Chapter 15

Design of Specific Peptide–Protein Recognition

Fan Zheng and Gevorg Grigoryan

Abstract

Selective targeting of protein–protein interactions in the cell is of great interest in biological research. Computational structure-based design of peptides to bind protein interaction interfaces could provide a potential means of generating such reagents. However, to avoid perturbing off-target interactions, methods that explicitly account for interaction specificity are needed. Further, as peptides often retain considerable flexibility upon association, their binding reaction is computationally demanding to model—a stark limitation for structure-based design. Here we present a protocol for designing peptides that selectively target a given peptide-binding domain, relative to a pre-specified set of possibly related domains. We recently used the method to design peptides that discriminate with high selectivity between two closely related PDZ domains. The framework accounts for the flexibility of the peptide in the binding site, but is efficient enough to quickly analyze trade-offs between affinity and selectivity, enabling the identification of optimal peptides.

Key words Interaction specificity, Computational protein design, PDZ–peptide interactions, Cluster expansion, Flexible peptide docking

1 Introduction

The loss of precise control over cellular protein interactions often results in disease [1]. Therefore, reagents that target protein interactions to rewire cellular signaling pathways in desired ways are of great relevance in both therapeutic development and mechanistic investigation [2]. A considerable fraction of the known cellular interactome is believed to be mediated by peptide-recognition domains (PRDs)—interaction-encoding modules that bind to short amino-acid stretches on their partner proteins [3–5]. Many PRD families are large, with members closely related in structure and sequence, but often having entirely divergent functions. Peptides are a natural choice for functional modulation of PRD-encoded interactions, because they are well suited to occupy the PRD binding site and are amenable to computational design. Further, recognition sequence preferences of several PRDs have

Barry L. Stoddard (ed.), *Computational Design of Ligand Binding Proteins*, Methods in Molecular Biology, vol. 1414,
DOI 10.1007/978-1-4939-3569-7_15,

been characterized experimentally [6–9], enabling the development of computational models for binding prediction either by direct training on high-throughput experimental data [10, 11], structure-based energy calculations [12–15], or combinations of the two [16, 17]. However, to effectively target a given interaction encoded by a PRD, the targeting peptide should in general be selective—i.e., it should avoid interactions with other proteins, including those within the same PRD family. Given the close similarity among family members, achieving such selectivity by design is not trivial. Peptides chosen purely for binding to the target are likely to also bind other family members, with unpredictable functional consequences.

Structure-based methods for modeling PRD–peptide binding have the potential to generalize across different PRDs [18]. However, the use of such techniques in designing selective recognition is complicated by the inherent flexibility of peptides, which places high computational demands on modeling. To mitigate this problem, we have developed a general computational framework that decouples the complexity of the structure-based simulation used to model PRD–peptide binding from the computational efficiency requirements imposed by the design of selectivity [19]. The framework uses the previously described method of cluster expansion (CE) [20, 21] to produce simple sequence-based expressions that rapidly estimate the results of detailed structure modeling techniques. The efficiency gained by CE enables the fast identification of optimal trade-offs between affinity for the targeted domain and selectivity against any number of undesired partners. The framework is detailed below.

2 Materials

The following resources or materials are needed to apply our framework:

1. A Unix-/Linux-based computing platform with:
   (a) A linear algebra engine (e.g., the proprietary MathWorks MATLAB or the open-source GNU Octave).
   (b) Macromolecular modeling suite Rosetta, version 3.4 or higher [22].
   (c) PyRosetta, a Python-based interface to Rosetta [23].
   (d) Highly desirable: access to a high-performance computing cluster with the ability to perform at least hundreds of jobs independently in parallel.
2. A basic understanding of and the capability to work with the computation resources in 1.

3. Optional, but highly desirable: experimentally validated examples of peptides that bind strongly and those that bind weakly (or undetectably) to members of the PRD family of interest.

3 Methods

In this section, we outline our framework for designing PRD-binding peptides. We will refer to our experience with using it to design PDZ-targeting peptides [19], but we believe that the framework should generalize to other systems. The procedure differs depending on whether the goal is to design high-affinity peptides for a single PRD or design selective peptides that bind one PRD (target) but not the others (competitors). In the latter case, binding to multiple PRDs has to be modeled. If not stated otherwise, it should be assumed that each discussed step is carried out for all PRDs being considered.

