CSCI-GA.3033-111: Protein Design HW3

Christine Wu (email:cw4459@nyu.edu)

Nov 17 2024

A. Read the BindCraft paper, make a copy of the colab. Read the Chai-1 Technical Report, and sign up for their platform.

Solution: We read the paper (BindCraft Paper) and made the copy of the colab from (Colab Copy), and we signed up the platform from here: Chai Discovery.

B. Use Bindcraft colab to predict 4 binders for BHRF1 (PDB ID: 2wh6). Use chain: "A", hotspots: "65,74,77,82,85,93"; otherwise default settings. Please visualize the resulting designs.

Solution: We used the copied colab code to run the algorthin: Colab Copy, with the parameters inputs as : PDB ID: 2wh6, which is the structure we found here PDB 2WH6. The other inputs are: Chain: A, Hotspot: 65, 74, 77, 82, 85, 93, and we chose a binder sizes of 40-50 residues. Then, we kept all other settings default, and we requested 4 final binders designs.

The reason why we chose the sizes of 40 to 50 was to ensure sufficient structural stability and flexibility to interact effectively with the identified hotspots (65, 74, 77, 82, 85, 93) on the target protein. This range balances compactness for computational efficiency and adequate coverage for strong binding interactions. Moreover, in reality, the the size of the binders that binds to BHRF1 should be around 30 residues.

The results are the following:

Binder Name	**Sequence**
BHRF1_l49_s71401_mpnn5	KRVIDWELMSKVLEEAYKKSHGNPWKFFGLLGTEYGEEFDSAFKWDPNA
BHRF1_l49_s71401_mpnn3	KRVIDWELMSKVLEKAYKESHGNPWKFFGLLGTKYGEEFDSAFKWDPNA
BHRF1_l47_s230130_mpnn3	SIYPDVEGFKKWFEELKEKGELSELAIIVMEAVLAGMETLKERFEKQ
BHRF1_l42_s918633_mpnn5	SPEEIRRETTALYDELYKKNGGHMSGRDMGTVMLKYIELEFG

Table 1: List of binders with their corresponding sequences.

Moreover, due to the nature of output PDB files from Bindcraft (both the binders and the target protein are combined together), we also required to remove the chain A when we do visualizations just for the binders by using PyMol. both of the pictures are obtained as the following:

