Artificial Intelligence in the Life Sciences 2 (2022) 100044

Contents lists available at ScienceDirect

Artificial Intelligence in the Life Sciences

journal homepage: www.elsevier.com/locate/ailsci

Open protocols for docking and MD-based scoring of peptide substrates

Rodrigo Ochoa[a,*], Ángel Santiago[b], Melissa Alegría-Arcos[c]

[a] Biophysics of Tropical Diseases, Max Planck Tandem Group, University of Antioquia, Antioquia, Medellin 050010, Colombia
[b] Departamento de Fisicoquímica, Facultad de Química, Universidad Nacional Autónoma de México, Mexico 04510, Mexico
[c] Facultad de Ingeniería y Negocios, Universidad de las Américas, Sede Providencia, Manuel Montt, Santiago 948, Chile

ARTICLE INFO

Keywords:
Peptide
Docking
Molecular dynamics
Machine learning

ABSTRACT

The study of protein-peptide interactions is an active research field from an experimental and computational perspective, with the latest presenting challenges to model and simulate the peptides' intrinsic flexibility. Predicting affinities towards protein systems of interest, such as proteases, is crucial to understand the specificity of the interactions and support the discovery of novel substrates. Here we provide a set of computational protocols to run structural and dynamical analysis of protein-peptide complexes from a binding perspective. The protocols are based on state-of-the-art methods, but the code is open and can be customized depending on the user needs. These include a fragment-growing peptide docking protocol to predict bound conformations of flexible peptides, a protocol to extract descriptors from protein-peptide molecular dynamics trajectories, and a workflow to build and test machine learning regression models. As a toy example, we applied the protocols to a serine protease structure with a set of known peptide substrates and random sequences to illustrate the use of the code, which is publicly available at: https://github.com/rochoa85/Protocols-Peptide-Binding

1. Introduction

The spatial arrangement of peptide substrates is important to predict potential interactions with their molecular targets [1]. The research on molecular docking of peptides has contributed to overcome, in part, the challenge of predicting bound conformations of these highly flexible molecular entities [2,3]. In this sense, multiple approaches have been published to tackle this computational problem [4]. Among them, methods using docking strategies where the peptide is grown step-by-step can help to solve the flexibility issues, while maintaining the capability of predicting energy-favorable conformations associated with potential biological activities [5,6]. These complexes can be sampled using techniques such as molecular dynamics (MD), where amino acid force field parameters can be implemented to study peptides and their interactions [7,8]. Other tools can be useful to predict beforehand the probable structural conformation of the peptides in comparison to experimental techniques [9,10].

To assess the binding affinity of peptides, multiple strategies have been proposed to capture their molecular flexibility. These include enhanced sampling approaches [11], alchemical free energy perturbations [12], or techniques able to explore the potential surface energy of the system. Other methods are based on implicit solvent calculation using Molecular Mechanics Poisson-Boltzmann Surface Area (MM/PBSA) approaches [13], with the possibility of adding quantum calculation using semi-empirical theories for catalytic residues [14]. In any of these scenarios, large computational resources are required to reproduce the binding landscape of peptides as ligands. One option is to use scoring functions used in molecular docking to score representative frames of MD trajectories, and calculate thermal averages that can be correlated with binding affinities [15]. Another approximation is to use machine learning methods to predict, from MD trajectories and molecular descriptors, any response variable such as affinity proxies [16,17].

Nowadays, the combination of physics-based approaches with machine learning models is useful to reduce the computational cost of running exhaustive simulations and improve the prediction performance of classical methods with available curated data [18]. This is the case of novel methods to accelerate quantum chemical calculations using models trained with pre-calculated parameters [19]. In the context of MD, initiatives are reported to extract descriptors from the trajectories and combine them with additional chemical data from the molecular entities. This is the case of the Molecular Dynamics Fingerprints package (MDFP) [20], where a set of molecular fingerprints can be computed by obtaining average energy terms and observables from the MD simulations such as the solvent accessible surface area, dipole moments, radius of gyration and evolution of the hydrogen bonds. The latest have been applied to predict free energies of solvation for small molecules [21], and for binding studies involving proteins [22]. The protocol can be adapted to include peptides as ligands for virtual screening studies

* Corresponding author.
E-mail address: rodrigo.ochoa@udea.edu.co (R. Ochoa).

https://doi.org/10.1016/j.ailsci.2022.100044
Received 30 June 2022; Received in revised form 30 September 2022; Accepted 12 October 2022
Available online 20 October 2022
2667-3185/© 2022 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

given that a predictive model can be convenient to avoid running exhaustive simulations.

In this article, we present a set of open Python scripts to run locally different computational workflows inspired by published methods for peptide modeling, such as peptide docking using a fragment-growing docking protocol for the prediction of bound peptide conformations, and a routine to capture descriptors from protein-peptide MD trajectories in order to predict observables such as average scoring values. We provide the open source code to reproduce the main protocols with small differences and without requiring access to web servers. As a toy example, a library of known substrates and random peptide sequences were docked to a granzyme B protease, where each complex was subjected to short MD simulations. Then, a set of descriptors was calculated per complex to generate a regression model able to predict with enough accuracy binding observables. For this example, we show the performance of some models under certain conditions explained in the Methods, but such variables can be changed by the user if a protein target is selected with previous knowledge of its binding site, as well as a list of peptide substrates that can be used to extrapolate the results into a novel chemical space using machine learning approaches. Instructions to run examples and install third-party modules required to perform the analysis are provided in the repository.

2. Methods

2.1. Customization and system requirements

The Python 3 scripts are available in the public repository: https://github.com/rochoa85/Protocols-Peptide-Binding. The code is provided under a MIT license for academic and development purposes. The project is split into three folders. The first one allows running the fragment-growing docking protocol with a protease receptor as an example. The second provides scripts to extract features of protein-peptide MD trajectories with the MDFP package. In addition, the PepFun protocol is used to calculate descriptors from the peptide sequence and structure as explained later in Section 2.5 and Supplementary Table 1. Finally, the third protocol provides a script to configure and run a regression model to predict score/affinity values based on the obtained descriptors. An example with a protease and a set of peptides evaluated in this work is included in the code.

All the classes and functions were written under the Ubuntu 20.04 operating system. Additional third-party tools (see Supplementary Table 2) can be installed through the available source code using the latest versions, or through Conda virtual environments.

2.2. Fragment-growing docking approach

As an open source alternative to dock flexible peptides in protein binding sites, we provide a protocol based on the incremental growth of a core peptide fragment into the protein binding site, maintaining a degree of flexibility after new amino acids are added in the peptide flanking regions. The method is inspired in the DINC2 strategy [23].