1. Download experimental structures of target and competitor PRDs from the Protein Data Bank (PDB), if they are available. The following preferences apply if multiple structures are available for a given PRD (in the order of priority): (a) an X-ray structure is preferred over an NMR structure, (b) a peptide-bound structure is preferred over an *apo* structure, and (c) a higher-resolution X-ray structure is preferred over a lower resolution one. If a given PRD has no experimental structures, use homology modeling (e.g., via the SWISS-MODEL server [24] or MODELLER [25]) to create a predicted structure. The template used in homology modeling should be a peptide-bound structure and otherwise as close in sequence to the relevant PRD as possible (*see* **Note 1**). If either an NMR structure or a homology model is used for a PRD, particular attention should be paid to the results in **step 5**.
2. Subject any homology models to continuous minimization in the presence of a known binding peptide. Because the backbone will be held fixed when sampling the bound state (*see* below), this step is recommended to make the PRD model resemble a peptide-bound state as much as possible. To this end, first align the homology model to the template by optimally superimposing the backbone of binding-site residues, and then copy the peptide backbone from the template to the PRD model. In PyRosetta [23], assign peptide side-chain identities according to a known ligand peptide (a ligand of a closely homologous PRD may be used if no ligand for the target is known) and repack all side chains in the model. Follow by applying full-atom minimization via the "dfpmin" algorithm in PyRosetta, with a tolerance of 0.01, allowing both

backbone torsion angles and side chain χ-angles to move. Note this assumes that the template used in homology modeling is close enough to the PRD of interest to have similar binding geometry and sequence preferences.

3. Collect a set of experimental PRD–peptide complex structures for use in seeding multiple simulation trajectories when modeling new PRD–peptide pairs. For example, for PDZ domains, we collected 51 unique complexes with peptides of at least six residues (Table 1). For each available complex, align its binding site onto that of the PRD of interest, and copy the peptide backbone from the complex onto the PRD (as in **step 2**). To automate the procedure of identifying binding sites in all experimental complexes, we recommend manually defining binding-site residues only in the PRD of interest and then using our substructure search engine MASTER [26] to automatically find corresponding residues in all complexes. We found the generation of diverse starting conformations to seed multiple sampling trajectories to be critical in modeling PDZ–peptide binding, presumably due to the considerable flexibility of the peptide in the binding site [19].
4. Given a peptide/PRD combination to be evaluated, run the Rosetta FlexPepDock ab initio protocol [27] for each of the starting conformations generated in **step 3**. We recommend asking each simulation to generate at least 500 structural models (from 500 independent Monte Carlo simulations). Therefore, in the PDZ example, for each peptide/PRD pair, $500 \times 51 = 25{,}500$ structural models would be generated. Rosetta FlexPepDock documentation is available at https://www.rosettacommons.org/docs/latest/application_documentation/docking/flex-pep-dock. Evaluate each model using the *talaris2013* Rosetta scoring function; in our experience, omitting backbone statistical energy terms "rama" and "omega" increases performance (*see* **Note 2**). The lowest score among all generated models should be used as the final predicted binding score for the given peptide/domain combination.
5. Use an experimental dataset as a benchmark to assess the accuracy of the structure-based simulation and the appropriateness of structural models used. Ideally, experimental data for the relevant PRDs should be used, but if such data are unavailable, results for highly homologous domains in the PRD family (those believed to share close binding preferences) may be used. Use the experimental data to build the benchmark dataset: sets of high-confidence binding peptides and weak/non-binding peptides for each PRD (*see* **Note 3**). Run the procedure in **step 4** to score each peptide/domain combination in the benchmark dataset. Use the Receiver Operating Characteristic (ROC) analysis to measure the ability of the simulation to sep-

Table 1
A set of experimental PDZ–peptide complex structures used to generate starting conformations for multiple simulation trajectories

PDB-ID	Chain-ID (domain)	Domain residue number range	Chain-ID (peptide)
1B8Q	A	11–90	B
1D5G	A	8–90	B
1KWA	A	3–82	B
1L6O	A	3–88	D
1N7F	A	5–84	C
1N7T	A	12–98	B
1OBY	B	2–74	Q
1Q3P	A	8–95	C
1RGR	A	4–88	B
1RZX	A	5–95	B
1TP3	A	13–91	B
1TP5	A	13–91	B
1U3B	A	4–88	A
1VJ6	A	8–90	B
1X8S	A	5–95	B
1YBO	A	88–160	C
1ZUB	A	23–107	B
2AIN	A	7–89	B
2EJY	A	3–81	B
2FNE	B	11–93	A
2HE2	A	7–85	B
2I04	B	3–83	D
2I0I	A	4–81	D
2I0L	A	2–83	C
2I1N	A	6–90	B
2IWP	A	3–83	B
2JIL	A	7–89	B
2JOA	A	5–88	B
2 K20	A	9–99	B
2KA9	A	5–89	B

