<u>Introduction</u>

The overarching goal of this workflow is to show Bench users how they may use their understanding of a complex protein-protein interface to generate *specificity variants* on that sequence, yielding *orthogonal interfaces*. To illustrate this general principle, this workflow describes the specific challenge of creating bispecific antibodies. One computational approach to this problem is described in the 2014 paper in Nature Biotechnology titled "Generation of bispecific IgG antibodies by structure-based design of an orthogonal Fab interface." In this workflow, we reimplement its basic ideas using Bench, rather than Rosetta. Within this context, we will examine two strategies for creating new orthogonal interfaces based on knob-hole mutagenesis and a charge-pairing strategy.

<u>Scientific Background</u>

The scientific aim of this workflow is to obtain sequences for two sets of antibody heavy and light chains that, when co-expressed, will pair specifically to form a bispecific antibody. Ordinarily, coexpression of mixed heavy and light chains results in some mis-assembly products, to the point that many researchers have avoided this particular strategy to create bispecific antibodies altogether. This workflow describes a method to prevent mis-association of heavy and light chains by designing an orthogonal interface. There are two distinct interfaces where one may wish to install mutations: the CH1/CL interface, which is part of the antibody's 'constant region', and the VH/VL interface, which is part of the variable region. Orthogonalizing the CH3 interfaces is also necessary, but outside the scope of this workflow; the same ideas can (and have) been used (Leaver-Fay *et al.*, 2016).

In this example, we are starting from a crystal structure of the monoclonal antibody we would like to orthogonalize, PGT128, which is a broadly-neutralizing HIV-1 antibody. This method requires that you have a crystal structure to start with. [This method could also begin from a high quality antibody model, but in this particular workflow and publication, a crystal structure is used.]

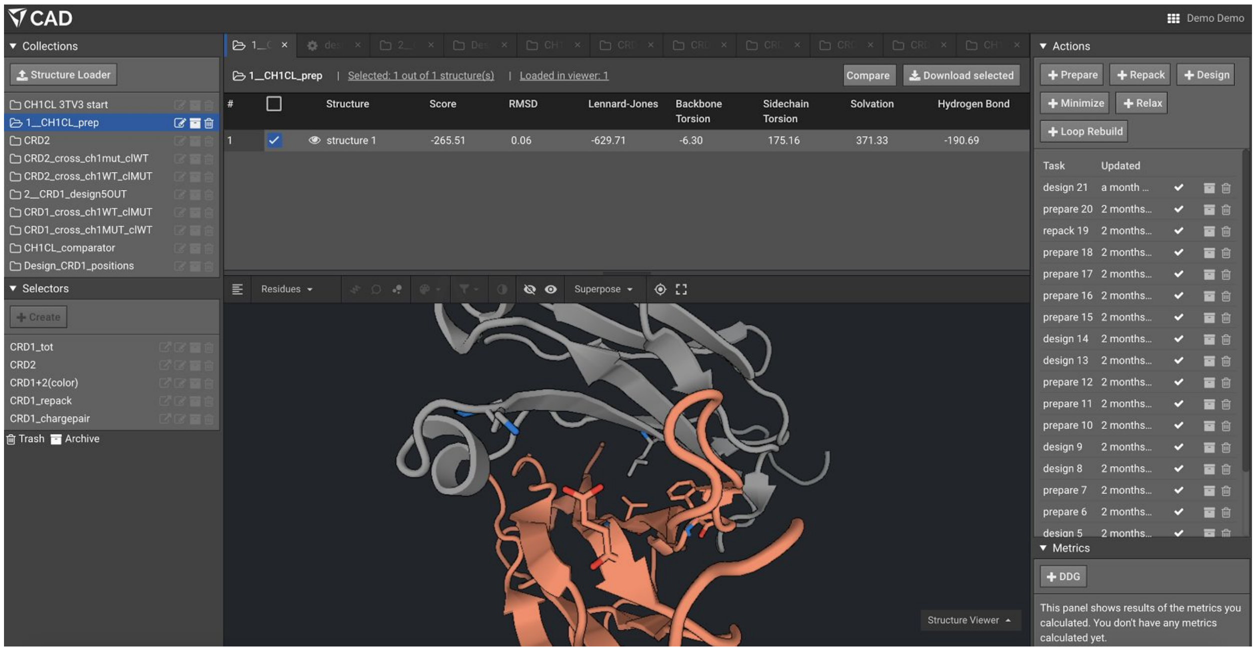
<u>Workflow</u>

Part One

First, we make two constant region redesigns that ought to be bispecific against the wild type:

