ppdx: automated modeling of protein-protein interaction descriptors for use with machine learning

Simone Conti[1], Victor Ovchinnikov[1] and Martin Karplus[1,2]

Correspondence to: Simone Conti (simonecnt@gmail.com), Martin Karplus (marci@tammy.harvard.edu)

[1] Department of Chemistry and Chemical Biology, Harvard University, Cambridge, MA 02138, USA

[2] Laboratoire de Chimie Biophysique, Institut de Science et d'Ingénierie Supramoléculaires, Université de Strasbourg, 67000 Strasbourg, France

Abstract

This paper describes ppdx, a python workflow tool that combines protein sequence alignment, homology modeling, and structural refinement, to compute a broad array of descriptors for characterizing protein-protein interactions. The descriptors can be used to predict various properties of interest, such as protein-protein binding affinities, or inhibitory concentrations (IC_{50}), using approaches that range from simple regression to more complex machine learning models. The software is highly modular. It supports different protocols for generating structures, and 95 descriptors can be currently computed. More protocols and descriptors can be easily added. The implementation is highly parallel and can fully exploit the available cores in a single workstation, or multiple nodes on a supercomputer, allowing many systems to be analyzed simultaneously. As an illustrative application, ppdx is used to parametrize a model that predicts the IC_{50} of a set of antigens and a class of antibodies directed to the influenza hemagglutinin stalk.

Keywords

Protein-Protein Interactions, Machine Learning, Binding Affinity, Scoring Functions, Protein Interaction Descriptors

Introduction

In computational protein design, one is often faced with the task of modeling protein-protein interfaces and characterizing their properties. For example, in antibody or antigen design, one often needs to compute a large number of protein-protein binding affinities upon sequence mutation. An ideal practical tool should be able to produce an accurate affinity estimate within seconds or minutes.

Modeling protein-protein interaction remains notoriously difficult[1,2]. The most accurate physics-based methods involve calculating binding free energy by simulating all-atom models in explicit solvent, computing potentials of mean force of physical separation, or using alchemical free energy perturbation theory[3,4]. Such methods are inefficient for high throughput applications, because they typically exhibit slow convergence and requiring simulation times of hours to days[5]. At the other end of the spectrum are methods that rely on a small number of structures (one or several) to predict affinity using phenomenological modeling[6–8]. They typically compute descriptors from the 3D model coordinates (e.g., the buried surface area, the number of hydrogen bonds, etc.) and use them in regression models. While such methods are generally fast because of the small number of structures required, their accuracy is variable and usually depends on the specific protein-protein complex under investigation; for example, some are parametrized to work for antibody/antigen complexes[9], or for cases in which binding does not result in significant conformational rearrangements[10].

The present software represents an approach to improve the second class of methods to create a general tool that achieves a compromise between speed and accuracy appropriate for computational protein design applications. Our efforts began with the adjustment of coefficients in a linear regression model for the specific application of predicting binding affinities between anti-HIV antibodies and their cognate antigens[11]. However, we soon realized that improved agreement with experiments could be obtained by using different descriptors, a different form of regression, or both. With the advent of easily accessible machine learning libraries[12,13], such as scikit-learn[14], we focused our efforts on developing method for the

calculation of a range of molecular properties and descriptors, which could be used in a machine learning model using only a few lines of code.

The simple standalone tools we had used initially[11,15] thus evolved into a more complete software package, called ppdx, for Protein-Protein ΔX, where X can, in principle, be any property of interest: a binding affinity, an inhibitory concentration (IC_{50}), or even rates of binding and unbinding. The package has three main components: (i) an interface to structure modeling tools for creating the 3D coordinates of the protein-protein complexes starting with the protein sequences and a template for the structure; (ii) computation of a variety of descriptors, which can be used to train regression or machine learning models; (iii) a simple interface to use the trained models for predictions of new protein-protein complexes. In the Methods section, we describe the protocols to generate protein-protein complex structures and the 95 different descriptors currently available for calculation. In the Results, we describe an illustrative application of ppdx to characterize 350 complexes between influenza hemagglutinins and antibody (Fab) fragments and to parametrize a model to compute their IC_{50}.

ppdx is highly modular, extensible, and can be run in parallel on a single workstation or on multiple nodes on a supercomputer. The code is freely available under GPLv3 license at https://github.com/simonecnt/ppdx

Methods

The overall procedure is to generate multiple models of the complex using protein structure modeling tools such as Modeller[16] or Rosetta[17,18], compute descriptors for each of them, and average the scores over the generated models. The computed descriptors can then be used to parametrize regressions or to train machine learning models to predict properties of related complexes, such as their binding affinity or IC_{50} upon mutation; see Figure 1. This code is based on earlier prototypes, which were described elsewhere[11,15].

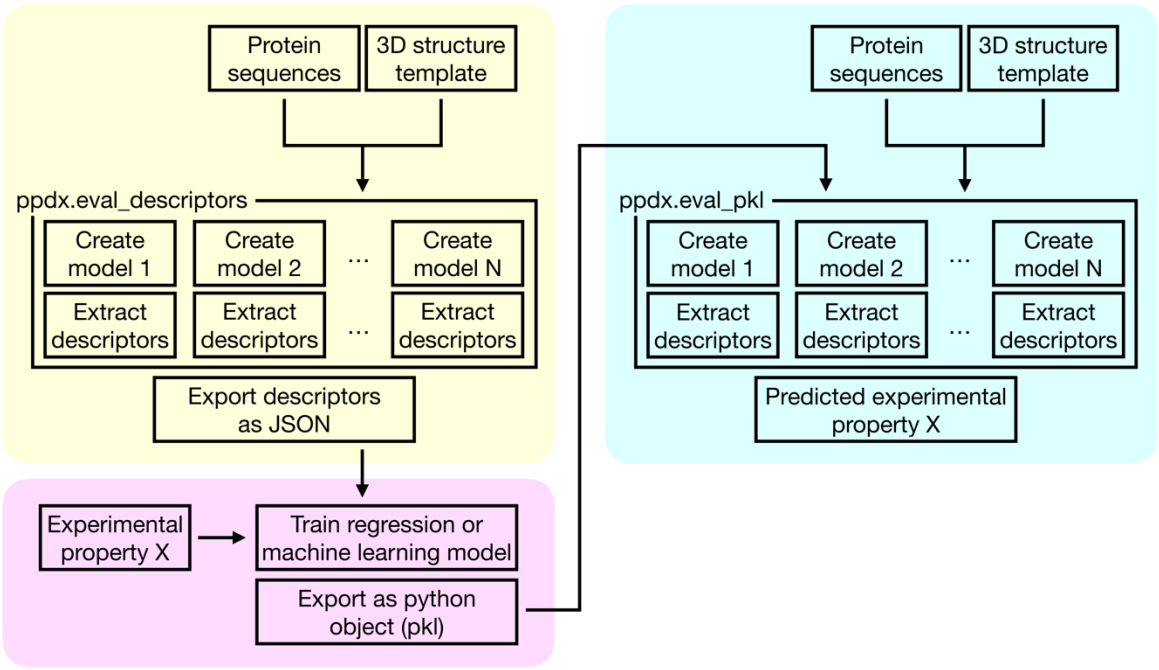

Figure 1. Scheme representing the workflow implemented in ppdx. On the left, ppdx is used to compute descriptors for a set of known protein-protein complexes (yellow-shaded section), and these are used, together with known experimental data, to train a model to predict such data (purple-shaded section); e.g., a binding affinity or an IC_{50} value. On the right (blue-shaded section), ppdx is used to apply the model to some new protein-protein complexes to predict the experimental property of interest.

Creating atomistic models

Starting from the amino acid sequences of the receptor and the ligand, and a template structure of the full complex, Modeller[16] or Rosetta[17,18] are used to create structural models of the protein-protein complexes of interest. Four protocols to generate the atomistic models are implemented, three using Modeller, and one using Rosetta; more protocols can be easily added, for example using other software suites such as AlphaFold[19] or I-TASSER[20]. The three Modeller protocols (modeller_veryfast, modeller_fast, and modeller_slow) differ in the Modeller md_level and repeat_optimization parameters. These protocols are increasingly computationally expensive and produce more refined models. For the Rosetta[17] protocol we use RosettaCM[18], which consists of two stages: first, ''threading'' the target sequence onto the

given template structure, and second, modeling missing residues and a final energy-based optimization. All four protocols generate by default one stochastic model; more models can be used, depending on the available computational resources. The structures are then processed by CHARMM[21], which adds missing coordinates, if any, and disulfide bonds, and refines the complete structure with the Steepest Descent (SD) and Adopted Basis Newton-Raphson (ABNR) energy minimizers; the free academic version of CHARMM is sufficient. Finally, the model of the complex is divided into models of the receptor and the ligand, which are not further optimized.