The methodology involves the following steps: (i) Modeling of a 3-mer peptide fragment corresponding to the central section of the peptide of interest given as input. The loop reconstruction module of Modeller is used to generate this starter 3-mer fragment from the amino acid sequence using the Python modules, which can be easily incorporated into the docking protocol [24]. The protonation states of the protein and peptide 3D structures are generated using PDB2PQR [25]. (ii) The fragment is docked in the protein active site using Autodock Vina with standard parameters [26]. The site is defined based on previous knowledge of the protein, and it is delimited by a grid search box at the center of the binding site that increases its size after growing the peptide sequence. The initial fragment has all their rotational bonds active. (iii) From the initial docking, the three best poses (i.e. best Autodock Vina scores) are used as starting points to add amino acids at each flanking region of the peptide. In that way, three solutions are generated from three different initial docked conformers. To add the flanking amino acids, the rotamers are predicted with Modeller using the rest of the structure bound to the protein as a template. The bonds of each new amino acid and the amino acids next to them are kept flexible, while the central part is configured as rigid based on the docked pose of the previous step. (iv) Finally, the new fragments are docked and the best pose per each of the three parallel runs is selected to repeat the growing process, until the peptide reaches its final size. The methodology is intrinsically parallelized by Autodock Vina itself and the multiprocessing module of Python.

2.3. Use case with a protease-peptide system

A benchmark/validation docking run using our fragment-growing approach was performed on the granzyme B protease (PDB id 1iau) [27,28]. This is a serine protease from the subfamily S01.010, according to the MEROPS database [29]. Multiple peptide substrates have been reported against this enzyme. A total of 513 8-mer sequences were selected from the MEROPS database based on filtering experimentally non-redundant cleaved substrates. To include a set of external peptides for prediction purposes, 365 random sequences were generated with the PepFun package [30] and docked to the same protease structure. The sequences were generated using the same 8-mer length, with uniformly distributed amino acids per position, and without conserving dominant residues on the peptide cleavage sites.

To assess the performance of our fragment-growing docking protocol, a set of reported proteases structures bound to peptides with extended conformations were subjected to re-docking calculations of the peptides using the following servers: ClusPro [31], Haddock [32], HPEPDock [33], MDockPep [34], CABSDock [35] and DINC2 [23]. To evaluate the performance, we compared the predicted peptides with respect to crystal conformations through RMSD calculations over backbone atoms. The studied complexes have the PDB IDs 1ou8, 2xxn, 3qdz, 3tjv and 6di8, and all of them are co-crystallized with peptides from 8 to 15 amino acids in range. The full peptide input conformations for redocking at each server were generated with PEP-FOLD [36], except for CABSDock and MDockPep which both generate the peptide structure input. To complement the assessment, we included a set of 10 additional protein-peptide complexes available in the PDB and curated based on the LEADS-PEP dataset [37]. The peptides selected are also bound in extended conformations. The same docking servers were used, and the RMSD evaluation metrics used for the proteases were also calculated. In all cases, the binding site residues or binding site grid were specified, depending on web server requirements.

2.4. MD simulations

Each granzyme B-peptide complex docked with our protocol was subjected to MD simulations using Gromacs version 5.1.4 [38]. The Amber99SB-ILDN protein force-field [39], a TIP3P water model [40], a modified Berendsen thermostat [41], and a Parrinello-Rahman barostat [42] were used during the equilibration and production phases. The complex was solvated in a cubic box of water with periodic boundaries at a distance of at least 8 Å from any atom of the protein. Counterions of Na^+ and Cl^- were included in the solvent to make the box neutral. The electrostatic interactions were calculated using the Particle Mesh Ewald (PME) method, with 1.0 nm short-range electrostatic and van der Waals cutoffs [43]. The equations of motion were solved with the leap-frog integrator [44], using a timestep of 2 femtoseconds (fs).

The equilibration consisted of 100 picoseconds (ps) of NVT ensemble, followed by 100 ps of NPT. Then a production NPT simulation was run during 10 ns. With the trajectories, a MD/scoring approach was implemented to calculate average scores using all the frames from the trajectories. Specifically, for each MD simulation, an interaction score was attributed: each frame was scored using the Autodock Vina scoring function (the same used in the fragment-growing docking protocol)

and the average value across all frames from the trajectory was stored, and used at later stages, for the generation of a regression model able to predict the same variable for novel peptide sequences.

2.5. Descriptors extraction

Based on the granzyme B-peptide MD trajectories performed with Gromacs, the MDFP tools libraries [20] were adapted and used to extract a set of protein-peptide MD-derived descriptors (https://github.com/rinikerlab/mdfptools/). Each descriptor is split into three positions (vector), which include the average, median and standard deviation value of the calculated property among the MD frames. To achieve this, the calculated MD trajectory is re-run with Gromacs to add new energy terms in the outputs per frame. These include the Coulomb and Lennard-Jones energy contributions between the peptide, the receptor and the water molecules. Other included descriptors are the SASA and radius of gyration, the charges calculated with the ParmEd module [45], the dipole moments and evolution of hydrogen bonds with the MDtraj module [46], and bioinformatics properties of the peptide using the PepFun package. A total of 70 descriptors per complex were calculated. The full list is available in the Supplementary Table S1.

Two scripts, one using bash command lines, and a second written in Python are included in the main code to extract these descriptors. The vectors per complex are stored as pickled objects in a folder, which can be read later by machine learning models. An example of a protein-peptide MD trajectory is provided in the code repository to reproduce the results.

2.6. Machine learning model setup and test

Finally, to test the prediction performance of the average score obtained from the MD frames, two regression models were configured using the granzyme B system. The models are a basic linear regression model, and a gradient boosting regressor with 500 estimators and a learning rate of 0.01. Both models are prepared using the scikit-learn module in Python [47]. For the initial training validation, the set of 513 peptide substrates were split in a 75/25 schema (i.e. 75% of the peptides to train, and 25% to test). This was performed using a maximum chemical diversity function, where both the training and test set contains the maximum number of diverse compounds to avoid biasing both chemical spaces. The chemical similarity is quantified using the ECFP4 Tanimoto coefficient [48], and the diverse subset are selected using the MaxMin algorithm in the RDKit. The R2 coefficient of determination and the Mean Squared Error (MSE) were calculated to assess the models. In the case of the gradient boosting regression model, the deviance of both training and test sets, as well as the feature and permutation importance were analyzed.

After the initial assessment, a final model was trained with the 513 peptide substrates, and applied using the 365 random sequences. The same R2 and MSE metrics were calculated to validate the model. A script to configure and run the predictions with the protease-peptide trajectories is also provided within the code. For this example, we add the average scores as a pre-calculated variable per peptide to be used as the model output. However, the users have the option to select as response any proxy affinity value, mostly motivated in the cases where running such calculations require demanding computational times and resources.

