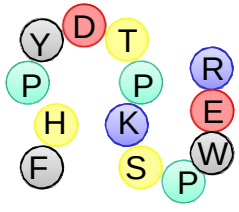
Protein Design

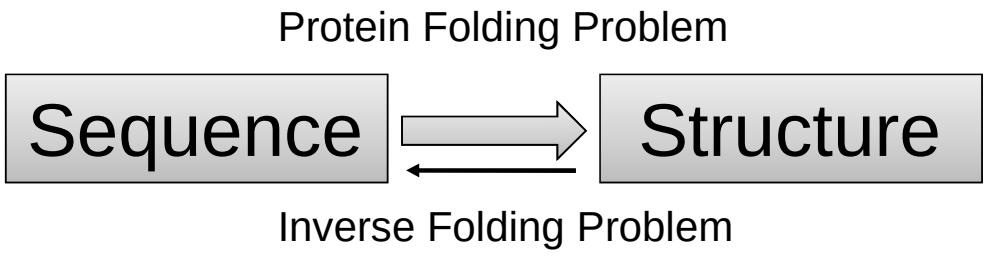
Marion Sauer

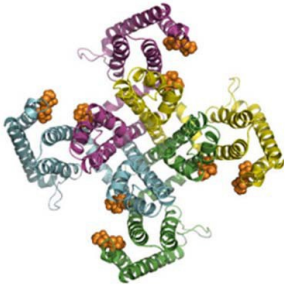
Rosetta Workshop

November 8, 2018

Protein design is the inverse protein-folding problem







Given a protein fold, which sequence(s) can assume that particular peptide geometry?

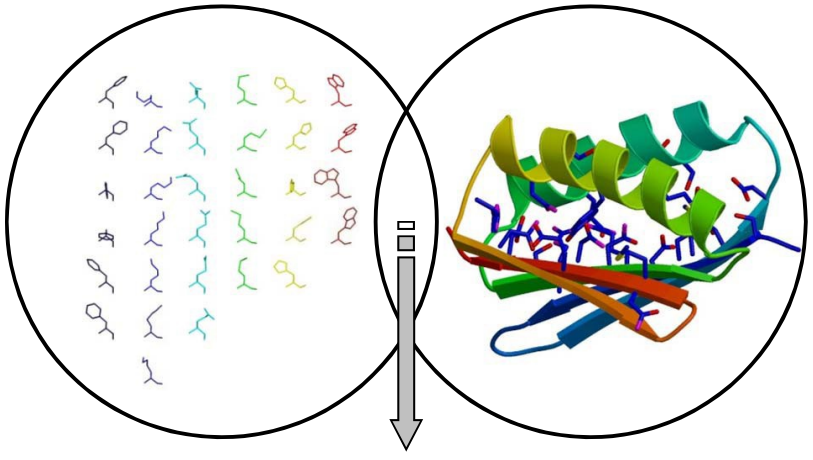
Using local rotamer libraries and the Rosetta energy function to optimize side chain placement

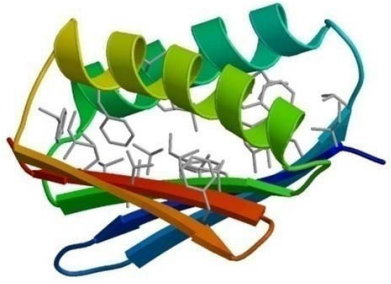
Local Rotamer Bias

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

Energy function

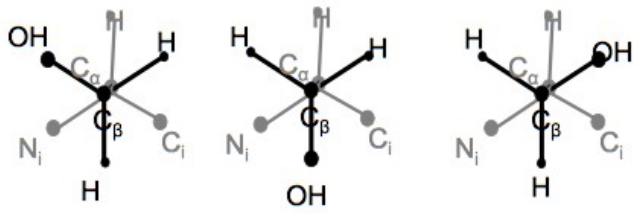
- *VDW interactions*
- *solvation*
- *hydrogen bonding potential*
- elec *interactions*
- *rotamer probability*

