LARGE-SCALE PROTEASE MULTISPECIFICITY:

STRUCTURE-BASED PREDICTION AND FITNESS LANDSCAPE ANALYSIS

By

ALIZA BATYA RUBENSTEIN

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Quantitative Biomedicine

Written under the direction of

Professor Sagar D. Khare

And approved by

_____

_____

_____

_____

New Brunswick, New Jersey

January 2018

ABSTRACT OF THE DISSERTATION

Large-scale Protease Multispecificity:

Structure-based Prediction and Specificity Landscape Analysis

By ALIZA RUBENSTEIN

Dissertation Director:

Professor Sagar D. Khare

Proteases are ubiquitous and significant to both normal cellular functioning and disease states. They are generally multispecific, cleaving a set of substrates without recognizing other peptides. Computational methods to predict and design protease multispecificity would advance our understanding of the biophysical basis of protease specificity, enable the characterization of novel proteases, allow the identification of novel biological roles for proteases, elucidate protease specificity landscapes and ultimately further the design of custom proteases to serve as therapeutics or protein-level knockout reagents in cell culture.

Current methods of computational protease specificity prediction are limited in a variety of ways. Techniques to classify substrates as cleaved or uncleaved are constrained by the

quality of the input data, cannot be easily generalized to other proteases, and require large training data sets to learn correlations between substrate positions. Methods that predict specificity profiles are computationally expensive and thus unable to be used directly within design. While fitness landscapes have been explored experimentally and via low-resolution computational models, no methods have yet explored the full fitness landscape using chemically realistic atomic-resolution computations.

In this dissertation, we further the understanding of protease multispecificity via a variety of experimental and computational techniques that can be generalized to other proteases. First, we develop a structure-based classifier that distinguishes robustly between cleaved and uncleaved substrates, benchmark the classifier performance for five model proteases, and apply the classifier in a blind test to identify novel substrates. Second, we implement a mean-field structure-based algorithm (MFPred) to rapidly and accurately predict protease specificity profiles, benchmark MFPred performance on a range of protease and protein-recognition domains, and demonstrate that MFPred accurately predicts the impact of receptor-side mutations, thus showing putative utility in protease design. Third, we construct a specificity landscape of hepatitis C virus NS3 protease using both experimental and computational methods and find evidence for a structural basis of mutational robustness. Finally, we compare the Rosetta and Amber energy functions used in the computational prediction of protease multispecificity in a systematic benchmark.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Dr. Sagar Khare, for his guidance, feedback, and encouragement. From suggesting research directions, to guiding me through the nitty-gritty of implementation, to advising me in the intricacies of writing for publication, your supervision has been invaluable. I must also thank Dr. Gail Arnold for your assistance and encouragement. It is a delight (and a healthy boost to my self-esteem!) to meet with you, and your advice has regularly been on-target.

I would also like to thank my committee members, Dr. David Case, Dr. Vikas Nanda, and Dr. Richard Bonneau for your advice and feedback. Several chapters in this dissertation were written collaboratively, and I gratefully acknowledge those collaborators. The experimental work and most of the computational work in Chapter 2 was performed by Dr. Manasi Pethe; I assisted with data curation, Rosetta simulations, and parameter optimization. I mainly completed the research in Chapter 3, with assistance in the data curation by Dr. Manasi Pethe. For Chapter 4, the experimental work was performed by Dr. Manasi Pethe and I executed the computational work and quantitative analyses. In Chapter 5, Dr. Hai Nguyen and Kristin Blacklock performed the Amber simulations, Kristin Blacklock completed the loop modeling Rosetta simulations, and I executed the remaining simulations and analyses. Dr. David Case provided supervision and feedback. I must also acknowledge Dr. Tanja Kortemme, Dr. Shane O'Connor, Dr. Frank DiMaio, and Dr. Hahnbeom Park for giving us some of the datasets used in Chapter 5 of the dissertation and for helpful discussions related to those datasets.

My journey through graduate school would not have been the same without the companionship and camaraderie of the Khare lab. Nancy Hernandez, Kristin Blacklock, Dr. Lu Yang, William Hansen, Dr. Brahm Yachnin, Elliot Dolan, Dmitri Zorine, and Dr. Manasi Pethe – your laughter and advice were always invaluable. I must express my enormous appreciation to Dr. Manasi Pethe; we collaborated closely for most of my graduate career. A collaboration of that length could easily have been difficult, yet your friendship made it easy, fun, and a fantastic learning experience.

I would like to acknowledge the National Science Foundation Graduate Research Fellowship, Grant No. DGE-1433187, for funding part of my graduate career.

I would never have made the choice to attend graduate school without the guidance of various mentors in my undergraduate career. Thank you, Dr. Robert Fardon, for teaching me physics, math, relativity, and most of all, how to love learning. Thank you to Professor Shmuel Fink, Professor Miriam Plonczak, Professor Andrew Schwimmer, and Professor Abraham Grund of the Touro College Computer Science Department. I still remember my first assignments (fountain and registration system!) for the learning and enjoyment they inspired. Though not technically my mentors, my cohort in CS at Touro College certainly taught me much about computers and even more about friendship; I owe a sincere thank you to Devorah Jacob, Cyril Rosen, Baila Pilicer, and Sarah Aberbach. I must also thank Dr. Robert Bressler of the Touro College Biology Department. I truly enjoyed working on the Touro College Science Journal with you and appreciate your career guidance. Additionally, I would like to acknowledge Dr. Dina Sokol (Brooklyn College) and Dr. Richard Bodnar (Queens College) for allowing me to engage in research under your supervision.

Several women scientists have served as both mentors and role models to me. I would like to thank Dr. Elena Zaslavsky for introducing me to computational biology – if not for her guidance I may never have entered this field. Elena, your advice was invaluable at each stage of my career, and you were instrumental in helping me to find a postdoc; I hope our collaborations are as fruitful in the future. I would also like to acknowledge Dr. Dina Sokol, Dr. Shana Posy and Dr. Tamar Rosenbloom for encouragement and advice. You were always available to guide me in both the nitty-gritty of graduate school and the larger questions of career choices. Finally, I must express my gratitude to Dr. Ora Schueler-Furman; from your feedback on my current research to your gracious assistance in my postdoc application process to your schmoozing with my children during Skype meetings, you taught me by example how to be a truly supportive mentor. I still regret that my move to Israel did not work out, and hope that it will, someday soon.

Family is always helpful during intense schooling, and my family by acquisition has been no exception. I started graduate school shortly after marriage, and my in-laws graciously stepped up to assist me. Mommy, Tatty, and the rest of the crew - from short stints of babysitting to week-long shifts so that I could attend conferences, from advice about the trivialities of life to supporting us through more intense trials, your love and support is so appreciated.

My siblings have always been an inescapable part of my life, and of incomparable assistance to me throughout graduate school. I would like to particularly thank Yehuda, for babysitting and helping us move; Sarah, for helpful discussions and sympathy; Henny, for insightful advice and volunteering to watch my kids; and Peryl, for being both

a voice of reason and an always available listening ear. Finally, I must express my enormous gratitude and appreciation to my wonderful parents, for stepping back and giving me the freedom to grow and stretch new wings, while providing a safety net to catch me after the inevitable falls (Mommy, pardon my clichéd metaphors!). Your love, support, and encouragement helped us every step of the way.

To my children – giving birth to each of you throughout graduate school made the experience both more interesting and more challenging than I could ever have imagined. You have enriched my life, made me prioritize my time, and gave meaning to my days. I must thank Bracha Tzipora for distracting me whenever I got too obsessed with my research and Yisrael for being the computer pressing-button man whenever I looked like I was getting any work done. Menachem Michel, your due date was great motivation to finish two papers, so thank you for providing that deadline.

I must acknowledge one final person, my husband, Leiby. I was accepted at Rutgers while dating you and started mere months after our marriage. Though you initially weren't accustomed to cooking, cleaning, or childcare, you went out of your comfort zone to help me through the roller coaster ride of graduate school. Thank you for your physical, emotional, and moral support - without the pacing and self-care that you forced me to pursue and gave me the wherewithal to do so, I would have burned out long ago.

Finally, I would like to thank G-d, for guiding me every step of the way.

DEDICATION

To Leiby,
Words cannot express my gratitude to you.

Table of Contents

List of Tables

List of Illustrations:

Chapter 1. Introduction

1.1. Motivation: Proteases

1.1.1. Biological relevance

Proteases, enzymes that cleave the peptide bond, constitute about 2% of the human genome[1]. They are involved in crucial functions in the human body and normal cellular functioning, such as apoptosis, digestion, hemostasis, reproduction and the immune system[2]. Proteolytic cascades play important roles in blood coagulation, complement fixation, fibrinolysis, development, matrix remodeling, differentiation, and wound healing[2,3]. Besides their biological significance in humans, proteases often drive viral maturation by cleaving the viral polyprotein[4]. The specificity of a given protease is dictated by its structure and determines its function[2,4].

1.1.2. Protease diversity

1.1.2.1. Structure and binding

Proteases have a variety of different folded structures, which often relate to their mechanism of binding. Several different interactions mediate the binding of protease to substrate, such as shape complementarity[5–7], hydrogen bonding[8,9], and electrostatics[10–14]. The fold of the protease may affect the mechanism of binding; protease folds that include grooves to bind the substrate usually require shape complementarity for binding[9], whereas exposed active sites often rely on hydrogen bonds to bind protease and substrate. Electrostatic-based binding requires the substrate to be enriched in amino acids with the charge that is opposite that of the active site residues.

As the shape of the binding site pocket is often instrumental in binding, Schechter and Berger have developed nomenclature that reflects this importance. The seven amino acids on the N-terminal side of the scissile bond of the substrate are labeled as P7, P6, P5…P1, while the seven amino acids on the C-terminal side are labeled P1'…P7'. Similar nomenclature is employed for the protease binding site as S7…S1, S1'…S7. This convention is used extensively and is used throughout this dissertation.

1.1.2.2. Specificity profile shape

Proteases often interact with many interaction partners and thus have multispecificity. Therefore, protease specificity cannot be expressed as a simple consensus sequence; a specificity profile, which gives a probability distribution of amino acids at each substrate site, is a more accurate expression of the protease specificity. The shape of this specificity profile varies between proteases. Some are more stringent, or conserved, and the shape of the specificity profile is said to be peaked, while others allow for a range of amino acids and are considered as flat in shape. Often, the fold of the protease and related mechanism of binding affect the shape of the binding profile. Proteases that bind based on shape complementarity may only bind a few amino acids that fit the protease binding pocket, while those that bind via electrostatics or hydrogen bonding may bind a broader range of amino acids.

Beyond multispecificity as recognition for a set of peptides (positive specificity), proteases often exhibit non-recognition of another set of interaction partners (negative specificity). This multispecificity is often particularly important for viral proteases.

These proteases must recognize and cleave a range of substrates, often sites on the viral polyprotein that are necessary to cleave for viral maturation (positive specificity). Concurrently, they must not cleave all other sites within the viral polyprotein (negative specificity)[15]. This multispecificity is maintained despite the high mutational load shared by many viruses; the basis for this mutational robustness is not well understood.

1.2. Objective

In light of the ubiquity and biological significance of protease-peptide interactions, we attempt to further the prediction and design of protease multispecificity. We investigate the structural and biochemical rules that form the basis for protease-substrate interactions. These rules are then used to implement computational techniques to predict protease multispecificity. These methods include a classifier that predicts whether a given substrate is cleaved or uncleaved and a mean-field based algorithm that predicts the specificity profile for a given protease. Additionally, we use both experimental and computational techniques to construct a specificity landscape for hepatitis C virus (HCV) NS3 protease, which allows us to further our understanding of its specificity and mutational robustness.

Greater understanding of protease specificity should enable computational design of both proteases and substrates. The substrate classifier can be used to design novel substrates for known proteases, while the rapid mean-field algorithm may be used to design novel proteases to cleave a given specificity profile. Insights gained from the specificity landscape can also be used in the design of novel proteases.

1.3. Outline of the dissertation

As mentioned above, our goal is furthering the prediction and design of protease multispecificity. We begin with the development of a discriminatory biophysical structure-based scoring function that can be used to classify substrates as cleaved or uncleaved by a given protease. In the process of developing the function, we investigate the score-terms that are important to the structural prediction of specificity (Chapter 2). Next, we implement a rapid, accurate structure-based algorithm for specificity profile prediction (Chapter 3). The score-terms discovered in Chapter 2 are implicit within the algorithm. We then use experimental and computational techniques, including the SVM developed in Chapter 2, to explore the specificity landscape of the HCV NS3 protease, thus demonstrating the accuracy of our specificity prediction and enabling a deeper understanding of mutational robustness (Chapter 4). One inherent limitation of these computational techniques is the energy function used (Rosetta); in fact, to circumvent this limitation, we use score-terms from Amber along with the Rosetta energy function in Chapter 2. To further investigate the limitations and strengths of these energy functions, we compare Amber and Rosetta in a systematic benchmark (Chapter 5).

Chapter 2. Large-scale structure-based prediction and identification of novel protease substrates using computational protein design

Note: Reproduced with permission from Pethe MA, Rubenstein AB, Khare SD, Large-scale structure-based prediction and identification of novel protease substrates using computational protein design. 2017. **429**(2):220-236.

2.1. Abstract

Characterizing the substrate specificity of protease enzymes is critical for illuminating the molecular basis of their diverse and complex roles in a wide array of biological processes. Rapid and accurate prediction of their extended substrate specificity would also aid in the design of custom proteases capable of selectively and controllably cleaving biotechnologically or therapeutically relevant targets. However, current *in silico* approaches for protease specificity prediction, rely on, and are therefore limited by, machine learning of sequence patterns in known experimental data. Here, we describe a general approach for predicting peptidase substrates *de novo* using protein structure modeling and biophysical evaluation of enzyme-substrate complexes. We construct atomic resolution models of thousands of candidate substrate-enzyme complexes for each of five model proteases belonging to the four major protease mechanistic classes (serine-, cysteine-, aspartyl- and metallo-proteases) and develop a discriminatory scoring function using enzyme design modules from Rosetta and Amber-MMPBSA. We rank putative substrates based on calculated interaction energy with a modeled near-attack conformation of the enzyme active site. We show that the energetic patterns obtained from these simulations can be used to robustly rank and classify known cleaved and

uncleaved peptides and that these structural-energetic patterns have greater discriminatory power compared to purely sequence-based statistical inference. Combining sequence and energetic patterns using machine-learning algorithms further improves classification performance, and analysis of structural models provides physical insight into the structural basis for the observed specificities. We further tested the predictive capability of the model by designing and experimentally characterizing the cleavage of four novel substrate motifs for the Hepatitis C virus NS3/4 protease using an *in vivo* assay. The presented structure-based approach is generalizable to other protease enzymes with known or modeled structures, and complements existing experimental methods for specificity determination.

2.2. Introduction

Proteolytic cleavage is a ubiquitous post-translational modification that controls the transmission of biological information[2,16,17]. Proteases encompass a structurally and mechanistically diverse class of enzymes that display a range of cleavage specificities reflecting their complex and diverse biological roles[2,4,18,19]. For example, proteases involved in digestion and extracellular matrix degradation, e.g. trypsins and matrix metalloproteases, respectively, show relatively relaxed specificity profiles[20], whereas those involved in apoptotic and thrombolytic cascades, e.g. caspases[21] and thrombin[22], respectively, are more selective in their cleavage motifs. In many viruses, protease-mediated cleavage of the viral polyprotein at specific sites is crucial for viral maturation[23]; as a result, these enzymes are highly selective in cleaving only a small set of polypeptide sequences, while not acting on other sequences in the polyprotein.

Accordingly, these enzymes have been successful drug targets for developing anti-viral therapies[24,25]. Thus, proteases are exemplars of enzymatic multi-specificity, which have likely evolved to act upon and cleave a range of substrates – their specificity profile – while simultaneously avoiding the cleavage of other substrates[15] (Figure 2.1D). Modeling of protease substrate specificity would illuminate the structural and physiochemical basis of these observed positive and negative selectivities, and aid protease biology by identifying novel substrates and biological roles of proteolysis.

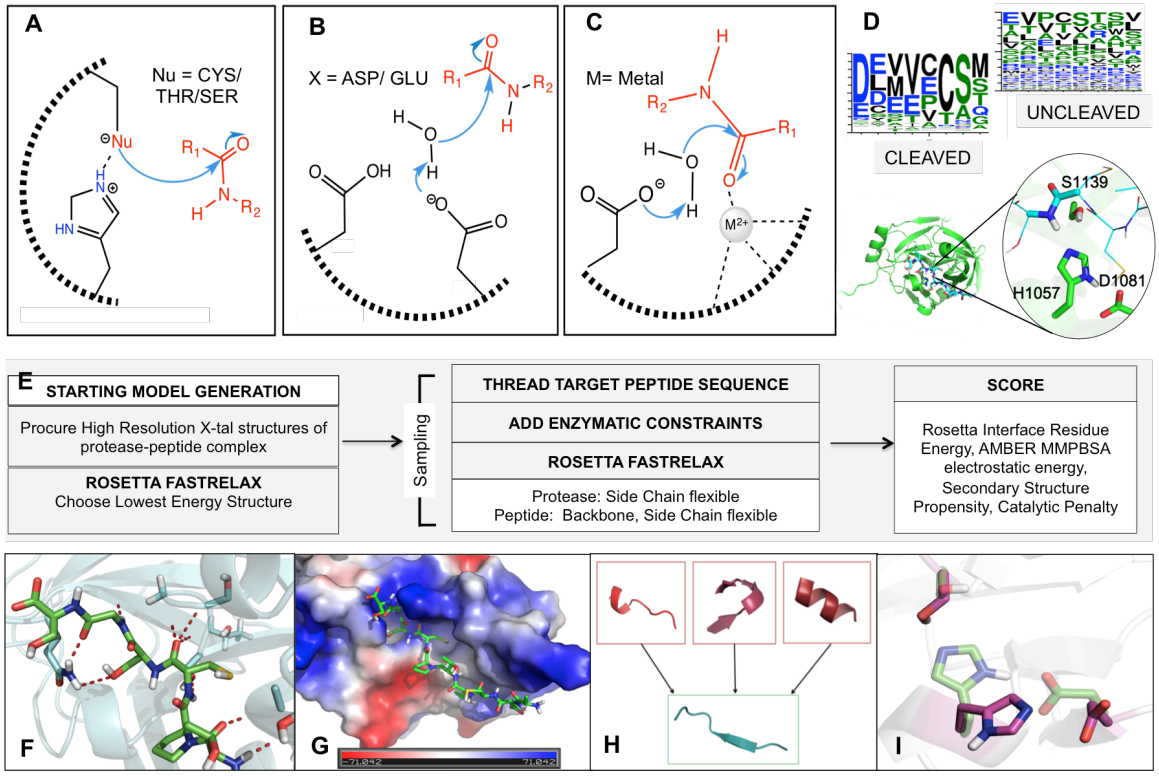


Figure 2.1. Overview of a general, energy-based discriminator
An illustration of the mechanism of steps leading to the formation of a common tetrahedral intermediate (TI) for serine-, cysteine-, threonine (A), aspartic, glutamic (B), and metallo-proteases (C). Protease active site cleft is depicted as a dashed arc. (D) Generation of atomic resolution models of the near attack conformation using high - resolution crystallographic structures and known cleaved and uncleaved sequence datasets. (E) The resulting complexes were allowed to relax into a minimum energy conformation using the described protocol (FastRelax) and scored using a linear combination of (F) the sum of the interface residues' Rosetta energy, (G) the sum of the interface residues' AMBER MMPBSA electrostatic scores, (H) a score that describes the

propensity of the peptide to adopt an extended conformation (reorganization penalty), and (I) the deviation of the active cleft residues from the idealized active conformations (a pseudo score-term). The linear combination of weighted scores were recombined according to this equation: Total_score = w1*Rosetta_Interface_Energy(Protease energy) + w2*Rosetta_Interface_Energy (Peptide energy) + w3* Catalytic constraint penalty + w4 *Reorganization Penalty + w5* Electrostatic Binding Energy; where w1 =1, w2 =1, w3 = 3.5, w4 = 0.01 , w5 = 0.5

Experimental methods to characterize protease specificity[26] range from low-throughput methods in which individual peptides or mixtures of peptides are assayed for cleavage[27–29] to high-throughput methods that allow identification of substrates on a proteome-wide scale[30,21,31–33]. However, substrate sequence space is large and different proteome-wide datasets often have little overlap, suggesting that many substrate sequences remain to be identified. Moreover, each experiment is limited to a single enzyme variant (typically the wild type). Computational approaches could, in principle, enable more rapid construction of specificity profiles, especially for naturally occurring or drug-resistant protease variants, and/or assist in library design for experimental specificity determination in a specific region of sequence space. Pattern recognition-based approaches have been used to predict substrate sequence preferences for various proteases based on machine learning from available experimental data[34–39]. However, these sequence-only approaches are constrained by the quality of the input data, and cannot be generalized to other proteases, or to variants of the same protease enzyme.

Proteolysis is a multi-step reaction involving the binding of the substrate and subsequent nucleophilic attack on the carbonyl group carbon of the scissile peptide bond to yield a tetrahedral intermediate (TI; Figure 2.1A-C)[17]. Steps after TI formation are mechanism-dependent: in cysteine, serine (and threonine) proteases, the intermediate

disproportionates to yield one product and the reaction proceeds via the formation of an enzyme-bound intermediate that is deacylated to yield the second product (Figure 2.1A). In aspartic (and glutamic), and metallo-proteases, which use a hydroxide nucleophile generated from a bound water molecule, the tetrahedral intermediate directly disproportionates into both products (Figure 2.1B, C). In principle, different steps could determine substrate specificity depending on the substrate and the mechanism under consideration. However, for all proteases, regardless of the mechanistic class they belong to, the first step, i.e., enzyme nucleophilic attack is required for turnover[17]. This observation led us to hypothesize that a model of the enzyme with the bound substrate and catalytic machinery modeled in a near-nucleophilic attack conformation would enable us to capture the energetics involved in substrate recognition and specificity.

Here, we develop a predictive biophysical model aimed at uncovering the underlying rules that govern protease-peptide molecular recognition and test its ability to classify known protease substrates from uncleaved ones. We construct a discriminative scoring function that includes descriptors of the energetics (including long-range electrostatic interactions) at the interface of the protease–peptide complex, the geometric compatibility of the substrate with the catalytically active state of the protease, and the reorganization penalty of a given substrate to adopt a favorable conformation in the protease active site[4]. We demonstrate the predictive capacity of this discriminator by the recapitulation of known cleavage specificities of five experimentally characterized proteases representing all the major mechanistic protease classes[18] (serine, cysteine, aspartic, and metallo- proteases). We demonstrate an application of our biophysical

discriminator by exploring previously uncharacterized, novel sequence motifs cleaved by the HCV NS3/4 protease via a yeast surface display-based assay[40] to identify novel cleaved sequences. Our biophysical structure-based model should allow the prediction of substrate specificities of experimentally uncharacterized proteases as well as protease variants (e.g. drug-resistant variants) and enable the structure-based design of proteases targeted to novel substrates.

2.3. Results

2.3.1. Rationale for the curation of Benchmark Datasets:

To develop and test a general structure- and energy-based prediction approach for protease specificity, we curated benchmark sequence sets for five diverse proteases. Each of these exhibit diverse mechanisms of action, varied folds and biological functions – TEV Protease (cysteine proteases), HCV NS3 protease (serine proteases), Granzyme B (serine protease), HIV Protease-1 (aspartyl protease) and Matrix Metalloprotease -2 (Metalloprotease). The sequence sets were composed of cleaved and uncleaved sequences identified in experiments or generated by examining naturally occurring targets (and non-targets) of each protease (see Methods). We preferentially chose datasets in which cleaved and uncleaved sequences were identified in the same experiment. For HCV NS3/4 protease, HIV Protease 1 and Granzyme B, we were able to identify experiment-derived datasets[36,41,42]. For TEV protease and MMP2 protease, we were able to obtain experimentally cleaved datasets[43–45] but uncleaved sequences were not available. Therefore, we generated a synthetic dataset of uncleaved sequences using a two-residue protein walk approach, utilized in previous computational and experimental work[36,41]. It is

possible that these synthetically generated uncleaved sequences may include a small number of cleaved sequences. However, experimental results from Shiryaev et al[41] suggest that misclassification of uncleaved sequences obtained using this approach is low. Therefore, in the absence of a directly experimentally determined uncleaved dataset for TEV protease and MMP2, we utilized this previously validated approach for uncleaved dataset creation.

2.3.2. Developing an energetic discriminatory scoring function based on structural simulations:

We hypothesized that determinants of substrate cleavage include (a) protease-peptide interfacial interactions, (b) the adoption of a catalytically competent conformation of the protease active site machinery in the bound state (near-attack conformation), and (c) a reorganization penalty that captures the propensity of a given substrate to adopt the extended conformation required for positioning the scissile bond in a cleavage-prone location in the protease active site. We created atomic resolution models for each peptide-protease complex and computed each of these terms as described below.

To model the conformation of each substrate peptide complexed with the active conformation of the protease, we created atomic resolution models within the context of the Rosetta macromolecular modeling software. Each known peptide substrate was threaded on the respective modeled near-attack conformation generated from the protease crystal structures (Figure 2.1D), and the resulting complex was allowed to computationally relax into a local energy minimum using Rosetta FastRelax[46], followed

by scoring this modeled conformation using Rosetta and Amber's MMPBSA modules (Figure 2.1E).

In addition to the interaction energy evaluated using Rosetta (Figure 2.1F), which includes a model of electrostatics, (called fa_elec), we also evaluated binding electrostatics by using Amber's MMPBSA module (Figure 2.1G). We reasoned that the Rosetta energy function has been weight optimized for all its component terms including fa_elec. Thus, we decided to include fa_elec even upon inclusion of the AMBER electrostatics score. We included two other terms in our discriminator scoring function: First, we included a term ("reorganization penalty") that captures the propensity of a given substrate to adopt the extended conformation observed in crystal structures of all proteases (Figure 2.1H). Second, the deviation of the active site from ideal catalytic geometry (a pseudo-energy term) upon energy minimization (Figure 2.1I), which captures the fit of a given substrate to the catalytically competent conformation of the protease, was included. These scores – energetic descriptors of the peptide-protease complex in a near-attack conformation – were combined using a linear weighting approach to obtain a discriminatory score function such that lower scores are predicted to energetically fit better in the active site (Figure 2.1F-I).

2.3.3. Recapitulation of known protease specificity profiles:

Each predicted substrate-binding set for each protease consists of a large set of evaluated peptide sequences, atomic-resolution bound structures, and predicted binding energies of individual peptides to the near-attack state of the enzyme. We compared our predictions

with experimentally determined specificity data from peptide library screening. Briefly, in these experiments, peptide (or peptide-cDNA fusion) libraries are generated and treated with protease of interest, cleaved and uncleaved populations of peptides are captured and identified using (deep) sequencing or mass spectrometry, and cleavage probability is assigned using Enrichment of a given peptide sequence in the cleaved population *versus* the uncleaved.

We found that for each of the five proteases, the distribution of discriminator scores was bimodal and cleaved and uncleaved sequences were separated in a statistically significant manner (*p*-values calculated using the Wilcoxon rank test; Figure 2.2A-E). To quantify the performance of the discriminator in the task of separating cleaved from uncleaved substrates, we performed a score threshold-based binary classification of the sequences into cleaved and uncleaved sets and calculated the area under the resulting receiver-operator curve (auROC; perfect discrimination would yield an auROC of 1.0; the expected auROC for a random ordering of the peptides is 0.5). The auROC values for the five proteases ranged between (0.86 for MMP-2 to 0.98 for TEV-PR), demonstrating robust discrimination using energetics (Figure 2.2G). The critical point of the auROC plot represents the optimal tradeoff between false positive and false negative rates. We found that false positive rates at critical points ranged from 0.04 (TEV-PR) to 0.24 (MMP-2), suggesting robust discrimination of the substrates into cleaved and uncleaved sets with a small but significant false positive rate (Table 2.1). We note that weights used for combining the five score terms were initially optimized to maximize discrimination for HCV NS3/4 protease (five weight terms over approximately 2100 data points), yet TEV-

PR displays the best performance in terms of both auROC and critical point values using this weight set. These results demonstrate the generality and robustness of the energy-based scoring function.

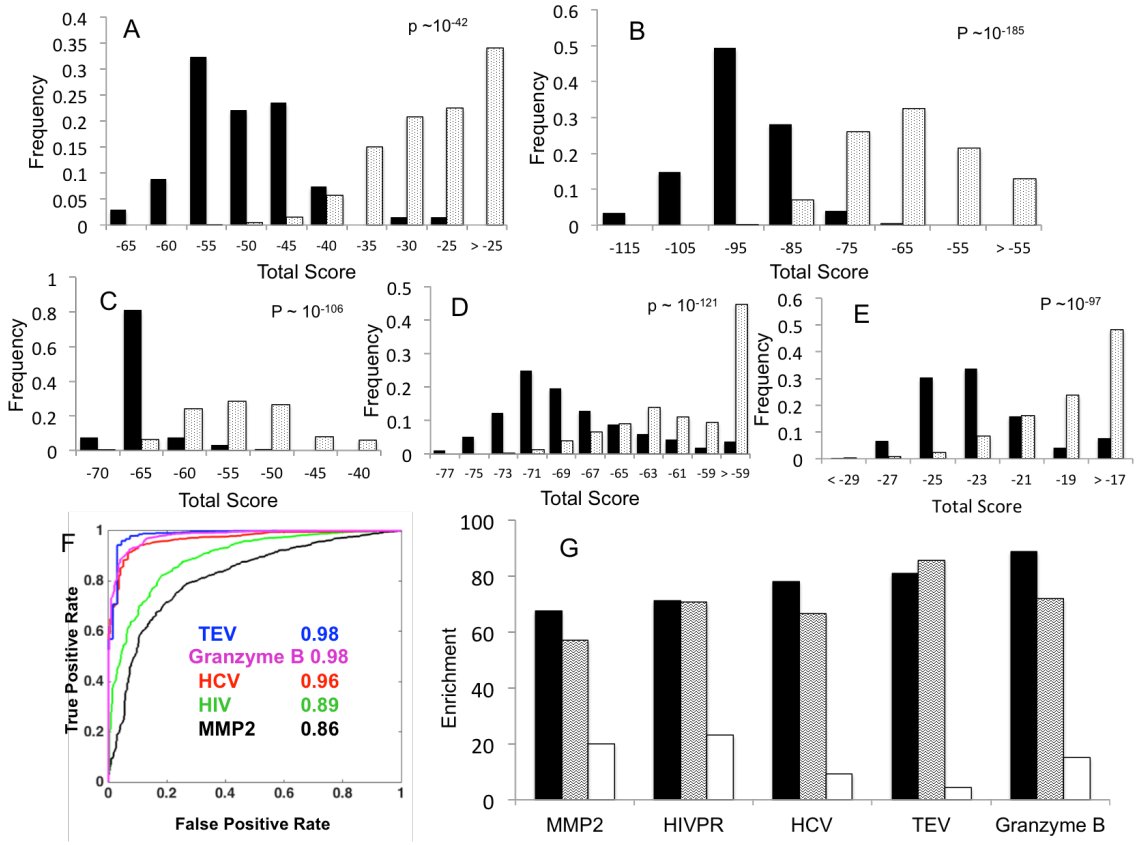


Figure 2.2. Distribution of Discriminator Scores
Score distributions for cleaved sequences (depicted in black) and uncleaved (depicted in dotted bars) for (A) TEV protease (B) Granzyme B (C) HCV (D) HIV (E) MMP2. The *p*–values were calculated using a Wilcoxon rank test. A threshold based binary classification of sequences into cleaved and uncleaved sequences using these scores was performed and the auROC (F) for the five proteases are indicated. (G) Enrichment of true cleaved sequences in the top-ranked pools. Enrichment ratio (black bars) = #true cleaved/ # of cleaved sequences in dataset. Background Enrichment (white bars), which represents fraction of cleaved sequences in the dataset, and Enrichment obtained from SitePrediction model (wavy bars) with 20% of the known cleaved sequences. In each case, the structure-based discriminator performs comparably to or better than SitePrediction.

Table 2.1. True positive and false positive rates observed for critical point of auROC.

Protease	TPR	FPR
HCV	0.92	0.08

TEV	0.96	0.04
HIV	0.82	0.18
Granzyme B	0.93	0.07
MMP2	0.76	0.24

To evaluate the ability of the discriminator to identify cleaved sequences from the entire pool of sequences – a task that would aid in novel substrate identification – we calculated the fraction of truly cleaved sequences in the top-scoring N_{cleaved} sequences, where N_{cleaved} is the number of cleaved sequences in the dataset. This Enrichment value is compared to background Enrichment, i.e. fraction of cleaved sequences in the dataset (reflecting a scenario when the ranking is performed by randomly shuffling the list of sequences). We find that in all cases a significantly higher fraction of sequences was enriched compared to the background with Enrichment ratios ranging from 3-fold (HIV-PR) to 19-fold (TEV-PR) (Figure 2.2F). We compared the Enrichment obtained using our discriminator with that obtained using SitePrediction[38] – a sequence-based machine learning method that relies on training with experimental data. For each protease, we trained a SitePrediction model with randomly chosen 20% of the known cleaved sequences and used the remaining dataset for testing. For all proteases, we find that our unbiased, biophysics-based approach yielded similar or higher Enrichment values as SitePrediction models trained separately on each individual protease. The lack of training on known experimental data makes the structure-based discriminator more widely applicable.

2.3.4. Optimization of scoring and sampling strategies:

To investigate the contribution of each score term and its weight in the discriminator scoring function, we evaluated the discrimination performance of various score term combinations. We found that while much of the discriminatory power could be attributed

to Rosetta interface residue energies, all five terms do contribute to the observed prediction metrics when they are serially included along with the Rosetta energy. While the increases in auROC compared to Rosetta energies-only scoring functions were modest, Enrichment values benefited significantly by the inclusion of the additional terms *e.g.*, for Granzyme B inclusion of the AMBER electrostatics score and secondary structure propensity increases Enrichment from 0.70 to 0.87 (Figure 2.3, Table 2.2). As auROC measures the overall difference in the two distributions (cleaved and uncleaved) and Enrichment measures the rank ordering of sequences, we conclude that inclusion of additional terms serves to subtly alter the calculated energy landscape and "rescue" some false negatives (cleaved sequences that score comparably to low-energy uncleaved ones).

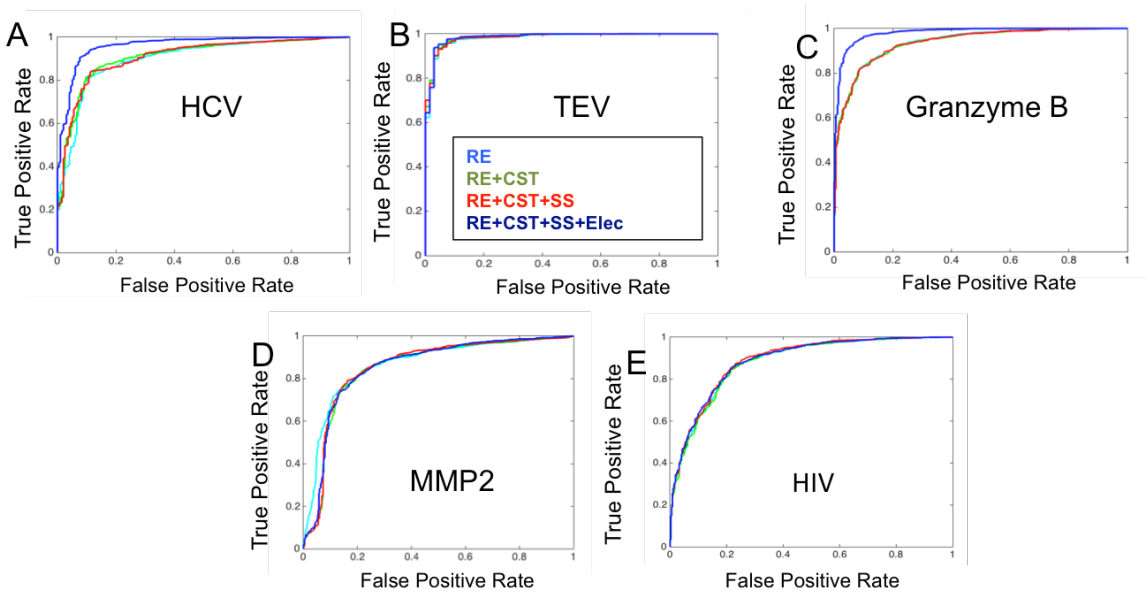


Figure 2.3. The additive effect of each energy term to the auROC.
Each plot shows the representative ROC curve for Rosetta Energy (sum of peptide and protease interface energy; depicted in light blue), Rosetta Energy + constraint score (Green), Rosetta Energy + constraint score + secondary structure propensity (red), Rosetta Energy + constraint score + secondary structure propensity + Electrostatic binding energy (dark blue). All score terms are seen to contribute to the discriminative efficiency of the score function.

Table 2.2. Results of a calculation to investigate the additive effect of each score term in the discriminatory score function.

Protease		RE+CST	RE+CST+Elec	RE+CST+Elec+SS
Granzyme B	Enrichment	0.70	0.68	0.87
	Fold increase	4.6	4.5	5.7
	AUC	0.93	0.93	0.98
HCV	Enrichment	0.64	0.76	0.80
	Fold increase	6.2	7.3	7.6
	AUC	0.93	0.97	0.97
TEV	Enrichment	0.72	0.72	0.80
	Fold increase	16.68	16.68	18.35
	AUC	0.98	0.98	0.98
HIV	Enrichment	0.69	0.68	0.69
	Fold increase	3.2	3.2	3.2
	AUC	0.90	0.90	0.90

We next investigated whether optimization of weights of the energetic scoring terms could improve performance. We used a grid-based optimization scheme in weight space to maximize Enrichment. While keeping Rosetta protease energy fixed, we optimized four free parameters by enumerating all combinations of peptide residue energy (0.3-1.3 in increments of 0.1, constraints (2.5-3.5 in increments of 0.1), secondary structure (0.005-0.02 in increments of 0.005), and electrostatics (0.1-0.3 in increments of 0.05). The ranges were chosen after a coarse-grained parameter sweep to find good starting parameters, and by considering the orders of magnitudes of raw scores of the score terms. For example, the raw score for the Secondary Structure Propensity term ranges between 0-200 (number of fragments from the top 200 that have an RMSD greater than 3.0 A compared to the crystallographic conformation of the peptide). As the Rosetta

residue energy weight was 1, we explored weight ranges of 0.005-0.02 for the secondary structure term. The results of this optimization are listed in Table 2.3.

Table 2.3. Results of a grid-based optimization scheme to maximize enrichment.

	Enrichment	*protease*	*peptide*	*cst*	*ss*	*Elec*
TEV	0.8088	1	0.3	2.5	0.005	0.25
HCV	0.7806	1	1	3.5	0.001	0.5
HIV	0.7112	1	0.8	3.4	0.013	0.1
GrB	0.8867	1	0.5	2.5	0.005	0.3
MMP2	0.6747	1	0.7	2.6	0.007	0.1

We next examined the impact of sampling flexibility of the backbone and side chain degrees of freedom at the protease-peptide interface (Figure 2.4) and found that limiting the backbone degrees of freedom of the protease, while sampling the full backbone DOFs of the peptide, yielded the highest Enrichment values. Previous studies with farnesyltransferase enzyme similarly observed that greater sampling of the peptide degrees of freedom increased performance[47]. When the protease backbone was allowed to move in an unconstrained manner, several uncleaved sequences adopted energetically favorable conformations. While some of these false positives can be attributed to limitations of the simulation force fields and sampling strategies, these results indicate that side chain flexibility in the protease pockets coupled to peptide backbone flexibility are key contributors to the molecular recognition observed at these interfaces.

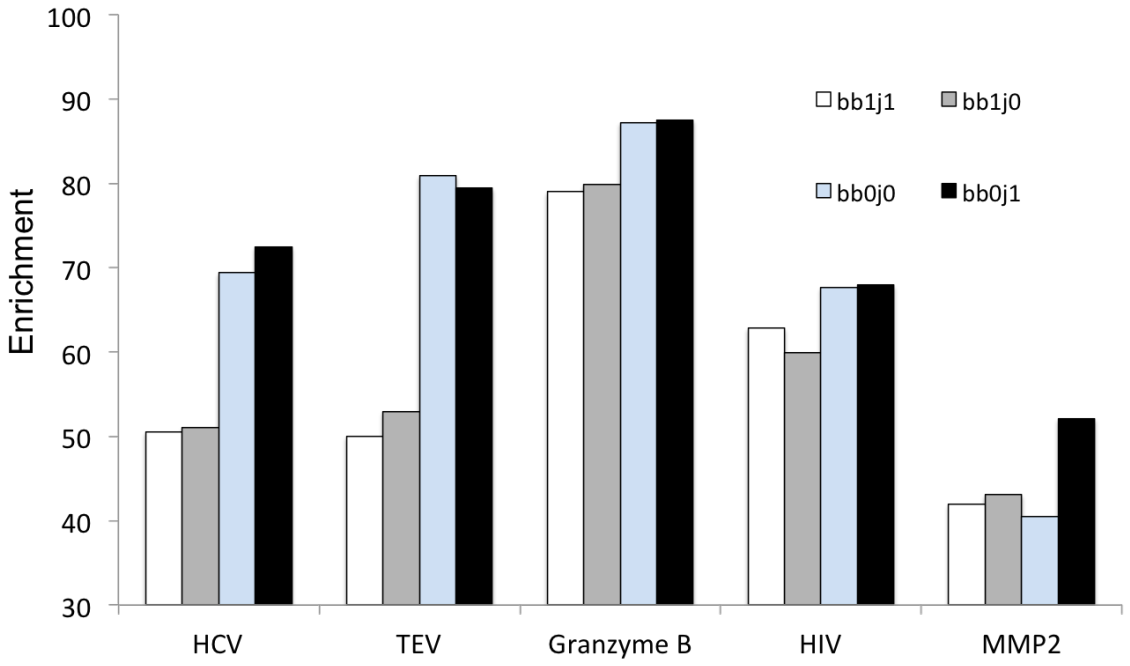


Figure 2.4. Impact of sampling flexibility of the protease backbone and sidechain degrees of freedom.
The peptide backbone and sidechains were flexible in all the simulations depicted in the figure. "bb" refers to the backbone of the protease such that bb=0 indicates that the backbone was not allowed to relax, bb=1 that the backbone was allowed to relax. "j" refers to the rigid body freedom of the peptide with respect to the protease. j=0 means that rigid body freedom was constrained during the simulation; j=1 rigid body flexibility allowed during simulation. The highest efficiency of discrimination was observed when the protease backbone was not allowed to relax, and the protease sidechains were flexible during the simulation.

Finally, we explored the contribution of maintaining, during each simulation, the scissile peptide bond in a near-attack conformation with respect to the protease catalytic machinery using geometric constraints, by performing simulations without these geometric constraints, and/or removing the constraint scores from the discriminator scoring function. In each case, a decrease in Enrichment was observed (Figure 2.5), providing further support for our rationale that specificity in protease-peptide molecular recognition is not simply a ground state binding phenomenon, but is contingent upon the

relative energetics of the near attack substrate conformation during the nucleophilic attack step.

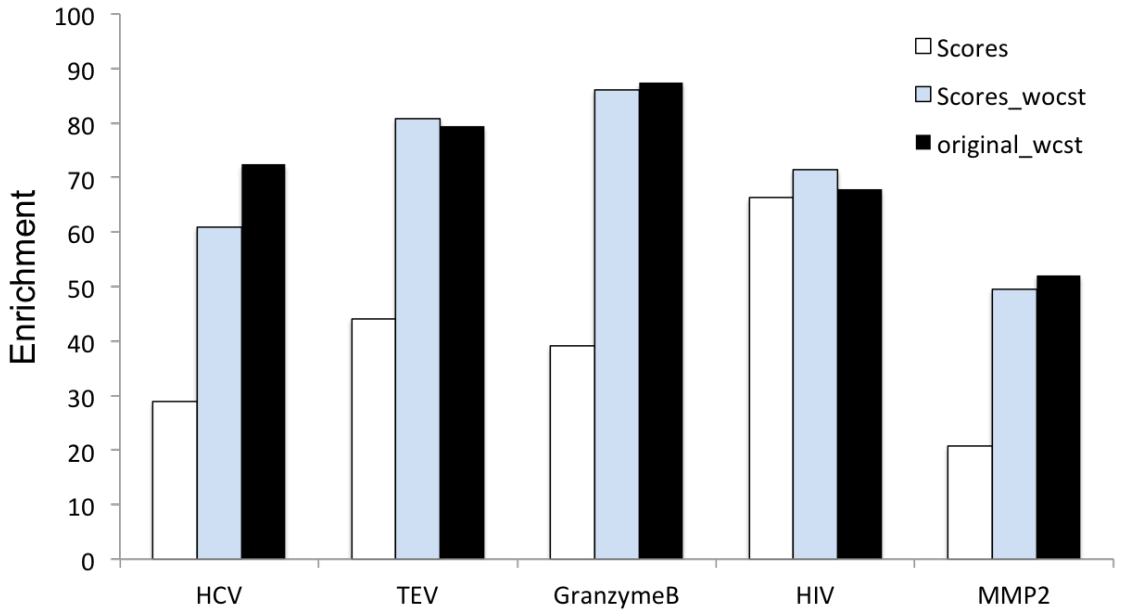


Figure 2.5. Contribution of maintaining near attack conformation with respect to protease catalytic machinery.
Three FastRelax protocols were performed to compare the effect of the presence of catalytic constraints during the FastRelax and scoring stage. Scores (white bars) depict enrichment values obtained when enzymatic constraints were excluded in the FastRelax step but were included in the scoring step. Scores_wocst (blue) depict experimental results where constraints were excluded from the FastRelax step as well as from the scoring calculation. Original_wcst (black) depict experimental results where FastRelax was performed with constraints and the constraint score was included in calculation of Enrichment. Highest enrichment is observed when catalytic constraints are included in both the FastRelax as well as scoring steps.

2.3.5. Combining sequence and energetic signatures using machine learning leads to higher discriminatory power

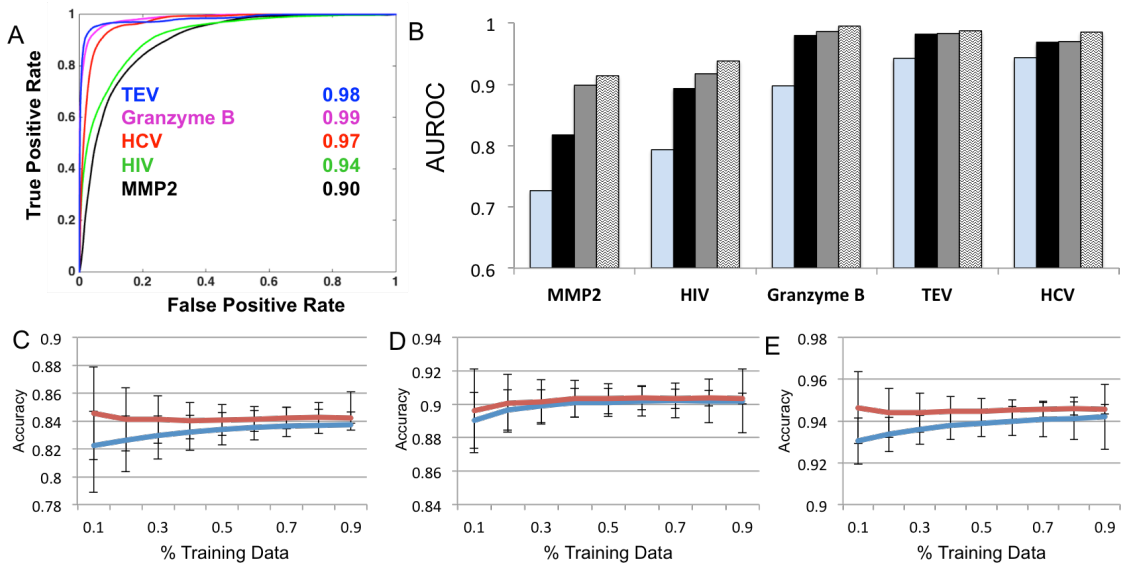


Figure 2.6. Combining sequence and energy signatures leads to higher discriminatory power
(A) The energetic features were used to train an SVM using a radial based function, which yielded higher auROC values for all proteases as compared to a linear combination of optimized weights. (B) auROCs obtained from support vector machines (SVMs) trained with sequence only (blue), energetic only (gray) and both sequence and energetic features (wavy) in a 5-fold cross-validation test. Black bars indicate auROC for the linear combination of weighted score terms. The combination of sequence and energy features consistently results in higher auROC values. (C) Accuracy as a function of training set size used for training for the (C) sequence, (D) energetic features, and (E) both sequence and energetic features. The accuracy values are not altered appreciably when a significantly smaller training dataset is used. In-set classification and generalization curves converge as a progressively higher fraction of the dataset is used for training. The classification curve is shown in red whereas the generalization curve is depicted in blue.

Current approaches for protease specificity prediction, including the SitePrediction tool discussed above, PCSS server[36] and PROSPER[35], use machine learning of sequence patterns in known experimental data. To more extensively compare our structure-based specificity prediction with current sequence-based approaches we trained support vector machines (SVMs) with sequence-only, energetic-only and both sequence and energetic features (Methods). For the energy-based SVM, the (unweighted) energy terms described above were treated as features ("interface protease residue energy", "interface peptide

residue energy", "constraints energy", "reorganization penalty" and "MMPBSA electrostatic binding energy"), whereas sequence-based features were generated using a protocol described by Barkan *et al*.[36]. We found robust discrimination of the substrate sequences using energy-based SVMs trained individually on each protease in 5-fold cross-validation test (Figure 2.6A). The values of auROC obtained using these SVMs are higher than those obtained with scoring using a linear weighting scheme (Figure 2.6B, black and gray bars), due likely to the use of a non-linear kernel function and training on individual datasets. When compared to a purely sequence-based SVM, the energy-based SVM consistently leads to higher auROC values for all datasets, and an SVM constructed based on sequence and energy features displays a high AUC value when compared to solely sequence-based and energy-based based SVMs (Figure 2.6B). These results indicate that structural/energetic features contribute information that is orthogonal to that obtained from sequence-only features.

To ensure that the increased discriminatory ability observed upon combining sequence- and energy-based features is not a result of data over-fitting, we performed a cross-validation procedure where in-set training (classification) and out-of-set testing (generalization) was performed by randomly splitting the datasets into training and test subsets[48] . We find that the performance of the method as indicated by the accuracy of prediction, does not appreciably alter when a significantly smaller training dataset is used for the energy-based SVMs, and the classification and generalization performance converge as the training set size increases (Figure 2.6C-E, Figure 2.7). The convergence between classification and generalization occurs at higher training set fraction for the

sequence-based SVMs than energy-based ones, demonstrating that the key energetic signatures underlying discrimination can be captured with a smaller dataset compared to the corresponding sequence signatures (Figure 2.7). Thus, energetic feature-based SVMs can outperform sequence-based ones, and the two sets of features can be combined to obtain more accurate classification than either set of features independently.

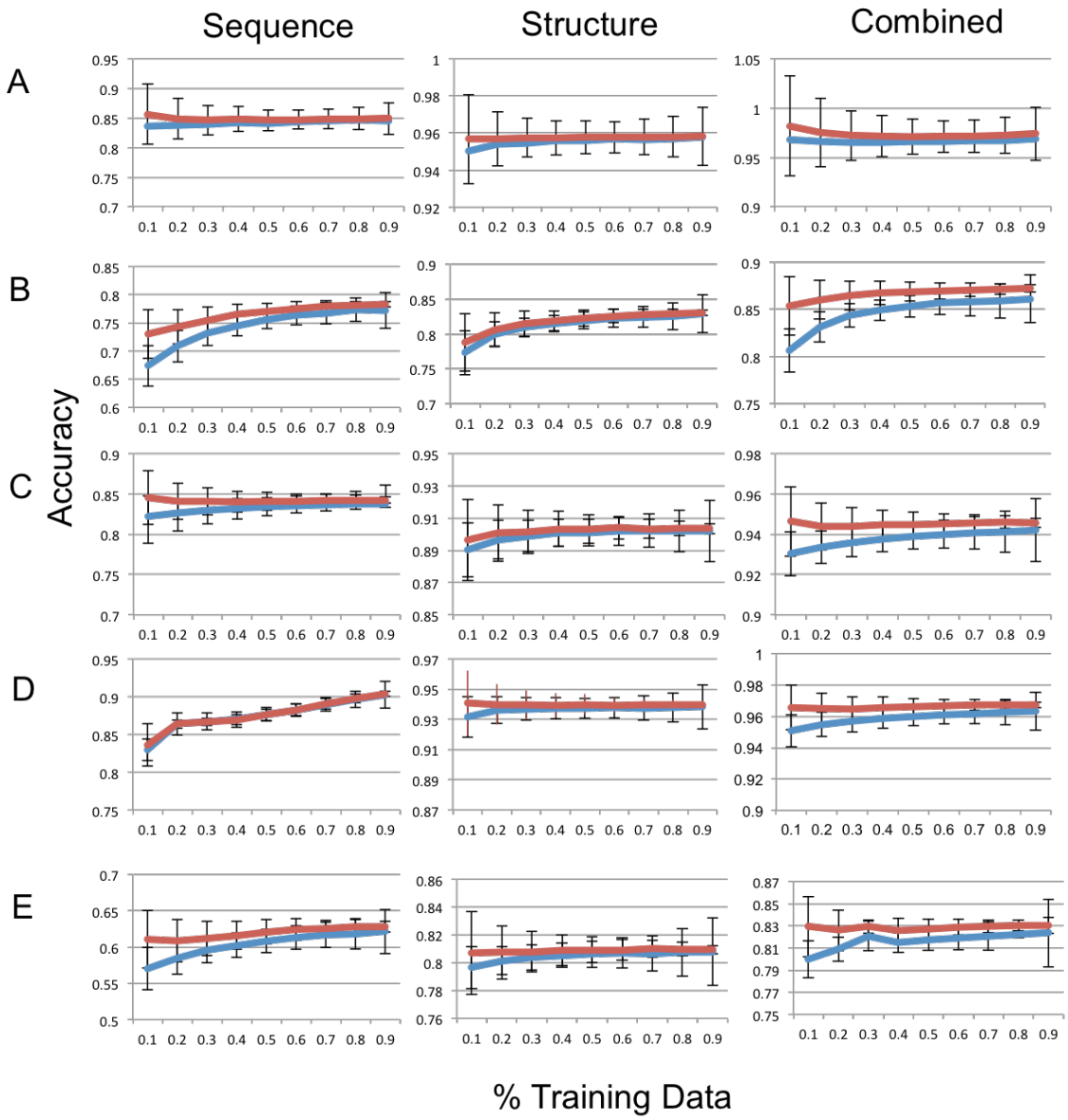


Figure 2.7. Accuracy versus Training Data size plots for Sequence, Structure and Combination SVMs.

To avoid over-fitting we performed a jack-knifing procedure where classification and generalization was performed by randomly splitting the datasets into training and test. (A) TEV (B) HIV (C) HCV (D) Granzyme B (E) MMP2

2.3.7. Multi-body interaction networks at the interface underlie improved discrimination

To investigate the underlying reasons for the observed increase in prediction efficiency when structural features are used, we identified several peptide sequences that are consistently misclassified by the sequence-based approach but are correctly classified by the structure-based approach. In several cases, we find that the increased classification ability could be attributed to interaction networks composed of multiple substrate and protease residues. A sequence-only approach would require a significantly larger training data than a relatively unbiased energy-based approach to directly "learn" multi-body correlations (interactions).

Three examples of these interaction networks are described below:

1. The structure-based discriminator can identify context-dependence of the substrate residue interactions more readily than a sequence-based approach, especially in cases where sequence preference at a given substrate site is not pronounced. For example, for the HIV protease, cleavage occurs between small non-polar amino acids and sequence preference at any other site is not particularly pronounced. Thus, GPGTASRP (Figure 2.8A) is misclassified as "cleaved" by the sequence-based SVM for HIV protease-1. There are no pronounced sequence preferences at position P3' (Figure 2.10). The

structural model of this sequence, however, shows that the guanidinium group of the arginine sidechain (P3') is packed in the vicinity of R8, a key residue, whose interaction with D29 is critical for HIV protease structural (dimer) stability [49]. Thus, the presence of an arginine at this P3' position would lead to lack of cleavage of the substrate, unless a secondary interaction relieves the electrostatic repulsion between the substrate arginine sidechain and the guanidinium group of R8. The subtle balance of these protease-substrate interactions can be captured by the electrostatics calculations in our approach.

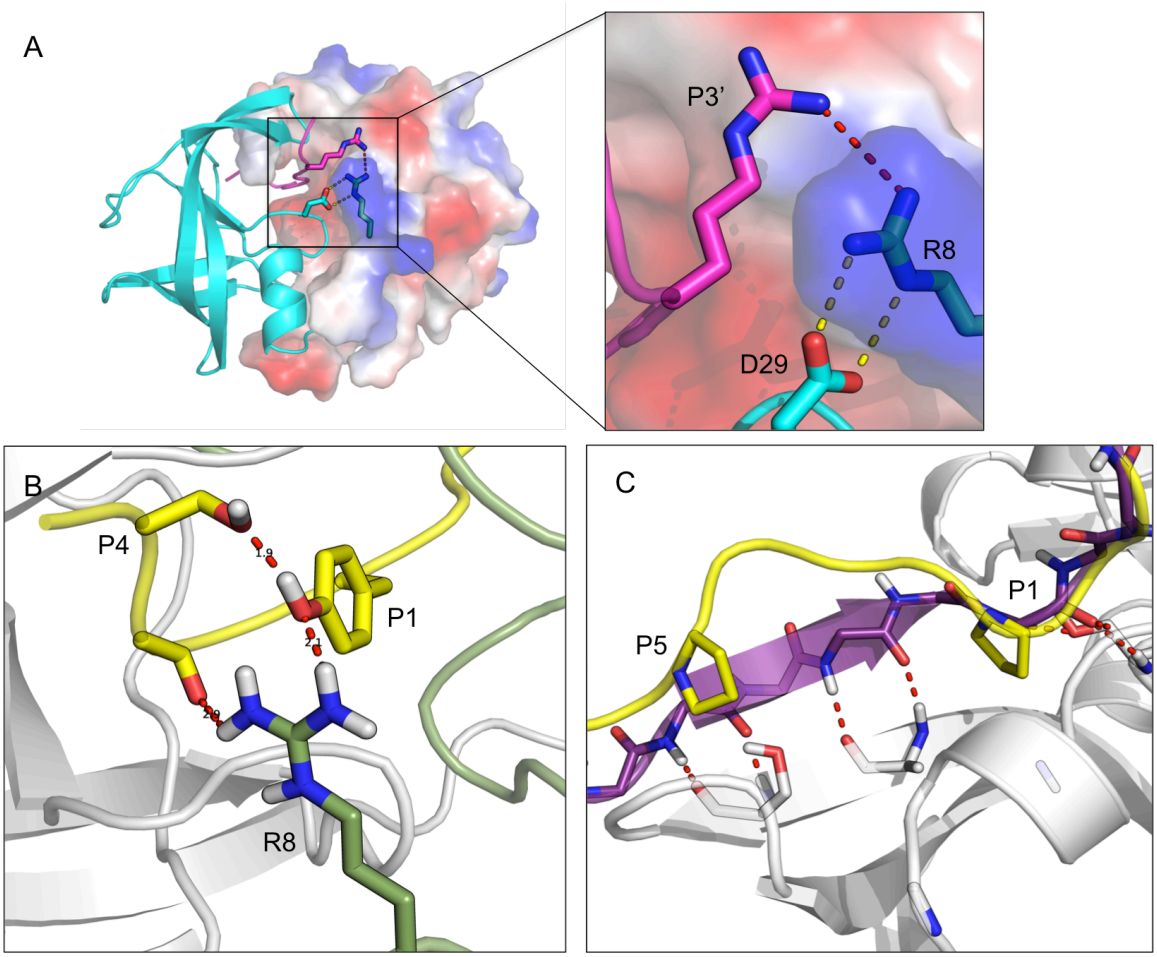


Figure 2.8. Multi-body interaction networks at the interface underlie improved discrimination.
Several sequences are misclassified by the Sequence-Based Discriminator whereas they are correctly classified when the Structure based Discriminator is used. (A) The sequence 'GPGTARSP' is misclassified by the sequence based SVM as 'cleaved' for the HIV protease sequence set. Residue P3' of the peptide is packed in the vicinity of ARG 8;

which is involved in a key interaction with ASP 29 necessary in maintaining the dimer interface. The P3' –ARG 8 repulsion leads to a destruction of one of the key interactions involved in dimer interface stabilization. One half of dimer surface is shown as a cartoon representation and the other as a charged surface to highlight the dimer interface of the HIV protease. This electrostatic repulsion is captured by the energy-based approach but not the sequence based approach, leading to a misclassification by the latter (B) 'SQAYPIVQ' is misclassified as an uncleaved sequence present in the HIV protease sequence set. The P1 tyrosine residue (yellow) along with the serine at P4 forms a favorable hydrogen bond network with ARG 8 (green) allowing for substrate cleavage. This favorable hydrogen-bonding network is likely not directly recognized by the sequence-based approach. (C)'KPAIIPDR' belongs to the HCV Protease sequence set which is misclassified as cleaved by the sequence-based approach. The presence of proline at positions P5 and P1(yellow) bends the substrate chain in an orientation that is unfavorable for cleavage. The extended conformation of a peptide, which allows hydrogen bond formation, leading to binding of the peptide and eventually cleavage, is highlighted (purple). The Rosetta energies correctly detect this disruption of the hydrogen bond network caused by the presence of proline residues between peptide (yellow) and protease.