1. In Bench, fetch 3TV3. This structure contains the constant region of the heavy (H) and light (L) chains. This PDB also contains the CH3 interface, where we won't be making any mutations, so for the purpose of this workflow we recommend removing it (as we have <here>).

2. Run ‘prepare’, producing a single decoy.



CAD — Demo Demo

Collections: Structure Loader; CH1CL 3TV3 start; 1__CH1CL_prep; CRD2; CRD2_cross_ch1mut_clWT; CRD2_cross_ch1WT_clMUT; 2__CRD1_design5OUT; CRD1_cross_ch1WT_clMUT; CRD1_cross_ch1MUT_clWT; CH1CL_comparator; Design_CRD1_positions

Selectors: Create; CRD1_tot; CRD2; CRD1+2(color); CRD1_repack; CRD1_chargepair

Trash Archive

1__CH1CL_prep | Selected: 1 out of 1 structure(s) | Loaded in viewer: 1 — Compare — Download selected

#	Structure	Score	RMSD	Lennard-Jones	Backbone Torsion	Sidechain Torsion	Solvation	Hydrogen Bond
1	structure 1	-265.51	0.06	-629.71	-6.30	175.16	371.33	-190.69

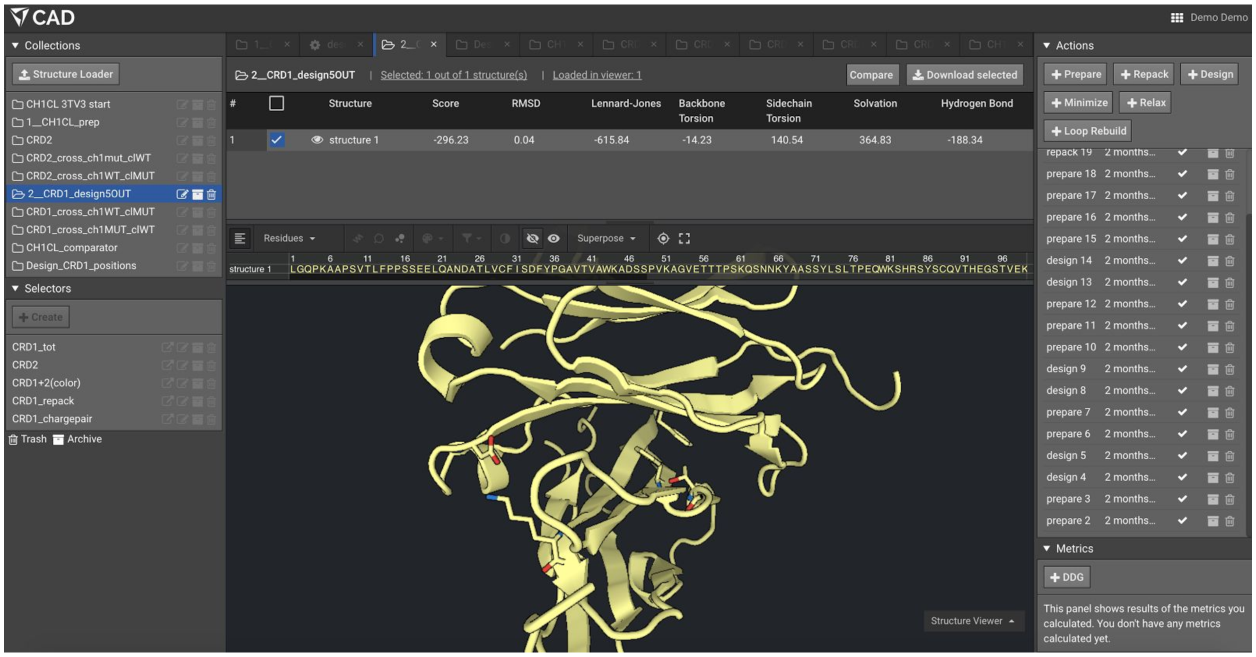
Actions: Prepare; Repack; Design; Minimize; Relax; Loop Rebuild

