Guide: using FastDesign to compare and optimize your binder redesigns

A recap...

Your COVID variant binder redesign project has come a long way! From first conceptualizing the project and understanding the system, then learning to use the command line, Foldit, and PyRosetta, to collectively identifying the critical mutations in the variants' spikes that have the potential to interfere with binding.

Now you're using your own *and* your team members' findings to create binder redesigns that work better against both COVID variants: Beta & Gamma than that the original binder, which was created specifically to attach to the Wild Type virus spike.

You got started in Foldit by manually "hand mutating" a binder scaffold (LCB1, LCB3, or AHB1) with the directions "anything goes" just try to arrive at the lowest Rosetta Energy Unit score possible with the best fitting interface.

Finding a universal design to move forward into PyRosetta testing

From there, you compared your results with your colleagues, sharing what worked and what didn't to arrive at a universal binder redesign concept to test with software beyond Foldit. At this point you want to model this universal binder redesign with the WT, Gamma, and Beta spike variants to make sure that it is indeed "universal"
To do this:

1. Download the appropriate scaffold files with the WT, Gamma, and Beta spike variants
2. Make the mutations in Foldit. Don't worry about minimizing between each mutation. Wait till you've made all the mutations, then go ahead and minimize globally (several iterations of shake and wiggle until the score appears to bottom out)
3. Compare final scores and save the structures (you'll want these PDBs for the next steps)

Table 1. Comparison of Rosetta Energy Unit score changes after global minimization of binder redesign with mutations *A33F, etc. (list all mutations) here* starting from scaffold LCB1/LCB2/AHB1

	Wild Type	**Beta**	**Gamma**
ΔFoldit Score (Change in Rosetta Energy Units)			

mutations list. Mutations that are **conserved between the different spike/virus variants are bolded.** Mutations that are *directly* interacting with spike variants are highlighted in yellow.	

*Δ: *calculate the score difference (delta) before and after you make the mutations. If the score decreases (goes in the negative direction) be sure to represent with a negative score change value in the table!*

This guide describes the next steps: in other words what and how you'll be testing your universal binder redesigns.

Use FastDesign to supplement your hand mutations

You might have already done this as part of your process in creating your universal redesign. If not though, no worries! Now is a great time to try.

Here you'll be telling PyRosetta to keep all your hard work (aka your hand-designed mutations) and to work around those, only running FastDesign on the other amino acids in the interface. Here is a template script/notebook. Keep an eye out for notes on the bits of code you need to change to adapt it to your PDB. Remember, you want to capture a universal binder design that works for both Beta & Gamma (and perhaps WT too).

Run this script with PDBs you generated in the last step (step 3, page 1). Compare: did FastDesign produce any of the same mutations on both/all spike variants. We'll call these "conserved" mutations.

Table 2. Comparison of Rosetta Energy Unit score changes after global minimization of binder redesigns AFTER running FastDesign on residues in the interface, excluding mutations *A33F, etc. (list all mutations) here* starting from scaffold LCB1/LCB2/AHB1

	Wild Type	**Beta**	**Gamma**
ΔFoldit Score (Change in Rosetta Energy Units)			
mutations suggested by FastDesign. Mutations that are **conserved between the different spike/virus variants are bolded.** Mutations that are *directly* interacting with spike variants are highlighted in yellow.			

Comparative analysis with RosettaFast Design

To investigate if your original, hand-designed mutations on the binder were also favorable from Rosetta's perspective, try running FastDesign on the original binder scaffolds with the different spike mutants, here's a template script/notebook to get the job done. After you've generated PDBs for these, minimize these in Foldit and keep track of your findings in a table

Table 3. Comparison of Rosetta Energy Unit score changes after global minimization after running FastDesign on residues in the interface between the starting scaffold LCB1/LCB2/AHB1 binder and the variants.

	Wild Type	**Beta**	**Gamma**
ΔFoldit Score (Change in Rosetta Energy Units)			
Interface mutations suggested by FastDesign. Mutations that are **conserved between the different spike/virus variants are bolded**			

**Highlight any mutations suggested by FastDesign that you also previously identified when you created your universal binders by hand mutating in Foldit. Also note if there are mutations that FastDesign suggested from this analysis and previously when you used FastDesign to supplement your hand mutations.*

Comparative analysis with AlphaFold2

Here you're going to gather data on how well your new redesigned binder is likely to fold up as you expect. We know the scaffolds fold up as they are modeled because we have the crystal structures. However, now that we've made a bunch of changes to the amino acid sequence there is a possibility that we've destroyed the shape! What better and timely way to make sure things are still folding as expect than to run them through AlphaFold. Here's the AlphaFold Google Colab Notebook, straight from the developers (wowow!).