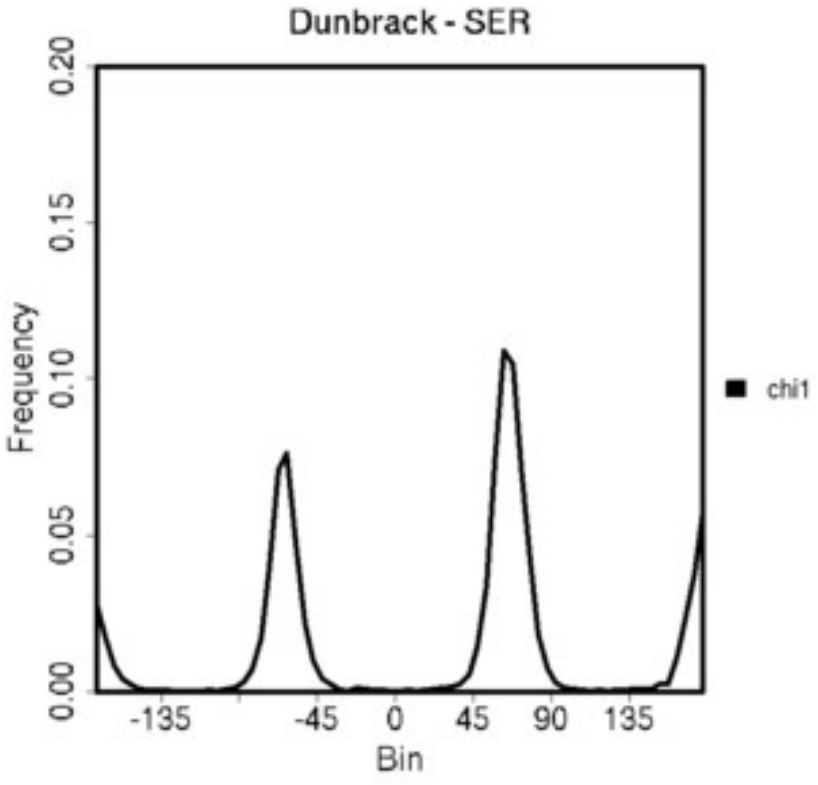




Simulated Annealing Monte Carlo optimization

Side chain rotamer optimization



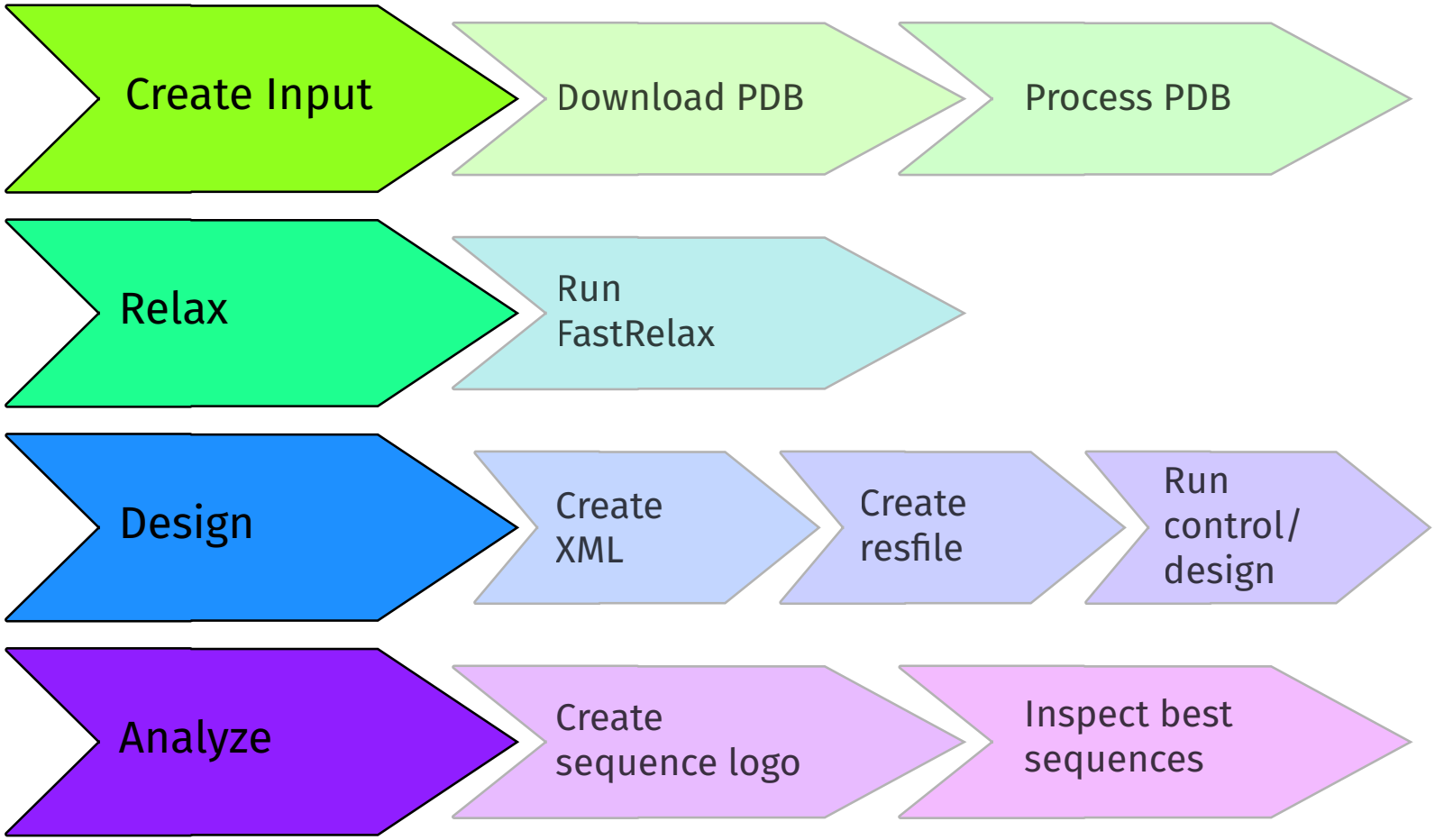


Rotamer = **Rota**tional iso**mer**

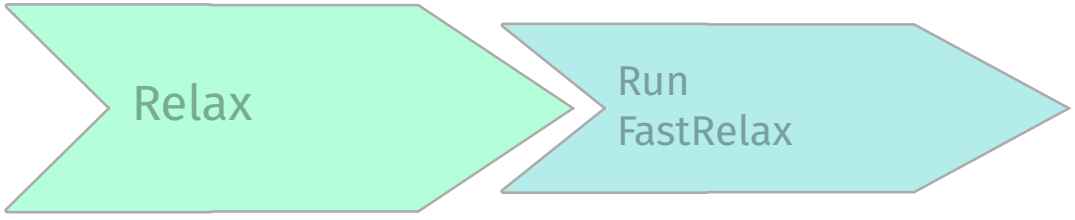
Sampling all χ angles is computationally expensive

Rotamer configurations most likely fall into discrete bins —simplify compute time with rotamer libraries