2. The energy-based discriminator is able to detect hydrogen bond networks between substrate residues, including those mediated by the protease structure. For example, for the sequence SQAYPIVQ (Figure 2.8B), the sidechain of the tyrosine residue at position P1 forms a hydrogen bond network with the P4 position on the substrate and the R8 of the protease chain. This likely allows the protease to recognize and cleave this substrate.

3. Another set of interactions that our structural approach correctly characterizes are those mediated by proline and glycine residues, as these have specific backbone preferences that can affect the peptide backbone conformation. Figure 2.8C is an example of a sequence, KPAIIPDR, which is experimentally shown to be uncleaved by the HCV NS3 protease. The sequence-only approach misclassifies this sequence as cleaved, likely because the non-polar isoleucine residues at the P1, P1' residues. However, the proline residues present at P5 and P1 substrate positions bend the substrate backbone into a conformation that results in the disruption of the stabilizing backbone hydrogen bond

network, which drives the extended substrate conformation optimal for cleavage. The Rosetta energy function detects the disruption of this backbone hydrogen bond network, and thus the energy-based approach correctly classifies this sequence as 'uncleaved'.

2.3.8. Discovering novel sequence specificities HCV NS3 4A Protease

To further investigate the predictive ability of the energetic-discriminator in a blind test, we used our simulations to identify novel cleaved substrates for the HCV NS3/4 protease. The residue identities on the substrate peptide at positions P6 through P2 were sampled and scored as described in Methods using the structure-based discriminator. A total of 26,400 candidate sequences were evaluated (out of the possible 20^5 = 3.2 million) in a two-step procedure of sequence sampling as described in Methods, low-scoring sequences were clustered and were further pruned to identify sequence motifs that were novel (i.e., absent from the dataset used for developing the discriminator). We identified four such sequence motifs (Figure 2.9A), whose scores overlapped with the distribution of scores obtained from known cleaved sets. At least one peptide sequence was selected from three of the four identified motifs, and these were tested experimentally using a Yeast Endoplasmic Reticulum Sequestration Screen (YESS system) based assay [40,50] (Figure 2.9B).

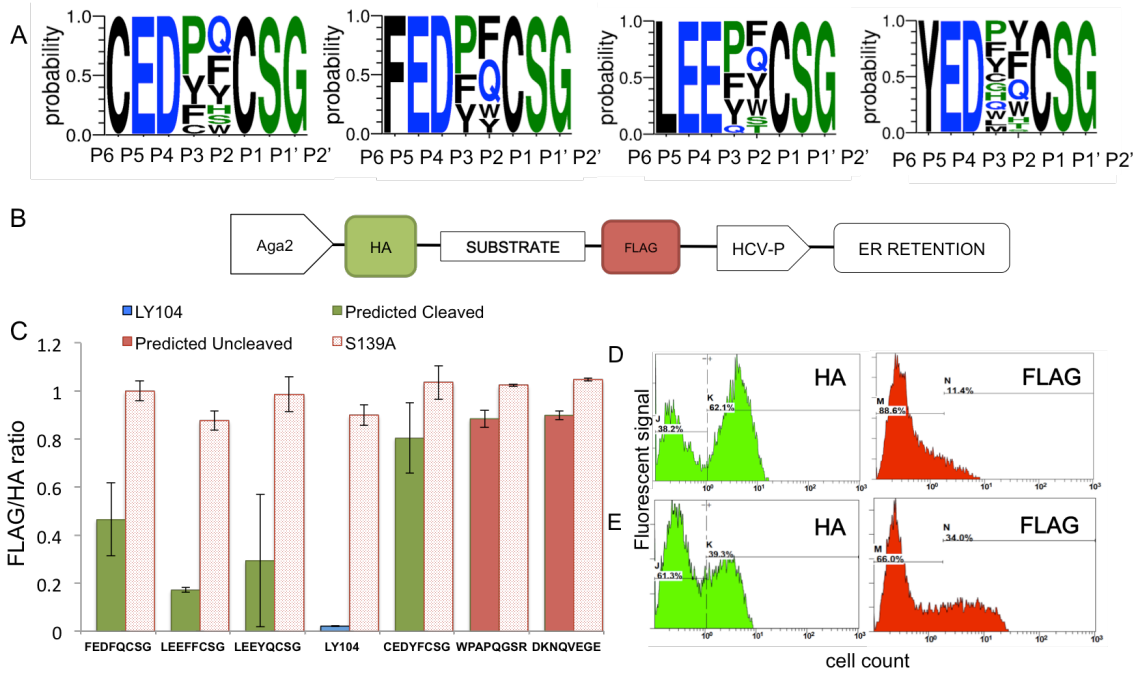


Figure 2.9. Discovering novel sequence specificities HCV NS3 4A Protease
(A) Sequence Logo plots of the identified four novel sequence motifs whose scores overlapped with the cleaved sequences in the benchmark dataset (B) Schematic of the vector (LY104) used for the YESS assay. The vector contains Aga2 cell surface signaling moiety followed by the substrate flanked between HA tag and FLAG tag which can be detected on the cell surface by fluorescently tagged antibodies. The protease and substrate are co-expressed in the ER of the yeast cell. If cleavage occurs the FLAG:HA ratio is 0, if substrate is uncleaved ratio is 1. (C) Results of the YESS assay test of the predicted cleaved sequences. Three out of the four tested sequences (predicted cleaved; green bar) showed a FLAG:HA ratio <0.5. The positive control (wild type shown in blue) showed an expected low FLAG/HA ratio whereas the negative control (known and predicted uncleaved sequences, red bars) showed high FLAG:HA ratios >0.85. The protease activity knockout mutant S139A (dotted red bars) showed FLAG:HA ratio >0.85 for all sequences, confirming that the sequences were cleaved because of the co-expressed HCV NS3 protease from the assay vector and not an endogenous yeast ER enzyme. (D) Cell cytometry histograms of LEEFFCSG, predicted cleaved sequence showing a 62.1% cell population signal for HA tag, 11.4% cell population signal for FLAG, thus showing a FLAG:HA ratio of 0.18 (E) Cell cytometry histograms for the negative control sequence DKNQVEGE, showing a 38.3% cell population signal for HA tag, and 34.0% for FLAG tag, thus exhibiting a FLAG:HA ratio of 0.88.

In this assay, the protease and substrate are co-expressed in active forms in the ER of yeast, and the substrate is targeted to the cell surface by fusion to the cell surface protein Aga2p. Proteolysis is detected using fluorescent antibodies against the HA and FLAG

tags that flank the substrate. We confirmed that the cleavage of the wild type substrate sequence (DEMEECA - canonical HCV NS3 cleavage sequence present between NS4A/4B on the polyprotein) results in the detachment of the FLAG tag from the AGA2 surface-signaling moiety, thus resulting in a FLAG:HA ratio of zero for complete cleavage and a ratio of one for no cleavage when an inactive variant of the protease (S139A) is used (Figure 2.9C). Several previous studies[41,51,52] have shown that the HCV protease cleaves between C/S or C/A residues (P1/P1') – however, the specificity at other positions can be broad and has not been explored fully. In all our predicted substrates (that we tested experimentally) the P1/P1' positions are still maintained as the known canonical sequence C/S, and our goal was prediction of different P6-P2 patterns. We, therefore, reasoned that the cleavage position of our substrates would not be altered as they retain the canonical P1/P1' cleavage pattern. The FLAG and HA signals were detected using flow cytometry. The observed FLAG/HA ratios (Figure 2.9 C, D) demonstrate that three out of four predicted sequences showed cleavage with ratios <0.5, whereas control assays with the S139A inactive protease variant showed significantly higher (>0.85) ratio, demonstrating that the observed cleavage is not due to a non-specific endogenous yeast enzyme.

Out of the four sequences that are predicted as cleaved, one sequence – CEDYFCSG – shows a high FLAG/HA ratio, and represents a prediction failure. These results are consistent with the ~75% True Positive and ~25% False Positive rates (Figure 2.2F) observed in the performance of the discriminator on known cleaved and uncleaved datasets, i.e., approximately one out of four sequences identified is expected to be a false

positive sequence. We also identified two predicted uncleaved substrates, and these show lack of cleavage when co-expressed with either wild type protease or the inactive protease variant, as expected. The FLAG:HA ratios for the novel identified substrates are higher than positive control LY104, indicating that the substrates identified are suboptimal. However, our test for novel substrates is particularly stringent as we chose sequence motifs that have previously not been identified in multiple studies of HCV NS3/4 protease. Thus, the developed discriminative score function and validating assay provide a method to screen for potential novel biological targets of this viral protease that is also a drug target.

2.4. Discussion:

Proteolytic cleavage is a key component of diverse and ubiquitous biological processes such as apoptosis, blood clotting, viral maturation, and cancer[3]. Developing a generalizable, predictive model for protease specificity would enable identification of potential novel substrates for furthering our understanding of protease biology and enhancing our ability to design inhibitor small molecules to chosen proteases. We developed a structure-based approach for specificity prediction using Rosetta and Amber force fields that provides atomic resolution insights into the molecular recognition at protease-substrate interfaces. We found that structural models robustly recapitulate known protease specificities for each of the four major protease classes (serine, cysteine, aspartic, and metallo-proteases) with little training on experimental data, and in several cross-validation tests. When combined with a machine learning algorithm our energy-based approach outperforms current bioinformatics-based approaches[37] on benchmark

sets, and a further increase in discrimination is achieved when both structure-based and sequence-based approaches are combined. To further test the utility of our approach in a blind manner, we used it to predict four novel substrate sequences for HCV NS3/4 protease, tested these predictions experimentally, and found that three of the four novel predicted cleaved sequences were cleaved by the protease; a success rate similar to that of the benchmark set was achieved in the blind experimental test.

The value of using energetic information in the discriminator is evident in the protease structure-dependent interaction networks that are captured in the energetic signatures. These interaction networks are equivalent to pairwise and multi-body correlations in the sequence data. Given 20 amino acid types at every substrate peptide position, a relatively large number of training sequences are required to "learn" pairwise and higher-order correlations between positions, whereas only ~2000 sequences (among them ~200 cleaved) are available in the experimental benchmark datasets. The structure-guided, energy-based discriminator has the advantage of being generalizable, relatively unbiased and is able to recapitulate key interactions that stabilize the peptidase – peptide interface as well as predict novel interactions not present in the training data. Success in using structure-based energetic signatures and molecular docking for binding partner identification has been achieved for several peptide recognition modules such as SH3 and PDZ domains[53–57], major histocompatibility complex[58] and for the enzymes methyltransferase[59], farnesyltransferase[47], and HIV protease[60,61]. We show here that a structure-based approach, guided by the knowledge of mechanism, can be successfully

integrated with machine learning to predict substrates for a mechanistically diverse enzyme family such as proteases with high accuracy.

Proteolytic sites in full-length proteins are more often found in exposed regions of the structure, and more frequently in flexible loops and beta conformations compared to buried regions and alpha helices[32]. A substrate sequence generally adopts an extended conformation in the protease active site[4], and surface-exposed loops and beta-strand regions are likely to pay a smaller reorganization penalty to adopt this extended conformation. Therefore, we incorporated the local structure preferences of the substrates in our datasets by computing local sequence-structure compatibility – an implicit assumption in our approach is that every candidate peptide sequence is equally accessible to the protease active site. This assumption is valid when analyzing the extended substrate specificity of the protease, but for the task of predicting cleavage sites in a given whole protein sequence, additional solvent accessibility and structural information are expected to modulate cleavability. Barkan *et al*. have shown that incorporation of such features improved prediction of cleavage sites in whole protein sequences. Furthermore, Julien *et al*.[21] found that cleavage efficiencies of protein substrates identified using a high throughput mass spectrometry-based approach and their synthetic peptide counterparts were correlated. Taken together, it appears likely that local primary sequence specificity (modeled here) largely determines the identity of cleavage sites, although the context of the cleavage site modulates the kinetics of cleavage.

Comparing the performance of the discriminator for the different protease systems included in the benchmark set highlights the strengths and limitations of our approach. Highest Enrichment of cleaved sequences in the top-ranked population is observed for TEV and Granzyme B proteases (Figure 2.2), where the active site is relatively rigid and steric effects and hydrogen bond interactions are the major contributors to specificity, highlighting the strength of the Rosetta force field in modeling these effects. However, performance is more modest for the metalloenzyme MMP-2, which features a zinc ion in the active site, and for the HIV protease, in which loop residues mediate molecular recognition. For these systems, inaccuracies in the modeling of flexibility of the active site conformation, and lack of explicit consideration of entropy changes can lead to increased misclassification. More exhaustive sampling of the backbone degrees of freedom of the loop structural elements is likely to improve performance as observed in other studies of peptide-protein molecular recognition[47,62]. Finally, while modeling catalytic residue conformations using geometric constraints appears to be a reasonable approximation for most systems considered here as evidenced by success in discrimination, electronic effects may be involved in the vicinity of the active site, especially for the metalloenzyme MMP-2. We also investigated alternative protonation states of key catalytic residues (nucleophiles serine, cysteine, hydroxyl and bases histidine, aspartic acid) in the MM-PBSA pipeline, but these charge changes did not lead to any appreciable increase in the performance (data not shown). It is likely that quantum mechanical (QM) calculations may be required to model these effects more accurately. However, the high computational cost of detailed QM simulations precludes the use of such calculations for the thousands of substrate-enzyme pairs considered in our study.

Advances in QM simulation methodology[63] and computational infrastructure are likely to bridge this gap in the future.

In contrast with sequence-based specificity prediction approaches, the unbiased nature of the biophysical substrate specificity predictor developed here should allow the modeling of specificity of protease variants for which experimental data are not available, such as newly emerged drug-resistant variants[64] of viral proteases as well as newly-discovered and/or uncharacterized proteases, whose sequences are homologous to proteases of known structure. Energy-based specificity prediction will also aid in the design of protease variants targeted to specific substrates. Current approaches for protease design rely on library-based screening/selection[40,44,65] *in vivo*. These directed evolutionary trajectories often proceed via incremental "generalist"[66] intermediates that display relaxed specificity, and are, therefore, toxic to cells (or the proteases undergo self-cleavage) and are never identified in the selection. A structure-guided computational design approach based on the evaluation of interaction energies of substrates with protease variants should allow for multiple simultaneous substitutions ("jumps" in the sequence landscape) to allow specificity switching without generating generalist toxic intermediates. Combining structural computation using the discriminator described here with directed evolution should enable more efficient protease specificity design.

2.5. Methods

2.5.1. Curation of Benchmark Datasets

Each protease used in the study exhibits diverse mechanisms of action, interface recognition modes, varied folds and biological functions (Figure 2.10) – e.g. TEV Protease (cysteine proteases), HCV NS3 protease (serine proteases), Granzyme B (serine protease), HIV Protease-1 (aspartyl protease) and Matrix Metalloprotease -2 (Metalloprotease). The sequences of cleaved and uncleaved substrate peptides for each protease were obtained as detailed below:

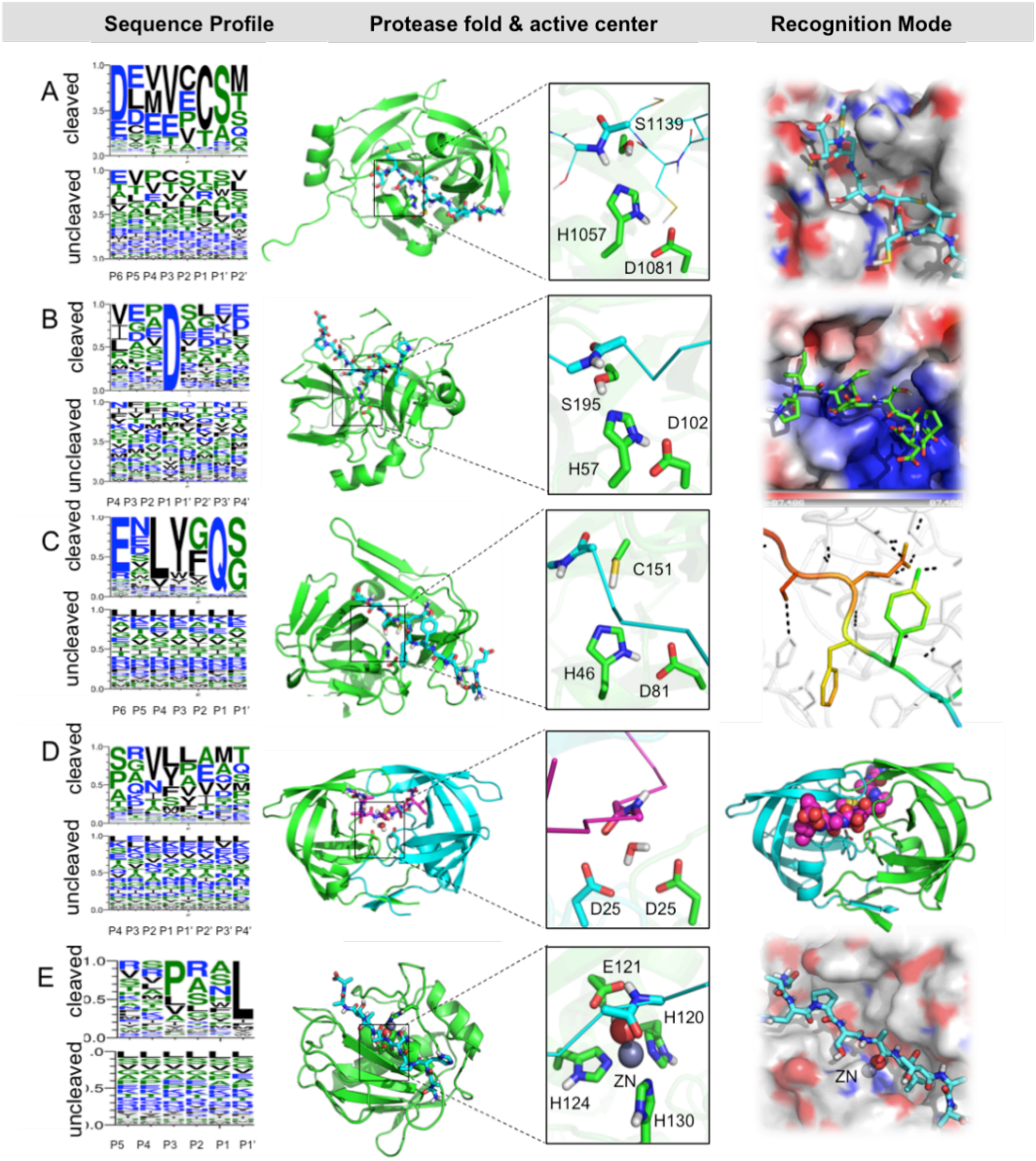


Figure 2.10. The cleaved and uncleaved dataset distributions, model generation and active site geometry of the starting crystal structure and mode of recognition of proteases used in the study.
(A) HCV Protease (PDB ID: 3M5N), a serine protease shows recognition via interfacial hydrogen bonding. (B) Granzyme B (PDB ID:1FI8) a serine protease shows an electrostatic mode of substrate recognition (C) TEV Protease, (PDB ID:1LVB), a cysteine protease displaying extensive hydrogen bonding at the protease-substrate interface (E) HIV Protease I (PDB ID: 1MT9), a symmetric aspartyl protease, working via proposed recognition mechanism - substrate-envelope hypothesis. (F) MMP2 (PDB ID: 3AYU), a zinc catalytic center

HCV protease: We obtained the cleaved and uncleaved sequence sets from a deep sequencing study by Shiryaev et al[41]. Only sequences with signals above a threshold (Z-score value> 3) at all three time points in their study were considered to avoid noise from deep sequencing analyses. We also incorporated sequences from a study by Rögnvaldsson et al[42]. Merging both individual sets generated a set with 196 cleaved and 1943 uncleaved sequences.

HIV-PR: 374 cleaved and 1251 uncleaved sequences were obtained from Rögnvaldsson *et al*.[42] .

TEV protease: The cleaved set of 68 sequences was curated from results obtained by Kostallas *et al*.[43] and Boulware *et al*.[44]. Due to the absence of a large uncleaved sequence dataset for the TEV protease, we synthetically generated the uncleaved dataset using a two-residue walk on the TEV polyprotein sequence. The TEV protease is expected to cleave only at one specific site in the polyprotein. Half of the sequences were randomly discarded to generate a dataset of 1520 uncleaved sequences. We ensured that the sequence distribution was not biased toward any specific amino acid type at any peptide position (Figure 2.10).

Granzyme B: The cleaved sequence set was obtained and uncleaved sequence set was adapted from Barkan *et al*.[36]. A subset of the uncleaved sequences was randomly chosen and the amino acid identity at P1 was randomly mutated to all amino acid identities except aspartate and glutamate. A total of 353 cleaved and 1973 uncleaved sequences were chosen.

Matrix Metalloprotease: The cleaved sequence set of 455 sequences was obtained from Ratnikov *et al*.[45]. To curate the uncleaved sequence set, we scanned the CutDB[67] database for MMP-2 protein substrates. Excluding the known cut sites in these proteins, the rest of the protein sequence was treated as uncleaved using a two-residue walk to generate an uncleaved sequence set of 1818 sequences for MMP-2.

2.5.2. Starting model generation for simulations:

We constructed models of peptide-protease bound complexes using high-resolution crystal structures culled from the Protein Data Bank (PDB) (Table 2.4)[9,6,68–70]. Crystal structures were filtered based on the following criteria: a resolution lower than 2.6 Å and a peptide or peptidomimetic inhibitor bound in the crystal structure. We remodeled the crystallographic conformation of the bound peptide to mimic the near-attack conformation for nucleophilic addition step of the proteolysis reaction by enforcing catalytic geometries obtained from mechanistic quantum mechanics simulations and/or crystal structures of proteases bound to inhibitors during Rosetta FastRelax simulations. The selected crystal structures were optimized using a Rosetta FastRelax protocol to find

a low energy, stable structure, which was used as a starting point in further calculations. Constraints were applied during FastRelax to maintain active site geometry and keep the protease in a catalytically active conformation. Co-ordinate constraints were also applied to the protease backbone to ensure that the structure does not drift away from the crystallographic conformation, while still minimizing energy, as previously described[71].

Table 2.4. Details of starting model generation for five proteases.

Protease	PDB ID	Resolution	Model Generation
HCV NS3 Protease	3M5L, 3M5N	1.9 Å	The P' residues of the bound peptide were built by overlaying PDB ID: 3M5N and PDB ID:3M5L (inhibitor bound crystal structure) thus allowing us to build a complete substrate bound complex
TEV Protease	1LVB, 1LVM	2.2 Å	Starting model generated from PDB by reverting C151A to WT
MMP2	3AYU, 1BQQ	2.0 Å	Starting model was generated by superimposing PDB ID: 1BQQ with PDB ID:3AYU(MMP2). The N terminal (P side) residues of the substrate were extended outward to build the complete substrate and were then relaxed to find an optimal substrate conformation
Granzyme B	1FI8	2.2 Å	The interface of the ecotin chain in the crystal structure, spanning eight residue substrate chain was used as the starting point for further calculations
HIV Protease 1	1MT9	2.0 Å	Starting model generated by inverting D25N and V82N from crystal structure to native residue identities

2.5.3. Calculating Rosetta and Amber energies

Starting from the relaxed crystal structure described above, we threaded the candidate peptide sequences to generate models of the protease-peptide complex corresponding to each sequence. The energy of the resulting conformation was minimized with constraints using Rosetta FastRelax and ten models were generated for each sequence. During this protocol, the protease backbone was constrained, protease side chains were allowed complete conformational flexibility, whereas peptide side chains and backbone were allowed to sample all degrees of freedom including backbone, sidechain and rigid body orientation with respect to the protease. The side chains of the catalytically active residues were constrained with respect to the scissile peptide bond of the substrate using enzyme design-style Rosetta constraints. This model represents a pre-transition state near-attack conformation for each of the peptide substrates for the protease. The resulting models were scored with Rosetta's Talaris2013 energy function.

Total residue energies for protease interface residues were extracted for all ten structures representing a single sequence, averaged and stored as "protease energy". Interface residues were defined as those whose C-alpha atom was within 8 Å of any peptide residue's C-alpha atom. We experimented with 8, 10, and 12 Å as the cutoff distance for defining the protease shell, but we found that the discriminator performance was robust to this cutoff value. The sum of total residue energies over all peptide residues was averaged and stored as "peptide energy". Total interface energy was defined as the sum of protease and peptide energies. These models were also scored for "constraint energy" based on the

deviation of active site residues geometries from idealized ones. Each energy term was used as a feature during machine learning (see below).

Sampling of the peptide backbone and protease and peptide side chains degrees of freedom was performed before calculating scores for a given complex structure. We optimized the structure sampling protocol by investigating several combinations of sidechain and backbone flexibility for the peptide and the protease, and their relative rigid-body transform. Allowing peptide backbone and sidechain flexibility, and protease sidechain flexibility afforded the highest discriminatory capability (Figure 2.4). All calculations were performed with the interface RosettaScripts[72,73]. Sample xml files used can be found in Appendix 1. The AMBER Tools 12 MMPBSA[74] application was used to calculate the electrostatic contribution to the bound state energy over the unbound energy for the protease–peptide complex. Run scripts are provided in Appendix 1.

2.5.4. Local sequence-structure compatibility

Rosetta's FragmentPicker[75] Tool was used to analyze the propensity of a peptide sequence to adopt an extended conformation that is found in protease active sites. We picked 200 fragments for a given peptide sequence, and calculated the RMSD of each fragment with the bound conformation of the peptide. The number of fragments with RMSD > 2.0 in the set of 200 top fragments compared to the bound conformation was used as the score.

2.5.5. Support Vector Machines

An SVM constructs a hyper plane between two sets of data points in multi-dimensional "feature" space, based on a predefined kernel function to maximally separate the two datasets. We used the built-in SVM function (MATLAB 2015) with a radial-based kernel function following Barkan *et al*.[36]. In the RBF kernel, parameters C and γ need to be adjusted: C, also called cost factor, is a regularization parameter that controls the trade-off between maximizing the margin and minimizing the prediction error, while γ is a kernel-type parameter that dominates the generalization ability of SVM by regulating the amplitude of the kernel function. We optimized the training parameters of SVM based on 5-cross-validation tests. C- and γ-values of 10 and 10, respectively, were used.

Sequence features: Each position within the sequence was considered to be one feature. The one letter amino acid codes were transformed into an index, which was calculated from the rank of the amino acid residue in an alphabetical ordering of all amino acids as well as on its position in the sequence from N to C terminus on the substrate chain as in Barkan *et al*.[36]. All 20 amino acids at each position in the peptide were assigned a number using the formula n*20+i, where n represents the position of the residue in the peptide sequence and i represents the position of the residue in an alphabetical ordering of amino acids by their one letter code.

***Structure features*:** Each contributing discriminator energy score was imported into the SVM as an independent feature. The structure-based Rosetta energies ("Interface residue peptide energy", "Interface residue protease energy", "Reorganization penalty",

"constraint energy") and Amber energy ("electrostatic energy") were used as features. The SVMs were cross-validated using an 80-20 bootstrap over 1000 iterations.

2.5.6. Generation of a computational library for HCV NS3/4A substrate from P6 through P2 positions

The mutational scanning was executed in two parts. We generated models of the protease–peptide complex for substrate positions P6 through P4, energy minimized and scored them using the computational protocol descried above. Ten structures were generated for each sequence. The models were evaluated using the weighted optimized energies as used in the discriminator. The top scoring 66 sequences were identified, and 26,400 models were generated by sampling P3 and P2 substrate positions for each sequence. These 26,400 models were subjected to energy minimization and score calculations as previously described. To calculate their final score, Rosetta interface energy, constraint energy, and AMBER MMPBSA electrostatic energy were used at the optimized Enrichment values. To reduce computational costs, the reorganization penalty score was not included in the final score calculation since it did not measurably change the auROC value in the benchmark set (Figure 2.3). The sequences that lay in the score distribution of the native cleaved sequences were further analyzed. These were filtered to be most different from the initial HCV cleaved sequence distribution and clustered using Hamming distance into 4 main sequence pools- CED*, LEE*, FED*, YED*. Representative sequences from the first three sequence clusters were tested experimentally.

2.5.7. Flow Cytometry:

We used the Yeast ER Sequestration and Screening Assay (YESS) for *in vivo* testing of predicted substrates of the HCV protease. The LY104 construct for the assay was a gift from Y. Li, B. Iverson, and G. Georgiou (University of Texas at Austin). The sequences to be tested were cloned into LY104 using a Restriction Free Cloning method[76]. Table 2.5 lists all the primers associated with the cloning protocol.

Table 2.5. Primers used for molecular cloning the sequences to be tested in the YESS assay into the assay (LY104) vector using RF cloning.

Sequence	Primers
LEEFFCSG	FOR: CGGTAGCGGAGGCGGAGGGTCGTTGGAAGAATTCTTCTGTTCAGGC
	REV: CTGCCTTTATCATCATCATCTTTATAATCACTGCCGCCTGAACAGAAGAATTCTTCC
LEEYQCSG	FOR: CGGTAGCGGAGGCGGAGGGTCGTTGGAAGAATATCAATGTTCAGGCG
	REV: CTGCCTTTATCATCATCATCTTTATAATCACTGCCGCCTGAACATTGATATTCTTCCAA
CEDYFCSG	FOR: CGGTAGCGGAGGCGGAGGGTCGTGTGAAGATYMTTTCTGTTCAGGCG
	REV: CTGCCTTTATCATCATCATCTTTATAATCACTGCCGCCTGAACAGAAAKRATCTTCACA
FEDFQCSG	FOR: CGGTAGCGGAGGCGGAGGGTCGTTCGAAGATTTCCAATGTTCAGGC
	REV: CTGCCTTTATCATCATCATCTTTATAATCACTGCCGCCTGAACATTGGAAATCTTCG

The positive control and test plasmids were then transformed into the EBY100 competent yeast strain. They were plated on selective complete (SC) media (20 g/L glucose) with a

selective amino acid mix (-Trp, - Ura). After two days of growth, a single colony was transferred to a 2 mL SC media culture tube supplemented with 2 μL of 1000x antibiotics (carbenicillin, kanamycin). The growth cultures were incubated for ~24h (OD_{600} 2.0 – 3.0) in a 30 ºC shaking incubator. 1.5 x 10^7 cells(OD_{600} ~0.5) were pelleted and resuspended in 2 mL induction media (20 g/L galactose, 2 g/L glucose) supplemented with 2 μL each of 1000x antibiotics (carbenicillin, kanamycin). The induction cultures were grown overnight at 30 ºC to an OD_{600} of 1-1.5. All spins in the protocol were done at 3000 r.c.f for 5 min. The induced cultures were pelleted and washed with 500 μL PBS followed by 500 μL PBS+ 0.5% BSA. 1 μL of each antibody stain (anti-FLAG, anti-HA) was incubated with 10^7 cells for 30 min at 4 ºC. The samples were resuspended by vortexing and incubated at RT for an additional 30 min. The cells were washed with 100μL PBS with 0.5% BSA, pelleted and then resuspended in 500 μL PBS. Samples were diluted to achieve a final concentration of 10^6 cells/mL and then FITC (anti-HA) and PE(anti-FLAG) intensities were detected using a Flow Cytometer (Beckman Coulter Gallios).

Chapter 3. MFPred: Rapid and Accurate Prediction of Protein-peptide Recognition Multispecificity Using Self-Consistent Mean Field Theory

3.1. Abstract

Multispecificity – the ability of a single receptor protein molecule to interact with multiple substrates – is a hallmark of molecular recognition at protein-protein and protein-peptide interfaces, including enzyme-substrate complexes. The ability to perform structure-based prediction of multispecificity would aid in the identification of novel enzyme substrates, protein interaction partners, and enable design of novel enzymes targeted towards alternative substrates. The relatively slow speed of current biophysical, structure-based methods limits their use for prediction and, especially, design of multispecificity. Here, we develop a rapid, flexible-backbone self-consistent mean field theory-based technique, MFPred, for multispecificity modeling at protein-peptide interfaces. We benchmark our method by predicting experimentally determined peptide specificity profiles for a range of receptors: protease and kinase enzymes, and protein recognition modules including SH2, SH3, MHC Class I and PDZ domains. We observe robust recapitulation of known specificities for all receptor-peptide complexes, and comparison with other methods shows that MFPred results in equivalent or better prediction accuracy with a ~10-1000-fold decrease in computational expense. We find that modeling bound peptide backbone flexibility is key to the observed accuracy of the

method. We used MFPred for predicting with high accuracy the impact of receptor-side mutations on experimentally determined multispecificity of a protease enzyme. Our approach should enable the design of a wide range of altered receptor proteins with programmed multispecificities.

3.2. Introduction

Many natural proteins, including signal transduction hubs and enzymes that process biological information, have evolved to be multispecific – they participate in specific interactions with several interaction partners[77,78]. Evolution of multispecificity includes selection for both positive and negative specificity, involving recognition and non-recognition, respectively, of sets of interaction partners[15]. Most multispecific interactions arise when the active site of a single receptor protein interacts with multiple binding partners of differing sequence[79]. Nature uses structurally conserved protein-recognition domains (PRDs), e.g., SH2, SH3 and PDZ domains, to mediate many multispecific interactions[80–85]. Thus, it is crucial that methods that model and modulate PRD specificity are able to accurately recapitulate their multispecific nature.

Similar to cascades composed of multispecific PRDs like SH3, SH2 and PDZ domains that mediate signal transduction, proteolytic cascades are ubiquitous in the post-translational transduction of biological information[1]. Protease activity and selectivity is involved in a diverse range of biological processes including digestion, blood clotting, apoptosis and cancer[86–89]. Proteases are inherently multispecific such that they recognize and proteolyze (or cleave) a range of substrates (positive specificity) while not

recognizing others (negative specificity)[15]. For example, viral proteases such as HCV protease that are involved in viral maturation cleave only specific sites in the viral polyprotein but do not cleave others[90]. These proteases may also have evolved the ability to cleave specific host proteins[91]. Prediction of protease multispecificity is, therefore, key for identifying their substrates under healthy and disease conditions. Additionally, designed proteases with programmed multispecificity have the potential to be used as therapeutics and protein-level knockout reagents in cell culture[92]. The ability to manipulate protease specificity computationally would enable the creation of such designer proteases with dialed-in recognition specificity, thereby providing tools to interrogate and intervene in biological processes.

Rational modulation of protein-protein or protein-peptide interaction multispecificity has met with limited success, except in a few notable cases, such as coiled-coil interfaces[93,94]. In principle, computational structure-based modeling methods should be able to recapitulate and modulate multispecificity. In fact, several methods relying on, among others, Monte-Carlo (MC) simulations in sequence and conformation space, and genetic algorithms (GA) have been developed to predict PRD multispecificity[56,59,95–97]. However, these methods are limited by the time required to enumerate a sufficiently large number of sequences to sample the substrate/peptide sequence space. As multispecific design entails additional sampling of (thousands) of receptor variants and modeling the multispecificity of each variant separately, using current methods to design receptors for and against specificity profiles is not computationally feasible.

We have developed a structure-based method that eliminates the expense of explicit sequence enumeration in multispecificity modeling. The method uses a self-consistent **M**ean-**F**ield theory-based **Pred**iction (MFPred) approach that expresses specificity as a sitewise probability distribution function that can be calculated relatively rapidly. We have benchmarked MFPred on four diverse proteases and compared the results to MC- and GA-based methods. MFPred has comparable accuracy to MC-based and GA-based methods and provides a tens- to thousands-fold speedup. We demonstrate the generality of MFPred by obtaining significant multispecificity predictions for five diverse classes of protein-recognition domains (PRDs). Finally, as a proof-of-concept for design, we demonstrate that MFPred can recapitulate experimentally determined changes in specificity profiles due to receptor-side mutations.

3.3. Results

3.3.1. Self-Consistent Mean Field Theory-Based Specificity Profile Prediction Algorithm

To predict the specificity profile, we consider an ensemble of peptide backbone conformations bound to a receptor. For each peptide backbone conformation, we simultaneously sample all rotameric conformations of all amino acids at all peptide residue positions while keeping the receptor backbone and sidechains in their crystallographic conformations. The sidechain conformations at a given peptide position are sampled in the “mean field” of all other sidechain conformations at all other positions and (fixed) receptor residues, as described in Methods. Next, the contribution of each peptide backbone conformation at each peptide position is accounted for by Boltzmann

averaging the mean-field specificity profile solution obtained in the previous step. The final specificity profile is constructed by combining these individual predictions. While the sequence specificity prediction described here can be performed using any (pairwise decomposable) energy function, we implemented our prediction method in the context of the Rosetta modeling suite, thus combining its sophisticated energy function with the speed of mean-field sampling (Figure 3.1).

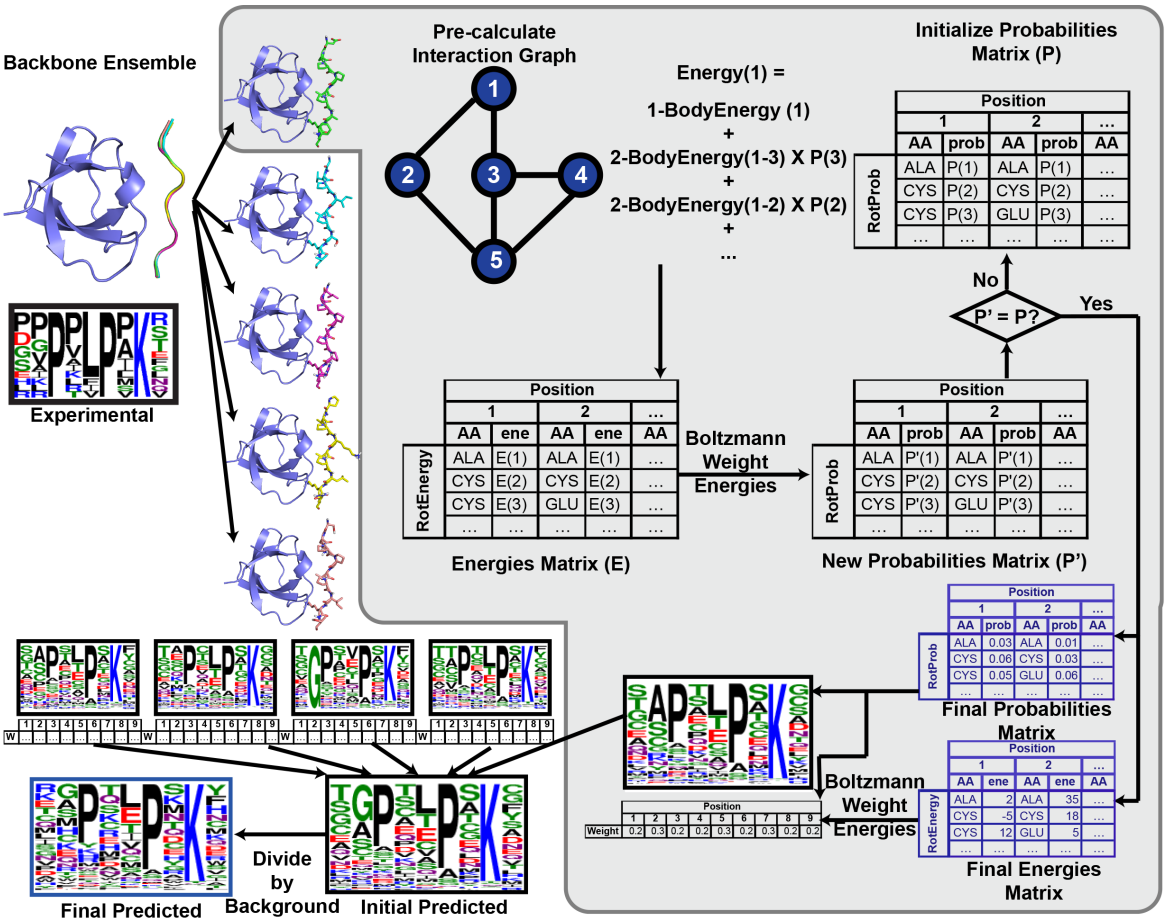


Figure 3.1. MFPred workflow.
MFPred input is a backbone ensemble of a protein/peptide complex, which is generated from a protein structure from the PDB (1CKA here) as described in Methods. For each backbone, Rosetta pre-calculates the interaction graph, which stores intrinsic rotamer one-body energies on the vertices (blue circles) and matrices of rotamer-rotamer two-body energies on the edges (black lines). A probabilities matrix (P) is initialized. Mean-field energies (E) are calculated using the interaction graph and P, and a new matrix, P' is generated from E. If P' is equal to P, convergence has been reached. If not, the process is

repeated by updating P with a combination of P and P'. Once convergence is reached, the final energies matrix and probabilities matrix is used to generate the Boltzmann weights of each backbone position, which is then used to average all the backbone specificity profiles together. This specificity profile is divided by the background specificity profile to reach the final predicted specificity profile.

3.3.2. Rationale for Choice of Benchmark Datasets

To test our MFPred method, we sought to first recapitulate experimentally determined specificity profiles of a variety of PRDs. We chose PRDs where both structural as well as specificity information has been experimentally determined. We focused primarily on protease enzymes for methodology development, and tested the generality of our approach with previously developed benchmarks for multispecificity prediction on PRDs such as a kinase enzyme, and SH3, SH2, MHC, and PDZ domains.

3.3.2.1. Protease set. We benchmarked our method on four protease enzymes that had both high-resolution crystal structures with a bound peptide in the Protein Data Bank (PDB) and experimental cleavage data (see Methods for details). The chosen proteases represent the vast diversity seen in structural fold, biological function, and mechanism of action amongst the protease enzyme family (Figure 3.2). Additionally, there is a mix of highly conserved and less specific positions among their specificity profiles, thus enabling us to determine how well MFPred performs with regard to varying degrees of flatness in the experimental specificity profile.

3.3.2.2. Testing on protein-recognition domains. To test the generality of the MFPred method, we curated a dataset consisting of a variety of non-protease PRDs that had high-resolution crystal structures as protein-peptide complexes in the PDB and experimental

binding specificity data available. We tested fourteen PRDs that comprise five classes of PRDs: kinases, SH2 domains, SH3 domains, PDZ domains, and MHC-I proteins. Including these diverse domains allows us to test the method on a range of underlying recognition modes, binding affinities and specificities; while proteases bind with relatively high dissociation constants to their substrates (K_M ~10 uM), SH2 domains have been known to bind with dissociation constants as low as 0.3 nM[98].

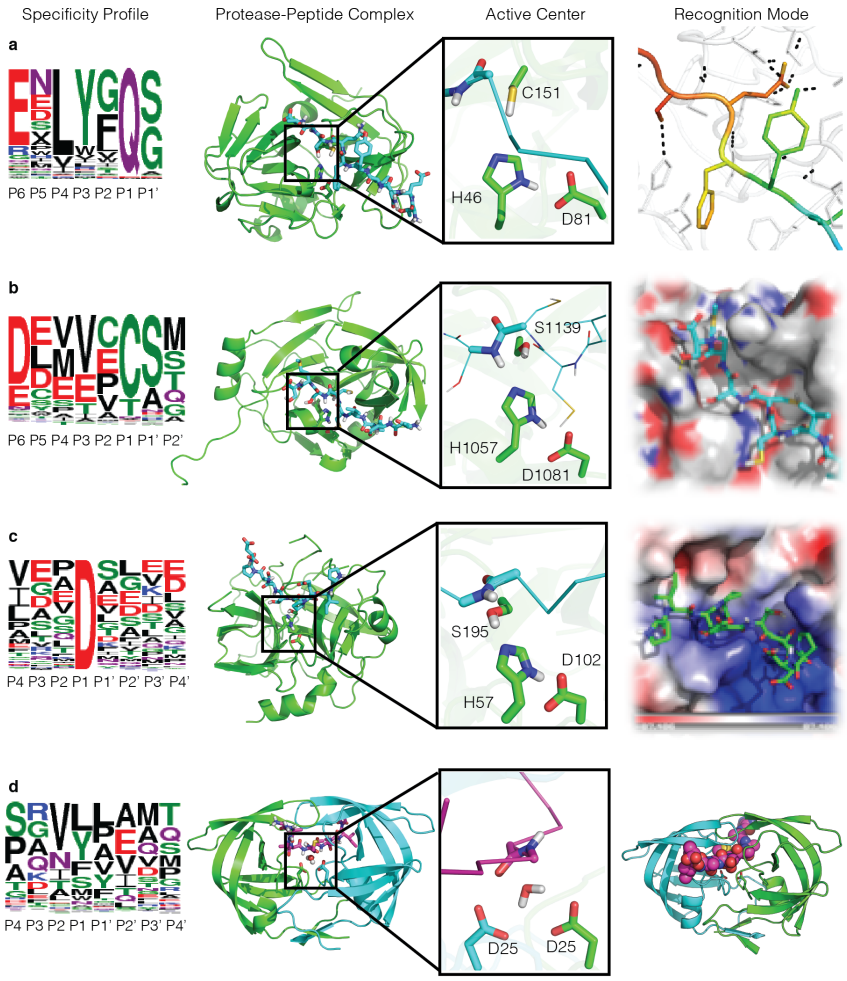


Figure 3.2. Protease benchmark specificity profiles, models, active centers, and recognition modes.
(a) Tobacco etch virus (TEV) protease is a cysteine protease displaying extensive hydrogen bonding recognizes substrates via interfacial hydrogen bonding at the protease substrate interface. (b) Hepatitis C virus (HCV) NS3 protease, a serine protease,

recognizes substrates through electrostatic interactions. (c) Granzyme B, a serine protease, recognizes substrates through electrostatic interactions. (d) Human immunodeficiency virus (HIV) protease I, a symmetric aspartyl protease, has been proposed to recognize substrates via the substrate – envelope hypothesis.

The binding specificities and mechanisms for each of these domains are distinct, thereby adding to the diversity of the test set. PDZ domains bind up to 7 C-terminal residues in a highly specific manner[82]. SH3 domains bind proline-rich regions that often form PPII helices[85]. SH2 domains show a preference for pTyr-containing peptides[99], while the context surrounding the pTyr residue determines the specificity of the peptide towards a distinct SH2 domain[100]. Kinases are one of the largest families in the eukaryotic genome and share a common fold that allows for the binding of ATP and a Ser, Thr, or Tyr residue-containing substrate[101]. Finally, MHC-I domains bind short pathogenic peptides to be presented to cytotoxic T lymphocytes (CTLs). MHC-I domains are promiscuous and may bind many peptides; generally, one or two substrate positions are conserved, while others are tolerant to mutations[102].

3.3.3. Choosing Metrics for Evaluation of Prediction Accuracy

We evaluated the performance of MFPred by quantifying the differences between predicted and experimentally determined specificity profiles using several metrics (see Appendix 3 for details). Four of these metrics, the cosine similarity, Frobenius norm, average absolute distance (AAD) and Jensen-Shannon divergence (JSD) are correlated, as shown in Figure 3.3. The Frobenius norm and AAD are distance-based metrics that have been used previously to compare profiles[56,95]. The Frobenius norm is more sensitive to flatness in the specificity profile than the AAD (Figure 3.4). Additionally, we

evaluated the profiles by their cosine similarity, which is another distance-based metric that is less sensitive to flatness than either AAD or Frobenius norm. It falls between 0 and 1, where 0 denotes a random prediction and 1 denotes a perfect prediction. The Jensen-Shannon divergence (JSD) has also been used in the past to evaluate profiles[95]. We used cosine distance as the general score of a profile, as it is easy to visualize and interpret. We evaluated the significance of each positional JSD score by scoring 100,000 random profiles against the experimental profile and thus determining the *p*-value of the JSD score (see Appendix 3 for details).

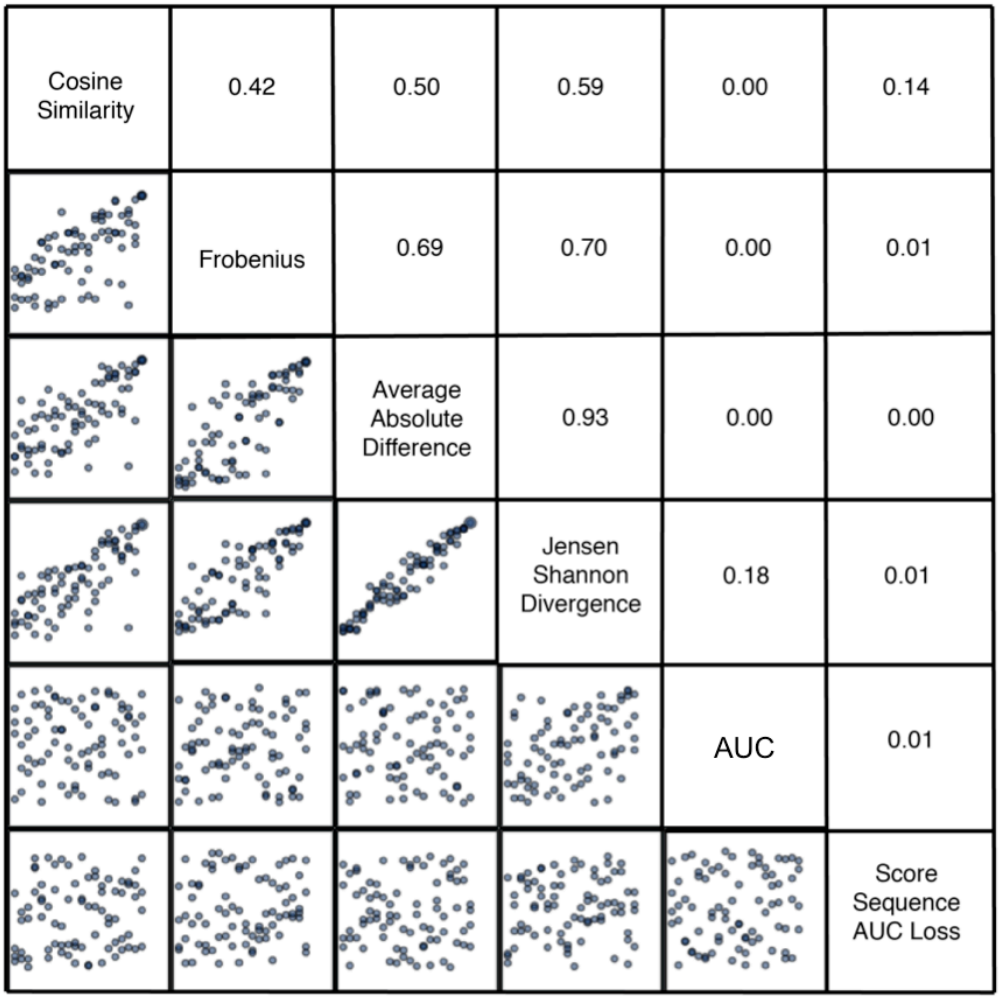


Figure 3.3. Specificity profile metric correlation
Correlation coefficients between pairs of metrics are shown in the upper diagonal while scatterplots are shown in the lower diagonal. Cosine similarities and AUC values are

shown as 1 – cosine and 1- AUC, respectively, so that a lower value represents a better prediction. Scatterplot points are colored by the number of bits in the predicted profile, with a darker blue representing fewer bits, or more peaked profiles

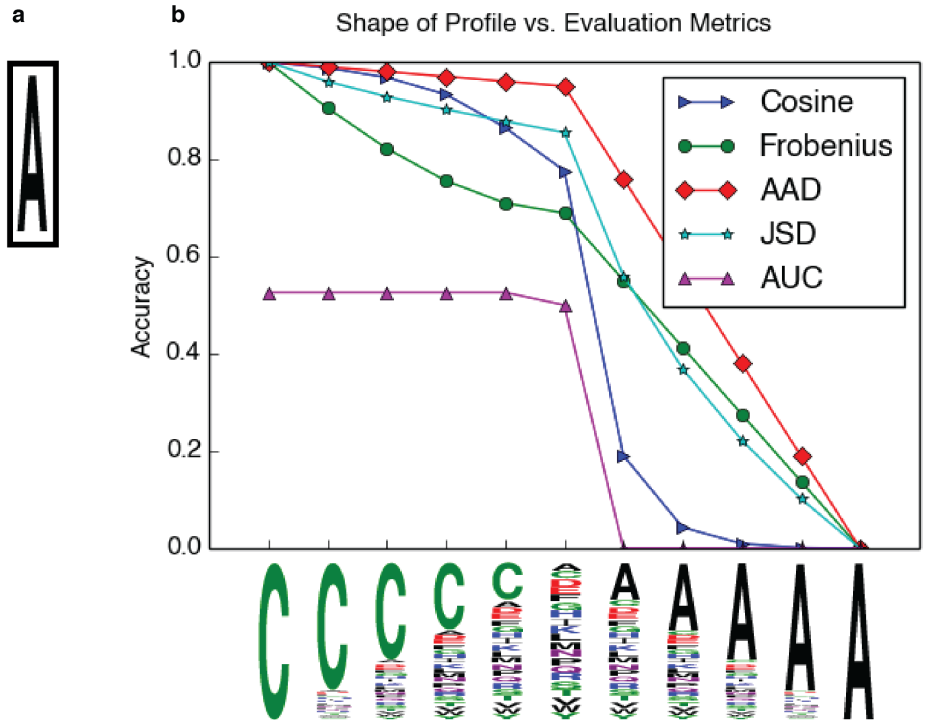


Figure 3.4. Profile shape affects evaluation metrics differently
(a) "Experimental" profile to compare to. (b) Each metric is affected differently by the shape of the profile (x-axis). Accuracy is normalized for all metrics so that the worst value for that metric corresponds to one. Both AUC and cosine are subtracted from 1, as well, to make the metrics consistent. Cosine similarity varies slightly with regard to flatness of the profile, whether or not the most frequent amino acid is correct. Frobenius distance varies more than the cosine similarity; it decreases somewhat consistently with the shape of the profile. While AAD does not vary much as a function of flatness when the most frequent amino acid is incorrect, it decreases very quickly when the most frequent amino acid is correct. JSD also varies more when the most frequent amino acid is correct, although to a lesser extent than AAD. AUC is relatively unaffected by flatness in the predicted profile; if the most frequent amino acid is incorrect, AUC is ~0.5 (or random), and if the most frequent amino acid is correct, it is zero.

We also used a second metric as a general score for each profile: area under the ROC (receiver operating characteristic) curve (AUC) is a non-distance-based metric that evaluates predictions based on their ranking more tolerated amino acids correctly[56]. It is

relatively unaffected by flatness (Figure 3.4) but will not evaluate well if either the experimental or predicted profile is close to uniform. It is not correlated with the above metrics. Additionally, we developed a new metric, Score Sequence AUC Loss (SSAL), which encapsulates the efficacy of the predicted specificity profile in differentiating between substrates which are recognized and cleaved by a given protease (cleaved sequences) and substrates which are not cleaved by that protease (uncleaved sequences). A perfect prediction scores an SSAL of zero. It does not correlate well with any other metric (Figure 3.3).

3.3.4. Recapitulation of protease specificity profiles

Proteolysis is a multi-step reaction, which involves substrate peptide binding, the formation of a tetrahedral intermediate (acylation) and hydrolytic cleavage of the tetrahedral intermediate (deacylation). We have previously found that modeling a near-attack conformation for the acylation step was successful in discriminating between known cleaved and uncleaved peptides[103]. Therefore, starting from structures of protease-substrate complexes in a near-attack conformation, we performed MFPred-based specificity prediction.

We found that MFPred robustly recapitulates protease specificity profiles (Figure 3.5b) in our benchmark set. The cosine similarities of the entire profiles range from 0.66 to 0.89, AUC ranges from 0.73 to 0.86, and SSAL ranges from 0.21 to 0.002. Out of 31 substrate positions across the protease dataset, 20 were predicted with a significant JSD *p*-value. The best prediction is obtained for the common biotechnologically used protease TEV-

PR. The predicted profile has a high cosine similarity of 0.89 (1 would be a perfectly accurate prediction). The primarily steric and hydrogen-bonding-based nature of molecular recognition at TEV-PR-substrate interfaces is well suited to the strengths of the Rosetta energy function underlying MFPred. Similarly, the profiles of HCV protease and granzyme B (GrB) protease are also generally recapitulated with a high degree of accuracy, except for positions with no marked preference for specific amino acids (flat positions) – positions P5 and P2 in HCV protease and positions P4, P1', and P2' in granzyme B protease. We attribute the lack of correlation at these flat positions to small errors in energy evaluations being equivalent to the size of the energy gaps being modeled, thus leading to erroneous ranking. Challenges in measuring prediction accuracy at flat positions have indeed been noted before[56].

The worst performance among the proteases in the benchmark set is observed for the prediction of HIV protease-1 (HIVPR1) specificity. This protease is known to have a relaxed specificity profile, with preference for small hydrophobic residues at P1 and P1' positions. The cavity of HIV protease-1 is large and peptides may adopt large variations in backbone conformation depending on their sidechains. Additionally, substrate binding involves flexibility on the protease side, with two loops ("flaps") that are mobile and close over the binding pocket. Incorporation of greater backbone flexibility on both the receptor and peptide parts of the HIVPR1-peptide interface may help improve predictions, as previously observed by us and others[47,62,103].

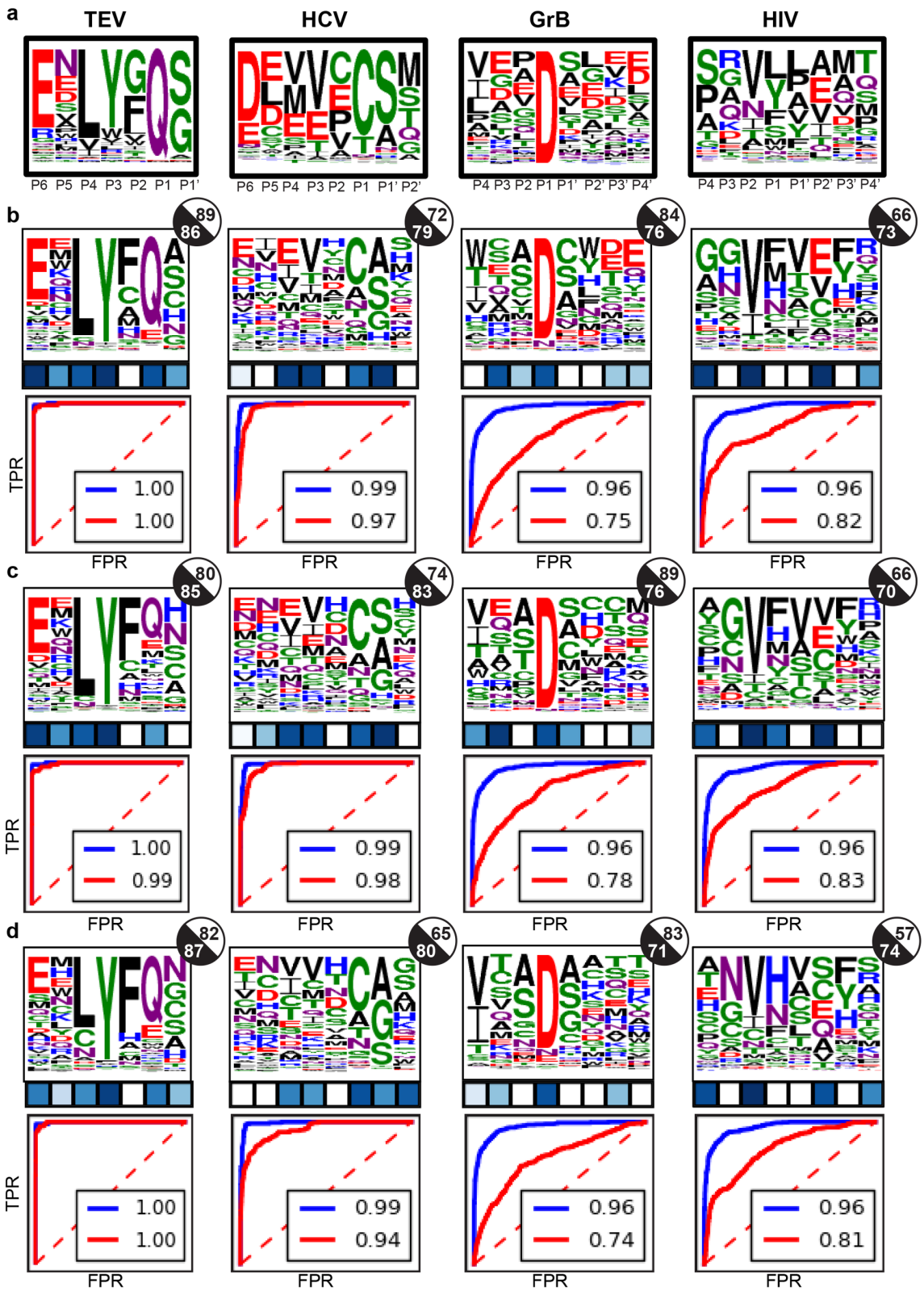


Figure 3.5. Comparison of backbone ensemble generation methods.

(a) Experimental specificity profiles. **(b)** MFPred on FastRelax backbone ensemble. The *p*-value of the JSD for a given position is represented by the color of the square under that position; white denotes a *p*-value > 0.5 and dark blue denotes a *p*-value of 0. A given circle to the right of a profile represents the cosine similarity (white) and AUC (black) of that profile. The ROC plots beneath each profile depict the SSAL calculation *via* the experimental ROC (blue) and predicted ROC (red) with their respective AUC values. **(c)** MFPred on FlexPepDock backbone ensemble. **(d)** MFPred on Backrub backbone ensemble.