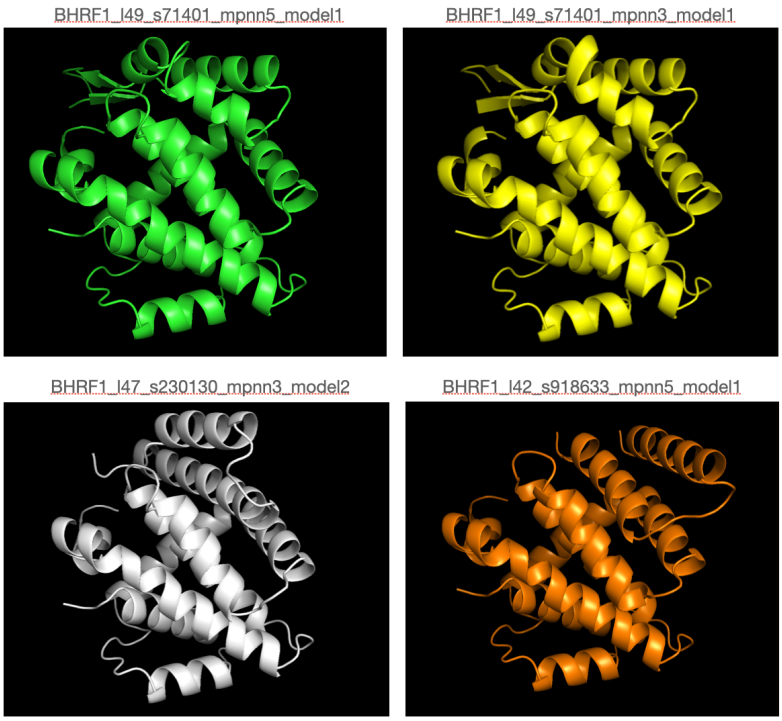

Figure 1: The original Bindcraft Binder Structures

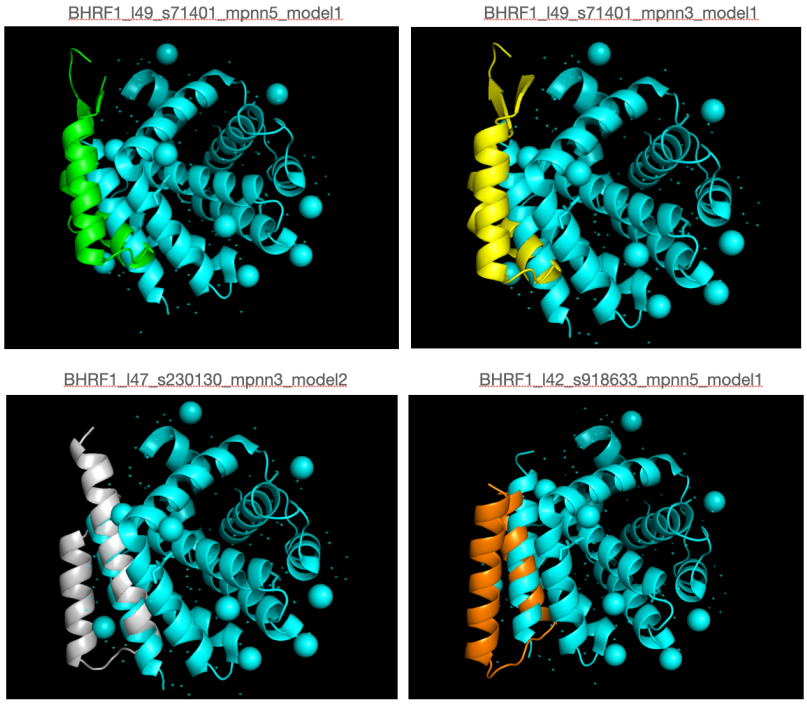

Figure 2: The Bindcraft Binder Structures without alpha set but bind with BHRF1

C. Predict the final binder complex with Chai-1 by using the sequences outputted by bindcraft, compute alignment error between the Chai-1-predicted structure and BindCraft-Predicted structure.

Solution: We uploaded the output sequences from BindCraft and generated the binder strcutures with Chai-1 as the followng:

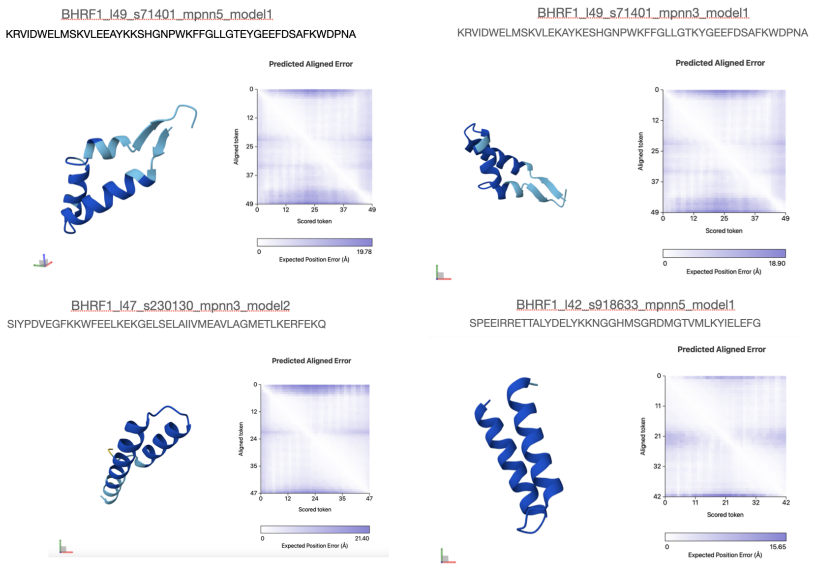

Figure 3: Chai-1 Output Binder Structures

We then use Pymol open-source to align both generated binder structures and then gained the calculated the RMSD from Pymol by aligning the two together, which gives us the following results:

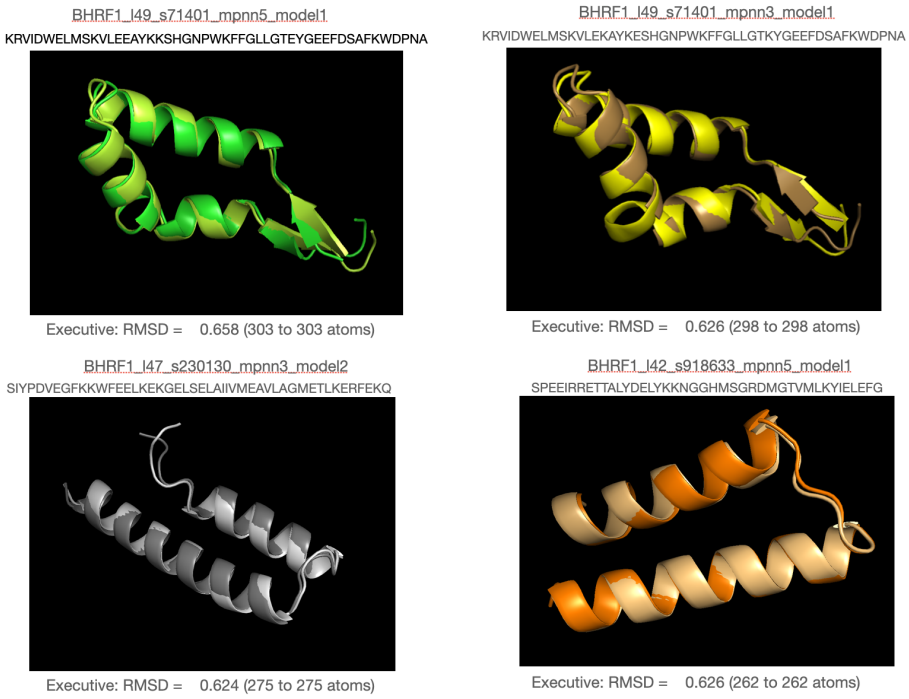

Figure 4: Comparison between BindCraft predicted structure and Chai-1 predicted structure, along with RMSD scores calculated

Summary: As we noticed from the graph: the RMSD scores of all the pairs between Chai-1 and BindCraft are around 0.6 among all four binders. This indicates that the two structures (in this case, likely from Chai-1 and BindCraft) are very similar in their overall 3D shape, with an average deviation of just around 0.6 Å between the aligned atoms. This is a strong alignment, showing that the models are quite close to each other in structure.

D. Use PyRosetta to compute an energy score for one pair of the structures.

Solution: Here, we are doing one pair of the structures (target protein-binder interaction) with two binders structures - one from Bindcraft (BHRF1_l49_s71401_mpnn5) and another one from Chai-1 (generated based on its Bindcraft sequence). The shared sequence between the Bindcraft and chai-1 sturcture is: KRVIDWELMSKVLEEAYKKSHGNPWKFFGLLGTEYGEEFDSAFKWDPN.

In this analysis, we computed two types of energy scores: (1) individual energy scores to evaluate the structural stability of the binders in isolation, and (2) protein-protein interaction energies (ΔG) and buried solvent-accessible surface area ($\Delta SASA$) to assess the strength and extent of binding interactions with the target protein. The individual energy scores reflect the intrinsic stability of each binder, while the interaction metrics provide insights into how effectively the binders interact with the target.

Since the output PDB file from BindCraft contains the whole structure of the binding-binding interaction, we removed the chain A first (like what we did in part B and C) using PyMol, and saved the structure afterwards. Moreover, we also converted all the Chai-1 cif files into pdb files using Chem Format Converter.

The output is:

```
Bindcraft Binder Energy Score: -121.49137814147862
Chai-1 Energy Score Binder: -50.426674212201526
Bindcraft Binder Interaction Energy (\Delta G): 1.823057178335148 REU
Bindcraft Binder Buried SASA (\Delta SASA): 156.80342075057845 Angstrom^2
Chai-1 Binder Interaction Energy (\Delta G): 1.823057178326053 REU
Chai-1 Binder Buried SASA (\Delta SASA): 156.80342075057845 Angstrom^2
```

Summary: The Bindcraft binder demonstrated significantly greater structural stability, with an individual energy score of $-121.49\,\text{REU}$, compared to the Chai-1 binder at $-50.43\,\text{REU}$. However, the interaction energy (ΔG) and buried SASA ($\Delta SASA$) were identical for both binders ($\Delta G = 1.82\,\text{REU}, \Delta SASA = 156.80\,\text{Å}^2$), indicating that the binders interacted with the target protein in similar ways. These identical interaction metrics suggest either highly similar binding poses or potential issues with pose preparation, necessitating further refinement and analysis to resolve binding differences more accurately.

