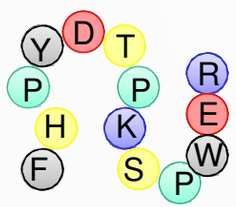
Affinity Maturation

- MeilerLab Rosetta Workshop April 2019
- Samuel Schmitz

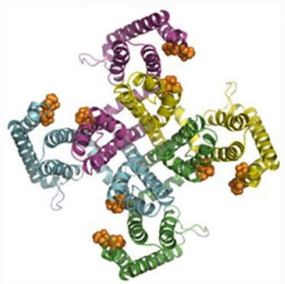
Protein design is the inverse folding problem



Protein Folding Problem

Sequence ⇌ Structure

Inverse Folding Problem

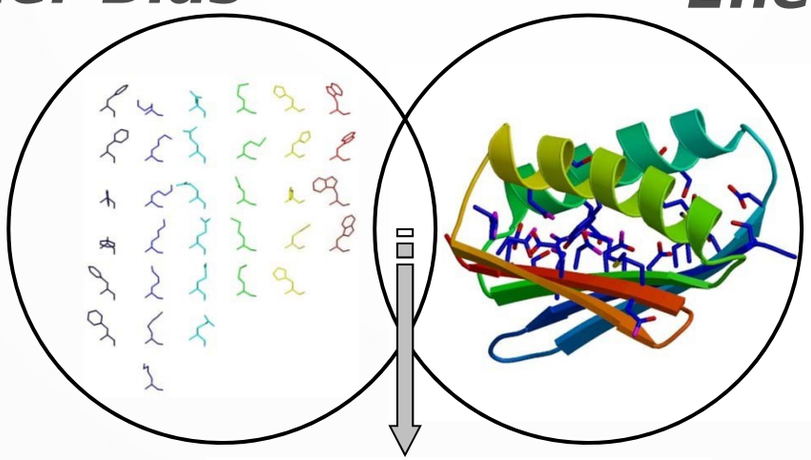


Given a protein fold – which primary sequence(s) can fold into it?

Using rotamer libraries and the Rosetta scoring function to optimize packing

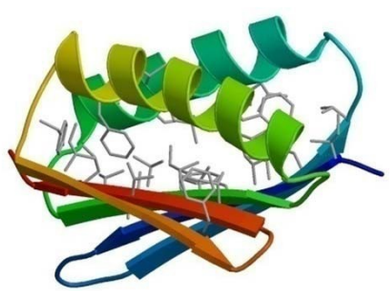
Local Rotamer Bias

Approximate interactions between sidechains using the distribution of sidechain conformations seen in known protein structures



Energy function

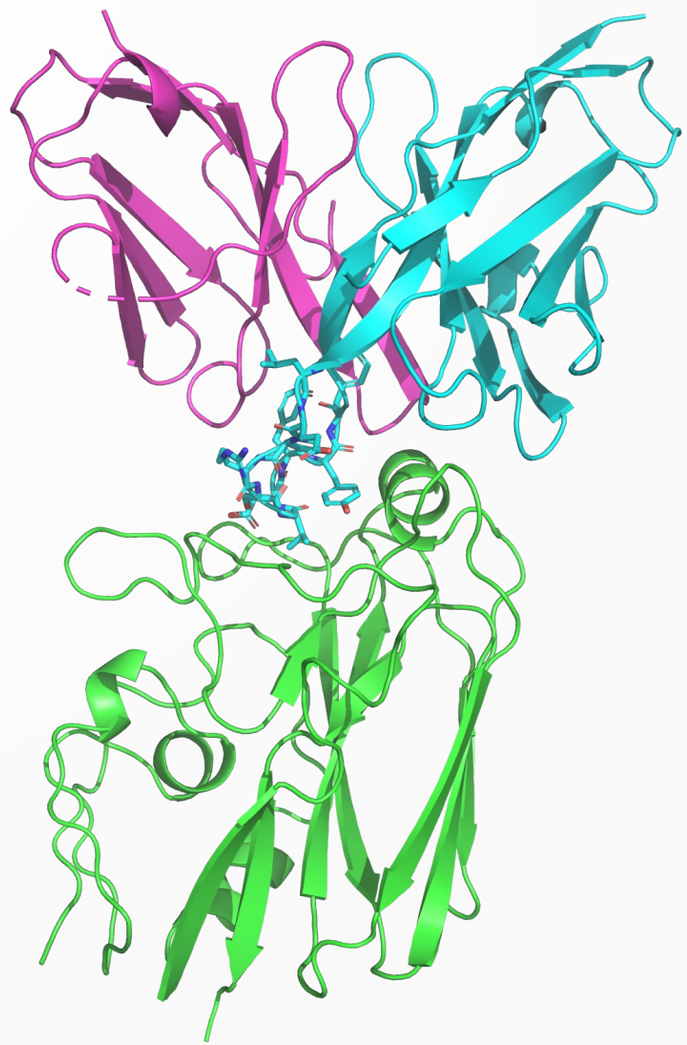
- ***VDW interactions***
- ***solvation***
- ***hydrogen bonding potential***
- **elec *interactions***
- ***rotamer probability***