Available descriptors

Descriptors (or features) of the protein-protein complexes are computed from the atomistic models. Currently, 95 descriptors (described below) can be computed; details and references are given in Table 1. Descriptors can be broadly classified into three groups: (1) molecular descriptors that are functions of the atomic coordinates, such as the number of hydrogen bonds between the receptor and the ligand[22–24], the buried surface area[25], and others[26,27]; (2) protein-protein docking scores computed from scoring functions[28–33]; where possible, intermediate components used in the computation of the final score are also extracted; (3) the change in protein folding propensity or stability upon binding. This last group contains descriptors with widely varying computational cost. Statistical potentials[34–38] are generally fast, and provide an energy for a protein (or protein complex) structure; they are used here to evaluate the change in the scores upon binding. We also consider several potentials with greater complexity[10,39–41]. One particularly popular approach to evaluate the interaction energy between two proteins is MMGBSA. The total energy of each protein or complex is estimated as the sum of a molecular mechanics term (MM), a polar solvation contribution evaluated in the Generalized-Born approximation (GB)[42], and a non-polar solvation term proportional to the solvent accessible surface area (SA). The MM term is modeled with the CHARMM36 classical force field[43], while a few implicit solvent models are considered for the GB term, some[44–47] as implemented in CHARMM[21], others[48–51] as implemented in OpenMM[52], and Modeller[53]. As a general protocol, the complex is minimized using the chosen GB model, and the interaction energy is evaluated from the difference of the energy of the minimized complex and the

receptor and ligand taken apart (without further minimizations). Two additional descriptors represent the harmonic entropy computed from elastic network models[54,55]. A coarse-grained model of the protein is constructed using only the beta carbon coordinate of each amino acid, with each pair within a prescribed separation distance connected by an elastic spring vibrating around the equilibrium distance taken from the initial coordinates. The force constant of the spring depends inversely on the equilibrium distance, e.g., atoms at greater distances have lower spring constant[56,57]. The entropy of this model is obtained from the eigenvalues of the Hessian matrix of the energy. The entropy difference is computed as difference between the entropy of the complex and the separated ligand and receptor.

Table 1. List of descriptors currently available.

Name	**Description**	**Software**
HB_BH	Number of hydrogen bonds as defined by Baker & Hubbard[22]	MDTraj[58]
HB_WN	Number of hydrogen bonds as defined by Wernet, Nilsson et al.[23]	MDTraj[58]
HB_KS	Hydrogen bond energy as defined by Kabsch & Sander[24]	MDTraj[58]
BSA	Total buried surface area upon binding[25]	MDTraj[58]
BSA_C	Charged buried surface area upon binding	Native (MDTraj[58])
BSA_A	Apolar buried surface area upon binding	Native (MDTraj[58])
BSA_P	Polar buried surface area upon binding	Native (MDTraj[58])
NIS_P	Fraction of polar non-interacting surface area[26]	Native (MDTraj[58])
NIS_C	Fraction of charged non-interacting surface area[26]	Native (MDTraj[58])
NIS_A	Fraction of apolar non-interacting surface area[26]	Native (MDTraj[58])
sticky_tot	Total “stickiness” as defined by Levy et al.[27]	Native (MDTraj[58])
sticky_avg	Average “stickiness” as defined by Levy et al.[27]	Native (MDTraj[58])
IC_TOT	Total number of interchain contacts[26]	Biopython[59]
IC_AA	Number of apolar-apolar interchain contacts[26]	Biopython[59]
IC_PP	Number of polar-polar interchain contacts[26]	Biopython[59]
IC_CC	Number of charged-charged interchain contacts[26]	Biopython[59]
IC_AP	Number of apolar-polar interchain contacts[26]	Biopython[59]
IC_CP	Number of charged-polar interchain contacts[26]	Biopython[59]
IC_AC	Number of apolar-charged interchain contacts[26]	Biopython[59]
ZRANK	ZRANK docking scoring function[28]	ZRANK[28]
ZRANK2	ZRANK2 docking scoring function[29,30] (with the -R option)	ZRANK[28]
pyDock	PyDock docking scoring function[31] (with pydock setup followed by pydock dockser)	pyDock[31]
pyDock_elec	Electrostatic component of pyDock scoring function[31]	pyDock[31]
pyDock_vdw	Van der Waals component of pyDock scoring function[31]	pyDock[31]
pyDock_desolv	Desolvation component of pyDock scoring function[31]	pyDock[31]

Name	Description	Software
ATTRACT	ATTRACT docking scoring function[32] on the structure prepared by REDUCE[60]	ATTRACT[32]
FireDock	FireDock docking scoring function[33] on the structure prepared by REDUCE[60]	FireDock[33]
FireDock_aVdW	Attractive Van der Waals component of FireDock scoring function[33]	FireDock[33]
FireDock_rVdW	Repulsive Van der Waals component of FireDock scoring function[33]	FireDock[33]
FireDock_ACE	Atomic contact energy (ACE) component of FireDock scoring function[33]	FireDock[33]
FireDock_inside	"Insideness" measure component of FireDock scoring function[33]	FireDock[33]
FireDock_aElec	Attractive electrostatic component of FireDock scoring function[33]	FireDock[33]
FireDock_rElec	Repulsive electrostatic component of FireDock scoring function[33]	FireDock[33]
FireDock_laElec	Attractive long-range electrostatic component of FireDock scoring function[33]	FireDock[33]
FireDock_lrElec	Repulsive long-range electrostatic component of FireDock scoring function[33]	FireDock[33]
FireDock_hb	Hydrogen bonds component of FireDock scoring function[33]	FireDock[33]
FireDock_piS	Pi-Stacking component of FireDock scoring function[33]	FireDock[33]
FireDock_catpiS	Cation-pi component of FireDock scoring function[33]	FireDock[33]
FireDock_aliph	Aliphatic component of FireDock scoring function[33]	FireDock[33]
RF_HA_SRS	"Heavy atoms" pairwise statistical potentials by Rykunov and Fiser[34,35]	RFPP[34,35]
RF_CB_SRS_OD	"CB directional" pairwise statistical potentials by Rykunov and Fiser[34,35]	RFPP[34,35]
ipot_aace167	AACE167 iPot contact potential[36]	iPot[36]
ipot_aace18	AACE18 iPot contact potential[36]	iPot[36]
ipot_aace20	AACE20 iPot contact potential[36]	iPot[36]
ipot_rrce20	RRCE20 iPot contact potential[36]	iPot[36]
SOAP-PP-Pair	SOAP potential for protein-protein interfaces[37]	Modeller[16]
SOAP-Protein-OD	SOAP potential for protein structures[37]	Modeller[16]
DOPE	DOPE potential in Modeller[38]	Modeller[16]
DOPE-HR	High-resolution DOPE potential in Modeller[38]	Modeller[16]
AGBNP	Implicit solvation energy according to the AGBNP[53] model.	Modeller[16]
FACTS_ELEC	Electrostatic interaction as computed in the FACTS implicit solvent model[44]	CHARMM[21]
FACTS_VDW	Van der Waals interaction as computed in the FACTS implicit solvent model[44]	CHARMM[21]
FACTS_GB	Generalized-Born interaction as computed in the FACTS implicit solvent model[44]	CHARMM[21]
FACTS_ASP	Apolar interaction as computed in the FACTS implicit solvent model[44]	CHARMM[21]
FACTS_POL	Polar interaction as computed in the FACTS implicit solvent model[44]	CHARMM[21]