3. Results and discussion

The three protocols covering peptide docking and scoring analysis were applied using a protease (granzyme B) system as an example. In the next section, we discuss their implementation and provide insights about their use for other applications.

Table 1
Protease-peptide (in gray) and LEADS-PEP protein-peptide complexes selected from the PDB to assess the fragment-growing peptide docking approach.

PDB id	Peptide sequence	Protocol RMSD (Å)
1ou8	GRHGAANDENY	3.2
2xxn	SVWIPVNEGASTSGM	4.8
3qdz	TPSILPAPR	4.1
3tjv	PTSYAGDDS	2.7
6di8	CGVPAIQPVLSGL	3.9
1elw	GPTIEEVD	2.4
1ntv	NFDNPVYRKT	2.9
2b9h	RRNLKGLNLNLH	3.6
2w0z	APPPRPPKP	2.2
2w10	PPPRPTAPKPLL	3.4
2xfx	VGYPKVKEEML	4.2
3ch8	PQPVDSWV	2.2
3obq	PTPSAPVPL	4.1
4btb	PPPPPPPPP	2.0
4eik	SLARRPLPPLP	3.5

3.1. Peptide docking protocol

The docking of highly flexible molecules is an active research area with different software available for that purpose. This includes packages such as FlexPepDock [49], GalaxyPepDock [50], and DynaDock [2], which use different computational strategies and scoring functions to predict refined protein-peptide complexes. However, most of these methods are available as public web servers where the code is not available, or as part of open pipelines demanding previous exhaustive analysis of the initial template, limiting their usability. In our case, the docking protocol implements open source software to dock peptides in a protein binding site, dealing with the intrinsic flexibility of the peptide, and allowing the customization of the protocol in case the user wants to modify not only parameters but substantial parts of the protocol. The workflow we propose is summarized in Fig. 1.

One goal of the protocol is to include the flexibility of the molecule, in this case the peptide, but gradually after each iteration. The internal rotational bonds of the peptide are configured first as flexible, but after growing the flanking regions the best conformations of the previous steps are kept rigid to diminish the computational time, allowing the new fragments to explore the best pose based on the rigid selected template, and the flexible new flanking amino acids. The protocol has benefited from the multithreading architecture available in Autodock Vina and auxiliary programs to model additional amino acids in the peptide, protonate according to the system requirements, and tailor the peptide based on the known binding site and biological background available.

To validate the protocol performance for proteases, we selected first a dataset of five protease-peptide complexes available in the PDB. The peptides were selected by taking into account full peptide substrates longer than 8 amino acids belonging to different families and with loop structures. One advantage is the availability of crystallized bound conformations to compare the docking results through RMSD values. We also performed the docking using alternative protocols for peptide docking (see Methods). In addition, we followed a similar docking analysis and RMSD calculation using a set of 10 protein-peptide complexes available in the LEADS-PEP dataset. A summary of the peptide RMSD for our fragment-growing protocol is provided in Table 1.

We found that our protocol predicts peptide conformations with RMSD values below 5 Å for the proteases included in the benchmarking, and a similar performance was found with the additional protein-peptide complexes reported in the LEADS-PEP dataset. In general, reproducing flexible backbone conformations is a subject of research that can be complemented with sampling of the conformational space using MD and other techniques, which is the case of our pipeline. Other alternatives to perform peptide docking presented similar, and in some cases

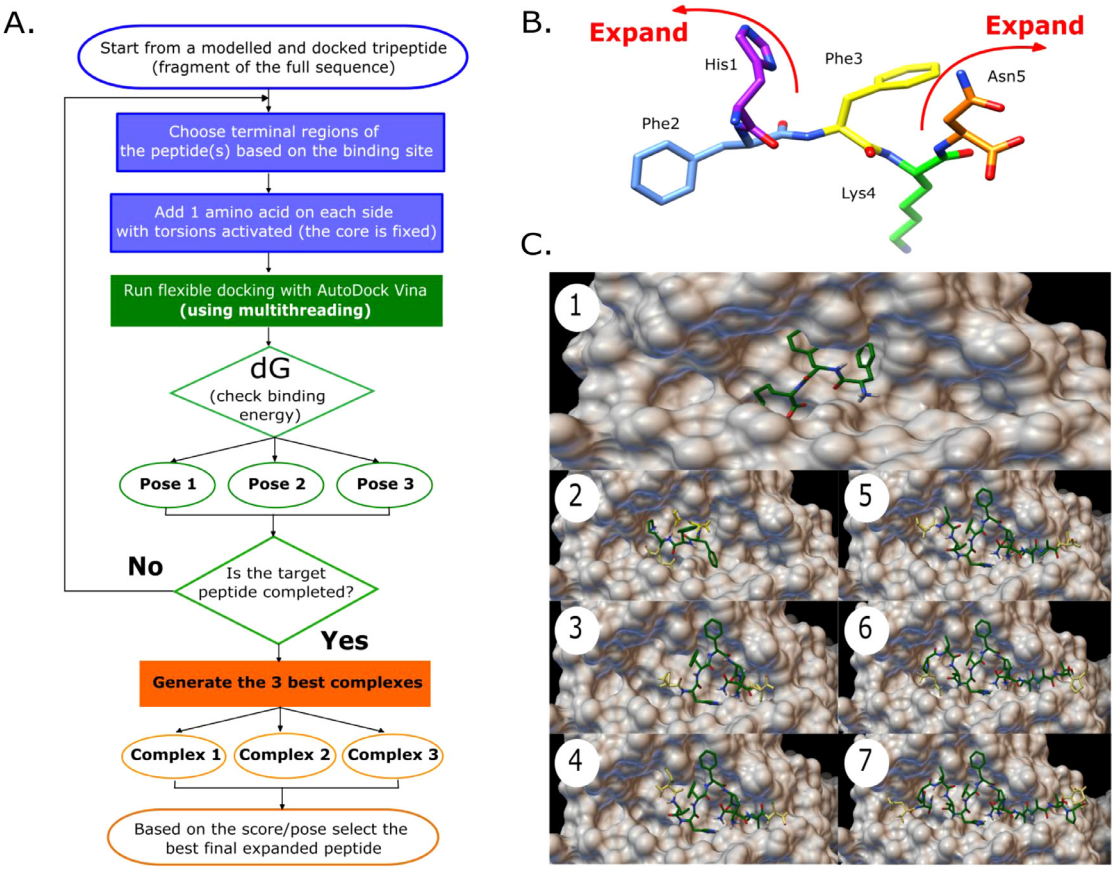

Fig. 1. Workflow of the protein-peptide fragment-growing docking protocol. (A) The protocol highlights the selection of the initial fragment, the docking, and the iterative expansion of the structure through the addition of amino acids in the flanking regions. The pipeline uses three starting docked fragments to grow the sequence in parallel. The best scored final complex can be used for further sampling steps. The blue section describes the expansion steps, the green section the docking and ranking, and the orange section the protocol output. (B) Example of a peptide fragment and how it is expanded through the flanking regions. (C) Snapshots of the evolution of a peptide subjected to the fragment-growing docking protocol. The initial fragment (in green) is grown by single amino acids at each flanking part of the peptide (in yellow). After each step, the docked part remains rigid, and the new additions are configured as flexible. The snapshots represent the growing steps of the docking protocol until completing the full bound peptide.

