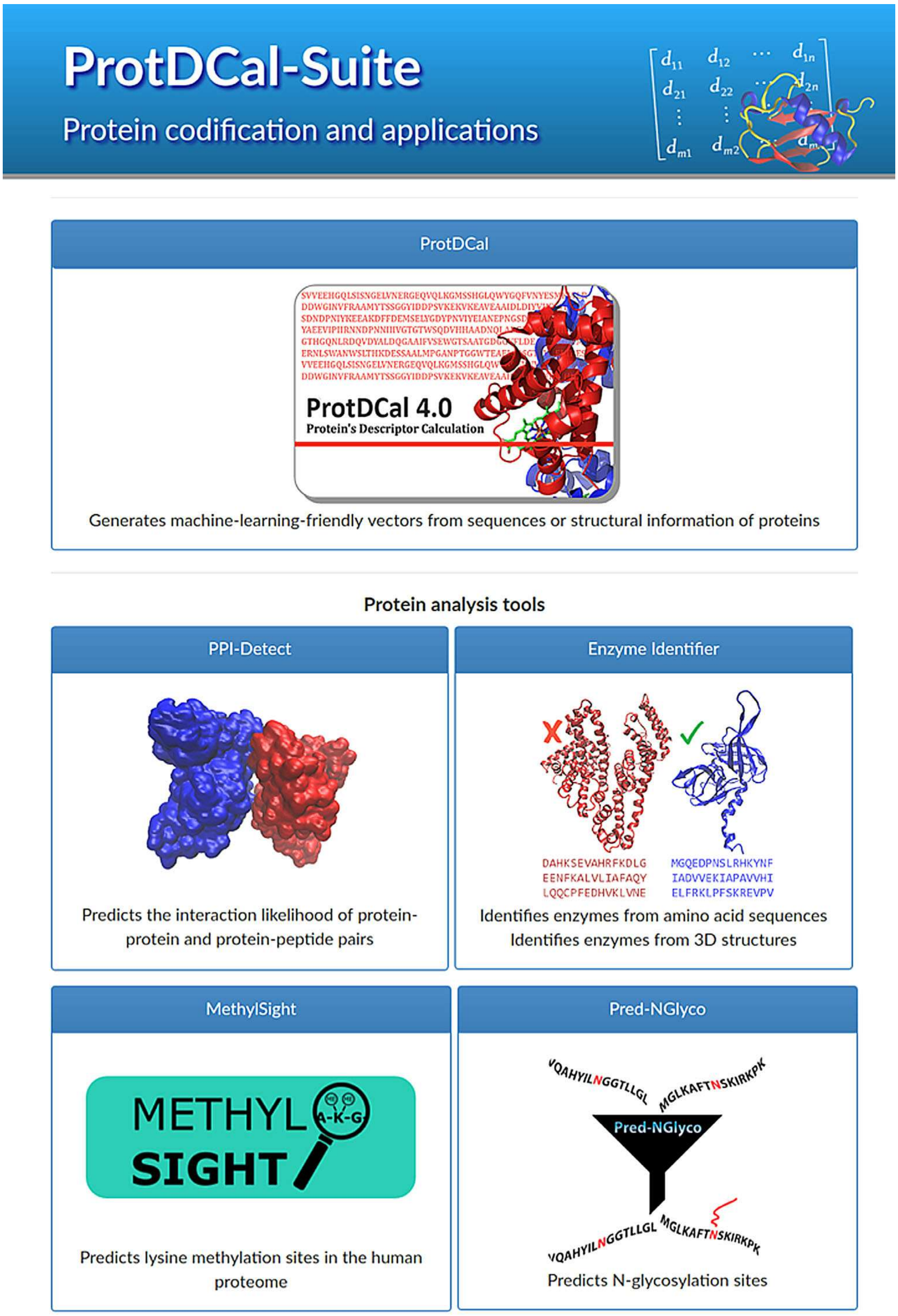
Once a job is completed, there are two main output interfaces depending on whether the used tool was (1) ProtDCal or (2) any of the ProtDCal-based applications. In the first case, the output is a download link to access the file containing the complete descriptor matrix. In addition, the output interface permits the user to post-process the computed descriptors using an unsupervised feature selection approach based on Shannon Entropy (see section *Analysis of ProtDCal´s outcome*). The use of Shannon Entropy allows for a preliminary reduction of the dimensionality of the descriptor matrix. For ProtDCal-based applications, the predictions are visualized directly in the web, using a tabular form. All the results can be downloaded in CSV format.

2.2 | ProtDCal

ProtDCal is a computational package[5] for encoding the sequences and structures of proteins into numerical descriptors. These descriptors are the input to machine-learning techniques (artificial neural networks,[21] support vector machine,[22] and random forest,[23] among others) used for the development of novel predictors of protein functions and properties. ProtDCal splits the protein into different residue groups. Then, the contributions of the residues in each group are aggregated using diverse descriptive statistics (such as averages, variance, minimum or maximum values). This aggregation gives rise to a large variety of scalar descriptors, each of which represents local or global properties of the protein. The resulting vector is applicable to data mining problems such as protein classification, similarity analysis, and function prediction.

2.2.1 | ProtDCal steps for calculating a protein descriptor

Figure 2 illustrates the process of obtaining the descriptor FD_AC2_GLY_Ar for the human prion protein fragment described by the PDB entry 1OEH[27] with sequence: HGGGTGQP. The notation used in ProtDCal to label the final descriptors directly refers to the options chosen by the user in the input step. A combinatorial algorithm composed of four steps (Figure 2, top), each with several options (that can be defined by the user) is implemented in ProtDCal.



Generates machine-learning-friendly vectors from sequences or structural information of proteins

Predicts the interaction likelihood of protein-protein and protein-peptide pairs

Identifies enzymes from amino acid sequences
Identifies enzymes from 3D structures

Predicts lysine methylation sites in the human proteome

Predicts N-glycosylation sites

FIGURE 1 Main interface of *ProtDCal-Suite*

The program computes all the combinations of defined options, thus producing one individual descriptor from each combination. The combination of the selected indices (In), vicinity operators (VO), groups (Gp), and aggregation operators (AO) results in a large set of descriptors for each protein. All these descriptors are univocally identified following the convention: In_VO_Gp_AO. In the example shown in Figure 2, the options selected to generate the descriptor are highlighted in red.

Next, we briefly describe, step by step, the general process of calculating the protein descriptors using ProtDCal, for the human prion protein fragment shown in Figure 2.

Step 1: *Residue codification (indices).* ProtDCal has implemented a list of indices (Tables S1–S4), mostly extracted from the AAindex database[28] that represent several structural and chemical physical properties of amino acids. For each residue in the protein, according to the indices selected by the user, an array of numerical values is created. This list of indices is

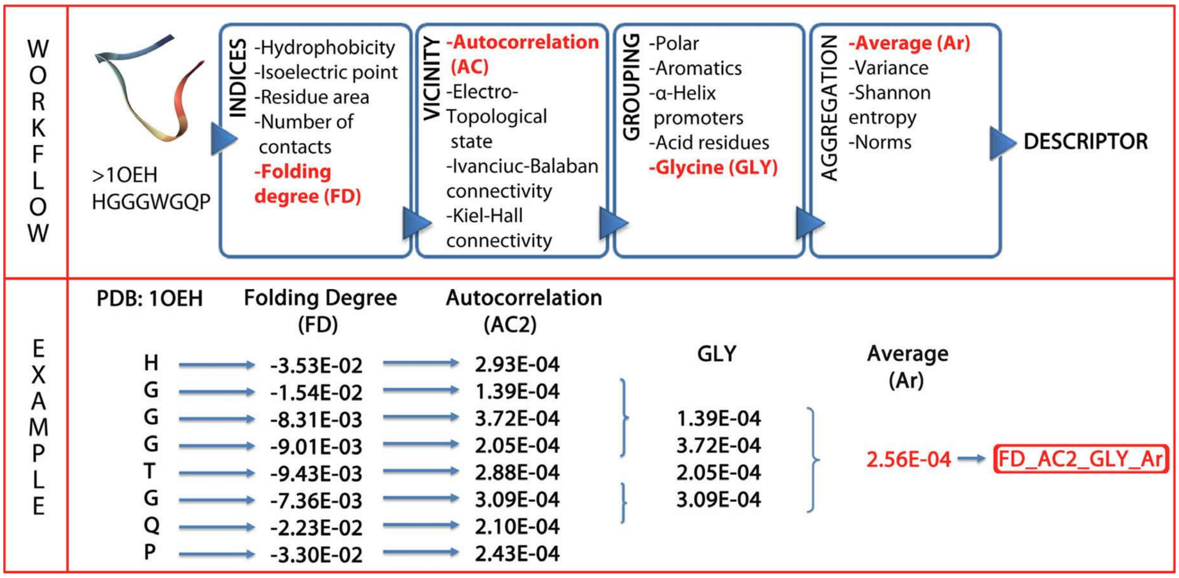


FIGURE 2 ProtDCal steps for calculating a protein descriptor. The fragment of a human prion protein (upper panel, far left) with Protein Data Bank[24–26] identification code 1OEH[27] is used as an example of protein under codification

then used to encode the residues in the protein in order to obtain sequence-based and 3D-structure protein descriptors.

In the example shown in Figure 2, we use the folding degree (FD) as residue index. FD is a geometrical parameter,[29] which significantly correlates with the folding rate constant and the average of the logarithm of the folding degree (lnFD) along all the residues in the protein.

$$\ln FD_i = -\frac{\sum_{j;|j-i|>1}^{N} |j-i| / d_{ij}^3}{N-x}$$

where d is the spatial Euclidian distance, N the length of the protein, and x a parameter that takes value 2 for terminal residues and 3 for all the others. In the example, FD is selected as index to provide an initial numerical characterization of all residues in the protein. In addition to the folding degree, more than 30 geometrical and chemical–physical indices (e.g., hydrophobicity, number of contacts, molar weight, solvent accessible surface area) are implemented in ProtDCal, which results in a great variability of the information captured by different descriptors.

Step 2: *Modification by vicinity*. Here, the numeric values in each array of index values are modified according to the values of neighboring residues within the sequence. Different definitions of "neighborhood" result in several potential vicinity operators (Table S5). The application of vicinity-modification operators to the values of a specific index array allows to include information in the final descriptor that reflects the ordering of the amino acids within the protein.

In the example of Figure 2, the autocorrelation operator of order 2 (AC2) is used to modify the initial FD values of each residue. This is achieved by incorporating information of the values from residues separated by two amino acids along the sequence. The operator is formulated as:

$$FD\_AC2_i = FD_i {*} FD_{i-k} + FD_i {*} FD_{i+k}$$

where i represents the *i-th* residue in a protein and k corresponds to the order of the autocorrelation.

Step 3: *Grouping*. Subarrays of groups of residues are formed, according to a set of grouping criteria implemented in ProtDCal[5] (Tables S6–S8). Among them, the entire protein forms the largest group, while the shortest group could contain a single type of residue. Such splitting of information in the amino acid sequence results in highly specific descriptors applicable to various protein analysis-related problems. In the example shown in Figure 2, the group is formed by all glycine residues (GLY) in the protein.

Step 4: *Aggregation operators*. Finally, an aggregation operator is applied to the columns of each matrix obtained after grouping, to transform such matrix into a final numeric descriptor. Available aggregation operators include the p-norms of orders $p = 1$ to $p = 3$,[30] central-tendency measures (geometric, average, and harmonic means, among others), dispersion and distribution parameters (kurtosis, variance, quartiles, skewness), and information-theoretic measures based on Shannon entropy[31] (Tables S9–S12). The different aggregation operators deliver distinct information about the property and the group used to generate the descriptors. In this way, descriptors derived from norms are most appropriate for modeling protein functions and classes that are dependent on protein size. On the contrary, for classes that are not related to the number of residues, descriptors obtained with dispersion and central tendency (means) aggregation operators may be preferable. In Figure 2, the arithmetic mean (Ar) is used to aggregate the values in the group into a single scalar value.

After following these four steps, the final descriptor resulting from the selected options (Figure 2, highlighted in red) is: FD_AC2_GLY_Ar. Hence, the structural information in this descriptor can be read as the average value (Ar) for all glycine amino acids (GLY), of the *modified* folding degree (FD) property, according to the autocorrelation (AC2) operator between neighboring residues.

2.2.2 | Analysis of ProtDCal's outcome

PROFEAT,[6] PROTEIN RECON,[32] and PseAAC[7,8] are among the most notable available tools for calculating large numbers of sequence-based physicochemical protein features. We used principal component analysis (PCA) to compare these methods to ProtDCal[5] (Figure 3). PCA was applied on the matrix of all computed descriptors. Then, the contribution of each program was measured using the loading values to evaluate the correlation between the original descriptors and the principal components. A given component is said to be loaded by a descriptor arising from one program when the correlation between the descriptor and a component is higher than 0.7.

The application of PCA resulted in 191 principal components, explaining 95% of the total variance in the descriptor data. Notably, while PROFEAT explains 45% of the variance (90 components loaded), ProtDCal descriptors are able to explain 52% of the variance (103 components loaded, Figure 3 top). Of the 20 top-ranked components (Figure 3, bottom), 16 have high loadings uniquely from ProtDCal. This analysis indicates that the components of ProtDCal capture most of the data variance. Importantly, ProtDCal captures information that it is not contained in other descriptors such as those of PROFEAT and PROTEIN RECON.

The information content of the structural descriptors generated by ProtDCal makes them suitable for modeling various functions and properties of proteins. However, given the large number of descriptors that ProtDCal delivers, the application of feature selection methods is required as an intermediate step between generating a raw feature matrix and training the final model. Machine-learning platforms, such as Weka[33] offer several methods to perform feature selection based on both unsupervised and supervised approaches. Depending of the size of the data set and the number of initial features, this step can be computationally demanding. Importantly, the resulting subset of features can determine the quality of the final model. Thus, to offer users an initial processing of the feature matrix, our web server characterizes each descriptor using standardized Shannon Entropy (sSE).

$$sSE = \frac{-\sum_{i=1}^{N} p_i \log p_i}{\log N}$$

where p_i is the probability that a randomly selected instance (protein) belongs to the interval i and N is the number of intervals over which the range of descriptor values is split. We use uniform splitting to obtain all the intervals. The number of instances in the data set determines the number of bins. In this way, the range of the sSE values for each descriptor is within (0,1), ranging from zero, corresponding to a total absence of variability, to one, corresponding to a uniformly distributed data set along the descriptor range. Accordingly, plots of the frequency histogram per interval of sSE and of the cumulative frequency along the data set are provided to the user (Figure 4). Then, users can perform an initial reduction of the feature matrix by requesting a subset

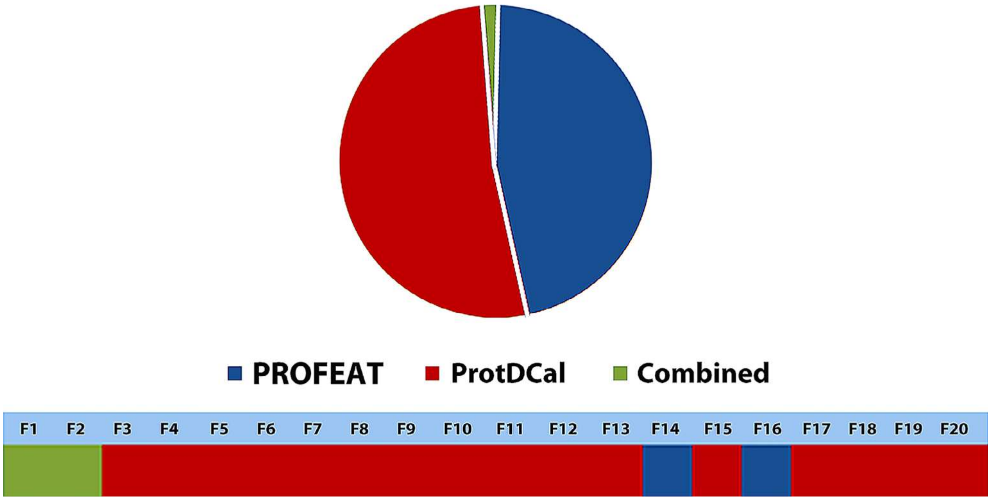


FIGURE 3 PCA test. Top: Pie chart showing all 191 principal components. Bottom: Bar diagram of the 20 top-ranked composed components of the test. The descriptors from the RECON program were highly redundant, thus they are found only within the first two "combined components."

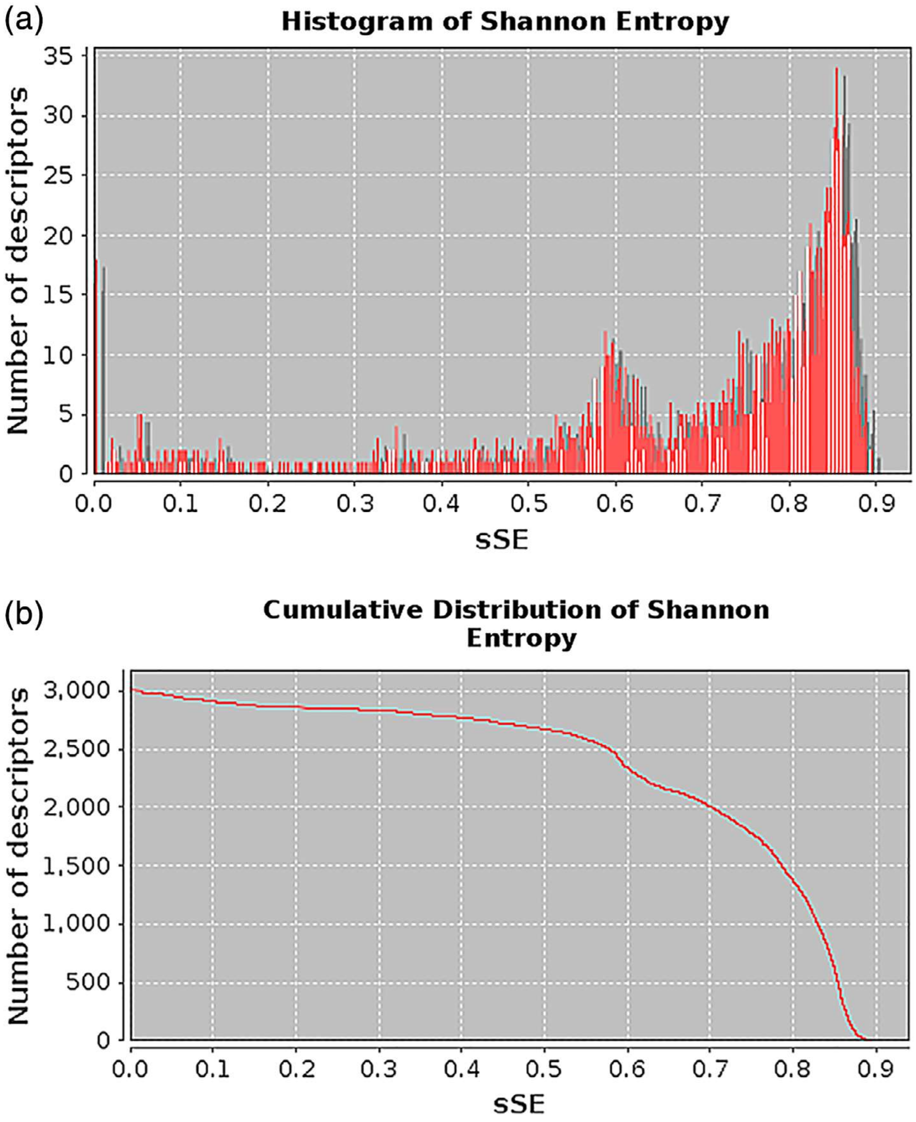


FIGURE 4 Illustration of the information content plots derived from a set of 3,000 descriptors calculated with ProtDCal. (a) A frequency histogram per interval of standardized Shannon entropy (sSE) is presented. (b) The cumulative frequency along the range of sSE is depicted

of descriptors within a custom interval of sSE. This preprocessing step presents, in a user-friendly manner, the dispersion of the obtained descriptors along the data set of input proteins. In addition, it enables the elimination of invariant descriptors that do not provide useful information. This step also allows discarding highly variable features that may not be as effective to model discrete properties, such as in a binary classification problem (e.g., active vs. inactive peptide drugs), where we generally seek descriptors following a bimodal distribution.