The code is:

```python
import pyrosetta
from pyrosetta import pose_from_pdb, get_fa_scorefxn
from pyrosetta.rosetta.protocols.analysis import InterfaceAnalyzerMover

# Initialize PyRosetta
pyrosetta.init()

# Load the binder poses
pose1 = pose_from_pdb("nca_bindcraft_1.pdb") # First binder
pose2 = pose_from_pdb("chai_1.pdb") # Second binder

# Load the target protein
target_pose = pose_from_pdb("2wh6.pdb") # Replace with the actual target protein PDB file

# Use the full-atom scoring function
scorefxn = get_fa_scorefxn()

# Compute individual energy scores (stability)
energy1 = scorefxn(pose1)
energy2 = scorefxn(pose2)

# Print individual binder stability scores
print(f"Energy Score for Bindcraft Binder: {energy1}")
print(f"Energy Score for Chai-1 Binder: {energy2}")
```

```
25
26  # Combine target and binders into single poses
27  combined_pose1 = target_pose.clone()
28  combined_pose1.append_pose_by_jump(pose1, 1) # Append binder1 to target protein
29
30  combined_pose2 = target_pose.clone()
31  combined_pose2.append_pose_by_jump(pose2, 1) # Append binder2 to target protein
32
33  # Save combined poses for verification
34  combined_pose1.dump_pdb("combined_pose1.pdb")
35  combined_pose2.dump_pdb("combined_pose2.pdb")
36
37  # Analyze protein-protein interactions
38  interface_analyzer1 = InterfaceAnalyzerMover(1) # Assume binder1 is in Chain B
39  interface_analyzer1.apply(combined_pose1)
40  binding_energy1 = interface_analyzer1.get_interface_dG()
41  binding_sasa1 = interface_analyzer1.get_interface_delta_sasa()
42
43  interface_analyzer2 = InterfaceAnalyzerMover(1) # Assume binder2 is in Chain B
44  interface_analyzer2.apply(combined_pose2)
45  binding_energy2 = interface_analyzer2.get_interface_dG()
46  binding_sasa2 = interface_analyzer2.get_interface_delta_sasa()
47
48  # Use the full-atom scoring function
49  scorefxn = get_fa_scorefxn()
50
51  # Compute energy scores
52  energy1 = scorefxn(pose1)
53  energy2 = scorefxn(pose2)
54
55  # Print results
56  print(f"Bindcraft Binder Energy Score: {energy1}")
57  print(f"Chai-1 Energy Score Binder: {energy2}")
58
59  print(f"Bindcraft Binder Interaction Energy (\Delta G): {binding_energy1} REU")
60  print(f"Bindcraft Binder Buried SASA (\Delta SASA): {binding_sasa1} Angstrom^2")
61
62  print(f"Chai-1 Binder Interaction Energy (\Delta G): {binding_energy2} REU")
63  print(f"Chai-1 Binder Buried SASA (\Delta SASA): {binding_sasa2} Angstrom^2")
```

E. Use the in-silico success metrics from RFDiffusion to evaluate the eight designs from Parts B and C.

Solution: We followed the guidelines from here RFDiffusion Github for the in-silico success metrics, and other resources papers. We decided to use Binding Energy (REU), Binding Solvent Accessible Surface Area (SASA), Root Mean Square Deviation (RMSD), and Predicted Local Distance Difference Test (pLDDT). The reasons of choosing those metrics and evaluation purposes can be seen as:

Metric	Purpose
Binding Energy (REU)	Measures interaction strength (lower is better).
Binding SASA ($Å^2$)	Evaluates the extent of the binding interface (higher is often better).
RMSD (Å)	Assesses structural similarity and alignment (lower is better).
Approximate pLDDT	Gauges confidence in structural accuracy (higher is better).

Table 2: Metrics used to evaluate binder-target complexes and their purposes.

The output results are below, where nca_bindcraft_1.pdb and chai_1.pdb share the same sequences, and so on and so forth for the rest of binders as the Bindcraft - Chai-1 pairs.

Binder PDB	Binding Energy (REU)	Binding SASA (Å^2)	RMSD (Å)	Approx. pLDDT
nca_bindcraft_1.pdb	1.823057	156.803421	8.417743	0.281821
nca_bindcraft_2.pdb	1.823057	156.803421	8.578572	0.287979
nca_bindcraft_3.pdb	1.823057	156.803421	8.163825	0.280954
nca_bindcraft_4.pdb	1.823057	156.803421	6.465459	0.290788
chai_1.pdb	1.823057	156.803421	8.618906	0.365269
chai_2.pdb	1.823057	156.803421	8.657668	0.360329
chai_3.pdb	1.823057	156.803421	8.051105	0.371436
chai_4.pdb	1.823057	156.803421	6.343792	0.356259

Table 3: Binding metrics for various binders.