Rosetta Design protocol overview



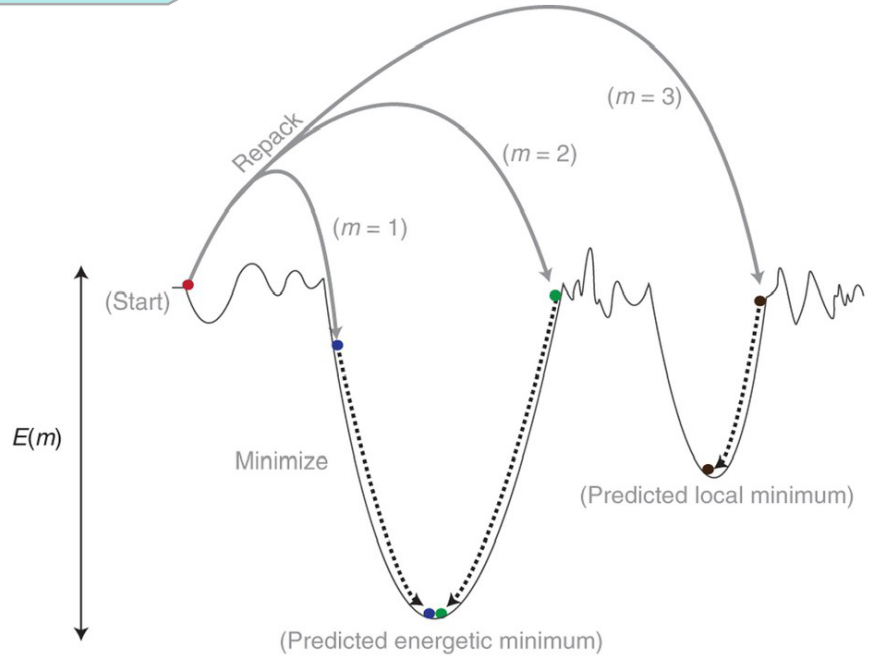
FastRelax



Cycles between minimization and repacking to optimize backbone and side chain coordinates/torsion angles

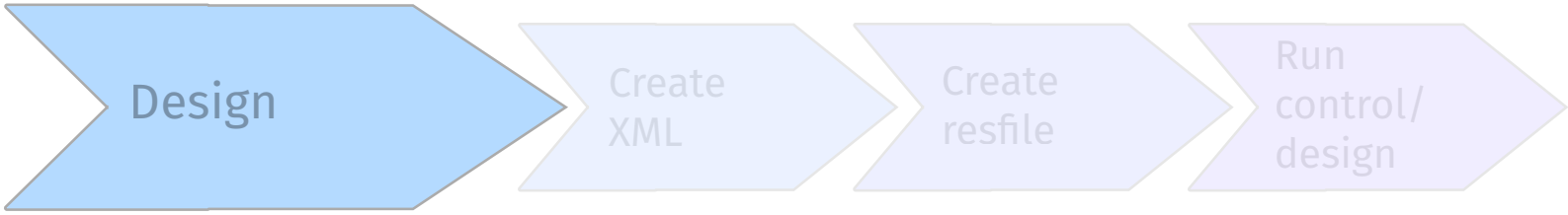
Benchmarks have shown that as little as 1 Å backbone perturbations can completely change sampled sequences