AlphaFold is doing a bad job modeling the binder-spike complex that you've been working with thus far. At this point, you will make a binder-only PDBs of your redesign by simply deleting the RBD/spike chain in PyMol. This binder-only PDB will be the one you'll be working with going forward with AlphaFold.

When you're ready to model your universal binder redesign, go to the template Google Colab AlphaFold notebook and run it. Finally align the AlphaFold prediction PDB with the original scaffold and WT complex. Here's a guide on aligning PDBs.

Note: to run AlphaFold, you'll need the amino acid sequence of your binder. There's a couple ways to do this. The PyRosetta code line pose.sequence() will work.

You can also use PyMol to remove the RBD/spike and export the file as a FASTA file

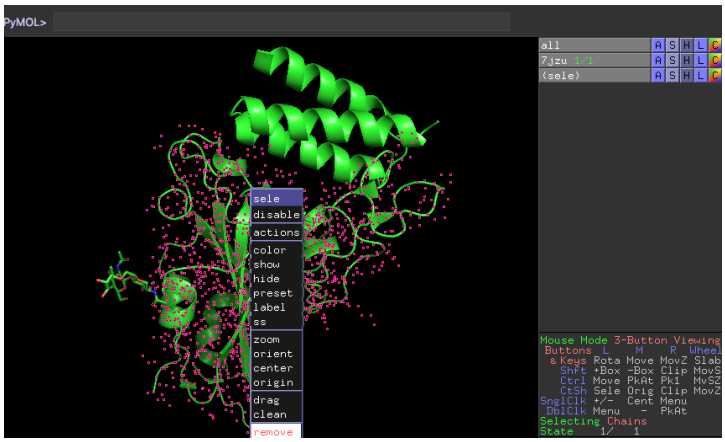


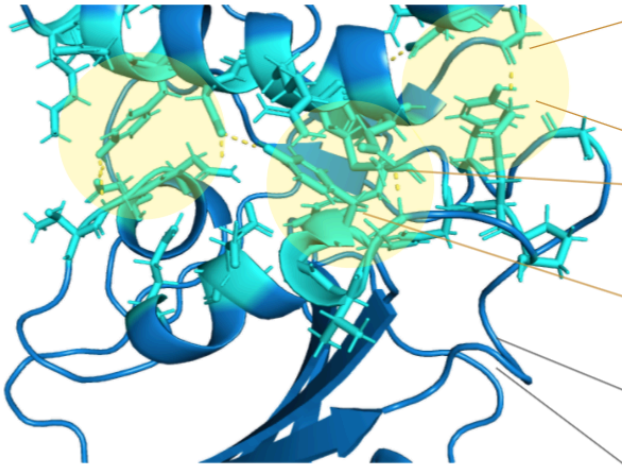
Figure 1. Select the chain you want to get rid of and right click to bring up the "remove" option.

Thinking forward… turning your results into a story with visuals!

Beyond the tables you put together in the earlier section of this guide, here are some ideas for how to describe your findings: The details are here: in this Slide Deck: Results! Week 6. If you've got a great idea for visualizing your results and you think it would be cool to share this, add a slide example in the deck to inspire your colleagues!

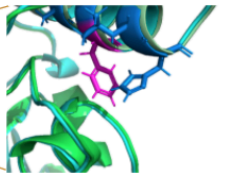
Deliverables

Figure 1. Setting the scene



Highlight the interface with WT binder. Show regions that have been changed by the variant's mutations.
**This image is not an accurate example!*

Figure 2a-n. Zoom into redesigned binder mutations that directly interact with variant's mutations



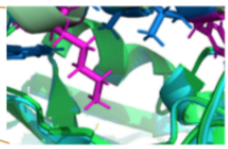


Figure 3a-n. Zoom into redesigned binder mutations that do NOT directly interact with variant's mutations

**Note this is a particular bad example because what I'm showing isn't even the binder*