RosettaRemodel

secondary structure ⟶ model

blueprint:

intuitive

direct correspondence to target

specifies fold trees

simple

versatile

minimalist

Blueprint

getBlueprintFromCoords.pl -pdbfile [pdb] > myBluprint.bp

Different tasks in one file

best with column capable editors

base blueprint

```
1 A .
2 D .
3 H .
4 E .
5 L .
6 I .
```

```
E,L,H,D
or
A,B,E,G,O
or
1,2,3
(D-amino acid)
or
I
(insertion)
```

extend

```
1 A .
2 D H
0 x H
0 x H
3 H H
4 E .
5 L .
6 I .
```

extend 2

```
1 A H
0 x H
0 x H
0 x H
3 H H
4 E .
5 L .
6 I .
```

extend 3

```
0 x H
0 x H
0 x H
1 A H
2 D .
3 H .
4 E .
5 L .
6 I .
```

extend 4

```
1 A .
2 D .
3 H .
4 E .
5 L .
6 I H
0 x H
0 x H
```

delete

```
1 A .
2 D H
4 E H
5 L .
6 I .
```

insertion
(-insert_segment_from_pdb)

rebuild

```
1 A .
2 D H
3 H H
4 E H
5 L .
6 I .
```

de novo

```
1 A H
0 x H
0 x H
0 x H
0 x H
0 x H
```

```
1 A H
0 x H
0 x I
0 x I
3 H L
4 E L
5 L .
6 I .
```

abego

```
1 A .
2 D A
3 H A
4 E A
5 L .
6 I .
```

mix

```
1 A .
2 D HA
3 H HA
4 E HA
5 L .
6 I .
```

The middle column is not used for anything, unless you want to pick fragments with sequence bias (-use_blueprint_sequence). Poor man's fold prediction.

Blueprint -- what you see is what you get

extend 3

```
0 x H
0 x H
0 x H
1 A H
2 D .
3 H .
4 E .
5 L .
6 I .
```

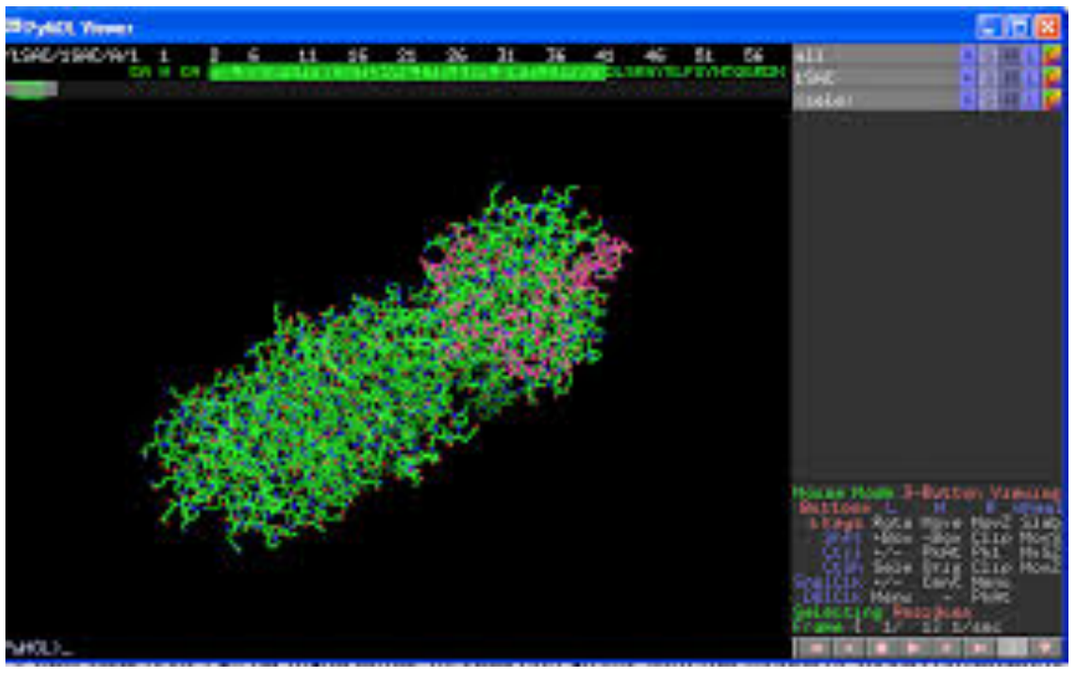
vi

:set number

extend 3

```
1 0 x H
2 0 x H
3 0 x H
4 1 A H
5 2 D .
6 3 H .
7 4 E .
8 5 L .
9 6 I .
```