ALWAYS use
`-relax:constrain_relax_to_start_coords`



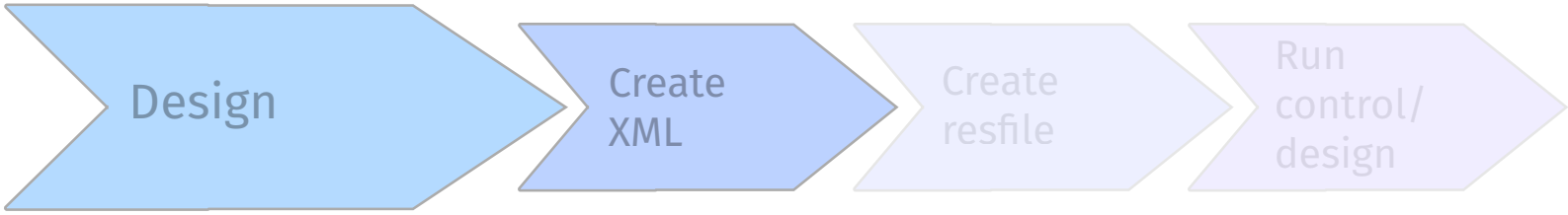
Combs, S. et al. (2013) Nature Protocols

Design with Rosetta Scripts



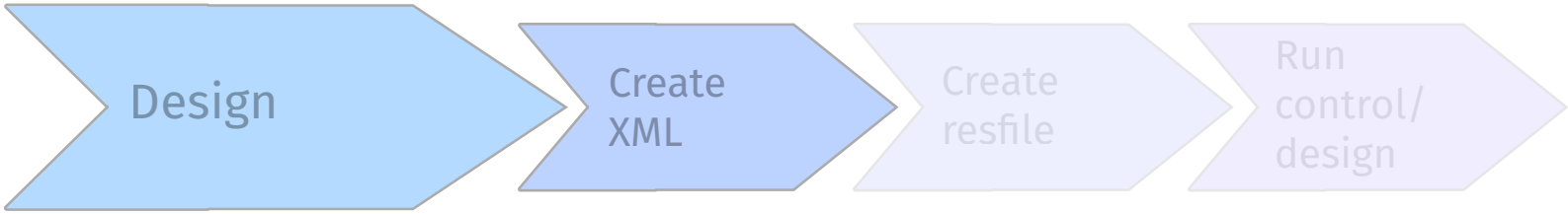
To run Rosetta design, where do you begin?

Design with Rosetta Scripts



Please open single_state_design/input_files/design.xml

Design with Rosetta Scripts

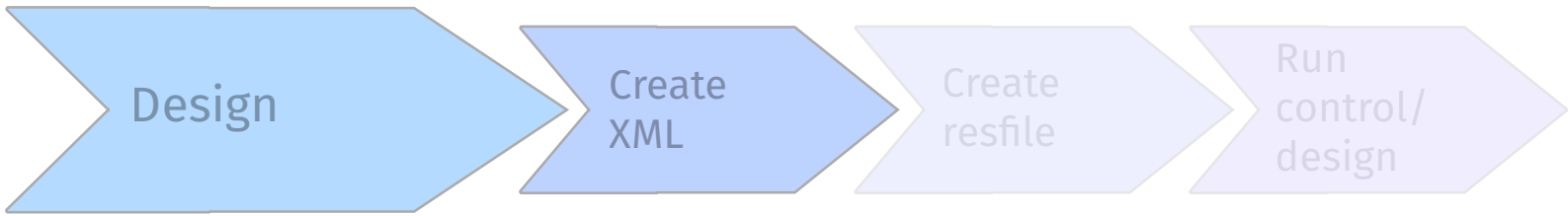


Please open single_state_design/input_files/design.xml

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

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

Using movers in an XML file



Design Mover

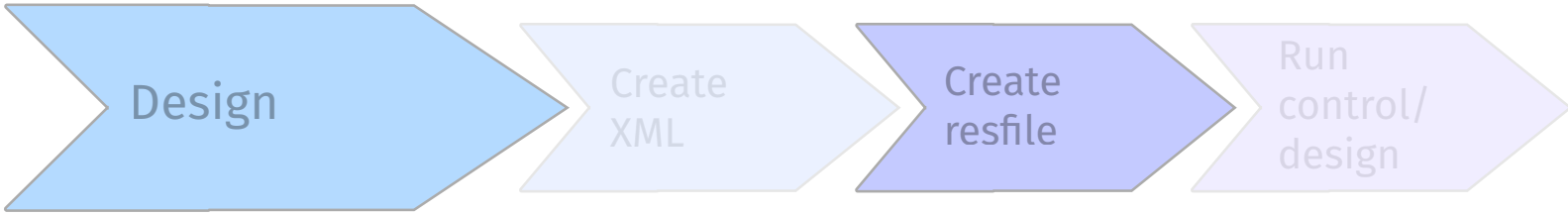
```
<PackRotamerMover name="design" scorefxn="REF2015"
task_operations="ifcl,rrf" />
```