Independent tools, such as the IMMAN program,[34] allow for the advanced use of SE and several other information theoretic measures for applying both unsupervised and supervised feature selection to a set of descriptors. Information gain[35,36] is another widely used measure for supervised feature selection in machine-learning approaches. In future developments of our web server, we intend to implement these and other feature-selection analysis tools, for post-processing the descriptors generated by the ProtDCal server.

2.3 | Protein analysis tools

ProtDCal's features have been used to develop predictors for protein analysis.[9,14–17,19,20] In *ProtDCal-Suite* we provide, for the first time, web access to some of these tools.

2.3.1 | Performance measures

Next, we summarize the set of measures used to evaluate the predictors implemented in the different protein analysis tools.

$$\text{Precision (Pr)} = \text{TP}/(\text{TP}+\text{FP})$$
$$\text{Sensitivity(Sn)} = \text{TP}/(\text{TP}+\text{FN})$$
$$\text{Specificity(Sp)} = \text{TN}/(\text{TN}+\text{FP})$$
$$\text{Accuracy (Acc)} = (\text{TP}+\text{TN})/(\text{TP}+\text{TN}+\text{FP}+\text{FN})$$

where TP means true positive predictions, TN corresponds to true negative predictions, FP represents false positives, and FN indicates false negative predictions.

2.3.2 | PPI-detect

PPI-Detect[19] is a support vector machine (SVM) model that allows predicting the likelihood of interactions between two proteins based on their sequence information. The method is based on a new formalism that transforms pairs of amino acid sequences into general-purpose-numerical descriptors, which are used as input to an SVM classifier.

The benchmark employed for PPI-Detect was created using the publicly available databases of protein domains interaction data: 3did[37] and IPfam,[38] containing pairs of domains reported as interacting, and Negatome 2.0,[39] containing pairs of domains with no reported interactions. For each domain, the corresponding sequences were obtained from Pfam, a database with a large collection of protein families.[40] The final dataset comprises 1,922 interacting pairs and 2,405 noninteracting pairs of domains. Then, the data set was split into training (3,491 pairs: 1,613 positive and 1,878 negatives) and test (836 pairs: 309 positives and 527 negatives).

The theoretical background of PPI-Detect is described elsewhere.[19] Shortly, we defined new pairwise protein descriptors as follows: Provided two amino acid sequences A and B, and the reaction:

$$2A + 2B = > AB + BA$$

where AB and BA are block copolymers formed by the sequences of A and B.

The pairwise descriptor D(A-B) is calculated as: D(A-B) = D(AB) + D(BA) − 2D(A) − 2D(B), where D(X) corresponds to the value of the single-chain descriptor for a given sequence X (A, B, AB, or BA in this example). The value of D(A-B) is related to the change in the topological information upon the dimerization process. We note that the contribution of the unaltered partners is removed, thus the descriptors are a numerical representation of the relation between the independent sequences. We obtained the individual descriptors using the electro-topological state (E-State) vicinity operator, which allows capturing the topological information of both the original and combined sequences.

The training was performed with the SVM package SMO[22,41] and the final model was selected with a linear kernel and a cost (C) for misclassified cases, C = 11.3. The results of an external test for PPI-Detect and the tools PIPE,[42] Pred-PPI,[43] and SPPS[44] indicate that PPI-Detect outperforms, in terms of accuracy, the other tools (Table 1).

PPI-Detect was successfully used to identify improved derivatives of EPI-X4,[45,46] an endogenous peptide inhibitor of the G-protein-coupled receptor CXCR4.[19]

2.3.3 | Enzyme identifier

Enzyme Identifier is a SVM predictor for identifying enzyme-like proteins[15] from sequence or structural data. Accordingly, two models are implemented in Enzyme Identifier: sequence-based (using FASTA Files) and structure-based (using PDB files).

TABLE 1 Comparison of the accuracy values for PPI-detect and other PPI predictors[19]

	PIPE	Pred-PPI	SPPS	PPI-detect
Accuracy (%)	63.9	43.5	61.7	66.1

Abbreviation: PPI, protein–protein interaction.

TABLE 2 Comparison of performance measures in 10-fold cross-validation for ProtDCal-based models (enzyme identifier) and other methods[15]

Reference	Accuracy (%)
Enzyme identifier (3D structures)[15]	82.0 ± 0.3
Shervashidze[48]	81.5 ± 1.5
Senelle[49]	80.3
Dobson et al.[47]	80.2 ± 1.2
Shervashidze et al.[50]	79.8 ± 0.4
Neumann et al.[51]	79.0 ± 0.2
Enzyme identifier (amino acid sequences)[a]	78.8 ± 0.2
Li et al.[52]	78.3
Bai and Hancock[53]	77.6
Orsini et al.[54]	76.6 ± 0.6
Kilhamn[55]	75.9
Johansson et al.[56]	75.4 ± 0.6

[a] Sequence-based model. Notice that all other models are based on 3D structural information.

The data set employed for training both models was taken from Dobson and Doig (D&D),[47] comprising a total of 1178 structurally diverse proteins (691 enzymes and 487 nonenzymes), extracted from the PDB and Medline Abstracts databases. The Enzyme Identifier SVM models were generated and validated using 10 × 10-fold CV. The accuracy values reported in Table 2 illustrate how this structure-based model outperforms structure-based predictors developed by other authors using the same data set.

In addition, the accuracy of the predictions of the 3D structure-based model was assessed in an external set of 52 proteins, which was structurally unrelated to the training data set. The accuracy obtained was 80.8%, while with the method of Dobson and Doig the reported accuracy is 79.0%.[47]

2.3.4 | Pred-NGlyco

Pred-NGlyco is a sequence based Random Forest (RF) model for predicting N-glycosylation sites in peptides and proteins. This model illustrated, for the first time, the applicability of ProtDCal's descriptors to model relevant protein structural

TABLE 3 Comparison of performance measures for Pred-NGlyco and other predictors in 10-fold cross validation test[5]

	Accuracy (%)	Sensitivity (%)	Specificity (%)
Pred-NGlyco	91.6	93.2	91.4
GPP	92.8	96.6	91.8
NetNGlyc	76.7	43.9	95.7
EnsembleGly[a]	95.0	98.0	77.0
ScanSite	79.8	72.7	81.9

[a]Evaluated in five-fold cross validation. The specificity value originally reported for EnsembleGly[59] actually corresponds to precision.

TABLE 4 Performance measures in external test set for Pred-NGlyco and GPP[5]

	Accuracy (%)	Sensitivity (%)	Specificity (%)	Precision (%)
GPP	66.2	97.2	62.7	22.7
Pred-NGlyco	87.1	93.5	86.4	43.6

data.[5] To build the model, 3,508 sequence-unique windows, with 15 amino acids of length, were extracted from an initial data set of 241 proteins in the OGLYCBASE[57] data set. Each window was centered on an asparagine residue and classified in glycosylated (positive) or nonglycosylated (negative). Then, ProtDCal sequence-based descriptors were computed for each position of these chains.

Feature selection was performed using a Wrapper approach, with a genetic algorithm as implemented in Weka.[33] The resulting model was compared via cross-validation to contemporary N-glycosylation predictors, such as GPP,[58] NetNglyc,[58] EnsembleGly,[59] and ScanSite.[60] The results (Table 3) indicated that, in general, Pred-NGlyco, EnsembleGly, and GPP outperform the methods NetNGlyc and ScanSite.

In addition, the Pred-NGlyco model was compared using an external test set to the predictor GPP[58] (Table 4, the web server associated with EnsembleGly is no longer available). The comparison shows higher performance for the Pred-NGlyco model with superior values of accuracy, specificity, and precision than those of GPP, while GPP showed slightly better sensitivity.

Like PPI-Detect and Enzyme Identifier, Pred-NGlyco is an example of the value of ProtDCal descriptors to model various biological data.

3 | SERVER DETAILS

The server is hosted in an Apache2 webserver and it was implemented in a two-layer architecture, divided into front-end and back-end. The front-end, written in PHP and JavaScript, is responsible for exchanging information with users. This layer is visualized with HTML5 and Bootstrap framework. All tools were implemented in the Java language using third-party libraries. The back-end is formed by a set of Perl scripts that manage job execution on a computer cluster system.

4 | CONCLUSIONS

ProtDCal-Suite is a valuable platform for the machine learning-based study of protein structure–function relationships. The principal module, ProtDCal, provides scientists with information-rich features datasets that describe key structural characteristics of proteins. These descriptors are highly suited for the training and evaluation of machine learning models used in the prediction of protein function. The information-theoretic post-processing of the generated protein descriptors enables rapid unsupervised feature selection, prior to the creation of the model.

The capability of ProtDCal to generate useful features was assessed in several studies developing novel machine learning-based tools.[9–19] Here, we present web interfaces for predicting the interaction likelihood of protein–protein and protein–peptide pairs (PPI-Detect), for identifying enzymes from amino acid sequences or 3D structures (Enzyme Identifier), and for predicting N-glycosylation sites in peptides and proteins (Pred-NGlyco).

In future, we will continue incorporating new applications based on ProtDCal features into *ProtDCal-Suite* to bring more functionalities to users. A next development will include a tool for the design of antibacterial peptides.

ACKNOWLEDGMENTS

E.S.-G. acknowledges the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy—EXC-2033—Project ID: 390677874 as well as the support of the Boehringer Ingelheim Foundation (Plus-3 grant). S. R.-M. and E.S.-G. acknowledge funding by the DFG—Project number 316249678—SFB 1279.

ORCID

Yasser B. Ruiz-Blanco https://orcid.org/0000-0001-5400-4427
James R. Green https://orcid.org/0000-0002-6039-2355
Elsa Sanchez-Garcia https://orcid.org/0000-0002-9211-5803

REFERENCES

1. Alberts B, Bray D, Hopkin K, et al. Protein structure and function. Essential cell biology. New York and London: Garland Science, 1997.

2. Ashburner M, Ball CA, Blake JA, et al. Gene ontology: Tool for the unification of biology. The gene ontology consortium. Nat Genet. 2000;25:25–29.
3. Rives AW, Galitski T. Modular organization of cellular networks. Proc Natl Acad Sci U S A. 2003;100:1128–1133.
4. Nelson D, Cox M. Principles of biochemistry. 4th ed. New York, NY: WH Freeman and Company, 2005.
5. Ruiz-Blanco YB, Paz W, Green J, Marrero-Ponce Y. ProtDCal: A program to compute general-purpose-numerical descriptors for sequences and 3D-structures of proteins. BMC Bioinform. 2015; 16:162.
6. Li ZR, Lin HH, Han LY, Jiang L, Chen X, Chen YZ. PROFEAT: A web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. Nucleic Acids Res. 2006;34:W32–W37.
7. Shen H-B, Chou K-C. PseAAC: A flexible web server for generating various kinds of protein pseudo amino acid composition. Analyt Biochem. 2008;373:386–388.
8. Liu B, Liu F, Wang X, Chen J, Fang L, Chou K-C. Pse-in-one: A web server for generating various modes of pseudo components of DNA, RNA, and protein sequences. Nucleic Acids Res. 2015;43: W65–W71.
9. Kleandrova VV, Ruso JM, Speck-Planche A, Dias Soeiro Cordeiro MN. Enabling the discovery and virtual screening of potent and safe antimicrobial peptides. Simultaneous prediction of antibacterial activity and cytotoxicity. ACS Combinat Sci. 2016;18:490–498.
10. Scheraga HA, Rackovsky S. Global informatics and physical property selection in protein sequences. Proc Natl Acad Sci U S A. 2016;113:1808–1810.
11. Simeon S, Li H, Win TS, et al. PepBio: Predicting the bioactivity of host defense peptides. RSC Adv. 2017;7:35119–35134.
12. García-Jacas CR, Cabrera-Leyva L, Marrero-Ponce Y, Suárez-Lezcano J, Cortés-Guzmán F, García-González LA. GOWAWA aggregation operator-based global molecular characterizations: Weighting atom/bond contributions (LOVIs/LOEIs) according to their influence in the molecular encoding. Mol Inform. 2018;37:1800039.
13. Johnson D. Biotherapeutics: Challenges and opportunities for predictive toxicology of monoclonal antibodies. Intl J Mol Sci. 2018; 19:3685.
14. Ruiz-Blanco YB, Marrero-Ponce Y, García-Hernández E, Green J. Novel "extended sequons" of human N-glycosylation sites improve the precision of qualitative predictions: An alignment-free study of pattern recognition using ProtDCal protein features. Amino Acids. 2017;49:317–325.
15. Ruiz-Blanco YB, Agüero-Chapin G, García-Hernández E, Álvarez O, Antunes A, Green J. Exploring general-purpose protein features for distinguishing enzymes and non-enzymes within the twilight zone. BMC Bioinform. 2017;18:349.
16. Speck-Planche A, Kleandrova VV, Ruso JM, Cordeiro DS. MNFirst multitarget chemo-bioinformatic model to enable the discovery of antibacterial peptides against multiple gram-positive pathogens. J Chem Inform Model. 2016;56:588–598.
17. Corral-Corral R, Beltrán JA, Brizuela CA, Del Rio G. Systematic identification of machine-learning models aimed to classify critical residues for protein function from protein structure. Molecules. 2017;22:1673.
18. Yang Y, Urolagin S, Niroula A, Ding X, Shen B, Vihinen M. PON-tstab: Protein variant stability predictor. Importance of training data 1uality. Intl J Mol Sci. 2018;19:1009.
19. Romero-Molina S, Ruiz-Blanco YB, Harms M, Münch J, Sanchez-Garcia E. PPI-detect: A support vector machine model for sequence-based prediction of protein–protein interactions. J Comput Chem. 2019;40:1233–1242.
20. Biggar KK, Ruiz-Blanco YB, Charih F, et al. MethylSight: Taking a wider view of lysine methylation through computer-aided discovery to provide insight into the human methyl-lysine proteome. bioRxiv. 2018;274688.
21. Hassoun MH. Fundamentals of artificial neural networks. Cambridge: MIT Press, 1995.
22. Platt JC. Fast training of support vector machines using sequential minimal optimization. Advances in kernel methods. Cambridge: MIT Press, 1998; p. 185–208.
23. Breiman L. Random forests. Mach Learn. 2001;45:5–32.
24. Berman H, Henrick K, Nakamura H. Announcing the worldwide Protein Data Bank. Nat Struct Mol Biol. 2003;10:980–980.
25. Berman H, Henrick K, Nakamura H, Markley JL. The worldwide Protein Data Bank (wwPDB): Ensuring a single, uniform archive of PDB data. Nucleic Acids Res. 2006;35:D301–D303.
26. wwPDB consortium. Protein Data Bank: The single global archive for 3D macromolecular structure data. Nucleic Acids Res. 2018; 47:D520–D528.
27. Zahn R. The octapeptide repeats in mammalian prion protein constitute a pH-dependent folding and aggregation site. J Mol Biol. 2003;334:477–488.
28. Kawashima S, Kanehisa M. AAindex: Amino acid index database. Nucleic Acids Res. 2000;28:374–374.
29. Ruiz-Blanco YB, Marrero-Ponce Y, Prieto PJ, Salgado J, García Y, Sotomayor-Torres CM. A Hooke's law-based approach to protein folding rate. J Theoret Biol. 2015;364:407–417.
30. Dunford, N. and Schwartz, J.T. (1958) Linear Operators, Part I: General Theory. Wiley-Interscience, New York.
31. Shannon CE. A mathematical theory of communication. Bell Syst Tech J. 1948;27:379–423.
32. Sukumar N, Breneman CM. QTAIM in drug discovery and protein modeling. The quantum theory of atoms in molecules. Weinheim, Germany: Wiley VCH Verlag, 2007.
33. Frank E, Hall MA, Witten IH. The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques". USA: Morgan Kaufmann Publishers Inc., 2016.
34. Urias RWP, Barigye SJ, Marrero-Ponce Y, García-Jacas CR, Valdes-Martiní JR, Perez-Gimenez F. IMMAN: Free software for information theory-based chemometric analysis. Mol Divers. 2015;19:305–319.
35. Quinlan JR. Induction of decision trees. Machine Learning. 1986; 1:81–106.
36. Lee C, Lee GG. Information gain and divergence-based feature selection for machine learning-based text categorization. Inf Process Manage. 2006;42:155–165.
37. Mosca R, Céol A, Stein A, Olivella R, Aloy P. 3did: A catalog of domain-based interactions of known three-dimensional structure. Nucleic Acids Res. 2014;42:D374–D379.
38. Finn RD, Miller BL, Clements J, Bateman A. iPfam: A database of protein family and domain interactions found in the Protein Data Bank. Nucleic Acids Res. 2014;42:D364–D373.
39. Blohm P, Frishman G, Smialowski P, et al. Negatome 2.0: A database of non-interacting proteins derived by literature mining, manual annotation and protein structure analysis. Nucleic Acids Res. 2014;42:D396–D400.

40. Finn RD, Coggill P, Eberhardt RY, et al. The Pfam protein families database: Towards a more sustainable future. Nucleic Acids Res. 2016;44:D279–D285.
41. Keerthi SS, Shevade SK, Bhattacharyya C, Murthy KRK. Improvements to Platt's SMO algorithm for SVM classifier design. Neural Comput. 2001;13:637–649.
42. Pitre S, Dehne F, Chan A, et al. PIPE: A protein-protein interaction prediction engine based on the re-occurring short polypeptide sequences between known interacting protein pairs. BMC Bioinform. 2006;7:365–365.
43. Guo Y, Yu L, Wen Z, Li M. Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences. Nucleic Acids Res. 2008;36:3025–3030.
44. Liu X, Liu B, Huang Z, Shi T, Chen Y, Zhang J. SPPS: A sequence-based method for predicting probability of protein-protein interaction partners. PLoS One. 2012;7:e30938.
45. Buske C, Kirchhoff F, Munch J. EPI-X4, a novel endogenous antagonist of CXCR4. Oncotarget. 2015;6:35137–35138.
46. Zirafi O, Kim KA, Standker L, et al. Discovery and characterization of an endogenous CXCR4 antagonist. Cell Rep. 2015;11:737–747.
47. Dobson PD, Doig AJ. Distinguishing enzyme structures from non-enzymes without alignments. J Mol Biol. 2003;330:771–783.
48. Shervashidze N. Scalable graph kernels. Tübingen, Germany: Universität Tübingen, 2012.
49. Senelle M. Measures on graphs: From similarity to density. Louvain-la-Neuve, Belgium: Université catholique de Louvain, 2014.
50. Shervashidze N, Schweitzer P, van Leeuwen EJ, Mehlhorn K, Borgwardt KM. Weisfeiler-Lehman Graph Kernels. J Mach Learn Res. 2011;12:2539–2561.
51. Neumann M, Garnett R, Bauckhage C, Kersting K. Propagation kernels: Efficient graph kernels from propagated information. Mach Learn. 2016;102:209–245.
52. Li G, Semerci M, Yener B, Zaki MJ. Effective graph classification based on topological and label attributes. Statist Analys Data Mining. 2012;5:265–283.
53. Bai L, Hancock ER. Depth-based complexity traces of graphs. Pattern Recogn. 2014;47:1172–1186.
54. Orsini F, Frasconi P, Raedt LD. Graph invariant kernels. IJCAI Proceedings-International Joint Conference on Artificial Intelligence. USA: AAAI Press / IJCAI, 2015.
55. Kilhamn J. Fast shortest-path kernel computations using approximate methods [master of science thesis]. University of Gothenburg; 2015.
56. Johansson FD, Frost O, Retzner C, Dubhashi D. Classifying large graphs with differential privacy. In: Torra V, Narukawa T, editors. Modeling decisions for artificial intelligence. Switzerland: Springer International Publishing, 2015, 2015; p. 3–17.
57. Gupta R, Birch H, Rapacki K, Brunak S, Hansen JE. O-GLYCBASE version 4.0: A revised database of O-glycosylated proteins. Nucleic Acids Res. 1999;27:370–372.
58. Hamby SE, Hirst JD. Prediction of glycosylation sites using random forests. BMC Bioinform. 2008;9:500.
59. Caragea C, Sinapov J, Silvescu A, Dobbs D, Honavar V. Glycosylation site prediction using ensembles of support vector machine classifiers. BMC Bioinform. 2007;8:438.
60. Obenauer JC, Cantley LC, Yaffe MB. Scansite 2.0: Proteome-wide prediction of cell signaling interactions using short sequence motifs. Nucleic Acids Res. 2003;31:3635–3641.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of this article.

