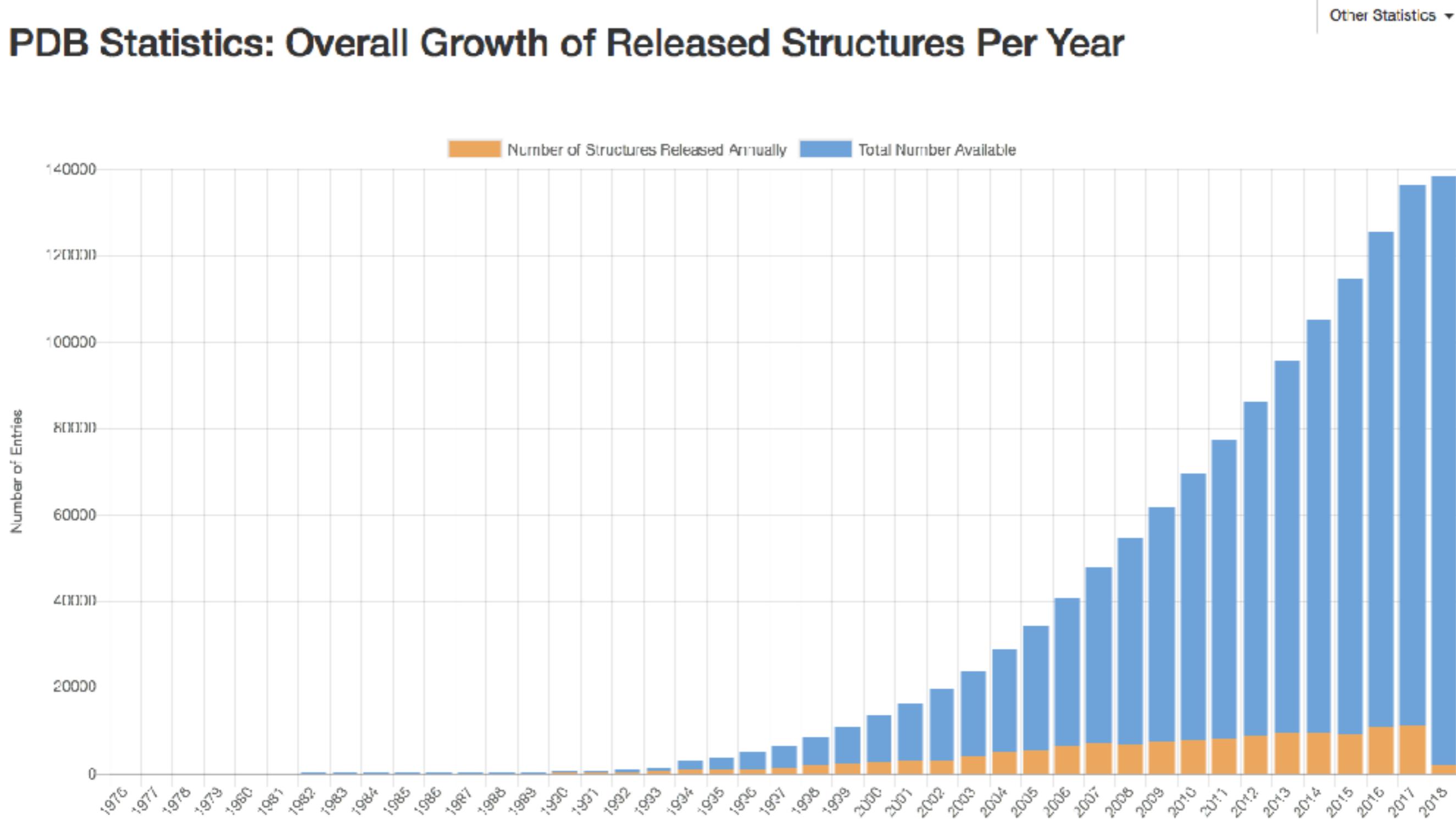


Structure determination has proceeded at a remarkable pace

PDB Statistics: Overall Growth of Released Structures Per Year



The Protein Data Bank (PDB) is a repository which stores structures of macromolecules

RCSB PDB Deposit Search Visualize Analyze Download Learn More MyPDB

RCSB PDB PROTEIN DATA BANK 138464 Macromolecular Structures Enabling Breakthroughs In Research and Education

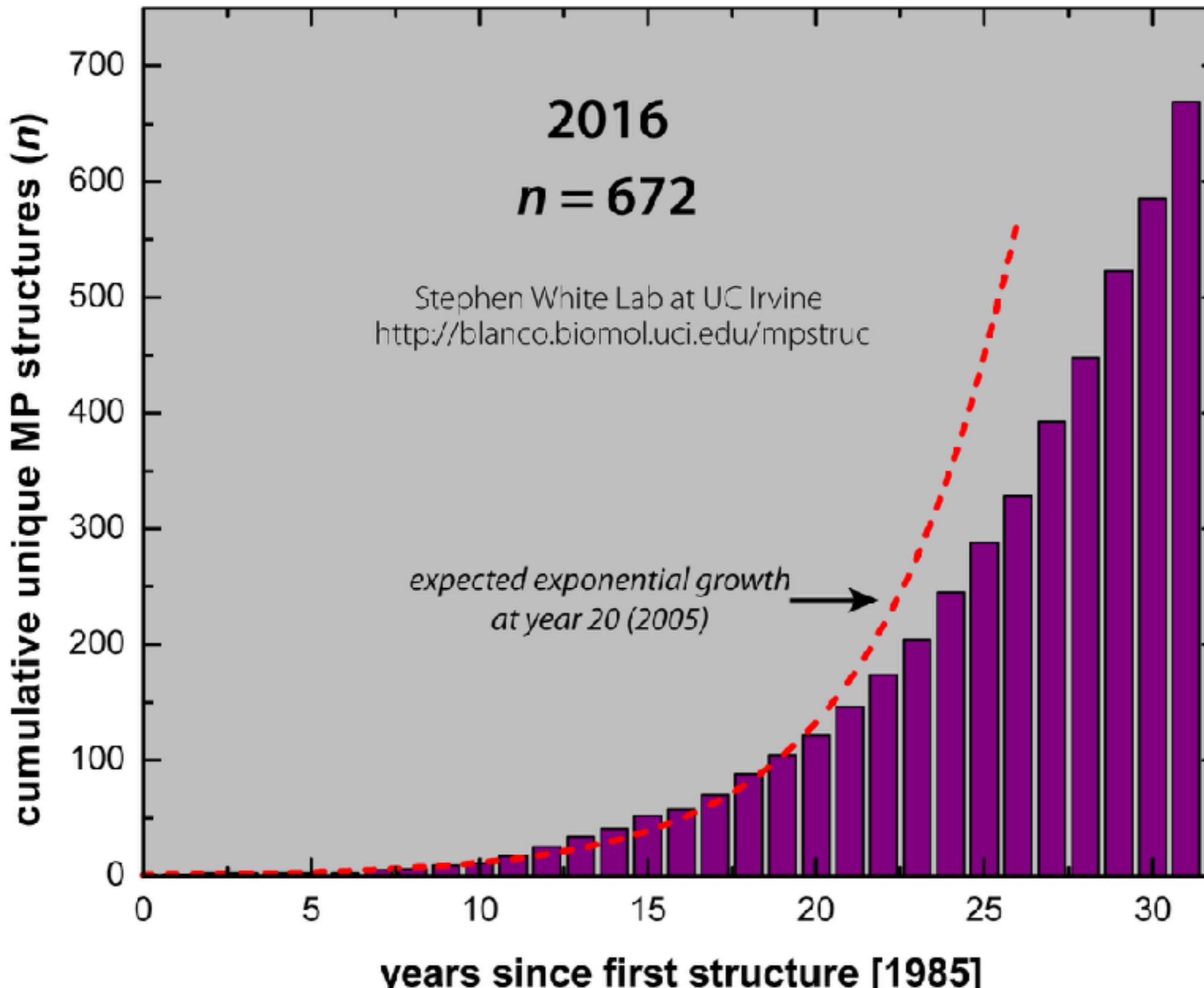
Search by PDB ID, author, macromolecule, sequence, or ligands Go Advanced Search | Browse by Annotations

PDB-101 Worldwide PDB EMDataBank NDB Nucleic Acid Database Worldwide Protein Data Bank Foundation

www.rcsb.org

www.rcsb.org

Membrane protein structures are still extremely rare



Most structures are obtained via crystallography

PDB Data Distribution by Experimental Method and Molecular Type

Other Statistics ▾

Experimental Method	Proteins	Nucleic Acids	Protein/NA Complex	Other	Total
X-Ray	116115	1916	5922	10	123963
NMR	10660	1236	249	8	12153
Electron Microscopy	1459	31	506	0	1996
Other	210	4	6	13	233
Multi Method	112	4	2	1	119
Total	128556	3191	6685	32	138464

113777 structures in the PDB have a structure factor file.

9490 structures in the PDB have an NMR restraint file.

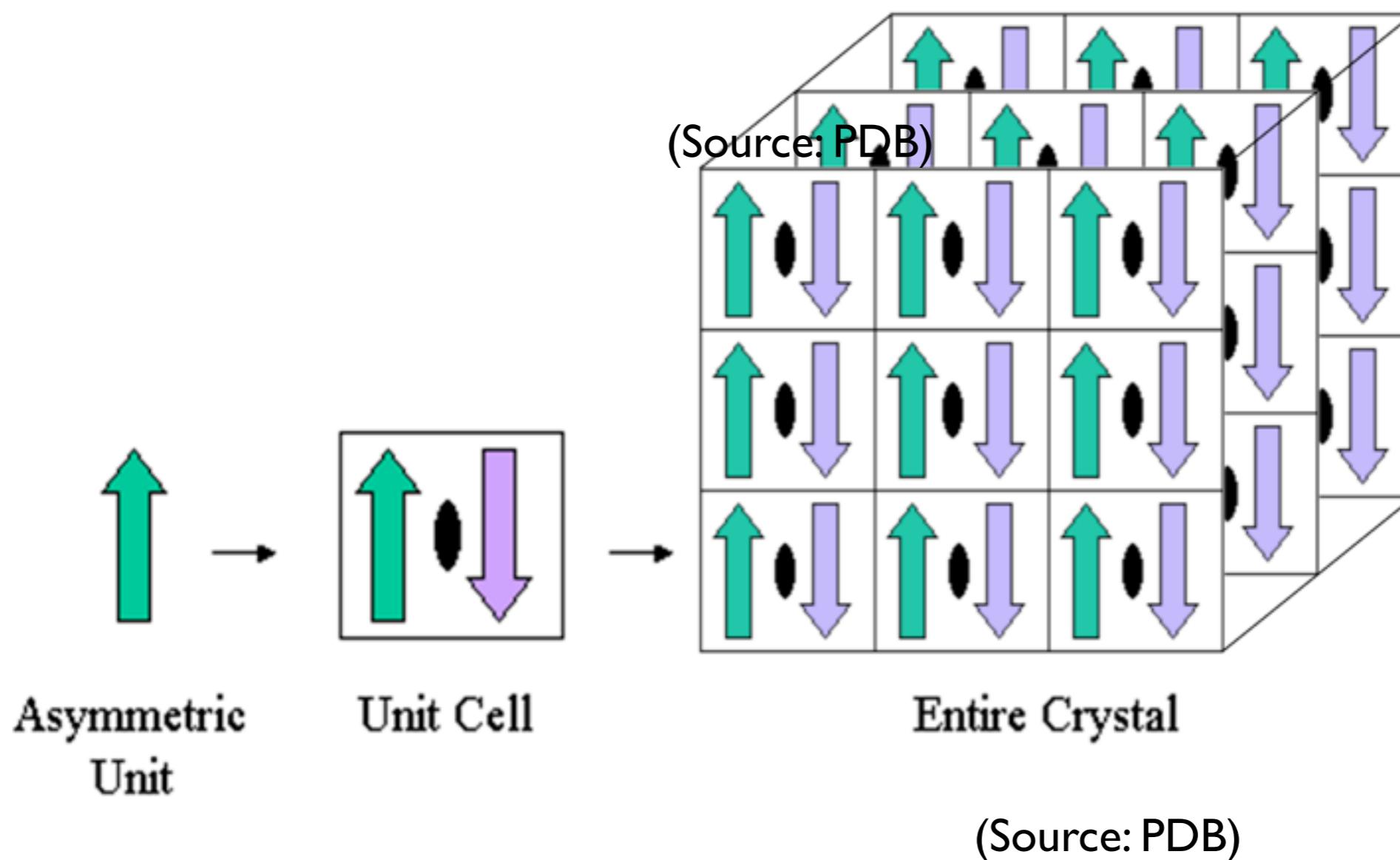
3242 structures in the PDB have a chemical shifts file.

2022 structures in the PDB have a 3DEM map file.

What's in a PDB file?