Task	Updated
design 21	a month ...
prepare 20	2 months...
repack 19	2 months...
prepare 18	2 months...
prepare 17	2 months...
prepare 16	2 months...
prepare 15	2 months...
design 14	2 months...
design 13	2 months...
prepare 12	2 months...
prepare 11	2 months...
prepare 10	2 months...
design 9	2 months...
design 8	2 months...
prepare 7	2 months...
prepare 6	2 months...
design 5	2 months...

Metrics: DDG

This panel shows results of the metrics you calculated. You don't have any metrics calculated yet.

3. Set up a design simulation starting from the prepared structure to install the mutation pattern of CRD1 (K129D, L135F on chain L; D146K, V190F, F174T on chain H).These design simulations should produce multiple decoys, perhaps 20.



CAD — Demo Demo

Collections: Structure Loader; CH1CL 3TV3 start; 1__CH1CL_prep; CRD2; CRD2_cross_ch1mut_clWT; CRD2_cross_ch1WT_clMUT; 2__CRD1_design5OUT; CRD1_cross_ch1WT_clMUT; CRD1_cross_ch1MUT_clWT; CH1CL_comparator; Design_CRD1_positions

Selectors: Create; CRD1_tot; CRD2; CRD1+2(color); CRD1_repack; CRD1_chargepair

Trash Archive

2__CRD1_design5OUT | Selected: 1 out of 1 structure(s) | Loaded in viewer: 1 — Compare — Download selected

#	Structure	Score	RMSD	Lennard-Jones	Backbone Torsion	Sidechain Torsion	Solvation	Hydrogen Bond
1	structure 1	-296.23	0.04	-615.84	-14.23	140.54	364.83	-188.34

```
            1    6    11   16   21   26   31   36   41   46   51   56   61   66   71   76   81   86   91   96
structure 1 LGQPKAAPSVTLFPPSSEELQANDATLVCFISDFYPGAVTVAWKADSSPVKAGVETTTPSKQSNNKYAASSYLSLTPEQWKSHRSYSCQVTHEGSTVEK
```