Task Operations

```
Include rotamer options from the command line
<InitializeFromCommandLine name="ifcl" />

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

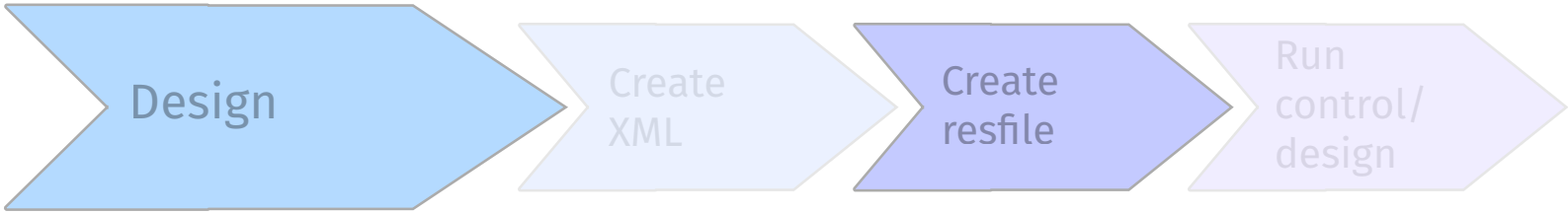
Creating a resfile



A Brief Guide

ALLAA	Allow all 20 canonical amino acids
ALLAAxc	Allow all 20 canonical amino acids excluding cysteine
POLAR	Allow only polar canonical amino acids
APOLAR	Allow only apolar canonical amino acids
PIKAA	Allow only the specified amino acids
NOTAA	Disallow only the specified amino acids
NATAA	Allow only the native amino acid (repack)
NATRO	Preserve the native rotamer
EMPTY	Disallow all 20 canonical amino acids
NC	Allow the specific non-canonical amino acid

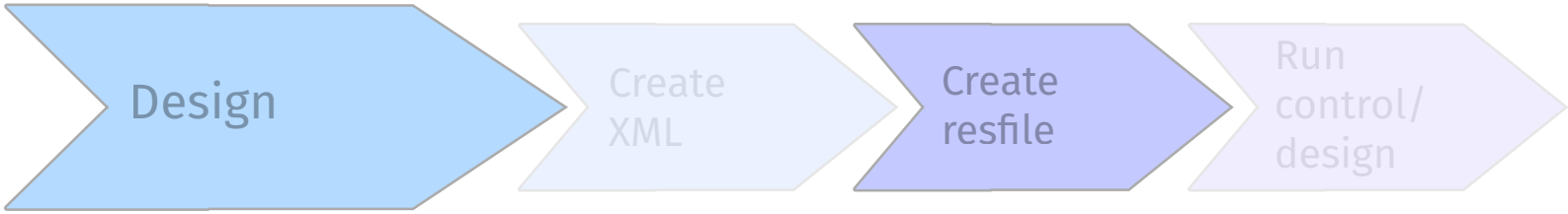
Creating a resfile



A Brief Guide

```
<Header> # instructions for all residues not mentioned in the body
START # Keyword! Defines resfile format
<Body> # specific instructions with format:
<PDB NUM> <PDB CHAIN> <COMMANDS>       # Specify for a residue
          or:
* <PDB CHAIN> <COMMANDS>               # Specify for a chain
```

Resfiles



A Few Examples

A resfile that designs everything

```
ALLAA
START
```

A resfile that does nothing

```
NATRO
START
```

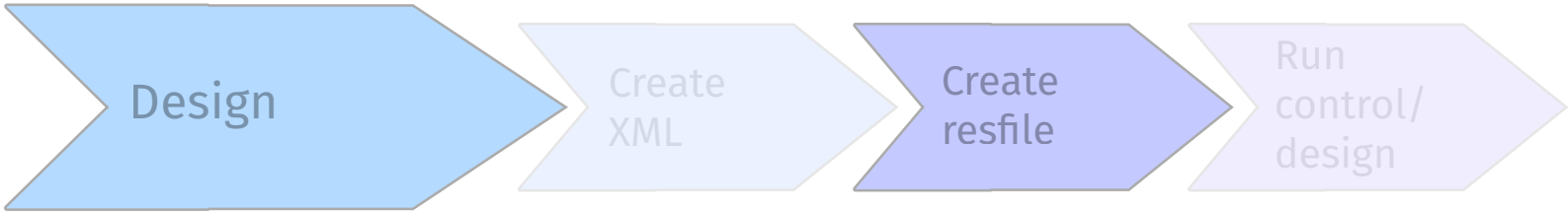
A resfile that repacks only chain H