- Coordinates:
 - Atom list
 - 3D position in space
 - “occupancy”
 - Missing information!
- Experimental details
 - What the system is, how the structure was solved, secondary structure information, protein sequence
- Other info:
 - i.e. citations

PDB files for crystals contain just a portion of the whole crystal structure -- the “asymmetric unit”

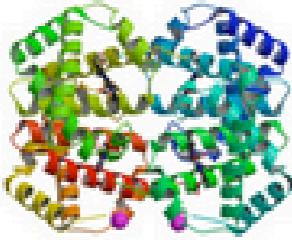


The biological assembly may or may not be the same as the asymmetric unit

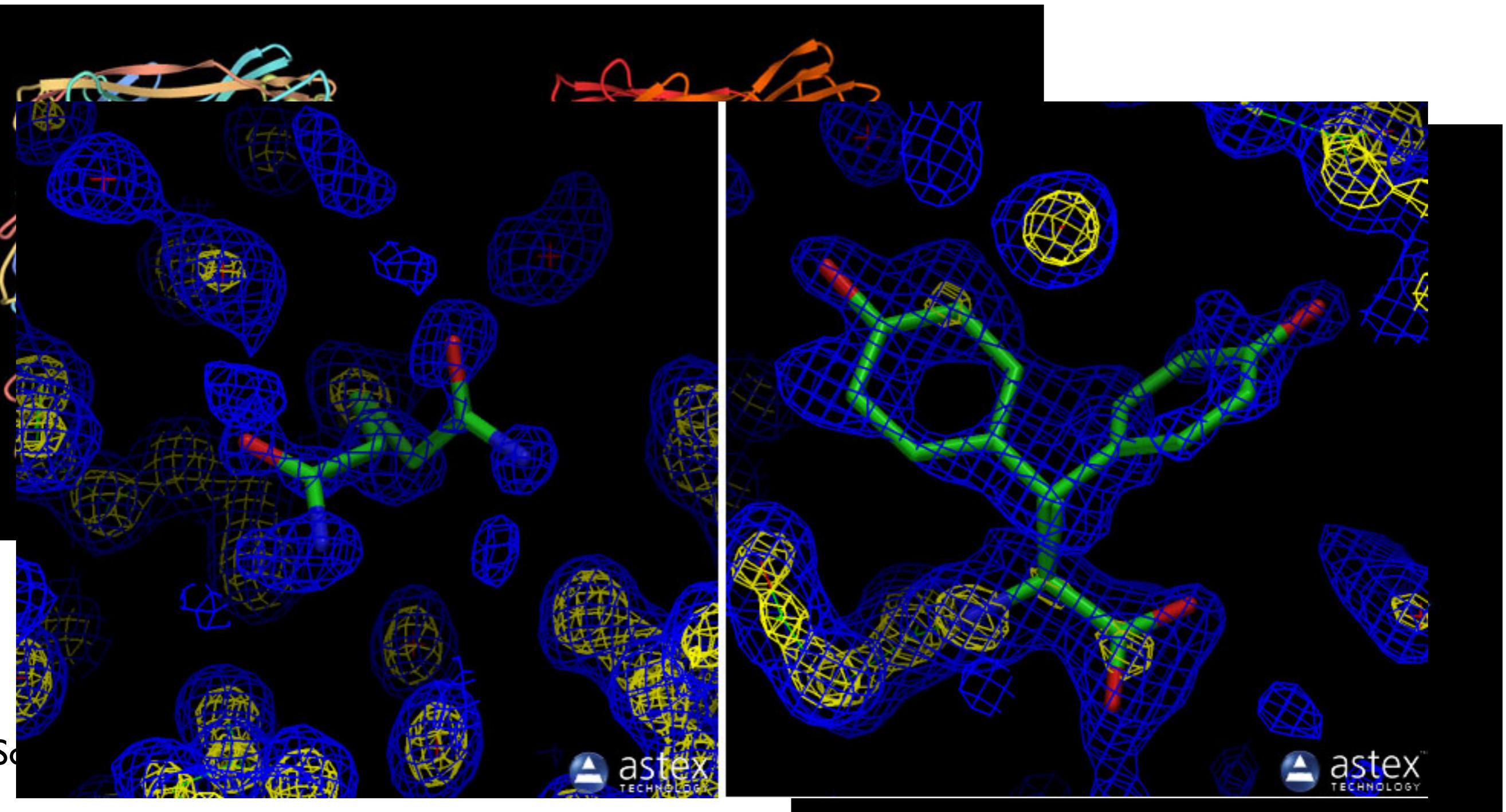
- Asymmetric unit can be:

- A single biological assembly
- Multiple biological assemblies
- Part of a biological assembly

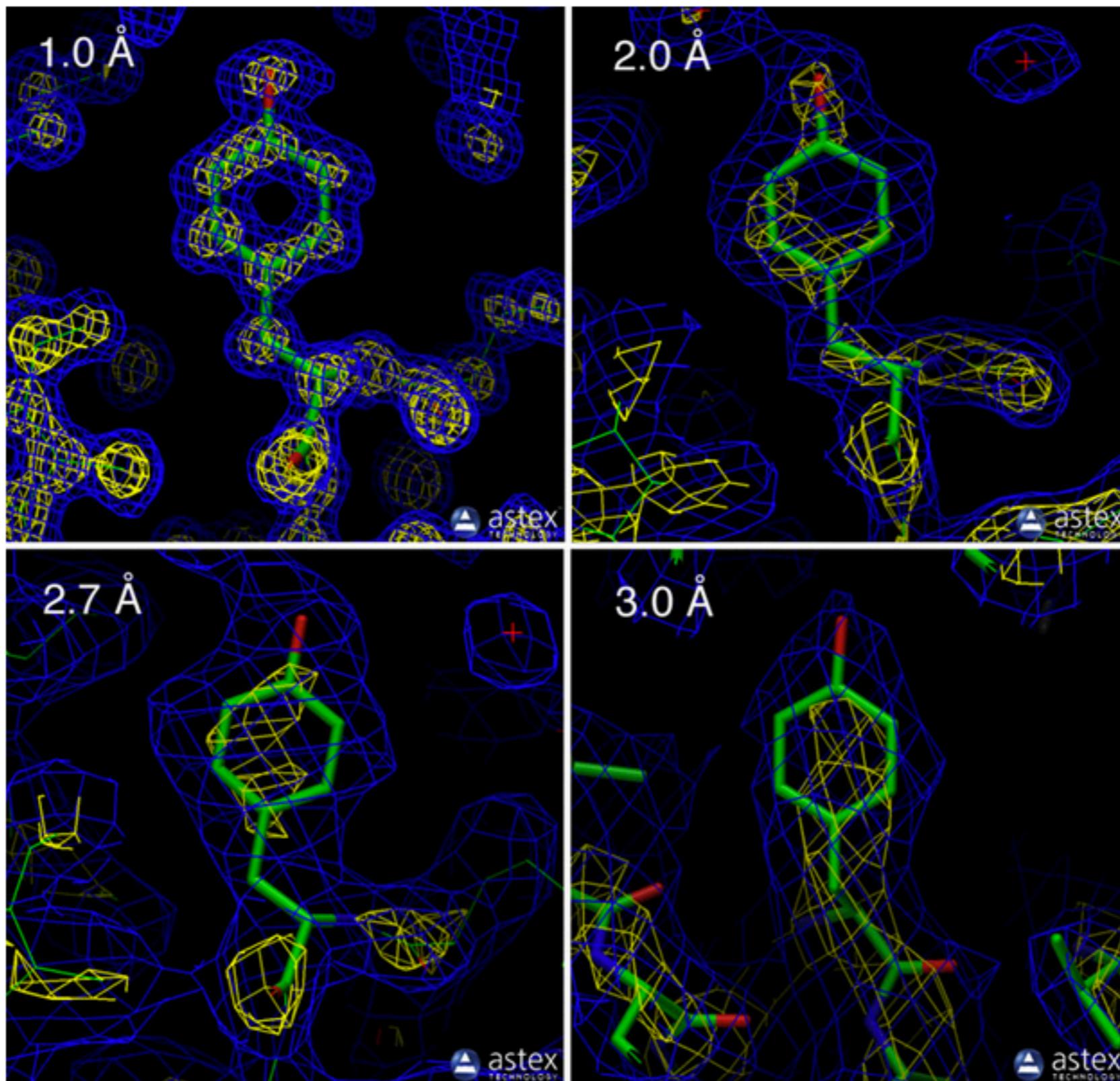
(Source: PDB)

Asymmetric unit with one biological assembly	Asymmetric unit with a portion of a biological assembly	Asymmetric unit with multiple biological assemblies
 Entry 2hhb contains one hemoglobin molecule (4 chains) in the asymmetric unit.	 Entry 1hho contains half a hemoglobin molecule (2 chains) in the asymmetric unit. A crystallographic two-fold axis generates the other 2 chains of the hemoglobin molecule.	 Entry 1hv4 contains two hemoglobin molecules (8 chains) in the asymmetric unit.

Structures may contain multiple chains and models; temperature factor (B value) information is also deposited



Resolution is an important aspect when looking at structures

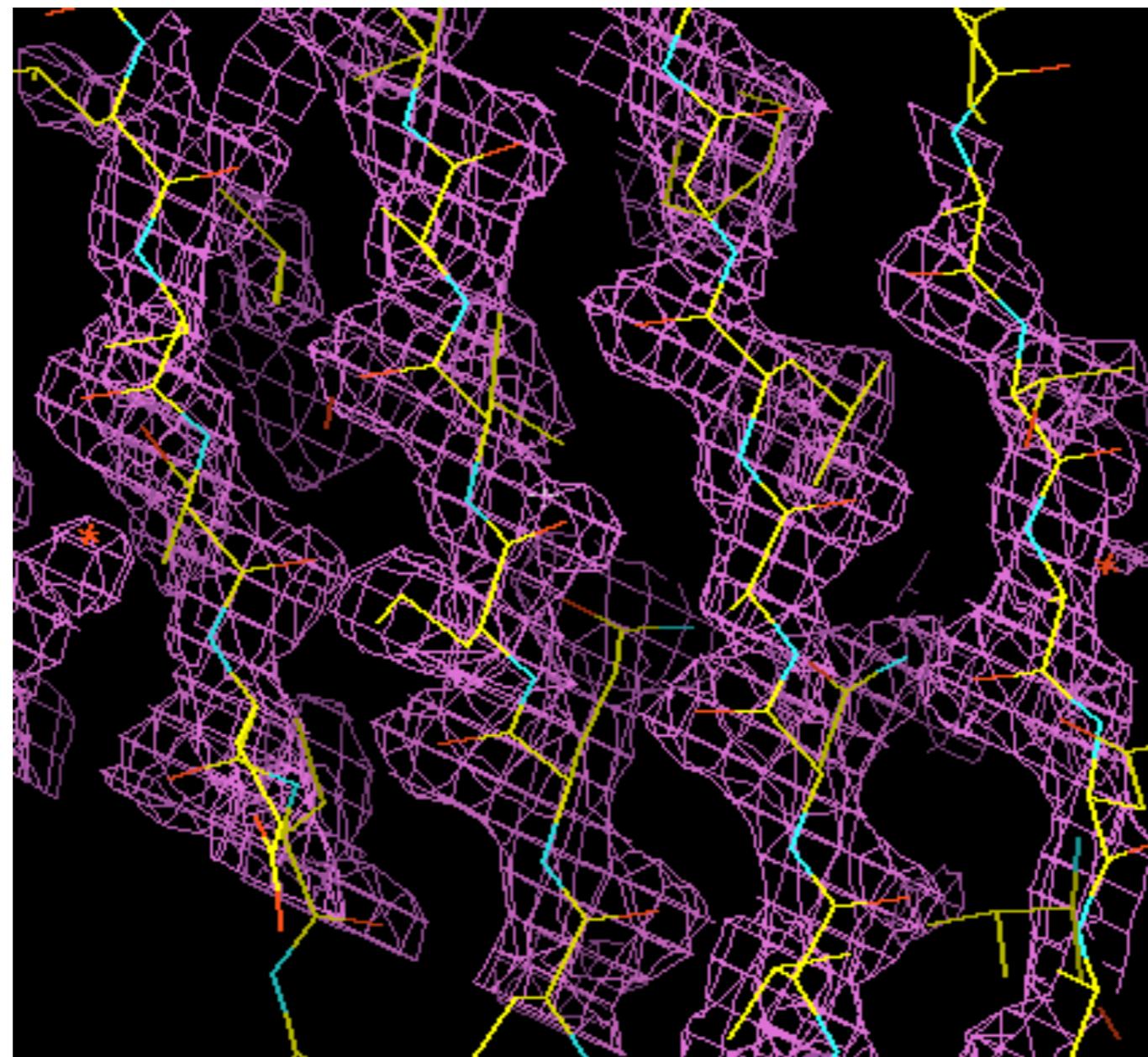


R-value and R-free are important factors in reliability

- R value: Measures how well simulated diffraction pattern matches experiment
 - Ideal: R=0
 - Typical: R~0.2
 - Random: R about 0.63
- But R value is used in building model, which introduces bias
- Better: Use R-free
 - R value on 10% of data which is held back and not used in building model. Typically slightly higher, around 0.26
 - Big concern if it is ever a lot higher than R -- suggests overfitting