Actions: Prepare; Repack; Design; Minimize; Relax; Loop Rebuild

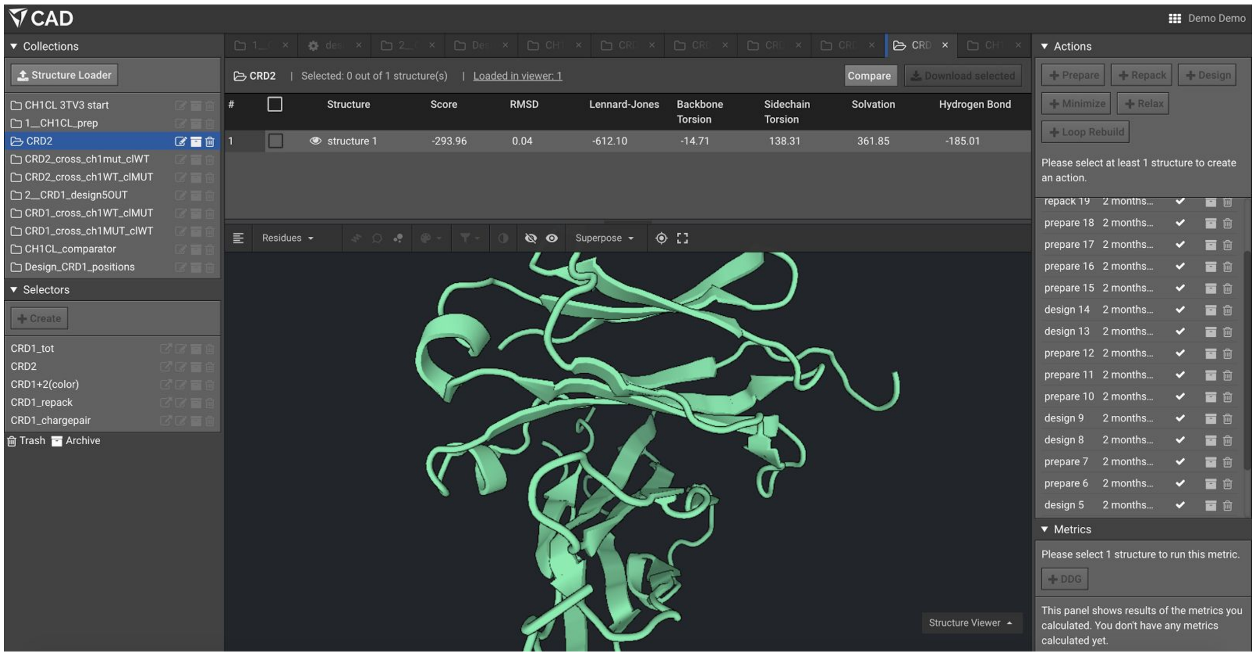
Task	Updated
repack 19	2 months...
prepare 18	2 months...
prepare 17	2 months...
prepare 16	2 months...
prepare 15	2 months...
design 14	2 months...
design 13	2 months...
prepare 12	2 months...
prepare 11	2 months...
prepare 10	2 months...
design 9	2 months...
design 8	2 months...
prepare 7	2 months...
prepare 6	2 months...
design 5	2 months...
design 4	2 months...
prepare 3	2 months...
prepare 2	2 months...

Metrics: DDG

This panel shows results of the metrics you calculated. You don't have any metrics calculated yet.

4. Set up a design simulation starting from the prepared structure to install the mutation pattern of CRD2 (S176W, L135Y on chain L; F174G, H172A on chain H). These design simulations should produce multiple decoys, perhaps 20.

5. Run ‘prepare’ on the best-scoring design results for each of the simulations in (3) and (4).



CH1CL 3TV3 start
1__CH1CL_prep
CRD2
CRD2_cross_ch1mut_clWT
CRD2_cross_ch1WT_clMUT
2__CRD1_design5OUT
CRD1_cross_ch1WT_clMUT
CRD1_cross_ch1MUT_clWT
CH1CL_comparator
Design_CRD1_positions

CRD1_tot
CRD2
CRD1+2(color)
CRD1_repack
CRD1_chargepair

CRD2 | Selected: 0 out of 1 structure(s) | Loaded in viewer: 1

#	Structure	Score	RMSD	Lennard-Jones	Backbone Torsion	Sidechain Torsion	Solvation	Hydrogen Bond
1	structure 1	-293.96	0.04	-612.10	-14.71	138.31	361.85	-185.01