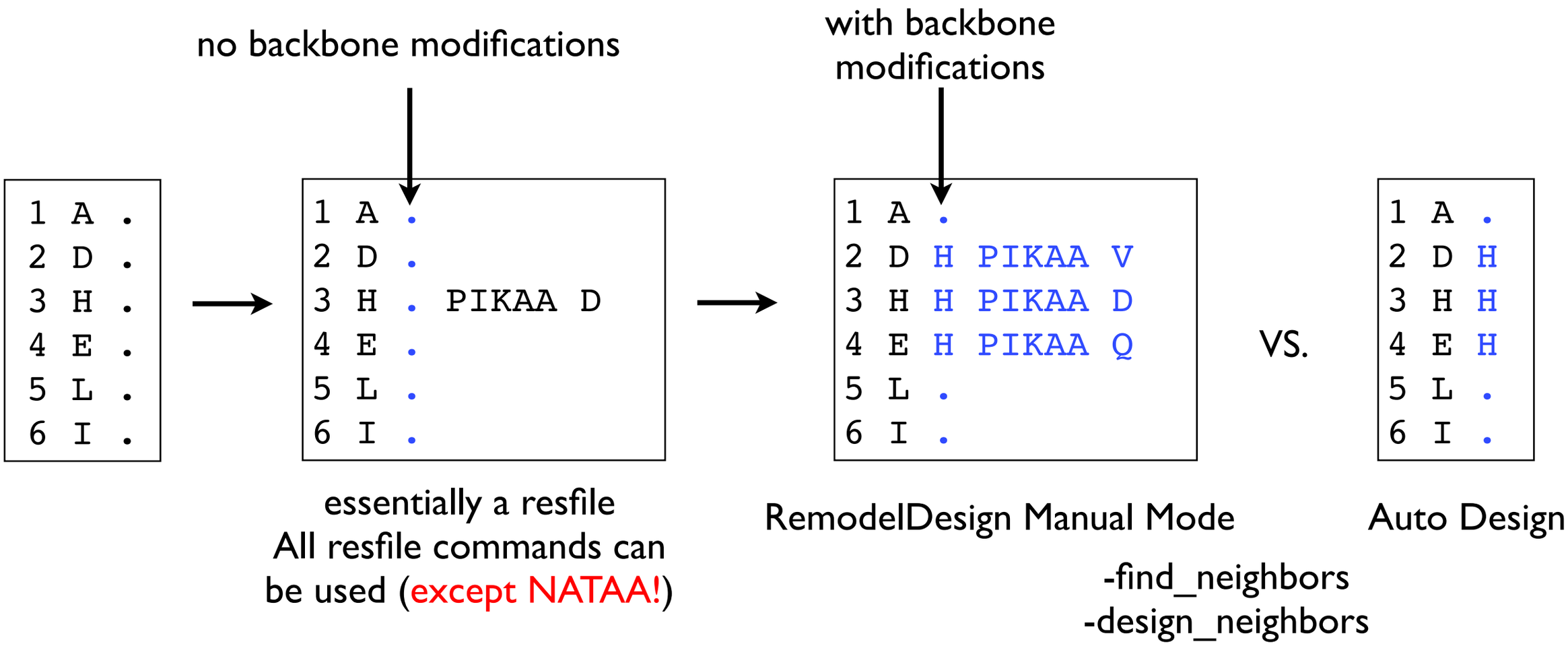
never count residues (if native seq is attached using the getBlueprintFromCoords script)



```
1 A .
2 D .
3 H .
4 E .
5 L .
6 I .
7 A .
8 V .
9 K .
```

Blueprint -- not just backbone controls

1. Sidechain



essentially a resfile
All resfile commands can be used (except NATAA!)