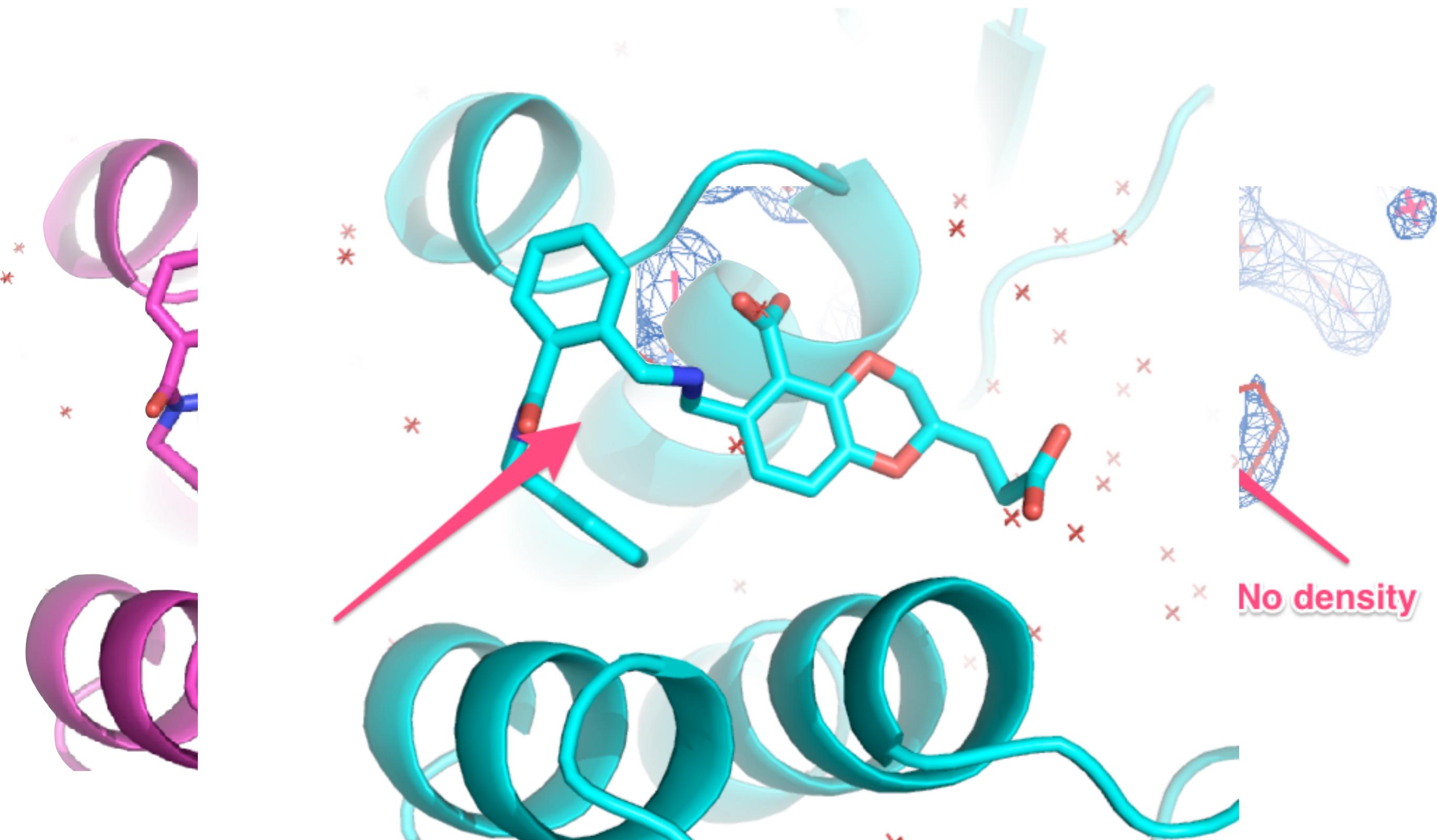
(Source: PDB)

At the very least you should visualize the electron density before using a structure for something

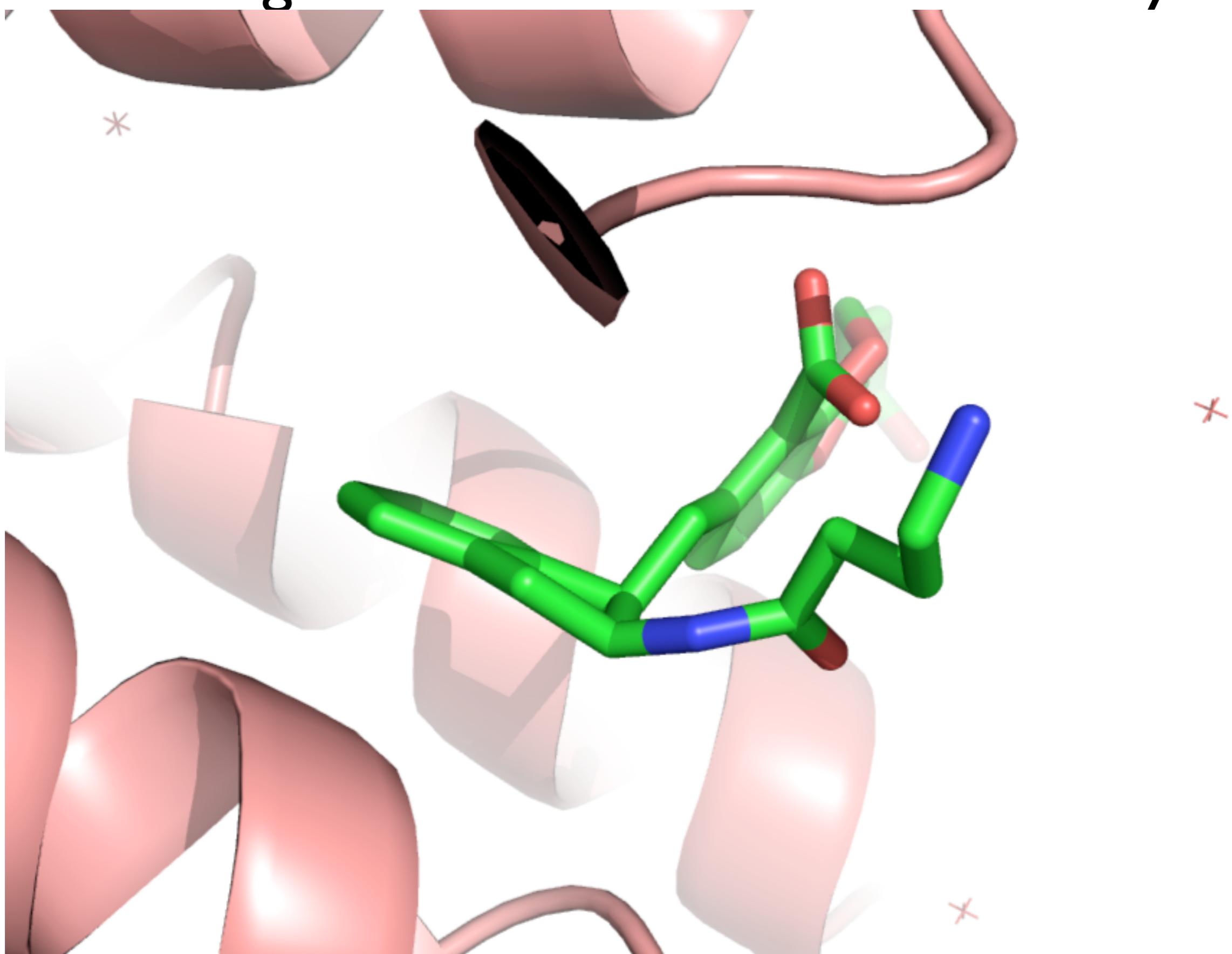


The structure
is a FIT to the
density

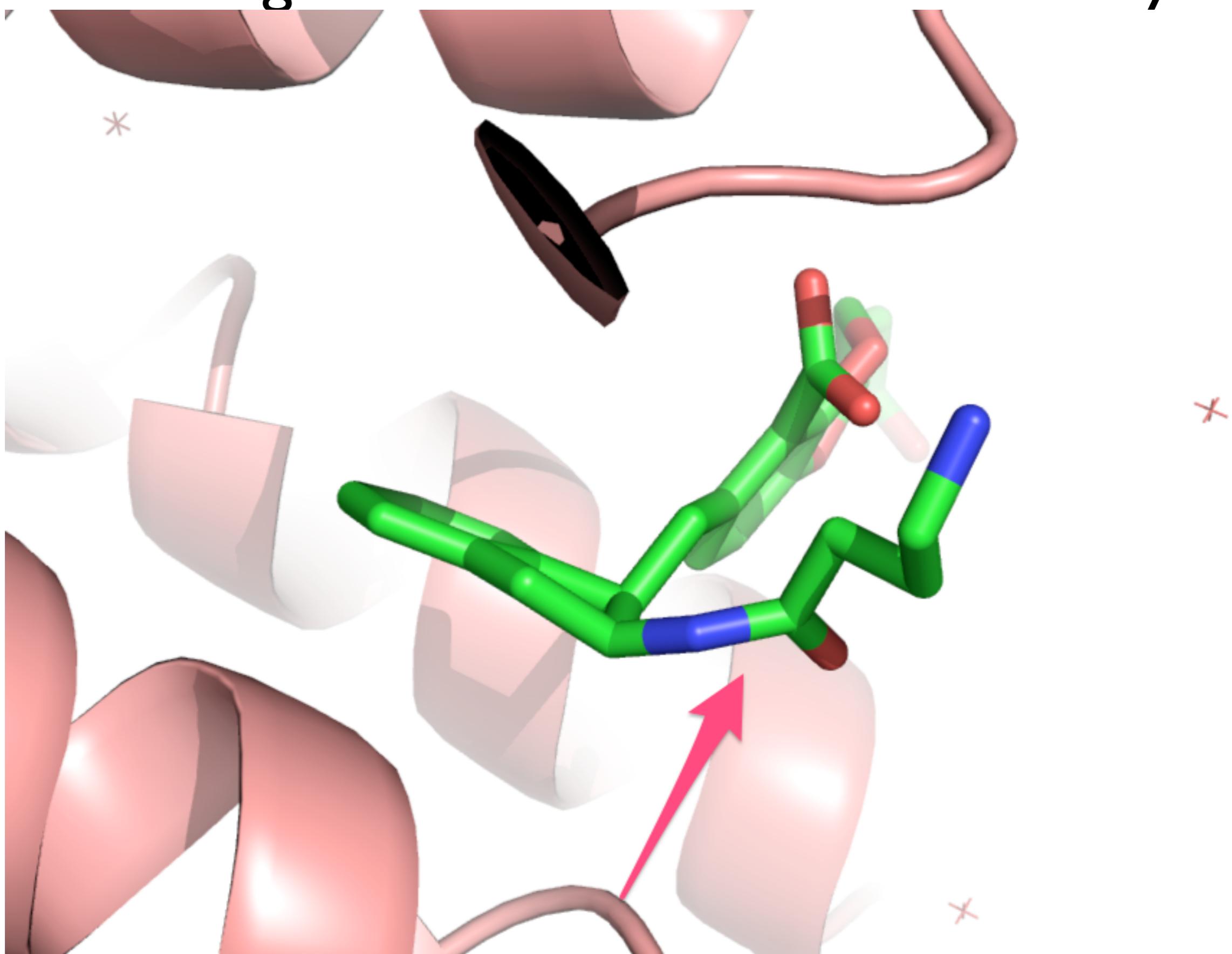
A lot of other things can get in the way: How sure are you of what you're looking at?



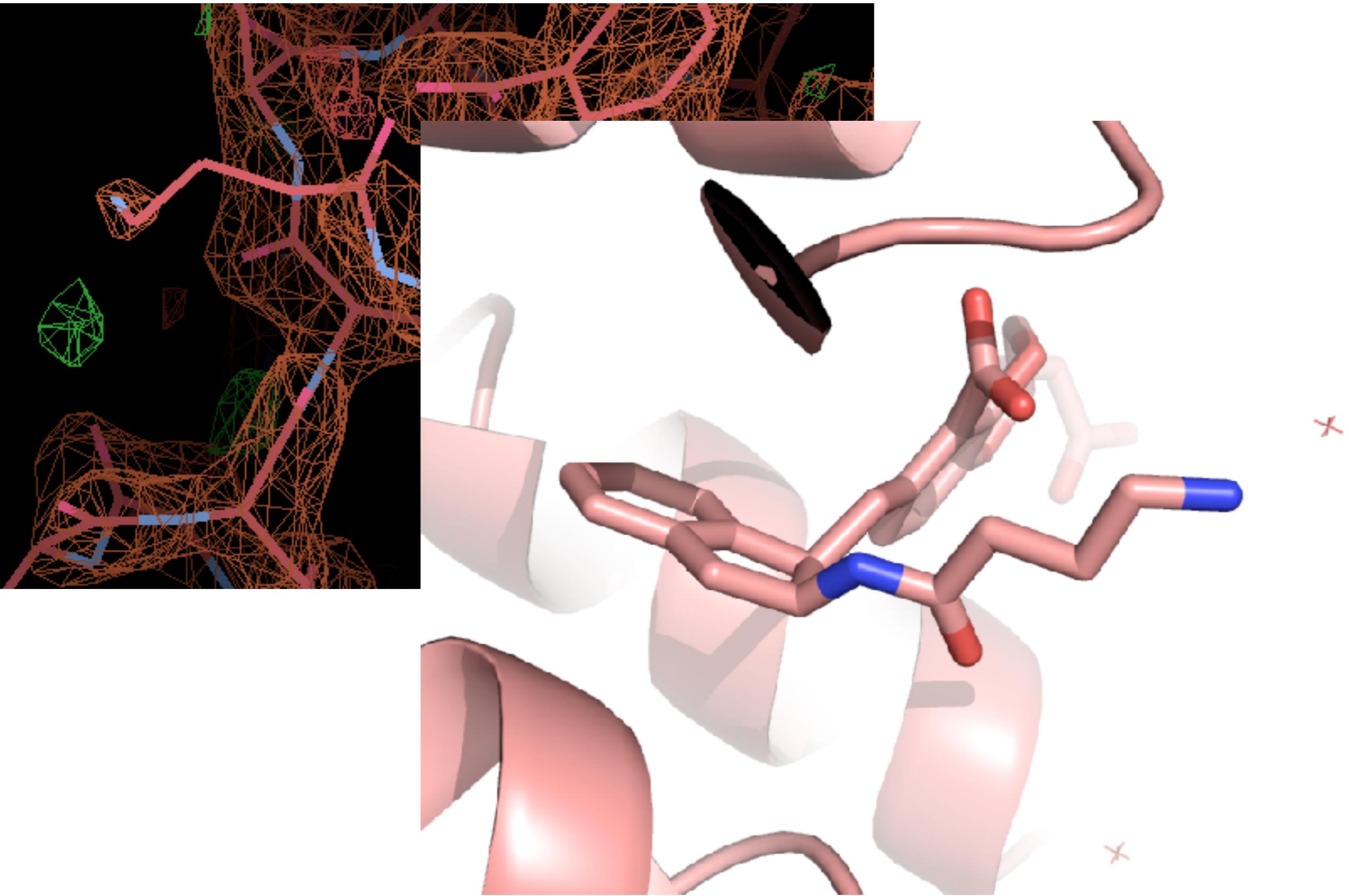
Molecular modeling is used in refinement. Do you have the right balance of forces and density?