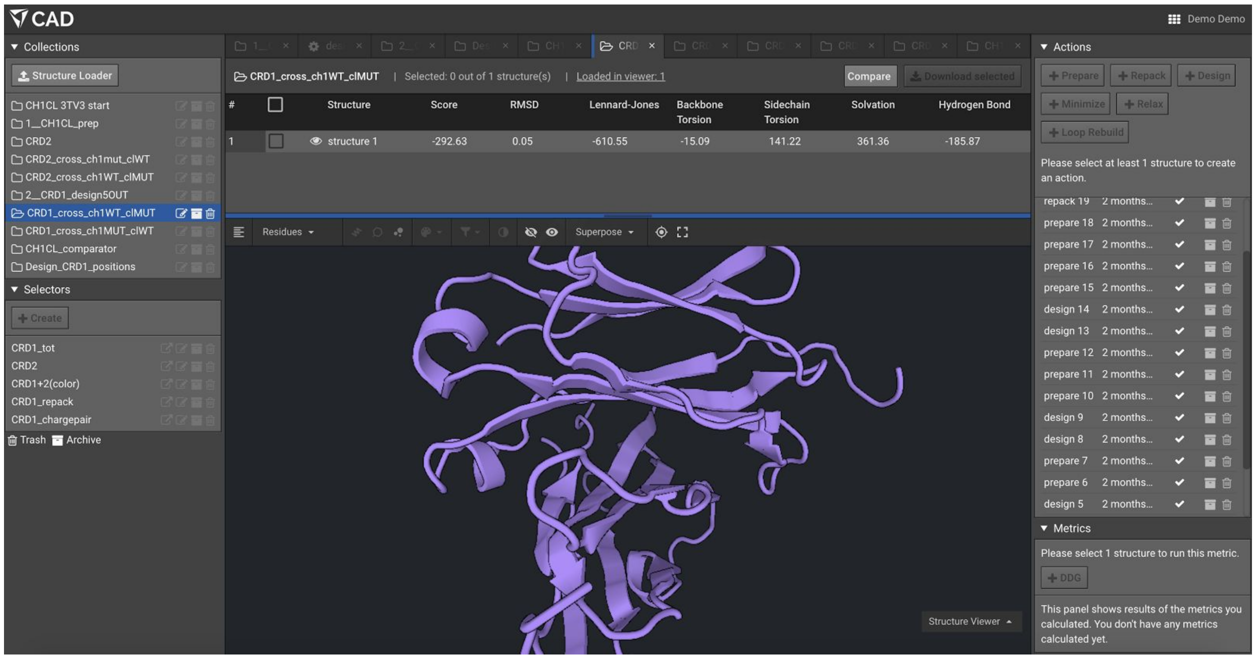
Please select at least 1 structure to create an action.

repack 19	2 months...	✓
prepare 18	2 months...	✓
prepare 17	2 months...	✓
prepare 16	2 months...	✓
prepare 15	2 months...	✓
design 14	2 months...	✓
design 13	2 months...	✓
prepare 12	2 months...	✓
prepare 11	2 months...	✓
prepare 10	2 months...	✓
design 9	2 months...	✓
design 8	2 months...	✓
prepare 7	2 months...	✓
prepare 6	2 months...	✓
design 5	2 months...	✓

Please select 1 structure to run this metric.

This panel shows results of the metrics you calculated. You don't have any metrics calculated yet.

6. Set up design-and-prepare simulations for each of the cross-products above. For example, to model the complex between wild-type light chain and CRD1 heavy chain, we need to make mutations D146K, V190F, F174T on chain H.



#		Structure	Score	RMSD	Lennard-Jones	Backbone Torsion	Sidechain Torsion	Solvation	Hydrogen Bond
1		structure 1	-292.63	0.05	-610.55	-15.09	141.22	361.36	-185.87

Please select at least 1 structure to create an action.

repack 19	2 months...	✓
prepare 18	2 months...	✓
prepare 17	2 months...	✓
prepare 16	2 months...	✓
prepare 15	2 months...	✓
design 14	2 months...	✓
design 13	2 months...	✓
prepare 12	2 months...	✓
prepare 11	2 months...	✓
prepare 10	2 months...	✓
design 9	2 months...	✓
design 8	2 months...	✓
prepare 7	2 months...	✓
prepare 6	2 months...	✓
design 5	2 months...	✓

Please select 1 structure to run this metric.

This panel shows results of the metrics you calculated. You don't have any metrics calculated yet.

7. Repack the native structure (20 decoys) and 'prepare' the top few decoys as above to obtain a wild-type score that is comparable to the above.