better results, but with the disadvantage of being accessed only through web servers, and limiting batch jobs for large peptide binders datasets. Among the external programs, it is important to mention that ClusPro and Haddock are rigid body docking approaches where the results depend on the peptide initial conformation. It means that the output will be affected if the structure of the input is very different from that of the crystallographic structure. Unlike the two previous cases, HPEPDock and DINC2 allow implicit flexibility in the peptide structure through ensemble docking or by fragment-based docking respectively, increasing the chances to find optimal conformations. MDockPep and CABSDock can model and dock the structure of the peptide from its sequence.

The docking protocol provided in our scripts is inspired by the DINC2 methodology [23], with differences around the parallelization of the code, the selection of candidates, and the requirement of peptide fragments as input files, which are modelled within our protocol. In the case of DINC2, more parallel runs are generated to select the candidates from a group of solutions. In our case, we provide three final protein-peptide complexes that can be filtered based on the predicted score or conformational pose. The number of runs can be changed by the user in the code. In terms of the assessment performance, one of the limitations is associated with the size of the peptide, as in our case. Specifically, the larger the size the greater the conformational freedom, as can be appreciated for larger peptides with higher RMSD values in Table 1. The RMSD for the external peptide docking servers are reported in Supplementary Table S3.

Differences in predictions can be related with the peptide initial conformation. In our docking protocol, the peptide input conformer is predicted, or a crystallized fragment from a structural complex can be used directly to grow the full peptide sequence. On the other hand, the inputs used for the servers were generated with PEP-FOLD. In spite of not accurately reproducing the crystal structures, our method can be used to screen massive amounts of ligands that can be refined as proposed. In addition, we compared the computational efficiency of our protocol against Autodock Vina itself, using one protein-peptide system as reference (PDB id 1bx2). We found that our method can run the docking in half of the time required by Vina with 24 CPU cores, and the final docking pose was less accurate with regard to the crystal structure when Vina was used alone (see Supplementary Fig. S1).

For the granzyme B system, we docked the 878 peptides at the crystal binding site, where 513 are known substrates and the remaining 365 are the random sequences generated with PepFun. After docking all the substrates and random peptide sequences to the granzyme B system, we followed a pipeline where each complex is subjected to MD simulations and a set of descriptors are extracted to build a predictive machine learning model (Fig. 2A). The latest is important because it is difficult to explore the conformational space using classic docking protocols. This can be overcomed by refining or sampling the complexes using MD, Metropolis Monte Carlo, among other sampling techniques. Our protocol generates short trajectories of the systems with MD equilibrium simulations of 10 ns, which is a suitable time for massive virtual screen-

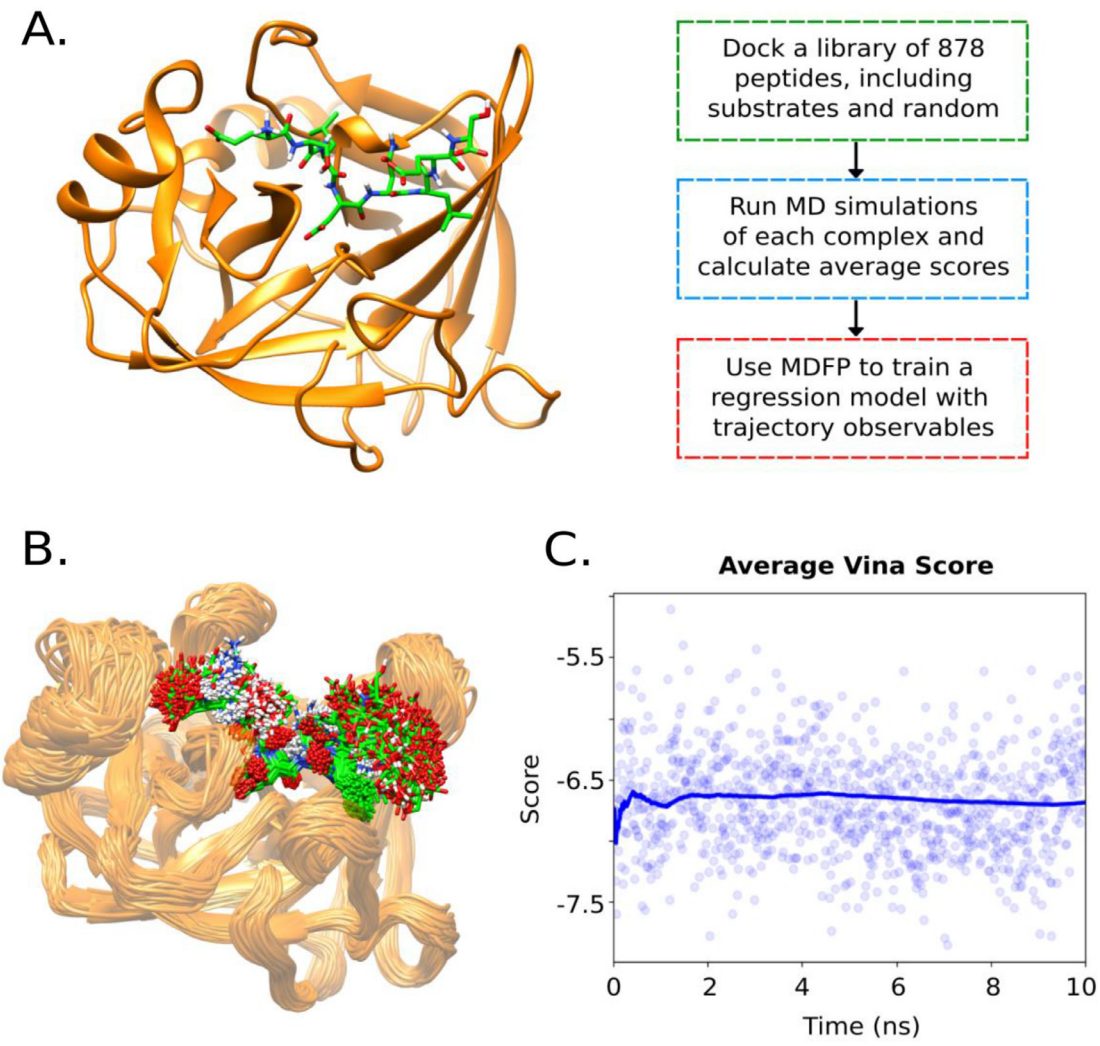

Fig. 2. Granzyme B system used as toy example to test the proposed protocols. (A) Representation of the protease structure and general pipeline followed during the application. (B) Overlapping of MD snapshots obtained after 10 ns simulations. (C) Example of the score average for one protease-peptide trajectory. The continuous line is the cumulative average.