Name	Description	Software
FACTS_TOT	Total interaction energy as computed in the FACTS implicit solvent model[44]	CHARMM[21]
GBMV_ELEC	Electrostatic interaction as computed in the GBMV implicit solvent model[45,46]	CHARMM[21]
GBMV_VDW	Van der Waals interaction as computed in the GBMV implicit solvent model[45,46]	CHARMM[21]
GBMV_GB	Generalized-Born interaction as computed in the GBMV implicit solvent model[45,46]	CHARMM[21]
GBMV_ASP	Apolar interaction as computed in the GBMV implicit solvent model[45,46]	CHARMM[21]
GBMV_POL	Polar interaction as computed in the GBMV implicit solvent model[45,46]	CHARMM[21]
GBMV_TOT	Total interaction energy as computed in the GBMV implicit solvent model[45,46]	CHARMM[21]
GBSW_ELEC	Electrostatic interaction as computed in the GBSW implicit solvent model[47]	CHARMM[21]
GBSW_VDW	Van der Waals interaction as computed in the GBSW implicit solvent model[47]	CHARMM[21]
GBSW_GB	Generalized-Born interaction as computed in the GBSW implicit solvent model[47]	CHARMM[21]
GBSW_ASP	Apolar interaction as computed in the GBSW implicit solvent model[47]	CHARMM[21]
GBSW_POL	Polar interaction as computed in the GBSW implicit solvent model[47]	CHARMM[21]
GBSW_TOT	Total interaction energy as computed in the GBSW implicit solvent model[47]	CHARMM[21]
CDIE_ELEC	Electrostatic interaction as computed in a constant dielectric model[43]	CHARMM[21]
CDIE_VDW	Van der Waals interaction as computed in a constant dielectric model[43]	CHARMM[21]
CDIE_TOT	Total interaction energy as computed in a constant dielectric model[43]	CHARMM[21]
RDIE_ELEC	Electrostatic interaction as computed in distance-dependent dielectric model[43]	CHARMM[21]
RDIE_VDW	Van der Waals interaction as computed in distance-dependent dielectric model[43]	CHARMM[21]
RDIE_TOT	Total interaction energy as computed in distance-dependent dielectric model[43]	CHARMM[21]
OMM_vacuum	Total interaction energy in vacuum[43]	OpenMM[52]
OMM_HCT	Total interaction energy with the HCT implicit solvent model[48]	OpenMM[52]
OMM_OBC1	Total interaction energy with the OBC1 implicit solvent model[49]	OpenMM[52]
OMM_OBC2	Total interaction energy with the OBC2 implicit solvent model[49]	OpenMM[52]
OMM_GBn	Total interaction energy with the GBn implicit solvent model[50]	OpenMM[52]
OMM_GBn2	Total interaction energy with the GBn2 implicit solvent model[51]	OpenMM[52]

Name	Description	Software
FoldX	FoldX[10,39] binding score computed with the AnalyseComplex tool after RepairPDB	FoldX[10,39]
FoldX_backbone_hbond	Backbone hydrogen bond component of the FoldX binding score	FoldX[10,39]
FoldX_sidechain_hbond	Sidechain hydrogen bond component of the FoldX binding score	FoldX[10,39]
FoldX_vdw	Van der Waals component of the FoldX binding score	FoldX[10,39]
FoldX_elec	Electrostatic component of the FoldX binding score	FoldX[10,39]
FoldX_solvation_polar	Polar solvation component of the FoldX binding score	FoldX[10,39]
FoldX_solvation_hydrophobic	Hydrophobic solvation component of the FoldX binding score	FoldX[10,39]
FoldX_entropy_sidechain	Sidechain entropy component of the FoldX binding score	FoldX[10,39]
FoldX_entropy_mainchain	Main chain entropy component of the FoldX binding score	FoldX[10,39]
Rosetta_dg	Rosetta binding score computed with the InterfaceAnalyzer tool[40] on a relaxed[41] structure	Rosetta[17,18]
Rosetta_sasa	SASA component of the Rosetta binding score	Rosetta[17,18]
Rosetta_hbond	Hydrogen bond component of the Rosetta binding score	Rosetta[17,18]
ENM_R6	Entropy from an Elastic Network Model where the force constant decreases proportionally to sixth power of the distance[56]	Native (numpy)
ENM_EXP	Entropy from an Elastic Network Model where the force constant decreases exponentially with the distance[57]	Native (numpy)
Prodigy_IC_NIS	Prodigy[26] scoring function for protein-protein itneractions.	MDTraj[58], Biopython[59]

Implementation, dependencies & parallelization

The ppdx code is implemented as a python library with two main functions provided to the end user: eval_descriptors and eval_pkl. The first is used to compute a chosen subset of descriptors for a list of protein-protein complexes. The second uses as inputs a list of protein-protein complexes and a previously generated regression or machine learning model and evaluates the trained protein-protein property as a function of the descriptors computed from the model coordinates; see Figure 1. To run ppdx either Modeller or Rosetta is needed to generate the models, and CHARMM is needed for energy minimization. Some descriptors can be computed using python modules, others require external executables; see Table 1. When generating models and computing descriptors, each task is independent, allowing trivial parallelization. Three execution modes are available: (1) serial, which does not use any parallelization, useful for testing and debugging; (2) using the Pool object in the multiprocessing Python library to

parallelize the workload on a single workstation; and (3) using the Parsl[61] library to upscale to multiple compute nodes, like on a supercomputer. The code is modular so that new protocols to generate structures, new descriptors, or new engines can be easily added. ppdx is freely available on GitHub, and contributions are welcome.

Protein-Protein complexes dataset & Machine Learning modeling

A test set of 350 protein-protein complexes was used for benchmarking and validation. This dataset consists of 14 antibodies targeting the stem of the influenza hemagglutinin protein, each bound to 25 different influenza virus strains. Experimental IC_{50} values are available for all 350 pairs[62]. Crystallographic structures 5JW4 and 3ZTJ are used as templates for the modeling. Models were generated using all four protocols, and all descriptors, except Firedock**,** were computed for all of them. We could not test Firedock (13 descriptors in total) with this dataset, due to their size (number of amino acids) which is too large for Firedock. The computed descriptors were used to train a machine learning model to reproduce experimental IC_{50} values (see Results). The template structures, the antibody and antigen sequences, and example scripts to train the machine learning model are available in the distributed software package.

Further examples

In the distributed software package, there are four examples of application of ppdx, in addition to the modeling of Influenza/Antibodies interaction described here. The first is an introductory example on how to model the barnase/barstar protein-protein complex[63] and how to compute the descriptors. Then there are two examples related to modeling HIV/Antibody interactions: the first is a benchmark of scoring functions for modeling the binding of the VRC01 antibody to HIV antigens[5,64]; the second is to compute the breadth of broadly neutralizing antibodies against HIV[11,15]. The last example consists of a subset of protein-protein interactions from the curated SKEMPI database[65]. We selected only protein-protein complexes with isothermal titration calorimetry (ITC) data. From these data we have the free energy, enthalpy, and entropy of binding. These examples are provided for the benefit of the user and are not discussed further in this work.

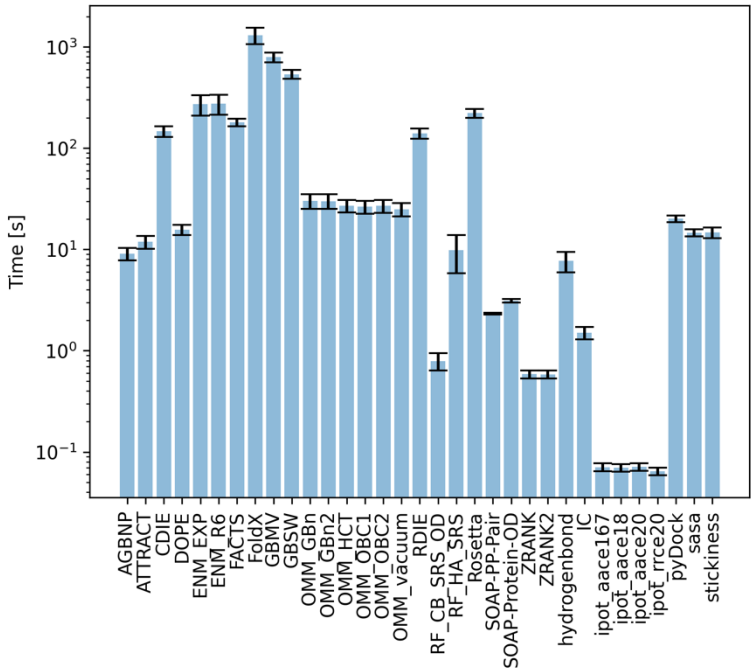

Figure 2. Average time, in seconds, needed to score one model with the different scoring functions. One or more descriptors are computed by each item in the bar plot, so the number of items is less than the total number of descriptors. Note the time is reported in logarithmic scale.