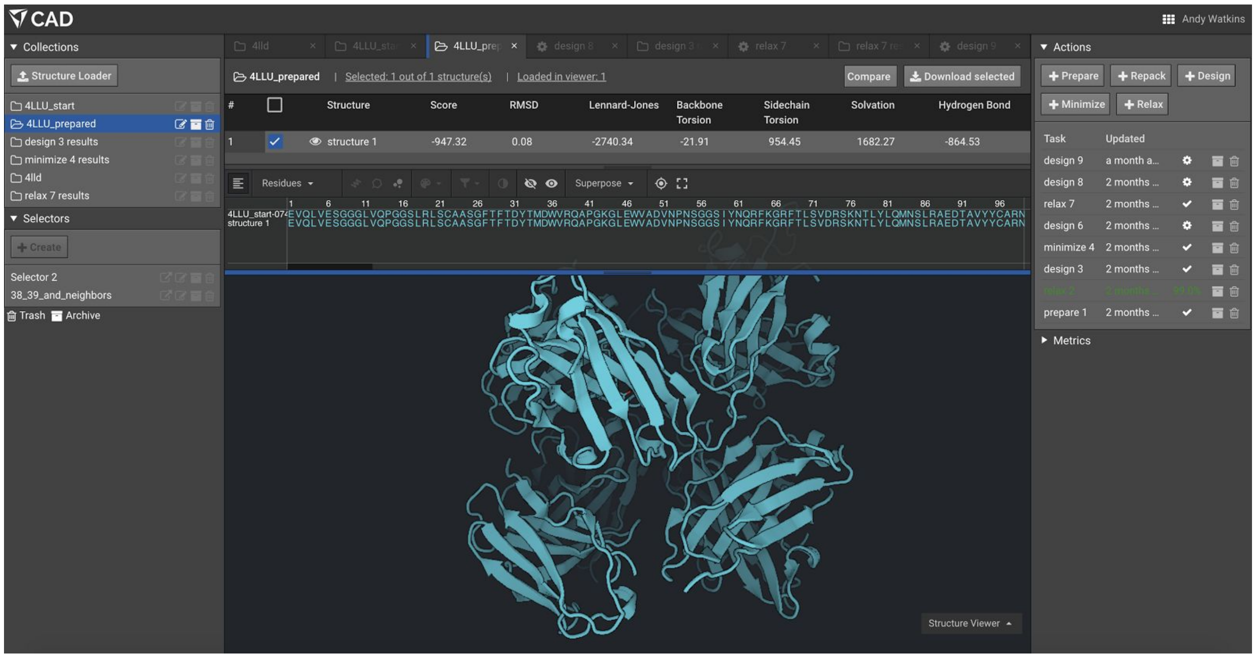
8. Compare the energies of the wild-type and the redesigns against the cross-interfaces. The wildtype and CRD1 should both have energies of about -296, with its cross-interfaces scoring -293 and -291; CRD2 has an energy of -294 and its cross-interfaces score -293 and -273.

Part Two

Second, we make two variable region redesigns that ought to be bispecific against *each other*:

1. In Bench, fetch 4LLU. This structure contains the Fab region of the heavy (H) and light (L) chains. It actually contains two copies, so you can either apply the following operations to each chain and divide any resulting energies by two, or you can manually remove two chains and then proceed with that PDB file. The latter method will run much faster, so we will assume that you have removed chains C and D in a text editor.