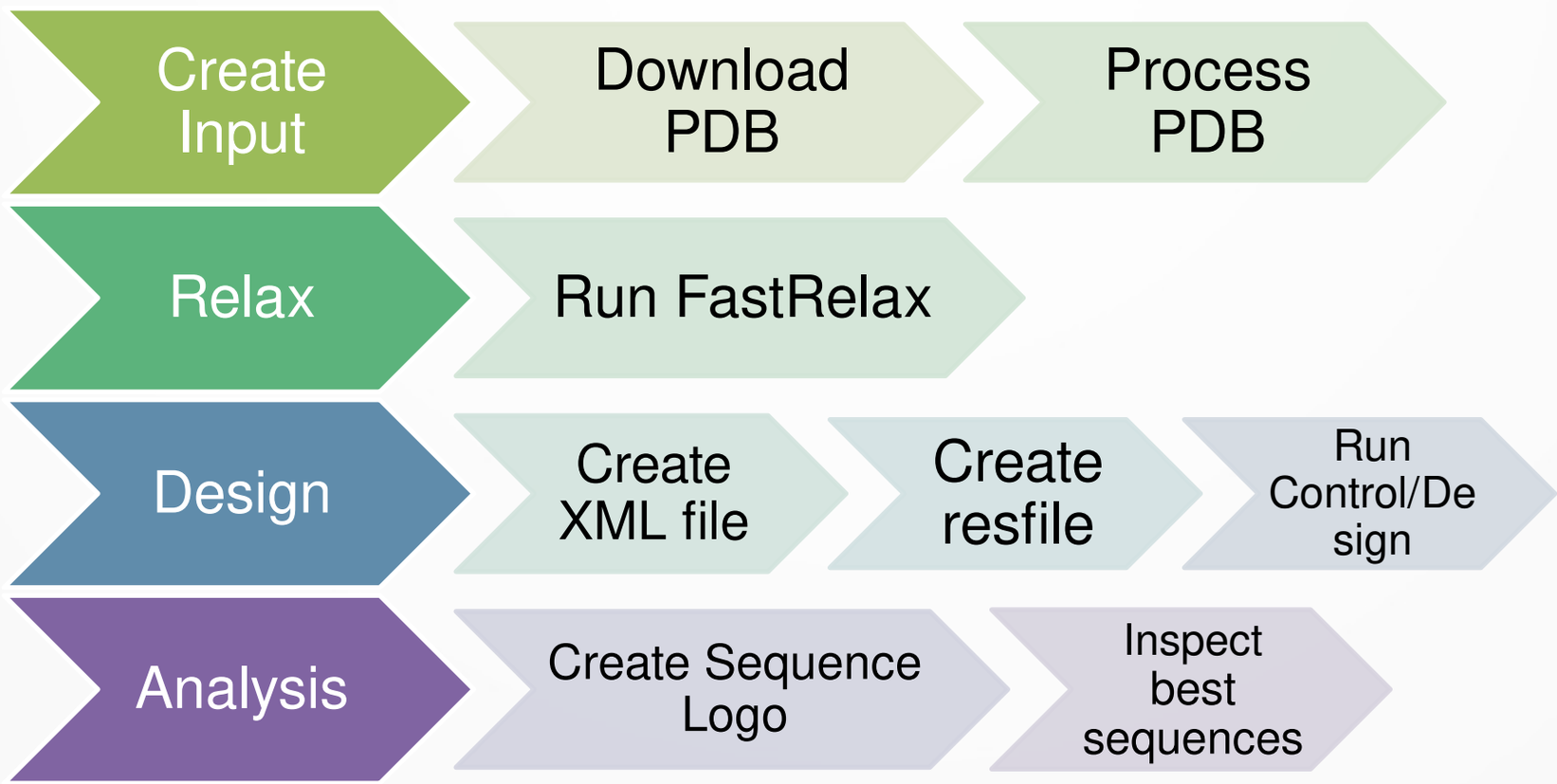
Simulated Annealing
Monte Carlo optimization