How to cite this article: Romero-Molina S, Ruiz-Blanco YB, Green JR, Sanchez-Garcia E. *ProtDCal-Suite*: A web server for the numerical codification and functional analysis of proteins. *Protein Science*. 2019; 28:1734–1743. https://doi.org/10.1002/pro.3673

ProtDCal-Suite: A web server for the numerical codification and functional analysis of proteins

Sandra Romero-Molina[1], Yasser B. Ruiz-Blanco[1*], James R. Green[2] and Elsa Sanchez-Garcia[1*]

[1] Computational Biochemistry, Center of Medical Biotechnology, University of Duisburg-Essen, Essen, North Rhine-Westphalia, 45117, Germany
[2] Systems and Computer Engineering, Carleton University, Ottawa, Ontario, K1S 5B6, Canada

Supplementary Information

Table of Content:

Table SM-1. Compendium of structural and chemical-physical amino acid properties.
Table SM-2. Formulae and description of 3D-thermodynamics indices.
Table SM-3. Formulae and description of thermodynamics indices for protein sequences.
Table SM-4. Formulae and description of topographic indices.
Table SM-5. Summary of vicinity operators.
Table SM-6. Summary of the definitions of residue-based groups.
Table SM-7. Summary of the definitions of property-based groups.
Table SM-8. Summary of the definitions of topographic-based groups.
Table SM-9. Aggregation operators: Distance invariants.
Table SM-10. Aggregation operators: Mean (first statistical moment) invariants.
Table SM-11. Aggregation operators: Statistical (highest statistical moments) invariants.
Table SM-12. Aggregation operators: Information-Theory-based invariants.

Table SM-1. Compendium of structural and chemical-physical amino acid properties.*

	Mw	HP	IP	ECI	L1-9	Z1	Z2	Z3	ISA	Xi	Pa	Pb	Pt	ΔHf	Ap
ALA	71	1.8	6.01	0.05	19.2	0.07	-1.73	0.09	62.9	-77.85	1.29	0.9	0.78	-433.66	202.42
ARG	156	-4.5	10.76	1.69	17.8	2.88	2.52	-3.44	52.98	108.86	0.96	0.99	0.88	-403.21	557.81
ASN	114	-3.5	5.41	1.31	21.72	3.22	1.45	0.84	17.87	-55.42	0.9	0.76	1.28	-466.91	377.84
ASP	115	-3.5	2.77	1.25	17.14	3.64	1.13	2.36	18.46	47.89	1.04	0.72	1.41	-518.1	360.26
CYS	103	2.5	5.07	0.15	18.83	0.71	-0.97	4.13	78.51	160.13	1.11	0.74	0.8	-425.69	236.80
GLN	128	-3.5	3.22	1.31	18.55	3.08	0.39	-0.07	19.53	134.68	1.44	0.75	1	-479.54	439.85
GLU	129	-3.5	5.65	1.36	17.31	2.18	0.53	-1.14	30.19	53.27	1.27	0.8	0.97	-531.69	417.46
GLY	57	-0.4	5.97	0.02	19.48	2.23	-5.36	0.3	19.93	-148.03	0.56	0.92	1.64	-420.86	172.08
HIS	137	-3.2	7.59	0.56	13.97	2.41	1.74	1.11	87.38	-4.57	1.22	1.08	0.69	-378.92	417.33
ILE	113	4.5	6.02	0.09	20.76	-4.44	-1.68	-1.03	149.77	-104.8	0.97	1.45	0.51	-449.27	309.12
LEU	113	3.8	5.98	0.01	17.65	-4.19	-1.03	-0.98	154.35	-148.5	1.3	1.02	0.59	-448.27	318.85
LYS	128	-3.9	9.74	0.53	17.05	2.84	1.41	-3.14	102.78	47.61	1.23	0.77	0.96	-446.97	409.91
MET	131	1.9	5.74	0.34	17.88	-2.49	-0.27	-0.41	132.22	46.37	1.47	0.97	0.39	-435.34	332.93
PHE	147	2.8	5.48	0.14	16.81	-4.92	1.3	0.45	189.42	47.67	1.07	1.32	0.58	-376.77	414.12
PRO	97	-1.6	6.48	0.16	18.55	-1.22	0.88	2.23	122.35	169.73	0.52	0.64	1.91	-422.17	261.24
SER	87	-0.8	5.68	0.56	18.91	1.96	-1.63	0.57	19.75	30.24	0.82	0.95	1.33	-479.75	265.01
THR	101	-0.7	5.87	0.65	17.15	0.92	-2.09	-1.4	59.44	46.04	0.82	1.21	1.03	-483.37	292.47
TRP	186	-0.9	5.89	1.08	20.94	-4.75	3.65	0.85	179.16	178.69	0.99	1.14	0.75	-365.49	530.87
TYR	163	-1.3	5.66	0.72	16.86	-1.39	2.32	0.01	132.16	49.11	0.72	1.25	1.05	-446.32	472.98
VAL	99	4.2	5.97	0.07	17.88	-2.69	-2.53	-1.29	120.91	-106.5	0.91	1.49	0.47	-434.3	276.26

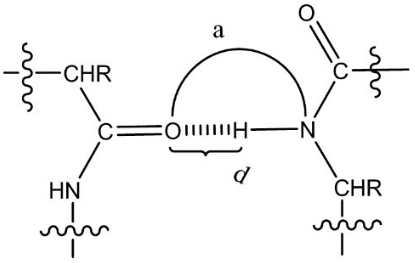
Mw Molar Weight [2]
HP Kyte-Doolitle's Hydrophobicity Scale [4]
IP Isoelectric Point [2]
ECI Electronic Charge Index[3]
L1-9 Compatibility parameter [5]
Z1 Composed parameter related with hydrophilicity [7]
Z2 Composed parameter related with steric features [7]
Z3 Composed parameter related with electronic features [7]

ISA Isotropic Surface Area [3]
Xi Compatibility parameter [5]
Pa Levitt's Probability of adopting alpha helix conformation [6]
Pb Levitt's Probability of adopting beta sheet conformation [6]
Pt Levitt's Probability of adopting beta turn conformation [6]
ΔHf(X) enthalpy of formation of the peptide: AAAA**X**AAAA. [5]
Ap Molecular area of non-carbon atoms in the sidechain

*This table was taken from the Supplementary Information of the ProtDCal manuscript.[1]

Table SM-2. Formulae and description of 3D-thermodynamics indices.*

Acronym	Formula	Description
Gc(F)	$G_c(F)_i = RT(N-1)p_i \ln p_i$, $p_i = \left(\dfrac{3}{2\pi(i-1)3.8^2}\right)^{3/2} e^{-\frac{3r_i^2}{2(i-1)3.8^2}}$	Configurational free energy of a folded state. Index based on a "random-flight" model of the protein chain. [8] Where r_i represents the distance to the first residue in the chain.
W(F)	$W_i^F = \sum_{j=1}^{N} \delta_{ij}^{ng} \delta_j^s N_j^w$	Number of water molecules close to a residue in a folded state.[9; 10] Where δ^{ng} takes value 1 if the pair of residues are neighbours, using a cutoff for the spatial distance (9.4 Å), or 0 otherwise. In the same way δ^s takes value 1 if the residue is superficial, using a cutoff for the solvent accessible surface area, or 0 otherwise. The parameters N^w represents the number of associated water molecules to the sidechain of a residue [11].
Gw(F)	$G_w(F)_i = -TR\delta_{hyd} \ln \dfrac{W_i^F!}{(W_i^F - N_i^w)!}$	Free energy contribution of the entropy of the first shell of water molecules in a folded state [10]. δ_{hyd} takes value 1 if the residue has non-zero N_i^w, or zero otherwise.
Gs(F)	$G_s(F)_i = H_i A_i^F$	Interfacial free energy contribution of a folded state. Where H_i is hydrophobicity in Kyte-Doolittle scale [4] and A^F is the solvent accessible surface area of a residue in a folded state.
ΔG_s	$\Delta G_{s\,i} = G_s(F)_i - G_s(U)_i$	Interfacial free energy variation.
HBd	$\Delta Hbd_i = 0.5\sum_{j=1}^{N}(\delta_{ij}^N + \delta_{ij}^O)$ Geometric definition of a H-bond:	Number of backbone's hydrogen bonds. Where δ_{ij}^N takes value 1 if the Nitrogen atom of residue *i* is H-bonded with the Oxygen atom of residue *j* and 0 otherwise. In the same way δ_{ij}^O takes value one if the Oxygen atom of residue

	 d < 2.5 Å a = 120º -- 180º	*i* is H-bonded with the Nitrogen atom of residue *j* and zero otherwise.
ΔG$_{el}$	$$\Delta G_{el\,i} = -\frac{\mathrm{k_{el}}}{2\bar{r}^2}\sum_{j=1}^{N}\frac{q_i q_j r_i r_j}{r_{ij}}$$	Free energy contribution of the charge distribution within the protein. The parameters *q* are the Electronic Charge Indices of each residue [3]. Parameter k_{el}= 7.608.
ΔG$_{w}$	$$\Delta G_{w\,i} = k_w(G_w(F)_i - G_w(U)_i)$$	Folding free energy contribution of the entropy of the first shell of water molecules.
ΔG$_{LJ}$	$$\Delta G_{LJ\,i} = \frac{\mathrm{k_{LJ}}}{2}\sum_{\substack{j=1;\\ \lvert j-i\rvert>1}}^{N}\left[\left(\frac{3.965}{r_{ij}}\right)^{12} - \left(\frac{3.965}{r_{ij}}\right)^{6}\right]$$	Residue-level Lennard-Jones interactions. Parameter k_{LJ} = 63.981.
ΔG$_{tor}$	$$\Delta G_{tor\,i} = k_{tor}[(\cos^2 2\phi_i - 1) + 0.256(\cos^2 2\psi_i - 1)]$$	Free energy contribution of backbone torsion angles. Parameter k_{tor}= 1.219.

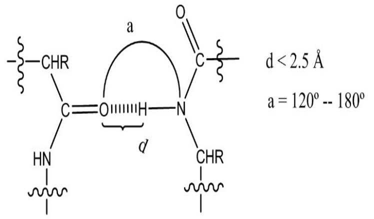
*This table was taken from the Supplementary Information of the ProtDCal manuscript.[1]

Table SM-3. Formulae and description of thermodynamics indices for protein sequences.*

Acronym	Formula	Description
W(U)	$W_i^U = \sum_{j=i-2}^{i+2} N_j^w$	Number of water molecules close to a residue in an unfolded state [10].
Gw(U)	$G_w(U)_i = -TR\delta_{hyd} \ln \frac{W_i^U!}{(W_i^U - N_i^w)!}$	Free energy contribution from the entropy of the first shell of water molecules in an unfolded state [10].
Gs(U)	$G_s(U)_i = H_i A_i^U$	Interfacial free energy contribution of an unfolded state

*This table was taken from the Supplementary Information of the ProtDCal manuscript.[1]

Table SM-4. Formulae and description of topographic indices.*

Acronym	Formula	Description
A_F	-	Solvent accessible surface area
ΔA	$\Delta A = A_F - A_U$	Buried area. Where A_U is the fully exposed surface area of each residue and A_F is the area in the folded state.
ΔA^{np}	$\Delta A^{np} = A^{np}_F - A^{np}_U$	Buried non-polar area. Here nitrogen atoms and oxygen atoms are excluded.
wSp	$wSp_i = \omega_i * \delta_i^s$	Weighted index of the solvent accessibility. Where ω represents any weighting property and the delta takes value 1 or 0 if the residue is considered superficial or internal respectively.
lnFD	$\ln FD_i = -\frac{\sum_{j;\|j-i\|>1}^{N} \|j-i\| / d_{ij}^3}{N-x}$	Logarithm of the Folding Degree. Where *d* is the spatial distance, *N* the length of the protein and *x* a parameter which takes value 2 for terminal residues and 3 for the others.
wR^2	$wRG_i^2 = \frac{w_i * d_i^2}{\sum_{i=1}^{N} w_i}$	Weighted Squared Radius. Where ω represents any weighting property and *d* is the spatial distance.
wΔHBd	$\Delta Hbd_i = \omega_i * (\delta_N + \delta_O)$ Geometric definition of a H-bond:  a O CHR C C O H N d HN CHR d < 2.5 Å a = 120º -- 180º	Weighted deficit or excess of the H-bond between the backbone atoms. Where δ_{ij}^N takes value 1 if the nitrogen atom of residue *i* is buried ($A_{(N)}$ < 0.01Å) and is not H-bonded with any oxygen atom or 0 otherwise. In the same way δ_{ij}^O takes value 1 if the oxygen atom of residue *i* is buried ($A_{(O)}$ < 0.01Å) and is not H-bonded with any nitrogen atom and 0 otherwise.
wNc	$wNc_i = 0.5\sum_{j\neq i}^{N} \omega_{ij}\delta_{ij}^c$	Weighted Number of Contact. Where δ_{ij}, takes value 1 when the contact conditions are fulfilled and 0 otherwise. A contacts is defined for pair of residues with spatial distances shorter than a cutoff *d* and topological distances longer than a cutoff *t*. The parameter ω_{ij} represents a weighting coefficient for each pair of residues. This parameter is computed as the product, $\omega_i\omega_j$, of the values, for each residue, of any property within a pool of 12 amino acid properties covering structural, physical-chemical features.
wFLC	$wFLC_i = \frac{\sum_{\|j-i\|\leq 4}^{N} \omega_{ij}\delta_{ij}^c}{\sum_{i=1}^{N}\sum_{j=1}^{N} \omega_{ij}\delta_{ij}^c}$	Weighted Fraction of Local Contacts. The parameters δ_{ij} and ω_{ij} means the same as previous but here the topological cutoff value is fixed in *t* = 1.
wNLC	$wNLC_i = 0.5\sum_{\|j-i\|\leq 4}^{N} \omega_{ij}\delta_{ij}^c$	Weighted Number of Local Contact The parameters δ_{ij} and ω_{ij} means the same as in *wNc* but here the topological cutoff value is fixed in *t* = 1.
wCO	$wCO_i = \frac{1}{2NN_c}\sum_{j\neq i}^{N} \omega_{ij}\delta^c_{ij}$	Weighted Relative Contact Order [12]. Where Nc represents the number of contacts in the protein.

wLCO	$wLCO_i = \frac{\sum_{j\neq i}^{N}\omega_{ij}\delta_{ij}^{c}}{N\sum_{j\neq i}^{N}\delta_{ij}^{c}}$	Weighted Local Contact Order. As difference with previous, the weighted contacts are divided by the same un-weighted local contact instead of all the contact in the protein.
wRWCO	$wRWCO_i = \frac{\sum_{j\neq i}^{N}\omega_{ij}\delta_{ij}^{c}}{N}$	Weighted Residue-Wise Contact Order [13].
wCTP	$wCTP_i = \frac{1}{2NN_c}\sum_{j\neq i}^{N}\omega_{ij}^{2}\delta_{ij}^{c}$	Weighted Chain Topology Parameter [14].
wCLQ	$wCLQ_i = \frac{\sum_{j<l}\delta_{ij}\delta_{il}\delta_{lj}\omega_{ij}\omega_{il}\omega_{lj}}{\sum_{j<l}\delta_{ij}\delta_{il}\omega_{ij}\omega_{il}}$	Weighted Cliquishness or Clustering Coefficient [15].
wPsi_H	$Psi\_H_i = \delta_i^{\psi H} * \omega_i$	Weighted Helix-like Psi angle. The delta takes value 1 if the angle is in the range [-77;-17] or 0 otherwise.
wPsi_S	$Psi\_S_i = \delta_i^{\psi S} * \omega_i$	Weighted Sheet-like Psi angle. The delta takes value 1 if the angle is in the range [94;154] or 0 otherwise.
wPsi_I	$Psi\_I_i = \delta_i^{\psi I} * \omega_i$	Weighted Irregular Psi angle. The delta takes value 1 if the angle is in one of the following ranges: [-180,-77), (-17;94), (154;180] or 0 otherwise.
wPhi_H	$Phi\_H_i = \delta_i^{\phi H} * \omega_i$	Weighted Helix like Phi angle. The delta takes value 1 if the angle is in the range [-87;-27] or 0 otherwise.
wPhi_S	$Phi\_S_i = \delta_i^{\phi S} * \omega_i$	Weighted Sheet like Phi angle. The delta takes value 1 if the angle is in the range [-159;-99] or 0 otherwise.
wPhi_I	$Phi\_I_i = \delta_i^{\phi I} * \omega_i$	Weighted Irregular Phi angle. The delta takes value 1 if the angle is in one of the following ranges: [-180,-159), (-99;-87), (-27;180] or 0 otherwise.
Phi	-	Phi dihedral angle
Psi	-	Psi dihedral angle
TCD	$wTCD_i = \frac{1}{2N^2}\sum_{j\neq i}^{N}\omega_{ij}\delta_{ij}^{c}$	Total Contact Distance [16].

*This table was taken from the Supplementary Information of the ProtDCal manuscript.[1]

Table SM-5. Weighting procedures (*Vicinity modifiers*) implemented in ProtDCal.*

Acronym	Formula	Description
$\mathrm{AC_i}^k$	$AC_i^k = \sum_{j\geq 1}^{N} L_i L_j \delta(d_{ij},k)$ $Condition: (d_{ij}=k)?\delta = 1 : \delta = 0$	Autocorrelation. Where, L_x are the index values of residues *i* and *j* and *k* is a topological distance cutoff and *N* is the total number of residues.
$\mathrm{GV_i}^k$	$GV_i^k = \frac{1}{N}\sum_{j=1;j\neq i}^{N}\frac{L_i L_j \delta(d_{ij},k)}{d_{ij}}$	Gravitational
$\mathrm{KH_i}^m$	$KH_i^m = \sum_{\alpha=1}^{A}\sqrt{\prod_{j=1}^{n_\alpha} L_{j\alpha}}$	Kier-Hall's connectivity-based operator. Where, *A* is the number of segments containing the residue *i*, with a maximum length of *m* residues, n_α is the number of residues in a sub-segment, $L_{j\alpha}$ is the index value of the residue *j* in the segment *α*.
$\mathrm{ES_i}$	$ES_i = L_i + \Delta L_i = L_i + \sum_{j=1;j\neq i}^{N}\frac{L_i - L_j}{(d_{ij}+1)^2}$	Electro-topological state (E-state index). Where, L_i is the intrinsic state (index) of the i^{th} residue and ΔL_i is the field effect on the *ith* residue calculated as perturbation of the index value (L_i) of i^{th} residue by all other residues in the protein, d_{ij} is the topological distance between the i^{th} and the j^{th} residue, and N is the total number of residues.
$\mathrm{IB_i}^2$	$IB_i^2 = (N-1)\sum_{j\neq i}^{N} a_{ij}\left(S_i S_j\right)^{-1/2}$ $S_i = L_i + \sum_{j\neq i}^{N} a_{ij} L_j$	Ivanciuc-Balaban. Where, a_{ij} represents th elements of the adjacency matrix, and *N* is the number of residues. The exponent 2 dues to the use of the exponent -1/2. Here the factor (N-1) represents the numbers of virtual bonds among residues.