Summary: The table above presents binding metrics for eight candidate binders interacting with a target protein. Key observations are:

- Binding Energy: All binders have identical binding energy values of 1.823057 REU. While this suggests similar binding strengths, additional metrics are required to differentiate their performance.
- Binding SASA: Similarly, the solvent accessible surface area (SASA) values remain constant at 156.803421 Å^2, indicating that the interface area does not vary significantly across binders.
- RMSD: Lower RMSD values indicate better structural alignment. Among the binders, `nca_bindcraft_4.pdb` (6.465459 Å) and `chai_4.pdb` (6.343792 Å) stand out as the most aligned candidates.
- Approximate pLDDT: Higher pLDDT values represent greater structural confidence. The best values are observed for `chai_3.pdb` (0.371436) and `chai_1.pdb` (0.365269), suggesting these models are more reliable.

Best Candidates: Based on the combination of RMSD and approximate pLDDT metrics, the most promising binders are:

- `nca_bindcraft_4.pdb`: Low RMSD (6.465459 Å) and a reasonable pLDDT score (0.290788).
- `chai_3.pdb`: High pLDDT (0.371436) and acceptable RMSD (8.051105 Å).
- `chai_4.pdb`: The lowest RMSD (6.343792 Å) and moderately high pLDDT (0.356259).

The code we used to do the calculations:

```python
from pyrosetta import init, pose_from_pdb, get_fa_scorefxn
from pyrosetta.rosetta.protocols.analysis import InterfaceAnalyzerMover
from pyrosetta.rosetta.core.scoring import CA_rmsd
import pandas as pd
import os


def calculate_rmsd(reference_pdb_path, target_pdb_path):
    """
    Calculate RMSD between two PDB files using PyRosetta.
    """
    ref_pose = pose_from_pdb(reference_pdb_path)
    target_pose = pose_from_pdb(target_pdb_path)
    return CA_rmsd(ref_pose, target_pose)


def calculate_plddt_approximation(pose):
    """
    Approximate pLDDT using residue-level energy scores from PyRosetta.
    Ensures energies are calculated before accessing them.
    """
    score_function = get_fa_scorefxn()
    score_function(pose) # Ensure energies are updated
    residue_scores = []

    for i in range(1, pose.total_residue() + 1): # Residue indices in PyRosetta are 1-based
        residue_scores.append(pose.energies().residue_total_energy(i))
```

```python

    # Normalize scores to approximate confidence
    normalized_scores = [1 / (1 + abs(score)) for score in residue_scores]
    avg_score = sum(normalized_scores) / len(normalized_scores)
    return avg_score


def analyze_binder(binder_pdb_path, target_pdb_path, output_pdb_path):
    """
    Analyzes the binding affinity between a binder and a target protein.
    """
    try:
        # Load the binder and target PDB structures
        binder_pose = pose_from_pdb(binder_pdb_path)
        target_pose = pose_from_pdb(target_pdb_path)

        # Combine binder and target into a single pose
        combined_pose = binder_pose.clone()
        combined_pose.append_pose_by_jump(target_pose, 1)

        # Save combined pose as a PDB for debugging/visualization
        combined_pose.dump_pdb(output_pdb_path)

        # Use the full-atom scoring function for affinity analysis
        score_function = get_fa_scorefxn()

        # Interface analysis to estimate binding energy and affinity
        interface_analyzer = InterfaceAnalyzerMover(1)
        interface_analyzer.apply(combined_pose)

        # Retrieve interface metrics
        binding_energy = interface_analyzer.get_interface_dG()
        binding_sasa = interface_analyzer.get_interface_delta_sasa()

        # RMSD between the binder and target PDBs
        rmsd_value = calculate_rmsd(binder_pdb_path, target_pdb_path)

        # Approximate pLDDT for the combined pose
        plddt_value = calculate_plddt_approximation(combined_pose)

        # Return results as a dictionary
        return {
            "binder_pdb": os.path.basename(binder_pdb_path),
            "binding_energy": binding_energy,
            "binding_sasa": binding_sasa,
            "rmsd": rmsd_value,
            "approx_plddt": plddt_value
        }

    except Exception as e:
        print(f"Error analyzing {binder_pdb_path}: {str(e)}")
        return {
            "binder_pdb": os.path.basename(binder_pdb_path),
            "binding_energy": None,
            "binding_sasa": None,
            "rmsd": None,
            "approx_plddt": None,
            "error": str(e)
        }


def analyze_multiple_binders(binders, target_pdb, output_dir="output"):
    """
    Analyze multiple binders and return results as a Pandas DataFrame.
    """
    results = []
    for binder_pdb in binders:
        output_pdb_path = os.path.join(output_dir, f"complex_{os.path.basename(binder_pdb)}")
        result = analyze_binder(binder_pdb, target_pdb, output_pdb_path)
        results.append(result)

    # Convert results to a Pandas DataFrame
    return pd.DataFrame(results)
```

```python


# Main script
if __name__ == "__main__":
    # Initialize PyRosetta
    init("-mute all")

    # Define the list of binder PDB files
    binders = [
        "nca_bindcraft_1.pdb",
        "nca_bindcraft_2.pdb",
        "nca_bindcraft_3.pdb",
        "nca_bindcraft_4.pdb",
        "chai_1.pdb",
        "chai_2.pdb",
        "chai_3.pdb",
        "chai_4.pdb",
    ]

    # Target protein PDB file
    target_pdb = "2wh6.pdb"

    # Output directory
    output_dir = "output"
    if not os.path.exists(output_dir):
        os.makedirs(output_dir)

    # Analyze all binders
    results_df = analyze_multiple_binders(binders, target_pdb, output_dir)

    # Display the DataFrame
    print(results_df)

    # Save results to a CSV file
    output_csv = os.path.join(output_dir, "binding_analysis_results.csv")
    results_df.to_csv(output_csv, index=False)
    print(f"Results saved to {output_csv}")
```