3.3.5. Modeling Backbone Flexibility is Key for Prediction Accuracy

To determine the contribution of modeling backbone flexibility to the accuracy of prediction and to investigate if backbone sampling could be optimized for specificity prediction, we generated MFPred profiles with different levels of backbone flexibility.

First, we found that predictions generated by starting from a single crystallographically-determined backbone structure for the peptide led to poor accuracy for HCV and HIV proteases (Figure 3.6f, h), indicating that incorporating peptide backbone diversity is a key requirement for the observed accuracy of prediction. Second, we generated peptide backbone ensembles by threading on a varying number of known substrate (cleaved) peptides using three different Rosetta-based backbone sampling protocols (FastRelax[46], FlexPepDock[104], and Backrub[105]) separately to further diversify the peptide backbone ensemble. In each case, geometric constraints[103] were used to limit the scissile peptide bond to a near-attack conformation and the catalytic residues to an active conformation. The MFPred simulations were then performed on all backbone ensembles and their results were compared to each other (Figure 3.5, Table 3.1).

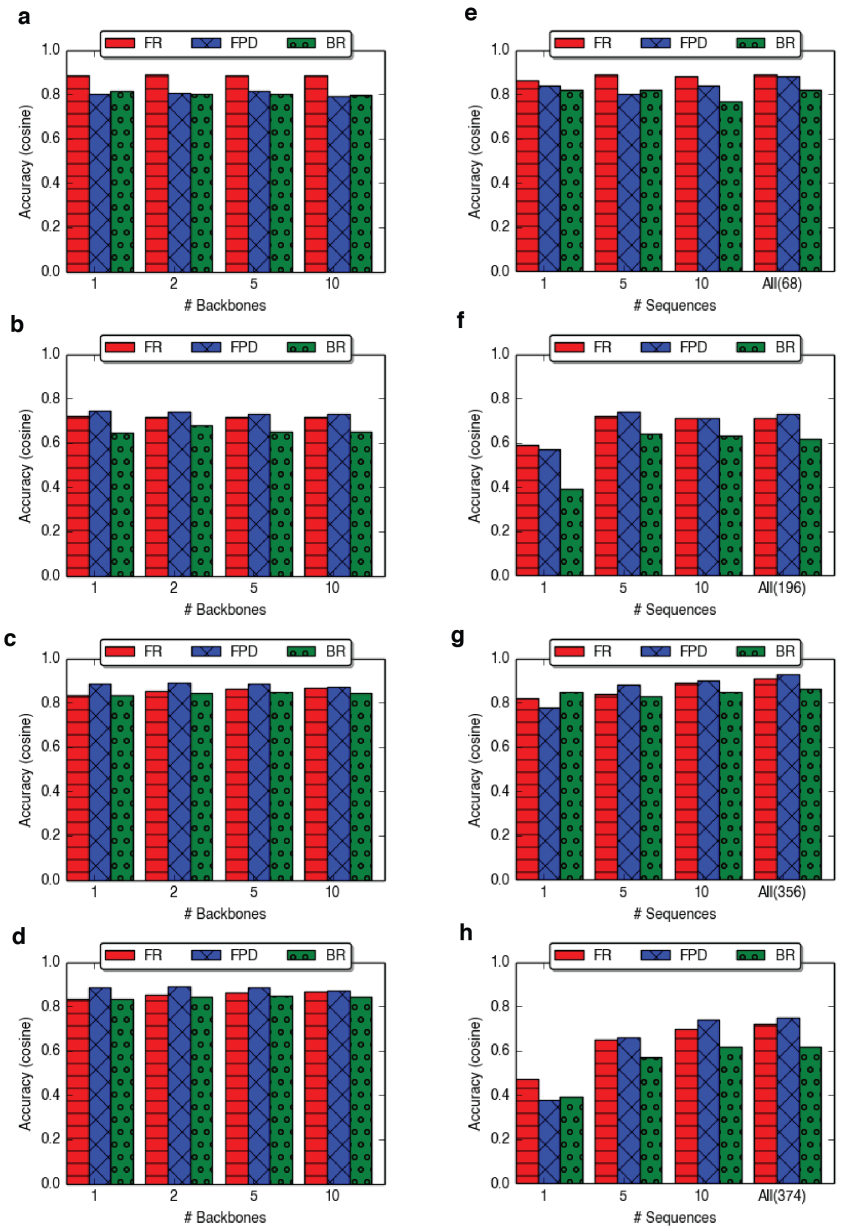


Figure 3.6. Number of sequence vs. accuracy and number of backbones vs. accuracy for methods of backbone ensemble generation

(a)- (d) Number of backbones per sequence vs. accuracy for TEV, HCV, Granzyme B and HIV, respectively. Each protocol begins with five sequences, which are then relaxed using FR, FPD or BR 1,2,5 or 10 times each. (e)-(h) Number of sequences vs. accuracy for TEV, HCV, Granzyme B and HIV, respectively. Number of sequences is varied over 1,5,10 all experimentally derived sequences, which is different for each protease.

Table 3.1. Results of all methods of backbone generation - FastRelax (FR), FlexPepDock (FPD), and backrub (BR) - on variously-sized backbone ensembles.

Protease	Method	#Seq	Cosine	Frob	AA	JSD	AUC	SSAL	Bits
TEV	**FR**	1	0.86	1.06	0.04	0.22	0.87	0.00	0.43
		5	0.89	0.85	0.04	0.21	0.86	0.00	-0.34
		10	0.88	0.86	0.04	0.20	0.91	0.00	-0.55
		All	0.89	0.84	0.03	0.20	0.91	0.00	-0.69
	FPD	1	0.84	1.08	0.04	0.23	0.86	0.00	0.23
		5	0.80	1.10	0.04	0.27	0.85	0.01	-0.64
		10	0.84	0.99	0.04	0.24	0.91	0.00	-0.64
		All	0.88	0.87	0.04	0.20	0.91	0.00	-0.72
	BR	1	0.82	1.11	0.04	0.25	0.84	0.00	-0.06
		5	0.82	1.06	0.05	0.26	0.87	0.00	-0.70
		10	0.77	1.17	0.05	0.29	0.89	0.00	-0.91
		All	0.82	1.06	0.05	0.27	0.89	0.00	-0.87
HCV	**FR**	1	0.59	1.37	0.06	0.35	0.77	0.08	-0.51
		5	0.72	1.13	0.05	0.31	0.79	0.02	-1.28
		10	0.71	1.15	0.05	0.30	0.82	0.02	-1.28
		All	0.71	1.14	0.05	0.29	0.84	0.02	-1.29
	FPD	1	0.57	1.45	0.06	0.35	0.76	0.09	-0.39
		5	0.74	1.10	0.05	0.30	0.83	0.02	-1.29
		10	0.71	1.14	0.05	0.30	0.80	0.01	-1.29
		All	0.73	1.12	0.05	0.28	0.87	0.01	-1.35
	BR	1	0.39	1.67	0.06	0.44	0.69	0.17	-0.83
		5	0.64	1.23	0.05	0.32	0.80	0.05	-1.20
		10	0.63	1.25	0.06	0.32	0.81	0.04	-1.22
		All	0.62	1.26	0.05	0.32	0.81	0.05	-1.31
GrB	**FR**	1	0.82	0.85	0.04	0.23	0.71	0.20	0.60
		5	0.84	0.73	0.04	0.20	0.76	0.21	0.07
		10	0.89	0.60	0.03	0.17	0.80	0.17	0.06
		All	0.91	0.53	0.03	0.13	0.87	0.15	-0.08
	FPD	1	0.78	1.04	0.04	0.25	0.72	0.19	0.83
		5	0.88	0.62	0.03	0.17	0.76	0.18	0.10
		10	0.90	0.59	0.03	0.15	0.80	0.17	0.02
		All	0.93	0.49	0.03	0.11	0.83	0.13	-0.08
	BR	1	0.85	0.74	0.04	0.22	0.71	0.19	0.38
		5	0.83	0.74	0.04	0.20	0.71	0.22	0.14
		10	0.85	0.70	0.04	0.19	0.72	0.22	0.09
		All	0.86	0.68	0.04	0.18	0.72	0.21	0.08
HIV	**FR**	1	0.47	1.55	0.06	0.42	0.66	0.17	0.96
		5	0.65	0.96	0.05	0.27	0.73	0.14	-0.01
		10	0.70	0.88	0.04	0.23	0.78	0.08	-0.04
		All	0.72	0.82	0.04	0.21	0.81	0.05	-0.21
	FPD	1	0.38	1.78	0.07	0.47	0.69	0.22	1.22
		5	0.66	0.96	0.05	0.28	0.70	0.13	-0.04
		10	0.74	0.81	0.04	0.22	0.78	0.07	-0.18
		All	0.75	0.77	0.04	0.19	0.83	0.05	-0.32
	BR	1	0.39	1.48	0.06	0.41	0.67	0.23	0.47
		5	0.57	1.06	0.05	0.30	0.74	0.15	-0.04
		10	0.62	0.98	0.05	0.27	0.73	0.14	-0.11
		All	0.62	0.96	0.05	0.27	0.73	0.11	-0.16
Most Similar			1.00	0.00	0.00	0.00	1.00	0.00	0.00
Most Different			0.00	$\sqrt{(2n)}$	0.06	1.00	0.00	1.00	4.32

[1]n refers to the number of positions in the profile

While the algorithm is relatively robust to the method of backbone generation as long as scissile bond geometry is maintained, the FastRelax (FR) protocol has a small improvement in overall performance over the FlexPepDock (FPD) protocol, with 20 significant *p*-values (out of 31) for FR vs. 19 for FPD, and FPD has a minor increase in overall performance over Backrub (BR), with 19 significant *p*-values for FPD vs. 18 for BR. The profile for TEV-PR is predicted best by FR, due to better prediction of Q at P1 and S at P1'. In the case of HIV protease-1, FR recapitulates the profile better than FPD and BR do. However, the performance of FPD is marginally better than that of FR and significantly more accurate than that of BR in the cases of HCV protease and granzyme B protease.

To determine how MFPred accuracy depends on the number and sequences of known cleaved substrates used to generate the backbone ensemble, we generated a peptide backbone conformational ensemble that was independent of peptide sequence. For all positions on the peptide backbone, we enumerated every combination of phi/psi dihedral angles that were x-15, x, and x+15, where x is the dihedral angle of the relaxed crystal structure peptide backbone. The resulting structures were filtered to remove those with clashes and to preserve hydrogen-bond interactions. The remaining structures were further clustered by all-heavy-atom RMSD of the peptide residues (see Appendix 2 for details) and MFPred was performed on the cluster centers. The resulting predictions are significantly less accurate than those of FR, FPD, or BR (Figure 3.7), indicating that successful prediction requires a backbone ensemble that is optimally positioned in the binding site for cleavage.

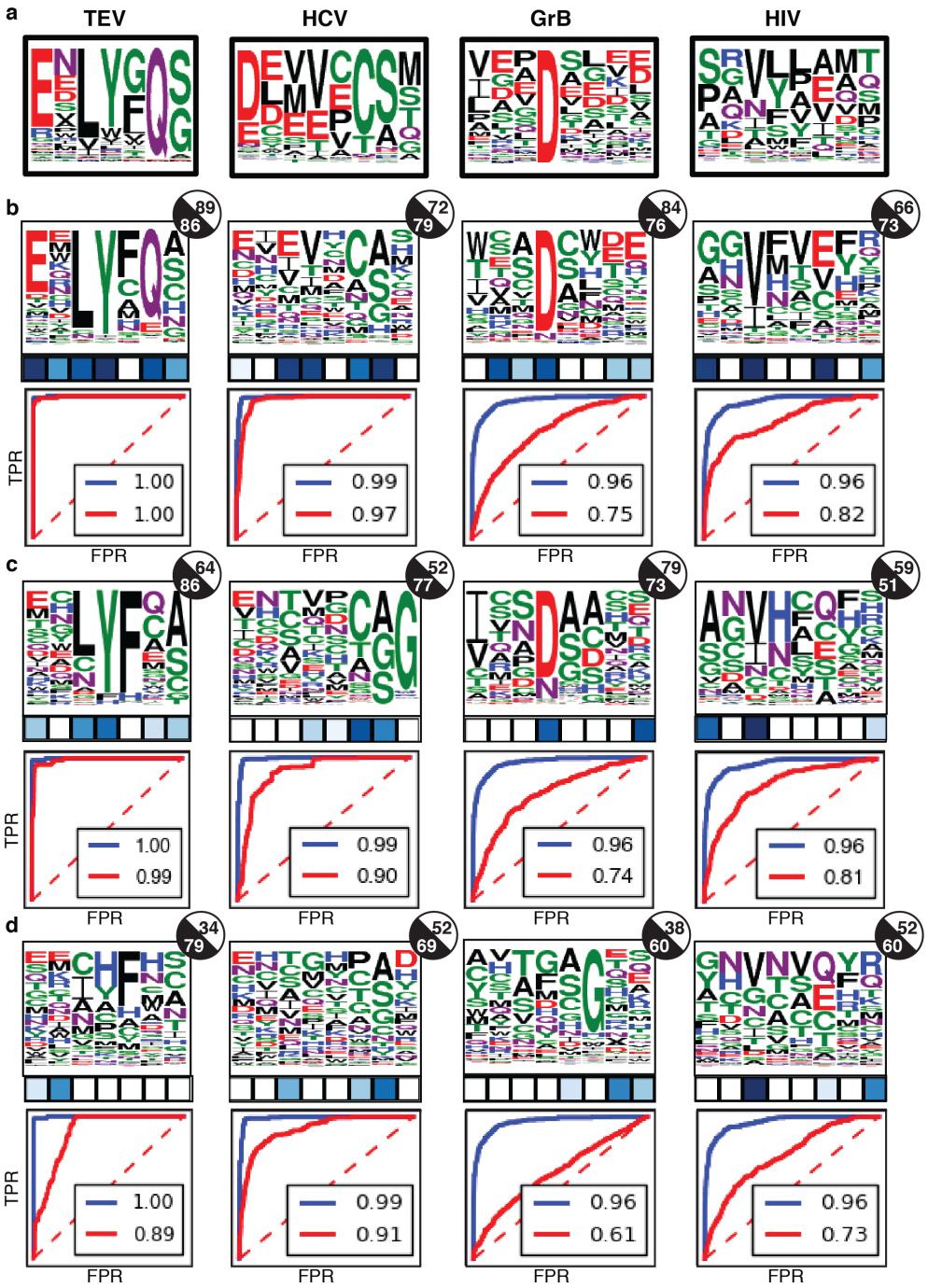


Figure 3.7 Incorporating cleaved sequences into backbone ensemble generation improves MFPred accuracy.

(a) Experimental specificity profiles (b) Results of running MFPred on backbone ensemble of five cleaved sequences FastRelaxed (c) Results of running MFPred on backbone ensemble generated by enumerating combinations of phi/psi angles. (d) Results of running MFPred on backbone ensemble of five uncleaved sequences FastRelaxed.

As a second test of the dependence of MFPred on the cleaved sequence information, we threaded five known uncleaved (*i.e.,* not bound by the protease in a productive conformation) sequences on the peptide backbone and then performed FastRelax on the resulting structures. The prediction accuracy of MFPred decreased on these structures (Figure 3.7), to the extent that the specificity profiles are almost uniform. Therefore, diversifying the peptide structure in suboptimal sequence space led to worse predictions than those obtained while diversifying it without any sequence information.

Next, to determine the impact of starting from bound complexes to generate MFPred predictions, we performed MFPred simulations on apo structures of two proteases: HCV NS3 protease and HIV protease-1 (Figure 3.8). As HIV protease-1 has two flaps that can assume either a closed or open form[106], we used both a 'closed apo' structure and an 'open apo' structure for our simulations. In each case the protease all-atom RMSD between bound and open states, as determined by PyMol[107], were 1.04 Å, 1.85 Å, and 2.00 Å. In all three cases, MFPred accuracy was higher when starting from the bound complex compared to the apo state. While the number of significant *p*-values remains similar, the overall cosine similarities, AUC, and SSAL decreased for the apo structure-based simulations. Additionally, the information content decreased significantly for the apo structures of HIV (0.72-0.74 bits) as opposed to the bound complex (1.18 bits). Overall, the prediction accuracies between apo and bound states were more similar for the HCV protease where small backbone changes in the protease are incurred upon binding, compared to HIV protease where larger differences in prediction accuracy were apparent. These results suggest that especially in cases where there is significant

backbone conformational change in the receptor upon peptide binding, such as the HIV protease, the incorporation of receptor flexibility may be needed for maintaining MFPred accuracy.

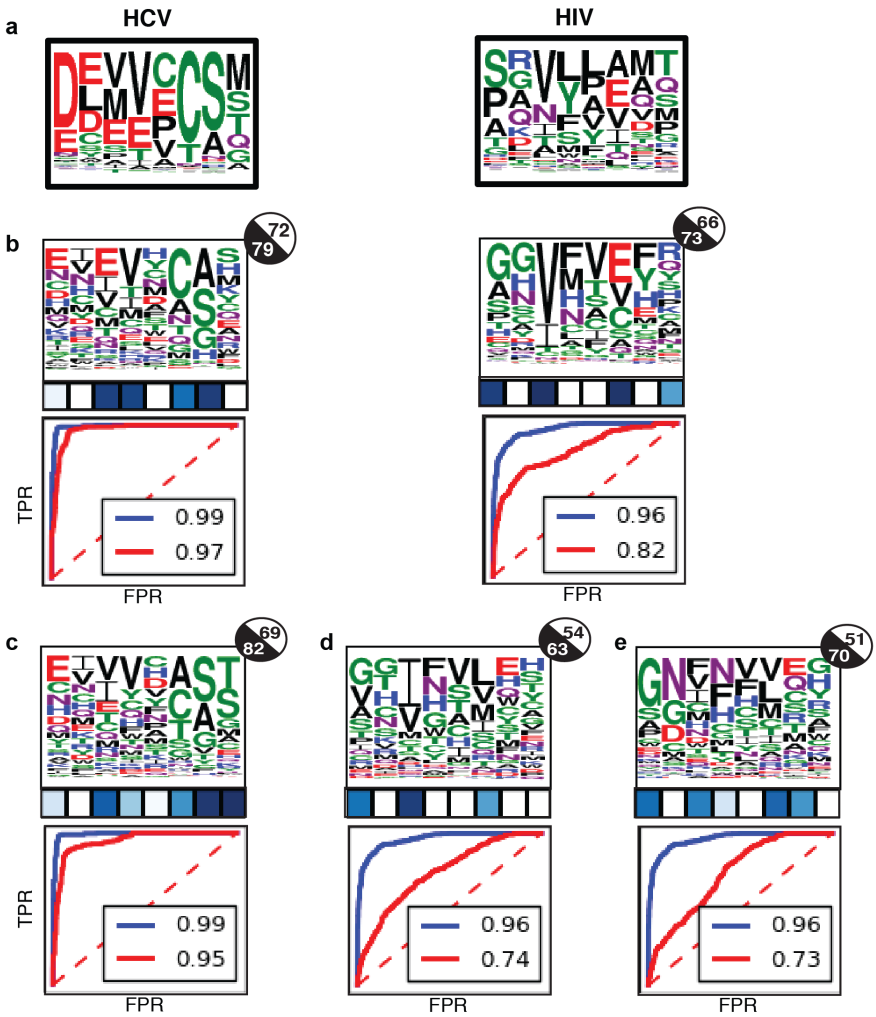


Figure 3.8. Using structures of receptor peptide complexes vs. apo structures improves the accuracy of MFPred.
(a) Experimental specificity profiles. (b) MFPred prediction on receptor – peptide complexes. (c) MFPred prediction on HCV NS3 Protease apo structure (PDB 3KF2). (d) MFPred prediction on HIV protease 1 closed form apo structure (PDB: 2HB4). (e) MFPred prediction on HIV protease 1 open form apo structure (PDB: 1PCO).

Finally, to investigate the dependence of performance accuracy on the number of known cleaved (recognized) sequences, we executed MFPred simulations on backbone ensembles generated from differing numbers of starting peptide sequences threaded on to the crystallographic backbone conformation. We varied the number of sequences used to generate the backbone ensemble from one sequence to five sequences to ten sequences to all known sequences in the benchmark set. We found that MFPred is highly dependent on *N*, the number of cleaved sequences used, when *N* is small (Figure 3.6e-h). However, as *N* increases, this effect is decreased. For TEV-PR and HCV protease, which have relatively few sequences (68 and 198 respectively), the prediction accuracy plateaus after ten sequences, although in some cases it may fluctuate slightly from five to ten to all sequences. However, for granzyme B and HIV proteases (356 and 374 cleaved sequences respectively), the accuracy of MFPred has a minor increase from ten to all sequences. Thus, there is a near-maximum of accuracy for each system; once that point of diminishing returns has been reached, incorporating more cleaved sequences does not lead to significant increases in the accuracy.

Besides determining that the level of backbone sampling was optimal for prediction, we also optimized sidechain sampling (Table 3.2). Using an older version of the rotamer library (2002)[108] decreased scores for all systems. Increasing the fineness of rotamer chi-angle sampling or removing the starting sidechain conformation from the rotamer sampling had little impact on the results. Packing protease sidechains around the peptide (between distances of 4-8 Angstroms) decreased the accuracy of the results. This may be explained by the finding that hot spot residues at protein-protein interfaces often adopt

strained rotamer configurations[109]; packing protease interface sidechains while designing peptide residues within MFPred may force protease sidechains to adopt conformations that are unfavorable for productive substrate binding.

Table 3.2. Effect of various Rosetta settings on MFPred predictions on five sequence backbones.

Protease	Method	Cosine	Frob	AAD	JSD	AUC	SSAL	Bits
TEV	**Current**	0.89	0.85	0.04	0.21	0.86	0.00	-0.34
	Dun02	0.86	0.97	0.04	0.24	0.86	0.00	-0.14
	Ex1aro,ex2aro	0.89	0.85	0.04	0.21	0.86	0.00	-0.34
	Ex3,ex4	0.88	0.87	0.04	0.22	0.86	0.00	-0.38
	No input sc	0.88	0.88	0.04	0.22	0.86	0.00	-0.46
	Pack prot 4	0.81	1.07	0.04	0.25	0.90	0.00	-0.56
	Pack prot 6	0.81	1.07	0.04	0.25	0.91	0.00	-0.59
	Pack prot 8	0.81	1.07	0.04	0.25	0.91	0.00	-0.60
HCV	**Current**	0.72	1.13	0.05	0.31	0.79	0.02	-1.28
	Dun02	0.64	1.24	0.06	0.35	0.78	0.02	-1.19
	Ex1aro,ex2aro	0.72	1.13	0.05	0.31	0.79	0.02	-1.28
	Ex3,ex4	0.71	1.14	0.05	0.31	0.77	0.03	-1.27
	No input sc	0.71	1.15	0.05	0.31	0.78	0.02	-1.29
	Pack prot 4	0.67	1.20	0.06	0.33	0.73	0.04	-1.21
	Pack prot 6	0.67	1.20	0.06	0.33	0.74	0.04	-1.21
	Pack prot 8	0.67	1.20	0.06	0.33	0.74	0.04	-1.20
GrB	**Current**	0.84	0.73	0.04	0.20	0.76	0.21	0.07
	Dun02	0.82	0.78	0.04	0.23	0.79	0.22	0.21
	Ex1aro,ex2aro	0.84	0.73	0.04	0.20	0.76	0.21	0.07
	Ex3,ex4	0.84	0.73	0.04	0.20	0.76	0.21	0.08
	No input sc	0.84	0.73	0.04	0.20	0.75	0.22	0.06
	Pack prot 4	0.81	0.80	0.04	0.23	0.77	0.25	0.22
	Pack prot 6	0.80	0.82	0.04	0.23	0.75	0.26	0.18
	Pack prot 8	0.81	0.80	0.04	0.23	0.76	0.25	0.21
HIV	**Current**	0.65	0.96	0.05	0.27	0.73	0.14	-0.01
	Dun02	0.59	1.08	0.05	0.32	0.68	0.14	0.10
	Ex1aro,ex2aro	0.65	0.96	0.05	0.27	0.73	0.14	-0.01
	Ex3,ex4	0.65	0.97	0.05	0.27	0.71	0.14	-0.01
	No input sc	0.63	0.98	0.05	0.28	0.70	0.15	-0.06
	Pack prot 4	0.63	1.01	0.05	0.30	0.71	0.14	0.08
	Pack prot 6	0.61	1.04	0.05	0.32	0.71	0.15	0.11
	Pack prot 8	0.60	1.05	0.05	0.31	0.69	0.15	0.05
Most		1.00	0.00	0.00	0.00	1.00	0.00	0.00
Most Different		0.00	$\sqrt{(2n)}$[1]	0.06	1.00	0.00	1.00	4.32

[1]n refers to the number of positions in the profile

3.3.6. Comparison of MFPred with Other Structure-Based Approaches

We compared our results to the two previously developed methods for specificity prediction that have been implemented in the Rosetta software. MFPred performed with

comparable or greater accuracy than the sequence_tolerance[56] and pepspec[95] methods (Figure 3.9, Table 3.3). Additionally, MFPred was between 23-fold to 120-fold faster than the pepspec method and between 154-fold to 1154-fold faster than the sequence_tolerance method, depending on the number of peptide backbone conformations and rotamers (Table 3.3). Furthermore, MFPred is more accurate on single backbones and smaller backbone ensembles than the other two methods; when performed on a backbone ensemble generated from five substrate sequences, MFPred predicts 19 out of 31 positions with a significant *p*-value, whereas only 11 of the positions predicted by sequence_tolerance and 8 of the positions predicted by pepspec yield significant *p*-values (Figure 3.9). When executed on a single backbone conformation, MFPred predicts 12 positions with a significant *p*-value, while both sequence_tolerance and pepspec predict only 8 positions with a significant *p*-value. Both sequence_tolerance and pepspec are designed to be used with larger peptide ensembles – their success is dependent on a diverse backbone ensemble – and, as expected, their prediction accuracy increases as the number of backbones in the ensemble rises (Figure 3.10a-d), with sequence_tolerance predicting 15 significant positions and pepspec predicting 16 significant positions on the backbone ensemble generated from all cleaved sequences (Figure 3.11). When performed on this expanded backbone ensemble, MFPred prediction accuracy was also higher, with 25 significant predictions. Thus, compared to two state-of-the-art existing methods, MFPred-based predictions are of comparable or higher accuracy, and can be obtained with 10-1000-fold higher computational efficiency.

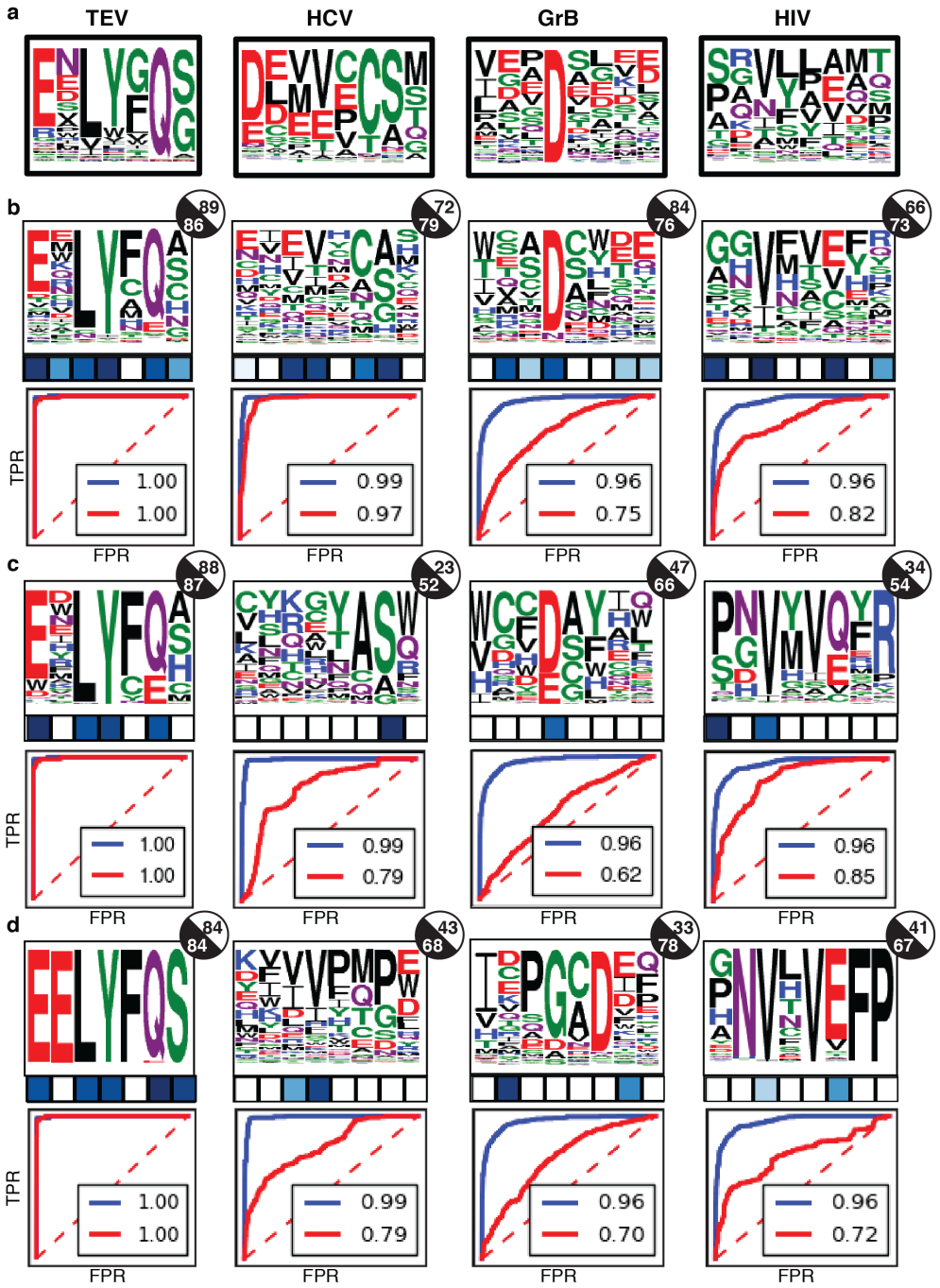


Figure 3.9. MFPred vs. other Rosetta prediction techniques on ensemble of five sequences.
(a) Experimental specificity profiles. (b) MFPred. (c) pepspec. (d) sequence_tolerance.

Table 3.3. Results of all methods on variously-sized backbone ensembles.

Protease	Method	#Seq	Time(m^{-1})	Cosine	Frob	AAD	JSD	AUC	SSAL	Bits
TEV	**MF**	1	0.18	0.86	1.06	0.04	0.22	0.87	0.00	0.43
		5	0.80	0.89	0.85	0.04	0.21	0.86	0.00	-0.34
		10	2.08	0.88	0.86	0.04	0.20	0.91	0.00	-0.55
		All (68)	11.97	0.89	0.84	0.03	0.20	0.91	0.00	-0.69
	ST	1	195.65	0.84	1.49	0.04	0.28	0.83	0.00	1.82
		5	923.91	0.84	1.49	0.04	0.28	0.84	0.00	1.79
		10	1827.32	0.84	1.49	0.04	0.28	0.85	0.00	1.82
		All (68)	12333.94	0.84	1.44	0.04	0.28	0.84	0.00	1.65
	PS	1	17.46	0.72	1.50	0.05	0.36	0.81	0.01	0.83
		5	96.01	0.85	1.06	0.04	0.24	0.92	0.00	0.44
		10	189.43	0.82	1.17	0.04	0.24	0.85	0.00	0.34
		All (68)	1290.41	0.86	1.04	0.03	0.21	0.86	0.00	0.27
HCV	**MF**	1	0.68	0.59	1.37	0.06	0.35	0.77	0.08	-0.51
		5	3.61	0.72	1.13	0.05	0.31	0.79	0.02	-1.28
		10	7.14	0.71	1.15	0.05	0.30	0.82	0.02	-1.28
		All (196)	132.15	0.71	1.14	0.05	0.29	0.84	0.02	-1.29
	ST	1	115.04	0.30	1.77	0.07	0.53	0.63	0.30	-0.59
		5	574.01	0.43	1.54	0.06	0.46	0.68	0.21	-0.93
		10	1101.15	0.44	1.49	0.07	0.44	0.70	0.17	-1.16
		All (196)	22239.05	0.43	1.51	0.07	0.44	0.67	0.17	-1.08
	PS	1	17.78	0.24	2.19	0.08	0.63	0.61	0.34	0.66
		5	91.68	0.37	1.69	0.07	0.55	0.55	0.20	-0.53
		10	171.30	0.61	1.30	0.06	0.39	0.73	0.05	-0.73
		All (196)	3462.64	0.63	1.26	0.06	0.36	0.71	0.05	-1.19
GrB	**MF**	1	0.34	0.82	0.85	0.04	0.23	0.71	0.20	0.60
		5	2.39	0.84	0.73	0.04	0.20	0.76	0.21	0.07
		10	5.24	0.89	0.60	0.03	0.17	0.80	0.17	0.06
		All (356)	145.63	0.91	0.53	0.03	0.13	0.87	0.15	-0.08
	ST	1	114.80	0.28	2.02	0.07	0.46	0.76	0.26	1.29
		5	544.28	0.33	1.71	0.06	0.35	0.78	0.26	0.68
		10	1109.45	0.35	1.62	0.05	0.31	0.82	0.17	0.55
		All (356)	39036.17	0.34	1.67	0.05	0.32	0.84	0.21	0.53
	PS	1	19.58	0.62	1.45	0.06	0.51	0.61	0.38	1.59
		5	101.24	0.63	1.15	0.06	0.39	0.70	0.34	0.68
		10	203.69	0.76	0.99	0.05	0.29	0.78	0.27	0.61
		All (356)	6814.15	0.88	0.64	0.03	0.17	0.86	0.18	0.13
HIV	**MF**	1	0.23	0.47	1.55	0.06	0.42	0.66	0.17	0.96
		5	1.29	0.65	0.96	0.05	0.27	0.73	0.14	-0.01
		10	3.15	0.70	0.88	0.04	0.23	0.78	0.08	-0.04
		All (374)	110.65	0.72	0.82	0.04	0.21	0.81	0.05	-0.21
	ST	1	92.37	0.40	2.48	0.08	0.64	0.62	0.19	2.78
		5	453.18	0.41	2.20	0.07	0.57	0.67	0.24	2.14
		10	907.90	0.45	2.05	0.07	0.51	0.73	0.16	1.93
		All (374)	34090.45	0.48	1.81	0.06	0.42	0.73	0.14	1.38
	PS	1	23.05	0.37	2.13	0.07	0.60	0.59	0.22	2.05
		5	109.77	0.55	1.54	0.06	0.40	0.69	0.11	1.21
		10	218.41	0.53	1.51	0.06	0.39	0.70	0.16	1.04
		All (374)	8134.56	0.57	1.23	0.05	0.28	0.76	0.10	0.33
Most Similar				1.00	0.00	0.00	0.00	1.00	0.00	0.00
Most Different				0.00	$\sqrt{(2n)}$[1]	0.06	1.00	0.00	1.00	4.32

[1]n refers to the number of positions in the profile

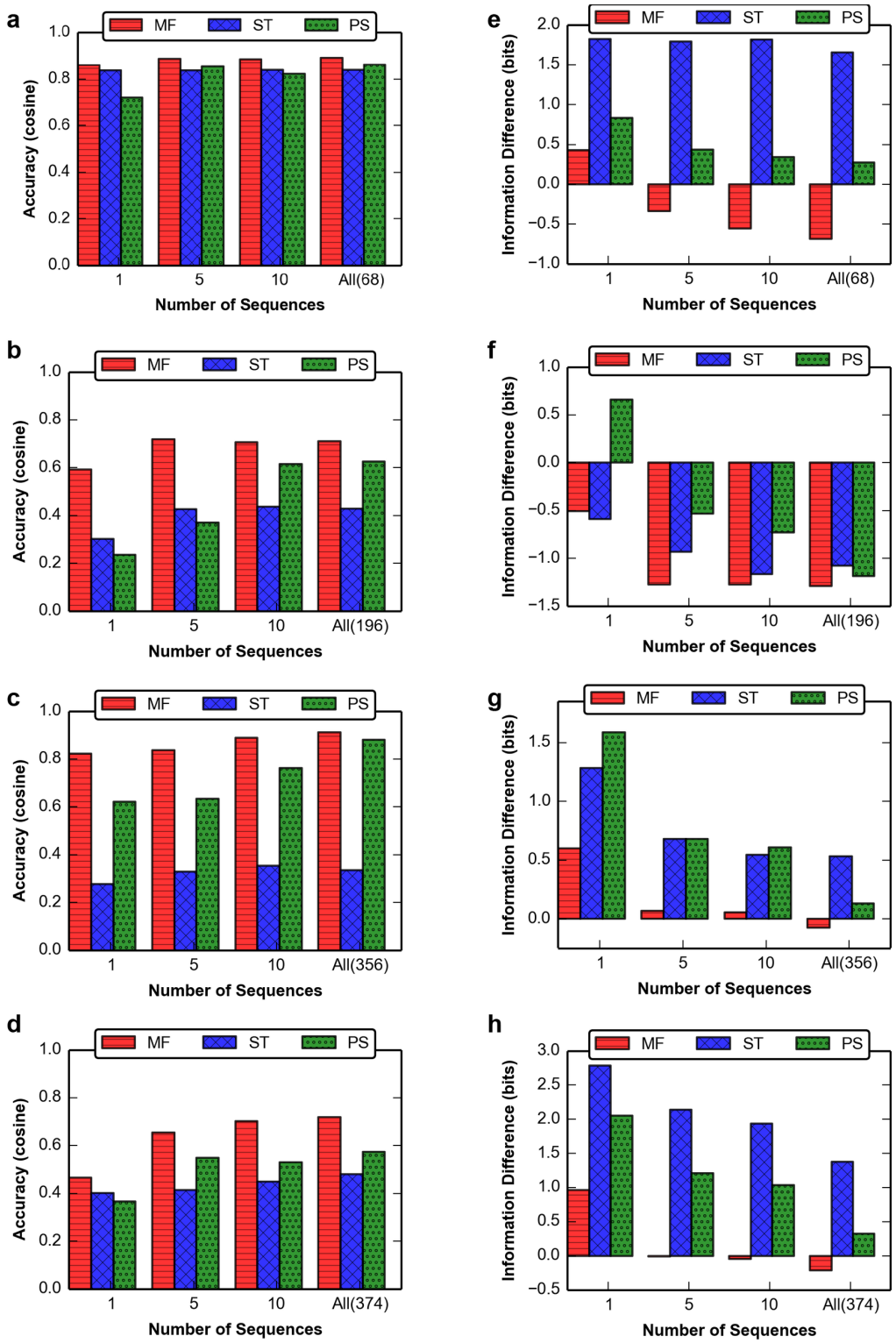


Figure 3.10. Number of sequences vs. accuracy and information for methods of profile prediction

(a)-(d) Number of sequences vs. accuracy for TEV, HCV, GrB, and HIV, respectively. Number of sequences is varied over 1-5-10-All experimentally derived sequences, which is different for each protease. **(e)-(h)** Number of sequences vs. information content

difference for TEV, HCV, GrB, and HIV, respectively. Information difference is equal to the predicted bits minus the experimental bits. An information difference that is close to zero approximates the experimental information content well; a positive information difference indicates a more peaked predicted than experimental profile while a negative information difference denotes a flatter predicted than experimental profile.

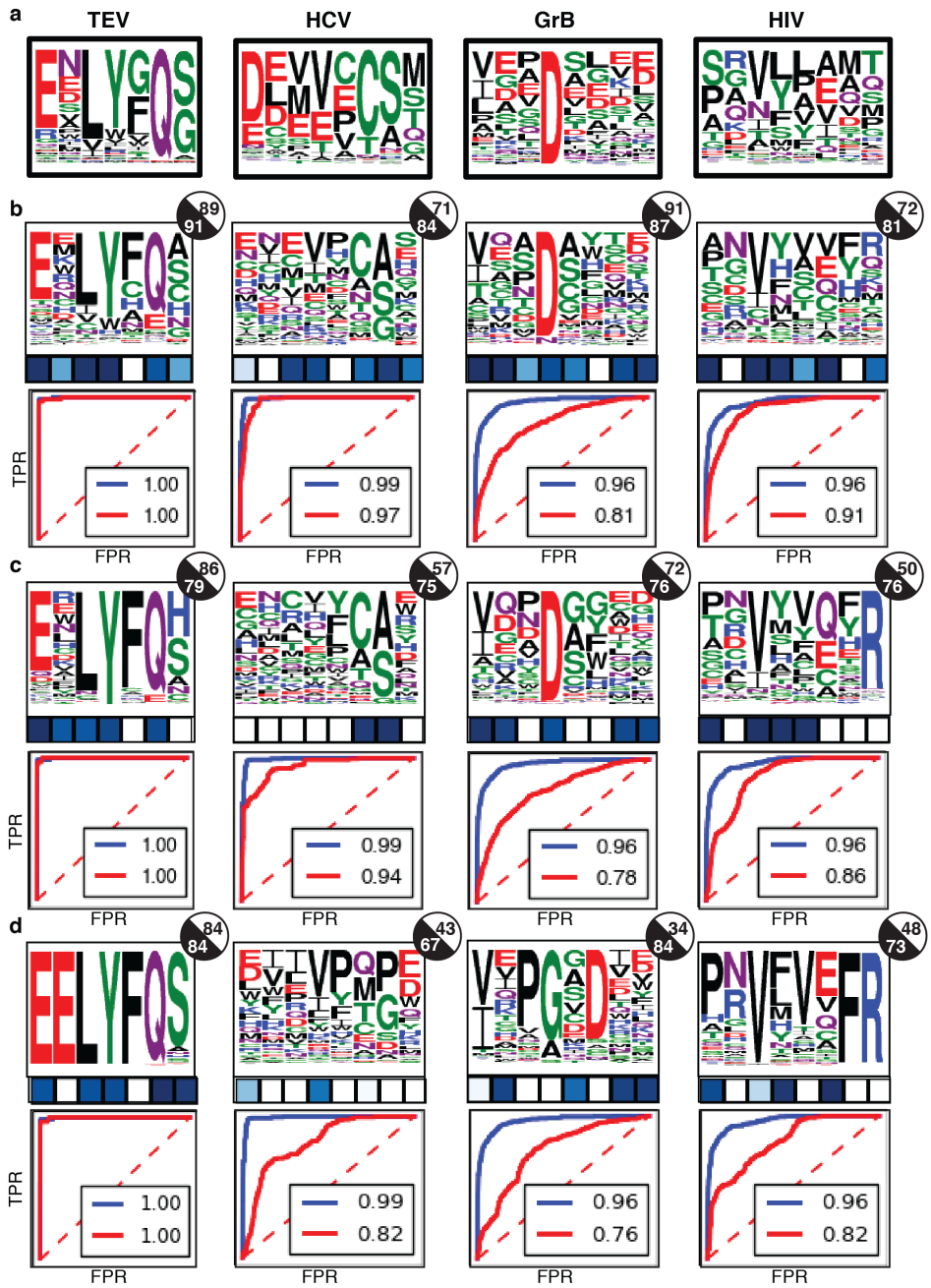


Figure 3.11. MFPred vs. other Rosetta prediction techniques on ensemble of all sequences.
(a) Experimental specificity profiles. (b) MFPred. (c) pepspec. (d) sequence_tolerance.

Besides informing us about the accuracy and speed of MFPred relative to existing methods, the comparison of MFPred to pepspec and sequence_tolerance allows us to categorize inaccuracies in MFPred predictions into those obtained from incorrect sequence sampling and those due to the Rosetta energy function or incomplete backbone conformational diversity. For example, MFPred on all cleaved backbones does not recover the experimentally determined high frequency for G at P2 of TEV-PR. Since both pepspec and sequence_tolerance also do not recover G at P2 with the same peptide backbone conformational ensemble, we attribute this inaccuracy to imperfections in the underlying Rosetta energy function and/or an incomplete peptide backbone ensemble used for prediction.

Generally, MFPred predicts lower information content (*i.e.* flatter shape) for the profiles than both sequence_tolerance and pepspec (Table 3.3, Figure 3.10e-h). In the cases of granzyme B protease and HIVPR1, the predicted lower information content is reflective of the experimentally determined profiles; however, in the case of TEV-PR MFPred underestimates the information content relative to pepspec and sequence_tolerance. All protocols underestimate the information content of the profile of HCV protease. This underestimation may be due to an incomplete experimental dataset or sampling/scoring inaccuracies as discussed above. Overall, the difference between the predicted information content and the experimental information content was smaller for MFPred than for sequence_tolerance and pepspec, especially when performed with smaller backbone ensembles.

3.3.7. Generalizing MFPred to other Protein-Recognition Domains

To investigate the generality of our method for specificity prediction, we utilized the MFPred method to predict the specificity profiles for a variety of peptide-recognition domains: kinase, SH2, SH3, PDZ, and MHC domains. We achieved 17 significant *p*-values out of 31 positions and high cosine similarities (0.77-0.85) for three out of five PRD classes: PKA (kinase), Src (SH2), and c-Crk (SH3) domains (Figure 3.12). However, these three systems had lower AUCs (0.60-0.65). This may be due to the inadequacy of AUC as a metric for scoring positions that have low information content in the experimentally-derived profile; if few of the experimental amino acid frequencies are greater than 10%, the AUC reveals little about the prediction accuracy.

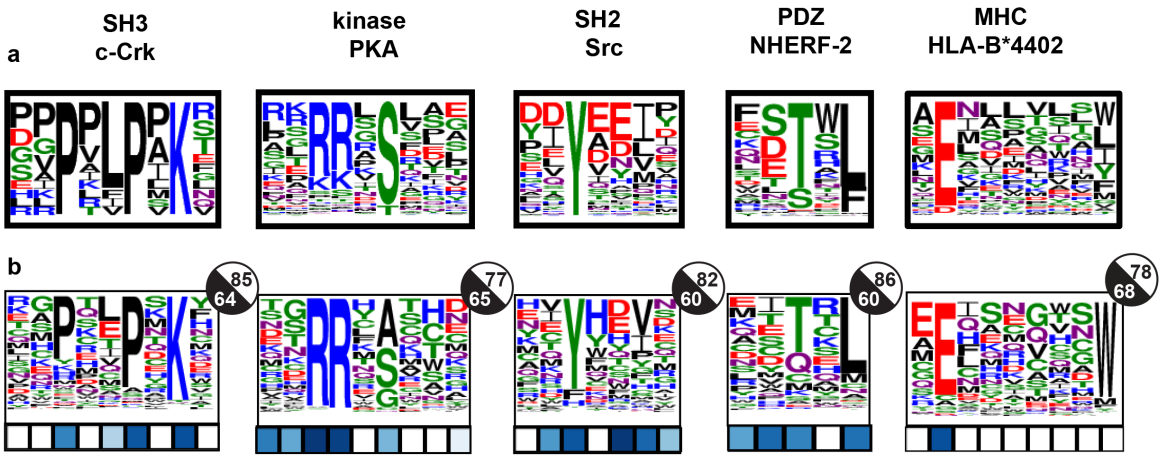


Figure 3.12. Generalize MFPred to PRD benchmark.
(a) Experimental specificity profiles. **(b)** MFPred prediction. The *p*-value of the JSD for a given position is represented by the color of the square under that position; white denotes a *p*-value > 0.5 and dark blue denotes a *p*-value of 0. A given circle to the right of a profile represents the cosine similarity (white) and AUC (black) of that profile. For the PDZ domain, prediction was performed at a kT of 0.6, which was found to be optimal for PDZ domains.

We predicted the specificity profiles of seven different PDZ domains: NHERF-2 PDZ2, PSD-95, AF-6 PDZ, Erbin PDZ, MPDZ-13, ZO-1 PDZ1, and DLG1-2 PDZ (Figure 3.12, Figure 3.13). The specificity of NHERF-2 PDZ-2 was already predicted computationally by Zheng et al.[110], who were able to achieve good prediction via the use of CLASSY and FlexPepDock. King and Bradley previously predicted the specificity profile for PSD-95 computationally using pepspec[95], while the five other PDZ domain specificities were previously predicted by Smith and Kortemme via sequence_tolerance[56]. Six out of seven PDZ domains were predicted with medium to high accuracies, with cosine similarities of 0.63-0.86, AUCs of 0.60 to 0.88, and 25 out of 38 significant *p*-values. However, the prediction accuracy of the final PDZ domain, AF-6 PDZ was much lower, with a cosine similarity of 0.43, AUC of 0.59, and no significant *p*-values. This low accuracy may be due to the flexibility of the AF-6 PDZ domain, which has been known to bind in multiple binding modes and can be characterized as belonging to multiple classes of PDZ domain specificity[111,112]. Like the HIV-PR1 case above, addition of receptor flexibility to MFPred may assist in AF-6 specificity profile recapitulation.

Finally, we tested the performance of MFPred on predicting MHC-I peptide recognition specificities. We selected four MHC-I domains with crystallographic structure availability and a large pool of known peptide binders[113]. The experimentally derived specificity profiles for the MHCs were highly conserved at one or two positions but relatively flat at others (Figure 3.12, Figure 3.14). The MFPred predictions reflected this pattern: while 30 out of 36 positions had *p*-values that were not significant, due to the high tolerance of a diversity of amino acid at those positions, the cosine similarity of the

predictions was high (0.63-0.78), reflecting good overall profile recapitulation (Figure 3.12, Figure 3.14). These results indicate that robust and accurate predictions of the specificity profiles of a variety of peptide-recognition domains can be obtained using the MFPred approach, pointing to its wide applicability, especially for cases where receptor backbone flexibility is minimal. Improved modeling of backbone conformational diversity, an area where methodological improvements are needed[114], is likely to improve prediction accuracy further.

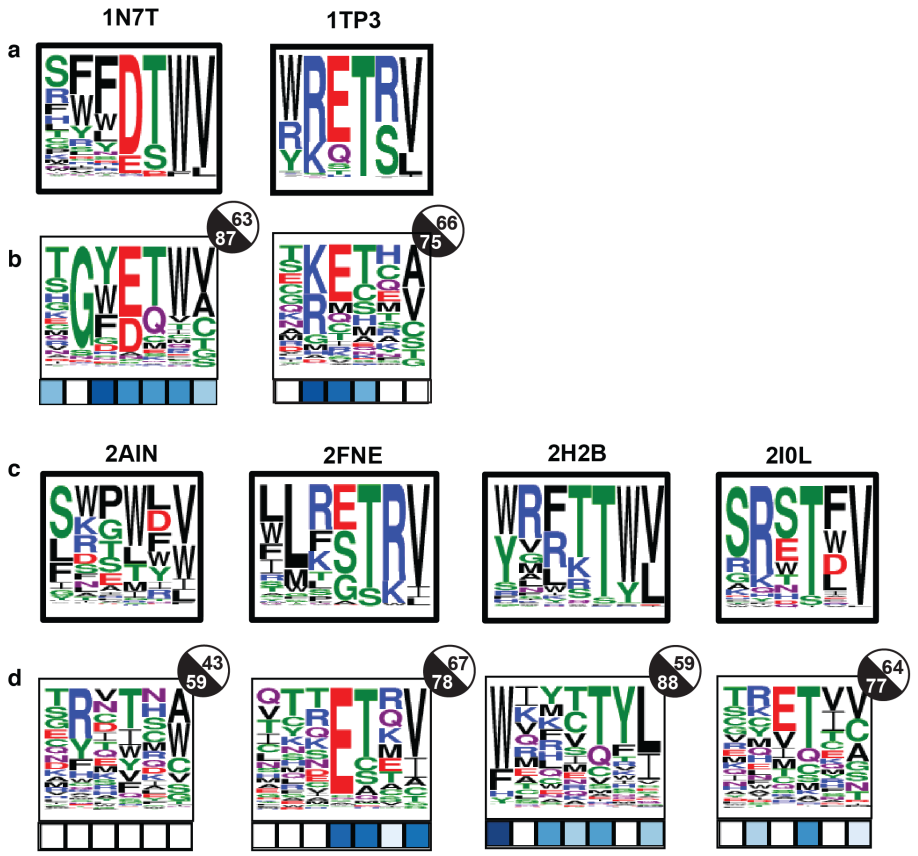


Figure 3.13. MFPred prediction for six PDZ domains.
(a, c) Experimental specificity profiles. (b, d) MFPred prediction. Prediction was performed at a kT of 0.6, which was found to be optimal for PDZ domains.

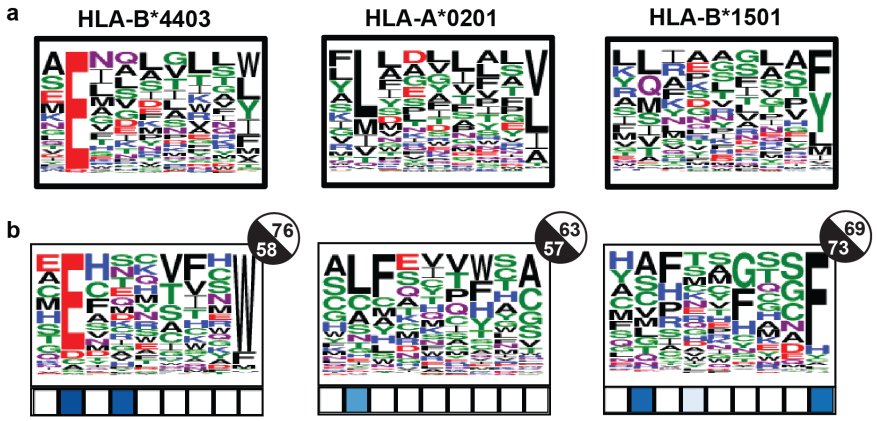


Figure 3.14. MFPred prediction for three MHC-I domains.
(a) Experimental specificity profiles. (b) MFPred prediction.

Prediction of changes in multispecificity upon receptor mutation

When used to design receptors for and against specificity profiles, MFPred should be able to accurately recapitulate changes in specificity profiles due to protease mutations, when simulations are performed on a constant set of backbones. As a proof of concept, we predicted the changes in the specificity profiles of two variants of granzyme B protease for which altered multispecificity has been experimentally determined (Figure 3.15). R192E granzyme B protease and R192E/N218A granzyme B protease have been shown to have decreased specificity for glutamic acid and increased specificity for lysine and arginine at P3[10,115]. To investigate whether MFPred can recapitulate mutant specificity profiles without changing the peptide backbone, we modeled the variants of granzyme B protease by performing the necessary mutations in Rosetta on the five FastRelaxed granzyme B protease backbones.

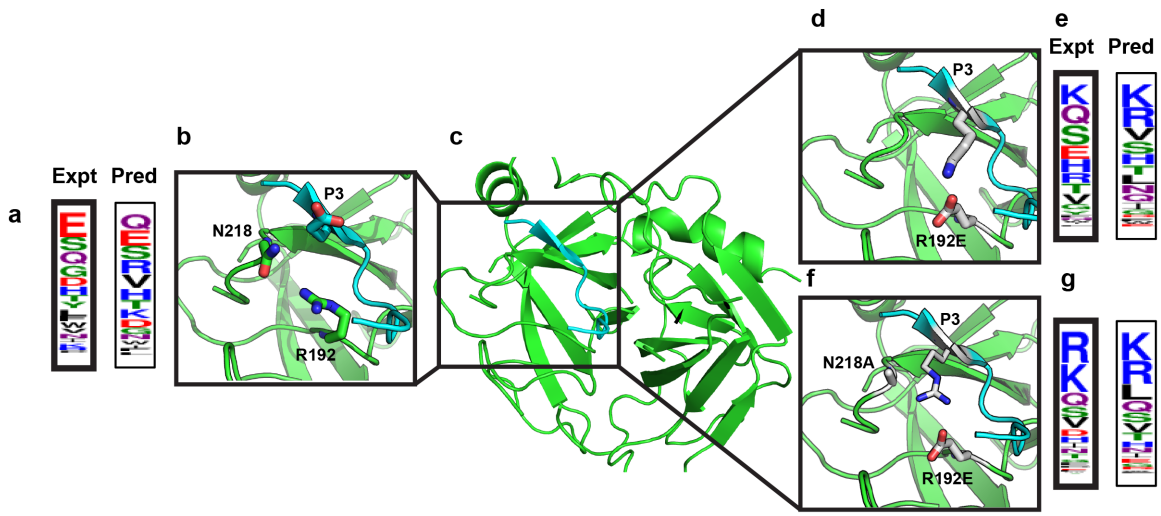


Figure 3.15. Proof-of-concept for design. Changes in specificity profile upon granzyme B protease mutation are recapitulated by MFPred.
(a) Experimental (bold) specificity (average of Harris et al.[10] and Ruggles et al.[115]) and predicted P3 specificity for WT granzyme B protease. **(b)-(c)**, WT granzyme B protease structure. **(d)** R192E granzyme B protease active site. **(e)** Experimental specificity (bold)[10] and predicted P3 specificity for R192E granzyme B protease. **(f)** R192E/N218A granzyme B protease active site. **(g)** Experimental specificity (bold)[115] and predicted P3 specificity for R192E/N218A granzyme B protease.

The MFPred-predicted specificity profile for the mutated structures accurately recapitulated the experimentally predicted specificity profile for the mutants. In the case of R192E, the change from a positively-charged arginine to a negatively-charged glutamic acid yields an increased frequency of positive amino acids such as lysine and arginine and a decreased frequency of the negative amino acid glutamic acid. MFPred predicts the shift toward lysine and arginine and away from glutamic acid correctly, although it upweights the frequency of arginine and downweights the frequency of glutamic acid relative to the experimental profile. In the case of R192E/N218A, the shift towards arginine and lysine is even more pronounced in the experimentally-derived profile. Sterically, the mutation of N to A may allow for the longer sidechains of R and K (relative to E) to fit at P3. MFPred correctly predicts this shift as well. The sensitivity

of MFPred to altered multispecificity at a given position due to a given receptor mutation should enable its use in designing for or against a given specificity profile.

3.4. Discussion

Protein-peptide interactions underlie much of biology, and the ability to computationally manipulate these interactions would enable intervention in many biological processes. The rational design of receptor proteins, including enzymes that act upon peptide substrates, for and against peptide recognition specificity profiles is an open challenge. Such design would benefit from a specificity profile prediction technique that is both (i) rapid enough to be used in each step of the design process, and (ii) able to predict changed specificity for receptor variants with a constant peptide backbone conformational ensemble. The MFPred method developed here represents a step forward in achieving in both of these goals. MFPred is able to predict profiles for both proteases and a diverse set of PRDs, and it can recapitulate changes in the profile of variant granzyme B. This result sets the stage for application of the MFPred algorithm to enable the design of proteins for and against specificity profiles, by combining the MFPred algorithm with multi-state design[116].

The MFPred method, implemented in the context of the Rosetta software, performs specificity profile prediction with equivalent or better accuracy when compared to two previously developed methods (pepspec, sequence_tolerance) in the Rosetta framework, but with a significant decrease in run time (~10- to 1000-fold). Practically, this means that given a receptor variant and a peptide backbone ensemble, a specificity profile can

be obtained, on a standard single processor, on a time-scale of seconds *vs*. hours required for other approaches. While pepspec and sequence_tolerance are less accurate on a smaller peptide backbone ensemble, MFPred is relatively robust to the size of the backbone ensemble. Additionally, MFPred can predict information content (determined from the amino acid frequency distribution at a given peptide position) better than other methods (Figure 3.10e-h). The ability to recapitulate information content should enable design for a narrow or wide range of amino acid types at a given peptide position, thereby allowing greater control over binding selectivity. The speed, prediction accuracy on a small backbone ensemble, and robust recapitulation of information content of MFPred are due to the mean-field approach of MFPred: rather than attempt to enumerate many sequences on varying backbones, MFPred predicts a specificity profile by treating amino acid energies as a Boltzmann probability distribution. However, optimal sampling of the peptide backbone conformational space by MFPred does require some prior knowledge in the form of several (~5) recognized substrates, which is not required for pepspec or sequence_tolerance.

While MFPred can rapidly and consistently generate recognition profiles with high accuracy compared to experimental data, it was not possible to achieve a perfect prediction using MFPred. Several reasons may underlie these limitations of MFPred. First, our experimental dataset may be incomplete: it comprises various *in vitro* and *in vivo* sources in the literature, each of which may have their biases. *In vitro* experimental profiles vary with the definition of a cleaved sequence; when few sequences are included in this definition, the profile will converge on a few optimal sequences. *In vivo*

experimental profiles are subject to biases due to biological factors[95]. Second, any specificity prediction challenge is composed of several, smaller problems – sampling the vast sequence space, sampling the significantly larger conformational space, and scoring the structures – each of contributes multiplicatively to the error-rate. In our study, the sequence sampling problem is solved by MFPred itself. As it is an approximation, MFPred may not sample the sequence space effectively; the free parameters, which are optimized for overall success, are sub-optimal for each system. This is especially true in the case of the temperature parameter, which we found to be the most system-dependent. Thus, application of MFPred to domain families that are not included in our benchmark set may require further system-specific optimization of model parameters to achieve comparable accuracy. In terms of structure sampling, our method of utilizing a small number of known recognized peptides to generate a backbone ensemble is an attempt to more efficiently sample the large backbone conformational space (which also determines sidechain sampling due to the use of a backbone-dependent rotamer library[117]); however, this space is so large, especially in the case of a flexible binding pocket such as the HIV protease-1, that sampling efficiency is still limited. The sampling of receptor backbone flexibility is also required in such cases, as evidenced by a decreased prediction accuracy when the apo-structure of the complex is used (Figure 3.8). Finally, we score the structures using an empirical energy function (from Rosetta); subtle errors in the energy function may also contribute to the observed inaccuracies. As both conformational and sequence sampling in the MFPred approach rely on, and are limited by, the underlying rotamer library and energy function as implemented in Rosetta, improvements in these features[117,118] should yield higher accuracy predictions.