*This table was taken from the Supplementary Information of the ProtDCal manuscript.[1]

Table SM-6. Summary of the definitions of residue-based groups.*

Acronym	Description
ALA	Represents all alanine residues contained in the protein
ARG	Represents all arginine residues contained in the protein.
ASN	Represents the all asparagine residues contained in the protein.
ASP	Represents the all aspartic residues contained in the protein.
CYS	Represents the all cysteine residues in the protein.
GLN	Represents the all glutamine residues in the protein.
GLU	Represents the all glutamic residues in the protein.
GLY	Represents the all glycine residues contained in the protein.
HIS	Represents the all histidine residues contained in the protein.
ILE	Represents the all isoleucine residues in the protein.
LEU	Represents the all leucine residues contained in the protein.
LYS	Represents the all lysine residues contained in the protein.
MET	Represents the all methionine residues contained in the protein.
PHE	Represents the all phenylalanine residues contained in the protein.
PRO	Represents the all proline residues contained in the protein.
SER	Represents the all Serine residues contained in the protein.
THR	Represents the all threonine residues contained in the protein.
TRP	Represents the all tryptophan residues contained in the protein.
TYR	Represents the all tyrosine residues contained in the protein.
VAL	Represents the all valine residues contained in the protein.

*This table was taken from the Supplementary Information of the ProtDCal manuscript.[1]

Table SM-7. Summary of the definitions of property-based groups.*

Acronym	Included Residues	Description
AHR	ALA, CYS, GLN, GLU, HIS, LEU, LYS, MET	Common residues in alpha helix motifs.
BSR	ILE, PHE, THR, TRP, TYR, VAL	Common residues in beta sheet motifs.
RTR	ASN, ASP, GLY, PRO, SER	Common residues in reverse turn motifs.
PCR	ARG, HIS, LYS	Positive-electric-charged residues.
NCR	ASP, GLU	Negative-electric-charged residues.
UCR	ASN, CYS, GLN, SER, THR, TYR	Uncharged residues.
ARM	HIS, PHE, TRP, TYR	Aromatic residues.
ALR	ALA, GLY, ILE, LEU, MET, PRO, VAL	Aliphatic residues.
UFR	GLY, PRO	Common residues promoting unfolding or distorted regions.
NPR	ALA, GLY, ILE, LEU, MET, PHE, PRO, TRP, VAL	Non-polar residues.
PLR	ARG, ASN, ASP, CYS, GLN, GLU, HIS, LYS, SER, THR, TYR	Polar residues.

*This table was taken from the Supplementary Information of the ProtDCal manuscript.[1]

Table SM-8. Summary of the definitions of topographic-based groups.*

Acronym	**Description**
HEX	All residues in alpha helix conformation
SHT	All residues in beta sheet conformation
TRN	All residues in reverse turn conformation
RCL	All residues in loops regions (Residues in TRN are excluded)
INT	Represents the all internal residues in the protein.
SUP	Represents all superficial residues contained in the protein.
PRT	The whole protein

*This table was taken from the Supplementary Information of the ProtDCal manuscript.[1]

Table SM-9. Aggregation operators: Distance invariants.*

Acronym	**Formula**	**Description**
N1	$N1 = \sum_{i=1}^{N} \lvert L_i \rvert$	Minkowski's norms (p = 1) Manhattan norm. Where L_i represents each index of the group of indices and N the number of indices in the group.
N2	$N2 = \sqrt{\sum_{i=1}^{N} \lvert L_i \rvert^2}$	Minkowski's norms (p = 2) Euclidean norm. Where L_i represents each index of the group of indices and N the number of indices in the group.
N3	$N3 = \sqrt[3]{\sum_{i=1}^{N} \lvert L_i \rvert^3}$	Minkowski's norms (p = 3). Where L_i represents each index of the group of indices and N the number of indices in the group.

*This table was taken from the Supplementary Information of the ProtDCal manuscript.[1]

Table SM-10. Aggregation operators: Means (first statistical moment) invariants.*

Acronym	Formula	Description
G	$G = \sqrt[N]{\prod_{i=1}^{N} L_i}$	Geometric Mean. Where N is the number of indices in the group.
Ar	$m_\alpha = \left(\frac{L_1^\alpha + L_2^\alpha + ... + L_N^\alpha}{N} \right)^{\frac{1}{\alpha}}$	Arithmetic Mean (potential with α = 1)
P2		Potential Mean (potential with α = 2)
P3		Potential Mean (potential with α = 3)
M		Harmonic Mean (potential with α = -1)

*This table was taken from the Supplementary Information of the ProtDCal manuscript.[1]

Table SM-11. Aggregation operators: Statistical (highest statistical moments) invariants.*

Acronym	Formula	Description
V	$\text{V} = \frac{\sum_{i=1}^{N}\left(L_i - \bar{L}\right)^2}{N\text{ - }1}$	Variance. Where N is the number of indices in the group.
S	$S = \frac{N(X_3)}{(N-1)(N-2)(DE)^3}$ $X_3 = \sum_{a=1}^{N}(L_a - \bar{L})^3$	Skewness. Where N is the number of indices in the group and (DE)^3 is the standard deviation raised to the 3rd power
K	$k = \frac{N(N+1)X_4 - 3(X_2)(X_2)(N-1)}{(N-1)(N-2)(N-3)(DE)^4}$ $X_j = \sum_{a=1}^{N}(L_a - \bar{L})^j$	Kurtosis. Where (DE)^4 is the standard deviation raised to the fourth power
DE	$DE = \sqrt{\frac{\left(\sum L_i - \bar{L}\right)^2}{N-1}}$	Standard Deviation
CV	$c_v = {}^{s}\!/_{\bar{L}}$	Variation Coefficient
RA	$RA = L_{\max} - L_{\min}$	Range
Q1	$\text{P25} = \left[\frac{\text{N}}{4} + \frac{1}{2}\right]$	Percentile 25. Where N is the number of indices in the group.
Q2	$\text{P50} = \left[\frac{\text{N}}{2} + \frac{1}{2}\right]$	Percentile 50. Where N is the number of indices in the group.
Q3	$\text{P75} = \left[\frac{3\text{N}}{4} + \frac{1}{2}\right]$	Percentile 75. Where N is the number of indices in the group.
I50	I50 = P75 – P25	Inter-quartile Range
MX	L_i maximum	Maximum value of the group of indices.
MN	L_j minimum	Minimum value of the group of indices.

*This table was taken from the Supplementary Information of the ProtDCal manuscript.[1]

Table SM-12. Aggregation operators: Information-Theory-based invariants. *

Acronym	Formula (Equation)	Description
MI	$MI = -\sum_{i=1}^{K} \frac{N_k}{N} \log_2 \frac{N_k}{N}$	Mean Information Content. Where N$_k$ is the number of indices in the same bin, K is the number of bins defined to compute the operator and N is the total number of indices in the group.
TI	$TI = N \log_2 N - \sum_{k=1}^{K} N_k \log_2 N_k$	Total Information Content.
SI	$SI = \frac{TI}{N \log_2 N}$	Standarized Infomation Content

*This table was taken from the Supplementary Information of the ProtDCal manuscript.[1]

References

1. Ruiz-Blanco YB, Paz W, Green J, Marrero-Ponce Y (2015) ProtDCal: A program to compute general-purpose-numerical descriptors for sequences and 3D-structures of proteins. BMC Bioinformatics 16:162.
2. Lehninger. Amino Acids, Peptides, and Proteins. (2005) Biochemistry. pp. 76-115.
3. Collantes ER, Dunn-III WJ (1995) Amino acid side chain descriptors for quantitative structure-activity relationship studies of peptide analogues. J Med Chem 38:2705-2713.
4. Kyte J, Doolitle RF (1982) A Simple Method for Displaying the Hydropathic Character of a Protein. . J Mol Biol 157:105-132.
5. S K, M K, J J (1999) Modeling of the amino acid side chain effects on peptide conformation. Bioorg Chem 27:434–442.
6. Levitt M (1978) Conformational Preferences of Amino Acids in Globular Proteins. Biochemistry 17.
7. Hellberg S. S, M., Skagerberg B., Wold, S. (1987) Peptide Quantitative Structure-Activity Relationship, a Multivariate Approach. J Med Chem 30:1126-1135. .
8. Kamide K, Dobashi T. Chapter 6: Statistical Mechanics and Excluded Volume of Polymer Chains. (2000) Physical Chemistry of Polymer Solutions Theoretical Background. Elsevier Science.
9. Ruiz-Blanco YB, García Y, Sotomayor-Torres CM, Marrero-Ponce Y (2010) New Set of 2D/3D Thermodynamic Indices for Proteins. A Formalism Based on "Molten Globule" Theory. Physics Procedia 8:63-72.
10. Ruiz-Blanco YB, Marrero-Ponce Y, Paz W, García Y, Salgado J (2013) Global Stability of Protein Folding from an Empirical Free Energy Function. Journal of Theoretical Biology 321:44-53.
11. Jiang L, Kuhlman B, Kortemme T, Baker D (2005) A "solvated rotamer" approach to modeling water-mediated hydrogen bonds at protein–protein interfaces. PROTEINS: Structure, Function, and Bioinformatics 58:893–904.
12. Plaxco KW, Simons KT, Baker D (1998) Contact Order, Transition State Placement and the Refolding Rates of Single Domain Proteins. J Mol Biol 277:985-994.
13. Burrage JSaK (2006) Predicting residue-wise contact orders in proteins by support vector regression. BMC Bioinformatics 425.
14. Nolting B, Schalike W, Hampel P, Grundig F, Gantert S, Sips N, Bandlow W, Qi PX (2003) Structural determinants of the rate of protein folding. J Theor Biol 223:299–307.
15. Micheletti C (2003) Prediction of Folding Rates and Transition-State Placement From Native-State Geometry. PROTEINS: Structure, Function, and Genetics 51:74–84.
16. Zhou H, Zhou Y (2002) Folding Rate Prediction Using Total Contact Distance. Biophysical Journal 82:458–463.

7 Discussion

In this work, I developed several ML-based tools for the design and optimization of bioactive peptides. To this end, I focused on different early-stage tasks of the drug design process. These tasks comprise the identification of protein-peptide and protein-protein interactions, the estimation of the binding affinity of protein-peptide and protein-protein complexes, and the identification of antibacterial peptides. Furthermore, I implemented several web applications to facilitate the use of the developed models. In this way, the predictors can be employed for (1) the massive *in silico* screening of peptide libraries to study protein-peptide interactions, (2) to identify and design peptides and protein derivatives with tuned binding affinity against a protein receptor, and (3) to identify protein fragments with antibacterial activity. In addition, I developed a web platform, ProtDCal-Suite, that provides access to these tools and to other ProtDCal-based applications. ProtDCal-Suite facilitates the functional analysis of proteins and peptides[72, 76, 86, 87, 212-214].

7.1 The models

First, I developed PPI-Detect, a sequence-based predictor of protein-protein and protein-peptide interactions (**Publication 1**). PPI-Detect receives as input the individual amino acid sequences of a protein-peptide or protein-protein pair and outputs the likelihood of the pair being involved in an interaction. I developed the ML model using as features the molecular descriptors implemented in the ProtDCal software[76], whose suitability in proteins-related QSAR development has been demonstrated in several studies[72, 76, 81, 86, 87, 90, 215]. To this end, and since the codification approach of ProtDCal was initially intended only for individual proteins, I defined a novel procedure to encode protein pairs and implemented a new functionality in ProtDCal to enable their future computation (**Publication 1, Figure 2**). Subsequently, I trained the ML model using a combination approach for protein pairs introduced and validated its effectiveness in the study of protein-protein and protein-peptide interactions. PPI-Detect can be applied to the identification of lead compounds from extensive *in silico* screening of protein-peptide interactions, especially in those cases where the primary structure of protein-peptide pairs is the only available information.

I further studied protein-peptide interactions by exploring the structural information of protein-peptide complexes (**Publication 2**). For this, the response variable modeled was the binding affinity (binding free energy, ΔG_{bind}) of protein-peptide complexes. The model delivers the strength of the interaction rather than identifying interacting/non-interacting binders, as was the case with PPI-Detect (**Publication 1**). Unlike PPI-Detect, the predictor of protein-peptide binding free energies is based on 3D structures. To facilitate the use of the created model, I developed a web tool named PPI-Affinity that allows, in addition to the estimation of BA, the optimization of a putative peptide sequence for which a 3D complex structure has been resolved. The implemented functionalities permit the generation of thousands of peptide derivatives by performing substitutions and/or deletions on the peptide residues located at the interface of contact of the protein-peptide complex. PPI-Affinity can find application in mutagenesis experiments and protein engineering. To my knowledge, without considering the study of peptides targeting MHC classes I and II[207, 208], this was the first protein-peptide BA predictor trained on data formed exclusively of protein-peptide structures. Additionally, as part of the study, I also modeled the BA of protein-protein complexes. Consequently, all the functionalities of PPI-Affinity can be exploited separately for both protein-peptide and protein-protein complexes.

Furthermore, I contributed to the development of ABP-Finder, a tool to identify antibacterial peptides and the Gram-staining type of the targeted bacteria (**Publication 3**). This study involved the development of two ML classifiers. The first model predicts the likelihood of a peptide exhibiting antibacterial activity. Putative antibacterial peptides are then fed to the second model, which classifies the activity according to the target bacteria. The classifications are exclusively Gram-positive (Gram+), exclusively Gram-negative (Gram-), or both types (broad-spectrum). My contribution to this project was mainly computational, by implementing the web server that enables the use of the developed models in the screening of large peptide libraries. The server permits the breakdown of protein sequences into short peptide fragments before the prediction. This facility paves the way for the discovery of protein domains with antibacterial activity.

7.2 Modelling process

I applied both classification (**Publication 1**) and regression (**Publication 2**) modeling approaches to effectively exploit the available data. I used regression as the primary strategy to create the BA predictors, given the numerical (continuous) nature of ΔG_{bind} values. However, modeling the BA of PPIs was a major challenge given the susceptibility of regression to noise which, due to different experimental conditions and measurement errors, is a common factor limiting the quality of reported data[216]. Previously, other authors have addressed the analysis of PPIs as a classification problem by introducing an artificial threshold value to define discrete classes[217]. Yet, for this option to be feasible, the values separating the classes must be very well defined, i.e., the distribution of binding free energies should be multimodal, or an unambiguous physically justified threshold should exist. That is not the case for the modeled data (**Publication 2**, **Figures SI-1 and SI-3 B**). To assess this, I evaluated PPI-Affinity and the

LUPIA[217] classifier in predicting the BA of protein-protein complexes as a classification rather than as a regression problem (**Publication 2**, **Table SI-5.1**). The evaluation on two test sets reflected the downside of using a threshold to discretize BA values. In both evaluations, LUPIA was overly optimistic by ranking most of the cases as with “high” affinity. Moreover, with the constant increase of available BA-related data (K_d, K_i, and IC_{50}), several ML models have been developed with a regression approach.

Since available datasets can be found for both endpoints, I studied both response variables. I created a binary classifier to discriminate between interacting and non-interacting protein-peptide and protein-protein sequence pairs (**Publication 1**), and regressors to calculate the ΔG_{bind} of protein-protein and protein-peptide complexes (**Publication 2**).

All modeling processes involved the definition of training and test sets. For PPI-Affinity and ABP-Finder, development sets were also defined. Development sets allow monitoring the generalization power of the models during the training steps. Although no development set was defined for the modeling of PPI-Detect, the lack of sequence identity in the analyzed data (**Publication 1**, **Figure 1**), as well as the size of the training set and the rigorous definition of three test sets (easy, mid-hard and very hard subsets), served likewise to ensure the robustness of the model. Among the three ML studies, ProtDCal features for protein structures (sequence-based and structure-based) were exploited to a large extent. The main difference between the calculated descriptors was the choice of indices and vicinity operators, the first two steps of the four comprising the ProtDCal pipeline to calculate the descriptors (**Publication 4**, **Figure 2**). The molecular descriptors modeled in PPI-Affinity accounted for structural information without vicinity modifications and aimed to encode contact information of residues at the PPI interface at a determined spatial distance. Indices and vicinity operators used in the modeling of PPI-Detect represented properties of the primary structure of proteins. In PPI-Detect, the application of the Electro-topological state (E-State) vicinity operator allowed to measure the total topological information of the PPI. The descriptors of the individual and aggregated sequence pairs were combined according to the following formulation (**Publication 1**, **Equation 9**):

$$D_{A-B} = D_{(AB)} + D_{(BA)} - 2D_{(A)} - 2D_{(B)} \qquad (14)$$

Where $D_{(A)}$, $D_{(B)}$, $D_{(AB)}$, and $D_{(BA)}$ correspond to the molecular descriptors of the sequences A and B, and that of the sequences AB and BA (formed by the concatenation of A and B), respectively.

This, accompanied by a large set of grouping and aggregation criteria (which are the last two configurable parameters of ProtDCal) produced a large set of molecular descriptors for each instance in the different data sets. The sets comprised 13248 and 23040 descriptors for PPI-Detect and PPI-Affinity, respectively. Such high multidimensional spaces aimed to explore all possible scope-related information to produce features correlating with the response variable. Next, I extracted the most informative descriptors by applying several unsupervised and supervised techniques. Among them, the Information Gain for classification tasks (**Publication 1**, **Publication 3**) and the correlation coefficient for regression (**Publication 2**) served to identify top-ranked features by evaluating the worth of each descriptor to predict the class. Additionally, unsupervised clustering allowed the removal of redundant dimensions based on the mutual correlation among features (**Publication 1**, **Publication 3**). In all studies, the final steps of the feature selection process involved the use of a supervised technique coupled with the classifier to explore relationships between different subsets of descriptors and the response variable. This strategy facilitated the detection of dependencies between the models' features.

In the case of PPI-Affinity, as the ΔG_{bind} values of most of the complexes in both datasets were in the center of the distribution (**Publication 2**, **Figures SI-1**, **SI-3-B**), once the final descriptors were fixed, I divided the training set into four subsets by subsampling the most populated ΔG_{bind} regions (**Publication 2**, **Figures SI-2 and SI-4**). This approach aimed to reduce large prediction errors in the least sampled BA ranges. Next, I modeled ΔG_{bind} on each dataset and followed an ensemble approach, based on vote selection, to create the final predictors. Ensemble-based predictions can reduce the dispersion of the estimates produced by single learners and thus achieve better performance[218]. Two ensemble-based tools (PPI-Affinity and ABP-Finder) are presented in this work, with robustness validated by their performance on several independent test sets.

In the modeling protocols, feature selection was followed by the optimization of the hyperparameters of the technique. In this step, the followed grid-search strategy allowed to explore a discrete set of hyperparameters to select those values producing the model that most approximates the objective function. Such selection involved the definition of different *ad-hoc* functions to evaluate the models using different test options and thus enhance the robustness of the final predictors. In PPI-Detect, I evaluated the models in 10-fold CV and selected the hyperparameters that maximized precision and recall values while reducing their deviation from the false positive rate (**Publication 1**, **Section SM-3**). In PPI-Affinity, I evaluated the performance of each model on the training set, in 10-fold CV, and on the development set. Then, I selected the best model by combining the three output correlations in a function that consolidated in a single measure the goodness-of-fit, generalization, and robustness of the model (**Publication 2**, **Figure 1-2**). In the training of SVM models, the polynomial and radial basis kernels were explored to determine the most suitable kernel function to predict the class. From them, the polynomial kernel was selected to train the models in both studies (**Publication 1**, **Publication 2**). The complexity hyperparameter (log_2C) of the technique took values in the range between -5 and 4 (with step=0.5) for all the models. However, I extended the minimum value to $log_2C = -7$ in the modeling of the BA of protein-peptide complexes, as optimal models initially fell in the lower limit of the defined range. As a result, two of the models had $log_2C = -6$, while the other two remained with $log_2C = -5$ (**Publication 2**, **Figure 2**). Another hyperparameter was the degree (D) of the polynomial kernel, with values ranging between one and three. Interestingly, the final PPI-Detect predictor is a linear model (D=1), while the ensembles of PPI-Affinity contain models trained with either D=1 or D=2. Such results indicate the presence of more complex nonlinear relationships in the structural data, and show how the followed modeling strategy complies with the principle of parsimony of always opting for the simplest solution.