Results

For testing and benchmarking of the software, all descriptors, except Firedock (see Methods), were computed for a dataset of 350 protein-protein complexes using the four protocols to generate the models. Twenty models were generated for each complex. In this section some analysis and comparisons are performed on these data. All data and the analysis code are available in the distributed software package.

Timing

The time needed to generate the structural models depends on the protocol used. Modeller with the modeller_veryfast needs, on average, 1.9 minutes per model of the antigen/antibody (Fab) complex. With the modeller_fast and modeller_slow protocols, the time increases to 3.7

and 10.3 minutes, respectively. The protocol implemented using Rosetta requires a significantly higher computational time of about 3 hours. It is not possible to say *a priori* which protocol is best, as that depends on the descriptors computed afterwards and the particular property chosen for prediction. The time to evaluate the descriptors is highly variable, ranging from tenths of a second to tens of minutes, depending on the descriptors; see Figure 2. The times to generate a model and to compute the descriptors also depend on the size of the proteins.

Model similarity & descriptor convergences

For each protein-protein complex under study, several models are generated, and the values of the descriptors are averaged in order to reduce statistical errors in the results. This raises the question of the number of models needed for the average value of a given descriptor to converge; see Figure 3. Examples of descriptors that converge rapidly are the BSA-based ones or ZRANK. Descriptors that need more models to converge are some implicit solvents, while other, e.g., AGBNP, GBMV_POL, or SOAP-Protein-OD do not appear to converge over the 20-model sample; see Figure 3. The reason for the different behaviors is the sensitivity of the descriptor to the atomic coordinates in the models. For example, in general it is difficult to get convergence of the electrostatic and implicit solvent descriptors, which are very sensitive to the position of charged atoms in the models.

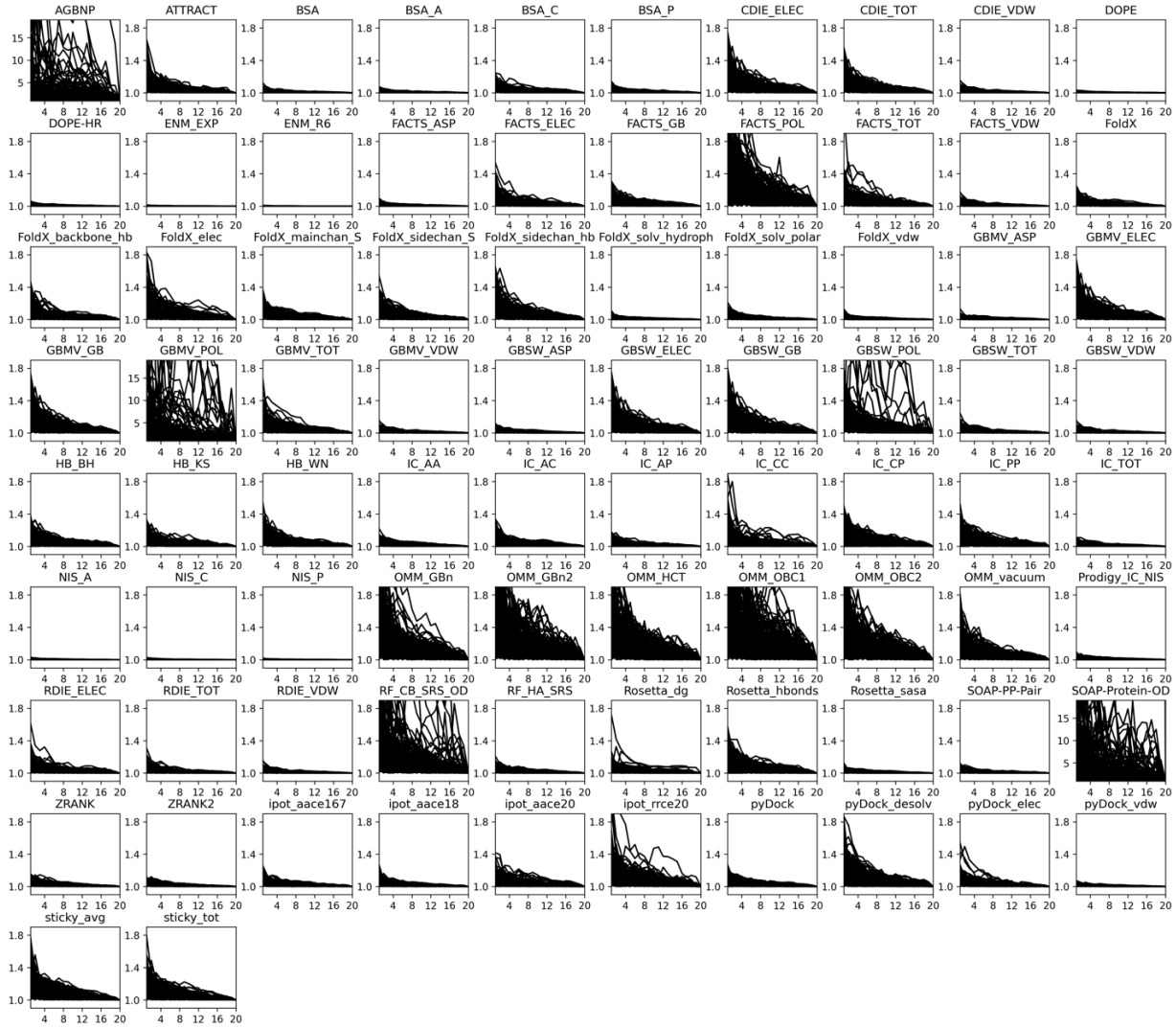

Figure 3. Convergence of each descriptor as a function of the number of models (generate with the modeller_fast protocol). Each plot is a different descriptor, and each trace is a different protein-protein complex. The absolute value of the averages at each number of models are normalized by the average at 20 models. A value of, e.g., 1.4 at 2 models, signifies the average is 40% higher (or lower) than the average computed over all 20 models. All plots use the same range in the vertical axis, except AGBNP, GBMV_POL, and SOAP-Protein-OD, for which the scale is tenfold bigger to show the much slower convergence. Higher resolution image is available in the software repository.