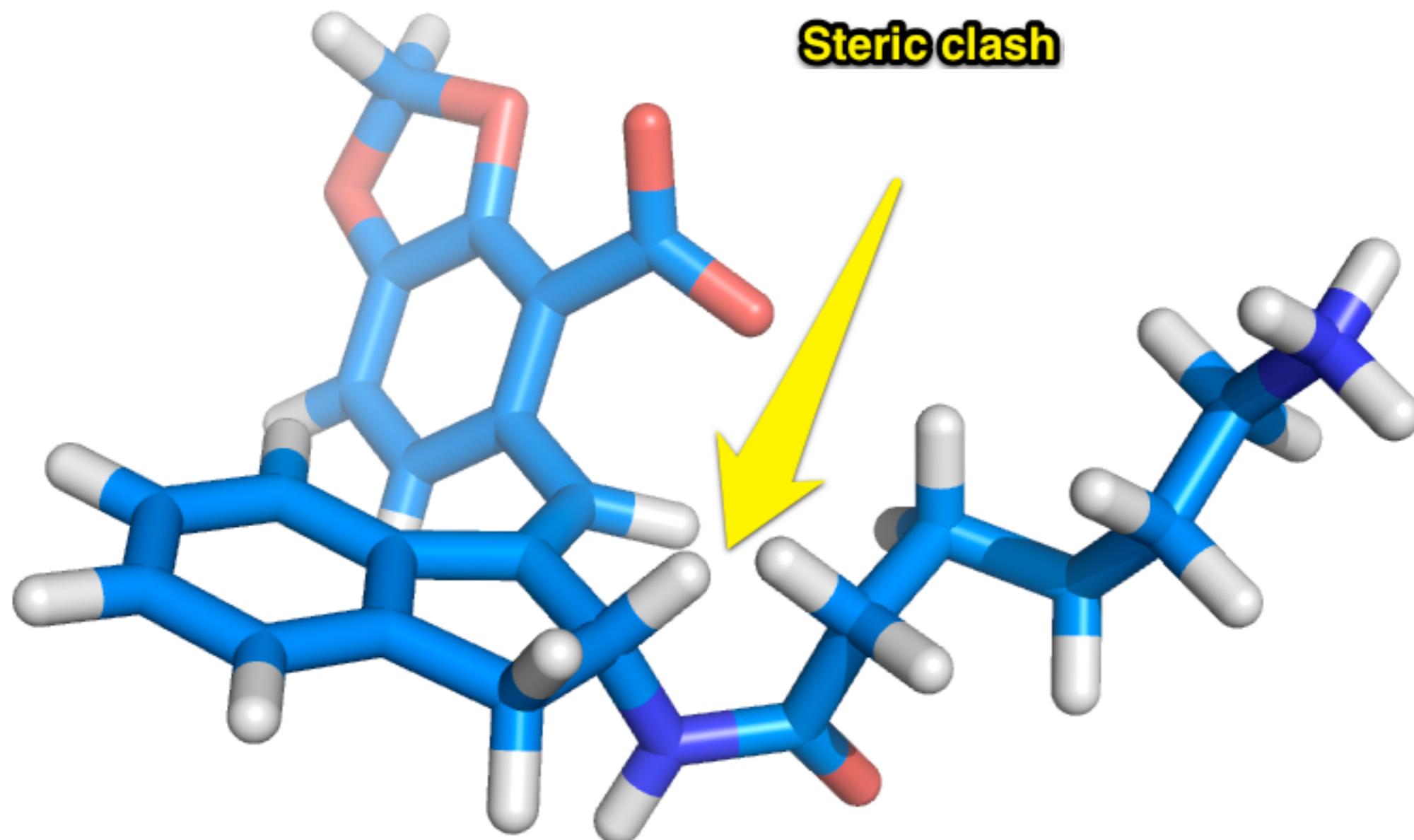
Molecular modeling is used in refinement. Do you have the right balance of forces and density?



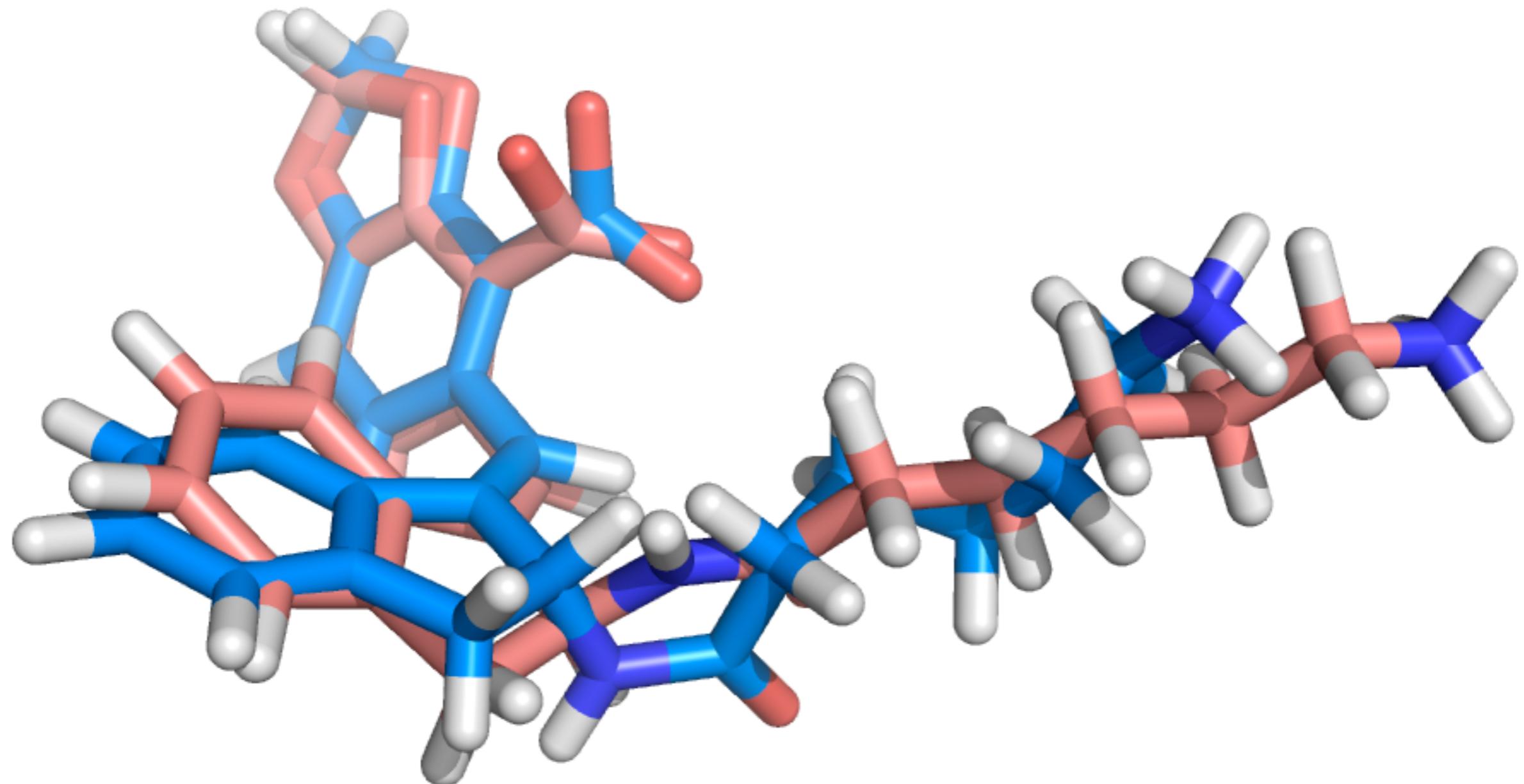
Molecular modeling is used in refinement. Do you have the right balance of forces and density?



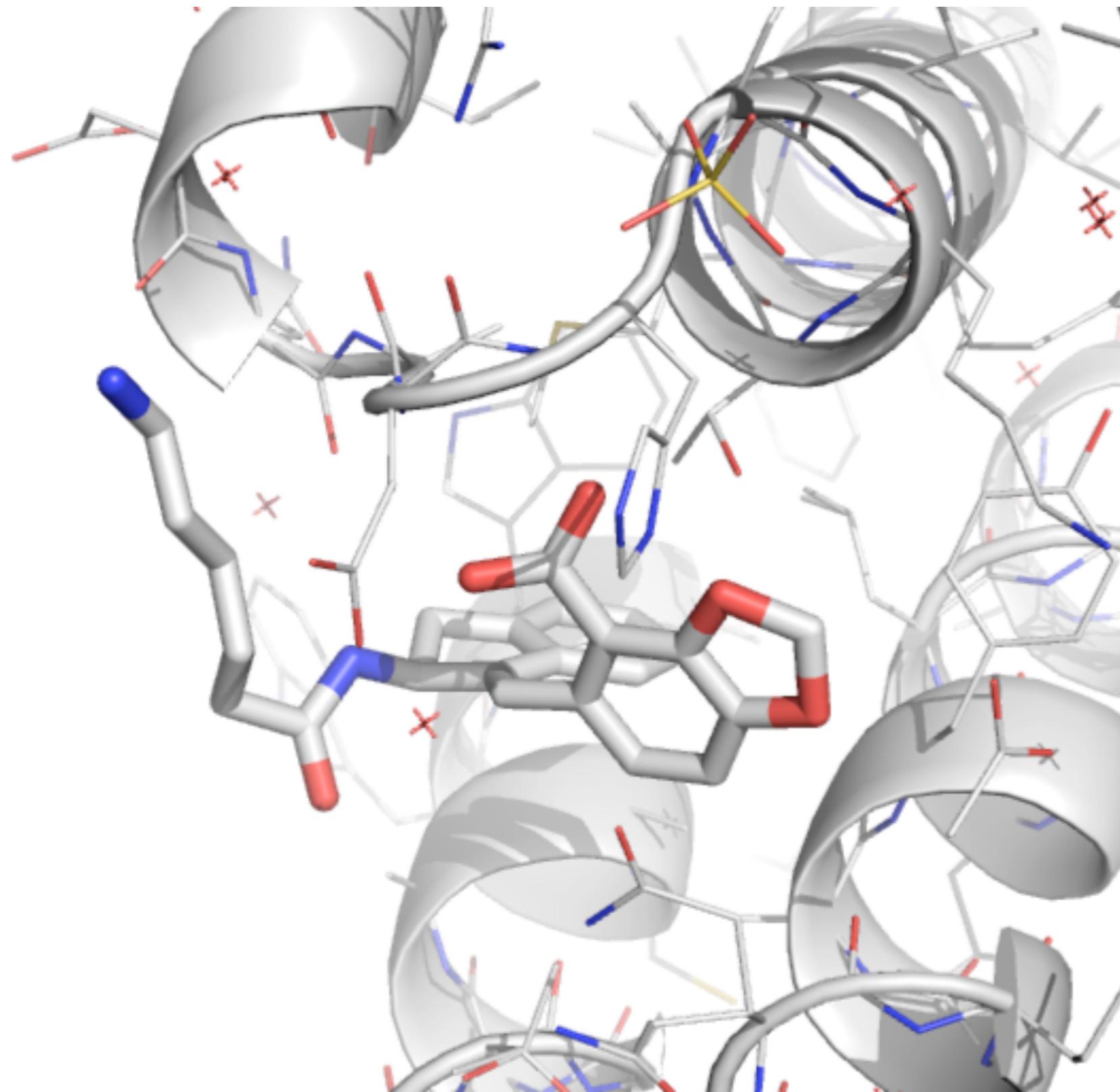
Molecular modeling is used in refinement. Do you have the right balance of forces and density?



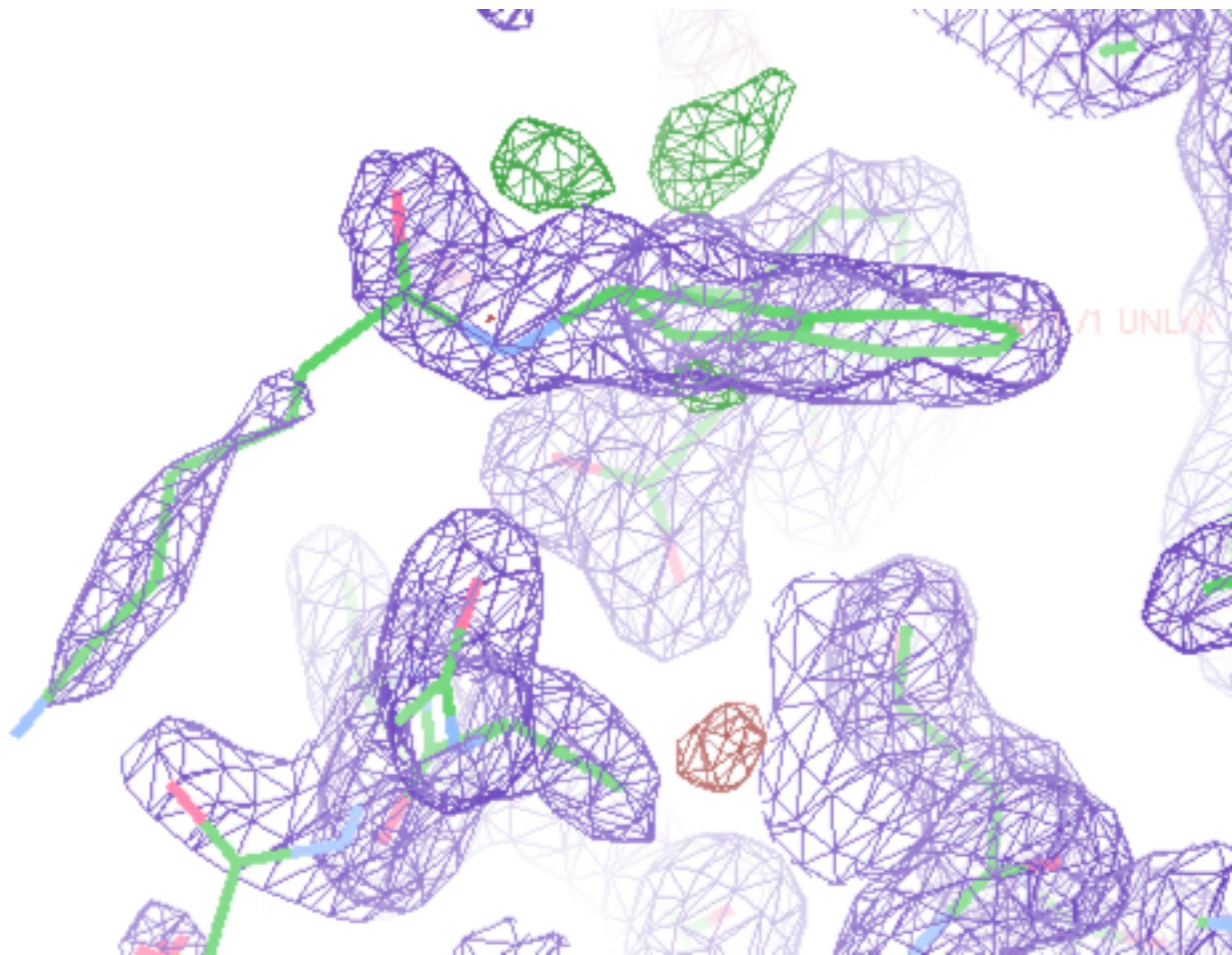
We proposed the opposite stereoisomer, which would eliminate the steric clash