7.3 Evaluation of the models

A proper evaluation of the models is paramount to ensure an accurate, robust, and stable production environment. As mentioned above, all models were evaluated in 10-fold cross-validation. This validation strategy provided an unbiased estimate of the generalization error with lower variance than that produced by a single split into training and test sets. However, we note that, to a lesser extent, CV involves the use of test samples in the fitting process. Therefore, validation on independent test sets, and comparisons with state-of-the-art methods, are crucial to ensure the generalization power and improvement of the models. Likewise, the use of appropriate metrics for the evaluation of the model is essential. The selection of the evaluation measures shall involve not only data types (classification or regression problems) but also their practical utility to solve the biological problem being addressed. Park[209] discussed the limitations of available sequence-based PPI predictors and highlighted the need for novel methods that consistently outperform those of the state-of-the-art in terms of area under the receiver operating characteristic curve (AUC) and precision-recall. AUC measures the performance of a binary classifier by comparing the true positive rate to the false positive rate obtained as the decision threshold varies. In this regard, Park[209] debated the superiority of precision-recall compared to AUC, as AUC underestimates the effect of absolute false positives, which might lead to overestimating the model's performance in drug discovery scenarios.

For the development of PPI-Detect, I used precision-recall curves (PRC) to evaluate the model, as it is highly relevant for biologists to avoid false positives when performing costly and time-consuming experiments. In a first evaluation using the entire training set and in 10-fold CV, PRC showed very low differences between these two testing approaches, evidencing the robustness of the fitting process (**Publication 1**, **Figure 3**). Subsequently, I plotted PRC to evaluate the model on the three initially defined test sets. Remarkably, models with more than 75% precision were obtained at a sensitivity of 50% for all test sets (**Publication 1**, **Figure 4**). The combination of mid-hard and very-hard subsets was used to compare PPI-Detect with three state-of-the-art methods. This evaluation evidenced the superior performance of PPI-Detect, as at 50% sensitivity, PPI-Detect achieved a precision of 80%, while all the other methods delivered precision values below 60% (**Publication 1**, **Figure 5**). Notably, despite the previous success of the PIPE method[115, 209], this tool did not benefit from the application of ML techniques and it lacks pairs of non-interacting (negative) data. Park[209] also emphasized the importance of using reliable non-interacting data to enhance model performance and mentioned the random sample of protein pairs with unproved interaction as a valid approach to generate such data. In this work, I used the negative PPI pairs reported in the Negatome database[62] to build the classifier. PPI-Detect delivered the best precision-recall balance (Pr=0.554, Sn=0.648) when compared to other state-of-the-art methods (**Publication 1**, **Table 1**). There, either higher precision or sensitivity values were achieved by PIPE (Pr=0.762, Sn=0.101) and Pred-PPI (Pr=0.396, Sn=0.88), yet with notable unbalance among these metrics. Such results evidenced the robustness of PPI-Detect and the effectiveness of our novel pair-wise codification approach to extend ProtDCal descriptors to a broader set of protein studies.

Likewise, I evaluated the performance of PPI-Affinity and other state-of-the-art BA predictors on several independent test sets. The protein-protein model was first assessed on two sets of protein-protein complexes. There, PPI-Affinity ranked second on the test set 1 (R=0.62, MAE=1.8 kcal/mol) and first on the test set 2 (R=0.50, MAE=1.8 kcal/mol) (**Publication 2**, **Table 1**). Overall, the performance of PPI-Affinity in both test sets was moderate but consistent, while the other methods exhibited low or notably fluctuating performance in both test sets. Finally, the ranking power of the protein-protein model was also assessed by predicting the binding free energies of 26 wild-type and 151 mutant protein-protein complexes taken from the SKEMPI v2.0 database[65] (**Publication 2**, **Figure 3**). The performance of PPI-Affinity on this set (R=0.78 and MAE=1.4 kcal/mol) was superior to the performance on the previously assessed test sets. Notably, such performance was just marginally inferior to that obtained when evaluating only on the 26 wild-type protein-protein complexes (R=0.77, MAE=1.1 kcal/mol). The evaluation on this third test set showed the potential of the model to characterize changes upon mutation, a functionality provided in the web server implementation (**Publication 2**,

Figure 5). Until the introduction of PPI-Affinity, BA predictors were trained on either protein-protein or protein-ligand datasets, with ligands accounting for small organic molecules (except for the MHC-I and MHC-II studies[207, 208]). In both modeling processes, protein-peptide complexes were poorly represented. Thus, the unsuitability of the other methods for the prediction of the BA of protein-peptide complexes was evidenced by assessing their performance on a set of 100 protein-peptide instances taken from the Biolip dataset[219] (**Publication 2**, **Table 2)**. There, the protein-peptide ensemble model of PPI-Affinity outranked all the assessed methods (R=0.55 and MAE=1.1 kcal/mol). Notably, below PPI-Affinity, the methods with the highest correlations delivered MAE values ranging between 8 and 10 kcal/mol, while methods with low MAE values (MAE < 2 kcal/mol) delivered the lowest correlation values (R < 0.24).

7.4 Evaluation on other experimental data

Finally, all the models were challenged to either replicate or predict the results of experimental measurements in collaboration with biologists. Mutants of EPI-X4, an endogenous inhibitor of the chemokine receptor CXCR4, were employed to assess the performance of PPI-Detect and PPI-Affinity. Two different datasets of EPI-X4 were leveraged, with activities measured in inhibition (**Publication 1**) and competition assays (**Publication 2**). In PPI-Detect, the model was used to study regions of CXCR4 interacting with 35 derivatives of CXCR4's endogenous ligand EPI-X4 (**Publication 1**, **Figure 6**). There, the precision values delivered by the model were between 50% and 70% on three of the four studied fragments of EPI-X4 (**Publication 1**, **Table 2**). Such results evidenced that EPI-X4 derivatives are potential binders of CXCR4. In addition, since these three EPI-X4 regions comprise the minor pocket of CXCR4, binding to the minor pocket is probably enhanced in those regions of the receptor. Subsequent studies involved the active participation of PPI-Detect in the study of EPI-X4 derivatives. To this end, ten thousand mutants were generated taking as template WSC02, a derivative of EPI-X4. From this screening, three of the generated derivatives were experimentally tested (**Publication 1**, **Table 3**). From them, a peptide (JM133) was found to be more active than EPI-X4 (**Publication 1**, **Figure 7**), which evidenced the capabilities of PPI-Detect for the virtual screening of PPIs.

In PPI-Affinity, I assessed the performance of the protein-peptide model in predicting the BA of 56 derivatives of EPI-X4 coupled to the CXCR4 receptor. For this, I evaluated the model for the identification of active peptides in two test modes: (1) defining as *active* peptides those with IC_{50} below the value of EPI-X4, and (2) defining as *active* those peptides with IC_{50} < 10,000 nM. To evaluate the screening performance, I calculated the Enrichment Factor on the top 5, 10, and 15 ranked peptides. This metric is useful in a scenario such as this, where the affinity of derivatives was measured up to IC_{50} = 10,000 nM in the experimental assays, denoting those with higher values as weak binders. In this analysis, PPI-Affinity predicted only one false positive in the top 15 when using the affinity of EPI-X4 as the cut-off value. For all the other calculated EF values, the model delivered the highest performance (EF=1.9) (**Publication 2**, **Figure 4**). The protein-peptide model of PPI-Affinity was also assessed on two sets of peptides bound to PDZ domains of the human high-temperature requirement serine proteases (HtrAs) HTRA1 and HTRA3 (**Publication 2**, **Tables 3-4**). In this assessment, most of the BA values calculated with PPI-Affinity had an error within the MAE of the model (**Publication 2**, **Table 2**).

Furthermore, I evaluated the ranking power of PPI-Affinity and other state-of-the-art BA predictors on these sets of derivatives by measuring the Kendall correlation coefficient between experimental and predicted binding affinities. This non-parametric measure is well-suited to address the small sizes of these datasets, containing each less than 15 peptides. In this evaluation, PPI-Affinity outranked the other methods with τ=0.59 (HTRA1) and τ=0.42 (HTRA3) (**Publication 2**, **Table 5**). Notably, PPI-Affinity showed the lowest variance among all the methods compared. RF-Score, the second-best method for predicting the BA of HTRA1 (t=0.56) suffered a fall in the performance for HTRA3 (t=0.27).

All the conducted evaluations evidenced the potential of the developed models in VS experiments and showed, with several examples, their practical utility in drug design. In addition to the methodological improvements achieved by the novel models, the evaluation on different and independent datasets unveiled a notable variance in the predictions of other state-of-the-art methods, especially in the case of BA predictors (**Publication 2**).

7.5 Implementation of web servers

As mentioned, in addition to the PPI-Detect and PPI-Affinity models, I implemented all the introduced tools as web servers to facilitate the screening of large libraries of compounds without installation requirements for the scientific community. All web interfaces implement validation functionalities to avoid data entry errors. For instance, the Protein Engineering module of PPI-Affinity (**Publication 2**) provides a four-steps form to customize mutant generation. There, information related to the structure provided by the user is interactively displayed and requested in a way that greatly minimizes errors related to user management. The aforementioned ML models were all created using the descriptors implemented in ProtDCal. This software was originally deployed as a stand-alone program written in the Java programing language. In this work, I also implemented a web platform, named ProtDCal-Suite, that allows calculating the vast variety of ProtDCal descriptors, as well as to provide access to the tools introduced in this thesis and to others that leveraged the protein codification of ProtDCal (**Publication 4**). ProtDCal-Suite provides a general framework for the study of proteins and peptides, permitting both (1) the generation of molecular descriptors for data-mining purposes, and (2) the application of our tools in the early steps of the peptide discovery process.

7.6 General considerations concerning the models

Several factors might be considered when analyzing the high variance presented by some state-of-the-art methods. For instance, many methods lack the definition of the applicability domain of the model, which is the 3rd principle defined by the OECD for the regulation of QSAR models[112]. Many ML models assume that new instances will come from an identical distribution to that of the training set[1], without considering that this accounts for a small portion of the chemical space. Consequently, if a tool lacks the definition of the AD, it is not possible to analyze whether test samples are simply outside the scope of these predictors or whether it may be a specific situation of some structure that causes an error in the prediction. This makes it difficult to analyze the errors obtained in the predictions and reflects the importance of specifying, for each predicted case, its projection into the AD of the model as part of the output of the tools. In this work, I defined the AD as the subspace specified by the value range of the variables of the models (molecular descriptors) in the training datasets (**Publication 1: Table SM6**, **Publication 2: Table SI-7**, **Table SI-8**, **Publication 3: Supplementary File 2**). By doing this, my aim was to release robust models providing reliable predictions. Nonetheless, there are other important considerations regarding the practical use of our tools:

1) The models can be applied only to linear sequences comprising natural and unmodified amino acids. Although this could appear as a limitation, it should be noted that the chemical space encompassing all possible combinations of the 20 standard amino acids is enormous and that the initial exploration of peptides libraries is aimed at finding hits to be further improved by techniques such as cyclization, *N*-methylation, and modifications of the amino acids, among others[33].
2) PPI-Detect was built using a training set where peptides have a minimum length of 16 amino acids, which hinders the applicability of the method to shorter peptides, generally preferred as therapeutics. However, this limitation is imposed by the available data and smaller peptides might be studied using PPI-Affinity, whose protein-peptide model was trained on peptides with sizes ranging between 3 and 30 amino acids.
3) It is known the concern of drug developers regarding the understanding of model features to be able to explain the success of ML models[1]. This was acknowledged by the 5th OECD principle[112], and in the widely accepted definition of molecular descriptor

given by Todeschini et. al.[71]. The codification approach implemented in ProtDCal significantly sacrifices the rigorous interpretation of the features. Nevertheless, the models introduced by us and other authors[72, 76, 81, 86, 87, 90, 212, 213, 215, 220] evidence the suitability of ProtDCal descriptors to train robust and generalizable models and their applicability to diverse types of studies. Tables summarizing the descriptors for each model are provided in the corresponding publications (**Publication 1: Table SM2**, **Publication 2: Tables SI-3.1 and SI-4.1**). Additionally, the formulations for all molecular descriptors are available in the section "Theory and Algorithms" of the ProtDCal-Suite paper (**Publication 4**).

4) As explained above, it should be noted that data-driven models are limited to some extent to the available information, which accounts for a reduced size of the chemical space. This might limit the AD of the models and thus the novelty of some predicted leads.
5) PPI-Detect and PPI-Affinity attempt to predict the on-target interactions of putative peptides, which does not directly imply bioactivity. For this, other factors such as metabolic activity, polypharmacological implications, and alternative binding modes, among others, must be considered[1].

Among all, it should be noted that *in silico* methods are approximate and thus the models can produce errors. Nevertheless, this does not hamper the application of the models. For instance, PPI-Detect was used for the screening of a peptide library to identify peptide inhibitors of *Escherichia coli* ATP synthase (high binding likelihood) which also display low binding likelihood to human ATP synthase[221]. In that work, the inhibitory activity of two of the top-ranked peptides identified by PPI-Detect was experimentally validated. Such results evidence the potential of PPI-Detect as a virtual screening tool. The authors conducted protein engineering computational experiments to generate fragments (peptides ranging between 20 and 40 residues) from F_OF_1-ATP synthase interfaces. As protein engineering tasks require algorithms capable of generating thousands of derivatives for a putative compound, the authors used the evolutionary algorithm implemented in ROSE[222] to generate mutants of the ATP synthase interfaces. Currently, this can be achieved by the functionalities implemented by me to allow the generation of mutants as part of the pipeline of PPI-Affinity (**Publication 2**, **Figure 5**).

7.7 Conclusions

The design of novel peptides requires the selection of putative hits compounds in the early stages of the process. In this search, *in silico* methods exploring the peptidome can reduce the cost of other techniques such as HTS, as well as extend the search to non-physical libraries. The amount of data collected in the fields of Proteomics and Peptidomics is continuously increasing, which opens the possibility of applying ML techniques to create predictive models that leverage the information content of available data. Several ML-based methods that analyze protein(peptide)-related endpoints exist. However, the value of any ML model resides in how well the model performs out-of-sample, and state-of-the-art methods so far left us room for improvement in terms of generalization on unseen data. Moreover, there was a niche in the state-of-the-art methods to predict the binding affinity of protein-peptide complexes.

I applied several computational techniques to develop novel methods contributing to the *de novo* design of bioactive peptides. The purpose of the tools is to detect promising peptide candidates in the early stages of drug discovery by providing high-value predictions. In this regard, the introduced methods improved state-of-the-art ML-based methods, which represents a scientific advance in the *in silico* study of bioactive peptides. Moreover, the development of a predictor of BA uniquely trained on protein-peptide complexes opened space for new research. The predictions made by using the server can generate important insights into the structural information of active compounds. Even if this knowledge must be validated experimentally, it reduces the time and costs associated with performing those experiments.

The developed ML-based tools can be leveraged in peptide discovery and optimization. In addition to complementing each other, the introduced methods may be used in a pipeline for the massive screening of protein-peptide associations. For instance, ABP-Finder can be used first to search peptide domains within a protein sequence with antibacterial activity. Then, those fragments identified as *active* peptides can be fed to PPI-Detect to predict the likelihood of interaction with a protein receptor. Next, the sequences detected as *interacting* can be delivered to other *in silico* approaches for building putative protein-peptide complexes. Subsequently, PPI-Affinity can be applied to the optimization of the peptide sequence or to rank interacting protein-peptide pairs according to their values of binding free energy.

The implementation of PPI-Detect, PPI-Affinity, ABP-Finder, and ProtDCal-Suite as web servers offers the opportunity of using the methods worldwide without the need to allocate large computational resources to the task. To date (November 26th, 2022), with an average of 18 jobs per day, 26134 jobs have been submitted from 57 countries to ProtDCal-Suite (published on July 4th, 2019) (**Figure 9**). Of them, 43% are from PPI-Detect (published on February 15th, 2019) and 19% from PPI-Affinity (published on June 2nd, 2022). Despite the recent introduction of ABP-Finder (published on November 26, 2022), the tool has already completed 92 jobs sent from eight countries.

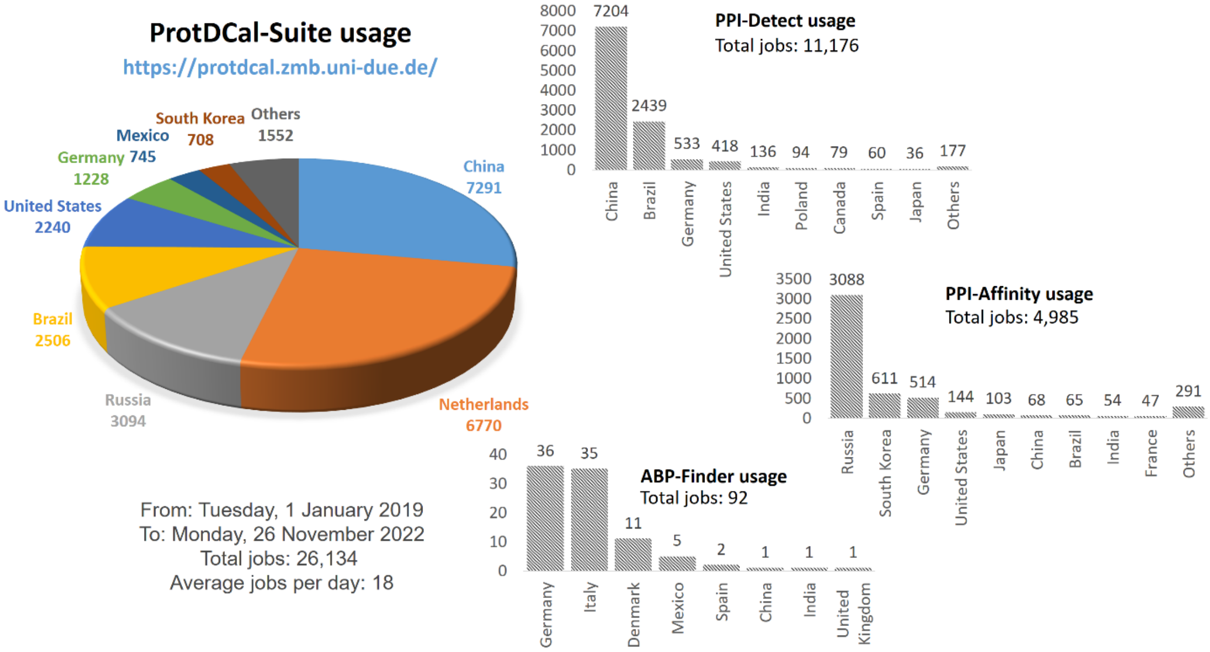


Figure 9. Worldwide usage of the tools developed within this work (stand November 26th, 2022)

Future work involves retraining the models to leverage the availability of novel data. In addition, we will introduce the BA predictor as a scoring function in docking algorithms. Forthcoming work also involves developing ML-based tools to predict ADME(T)-related properties to monitor the pharmaceutical profiles of putative peptides and optimize activity and stability. PPI-Detect, PPI-Affinity, and ABP-Finder can be used for the virtual screening of the peptidome to discover and optimize bioactive peptides against disorders such as cancer and infection, with implications for human health and societal well-being.

8 References

1. Thomas, M.; Boardman, A.; Garcia-Ortegon, M.; Yang, H.; de Graaf, C.; Bender, A., Applications of Artificial Intelligence in Drug Design: Opportunities and Challenges. In *Artificial Intelligence in Drug Design*, Heifetz, A., Ed. Springer US: New York, NY, 2022; pp 1-59.

2. Schlander, M.; Hernandez-Villafuerte, K.; Cheng, C.-Y.; Mestre-Ferrandiz, J.; Baumann, M., How Much Does It Cost to Research and Develop a New Drug? A Systematic Review and Assessment. *PharmacoEconomics* **2021,** *39* (11), 1243-1269.