(continued)

Table 1
(continued)

PDB-ID	Chain-ID (domain)	Domain residue number range	Chain-ID (peptide)
2KBS	A	4–83	B
2KPL	A	17–97	B
2KQF	A	8–91	B
2KYL	A	8–91	B
2L4T	A	17–110	B
2OPG	B	5–87	A
2OQS	A	2–86	B
2OS6	A	11–83	B
2PZD	A	1–85	B
2QBW	A	2–97	B
2UZC	B	3–81	A
2 V90	E	6–85	C
2VRF	B	7–87	A
3B76	B	11–94	A
3CBX	B	7–88	A
3CBY	B	4–86	A
3CC0	C	4–88	A
3CH8	A	2–95	P
3DIW	B	7–100	D
3GGE	A	9–88	B
3LNY	A	8–90	B

This table was created by filtering search results from extended PDZ domain database (http://bcz102.ust.hk/pdzex/) [31]

arate true binders from weak/non-binders, using Area Under the Curve (AUC) for quantification [28]. AUC values above ~0.7 would indicate a reasonable structural model and simulation approach.

6. Define amino acids allowed at each position of the peptide—i.e., the design alphabet. We strongly recommend constraining the alphabet based on any known information about the PRD family in general and the specific targeted domain(s). This keeps the sequence space from being unnecessarily large, limiting computational complexity. Further, patterning of allowed amino acids based on strong experimentally observed preferences limits the effect of error present in any modeling

approach. In our PDZ-targeting study, we were able to design highly selective binders by computationally considering a sequence space of only 8400 peptides [19].

7. Given the computationally complex modeling procedure described in **step 4**, it will likely be prohibitively expensive to enumerate even moderately large peptide sequence spaces (e.g., the procedure takes over 400 CPU hours per peptide in our PDZ example). On the other hand, given a specific PRD, the final score of the simulation depends only on the peptide sequence. Thus, the next step is to derive an analytical mapping from peptide sequence to predicted binding score, for each PRD of interest. We previously described a method for finding such a mapping, called cluster expansion (CE) [20]. In short, CE expresses the result of a structure-based computational procedure as a series expansion in contributions from amino-acid clusters of increasing size—we call these cluster functions or CFs. For example, if $E(\vec{\sigma})$ represents the binding score from the procedure in **step 4**, for a peptide sequence $\vec{\sigma}$ and a given domain, the CE expression states

$$E(\vec{\sigma}) = C + \sum_{\substack{i=1 \\ \sigma_i \neq \rho_i}}^{L} f_i(\sigma_i) + \sum_{\substack{i=1 \\ \sigma_i \neq \rho_i}}^{L-1} \sum_{\substack{j=i+1 \\ \sigma_j \neq \rho_j}}^{L} f_{ij}(\sigma_i, \sigma_j) + \ldots$$

where L is peptide length, $\vec{\rho}$ is a reference sequence, and σi and ρi are the amino acids in the i-th position of $\vec{\sigma}$ and $\vec{\rho}$, respectively. The significance of the reference sequence is that the summations in the expression extend only over combinations of positions (clusters) occupied by amino acids differing from the corresponding ones in $\vec{\rho}$. Thus, C represents the binding score for $\vec{\rho}$ (i.e., the reference CF), whereas the remaining terms capture the additional contributions of amino acids in $\vec{\sigma}$ that differ from $\vec{\rho}$ (i.e., higher-order CFs). The first summation considers point CFs, with $fi(\sigma i)$ representing the effective contribution of amino acid σi at position i. Similarly, the second summation considers pair CFs, with $fij(\sigma i, \sigma j)$ representing the additional pairwise contribution due to having σi at position i and σj at position j simultaneously. To be exact, the expansion must consider all higher-order contributions, up to L-tuples, but in most cases this is impractical. Instead, one can choose to preserve only lower-order CFs (e.g., including only up to pairwise contributions), and use a training set of sequences with pre-computed scores to deduce CF values that optimize the accuracy of the truncated expansion [20].

Based on our PDZ study, a CE with up to pair CFs should represent peptide–PRD interactions reasonably well [19], though higher-order terms can still be added if needed [20].