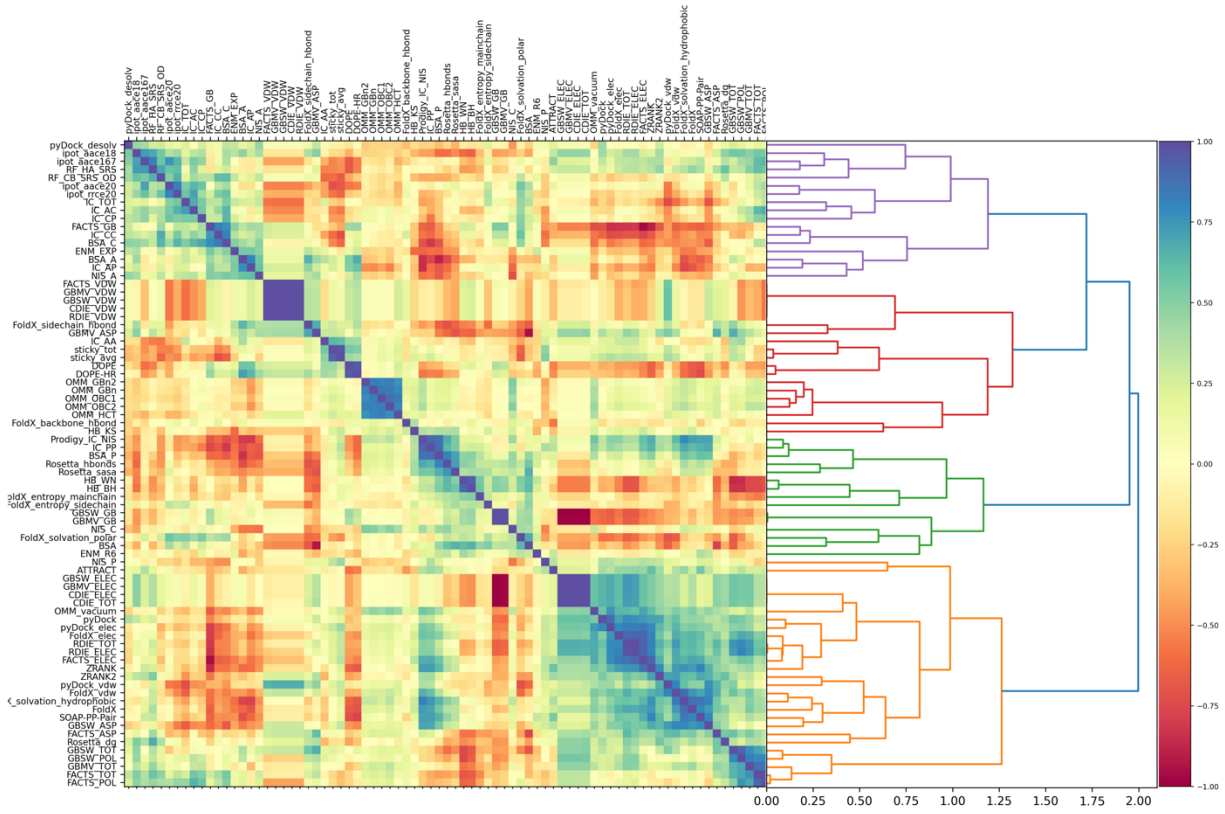

Figure 4. On the left side, cross correlation between each pair of descriptors. Upper and lower triangular matrix contains the Pearson and Spearman correlation coefficients, respectively. Blue and red correspond to correlation and anticorrelation, respectively. On the right side, hierarchical clustering of the descriptors based on their cross-correlations. Higher resolution images are available in the software repository.

Descriptor cross-correlations

Given the similar basis of many of the descriptors described above, cross-correlations among them are expected. This is confirmed in Figure 4A, where the Pearson and Spearman correlation coefficients are reported for each pair of descriptors (in the upper and lower triangular portion of the matrix, respectively). For example, there is high correlation among descriptors for the electrostatic interaction energy, which are anticorrelated to the GB implicit solvent energy. This is expected because the GB polar solvation energy acts to reduce Coulombic interactions via solvent screening. There is correlation also among statistical

potentials. Descriptors can also be clustered on the basis of their correlations, which could help select uncorrelated subsets to reduce the model dimension; see Figure 4B. A related and very interesting question is whether a small subset of descriptors can capture the variability in the full set. This can be explored by Principal Component Analysis (PCA), in which the largest principal values (variances) correspond to the most important components; see Figure 5. Most of the total variance (about 90%) is explained by about ten principal components; the explained variance rises rapidly, then forms a slowly converging tail at higher values. Overall, the data suggest that around ten principal components can capture most of the information in the descriptors.

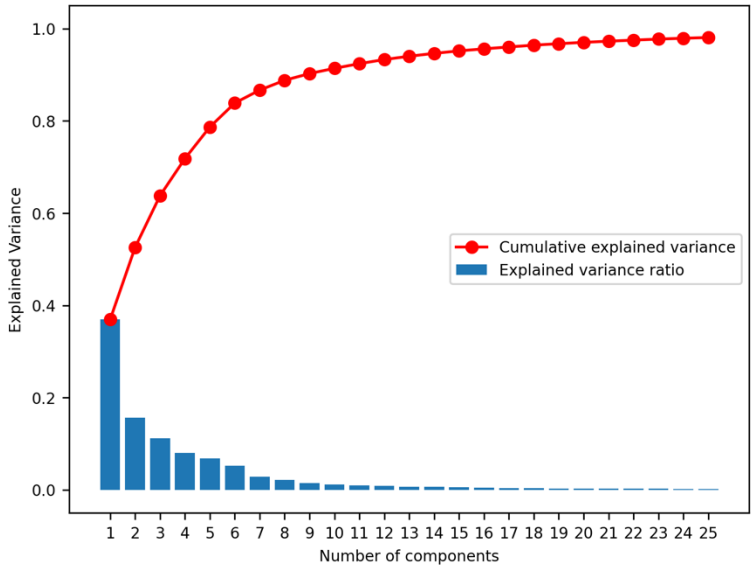

Figure 5. Fraction and cumulative explained variance for the sorted, most important, components from the principal component analysis (PCA) of all descriptors.

Correlations across protocols

It is also useful to compare descriptor values obtained with different modeling protocols. The underlying questions are whether different protocols generate significantly different models and how sensitive the descriptors are to such differences. In Figure 6, the Pearson correlation coefficients are shown for a selection of descriptors (chosen to minimize the required

computational time), and for each pair of modeling protocols. Overall, the majority of descriptors correlate significantly across protocols, e.g., the BSA, NIS and iPot descriptors, but some, e.g., Rosetta_dg or IC_TOT, show anticorrelations. The descriptors that are sensitive to atomic positions (e.g., the implicit solvation energies) also tend to be sensitive to the protocol used with small cross-correlations. Some descriptors show more complex behavior: the modeller_veryfast protocol shows very little correlation with modeller_fast, modeller_slow or Rosetta, while the latter three are correlated with each other. Looking across protocols, the strongest correlation is between modeller_fast and modeller_slow, suggesting these two protocols are generating similar ensembles of structures. The ZRANK and iPot descriptors exhibit a software dependence on the correlations; they are high across the Modeller protocols but significantly decreased when compared to Rosetta.

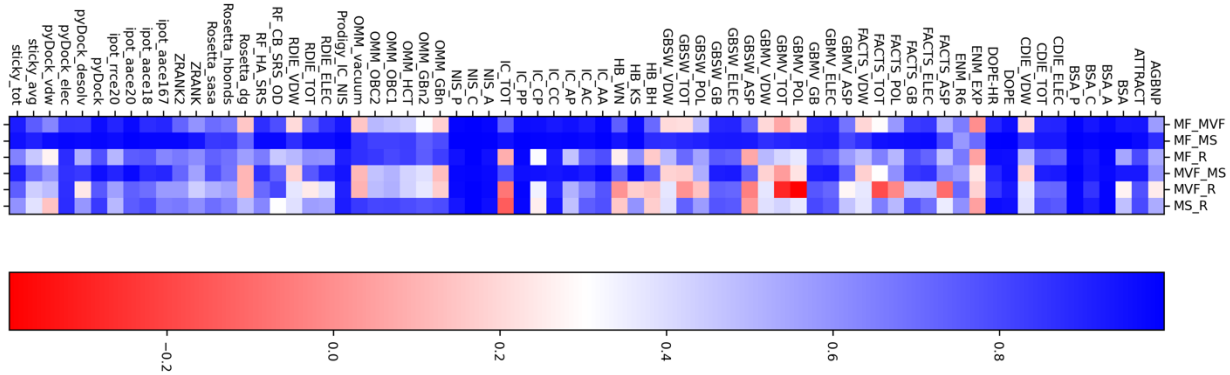

Figure 6. Pearson correlations coefficients of the descriptors computed with structures generated by different protocols. Each column is a descriptor, and each row is for a pair of protocols (MVF: modeller_veryfast, MF: modeller_fast, MS: modeller_slow, R: rosetta). Values close to one (blue) correspond to a perfect correlation, close to zero, or less, (red), correspond to no correlation or slight anticorrelation.

Machine learning modeling

As an example, the descriptors computed by ppdx were used to train a machine learning model to predict the experimental IC_{50} values for 350 influenza antigen/antibody complexes (more precisely, the pIC_{50} were computed). We chose to use a random forest model[66,67], due to its ability to provide an estimate of the importance of each descriptor in the model[15]. For the purpose of this demonstration, we used the modeller_veryfast protocol to generate the structures. To estimate which descriptors would be most informative for the pIC_{50} model, we trained models using just one descriptor, or, starting from the full list, we removed the least important descriptor, and retrained the model, until only one descriptor remained. These tests showed that descriptors based on the buried surface area were most important; therefore, we chose the following seven: BSA, BSA_A, BSA_P, BSA_C, NIS_A, NIS_P, NIS_C. While some of these correlate with each other, all seven are computed simultaneously using MDTraj[58], which minimizes the computational costs. For the final random forest model, the number of trees in the forest, the maximum depth of the tree, and the minimum number of samples required to be at a leaf node, were optimized by a grid-search over the parameter space using five-fold cross-validation. Model training was performed using a random 50% sample of the available data, while the remaining 50% were used as validation. Due to the limited amount of data available, significant differences are obtained depending on the random split into training and validation sets. The Pearson correlation coefficient of the validation set (against the experimental values) for multiple random splits, and retraining, of the random forest model was, on average, 0.53±0.12 (at 95% confidence interval). The correlation between experimental and computed values for one of the best "cherry-picked" models is 0.71±0.10 (at 95% confidence interval); see Figure 7. The error in the Pearson correlation coefficient in a given model was estimated by bootstrapping (with replacement) the predicted values. For the model in Figure 7, the standard deviation in the Pearson correlation coefficient is 0.05. Correlations in the training set were generally very high (>0.9).