2.	Run ‘prepare’, producing a single decoy.



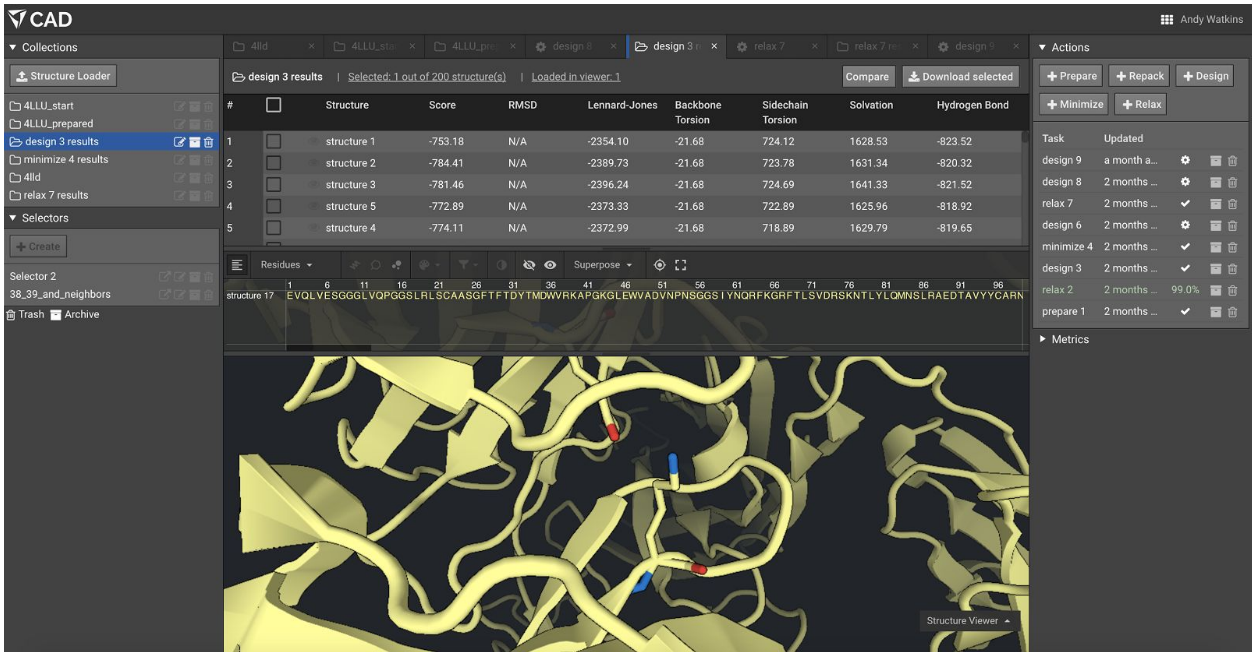
4LLU_prepared | Selected: 1 out of 1 structure(s) | Loaded in viewer: 1

#		Structure	Score	RMSD	Lennard-Jones	Backbone Torsion	Sidechain Torsion	Solvation	Hydrogen Bond
1	✓	structure 1	-947.32	0.08	-2740.34	-21.91	954.45	1682.27	-864.53

```
                1    6    11   16   21   26   31   36   41   46   51   56   61   66   71   76   81   86   91   96
4LLU_start-074 EVQLVESGGGLVQPGGSLRLSCAASGFTFTDYTMDWVRQAPGKGLEWVADVNPNSGGSIYNQRFKGRFTLSVDRSKNTLYLQMNSLRAEDTAVYYCARN
structure 1    EVQLVESGGGLVQPGGSLRLSCAASGFTFTDYTMDWVRQAPGKGLEWVADVNPNSGGSIYNQRFKGRFTLSVDRSKNTLYLQMNSLRAEDTAVYYCARN
```

Task	Updated	
design 9	a month a...	⚙
design 8	2 months ...	⚙
relax 7	2 months ...	✓
design 6	2 months ...	⚙
minimize 4	2 months ...	✓
design 3	2 months ...	✓
relax 2	2 months ...	99.0%
prepare 1	2 months ...	✓

3.	Set up a design simulation starting from the prepared structure to install the mutation pattern of VRD1 (D1R, Q38D on chain B; R63E, Q39K on chain A). These design simulations should produce multiple decoys, perhaps 20.



design 3 results | Selected: 1 out of 200 structure(s) | Loaded in viewer: 1

#		Structure	Score	RMSD	Lennard-Jones	Backbone Torsion	Sidechain Torsion	Solvation	Hydrogen Bond
1		structure 1	-753.18	N/A	-2354.10	-21.68	724.12	1628.53	-823.52
2		structure 2	-784.41	N/A	-2389.73	-21.68	723.78	1631.34	-820.32
3		structure 3	-781.46	N/A	-2396.24	-21.68	724.69	1641.33	-821.52
4		structure 5	-772.89	N/A	-2373.33	-21.68	722.89	1625.96	-818.92
5		structure 4	-774.11	N/A	-2372.99	-21.68	718.89	1629.79	-819.65

```
              1    6    11   16   21   26   31   36   41   46   51   56   61   66   71   76   81   86   91   96
structure 17 EVQLVESGGGLVQPGGSLRLSCAASGFTFTDYTMDWVRKAPGKGLEWVADVNPNSGGSIYNQRFKGRFTLSVDRSKNTLYLQMNSLRAEDTAVYYCARN
```

Task	Updated	
design 9	a month a...	⚙
design 8	2 months ...	⚙
relax 7	2 months ...	✓
design 6	2 months ...	⚙
minimize 4	2 months ...	✓
design 3	2 months ...	✓
relax 2	2 months ...	99.0%
prepare 1	2 months ...	✓

4. Set up a design simulation starting from the prepared structure to install the mutation pattern of VRD2 (Q38R on chain B, Q39Y on chain A). These design simulations should produce multiple decoys, perhaps 20.