Point CFs at all positions should be included. To reduce computational complexity, pair CFs can be restricted to position pairs likely to host side chains that interact either directly or through a common site on the PRD. For example, when building CEs for PDZ–peptide interactions, we omitted pair CFs between adjacent peptide positions, as these alternate in pointing either into or away from the binding interface, making coupling between them less likely [19]. Once a cluster is included in a CE (e.g., a pair cluster), every combination of non-reference amino acids at the corresponding positions produces a unique CF. Thus, the number of CFs to be considered is related to the size of design alphabet. For example, in our PDZ study, allowing 2–8 amino acids at six peptide positions resulted in 77 CFs (the reference CF, 24 point CFs, and 52 pair CFs) [19].

8. Generate sequences for CE training by randomly drawing from the design alphabet. The number of sequences should be at least twice the number of CFs to be considered (determined in **steps 6** and **7**). These sequences will be subjected to structure-based simulations, so choosing a design alphabet to be only as large as necessary (**step 6**) helps keep training time manageable. Figure 1 uses the PDZ example to show how the complexity of CE training increases with increasing number of amino acids allowed at each position. The random sequence

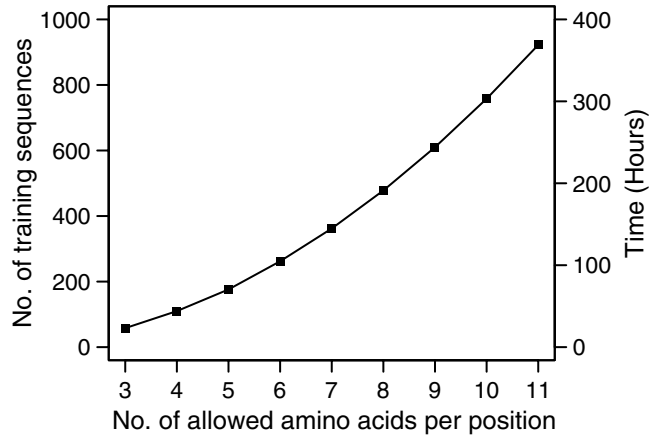

Fig. 1 The computational complexity of generating the CE training set increases with the number of amino acids allowed at each position. The clusters allowed in our PDZ study [19] are used in this estimation. The number of training sequences (*left-axis*) is estimated as twice the number of candidate cluster functions (CFs); time is estimated by assuming that a 1000-core compute cluster is available and that the simulations for one peptide take 400 wall-clock hours when run in serial (*see* **step 4**)

generation can be biased toward any known binding sequence preferences in order to concentrate the sampling (and ultimately CE accuracy) toward more relevant sequence spaces. No matter how the random set is generated, we recommend checking it for reasonable coverage of all CFs to be considered (e.g., at least three examples of each CF should be present). For any underrepresented CFs, sequences that contain them (but are otherwise random) should be added to balance the training set.

9. Run the simulation protocol in **step 4** for all sequences in the training set with all PRDs of interest, extracting the final binding score for each.
10. Train a CE model for each PRD by deriving optimal CF weights. In a linear algebra engine (e.g., MATLAB or Octave), create an $m \times n$ model matrix M, where m is the number of training sequences and n is the number of cluster functions considered ($m > n$). $M(i,j)$ should contain the number of times the j-th CF occurs in the i-th sequence. Typically, this will be either 1 or 0 (when the i-th sequence either does or does not involve the j-th CF, respectively), but can also be a larger integer in cases with structural symmetry, where a CF may occur multiple times within a sequence (e.g., with coiled coils; *see* Ref. [20]). Create also an $m \times 1$ vector E, whose i-th element is the structure-based binding score of the i-th sequence calculated in **step 9**. Optimal CF weights can then be obtained by finding the $m \times 1$ vector b that minimizes the mean squared difference between $E = Mb$ (CE-predicted scores) and E, with the j-th element of b representing the weight of the j-th CF. The least-square solution can be easily found using the method of pseudo-inverse as $\left(M^T M\right)^{-1} M^T E$. In MATLAB or Octave, this corresponds to the expression:

$$\mathrm{b} = \left(\mathrm{M'} * \mathrm{M}\right)\text{^}\left(-1\right) * \mathrm{M'} * \mathrm{E}$$