RemodelDesign Manual Mode

-find_neighbors
-design_neighbors

Auto Design

Blueprint -- not just backbone controls

2. Constraint assignment (ENZDES style)

```
1 A .
2 D .
3 H .
4 E .
5 L .
6 I .
```

→

```
1 A .
2 D H PIKAA A CST1A
3 H H ALLAA
4 E H ALLAA
5 L L PIKAA H CST1B
6 I L ALLAA
0 x L PIKAA A CST2A
0 x L PIKAA H CST2B
```

1

```
CST::BEGIN
TEMPLATE:: ATOM_MAP: 1 atom_name: Nbb
TEMPLATE:: ATOM_MAP: is_backbone
TEMPLATE:: ATOM_MAP: 1 residue3: A

TEMPLATE:: ATOM_MAP: 2 atom_type: Nhis,
TEMPLATE:: ATOM_MAP: 2 residue1: H

CONSTRAINT:: distanceAB: 2.00 0.30 100.00 1
CONSTRAINT:: angle_A: 105.10 6.00 100.00 360.00
CONSTRAINT:: angle_B: 116.90 5.00 50.00 360.00
CONSTRAINT:: torsion_A: 105.00 10.00 50.00 360.00
CONSTRAINT:: torsion_B: 180.00 10.00 25.00 180.00
CONSTRAINT:: torsion_AB: 0.00 0.00 0.00 180.00
CST::END
```

A

B

2

```
CST::BEGIN
TEMPLATE:: ATOM_MAP: 1 atom_name: Nbb
TEMPLATE:: ATOM_MAP: is_backbone
TEMPLATE:: ATOM_MAP: 1 residue3: A

TEMPLATE:: ATOM_MAP: 2 atom_type: Nhis,
TEMPLATE:: ATOM_MAP: 2 residue1: H

CONSTRAINT:: distanceAB: 2.00 0.30 100.00 1
CST::END
```

A

B

Blueprint -- not just backbone controls

3. Disulfide control (with -build_disulf)

```
1 A .
2 D .
3 H .
4 E .
5 L .
6 I .
```



```
1 A .
2 D . DS_start DS_stop
3 H H ALLAA
4 E H ALLAA
5 L L PIKAA H CST1B
6 I L ALLAA DM_start
0 x L PIKAA A CST2A
0 x L PIKAA H CST2B DM_stop
```

Blueprint -- not just backbone controls

4. NCAA control

```
1 A .
2 D .
3 H .
4 E .
5 L .
6 I .
```

→

```
1 A . EMPTY NC BPY
2 D .
3 H 1 EMPTY NC DALA
4 E 1 EMPTY NC DGLY NC DMET NC DPRO
5 L 1 EMPTY NC DASP
6 I L ALLAA
0 x L PIKAA A
0 x L PIKAA H
```