ing campaigns in order to increase conformational variability within the same minima found during the fitting process (Fig. 2B), and therefore statistically improve the predicted score by scoring each frame and averaging it as a proxy affinity response (Fig. 2C).

3.2. Model descriptors

From the MD simulations for all the 878 peptides (513 substrates and 365 random sequences), 70 previously described descriptors (see Methods) were calculated for the machine learning (ML) setup and selected as independent variables. The average Autodock Vina score obtained from the MD simulations for each protein-peptide complex was selected as the dependent variable to be predicted in the regression model. This combined MD/scoring approach has been implemented in the past to filter candidates that agree with experimental data [51,52]. However, users can run more exhaustive free energy calculations to predict the affinities or to include experimental values if these are available, in order to justify the setup of a regression model able to predict energies using solely basic MD simulations. This reduces the required computational resources and provides a hybrid MD/ML approach that can be more efficient for massive analysis such as protease novel substrate recognition [53].

The example provided is an illustrative case where capturing information from MD simulations can allow the reduction of prospective simulation time by preparing a model able to predict a particular MD-related observable. It means that information from MD trajectories can be captured in an ML model trained with MD-based descriptors, or if an experimental value is available, a descriptive model can be generated and complemented using the same MD-based descriptors. To visualize how these 70 descriptors are distributed among the included sequences, histograms of some properties are shown in Supplementary Fig. S2. We also plotted the distribution of the average scores to assess the variability of the metric to be predicted in the example, which is suitable for the machine learning application (Supplementary Fig. S3). A summary of the suggested model is shown in Fig. 3.

3.3. Machine learning model performance

As a toy example, we provide a simple analysis of training and testing a regression model using the descriptors calculated in the previous step. The goal is to illustrate one way to run the analysis, but the user has the option to customize the protocols based on the system and data available to build their own models.

For this application and based on the defined set of descriptors, we trained two regression models using the 513 peptide substrates as the

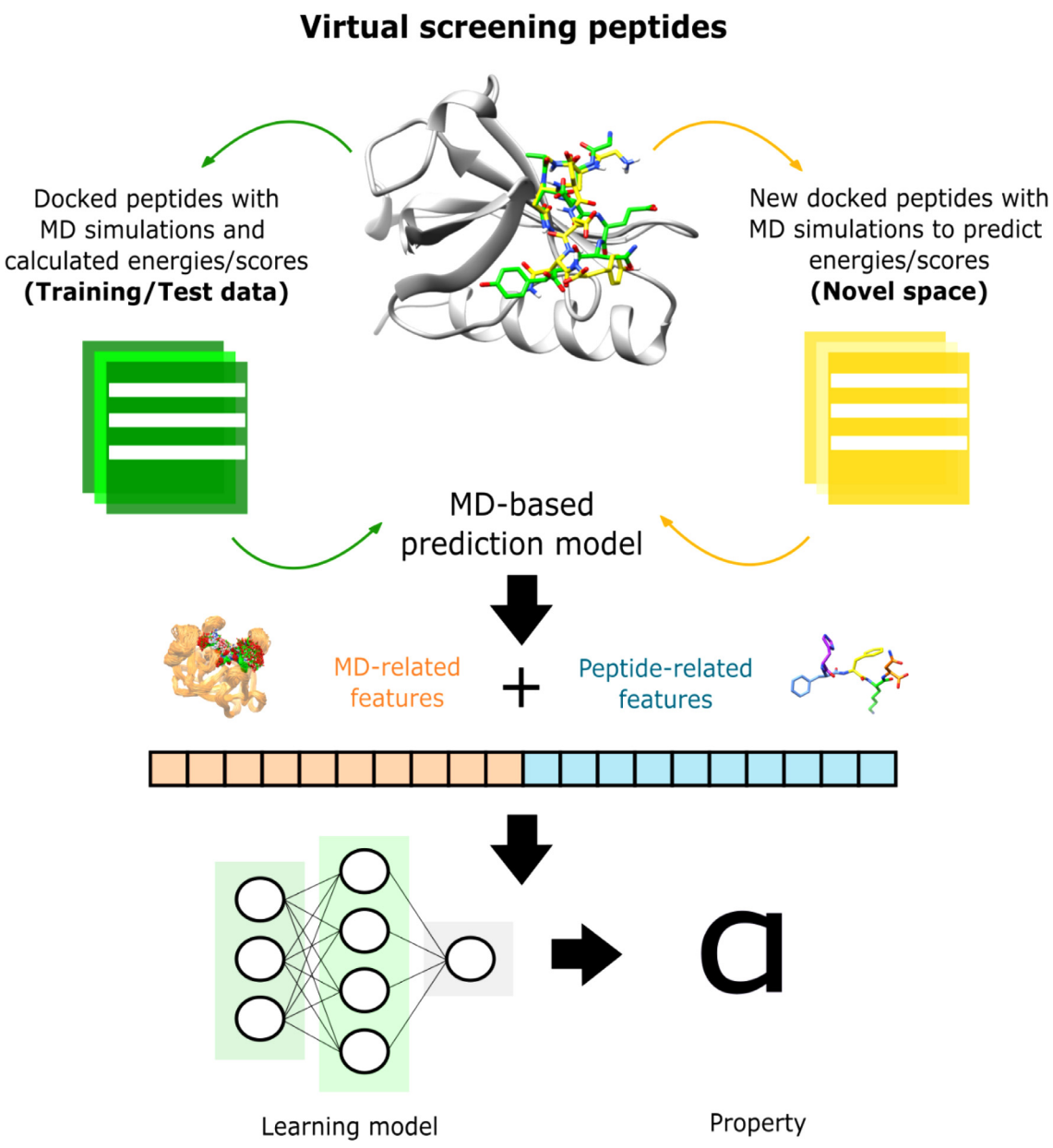

Fig. 3. Overall machine learning strategy. This includes the definition of training and test datasets based on sampled substrates and random docked peptides. The MD simulations and the peptide intrinsic properties are used to extract a set of descriptors to use them as input in predictive regression models.

training set, and the 365 random peptides as the test set. However, with the 513 peptides we did a 75/25 training/test schema using a maximal chemically diversity analysis with the included peptides (see Methods). The main regression metrics for this analysis is provided in the Supplementary Table S4. The results for the final training and test using the average score as the output variable is shown in Table 2.