```
NATRO
START
* H NATAA
```

A resfile that repacks and designs

```
NATAA
START
24 PIKAA LVI
```

Resfiles

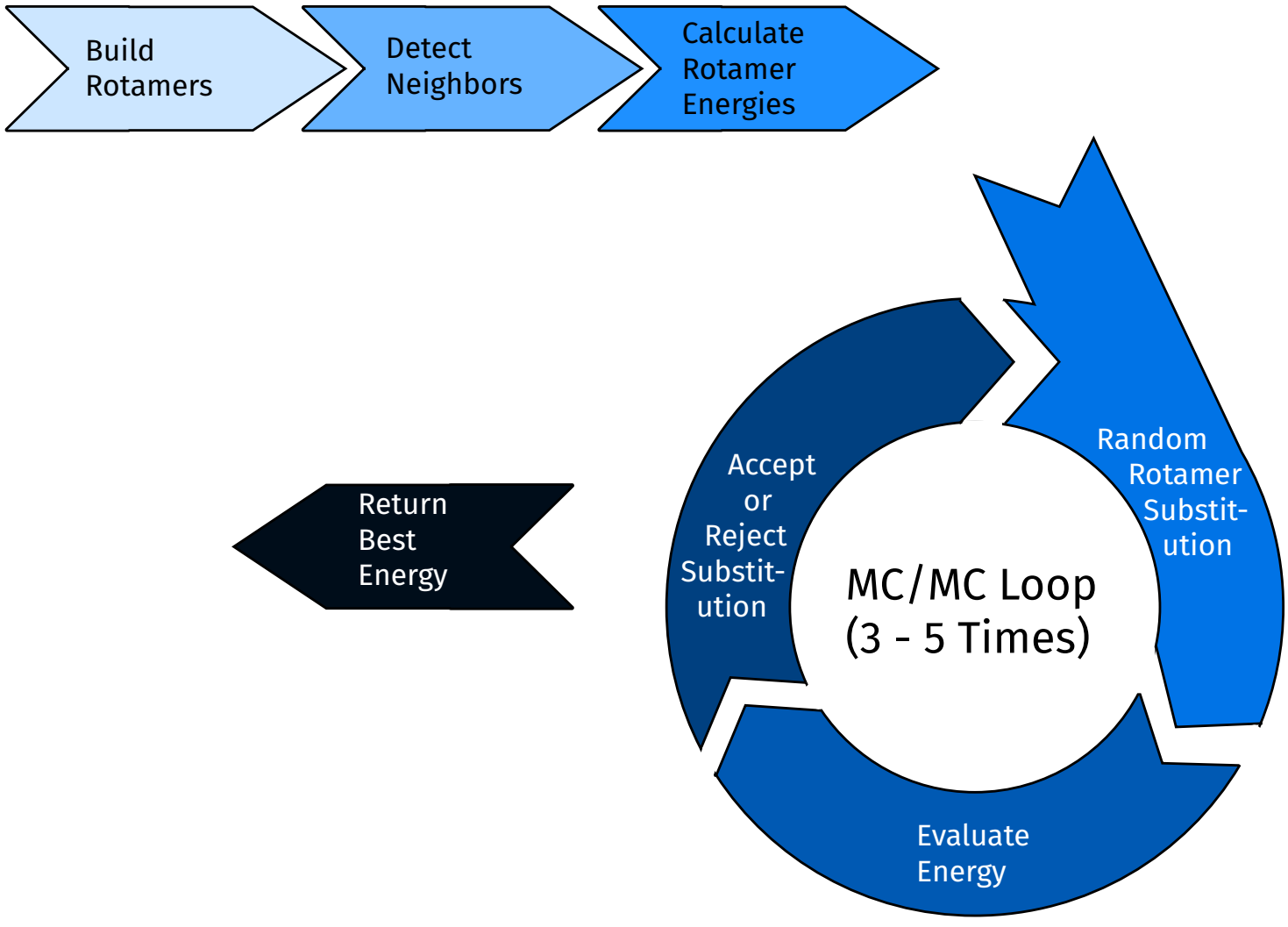


What would this resfile achieve?

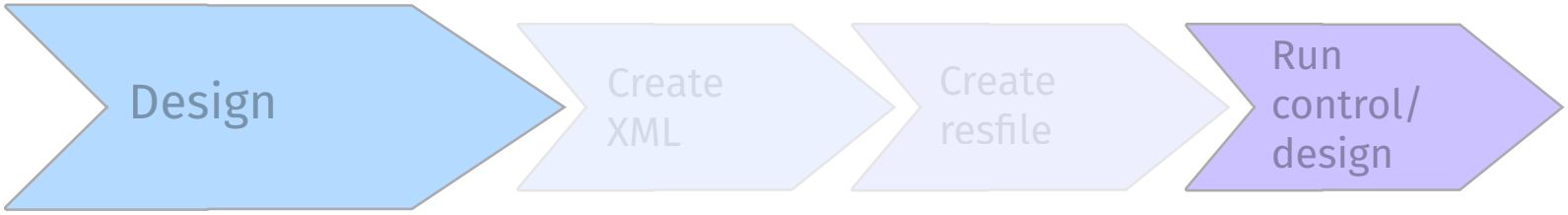
```
NATRO
EX 1 EX 2
START
30 A POLAR NOTAA K
20-35 A ALLAAxc
```

Resfiles control the Packer

The Packer



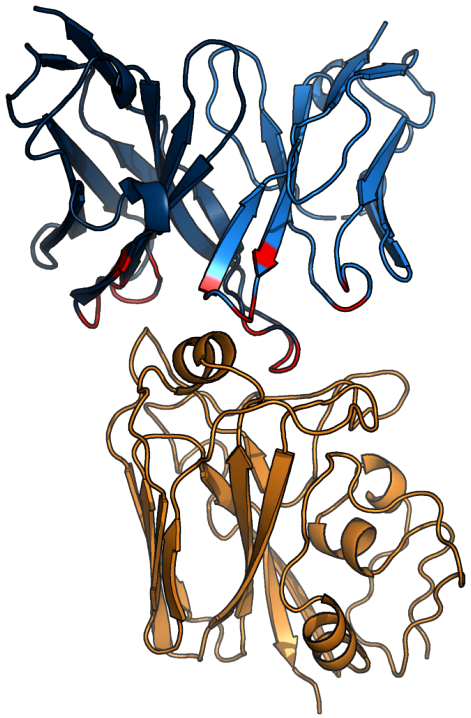
Running a Design Experiment: SSD



Single-State Design

CH67 Ab bound to H1 SolomonIslands/03/2006

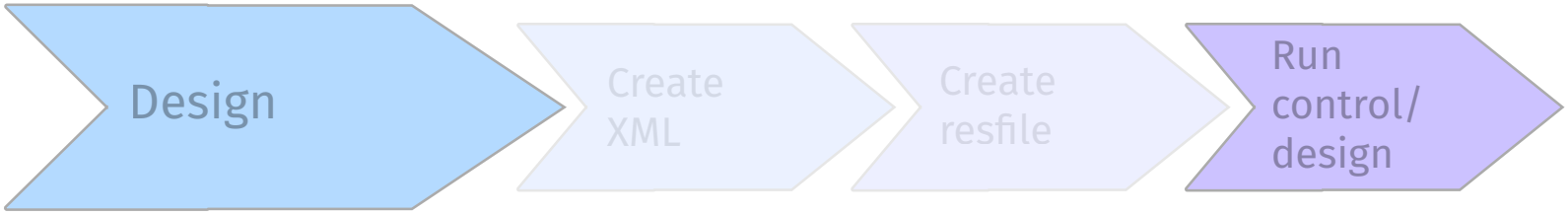
Optimize interface interactions by designing CH67 residues within 5 Å of antigen amino acids



Chains H,L
(side1)
Design interface residues

Chain A
(side2)
Allow interface repacking

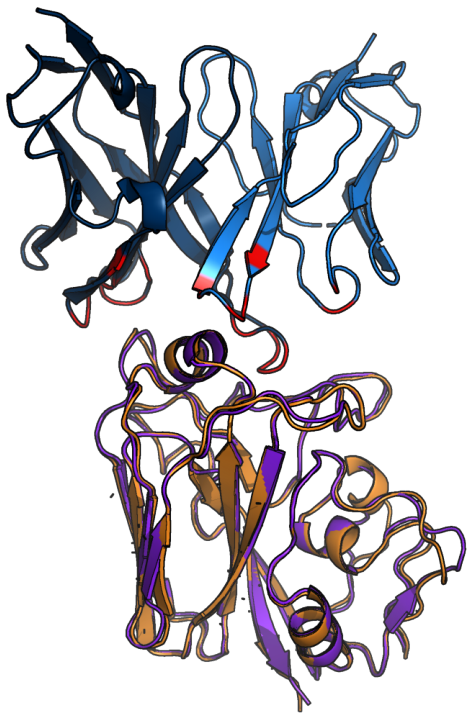
Running a Design Experiment: MSD



MultiState Design

CH67 Ab bound to H1
SolomonIslands/03/2006
or California/07/2009