3.5. Methods

3.5.1. Inputs

Table 3.4. Details of model generation for four proteases and fourteen PRDs

Protein	PDB ID	Resolution	Notes
HCV NS3 **Protease**	3M5L, 3M5N	1.9 Å	The P' residues of the bound peptide were built by overlaying PDB ID: 3M5N and PDB ID:3M5L (inhibitor-bound crystal structure) thus allowing us to build a complete substrate bound complex
HCV NS3 **Protease** (apo)	3KF2	2.5 Å	PDB ID: 3KF2, the apo structure of HCV NS3 protease, was superimposed with the complex built from 3M5L and 3M5N (above) and the peptide from that model was added to the apo structure to generate the starting model.
TEV **Protease**	1LVB, 1LVM	2.2 Å	Starting model generated from PDB by reverting C151A to WT
Granzyme B (**Protease**)	1FI8	2.2 Å	The interface of the ecotin chain in the crystal structure, spanning eight residue substrate chain was used as the starting point for further calculations
HIV **Protease** 1	1MT9	2.0 Å	Starting model generated by inverting D25N and V82N from crystal structure to native residue

			identities
HIV **Protease** 1 (apo)	2HB4	2.15 Å	PDB ID: 2HB4, the closed-form apo structure of HIV protease-1, was superimposed with the complex built from 1MT9 (above) and the peptide from that model was added to the apo structure to generate the starting model.
HIV **Protease** 1 (apo)	2PC0	1.4 Å	PDB ID: 2PC0, the open-form apo structure of HIV protease-1, was superimposed with the complex built from 1MT9 (above) and the peptide from that model was added to the apo structure to generate the starting model.
c-Crk **SH3**-N	1CKA	1.5 Å	
cAMP-dependent PKA (**kinase**)	1L3R	2.0 Å	
Src **SH2**	1SPS	2.7 Å	
PSD-95 **PDZ3**	1TP3	1.99 Å	
NHERF-2 **PDZ2**	2HE4	1.45 Å	

AF-6 **PDZ**	2AIN	(NMR)	First model in NMR ensemble was taken.
Erbin **PDZ**	1N7T	(NMR)	First model in NMR ensemble was taken.
MPDZ-13 (**PDZ**)	2FNE	1.83 Å	
ZO-1 **PDZ1**	2H2B	1.6 Å	
DLG1-2 (**PDZ**)	2I0L	2.31 Å	
HLA-A*0201 (**MHC**)	1QSF	2.8 Å	
HLA-B*1501 (**MHC**)	1XR9	1.79 Å	
HLA-B*4402 (**MHC**)	1M6O	1.6 Å	
HLA-B*4403 (**MHC**)	1N2R	1.7 Å	

Structure Preparation. Crystal structures of the four protease-peptide complexes, fourteen protein-recognition domains, and three protease apo structures were procured from the Protein Data Bank (PDB) (Table 3.4)[9,6,69,70,99,106,111,119–129]. Structures were filtered for a resolution equal to or lower than 2.8 Å and a bound peptide or peptidomimetic inhibitor. Active site mutations were reverted to the wild-type residues.

The selected crystal structures were optimized using Rosetta FastRelax to find a low energy structure, which was used as a starting point in further calculations. In the case of the protease enzymes, constraints were applied to catalytic residues during FastRelax to maintain active site geometry and keep the protease in a pre-transition-state near-attack conformation, and coordinate constraints were applied to the backbone to ensure that the enzyme did not unfold; we did not apply constraints in the general PRD benchmark, as constraints were found to decrease prediction accuracy in those cases. Peptide side chains and backbone were allowed to sample all degrees of freedom including rotation, translation, and rigid body orientation with respect to the protease. The models were scored with Rosetta's talaris2013 energy function.

The apo crystal structures were aligned with the relaxed models of the protease-peptide complexes using PyMol[107], and the peptides from the protease-peptide complexes were placed within the apo models. The crystal structures were further optimized using Rosetta FastRelax as described above.

Experimental Sequence Profiles and Cleaved/Uncleaved Sequences. The sequences of cleaved and uncleaved substrate peptides for each protease and bound peptides for each PRD were obtained as described in Table 3.5. For further details on the curation of the protease datasets, please see our recent study[103]. To generate a specificity profile for each protease, we first removed duplicates from the set of cleaved peptides and then calculated the frequency of each amino acid at each position. We followed the same procedure for the PRDs; however, we did not remove duplicates from those sets. The sequence sets are provided in S1 Dataset.

Table 3.5. Substrates for proteases and PRDs.

Protease	**# Cleaved**	**# Uncleaved**	**References**
TEV-PR	68	1520	• Kostallas et al.[43] • Boulware et al.[44]
HCV protease	196	1943	• Shiryaev et al.[41] • Rögnvaldsson et al.[42]
Granzyme B protease	353	1973	• Barkan et al.[36]
HIV-PR	374	1251	• Rögnvaldsson et al.[42]
PRD	**#Bound *in vitro***	**#Bound *in vivo***	**References**
c-Crk SH3-N	13	N/A	• Sparks et al.[85]
cAMP-dependent PKA	346	19	• PhosphoELM[130] • Schutkowski et al.[80]
Src SH2	13	117	• PepCyber[131] • Khati et al.[81]
PSD-95 PDZ3	93	2	• PDZBase[132] • Tonikian et al.[82]
NHERF-2 PDZ2	132	N/A	• Vouilleme et al.[83] • Stiffler et al.[84] • Tonikian et al.[82]
AF-6 PDZ	176	N/A	• Tonikian et al.[82]

Erbin PDZ	86	N/A	• Tonikian et al.[82]
MPDZ-13 (PDZ)	91	N/A	• Tonikian et al.[82]
ZO-1 PDZ1	71	N/A	• Tonikian et al.[82]
DLG1-2 (PDZ)	58	N/A	• Tonikian et al.[82]
HLA-A*0201 (MHC)	3273	N/A	• Vita et al.[113]
HLA-B*1501 (MHC)	1187	N/A	• Vita et al.[113]
HLA-B*4402 (MHC)	236	N/A	• Vita et al.[113]
HLA-B*4403 (MHC)	207	N/A	• Vita et al.[113]

3.5.2. Backbone Ensemble Generation

We generated a flexible backbone ensemble by constructing models of the proteins bound to several cleaved sequences, and then diversifying those models via FastRelax[46], FlexPepDock[104], or Backrub[105] backbone sampling protocols, as described in detail below. For each protein, *N* cleaved sequences were chosen from the dataset by sorting the sequences in alphabetical order and then choosing evenly spaced sequences from the sorted dataset. Two alternative methods of picking cleaved sequences - randomly, or at even intervals from a set sorted by hamming distance from an arbitrarily chosen cleaved sequence - did not impact the results.

Then those *N* cleaved sequences were threaded onto the original FastRelaxed protein-peptide complex to create *N* structure-sequence models. Each model was subjected to 10 trajectories of FastRelax simulations, 10 trajectories of FlexPepDock refine simulations,

or 10 trajectories of Backrub simulations, and the resulting 10 models were considered to be the backbone conformational ensemble. As we found that the FastRelax protocol was more accurate than FlexPepDock and Backrub, we used FastRelax alone in the final version of the protocol. The model was constrained to active catalytic geometry for the proteases; we did not apply constraints to the PRD systems. Finally, the *x* lowest-scoring models for each sequence (with *x* dependent on the protocol in question, and generally set as 1) were chosen as the final backbone ensemble.

3.5.3. Mean-Field Algorithm

Various self-consistent mean-field theory-based methods have been developed for use in protein sidechain packing and design[133–140]. In the canonical self-consistent mean field theory-based method for protein sidechain packing as proposed by Koehl and Delarue[133], the energy landscape is investigated by using an effective energy potential to approximate the effects of all possible rotamers at all positions to be modeled. Thus, the mean-field energy of rotamer *r* occurring at position *i* is determined by Eq. 3.1:

$$E(i,r) = e(i_r) + \sum_{j=1, j\neq i}^{N} \sum_{s=1}^{K_j} e(i_r, j_s) P(j,s) \tag{3.1}$$

$e(i_r)$ represents the one-body energy of the rotamer, or the energy between a residue and the fixed components of the protein. $e(i_r, j_s)$ represents the two-body energy between a rotamer *r* at position *i* and a rotamer *s* at position *j*. Energies are truncated at a threshold that we optimized as a free parameter. *P(j, s)* represents the probability of rotamer *s*

occurring at position j and is initially given as $1/K_j$, where K_j is the total number of available rotamers at position j (obtained from a rotamer library).

A probability matrix (**P**) of size $N \times K_{max}$, where N is the number of positions to be analyzed and K_{max} is the maximum number of rotamers at any position, is used to model the probabilities of each rotamer occurring. Once the effective energy of each rotamer is determined using Equation 3.1, the probability of each rotamer is:

$$P(j,s) = \frac{e^{-\beta E(j,s)}}{\sum_{x=1}^{K_j} e^{-\beta E(j,x)}} \tag{3.2}$$

β (= $1/kT$) is also optimized as a free parameter. The algorithm iterates between the two equations until convergence is reached. We use a pre-calculated interaction graph in Rosetta[141] to store the one-body and two-body energies, which do not change between iterations, so the iteration is rapid. Convergence is improved with the use of a memory in the updating of **P**, so that the probability matrix after iteration x is given by $P_x = \lambda P_{x-1} + (1-\lambda)P_x$, where λ is a free parameter between 0 and 1. Once convergence is reached, the probability matrix **P** can be used to obtain the probability for every rotamer.

We extended the algorithm for use with a flexible backbone and with any given amino acid alphabet. Given an ensemble of backbone conformations, the probability matrix **P** is calculated for each backbone using the canonical self-consistent mean field method, while allowing each position to take on any amino acid, so that the vector for that

position contains all the rotamers for all amino acids at that position. $P_{aa}(bb, i)$, the probability of amino acid *aa* occurring at position *i* in backbone *bb*, is determined for all amino acids at all positions in all backbones:

$$P_{aa}(bb,i) = \frac{\sum_{r=1}^{K_{aa}} P_{bb}(i,r) \Big/ K_{aa}^{\gamma}}{\sum_{x=1}^{20} \sum_{r=1}^{K_x} P_{bb}(i,r) \Big/ K_x^{\gamma}}$$

(3.3)

where K_{aa} is the number of rotamers available to amino acid aa at position *i*, and γ is a free parameter optimized to 0.8 in our implementation. Dividing the sum of probabilities over all amino acids by K_{aa}^{γ} thus corrects for cases where numerous rotamers of an amino acid artificially inflate the probability of a specific amino acid occurring (Figure 3.16). The probability matrices for all backbones are then averaged together using a Boltzmann-weighting scheme in a two-step process. First, $E_{bb}(i,aa)$, the weighted sum of the energies for rotamers of amino acid *aa* at position *i* in backbone *bb*, divided by K_{aa}^{γ}, is calculated (Equation 3.4). Then $E_{bb}(i,aa)$ is used to find $W(i)$, the probability of backbone bb occurring at position i (Equation 3.5). *M* is the number of (peptide) backbones in the ensemble.

$$E_{bb}(i,aa) = \frac{\sum_{r=1}^{K_{aa}} E_{bb}(i,r) P_{bb}(i,r)}{K_{aa}^{\gamma}}$$

(3.4)

$$W(i) = \frac{e^{-\beta \sum_{aa=1}^{20} E_{bb}(i,aa)}}{\sum_{s=1}^{M} e^{-\beta \sum_{aa=1}^{20} E_s(i,aa)}}$$

(3.5)

Finally, a weighted average P is determined and taken to be the predicted specificity profile for that protease:

$$P(i, aa) = \sum_{bb=1}^{M} P_{aa}(bb, i) W(i) \tag{3.6}$$

Thus, MFPred can be used for prediction of multispecificity for both one backbone and multiple backbone conformations.

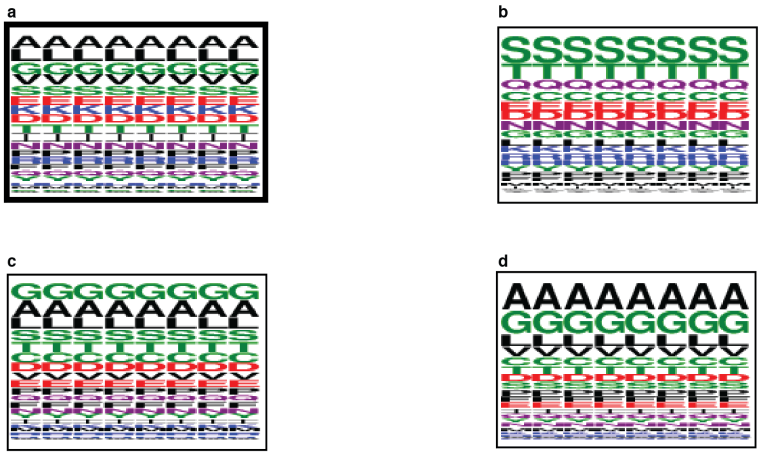


Figure 3.16. The need for γ in the mean-field algorithm when averaging rotamers of an amino acid to find the probability of that amino acid.
(a) Background amino acid composition as defined in Rosetta database (P_AA). This is the gold-standard which we attempted to match in our background profile generation (see Methods). (b) MFPred background prediction with γ=0, i.e. the rotamer probabilities are simply summed to find the amino acid probability. Serine and threonine are overrepresented as the Rosetta Dunbrack library contains many more rotamers for S and T, and glycine and alanine are underrepresented due to having only one rotamer each. (c) MFPred background prediction with γ=0.8 (current settings). This is closest to the P_AA distribution (Frobenius distance of 0.24). (d) MFPred background prediction with γ=1.0, *i.e* the amino acid probability is simply the average of the rotamer probabilities. While this is better than γ=0, alanine and glycine are now overrepresented and serine and threonine are underrepresented. Frobenius distance is 0.39.

3.5.4. Parameter Optimization of MFPred

To optimize four free parameters for MFPred (lambda, γ, threshold, and kT), we enumerated all combinations of lambda (0.25, 0.5, 0.75), γ (0, 0.2, 0.4, 0.6, 0.8, 1.0), threshold (5, 10, 50, 100, 250, 500), and kT (0.2, 0.4, 0.6, 0.8, 1.0). We selected 68 structures from the peptiDB (a peptide-protein complex database)[142] that met our criteria of having at least eight peptide residues. The structures were input into MFPred as a backbone ensemble and all combinations of the above parameters were tested. The resulting background specificity profiles were compared to the background residue distribution in the Rosetta database (Figure 3.16, Figure 3.17) and the combination of parameters with the lowest cosine distance from the known background distribution was chosen as our final set of parameters. While varying lambda had little impact on the results, all other parameters had a significant, system-dependent impact on the results.

3.5.5. Enrichment over Background

Since the MFPred predictions did include some noise due to the background distribution, we divided its predictions by the background profile to find the final prediction. The background profile was determined by averaging the frequencies of each position in the peptiDB profile. We divided each amino acid frequency in the initial predicted profile by the frequency of that amino acid in the background profile to find the final profile (Figure 3.17).

3.5.6. Software Availability

MFPred is available as a RosettaScripts Mover within the master branch of Rosetta. Sample cases for how to use MFPred can be found in Appendix 2 and in online Rosetta documentation.

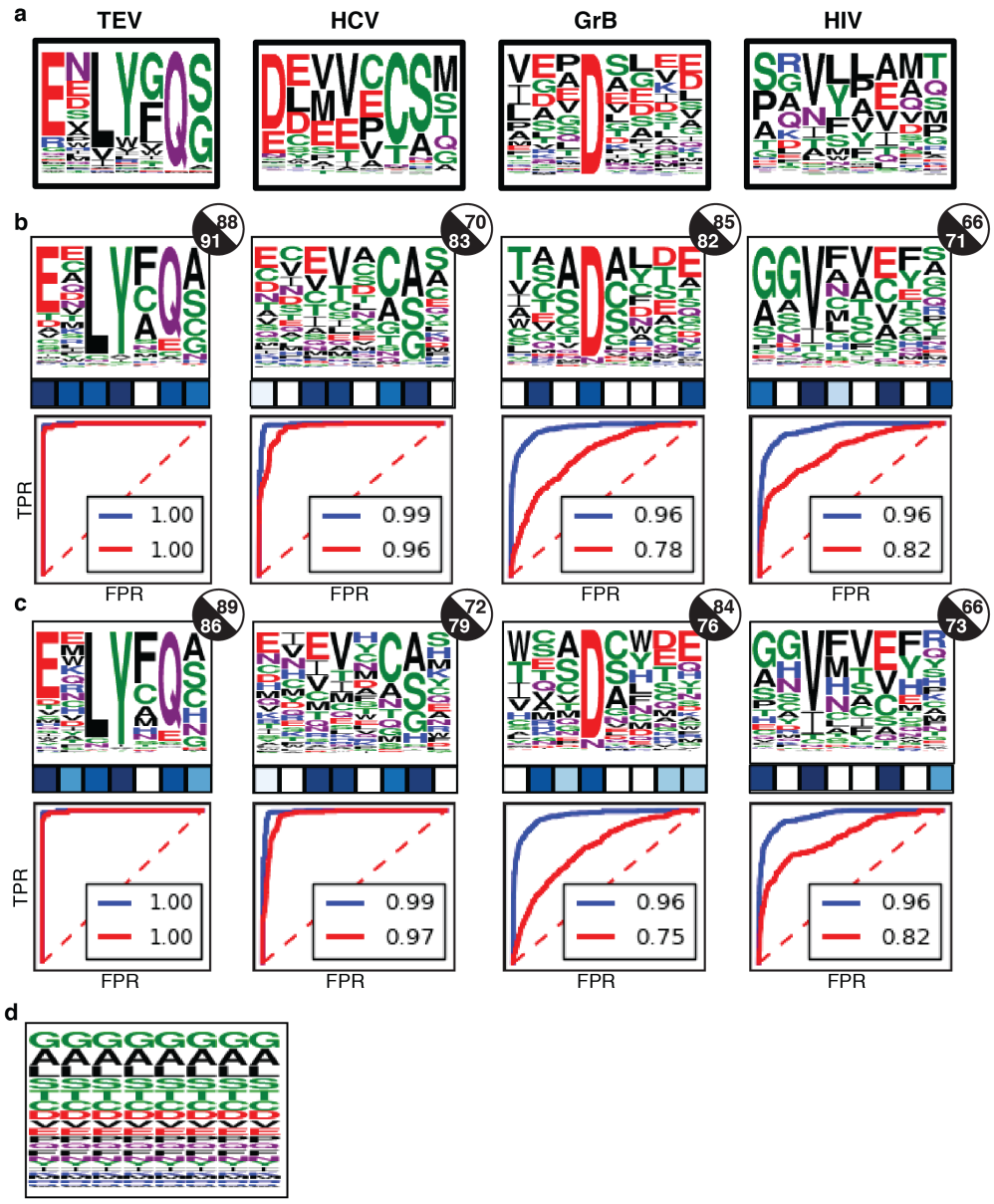


Figure 3.17 Enriching specificity profiles over background specificity profile improves accuracy.
(a) Experimental specificity profiles. (b) Initial MFPred-predicted specificity profiles. (c) Specificity profiles divided by background specificity profile. (d) Background specificity profile.

Chapter 4. Biophysical determinants of mutational robustness in a viral molecular fitness landscape

4.1. Abstract

Biophysical interactions between proteins and peptides are key determinants of genotype-fitness landscapes, but an understanding of how molecular structure and residue-level energetics at protein-peptide interfaces shape functional landscapes remains elusive. Combining information from yeast-based library screening, next-generation sequencing and structure-based modeling, we report comprehensive sequence-energetics-function mapping of the specificity landscape of the Hepatitis C Virus (HCV) NS3/4A protease, whose function – site-specific cleavages of the viral polyprotein – is a key determinant of viral fitness. We elucidate the cleavability of 3.2 million substrate variants by the HCV protease and find extensive clustering of cleavable and uncleavable motifs in sequence space indicating mutational robustness, and thereby providing a plausible molecular mechanism to buffer the effects of low replicative fidelity of this RNA virus. Specificity landscapes of known drug-resistant variants are similarly clustered. Our results highlight the key and constraining role of molecular-level energetics in shaping plateau-like fitness landscapes from quasi-species theory.

4.2. Introduction

RNA viruses, e.g., influenza, Hepatitis C virus (HCV) and Human Immunodeficiency virus (HIV), are under a heavy mutational load due to the extremely high error-rates of their RNA polymerases[143–145]. As a result of this low replication fidelity, these viruses exist as a population of variants called quasispecies[146,147], even within a single host

individual[148]. While this genetic diversity and a large population size is believed to increase viral adaptive potential against antiviral therapies[149–151], low replication fidelity may also lead to too many mutations, causing an "error catastrophe" and extinction[152,153]. The underlying biomolecular structures and interactions in the virus must, therefore, be robust to genetic variability such that they provide a buffer against the deleterious impacts of a high mutational load[154,155]. Tawfik and co-workers have hypothesized that viral proteins possess "gradient robustness" in which individual mutations have small and largely additive effects on stability leading to a slower loss of function compared to "threshold robustness" exhibited by proteins in general[156]. It has been argued that mutational robustness may itself promote adaptiveness if the number of phenotypes accessible to a variant through mutation is smaller than the total number of phenotypes possible[157,158]. How is mutational robustness encoded at the molecular level in RNA viruses such as HCV? How is structural integrity and interaction fidelity maintained in the face of a large mutational load, and what, if any, are the limits imposed by the underlying molecular interactions on mutational robustness and adaptive potential? The degeneracy of the genetic code, the thermodynamic and kinetic stabilities of RNA and proteins, and the presence of molecular chaperones, may all contribute to the robustness of the structures of individual viral biomolecules[145]. However, how viral protein-based interactions, especially those that are critical for viral propagation, encode "fuzziness"[156] leading to mutational robustness at the molecular level is not well understood.

At the molecular level, the balance between mutational robustness and functional plasticity is encapsulated in the notion of molecular fitness landscapes[159], which are high-

dimensional maps that relate the sequence of individual biomolecular variants to their functional and/or evolutionary fitness[160,161]. Analysis of mutational trajectories on these landscapes provides insight into the constraints placed on evolution by the physiochemical properties of biomolecules, allowing, in principle, reconstruction as well as forward prediction of molecular evolution[162–166]. The molecular fitness landscape has long been theoretically postulated[159] and recent empirically determined sequence-function mappings of proteins[167–175] have enabled the partial construction of fitness landscapes. These reconstructed landscapes permit testing of possible evolutionary scenarios and provide insight into properties such as mutational robustness and non-additivity (epistasis) of mutational effects[176,162,177–182]. Empirical sequence-function relationships also enable biomolecular engineering for new or improved functions[172,183–186].

Typically, sequence-function mapping of proteins and protein-protein interactions described above involves partial enumeration of the possible sequence diversity (for example, all single mutations and a subset of double mutations at a large number of protein residue positions) and high-throughput functional evaluation coupled with deep sequencing[187–189]. Statistical and/or biophysical models can be used to make inferences about the regions of sequence space not sampled[183,188]. However, comprehensive construction of the fitness landscape requires enumeration and evaluation of the complete sequence diversity (all higher-order mutations at all residue positions). Laub and co-workers have pioneered studies in which the entire combinatorial diversity is experimentally sampled, albeit at a smaller number of positions[173,190]. The astronomical size of sequence space, however, makes the comprehensive experimental evaluation of

sequence-function landscapes with any one experimental approach difficult. Computational biophysical methods may, in principle, assist in creation and analysis of functional and fitness landscapes[191]. Indeed, evolutionary landscapes of simple protein models, such as lattice models, have been extensively investigated using biophysical evolutionary theory and computational simulations[192–202], and deep connections with population genetics theories have been discovered[198,203,204]. While pioneering and crucial insights have been obtained in these studies, chemically realistic atomic-resolution structure-based elucidation of functional landscapes has not been performed so far, due both to high computational cost as well as inaccuracies in simulation force fields which preclude accurate biophysical evaluation of mutational effects on protein-protein interactions.

Here, we use a combination of experimental (biochemical) and computational techniques to elucidate the specificity landscape of the interaction between HCV NS3/4A protease enzymes and its substrates. This enzyme-substrate interaction is key for viral maturation as it cleaves exclusively at four specific sites in the viral polyprotein (Figure 4.1A) to release individual non-structural proteins[23], and also mediates inactivation of key human immunity proteins[205]. The cleavage specificity of the protease is thus a key determinant of viral fitness, and its proper functioning includes negative specificity – the lack of cleavage of non-canonical sites on the viral protein and of most host cell proteins (Figure 4.1A). The molecular interactions underlying both positive and negative specificities must be robust to mutations as the HCV virus RNA polymerase has a high error-rate[206], but how and whether this robustness is encoded in the protease-substrate interactions is

not known. Using yeast surface display, next-generation sequencing and a machine-learning approach which combines features from experimental data and atomistic computational simulations (utilizing the Rosetta and Amber force fields) that we recently developed[103,207], we construct the specificity landscape (with cleavability assignments made for 3.2 million substrate pentapeptide sequences) of the HCV NS3/4A protease and three of its known drug-resistant variants[64]. We demonstrate that energetic features of protease-substrate interactions inherently encode mutational robustness, and that the connectivity patterns in the specificity landscape may act as a "biophysical capacitor" for maintaining protease function in the face of high mutational load.

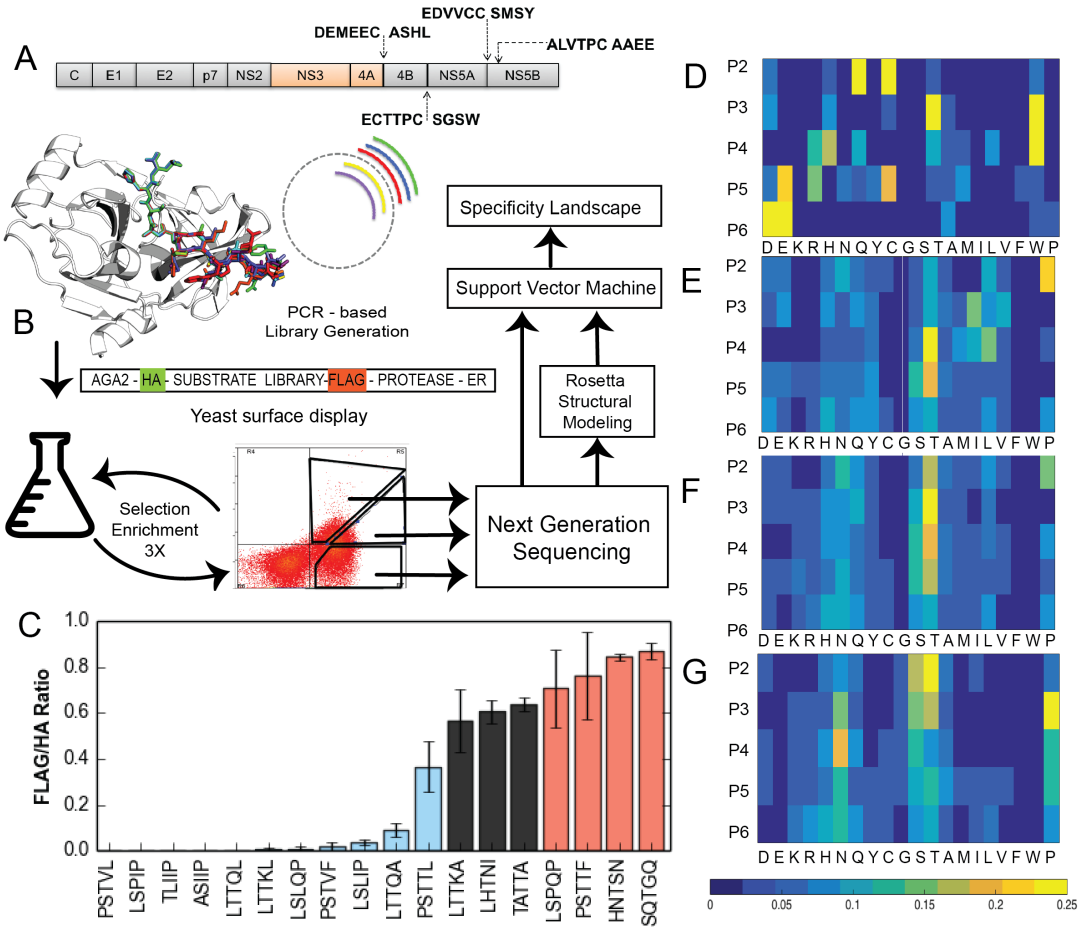


Figure 4.1. Overview of experimental workflow, validation of results

(A) The HCV viral polyprotein depicting marked biological cleavage sites for the HCV NS3 protease (B) overview of the experimental and computational workflow. (C) Validation of FACS gates for cleaved, partially cleaved and uncleaved sequences using yeast surface display assay (D) Sequences taken from in vivo samples of HCV patients (8726) as compared to (E) sequences determined by our assay as cleaved (7472), (F) as partially cleaved (8737), and (G) as uncleaved (14702)

4.3. Results

HCV NS3/4A protease is known to cleave four canonical cleavage sites on the hepatitis C viral polyprotein (Figure 4.1A), causing a cascade of viral assembly and maturation events. These cleavages (and a lack of cleavage of other parts of the polyprotein) are thus, critical for viral fitness. The high mutational load on the HCV polyprotein can lead to sequence variation in both the protease and substrate regions[208]. At the protein level, the distribution of mutational effects in a folded protein (protease) are modulated by both the thermodynamic stability and function (binding and cleavage), while the peptide substrate regions, which are found in flexible linker regions of the HCV polyprotein and connect component proteins, do not have a native tertiary structure. Therefore, we reasoned that a more direct sequence-cleavability mapping can be made for diversity in the substrate region without the need to additionally deconvolute the contribution from stability effects on tertiary structure. Secondly, it is more feasible to enumerate and evaluate by sequencing the substrate combinatorial diversity due to its shorter length (~7 residues) compared to the protease (>200 residues). Therefore, we mapped the viral protease-substrate interaction landscape for the HCV NS3/4A protease by considering all possible pentapeptide sequence combinations in its sequence recognition site at positions P6 through P2 following the Schechter and Berger nomenclature[209]. Positions P1 and P1', between which the scissile bond is present, were maintained as C and A, respectively, in this study. In the rest of this paper, we refer to individual pentapeptide patterns (e.g., the

canonical cleavage sites DEMEE, EDVVC, ECTTP, ALVTP) and omit the identity of the P1 and P1' residues.

4.3.1. Exploration of the (P6-P2) specificity landscape of the HCV NS3/4A protease reveals a diverse specificity profile

To mimic the viral intrachain arrangement of substrate libraries and the protease, we utilized a modified version of the assay described by Iverson, Georgiou and co-workers[40] as depicted in Figure 4.1B. A mutagenic library was created incorporating degenerate codons at P6-P2 specificity defining substrate positions[210,211]. In our assay, substrates are transported to the surface of yeast cells in a cleavage-dependent manner: the degree of cleavage is estimated by measuring the relative levels of substrate-flanking FLAG and HA tags using fluorescent, labeled antibodies. We have previously used this assay to test known and novel substrates of the HCV protease[103]. A first round of yeast surface display assay and Fluorescence Assisted Cell Sorting (FACS) was performed with an inactive protease variant (S139A) to select for high expression of library variants, for removing sequences containing stop codons in the substrate region, and to deplete substrate sequences that are cleaved by yeast ER proteases[212].

The resulting substrate variants from the pre-selection were subjected to rounds of yeast surface display assay and FACS with an active protease containing construct to select cleaved, partially cleaved and uncleaved variants using three sorting gates (Figure 4.1B), based on the relative levels of anti-HA and anti-FLAG fluorescence values (FLAG/HA ratio, ranging between 0, for completely cleaved, and 1, for completely uncleaved). Sorting gates were defined based on the distribution of populations observed for known

cleaved and uncleaved sequences[103]. This procedure was coupled with rounds of growth and selection to improve signal:noise for variants in each pool. Sequence profiles of the unselected population and isolated functional variants were determined using next-generation sequencing technology (Illumina NextSeq). Analysis of unique sequences in all sequenced pools showed that we identified a total of ~1.3 million sequences corresponding to ~30% of the possible amino acid diversity (3.2 million; Appendix 4). Analysis of sequencing and technical replicates as well as overlap between the sequence pools was used to determine a count threshold (raw count 11) to remove noise from the sequencing data (Appendix 4 and Figure 4.2). Based on these criteria, we identified 7472, 8737 and 14702 unique pentapeptide sequences in the cleaved, partially cleaved and uncleaved pools. In parallel, we performed Rosetta simulations on all 3.2 million sequences in the P6-P2 region, and used a Support Vector Machine to predict the complete protease-substrate interaction landscape using sequence information procured from the aforementioned library and Rosetta-generated energetic features (Figure 4.1B).

Several novel substrates identified from the three variant populations were tested as clonal populations in the yeast surface display assay system (Figure 4.1C, Figure 4.3) to validate that individual sequences fall into the gates used for selection from the library (Figure 4.4A-C). A subset of these sequences was also tested in vitro to ensure that the cleavage properties observed in the yeast system were reproduced with purified protease and substrates (Figure 4.4D).

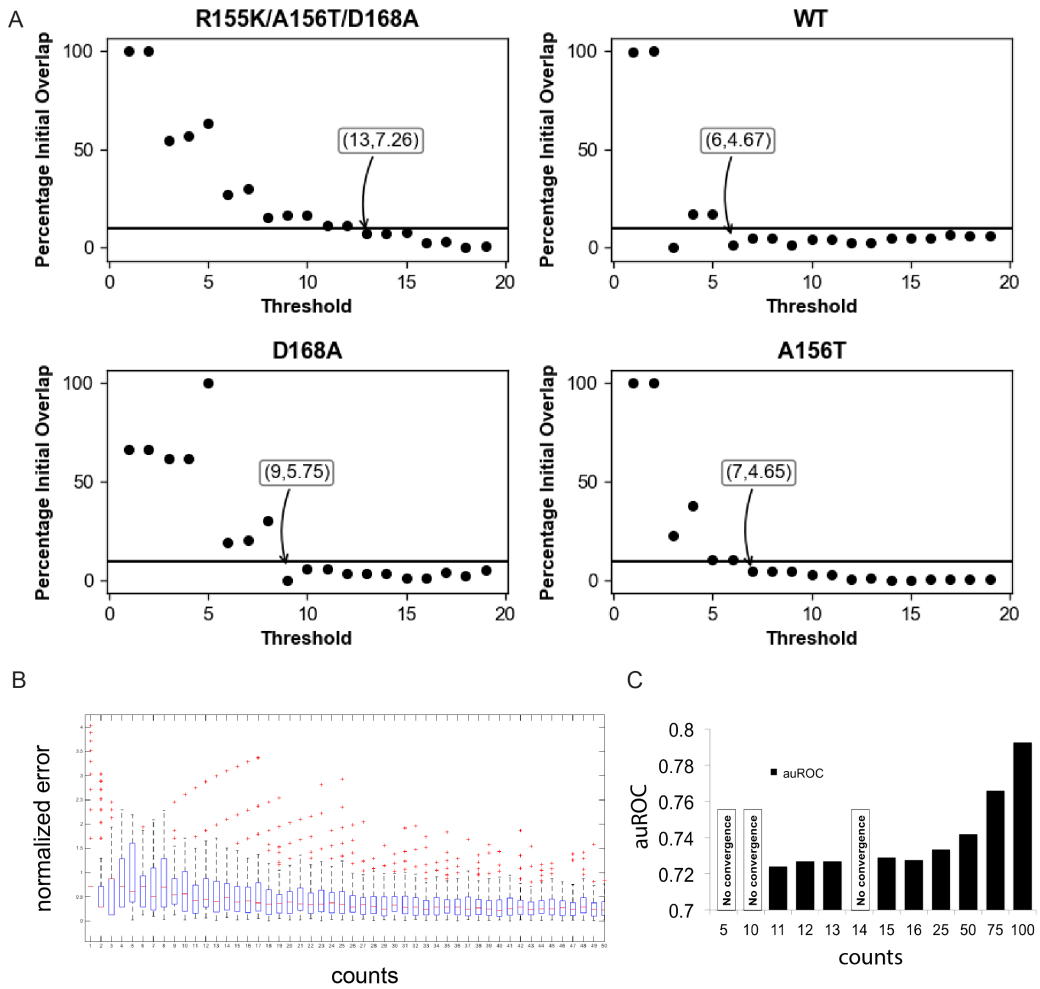


Figure 4.2. Threshold determination

(A) Threshold vs. percentage of initial overlap between cleaved and uncleaved sequences for all variants. The final threshold beyond which all other thresholds have a percentage overlap that is <= 10% is marked with an arrow (B) Duplicate population analysis. Normalized error is calculated for biological duplicates of cleaved samples by the formula: | (counts_S2 - counts_S) | / counts_S2, where sample S and S2 are biological duplicates (C) the Area Under the Curve for the ROC plot, when the SVM is used to classify cleaved versus uncleaved sequence pools at various count thresholds.

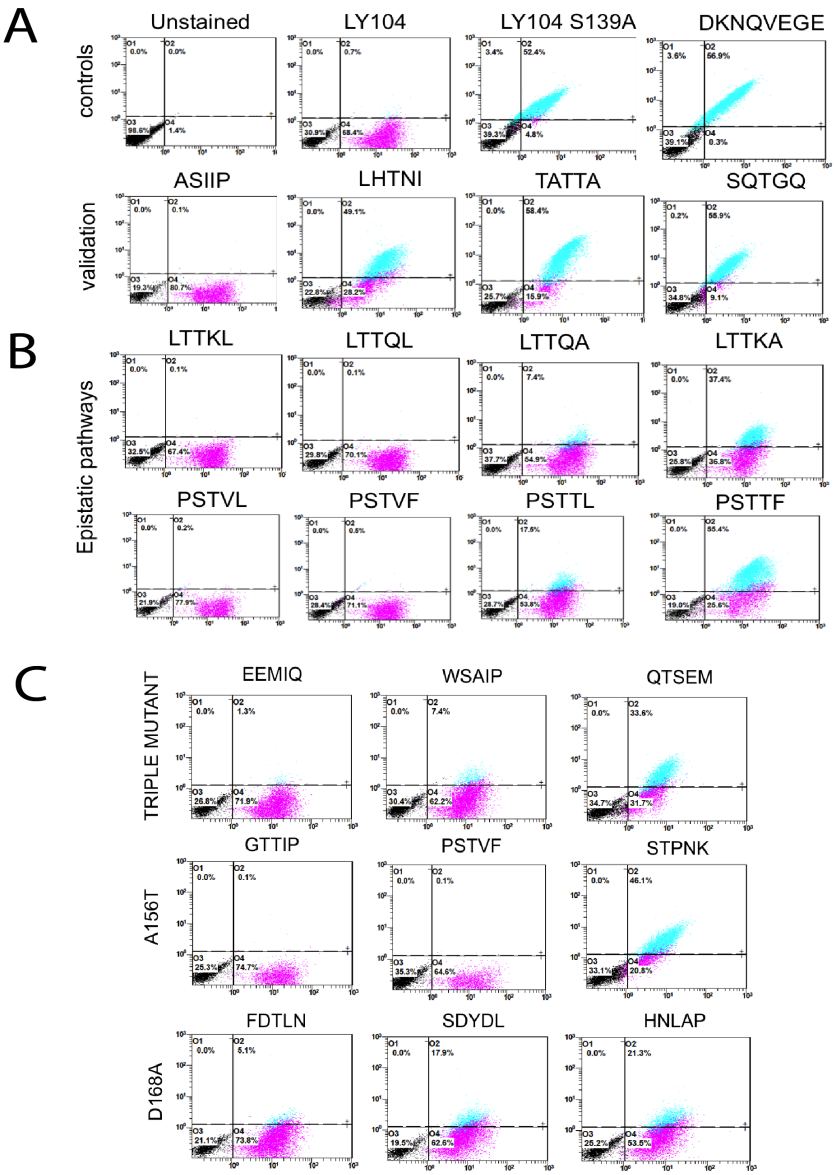


Figure 4.3. 2D plots of anti HA and anti-FLAG antibody signals seen in the flow cytometry assay

(A) display controls (B) Epistatic pathway validation (C) Drug resistant mutant validation plots

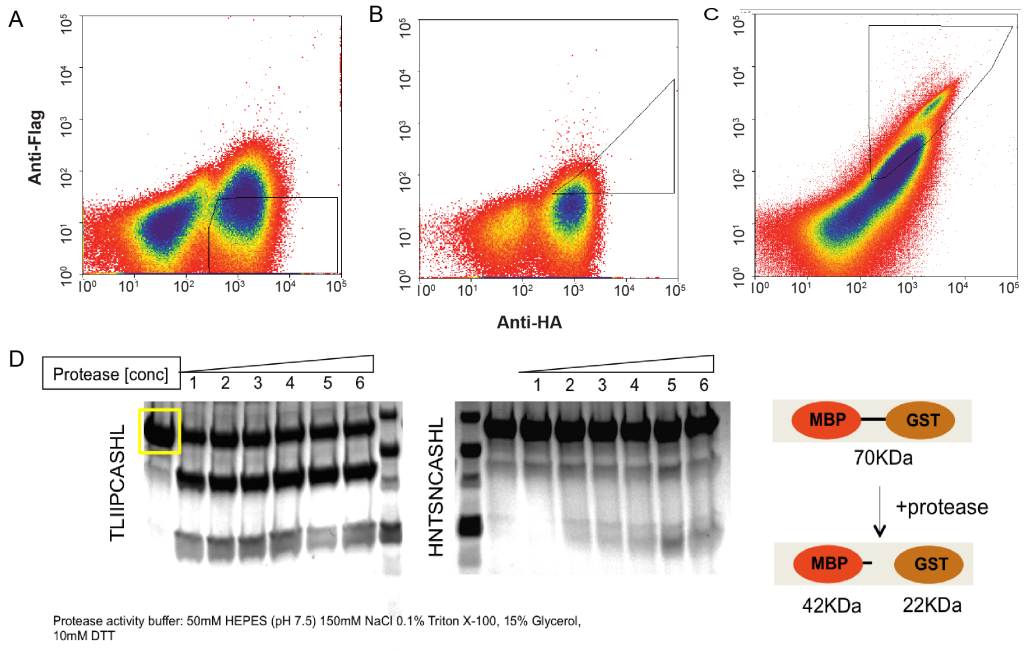


Figure 4.4. Flow cytometry 2D plots showing anti HA and anti-FLAG stains for cell populations collected after enrichment round three
(A) plot showing gate and cell population for cleaved (B) partially cleaved and (C) uncleaved populations (D) *in vitro* gel-based assay using an MBP- GST fusion protein (70KDa). Upon overnight incubation with increasing concentrations of the protease – 500 nM, 700 nM, 1uM, 2uM, 3uM, 4uM (wells #1 through #6) results in cleavage for substrate TLIIPCASHL whereas HNTSNCASHL displays no cleavage

We next analyzed the profiles of sequences in each pool. For the cleaved sequence pool, the obtained substrate sequence ensemble has greater diversity compared to substrates identified from viral genomes sequenced from patient populations (Methods, Figure 3.1D). For example, we observe that a more diverse subset of amino acids is tolerated at substrate positions P6 and P5 in our cleaved and partially cleaved pools (Figure 4.1E, F) whereas the patient isolated genomes display a high enrichment of Asp and Glu specifically at these positions. Another notable difference observed was the enrichment of small hydrophilic residues (Figure 4.1E, F), Ser (at P5) and Thr (at P4) in the cleaved and partially cleaved populations, in contrast to enrichment at P3 and P2 in the uncleaved population (Figure 4.1G). Strikingly, we found prolines enriched at position P2 in the

cleaved and partially cleaved populations and at P3 in the uncleaved populations, which corresponds well with the fact that 2 out of 4 canonical cleaved sequences have proline at P2 (ECTTP, ALVTP). While some of the above trends are also reflected in the sequences we tested during our method validation (Figure 4.1C), it is evident that individual positional enrichments cannot be directly used to predict the pool assignments of individual sequences. For example, His is enriched at P6 in the cleaved sequence pool, however the sequence HNTSN is experimentally determined to be in the uncleaved pool (Figure 4.1C, Figure 4.4). While individual positional preferences of amino acids are useful, these results clearly indicated that molecular recognition between the protease and substrate pools is highly (sequence) context-dependent. We concluded that interactions between substrate sidechains (mediated possibly via interaction networks in the protease) influence the cleavability, thereby motivating the need for an analysis of the determined specificity landscape using properties of whole pentapeptide sequences.

4.3.2. Clustering among cleaved, partially cleaved and uncleaved substrates

To visualize the functionally labeled sequence space of the experimentally derived substrates, we generated a force-directed graph[213,214] (Figure 4.5A) in which each node represents a sequence and is colored according to the functional pool to which it belongs. Nodes are connected by an edge if they differ by one amino acid (Hamming distance = 1). Cleaved substrates exhibit significant clustering in the resulting graph (Figure 4.5A). To examine the landscape in greater detail around the cleaved sequences, we generated a sub-graph of the cleaved sequences (Figure 4.5B). We identified four clusters in this graph using the Gephi[213] modularity algorithm and determined corresponding profiles for

each cluster. One identified cluster is clearly related to a canonical substrate, DEMEE. The other three clusters appear to have similarities with the other three canonical substrates (ALVTP, ECTTP, and EDVVC) but are less distinct from each other compared to the DEMEE cluster. These results indicate that the four canonical cleaved sites in the viral polyprotein are all members of mutationally robust clusters. Single amino acid changes within the cluster lead to other cleaved sequences, thereby buffering the impact of the heavy mutational load on the virus.

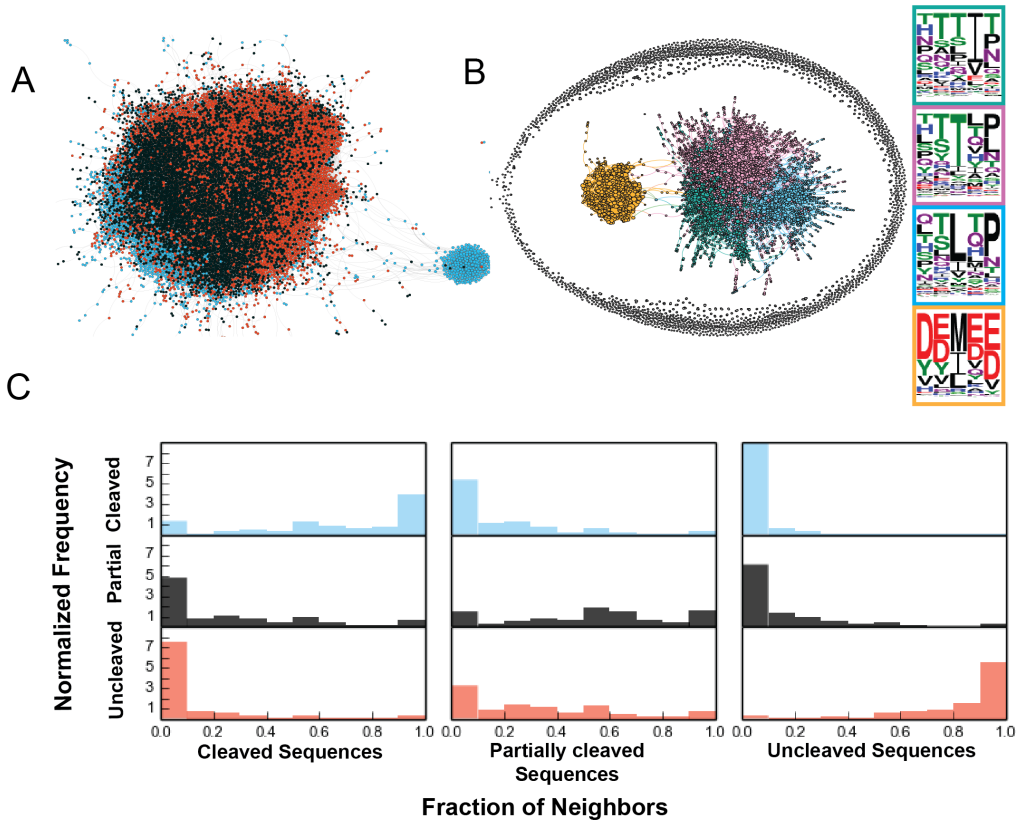


Figure 4.5. Force directed graph representation of experimental landscape; Neighbor analysis
(A) Force- directed graph of amino acid sequence space. Blue nodes are cleaved, red are uncleaved, and black is partially cleaved. Edges connect nodes that are within one hamming distance of each other (B) Force- directed graph of cleaved sequence. Colors denote clusters which are shown as specificity profiles outlined in the same color as the corresponding cluster (C) Frequency of neighbors for cleaved, partially cleaved, and

uncleaved sequences denoting cleaved neighbors shown in blue bars, uncleaved neighbors depicted in red and partially cleaved neighbors depicted as black.

To determine if this clustering behavior observed in the cleaved sequence pool is also found in the partially cleaved and uncleaved pools, we calculated the fraction of neighbors for sequences with neighbors that belong to the same functional pool (Figure 4.5C). We find that similar to cleaved sequences, uncleaved sequences are also most frequently surrounded by uncleaved neighbors indicating clustering behavior for this functional pool as well. On average, cleaved sequence neighbors are 66.4% cleaved, and uncleaved sequence neighbors are 83.3% uncleaved. Partially cleaved sequences are the least clustered among the three pools, having on average 53% neighbors belonging to the same pool. These distributions indicate that in the specificity landscape, clusters of partially cleaved sequences surround clusters of cleaved and uncleaved ones.

To delineate how the three functional populations, which appear to be individually clustered in sequence space, are connected to each other, we used the PageRank metric[215]. This metric predicts the likelihood of reaching a node given a random walk on the substrate specificity landscape starting from a chosen sequence. Strikingly, partially cleaved substrates have higher PageRanks (Figure 4.6A) than either cleaved or uncleaved substrates, indicating that they are most likely to be reached on long unbiased evolutionary trajectories starting from the canonical cleaved sequence DEMEE, the sequence that was used as the template for library generation. These connectivity patterns imply that partially cleaved node clusters may act as an evolutionary buffer on the substrate landscape, thereby enhancing robustness.

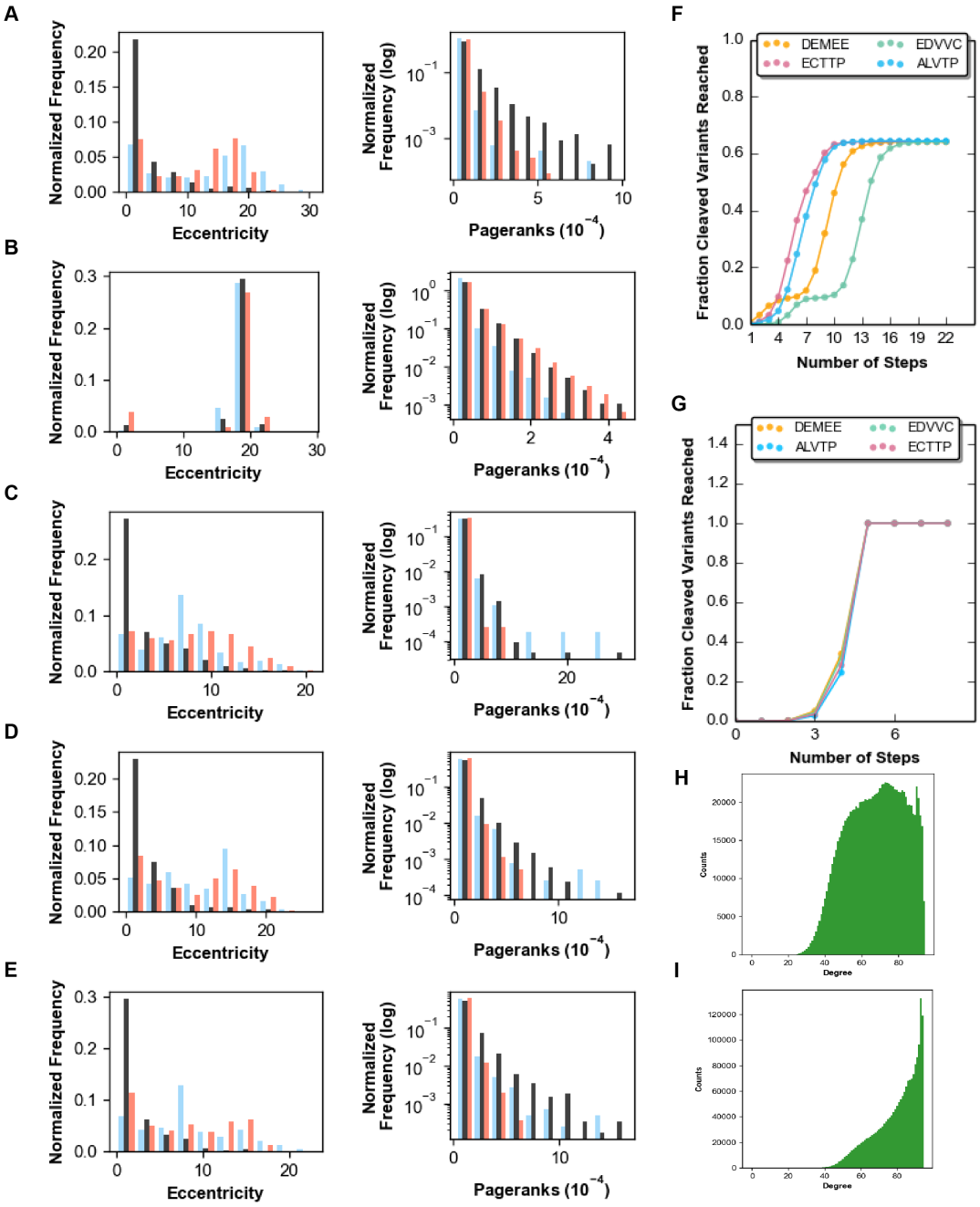


Figure 4.6. Graph metrics for WT and mutant protease

Cleaved (blue), uncleaved (red) and partially cleaved (black) graph metrics for (A) wild type HCV (B) randomly generated graph (C) R155K/A156T/D168A triple mutant (D) A156T and (E) D168A. Partially cleaved sequences generally have higher pageranks and lower eccentricity. Number of mutations vs. fraction cleaved variants reached for (F) experimental and (G) SVM-generated graphs. Degree distribution for cleaved sequences subset (H) and uncleaved sequences subset (I) of SVM derived graph.

The graph generated by the experimentally derived sequences is incomplete (~30,000 nodes out of the 3.2 million possible). To test if the observed clustering and PageRank distributions are an artifact of the limited sampling in the experiment, we generated a control random graph (Figure 4.7A) with the same number of nodes and edges, but having a randomly rewired connectivity. Both partially cleaved and uncleaved sequences are found to have higher pageranks than cleaved sequences in this random graph, indicating that the higher pageranks of partially cleaved sequences than cleaved and uncleaved sequences in the original experimental graph is significant.

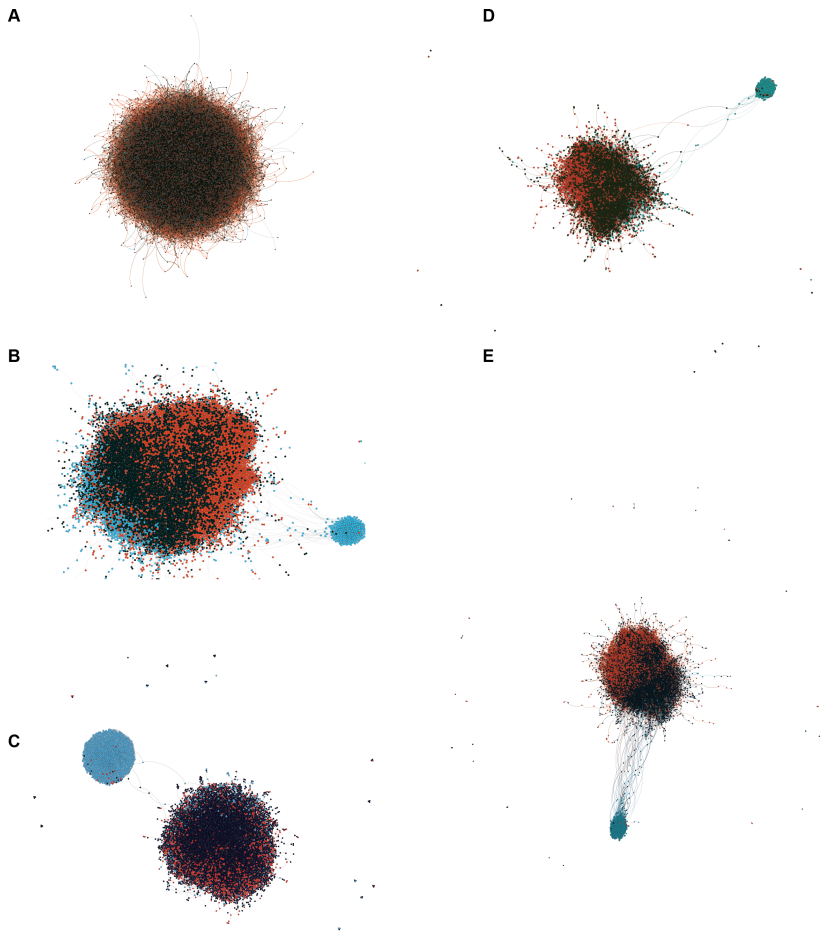


Figure 4.7 Force – directed graphs for WT and mutant proteases
(A) randomly generated graph (B) wild type HCV protease (C) R155K/A156T/D168A triple mutant (D) D168A variant (E) A156T mutant

4.3.3. Energetic features derived from Rosetta modeling enable reconstruction of the complete protease-pentapeptide substrate landscape

While the experimentally-derived populations of the cleaved, partially cleaved and uncleaved sequences revealed striking clustering patterns in sequence space, it is not clear if these connectivity patterns would be preserved in a complete graph containing the complete diversity at five positions (3.2 million sequences). Therefore, to predict cleavability of all possible 3.2 million sequences in the interaction landscape, we used a Support Vector Machine (SVM)-based method that we developed previously[103]. Briefly, each sequence was threaded onto a bound complex based on a modeled near-attack conformation a crystal structure of the protease, and the complex was then relaxed to maintain favorable catalytic geometry. Energy evaluation of each of the 3.2 million complexes was performed using Rosetta and Amber simulation packages. A binary classification (cleaved/uncleaved) SVM was trained on a subset of experimentally identified sequences that passed a more stringent threshold of enrichment compared to the unselected pool in our assay (1817 cleaved and 3605 uncleaved sequences) as well as sequences identified by Shiryaev *et al*.[41] for a total of 7342 unique sequences. Training features consisted of structure-based features (energies of interaction) and sequence-based features (see Appendix 4, Figure 4.8A). We initially cross-validated the SVM on the training set using an 80:20 split with 100 iterations, which yielded an average auROC of 0.96 (Figure 4.8B) indicating high recapitulation of training data (a perfect performance would lead to an auROC of 1). We then used the SVM to predict cleaved and uncleaved labels for the remaining 3,192,658 sequences. These predictions have a precision of 0.96 at a recall level of 0.91 for an overall accuracy of 0.96 (Figure 4.8B) for

all experimental sequences. We experimentally validated cleavage predictions for six substrates as clonal populations using the yeast assay and find good agreement with the SVM-based predictions (Figure 4.9A). We visualized a sub-graph of predicted cleaved sequences present at a distance > 2 from the hyper-plane constructed by the SVM (Figure 4.8C). The experimentally identified cleaved sequences are recapitulated well, and distributed evenly across the predicted cleaved population.

4.3.4. Structural and energetic bases for observed specificity patterns

Having obtained and validated predictions of cleavability by combining experimental and computational data, we turned to structural models of protease-substrate complexes to obtain insight into the underlying structural basis of observed specificity patterns. For example, a comparative analysis of the partially cleaved substrate 'TATTA' and canonical substrate 'EDVVC' reveals that the former, composed of small residues does not completely occupy the substrate cavity volume (Figure 4.9B, C) whereas 'EDVVC' occupies the entire cavity. The lack of voids at the interface and several hydrogen bonds formed by the canonical lead to better binding (Binding interaction energy = -80.2 Rosetta energy units (Reu), as opposed to -77.5 Reu for TATTA), resulting in better cleavage for this substrate. Similarly, models of the uncleaved sequence FWPPM (Figure 4.9D) reveals that the side chains are found to have steric clashes with the protease side chains. Apart from sidechain-based interaction patterns, models also capture backbone conformational changes that affect the orientation of the substrate in the active site. For example, in the model corresponding to the sequence RPGPG (uncleaved), the proline present at P3 in RPGPG (Figure 4.9E) bends the peptide chain away from the protease,

resulting in breaking of the crucial backbone hydrogen bond patterns that are characteristic of protease-substrate interactions[4].