Affinity maturation of the antibody-antigen complex

- Sequence redesign of the protein-protein sequence
- Using one conformation or multiple conformations (“**state**”)
- *In-silico* affinity maturation to improve the interface score for tighter binding



Single state design overview

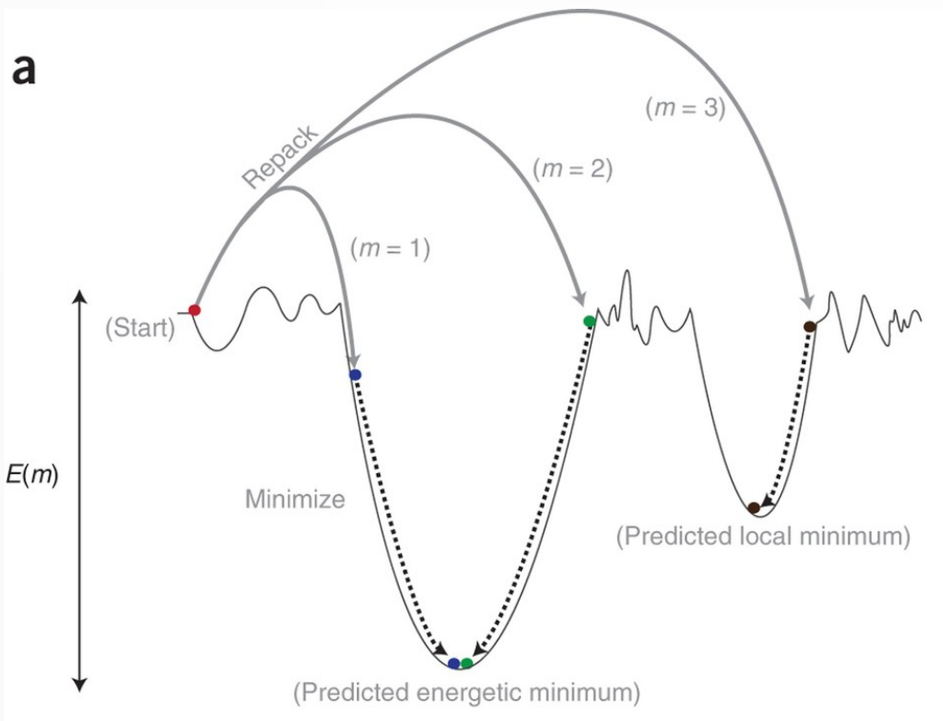


Relax

Run FastRelax

FastRelax is designed to optimize the protein backbone/side chains to model at an energy minimum

Helps relieve clashes that may introduce artifacts into design



Combs, *et al*, Nat. Prot. 2013

Please open
~/rosetta_workshop/tutorials/protein_design/single_state_design/input_files/design.xml

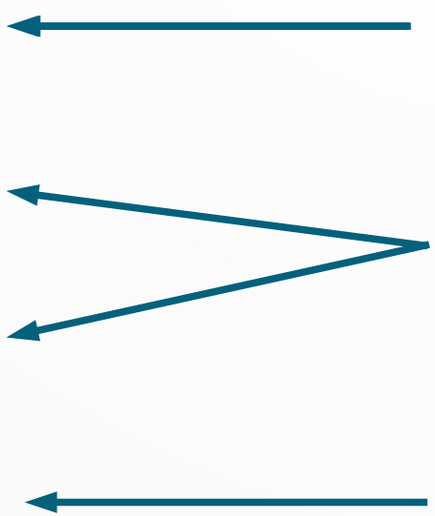
Where should you start looking?

```
<PROTOCOLS>
    Run the design protocol
        <Add mover="design" />

    Calculate interface metrics for the final sequence
        <Add mover="analyze" />
</PROTOCOLS>
```

```
Design and repack residues based on resfile
<ReadResfile name="rrf" filename="4HKX.resfile"/>
```

```
NATRO
start
30 H ALLAA
31 H ALLAA
...
152 L ALLAA
155 L ALLAA
...
333 A NATAA
334 A NATAA
```