In general, we observed a better performance of the linear regression model. However, in both cases the performances were acceptable with R2 values over 0.7. In particular, the gradient boosting technique allowed us to visualize per iteration the deviance of the results, and check which features are contributing the most to the predictions (Fig. 4).

Regarding the features, the most relevant are those derived from the protease-peptide energy terms, including the Coulomb and Lennard-Jonnes average and median descriptors. Some ligand-based features are also highlighted such as the number of rotatable bonds, which is an indicator of the peptide flexibility. An additional permutation importance analysis was performed with similar results about the most relevant variables (Supplementary Fig. S4). The protocol to reproduce the training of both models, as well as generate the gradient boosting related figures is available within the code repository.

Table 2
Regression performances for the gradient boosting regressor and the linear regression model trained with the peptide substrates and tested with the random peptides. The metrics are associated with the predictions using the test set.

Metrics	Gradient boosting	Linear regression
Mean Squared Error (MSE)	0.195	0.075
Pearson correlation	0.877	0.947
R^2	0.721	0.892

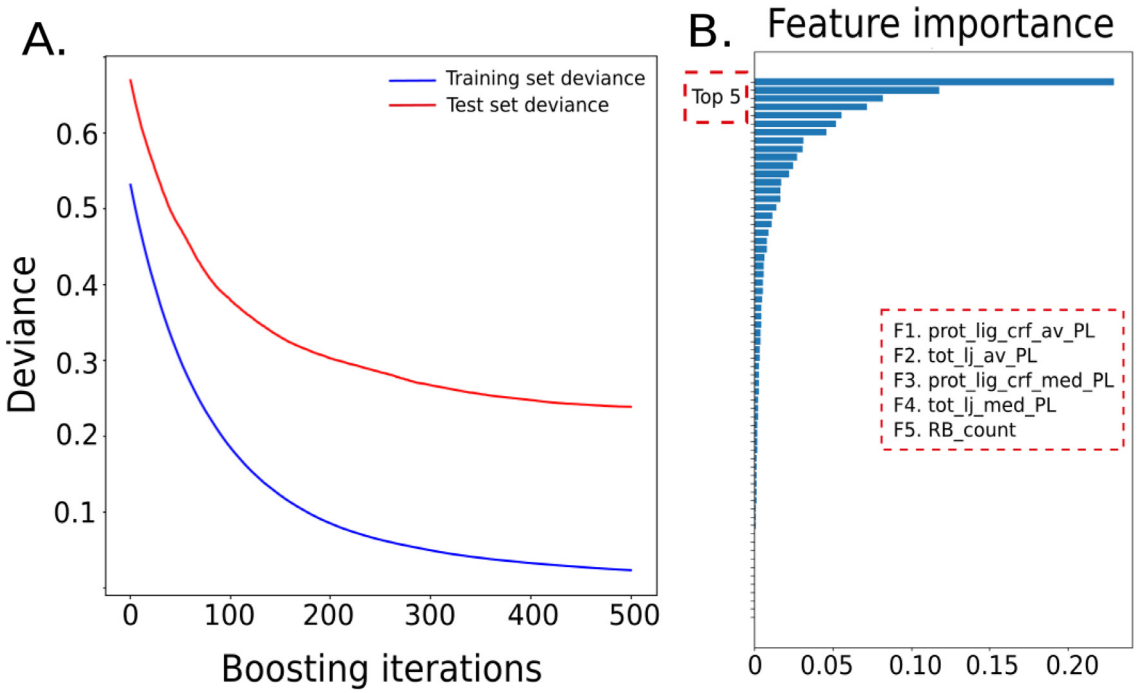

Fig. 4. Gradient boosting regression model deviance and feature importance. (A) Deviance evolution of the model with the training set deviance in blue and the test set deviance in red. (B) Feature importance with the top 5 features of the model mentioned in the red square.

3.4. Technical considerations

The protocols were designed under the Ubuntu 20.04 operating system. However, the project can be installed in any Conda virtual environment with the required dependencies. In the case of the docking protocol, it depends on calls to bash commands that were originally tested in a Linux environment. All the workflows depend on the addition of third-party tools, which are also open source, but will require building the source code in the local machine where the project will be implemented.

The docking method we provide is a supervised approach, initially motivated by a local project to dock epitopes into the MHC class II binding site [54]. This means that instead of providing a blind docking strategy where the binding site is undetermined, the user is required to customize the process based on previous knowledge of the protein and binding site, in order to focus the analysis on managing the peptide intrinsic flexibility and reproducing plausible bound conformations. To allow this, a grid search space should be assigned based on a required size. A set of known binding site coordinates must be provided to dock the initial peptide fragment, which can affect the starting point to grow the bound peptide structure. We recommend that if a crystallized protein-peptide complex exists, a tripeptide obtained directly from the reference complex should be considered for the fragment-growing docking approach, to improve the results.

Finally, we suggest the parameters for running the MD simulations, but the user can configure them based on their own necessities. The only requirement is to use Gromacs for the calculations.

4. Conclusion

The computational study of how peptides interact with other molecular entities is crucial to accelerate the design of novel sequences with better properties, including their affinities. In this work, we provide three open protocols that can be implemented to any protein-peptide of interest but exemplified in the context of a granzyme B system with available biological and structural knowledge. The protocols allow the analysis of massive peptide substrates through fragment-based docking and MD sampling and scoring of the molecules. The code is open and can be modified to fit the user necessities, add new types of predictions and simulations, and automatize the pipeline for virtual screening purposes.

Data availability

The code, examples, and instructions to run the protocols are publicly available at: https://github.com/rochoa85/Protocols-Peptide-Binding.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The computations were performed in a local server of the Max Planck tandem group with an NVIDIA Titan X GPU. The project was funded by Minciencias, University of Antioquia, Ruta N, Colombia, and the Max Planck Society, Germany.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ailsci.2022.100044.