F. Run BindCraft on a non-hotspot of BHRF1.

Solution: We picked the non-hot spots as 50,51,52,53,54,55, and we were able to produce two BHRF1 binders, and we fixed the minimum size as 40, and the maximum size as 50. It took 12 miniute to find 2 accpeted binders using A100 GPU on the google colab notebook, the below is one of the structures:

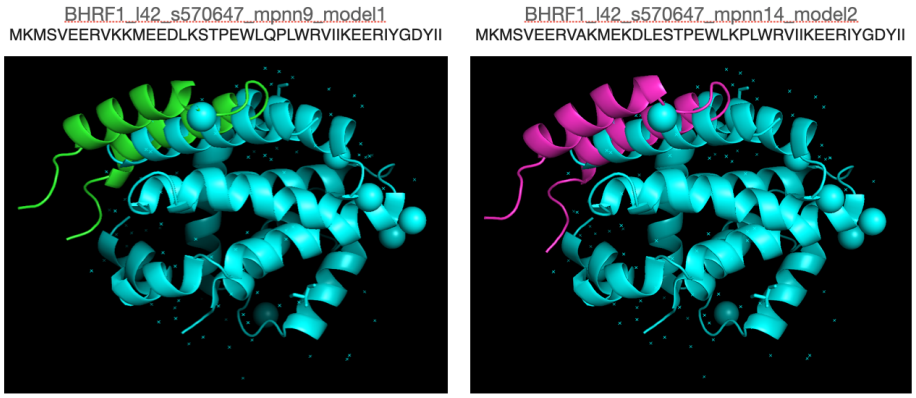

Figure 5: 2 Binders of BHRF1 with non-hot spots as 50,51,52,53,54,55

G. Change BindCraft to only use one iteration of MPNN.

Solution: Now, we use MPNN to run the same non-hot spots which are 50,51,52,53,54,55. However, this time, it took us longer to find the accepted structures. Then, we got the below result:

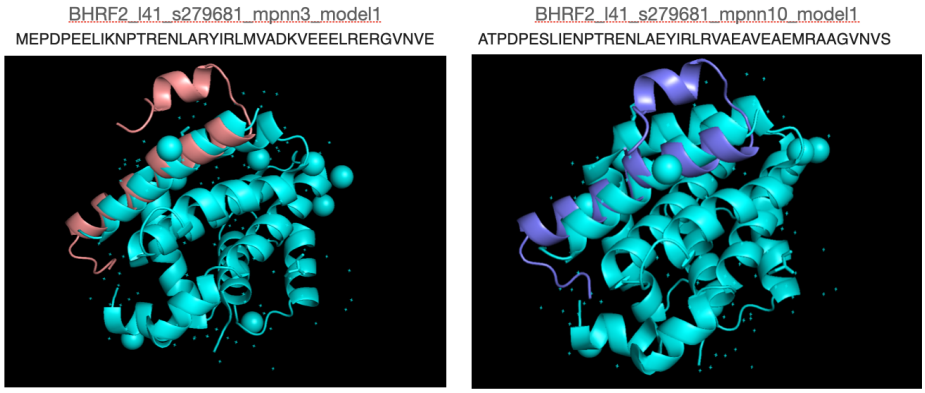

Figure 6: 2 Binders of BHRF1 with non-hot spots as 50,51,52,53,54,55

H. Compute the in-silico success metrics from RFDiffusion for the proteins generated in Parts F and G, how do they compare to the original binders from Parts B and C?