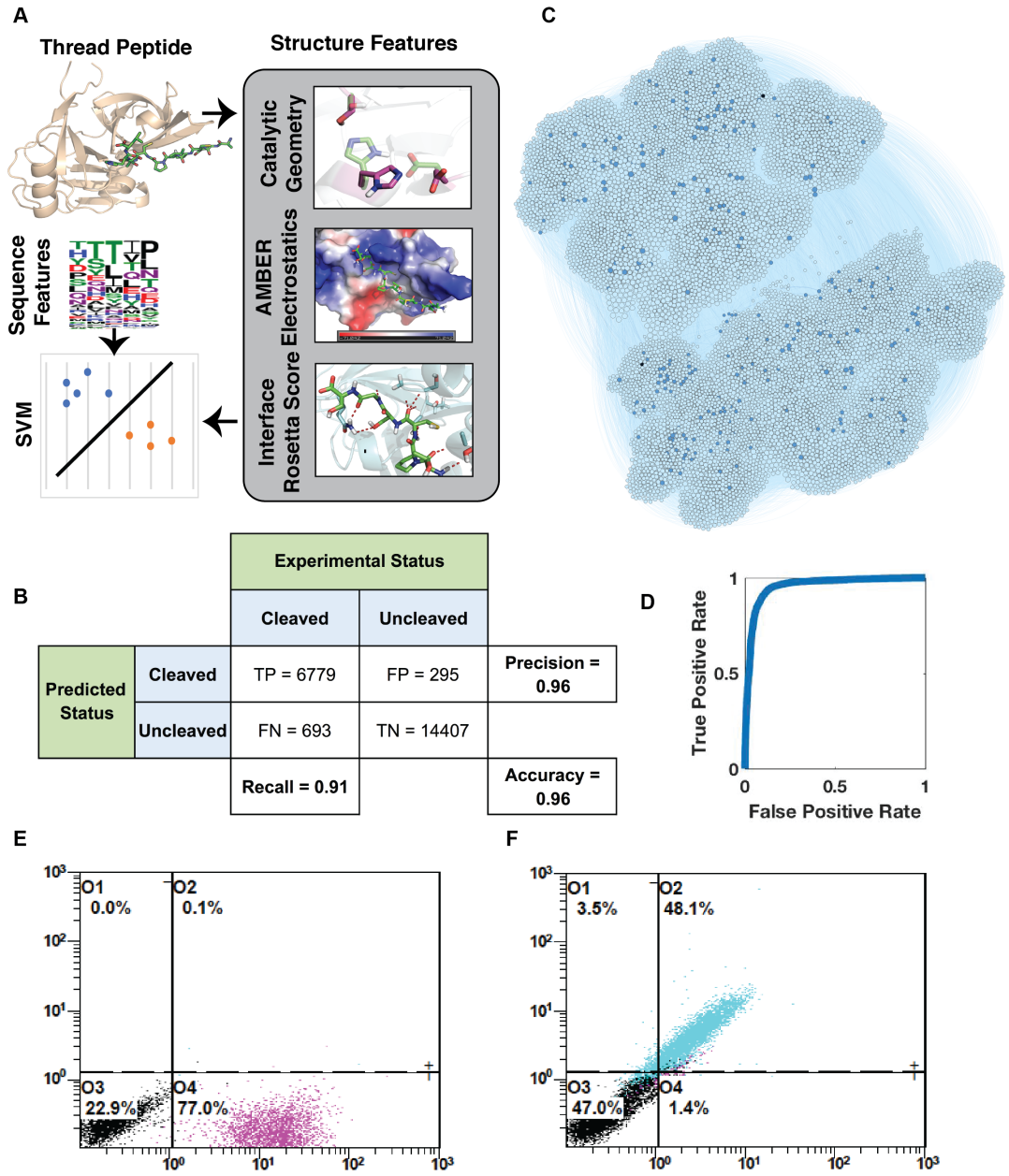


Figure 4.8. SVM generation workflow, contingency table and validation results
(A) Schematic workflow for SVM generation (B) Sub-graph of SVM predicted cleaved sequences with a distance > 2 from the hyperplane. Experimental cleaved sequences are dark blue and experimental partially cleaved sequences are depicted as black. (C) Contingency table for SVM prediction (D) ROC plot of cross-validation on training set

for SVM (E) Flow cytometry plot for ECTIP (SVM- predicted cleaved) (F) Flow cytometry plot for RPGPG (SVM – predicted uncleaved)

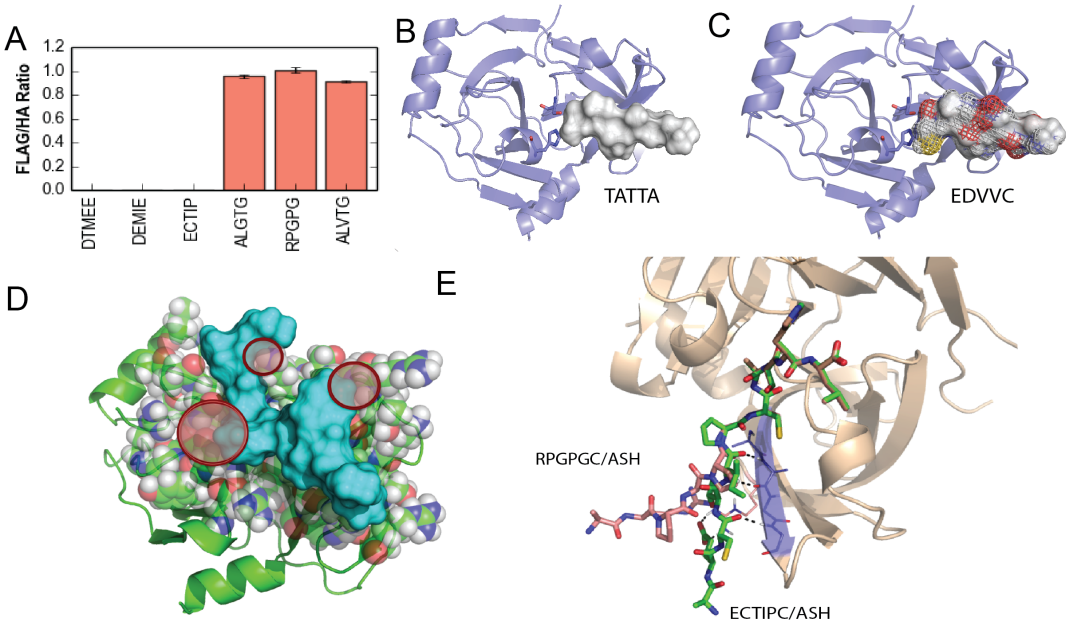


Figure 4.9. Structural basis for SVM prediction & validation
(A) Validation assay performed for three predicted cleaved and uncleaved sequences using a yeast surface display based technique (B) and (C) depict the volume occupied by TATTA and EDVVC, EDVVC occupies an optimal volume, making good contacts with the protease residue side chains. TATTA fits in the available space but does not make optimal contacts, thus resulting in suboptimal interaction energetics making TATTA a suboptimal substrate (D) Peptide (surface shown in blue) "FWPPM" sterically clashing against the protease chain (E) Structure of two models, ECTIP (cleaved) and RPGPG (uncleaved)

Structural analysis also allows rationalization of non-additive (epistatic) patterns between amino acid substitutions. We detected the presence of both positive and negative epistasis in our experimental data, and further investigated two cases (Figure 4.10A). We examined a predicted negative epistasis pathway (Figure 4.10B), where single-mutant P at position P4 and single-mutant Q at position P3 both result in a cleaved substrate but the double-mutant PQ at position P3-P4 is uncleaved. We measured the mutual information

(Figure 4.10C; Appendix 4) between positions P3 and P4 in the experimentally derived cleaved sequence pool and found that both L at P3 and Q at P4 (corresponding to sequence LSLQP) and P at P3 and I at P4 (corresponding to sequence LSPIP) are correlated, indicating that these two amino acid preferences are found in the experimentally-derived cleaved population at a higher incidence than expected by their individual incidence. However, the correlation for P at P4 and Q at P3 (corresponding to sequence LSPQP) is low, suggesting that the PQ pattern is depleted in the cleaved sequence population. Structurally, the sequence LSPQP (Figure 4.10D) may have an increased PPII (polyproline-II) helix propensity[216,217], causing the substrate to twist out of a catalysis-competent binding conformation in our models. The PPII helix propensity for the sequence LSPIP is lower thus resulting in retention of the extended substrate binding conformation that is favorable for catalysis[4]. Thus, analysis of models of individual substrates provides atom-resolution insights into how the underlying biophysics of molecular recognition by the protease shapes the observed specificity landscapes, including non-additive effects.

Having validated (Figure 4.10E) these examples of double-mutant epistatic networks, we enumerated the double mutant epistatic networks present in the experimental data, and found that the majority of these epistatic networks (60.7%) involved cleaved and partially cleaved sequences only. The preponderance of epistatic networks at the cleaved/partially cleaved boundary indicates that the boundary between cleaved and partially cleaved sequences is more rugged than the boundary between cleaved and uncleaved sequences, further highlighting the role of partially cleaved sequences as a biophysical buffer in sequence space, leading to “gradient robustness” proposed by Tawfik and co-workers.

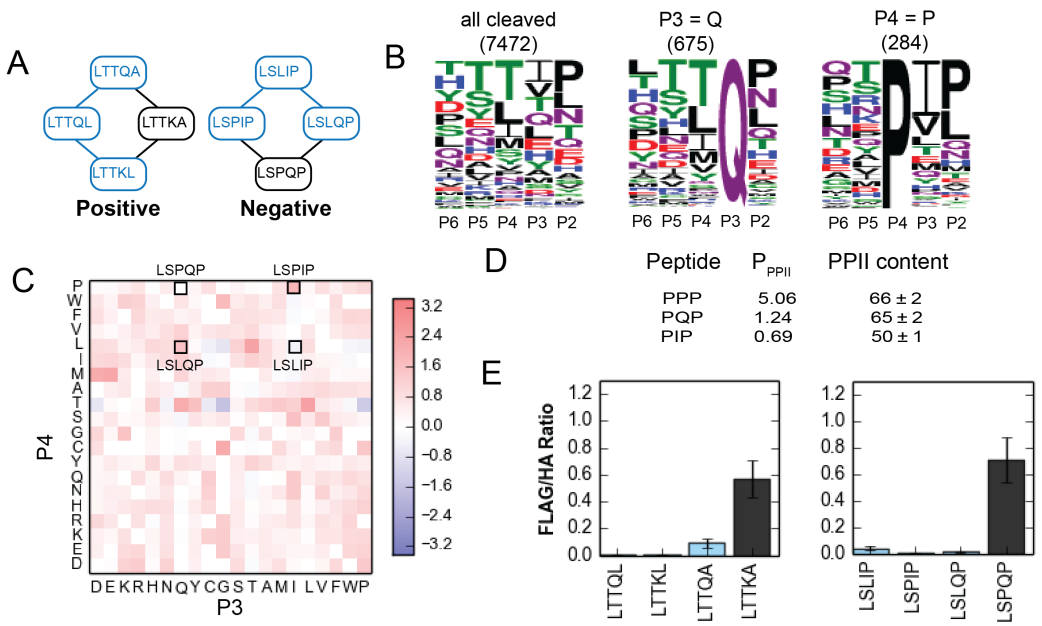


Figure 4.10. Structural basis underlying epistasis found on the interaction landscape.
(A) Examples of positive and negative epistasis. Cleaved sequences are highlighted in blue, partially cleaved in red. (B) Specificity profiles for entire cleaved set (left), sequences with glutamine at P3 (middle), and proline at P4 (right). (C) Heatmap of correlations between positions 3 and 4, as measured by mutual information. (D) Polyproline II structure propensity of peptides (see text). (E) Experimental validation of the sequences in both positive and negative epistatic pathways, performed using yeast surface display. Blue bars indicate sequences that are expected to be cleaved and black bars indicate sequences that are expected to be partially cleaved.

4.4.5. Mutational robustness and possible evolutionary trajectories in the experimentally-determined and computationally reconstructed landscape

Having computed the entire P6-P2 specificity landscape, we next examined the connectivity patterns between cleaved and uncleaved sequences in this reconstructed landscape. As with the experimentally determined landscape, the reconstructed landscape also shows clear evidence of clustering between cleaved and uncleaved nodes (Figure 4.6G-I), indicating that mutational robustness extends to regions of sequence space not

covered in our library, and is an essential feature of this protease-substrate interface. As our SVM-based approach is a binary classification scheme, partially cleaved sequences are classified in either cleaved or uncleaved pools. Attempts to build a 3-way classifier failed due both to the noise from the experiments as well as difficulty in estimating small energy differences in Rosetta simulations. Further improvements in each methodology may allow the prediction of partially cleaved sequences.

As the hepatitis C virus is subject to a considerable amount of evolutionary drift, we investigated the impact of the pathways of drifting on the landscape on maintaining function. For the experimentally determined landscape, we calculated the number of mutations from each canonical sequence to the functional boundary and plotted the fraction of cleaved substrates that can be reached at each step (Figure 4.6F). The curves for both DEMEE and EDVVC reach a small initial plateau and then rise sharply, indicating that both are surrounded by a cluster of cleaved sequences and then must bridge a largely non-functional region of the graph to reach the rest of the cleaved sequences, whereas the curves for both ALVTP and ECTTP rise steadily, indicating that the topology surrounding these sequences is less rugged.

Both the reconstructed and experimentally-derived landscapes feature several “novel” cleaved sequence patterns (defined as >3 substitutions away from a canonical recognition motif). To investigate if these novel sequences can be reached, as an example, we generated a sub graph of the sequence space connecting the canonical cleaved sequences (DEMEE, EDVVC, ECTTP, ALVTP) with each other as well as the novel cleaved

sequences, e.g., PSTVF (Figure 4.11A). Analysis of all inter-node shortest paths on these networks shows that there exist many paths between canonical and novel sequences that do not include uncleaved nodes (viable paths) while some paths involve traversal of at least one predicted uncleaved node (unviable paths; Figure 4.11B). All canonical sequences are more connected to each other than to any of the novel sequence motifs, suggesting that the latter may be "kinetically" less accessible during evolutionary drifts.

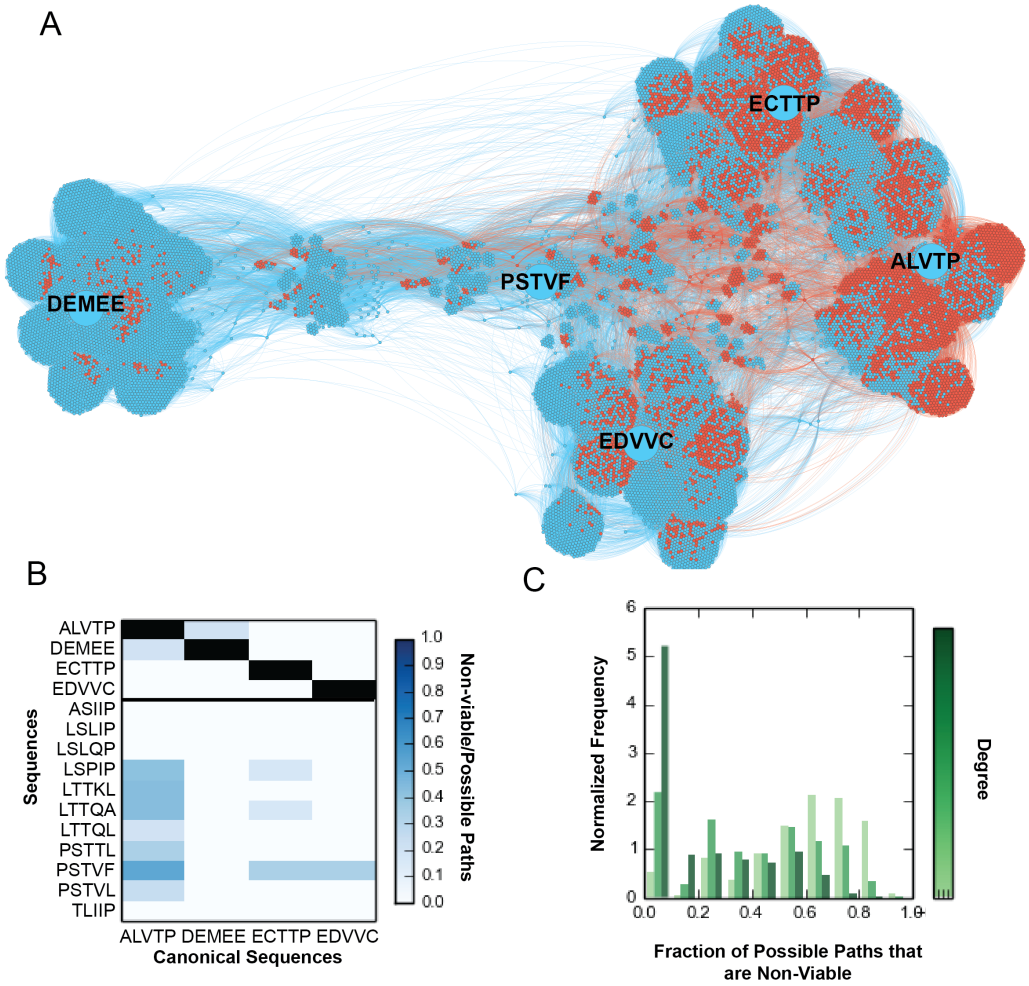


Figure 4.11. Force directed graph representation between five canonical and novel sequences and graph metrics for validation

(A) Force-directed interaction graph between the five canonical sequences – DEMEE, ECTTP, EDVVC, ALVTP and the novel cleaved sequence PSTVF (depicted by large blue nodes). The graph depicts neighbors of all intermediate sequences between PSTVF and all canonical sequences. The cleaved sequences in the interaction pathways are

denoted by blue nodes and the uncleaved are denoted by red (B) The fraction of uncleaved nodes present in the shortest paths from both canonical sequences and novel sequences to all canonical sequences (C) Degree vs. fraction of the shortest paths uncleaved between all novel sequences and all canonical sequences.

We calculated the fraction of non-viable paths between canonical sequences and compared it to the fraction of non-viable paths between canonical sequences and novel sequences. The latter shows a higher, albeit still small, fraction of non-viable paths (Figure 4.11B). We also find that those novel cleaved sequences that have a higher fraction of cleaved neighbors (higher degree) are more likely to have a higher fraction of viable trajectories to canonical nodes (Figure 4.11C). Thus, it appears that the higher single mutational robustness of a given novel sequence is correlated with its ability to be reachable from/to canonical sequences that are at least three amino acid substitutions away in sequence space. Further contributions from codon usage in the host context may modulate the reachability of different substrates by making some amino acid changes even less likely. Our analysis above leaves out these contributions to selectively delineate the impact of amino acid-level effects.

4.3.5. Protease specificity landscape may contribute to negative selection

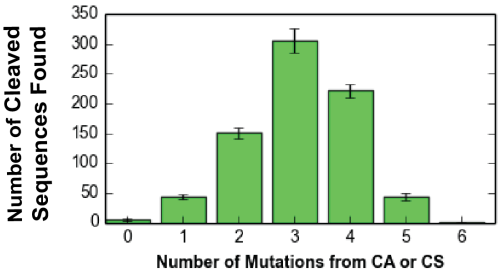
Sequences of patient-derived genomes indicate that the HCV virus is under strong negative selection[218,219]. Although the underlying mechanisms are not well understood, several factors have been invoked to explain the observation of a low dN/dS ratio (number of non-synonymous to synonymous substitutions in the genome) in the patient-derived populations including intrahost competition between quasispecies, and immune evasion[220]. Given the centrality of the protease in viral maturation, we asked if

maintenance of cleavability (and uncleavability) in different parts of the polyprotein also contributes to negative selection, and what, if any, are the limits imposed by the recognizability of different polyprotein regions by the protease on their variability.

As our reconstructed landscape provides information on all pentapeptide sequence combinations (followed by Cys-Ala), we asked if overlapping pentapeptides in the other parts of the polyprotein (apart from the known cleavage sites) are likely to be cleaved, especially if they acquired a Cys-Ala pattern in the two immediately downstream residues (thereby acquiring the necessary heptapeptide pattern that would be cleaved). If several regions of the polyprotein are poised to be cleaved upon acquisition of the Cys-Ala motif, an error catastrophe may ensue upon increasing the mutational load. We performed a genome-wide comparison of patient derived sequences with sequences predicted as cleaved by our SVM classifier. Each viral genome[219] was split into overlapping 5-mer peptide sequence fragments using a one-residue sliding window method. These 5-mers were compared to the pentapeptide sequences predicted by our approach as cleaved. If the patient-derived pentapeptide sequence was found in the cleaved pool, we calculated the minimum nucleotide mutational distance of the successive two residues from the DNA sequences that code for 'CA' and 'CS' which are known to be the canonical P1-P1' sites favoring cleavage by the HCV NS3/4A protease (Figure 4.1A). The results (Figure 4.12A, B, Figure 4.13A-H) indicate that the majority (~70%) of patient-derived translated pentapeptides are found in the uncleaved pool. Of the remaining (~30%) 5-mer sequences that are identified as potentially cleavable (if they acquire a CA or CS as the following two amino acids), 74.1% pentapeptides from all genotypes of the virus require more than

three nucleotide changes to acquire a 'CA' or 'CS' at the P1-P1' sites (Figure 4.13A-H). The avoidance of acquisition of a cleavable sequence in other regions of the protein, made feasible by codon usage, may thus contribute to the previously described negative selection pressure on the HCV genome[218], and may be reflected in the measured low dN/dS rates in the non-structural regions of the protein[219]. Additional avoidance of non-productive cleavage may also result at the structural level from altered dynamics[221] and/or the post-translational structural context of the potentially cleavable regions – these may be buried (inaccessible to the protease) or adopt secondary structures that are incompatible with the extended conformation required to fit in the protease active site[36,21].

A



B

Genotype	Average Cleaved	Average Uncleaved
1a	748	2246
1b	780	2221
2a	803	2200
3a	791	2193
4	792	2179
5a	780	2197
6a	779	2216

Figure 4.12. Evidence for negative selection of canonical substrate areas
(A) Bar plot depicting the number of DNA mutations required to mutate from current protein sequence to 'CS' which is the scissile bond sequence for the HCV NS3/4A protease (B) Table depicting the classification of all genotype derived 5-mers as classified by our SVM based predictor

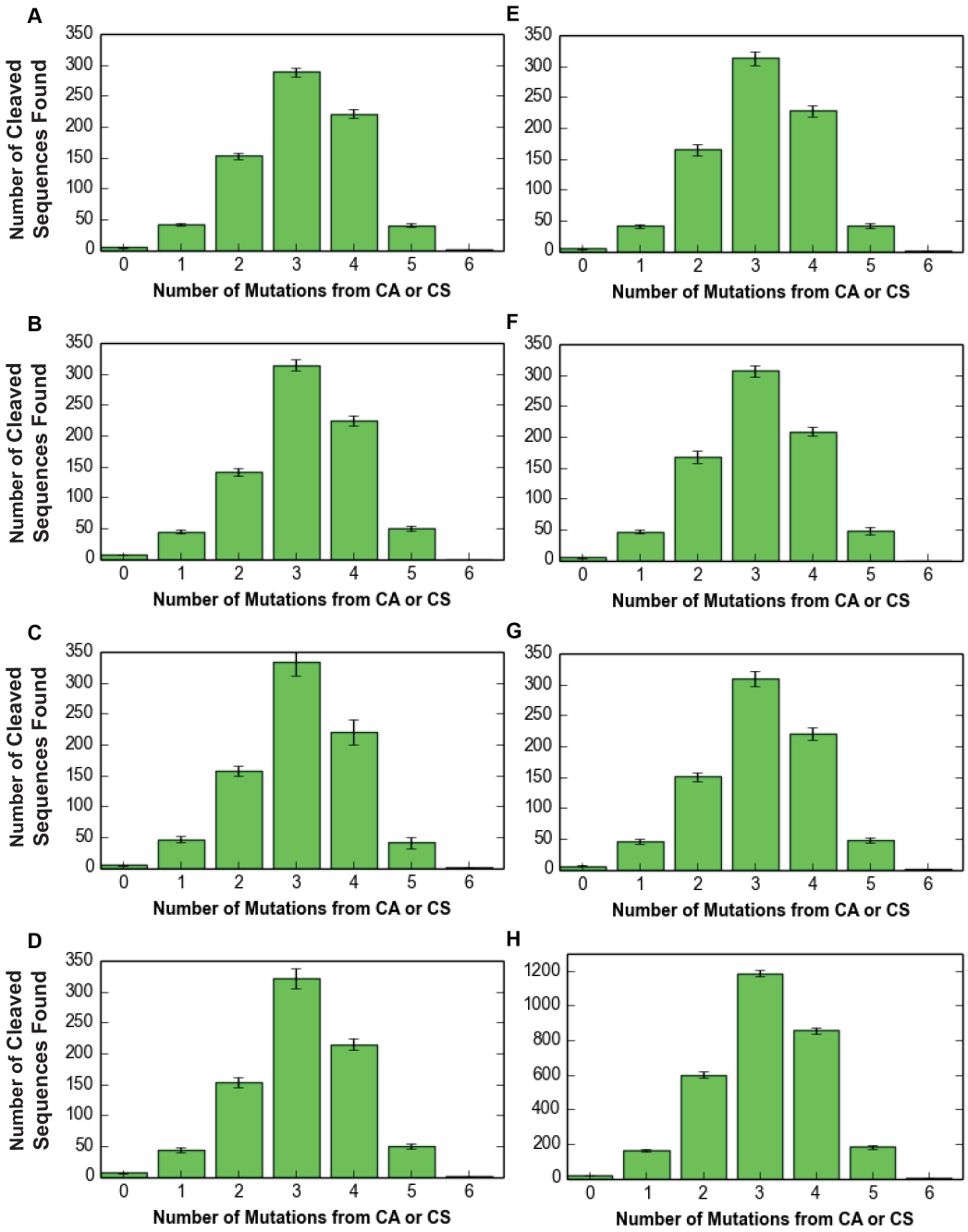


Figure 4.13. Plot depicting the number of DNA mutation required to mutate from current protein sequence to 'CS' which is the scissile bond sequence for the HCV NS3/4A protease for all genotypes

(A) strain 1a (B) strain 1b (C) strain 2 (D) strain 3 (E) strain 4 (F) strain 5 (G) strain 6 (H) control. Control is the distance from CA/CS for all 2-mers in all genotypes

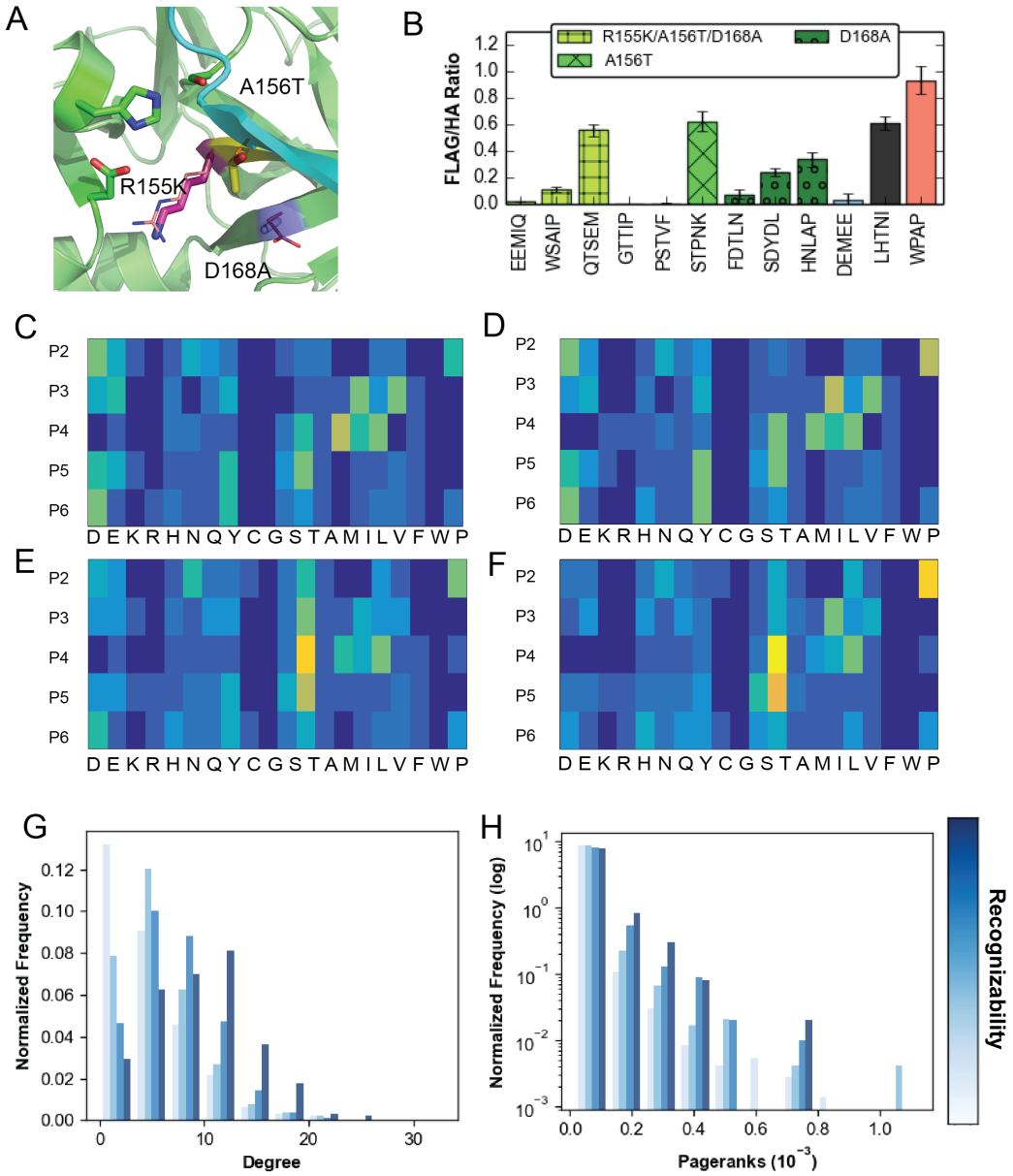


Figure 4.14. Validation, graph metrics and specificity profile for Drug resistant mutant proteases

(A) Drug-resistant variant structures. Mutations are outlined in sticks and WT residues in lines. Active site residues are represented as green sticks (B) Validation assay performed using yeast surface display for each of the mutants (C-F) Mutant specificity logos for the triple mutant, D168A, A156T and wild type showing that the mutants have very similar specificity profiles with slight variation as compared to the WT (G-H) Substrate sequences that are recognized by a greater number of variants have higher degrees (G) and pageranks (H)

4.3.6. Specificity landscapes of Drug Resistant Protease variants

As the NS3/4A protease plays a key role in the viral assembly and maturation process, it is a target for therapeutics that aim at neutralizing viral activity. However, due to prevalence of quasispecies that are lurking at low levels in the population[222], several viral variants get exposed to the drug. Some of these develop resistance, and propagate to form Resistance Associated Variants (RAVs). To investigate how drug-resistant variants of the protease affect the mutational robustness, we explored the specificity landscape for three RAVs – A156T, D168A, R155K/A156T/D168A (Figure 4.14A). If the connectivity patterns of the sequences recognized by the RAVs are dramatically different and less clustered, it would indicate that their evolutionary fitness might be more limited under the heavy mutational load, as drifts on the substrate side would abolish the molecular interaction required for viral maturation. In this scenario, treatment with mutagens may be a desirable therapeutic strategy to induce error catastrophes. On the other hand, if similar mutationally robust connectivity is detected, the RAVs are likely to have a similar evolutionary potential as the wild type, and have an additional selective advantage in the population in the presence of the drug.

To obtain the landscapes of the protease variants (Figure 4.5B-E), we generated the library using a PCR amplification based strategy; isolated functional variants using FACS, deep sequenced the isolated populations and validated mutants (Figure 4.3, Figure 4.14B) identified from these populations using the yeast surface display assay. We find that the RAVs demonstrate a similar sequence profile to each other and to the wild type protease (Figure 4.14C-F). Upon comparing the graphical properties of the specificity landscapes of the various protease variants, we observe that substrates that are

experimentally detected in the cleaved pools of a greater number of protease variants are more reachable (higher pageranks) and more connected (higher degree) in each graph (Figure 4.14G, H). As our goal was to compare gross features of the specificity landscapes for the wild type and variant proteases, we did not perform detailed structure-based calculations for RAVs. Nonetheless, these data indicate that more recognizable substrates appear to be more robust to changes in the protease, and indeed, mutational robustness is a key feature of this specificity landscape.

4.4. Discussion

For RNA viruses, such as HCV, which have a high mutation rate, it has been hypothesized that viral evolution occurs via "survival of the flattest": the most conserved viral form is not necessarily the most fit, but instead is the one most robust to mutation – thus mutational robustness may provide an evolutionary advantage[145,151,153]. Our data, based on combining information gleaned from library screening in yeast, deep sequencing, and structure-based modeling, provide atomic-resolution insight into how mutational robustness may be encoded in the molecular recognition landscapes involved in viral maturation, and indicate that cleavage specificity of the HCV NS3/4a protease is robust to patient-derived mutations in both the substrate regions as well as the protease. However, molecular interaction between the protease and substrate, which key for viral survival, is but one of the many evolutionary forces at play, especially in the "wild"[223]. Other factors such as the intrahost population size, stability and structure of the viral RNA genome, and interactions between the host and viral machineries and other

environment dependent factors are also important to consider while considering evolutionary demands and trajectories.

We used a yeast surface display based assay that relies on the cleavage of the substrate region in the ER of yeast followed by cell sorting into gates and deep sequencing. We note that our assay is qualitative, and does not permit association of the detected signal from deep sequencing with quantitative cleavability of substrates. Indeed, while we have validated that assignments to the three different pools is accurate with at least ~20 individual sequences, the identified cleaved and partially cleaved substrates may represent a wide range of catalytic efficiencies. A limitation of our technique is that it flattens this diversity into two pools. On the other hand, the assay construct with the protease and substrate on the same chain is a good representation of the situation in the virus, where the substrates of the protease are part of the same polyprotein (although both *cis* and *trans* cleavages occur) leading to high effective concentrations of substrates ([S] >> K_M) *in vivo*. Under these saturating conditions in the virus and in our assay, we argue that selectivity and catalytic efficiency are both determined to a great extent by the goodness of fit of various substrates in the protease active site (i.e. by the *relative* binding between the different substrates). Similarly, our machine learning approach to combine experimental and computational data also is not without errors, showing a false-positive rate of ~5-10% on the experimental data. While we have validated several predictions on individual sequences (Figures 4.1, 4.9, 4.14), it is possible that some individual sequences may be mispredicted. However, the overall trends regarding the connectivity patterns observed for the entire landscape should be robust to the misprediction noise. Further

ongoing development of the computational and experimental methods that we utilized is expected to help increase the accuracy of the approach.

HCV infects ~3% of the world population and the limited number of available viral genome sequences show low sequence heterogeneity in the protease and its substrate regions. Nevertheless, resistance mutations upon protease inhibitor drug treatment arise in a facile manner in the patient population, suggesting that genetic heterogeneity (quasispecies) indeed exists, possibly at levels too low for being captured in patient-derived sequencing. Spontaneous emergence of diverse HCV protease mutations (including drug-resistant mutations) was demonstrated recently by Liu and colleagues in continuous evolution studies of the protease[224], as well as by Sanjuan and colleagues in viral replicon assays coupled to ultradeep sequencing[208]. Our results show how genetic heterogeneity is entirely consistent with the robustness of a key protease-peptide interaction in the virus, and therefore, provide a biophysical baseline for understanding evolvability of HCV, and for evaluating inhibitor drug resistance risks. For example, our analysis suggests that viral evolution occurring at the substrate sites on the polyprotein could also contribute to drug resistance. Due to the flatness of the specificity landscape and high inter-connectedness of partially cleaved and fully cleaved clusters, novel sequences that are better substrates of drug-resistant variants may easily arise. Thus, considering both substrate and protease variation in evaluating and designing anti-viral therapies may be necessary. This mode of substrate coevolution-based drug resistance has been observed in HIV-1[225]. At the same time, our analysis of the dominant HCV sequences obtained from patients suggests that the protease substrate interactions may

also contribute to negative selection and help limit the acquisition of heterogeneity – the sequences of sites in the protease that are potentially cleavable upon acquisition of CA/CS at the P1-P1' junction (Figure 4.12) appear to be mutationally distant from doing so. Thus, the protease-substrate interaction landscape reveals that the balance between mutational robustness, negative selection and adaptive potential to environmental changes may be necessary to consider for understanding and therapeutic interventions.

In summary, our exploration of a viral molecular specificity landscape uncovers novel specificities for the HCV NS3/4A protease and data provides a biophysical basis for the mutational robustness observed for a key interaction required in HCV propagation. Given the widespread prevalence of HCV, insights obtained here may help in better understanding, and tackling the evolutionary trajectories of this ever-changing virus. The developed specificity landscape enumeration approach is general, and combining experimental deep sequencing and Rosetta-based structural modeling at a matching high throughput, followed by statistical machine learning, may be useful for elucidating a significantly larger space of sequence-function relationships for a variety of other systems.

Chapter 5. A Pareto-optimal approach for structure evaluation using Amber and Rosetta energy functions.

5.1. Abstract

An accurate energy function is an essential component of biomolecular structural modeling and design. The comparison of differently derived energy functions enables analysis of the strengths and weaknesses of each energy function, and provides independent benchmarks for evaluating improvements within a given energy function. We compared the molecular mechanics Amber empirical energy function to two versions of the Rosetta energy function (talaris2014 and REF2015) in decoy discrimination and loop modeling tests. Both Rosetta's talaris2014 and Amber's ff14SBonlySC energy functions performed well in scoring the native state as the lowest energy conformation in many cases. In 24/150 cases with Rosetta, and in 2/150 cases using Amber, a false minimum is found that is absent in the alternative landscape. In 21/150 cases, both energy function-generated landscapes featured false minima. The newest version of the Rosetta energy function, REF2015, which has more physically-derived terms than talaris2014, performs significantly better, highlighting the improvements made to the Rosetta scoring approach. To take advantage of the semi-orthogonal nature of these energy functions, we developed a Pareto optimization approach that combines Amber and Rosetta energy landscapes to predict the most near-native model for a given protein. This algorithm improves upon predictions from either energy function in isolation, and should aid in model selection for structure prediction and loop modeling tasks.

5.2. Introduction

Computational protein structure prediction is dependent on an accurate energy function. The native state of a protein is expected to be found uniquely at the minimum of the energy function[226]; therefore, the energy function must robustly discriminate between native and non-native conformations. A variety of energy functions to predict protein structure have been implemented over the past forty years[227–233]. These potentials largely fall into one of two categories: molecular mechanics force fields that rely on the combination of various empirical potentials such as Lennard-Jones, torsional energies, Coulombic interactions, and desolvation penalties[228,229,232] and statistical or knowledge-based potentials that depend on characteristics of known protein structures[227,230,231]. While molecular mechanics force-fields are generally parameterized on small molecule properties[232,234–236], statistical potential parameter optimization is often guided by known biomolecular structures[117,237,238]. Each approach has its own drawback: since parameters in physically derived force-fields are fit based on small molecule properties, they may not be suited to macromolecules[239,240]: for example, force-fields will often display biases towards certain secondary structure propensities[239,241]. On the other hand, statistical potentials are trained on specific datasets of large biomolecules, and data sparseness may lead to overfitting[118].

The Rosetta macromolecular modeling program energy function combines elements of both categories; it contains physical force-field terms (Lennard-Jones interactions, electrostatic interactions, desolvation penalties, etc.) and statistical potentials (probability of amino acid identity given backbone angles, probability of backbone angles given amino acid identity, probability of backbone-dependent rotamer, etc.)[242]. The most recent

Rosetta energy function (REF2015) is parameterized on both small molecule properties and large sets of biomolecular structures[243], although previous energy functions were generally parameterized on known biomolecular structures alone[238]. While efforts have been made to compare the performance of various empirical force-fields[241,244,245], little attention has been focused on the comparison between the Rosetta energy function and empirical force-fields.

The Amber ff14SBonlySC force field[235] uses a standard fixed-charge molecular mechanics potential, with torsion potentials based entirely on fits to quantum chemistry data. It is very like the more commonly-used ff14SB protein force field, but does not include the empirical modifications to backbone torsion potentials that are present in ff14SB, and which provide an improved balance of secondary structure in explicit solvent simulations. Hence, ff14SBonlySC is more "physics-based" than is ff14SB, and it arguably better suited for the implicit solvent simulations used here, since the empirical backbone torsional potentials in ff14SB might be specific to its use of explicit solvent simulations. The ff14SBonlySC force field, in combination with a generalized Born implicit solvent model[246], has been shown to fold a variety of single-domain proteins using unrestrained molecular dynamics simulations[247].

Comparing the Amber force-field and Rosetta energy function performance at structure prediction elucidates the strengths and areas of improvements for each energy function. As Rosetta energy functions have been developed based on improving performance for certain modeling datasets, testing their performance on the same macromolecular datasets

may result in overfitting of the Rosetta energy function, while comparing their performance to that of a physics-based Amber energy function is a relatively unbiased comparison for evaluating performance improvements. Finally, selecting a correct near-native model for a given sequence is an elementary challenge; the Pareto-optimal combination of these two semi-orthogonal energy functions provides a method for model selection that is able to select more accurate models.

5.3. Results

5.3.1. Performance of Amber and Rosetta energy functions in discriminating between native and non-native structures

Protein free energy landscapes involve folding funnels[248–250] which enable the folding chain to efficiently find the native state[226,248], and their existence implies that the higher energy of non-native (decoy) structures compared to the native (e.g., crystallographically-determined) structure drives protein folding. Therefore, a common test used for evaluating[247,251] and improving[118] energy functions is the decoy discrimination test, in which the evaluated scores of decoy structures are compared to that of near-native structures. High-RMSD decoys which have comparable energies to near-native structures are classified as "false minima", and are indicative of inaccuracies in the energy function (Figure 5.1A-C, black points). The *B* metric[118], ranging from 0 to 1, quantifies the existence of false minima in a set of structures upon evaluation with a given energy function, with values close to 1 indicating a smooth folding funnel with no false minima. Conversely, a lower *B* value indicates that one or more false minima exist.

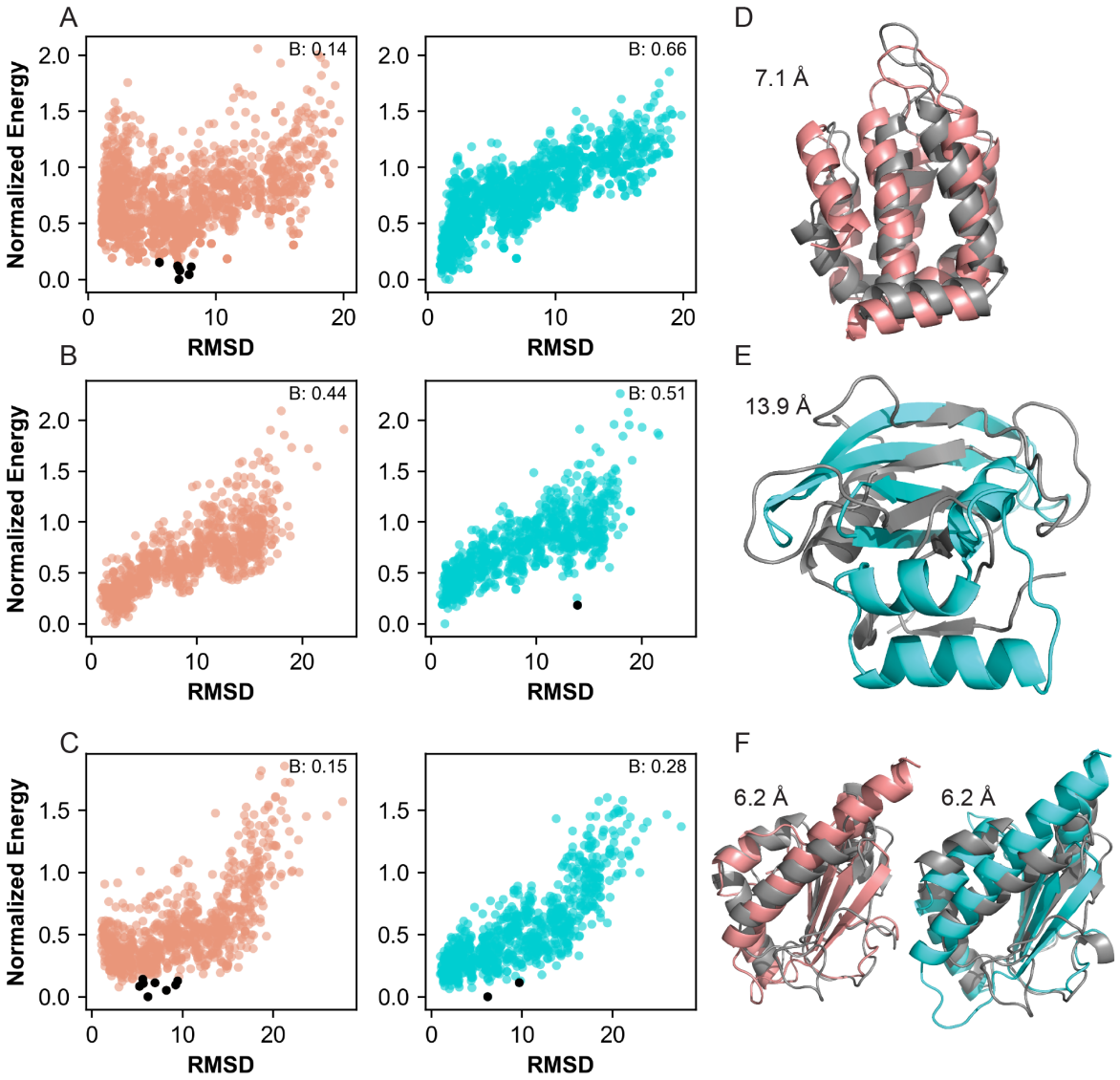


Figure 5.1. Energy landscape examples for cases of false minima.
(A-C) Energy landscapes for 2QY7, 1T2I, and 1SEN respectively. Each dot on the plot represents one decoy conformation. The x-axis is RMSD from native and the y-axis is normalized energy. False minima (defined as decoys within top 10 energies but with RMSD > 5.0 Å) are depicted in black. The *B* metric, which represents the efficacy of the score-function at differentiating between native and non-native decoys, is shown at the top right corner of each plot. Rosetta plots are to the left, in salmon, and Amber plots are to the right, in turquoise. (D-F) Superimposed native (gray) and Rosetta lowest-ranking false minimum decoy (salmon) and/or superimposed native (gray) and Amber lowest-ranking false minimum decoy (turquoise) for 2QY7, 1T2I, and 1SEN respectively.

We compared the performance of the Amber energy function and Rosetta energy function at ranking native state structures lower than decoy conformations for a set of 150 proteins. Amber ff14SBonlySC generally performed better than Rosetta talaris2014, scoring significantly higher *B* metrics for many systems (Figure 5.2A). We also compared Amber to the newer default Rosetta energy function, REF2015[242], and found that while Amber did have a higher *B* metric for several systems, several other systems had a higher *B* metric when scored by REF2015, thus showing the improvement of REF2015 over talaris2014 when compared to Amber as an unbiased benchmark. Nonetheless, the comparative performance of the two energy functions (Amber and REF2015; Figure 5.2B) shows that each has its strengths and limitations (Table 5.1). Our analysis was carried out with the talaris2014 energy function, and we refer to it as the Rosetta energy function in the remainder of this paper.

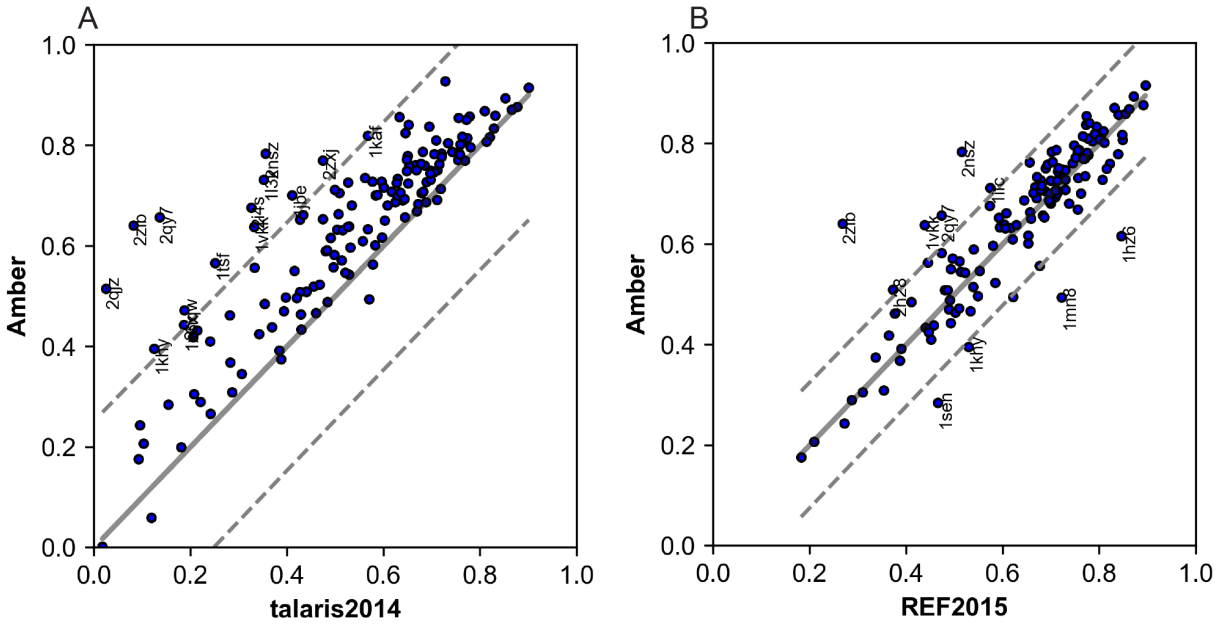


Figure 5.2. Comparison of Rosetta and Amber performance.
Scatterplots to depict general performance of Rosetta talaris2014 scoring function vs. Amber scoring function (A) and Rosetta REF2015 scoring function vs. Amber scoring

function (B) over entire decoy discrimination set. Each dot represents the *B* metric for one system. The black line is x=y and the dashed line represents the 95% prediction interval. Any points that lie outside the 95% prediction interval are annotated with the PDB ID of that system.

We examined cases in which either Amber, Rosetta, or both were unable to correctly rank high-RMSD decoy conformations, scoring them as low-scoring instead of high-scoring. A false minimum is defined as a decoy within the top-10 ranked decoys that has a C-α RMSD from native of greater than 5 Å. Three of these cases are shown in Figure 5.1. 2QY7 (Figure 5.1A, D) has several false minima for Rosetta but none for Amber. Generally, Rosetta alone had at least one false minimum in 16% of structures. 1T2I (Figure 5.1B, E) has a false minimum for Amber but none for Rosetta; 1.32% of systems have at least one false minimum for Amber alone. 1SEN (Figure 5.1C, F) has false minima for both Amber and Rosetta, as do 14% of overall structures (Table 5.1).

Table 5.1. *B* metric, false minima, and Pareto summary comparisons for Amber ff14SBonlySC, Rosetta talaris2014, and Rosetta REF2015 energy functions.

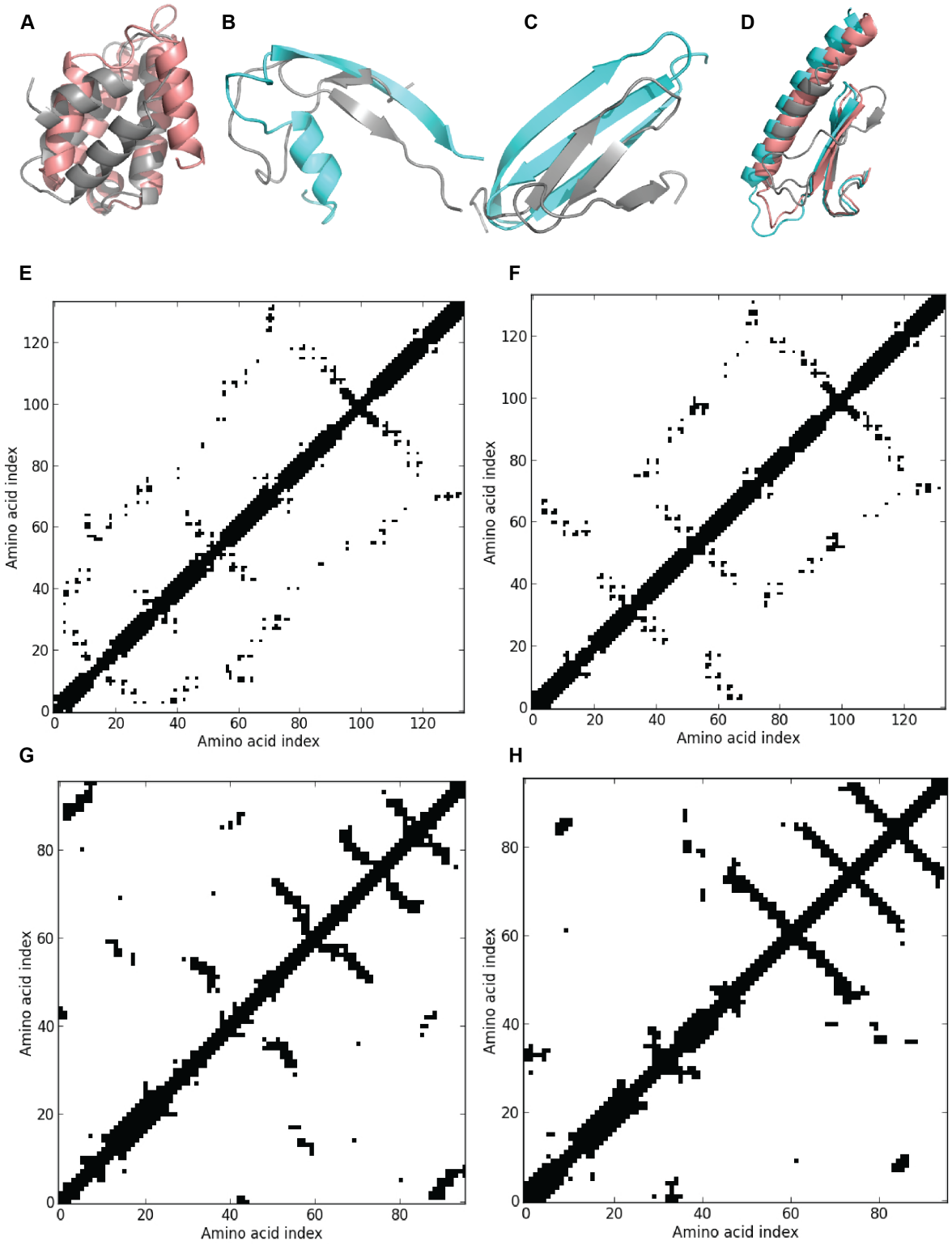
	No. Cases/Total No. Proteins
Decoy Discrimination	
ff14SBonlySC B > talaris2014 B by 0.1	54/150
talaris2014 B > ff14SBonlySC B by 0.1	0/150
ff14SBonlySC B > REF2015 B by 0.1	6/140
REF2015 B > ff14SBonlySC B by 0.1	9/140
False minima in ff14SBonlySC only (not talaris2014)	2/150
False minima in talaris2014 only (not ff14SBonlySC)	24/150
False minima in ff14SBonlySC and talaris2014	21/150
False minima in REF2015 only (not ff14SBonlySC)	0/140
False minima in ff14SBonlySC and REF2015	10/140

Pareto selected RMSD < ff14SBonlySC selected RMSD by 1 Å	10/150
Pareto selected RMSD < talaris2014 selected RMSD by 1 Å	21/150
Pareto selected RMSD < ff14SBonlySC selected and talaris2014 RMSD by 1 Å	1/150
Loop Modeling	
ff14SBonlySC B > talaris2014 B	15/39
talaris2014 B > ff14SBonlySC B	7/39

Superimpositions of false minima decoys with native decoys show their distinct non-native conformations involving both misprediction of secondary structure elements as well as their incorrect relative placement in tertiary structures. In the case of 2QY7, a Rosetta false minimum, the four-helical bundle found in the native structure is perturbed in the false minimum, as the order of the first two helices is reversed; thus, they do not contact the other two helices as tightly as that of the native structure (Figure 5.3E, I-J). The difference between the native structure of 1T2I and its Amber false minimum is subtler. While the contact maps for the native and false minimum conformations are similar, except for a small contact region in the native structure between residues 40 and 59 that does not appear in the false minimum (Figure 5.3K-L), the false minimum is slightly more compact and has a more ordered secondary structure. Two beta sheet regions in the false minimum are beta strands/unordered in the native structure and two alpha helices in the false minimum are beta strands in the native structure (Figure 5.3F-G).

The case of 1SEN, which has false minima for both Rosetta and Amber, is like 1T2I in that the false minima are more ordered than the native structure, although the native

structure forms more contacts than do the false minima (Figure 5.3A-D). Residues 85-96 form a tight beta hairpin in the false minima, whereas the native residues 85-96 has a longer loop between the beta strands, resulting in a shorter, less tight, beta hairpin. Additionally, residues 94-109 in the native are entirely disordered, while that of the false minimum begins as a beta strand and ends in an alpha helix (Figure 5.3H). Decoys that are predicted as false minima often have the same overall structure and contact maps as native structures, yet secondary structure differences may result in large structural deviation. In some cases, false minima contain more ordered secondary structures yet fewer contacts than native conformations; the propensity away from disordered loops may result in lower energies for these false minima.


A
B
C
D
E
F
G
H
Amino acid index
Amino acid index

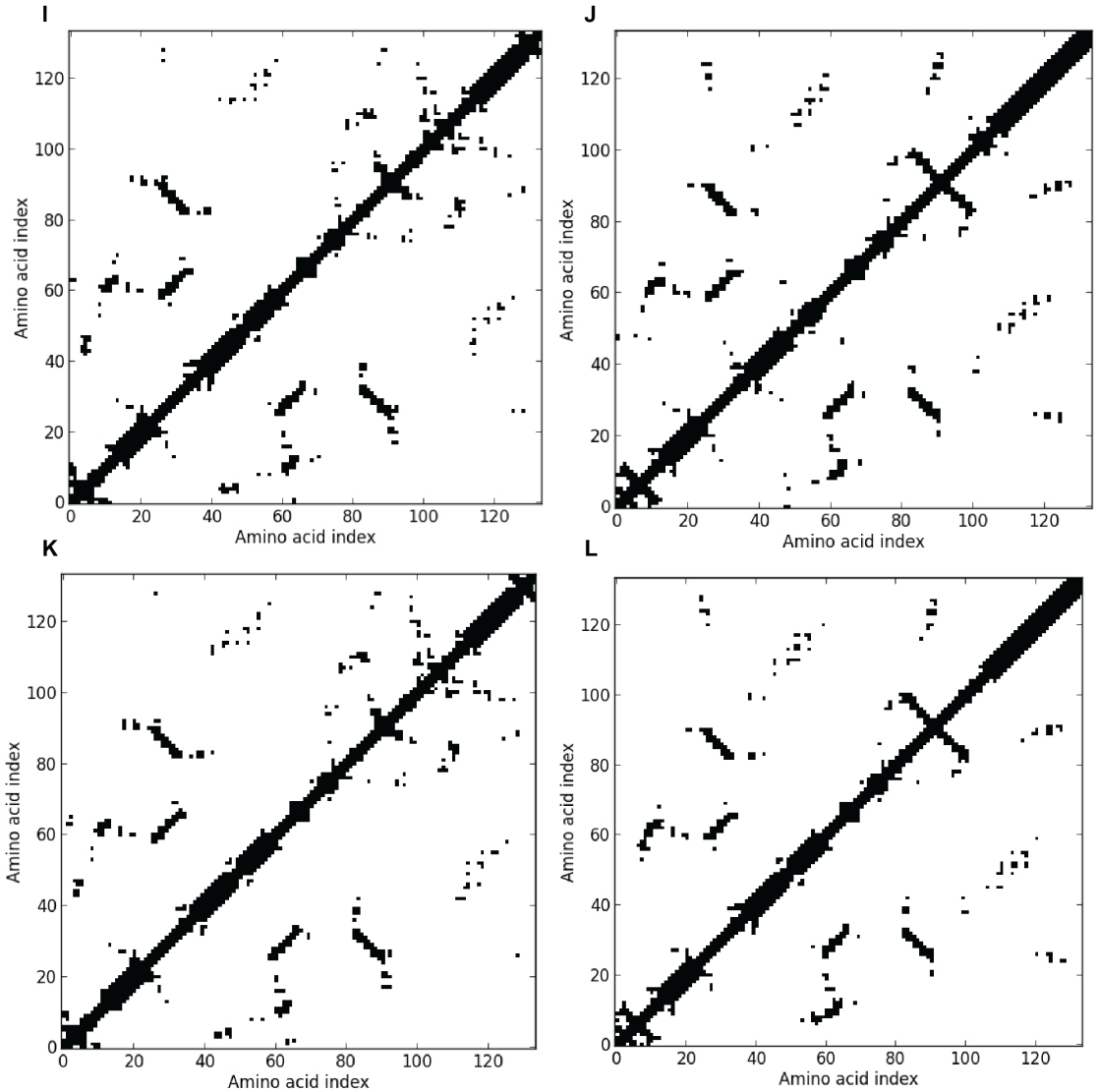


Figure 5.3. Structural analysis of false minima.
(A) Residues 1-81 of 2QY7. Gray is native, salmon is Rosetta false minimum. (B-C) Residues 31-59 and 61-96 of 1T2I respectively. Gray is native, turquoise is Amber false minimum. (D) Residues 75-135 of 1SEN. Gray is native, salmon is Rosetta false minimum, and turquoise is Amber false minimum. (E) Contact map for 2QY7 lowest-Rosetta scored native structure. (F) Contact map for 2QY7 Rosetta false minimum. (G) Contact map for 1T2I lowest-Amber scored native structure. (H) Contact map for 1T2I Amber false minimum. (I) Contact map for 1SEN lowest-Rosetta scored native structure. (J) Contact map for 1SEN Rosetta false minimum. (K) Contact map for 1SEN lowest-Amber scored native structure. (L) Contact map for 1SEN Amber false minimum. Contact maps were generated using the Protein Contact Maps tool at nanoHUB.org[252].