The newly refined model eliminates the steric clash



And it fits the density as well as the original model



PDB information includes sequence info

SEQRES	1	B	19	DT	DG	DG	DA	DG	DA	DT	DG	DA	DC	DG	DT	DC
SEQRES	2	B	19	DA	DT	DC	DT	DC	DC							
SEQRES	1	A	63	MET	ILE	VAL	PRO	GLU	SER	SER	ASP	PRO	ALA	ALA	LEU	LYS
SEQRES	2	A	63	ARG	ALA	ARG	ASN	THR	GLU	ALA	ALA	ARG	ARG	SER	ARG	ALA
SEQRES	3	A	63	ARG	LYS	LEU	GLN	ARG	MET	LYS	GLN	LEU	GLU	ASP	LYS	VAL
SEQRES	4	A	63	GLU	GLU	LEU	LEU	SER	LYS	ASN	TYR	HIS	LEU	GLU	ASN	GLU
SEQRES	5	A	63	VAL	ALA	ARG	LEU	LYS	LYS	LEU	VAL	GLY	GLU	ARG		

Amino acids	ALA, CYS, ASP, GLU, PHE, GLY, HIS, ILE, LYS, LEU, MET, ASN, PRO, GLN, ARG, SER, THR, VAL, TRP, TYR
Deoxyribonucleotides	DA, DC, DG, DT, DI
Ribonucleotides	A, C, G, U, I

(Source: PDB)

Coordinates are a key section in the PDB file

```

1 2 3 4 5 6 7 8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
MODEL 1
ATOM 1 N ALA A 1 11.104 6.134 -6.504 1.00 0.00 N
ATOM 2 CA ALA A 1 11.639 6.071 -5.147 1.00 0.00 C
...
...
...
ATOM 293 1HG GLU A 18 -14.861 -4.847 0.361 1.00 0.00 H
ATOM 294 2HG GLU A 18 -13.518 -3.769 0.084 1.00 0.00 H
TER 295 GLU A 18
ENDMDL
MODEL 2
ATOM 296 N ALA A 1 10.883 6.779 -6.464 1.00 0.00 N
ATOM 297 CA ALA A 1 11.451 6.531 -5.142 1.00 0.00 C
...
...
ATOM 588 1HG GLU A 18 -13.363 -4.163 -2.372 1.00 0.00 H
ATOM 589 2HG GLU A 18 -12.634 -3.023 -3.475 1.00 0.00 H
TER 590 GLU A 18

```

If you want to work with PDB files, look up the format. Fixed width!

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"ATOM "	
7 - 11	Integer	serial	Atom serial number.
13 - 16	Atom	name	Atom name.
17	Character	altLoc	Alternate location indicator.
18 - 20	Residue name	resName	Residue name.
22	Character	chainID	Chain identifier.
23 - 26	Integer	resSeq	Residue sequence number.
27	AChar	iCode	Code for insertion of residues.
31 - 38	Real(8.3)	x	Orthogonal coordinates for X in Angstroms.
39 - 46	Real(8.3)	y	Orthogonal coordinates for Y in Angstroms.
47 - 54	Real(8.3)	z	Orthogonal coordinates for Z in Angstroms.
55 - 60	Real(6.2)	occupancy	Occupancy.
61 - 66	Real(6.2)	tempFactor	Temperature factor.
77 - 78	LString(2)	element	Element symbol, right-justified.
79 - 80	LString(2)	charge	Charge on the atom.

Crystal structures are limiting in some respects

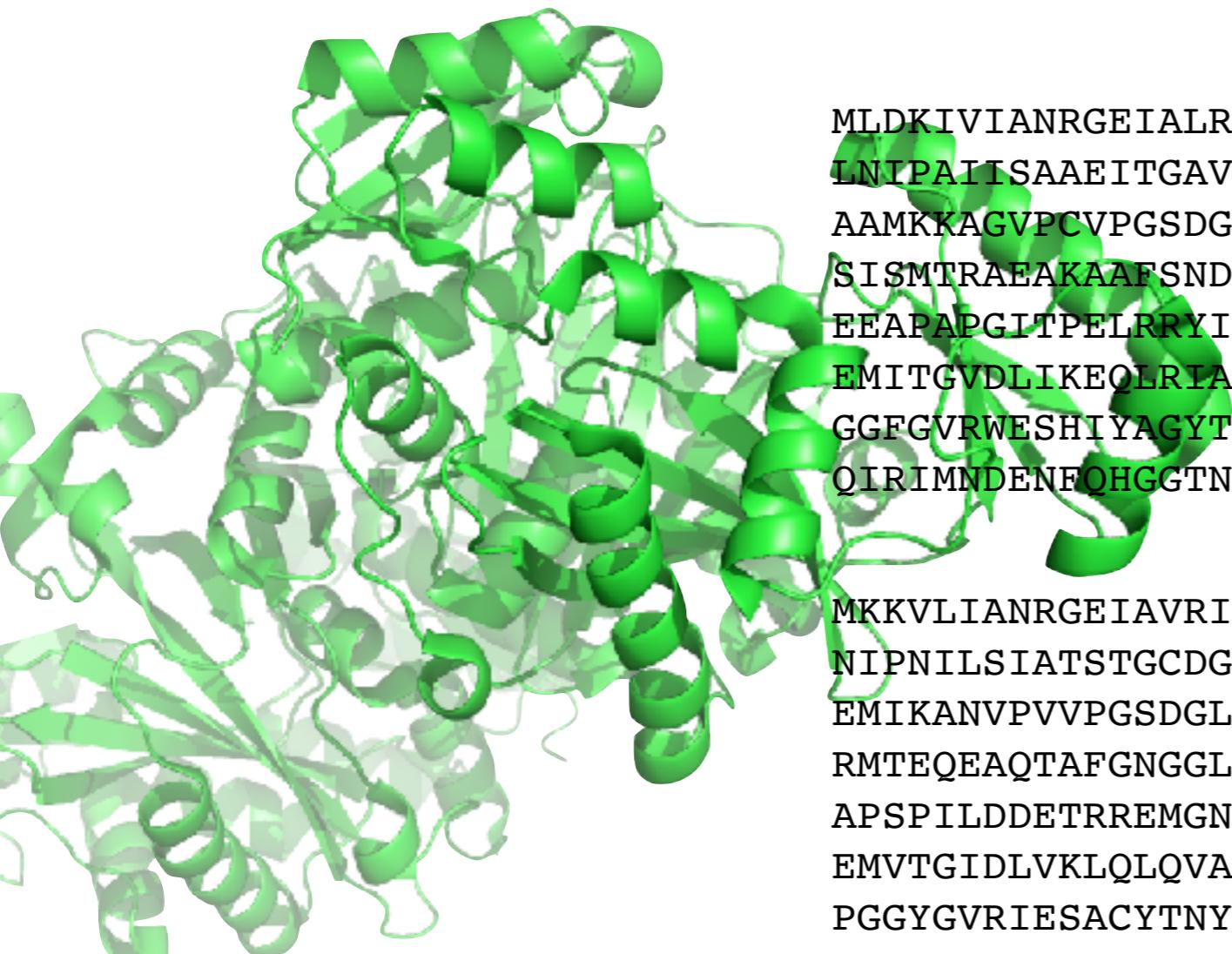
- Proteins are not really static
- They are a *model* fit to the data (the data is the diffraction pattern/electron density)
 - It's easy to confuse them with the data itself
 - Bad modeling yields a bad model!
 - Hetero atoms are particularly problematic (i.e. ligands)
- Crystal conditions, cosolvents, temperature can be important

Homology modeling seeks to solve a common protein structure problem

- We have a protein sequence of interest
 - ex. MET ALA ILE VAL ARG ALA HIS LEU LYS ILE TYR GLY ARG VAL GLN GLY VAL GLY PHE ARG TRP SER MET GLN ARG GLU ...
 - that is, MAIVRAHLKIYGRVQGVGFAWSMNRE...
- We want to predict its structure
 - There are structures of related proteins available
- How do we use the structures of these related proteins to help with our problem?

Homology or comparative models predict new structures based on knowns

- Example: Could we predict the structure of BC in *S. Aureus* based on its structure in *E. Coli*?



MLDKIVIANRGEIALRILRACKELGIKTVAVHSSADRDLKHVLLADETVCIGPAPSVKSY
LNIPAIISAAEITGAVAIHPGYGFLSENANFAEQVERSGFIFIGPKAETIRLMGDKVSAI
AAMKKAGVPCVPGSDGPLGDDMDKNRAIAKRIGYPVIIKASGGGGRGMRVVRGDAELAQ
SISMTRAEEAKAAF SNDMVYMEKYLENPRHVEIQVLADGQGNAYLAERDCSMQRRHQKVV
EEAAPAPGITPELRRYIGERCAKACVDIGYRGAGTFEFLFENGFYFIEMNTRIQVEHPVT
EMITGVDLIKEQLRIAAGQPLSIKQEEVHVRGHAVECRINAEDPNTFLPSPGKITRFHAP
GGFGVRWESHIYAGYTVPYYDSMIGKLICYGENRDVAIARMKNALQELIIDGIKTNVDL
QIRIMMDENFOHGGTNIHYLEKKLGLQEK

MKKVLIANRGEIAVRIIRACRDLGIQTVAIYSEGDKDALHTQIADEAYCVGPTLSKDSYL
NIPNILSIATSTGCDGVHPGYGFLAENADFAELCEACQLFIGPSYQSIQKMGIKDVAKA
EMIKANVPVPGSDGLMKDVSEAKKIAKKIGYPVIIKATAGGGKGIRVARDEKELETGF
RMTEQEAQTAFGNGGLYMEKFIEFRHIEIQIVGDSYGNVIHLGERDCTIQRRMQKLVEE
APSPILDDETTRREMGNAAVRAAKAVNYENAGTIEFIYDLNDNKFYFMEMNTRIQVEHPVT
EMVTGIDLVLQLQVAMGDVLPYKQEDIKTGHAIEFRINAENPYKNFMPSPGKIEQYLA
PGGYGVRIESACYTNYTIPYYDSMVAKLIIHEPTRDEAIMAGIRALSEFVVLGIDTTIP
FHIKLLNNNDIFRSGKFNTNFLEQNSIMNDEG

At this point, it's hard to say -- how similar is the sequence?

- First, let's try to align the sequences and then color code by amino acid category:

sp P24182 ACCC_ECOLI tr Q99TW7 Q99TW7_STAAM	IGPAPSVKSYLNIPAIISAAEITGAVAIHPGYGFLSENANFAEQVERS VGPTLSKDSYLNIPNILSIATSTGCDGVHPGYGFLAENADFAELCEAC	100 99
--	--	-------------------------

sp P24182 ACCC_ECOLI	IFIGPKAETIRLMGDKVSAIAAMKKAGVPCVPGSMDGPLGDDMDKNRAIAK	150
tr Q99TW7 Q99TW7_STAAM	KFIGPSYQSIQKMGIKDVAKAEMIKANVPVVPGSDG-LMKDVSEAKKIAK	148

sp P24182 ACCC_ECOLI	RIGYPVIIIKASGGGGGRGMRVVRGDAELAQSIISMTRAEAKAAF SNDMVYM	200
tr Q99TW7 Q99TW7_STAAM	KIGYPVIIIKATAGGGGKGIRVARDEKELETGFRMTEQEAQTAFGNGGLYM	198

sp P24182 ACCC_ECOLI	EKYLENPRHVEIQVLADGQGNIAIYLAERDCSMQRRHQKVVEEAPAPGITP	250
tr Q99TW7 Q99TW7_STAAM	EKFIENFRHIEIQIVGDSYGNVIHLGERDCTIQRRMQLVVEEAPSPILDD	248

sp P24182 ACCC_ECOLI	ELRRYIGERCAKACVDIGYRGAGTFEFLFENGE--FYFIEMNTRIQVEHP	298
tr Q99TW7 Q99TW7_STAAM	ETRREMGNAAVRAAKAVNYENAGTIEFIYDLNDNKFYFMEMNTRIQVEHP	298