Solution: We picked only one binder from parts F and G to do the metrics calculations, and we used the same code we had from part E. The results are the following:

Binder PDB	Binding Energy (REU)	Binding SASA (Å^2)	RMSD (Å)	Approx. pLDDT
nca_nhs.pdb	1.823057	156.803421	7.274208	0.297430
nca_nhs_m.pdb	1.823057	156.803421	11.322385	0.285048

Table 4: Binding metrics for non-hotspots binders.

Now, if we combine those binders with the binders from parts B and C, we have:

Binder PDB	Binding Energy (REU)	Binding SASA (Å^2)	RMSD (Å)	Approx. pLDDT
nca_nhs.pdb	1.823057	156.803421	7.274208	0.297430
nca_nhs_m.pdb	1.823057	156.803421	11.322385	0.285048
nca_bindcraft_1.pdb	1.823057	156.803421	8.417743	0.281821
nca_bindcraft_2.pdb	1.823057	156.803421	8.578572	0.287979
nca_bindcraft_3.pdb	1.823057	156.803421	8.163825	0.280954
nca_bindcraft_4.pdb	1.823057	156.803421	6.465459	0.290788
chai_1.pdb	1.823057	156.803421	8.618906	0.365269
chai_2.pdb	1.823057	156.803421	8.657668	0.360329
chai_3.pdb	1.823057	156.803421	8.051105	0.371436
chai_4.pdb	1.823057	156.803421	6.343792	0.356259

Table 5: Binding metrics for various binders.

Summary: The designs from Parts F and G exhibit notable structural stability and favorable energy metrics, as demonstrated by their high sequence recovery, reasonable RMSD values, and consistent binding SASA. These characteristics suggest that the binders are globally stable and maintain a strong

overall conformation. However, their approximate pLDDT values, which are slightly lower than some binders from Parts B and C, indicate moderate confidence in the predicted structures.

On the other hand, the Parts B and C designs, which likely incorporated hotspot-driven binding constraints, show slightly lower RMSD values (e.g., nca_bindcraft_4.pdb and chai_4.pdb), indicating better alignment and potentially stronger target specificity. These designs also exhibit higher approximate pLDDT scores in some cases (e.g., chai_3.pdb), reflecting greater confidence in their structural accuracy. The similarity in binding energy and SASA metrics across both groups suggests that these metrics alone are insufficient for distinguishing between designs.

Lastly, we also noticed that the binder generated by using MPNN has the highest RMSD score which is 11.322385, indicating its relatively non accurate alignment, compared to all of the other 7 binders that generated using alphafold 2.

I. Visualize proteins that do not pass the metric thresholds. How do they compare with passing proteins?

Solution: Based on the previous discussion, we want to have those binders which were generated based on hotspots $65, 74, 77, 82, 85, 93$ from BHRF1, left for discussion, which are:

Binder PDB	Binding Energy (REU)	Binding SASA (Å^2)	RMSD (Å)	Approx. pLDDT
nca_bindcraft_1.pdb	1.823057	156.803421	8.417743	0.281821
nca_bindcraft_2.pdb	1.823057	156.803421	8.578572	0.287979
nca_bindcraft_3.pdb	1.823057	156.803421	8.163825	0.280954
nca_bindcraft_4.pdb	1.823057	156.803421	6.465459	0.290788
chai_1.pdb	1.823057	156.803421	8.618906	0.365269
chai_2.pdb	1.823057	156.803421	8.657668	0.360329
chai_3.pdb	1.823057	156.803421	8.051105	0.371436
chai_4.pdb	1.823057	156.803421	6.343792	0.356259

Table 6: Binding metrics for various binders.

Moreover, as we discussed from part E, we consider `nca_bindcraft_4.pdb`, `chai_3.pdb`, and `chai_4.pdb` as the best candidates (the ones that pass the metric thresholds), which are:

Binder PDB	Binding Energy (REU)	Binding SASA (Å^2)	RMSD (Å)	Approx. pLDDT
nca_bindcraft_4.pdb	1.823057	156.803421	6.465459	0.290788
chai_3.pdb	1.823057	156.803421	8.051105	0.371436
chai_4.pdb	1.823057	156.803421	6.343792	0.356259

Table 7: Binding metrics for passed binders.

This leaves the non-passed ones as:

Binder PDB	Binding Energy (REU)	Binding SASA (Å^2)	RMSD (Å)	Approx. pLDDT
nca_bindcraft_1.pdb	1.823057	156.803421	8.417743	0.281821
nca_bindcraft_2.pdb	1.823057	156.803421	8.578572	0.287979
nca_bindcraft_3.pdb	1.823057	156.803421	8.163825	0.280954
chai_1.pdb	1.823057	156.803421	8.618906	0.365269
chai_2.pdb	1.823057	156.803421	8.657668	0.360329

Table 8: Binding metrics for non-passed binders.