Do only design the residues specified below (the interface)

Free sequence design on heavy and light chain interface residues

Repack the antigen, do not mutate

Use the python script located in
~/rosetta_workshop/tutorials/protein_design/scripts/define_interface.py

Calculates residues on each side of the interface using a side chain cutoff (default 5 A)

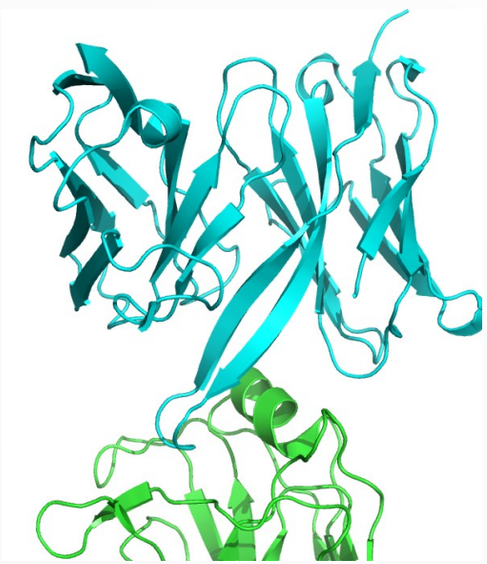
If any atom of a residue is within 5 A of any atom of a residue on the opposing chain – it's considered to be an interface residue

Single State Design (SSD):

- Optimize the sequence for one antibody-antigen conformation (**design**)
- Repack the wild-type sequence to descend to the lowest possible score (**control**)

Antibody

Antigen 1



PDB ID 4yk4

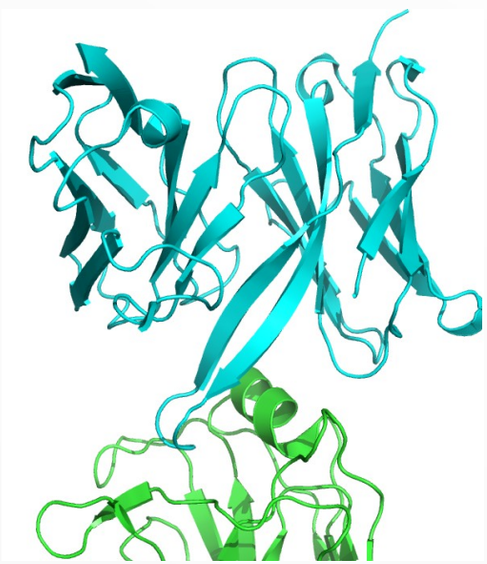
Sevy, A. M., Jacobs, T. M., Crowe, J. E. & Meiler, J. *PLoS Comput. Biol.* **11,** e1004300 (2015).

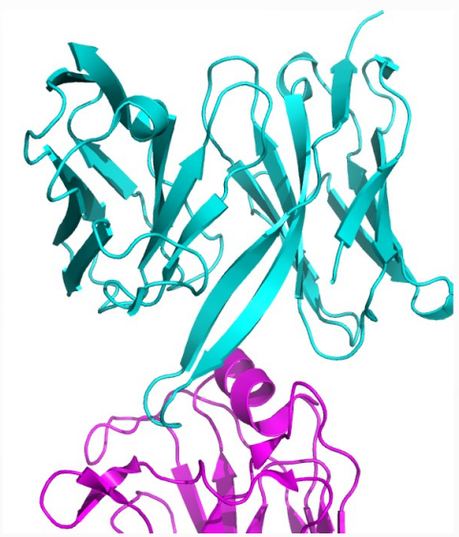
Multi State Design (MSD):

- Optimize the sequence for **two** antibody-antigen conformations (**design**)
- Repack the wild-type sequence to descend to the lowest possible score for **each** state (**control**)