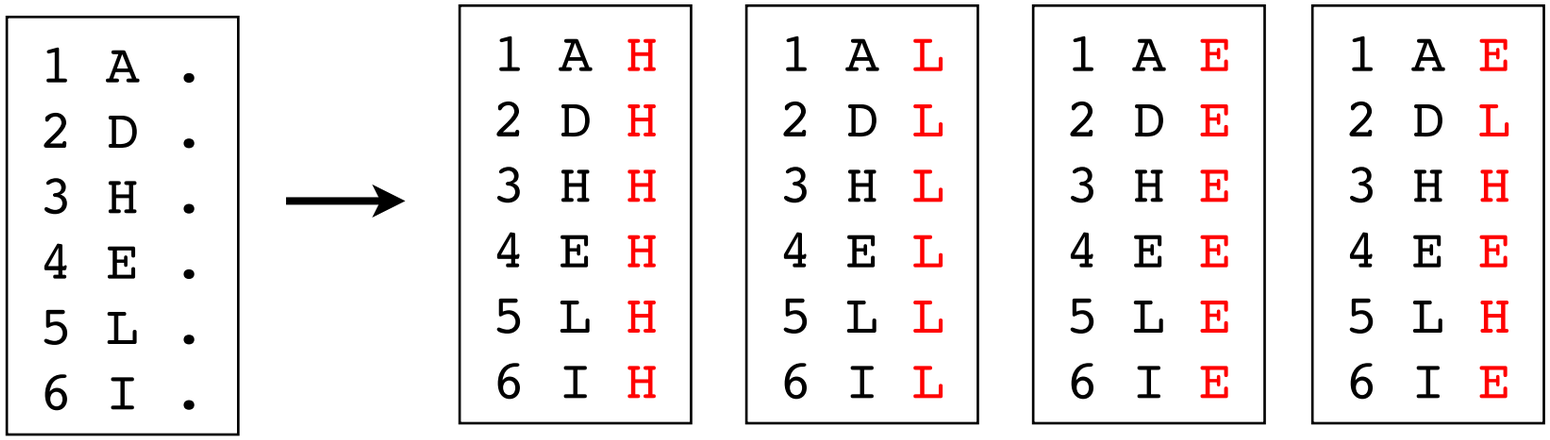
← -extra_res_fa [params file]
-extra_res_cen [centorid params file]

Blueprint -- not just backbone controls

5. relax mode (marking movable)