3. Zhavoronkov, A., Artificial Intelligence for Drug Discovery, Biomarker Development, and Generation of Novel Chemistry. *Molecular Pharmaceutics* **2018,** *15* (10), 4311-4313.

4. Incaviglia, I.; Frutiger, A.; Blickenstorfer, Y.; Treindl, F.; Ammirati, G.; Lüchtefeld, I.; Dreier, B.; Plückthun, A.; Vörös, J.; Reichmuth, A. M., An Approach for the Real-Time Quantification of Cytosolic Protein-Protein Interactions in Living Cells. *ACS Sensors* **2021,** *6* (4), 1572-1582.

5. Blaszczak, E.; Lazarewicz, N.; Sudevan, A.; Wysocki, R.; Rabut, G., Protein-fragment complementation assays for large-scale analysis of protein-protein interactions. *Biochemical Society Transactions* **2021,** *49* (3), 1337-1348.

6. Wang, H.; Dawber, R. S.; Zhang, P.; Walko, M.; Wilson, A. J.; Wang, X., Peptide-based inhibitors of protein-protein interactions: biophysical, structural and cellular consequences of introducing a constraint. *Chemical Science* **2021,** *12* (17), 5977-5993.

7. Peng, X.; Wang, J.; Peng, W.; Wu, F.-X.; Pan, Y., Protein–protein interactions: detection, reliability assessment and applications. *Briefings in Bioinformatics* **2017,** *18* (5), 798-819.

8. Lu, H.; Zhou, Q.; He, J.; Jiang, Z.; Peng, C.; Tong, R.; Shi, J., Recent advances in the development of protein–protein interactions modulators: mechanisms and clinical trials. *Signal Transduction and Targeted Therapy* **2020,** *5* (1), 213.

9. Zinzalla, G.; Thurston, D. E., Targeting protein–protein interactions for therapeutic intervention: a challenge for the future. *Future Medicinal Chemistry* **2009,** *1* (1), 65-93.

10. Kuzmanov, U.; Emili, A., Protein-protein interaction networks: probing disease mechanisms using model systems. *Genome Medicine* **2013,** *5* (4), 37.

11. Thompson, T. B.; Chaggar, P.; Kuhl, E.; Goriely, A.; for the Alzheimer’s Disease Neuroimaging, I., Protein-protein interactions in neurodegenerative diseases: A conspiracy theory. *PLOS Computational Biology* **2020,** *16* (10), e1008267.

12. Carro, L., Protein–protein interactions in bacteria: a promising and challenging avenue towards the discovery of new antibiotics. *Beilstein Journal of Organic Chemistry* **2018,** *14*, 2881-2896.

13. Ryu, J. Y.; Kim, J.; Shon, M. J.; Sun, J.; Jiang, X.; Lee, W.; Yoon, T.-Y., Profiling protein–protein interactions of single cancer cells with in situ lysis and co-immunoprecipitation. *Lab on a Chip* **2019,** *19* (11), 1922-1928.

14. Modell, A. E.; Blosser, S. L.; Arora, P. S., Systematic Targeting of Protein–Protein Interactions. *Trends in Pharmacological Sciences* **2016,** *37* (8), 702-713.

15. Wang, X.; Ni, D.; Liu, Y.; Lu, S., Rational Design of Peptide-Based Inhibitors Disrupting Protein-Protein Interactions. *Frontiers in chemistry* **2021,** *9*, 682675-682675.

16. Hughes, J. P.; Rees, S.; Kalindjian, S. B.; Philpott, K. L., Principles of early drug discovery. *British Journal of Pharmacology* **2011,** *162* (6), 1239-1249.

17. Govardhanagiri, S.; Bethi, S.; Nagaraju, G. P., Small Molecules and Pancreatic Cancer Trials and Troubles. In *Breaking Tolerance to Pancreatic Cancer Unresponsiveness to Chemotherapy*, 2019; Vol. 5, pp 117-131.

18. Cunningham, A. D.; Qvit, N.; Mochly-Rosen, D., Peptides and peptidomimetics as regulators of protein–protein interactions. *Current Opinion in Structural Biology* **2017,** *44*, 59-66.

19. Guo, W.; Wisniewski, J. A.; Ji, H., Hot spot-based design of small-molecule inhibitors for protein–protein interactions. *Bioorganic & Medicinal Chemistry Letters* **2014,** *24* (11), 2546-2554.

20. Wang, L.; Wang, N.; Zhang, W.; Cheng, X.; Yan, Z.; Shao, G.; Wang, X.; Wang, R.; Fu, C., Therapeutic peptides: current applications and future directions. *Signal Transduction and Targeted Therapy* **2022,** *7* (1), 48.

21. Alberts, B.; Johnson, A.; Lewis, J.; Raff, M.; Roberts, K.; Walter, P., The Shape and Structure of Proteins. In *Molecular biology of the cell. 4th edn*, Garland Science: New York, 2002.

22. Apostolopoulos, V.; Bojarska, J.; Chai, T.-T.; Elnagdy, S.; Kaczmarek, K.; Matsoukas, J.; New, R.; Parang, K.; Lopez, O. P.; Parhiz, H.; Perera, C. O.; Pickholz, M.; Remko, M.; Saviano, M.; Skwarczynski, M.; Tang, Y.; Wolf, W. M.; Yoshiya, T.; Zabrocki, J.; Zielenkiewicz, P.; AlKhazindar, M.; Barriga, V.; Kelaidonis, K.; Sarasia, E. M.; Toth, I., A Global Review on Short Peptides: Frontiers and Perspectives. *Molecules* **2021,** *26* (2).

23. Sánchez, A.; Vázquez, A., Bioactive peptides: A review. *Food Quality and Safety* **2017,** *1* (1), 29-46.

24. Chakrabarti, S.; Guha, S.; Majumder, K., Food-Derived Bioactive Peptides in Human Health: Challenges and Opportunities. *Nutrients* **2018,** *10* (11).

25. Hafeez, Z.; Benoit, S.; Cakir-Kiefer, C.; Dary, A.; Miclo, L., Food protein-derived anxiolytic peptides: their potential role in anxiety management. *Food & Function* **2021,** *12* (4), 1415-1431.

26. Maestri, E.; Marmiroli, M.; Marmiroli, N., Bioactive peptides in plant-derived foodstuffs. *Journal of Proteomics* **2016,** *147*, 140-155.

27. Nongonierma, A. B.; FitzGerald, R. J., The scientific evidence for the role of milk protein-derived bioactive peptides in humans: A Review. *Journal of Functional Foods* **2015,** *17*, 640-656.

28. Halim, N. R. A.; Yusof, H. M.; Sarbon, N. M., Functional and bioactive properties of fish protein hydolysates and peptides: A comprehensive review. *Trends in Food Science & Technology* **2016,** *51*, 24-33.

29. Huan, Y.; Kong, Q.; Mou, H.; Yi, H., Antimicrobial Peptides: Classification, Design, Application and Research Progress in Multiple Fields. *Frontiers in Microbiology* **2020,** *11*, 582779.

30. Lau, J. L.; Dunn, M. K., Therapeutic peptides: Historical perspectives, current development trends, and future directions. *Bioorganic & Medicinal Chemistry* **2018,** *26* (10), 2700-2707.

31. Fosgerau, K.; Hoffmann, T., Peptide therapeutics: current status and future directions. *Drug Discovery Today* **2015,** *20* (1), 122-128.

32. Raeisi Estabragh, M. A.; Bami, M. S.; Ohadi, M.; Banat, I. M.; Dehghannoudeh, G., Carrier-Based Systems as Strategies for Oral Delivery of Therapeutic Peptides and Proteins: A Mini-Review. *International Journal of Peptide Research and Therapeutics* **2021,** *27* (2), 1589-1596.

33. Di, L., Strategic Approaches to Optimizing Peptide ADME Properties. *The AAPS Journal* **2015,** *17* (1), 134-143.

34. Brock, W. H.; Jensen, K. A.; Jørgensen, C. K.; Kauffman, G. B., The origin and dissemination of the term "ligand" in chemistry. *Polyhedron* **1983,** *2* (1), 1-7.

35. Wang, Y.; Xing, J.; Xu, Y.; Zhou, N.; Peng, J.; Xiong, Z.; Liu, X.; Luo, X.; Luo, C.; Chen, K.; Zheng, M.; Jiang, H., In silico ADME/T modelling for rational drug design. *Quarterly Reviews of Biophysics* **2015,** *48* (4), 488-515.

36. Salunke, D. M., Multiple target sites for designing candidate drugs. *Biochemical Journal* **2018,** *475* (5), 977-979.

37. Prieto-Martínez, F. D.; López-López, E.; Juárez-Mercado, K. E.; Medina-Franco, J. L., Chapter 2 - Computational Drug Design Methods—Current and Future Perspectives. In *In Silico Drug Design*, Roy, K., Ed. Academic Press: 2019; pp 19-44.

38. Lee, A. C.; Harris, J. L.; Khanna, K. K.; Hong, J.-H., A Comprehensive Review on Current Advances in Peptide Drug Development and Design. *International Journal of Molecular Sciences* **2019,** *20* (10), 2383.

39. Vázquez, J.; López, M.; Gibert, E.; Herrero, E.; Luque, F. J., Merging Ligand-Based and Structure-Based Methods in Drug Discovery: An Overview of Combined Virtual Screening Approaches. *Molecules* **2020,** *25* (20), 4723.

40. Mao, J.; Akhtar, J.; Zhang, X.; Sun, L.; Guan, S.; Li, X.; Chen, G.; Liu, J.; Jeon, H.-N.; Kim, M. S.; No, K. T.; Wang, G., Comprehensive strategies of machine-learning-based quantitative structure-activity relationship models. *iScience* **2021,** *24* (9), 103052.

41. Cai, Z.; Zafferani, M.; Akande, O. M.; Hargrove, A. E., Quantitative Structure–Activity Relationship (QSAR) Study Predicts Small-Molecule Binding to RNA Structure. *Journal of Medicinal Chemistry* **2022,** *65* (10), 7262-7277.

42. Hansch, C.; Maloney, P. P.; Fujita, T.; Muir, R. M., Correlation of Biological Activity of Phenoxyacetic Acids with Hammett Substituent Constants and Partition Coefficients. *Nature* **1962,** *194* (4824), 178-180.

43. Samuel, A., Eight-move opening utilizing generalization learning,(See Appendix B, Game G-43.1 Some Studies in Machine Learning Using the Game of Checkers). *IBM Journal* **1959**, 210-229.

44. El Bouchefry, K.; de Souza, R. S., Chapter 12 - Learning in Big Data: Introduction to Machine Learning. In *Knowledge Discovery in Big Data from Astronomy and Earth Observation*, Škoda, P.; Adam, F., Eds. Elsevier: 2020; pp 225-249.

45. Abu-Mostafa, Y. S.; Magdon-Ismail, M.; Hsuan-Tienm, L., *Learning from Data: a Short Course*. AMLBook.com: 2012.

46. Mohri, M.; Rostamizadeh, A.; Talwalkar, A., *Foundations of Machine Learning*. The MIT Press Cambridge, Massachusetts, 2018.

47. Dunjko, V.; Taylor, J. M.; Briegel, H. J., Quantum-Enhanced Machine Learning. *Physical Review Letters* **2016,** *117* (13), 130501.

48. Sutton, R. S.; Barto, A. G., *Reinforcement Learning: An Introduction, second edition*. The MIT Press: Cambridge, Massachusetts, 2018.

49. LeCun, Y.; Bengio, Y.; Hinton, G., Deep learning. *Nature* **2015,** *521*, 436-444.

50. Alloghani, M.; Al-Jumeily, D.; Mustafina, J.; Hussain, A.; Aljaaf, A. J., A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science. In *Supervised and Unsupervised Learning for Data Science*, Springer International Publishing: Cham, 2020; pp 3-21.

51. Höge, M.; Wöhling, T.; Nowak, W., A Primer for Model Selection: The Decisive Role of Model Complexity. *Water Resources Research* **2018,** *54* (3), 1688-1715.

52. Belkin, M.; Hsu, D.; Ma, S.; Mandal, S., Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences* **2019,** *116* (32), 15849-15854.

53. Beyeler, M.; Rounds, E.; Carlson, K.; Dutt, N.; Krichmar, J., Neural correlates of sparse coding and dimensionality reduction. *PLOS Computational Biology* **2019,** *15*, e1006908.

54. Bishop, C. M., *Pattern Recognition and Machine Learning*. 2006.

55. Sarker, I. H., Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science* **2021,** *2* (3), 160.

56. Breiman, L., Bagging predictors. *Machine Learning* **1996,** *24* (2), 123-140.

57. Freund, Y.; Schapire, R. E., A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences* **1997,** *55* (1), 119-139.

58. Breiman, L., Stacked regressions. *Machine Learning* **1996,** *24* (1), 49-64.

59. Aslam, B.; Basit, M.; Nisar, M. A.; Khurshid, M.; Rasool, M. H., Proteomics: Technologies and Their Applications. *Journal of Chromatographic Science* **2017,** *55* (2), 182-196.

60. Mosca, R.; Céol, A.; Stein, A.; Olivella, R.; Aloy, P., 3did: a catalog of domain-based interactions of known three-dimensional structure. *Nucleic Acids Research* **2014,** *42* (D1), D374-D379.

61. Finn, R. D.; Miller, B. L.; Clements, J.; Bateman, A., iPfam: a database of protein family and domain interactions found in the Protein Data Bank. *Nucleic Acids Research* **2014,** *42* (D1), D364-D373.

62. Blohm, P.; Frishman, G.; Smialowski, P.; Goebels, F.; Wachinger, B.; Ruepp, A.; Frishman, D., Negatome 2.0: a database of non-interacting proteins derived by literature mining, manual annotation and protein structure analysis. *Nucleic Acids Research* **2014,** *42* (D1), D396-D400.

63. Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E., The Protein Data Bank. *Nucleic Acids Research* **2000,** *28* (1), 235-242.

64. Wang, R.; Fang, X.; Lu, Y.; Wang, S., The PDBbind Database: Collection of Binding Affinities for Protein−Ligand Complexes with Known Three-Dimensional Structures. *Journal of Medicinal Chemistry* **2004,** *47* (12), 2977-2980.

65. Jankauskaitė, J.; Jiménez-García, B.; Dapkūnas, J.; Fernández-Recio, J.; Moal, I. H., SKEMPI 2.0: an updated benchmark of changes in protein–protein binding energy, kinetics and thermodynamics upon mutation. *Bioinformatics* **2019,** *35* (3), 462-469.

66. Aguilera-Mendoza, L.; Marrero-Ponce, Y.; Beltran, J. A.; Tellez Ibarra, R.; Guillen-Ramirez, H. A.; Brizuela, C. A., Graph-based data integration from bioactive peptide databases of pharmaceutical

interest: toward an organized collection enabling visual network analysis. *Bioinformatics* **2019,** *35* (22), 4739-4747.

67. Beginner's Guide to the PDBbind Database (v.2020). http://www.pdbbind.org.cn/download/pdbbind_2020_intro.pdf.

68. Cherkasov, A.; Muratov, E. N.; Fourches, D.; Varnek, A.; Baskin, I. I.; Cronin, M.; Dearden, J.; Gramatica, P.; Martin, Y. C.; Todeschini, R.; Consonni, V.; Kuz'min, V. E.; Cramer, R.; Benigni, R.; Yang, C.; Rathman, J.; Terfloth, L.; Gasteiger, J.; Richard, A.; Tropsha, A., QSAR Modeling: Where Have You Been? Where Are You Going To? *Journal of Medicinal Chemistry* **2014,** *57* (12), 4977-5010.

69. Jiang, D.; Wu, Z.; Hsieh, C.-Y.; Chen, G.; Liao, B.; Wang, Z.; Shen, C.; Cao, D.; Wu, J.; Hou, T., Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models. *Journal of Cheminformatics* **2021,** *13* (1), 12.

70. Liu, B.; Liu, F.; Wang, X.; Chen, J.; Fang, L.; Chou, K.-C., Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences. *Nucleic acids research* **2015,** *43* (W1), W65-W71.

71. Todeschini, R.; Consonni, V., *Handbook of Molecular Descriptors*. Wiley-VCH: 2000.

72. Ruiz-Blanco, Y. B.; Agüero-Chapin, G.; García-Hernández, E.; Álvarez, O.; Antunes, A.; Green, J., Exploring general-purpose protein features for distinguishing enzymes and non-enzymes within the twilight zone. *BMC bioinformatics* **2017,** *18* (1), 349-349.

73. Shen, H.-B.; Chou, K.-C., PseAAC: A flexible web server for generating various kinds of protein pseudo amino acid composition. *Analytical Biochemistry* **2008,** *373* (2), 386-388.

74. Liu, B.; Wu, H.; Chou, K., Pse-in-One 2.0: An Improved Package of Web Servers for Generating Various Modes of Pseudo Components of DNA, RNA, and Protein Sequences. *Natural Science* **2017,** *9* (4), 67-91.

75. Rao, H. B.; Zhu, F.; Yang, G. B.; Li, Z. R.; Chen, Y. Z., Update of PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. *Nucleic Acids Research* **2011,** *39* (suppl_2), W385-W390.

76. Ruiz-Blanco, Y. B.; Paz, W.; Green, J.; Marrero-Ponce, Y., ProtDCal: A program to compute general-purpose-numerical descriptors for sequences and 3D-structures of proteins. *BMC Bioinformatics* **2015,** *16* (1), 162.

77. Dong, J.; Yao, Z.-J.; Zhang, L.; Luo, F.; Lin, Q.; Lu, A.-P.; Chen, A. F.; Cao, D.-S., PyBioMed: a python library for various molecular representations of chemicals, proteins and DNAs and their interactions. *Journal of Cheminformatics* **2018,** *10* (1), 16.

78. Moriwaki, H.; Tian, Y.-S.; Kawashita, N.; Takagi, T., Mordred: a molecular descriptor calculator. *Journal of Cheminformatics* **2018,** *10* (1), 4.

79. Dong, J.; Zhu, M.-F.; Yun, Y.-H.; Lu, A.-P.; Hou, T.-J.; Cao, D.-S., BioMedR: an R/CRAN package for integrated data analysis pipeline in biomedical study. *Briefings in Bioinformatics* **2021,** *22* (1), 474-484.

80. Sukumar, N.; Breneman, C. M., *QTAIM in Drug Discovery and Protein Modeling*. 2007; p 471-498.

81. Kleandrova, V. V.; Ruso, J. M.; Speck-Planche, A.; Dias Soeiro Cordeiro, M. N., Enabling the Discovery and Virtual Screening of Potent and Safe Antimicrobial Peptides. Simultaneous Prediction of Antibacterial Activity and Cytotoxicity. *ACS Combinatorial Science* **2016,** *18* (8), 490-498.

82. Youmans, M.; Spainhour, C.; Qiu, P. In *Long short-term memory recurrent neural networks for antibacterial peptide identification*, 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 13-16 Nov. 2017; pp 498-502.

83. Kleandrova, V. V.; Rojas-Vargas, J. A.; Scotti, M. T.; Speck-Planche, A., PTML modeling for peptide discovery: in silico design of non-hemolytic peptides with antihypertensive activity. *Molecular Diversity* **2022,** *26* (5), 2523-2534.

84. Kizhedath, A.; Karlberg, M.; Glassey, J., Cross-Interaction Chromatography-Based QSAR Model for Early-Stage Screening to Facilitate Enhanced Developability of Monoclonal Antibody Therapeutics. *Biotechnology Journal* **2019,** *14* (8), 1800696.

85. Karlberg, M.; de Souza, J. V.; Fan, L.; Kizhedath, A.; Bronowska, A. K.; Glassey, J., QSAR Implementation for HIC Retention Time Prediction of mAbs Using Fab Structure: A Comparison between Structural Representations. *International Journal of Molecular Sciences* **2020,** *21* (21).