Antibody

Antigen 1

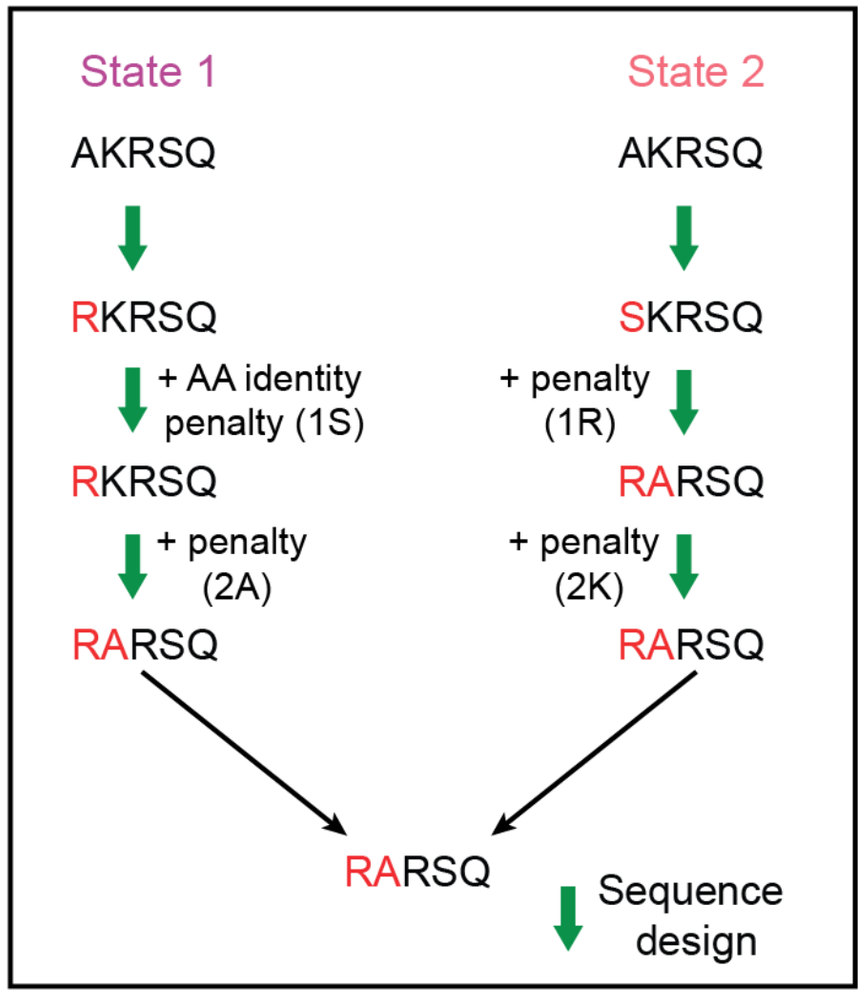




Antigen 2

PDB ID 4yk4

REstrained **CON**vergence (RECON)



Sevy, A. M., Jacobs, T. M., Crowe, J. E. & Meiler, J. *PLoS Comput. Biol.* **11,** e1004300 (2015).

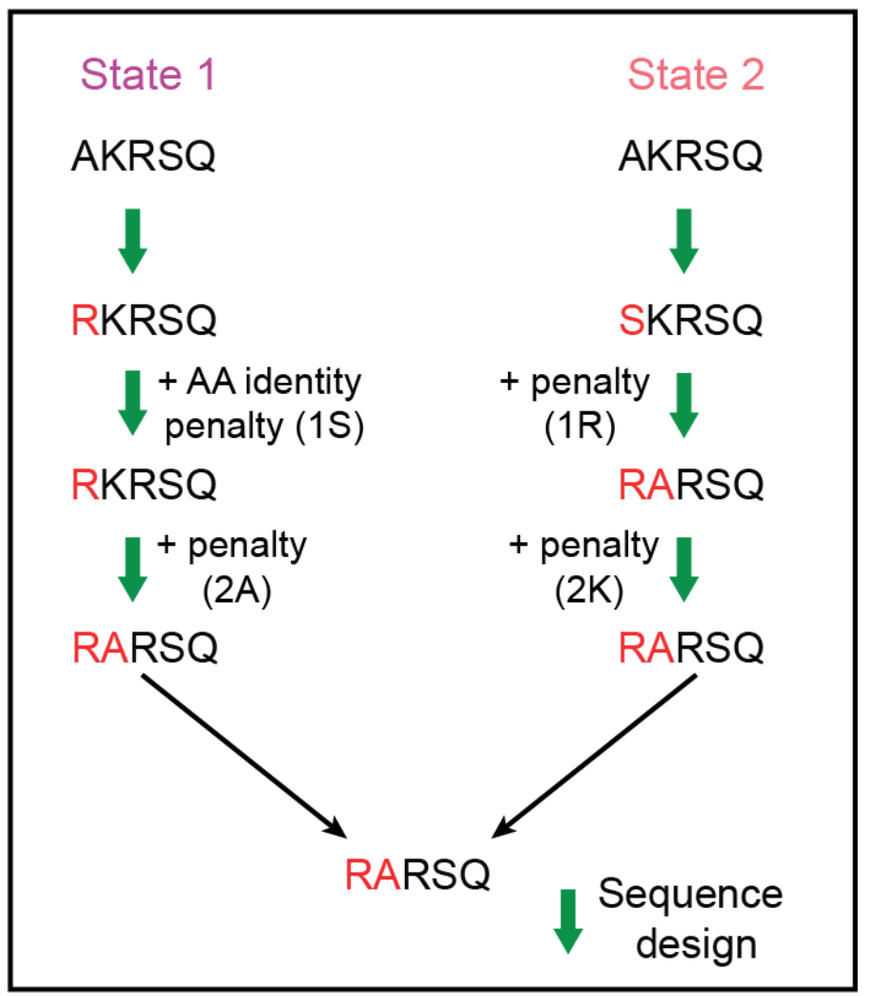
REstrained **CON**vergence (RECON)