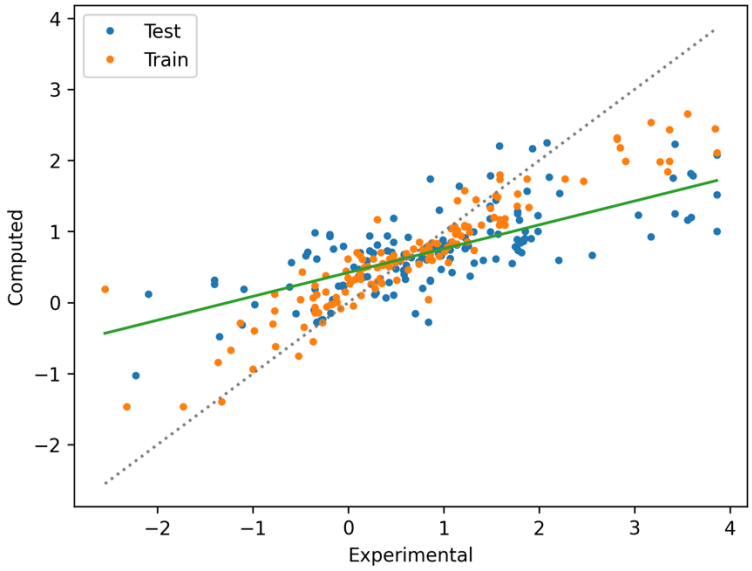

Figure 7. Correlation between the experimental and computed IC_{50} values for the random forest model with the highest expected Pearson correlation coefficient. The training set is in orange and the validation set is in blue. The green line corresponds to the linear regression in the test set. The Pearson correlation coefficient in the test set is 0.71±0.10 (at 95% confidence interval).

Concluding Discussion

This paper describes ppdx, a python package to model properties of protein-protein complexes. Multiple protocols are implemented to generate the all-atom structures of the complexes from which a large number of descriptors can be computed and used to train machine learning models. Properties of the currently implemented descriptors are described and analyzed in terms of their cross-correlation and sensitivity to the choice of protocols to generate the complex structures. The software is modular, and easily extensible. It is highly parallel, able to run on multiple CPU cores and multiple nodes on supercomputers. An example application to train a machine learning model to predict pIC_{50} values for 350 influenza antigen/antibody complexes using descriptors computed by ppdx yielded an average correlation coefficient of 0.53±0.12 for the experimental dataset, with the best model reaching a correlation coefficient of 0.71. While these correlation values are already significant, optimization is likely to lead to

further improvements. For example, while we generated a 20-model ensemble for each complex from which descriptor values were computed as ensemble averages, the number of models could be increased to better sample the conformational space of the complexes. Moreover, when computing the averages, an influential factor is the choice of the average, e.g., a simple average (as done here) vs. a weighted average (e.g., with Boltzmann weights). The answer would depend on the kind of distribution generated by the modeling protocol, which is not known a priori, unless one uses sophisticated methods such as molecular dynamics or Monte Carlo simulations to sample from a true thermodynamic ensemble. For practical reasons, the median value could also be a valid option, e.g., to protect from outliers arising from faulty models, or glitches in the scoring functions, which can be difficult to detect in high throughput scenarios. When comparing descriptors computed on structures generated by different modeling protocols, we also noted that different degrees of correlation exist with some descriptors particularly susceptible to the modeling protocol. This implies that, when describing a scoring function based on descriptors computed for protein-protein complex structures, it is important to specify how the complex models are constructed, i.e., the method of model generation is in itself a component of the scoring function. Our hope is that ppdx will motivate such explorations, facilitate the development of new scoring functions in machine learning methods, and, more generally, simplify the study of protein-protein interactions.

Acknowledgments

Financial support for this project was provided by the Bill & Melinda Gates Foundation and Flu Lab under grant opportunity OPP1214161, and by the CHARMM Development Project. Computer resources were provided by the National Energy Resource Scientific Computing Center (NERSC), which was supported by the Office of Science of the US Department of Energy under contract no. DE-AC02-05CH11231, and by the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under contract DE- AC05-00OR22725. The findings and conclusions contained within are those of the authors and do not necessarily reflect positions or policies of the Bill and Melinda Gates Foundation. Under the grant

conditions of the Foundation, a Creative Commons Attribution 4.0 Generic License has already been assigned to the Author Accepted Manuscript version that might arise from this submission.

Data availability

The ppdx source code and all examples and analyses are available online at https://github.com/simonecnt/ppdx

References

(1) Kastritis, P. L.; Bonvin, A. M. J. J. On the Binding Affinity of Macromolecular Interactions: Daring to Ask Why Proteins Interact. *J. R. Soc. Interface* **2013**, *10* (79), 20120835. https://doi.org/10.1098/rsif.2012.0835.

(2) Geng, C.; Xue, L. C.; Roel-Touris, J.; Bonvin, A. M. J. J. Finding the ΔΔG Spot: Are Predictors of Binding Affinity Changes upon Mutations in Protein–Protein Interactions Ready for It? *WIREs Comput. Mol. Sci.* **2019**, *9* (5), e1410. https://doi.org/10.1002/wcms.1410.

(3) Wang, J.; Deng, Y.; Roux, B. Absolute Binding Free Energy Calculations Using Molecular Dynamics Simulations with Restraining Potentials. *Biophys. J.* **2006**, *91* (8), 2798–2814. https://doi.org/10.1529/biophysj.106.084301.

(4) Chodera, J. D.; Mobley, D. L.; Shirts, M. R.; Dixon, R. W.; Branson, K.; Pande, V. S. Alchemical Free Energy Methods for Drug Discovery: Progress and Challenges. *Curr. Opin. Struct. Biol.* **2011**, *21* (2), 150–160. https://doi.org/10.1016/j.sbi.2011.01.011.

(5) Clark, A. J.; Gindin, T.; Zhang, B.; Wang, L.; Abel, R.; Murret, C. S.; Xu, F.; Bao, A.; Lu, N. J.; Zhou, T.; Kwong, P. D.; Shapiro, L.; Honig, B.; Friesner, R. A. Free Energy Perturbation Calculation of Relative Binding Free Energy between Broadly Neutralizing Antibodies and the Gp120 Glycoprotein of HIV-1. *J. Mol. Biol.* **2017**, *429* (7), 930–947. https://doi.org/10.1016/j.jmb.2016.11.021.

(6) Pirhadi, S.; Shiri, F.; B. Ghasemi, J. Multivariate Statistical Analysis Methods in QSAR. *RSC Adv.* **2015**, *5* (127), 104635–104665. https://doi.org/10.1039/C5RA10729F.

(7) Danishuddin; Khan, A. U. Descriptors and Their Selection Methods in QSAR Analysis: Paradigm for Drug Design. *Drug Discov. Today* **2016**, *21* (8), 1291–1302. https://doi.org/10.1016/j.drudis.2016.06.013.

(8) Vreven, T.; Hwang, H.; Pierce, B. G.; Weng, Z. Prediction of Protein–Protein Binding Free Energies. *Protein Sci.* **2012**, *21* (3), 396–404. https://doi.org/10.1002/pro.2027.

(9) Sirin, S.; Apgar, J. R.; Bennett, E. M.; Keating, A. E. AB-Bind: Antibody Binding Mutational Database for Computational Affinity Predictions. *Protein Sci.* **2016**, *25* (2), 393–409. https://doi.org/10.1002/pro.2829.

(10) Guerois, R.; Nielsen, J. E.; Serrano, L. Predicting Changes in the Stability of Proteins and Protein Complexes: A Study of More Than 1000 Mutations. *J. Mol. Biol.* **2002**, *320* (2), 369–387.

(11) Sprenger, K. G.; Conti, S.; Ovchinnikov, V.; Chakraborty, A. K.; Karplus, M. Multiscale Affinity Maturation Simulations to Elicit Broadly Neutralizing Antibodies against HIV. **2022**.