It turns out the structure is already available, and almost identical despite sequence differences

sp|P24182|ACCC_ECOLI
tr|Q99TW7|Q99TW7_STAAM

MLDKIVIANRGEIALRILRACKELGIKTVAVHSSADRDLI
-MKKVLIANRGEIAVRIIRACRDLGIQTVAIYSEGDKDIA
* *

sp|P24182|ACCC_ECOLI
tr|Q99TW7|Q99TW7_STAAM

IGPAPSVKSYLNIPAIISAAEITGAVAIHPGYGFLS
VGPTLSKDSYLNIPNILSIATSTGCDGVHPGYGFL
* *

sp|P24182|ACCC_ECOLI
tr|Q99TW7|Q99TW7_STAAM

IFIGPKAETIRLMGDKVSAIAAMKKAGVPCVPGS
KFIGPSYQSIQKMGIKDVAKAEMIKANVPVVPGS
* *

sp|P24182|ACCC_ECOLI
tr|Q99TW7|Q99TW7_STAAM

RIGYPVIIKASGGGGGRGMRVVRGDAELAQSI
KIGYPVIIKATAAGGGKGIRVARDEKELETGFR
* *

sp|P24182|ACCC_ECOLI
tr|Q99TW7|Q99TW7_STAAM

EKYLENPRHVEIQVLADGQGNIAIYLAERD
EKFIENFRHIEIQTIVGDSYGNVIHLGER
* *

sp|P24182|ACCC_ECOLI
tr|Q99TW7|Q99TW7_STAAM

ELRRYIGERCAKACVDIGYRGAGTFEFLFENG
ETRREMGNAAVRAAKAVNYENAGTIEIFIYDLNDNK
* *

sp|P24182|ACCC_ECOLI
tr|Q99TW7|Q99TW7_STAAM

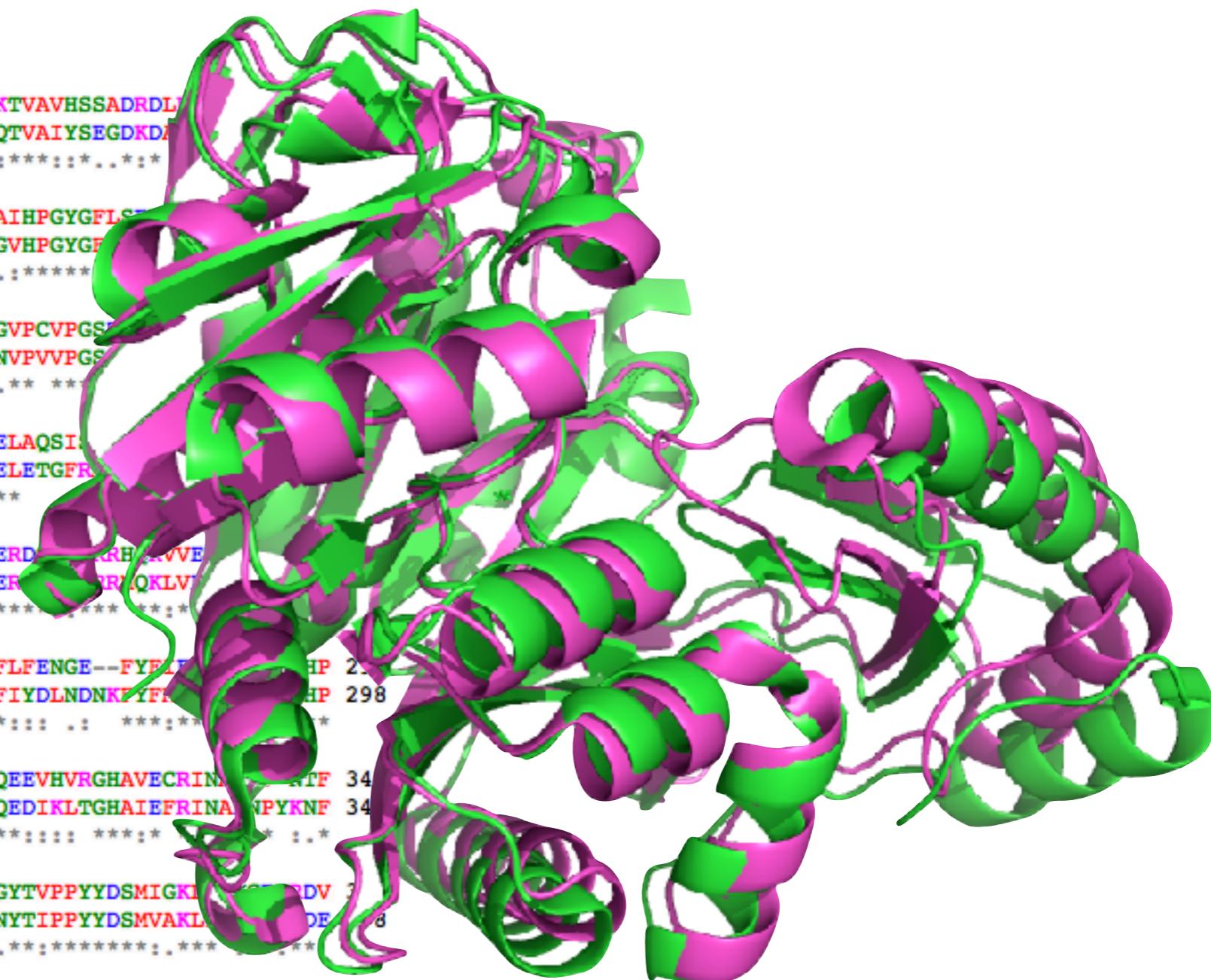
VTEMITGVLDLKEQLRIAAGQPLSIKQEEVHVRGHAVECRIN
VTEMVTGIDLVLQLQLQVAMGDVLPYKQEDIKLTGHIAEFRINA
* *

sp|P24182|ACCC_ECOLI
tr|Q99TW7|Q99TW7_STAAM

LPSPGKITRFHAPGGFGVRWESHIYAGYTVPYYDSMIGK
MPSPGKIEQYLAPGGYGVRIESACYTNYTIPYYDSMVAKL
* *

sp|P24182|ACCC_ECOLI
tr|Q99TW7|Q99TW7_STAAM

AIARMKNALQELIIDGIKTNVDLQIRIMNDENFQHGGTNIHYLEKKLGLQ 447
AIMAGIRALSEFVVLGIDTTIPFHIKLLNNDFRSGKFNTNFLEQNSIMN 448
* *



This motivates homology or comparative modeling: Often, proteins have high structural similarity even at low sequence identity

- Many tools available for homology modeling
- In general, high sequence identity is not necessary
 - Above 50% sequence identity, models are generally quite good, with errors mostly in sidechain positioning
 - In 30-50% range, errors can be more severe, but often still tolerable (generally considered acceptable)
 - Below 30%, all bets are off

Sequence identity and sequence similarity are related but different concepts

- A low sequence identity scenario can be OK if sequence similarity is high
- Sequence identity: Fraction of amino acids that are exactly the same
- Sequence similarity: Fraction that have similar properties (i.e. polar, positively charge, negatively charged, hydrophobic)

For high sequence identity models to work, there are two key ingredients

- Good choice of related (template) structure(s)
- Good sequence alignment

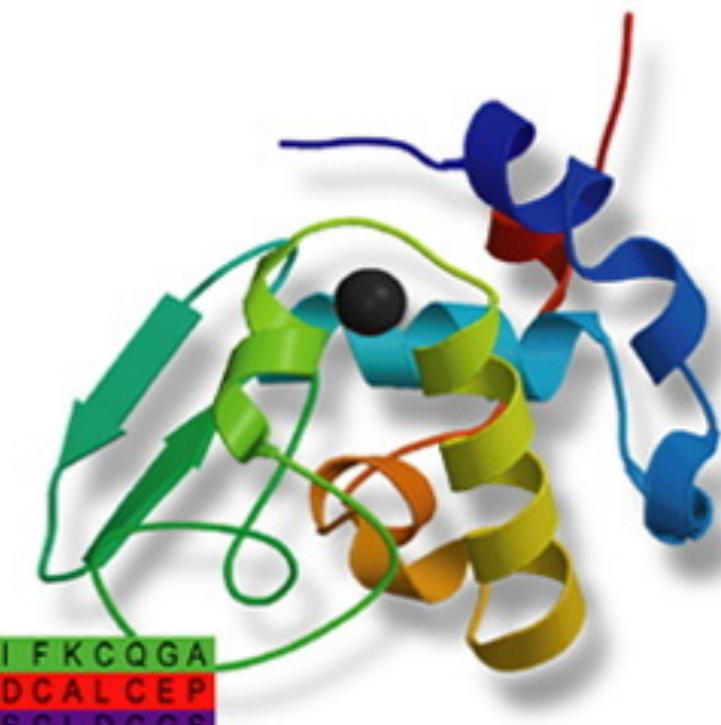
AAB24882	TYHMCQFHCRYVNNHSGEKL <color>LYECNERSKA</color> FSCP <color>SHLQCHKRRQIGEK</color> I <color>THEHNQCGKAFPT</color> 60
AAB24881	-----YE <color>CNQCGKAF</color> AQHSSLK <color>CHYRTHI</color> G <color>EKPYECNQCGKAFSK</color> 40
	*****: .***: * *:*** * :****. :* *****..
AAB24882	PSHLQYHERTHTGEKPYE <color>CHQCGQAFKKCSLLQRHKRTHTGEKPYE-CNQCGKAF</color> AQ- 116
AAB24881	HSHLQCHKR <color>THTGEKPYECNQCGKAFSQHGLLQRHKRTHTGEKPYMNVINMV</color> KPLHNS 98
	***** *:*****:*****:***:***.: .*****:***** : *.: :

We will look at Modeller, a useful tool for homology modeling (many other good ones out there)

Modeller

Program for Comparative Protein Structure Modelling by Satisfaction of Spatial Restraints

A	I	L	V	G	S	M	P	R	R	D	G	M	E	R	K	D	L	L	K	A	N	V	K	I	F	K	C	Q	G	A
V	E	V	C	P	V	D	C	F	Y	E	G	P	N	F	L	V	I	H	P	D	E	C	I	D	C	A	L	C	E	P
G	A	C	K	P	E	C	P	V	N	I	Q	G	S	-	-	I	Y	A	I	D	A	D	S	C	I	D	C	G	S	
C	-	-	I	A	C	G	A	C	K	P	E	C	P	V	N	I	Q	G	S	-	-	I	Y	A	I	D	A	D	S	



<http://salilab.org/modeller/documentation.html> -- has tutorials! (We'll follow one of these)

conda install -c salilab modeller

The process starts with picking our target -- i.e. lactate dehydrogenase from a particular bacteria (Tv)

- Step 1: Get our target sequence and search for related structures

```
>P1 ;TvLDH
sequence:TvLDH:::::0.00: 0.00
MSEAAHVLITGAAGQIGYILSHWIASGELYGDRQVYLHLLDIPPAMNRLTALTMELEDCAFPHLAGFVATTDPKA
AFKDIDCAFLVASMPLKPGQVRADLISSNSVIFKNTGEYLSKWAKPSVKVLVIGNPDNTNCEIAMLHAKNLKPE
FSSLSMLDQNRAYYEVASKLGVDVHDIIIVWGNHGESMVADLTQATFTKEGKTQKVVDVLDHDYVFDTFFKKI
GHRAWDILEHRGFTSAASPTKAAIQHMKAWLFGTAPGEVLSMGIPVPEGNPYGIKPGVVFSFPCNVDKEGKIHV
EGFKVNDWLREKLDFTKEKDLFHEKEIALNHLAQGG*
```

- The tutorial provides a Python script which uses Modeller to do this search
- Searches a clustered set of PDB sequences (at 95% similarity) to find those similar to this

This is called building a profile; once we do it, we need to decide what structure to use as a template

```
# Number of sequences:      30
# Length of profile :     335
# N_PROF_ITERATIONS :      1
# GAP_PENALTIES_1D :   -900.0    -50.0
# MATRIX_OFFSET :        0.0
# RR_FILE :           ${MODINSTALL8v1}/modlib//as1.sim.mat
  1  TvLDH
  2  1a5z
  3  1b8pA
  4  1bdmA
  5  1t2dA
  6  1civA
  7  2cmd
  8  1o6zA
  9  1ur5A
 10  1guzA
 11  1gv0A
 12  1hyeA
 13  1ic2A
 14  1i10A
 15  1ldnA
 16  61dh
 17  214x
 18  51dh
 19  91dtA
 20  111c
 21  111dA
 22  5mdhA
 23  7mdhA
 24  1mldA
 25  1oc4A
 26  1ajuA
 27  1pzgA
 28  1smkA
 29  1sovA
 30  1y6jA

          S   0   335     1   335     0   0   0   0   0
          X   1   312     75  242    63  229  164  28.  .83E-04
          X   1   327     7   331     6  329  316  42.  0.0
          X   1   318     1   325     1  310  309  45.  0.0
          X   1   315     5   256     4  250  238  25.  0.66E-04
          X   1   374     6   334    33  318  325  35.  0.0
          X   1   312     7   320     3  303  289  27.  0.16E-05
          X   1   303     7   320     3  287  276  26.  0.27E-05
          X   1   299    13  191     9  171  158  11.  0.25E-02
          X   1   305    13  301     8  280  266  25.  0.28E-08
          X   1   301    13  323     8  289  274  26.  0.28E-04
          X   1   307     7  191     3  183  173  29.  0.4E-07
          X   1   332    85  300    94  304  207  26.  0.56E-05
          X   1   331    85  295    93  298  196  26.  0.56E-05
          X   1   316    78  298    73  301  211  26.  0.9E-03
          X   1   329    47  301    56  302  244  23.  0.34E-2
          X   1   331    66  306    67  306  227  23.  0.35E-04
          X   1   333    85  300    94  304  207  26.  0.30E-05
          X   1   331    85  301    93  304  207  26.  0.30E-05
          X   1   321    64  239    53  234  161  26.  0.20E-03
          X   1   313    13  242     9  233  216  31.  0.31E-07
          X   1   333     2  332     1  331  328  44.  0.0
          X   1   351     6  334    14  339  329  34.  0.0
          X   1   313     5  198     1  199  183  26.  0.13E-05
          X   1   315     5  191     4  186  174  28.  0.18E-04
          X   1   294    78  320    68  285  218  28.  0.43E-05
          X   1   327    74  191    71  190  114  30.  0.16E-06
          X   1   313     7  202     4  198  188  34.  0.0
          X   1   316    81  256    76  248  160  27.  0.93E-03
          X   1   289    77  191    58  167  109  33.  0.32E-09
```

PDB code

len. iden% val.