Note that matrix M has to be rank n, meaning that CFs have to represent orthogonal information and may not be linear combinations of each other (if M is not rank n, it often means an error was made either in encoding the model matrix or in defining CFs). Rather than including all candidate CFs into M at once and obtaining the best-fitting b, we recommend using our previously described strategy to prevent overtraining. The quality of a CE model (with a specific subset of CFs included) can be conveniently estimated as the average error with which the score of each sequence is predicted when that sequence is left out of the training set—the cross-validation root-mean-square error (CV-RMS). This value can be computed in closed form as

$$\sqrt{\frac{1}{n}\sum_{n}^{i=1}\left(\frac{E_i-\check{E}_i}{1-M_{i\cdot}\left(M^TM\right)^{-1}M_{i\cdot}^T}\right)^2},$$

where $Mi_{\cdot}$ represents the i-th row of matrix M. In MATLAB or Octave, this can be computed via the expression:

```
sqrt(sum(((E-M*b)./(1-sum(M.*(M*((M'*M)^(-1))'),
2))).^2)/length(E))
```

Thus, first train a CE model including all CFs (constant, point, and pair)—the all-inclusive model. Next, train another CE model with only constant and point CFs—the current model. Then, consider pair CFs, in decreasing order of their weights in the all-inclusive model, for addition to the current model. Each time a pair CF is considered for addition, train a new CE model that includes all CFs in the current model and the candidate pair CF, and evaluate the resulting CV-RMS. If it is lower than that of the current model, update the current model to include the CF; otherwise, discard the pair CF. Repeat until all pair CFs are considered. We have found this simple procedure to work well in practice, as in our PDZ-targeting study, but we have also proposed a more principled and general-purpose statistical method for choosing CFs to maximize CE accuracy [29].

11. Randomly generate a test set containing sequences not included in the training set, following the same procedure as in **step 8**. The number of sequences in the test set need only be large enough to provide a reliable estimate of CE error. Run the protocol in **step 4** for these sequences, and compute the root-mean-square of the difference between the resulting binding scores and scores calculated by the CE model from above (test-set RMS). This metric is a better indicator of expected CE error and is generally marginally higher than CV-RMS. Evaluate the quality of the CE model in the context of the ROC analysis in **step 5**. CE error should be lower than the score differences that tend to differentiate known binders from non-binders. If this is not the case, then the CE model is not of sufficient accuracy for specificity design, with several possible root causes: (1) important clusters were missed in **step 7**; (2) training set for CE was too small, such that important CF contributions could not be discerned; or (3) the structure-based score being considered is not easily expandable in terms of low-order CFs and may require more context for higher accuracy (e.g., triplet CFs may be necessary; *see* Ref. [20]).
12. Identify optimal peptide sequences for experimental characterization. In an earlier study, we described CLASSY, a framework that feeds CE models into an integer linear programming (ILP)

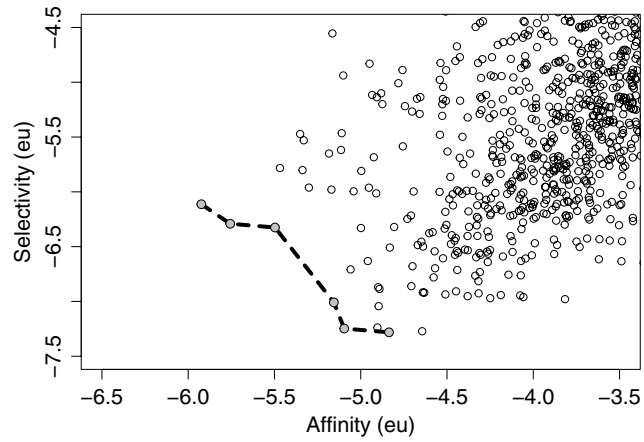

Fig. 2 An example predicted affinity/selectivity landscape, zoomed in around optimal sequences. Scores are shown in Rosetta energy units (eu). Each dot represents a peptide sequence; *X* and *Y* coordinates indicate affinity and selectivity scores, respectively (*see* Ref. [19]), with more negative numbers corresponding to higher affinity and selectivity. Sequences on the pareto-optimal front (i.e., those for which affinity and selectivity cannot be improved simultaneously; gray points) are connected with dashed lines. Adapted from Fig. 4a in Ref. [19]

framework to select sequences that make optimal trade-offs between affinity and selectivity [21]. Alternatively, in circumstances where the peptide sequence space is sufficiently small (i.e., $\leq 10^{10}$ sequences), given that the CE model typically takes less than 1 μs per peptide to evaluate, the entire sequence space can be simply enumerated. Either way, the goal is to find all peptide sequences that cannot be simultaneously improved in both predicted binding score and selectivity (i.e., the difference in binding scores between the target complex and the best-scoring off-target complex) [21]. These sequences lie at the edge of affinity/selectivity space (the so-called pareto-optimal front [30]) and are the only candidates worth considering, due to the simple fact that all other sequences can be simultaneously improved in both parameters. The pareto-optimal front is easy to visualize on a plot of affinity versus selectivity, where each point represents a sequence (Figure 2 shows a plot corresponding to one of the designs from our PDZ study [19]).