References

[1] London N, Raveh B, Schueler-Furman O. Peptide docking and structure-based characterization of peptide binding: from knowledge to know-how. Curr Opin Struct Biol 2013;23:894–902. doi:10.1016/j.sbi.2013.07.006.
[2] Antes I. DynaDock: a new molecular dynamics-based algorithm for protein-peptide docking including receptor flexibility. Proteins 2010;78:1084–104. doi:10.1002/prot.22629.
[3] Florez AM, Suarez-Barrera MO, Morales GM, Rivera KV, Orduz S, Ochoa R, Guerra D, Muskus C. Toxic activity, molecular modeling and docking simulations of bacillus thuringiensis cry11 toxin variants obtained via DNA shuffling. Front Microbiol 2018;9:2461. doi:10.3389/fmicb.2018.02461.

[4] Weng G, Gao J, Wang Z, Wang E, Hu X, Yao X, Cao D, Hou T. Comprehensive evaluation of fourteen docking programs on protein-peptide complexes. J Chem Theory Comput 2020;16:3959–69. doi:10.1021/acs.jctc.9b01208.

[5] Unal EB, Besray Unal E, Gursoy A, Erman B. VitAL: viterbi algorithm for de novo peptide design. PLoS One 2010;5:e10926. doi:10.1371/journal.pone.0010926.

[6] Antunes DA, Devaurs D, Moll M, Lizée G, Kavraki LE. General prediction of peptide-MHC binding modes using incremental docking: a proof of concept. Sci Rep 2018;8. doi:10.1038/s41598-018-22173-4.

[7] Ochoa R, Laio A, Cossio P. Predicting the affinity of peptides to major histocompatibility complex class II by scoring molecular dynamics simulations. J Chem Inf Model 2019;59:3464–73. doi:10.1021/acs.jcim.9b00403.

[8] Ochoa R, Soler MA, Laio A, Cossio P. Assessing the capability of in silico mutation protocols for predicting the finite temperature conformation of amino acids. Phys Chem Chem Phys 2018;20:25901–9. doi:10.1039/C8CP03826K.

[9] Kamenik AS, Lessel U, Fuchs JE, Fox T, Liedl KR. Peptidic macrocycles - conformational sampling and thermodynamic characterization. J Chem Inf Model 2018;58:982–92. doi:10.1021/acs.jcim.8b00097.

[10] Yan Y, Zhang D, Huang S-Y. Efficient conformational ensemble generation of protein-bound peptides. J Cheminform 2017;9:59. doi:10.1186/s13321-017-0246-7.

[11] Rastelli G, Pinzi L. Refinement and rescoring of virtual screening results. Front Chem 2019;7:498. doi:10.3389/fchem.2019.00498.

[12] Chodera JD, Mobley DL, Shirts MR, Dixon RW, Branson K, Pande VS. Alchemical free energy methods for drug discovery: progress and challenges. Curr Opin Struct Biol 2011;21:150–60. doi:10.1016/j.sbi.2011.01.011.

[13] Genheden S, Ryde U. The MM/PBSA and MM/GBSA methods to estimate ligand-binding affinities. Expert Opin Drug Discov 2015;10:449–61. doi:10.1517/17460441.2015.1032936.

[14] Jacob K S, Ganguly S, Kumar P, Poddar R, Kumar A. Homology model, molecular dynamics simulation and novel pyrazole analogs design of Candida albicans CYP450 lanosterol 14 α-demethylase, a target enzyme for antifungal therapy. J Biomol Struct Dyn 2017;35:1446–63. doi:10.1080/07391102.2016.1185380.

[15] Amaro RE, Baron R, McCammon JA. An improved relaxed complex scheme for receptor flexibility in computer-aided drug design. J Comput-Aided Mol Des 2008;22:693–705. doi:10.1007/s10822-007-9159-2.

[16] Wang DD, Ou-Yang L, Xie H, Zhu M, Yan H. Predicting the impacts of mutations on protein-ligand binding affinity based on molecular dynamics simulations and machine learning methods. Comput Struct Biotechnol J 2020;18:439–54. doi:10.1016/j.csbj.2020.02.007.

[17] Ballester PJ, Mitchell JBO. A machine learning approach to predicting protein-ligand binding affinity with applications to molecular docking. Bioinformatics 2010;26:1169–75. doi:10.1093/bioinformatics/btq112.

[18] Wang Y, Lamim Ribeiro JM, Tiwary P. Machine learning approaches for analyzing and enhancing molecular dynamics simulations. Curr Opin Struct Biol 2020;61:139–45. doi:10.1016/j.sbi.2019.12.016.

[19] Dral PO. Quantum chemistry in the age of machine learning. J Phys Chem Lett 2020;11:2336–47. doi:10.1021/acs.jpclett.9b03664.

[20] Riniker S. Molecular dynamics fingerprints (MDFP): machine learning from MD data to predict free-energy differences. J Chem Inf Model 2017;57:726–41. doi:10.1021/acs.jcim.6b00778.

[21] Wang S, Riniker S. Use of molecular dynamics fingerprints (MDFPs) in SAMPL6 octanol-water log P blind challenge. J Comput-Aided Mol Des 2020;34:393–403. doi:10.1007/s10822-019-00252-6.

[22] Esposito C, Wang S, Lange UEW, Oellien F, Riniker S. Combining machine learning and molecular dynamics to predict P-glycoprotein substrates. J Chem Inf Model 2020;60:4730–49. doi:10.1021/acs.jcim.0c00525.

[23] Antunes DA, Moll M, Devaurs D, Jackson KR, Lizée G, Kavraki LE. DINC 2.0: a new protein–peptide docking webserver using an incremental approach. Cancer Res 2017;77:e55–7. doi:10.1158/0008-5472.can-17-0511.

[24] Webb B, Sali A. Comparative protein structure modeling using MODELLER. Curr Protoc Protein Sci 2016;86:2.9.1–2.9.37. doi:10.1002/cpbi.3.

[25] Dolinsky TJ, Nielsen JE, McCammon JA, Baker NA. PDB2PQR: an automated pipeline for the setup of Poisson-Boltzmann electrostatics calculations. Nucleic Acids Res 2004;32:W665–7. doi:10.1093/nar/gkh381.

[26] Trott O, Olson AJ. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. J Comput Chem 2010;31:455–61. doi:10.1002/jcc.21334.

[27] Rotonda J, Garcia-Calvo M, Bull HG, Geissler WM, McKeever BM, Willoughby CA, Thornberry NA, Becker JW. The three-dimensional structure of human granzyme B compared to caspase-3, key mediators of cell death with cleavage specificity for aspartic acid in P1. Chem Biol 2001;8:357–68. doi:10.1016/S1074-5521(01)00018-7.