Next, we zero in on some specific structures which look interesting

# Number of sequences:	30	# Length of profile :	335	# N_PROF_ITERATIONS :	1	# GAP_PENALTIES_1D :	-900.0 -50.0	# MATRIX_OFFSET :	0.0	# RR_FILE :	\$(MODINSTALL8v1)/modlib//asl.sim.mat
1	TvLDH	S	0	335	1	335	0	0	0	0.	0.0
2	1a5z	X	1	312	75	242	63	229	164	28.	0.83E-08
3	1b8pA	X	1	327	7	331	6	325	316	42.	0.0
4	1bdmA	X	1	318	1	325	1	310	309	45.	0.0
5	1c2dm	X	1	315	5	256	4	250	238	25.	0.66E-04
6	1civA	X	1	374	6	334	33	358	325	35	0.0
7	2cmd	X	1	312	7	320	3	303	289	27.	0.16E-05
8	1o6zA	X	1	303	7	320	3	287	278	26.	0.27E-05
9	1ur5A	X	1	299	13	191	9	171	158	31.	0.25E-02
10	1guzA	X	1	305	13	301	8	280	265	25.	0.28E-08
11	1gv0A	X	1	301	13	323	8	289	274	26.	0.28E-04
12	1hyeA	X	1	307	7	191	3	183	173	29.	0.14E-07
13	1i0zA	X	1	332	85	300	94	304	207	25.	0.66E-05
14	1i10A	X	1	331	85	295	93	298	196	26.	0.86E-05
15	1ldnA	X	1	316	78	298	73	301	214	26.	0.19E-03
16	6ldh	X	1	329	47	301	56	302	244	23.	0.17E-02
17	2ldx	X	1	331	66	306	67	306	227	26.	0.25E-04
18	5ldh	X	1	333	85	300	94	304	207	26.	0.30E-05
19	9ldtA	X	1	331	85	301	93	304	207	26.	0.10E-05
20	1llc	X	1	321	64	239	53	234	164	26.	0.20E-03
21	1lldA	X	1	313	13	242	9	233	216	31.	0.31E-07
22	5mdhA	X	1	333	2	332	1	331	328	44.	0.0
23	7mdhA	X	1	351	6	334	14	339	325	34.	0.0
24	1mldA	X	1	313	5	198	1	189	183	26.	0.13E-05
25	1oc4A	X	1	315	5	191	4	186	174	28.	0.18E-04
26	1ojuA	X	1	294	78	320	68	285	218	28.	0.43E-05
27	1pwcA	X	1	327	74	191	71	190	114	30.	0.16E-06
28	1smkA	X	1	313	7	202	4	198	188	34.	0.0

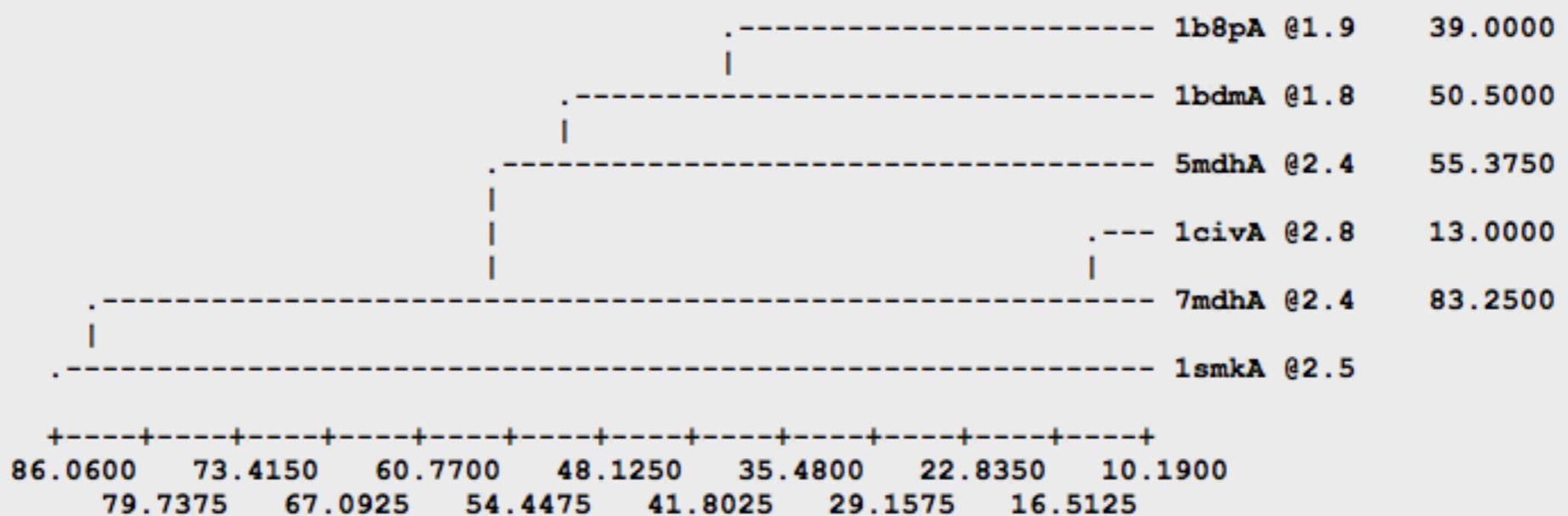
We then run a structural comparison on these to get some additional insight

Sequence identity comparison (ID_TABLE) :

Diagonal ... number of residues;
Upper triangle ... number of identical residues;
Lower triangle ... % sequence identity, id/min(length).

	1b8pA @1	11bdmA @1	1civA @2	25mdhA @2	27mdhA @2	21smkA @2
1b8pA @1	327	194	147	151	153	49
1bdmA @1	61	318	152	167	155	56
1civA @2	45	48	374	139	304	53
5mdhA @2	46	53	42	333	139	57
7mdhA @2	47	49	87	42	351	48
1smkA @2	16	18	17	18	15	313

Weighted pair-group average clustering based on a distance matrix:



We then run a structural comparison on these to get some additional insight

Sequence identity comparison (ID_TABLE) :

Diagonal ... number of residues;
Upper triangle ... number of identical residues;
Lower triangle ... % sequence identity, id/min(length).

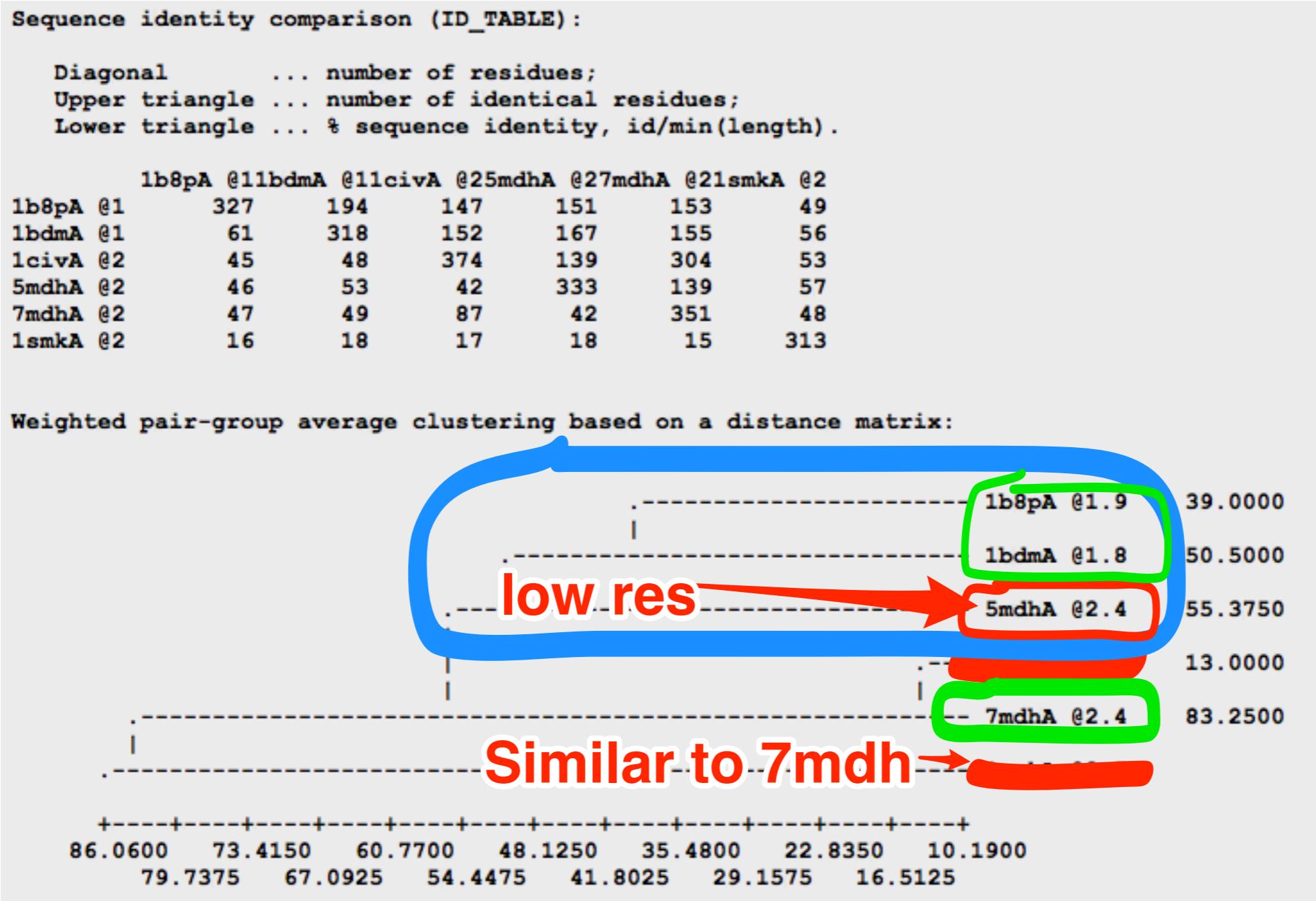
	1b8pA @1	1lbdmA @1	1civA @2	5mdhA @2	7mdhA @2	1smkA @2	
1b8pA @1	327	194	147	151	153	49	
1lbdmA @1	61	318	152	167	155	56	
1civA @2	45	48	374	139	304	53	
5mdhA @2	46	53					
7mdhA @2	47	49					
1smkA @2	16	18	17	18	15	313	

High sequence similarity

Weighted pair-group average clustering based on a distance matrix:



We then run a structural comparison on these to get some additional insight



For high sequence identity models to work, there are two key ingredients

- Good choice of related (template) structure(s)
- Good sequence alignment

AAB24882	TYHMCQFHCRYVNNHSGEKL <color>LYECNERSKA</color> FSCP <color>SHLQCHKRRQIGEK</color> T <color>HENQCGKAFPT</color> 60
AAB24881	-----YE <color>CNQCGKAF</color> AQHSSLK <color>CHYRTHI</color> G <color>EKPYECNQCGKAFSK</color> 40
	*****: . ***: * *:*** * :*****, :* *****.. .
AAB24882	PSHLQYHERTHTGEKPYE <color>CHQCGQAFKKCSLLQRHKRTHTGEKPYE-CNQCGKAF</color> AQ- 116
AAB24881	HSHLQCHKR <color>THTGEKPYECNQCGKAFSQHGLLQRHKRTHTGEKPYMNVINMVKPLHNS</color> 98
	***** * :*****:*****:***:***.: .*****:***** : *.: :

Next, we do a sequence alignment in Modeller

		10	20	30	40	50	60							
_aln.pos	1bdmA	MKAPVRVAVTGAAGQIGYSLLFRIAAGEMLGKDQPVILQLLEIPQAMKALEGVVMELEDCAFPLLAGL												
	TvLDH	MSEAAHVVLITGAAGQIGYILSHWIASGELYG-DRQVYLHLLDIPPAMNRLTALTMELEDCAFPHLAGF												
_consrvd		*	*	*****	*	**	*	*	*	*	*	*	*****	***
		70	80	90	100	110	120	130						
_aln.p	1bdmA	EATDDPDVAFKDADYALLVGAAPRL-----	QVNNGKIFTEQGRALAEVAKKDVKVLVVGNPANTN											
	TvLDH	VATTDPKAASFIDCAFLVASMPLKPGQVRADLISSNSVIFKNTGEYLSKWA	KPSVKVLVIGNPDNTN											
_consrvd		***	***	*****	*	**	*	*	*	**	*****	***	***	
		140	150	160	170	180	190	200						
_aln.pos	1bdmA	ALIAKNAPGLNPRNFTAMTRLDHNRAKAQLAKKTGTGVDRIRRMTVWGNHSSIMFPDLFHAEVD--												
	TvLDH	CEIAMLHAKNLKPENFSSLMSMLDQNRAYYEVASKLGVDVKDVHDIIIVWGNHGESMVADLTQATFTKEG												
_consrvd		**	*	* * **	** ***	* * *	*	*****	*	**	*			
		210	220	230	240	250	260	270						
_aln.pos	1bdmA	-GRPALELVDMEWYEKFPIPTVAQRGAIIIQARGASSAASAANAAIEHIRDWALGTPEGDWVSMAVPS												
	TvLDH	KTQKVVDVLDHDYVFDTFFKKIGHRAWDILEHRGFTSAASPTKAAIQHMKAWLFGTAPGEVLSMGIPV												
_consrvd		*	*	*	**	****	*** *	*	**	*	**	*		
		280	290	300	310	320	330							
_aln.pos	1bdmA	Q--GEYGIPEGIVYSFPVTAK-DGAYRVVEGLEINEFARKRMEITAQELLDEMEQVKAL--GLI												
	TvLDH	PEGNPYGIKPGVVFSFPCNVDKEGKIHVVEGFKVNDWLREKLDFTEKDLFHEKEIALNHLAQGG												
_consrvd		***	*	***	*	****	*	*	*	*	*	*		

The next step is building a model -- or really, multiple models

- Model building involves something like:
 - Fit the model backbone trace along the template
 - Build in the sidechains
 - Refine model based on energies
 - Optimize sidechains