5.3.2. Per-residue Rosetta energy decomposition

To investigate whether certain residues, structural elements, or energy terms contribute more to false minima conformations, we analyzed the per-residue score decomposition for Rosetta scores for the three systems outlined above (2QY7, 1T2I, and 1SEN). We were unable to perform the same decomposition for Amber as the GB solvation term is not pairwise-decomposable[253]. We calculated the Z-scores for each residue over the lowest-scoring native and false minimum conformations. We identified residues as possibly implicated in false minima if the false minimum residue Z-score score was lower than the native residue Z-score by at least one (i.e. the distance between the two was greater than one standard deviation). We have highlighted these residues (Figure 5.4A-C). False minima contributing residues were distributed over the conformations and did not cluster to any particular region. Moreover, false minima contributing residues were found in various types of secondary structure: alpha helices, beta strands, and loops. It is therefore currently not possible to attribute Rosetta false minima to any single per-residue propensity, but as expected, several small errors in energy estimation may lead to the observed incorrect scoring.

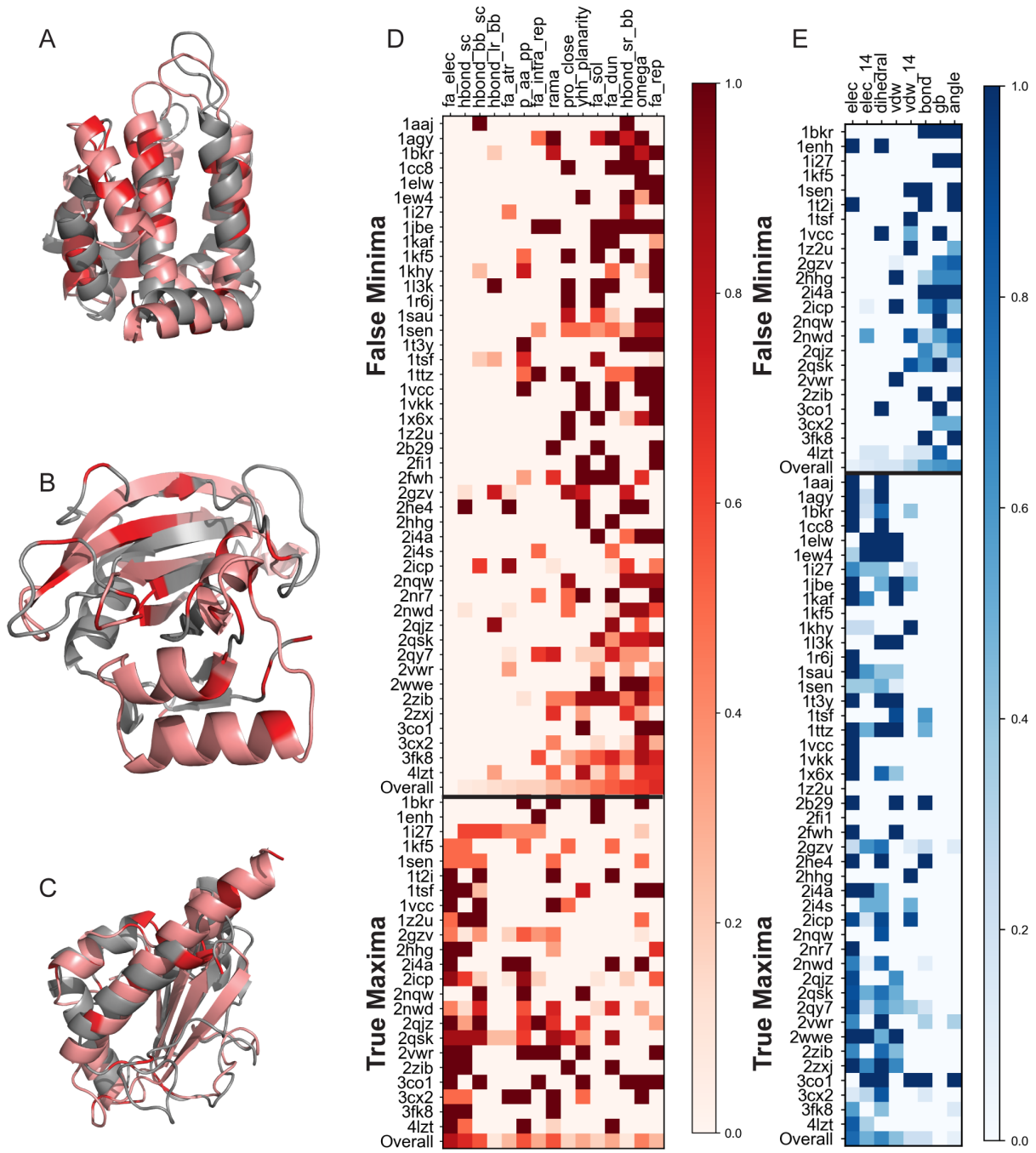


Figure 5.4. Per-residue and per-score-term propensity of score-functions toward false minima.
(A-C) Native (gray) and Rosetta-minimized (salmon) structures of 2QY7, 1T2I, and 1SEN respectively. Rosetta-minimized residues that are scored by Rosetta as greater than 1 standard deviation away from the corresponding native residue are highlighted in red. Heatmaps of per-structure, score-term contribution to Rosetta-determined (D) and Amber-determined (E) false minima and true maxima. The row marked Overall shows the percentage of structures that indicate some degree of implication for that score-term.

5.3.3. Per-scoreterm contributions of Amber and Rosetta

We reasoned that insight about the performance and pathologies of each energy function could be gained by identifying the energy terms that are responsible for correct and incorrect evaluations within the same energy function. For example, we asked which terms in the Amber energy function help it avoid mis-scoring a decoy (called Amber true maximum) that is identified as a false minimum in the Rosetta landscape (called Rosetta false minimum), and vice versa.

We identified terms that contribute to false minima and true maxima by calculating the Z-scores per decoy set and native set for each protein. If the lowest native score-term Z-score is greater than the false minimum score-term by at least one, that term is implicated in that false minimum. The reverse (i.e. true maximum score-term Z-score is greater than the lowest native score-term Z-score by at least one) is true for identifying true maximum contributing score-terms. The heatmap in Figure 5.4D depicts the fraction of Rosetta false minima decoys (top) and true maxima decoys (bottom) that show some degree of implication for each score-term. This is calculated both on a per-protein basis and over the entire false minima/true maxima sets. Several score-terms, including hbond_sr_bb, fa_dun, fa_rep and omega, are implicated in a majority of false minima in the Rosetta talaris2014 energy function. A set of other score-terms contribute to a majority of Rosetta true maxima (or Amber false minima), including rama, hbond_bb_sc, hbond_sc, p_aa_pp, and fa_elec. These are score-terms that are not usually implicated in Rosetta false minima, thus demonstrating that the score-terms that contribute to the two trends (towards false minima and true maxima) are mutually exclusive. Except fa_elec, the other

terms identified as helping "rescue" Amber false minima are all PDB-statistics derived, and it is not surprising that they are implicated in correcting the errors of the more physics-based Amber energy function.

We next performed a similar analysis on Amber score-terms for both Amber false minima and Amber true maxima (Figure 5.4E). We found that bond, angle, and gb are responsible for more than 50% of Amber false minima and that dihedral and elec are implicated in rescuing Rosetta false minima (Amber true maxima). We found that score-terms that are responsible for false minima are not implicated in true maxima and vice versa. Similar to the identification of statistically-derived terms in Rosetta as being responsible for correctly scoring Amber false minima, we find that physics-based terms, i.e., elec (which is counterbalanced by gb) and dihedral potentials, that are orthogonal to the talaris2014 Rosetta energy function, are implicated in the rescue of Rosetta false minima by Amber.

5.3.4. Pareto-selected decoys improve decoy selection

Based on the results above indicating that the rescue of false minima in the landscape generated by one energy function can be effected using the other energy function due to additional terms or different parameterization of terms, we sought to develop an approach to productively combine the two landscapes for model selection. In model selection (for example in protein structure prediction) the challenge is to select a near-native conformation from a set of decoy conformations based on one or more energy values or other features. Typically, an energy value obtained from a single energy function is used.

In the current benchmark set, if model selection is performed by the Rosetta and Amber energy functions individually, the Rosetta lowest-scored decoy has an RMSD of > 5.0 Å for thirteen out of 150 systems, while the lowest-scored Amber decoy has an RMSD of > 5.0 Å for seven systems (four of which overlap with the aforementioned Rosetta systems). We designed a Pareto optimization-based algorithm (see Methods) to select a decoy conformation based on both sets of ranks to improve the chances of selecting a near-native decoy.

We found that our Pareto algorithm improved model selection for both Rosetta and Amber rankings (Figure 5.5D-E, Figure 5.6), although it improved model selection for Rosetta to a greater extent. The Pareto-selected decoy had a lower RMSD than the lowest-scoring Rosetta decoy by at least 1 Å for ten out of the thirteen cases mentioned above and a lower RMSD than the lowest-scoring Amber decoy by at least 1 Å for four out of the seven cases mentioned above. More generally, the Pareto-selected decoy had a lower RMSD than the lowest-scoring Rosetta decoy for 22 out of 150 cases and a lower RMSD than the lowest-scoring Amber decoy for 11 out of 150 cases.

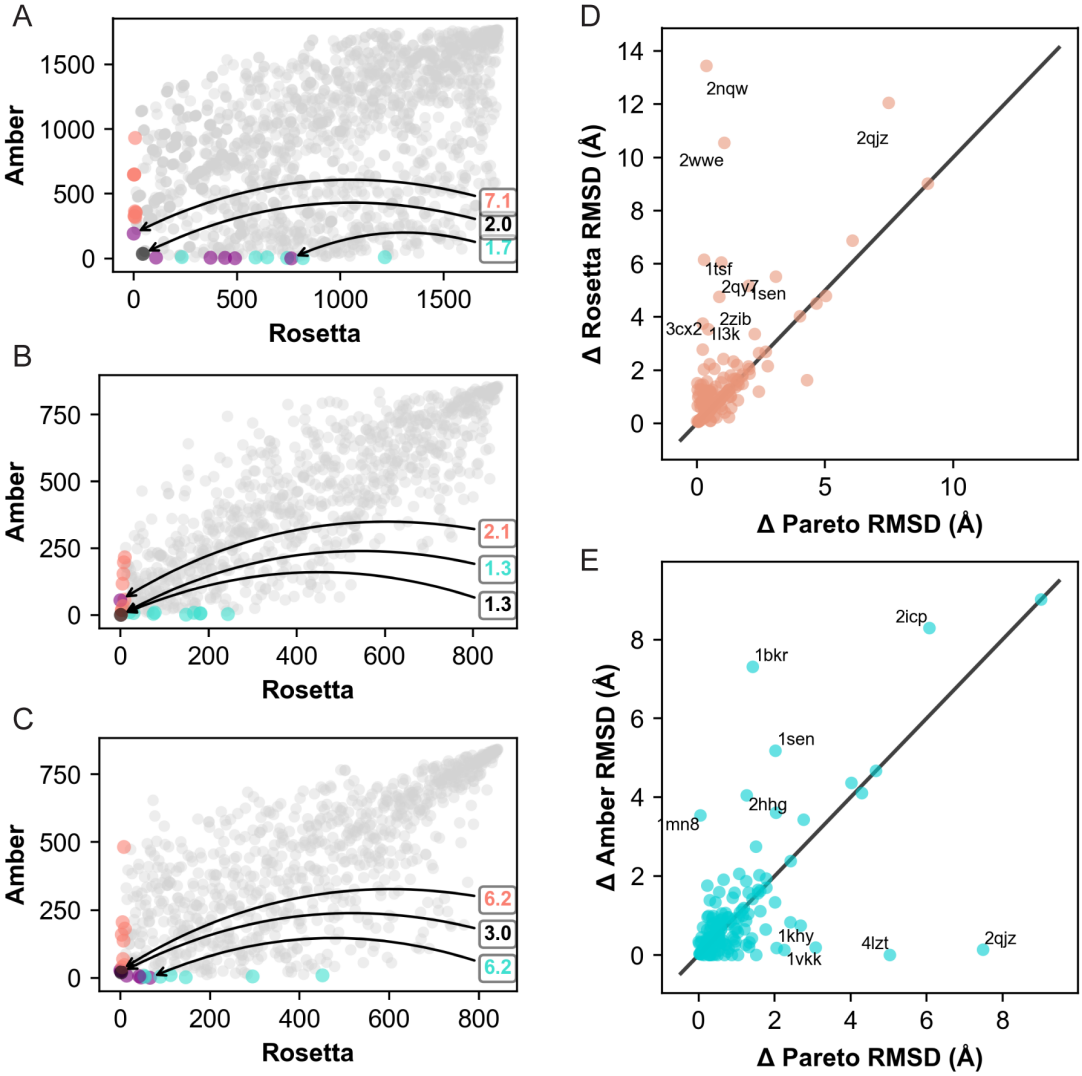


Figure 5.5. Pareto-optimal decoys.
(A-C) Scatterplots of Rosetta-rank vs. Amber-rank for all decoys of 2QY7, 1T2I, and 1SEN respectively. Each point represents one decoy conformation. The set of Pareto solutions is purple, the top-10 ranked Amber decoys are turquoise, the top-10 ranked Rosetta decoys are salmon, and the chosen Pareto solution is black. Annotations represent the RMSD in Å from native for the top-ranked Amber decoy (turquoise), top-ranked Rosetta decoy (salmon), and chosen Pareto solution (black). Scatterplots that show the efficacy of the Pareto solution at minimizing the distance from native relative to the top-ranked Rosetta decoy (D) and top-ranked Amber decoy (E). Each point represents one system. The x-axis is the difference between the Pareto solution RMSD from native and the RMSD of the minimal-RMSD decoy conformation, while the y-axis is the difference between the Rosetta lowest-ranked conformation (D) or Amber lowest-ranked conformation (E) RMSD from native and the RMSD of the minimal-RMSD decoy conformation. Points that fall outside the 95% prediction interval are annotated.

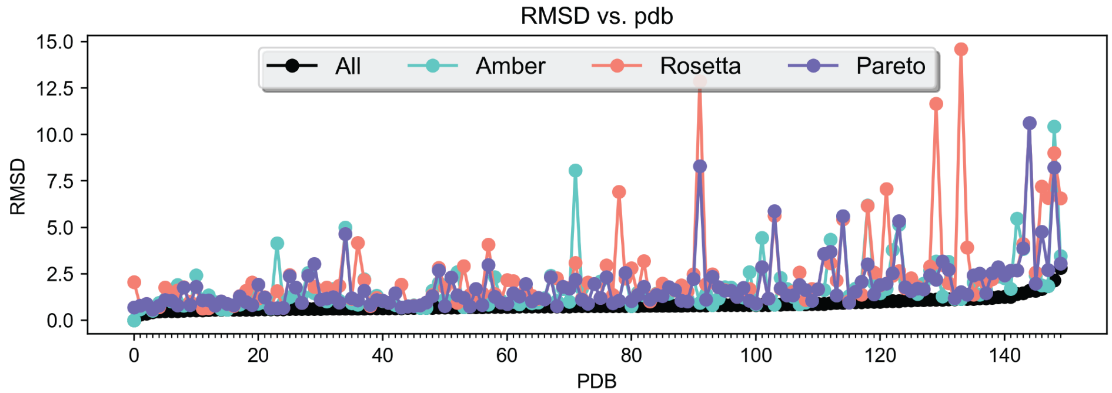


Figure 5.6. Plot of minimal (All), Pareto-selected, Rosetta-selected, and Amber-selected RMSD for each system.

We examined the false minima cases described above (2QY7, 1T2I, and 1SEN) and found that the Pareto-selected decoy generally had a lower RMSD than that of Rosetta- or Amber-selected decoys (Figure 5.5A-C). However, for 2QY7, which contains a false minimum for Rosetta but not for Amber, the Amber-selected decoy had a slightly lower RMSD than that of the Pareto-selected decoy (1.7 Å vs. 2.0 Å). Nevertheless, the Pareto-selected decoy RMSD is significantly lower than that of the Rosetta-selected decoy (2.0 Å vs. 7.1 Å). Thus, a Pareto optimality framework allows combining the two energy functions productively to select a near-native model.

5.3.5. Loop Modeling

The conformational variability of loops plays a multi-functional role in protein structure and function. They are implicated in stability and folding pathways[254], binding and active sites[255,256], and binding other proteins[257,258]. Efficient sampling algorithms have been developed[256–259], but loop structure prediction efforts can be limited by energy functions, as the energy gaps between loops are smaller and minima are narrow[260]. Therefore, we tested both Amber and Rosetta energy functions on a loop modeling benchmark obtained

from T. Kortemme and S. O'Connor. In this benchmark, most of the structure remains the same over the set of decoys; the difference lies in a small loop region, which can vary highly in RMSD. The energy gaps between structures are therefore smaller; thus, loop modeling provides a more stringent test to distinguish between energy functions.

We found that Amber ranked loops more accurately than did Rosetta (Figure 5.7C). Several systems had significantly higher *B* values with Amber than with Rosetta. Figure 5A depicts the energy landscapes for one of these structures (1TCA). The Amber funnel is steeper than that of Rosetta, which is reflected in its higher *B* (0.86 vs. 0.37). The lowest-energy and highest-energy loop conformations are shown for both Rosetta and Amber in Figure 5.7B. Both Rosetta and Amber rank the lowest-energy and highest-energy conformations correctly.

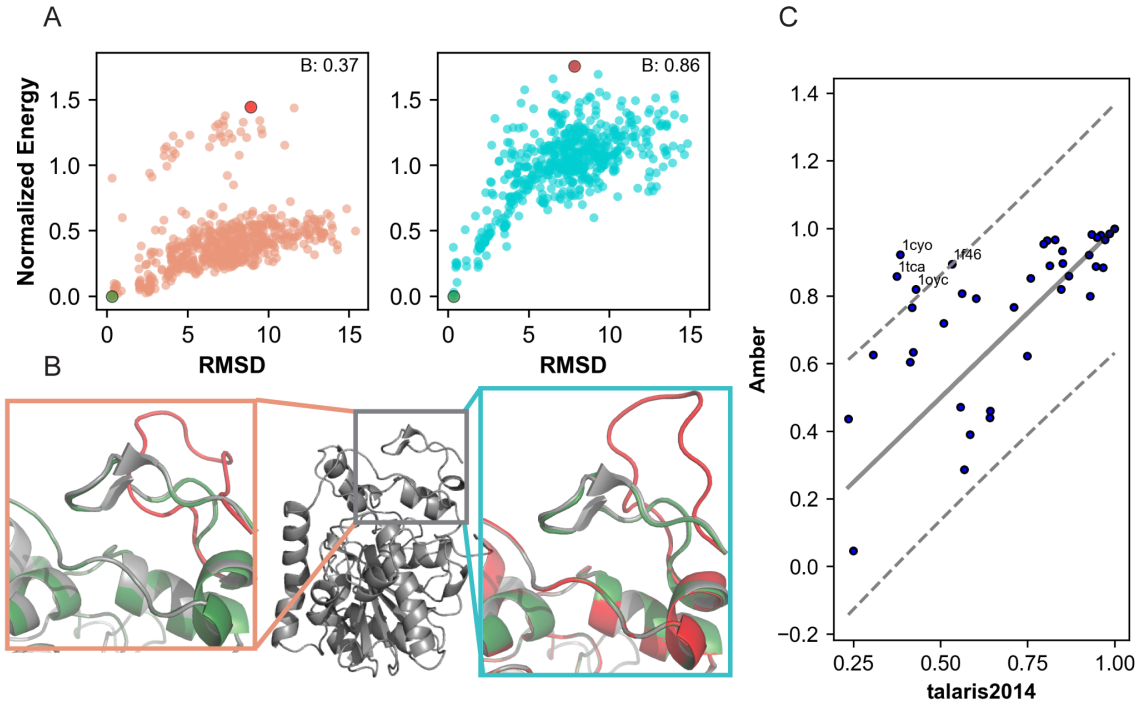


Figure 5.7. Loop modeling benchmark.

(A) Energy landscape for 1TCA. Each dot on the plot represents one decoy conformation. The x-axis is RMSD from native and the y-axis is normalized energy. The *B* metric, which represents the efficacy of the score-function at differentiating between native and non-native decoys, is shown at the top right corner of each plot. Rosetta plots are to the left, in salmon, and Amber plots are to the right, in turquoise. The lowest-energy decoy conformation in each plot is shown in green and the highest-energy decoy conformation is shown in red. (B) Native structure of 1TCA (gray) and close-ups of loop conformations for lowest-energy decoys (green) and highest-energy decoys (red) for Rosetta (salmon box) and Amber (turquoise box). (C) General performance of Rosetta talaris2014 scoring function vs. Amber scoring function over the entire loop modeling set. Each dot represents the *B* metric for one system. The black line is x=y and the dashed line represents the 95% prediction interval. Any points that lie outside the 95% prediction interval are annotated with the PDB ID of that system.

5.4. Discussion

It can be difficult to systematically compare energy functions derived by different methods. Systematic comparison of Amber ff14SBonlySC (a physically-derived energy function) and Rosetta talaris2014 (both physical and statistical based) reveals the strengths and weaknesses of each energy function. Generally, Amber ff14SBonlySC performs better than Rosetta talaris2014 at both decoy discrimination and loop modeling. However, comparison of Amber ff14SBonlySC to Rosetta REF2015 (the newer, default Rosetta energy function) reveals that REF2015, which has more physically-derived terms than talaris2014, performs comparably well to Amber ff14SBonlySC. Examination of Rosetta talaris2014 score-terms that rescue Amber ff14SBonlySC false minima and Amber ff14SBonlySC score-terms that correct Rosetta talaris2014 false minima reveals two possible sources for the performance improvement of REF2015. While two of the Rosetta score-terms and two of the Amber score-terms that contribute to the correction of false minima are counterparts to each other (Amber dihedral and Rosetta rama, and Amber elec and Rosetta fa_elec), subtle nuances in their derivation and parameterization appear to influence the propensity of each energy function toward false minima.

Although rama and dihedral both score the propensity of the backbone dihedral angles, rama does so in a statistically-derived manner while dihedral is based on fits to quantum chemistry data. Both elec and fa_elec are derived from a Coulombic model, yet they are differently parameterized; the Amber elec is parameterized *via* small-molecule properties, whereas fa_elec is optimized on larger biomolecular structures. The improvement of Rosetta REF2015 over Rosetta talaris2014 may be caused by its greater inclusion of physical-derived terms (bond, angle, etc.) and/or its parameterization on both small-molecule properties and larger biomolecular structures.

Model selection, or the ability to select a near-native decoy from a set of decoy conformations is a general problem in protein structure prediction. If low-energy decoys exist in false minima in the energy landscape, it is difficult to identify conformations that are near-native. Since Amber and Rosetta provide different, semi-orthogonal information, Pareto-optimal solutions enable the identification of near-native decoys. The Pareto-based algorithm that we have implemented improves model selection for 15% of structures over Rosetta model selection and 7.3% of structures over Amber model selection. The model selection algorithm is extensible to any two sets of energy functions or model ranks for one set of models and can thus be used to combine any two sources of information to produce meaningful improvements in near-native decoy selection.

The approach described here should enable comparative analysis and combination of future versions of both Amber and Rosetta scoring functions, and enable a variety of biomolecular modeling tasks.

5.5. Methods

5.5.1. Benchmark Sets

5.5.1.1. Structure Prediction

To evaluate and compare the performance of Rosetta and Amber energy functions, we used two benchmark sets, a structure prediction (decoy discrimination) set and a loop modeling set. The decoy discrimination benchmark set includes a total of 150 proteins, a combination of two independent decoy sets used in previous studies[118,261]. The proteins in the set are monomeric and have crystallographic native structures available in the RCSB PDB[262] with resolution < 2.0 Å. The protein lengths range from 50 to 200 residues and have a diverse range of topologies. The decoy sets were originally generated using biased and unbiased ab-initio sampling runs[251] followed by parallel loophash sampling (PLS)[263]. This produced 40,000-200,000 decoys per protein, ~1000 representative low-energy structures of which were chosen for each protein to cover the range of possible C-α RMSD values.

5.5.1.2. Loop Modeling

The loop modeling benchmark set consisted of the 45-PDB dataset for 12-residue loops in the monomeric protein loops training set of the 2016 Collaborative Assessment and Development of Rosetta Energetics and Sampling (CADRES). This loop modeling

benchmark set was obtained from Shane O'Connor and Tanja Kortemme (personal communication).

5.5.2. Structure Preparation

5.5.2.1. Rosetta

5.5.2.1.1. Structure Prediction

For each protein, the native structure was downloaded from the RSCB PDB and residues were trimmed from the structure to match the sequence of the crystal with the decoy structure in the benchmark sets. Native structures were necessary to evaluate RMSD from native for decoy conformations. Native structures were then relaxed using FastRelax[251] with the talaris2014[238] scorefunction to relieve any clashes. One hundred relaxation trajectories were simulated to generate one hundred relaxed native-like decoys. These native-like decoys were used for false minima analysis. Then, these one hundred native-like decoys, along with the ~1000 pre-sampled decoys, were subjected to backbone and sidechain minimization using talaris2014 and the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) minimizer implementation with inexact line search conditions (lbfgs_armijo_nonmonotone) over a maximum of 2000 iterations for convergence. C-α atom RMSD was calculated for all decoys.

The REF2015 dataset was obtained from F. DiMaio and H. Park[118]. For this dataset, each decoy was relaxed with 3 cycles torsion-space minimization and 2 cycles Cartesian mode[261] using the REF2015 energy function[242]. Only 140 out of 150 protein systems were included in this set due to the lower quality of experimentally determined structures

for 10 systems (H. Park and F. DiMaio, personal communication, July 5, 2017). Those 10 systems are ignored when comparing REF2015 to Amber.

5.5.2.1.2. Loop Modeling

The native crystal structures for each set were downloaded from the RCSB PDB and trimmed of excess residues that were not found in the decoy PDB structures. The backbone and sidechain geometries for residues in the loop region of each decoy structure were minimized in Rosetta using the talaris2014 scorefunction and the lbfgs_armijo_nonmonotone over a maximum of 2000 iterations for convergence. C-α RMSDs were calculated with respect to the crystal structure over loop residues only without fitting; since the protein scaffold was fixed during optimization, this statistic describes the extent of loop deviation.

5.5.2.2. Amber

5.5.2.2.1. Structure Prediction

Hydrogens were removed from the crystal structures and decoy PDBs, and initial structures were built using the tLEaP module of AmberTools[253] with the ff14SBonlySC[235] forcefield parameters. Minimizations were carried out for a maximum of 1000 steps under the LBFGS quasi-Newton algorithm[264] with a convergence criterion of 0.01 kcal/mol-A. Solvent effects were treated with a generalized Born implicit solvent model (GB-Neck2[246]) implemented in the Amber16[253] package with mbondi3 radii and a cutoff value of 999A for nonbonded interactions. Total potential energies of minimized structures and C-α RMSDs with respect to the crystal structure were obtained using the

pytraj 2.0.0 interactive molecular dynamics simulation data analysis Python package[265], which is a Python interface for cpptraj in AmberTools16[253].

5.5.2.2.2. Loop Modeling

Initial structures were obtained, prepared, and built as previously described in the Structure Prediction Benchmark set, but included the addition of positional restraints on all non-loop-residue atoms except for hydrogens with a force constant of 10.0 kcal mol-1 A-2. Minimization was performed and energies were gathered in a similar fashion to the Structure Prediction Benchmark, while C-α RMSDs were calculated over the loop residues only as in the Rosetta calculation. Loop residues are defined in Appendix 5.

Six sets of decoy structures were unable to be minimized in Amber due to missing residues, and those sets were not considered in subsequent analyses (1cb0, 1dts, 1m3s, 1ms9, 1t1d, and 2pia).

5.5.3. Energy Landscape Generation

Energy landscapes (RMSD vs. normalized energy scatterplots) were generated for all proteins for both Rosetta and Amber. The ideal shape of an energy landscape is that of a funnel (i.e. Figure 5.1A, turquoise plot) where the lowest-scoring decoy conformations are of near-native RMSD. We use the binned Boltzmann metric (see below) to evaluate the funnel shape of each energy landscape.

5.5.3.1. Energy Normalization

For each set of energies per scorefunction per protein, energies are normalized so that the gap between the 5th percentile of and the 95th percentile is equal to 1. This is accomplished via the following equation:

$$E_{i(norm)} = (E_i - E_{min})/(E_{95th} - E_{5th})$$

E_i refers to the raw energy of decoy i. E_{min} is the minimum energy value, E_{95th} is the 95th percentile energy, and E_{5th} is the 5th percentile energy.

5.5.3.2. Funnel Evaluation Metric

We use the binned Boltzmann metric, B, for energy landscape evaluation, as described previously[118]. This metric finds the Boltzmann probability of selecting native-like decoys over high-RMSD decoys based on their energy values. As in previous work[261], the metric is averaged over multiple thresholds for determining native-like status for each decoy.

$$B = \frac{\sum_j (\sum_i d_{ij} P_i / \sum_i P_i)}{N_j}$$

$$P_i = e^{-kTE_{i(norm)}}$$

The conformation index is i and j is the native threshold definition index. Cutoffs are 0.5, 1.0, 1.25, 1.5, 1.75, 2.0, 2.25, 2.5, 2.75, 3.0, 3.5, 4.0, 5.0, 6.0 Å and N_j is 14. $E_{i(norm)}$ is the score of decoy i as determined in Rosetta or Amber and normalized as described above. The value kT is the Boltzmann temperature of the ensemble and is set to 10. d_{ij} determines whether decoy i is considered native at threshold j; it is set to 1 if it is native and 0 if it is not. As the sum of the probabilities of the non-native-like conformations approaches 0, the numerator ($\sum_i d_{ij} P_i$) approaches the value of the denominator ($\sum_i P_i$), so that the value of B approaches to 1. As mentioned in Park et al.[118], the B metric is

better than the previously used *S* metric[261] due to a larger increase in the metric for a poor energy landscape vs. a good energy landscape than the increase from an already good energy landscape to a steeper energy landscape. Additionally, it is a smoother metric that is less affected by single-decoy outliers.

5.5.4. Pareto Optimization

For each protein, a Pareto solution was selected by finding the decoy with Pareto-optimal Amber and Rosetta ranks. First, the Amber scores and Rosetta scores were converted into ranks so that the rank of decoy *a* was less than the rank of decoy *b* if the energy of decoy *a* was less than the energy of decoy *b*. Second, the Pareto-minimal solutions are found as follows. Decoy *a* is defined as dominating decoy *b* if both ranks (Rosetta and Amber) of decoy *a* are <= both ranks of decoy *b*. Pareto-optimal decoys are decoys that dominate at least one other decoy and are not dominated by any decoys. From among the set of Pareto-optimal decoys, the decoy that has the lowest sum of ranks is chosen as the solution. In the rare case that more than one decoy has a minimal sum of ranks, a decoy is arbitrarily chosen from the minimal-sum-ranks decoys.

Chapter 6. Conclusion

6.1. Summary

Proteolytic cleavage is a crucial mechanism in normal cellular functioning. The ability to predict and manipulate protease specificity has important implications in understanding disease, synthetic biology, and drug design. We developed a structure-based predictive model of protease specificity that uses Rosetta and Amber force fields to classify substrates as cleaved or uncleaved. It is accurate, outperforms current machine learning approaches, and can be generalized to other proteases. Additionally, as we have demonstrated, it can be used to identify potential novel substrates.

Next, we implemented a mean-field structure-based algorithm (MFPred) to predict protease specificity profiles rapidly and accurately. MFPred is of equivalent or better accuracy and ~10-1000-fold faster than current computational specificity prediction methods. It is rapid enough to be used in each step of design and we demonstrate its ability to accurately predict the impact of protease mutations on substrate specificity without changing the substrate backbone ensemble.

Third, we constructed a comprehensive specificity landscape for HCV NS3 protease using a combination of experimental and computational techniques. The landscape provides insights into how mutational robustness may be encoded at a molecular level. The method that we use to construct the landscape can be generalized to other proteases as well.

Finally, we compare the performance of Rosetta and Amber energy functions. We find that while Amber ff14SBonlySC performs better than Rosetta talaris2014, Rosetta REF2015 performs comparably to Amber ff14SBonlySC. The parameterization of REF2015 on small-molecule properties, as well as the inclusion of more physical-derived terms within REF2015, may contribute to the performance improvement. We develop an algorithm to improve model selection by using semi-orthogonal information from both energy functions.

6.2. Strengths

The methods developed within this dissertation have several strengths over current related methods. One general advantage common to all structure-based approaches is that they are easily generalizable to new proteases. This is especially true in the case of our discriminatory scoring function, which can be extended to novel protease variants, unlike current pattern-recognition machine-learning techniques. However, while MFPred can be extended to new proteases, it does require several known substrates for accurate prediction. Also, the discriminatory scoring function captures interaction networks and pairwise correlations more accurately than do current methods. Third, it is not biased by the quality of the input data and is thus able to predict novel interactions that are not present in the training data.

The prediction performance of MFPred is equivalent to or better than existing structure-based methods and it is rapid enough to be used within design. The performance of MFPred is robust to the size of the flexible backbone ensemble; while other current

methods require a large backbone ensemble as a prerequisite to accurate specificity prediction, MFPred performs well on a small ensemble (<6 backbones). MFPred also predicts information content, or the shape of the specificity profile, more accurately than do other current methods, which is important when designing highly selective proteases.

Next, our approach to specificity landscape construction allows for the exploration of the full specificity landscape as opposed to a limited subset of sequence space. Again, this approach can be extended to other proteases. Additionally, insights into the mutational robustness of HCV NS3 protease may assist in understanding and preventing drug resistance.

Finally, our systematic comparison of Amber and Rosetta score-functions enables insights into the strengths and weaknesses of each energy function. Our model selection algorithm allows for a greater likelihood of selecting a native-like decoy from within a large decoy ensemble.

6.3. Limitations

Our structure-based approach to specificity prediction relies on two major, related suppositions. First, we assume that binding presumes cleavage and second, we assume that every candidate peptide has equal access to the protease active site. Both assumptions are not consistently true, especially for cases where a putative substrate is found within a folded protein.

Additionally, the structure-based approach is less accurate for proteases that are more flexible and/or contain loops near the active site, such as MMP2 and HIV-PR 1. Increased protease backbone sampling may therefore improve prediction accuracy. We may also be neglecting important electrostatic effects within the active site in the prediction of protease multispecificity.

A third general limitation of this approach is that we have tuned the free parameters for general good performance for all systems. While these parameters appear to be optimal for most systems in the case of the discriminatory scoring function, in the case of MFPred, the temperature parameter appears to be system-dependent and possibly sub-optimal for individual proteases.

6.4. Implications

We have developed a toolbox of techniques for computational protease design. Custom-designed proteases can be used to interrogate and intervene in biological processes. Current protease design approaches rely on directed evolution[40,44,65] *in vivo*, which proceed via incremental "generalist"[66] intermediates that display relaxed specificity, and are, therefore, toxic to cells. Structure-guided computational design, aided by the developed substrate classifier, specificity profile prediction, and specificity landscape elucidation should allow for multiple simultaneous substitutions to allow specificity switching without toxic intermediates. Combining structural computation with directed evolution should enable more efficient protease specificity design.

Appendix 1. Supplementary Methods for Chapter 2

The MMPBSA calculation includes the following steps:

1. Preparation of AMBER input .pdb files
2. Preparation of input parameter and topology files
3. MMPBSA Calculation

Description of each of the steps below:

In order to transform a pdb file into an AMBER readable format the hydrogens and virtual atoms are stripped. The subsequent file is loaded into AMBER using the following script using a tleap interface.

```
source leaprc.gaff
source leaprc.ff12SB
loadamberparams frcmod.ionsjc_tip3p
d$i = loadpdb "toload_$i.pdb"
addions d$i Cl- 0
charge d$i
saveamberparm d$i d$i.prmtop d$i.inpcrd
quit
```

The files saved as d$i.prmtop and d$i.inpcrd are inputs to the ante-MMPBSA.py program which generates the receptor-ligand, receptor only and ligand only topology files. An AMBER topology file is used to specify atom types, charges, etc. The inpcrd / input coordinate file is used to build the connections which forms the overall structure of the pdb.

```
ante-MMPBSA.py -p d$i.prmtop -c d_c$i.prmtop -s @Cl-
ante-MMPBSA.py -p d_c$i.prmtop -r d_r$i.prmtop -l d_l$i.prmtop -n : “residue range”
```

Residue range: specify the pose numbering of the peptide

The final step involves using the inpcrd and prmtop files to calculate the MMPBSA contribution of the complex. This is done by calculating the electrostatic energy of the peptide and protease separately as well as in a bound state

The following commandline is used for MMPBSA calculation

```
MMPBSA.py -O -i mmpbsa.in -o FINAL_RESULTS_MMPBSA.dat -sp d$i.prmtop -cp
d_c$i.prmtop -rp d_r$i.prmtop -lp d_l$i.prmtop -y *.inpcrd
```

For MMP2: The pdbs in these cases needed to be analyzed differently because of the presence of heteroatoms such as Zinc and Water that are involved in the active sites respectively.

The water is modeled using the TP5.lib and the following command is added to the prep script

Sample Scripts:

Sample xml for initial Relax:

```
<dock_design>
    <SCOREFXNS>
      <myscore weights=enzdes.wts/>
    </SCOREFXNS>
    <TASKOPERATIONS>
      <ProteinInterfaceDesign name=pido design_chain2=0 modify_after_jump=1/>
      <InitializeFromCommandline name=init/>
      <ReadResfile name=rrf filename="PATH TO RESFILE"/>
    </TASKOPERATIONS>

    <FILTERS>
    </FILTERS>

    <MOVERS>
     <AddOrRemoveMatchCsts name=cstadd cst_instruction=add_new/>
```

```
<FastRelax name=fastrelax scorefxn=myscore repeats=8 task_operations=pido,init>
<MoveMap name=mm>
        <Chain number=2 chi=1 bb=1/>
        <Chain number=1 chi=1 bb=1/>
        <Jump number =1 setting=1/>
</MoveMap>
</FastRelax>
<TaskAwareMinMover name =min_pro task_operations=rrf scorefxn=myscore
chi=1 bb=0 jump=0/>
<PackRotamersMover name=repack task_operations=rrf/>
<ConstraintSetMover name=protease_cst
cst_file="PATH_TO_PROTEASE_BACKBONE_HEAVY_ATOM_CONSTRAINT_FI
LE"/>

</MOVERS>
<APPLY_TO_POSE>
</APPLY_TO_POSE>

  <PROTOCOLS>
      <Add mover_name=protease_cst/>
      <Add mover_name=repack/>
      <Add mover_name=min_pro/>
      <Add mover_name=cstadd/>
```

```
        <Add mover_name=fastrelax/>
</PROTOCOLS>
</dock_design>
```

Command line:

```
~<PATH_TO_ROSETTA_BIN> rosetta_scripts.static.linuxgccrelease  -jd2:ntrials 1 -nstruct 20  -parser:protocol <PATH_TO_RELAX_XML> -database <PATH_TO_DATABASE> -out::prefix Job_${i}_ -s <PATH_TO_STARTING_PDB> -run:preserve_header -enzdes::cstfile <PATH_TO_CONSTRAINT_FILE>  -out:file:output_virtual @<PATH_TO_FLAGS_FILE>
```

Sample Script For Mutate, FastRelax, Scoring

```
#MUTATERUN
<PATH_TO_EXECUTABLE>/rosetta_scripts.static.linuxgccrelease -nstruct 10  -jd2:ntrials 1 -parser:protocol <PATH_TO_XML> -database <PATH_TO_DATABASE> -out::prefix $1_mut_ -s <PATH_TO_STARTING_PDB> -enzdes:cstfile <PATH_TO_CSTFILE> -run:preserve_header @<PATH_TO_FLAGSFILE> > design.log

find `pwd` -name "$1_mut_*00*pdb" > tlist

cp ~/Rosetta/main/database/scoring/weights/talaris2013 ./
```

```
#SCORINGRUN
~/Rosetta/main/source/bin/rosetta_scripts.static.linuxgccrelease  -jd2:ntrials 1 -
parser:protocol <PATH_TO_SCORING_XML> -database <PATH_TO_DATABASE> -
out::prefix Scores_ -l tlist -in:file:native <PATH_TO_STARTINGPDB> -
run:preserve_header @<PATH_TO_FLAGSFILE> -score:weights talaris2013 >
scoring.log

ls Scores_*.pdb > slist

#CSTRUN
~/Rosetta/main/source/bin/rosetta_scripts.static.linuxgccrelease  -jd2:ntrials 1 -
parser:protocol <PATH_TO_XML> -database ~/Rosetta/main/database/  -out::prefix
$1_cst_  -l tlist -enzdes:cstfile <PATH_TO_CSTFILE> -run:preserve_header
@<PATH_TO_FLAGSFILE> -jd2:enzdes_out > cst.log
```

Protease Mutate:

```
<dock_design>
    <SCOREFXNS>
            <myscore weights=enzdes.wts/>
    </SCOREFXNS>
    <TASKOPERATIONS>
      <ProteinInterfaceDesign name=pido design_chain2=0 modify_after_jump=0/>
```

```
    <InitializeFromCommandline name=init/>
    <ReadResfile name=rrf filename="PATH_TO_RESFILE"/>
  </TASKOPERATIONS>

  <FILTERS>
  </FILTERS>

  <MOVERS>
    <MutateResidue name=mut1 target=Res#1 new_res=DM1/>
    <MutateResidue name=mut2 target= Res#2 new_res=DM2/>
    <MutateResidue name=mut3 target= Res#3new_res=DM3/>
    <MutateResidue name=mut4 target= Res#4 new_res=DM4/>
    <MutateResidue name=mut5 target= Res#5 new_res=DM5/>
    <MutateResidue name=mut6 target= Res#6 new_res=DM6/>
<AddOrRemoveMatchCsts name=cstadd cst_instruction=add_new/>
    <FastRelax name=fastrelax scorefxn=myscore repeats=8 task_operations=pido,init>
    <MoveMap name=mm>
            <Chain number=2 chi=1 bb=1/>
            <Chain number=1 chi=1 bb=0/>
            <Jump number =1 setting=1/>
    </MoveMap>
    </FastRelax>
    <TaskAwareMinMover name =min_pro task_operations=rrf chi=1 bb=0 jump=0/>
```

```
    <PackRotamersMover name=repack task_operations=rrf/>

</MOVERS>
<APPLY_TO_POSE>
</APPLY_TO_POSE>

  <PROTOCOLS>
      <Add mover_name=mut1/>
      <Add mover_name=mut2/>
      <Add mover_name=mut3/>
      <Add mover_name=mut4/>
      <Add mover_name=mut5/>
      <Add mover_name=mut6/>
     <Add mover_name=repack/>
      <Add mover_name=cstadd/>
      <Add mover_name=fastrelax/>
</PROTOCOLS>
</dock_design>
```

SCORING XML

CST XML

```
<dock_design>
```

```
<SCOREFXNS>
        <myscore weights=enzdes.wts/>
</SCOREFXNS>
<TASKOPERATIONS>
  <InitializeFromCommandline name=init/>
</TASKOPERATIONS>

<FILTERS>
  <EnzScore name="cstenergy" scorefxn=myscore whole_pose=1 score_type=cstE
energy_cutoff=99999.0/>
</FILTERS>

<MOVERS>
  <AddOrRemoveMatchCsts name=cstadd cst_instruction=add_new/>
</MOVERS>

<APPLY_TO_POSE>
</APPLY_TO_POSE>

<PROTOCOLS>
  <Add mover_name=cstadd/>
  <Add filter_name=cstenergy/>
</PROTOCOLS>
```

</dock_design>

AMBER MMPBSA

```
cat >tleap.in <<EOF
source leaprc.gaff
source leaprc.ff12SB_manasi
loadamberparams frcmod.ionsjc_tip3p
loadamberparams frcmod.ionslrcm_hfe_tip3p
d$i = loadpdb "toload_$i.pdb"
charge d$i
saveamberparm d$i d$i.prmtop d$i.inpcrd
quit
EOF
tleap -f tleap.in

ante-MMPBSA.py -p d$i.prmtop -c d_c$i.prmtop -s @Cl-
ante-MMPBSA.py -p d_c$i.prmtop -r d_r$i.prmtop -l d_l$i.prmtop -n :199-208

MMPBSA.py -O -i mmpbsa.in -o FINAL_RESULTS_MMPBSA.dat -cp d_c$i.prmtop -
rp d_r$i.prmtop -lp d_l$i.prmtop -y d$i.inpcrd
```

MATLAB

```
function [test, testlab, ttcleaved, to, ts, train, trainlab,  a, f, X, Y, T, AUC, AUCav, Std,
Performanceav,Stdp] = coduh(A, LABELS, cleaved, uncleaved, boxconstraint, rbfsigma)

clearvars -except A LABELS cleaved uncleaved boxconstraint rbfsigma TABLE

X = [];
Y = [];
T = [];
AUC = [];


 [numberofelements len] = size(A);
tic

for s = 1:1000

    zcleaved = ceil(0.2%*cleaved);
    zuncleaved = ceil(0.2%*uncleaved);
    ttcleaved = randperm(cleaved,zcleaved);
%generatingRandomFromNumLength
    ttuncleaved = randperm((numberofelements - cleaved), zuncleaved) + cleaved;
```

```
    t = vertcat(ttcleaved',ttuncleaved');
    to(:,s) = vertcat(ttcleaved',ttuncleaved');
    ts(s) = length(t);
    z = zcleaved + zuncleaved;

test(:,:,s) = A(t,:);
testlab(:,:,s) = LABELS(t,:);

    x = numberofelements - z;
    train(:,:,s) = zeros(x, len);
    trainlab(:,:,s)= cell(x,1);

    clear n1;
    n1 = 1;

    for i = 1:numberofelements

        if i ~= t(:)
            train(n1,:,s) = A(i,:);
            trainlab(n1,s) = LABELS(i);
            n1 = n1 + 1;
        end
```

```
    end



svmrbf =[];

svmrbf=svmtrain(train(:,:,s), trainlab(:,s), 'kernel_function', 'rbf', 'boxconstraint',

boxconstraint, 'rbf_sigma', rbfsigma);



%%TEST%%

V = svmclassify(svmrbf,test(:,:,s));

result = transpose(V);

a(:,s)=transpose(result);


shift = svmrbf.ScaleData.shift;

scale = svmrbf.ScaleData.scaleFactor;

Xnew = bsxfun(@plus,test(:,:,s),shift);

Xnew = bsxfun(@times,Xnew,scale);

sv = svmrbf.SupportVectors;

alphaHat = svmrbf.Alpha;

bias = svmrbf.Bias;

kfun = svmrbf.KernelFunction;

kfunargs = svmrbf.KernelFunctionArgs;

f(:,s) = kfun(sv,Xnew,kfunargs{:})'*alphaHat(:) + bias;
```

```
[X(:,s),Y(:,s),T(:,s),AUC(s)] = perfcurve(testlab(:,:,s), f(:,s) ,'CLEAVED', 'Xcrit','reca',
'YCrit', 'prec' );

AUCav = mean(AUC);
Std = std(AUC);

%ACCURACY
tf(:,s) = strcmp (a(:,s), testlab(:,s));
Performance(s) = sum(tf(:,s)) / numel(a(:,s));
Performanceav = mean(Performance);
Stdp = std(Performance);

% %TRAIN
Vtrain = svmclassify(svmrbf,train(:,:,s));
resulttrain = transpose(Vtrain);

%clear train end
atrain(:,s)=transpose(resulttrain);



shift = svmrbf.ScaleData.shift;
scale = svmrbf.ScaleData.scaleFactor;
```

```
Xnew1 = bsxfun(@plus,train(:,:,s),shift);
Xnew1 = bsxfun(@times,Xnew1,scale);
sv = svmrbf.SupportVectors;
alphaHat = svmrbf.Alpha;
bias = svmrbf.Bias;
kfun = svmrbf.KernelFunction;
kfunargs = svmrbf.KernelFunctionArgs;
ftrain(:,s) = kfun(sv,Xnew1,kfunargs{:})'*alphaHat(:) + bias;
display(f(:,s));

[Xtraintemp,Ytraintemp,Ttraintemp,AUCtrain(s)]=
perfcurve(trainlab(:,:,s),ftrain(:,s),'CLEAVED');
[r] = length(Xtraintemp);
Xtrain(1:r, s) = Xtrain(1:r, s) + Xtraintemp;
Ytrain(1:r, s) = Ytrain(1:r, s) + Ytraintemp;
Ttrain(1:r, s) = Ttrain(1:r, s) + Ttraintemp;

clear Xtraintemp Ytraintemp Ttraintemp

[Xtrain(:,s),Ytrain(:,s),Ttrain(:,s),AUCtrain(s)]=
perfcurve(trainlab(:,:,s),ftrain(:,s),'CLEAVED');
```

```
AUCtrainav = mean(AUCtrain);

Stdtrain = std(AUCtrain);


tftrain(:,s) = strcmp (atrain(:,s), trainlab(:,s));

Performancetrain (s)= sum(tftrain(:,s)) / numel(atrain(:,s));

Performancetrainav = mean(Performancetrain);

Stdptrain = std(Performancetrain);


s

end

toc
```

Appendix 2. Supplementary Software for Chapter 3

Running Entire MFPred pipeline:

Inputs

- Crystallographic pdb of protein-peptide complex
- List of five substrate sequences to thread on

Process

1. Initial Relax
    a. Run on initial crystallographic pdb to get rid of internal clashes
2. Thread Peptide-FastRelax
    a. Run this step for each substrate sequence
3. MFPred
    a. Choose the lowest-scoring pdb from 2a for each substrate sequence and use a list of paths to these pdbs as the input for MFPred
4. Distances.py (optional)

Outputs

- Transfac file for each pdb and averaged transfac file
- Distance file (distances per-column and overall)

Initial Relax

Inputs

1. <PATH_TO_XTAL_PDB> Crystallographic pdb of protein-peptide complex

Retrieve from pdb

2. **<PATH_TO_ENZDES_CSTFILE> (for proteases only)**

   Generate yourself based on protease catalytic geometry

3. **<PATH_TO_COO_CSTFILE> (for proteases only)**

   Use a modified version of sidechain_cst_3.py (at /source/src/apps/public/relax_w_allatom_cst/sidechain_cst_3.py in the Rosetta source code) to generate constraints with settings of 0.1 and 0.5 on the protease atoms.

4. **<RESFILE>**

   NATRO all, NATAA peptide residues

5. **<XML_FILE>**

   Sample xml:

```
<ROSETTASCRIPTS>
    <SCOREFXNS>
        <ScoreFunction name="myscore" weights="<SCORE_FUNCTION>".wts/>
    </SCOREFXNS>
    <TASKOPERATIONS>
        <ProteinInterfaceDesign design_chain2="0" modify_after_jump="1"
name="pido"/>
        <InitializeFromCommandline name="init"/>
        <ReadResfile name="rrf"/>
    </TASKOPERATIONS>
    <FILTERS/>
    <MOVERS>
```

```
        <AddOrRemoveMatchCsts cst_instruction="add_new" name="cstadd"/>
        <FastRelax name="fastrelax" repeats="8" scorefxn="myscore"
task_operations="pido,init">
            <MoveMap name="mm">
                <Chain bb="1" chi="1" number="2"/>
                <Chain bb="1" chi="1" number="1"/>
                <Jump number="1" setting="1"/>
            </MoveMap>
        </FastRelax>
        <TaskAwareMinMover bb="0" chi="1" jump="0" name="min_pro"
scorefxn="myscore" task_operations="rrf"/>
        <PackRotamersMover name="repack" task_operations="rrf"/>
        <ConstraintSetMover name="protease_cst"/>
    </MOVERS>
    <APPLY_TO_POSE/>
    <PROTOCOLS>
        <Add mover_name="protease_cst"/>
        <Add mover_name="repack"/>
        <Add mover_name="min_pro"/>
        <Add mover_name="cstadd"/>
        <Add mover_name="fastrelax"/>
    </PROTOCOLS>
</ROSETTASCRIPTS>
```

6. <PATH_TO_FLAGS>

-mute core.io.database

-packing::use_input_sc

-packing::extrachi_cutoff 1

-packing::ex1

-packing::ex2

-linmem_ig 10

-out:file::output_virtual

Process

Run on initial crystallographic pdb to get rid of internal clashes.

Command Line:

<ROSETTA_BIN>rosetta_scripts.static.linuxgccrelease -jd2:ntrials 1 -nstruct 1000 -parser:protocol <XML_FILE> -database <ROSETTA_DB> -s <PATH_TO_XTAL_PDB> -run:preserve_header -enzdes::cstfile <PATH_TO_ENZDES_CSTFILE> -constraints:cst_file <PATH_TO_COO_CSTFILE> -resfile <PATH_TO_RESFILE> @<PATH_TO_FLAGS>

Outputs

1000 “relaxed” pdb files. Use lowest scoring pdb file as input for the next step.

Remarks

Differences between protease and PRD:

Protease:

command line includes: -enzdes::cstfile <PATH_TO_ENZDES_CSTFILE> -constraints:cst_file <PATH_TO_COO_CSTFILE>

<SCORE_FUNCTION>: talaris2013_cst

PRD:

command line does not include constraint parameters

<SCORE_FUNCTION>: talaris2013

Thread Peptide-FastRelax

Inputs

1. **<STARTING_RELAXED_MODEL> Lowest scoring pdb from <u>Initial Relax</u> step.**
2. **<PATH_TO_ENZDES_CSTFILE> (for proteases only)**

   Generate yourself based on protease catalytic geometry
3. **<RESFILE>**

   NATRO all, NATAA peptide residues
4. **<XML_FILE>**

   Sample xml:

   <ROSETTASCRIPTS>

   <SCOREFXNS>

   </SCOREFXNS>

   <TASKOPERATIONS>

```
    <ProteinInterfaceDesign name="pido" design_chain2="0"
modify_after_jump="1" />
    <InitializeFromCommandline name="init"/>
    <ReadResfile name="rrf" filename=<RESFILE> />
  </TASKOPERATIONS>

  <FILTERS>
  </FILTERS>

  <MOVERS>
   <MutateResidue name="mut1" target="<PEPT_RES1>" new_res="DM1"/>
   <MutateResidue name="mut2" target="<PEPT_RES2>" new_res="DM2"/>
   <MutateResidue name="mut3" target="<PEPT_RES3>" new_res="DM3"/>
   <MutateResidue name="mut4" target="<PEPT_RES4>" new_res="DM4"/>
   <MutateResidue name="mut5" target="<PEPT_RES5>" new_res="DM5"/>
   <MutateResidue name="mut6" target="<PEPT_RES6>" new_res="DM6"/>
   <MutateResidue name="mut7" target="<PEPT_RES7>" new_res="DM7"/>
   <AddOrRemoveMatchCsts name="cstadd" cst_instruction="add_new" />
   <FastRelax name="fastrelax" repeats="8" task_operations="pido,init">
   <MoveMap name="mm">
          <Chain number="2" chi="1" bb="1"/>
          <Chain number="1" chi="1" bb="0"/>
          <Jump number="1" setting="1"/>
```

```
        </MoveMap>
        </FastRelax>
        <PackRotamersMover name="repack" task_operations="rrf"/>

    </MOVERS>
    <APPLY_TO_POSE>
    </APPLY_TO_POSE>

      <PROTOCOLS>
            <Add mover_name="mut1"/>
            <Add mover_name="mut2"/>
            <Add mover_name="mut3"/>
            <Add mover_name="mut4"/>
            <Add mover_name="mut5"/>
            <Add mover_name="mut6"/>
            <Add mover_name="mut7"/>
            <Add mover_name="cstadd"/>
            <Add mover_name="repack"/>
            <Add mover_name="fastrelax"/>
    </PROTOCOLS>
    </ROSETTASCRIPTS>
```

5. **<PATH_TO_FLAGS>**

-mute core.io.database

-packing::use_input_sc

-packing::extrachi_cutoff 1

-packing::ex1

-packing::ex2

-linmem_ig 10

-out:file::output_virtual

Process

Run on lowest scoring relaxed pdb from Initial Relax one time per substrate sequence. Substitute your peptide sequence for <PEPT_RES1>, etc. in xml script. Add more <MutateResidue> movers as needed. Generates 10 relaxed protease-peptide complexes with that substrate sequence threaded on. Select lowest-scoring complex from these 10 complexes for MFPred step.

Command Line:

<ROSETTA_BIN>rosetta_scripts.static.linuxgccrelease -nstruct 10 -jd2:ntrials 1 -parser:protocol <XML_FILE> -database /home/arubenstein/Rosetta/main/database/ <CONST_ARG> -s <STARTING_RELAXED_MODEL> -run:preserve_header -overwrite @<PATH_TO_FLAGS> -score:weights <SCORE_FUNCTION>

Outputs

10 “relaxed” pdb files. Use lowest scoring pdb file as input for the next step.

Remarks

Differences between protease and PRD:

Protease:

command line includes <CONST_ARG>: -enzdes::cstfile

<PATH_TO_ENZDES_CSTFILE>

<SCORE_FUNCTION>: talaris2013_cst

PRD:

command line does not include constraint parameters

<SCORE_FUNCTION>: talaris2013

MFPred

Inputs

1. **<PATH_TO_INPUT_PDB> Lowest scoring pdb from <u>Initial Relax</u> step.**
2. **<LIST_PDB_COMPLEXES>**

   List of paths to lowest-scoring pdbs for each of the <u>Thread Peptide</u> runs in the previous step.
3. **<RESFILE>**

   NATRO all, NATAA peptide residues that should not be designed (flanking residues), ALLAA peptide residues for which a specificity profile should be predicted.
4. **<XML_FILE>**

   Sample xml:

   <ROSETTASCRIPTS>

```
<TASKOPERATIONS>
    <InitializeFromCommandline name="init" />
    <ReadResfile name="rrf" />
</TASKOPERATIONS>
<SCOREFXNS>
</SCOREFXNS>
<FILTERS>
</FILTERS>
<MOVERS>
    <GenMeanFieldMover name="boltz" threshold="5" lambda_memory="0.5"
tolerance="0.0001" temperature="0.8" task_operations="rrf,init"/>
</MOVERS>
<APPLY_TO_POSE>
</APPLY_TO_POSE>
<PROTOCOLS>
    <Add mover_name="boltz"/>
</PROTOCOLS>
</ROSETTASCRIPTS>
```

5. **<PATH_TO_FLAGS>**

```
-mute core.io.database
-packing::use_input_sc
-packing::extrachi_cutoff 1
-packing::ex1
```

-packing::ex2

-out:file::output_virtual

6. **<EXPT_SPEC_PROFILE> (optional)**

   Path to known (experimentally-derived) specificity profile. MFPred protocol will output certain distances from this profile in the log if this parameter is given.

7. **<ROT_NORM_PARAM> (optional)**

   This is the γ parameter described in the paper. The default is 0.8.

8. **<BB_AVERAGE_PARAM>**

   This is the γ parameter described in the paper. The default is 0.8.

Process

Run on backbone ensemble as generated in Thread Peptide step. Runs MFPred algorithm on residues that are designated as packed/designed in the TaskOperations.

Command Line:

<ROSETTA_BIN>rosetta_scripts.static.linuxgccrelease -database <ROSETTA_DB> -parser:protocol <XML_FILE> -s <PATH_TO_INPUT_PDB> -rot_norm_weight <ROT_NORM_PARAM> -bb_average_weight <BB_AVERAGE_PARAM> -spec_profile <EXPT_SPEC_PROFILE> -bb_list <LIST_PDB_COMPLEXES> -dump_transfac <PATH_TO_OUTPUT_TRANSFAC> -resfile <RESFILE> -nooutput true –score:weights talaris2013 @<PATH_TO_FLAGS>

Outputs

Log contains probabilities per rotamer, probabilities per amino acid, and distances from experimental specificity profile (if provided). If <PATH_TO_OUTPUT_TRANSFAC> is provided, dumps one transfac file per backbone, file with backbone Boltzmann probabilities, and one averaged transfac file for the ensemble as a whole.