(12) Butler, K. T.; Davies, D. W.; Cartwright, H.; Isayev, O.; Walsh, A. Machine Learning for Molecular and Materials Science. *Nature* **2018**, *559* (7715), 547–555.

(13) Mitchell, J. B. O. Machine Learning Methods in Chemoinformatics. *WIREs Comput. Mol. Sci.* **2014**, *4* (5), 468–481. https://doi.org/10.1002/wcms.1183.

(14) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, É. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825−2830.

(15) Conti, S.; Karplus, M. Estimation of the Breadth of CD4bs Targeting HIV Antibodies by Molecular Modeling and Machine Learning. *PLOS Comput. Biol.* **2019**, *15* (4), e1006954. https://doi.org/10.1371/journal.pcbi.1006954.
(16) Webb, B.; Sali, A. Comparative Protein Structure Modeling Using MODELLER. In *Current Protocols in Bioinformatics*; John Wiley & Sons, Inc., 2002.
(17) Leaver-Fay, A.; Tyka, M.; Lewis, S. M.; Lange, O. F.; Thompson, J.; Jacak, R.; Kaufman, K.; Renfrew, P. D.; Smith, C. A.; Sheffler, W.; Davis, I. W.; Cooper, S.; Treuille, A.; Mandell, D. J.; Richter, F.; Ban, Y.-E. A.; Fleishman, S. J.; Corn, J. E.; Kim, D. E.; Lyskov, S.; Berrondo, M.; Mentzer, S.; Popović, Z.; Havranek, J. J.; Karanicolas, J.; Das, R.; Meiler, J.; Kortemme, T.; Gray, J. J.; Kuhlman, B.; Baker, D.; Bradley, P. Rosetta3: An Object-Oriented Software Suite for the Simulation and Design of Macromolecules. *Methods Enzymol.* **2011**, *487*, 545–574.
(18) Song, Y.; DiMaio, F.; Wang, R. Y.-R.; Kim, D.; Miles, C.; Brunette, T.; Thompson, J.; Baker, D. High-Resolution Comparative Modeling with RosettaCM. *Structure* **2013**, *21* (10), 1735–1742.
(19) Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; Bridgland, A.; Meyer, C.; Kohl, S. A. A.; Ballard, A. J.; Cowie, A.; Romera-Paredes, B.; Nikolov, S.; Jain, R.; Adler, J.; Back, T.; Petersen, S.; Reiman, D.; Clancy, E.; Zielinski, M.; Steinegger, M.; Pacholska, M.; Berghammer, T.; Silver, D.; Vinyals, O.; Senior, A. W.; Kavukcuoglu, K.; Kohli, P.; Hassabis, D. Applying and Improving AlphaFold at CASP14. *Proteins Struct. Funct. Bioinforma.* **2021**, *89* (12), 1711–1721. https://doi.org/10.1002/prot.26257.
(20) Yang, J.; Yan, R.; Roy, A.; Xu, D.; Poisson, J.; Zhang, Y. The I-TASSER Suite: Protein Structure and Function Prediction. *Nat. Methods* **2015**, *12* (1), 7–8. https://doi.org/10.1038/nmeth.3213.
(21) Brooks, B. R.; Brooks, C. L.; Mackerell, A. D.; Nilsson, L.; Petrella, R. J.; Roux, B.; Won, Y.; Archontis, G.; Bartels, C.; Boresch, S.; Caflisch, A.; Caves, L.; Cui, Q.; Dinner, A. R.; Feig, M.; Fischer, S.; Gao, J.; Hodoscek, M.; Im, W.; Kuczera, K.; Lazaridis, T.; Ma, J.; Ovchinnikov, V.; Paci, E.; Pastor, R. W.; Post, C. B.; Pu, J. Z.; Schaefer, M.; Tidor, B.; Venable, R. M.; Woodcock, H. L.; Wu, X.; Yang, W.; York, D. M.; Karplus, M. CHARMM: The Biomolecular Simulation Program. *J. Comput. Chem.* **2009**, *30* (10), 1545–1614.
(22) Baker, E. N.; Hubbard, R. E. Hydrogen Bonding in Globular Proteins. *Prog. Biophys. Mol. Biol.* **1984**, *44* (2), 97–179.
(23) Wernet, P.; Nordlund, D.; Bergmann, U.; Cavalleri, M.; Odelius, M.; Ogasawara, H.; Näslund, L. Å.; Hirsch, T. K.; Ojamäe, L.; Glatzel, P.; Pettersson, L. G. M.; Nilsson, A. The Structure of the First Coordination Shell in Liquid Water. *Science* **2004**, *304* (5673), 995–999.
(24) Kabsch, W.; Sander, C. Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen-Bonded and Geometrical Features. *Biopolymers* **1983**, *22* (12), 2577–2637.
(25) Shrake, A.; Rupley, J. A. Environment and Exposure to Solvent of Protein Atoms. Lysozyme and Insulin. *J. Mol. Biol.* **1973**, *79* (2), 351–371.
(26) Vangone, A.; Bonvin, A. M. Contacts-Based Prediction of Binding Affinity in Protein–Protein Complexes. *eLife* **2015**, *4*, e07454.
(27) Levy, E. D.; De, S.; Teichmann, S. A. Cellular Crowding Imposes Global Constraints on the Chemistry and Evolution of Proteomes. *Proc. Natl. Acad. Sci.* **2012**, *109* (50), 20461–20466.
(28) Pierce, B.; Weng, Z. ZRANK: Reranking Protein Docking Predictions with an Optimized Energy Function. *Proteins Struct. Funct. Bioinforma.* **2007**, *67* (4), 1078–1086.
(29) Pierce, B.; Weng, Z. A Combination of Rescoring and Refinement Significantly Improves Protein Docking Performance. *Proteins Struct. Funct. Bioinforma.* **2008**, *72* (1), 270–279.
(30) Mintseris, J.; Pierce, B.; Wiehe, K.; Anderson, R.; Chen, R.; Weng, Z. Integrating Statistical Pair Potentials into Protein Complex Prediction. *Proteins Struct. Funct. Bioinforma.* **2007**, *69* (3), 511–520.
(31) 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. Bioinforma.* **2007**, *68* (2), 503–515.
(32) Zacharias, M. ATTRACT: Protein–Protein Docking in CAPRI Using a Reduced Protein Model. *Proteins Struct. Funct. Bioinforma.* **2005**, *60* (2), 252–256.
(33) Andrusier, N.; Nussinov, R.; Wolfson, H. J. FireDock: Fast Interaction Refinement in Molecular Docking. *Proteins Struct. Funct. Bioinforma.* **2007**, *69* (1), 139–159.