86. Ruiz-Blanco, Y. B.; Marrero-Ponce, Y.; García-Hernández, E.; Green, J., Novel "extended sequons" of human N-glycosylation sites improve the precision of qualitative predictions: an alignment-free study of pattern recognition using ProtDCal protein features. *Amino Acids* **2017,** *49* (2), 317-325.

87. Biggar, K. K.; Charih, F.; Liu, H.; Ruiz-Blanco, Y. B.; Stalker, L.; Chopra, A.; Connolly, J.; Adhikary, H.; Frensemier, K.; Hoekstra, M.; Galka, M.; Fang, Q.; Wynder, C.; Stanford, W. L.; Green, J. R.; Li, S. S. C., Proteome-wide Prediction of Lysine Methylation Leads to Identification of H2BK43 Methylation and Outlines the Potential Methyllysine Proteome. *Cell Reports* **2020,** *32* (2), 107896.

88. Yang, Y.; Urolagin, S.; Niroula, A.; Ding, X.; Shen, B.; Vihinen, M., PON-tstab: Protein Variant Stability Predictor. Importance of Training Data Quality. *International Journal of Molecular Sciences* **2018,** *19* (4), 1009.

89. Yang, Y.; Ding, X.; Zhu, G.; Niroula, A.; Lv, Q.; Vihinen, M., ProTstab – predictor for cellular protein stability. *BMC Genomics* **2019,** *20* (1), 804.

90. Corral-Corral, R.; Beltrán, J. A.; Brizuela, C. A.; Del Rio, G., Systematic Identification of Machine-Learning Models Aimed to Classify Critical Residues for Protein Function from Protein Structure. *Molecules* **2017,** *22* (10), 1673.

91. Mou, Z.; Eakes, J.; Cooper, C. J.; Foster, C. M.; Standaert, R. F.; Podar, M.; Doktycz, M. J.; Parks, J. M., Machine learning-based prediction of enzyme substrate scope: Application to bacterial nitrilases. *Proteins: Structure, Function, and Bioinformatics* **2021,** *89* (3), 336-347.

92. Staszak, M.; Staszak, K.; Wieszczycka, K.; Bajek, A.; Roszkowski, K.; Tylkowski, B., Machine learning in drug design: Use of artificial intelligence to explore the chemical structure–biological activity relationship. *WIREs Computational Molecular Science* **2022,** *12* (2), e1568.

93. Wang, H.; Bah, M. J.; Hammad, M., Progress in Outlier Detection Techniques: A Survey. *IEEE Access* **2019,** *7*, 107964-108000.

94. Devijver, P. A.; Kittler, J., *Pattern recognition: a statistical approach*. Prentice-Hall London, 1982.

95. Bellman, R., *Adaptive control processes: a guided tour*. Princeton University Press.: Princeton, Nueva Jersey, 1961.

96. Hughes, G., On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory* **1968,** *14* (1), 55-63.

97. Kent, J. T., Information gain and a general measure of correlation. *Biometrika* **1983,** *70* (1), 163-173.

98. John, G. H.; Kohavi, R.; Pfleger, K., Irrelevant Features and the Subset Selection Problem. In *Machine Learning: Proceedings of the Eleventh International Conference*, Morgan Kaufmann: 1994; pp 121-129.

99. Kohavi, R.; John, G. H., Wrappers for feature subset selection. *Artificial Intelligence* **1997,** *97* (1), 273-324.

100. Carracedo-Reboredo, P.; Liñares-Blanco, J.; Rodríguez-Fernández, N.; Cedrón, F.; Novoa, F. J.; Carballal, A.; Maojo, V.; Pazos, A.; Fernandez-Lozano, C., A review on machine learning approaches and trends in drug discovery. *Computational and Structural Biotechnology Journal* **2021,** *19*, 4538-4558.

101. Cortes, C.; Vapnik, V., Support-vector networks. *Machine Learning* **1995,** *20* (3), 273-297.

102. Pino, R.; Mendoza, R.; Sambayan, R., Optical character recognition system for Baybayin scripts using support vector machine. *PeerJ Computer Science* **2021,** *7*, e360.

103. Breiman, L., Random Forests. *Machine Learning* **2001,** *45* (1), 5-32.

104. Feurer, M.; Hutter, F., Hyperparameter Optimization. In *Automatic Machine Learning: Methods, Systems, Challenges*, Springer: 2019; pp 3-38.

105. Olson, R. S.; Cava, W. L.; Mustahsan, Z.; Varik, A.; Moore, J. H., Data-driven advice for applying machine learning to bioinformatics problems. In *Biocomputing 2018*, World scientific: 2018; pp 192-203.

106. Hsu, C. W.; Chang, C. C.; Lin, C. J., A Practical Guide to Support Vector Classification. **2003**.

107. Leo, B., Heuristics of instability and stabilization in model selection. *The Annals of Statistics* **1996,** *24* (6), 2350-2383.

108. Chen, P.-H. C.; Liu, Y.; Peng, L., How to develop machine learning models for healthcare. *Nature Materials* **2019,** *18* (5), 410-414.

109. Shipe, M. E.; Deppen, S. A.; Farjah, F.; Grogan, E. L., Developing prediction models for clinical use using logistic regression: an overview. *Journal of Thoracic Disease* **2019,** *11* (Suppl 4), S574-S584.

110. Netzeva, T. I.; Worth, A. P.; Aldenberg, T.; Benigni, R.; Cronin, M. T. D.; Gramatica, P.; Jaworska, J. S.; Kahn, S.; Klopman, G.; Marchant, C. A.; Myatt, G.; Nikolova-Jeliazkova, N.; Patlewicz, G. Y.; Perkins, R.; Roberts, D. W.; Schultz, T. W.; Stanton, D. T.; van de Sandt, J. J. M.; Tong, W.; Veith, G.; Yang, C., Current Status of Methods for Defining the Applicability Domain of (Quantitative) Structure-Activity Relationships: The Report and Recommendations of ECVAM Workshop 521,2. *Alternatives to Laboratory Animals* **2005,** *33* (2), 155-173.

111. Muratov, E. N.; Bajorath, J.; Sheridan, R. P.; Tetko, I. V.; Filimonov, D.; Poroikov, V.; Oprea, T. I.; Baskin, I. I.; Varnek, A.; Roitberg, A.; Isayev, O.; Curtalolo, S.; Fourches, D.; Cohen, Y.; Aspuru-Guzik, A.; Winkler, D. A.; Agrafiotis, D.; Cherkasov, A.; Tropsha, A., QSAR without borders. *Chemical Society Reviews* **2020,** *49* (11), 3525-3564.

112. OECD, *Guidance Document on the Validation of (Quantitative) Structure-Activity Relationship [(Q)SAR] Models*. 2014.

113. Göller, A. H.; Kuhnke, L.; ter Laak, A.; Meier, K.; Hillisch, A., Machine Learning Applied to the Modeling of Pharmacological and ADMET Endpoints. In *Artificial Intelligence in Drug Design*, Heifetz, A., Ed. Springer US: New York, NY, 2022; pp 61-101.

114. Dhakal, A.; McKay, C.; Tanner, J. J.; Cheng, J., Artificial intelligence in the prediction of protein–ligand interactions: recent advances and future directions. *Briefings in Bioinformatics* **2022,** *23* (1), bbab476.

115. Pitre, S.; Dehne, F.; Chan, A.; Cheetham, J.; Duong, A.; Emili, A.; Gebbia, M.; Greenblatt, J.; Jessulat, M.; Krogan, N.; Luo, X.; Golshani, A., PIPE: a protein-protein interaction prediction engine based on the re-occurring short polypeptide sequences between known interacting protein pairs. *BMC Bioinformatics* **2006,** *7* (1), 365.

116. Liu, X.; Liu, B.; Huang, Z.; Shi, T.; Chen, Y.; Zhang, J., SPPS: A Sequence-Based Method for Predicting Probability of Protein-Protein Interaction Partners. *PLOS ONE* **2012,** *7* (1), e30938.

117. Guo, Y.; Yu, L.; Wen, Z.; Li, M., Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences. *Nucleic Acids Research* **2008,** *36* (9), 3025-3030.

118. Vangone, A.; Bonvin, A. M., Contacts-based prediction of binding affinity in protein-protein complexes. *eLife* **2015,** *4*, e07454-e07454.

119. Ballester, P. J.; Mitchell, J. B. O., A machine learning approach to predicting protein-ligand binding affinity with applications to molecular docking. *Bioinformatics (Oxford, England)* **2010,** *26* (9), 1169-1175.

120. Jiménez, J.; Škalič, M.; Martínez-Rosell, G.; De Fabritiis, G., KDEEP: Protein–Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks. *Journal of Chemical Information and Modeling* **2018,** *58* (2), 287-296.

121. Abbasi, W. A.; Yaseen, A.; Hassan, F. U.; Andleeb, S.; Minhas, F. U. A. A., ISLAND: in-silico proteins binding affinity prediction using sequence information. *BioData Mining* **2020,** *13* (1), 20.

122. Ravikant, D. V. S.; Elber, R., PIE-efficient filters and coarse grained potentials for unbound protein-protein docking. *Proteins* **2010,** *78* (2), 400-419.

123. Chakraborty, A.; Mitra, S.; De, D.; Pal, A. J.; Ghaemi, F.; Ahmadian, A.; Ferrara, M., Determining Protein–Protein Interaction Using Support Vector Machine: A Review. *IEEE Access* **2021,** *9*, 12473-12490.

124. Priya, S.; Tripathi, G.; Singh, D. B.; Jain, P.; Kumar, A., Machine learning approaches and their applications in drug discovery and design. *Chemical Biology & Drug Design* **2022,** *100* (1), 136-153.

125. Wang, J.; Dou, X.; Song, J.; Lyu, Y.; Zhu, X.; Xu, L.; Li, W.; Shan, A., Antimicrobial peptides: Promising alternatives in the post feeding antibiotic era. *Medicinal Research Reviews* **2019,** *39* (3), 831-859.

126. Alghamdi, S., The role of vaccines in combating antimicrobial resistance (AMR) bacteria. *Saudi Journal of Biological Sciences* **2021,** *28* (12), 7505-7510.

127. Hu, D.; Zou, L.; Gao, Y.; Jin, Q.; Ji, J., Emerging nanobiomaterials against bacterial infections in postantibiotic era. *VIEW* **2020,** *1* (3), 20200014.

128. Serafim, M. S. M.; Kronenberger, T.; Oliveira, P. R.; Poso, A.; Honório, K. M.; Mota, B. E. F.; Maltarollo, V. G., The application of machine learning techniques to innovative antibacterial discovery and development. *Expert Opinion on Drug Discovery* **2020,** *15* (10), 1165-1180.

129. Viira, B.; Gendron, T.; Lanfranchi, D. A.; Cojean, S.; Horvath, D.; Marcou, G.; Varnek, A.; Maes, L.; Maran, U.; Loiseau, P. M.; Davioud-Charvet, E., In Silico Mining for Antimalarial Structure-Activity Knowledge and Discovery of Novel Antimalarial Curcuminoids. *Molecules* **2016,** *21* (7), 853.

130. Pang, Y.; Yao, L.; Jhong, J.-H.; Wang, Z.; Lee, T.-Y., AVPIden: a new scheme for identification and functional prediction of antiviral peptides based on machine learning approaches. *Briefings in Bioinformatics* **2021,** *22* (6), bbab263.

131. Xing, J.; Zhang, R.; Jiang, X.; Hu, T.; Wang, X.; Qiao, G.; Wang, J.; Yang, F.; Luo, X.; Chen, K.; Shen, J.; Luo, C.; Jiang, H.; Zheng, M., Rational design of 5-((1H-imidazol-1-yl)methyl)quinolin-8-ol derivatives as novel bromodomain-containing protein 4 inhibitors. *European Journal of Medicinal Chemistry* **2019,** *163*, 281-294.

132. Zhang, H.; Liu, W.; Liu, Z.; Ju, Y.; Xu, M.; Zhang, Y.; Wu, X.; Gu, Q.; Wang, Z.; Xu, J., Discovery of indoleamine 2,3-dioxygenase inhibitors using machine learning based virtual screening. *Medicinal Chemistry Communications* **2018,** *9* (6), 937-945.

133. Fan, C.; Huang, Y., Identification of novel potential scaffold for class I HDACs inhibition: An in-silico protocol based on virtual screening, molecular dynamics, mathematical analysis and machine learning. *Biochemical and Biophysical Research Communications* **2017,** *491* (3), 800-806.

134. Yu, M.; Gu, Q.; Xu, J., Discovering new PI3Kα inhibitors with a strategy of combining ligand-based and structure-based virtual screening. *Journal of Computer-Aided Molecular Design* **2018,** *32* (2), 347-361.

135. Park, K., A review of computational drug repurposing. *Translational and clinical pharmacology* **2019,** *27* (2), 59-63.

136. Schuler, J.; Hudson, M. L.; Schwartz, D.; Samudrala, R., A Systematic Review of Computational Drug Discovery, Development, and Repurposing for Ebola Virus Disease Treatment. *Molecules* **2017,** *22* (10), 1777.

137. Dotolo, S.; Marabotti, A.; Facchiano, A.; Tagliaferri, R., A review on drug repurposing applicable to COVID-19. *Briefings in Bioinformatics* **2021,** *22* (2), 726-741.

138. Beck, B. R.; Shin, B.; Choi, Y.; Park, S.; Kang, K., Predicting commercially available antiviral drugs that may act on the novel coronavirus (SARS-CoV-2) through a drug-target interaction deep learning model. *Computational and Structural Biotechnology Journal* **2020,** *18*, 784-790.

139. Alimadadi, A.; Aryal, S.; Manandhar, I.; Munroe, P. B.; Joe, B.; Cheng, X., Artificial intelligence and machine learning to fight COVID-19. *Physiological Genomics* **2020,** *52* (4), 200-202.

140. Lo, Y.-C.; Rensi, S. E.; Torng, W.; Altman, R. B., Machine learning in chemoinformatics and drug discovery. *Drug Discovery Today* **2018,** *23* (8), 1538-1546.

141. Mouchlis, V. D.; Afantitis, A.; Serra, A.; Fratello, M.; Papadiamantis, A. G.; Aidinis, V.; Lynch, I.; Greco, D.; Melagraki, G., Advances in De Novo Drug Design: From Conventional to Machine Learning Methods. *International Journal of Molecular Sciences* **2021,** *22* (4), 1676.

142. Basith, S.; Manavalan, B.; Hwan Shin, T.; Lee, G., Machine intelligence in peptide therapeutics: A next-generation tool for rapid disease screening. *Medicinal Research Reviews* **2020,** *40* (4), 1276-1314.

143. Tyagi, A.; Kapoor, P.; Kumar, R.; Chaudhary, K.; Gautam, A.; Raghava, G. P. S., In Silico Models for Designing and Discovering Novel Anticancer Peptides. *Scientific Reports* **2013,** *3* (1), 2984.

144. Hajisharifi, Z.; Piryaiee, M.; Mohammad Beigi, M.; Behbahani, M.; Mohabatkar, H., Predicting anticancer peptides with Chou's pseudo amino acid composition and investigating their mutagenicity via Ames test. *Journal of Theoretical Biology* **2014,** *341*, 34-40.

145. Vijayakumar, S.; Ptv, L., ACPP: A Web Server for Prediction and Design of Anti-cancer Peptides. *International Journal of Peptide Research and Therapeutics* **2015,** *21* (1), 99-106.

146. Chen, W.; Ding, H.; Feng, P.; Lin, H.; Chou, K.-C., iACP: a sequence-based tool for identifying anticancer peptides. *Oncotarget* **2016,** *7* (13), 16895–16909.

147. Li, F.-M.; Wang, X.-Q., Identifying anticancer peptides by using improved hybrid compositions. *Scientific Reports* **2016,** *6* (1), 33910.

148. Khan, F.; Akbar, S.; Basit, A.; Khan, I.; Akhlaq, H., Identification of Anticancer Peptides Using Optimal Feature Space of Chou's Split Amino Acid Composition and Support Vector Machine. In *Proceedings of the 2017 4th International Conference on Biomedical and Bioinformatics Engineering*, Association for Computing Machinery: Seoul, Republic of Korea, 2017; pp 91–96.

149. Manavalan, B.; Basith, S.; Hwan Shin, T.; Choi, S.; Ok Kim, M.; Lee, G., MLACP: machine-learning-based prediction of anticancer peptides. *Oncotarget* **2017,** *8* (44), 77121–77136.

150. Xu, L.; Liang, G.; Wang, L.; Liao, C., A Novel Hybrid Sequence-Based Model for Identifying Anticancer Peptides. *Genes (Basel)* **2018,** *9* (3), 158.

151. Wei, L.; Zhou, C.; Chen, H.; Song, J.; Su, R., ACPred-FL: a sequence-based predictor using effective feature representation to improve the prediction of anti-cancer peptides. *Bioinformatics* **2018,** *34* (23), 4007-4016.

152. Boopathi, V.; Subramaniyam, S.; Malik, A.; Lee, G.; Manavalan, B.; Yang, D.-C., mACPpred: A Support Vector Machine-Based Meta-Predictor for Identification of Anticancer Peptides. *International Journal of Molecular Sciences* **2019,** *20* (8), 1964.

153. Schaduangrat, N.; Nantasenamat, C.; Prachayasittikul, V.; Shoombuatong, W., ACPred: A Computational Tool for the Prediction and Analysis of Anticancer Peptides. *Molecules* **2019,** *24* (10), 1973.

154. Yi, H.-C.; You, Z.-H.; Zhou, X.; Cheng, L.; Li, X.; Jiang, T.-H.; Chen, Z.-H., ACP-DL: A Deep Learning Long Short-Term Memory Model to Predict Anticancer Peptides Using High-Efficiency Feature Representation. *Molecular Therapy - Nucleic Acids* **2019,** *17*, 1-9.

155. Wei, L.; Zhou, C.; Su, R.; Zou, Q., PEPred-Suite: improved and robust prediction of therapeutic peptides using adaptive feature representation learning. *Bioinformatics* **2019,** *35* (21), 4272-4280.

156. Wu, C.; Gao, R.; Zhang, Y.; De Marinis, Y., PTPD: predicting therapeutic peptides by deep learning and word2vec. *BMC Bioinformatics* **2019,** *20* (1), 456.

157. Rao, B.; Zhou, C.; Zhang, G.; Su, R.; Wei, L., ACPred-Fuse: fusing multi-view information improves the prediction of anticancer peptides. *Briefings in Bioinformatics* **2020,** *21* (5), 1846-1855.

158. Kumar, R.; Chaudhary, K.; Singh Chauhan, J.; Nagpal, G.; Kumar, R.; Sharma, M.; Raghava, G. P. S., An in silico platform for predicting, screening and designing of antihypertensive peptides. *Scientific Reports* **2015,** *5* (1), 12512.

159. Win, T. S.; Schaduangrat, N.; Prachayasittikul, V.; Nantasenamat, C.; Shoombuatong, W., PAAP: a web server for predicting antihypertensive activity of peptides. *Future Medicinal Chemistry* **2018,** *10* (15), 1749-1767.

160. Manavalan, B.; Basith, S.; Shin, T. H.; Wei, L.; Lee, G., mAHTPred: a sequence-based meta-predictor for improving the prediction of anti-hypertensive peptides using effective feature representation. *Bioinformatics* **2019,** *35* (16), 2757-2765.

161. Usmani, S. S.; Bhalla, S.; Raghava, G. P. S., Prediction of Antitubercular Peptides From Sequence Information Using Ensemble Classifier and Hybrid Features. *Frontiers in Pharmacology* **2018,** *9*, 954.

162. Manavalan, B.; Basith, S.; Shin, T. H.; Wei, L.; Lee, G., AtbPpred: A Robust Sequence-Based Prediction of Anti-Tubercular Peptides Using Extremely Randomized Trees. *Computational and Structural Biotechnology Journal* **2019,** *17*, 972-981.

163. Khatun, S.; Hasan, M.; Kurata, H., Efficient computational model for identification of antitubercular peptides by integrating amino acid patterns and properties. *FEBS Letters* **2019,** *593* (21), 3029-3039.