13. The number of sequences on the pareto-optimal front is often small enough to allow for the manual inspection of each [19, 21]. We recommend re-scoring each of these sequences by the structure-based framework in **step 4** to check for the possibility of anomalous CE error (discard any candidates scoring significantly less favorably in either affinity or selectivity by the structure-based framework than the CE model), manually analyzing the corresponding structural models for biophysical

plausibility (discard candidates with potential structural problems not properly recognized by the structural modeling framework), and finally choosing among remaining candidates based on the predicted scores. Depending on the availability of time and computational resources, one may also perform explicit-solvent molecular dynamics simulations of chosen candidates to build further support of at least local stability of the peptide in the binding site. Although relevant timescales will differ between systems, at least 10–100 ns of sampling is likely required in most situations to make any relevant observations. Additional issues in selecting candidate sequences are discussed in **Note 4**.

4 Notes

1. Our analysis showed that when a homologous template for a PDZ domain has around 35–45 % sequence identity to the target sequence, the Cα RMSD between the binding pockets of the true structure and the homology model has a median of 1.4 Å [19]. Also, when comparing *apo* and peptide-bound structures of PDZ domains, we noticed that PDZ binding sites tend to widen upon peptide binding [19]. Backbone rearrangements are not modeled in the Rosetta FlexPepDock, but it was shown that although these rearrangements are small, they are enough to affect the outcomes of the structural simulation significantly [27]. Therefore, peptide-bound structures are strongly preferred as homology-modeling templates. For example, in our previous work, we found that a PDZ domain homology model based on a peptide-bound structure with 40 % sequence identity performed much better in binding prediction than one based on an *apo* structure with 50 % sequence identity (unpublished data).
2. In our PDZ study, we conducted benchmark tests for two PDZ domains, NHERF-2 PDZ2 (N2P2) and MAGI-3 PDZ6 (M3P6), with Rosetta 3.4 [22] using the scoring function *score12*. We observed that dropping the backbone statistical terms "rama" and "omega" significantly improved performance [19]. The AUCs before and after omitting these terms were 0.57 and 0.77 for M3P6 (25 binders and 16 non-binders in the benchmark set; Fig. 3). In preparation of this manuscript, we also tested the performance of the new scoring function *talaris2013* used in a newer version of Rosetta (Rosetta_2014.35.57232_bundle), and the AUCs before and after dropping "rama" and "omega" were 0.71 and 0.76 for M3P6. This omission also marginally improves the performance on N2P2 (AUCs 0.86 and 0.91 before and after dropping),

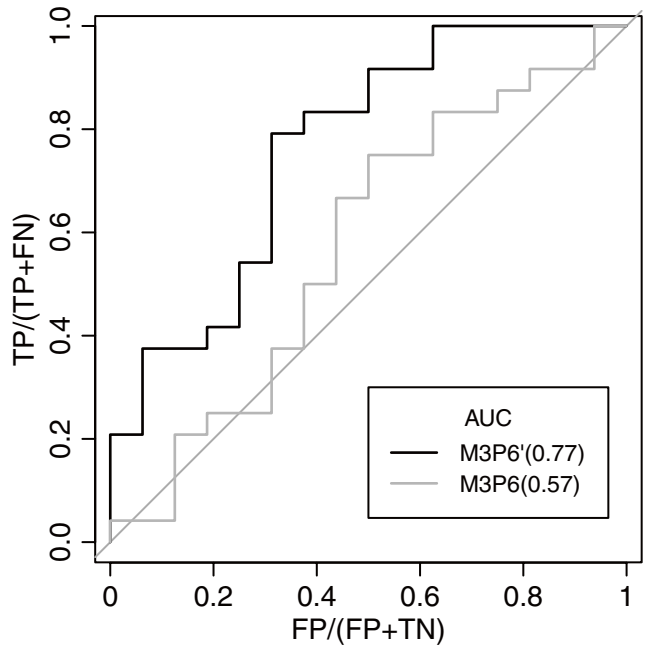

Fig. 3 An example ROC analysis, assessing the performance on differentiating binders from non-binders for M3P6. Default Rosetta scoring function *score12* (*gray line*, labeled as M3P6) and a modified version that omits "rama" and "omega" (*black line*, labeled as M3P6′) are compared. *Numbers in parenthesis* indicate the area under curve (AUC) for each case. *TP:* number of true positives, *FP:* number of false positives, *TN:* number of true negatives, *FN:* number of false negatives. Adapted from Fig. 2 in Ref. [19]