[28] Ochoa R, Magnitov M, Laskowski RA, Cossio P, Thornton JM. An automated protocol for modelling peptide substrates to proteases. BMC Bioinf 2020;21:586. doi:10.1186/s12859-020-03931-6.

[29] Rawlings ND, Morton FR, Kok CY, Kong J, Barrett AJ. MEROPS: the peptidase database. Nucleic Acids Res 2007;36:D320–5. doi:10.1093/nar/gkm954.

[30] Ochoa R, Cossio P. PepFun: open source protocols for peptide-related computational analysis. Molecules 2021;26. doi:10.3390/molecules26061664.

[31] Kozakov D, Hall DR, Xia B, Porter KA, Padhorny D, Yueh C, Beglov D, Vajda S. The ClusPro web server for protein-protein docking. Nat Protoc 2017;12:255–78. doi:10.1038/nprot.2016.169.

[32] de Vries SJ, van Dijk M, Bonvin AMJJ. The HADDOCK web server for data-driven biomolecular docking. Nat Protoc 2010;5:883–97. doi:10.1038/nprot.2010.32.

[33] Zhou P, Jin B, Li H, Huang S-Y. HPEPDOCK: a web server for blind peptide-protein docking based on a hierarchical algorithm. Nucleic Acids Res 2018;46:W443–50. doi:10.1093/nar/gky357.

[34] Xu X, Yan C, Zou X. MDockPeP: an ab-initio protein-peptide docking server. J Comput Chem 2018;39:2409–13. doi:10.1002/jcc.25555.

[35] Kurcinski M, Jamroz M, Blaszczyk M, Kolinski A, Kmiecik S. CABS-dock web server for the flexible docking of peptides to proteins without prior knowledge of the binding site. Nucleic Acids Res 2015;43:W419–24. doi:10.1093/nar/gkv456.

[36] Lamiable A, Thévenet P, Rey J, Vavrusa M, Derreumaux P, Tufféry P. PEP-FOLD3: faster de novo structure prediction for linear peptides in solution and in complex. Nucleic Acids Res 2016;44:W449–54. doi:10.1093/nar/gkw329.

[37] Hauser AS, Windshügel B. LEADS-PEP: a benchmark data set for assessment of peptide docking performance. J Chem Inf Model 2016;56:188–200. doi:10.1021/acs.jcim.5b00234.

[38] Hess B, Kutzner C, van der Spoel D, Lindahl E. GROMACS 4: algorithms for highly efficient, load-balanced, and scalable molecular simulation. J Chem Theory Comput 2008;4:435–47. doi:10.1021/ct700301q.

[39] Lindorff-Larsen K, Piana S, Palmo K, Maragakis P, Klepeis JL, Dror RO, Shaw DE. Improved side-chain torsion potentials for the Amber ff99SB protein force field. Proteins 2010;78:1950–8. doi:10.1002/prot.22711.

[40] Jorgensen WL, Jenson C. Temperature dependence of TIP3P, SPC, and TIP4P water from NPT Monte Carlo simulations: seeking temperatures of maximum density. J Comput Chem 1998;19:1179–86. doi:10.1002/(sici)1096-987x(19980730)19:10<1179::aid-jcc6>3.0.co;2-j.

[41] Bussi G, Donadio D, Parrinello M. Canonical sampling through velocity rescaling. J Chem Phys 2007;126:014101. doi:10.1063/1.2408420.

[42] Parrinello M, Rahman A. Crystal structure and pair potentials: a molecular-dynamics study. Phys Rev Lett 1980;45:1196–9. doi:10.1103/physrevlett.45.1196.

[43] Di Pierro M, Elber R, Leimkuhler B. A stochastic algorithm for the isobaric-isothermal ensemble with ewald summations for all long range forces. J Chem Theory Comput 2015;11:5624–37. doi:10.1021/acs.jctc.5b00648.

[44] Janezic D, Merzel F. An efficient symplectic integration algorithm for molecular dynamics simulations. J Chem Inf Comput Sci 1995;35:321–6. doi:10.1021/ci00024a022.

[45] Shirts MR, Klein C, Swails JM, Yin J, Gilson MK, Mobley DL, Case DA, Zhong ED. Lessons learned from comparing molecular dynamics engines on the SAMPL5 dataset. J Comput-Aided Mol Des 2017;31:147–61. doi:10.1007/s10822-016-9977-1.

[46] McGibbon RT, Beauchamp KA, Harrigan MP, Klein C, Swails JM, Hernández CX, Schwantes CR, Wang L-P, Lane TJ, Pande VS. MDTraj: a modern open library for the analysis of molecular dynamics trajectories. Biophys J 2015;109:1528–32. doi:10.1016/j.bpj.2015.08.015.

[47] G. Varoquaux, L. Buitinck, G. Louppe, O. Grisel, F. Pedregosa, A. Mueller, Scikit-Learn GetMob Mob Comput Commun 19 (2015) 29–33. 10.1145/2786984.2786995.

[48] Gardiner EJ, Holliday J, O'Dowd C, Willett P. Effectiveness of 2D fingerprints for scaffold hopping. Future Med Chem 2011;3:405–14. doi:10.4155/fmc.11.4.

[49] Raveh B, London N, Schueler-Furman O. Sub-angstrom modeling of complexes between flexible peptides and globular proteins. Proteins 2010;78:2029–40. doi:10.1002/prot.22716.

[50] Lee H, Heo L, Lee MS, Seok C. GalaxyPepDock: a protein-peptide docking tool based on interaction similarity and energy optimization. Nucleic Acids Res 2015;43:W431–5. doi:10.1093/nar/gkv495.

[51] Ochoa R, Watowich SJ, Flórez A, Mesa CV, Robledo SM, Muskus C. Drug search for leishmaniasis: a virtual screening approach by grid computing. J Comput-Aided Mol Des 2016;30:541–52. doi:10.1007/s10822-016-9921-4.

[52] Soler MA, Medagli B, Semrau MS, Storici P, Bajc G, de Marco A, Laio A, Fortuna S. A consensus protocol for the in silico optimisation of antibody fragments. Chem Commun 2019;55:14043–6. doi:10.1039/C9CC06182G.

[53] Barkan DT, Hostetter DR, Mahrus S, Pieper U, Wells JA, Craik CS, Sali A. Prediction of protease substrates using sequence and structure features. Bioinformatics 2010;26:1714–22. doi:10.1093/bioinformatics/btq267.

[54] Ochoa R, Lunardelli VAS, Rosa DS, Laio A, Cossio P, Multiple-Allele MHC. Class II epitope engineering by a molecular dynamics-based evolution protocol. Front Immunol 2022;13:862851. doi:10.3389/fimmu.2022.862851.