Distances.py

Inputs

1. Transfac file as output by MFPred

2. Experimental specificity profile

Process

```
import os
import sys
import numpy as np
import math
from sklearn import metrics
import matplotlib.pyplot as plt
from pylab import *


def binarizeList ( firstList ):
    binary_freq = []


    choose_val = 0.10
```

```
    max_val = max(firstList)
    if max_val < 0.10:
        if max_val > 0.09:
          choose_val = 0.09
        elif max_val > 0.08:
            choose_val = 0.08
        elif max_val > 0.07:
            choose_val = 0.07

    for val in firstList:
        if val > choose_val:
            binary_freq.append( 1 )
        else:
            binary_freq.append( 0 )
    return binary_freq

def areaUnderCurve ( firstList, secondList ):
    binary_freq = binarizeList( firstList )
    fpr, tpr, _ = metrics.roc_curve(binary_freq, secondList)
    auc = metrics.auc(fpr,tpr)
    return auc

def shannonEntropy( firstList ):
```

```
    sE = -1.0 * np.sum( [ p * math.log(p,2) for p in firstList if p != 0.0 ] )
    return sE

def JSDivergence( firstList, secondList ):

    firstSE = shannonEntropy( firstList )

    secondSE = shannonEntropy( secondList )

    combList = [ 0.5 * fL + 0.5 * sL for fL,sL in zip(firstList, secondList) ]

    combSE = shannonEntropy( combList )

    return combSE - 0.5 * firstSE - 0.5 * secondSE

def cosineDist( firstList, secondList):

    dotP = np.dot(firstList, secondList)

    sqrt_1 = math.sqrt( np.sum( np.power( firstList,2 ) ) )
    sqrt_2 = math.sqrt( np.sum( np.power( secondList,2 ) ) )

    return dotP/(sqrt_1 * sqrt_2)
```

```
def frobDist( firstList, secondList):

    diff_lists = np.subtract(firstList,secondList)
    terms = np.power( diff_lists,2)
    return math.sqrt( np.sum( terms ) )

def aveAbsDist( firstList, secondList ):

    diff_lists = np.fabs( np.subtract( firstList, secondList) )
    return sum( diff_lists ) / len( diff_lists )

def readSpecProfileList( filename ):
    with open(filename) as transfac_file:
        transfac = transfac_file.readlines()

    motifWidth = len(transfac)-2

    aaAlpha = transfac[1].split()[1:]

    freq = [{k: 0.0 for k in aaAlpha} for i in range(motifWidth)]

    t_read = transfac[2:]
```

```
    for pos,line in enumerate( t_read,0 ):
        for aa_ind,f in enumerate( line.split()[1:], 0):
            freq[pos][aaAlpha[aa_ind]] = float(f)

    freqList = [ [ val for key,val in sorted(pos.iteritems()) ] for pos in freq ]

    return freqList

def main(args):
    infile = args[1]
    infile_expt = args[2]



    expt = os.path.basename(infile_expt).rstrip()
    expt = expt.rsplit('.',1)[0]

    tokens=infile.rsplit('.',1)
    file=tokens[0]

    outfile= '%s_dist.txt' % (file)
    outfile_heat= '%s_heat.png' % (file)
```

```
freq_in = readSpecProfileList( infile )

freq_expt = readSpecProfileList( infile_expt )

nda_freq_in = np.array( [ freq_in] )

nda_freq_expt = np.array( [ freq_expt] )

flat_freq_in = np.ndarray.flatten( nda_freq_in )

flat_freq_expt = np.ndarray.flatten( nda_freq_expt )


c = [ cosineDist( i, g ) for i,g in zip( freq_in, freq_expt ) ]

f = [ frobDist( i, g ) for i,g in zip( freq_in, freq_expt ) ]

a = [ aveAbsDist( i, g ) for i,g in zip( freq_in, freq_expt ) ]

jsd1 = [ JSDivergence ( i, g ) for i,g in zip( freq_in, freq_expt )]

auc = [ areaUnderCurve ( i, g ) for i, g in zip( freq_expt, freq_in )]

avg_c = cosineDist( flat_freq_in, flat_freq_expt )

avg_f = frobDist( flat_freq_in, flat_freq_expt )

avg_a = aveAbsDist( flat_freq_in, flat_freq_expt )

avg_jsd = np.sum(jsd1) / len(jsd1)

avg_auc = np.sum(auc) / len(auc)


c.append(avg_c)

f.append(avg_f)

a.append(avg_a)

jsd1.append(avg_jsd)

auc.append(avg_auc)
```

```
    dist_out = open(outfile,"w")

    dist_out.write("Metric\t")
    dist_out.write("\t".join([ "Col{0}".format(i) for i in xrange(1,len(c)) ]))
    dist_out.write("\tAvg\nCosine\t")

    dist_out.write("\t".join(map(str,c)))
    dist_out.write("\nFrobenius\t")
    dist_out.write("\t".join(map(str,f)))
    dist_out.write("\nAAD\t")
    dist_out.write("\t".join(map(str,a)))
    dist_out.write("\nJSD\t")
    dist_out.write("\t".join(map(str,jsd1)))
    dist_out.write("\nAUC\t")
    dist_out.write("\t".join(map(str,auc)))
    dist_out.write("\n")

    dist_out.close()

if __name__ == "__main__":
  main(sys.argv)
```

Outputs

Distances file: the name of this file is <INPUT_FILE>_dist.txt. Contains one line per metric. Each line contains one value per column and the last value is the average of the columns.

Non-MFPred pipeline software – used for controls and/or optimization of protocol:

Backbone Ensemble Generation

Thread Peptide Alone (pre-flexpepdock or pre-backrub)

Command Line:

<ROSETTA_BIN>rosetta_scripts.static.linuxgccrelease -nstruct 1 -jd2:ntrials 1 -parser:protocol <XML_FILE> -database <ROSETTA_DB> <CONST_ARG> -s <STARTING_RELAXED_MODEL> -run:preserve_header -overwrite @<PATH_TO_FLAGS>

Sample xml:

```
<ROSETTASCRIPTS>
    <SCOREFXNS/>
    <TASKOPERATIONS>
        <InitializeFromCommandline name="init"/>
        <ReadResfile filename="<RESFILE>" name="rrf"/>
    </TASKOPERATIONS>
    <FILTERS/>
```

```
<MOVERS>
    <MutateResidue name="mut1" target="<PEPT_RES1>" new_res="DM1"/>
    <MutateResidue name="mut2" target="<PEPT_RES2>" new_res="DM2"/>
    <MutateResidue name="mut3" target="<PEPT_RES3>" new_res="DM3"/>
    <MutateResidue name="mut4" target="<PEPT_RES4>" new_res="DM4"/>
    <MutateResidue name="mut5" target="<PEPT_RES5>" new_res="DM5"/>
    <MutateResidue name="mut6" target="<PEPT_RES6>" new_res="DM6"/>
    <MutateResidue name="mut7" target="<PEPT_RES7>" new_res="DM7"/>
   <AddOrRemoveMatchCsts cst_instruction="add_new" name="cstadd"/>
   <PackRotamersMover name="repack" task_operations="rrf,init"/>
</MOVERS>
<APPLY_TO_POSE/>
<PROTOCOLS>
    <Add mover_name="mut1"/>
    <Add mover_name="mut2"/>
    <Add mover_name="mut3"/>
    <Add mover_name="mut4"/>
    <Add mover_name="mut5"/>
    <Add mover_name="mut6"/>
    <Add mover_name="mut7"/>
    <Add mover_name="cstadd"/>
    <Add mover_name="repack"/>
</PROTOCOLS>
```

</ROSETTASCRIPTS>

Resfile:

NATRO all, NATAA peptide residues

Flags:

-mute core.io.database

-packing::use_input_sc

-packing::extrachi_cutoff 1

-packing::ex1

-packing::ex2

-linmem_ig 10

-out:file::output_virtual

FlexPepDock

Command line:

<ROSETTA_BIN>_scripts.static.linuxgccrelease -parser:protocol ~/mean_field/xml/flexpepdock.xml -database <ROSETTA_DB> -s <STARTING_THREADED_MODEL> -ex1 -ex2 -ex1aro -ex2aro -extrachi_cutoff 0 -nstruct 10 -enzdes:cstfile <PATH_TO_ENZDES_CSTFILE> -score:weights talaris2013_cst -run:preserve_header -packing:use_input_sc

Sample xml:

<ROSETTASCRIPTS>

```
    <TASKOPERATIONS>
    </TASKOPERATIONS>
    <SCOREFXNS>
    </SCOREFXNS>
    <FILTERS>
    </FILTERS>
    <MOVERS>
        <AddOrRemoveMatchCsts name="cstadd" cst_instruction="add_new" />
        <FlexPepDock name="fpd" pep_refine="1" />
    </MOVERS>
    <APPLY_TO_POSE>
    </APPLY_TO_POSE>
    <PROTOCOLS>
        <Add mover_name="cstadd"/>
        <Add mover_name="fpd"/>
    </PROTOCOLS>
</ROSETTASCRIPTS>
```

Backrub

Command line:

<ROSETTA_BIN>backrub_cst.linuxgccrelease -run:preserve_header -score:weights talaris2013_cst -database <ROSETTA_DB> -s <STARTING_THREADED_MODEL> -ex1 -ex2 -ex1aro -ex2aro -extrachi_cutoff 0 -backrub:minimize_movemap

<MOVEMAP_FILE> -backrub:ntrials 10000 -backrub:pivot_residues 215 216 217 218 219 220 221 222 223 224 -overwrite -enzdes:cstfile <PATH_TO_ENZDES_CSTFILE> -packing:use_input_sc

Movemap:

RESIDUE * CHI

JUMP * YES

CHAIN 2 BBCHI

Backrub_cst app:

This app is a version of the general backrub app that includes Enzdes style constraint as a mover. Currently, the general backrub app has been moved to a new Mover called BackrubProtocol mover – had this been available at the time of benchmarking, this would have been used instead.

Enumerate_dihedral

Command line:

<ROSETTA_BIN>enumerate_dihedral.linuxgccrelease -database <ROSETTA_DB> -s <STARTING_RELAXED_MODEL> -anchor_res <FIXED_RES_P1> -run:preserve_header

Enumerate dihedral app:

This app is in my pilot apps folder within the Rosetta source code (Rosetta/main/source/src/apps/pilot/arubenstein/enumerate_dihedral.cc).

Clustering via AmberTools cpptraj:

Run tleap to convert pdb to topology and coordinate files:

tleap.in:

```
source leaprc.ff14SB
source leaprc.phosaa10
loadAmberParams frcmod.ionsjc_tip3p
pdb = loadpdb <PDB_NAME>
addions pdb Cl- 0
addions pdb Na+ 0
#solvatebox pdb TIP3PBOX 10.0
saveamberparm pdb <PDB_NAME>.top <PDB_NAME>.crd
```

Run:

```
tleap -f tleap.in
```

Run cpptraj to cluster:

cpptraj.in file:

```
parm <TOPO_FILE_1>
trajin <COORD_FILE_1>
```

parm <TOPO_FILE_2>

trajin <COORD_FILE_2>

.

.

.

cluster hieragglo clusters <N_CLUSTERS> rms :<PEPT_BEG_RES>-<PEPT_END_RES> repout <N_CLUSTERS> repfmt pdb

Run:

cpptraj -i 'cpptraj.in'

Multispecificity Prediction Controls for MFPred

Monte-Carlo (pepspec)

Command-line:

<ROSETTA_BIN>mc_no_sa.linuxgccrelease -database <ROSETTA_DB> -pepspec:pdb_list <BACKBONE_ENSEMBLE_LIST> -save_low_pdbs false -pepspec:n_peptides 1 -pepspec:use_input_bb true -ex1 -ex2 -extrachi_cutoff 0 -pepspec:diversify_lvl 50 -pepspec:run_sequential -use_input_sc

Mc_no_sa app:

This app is a version of the general pepspec app that includes profiling (necessary to extract running times and determine speedup).

Genetic Algorithm (sequence_tolerance)

Command-line:

<ROSETTA_BIN>sequence_tolerance_control.linuxgccrelease -database <ROSETTA_DB> -s <MODEL_FROM_BACKBONE_ENSEMBLE> -ex1 -ex2 -ex1aro -ex2aro -extrachi_cutoff 0 -ms:generations 5 -ms:pop_size 2000 -ms:pop_from_ss 1 -ms:checkpoint:prefix <PREFIX> -ms:checkpoint:interval 200 -ms:checkpoint:gz -seq_tol:fitness_master_weights 1 1 1 2 -resfile <RESFILE>

Resfile:

NATAA residues according to seqtol_resfile.py, ALLAA peptide residues

Sequence_tolerance_control app:

This app is a version of the general sequence_tolerance app that includes profiling (necessary to extract running times and determine speedup).

Appendix 3. Explanation of Metrics in Chapter 3

We used several metrics and distances to evaluate specificity profile predictions. The Frobenius distance is defined as:

$$Frobenius(E,P) = \sqrt{\sum_{i=1}^{N}(E_i - P_i)^2}$$

where E is a vector of experimentally determined amino acid frequencies and P is a vector of predicted frequencies. To calculate the Frobenius distance of the entire profile, we simply flattened the experimental and predicted profiles into one vector each. Two identical probability distributions have a Frobenius distance of 0, while two most divergent distributions have a Frobenius distance of $(2n)^{1/2}$, where n is equal to the number of positions in the profile.

The Average Absolute Distance (AAD) is defined as:

$$AAD(E,P) = \frac{1}{N}\sum_{i=1}^{N}|E_i - P_i|$$

Again, to calculate the AAD of the entire profile, we flattened each profile to a single vector. AAD ranges between 0 to 1, with 0 as the best score and 1 as the worst score. According to Smith and Kortemme, an AAD of less than 6% (or 0.06) is considered to be a good prediction.

The cosine similarity is defined as:

$$Cosine(E,P) = \frac{\sum_{i=1}^{N} E_i P_i}{\sqrt{\sum_{i=1}^{N} E_i^2}\sqrt{\sum_{i=1}^{N} P_i^2}}$$

We flattened each profile to a single vector. Two identical specificity profiles have a cosine distance of 1 whereas two most divergent profiles have a similarity of 0.

Jensen-Shannon Divergence (JSD) is defined as:

$$JSD(E,P) = H\left(\sum_{i=1}^{N} 0.5E_i + 0.5P_i\right) - 0.5\sum_{i=1}^{N} H(E_i) - 0.5\sum_{i=1}^{N} H(P_i)$$

where H is Shannon entropy, defined as:

$$H(E) = -\sum_{i=1}^{N} E_i \log_2 E_i$$

We calculated the JSD of the entire profile by averaging the JSD of each vector (or position) in the profile. A JSD of zero denotes two identical profiles, whereas a JSD of 1 denotes two entirely divergent profiles. While JSD is not considered a proper metric, it does provide information regarding how divergent two profiles are.

Area under the ROC curve, or AUC, as developed by Smith and Kortemme [56], is another measure that we used to evaluate the profiles. We plotted an ROC curve for each predicted profile based on how well the most frequent experimental amino acids (defined as > 10%) are recapitulated in the predicted profile. We then calculated the area under the curve, which denotes the probability that the predicted profile ranks a positive amino acid as higher than a negative amino acid. An AUC of 1 represents a perfect prediction, while an AUC of 0.5 is equivalent to a random prediction.

Last, we developed a new distance, referred to as the Score-Sequence AUC Loss (SSAL). This distance also takes advantage of an ROC curve, although this one is slightly different. We use the experimental profile to generate a score for each cleaved and uncleaved sequence by taking the sum of the probabilities of each amino acid in the sequence occurring at its position:

$$Score(S) = \sum_{i=1}^{len(S)} E_i(S_i)$$

We then plot an ROC curve that demonstrates how well the scores rank the cleaved vs. uncleaved sequences and calculate its AUC. We repeat the entire process with the predicted profile, and then subtract the predicted ROC-AUC from the experimental ROC-AUC. The result is the SSAL, which denotes how well the predicted profile differentiates between cleaved/uncleaved sequences vs. the experimental profile.

In order to transform the values of the distances to *p*-values, we generated 100,000 random profiles by randomly sampling columns of our protease and PRD experimental profile library and randomly shuffling the amino acid identity of their frequencies so as to generate profiles with similar information content. We then calculated their per-column and overall distance from each experimental profile for each of the six measures. The ranking of a given predicted profile distance value in its given distance list was then used to find the *p*-value.

Appendix 4. Supplementary Methods for Chapter 4

Two step screening approach to avoid stop codons:

The LY104 vector was a gift from Y. Li, B. Iverson, and G. Georgiou (University of Texas at Austin). The library was constructed using a two-step screening approach to avoid enrichment of false positives. The first step was an expression screen, which was done by combining the library with a protease inactive vector (LY104 S139A knockout). The recombination was performed by homologous recombination technique in yeast EBY100 cells. We modified an electroporation-based method as described in[210]. The transformed library was allowed to grow for 48 hours at 30 C, up to an OD_{600} of 2.0. Dilutions of 1/10, 1/100 and 1/1000th were plated from the initial culture to calculate the transformation efficiency and library size. The double positive cell population was isolated and enriched using a Fluorescence Assisted Cell Sorting technique. The expressible library was then recombined with a vector containing the active protease, using the aforementioned homologous recombination technique. This library of functional variants was allowed to grow up to 48 hours at 30 C and then sorted into three sequence pools – cleaved, partially cleaved and uncleaved. The gates for the FACS were defined using clonal substrates that displayed varying levels of cleavage activities. The three sequence pools were enriched via three rounds of successive selection (using FACS) and growth. The DNA from the three sequence pools was extracted using the Omega E.Z.N.A yeast plasmid kit. Biological duplicates were sequenced to get an estimate of error correction necessary for post processing this data.

Library Generation methodology:

The library was constructed using a PCR amplification based technique using NNK mixed base oligonucleotides (Integrated DNA Technologies). The LY104 vector was linearized using DNA oligonucleotides (IDT). The NNK library insert (~576 bp) and linearized vector (~6000 bp) were combined using Homologous Recombination using electro-competent EBY100 yeast cells. The transformed EBY100 cells were rescued using a YPD medium and allowed to grow in a 250 mL Selective Complete Growth Medium (-UW). The media was supplemented with 250 µL of Ampicillin and Kanamycin to avoid bacterial contamination.

Library Testing and Enrichment:

The transformed library was allowed to grow for ~48 hrs (upto OD_{600} 2.0) and then induced and tested using Flow cytometry. 1.5 x 10^7cells(OD_{600} ~0.5) were pelleted and resuspended in 2 mL induction media (20g/L galactose, 2 g/L glucose) supplemented with 2 µL each of 1000x antibiotics (carbenicillin, kanamycin). The induction cultures were grown overnight at 30 C (225 rpm) to an OD_{600} of 1-1.5. All spins in the protocol were done at 3000 r.c.f for 5 min. The induced cultures were pelleted and washed with 500 µL PBS followed by 500 µL PBS+ 0.5% BSA. 1 µL of each antibody stain (anti-FLAG PE from Prozyme, PJ315 and anti-HA FITC from Genscript, A01621) was incubated with 10^7 cells for 30 min at 4 C. The samples were resuspended by vortexing and incubated at RT for an additional 30 min. The cells were washed with 100µL PBS with 0.5% BSA, pelleted and then resuspended in 500 µL PBS. Samples were diluted to achieve a final concentration of 10^6 cells/mL and then FITC (anti-HA) and PE (anti-FLAG) intensities were detected using a Flow Cytometer (Beckman Coulter Gallios).

The tested cells were then enriched using a MoFlo XDP Cell Sorter (final cell density 10^7). Up to 10^6 cells were collected and grown in the Selective Complete Growth Media for 48 Hours. Two rounds of sorting and enrichment were carried out to select for clones that were expressed. The selected cells were grown for 48 hours. The DNA from the selected cell population was extracted using E.Z.N.A Zymoprep Kit (Omega).

Cell Sorting into Cleaved, Uncleaved, Partially Cleaved Populations:

The expressible fraction of the library was combined with the active LY104 vector using a second round of Homologous recombination following the same protocol as mentioned above. Using the MoFlo XDP Cell Sorter we defined Cleaved, Uncleaved and Partially cleaved gates for further selection of this population. These gates were defined based on previously experimentally tested sequences.

These cells from the selected population were put through three rounds of enrichment and sorting. In the first round of sorting, cells were collected into two gates – Cleaved and Uncleaved. The Uncleaved sample was further enriched in the second sorting round whereas the Cleaved population was separated into Cleaved and Partially cleaved gates. Cells were collected for each sorting round until a cell count of 10^6 was reached. At the culmination of each sorting round, DNA was collected from each population by using a Zymoprep Kit (Omega).

Preparation for Illumina Sequencing Run:

The DNA samples collected from each of the populations were prepared by 25 cycle amplification[211] with inner primers (Supplementary Table 3). The samples were then run on a 1% Agarose gel to confirm the amplification of a single species. These were further amplified using 8 PCR cycles to include the DNA barcode used in the deep sequencing protocol and checked for quality using a Bioanalyzer 2100. The Deep sequencing was performed on a NextSeq 500 (Illumina) giving a 75 bp paired end read.

I. Expression Protocols

II. Protease expression:

Expression and purification protocol was a modification of previously published protocols[64,266,267]. Transformed BL21 (DE3) *E. coli* cells were grown at 37°C and induced at an optical density of 0.6 by adding 1 mM IPTG. Cells were harvested after 5 hours of expression, pelleted, and frozen at −80°C for storage. Cell pellets were thawed, resuspended in 5 mL/g of resuspension buffer (50 mM phosphate buffer, 500 mM NaCl, 10% glycerol, 30 mM imidazole, 2 mM β-ME, pH 7.5) and lysed with a sonicator. The soluble fraction was retained, applied to a nickel column (Qiagen), washed with resuspension buffer, and eluted with resuspension buffer supplemented with 200 mM imidazole. The eluent was dialyzed overnight (MWCO 10 kD) into a protease storage buffer (20mM Tris.HCl,pH 8.0, Glycerol 20%, 100 mM KCl, 1mM DTT, 0.2 mM EDTA) to remove the imidazole. The purified protein was then flash frozen and stored at -80 C.

Substrate (MBP-GST construct) expression: The transformed BL21(DE3) cells were grown at 37 C to an optical density of 0.6 and induced using 0.2 mM IPTG. Upon induction the cells were grown overnight at 18 C. the cells were harvested and the cell pellet was resuspended in a resuspension buffer (50 mM Tris.HCl, pH8.0, 500 mM NaCl, 30 mM immidazole). The resuspended cells were lysed via sonication and the soluble fraction was applied to a Nickel column (Qiagen). The column was washed using the resuspension buffer and then the protein eluted using an Elution buffer- 50 mM Tris.HCl, pH8.0, 150 mM NaCl, 300 mM imidazole. The protein was dialyzed overnight to remove the imidazole and frozen until use.

Gel based validation assay: The frozen aliquots of substrate solutions were thawed and dialyzed overnight into the reaction buffer (50mM HEPES (pH 7.5), 150 mM NaCl, 0.1% Triton X-100, 15% Glyecerol, 10mM DTT). 28.5 nM substrate was incubated overnight with 500nM, 700nM, 1uM, 2uM, 3uM and 4 uM protease. The resultant reactions were run on a SDS PAGE gel to check for cleavage activity.

III. Sequence Processing

A. Sequence Alignment and Trimming

Data was received oriented in the correct orientation and filtered for quality of 20. Each sequence was searched for the presence and location of “TCTTTATAA”, a unique string within the WT sequence, to align the sequences. If the index of “TCTTTATAA” in sequence *a* is less than the index of “TCTTTATAA” in the WT sequence, the beginning of sequence *a* is padded to match the beginning of the WT sequence. If the index of

"TCTTTATAA" in sequence *a* is greater than the index of "TCTTTATAA" in the WT sequence, the beginning of sequence *a* is truncated to match the beginning of the WT sequence. If "TCTTTATAA" is not found in sequence *a*, it is discarded. If the padded or truncated sequence *a* is shorter than the index of the library region in the WT sequence, sequence *a* is discarded. If sequence *a* is longer than the index of the library region but shorter than the WT sequence, the end of sequence *a* is padded to match the WT sequence. Finally, we check that the padded or truncated sequence *a* matches the WT sequence entirely except for the library region. If it does not match the WT sequence, we discard sequence *a*.

B. Threshold Determination

After aligning and trimming sequences, we calculate a normalized count of each sequence so that the sum of the normalized counts in each population is equal. This is achieved by multiplying each sequence count in population *a* by a normalization factor that is equal to the number of sequences in the largest library divided by the number of sequences in library *a*. Then, to determine the minimum frequency of each sequence in the population above which we are confident of the validity of its representation in the library, we used several methods:

1) **Overlap between cleaved and uncleaved sequences**:

We expect little overlap between the populations of cleaved and uncleaved sequences. However, at low counts, there is some overlap between the two populations. For each threshold, we calculated the number of sequences that overlapped between cleaved and uncleaved sequences, and normalized by the count of unique translated cleaved DNA

sequences at that threshold. We determined the amount of overlap as a percentage of the initial overlap between the populations at a threshold of 1, and then found the threshold that gave <= 10% of the initial overlap (see Figure 3.2). We repeated this analysis for all four variant populations. The threshold was less than or slightly greater than 11 for all variants.

2) **Duplicate population error:**

We sampled technical duplicates for the third round of enrichment for cleaved, uncleaved and partially cleaved sequence pools. As a post - processing step in the pipeline, we introduced duplicate population error correction, by plotting the difference in counts for common sequences of the technical duplicate samples and plotting against the counts in the first sample.

3) **SVM Convergence**:

To select for the threshold that gave us the most distinct populations, we generated cleaved and uncleaved sequence sets for thresholds 5,10,11,12,13,14, 15, 16, 25, 50, 75 and 100. Using an SVM based technique described previously[103] we calculated the auROC for all cleaved and uncleaved sequence populations for the listed thresholds. This enabled us to identify which threshold increases the distinction between the two populations.

We decided upon a frequency threshold of 11 as one that satisfies all categories of threshold determination.

C. Enrich Software

We used a modified version of the Enrich software[169] to find an enrichment ratio (ER) for each sequence. We only included sequences that had a normalized count (as defined above) of greater than or equal to eleven for both the unselected and selected populations. The enrichment ratio of sequence v in population X is defined using Equation 1. $F_{v,X}$ is the frequency of sequence v in population X.

$$ER_{v,X} = \log_2 \frac{F_{v,X}}{F_{v,input}}$$

Eq. 1

D. Population Categorization

Sequences were sorted into one of three pools (cleaved, uncleaved and partially cleaved), based on the following criteria. Sequences that had a positive ER for more than one pool were discarded. Sequences that had a positive ER for either or both replicates for one pool only were assigned to that pool. Negative ERs were ignored.

We also sorted a second set with more stringent criteria, which was then used for training the SVM. For this set, if a sequence was found in more than one pool (even if it had a negative ER in the second pool), it was discarded. Additionally, only sequences with a positive ER > 2.0 were considered.

Computational

Graph Generation

Graph generation was done using Gephi 0.9.1[268]. Nodes were assigned a fitness of 2.0 for cleaved nodes, 1.5 for partially cleaved nodes, and 1.0 for uncleaved nodes. Edge directionality was determined by distance from DEMEE, the starting sequence for library generation; in the case of edge *a* connecting nodes *b* and *c*, the node with a smaller hamming distance from DEMEE was chosen as the source for edge *a*. Edge weight was defined as the ratio of the starting sequence fitness to the ending sequence fitness. The graph layout was run in two steps, starting with a Fruchterman-Reingold layout to separate the nodes and then ending with the ForceAtlas2 layout to generate a force-directed graph. All statistics were run with Gephi default settings.

Random Graph

The edges in the wild-type HCV graph were randomized using the following process. The source of each edge was kept and a population (cleaved, partially cleaved, or uncleaved) was randomly chosen for the target of the edge. The target of the edge was then randomly chosen from among that population.

SVM Sequence Features

We used an encoding scheme that included twenty binary features per amino acid residue, where one of those features was a one and the rest were zeroes. The placement of the one was dependent on the identity of the amino acid. With five amino acid residues per sequence, this resulted in 100 total sequence features.

Mutual Information

Correlation between residues at different positions was calculated using a mutual-information based metric (Equation 2), with modifications based on Buslje et al. (Equation 3)[269] and Gouveia-Oliveira and Pedersen (Equation 4)[270]. We begin with MI between amino acid *a* at position *i* and amino acid *b* at position *j* defined as:

$$MI_{a_i b_j} = \log \frac{P(a_i b_j)}{P(a_i) \cdot P(b_j)}$$

Eq. 2

$P(a_i)$ and $P(a_i b_j)$ are defined with a pseudocount to correct for MSAs with low counts.

$$P(a_i) = \frac{\lambda + N(a_i)}{\sum_x \lambda + N(x_i)}$$

Eq. 3

$N(a_i)$ is the count of amino acid *a* appearing at position i. λ is equal to the length of sequences in the MSA divided by 20 for single-amino acid counts ($N(a_i)$) and 400 for double-amino acid counts $N(a_i b_j)$. We also modified MI to include row-column weighting:

$$MI_{rcw} = \frac{MI_{a_i b_j}}{(\sum_x MI_{x_i b_j} + \sum_y MI_{a_i y_j} - MI_{a_i b_j})/19}$$

Eq. 4

Obtaining viral genomes from patient populations: The list of complete viral polyprotein genomes was accessed and downloaded from NCBI. These genomes were checked to ensure that the sequence covered all NS3 substrate regions. We translated the DNA sequence that we downloaded from NCBI into a protein sequence and compared

the five substrate regions “DLEVVTST”, “DEMEECASHL”, “EDVVCCSM”, ECTTPCSGS” and “ALVTPCASH” to discover the diversity found in the substrate region for the patient genomes.

The dataset of aligned genomes utilized in Cuypers et al. was used for dN/dS measurements and for the mapping of predicted cleaved and uncleaved sequences within the genome[219].

Supplementary Tables:

1. **Genes:**

Gene	DNA sequence
HCV protease (PDB ID: 3SV6)	**CGGATAACAA TTCCCCTCTA GAAATAATTT TGTTTAACTT TAAGAAGGAG ATATACATATG**GGC AGT CAC ATG GCC TCG ATG AAA AAG AAA GGC TCT GTG GTG ATC GTG GGG CGC ATC AAC CTG TCT GGC GAT ACC GCG TAC GCG CAA CAG ACG CGG GGT GAG GAA GGC TGT CAG GAG ACC TCG CAA ACG GGT CGT GAT AAA AAC CAG GTA GAG GGT GAA GTG CAG ATT GTG AGT ACA GCG ACG CAG ACC TTT CTG GCC ACC TCG ATC AAT GGT GTA CTG TGG ACG GTA TAT CAT GGT GCT GGC ACA CGT ACT ATT GCG TCG CCG AAA GGC CCT GTG ACG CAG ATG TAC ACA AAT GTG GAC AAA GAT TTG GTG GGA TGG CAG GCT CCG CAA GGT AGC CGC AGT TTG ACT CCT TGT ACG TGC GGT TCG TCA GAT CTG TAT CTT GTG ACT CGC CAC GCG GAT GTC ATC CCG GTA CGC CGC CGT GGC GAT TCC CGT GGT TCT CTG CTT TCT CCG CGC CCT ATC TCA TAT CTT AAA GGT TCA AGT GGA GGA CCA CTG TTA TGT CCG GCG GGG CAC GCA GTC GGA ATT TTT CGT GCG GCG GTT TCT ACT CGG GGA GTT GCA AAA GCT GTT GAC TTC ATT CCG GTT GAA TCT TTG GAA ACA ACC ATG CGG TCG CCG **CTCGAGCAC CATCACCACC ACCACTGA**

2. **Cell sorting statistics:**

	Functional pool	Sort Round	Cell #
1	CLEAVED	1	420 K

	UNCLEAVED		109 K
	CLEAVED MIDDLE UNCLEAVED	2	1.05M 105K 775K + 295K
	CLEAVED MIDDLE UNCLEAVED	3	1.55M 89K 675K
2	CLEAVED UNCLEAVED	1	1 M 205 K
	CLEAVED MIDDLE UNCLEAVED	2	1.15M 300K 1.05 M
	CLEAVED MIDDLE UNCLEAVED	3	2M 262K 707K
9	CLEAVED	1	812K + 2.65 M
	UNCLEAVED		359K
	CLEAVED MIDDLE UNCLEAVED	2	1.4 M 94 K 1.02 M
	CLEAVED MIDDLE UNCLEAVED	3	1.77 M 324 K 1.5 M
10	CLEAVED UNCLEAVED	1	2.7 M 646 K
	CLEAVED MIDDLE UNCLEAVED	2	1.04M 183K 1.06 M
	CLEAVED MIDDLE UNCLEAVED	3	1.59M 1.16M 1.5M

3. List of oligos for next - sequencing library generation

Primer	DNA Sequence
NNK library reverse primer	TTTCACTGCCTTTATCATCATCATCTTTATAATCACTGCC CAAATGAGAAGCACAMNNMNNMNNMNNMNNCGACCC TCCGCCTCCGCTACCGCCTCCACC
Library insert forward primer	CTGGGGTAATTAATCAGCGAAGCGATGATTTTTGATCTA TTAACAGATATATAAATGC
Vector forward primer	GGCAGTGATTATAAAGATGATGATGATAAAGGCAGTGA AA

Vector reverse primer	GCATTTATATATCTGTTAATAGATCAAAAATCATCGCTT CGCTGATTAATTACCCCAG
Insert amplification post library generation	TTTCACTGCCTTTATCATCATCATCTTTATAATCACTGCC

4. List of oligos for Illumina sample prep and sequencing

Primers	Sequence
Illumina Insert Amplification Forward	CGT TCC AGA CTA CGC TCT GCA GGC TA
Illumina Insert Amplification Reverse	GGC AGT GAT TAT AAA GAT GAT GAT GAT AAA GGC AGT G
Sequencing LYSeq_114	GCC GGA CAG GAT GAT TCT GCC TAC GAT TAC TAC TGA GCC
Sequencing P104	GGATATTACATGGGAAAACATGTTGTTTACGGAG

5. Deep sequencing processing statistics

Variant	Population	Initial		Post-thresholding		Post-categorization
		Unique Counts	Unique Ratios	Unique Counts	Unique Ratios	Unique Sequences
WT	Background	379360		74574		
	Cleaved-Rep1	216253	84771	30327	23549	7472
	Cleaved-Rep2	260763	95729	29237	23691	
	Partial-Rep1	219368	89829	32354	22689	8737
	Partial-Rep2	354252	123572	28630	21985	
	Uncleaved-Rep1	587739	183535	39234	32297	14702
	Uncleaved-Rep2	473114	160979	39114	32052	
R155K/ A156T/ D168A	Background	339048		64405		
	Cleaved	139721	50894	16373	10947	3135
	Partial	270662	108407	40601	29623	11562
	Uncleaved	209208	75868	23431	10424	3703
A156T	Background	367895		68198		
	Cleaved	140478	52198	18717	9910	3644
	Partial	251273	95065	26347	17150	8461
	Uncleaved	277993	109683	29934	17593	9564
D168A	Background	314941		65786		
	Cleaved	197577	65956	19017	10347	4350

	Partial	336653	108566	30534	16928	5780
	Uncleaved	286783	96577	26992	15154	7514

6. List of oligos for testing substrates in yeast surface display

Primers	DNA Sequence
TLIIPCASHL forward	CGGTAGCGGAGGCGGAGGGTCGACATTGATTATTCCTTGTGC
TLIIPCASHL reverse	CTTTATAATCACTGCCCAAATGAGAAGCACAAGGAATAATCAATGTCGAC
ASIIPCASHL forward	CGGTAGCGGAGGCGGAGGGTCGGCGTCAATTATTCCTTGTG
ASIIPCASHL reverse	CTTTATAATCACTGCCCAAATGAGAAGCACAAGGAATAATTGACGCCGA
TATTA forward	CGGTAGCGGAGGCGGAGGGTCGACAGCGACAACAGCGT
TATTA reverse	CTTTATAATCACTGCCCAAATGAGAAGCACACGCTGTTGTCGCTGT
LHTNI forward	GGTAGCGGAGGCGGAGGGTCGTTGCAT ACAAATATT TGTGCTTCTCATTTG
LHTNI reverse	TTATCATCATCATCTTTATAATCACTGCCCAAATGAGAAG CACAAATATTTGTATGCAA
HNTSN forward	GGTAGCGGAGGCGGAGGGTCGCAT AAT ACA TCA AAT TGTGCTTCTCATTTG
HNTSN reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACAATTTGATGTATTATG
SQTGQ forward	GGTAGCGGAGGCGGAGGGTCGTCA CAA ACA GGT CAA TGTGCTTCTCATTTG
SQTGQ reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACATTGACCTGTTTGTGA
PSTVL forward	GGTAGCGGAGGCGGAGGGTCGCCT TCA ACA GTG TTG TGTGCTTCTCATTTG
PSTVL reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACACAACACTGTTGAAGG
PSTTL forward	GGTAGCGGAGGCGGAGGGTCGCCT TCA ACA ACA TTG TGTGCTTCTCATTTG
PSTTL reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACACAATGTTGTTGAAGG
PSTVF forward	GGTAGCGGAGGCGGAGGGTCGCCT TCA ACA GTG TTC TGTGCTTCTCATTTG
PSTVF reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACAGAACACTGTTGAAGG
PSTTF forward	GGTAGCGGAGGCGGAGGGTCGCCT TCA ACA ACA TTC TGTGCTTCTCATTTG

PSTTF reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACAGAATGTTGTTGAAGG
LSLQP forward	GGTAGCGGAGGCGGAGGGTCGTTG TCA TTG CAA CCT TGTGCTTCTCATTTG
LSLQP reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACAAGGTTGCAATGACAA
LSPQP forward	GGTAGCGGAGGCGGAGGGTCG TTG TCA CCT CAA CCT TGTGCTTCTCATTTG
LSPQP reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACAAGGTTGAGGTGACAA
LSLIP forward	GGTAGCGGAGGCGGAGGGTCG TTG TCA TTG ATT CCT TGTGCTTCTCATTTG
LSLIP reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACAAGGAATCAATGACAA
LSPIP forward	GGTAGCGGAGGCGGAGGGTCG TTG TCA CCT ATT CCT TGTGCTTCTCATTTG
LSPIP reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACAAGGAATAGGTGACAA
LTTQA forward	GGTAGCGGAGGCGGAGGGTCG TTG ACA ACA CAA GCG TGTGCTTCTCATTTG
LTTQA reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACACGCTTGTGTTGTCAA
LTTKA forward	GGTAGCGGAGGCGGAGGGTCG TTG ACA ACA AAG GCG TGTGCTTCTCATTTG
LTTKA reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACACGCCTTTGTTGTCAA
LTTQL forward	GGTAGCGGAGGCGGAGGGTCG TTG ACA ACA CAA TTG TGTGCTTCTCATTTG
LTTQL reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACACAATTGTGTTGTCAA
LTTKL forward	GGTAGCGGAGGCGGAGGGTCG TTG ACA ACA AAG TTG TGTGCTTCTCATTTG
LTTKL reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACACAACTTTGTTGTCAA
ECTIP forward	GGTAGCGGAGGCGGAGGGTCG GAA TGT ACA ATT CCTTGTGCTTCTCATTTG
ECTIP reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACAAGGAATTGTACATTC
DTMEE forward	GGTAGCGGAGGCGGAGGGTCG GAT ACA ATG GAA GAATGTGCTTCTCATTTG
DTMEE reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACATTCTTCCATTGTATC
DEMIE forward	GGTAGCGGAGGCGGAGGGTCG GAT GAA ATGATT GAA TGTGCTTCTCATTTG
DEMIE reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC

	ACATTCAATCATTTCATC
ALGTG forward	GGTAGCGGAGGCGGAGGGTCG GCG TTG GGT ACA GGT TGTGCTTCTCATTTG
ALGTG reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACAACCTGTACCCAACGC
RPGPG forward	GGTAGCGGAGGCGGAGGGTCG CGC CCT GGT CCT GGT TGTGCTTCTCATTTG
RPGPG reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACAACCAGGACCAGGGCG
ALVTG forward	GGTAGCGGAGGCGGAGGGTCG GCG TTG GTG ACA GGT TGTGCTTCTCATTTG
ALVTG reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACAACCTGTCACCAACGC
EEMIQ forward	GGTAGCGGAGGCGGAGGGTCG GAA GAA ATG ATT CAA TGTGCTTCTCATTTG
EEMIQ reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACATTGAATCATTTCTTC
QTSEM forward	GGTAGCGGAGGCGGAGGGTCG CAA ACA TCA GAA ATG TGTGCTTCTCATTTG
QTSEM reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACACATTTCTGATGTTTG
WSAIP forward	GGTAGCGGAGGCGGAGGGTCG TGG TCA GCG ATT CCT TGTGCTTCTCATTTG
WSAIP reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACAAGGAATCGCTGACCA
STPNK forward	GGTAGCGGAGGCGGAGGGTCG TCA ACA CCT AAT AAG TGTGCTTCTCATTTG
STPNK reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACACTTATTAGGTGTTGA
GTTIP forward	GGTAGCGGAGGCGGAGGGTCG GGT ACA ACA ATT CCT TGTGCTTCTCATTTG
GTTIP reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACAAGGAATTGTTGTACC
HNLAP forward	GGTAGCGGAGGCGGAGGGTCG CAT AAT TTG GCG CCT TGTGCTTCTCATTTG
HNLAP reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACAAGGCGCCAAATTATG
FDTLN forward	GGTAGCGGAGGCGGAGGGTCG TTC GAT ACA TTG AAT TGTGCTTCTCATTTG
FDTLN reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACAATTCAATGTATCGAA
SDYDL forward	GGTAGCGGAGGCGGAGGGTCG TCA GAT TAT GAT TTG TGTGCTTCTCATTTG
SDYDL reverse	TATCATCATCATCTTTATAATCACTGCCCAAATGAGAAGC ACACAAATCATAATCTGA

7. Primers to generate Drug Resistant Mutants

Primer	Sequence
A156T forward	CGTGGGCATATTTAGGACAGCGGTGTGCACCCG
A156T reverse	CGGGTGCACACCGCTGTCCTAAATATGCCCACG
D168A forward	CTAAGGCGGTGGCGTTTATCCCTGTGGAGAAC
D168A reverse	GTTCTCCACAGGGATAAACGCCACCGCCTTAG
Triple Mutant forward	CGTGGGCATATTTAAGACAGCGGTGTGCACCCG
Triple Mutant reverse	CGGGTGCACACCGCTGTCTTAAATATGCCCACG

8. Vector amplification primers for YESS assay

Primers	DNA Sequence
Vector amplification LY104 for-Gibson	CGACCCTCCGCCTCCGCTACC
Vector amplification LY104 rev-Gibson	TGTGCTTCTCATTTGGGCAGTGATTATAAAGATGATGATGATAA

9. SVM parameter tuning: grid search for optimal boxconstraint and rbfsigma parameters. Average AUC is for each set of parameters run with an 80:20 split on the WT experimental full data set for 100 iterations. N/A is shown if the SVM did not converge with these parameters. A boxconstraint of 1 and rbfsigma of 10 was decided on for future calculations.

		boxconstraint					
	AUC	**0.01**	**0.1**	**1**	**10**	**100**	**1000**
rbfsigma	**0.01**	0.5	0.5	0.5	0.5	0.5	0.5
	0.1	0.5	0.5	0.5	0.5	0.5	0.5
	1	0.8715	0.8718	0.872	0.872	0.8723	0.8721
	10	0.9549	0.9811	0.9839	0.9829	0.9809	0.981
	100	0.9695	0.9696	0.975	0.919	0.9825	N/A
	1000	0.9691	0.9691	0.9693	0.9691	0.9748	0.9819

Appendix 5. Definitions for the loops in the loop modeling benchmark for Chapter 5

{

"154l": {

"chainID": "A",

"DOI": "10.1110/ps.9.9.1753",

"EndResidueID": " 164 ",

"PassedFilter": false,

"Sequence": "NVRSYARMDIGT",

"StartResidueID": " 153 "

},

"1a8d": {

"chainID": "A",

"DOI": "10.1002/prot.21990",

"EndResidueID": " 166 ",

"PassedFilter": true,

"Sequence":
"DLPDKFNAYLAN",
"StartResidueID": " 155 "
},
"1arb": {
"chainID": "A",
"DOI":
"10.1002/prot.21990",
"EndResidueID": " 193 ",
"PassedFilter": true,
"Sequence":
"WQPSGGVTEPGS",
"StartResidueID": " 182 "
},
"1arp": {
"chainID": "A",
"DOI":
"10.1110/ps.9.9.1753",

"EndResidueID": " 212 ",

"PassedFilter": false,

"Sequence":
"LDSTPQVFDTQF",

"StartResidueID": " 201 "

},

"1bhe": {

"chainID": "A",

"DOI":
"10.1002/prot.21990",

"EndResidueID": " 132 ",

"PassedFilter": true,

"Sequence":
"GQGGVKLQDKKV",

"StartResidueID": " 121 "

},

"1bn8": {

"chainID": "A",

"DOI": "10.1002/prot.21990",

"EndResidueID": " 309 ",

"PassedFilter": true,

"Sequence": "STSSSSYPFSYA",

"StartResidueID": " 298 "

},

"1c5e": {

"chainID": "A",

"DOI": "10.1002/prot.21990",

"EndResidueID": " 79 ",

"PassedFilter": true,

"Sequence": "YEDVLWPEAASD",

"StartResidueID": " 68 "

},

"1cb0": {

"chainID": "A",

"DOI":
"10.1002/prot.21990",

"EndResidueID": " 44 ",

"PassedFilter": true,

"Sequence":
"YVDTPFGKPSDA",

"StartResidueID": " 33 "

},

"1cnv": {

"chainID": "A",

"DOI":
"10.1002/prot.21990",

"EndResidueID": " 199 ",

"PassedFilter": true,

"Sequence":

"FYNDRSCQYSTG",

"StartResidueID": " 188 "

},

"1cs6": {

"chainID": "A",

"DOI":
"10.1002/prot.21990",

"EndResidueID": " 150 ",

"PassedFilter": true,

"Sequence":
"NEFPNFIPADGR",

"StartResidueID": " 139 "

},

"1ctm": {

"chainID": "A",

"DOI":
"10.1110/ps.9.9.1753",

"EndResidueID": " 20 ",

"PassedFilter": false,

"Sequence":
"YENPREATGRIV",

"StartResidueID": "   9 "

},

"1cyo": {

"chainID": "A",

"DOI":
"10.1110/ps.9.9.1753",

"EndResidueID": " 23 ",

"PassedFilter": true,

"Sequence":
"IQKHNNSKSTWL",

"StartResidueID": " 12 "

},

"1dqz": {

"chainID": "A",

"DOI": "10.1002/prot.21990",

"EndResidueID": " 218 ",

"PassedFilter": true,

"Sequence": "CGNGTPSDLGGD",

"StartResidueID": " 207 "

},

"1dts": {

"chainID": "A",

"DOI": "10.1110/ps.9.9.1753",

"EndResidueID": " 52 ",

"PassedFilter": true,

"Sequence": "SGSEKTPEGLRN",

"StartResidueID": " 41 "

},

"1eco": {

"chainID": "A",

"DOI":

"10.1110/ps.9.9.1753",

"EndResidueID": " 46 ",

"PassedFilter": true,

"Sequence":

"MAKFTQFAGKDL",

"StartResidueID": " 35 "

},

"1ede": {

"chainID": "A",

"DOI":

"10.1110/ps.9.9.1753",

"EndResidueID": " 161 ",

"PassedFilter": true,

"Sequence":

"CLMTDPVTQPAF",

"StartResidueID": " 150 "

},

"1exm": {

"chainID": "A",

"DOI":

"10.1002/prot.21990",

"EndResidueID": " 300 ",

"PassedFilter": true,

"Sequence":

"RGVSREEVERGQ",

"StartResidueID": " 289 "

},

"1ezm": {

"chainID": "A",

"DOI":

"10.1110/ps.9.9.1753",

"EndResidueID": " 133 ",

"PassedFilter": true,

"Sequence":
"FGDGATMFYPLV",

"StartResidueID": " 122 "

},

"1f46": {

"chainID": "A",

"DOI":
"10.1002/prot.21990",

"EndResidueID": " 70 ",

"PassedFilter": true,

"Sequence":
"MVKPGTFDPEMK",

"StartResidueID": " 59 "

},

"1hfc": {

"chainID": "A",

"DOI": "10.1110/ps.9.9.1753",

"EndResidueID": " 176 ",

"PassedFilter": false,

"Sequence": "RGDHRDNSPFDG",

"StartResidueID": " 165 "

},

"1i7p": {

"chainID": "A",

"DOI": "10.1002/prot.21990",

"EndResidueID": " 46 ",

"PassedFilter": true,

"Sequence": "LPSPQHILGLPI",

"StartResidueID": " 35 "

},

"1ivd": {

"chainID": "A",

"DOI":
"10.1110/ps.9.9.1753",

"EndResidueID": " 376 ",

"PassedFilter": false,

"Sequence":
"TISKDLRSGYET",

"StartResidueID": " 365 "

},

"1m3s": {

"chainID": "A",

"DOI":
"10.1002/prot.21990",

"EndResidueID": " 79 ",

"PassedFilter": true,

"Sequence":

"VGEILTPPLAEG",

"StartResidueID": " 68 "

},

"1ms9": {

"chainID": "A",

"DOI":
"10.1002/prot.21990",

"EndResidueID": " 540 ",

"PassedFilter": true,

"Sequence":
"GSTPVTPTGSWE",

"StartResidueID": " 529 "

},

"1msc": {

"chainID": "A",

"DOI":
"10.1110/ps.9.9.1753",

"EndResidueID": " 20 ",

"PassedFilter": true,

"Sequence":
"LVDNGGTGDVTV",

"StartResidueID": "   9 "

},

"1my7": {

"chainID": "A",

"DOI":
"10.1002/prot.21990",

"EndResidueID": " 75 ",

"PassedFilter": true,

"Sequence":
"TPPYADPSLQAP",

"StartResidueID": " 64 "

},

"1onc": {

"chainID": "A",

"DOI": "10.1110/ps.9.9.1753",

"EndResidueID": " 33 ",

"PassedFilter": true,

"Sequence": "MSTNLFHCKDKN",

"StartResidueID": " 22 "

},

"1oth": {

"chainID": "A",

"DOI": "10.1002/prot.21990",

"EndResidueID": " 47 ",

"PassedFilter": true,

"Sequence": "QKGEYLPLLQGK",

"StartResidueID": " 36 "

},

"1oyc": {

"chainID": "A",

"DOI":

"10.1002/prot.21990",

"EndResidueID": " 214 ",

"PassedFilter": true,

"Sequence":

"DPHSNTRTDEYG",

"StartResidueID": " 203 "

},

"1pbe": {

"chainID": "A",

"DOI":

"10.1110/ps.9.9.1753",

"EndResidueID": " 140 ",

"PassedFilter": true,

"Sequence":

"LHDLQGERPYVT",

"StartResidueID": " 129 "

},

"1pmy": {

"chainID": "A",

"DOI":
"10.1110/ps.9.9.1753",

"EndResidueID": " 88 ",

"PassedFilter": false,

"Sequence":
"KCAPHYMMGMVA",

"StartResidueID": " 77 "

},

"1prn": {

"chainID": "A",

"DOI":
"10.1110/ps.9.9.1753",

"EndResidueID": " 26 ",

"PassedFilter": false,

"Sequence":
"VEDRGVGLEDTI",

"StartResidueID": " 15 "

},

"1qlw": {

"chainID": "A",

"DOI":
"10.1002/prot.21990",

"EndResidueID": " 38 ",

"PassedFilter": true,

"Sequence":
"ETLSLSPKYDAH",

"StartResidueID": " 27 "

},

"1rcf": {

"chainID": "A",

"DOI": "10.1110/ps.9.9.1753",

"EndResidueID": " 99 ",

"PassedFilter": false,

"Sequence": "TGDQIGYADNFQ",

"StartResidueID": " 88 "

},

"1rro": {

"chainID": "A",

"DOI": "10.1110/ps.9.9.1753",

"EndResidueID": " 28 ",

"PassedFilter": true,

"Sequence": "ECQDPDTFEPQK",

"StartResidueID": " 17 "

},

"1scs": {

"chainID": "A",

"DOI":
"10.1110/ps.9.9.1753",

"EndResidueID": " 210 ",

"PassedFilter": false,

"Sequence":
"IKSPDSHPADGI",

"StartResidueID": " 199 "

},

"1srp": {

"chainID": "A",

"DOI":
"10.1110/ps.9.9.1753",

"EndResidueID": " 318 ",

"PassedFilter": true,

"Sequence":

"SDVGGLKGNVSI",

"StartResidueID": " 308 "

},

"1t1d": {

"chainID": "A",

"DOI":
"10.1002/prot.21990",

"EndResidueID": " 138 ",

"PassedFilter": true,

"Sequence":
"SGGRLRRPVNVP",

"StartResidueID": " 127 "

},

"1tca": {

"chainID": "A",

"DOI":
"10.1110/ps.9.9.1753",

"EndResidueID": " 316 ",

"PassedFilter": true,

"Sequence":
"AVGKRTCSGIVT",

"StartResidueID": " 305 "

},

"1thg": {

"chainID": "A",

"DOI":
"10.1110/ps.9.9.1753",

"EndResidueID": " 137 ",

"PassedFilter": true,

"Sequence":
"WIYGGAFVYGSS",

"StartResidueID": " 126 "

},

"1thw": {

"chainID": "A",

"DOI": "10.1110/ps.9.9.1753",

"EndResidueID": " 189 ",

"PassedFilter": true,

"Sequence": "PDAFSYVLDKPT",

"StartResidueID": " 178 "

},

"1tib": {

"chainID": "A",

"DOI": "10.1110/ps.9.9.1753",

"EndResidueID": " 110 ",

"PassedFilter": true,

"Sequence": "EINDICSGCRGH",

"StartResidueID": " 99 "

},

"1tml": {

"chainID": "A",

"DOI":

"10.1110/ps.9.9.1753",

"EndResidueID": " 254 ",

"PassedFilter": true,

"Sequence":

"STTNTGDPMIDA",

"StartResidueID": " 243 "

},

"1xif": {

"chainID": "A",

"DOI":

"10.1110/ps.9.9.1753",

"EndResidueID": " 214 ",

"PassedFilter": true,

"Sequence":

"IERLERPELYGV",

"StartResidueID": " 203 "

},

"2cpl": {

"chainID": "A",

"DOI":
"10.1110/ps.9.9.1753",

"EndResidueID": " 155 ",

"PassedFilter": true,

"Sequence":
"FGSRNGKTSKKI",

"StartResidueID": " 144 "

},

"2cyp": {

"chainID": "A",

"DOI":
"10.1110/ps.9.9.1753",

"EndResidueID": " 202 ",

"PassedFilter": false,

"Sequence":
"WGAANNVFTNEF",

"StartResidueID": " 191 "

},

"2ebn": {

"chainID": "A",

"DOI":
"10.1110/ps.9.9.1753",

"EndResidueID": " 143 ",

"PassedFilter": true,

"Sequence":
"YQTPPPSGFVTP",

"StartResidueID": " 132 "

},

"2exo": {

"chainID": "A",

"DOI": "10.1110/ps.9.9.1753",

"EndResidueID": " 304 ",

"PassedFilter": true,

"Sequence": "LVWDASYAKKPA",

"StartResidueID": " 293 "

},

"2pgd": {

"chainID": "A",

"DOI": "10.1110/ps.9.9.1753",

"EndResidueID": " 372 ",

"PassedFilter": false,

"Sequence": "WRGGCIIRSVFL",

"StartResidueID": " 361 "

},

"2pia": {

"chainID": "A",

"DOI":
"10.1002/prot.21990",

"EndResidueID": " 41 ",

"PassedFilter": true,

"Sequence":
"DPQGAPLPPFEA",

"StartResidueID": " 30 "

},

"2rn2": {

"chainID": "A",

"DOI":
"10.1110/ps.9.9.1753",

"EndResidueID": " 101 ",

"PassedFilter": true,

"Sequence":

"WKTADKKPVKNV",

"StartResidueID": " 90 "

},

"2sil": {

"chainID": "A",

"DOI":
"10.1110/ps.9.9.1753",

"EndResidueID": " 265 ",

"PassedFilter": true,

"Sequence":
"ETKDFGKTWTEF",

"StartResidueID": " 254 "

},

"2sns": {

"chainID": "A",

"DOI":
"10.1110/ps.9.9.1753",

"EndResidueID": " 122 ",

"PassedFilter": false,

"Sequence":
"VAYVYKPNNTHE",

"StartResidueID": " 111 "

},

"2tgi": {

"chainID": "A",

"DOI":
"10.1110/ps.9.9.1753",

"EndResidueID": " 59 ",

"PassedFilter": true,

"Sequence":
"CPYLWSSDTQHS",

"StartResidueID": " 48 "

},

"3cla": {

"chainID": "A",

"DOI": "10.1110/ps.9.9.1753",

"EndResidueID": " 182 ",

"PassedFilter": true,

"Sequence": "AKYQQEGDRLLL",

"StartResidueID": " 171 "

},

"3cox": {

"chainID": "A",

"DOI": "10.1110/ps.9.9.1753",

"EndResidueID": " 489 ",

"PassedFilter": false,

"Sequence": "VPGNVGVNPFVT",

"StartResidueID": " 478 "

},

"3hsc": {

"chainID": "A",

"DOI":
"10.1110/ps.9.9.1753",

"EndResidueID": " 81 ",

"PassedFilter": true,

"Sequence":
"RLIGRRFDDAVV",

"StartResidueID": " 70 "

}

"451c": {

"chainID": "A",

"DOI":
"10.1110/ps.9.9.1753",

"EndResidueID": " 27 ",

"PassedFilter": false,

"Sequence":

"HAIDTKMVGPAY",

"StartResidueID": " 16 "

},

"4enl": {

"chainID": "A",

"DOI":
"10.1110/ps.9.9.1753",

"EndResidueID": " 383 ",

"PassedFilter": false,

"Sequence":
"SHRSGETEDTFI",

"StartResidueID": " 372 "

},

"4i1b": {

"chainID": "A",

"DOI":
"10.1110/ps.9.9.1753",

"EndResidueID": " 55 ",

"PassedFilter": true,

"Sequence":
"FVQGEESNDKIP",

"StartResidueID": " 44 "

}

Appendix 6. Supplementary Software for Chapter 5

Native Relax

Inputs

1. <PATH_TO_XTAL_PDB> Crystallographic pdb of protein-peptide complex

Retrieve from pdb

2. <XML_FILE>

Sample xml:

```
<ROSETTASCRIPTS>
        <SCOREFXNS>
           <ScoreFunction name="myscore" weights="talaris2014.wts"/>
        </SCOREFXNS>
        <TASKOPERATIONS>
           <InitializeFromCommandline name="init"/>
        </TASKOPERATIONS>
        <FILTERS/>
        <MOVERS>
                <FastRelax name="fastrelax" repeats="8" scorefxn="myscore"
task_operations="init"/>
        </MOVERS>
        <APPLY_TO_POSE/>
        <PROTOCOLS>
           <Add mover_name="fastrelax"/>
```

```
        </PROTOCOLS>
</ROSETTASCRIPTS>
```

Process

Run on initial crystallographic pdb 100 times to get rid of internal clashes.

Command Line:

```
<ROSETTA_BIN>rosetta_scripts.static.linuxgccrelease -database <ROSETTA_DB> -ex1 -ex2 -extrachi_cutoff 1 -use_input_sc -s <PATH_TO_XTAL_PDB> -parser:protocol <XML_FILE> -score:weights talaris2014
```

Outputs

100 relaxed pdb files.

Rosetta Minimize

Inputs

6. **<PATH_TO_DECOY>** path to decoy conformation or native-like conformation to minimize
7. **<PATH_TO_XTAL_PDB>** path to crystallographic native structure from which to calculate RMSD
8. **<XML_FILE>**

Sample xml:

```
<ROSETTASCRIPTS>
      <SCOREFXNS>
                <ScoreFunction name="myscore" weights”talaris2014” />
        </SCOREFXNS>
      <TASKOPERATIONS>
      </TASKOPERATIONS>

      <FILTERS>
        <Rmsd name="rmsd" threshold="100" superimpose="1" >
        </Rmsd>
      </FILTERS>

      <MOVERS>
        <MinMover name="min_sc_bb" scorefxn="myscore" chi="1" bb="1"
tolerance="0.0001" max_iter="2000" type="lbfgs_armijo_nonmonotone" >
        </MinMover>

      </MOVERS>
<APPLY_TO_POSE>
</APPLY_TO_POSE>

      <PROTOCOLS>
        <Add mover_name="min_sc_bb"/>
        <Add filter_name="rmsd"/>
</PROTOCOLS>
</ROSETTASCRIPTS>
```

Process

Run on all decoy conformations and relaxed native conformations for each protein.

Command Line:

```
<ROSETTA_BIN>rosetta_scripts.static.linuxgccrelease -database <ROSETTA_DB> -ex1 -ex2
-extrachi_cutoff 1 -use_input_sc -s <PATH_TO_DECOY> -parser:protocol <XML_FILE> -
score:weights talaris2014 -in:file:native <PATH_TO_XTAL_PDB> -nblist_autoupdate
```

Outputs

1 minimized pdb file and one line in a score file, including total_score and rmsd values for each minimized conformation.

Rosetta Relax (REF2015)

Inputs

1. **<PATH_TO_DECOY>** path to decoy conformation
2. **<XML_FILE>**

Sample xml:

```
<ROSETTASCRIPTS>
        <SCOREFXNS>
           <ScoreFunction name="myscore" weights="REF2015.wts"/>
        </SCOREFXNS>
        <TASKOPERATIONS>
           <InitializeFromCommandline name="init"/>
        </TASKOPERATIONS>
      <FILTERS>
        <Rmsd name="rmsd" threshold="100" superimpose="1" >
        </Rmsd>
      </FILTERS>
```

```
        <MOVERS>
                <FastRelax name="fastrelax" repeats="8" scorefxn="myscore"
task_operations="init"/>
        </MOVERS>
        <APPLY_TO_POSE/>
        <PROTOCOLS>
           <Add mover_name="fastrelax"/>
        <Add filter_name="rmsd"/>
        </PROTOCOLS>
</ROSETTASCRIPTS>
```

3. <RELAX_SCRIPT>

```
switch:torsion
repeat 3

ramp_repack_min 0.02  0.01     1.0  50
ramp_repack_min 0.250 0.01     0.5  50
ramp_repack_min 0.550 0.01     0.0 100
ramp_repack_min 1     0.00001  0.0 200
accept_to_best
endrepeat

switch:cartesian
repeat 2
ramp_repack_min 0.02  0.01     1.0  50
ramp_repack_min 0.250 0.01     0.5  50
ramp_repack_min 0.550 0.01     0.0 100
ramp_repack_min 1     0.00001  0.0 200
accept_to_best
endrepeat
```

Process

Run on all decoy conformations for each protein.