Optimize interface interactions by designing CH67 residues within 5 Å of antigen amino acids



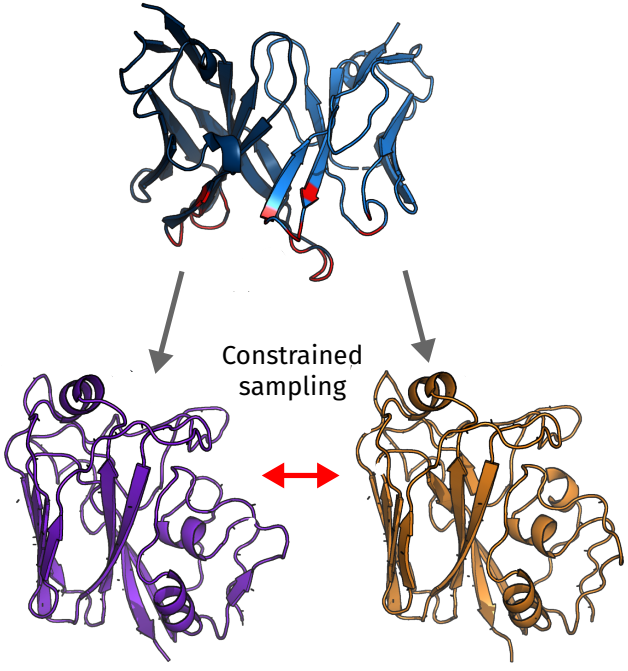
Chains H,L
(side1)
Design interface residues

Chain A/A
(side2)
Allow interface repacking

A closer look at MSD

MSD

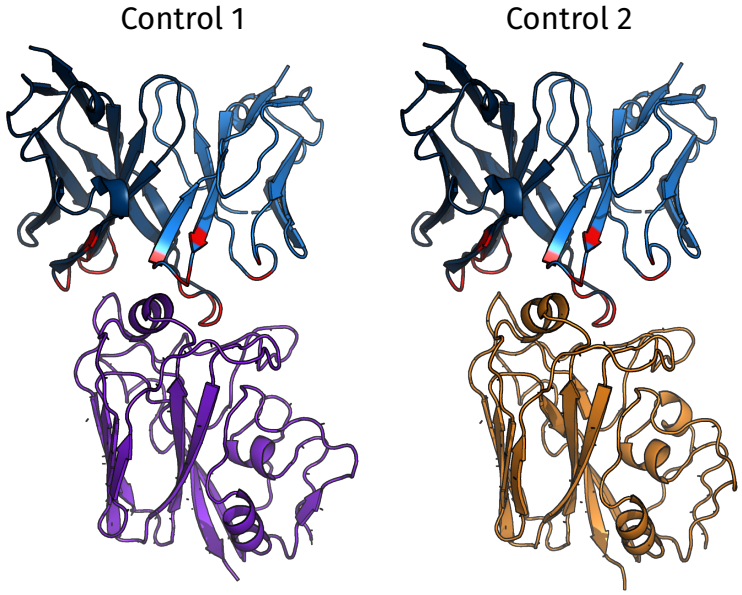
Design each interaction simultaneously



Identifies lowest-energy sequence suitable for both interfaces

MSD Control

Repack interface independently



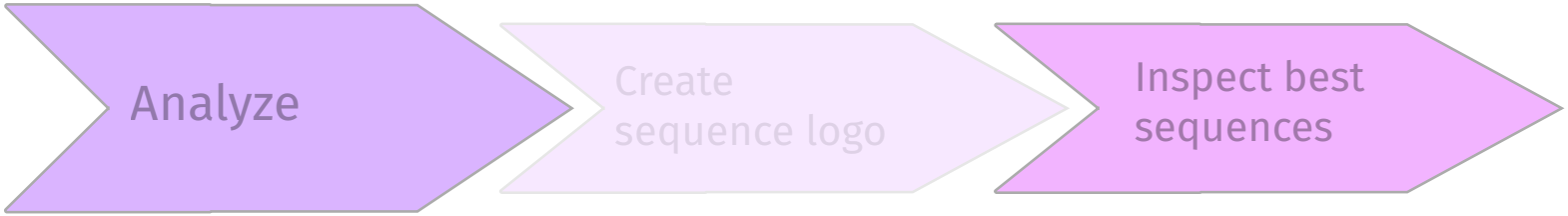
Identifies lowest-energy conformation possible of the native sequence for each pose

Sequence Logos

Useful to quickly identify sequence profiles — Shannon entropy bit score represents the log probability of a residue population at a site

http://weblogo.berkeley.edu/

Lowest Energy Sequence



Analysis Metric	Score Term	Definition
Total Score	total_score, complex_normalized	Score of the entire complex
Interface Score	side1/2_score, side1/2_normalized	Score of the residues defined as the interface
Binding Energy	ddG - dG_separated	Difference in energy between the bound and unbound partners
Binding Density	$\frac{\text{dG\_separated}}{\text{dSASA} \times 100}$	Prevents a low binding energy score by increasing the buried surface area

Analysis Mover


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

packstat	Activates packstat calculation; slow so by default is "0".
pack_input	Pre-pack before separating chains when calculating binding energy? Useful if there are non-Rosetta inputs.
pack_separated	Repack the exposed interfaces when calculating the binding energy? Usually a good idea.
fixedchains	Comma-delimited list of chain IDs to define one side of the interface

Protein Design Tutorial

Please begin the Rosetta protein design tutorial found at
`~/rosetta_workshop/tutorials/protein_design/`

If you have future questions, email
marion.f.sauer@vanderbilt.edu