-bypass_fragments

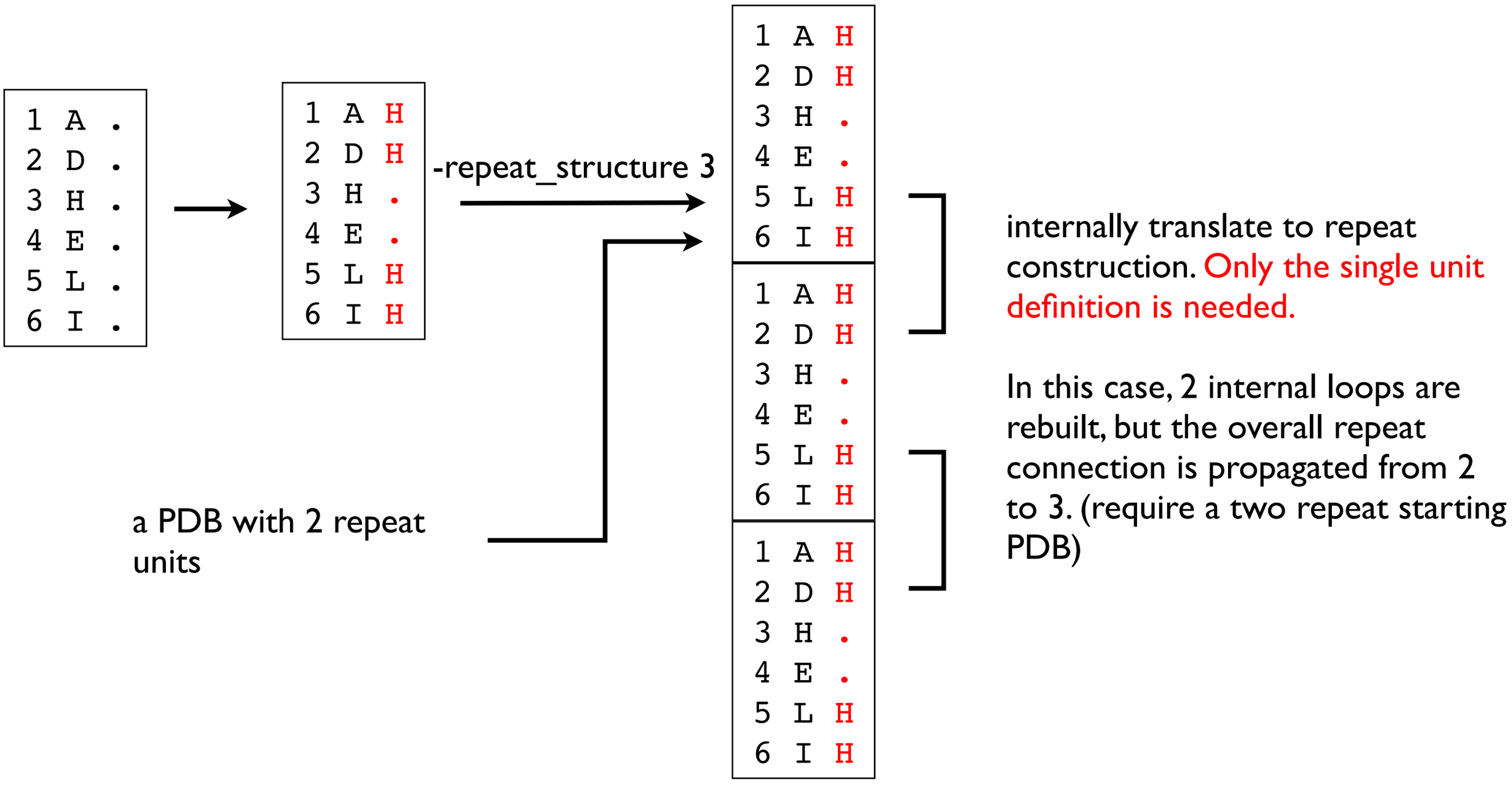
-use_pose_relax or -use_cart_relax



basically what is now called fast design, I think

Blueprint -- not just backbone controls

6. repeat mode
with -repeat_structure 3



Blueprint -- not just backbone controls

7. de novo build

```
1 A H
0 x H
0 x H
0 x H
0 x H
```

← de novo build always require the first residue and a dummy pdb with some junk residues

-repeat_structure 4

it'll be treated as if specifying the blueprint on the right, and build 4 identical repeating units. Only the single unit definition is needed.

```
1 A H
0 x H
0 x H