MSDMover 1

MSDMover 2

MSDMover 3

FindConsensusSequence



Sevy, A. M., Jacobs, T. M., Crowe, J. E. & Meiler, J. *PLoS Comput. Biol.* **11,** e1004300 (2015).

Please open
~/rosetta_workshop/tutorials/protein_design/multi_state_design/input_files/
design.xml

```
<PROTOCOLS>
    Run four rounds of design
    <Add mover=msd1 />
    <Add mover=msd2 />
    <Add mover=msd3 />
    <Add mover=msd4 />

    Find a consensus sequence for all states
     <Add mover=finish />

    Calculate interface metrics for the final sequence
     <Add mover=analyze />
</PROTOCOLS>
```

Multiple design operations with gradually forcing the design to one consensus sequence

Agree on the final consensus sequence (if yet unclear)

Analysis

- **Total score:** score of the entire complex
- **Interface score:** score of residues that are at the interface
- **Binding energy:** difference in energy between the bound and unbound partners

$$ddG - dG\_separated$$

- **Binding density:** ddG divided by the buried surface area. Prevents a low binding energy by increasing buried surface area.

$$\frac{dG\_separated}{dSASA \times 100}$$

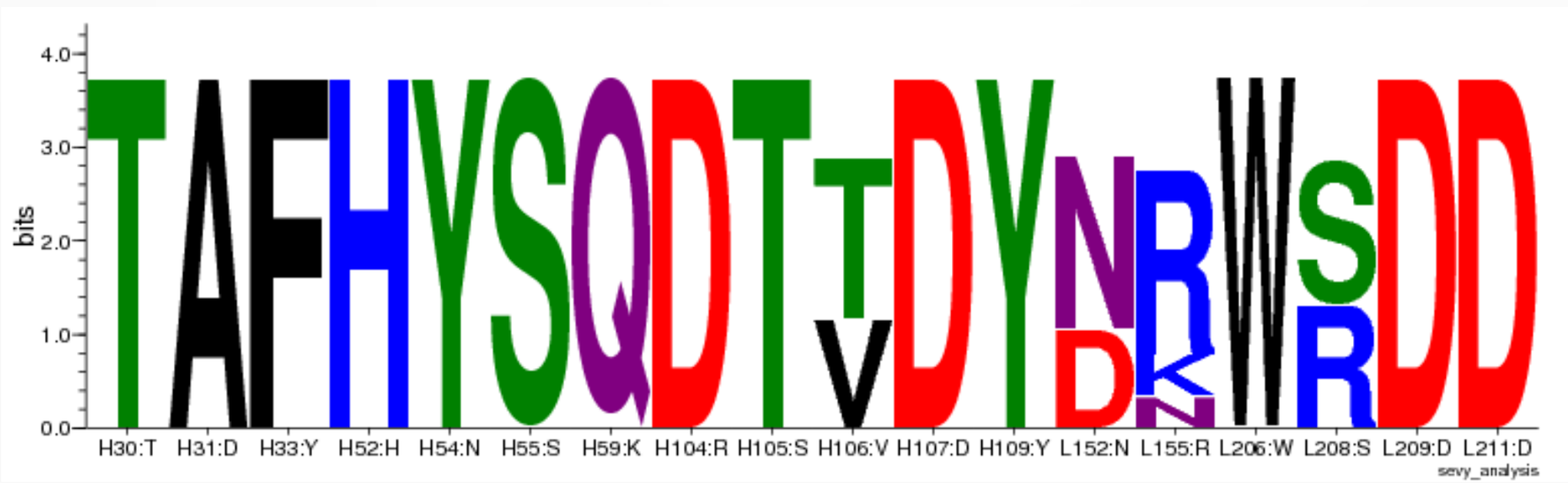
Analysis

```
<InterfaceAnalyzerMover name="analyze" scorefxn="REF2015"
packstat="0" pack_input="0" pack_separated="1"
fixedchains="H,L" />
```