Command Line:

```
<ROSETTA_BIN>rosetta_scripts.static.linuxgccrelease -database <ROSETTA_DB> -s
<PATH_TO_DECOY> -parser:protocol <XML_FILE> -score:weights REF2015 -relax:script
<RELAX_SCRIPT>
```

Outputs

1 relaxed pdb file and one line in a score file, including total_score and rmsd values for each minimized conformation.

Pareto Solution

Inputs

9. **<Amber scores>** score-file that contains Amber energies for each decoy for a given protein
10. **<Rosetta scores>** score-file that contains Rosetta energies for each decoy for a given protein

Process

Python script that includes the following four methods (dominates and cull taken with slight modifications from Yakym Pirozhenko under Creative Commons license 3.0 and gen_ranks taken from mjolka under Creative Commons license 3.0):

```
def dominates(row, rowCandidate):
    return all(r <= rc for r, rc in zip(row, rowCandidate))

def cull(pts, dominates):
    dominated = []
    cleared = []
    remaining = pts
    while remaining:
        candidate = remaining[0]
        new_remaining = []
        for other in remaining[1:]:
            [new_remaining, dominated][dominates(candidate, other)].append(other)
        if not any(dominates(other, candidate) for other in new_remaining):
            cleared.append(candidate)
        else:
            dominated.append(candidate)
        remaining = new_remaining
    return cleared, dominated

def find_lowest_point( list_pts ):
    first_rank_list = [ p[0] for p in list_pts ]
    second_rank_list = [ p[1] for p in list_pts ]
    min_rank = min(first_rank_list + second_rank_list)
```

```
    min_point = [ (e1, e2, r) for e1, e2, r in list_pts if min_rank == e1 or
min_rank == e2 ][0]
    return min_point

def gen_ranks(list_energies):
    indices = list(range(len(list_energies)))
    indices.sort(key=lambda x: list_energies[x])
    output = [0] * len(indices)
    for i, x in enumerate(indices):
        output[x] = i
    return output

def main(amber_scorefile, rosetta_scorefile):

    #read in amber_scores and rosetta_scores using external library methods READ
    amber_scores = READ(amber_scorefile)
    rosetta_scores = READ(rosetta_scorefile)

    #generate ranks for scores
    amber_ranks = gen_ranks(amber_scores)
    rosetta_ranks = gen_ranks(rosetta_scores)

    pts = map(list, zip(d1e_ranks, d2e_ranks))

    #find the set of Pareto-optimal solutions
    cleared, dominated = cull(pts, dominates)

    #find the set of solutions that have a minimal sum of ranks
    pareto_equal_min = min([ e1+e2 for e1,e2 in cleared.items() ])
    list_pts =  [ (rosetta,amber) for rosetta,amber in cleared if amber+rosetta ==
pareto_equal_min ]
```

```
    #find the subset of solutions that have the lowest rank of all the solutions
with a minimal sum of ranks
    pareto_solution = find_lowest_point( list_pts )
```

Outputs

This gives the ranks of the Pareto-optimal solution, which can be used to find the decoy name or rmsd or any other information about the decoy.

Amber – Generate Initital Structures with tLEaP

Inputs

1. <PDB_ID>

2. <TLEAP_INFILE>

```
source leaprc.protein.ff14SBonlysc
m = loadpdb NoH_<PDB_ID>.pdb
set default pbradii mbondi3
saveamberparm m <PDB_ID>.parm7 NoH_<PDB_ID>.rst7
quit
```

Process

Replaces hydrogen atoms and converts PDB files of crystal structures and decoys to one topology file (parm7) and individual coordinate files (rst7).

Command Line:

```
tleap -f <TLEAP_INFILE>
```

Outputs

One topology file <PDB_ID>.parm7 and all decoy structures as coordinates files <PDB_ID>.rst7

Amber – Minimization

Inputs

1. **<PDB_ID>**

2a. <MIN_SCRIPT> (without restraints)

```
$cntrl
    imin = 1, maxcyc=1000,
    ntx = 1,
    ntxo = 2,
    ntwr = 100, ntpr = 100,
    cut = 999.0,
    ntb = 0, igb = 8,
    ntr = 1,
    ntmin=3, drms=0.01,
```

```
/
```

OR 2b. <MIN_SCRIPT> (with restraints)

```
$cntrl
    imin = 1, maxcyc=1000,
    ntx = 1,
    ntxo = 2,
    ntwr = 100, ntpr = 100,
    cut = 999.0,
    ntb = 0, igb = 8,
    ntr = 1,
    restraint_wt = 10,
    restraintmask = "!:<LOOP_RESIDUES> & !@H=",
    ntmin=3, drms=0.01,
/
```

Process

Run Amber minimization on all decoy conformations and native crystal conformations for each protein.

Command Line:

```
sander -O -i <MIN_SCRIPT> -p <PDB_ID>.parm7 -c <PDB_ID>.rst7 -r
<PDB_ID>.min.rst7 -ref <PDB_ID>.rst7 -o <PDB_ID>.min.out
```

Amber – Scoring

Inputs

1. **<PDB_ID>**
2. **<OUTPUT_FILE>**

Process

Command Line: python GetEnergies.py –i <PDB_ID> -o <OUTPUT_FILE>

Sample python script for gathering scores from loop modeling benchmark set:

```
import pytraj as pt
import sander
import os, sys, getopt
from glob import glob
from mpi4py import MPI
import pandas as pd

loopdef = pd.read_json('loopdefs.json')
```

```
# create mpi handler to get cpu rank
comm = MPI.COMM_WORLD

PATH_TO_NATIVES = '<PATH/TO/NATIVES>’
PATH_TO_DECOYDISC = "<PATH/TO/DECOYS/>"

def chunks(l,n):
    n = max(1,n)
    return [l[i:i+n] for i in range(0, len(l), n)]

def get_ca_rmsd( traj, mask, metric ):

    ca_rmsd = pt.pairwise_rmsd(traj, mask=mask, metric=metric)
    return ca_rmsd;

def get_energies( traj, igb_value ):



    energy_data = pt.pmap_mpi(pt.energy_decomposition, traj,
igb=igb_value )
    return energy_data;



def get_energy_term( traj, term ):

    energy_data = pt.pmap_mpi(pt.energy_decomposition, traj, igb=8)
```

```
    return energy_data[term];



def main(argv):
    args = sys.argv

    input_pdb = ''
    outfile = ''

    try:
        opts, args=getopt.getopt(sys.argv[1:], "ho:i:o:",
["in:file:i=", "out:file:scorefile="])
    except getopt.GetoptError:
        print('Unknown flag given.\nKnown flags:\n\t-h\n\t-n
<native>')
        sys.exit()

    for opt, arg in opts:
        if opt == '-h':
            print('GetEnergies.py --in:file:s <input_pdb_id> --
out:file:scorefile <output_filename>')
            sys.exit()
        elif opt in ("-i", "--in:file:i"):
            input_pdb = arg
        elif opt in ("-o", "--out:file:scorefile"):
            outfile = arg
```

```
    if input_pdb == '':
        print('ERROR: No PDB ID supplied.')
        sys.exit()
    elif len(input_pdb) != 4:
        print("ERROR: Input PDB should be 4 letter PDB code.")
        sys.exit()
    input_pdb = input_pdb.lower()

    if outfile == '':
        outfile = 'Scores.sc'



##################################################################
##

##################################################################
##

    print( "===== Native RST7: {PDB} =====".format(PDB=input_pdb) )

    #Native is crystal structure.
    native_rst7 = PATH_TO_NATIVES+"/{PDB}.rst7".format(
PDB=input_pdb )
    native_parm = PATH_TO_NATIVES+"/{PDB}.parm7".format(
PDB=input_pdb )

    print("Native RST7\t{natrst}".format(natrst=native_rst7))
```

```
    print("Native PARM7\t{natparm}".format(natparm=native_parm))

    os.chdir(PATH_TO_DECOYDISC+"{PDB}/".format( PDB=input_pdb ) )
    print(PATH_TO_DECOYDISC+"{PDB}/".format( PDB=input_pdb ) )
    min_decoys = glob('min*.rst7')  ## List of rst7s ["<rst7_1>",
"<rst7_2>", ...]

    print("== Analyzing %i mols==" % len(min_decoys))

    min_decoys.insert(0, native_rst7)

    print("\t=== Loading Trajectories ===")
    traj = pt.iterload(min_decoys, native_parm)
    print(traj)

    print("\t=== Getting RMSDs ===")
    ca_mask = '@CA & :{loop_start}-
{loop_end}'.format(loop_start=int(loopdef[input_pdb]['StartResidueID
'].strip()), loop_end=int(loopdef[input_pdb]['EndResidueID'].strip()
))
    print ca_mask

    ca_rmsd_nofit = get_ca_rmsd( traj, ca_mask, 'nofit' )
    ca_rmsd_fit = get_ca_rmsd( traj, ca_mask, 'rms' )

    print("\t=== Getting Energy Decomposition ===")
    energy_data = get_energies( traj, 8 )
```

```
    print("\t\tFinished")

    if energy_data:

        energy_data['rmsd'] = ca_rmsd_nofit[0]
        energy_data['rmsd_suploop'] = ca_rmsd_fit[0]

        ekeys = energy_data.keys()
        ekeys.sort()

        print("Outfile: " + outfile)
        with open(outfile,'w') as scorefile:
            header = 'description\t'
            for key in ekeys:
                header += key + '\t'
            scorefile.write(header+"\n")

            for pdb_index in range(len(min_decoys)):
                scoreline = min_decoys[pdb_index]+'\t'
                for key in ekeys:
                    scoreline += '%s\t' %
str(energy_data[key][pdb_index])
                scorefile.write(scoreline+"\n")

if __name__ == "__main__":
    main(sys.argv[1:])
```

Outputs

One output file with Amber scores and RMSDs for native crystal structure and decoys.

Bibliography

(1) Li, Q.; Yi, L.; Marek, P.; Iverson, B. L. *FEBS Lett*. **2013**, *587*, 1155–1163.

(2) Hedstrom, L. *Chem. Rev*. **2002**, *102* (12), 4501–4524.

(3) Puente, X. S.; Sánchez, L. M.; Overall, C. M.; López-Otín, C. *Nat. Rev. Genet*. **2003**, *4* (7), 544–558.

(4) Tyndall, J. D. A.; Nall, T.; Fairlie, D. P. *Chem. Rev*. **2005**, *105* (3), 973–999.

(5) Prabu-Jeyabalan, M.; Nalivaika, E.; Schiffer, C. A. *Structure* **2002**, *10* (3), 369–381.

(6) Romano, K. P.; Ali, A.; Royer, W. E.; Schiffer, C. A. *Proc. Natl. Acad. Sci. U. S. A*. **2010**, *107* (49), 20986–20991.

(7) Shen, Y.; Altman, M. D.; Ali, A.; Nalam, M. N. L.; Cao, H.; Rana, T. M.; Schiffer, C. A.; Tidor, B. *ACS Chem. Biol*. **2013**, *8* (11), 2433–2441.

(8) Lin, C. *HCV NS3-4A Serine Protease*; Horizon Bioscience, 2006.

(9) Phan, J.; Zdanov, A.; Evdokimov, A. G.; Tropea, J. E.; Peters, H. K.; Kapust, R. B.; Li, M.; Wlodawer, A.; Waugh, D. S. *J Biol Chem* **2002**, *277* (52), 50564–50572.

(10) Harris, J. L.; Peterson, E. P.; Hudig, D.; Thornberry, N. A.; Craik, C. S. *J. Biol. Chem*. **1998**, *273* (42), 27364–27373.

(11) Casciola-Rosen, L.; Garcia-Calvo, M.; Bull, H. G.; Becker, J. W.; Hines, T.; Thornberry, N. A.; Rosen, A. **2006**.

(12) Matthews, D. J.; Goodman, L. J.; Gorman, C. M.; Wells, J. A. *Protein Sci*. **1994**, *3* (8), 1197–1205.

(13) Rockwell, N. C.; Krysan, D. J.; Komiyama, T.; Fuller, R. S. *Chem. Rev*. **2002**, *102* (12), 4525–4548.

(14) Walker, J. A.; Molloy, S. S.; Thomas, G.; Sakaguchi, T.; Yoshida, T.; Chambers, T. M.; Kawaoka, Y. *J. Virol*. **1994**, *68* (2), 1213–1218.

(15) Tawfik, D. S. *Curr. Opin. Chem. Biol*. **2014**, *21*, 73–80.

(16) López-Otín, C.; Bond, J. S. *J. Biol. Chem*. **2008**, *283* (45), 30433–30437.

(17) Hedstrom, L. *Chem. Rev*. **2002**, *102* (12), 4429–4430.

(18) Powers, J. C.; Odake, S.; Oleksyszyn, J.; Hori, H.; Ueda, T.; Boduszek, B.; Kam, C. *Agents Actions. Suppl*. **1993**, *42*, 3–18.

(19) Rawlings, N. D.; Salvesen, G. In *Handbook of Proteolytic Enzymes*; 2013.

(20) Rawlings, N. D.; Barrett, A. J.; Bateman, A. *Nucleic Acids Res*. **2010**, *38* (Database issue), D227-33.

(21) Julien, O.; Zhuang, M.; Wiita, A. P.; O'Donoghue, A. J.; Knudsen, G. M.; Craik, C. S.; Wells, J. A. *Proc. Natl. Acad. Sci. U. S. A*. **2016**, *113* (14), E2001-10.

(22) Di Cera, E.; Cantwell, A. M. *Ann. N. Y. Acad. Sci*. **2001**, *936*, 133–146.

(23) Scheel, T. K. H.; Rice, C. M. *Nat. Med.* **2013**, *19* (7), 837–849.

(24) Drag, M.; Salvesen, G. S. *Nat. Rev. Drug Discov.* **2010**, *9* (9), 690–701.

(25) Eder, J.; Hommel, U.; Cumin, F.; Martoglio, B.; Gerhartz, B. *Curr. Pharm. Des.* **2007**, *13* (3), 271–285.

(26) Poreba, M.; Drag, M. *Curr. Med. Chem.* **2010**, *17* (33), 3968–3995.

(27) Backes, B. J.; Harris, J. L.; Leonetti, F.; Craik, C. S.; Ellman, J. A. *Nat. Biotechnol.* **2000**, *18* (2), 187–193.

(28) Turk, B. E.; Huang, L. L.; Piro, E. T.; Cantley, L. C. *Nat. Biotechnol.* **2001**, *19* (7), 661–667.

(29) Fretwell, J. F.; K Ismail, S. M.; Cummings, J. M.; Selby, T. L. *Mol. Biosyst.* **2008**, *4* (8), 862–870.

(30) Ratnikov, B.; Cieplak, P.; Smith, J. W. *Methods Mol. Biol.* **2009**, *539*, 93–114.

(31) van den Berg, B. H. J.; Tholey, A. *Proteomics* **2012**, *12* (4–5), 516–529.

(32) Agard, N. J.; Mahrus, S.; Trinidad, J. C.; Lynn, A.; Burlingame, A. L.; Wells, J. A. *Proc. Natl. Acad. Sci. U. S. A.* **2012**, *109* (6), 1913–1918.

(33) Vizovišek, M.; Vidmar, R.; Fonović, M.; Turk, B. *Biochimie* **2016**, *122*, 77–87.

(34) Boyd, S. E.; Garcia de la Banda, M.; Pike, R. N.; Whisstock, J. C.; Rudy, G. B. *Proc. IEEE Comput. Syst. Bioinform. Conf.* **2004**, 372–381.

(35) Song, J.; Tan, H.; Perry, A. J.; Akutsu, T.; Webb, G. I.; Whisstock, J. C.; Pike, R. N. *PLoS One* **2012**, *7* (11), e50300.

(36) Barkan, D. T.; Hostetter, D. R.; Mahrus, S.; Pieper, U.; Wells, J. A.; Craik, C. S.; Sali, A. *Bioinformatics* **2010**, *26* (14), 1714–1722.

(37) Song, J.; Tan, H.; Boyd, S. E.; Shen, H.; Mahmood, K.; Webb, G. I.; Akutsu, T.; Whisstock, J. C.; Pike, R. N. *J. Bioinform. Comput. Biol.* **2011**, *9* (1), 149–178.

(38) Verspurten, J.; Gevaert, K.; Declercq, W.; Vandenabeele, P. *Trends Biochem. Sci.* **2009**, *34* (7), 319–323.

(39) Li, B.-Q.; Cai, Y.-D.; Feng, K.-Y.; Zhao, G.-J. *PLoS One* **2012**, *7* (9), e45854.

(40) Yi, L.; Gebhard, M. C.; Li, Q.; Taft, J. M.; Georgiou, G.; Iverson, B. L. *Proc. Natl. Acad. Sci. U. S. A.* **2013**, *110* (18), 7229–7234.

(41) Shiryaev, S. A.; Thomsen, E. R.; Cieplak, P.; Chudin, E.; Cheltsov, A. V; Chee, M. S.; Kozlov, I. A.; Strongin, A. Y. *PLoS One* **2012**, *7* (4), e35759.

(42) Rögnvaldsson, T.; Etchells, T. a; You, L.; Garwicz, D.; Jarman, I.; Lisboa, P. J. G. *BMC Bioinformatics* **2009**, *10*, 149.

(43) Kostallas, G.; Löfdahl, P.-Å.; Samuelson, P. *PLoS One* **2011**, *6* (1), e16136.

(44) Boulware, K. T.; Jabaiah, A.; Daugherty, P. S. *Biotechnol. Bioeng.* **2010**, *106* (3), 339–346.

(45) Ratnikov, B. I.; Cieplak, P.; Gramatikoff, K.; Pierce, J.; Eroshkin, A.; Igarashi, Y.; Kazanov, M.; Sun, Q.; Godzik, A.; Osterman, A.; Stec, B.; Strongin, A.; Smith, J.

W. *Proc. Natl. Acad. Sci.* **2014**, *111* (40), E4148–E4155.

(46) Tyka, M. D.; Keedy, D. A.; Andr??, I.; Dimaio, F.; Song, Y.; Richardson, D. C.; Richardson, J. S.; Baker, D. *J. Mol. Biol.* **2011**, *405* (2), 607–618.

(47) London, N.; Lamphear, C. L.; Hougland, J. L.; Fierke, C. A.; Schueler-Furman, O. *PLoS Comput. Biol.* **2011**, *7* (10).

(48) Baugh, E. H.; Simmons-Edler, R.; Müller, C. L.; Alford, R. F.; Volfovsky, N.; Lash, A. E.; Bonneau, R. *Nucleic Acids Res.* **2016**, *44* (6), 2501–2513.

(49) Appadurai, R.; Senapati, S. *Biochemistry* **2016**, *55* (10), 1529–1540.

(50) Yi, L.; Taft, J. M.; Li, Q.; Gebhard, M. C.; Georgiou, G.; Iverson, B. L. *Methods Mol. Biol.* **2015**, *1319*, 81–93.

(51) Grakoui, A.; McCourt, D. W.; Wychowski, C.; Feinstone, S. M.; Rice, C. M. *J. Virol.* **1993**, *67* (5), 2832–2843.

(52) Grakoui, A.; Wychowski, C.; Lin, C.; Feinstone, S. M.; Rice, C. M. *J. Virol.* **1993**, *67* (3), 1385–1395.

(53) Hou, T.; Zhang, W.; Case, D. A.; Wang, W. *J. Mol. Biol.* **2008**, *376* (4), 1201–1214.

(54) Teyra, J.; Sidhu, S. S.; Kim, P. M. *FEBS Lett.* **2012**, *586* (17), 2631–2637.

(55) Li, N.; Hou, T.; Ding, B.; Wang, W. *Proteins* **2011**, *79* (11), 3208–3220.

(56) Smith, C. A.; Kortemme, T. *J. Mol. Biol.* **2010**, *402* (2), 460–474.

(57) Crivelli, J. J.; Lemmon, G.; Kaufmann, K. W.; Meiler, J. *J. Comput. Aided. Mol. Des.* **2013**, *27* (12), 1051–1065.

(58) Yanover, C.; Bradley, P. *Proc. Natl. Acad. Sci. U. S. A.* **2011**, *108* (17), 6981–6986.

(59) Lanouette, S.; Davey, J. A.; Elisma, F.; Ning, Z.; Figeys, D.; Chica, R. A.; Couture, J. F. *Structure* **2015**, *23* (1), 206–215.

(60) Chaudhury, S.; Gray, J. J. *Structure* **2009**, *17* (12), 1636–1648.

(61) Jensen, J. H.; Willemoës, M.; Winther, J. R.; De Vico, L. *PLoS One* **2014**, *9* (5), e95833.

(62) Smith, C. A.; Kortemme, T. *PLoS One* **2011**, *6* (7), e20451.

(63) Liu, F.; Luehr, N.; Kulik, H. J.; Martínez, T. J. *J. Chem. Theory Comput.* **2015**, *11* (7), 3131–3144.

(64) Romano, K. P.; Ali, A.; Aydin, C.; Soumana, D.; Ozen, A.; Deveau, L. M.; Silver, C.; Cao, H.; Newton, A.; Petropoulos, C. J.; Huang, W.; Schiffer, C. A. *PLoS Pathog* **2012**, *8* (7), e1002832.

(65) Varadarajan, N.; Rodriguez, S.; Hwang, B.-Y.; Georgiou, G.; Iverson, B. L. *Nat. Chem. Biol.* **2008**, *4* (5), 290–294.

(66) Khersonsky, O.; Tawfik, D. S. *Annu. Rev. Biochem.* **2010**, *79*, 471–505.

(67) Igarashi, Y.; Eroshkin, A.; Gramatikova, S.; Gramatikoff, K.; Zhang, Y.; Smith, J.

W.; Osterman, A. L.; Godzik, A. *Nucleic Acids Res*. **2007**, *35* (Database issue), D546-9.

(68) Hashimoto, H.; Takeuchi, T.; Komatsu, K.; Miyazaki, K.; Sato, M.; Higashi, S. *J. Biol. Chem*. **2011**, *286* (38), 33236–33243.

(69) Prabu-Jeyabalan, M.; Nalivaika, E. A.; King, N. M.; Schiffer, C. A. *J. Virol*. **2003**, *77* (2), 1306–1315.

(70) Waugh, S. M.; Harris, J. L.; Fletterick, R.; Craik, C. S. *Nat. Struct. Biol*. **2000**, *7* (9), 762–765.

(71) Nivón, L. G.; Moretti, R.; Baker, D.; Davis, I.; Baker, D.; Gray, J.; Moughon, S.; Wang, C.; Schueler-Furman, O.; Kuhlman, B.; Siegel, J.; Zanghellini, A.; Lovick, H.; Kiss, G.; Lambert, A.; Jiang, L.; Althoff, E.; Clemente, F.; Doyle, L.; Rothlisberger, D.; Rothlisberger, D.; Khersonsky, O.; Wollacott, A.; Jiang, L.; DeChancie, J.; Fleishman, S.; Whitehead, T.; Ekiert, D.; Dreyfus, C.; Corn, J.; Kuhlman, B.; Dantas, G.; Ireton, G.; Varani, G.; Stoddard, B.; Koga, N.; Tatsumi-Koga, R.; Liu, G.; Xiao, R.; Acton, T.; Tyka, M.; Keedy, D.; Andre, I.; Dimaio, F.; Song, Y.; DiMaio, F.; Terwilliger, T.; Read, R.; Wlodawer, A.; Oberdorfer, G.; Word, J.; Lovell, S.; Richardson, J.; Richardson, D.; Leaver-Fay, A.; O'Meara, M.; Tyka, M.; Jacak, R.; Song, Y.; Song, Y.; Tyka, M.; Leaver-Fay, A.; Thompson, J.; Baker, D. *PLoS One* **2013**, *8* (4), e59004.

(72) Fleishman, S. J.; Leaver-Fay, A.; Corn, J. E.; Strauch, E. M.; Khare, S. D.; Koga, N.; Ashworth, J.; Murphy, P.; Richter, F.; Lemmon, G.; Meiler, J.; Baker, D. *PLoS One* **2011**, *6* (6), 1–10.

(73) Richter, F.; Leaver-Fay, A.; Khare, S. D.; Bjelic, S.; Baker, D. *PLoS One* **2011**, *6* (5), e19230.

(74) Miller, B. R.; McGee, T. D.; Swails, J. M.; Homeyer, N.; Gohlke, H.; Roitberg, A. E. *J. Chem. Theory Comput*. **2012**, *8* (9), 3314–3321.

(75) Gront, D.; Kulp, D. W.; Vernon, R. M.; Strauss, C. E. M.; Baker, D. *PLoS One* **2011**, *6* (8), e23294.

(76) Bond, S. R.; Naus, C. C. *Nucleic Acids Res*. **2012**, *40* (Web Server issue), W209-13.

(77) Kim, P.; Long, L.; Yu, X.; Mark, G. *Science* **2006**, *314* (December), 1938–1941.

(78) Erijman, A.; Aizner, Y.; Shifman, J. M. *Biochemistry* **2011**, *50*, 602–611.

(79) Schreiber, G.; Keating, A. E. *Curr. Opin. Struct. Biol*. **2011**, *21* (1), 50–61.

(80) Schutkowski, M.; Reimer, U.; Panse, S.; Dong, L.; Lizcano, J. M.; Alessi, D. R.; Schneider-Mergener, J. *Angew. Chemie - Int. Ed*. **2004**, *43* (20), 2671–2674.

(81) Khati, M.; Pillay, T. S. *Anal. Biochem*. **2004**, *325* (1), 164–167.

(82) Tonikian, R.; Zhang, Y.; Sazinsky, S. L.; Currell, B.; Yeh, J. H.; Reva, B.; Held, H. A.; Appleton, B. A.; Evangelista, M.; Wu, Y.; Xin, X.; Chan, A. C.; Seshagiri, S.; Lasky, L. A.; Sander, C.; Boone, C.; Bader, G. D.; Sidhu, S. S. *PLoS Biol*. **2008**, *6* (9), 2043–2059.

(83) Vouilleme, L.; Cushing, P. R.; Volkmer, R.; Madden, D. R.; Boisguerin, P. *Angew. Chemie - Int. Ed.* **2010**, *49* (51), 9912–9916.

(84) Stiffler, M. A.; Chen, J. R.; Grantcharova, V. P.; Lei, Y.; Fuchs, D.; Allen, J. E.; Zaslavskaia, L. A.; MacBeath, G. *Science* **2007**, *317* (5836), 364–369.

(85) Sparks, A. B.; Rider, J. E.; Hoffman, N. G.; Fowlkes, D. M.; Quillam, L. A.; Kay, B. K. *Proc. Natl. Acad. Sci. U. S. A.* **1996**, *93* (4), 1540–1544.

(86) Chapman, H. A.; Riese, R. J.; Shi, G. P. *Annu. Rev. Physiol.* **1997**, *59*, 63–88.

(87) Hirsch, T.; Xiang, J.; Chao, D. T.; Korsmeyer, S. J.; Scaife, J. F.; Colell, A.; Morales, A.; Ferna, J. C.; Adachi, S.; Cross, a R.; Babior, B. M.; Gottlieb, R. a; Newmeyer, D. D.; Green, D. R.; Eguchi, Y.; Shimizu, S.; Tsujimoto, Y.; Newmeyer, D.; Farschon, D. M.; Reed, J. C.; Liu, X.; Kim, C. N.; Yang, J.; Jemmerson, R.; Wang, X.; Kluck, R. M.; Li, P.; Heiden, M. G. Vander; Chandel, N. S.; Williamson, E. K.; Schumacker, P. T.; Jurgensmeier, J. M.; Kuwana, T.; Mancini, M.; Kuida, K.; Susin, S. a; Susin, S.; Bredesen, D. E.; Jacobson, M. D.; Raff, M. C.; Esposti, M. D.; Mclennan, H.; Petit, P. X.; Zamzami, N.; Mignotte, B.; Kroemer, G.; Qian, T.; Herman, B.; Lemasters, J.; Bernardi, P.; Broekemeier, K. M.; Pfeiffer, D. R.; Marzo, I.; Ichas, F.; Jouaville, L. S.; Metivier, D.; Wyllie, a H.; Kerr, J. F. R.; Currie, a R.; Krajewski, S.; Hengartner, M. O.; Horvitz, H. R.; Marchetti, P.; Kane, D. J.; Hurt, K. J.; Baffy, G.; Miyashita, T.; Williamson, J. R. *Science* **1998**, *281* (August), 1312–1316.

(88) Monahan, P.; Di Paola, J. *Semin. Thromb. Hemost.* **2010**, *36* (5), 498–509.

(89) Pampalakis, G.; Sotiropoulou, G. *Biochim. Biophys. Acta - Rev. Cancer* **2007**, *1776* (1), 22–31.

(90) Rice, S. *Nat.Med* **2014**, *19* (7), 837–849.

(91) Kerekatte, V.; Keiper, B. D.; Badorff, C.; Cai, A.; Knowlton, K. U.; Rhoads, R. E. *J. Virol.* **1999**, *73*, 709–717.

(92) Craik, C. S.; Page, M. J.; Madison, E. L. *Biochem. J.* **2011**, *435* (1), 1–16.

(93) Newman, J. R. S.; Keating, A. E. *Science* **2003**, *300* (5628), 2097–2101.

(94) Havranek, J. J.; Harbury, P. B. *Nat. Struct. Biol.* **2002**, *10*, 45–52.

(95) King, C. A.; Bradley, P. *Cancer Res.* **2010**, 3437–3449.

(96) Wollacott, A. M.; Desjarlais, J. R. *J. Mol. Biol.* **2001**, *313* (2), 317–342.

(97) Grigoryan, G.; Reinke, A. W.; Keating, A. E. *Nature* **2009**, *458* (7240), 859–864.

(98) Felder, S.; Zhou, M.; Hu, P.; Urena, J.; Ullrich, A.; Chaudhuri, M.; White, M.; Shoelson, S. E.; Schlessinger, J. *Mol. Cell. Biol.* **1993**, *13* (3), 1449–1455.

(99) Waksman, G.; Shoelson, S. E.; Pant, N.; Cowburn, D.; Kuriyan, J. *Cell* **1993**, *72*, 779–790.

(100) Domchek, S. M.; Auger, K. R.; Chatterjee, S.; Burke, T. R.; Shoelson, S. E. *Biochemistry* **1992**, *31*, 9865–9870.

(101) Ubersax, J. A.; Ferrell, J. E. *Nat. Rev. Mol. Cell Biol.* **2007**, *8*, 530–541.

(102) Lundegaard, C.; Lund, O.; Buus, S.; Nielsen, M. *Immunology* **2010**, *130* (3), 309–318.

(103) Pethe, M. A.; Rubenstein, A. B.; Khare, S. D. *J. Mol. Biol.* **2017**, *429* (2), 220–236.

(104) Raveh, B.; London, N.; Schueler-Furman, O. *Proteins Struct. Funct. Bioinforma.* **2010**, *78* (9), 2029–2040.

(105) Smith, C. A.; Kortemme, T. *J. Mol. Biol.* **2008**, *380* (4), 742–756.

(106) Heaslet, H.; Rosenfeld, R.; Giffin, M.; Lin, Y. C.; Tam, K.; Torbett, B. E.; Elder, J. H.; McRee, D. E.; Stout, C. D. *Acta Crystallogr. Sect. D Biol. Crystallogr.* **2007**, *63* (8), 866–875.

(107) p Version 1.8.0.3, Schrodinger, LLC.

(108) Dunbrack, R. *Curr. Opin. Struct. Biol.* **2002**, *12* (4), 431–440.

(109) Watkins, A. M.; Bonneau, R.; Arora, P. S. *J. Am. Chem. Soc.* **2016**, *138*, 10386–10389.

(110) Zheng, F.; Jewell, H.; Fitzpatrick, J.; Zhang, J.; Mierke, D. F.; Grigoryan, G. *J. Mol. Biol.* **2015**, *427* (2), 491–510.

(111) Chen, Q.; Niu, X.; Xu, Y.; Wu, J.; Shi, Y. *Protein Sci.* **2007**, *16* (6), 1053–1062.

(112) Fujiwara, Y.; Goda, N.; Tamashiro, T.; Narita, H.; Satomura, K.; Tenno, T.; Nakagawa, A.; Oda, M.; Suzuki, M.; Sakisaka, T.; Takai, Y.; Hiroaki, H. *Protein Sci.* **2015**, *24* (3), 376–385.

(113) Vita, R.; Overton, J. A.; Greenbaum, J. A.; Ponomarenko, J.; Clark, J. D.; Cantrell, J. R.; Wheeler, D. K.; Gabbard, J. L.; Hix, D.; Sette, A.; Peters, B. *Nucleic Acids Res.* **2015**, *43* (D1), D405–D412.

(114) Khare, S. D.; Fleishman, S. J. *FEBS Lett.* **2013**, *587* (8), 1147–1154.

(115) Ruggles, S. W.; Fletterick, R. J.; Craik, C. S. *J. Biol. Chem.* **2004**, *279* (29), 30751–30759.

(116) Leaver-Fay, A.; Jacak, R.; Stranges, P. B.; Kuhlman, B. *PLoS One* **2011**, *6* (7).

(117) Shapovalov, M. V.; Dunbrack, R. L. *Structure* **2011**, *19* (6), 844–858.

(118) Park, H.; Bradley, P.; Greisen, P.; Liu, Y.; Mulligan, V. K.; Kim, D. E.; Baker, D.; Dimaio, F. *J. Chem. Theory Comput.* **2016**, *12* (12), 6201–6212.

(119) Saro, D.; Martin, P.; Vickrey, J. F.; Griffin, A.; Kovari, L. C.; Spaller, M. R. *To be Publ.*

(120) Madhusudan; Akamine, P.; Xuong, N.-H.; Taylor, S. S. *Nat. Struct. Mol. Biol.* **2002**, *9* (4), 273–277.

(121) Wu, X.; Knudsen, B.; Feller, S. M.; Zheng, J.; Sali, A.; Cowburn, D.; Hanafusa, H.; Kuriyan, J. *Structure* **1995**, *3* (2), 215–226.

(122) Elkins, J. M.; Papagrigoriou, E.; Berridge, G.; Yang, X.; Phillips, C.; Gileadi, C.; Savitsky, P.; Doyle, D. A. *Protein Sci.* **2007**, *16*, 683–694.

(123) Skelton, N. J.; Koehler, M. F. T.; Zobel, K.; Wong, W. L.; Yeh, S.; Pisabarro, M.

T.; Yin, J. P.; Lasky, L. A.; Sidhu, S. S. *J. Biol. Chem.* **2003**, *278* (9), 7645–7654.

(124) Appleton, B. A.; Zhang, Y.; Wu, P.; Yin, J. P.; Hunziker, W.; Skelton, N. J.; Sidhu, S. S.; Wiesmann, C. *J. Biol. Chem.* **2006**, *281* (31), 22312–22320.

(125) Zhang, Y.; Dasgupta, J.; Ma, R. Z.; Banks, L.; Thomas, M.; Chen, X. S. *J. Virol.* **2007**, *81* (7), 3618–3626.

(126) Ding, Y. H.; Baker, B. M.; Garboczi, D. N.; Biddison, W. E.; Wiley, D. C. *Immunity* **1999**, *11* (1), 45–56.

(127) Røder, G.; Blicher, T.; Justesen, S.; Johannesen, B.; Kristensen, O.; Kastrup, J.; Buus, S.; Gajhede, M. *Acta Crystallogr. Sect. D Biol. Crystallogr.* **2006**, *62* (11), 1300–1310.

(128) Macdonald, W. A.; Purcell, A. W.; Mifsud, N. A.; Ely, L. K.; Williams, D. S.; Chang, L.; Gorman, J. J.; Clements, C. S.; Kjer-Nielsen, L.; Koelle, D. M.; Burrows, S. R.; Tait, B. D.; Holdsworth, R.; Brooks, A. G.; Lovrecz, G. O.; Lu, L.; Rossjohn, J.; McCluskey, J. *J. Exp. Med.* **2003**, *198* (5), 679–691.

(129) Cummings, M. D.; Lindberg, J.; Lin, T. I.; De Kock, H.; Lenz, O.; Lilja, E.; Feiländer, S.; Baraznenok, V.; Nyström, S.; Nilsson, M.; Vrang, L.; Edlund, M.; Rosenquist, Å.; Samuelsson, B.; Raboisson, P.; Simmen, K. *Angew. Chemie - Int. Ed.* **2010**, *49* (9), 1652–1655.

(130) Dinkel, H.; Chica, C.; Via, A.; Gould, C. M.; Jensen, L. J.; Gibson, T. J.; Diella, F. *Nucleic Acids Res.* **2011**, *39* (SUPPL. 1), D261-7.

(131) Gong, W.; Zhou, D.; Ren, Y.; Wang, Y.; Zuo, Z.; Shen, Y.; Xiao, F.; Zhu, Q.; Hong, A.; Zhou, X.; Gao, X.; Li, T. *Nucleic Acids Res* **2008**, *36* (Database issue), D679-83.

(132) Beuming, T.; Skrabanek, L.; Niv, M. Y.; Mukherjee, P.; Weinstein, H. *Bioinformatics* **2005**, *21* (6), 827–828.

(133) Koehl, P.; Delarue, M. *J. Mol. Biol.* **1994**, *239* (2), 249–275.

(134) Delarue, M.; Koehl, P. *Pac.Symp.Biocomput.* **1997**, 109.

(135) Lee, C. *J. Mol. Biol.* **1994**, *236* (3), 918–939.

(136) Voigt, C. A.; Mayo, S. L.; Arnold, F. H.; Wang, Z.-G. *Proc. Natl. Acad. Sci.* **2001**, *98* (7), 3778–3783.

(137) Saven, J. G.; Wolynes, P. G. *J. Phys. Chem. B* **1997**, *101* (41), 8375–8389.

(138) Xiao, X.; Hall, C. K.; Agris, P. F. *J. Biomol. Struct. Dyn.* **2014**, *32* (10), 1523–1536.

(139) Mendes, J.; Soares, C. M.; Carrondo, M. A. *Biopolymers* **1999**, *50* (2), 111–131.

(140) Kono, H. *J. Comput. Chem.* **1996**, *17* (14), 1667–1683.

(141) Leaver-Fay, A.; Kuhlman, B.; Snoeyink, J. *Pacific Symp. Biocomput.* **2005**, 16–27.

(142) London, N.; Movshovitz-Attias, D.; Schueler-Furman, O. *Structure* **2010**, *18* (2), 188–199.

(143) Domingo, E.; Holland, J. J. *Annu. Rev. Microbiol.* **1997**, *51* (1), 151–178.

(144) Holland, J.; Spindler, K.; Horodyski, F.; Grabau, E.; Nichol, S.; VandePol, S. *Science* **1982**, *215* (4540), 1577–1585.

(145) Lauring, A. S.; Frydman, J.; Andino, R. *Nat. Rev. Microbiol.* **2013**, *11* (5), 327–336.

(146) Andino, R.; Domingo, E. *Virology*. 2015, pp 46–51.

(147) Eigen, M. *Sci. Am.* **1993**, *269* (1), 42–49.

(148) Cristina, J.; del Pilar Moreno, M.; Moratorio, G. *Virus Res* **2007**, *127* (2), 185–194.

(149) Elde, N. C.; Child, S. J.; Eickbush, M. T.; Kitzman, J. O.; Rogers, K. S.; Shendure, J.; Geballe, A. P.; Malik, H. S. *Cell* **2012**, *150* (4), 831–841.

(150) Goldberg, D. E.; Siliciano, R. F.; Jacobs Jr., W. R. *Cell* **2012**, *148* (6), 1271–1283.

(151) Wilke, C. O.; Wang, J. L.; Ofria, C.; Lenski, R. E.; Adami, C. *Nature* **2001**, *412* (6844), 331–333.

(152) Eigen, M. *Proc Natl Acad Sci U S A* **2002**, *99* (21), 13374–13376.

(153) Lauring, A. S.; Andino, R.; Boone, C.; Holden, D. W.; Liu, T. *PLoS Pathog.* **2010**, *6* (7), e1001005.

(154) Elena, S. F.; Carrasco, P.; Daros, J. A.; Sanjuan, R. *EMBO Rep* **2006**, *7* (2), 168–173.

(155) Masel, J.; Siegal, M. L. *Trends Genet* **2009**, *25* (9), 395–403.

(156) Tokuriki, N.; Oldfield, C. J.; Uversky, V. N.; Berezovsky, I. N.; Tawfik, D. S. *Trends Biochem Sci* **2009**, *34* (2), 53–59.

(157) Draghi, J. A.; Parsons, T. L.; Wagner, G. P.; Plotkin, J. B. *Nature* **2010**, *463* (7279), 353–355.

(158) Wilke, C. O.; Adami, C. *Mutat. Res. Mol. Mech. Mutagen.* **2003**, *522* (1), 3–11.

(159) Smith, J. M. *Nature* **1970**, *225* (5232), 563–564.

(160) de Visser, J. A.; Krug, J. *Nat Rev Genet* **2014**, *15* (7), 480–490.

(161) Wright, S. *Genetics* **1931**, *16* (2), 97–159.

(162) Harms, M. J.; Thornton, J. W. *Nat. Rev. Genet.* **2013**, *14* (8), 559–571.

(163) Bridgham, J. T. *Science* **2006**, *312* (5770), 97–101.

(164) Kondrashov, D. A.; Kondrashov, F. A. *Trends Genet* **2015**, *31* (1), 24–33.

(165) Romero, P. A.; Arnold, F. H. *Nat Rev Mol Cell Biol* **2009**, *10* (12), 866–876.

(166) Weinreich, D. M.; Delaney, N. F.; Depristo, M. A.; Hartl, D. L. *Science* **2006**, *312* (5770), 111–114.

(167) Bandaru, P.; Shah, N. H.; Bhattacharyya, M.; Barton, J. P.; Kondo, Y.; Cofsky, J. C.; Gee, C. L.; Chakraborty, A. K.; Kortemme, T.; Ranganathan, R.; Kuriyan, J. *Elife* **2017**, *6*.

(168) Firnberg, E.; Labonte, J. W.; Gray, J. J.; Ostermeier, M. *Mol Biol Evol* **2014**, *31* (6), 1581–1592.

(169) Fowler, D. M.; Araya, C. L.; Fleishman, S. J.; Kellogg, E. H.; Stephany, J. J.; Baker, D.; Fields, S. *Nat. Methods* **2010**, *7* (9), 741--U108.

(170) Hietpas, R. T.; Jensen, J. D.; Bolon, D. N. A. *Proc. Natl. Acad. Sci. U. S. A.* **2011**, *108* (19), 7896–7901.

(171) Kim, I.; Miller, C. R.; Young, D. L.; Fields, S. *Mol. Cell. Proteomics* **2013**, *12* (11), 3370–3378.

(172) McLaughlin Jr., R. N.; Poelwijk, F. J.; Raman, A.; Gosal, W. S.; Ranganathan, R. *Nature* **2012**, *491* (7422), 138–142.

(173) Podgornaia, A. I.; Laub, M. T. *Science* **2015**, *347* (6222), 673–677.

(174) Sarkisyan, K. S.; Bolotin, D. A.; Meer, M. V; Usmanova, D. R.; Mishin, A. S.; Sharonov, G. V; Ivankov, D. N.; Bozhanova, N. G.; Baranov, M. S.; Soylemez, O.; Bogatyreva, N. S.; Vlasov, P. K.; Egorov, E. S.; Logacheva, M. D.; Kondrashov, A. S.; Chudakov, D. M.; Putintseva, E. V; Mamedov, I. Z.; Tawfik, D. S.; Lukyanov, K. A.; Kondrashov, F. A. *Nature* **2016**, *533* (7603), 397–401.

(175) Wrenbeck, E. E.; Azouz, L. R.; Whitehead, T. A. *Nat Commun* **2017**, *8*, 15695.

(176) Breen, M. S.; Kemena, C.; Vlasov, P. K.; Notredame, C.; Kondrashov, F. A. *Nature* **2012**, *490* (7421), 535–538.

(177) Sailer, Z. R.; Harms, M. J. *Genetics* **2017**.

(178) Blanquart, F.; Bataillon, T. *Genetics* **2016**, *203* (2), 847–862.

(179) Hartl, D. L. *Curr Opin Microbiol* **2014**, *21*, 51–57.

(180) Thyagarajan, B.; Bloom, J. D. *Elife* **2014**, *3*.

(181) Weinreich, D. M.; Lan, Y.; Wylie, C. S.; Heckendorn, R. B. *Curr Opin Genet Dev* **2013**, *23* (6), 700–707.

(182) Wu, N. C.; Dai, L.; Olson, C. A.; Lloyd-Smith, J. O.; Sun, R. *Elife* **2016**, *5*.

(183) Jenson, J. M.; Ryan, J. A.; Grant, R. A.; Letai, A.; Keating, A. E. *Elife* **2017**, *6*.

(184) Klesmith, J. R.; Bacik, J. P.; Michalczyk, R.; Whitehead, T. A. *ACS Synth Biol* **2015**, *4* (11), 1235–1243.

(185) Tinberg, C. E.; Khare, S. D.; Dou, J.; Doyle, L.; Nelson, J. W.; Schena, A.; Jankowski, W.; Kalodimos, C. G.; Johnsson, K.; Stoddard, B. L.; Baker, D. *Nature* **2013**, *501* (7466), 212–216.

(186) Whitehead, T. A.; Chevalier, A.; Song, Y.; Dreyfus, C.; Fleishman, S. J.; De Mattos, C.; Myers, C. A.; Kamisetty, H.; Blair, P.; Wilson, I. A.; Baker, D. *Nat. Biotechnol.* **2012**, *30* (6), 543–548.

(187) Fowler, D. M.; Fields, S. *Nat Methods* **2014**, *11* (8), 801–807.

(188) Klesmith, J. R.; Bacik, J. P.; Wrenbeck, E. E.; Michalczyk, R.; Whitehead, T. A. *Proc Natl Acad Sci U S A* **2017**, *114* (9), 2265–2270.

(189) Reich, L. L.; Dutta, S.; Keating, A. E. *J Mol Biol* **2015**, *427* (11), 2135–2150.

(190) Aakre, C. D.; Herrou, J.; Phung, T. N.; Perchuk, B. S.; Crosson, S.; Laub, M. T. *Cell* **2015**, *163* (3), 594–606.

(191) Rodrigues, J. V; Bershtein, S.; Li, A.; Lozovsky, E. R.; Hartl, D. L.; Shakhnovich, E. I. *Proc Natl Acad Sci U S A* **2016**, *113* (11), E1470-8.

(192) Sailer, Z. R.; Harms, M. J.; Dean, A. M.; Usmanova, D. R.; Mishin, A. S.; Sharonov, G. V. *PLOS Comput. Biol.* **2017**, *13* (5), e1005541.

(193) Bloom, J. D.; Wilke, C. O.; Arnold, F. H.; Adami, C. *Biophys. J.* **2004**, *86* (5), 2758–2764.

(194) Bornberg-Bauer, E.; Chan, H. S. *Proc. Natl. Acad. Sci.* **1999**, *96* (19), 10689–10694.

(195) DePristo, M. A.; Weinreich, D. M.; Hartl, D. L. *Nat Rev Genet* **2005**, *6* (9), 678–687.

(196) Ding, F.; Dokholyan, N. V. *PLoS Comput Biol* **2006**, *2* (7), e85.

(197) Drummond, D. A.; Wilke, C. O. *Cell* **2008**, *134* (2), 341–352.

(198) Echave, J.; Wilke, C. O. *Annu Rev Biophys* **2017**, *46*, 85–103.

(199) Manhart, M.; Morozov, A. V. *Proc Natl Acad Sci U S A* **2015**, *112* (6), 1797–1802.

(200) Sikosek, T.; Chan, H. S. *J R Soc Interface* **2014**, *11* (100), 20140419.

(201) van Nimwegen, E.; Crutchfield, J. P.; Huynen, M. *Proc Natl Acad Sci U S A* **1999**, *96* (17), 9716–9720.

(202) Yang, J. R.; Liao, B. Y.; Zhuang, S. M.; Zhang, J. *Proc Natl Acad Sci U S A* **2012**, *109* (14), E831-40.

(203) Bershtein, S.; Serohijos, A. W.; Shakhnovich, E. I. *Current Opinion in Structural Biology*. 2017, pp 31–40.

(204) Serohijos, A. W.; Shakhnovich, E. I. *Curr Opin Struct Biol* **2014**, *26*, 84–91.

(205) Meylan, E.; Curran, J.; Hofmann, K.; Moradpour, D.; Binder, M.; Bartenschlager, R.; Tschopp, J. *Nature* **2005**, *437* (7062), 1167–1172.

(206) Powdrill, M. H.; Tchesnokov, E. P.; Kozak, R. A.; Russell, R. S.; Martin, R.; Svarovskaia, E. S.; Mo, H.; Kouyos, R. D.; Gotte, M. *Proc Natl Acad Sci U S A* **2011**, *108* (51), 20509–20513.

(207) Rubenstein, A. B.; Pethe, M. A.; Khare, S. D.; Wang, Z.-G.; Weinstein, H.; Shen, Y. *PLOS Comput. Biol.* **2017**, *13* (6), e1005614.

(208) Geller, R.; Estada, U.; Peris, J. B.; Andreu, I.; Bou, J. V; Garijo, R.; Cuevas, J. M.; Sabariegos, R.; Mas, A.; Sanjuan, R. *Nat. Microbiol.* **2016**, *1* (7).

(209) Schechter, I.; Berger, A. *Biochem. Biophys. Res. Commun.* **1967**, *27* (2), 157–162.

(210) Benatuil, L.; Perez, J. M.; Belk, J.; Hsieh, C.-M. *Protein Eng. Des. Sel.* **2010**, *23* (4), 155–159.

(211) Kowalsky, C. A.; Klesmith, J. R.; Stapleton, J. A.; Kelly, V.; Reichkitzer, N.; Whitehead, T. A. *PLoS One* **2015**, *10* (3), 1–23.

(212) Li, Z.; Zhang, Y.; Liu, Y.; Shao, X.; Luo, Q.; Cai, Q.; Zhao, Z. *Medicine (Baltimore)*. **2017**, *96* (19), e6830.

(213) Amat, C. B. *Rev Esp Doc Cient* **2016**, *39*.

(214) Jacomy, M.; Venturini, T.; Heymann, S.; Bastian, M. *PLoS One* **2014**, *9* (6), e98679.

(215) Brin, S.; Page, L. *Comput. Networks ISDN Syst.* **1998**, *30* (1–7), 107–117.

(216) Kelly, M. A.; Chellgren, B. W.; Rucker, A. L.; Troutman, J. M.; Fried, M. G.; Miller, A. F.; Creamer, T. P. *Biochemistry* **2001**, *40* (48), 14376–14383.

(217) Vila, J. A.; Baldoni, H. A.; Ripoll, D. R.; Ghosh, A.; Scheraga, H. A. *Biophys. J.* **2004**, *86* (2), 731–742.

(218) Campo, D. S.; Dimitrova, Z.; Mitchell, R. J.; Lara, J.; Khudyakov, Y. *Proc Natl Acad Sci U S A* **2008**, *105* (28), 9685–9690.

(219) Cuypers, L.; Li, G.; Libin, P.; Piampongsant, S.; Vandamme, A.-M.; Theys, K. *Viruses* **2015**, *7* (9), 5018–5039.

(220) Skums, P.; Bunimovich, L.; Khudyakov, Y. *Proc Natl Acad Sci U S A* **2015**, *112* (21), 6653–6658.

(221) Fuchs, J. E.; von Grafenstein, S.; Huber, R. G.; Wallnoefer, H. G.; Liedl, K. R. *Proteins* **2014**, *82* (4), 546–555.

(222) Farci, P.; Shimoda, A.; Coiana, A.; Diaz, G.; Peddis, G.; Melpolder, J. C.; Strazzera, A.; Chien, D. Y.; Munoz, S. J.; Balestrieri, A.; Purcell, R. H.; Alter, H. J. *Science* **2000**, *288* (5464), 339–344.

(223) Boucher, J. I.; Bolon, D. N. A.; Tawfik, D. S. *Protein Science*. 2016, pp 1219–1226.

(224) Dickinson, B. C.; Packer, M. S.; Badran, A. H.; Liu, D. R. *Nat Commun* **2014**, *5*, 5352.

(225) Dam, E.; Quercia, R.; Glass, B.; Descamps, D.; Launay, O.; Duval, X.; Krausslich, H. G.; Hance, A. J.; Clavel, F.; Group, A. S. *PLoS Pathog* **2009**, *5* (3), e1000345.

(226) Anfinsen, C. B. *Science* **1973**, *181* (4096), 223–230.

(227) Lu, H.; Skolnick, J. *Proteins Struct. Funct. Genet.* **2001**, *44* (3), 223–232.

(228) 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. *J. Comput. Chem.* **2009**, *30* (10), 1545–1614.

(229) Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz, K. M.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A. *J. Am. Chem. Soc.* **1995**, *117* (19), 5179–5197.

(230) Jernigan, R. L.; Bahar, I. *Current Opinion in Structural Biology*. 1996, pp 195–209.

(231) Shen, M.-Y.; Sali, A. *Protein Sci.* **2006**, *15* (11), 2507–2524.

(232) Ponder, J. W.; Case, D. A. *Advances in Protein Chemistry*. 2003, pp 27–85.

(233) Simons, K. T.; Kooperberg, C.; Huang, E.; Baker, D. *J. Mol. Biol.* **1997**, *268* (1), 209–225.

(234) Jorgensen, W. L.; Maxwell, D. S.; Tirado-Rives, J. *J. Am. Chem. Soc.* **1996**, *118* (45), 11225–11236.

(235) Maier, J. A.; Martinez, C.; Kasavajhala, K.; Wickstrom, L.; Hauser, K. E.; Simmerling, C. *J. Chem. Theory Comput.* **2015**, *11* (8), 3696–3713.

(236) MacKerell, A. D.; Bashford, D.; Bellott, M.; Dunbrack, R. L.; Evanseck, J. D.; Field, M. J.; Fischer, S.; Gao, J.; Guo, H.; Ha, S.; Joseph-McCarthy, D.; Kuchnir, L.; Kuczera, K.; Lau, F. T. K.; Mattos, C.; Michnick, S.; Ngo, T.; Nguyen, D. T.; Prodhom, B.; Reiher, W. E.; Roux, B.; Schlenkrich, M.; Smith, J. C.; Stote, R.; Straub, J.; Watanabe, M.; Wiórkiewicz-Kuczera, J.; Yin, D.; Karplus, M. *J. Phys. Chem. B* **1998**, *102* (18), 3586–3616.

(237) Xu, D.; Zhang, Y. *Proteins Struct. Funct. Bioinforma.* **2012**, *80* (7), 1715–1735.

(238) O'Meara, M. J.; Leaver-Fay, A.; Tyka, M. D.; Stein, A.; Houlihan, K.; Dimaio, F.; Bradley, P.; Kortemme, T.; Baker, D.; Snoeyink, J.; Kuhlman, B. *J. Chem. Theory Comput.* **2015**, *11* (2), 609–622.

(239) Best, R. B.; Zhu, X.; Shim, J.; Lopes, P. E. M.; Mittal, J.; Feig, M.; MacKerell, A. D. *J. Chem. Theory Comput.* **2012**, *8* (9), 3257–3273.

(240) Raval, A.; Piana, S.; Eastwood, M. P.; Dror, R. O.; Shaw, D. E. *Proteins Struct. Funct. Bioinforma.* **2012**, *80* (8), 2071–2079.

(241) Lindorff-Larsen, K.; Maragakis, P.; Piana, S.; Eastwood, M. P.; Dror, R. O.; Shaw, D. E. *PLoS One* **2012**, *7* (2).

(242) Alford, R. F.; Leaver-Fay, A.; Jeliazkov, J. R.; O'Meara, M. J.; DiMaio, F. P.; Park, H.; Shapovalov, M. V.; Renfrew, P. D.; Mulligan, V. K.; Kappel, K.; Labonte, J. W.; Pacella, M. S.; Bonneau, R.; Bradley, P.; Dunbrack, R. L.; Das, R.; Baker, D.; Kuhlman, B.; Kortemme, T.; Gray, J. J. *J. Chem. Theory Comput.* **2017**, *13* (6), 3031–3048.

(243) Park, H.; Bradley, P.; Greisen, P.; Liu, Y.; Mulligan, V. K.; Kim, D. E.; Baker, D.; DiMaio, F. *J. Chem. Theory Comput.* **2016**, *12* (12), 6201–6212.

(244) Beauchamp, K. A.; Lin, Y. S.; Das, R.; Pande, V. S. *J. Chem. Theory Comput.* **2012**, *8* (4), 1409–1414.

(245) Cino, E. A.; Choy, W. Y.; Karttunen, M. *J. Chem. Theory Comput.* **2012**, *8* (8), 2725–2740.

(246) Nguyen, H.; Roe, D. R.; Simmerling, C. *J. Chem. Theory Comput.* **2013**, *9* (4), 2020–2034.

(247) Nguyen, H.; Maier, J.; Huang, H.; Perrone, V.; Simmerling, C. *J. Am. Chem. Soc.* **2014**, *136* (40), 13959–13962.

(248) Dill, K. a; MacCallum, J. L. *Science* **2012**, *338* (6110), 1042–1046.

(249) Wolynes, P. G.; Onuchic, J. N.; Thirumalai, D. *Science* **1995**, *267* (5204), 1619–

1620.

(250) Shakhnovich, E. *Chemical Reviews*. 2006, pp 1559–1588.

(251) Tyka, M. D.; Keedy, D. A.; Andre, I.; Dimaio, F.; Song, Y.; Richardson, D. C.; Richardson, J. S.; Baker, D. *J. Mol. Biol.* **2011**, *405* (2), 607–618.

(252) Rafferty, B.; Flohr, Z. C.; Martini, A. 2014.

(253) Case, D. A.; Betz, R. M.; Cerutti, D. S.; Cheatham III, T. E.; Darden, T. A.; Duke, R. E.; Giese, T. J.; Gohlke, H.; Goetz, A. W.; Homeyer, N.; Izadi, S.; Janowski, P.; Kaus, J.; Kovalenko, A.; Lee, T. S.; LeGrand, S.; Li, P.; Lin, C.; Luchko, T.; Luo, R.; Madej, B.; Mermelstein, D.; Merz, K. M.; Monard, G.; Nguyen, H.; Nguyen, H. T.; Omelyan, I.; Onufriev, A.; Roe, D. R.; Roitberg, A.; Sagui, C.; Simmerling, C. L.; Botello-Smith, W. M.; Swails, J.; Walker, R. C.; Wang, J.; Wolf, R. M.; Wu, X.; Xiao, L.; Kollman, P. A. *AMBER 2016*; University of California: San Francisco, 2016.

(254) Xiang, Z.; Soto, C. S.; Honig, B. *Proc. Natl. Acad. Sci. U. S. A.* **2002**, *99*, 7432–7437.

(255) Fiser, A.; Kinh Gian Do, R.; Sali. *Protein Sci.* **2000**, *9*, 1753–1773.

(256) Murphy, P. M.; Bolduc, J. M.; Gallaher, J. L.; Stoddard, B. L.; Baker, D. *Proc. Natl. Acad. Sci. U. S. A.* **2009**, *106* (23), 9215–9220.

(257) Wang, C.; Bradley, P.; Baker, D. *J. Mol. Biol.* **2007**, *373* (2), 503–519.

(258) Mandell, D. J.; Coutsias, E. a; Kortemme, T. *Nat. Methods* **2009**, *6* (8), 551–552.

(259) Ollikainen, N.; Smith, C. A.; Fraser, J. S.; Kortemme, T. *Methods Enzymol.* **2013**, *18* (9), 1199–1216.

(260) Stein, A.; Kortemme, T. *PLoS One* **2013**, *8* (5).

(261) Conway, P.; Tyka, M. D.; DiMaio, F.; Konerding, D. E.; Baker, D. *Protein Sci.* **2014**, *23* (1), 47–55.

(262) Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. *Nucleic Acids Res.* **2000**, *28*, 235–242.

(263) Tyka, M. D.; Jung, K.; Baker, D. *J. Comput. Chem.* **2012**, *33* (31), 2483–2491.

(264) Liu, D. C.; Nocedal, J. *Math. Program.* **1989**, *45* (1–3), 503–528.

(265) Nguyen, H.; Roe, D. R.; Swails, J. M.; Case, D. A. *Manuscr. Prep.* **2017**.

(266) Wittekind, M.; Weinheimer, S.; Zhang, Y.; Goldfarb, V. Modified forms of hepatitis C NS3 protease for facilitating inhibitor screening and structural studies of protease:inhibitor complexes, 2001.

(267) Gallinari, P.; Brennan, D.; Nardi, C.; Brunetti, M.; Tomei, L.; Steinkuhler, C.; De Francesco, R. *J Virol* **1998**, *72* (8), 6758–6769.

(268) Bastian, M.; Heymann, S.; Jacomy, M. *Third Int. AAAI Conf. Weblogs Soc. Media* **2009**, 361–362.

(269) Buslje, C. M.; Santos, J.; Delfino, J. M.; Nielsen, M. *Bioinformatics* **2009**, *25* (9), 1125–1131.

(270) Gouveia-Oliveira, R.; Pedersen, A. G. *Algorithms Mol. Biol.* **2007**, *2*, 12.