(34) Rykunov, D.; Fiser, A. Effects of Amino Acid Composition, Finite Size of Proteins, and Sparse Statistics on Distance-Dependent Statistical Pair Potentials. *Proteins Struct. Funct. Bioinforma.* **2007**, *67* (3), 559–568.
(35) Rykunov, D.; Fiser, A. New Statistical Potential for Quality Assessment of Protein Models and a Survey of Energy Functions. *BMC Bioinformatics* **2010**, *11*, 128.
(36) Anishchenko, I.; Kundrotas, P. J.; Vakser, I. A. Contact Potential for Structure Prediction of Proteins and Protein Complexes from Potts Model. *Biophys. J.* **2018**, *115* (5), 809–821.
(37) Dong, G. Q.; Fan, H.; Schneidman-Duhovny, D.; Webb, B.; Sali, A. Optimized Atomic Statistical Potentials: Assessment of Protein Interfaces and Loops. *Bioinformatics* **2013**, *29* (24), 3158–3166.
(38) Shen, M.; Sali, A. Statistical Potential for Assessment and Prediction of Protein Structures. *Protein Sci.* **2006**, *15* (11), 2507–2524.
(39) Schymkowitz, J.; Borg, J.; Stricher, F.; Nys, R.; Rousseau, F.; Serrano, L. The FoldX Web Server: An Online Force Field. *Nucleic Acids Res.* **2005**, *33*, W382–W388.
(40) Stranges, P. B.; Kuhlman, B. A Comparison of Successful and Failed Protein Interface Designs Highlights the Challenges of Designing Buried Hydrogen Bonds. *Protein Sci.* **2013**, *22* (1), 74–82.
(41) Nivón, L. G.; Moretti, R.; Baker, D. A Pareto-Optimal Refinement Method for Protein Design Scaffolds. *PLOS ONE* **2013**, *8* (4), e59004.
(42) Still, W. C.; Tempczyk, A.; Hawley, R. C.; Hendrickson, T. Semianalytical Treatment of Solvation for Molecular Mechanics and Dynamics. *J. Am. Chem. Soc.* **1990**, *112* (16), 6127–6129.
(43) Best, R. B.; Zhu, X.; Shim, J.; Lopes, P. E. M.; Mittal, J.; Feig, M.; MacKerell, A. D. Optimization of the Additive CHARMM All-Atom Protein Force Field Targeting Improved Sampling of the Backbone φ, ψ and Side-Chain X1 and X2 Dihedral Angles. *J. Chem. Theory Comput.* **2012**, *8* (9), 3257–3273.
(44) Haberthür, U.; Caflisch, A. FACTS: Fast Analytical Continuum Treatment of Solvation. *J. Comput. Chem.* **2008**, *29* (5), 701–715.
(45) Lee, M. S.; Salsbury, F. R.; Brooks III, C. L. Novel Generalized Born Methods. *J. Chem. Phys.* **2002**, *116* (24), 10606–10614.
(46) Lee, M. S.; Feig, M.; Salsbury, F. R.; Brooks, C. L. New Analytic Approximation to the Standard Molecular Volume Definition and Its Application to Generalized Born Calculations. *J. Comput. Chem.* **2003**, *24* (11), 1348–1356.
(47) Im, W.; Lee, M. S.; Brooks, C. L. Generalized Born Model with a Simple Smoothing Function. *J. Comput. Chem.* **2003**, *24* (14), 1691–1702.
(48) Hawkins, G. D.; Cramer, C. J.; Truhlar, D. G. Parametrized Models of Aqueous Free Energies of Solvation Based on Pairwise Descreening of Solute Atomic Charges from a Dielectric Medium. *J. Phys. Chem.* **1996**, *100* (51), 19824–19839.
(49) Onufriev, A.; Bashford, D.; Case, D. A. Exploring Protein Native States and Large-Scale Conformational Changes with a Modified Generalized Born Model. *Proteins Struct. Funct. Bioinforma.* **2004**, *55* (2), 383–394.
(50) Mongan, J.; Simmerling, C.; McCammon, J. A.; Case, D. A.; Onufriev, A. Generalized Born Model with a Simple, Robust Molecular Volume Correction. *J. Chem. Theory Comput.* **2007**, *3* (1), 156–169.
(51) Nguyen, H.; Roe, D. R.; Simmerling, C. Improved Generalized Born Solvent Model Parameters for Protein Simulations. *J. Chem. Theory Comput.* **2013**, *9* (4), 2020–2034.
(52) Eastman, P.; Swails, J.; Chodera, J. D.; McGibbon, R. T.; Zhao, Y.; Beauchamp, K. A.; Wang, L.-P.; Simmonett, A. C.; Harrigan, M. P.; Stern, C. D.; Wiewiora, R. P.; Brooks, B. R.; Pande, V. S. OpenMM 7: Rapid Development of High Performance Algorithms for Molecular Dynamics. *PLOS Comput. Biol.* **2017**, *13* (7), e1005659.
(53) Gallicchio, E.; Levy, R. M. AGBNP: An Analytic Implicit Solvent Model Suitable for Molecular Dynamics Simulations and High-Resolution Modeling. *J. Comput. Chem.* **2004**, *25* (4), 479–499.
(54) Tirion, M. M. Large Amplitude Elastic Motions in Proteins from a Single-Parameter, Atomic Analysis. *Phys. Rev. Lett.* **1996**, *77* (9), 1905–1908.
(55) Hinsen, K. Analysis of Domain Motions by Approximate Normal Mode Calculations. *Proteins Struct. Funct. Bioinforma.* **1998**, *33* (3), 417–429.
(56) Hinsen, K.; Petrescu, A.-J.; Dellerue, S.; Bellissent-Funel, M.-C.; Kneller, G. R. Harmonicity in Slow Protein Dynamics. *Chem. Phys.* **2000**, *261* (1), 25–37.

(57) Atilgan, A. R.; Durell, S. R.; Jernigan, R. L.; Demirel, M. C.; Keskin, O.; Bahar, I. Anisotropy of Fluctuation Dynamics of Proteins with an Elastic Network Model. *Biophys. J.* **2001**, *80* (1), 505–515.
(58) McGibbon, R. T.; Beauchamp, K. A.; Harrigan, M. P.; Klein, C.; Swails, J. M.; Hernández, C. X.; Schwantes, C. R.; Wang, L.-P.; Lane, T. J.; Pande, V. S. MDTraj: A Modern Open Library for the Analysis of Molecular Dynamics Trajectories. *Biophys. J.* **2015**, *109* (8), 1528–1532.
(59) Cock, P. J. A.; Antao, T.; Chang, J. T.; Chapman, B. A.; Cox, C. J.; Dalke, A.; Friedberg, I.; Hamelryck, T.; Kauff, F.; Wilczynski, B.; de Hoon, M. J. L. Biopython: Freely Available Python Tools for Computational Molecular Biology and Bioinformatics. *Bioinformatics* **2009**, *25* (11), 1422–1423. https://doi.org/10.1093/bioinformatics/btp163.
(60) Word, J. M.; Lovell, S. C.; Richardson, J. S.; Richardson, D. C. Asparagine and Glutamine: Using Hydrogen Atom Contacts in the Choice of Side-Chain Amide Orientation11Edited by J. Thornton. *J. Mol. Biol.* **1999**, *285* (4), 1735–1747.
(61) Babuji, Y.; Woodard, A.; Li, Z.; Katz, D. S.; Clifford, B.; Kumar, R.; Lacinski, L.; Chard, R.; Wozniak, J. M.; Foster, I.; Wilde, M.; Chard, K. Parsl: Pervasive Parallel Programming in Python. In *Proceedings of the 28th International Symposium on High-Performance Parallel and Distributed Computing*; HPDC '19; Association for Computing Machinery: New York, NY, USA, 2019; pp 25–36. https://doi.org/10.1145/3307681.3325400.
(62) Kallewaard, N. L.; Corti, D.; Collins, P. J.; Neu, U.; McAuliffe, J. M.; Benjamin, E.; Wachter-Rosati, L.; Palmer-Hill, F. J.; Yuan, A. Q.; Walker, P. A.; Vorlaender, M. K.; Bianchi, S.; Guarino, B.; Marco, A. D.; Vanzetta, F.; Agatic, G.; Foglierini, M.; Pinna, D.; Fernandez-Rodriguez, B.; Fruehwirth, A.; Silacci, C.; Ogrodowicz, R. W.; Martin, S. R.; Sallusto, F.; Suzich, J. A.; Lanzavecchia, A.; Zhu, Q.; Gamblin, S. J.; Skehel, J. J. Structure and Function Analysis of an Antibody Recognizing All Influenza A Subtypes. *Cell* **2016**, *166* (3), 596–608. https://doi.org/10.1016/j.cell.2016.05.073.
(63) Buckle, A. M.; Schreiber, G.; Fersht, A. R. Protein-Protein Recognition: Crystal Structural Analysis of a Barnase-Barstar Complex at 2.0-A Resolution. *Biochemistry* **1994**, *33* (30), 8878–8889.
(64) Zhou, T.; Georgiev, I.; Wu, X.; Yang, Z.-Y.; Dai, K.; Finzi, A.; Kwon, Y. D.; Scheid, J. F.; Shi, W.; Xu, L.; Yang, Y.; Zhu, J.; Nussenzweig, M. C.; Sodroski, J.; Shapiro, L.; Nabel, G. J.; Mascola, J. R.; Kwong, P. D. Structural Basis for Broad and Potent Neutralization of HIV-1 by Antibody VRC01. *Science* **2010**, *329* (5993), 811–817.
(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. https://doi.org/10.1093/bioinformatics/bty635.
(66) Ho, T. K. Random Decision Forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*; 1995; Vol. 1, pp 278–282.
(67) Svetnik, V.; Liaw, A.; Tong, C.; Culberson, J. C.; Sheridan, R. P.; Feuston, B. P. Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. *J. Chem. Inf. Comput. Sci.* **2003**, *43* (6), 1947–1958.