0 x H
0 x H
---
1 A H
0 x H
0 x H
0 x H
0 x H
---
1 A H
0 x H
0 x H
0 x H
0 x H
---
1 A H
0 x H
0 x H
0 x H
0 x H
```

How to run RosettaRemodel

For very basic features, all you really need is this:

remodel.static.linuxgccrelease -s [pdb name] -remodel:blueprint [blueprint name]

But commonly used with some of these:

run control

-num_trajectory 1 (default is 10)
-save_top 1 (default is 5)
-remodel:quick_and_dirty
-out:prefix [name] (otherwise 1.pdb, 2.pdb, 3.pdb....etc.)
-jd2:no_output (so it doesn't write out the useless XXX_0001.pdb)

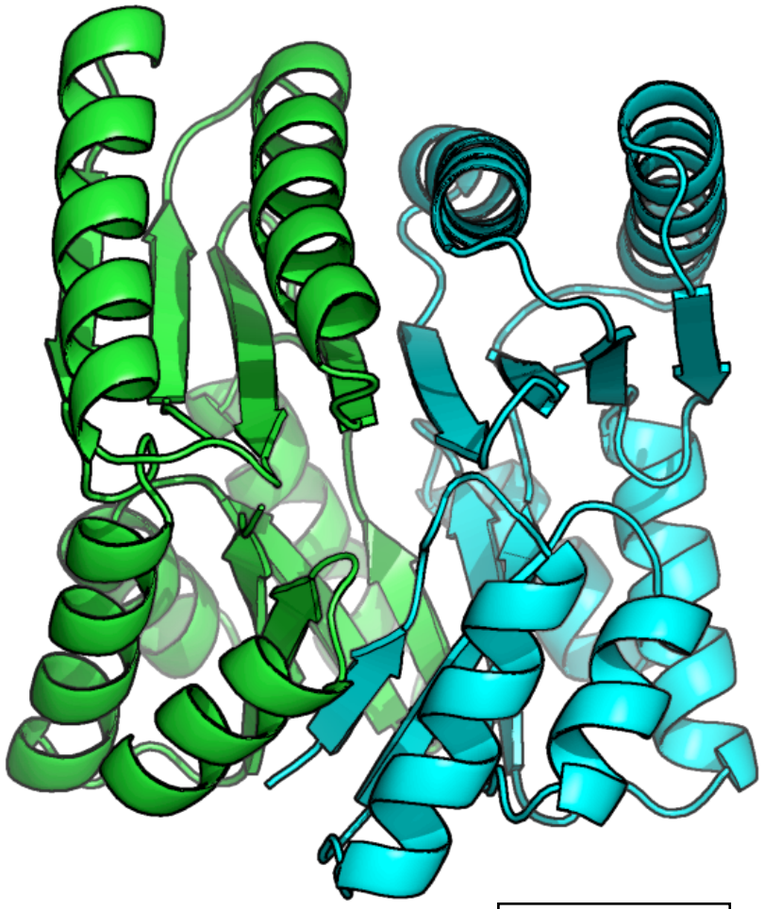
reweight centroid terms:

-vdw 1.0
-hb_srbb 1.0
-hb_lrbb 1.0
-rsigma 1.0
-sspair 1.0

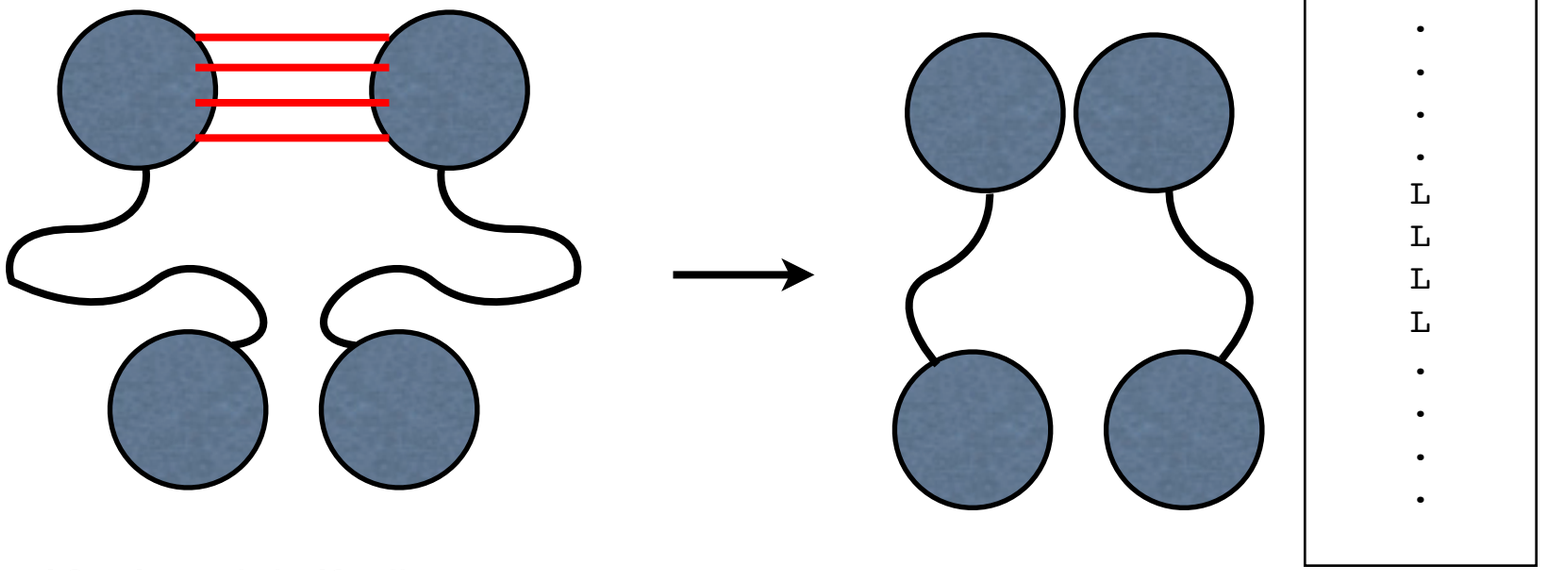
fullatom weight

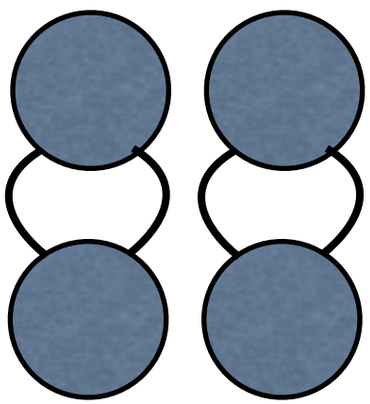
-soft_rep_design
-beta

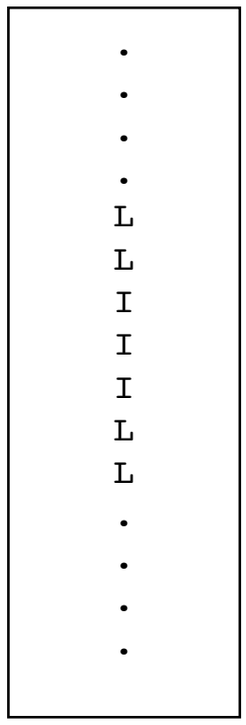
notes on working with symmetry



-symmetry_definition C2_Z.sym







-s tim10_aligned_half.pdb
-remodel:blueprint 2_74_2.bp
-insert_segment_from_pdb 4pww_A.pdb
-remodel:design:find_neighbors
-remodel:design:design_neighbors
-remodel:use_cart_relax
-num_trajectory 1
-chain A
-overwrite
-jd2:no_output
-symmetry_definition C2_Z.sym
-cst_file 2_74_2.cst
-cstfilter 500
-max_linear_chainbreak 3

anchor residue
foldtree rooting

PDB

blueprint

-s

-remodel:blueprint

on the fly fragment picking (full, 9, 3, 1mers)

-no_jumps

fragment/CCD

-bypass_closure

-max_linear_chainbreak

closure?

-bypass_fragments

-remodel:design:no_design

Design (6 different modes by blueprint and flags)

-core_cutoff

-boundary_cutoff

-find_neighbors

-design_neighbors

-remodel:dr_cycles [num]

-remodel:quick_and_dirty

-use_cart_relax

-use_pose_relax

CCD refine

Cart Refine

-free_relax

Pose relax Refine

-free_relax

accumulator

-save_top [num]

final PDB

PDB

blueprint

cst

enzdes_cst

sym_def

domain insert

-s

-remodel:blueprint

-cstfile

-enzdes:cst_file

-symmetry_definition

-insert_segment_from_pdb

on the fly fragment picking (full, 9, 3, 1mers)

segment file

-use_loophash -lh:db_path -lh:max_ex_limit -lh:max_radius -lh_filter

-set_segment [segment file]

loophash

set_segment

fragment/CCD

-staged_sampling (abinito)

-randomize_loops

-randomize_loops

-no_jumps

-simultaneous_cycles

-independent_cycles

-boost_closure_cycles

-repeat_structure [num]

repeat mode

-helical_rise

-helical_radius

-helical_omega

-cyclic_peptide

-bypass_closure

-max_linear_chainbreak

closure?

-bypass_fragments

-build_disulf -match_rt_limit

disulfide

-cen_minimize

centroid minimize (use with caution!)

accumulator

-save_top [num]

-remodel:design:no_design

Design (6 different modes by blueprint and flags)

-core_cutoff

-boundary_cutoff

-find_neighbors

-design_neighbors

-remodel:dr_cycles [num]

-remodel:quick_and_dirty

-use_cart_relax

-use_pose_relax

CCD refine

Cart Refine

-free_relax

Pose relax Refine

-free_relax

filter by cst

-cstfilter [value]

-run_confirmation

conformation KIC/CCD

-swap_refine_confirm_protocols

cluster (don't normally use)

-use_clusters

repeat filter

-filter_rise -filter_radius -filter_omega

final PDB

-preserve_header