although this domain has fewer data points in our benchmark set (7 binders and 8 non-binders). Importantly, as no experimental structures of M3P6 were available, we used a homology model for simulating M3P6–peptide interactions in our study. Given that the improvement due to omitting "rama" and "omega" is larger for M3P6, it may be that the terms present more of an issue for homology models than crystal structures. Still, omitting the terms appears to improve the performance in general (including additional PDZ domains we have tested since our study; data not shown), and this may be due to the fact that Rosetta scoring functions are generally optimized to recognize/reproduce ground state-like conformations.

3. The benchmark dataset in our PDZ domain study came from the work of MacBeath and coworkers, which characterized binding affinities for a large number of PDZ–peptide pairs [7]. The authors reported dissociation constants if they were below 100 μM, or simply labeled interactions as "weak" in the opposite case. Thus, we naturally chose 100 μM as the cutoff for separating "binders" from "non-binders" for ROC analysis

[19]. If quantitative affinity measurements are not available, SPOT-array or phage-display data can also be used to classify sequences into two categories. However, one should use caution with such data, as they are in general more error prone, especially with respect to false negatives (i.e., true binders that are not detected in the assay).

4. It may be unnecessary to experimentally test all candidate sequences selected in **steps 12** and **13**. It is generally advantageous to characterize sequences spanning different levels of selectivity, to determine whether predicted affinity/selectivity trade-offs are correct. When possible and applicable, choose sequence subsets with diverse structural strategies for reaching either affinity or selectivity.

References

1. Ryan DP, Matthews JM (2005) Protein-protein interactions in human disease. Curr Opin Struct Biol 15(4):441–446
2. Bashor CJ, Horwitz AA, Peisajovich SG, Lim WA (2010) Rewiring cells: synthetic biology as a tool to interrogate the organizational principles of living systems. Annu Rev Biophys 39:515–537
3. Pawson T, Nash P (2003) Assembly of cell regulatory systems through protein interaction domains. Science 300(5618):445–452
4. Kuriyan J, Cowburn D (1997) Modular peptide recognition domains in eukaryotic signaling. Annu Rev Biophys Biomol Struct 26:259–288
5. Neduva V, Linding R, Su-Angrand I, Stark A, de Masi F, Gibson TJ, Lewis J, Serrano L, Russell RB (2005) Systematic discovery of new recognition peptides mediating protein interaction networks. PLoS Biol 3(12):2090–2099
6. Tonikian R, Zhang YN, Sazinsky SL, Currell B, Yeh JH, Reva B, Held HA, Appleton BA, Evangelista M, Wu Y, Xin XF, Chan AC, Seshagiri S, Lasky LA, Sander C, Boone C, Bader GD, Sidhu SS (2008) A specificity map for the PDZ domain family. PloS Biol 6(9):2043–2059
7. Stiffler MA, Chen JR, Grantcharova VP, Lei Y, Fuchs D, Allen JE, Zaslavskaia LA, MacBeath G (2007) PDZ domain binding selectivity is optimized across the mouse proteome. Science 317(5836):364–369
8. Jones RB, Gordus A, Krall JA, MacBeath G (2006) A quantitative protein interaction network for the ErbB receptors using protein microarrays. Nature 439(7073):168–174
9. Birnbaum ME, Mendoza JL, Sethi DK, Dong S, Glanville J, Dobbins J, Ozkan E, Davis MM, Wucherpfennig KW, Garcia KC (2014) Deconstructing the peptide-MHC specificity of T cell recognition. Cell 157(5):1073–1087
10. Chen JR, Chang BH, Allen JE, Stiffler MA, MacBeath G (2008) Predicting PDZ domain-peptide interactions from primary sequences. Nat Biotechnol 26(9):1041–1045
11. Kamisetty H, Ghosh B, Langmead CJ, Bailey-Kellogg C (2014) Learning sequence determinants of protein: protein interaction specificity with sparse graphical models, Research in computational molecular biology. Springer, New York, pp 129–143
12. London N, Lamphear CL, Hougland JL, Fierke CA, Schueler-Furman O (2011) Identification of a Novel Class of Farnesylation Targets by Structure-Based Modeling of Binding Specificity. PloS Comput Biol 7(10):e1002170
13. London N, Gulla S, Keating AE, Schueler-Furman O (2012) In silico and in vitro elucidation of BH3 binding specificity toward Bcl-2. Biochemistry 51(29):5841–5850
14. Roberts KE, Cushing PR, Boisguerin P, Madden DR, Donald BR (2012) Computational design of a PDZ domain peptide inhibitor that rescues CFTR activity. PloS Comput Biol 8(4)
15. Yanover C, Bradley P (2011) Large-scale characterization of peptide-MHC binding landscapes with structural simulations. Proc Natl Acad Sci U S A 108(17):6981–6986
16. DeBartolo J, Dutta S, Reich L, Keating AE (2012) Predictive Bcl-2 family binding models rooted in experiment or structure. J Mol Biol 422(1):124–144
17. DeBartolo J, Taipale M, Keating AE (2014) Genome-wide prediction and validation of