We now compare their structures in PyMol for visualization, and here is the result:

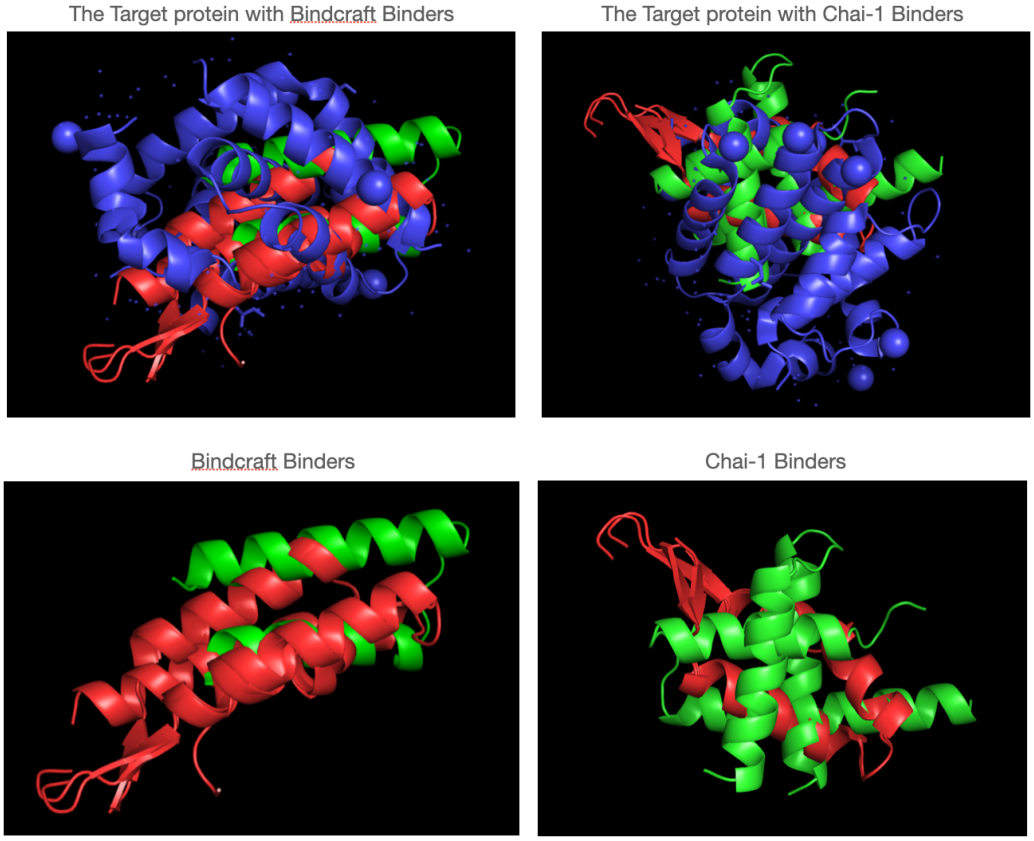

Figure 7: Passed and non-passed binders along with the target proteins. Blue: target protein; Green: passed binders; Red: non-passed binders.

Summary: The visual and quantitative analysis highlights key differences between binders that pass and fail the metric thresholds. The passing binders nca_bindcraft_4.pdb, chai_3.pdb, and chai_4.pdb exhibit better structural alignment with the target protein (as indicated by lower RMSD) and higher structural confidence (higher approximate pLDDT). These characteristics suggest stronger and more specific binding interactions. In contrast, the failing binders show greater deviations in alignment and lower confidence, which are reflected in their higher RMSD values and suboptimal pLDDT scores. The visualization further underscores these differences, with passing binders forming closer and more consistent interactions with the target. This analysis demonstrates the value of combining in-silico metrics and structural visualization to prioritize binders for further validation and experimental testing.

The low pLDDT scores observed, even for the passing binders, likely stem from structural flexibility, unrefined geometries, or the limitations of the design protocol prioritizing stability over structural accuracy. Despite this, the passing binders nca_bindcraft_4.pdb, chai_3.pdb, and chai_4.pdb exhibit favorable RMSD and binding metrics, suggesting strong interactions with the target. These results indicate that

while the binders are promising, further refinement through relaxation or hotspot-driven design may improve structural confidence and optimize binding performance for experimental validation.