5. Run 'prepare' on the best-scoring design results for each of the simulations in (3) and (4).

6. Set up design-and-prepare simulations for each of the cross-products above. For example, to model the complex between VRD1 light chain and VRD2 heavy chain, we need to make mutations D1R, Q38D on chains B and Q39Y on chains A. (This produces an anion-anion charge pair and an anion-pi interaction.)

7. Compare the energy of the desired dimers (VRD1/VRD1; VRD2/VRD2) to the cross-products.

Part Three

Third, we show how to use Bench to create new specificity mediating mutations using the knob-in-hole or charge-swapping strategy. Though in the first two sections we illustrated how to recapitulate and support mutations obtained through expert knowledge, here we establish how one might pick a set of mutations over another by measuring for specificity.

1. Load the trimmed 4LLU PDB we provided previously.

2. Run 'prepare', producing a single decoy.

3. Locate a charge pair at the interface; for this workflow, suppose you located D1 on chain B and R62 on chain A.

4. Set up a selector that includes those two residues and their immediate neighbors.

5. Using a design simulation that repacks those residues' neighbors, force a mutation to one side of the interface to match the charge of the other side -- for the purpose of this workflow, demand that D1 mutate to either K or R. (Do not permit it to remain D or E.)

6. Run 'prepare' on the design results and compare its score to the prepared wildtype structure to confirm that the mismatched charges give a poor score. Pick the worse-scoring sequence of the resulting options.

7. Using the same selector, applied to the design result from step 5, set up a similar design simulation, to demand that R62 mutate to either D or E.

8. Run ‘prepare’ on the design results and compare its score to the prepared wildtype structure to confirm that the swapped charges give a good score. Pick the best-scoring sequence of the resulting options.

FAQ

Q: How can I identify sets of ‘complementary’ mutations that force this bispecific phenomenon?

A: To review, the heterodimerization problem works like this: given one sequence A/B, generate a related new sequence A’/B’ such that the complexes A/B and A’/B’ are greatly favored over formation of A/B’ or A’/B. Two straightforward strategies for generating these new sequences can be implemented in Bench. First, you may use Bench to find locations where well-oriented hydrogen bonding interactions or charge pairs are anchoring an interface together, and make simple pairs of mutations on either side, as we show in Part Three. For example, if A/B have an E/K charge pair, swapping to give A’/B’ a K/E charge pair should disfavor the A’/B complex (K/K) as well as the A/B’ complex (E/E). Second, you may conduct ‘mutation-rescue’ simulations intended to perturb the interface with more aggressive mutations, doing this with large/small amino acid alterations is the classic “knobs in hole” approach to interface engineering (see for example Ridgway JB *et al.*, Protein Eng, 1996). To wit – suppose you force a mutation that introduces a ‘bump’ at the A/B interface, on sequence A. This mutation is unfavorable; it destabilizes the A/B interface to form a poorer A’/B interface. Then, you may set up a design simulation on the interface sequence of B to ‘rescue’ good binding to produce A’/B’. Finally, you may use the procedure outlined above in Parts One and Two to validate your predictions about how these complementary mutations should behave.

Q: Why is the final step for the constant region redesigns important for quantitative energy comparison – why can’t I just compare the CRD1 and CRD2 scores to the wild type score obtained in the second step (initial prepare)?

A: We always want to compare structural models to which the same amount of sampling and optimization have been applied. The design results have been prepared, then globally repacked, then prepared again. Comparing to the original structure that has just been prepared once and experienced no global repack is ‘unfair’.