164. Chen, W.; Feng, P.; Nie, F., iATP: A Sequence Based Method for Identifying Anti-tubercular Peptides. *Medicinal Chemistry* **2020,** *16* (5), 620-625.

165. Gupta, S.; Sharma, A. K.; Shastri, V.; Madhu, M. K.; Sharma, V. K., Prediction of anti-inflammatory proteins/peptides: an insilico approach. *Journal of Translational Medicine* **2017,** *15* (1), 7.

166. Manavalan, B.; Shin, T. H.; Kim, M. O.; Lee, G., AIPpred: Sequence-Based Prediction of Anti-inflammatory Peptides Using Random Forest. *Frontiers in Pharmacology* **2018,** *9*, 276.

167. Khatun, M. S.; Hasan, M. M.; Kurata, H., PreAIP: Computational Prediction of Anti-inflammatory Peptides by Integrating Multiple Complementary Features. *Frontiers in Genetics* **2019,** *10*, 129.

168. Rajput, A.; Gupta, A. K.; Kumar, M., Prediction and Analysis of Quorum Sensing Peptides Based on Sequence Features. *PLOS ONE* **2015,** *10* (3), e0120066.

169. Wei, L.; Hu, J.; Li, F.; Song, J.; Su, R.; Zou, Q., Comparative analysis and prediction of quorum-sensing peptides using feature representation learning and machine learning algorithms. *Briefings in Bioinformatics* **2020,** *21* (1), 106-119.

170. Holton, T. A.; Pollastri, G.; Shields, D. C.; Mooney, C., CPPpred: prediction of cell penetrating peptides. *Bioinformatics* **2013,** *29* (23), 3094-3096.

171. Gautam, A.; Chaudhary, K.; Kumar, R.; Sharma, A.; Kapoor, P.; Tyagi, A.; Raghava, G. P. S.; Open source drug discovery consortium, In silico approaches for designing highly effective cell penetrating peptides. *Journal of Translational Medicine* **2013,** *11* (1), 74.

172. Sanders, W. S.; Johnston, C. I.; Bridges, S. M.; Burgess, S. C.; Willeford, K. O., Prediction of Cell Penetrating Peptides by Support Vector Machines. *PLOS Computational Biology* **2011,** *7* (7), e1002101.

173. Dobchev, D. A.; Mager, I.; Tulp, I.; Karelson, G.; Tamm, T.; Tamm, K.; Janes, J.; Langel, U.; Karelson, M., Prediction of Cell-Penetrating Peptides Using Artificial Neural Networks. *Current Computer-Aided Drug Design* **2010,** *6* (2), 79-89.

174. Tang, H.; Su, Z.-D.; Wei, H.-H.; Chen, W.; Lin, H., Prediction of cell-penetrating peptides with feature selection techniques. *Biochemical and Biophysical Research Communications* **2016,** *477* (1), 150-154.

175. Wei, L.; Tang, J.; Zou, Q., SkipCPP-Pred: an improved and promising sequence-based predictor for predicting cell-penetrating peptides. *BMC Genomics* **2017,** *18* (7), 742.

176. Wei, L.; Xing, P.; Su, R.; Shi, G.; Ma, Z. S.; Zou, Q., CPPred-RF: A Sequence-based Predictor for Identifying Cell-Penetrating Peptides and Their Uptake Efficiency. *Journal of Proteome Research* **2017,** *16* (5), 2044-2053.

177. Manavalan, B.; Subramaniyam, S.; Shin, T. H.; Kim, M. O.; Lee, G., Machine-Learning-Based Prediction of Cell-Penetrating Peptides and Their Uptake Efficiency with Improved Accuracy. *Journal of Proteome Research* **2018,** *17* (8), 2715-2726.

178. Pandey, P.; Patel, V.; George, N. V.; Mallajosyula, S. S., KELM-CPPpred: Kernel Extreme Learning Machine Based Prediction Model for Cell-Penetrating Peptides. *Journal of Proteome Research* **2018,** *17* (9), 3214-3222.

179. Qiang, X.; Zhou, C.; Ye, X.; Du, P.-f.; Su, R.; Wei, L., CPPred-FL: a sequence-based predictor for large-scale identification of cell-penetrating peptides by feature representation learning. *Briefings in Bioinformatics* **2020,** *21* (1), 11-23.

180. Wang, S. F.; Cao, Z. C.; Li, M. Y.; Yue, Y. T., G-DipC: An Improved Feature Representation Method for Short Sequences to Predict the Type of Cargo in Cell-Penetrating Peptides. *IEEE/ACM transactions on computational biology and bioinformatics* **2020,** *17* (3), 739-747.

181. Lee, H.-T.; Lee, C.-C.; Yang, J.-R.; Lai, J. Z. C.; Chang, K. Y., A Large-Scale Structural Classification of Antimicrobial Peptides. *BioMed Research International* **2015,** *2015*, 475062.

182. Bhadra, P.; Yan, J.; Li, J.; Fong, S.; Siu, S. W. I., AmPEP: Sequence-based prediction of antimicrobial peptides using distribution patterns of amino acid properties and random forest. *Scientific Reports* **2018,** *8* (1), 1697.

183. Yan, J.; Bhadra, P.; Li, A.; Sethiya, P.; Qin, L.; Tai, H. K.; Wong, K. H.; Siu, S. W. I., Deep-AmPEP30: Improve Short Antimicrobial Peptides Prediction with Deep Learning. *Molecular Therapy - Nucleic Acids* **2020,** *20*, 882-894.

184. Lata, S.; Mishra, N. K.; Raghava, G. P. S., AntiBP2: improved version of antibacterial peptide prediction. *BMC Bioinformatics* **2010,** *11* (1), S19.

185. Veltri, D. P. A Computational and Statistical Framework for Screening Novel Antimicrobial Peptides. George Mason University, 2015.

186. Veltri, D.; Kamath, U.; Shehu, A., Deep learning improves antimicrobial peptide recognition. *Bioinformatics* **2018,** *34* (16), 2740-2747.

187. Schaduangrat, N.; Nantasenamat, C.; Prachayasittikul, V.; Shoombuatong, W., Meta-iAVP: A Sequence-Based Meta-Predictor for Improving the Prediction of Antiviral Peptides Using Effective Feature Representation. *International Journal of Molecular Sciences* **2019,** *20* (22), 5743.

188. Agrawal, P.; Bhalla, S.; Chaudhary, K.; Kumar, R.; Sharma, M.; Raghava, G. P. S., In Silico Approach for Prediction of Antifungal Peptides. *Frontiers in Microbiology* **2018,** *9*, 323.

189. Joseph, S.; Karnik, S.; Nilawe, P.; Jayaraman, V. K.; Idicula-Thomas, S., ClassAMP: A Prediction Tool for Classification of Antimicrobial Peptides. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **2012,** *9* (5), 1535-1538.

190. Lin, W.; Xu, D., Imbalanced multi-label learning for identifying antimicrobial peptides and their functional types. *Bioinformatics* **2016,** *32* (24), 3745-3752.

191. Meher, P. K.; Sahu, T. K.; Saini, V.; Rao, A. R., Predicting antimicrobial peptides with improved accuracy by incorporating the compositional, physico-chemical and structural features into Chou's general PseAAC. *Scientific Reports* **2017,** *7* (1), 42362.

192. Gull, S.; Shamim, N.; Minhas, F., AMAP: Hierarchical multi-label prediction of biologically active and antimicrobial peptides. *Computers in Biology and Medicine* **2019,** *107*, 172-181.

193. Pinacho-Castellanos, S. A.; García-Jacas, C. R.; Gilson, M. K.; Brizuela, C. A., Alignment-Free Antimicrobial Peptide Predictors: Improving Performance by a Thorough Analysis of the Largest Available Data Set. *Journal of Chemical Information and Modeling* **2021,** *61* (6), 3141-3157.

194. Xiao, X.; Wang, P.; Lin, W.-Z.; Jia, J.-H.; Chou, K.-C., iAMP-2L: A two-level multi-label classifier for identifying antimicrobial peptides and their functional types. *Analytical Biochemistry* **2013,** *436* (2), 168-177.

195. Chung, C.-R.; Kuo, T.-R.; Wu, L.-C.; Lee, T.-Y.; Horng, J.-T., Characterization and identification of antimicrobial peptides with different functional activities. *Briefings in Bioinformatics* **2020,** *21* (3), 1098-1114.

196. Su, X.; Xu, J.; Yin, Y.; Quan, X.; Zhang, H., Antimicrobial peptide identification using multi-scale convolutional network. *BMC Bioinformatics* **2019,** *20* (1), 730.

197. Ettayapuram Ramaprasad, A. S.; Singh, S.; Gajendra P. S, R.; Venkatesan, S., AntiAngioPred: A Server for Prediction of Anti-Angiogenic Peptides. *PLOS ONE* **2015,** *10* (9), e0136990.

198. Blanco, J. L.; Porto-Pazos, A. B.; Pazos, A.; Fernandez-Lozano, C., Prediction of high anti-angiogenic activity peptides in silico using a generalized linear model and feature selection. *Scientific Reports* **2018,** *8* (1), 15688.

199. Lin, C.; Wang, L.; Shi, L., AAPred-CNN: Accurate predictor based on deep convolution neural network for identification of anti-angiogenic peptides. *Methods* **2022,** *204*, 442-448.

200. Laengsri, V.; Nantasenamat, C.; Schaduangrat, N.; Nuchnoi, P.; Prachayasittikul, V.; Shoombuatong, W., TargetAntiAngio: A Sequence-Based Tool for the Prediction and Analysis of Anti-Angiogenic Peptides. *International Journal of Molecular Sciences* **2019,** *20* (12), 2950.

201. Nagpal, G.; Usmani, S. S.; Dhanda, S. K.; Kaur, H.; Singh, S.; Sharma, M.; Raghava, G. P. S., Computer-aided designing of immunosuppressive peptides based on IL-10 inducing potential. *Scientific Reports* **2017,** *7* (1), 42851.

202. Singh, O.; Hsu, W.-L.; Su, E. C., ILeukin10Pred: A Computational Approach for Predicting IL-10-Inducing Immunosuppressive Peptides Using Combinations of Amino Acid Global Features. *Biology* **2022,** *11* (1), 5.

203. Poorinmohammad, N.; Mohabatkar, H.; Behbahani, M.; Biria, D., Computational prediction of anti HIV-1 peptides and in vitro evaluation of anti HIV-1 activity of HIV-1 P24-derived peptides. *Journal of Peptide Science* **2015,** *21* (1), 10-16.

204. Mathiyazhagan, B.; Liyaskar, J.; Azar, A. T.; Inbarani, H. H.; Javed, Y.; Kamal, N. A.; Fouad, K. M., Rough Set Based Classification and Feature Selection Using Improved Harmony Search for Peptide Analysis and Prediction of Anti-HIV-1 Activities. *Applied Sciences* **2022,** *12* (4), 2020.

205. Poorinmohammad, N.; Mohabatkar, H., A Comparison of Different Machine Learning Algorithms for the Prediction of Anti-HIV-1 Peptides Based on Their Sequence-Related Properties. *International Journal of Peptide Research and Therapeutics* **2015,** *21* (1), 57-62.

206. Kilburg, D.; Gallicchio, E., Chapter Two - Recent Advances in Computational Models for the Study of Protein–Peptide Interactions. In *Advances in Protein Chemistry and Structural Biology*, Christov, C. Z., Ed. Academic Press: 2016; Vol. 105, pp 27-57.

207. Hu, J.; Liu, Z., DeepMHC: Deep Convolutional Neural Networks for High-performance peptide-MHC Binding Affinity Prediction. *bioRxiv* **2017**, 239236.

208. Jurtz, V.; Paul, S.; Andreatta, M.; Marcatili, P.; Peters, B.; Nielsen, M., NetMHCpan-4.0: Improved Peptide-MHC Class I Interaction Predictions Integrating Eluted Ligand and Peptide Binding Affinity Data. *Journal of immunology (Baltimore, Md. : 1950)* **2017,** *199* (9), 3360-3368.

209. Park, Y., Critical assessment of sequence-based protein-protein interaction prediction methods that do not require homologous protein sequences. *BMC Bioinformatics* **2009,** *10* (1), 419.

210. Xue, L. C.; Rodrigues, J. P.; Kastritis, P. L.; Bonvin, A. M.; Vangone, A., PRODIGY: a web server for predicting the binding affinity of protein–protein complexes. *Bioinformatics* **2016,** *32* (23), 3676-3678.

211. Liu, S.; Zhang, C.; Zhou, H.; Zhou, Y., A physical reference state unifies the structure-derived potential of mean force for protein folding and binding. *Proteins: Structure, Function, and Bioinformatics* **2004,** *56* (1), 93-101.

212. Romero-Molina, S.; Ruiz-Blanco, Y. B.; Mieres-Perez, J.; Harms, M.; Münch, J.; Ehrmann, M.; Sanchez-Garcia, E., PPI-Affinity: A Web Tool for the Prediction and Optimization of Protein–Peptide and Protein–Protein Binding Affinity. *Journal of Proteome Research* **2022,** *21* (8), 1829–1841.

213. Romero-Molina, S.; Ruiz-Blanco, Y. B.; Harms, M.; Münch, J.; Sanchez-Garcia, E., PPI-Detect: A support vector machine model for sequence-based prediction of protein–protein interactions. *Journal of Computational Chemistry* **2019,** *40* (11), 1233-1242.

214. Romero-Molina, S.; Ruiz-Blanco, Y. B.; Green, J. R.; Sanchez-Garcia, E., ProtDCal-Suite: A web server for the numerical codification and functional analysis of proteins. *Protein Science* **2019,** *28* (9), 1734-1743.

215. Speck-Planche, A.; Kleandrova, V. V.; Ruso, J. M.; D. S. Cordeiro, M. N., First Multitarget Chemo-Bioinformatic Model To Enable the Discovery of Antibacterial Peptides against Multiple Gram-Positive Pathogens. *Journal of Chemical Information and Modeling* **2016,** *56* (3), 588-598.

216. Li, G.; Zrimec, J.; Ji, B.; Geng, J.; Larsbrink, J.; Zelezniak, A.; Nielsen, J.; Engqvist, M. K. M., Performance of Regression Models as a Function of Experiment Noise. *Bioinformatics and Biology Insights* **2021,** *15*, 11779322211020315.

217. Abbasi, W. A.; Asif, A.; Ben-Hur, A.; Minhas, F. u. A. A., Learning protein binding affinity using privileged information. *BMC Bioinformatics* **2018,** *19* (1), 425.

218. Huang, F.; Xie, G.; Xiao, R. In *Research on Ensemble Learning*, 2009 International Conference on Artificial Intelligence and Computational Intelligence, 2009; pp 249-252.

219. Yang, J.; Roy, A.; Zhang, Y., BioLiP: a semi-manually curated database for biologically relevant ligand-protein interactions. *Nucleic Acids Research* **2013,** *41* (Database issue), D1096-103.

220. Ruiz-Blanco, Y. B.; Agüero-Chapin, G.; Romero-Molina, S.; Antunes, A.; Olari, L.-R.; Spellerberg, B.; Münch, J.; Sanchez-Garcia, E., ABP-Finder: A Tool to Identify Antibacterial Peptides and the Gram-Staining Type of Targeted Bacteria. *Antibiotics* **2022,** *11* (12), 1708.

221. Ruiz-Blanco, Y. B.; Ávila-Barrientos, L. P.; Hernández-García, E.; Antunes, A.; Agüero-Chapin, G.; García-Hernández, E., Engineering protein fragments via evolutionary and protein–protein interaction algorithms: de novo design of peptide inhibitors for FOF1-ATP synthase. *FEBS Letters* **2021,** *595* (2), 183-194.

222. Stoye, J.; Evers, D.; Meyer, F., Rose: generating sequence families. *Bioinformatics* **1998,** *14* (2), 157-163.

9 Appendix

9.1 List of Abbreviations

A	AA	amino acids
	ABPs	antibacterial peptides
	Acc	accuracy
	ACk	autocorrelation operator of order k
	AIDS	acquired immunodeficiency syndrome
	AD	applicability domain
	ADME	absorption, distribution, metabolism, excretion
	Ar	arithmetic mean
	AMPs	antimicrobial peptides
	AT	activity test
	AUC	area under the ROC curve
B	BA	binding affinity
	BS	broad-spectrum
C	C	cost, complexity of the polynomial kernel
	CADD	computer-aided drug design
	CV	cross validation
	CXCR4	CXC chemokine receptor 4
D	D	degree of the polynomial kernel
	Dev	development set
E	EF	enrichment factor
	EPI-X4	endogenous peptide inhibitor of CXCR4
	ERM	empirical risk minimization
	E-State	electro-topological state
F	F1	F1-Score
	FN	false negatives
	FD	folding degree
	FP	false positives
	FRS	fitness-robustness score
G	Gram-	gram negative staining type
	Gram+	gram positive staining type
	GLY	glycine amino acids
H	HSP	High scoring pairs

	HIV-1	human immunodeficiency virus type I
	HtrAs	high-temperature requirement serine proteases
	HTS	high-throughput screening
I	I	topological distance
	IC50	half-maximal inhibitory concentration
	ICs	network of inter-residue contacts
	ID	identification code
	IDL	individual descriptor labels
	IG	information gain
K	Kd	dissociation constant
	Ki	inhibition constant
L	lnFD	Logarithm of the folding degree
M	MAE	mean absolute error
	MCC	mathew correlation coefficient
	MHC	major histocompatibility complexes
	ML	machine learning
	Mw	molar weights
N	NIG	normalized information gain
	NIS	non-interacting surface
	NPR	nonpolar residues
O	OECD	organization for economic co-operation and development
P	PCA	principal component analysis
	PCPr	prevalence-corrected precision
	PCR	positively charged residues
	PDB	protein data bank
	PPI	protein-protein interactions
	PPIs	protein-protein interactions
	Pr	precision
	PRC	precision-recall curves
	PseACC	pseudo amino acid composition features
Q	QSAR	quantitative structure-activity relationship
R	R	pearson’s correlation coefficient
	RF	random forest
	R_s	spearman's rank correlation coefficient

	ROC	receiver operating characteristic
	RWCO	reside-wise contact order
S	SDF-1 or CXCL1	stromal-cell-derived factor-1
	SL	supervised learning
	Sn	sensitivity or recall
	sSE	standardized Shannon entropy
	St	sensitivity or recall
	Sp	specificity
	SVM	support vector machines
	SVR	support vector regression
	SWOT	strengths, weaknesses, opportunities, and threats
T	T	toxicity
	Tau	Kendall’s tau correlation coefficient
	TN	true negatives
	TP	true positives
U	UL	unsupervised learning
V	V	variance
	VS	virtual screening

9.2 List of Figures

Figure 1. Stages of the drug design process[1] 12
Figure 2. Mechanisms of action (orthosteric or allosteric) of modulators leading to the inhibition or stabilization of PPIs (from Modell, A. E. et al.[14])[Note] 13
Figure 3. Analysis of the strengths, weaknesses, opportunities, and threats (SWOT) of peptides (from Fosgerau and Hoffmann[31])[Note] 16
Figure 4. The learning curve arising from bias-variance trade-off (adapted from Beyeler et al.[53])[Note] 19
Figure 5. Machine learning algorithms 30
Figure 6. Hyperplane with the maximum margin for a SVM trained on a data set with two classes (adapted from Pino et al.[102])[Note] 31
Figure 7. Internal and external validation. 35
Figure 8. Standard process to develop ML-based QSAR models 40
Figure 9. Worldwide usage of the tools developed within this work (stand November 26th, 2022) 216

9.3 List of Tables

Table 1. List of tools developed within this work. 11
Table 2. Confusion matrix for a binary classification problem. 35
Table 3. List of performance measures used in the classification problems addressed in this work. 36
Table 4. List of performance measures used in the regression problems addressed in this work. 37

9.4 Curriculum Vitae

“The curriculum vitae is not included in the online version for data protection reasons.”