- packstat: activates packstat calculation (packing statistics, Rosetta holes); can be slow so it defaults to off
- fixedchains: comma-delimited list of chain ids to define a group in the interface.
- pack_separated: repack the exposed interfaces when calculating binding energy? Usually a good idea.
- pack_input: prepack before separating chains when calculating binding energy? Useful if these are non-Rosetta inputs

Useful to quickly see which residues are being designed, and what amino acids are being put there

Made by WebLogo application through
~/rosetta_workshop/tutorials/protein_design/scripts/design_analysis.py



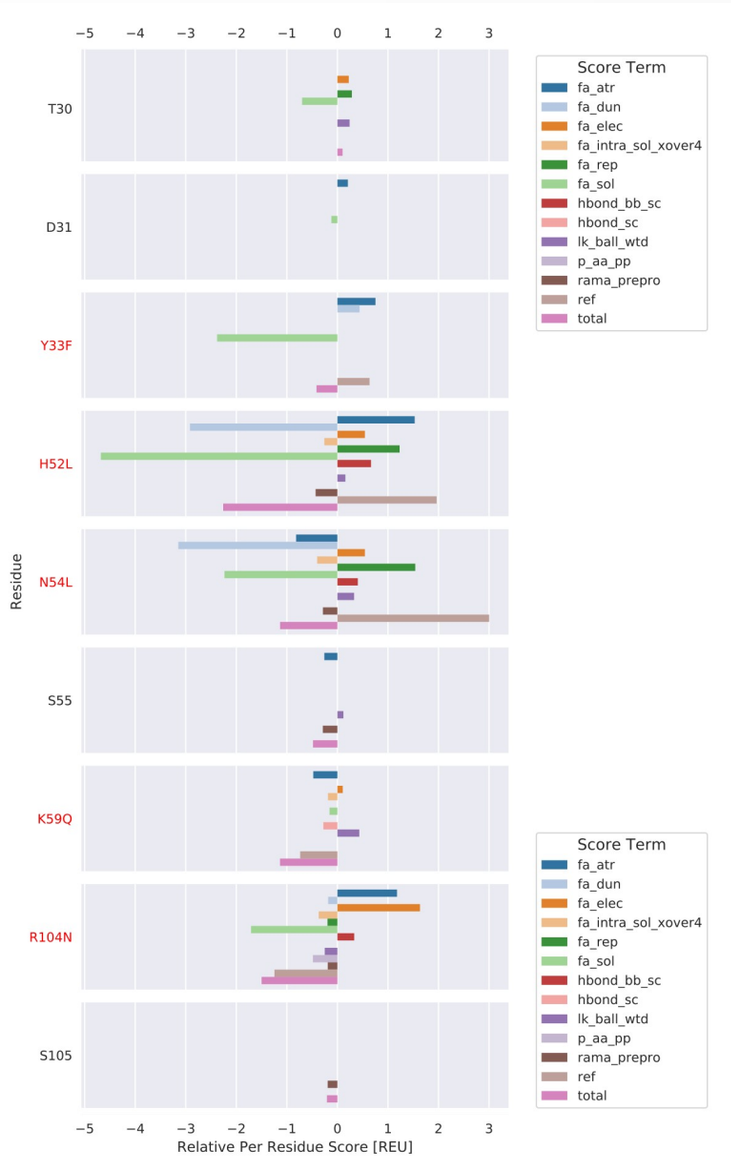
http://weblogo.berkeley.edu/

- Position specific score changes for each Rosetta scoring term
- REF 2015 scoring terms:

The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design
Alford et al (2017)

- Made by supplementary script through

~/rosetta_workshop/tutorials/protein_design/scripts/PerResidueEnergies.py



Please begin the Rosetta protein design tutorial found at

`~/rosetta_workshop/tutorials/protein_design/`