peptides that bind human prosurvival Bcl-2 proteins. PloS Comput Biol 10(6)

18. King CA, Bradley P (2010) Structure-based prediction of protein-peptide specificity in Rosetta. Proteins 78(16):3437–3449
19. Zheng F, Jewell H, Fitzpatrick J, Zhang J, Mierke DF, Grigoryan G (2015) Computational design of selective peptides to discriminate between similar PDZ domains in an oncogenic pathway. J Mol Biol 427(2):491–510
20. Grigoryan G, Zhou F, Lustig SR, Ceder G, Morgan D, Keating AE (2006) Ultra-fast evaluation of protein energies directly from sequence. PloS Comput Biol 2(6):551–563
21. Grigoryan G, Reinke AW, Keating AE (2009) Design of protein-interaction specificity gives selective bZIP-binding peptides. Nature 458(7240):859–U852
22. Leaver-Fay A, Tyka M, Lewis SM, Lange OF, Thompson J, Jacak R, Kaufman K, Renfrew PD, Smith CA, Sheffler W, Davis IW, Cooper S, Treuille A, Mandell DJ, Richter F, Ban YEA, Fleishman SJ, Corn JE, Kim DE, Lyskov S, Berrondo M, Mentzer S, Popovic Z, Havranek JJ, Karanicolas J, Das R, Meiler J, Kortemme T, Gray JJ, Kuhlman B, Baker D, Bradley P (2011) Rosetta3: an object-oriented software suite for the simulation and design of macromolecules. Methods Enzymol 487:545–574
23. Chaudhury S, Lyskov S, Gray JJ (2010) PyRosetta: a script-based interface for implementing molecular modeling algorithms using Rosetta. Bioinformatics 26(5):689–691
24. Arnold K, Bordoli L, Kopp J, Schwede T (2006) The SWISS-MODEL workspace: a web-based environment for protein structure homology modelling. Bioinformatics 22(2):195–201
25. Eswar N, Webb B, Marti-Renom MA, Madhusudhan MS, Eramian D, Shen MY, Pieper U, Sali A (2006) Comparative protein structure modeling using Modeller. Current protocols in bioinformatics Chapter 5: Unit 5 6
26. Zhou J, Grigoryan G (2015) Rapid search for tertiary fragments reveals protein sequence-structure relationships. Protein Sci 24(4):508–524
27. Raveh B, London N, Zimmerman L, Schueler-Furman O (2011) Rosetta FlexPepDock ab-initio: simultaneous folding, docking and refinement of peptides onto their receptors. PloS One 6(4)
28. Fawcett T (2006) An introduction to ROC analysis. Pattern Recogn Lett 27(8):861–874
29. Hahn S, Ashenberg O, Grigoryan G, Keating AE (2010) Identifying and reducing error in cluster-expansion approximations of protein energies. J Comput Chem 31(16):2900–2914
30. He L, Friedman AM, Bailey-Kellogg C (2012) A divide-and-conquer approach to determine the Pareto frontier for optimization of protein engineering experiments. Proteins 80(3):790–806
31. Wang CK, Pan LF, Chen J, Zhang MJ (2010) Extensions of PDZ domains as important structural and functional elements. Protein Cell 1(8):737–751