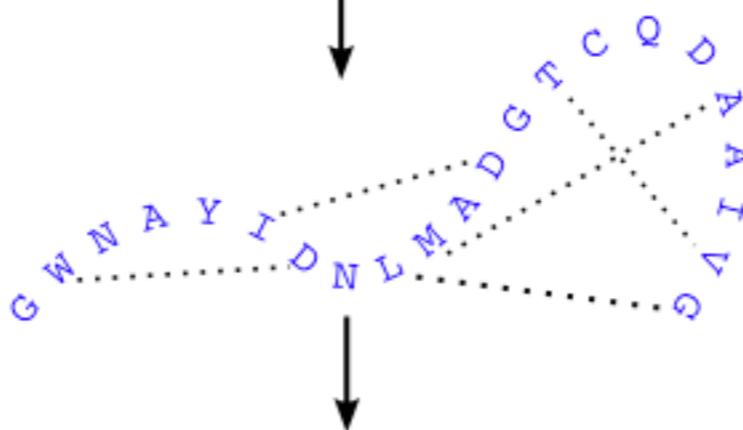
Model building works on identifying and optimizing subject to spatial restraints and energy terms

1. Align sequence with structures

Template structure(s)
Target sequence

SWQTYVDTNLVGTGAVTQA - AI
- GWNAYIDNLMADGTCQDAIIVG

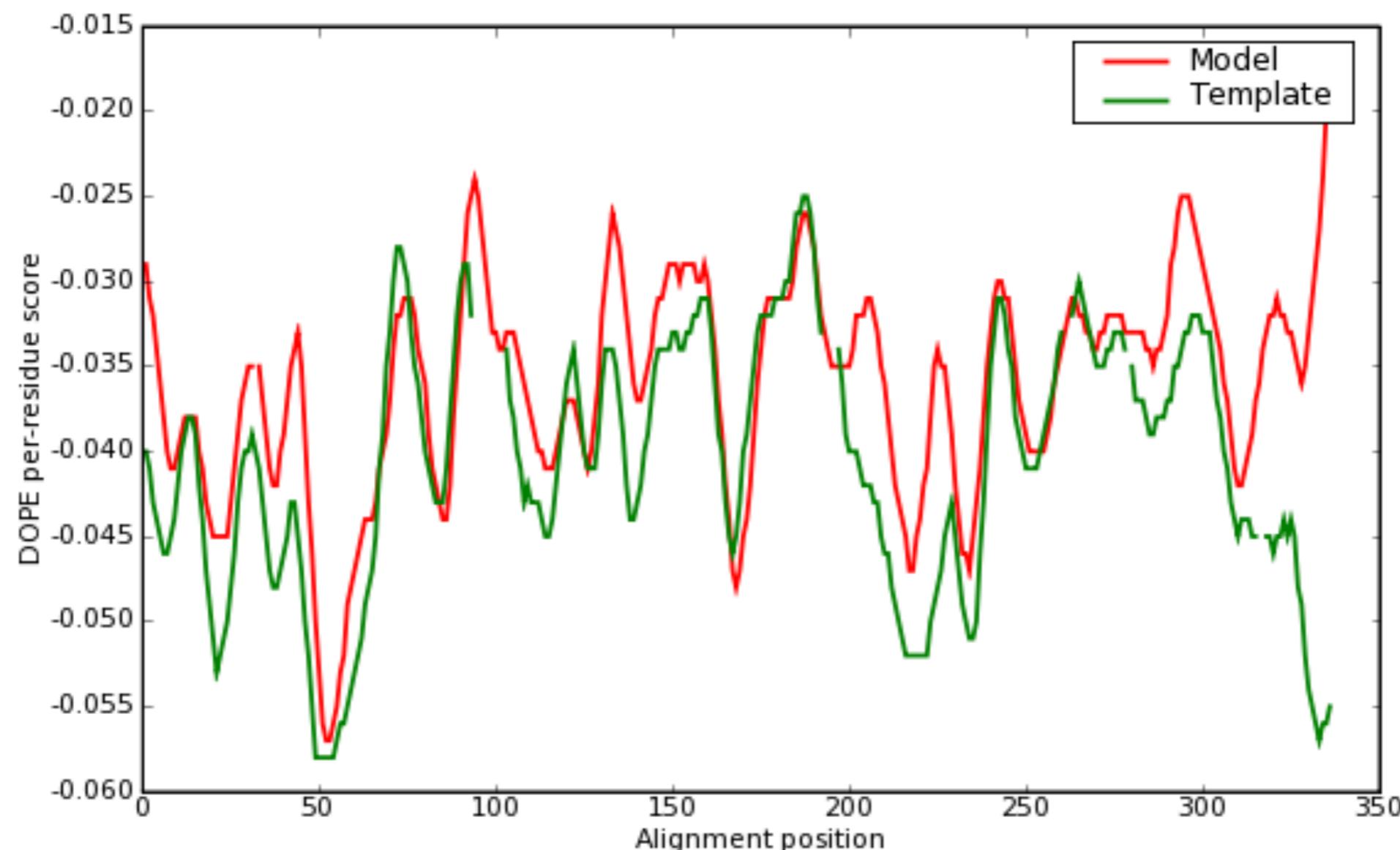
2. Extract spatial restraints



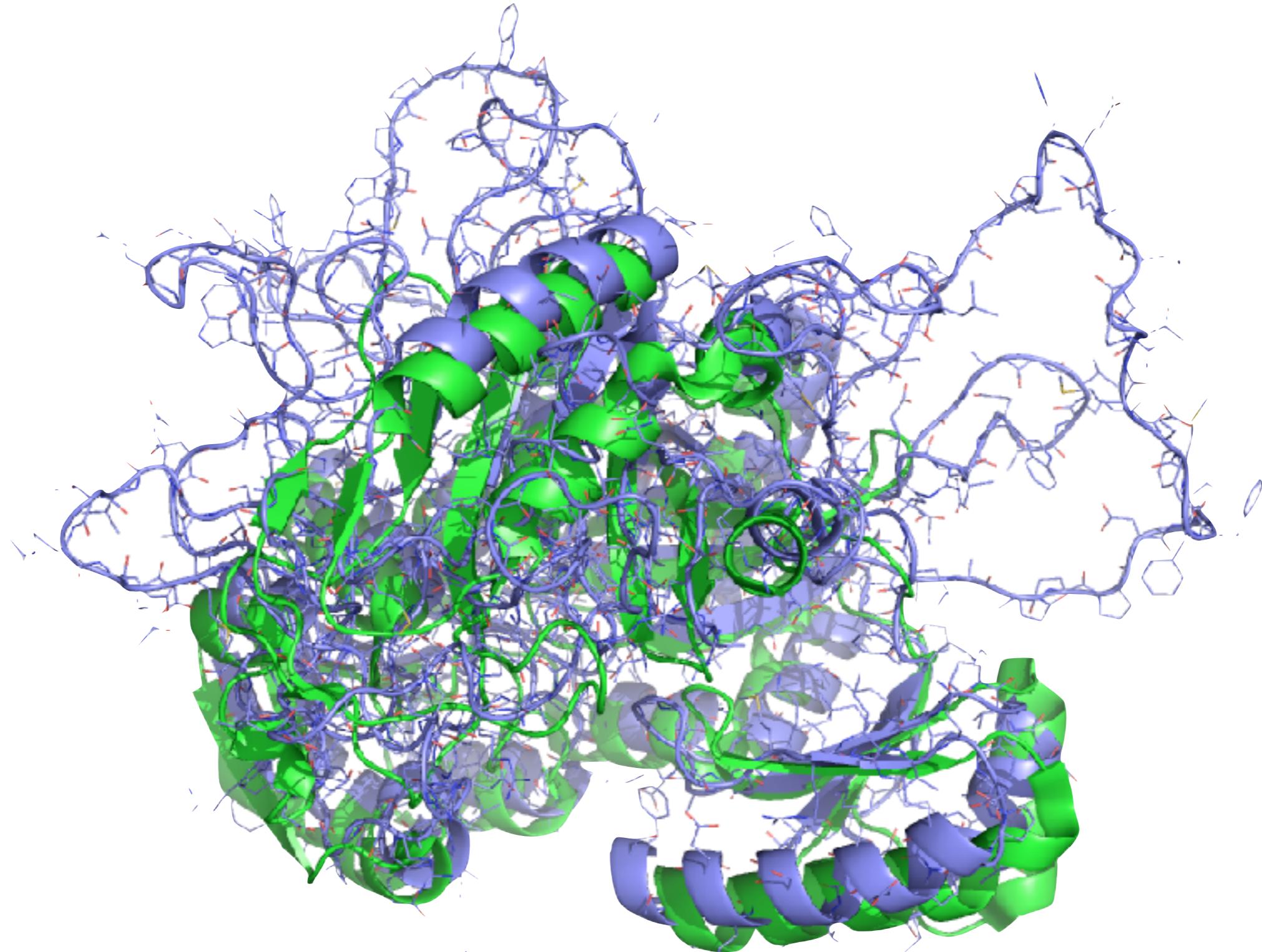
3. Satisfy spatial restraints



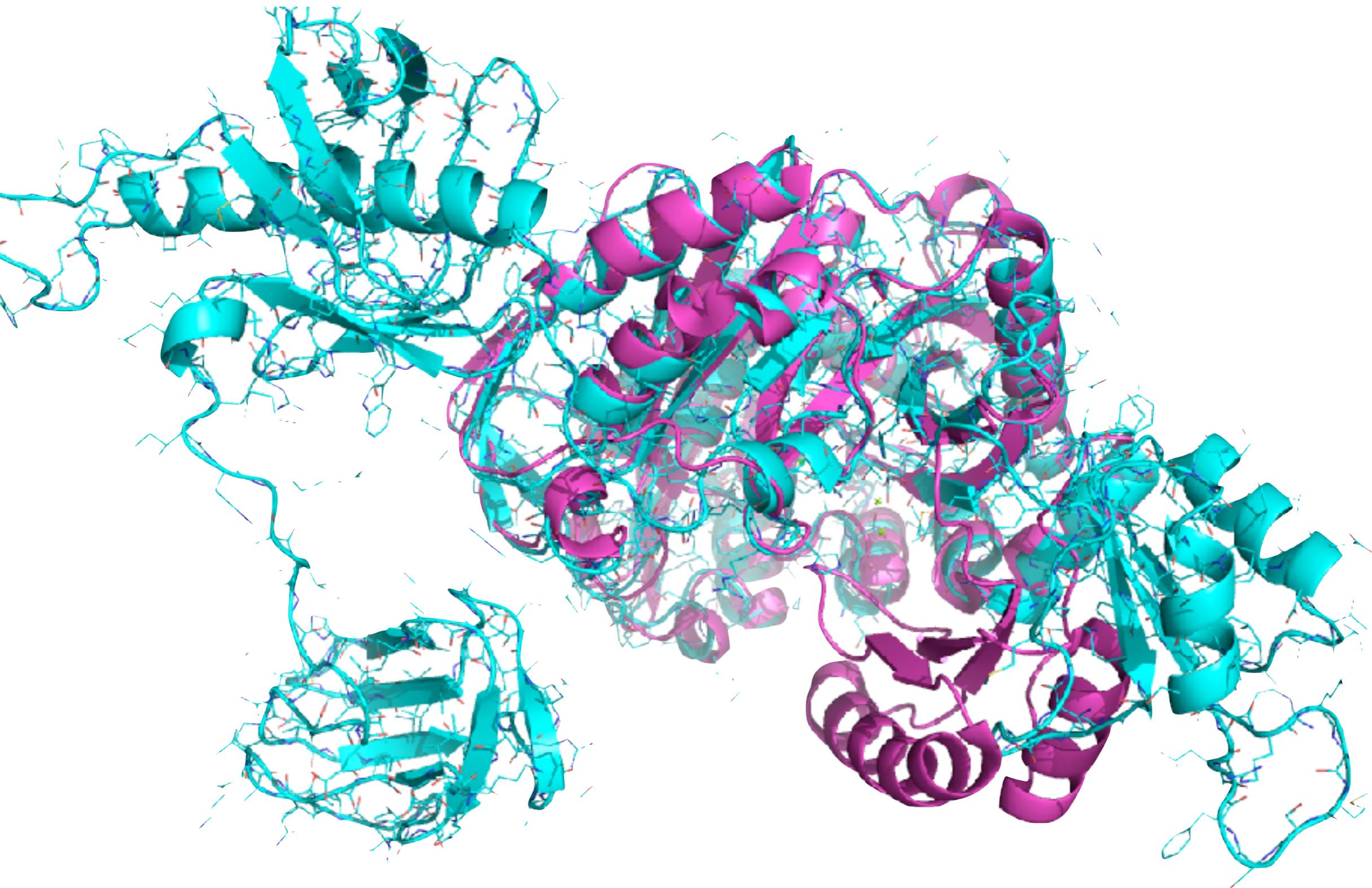
We evaluate the model as a final step with an energy score along the sequence



Looking at the actual model and its overlay onto the template is always helpful



In this case, fixing the alignment solves the problem



More advanced homology modeling can involve additional elements

- Using multiple template structures
 - i.e. one template gives info about one part of the structure, another part about another
- Building in constraints based on experimental data and/or binding partners
- Model refinement, such as with simulations

Ensembler

Anaconda Cloud 1.0.5 docs latest build passing

Software pipeline for automating omics-scale protein modeling and simulation setup.

Online documentation

- Go to the [official online documentation](#).
- Read a preprint of the paper [on bioRxiv](#).
- See the example dataset from [modeling all human tyrosine kinases](#).

Authors

- Daniel L. Parton | daniel.parton@choderalab.org
- John D. Chodera | john.chodera@choderalab.org
- Patrick B. Grinaway | patrick.grinaway@choderalab.org

Overview of pipeline

1. Retrieve protein target sequences and template structures.
2. Build models by mapping each target sequence onto every available template structure, using [Modeller](#).
3. Filter out non-unique models (based on a RMSD cutoff).
4. Refine models with implicit solvent molecular dynamics simulation.
5. Refine models with explicit solvent molecular dynamics simulation.
6. (optional) Package and/or compress the final models, ready for transfer or for set-up on other platforms such as [Folding@Home](#).

Python can interface easily with the PDB itself: Example, solid state NMR structures

```
import urllib.request
url = 'http://www.rcsb.org/pdb/rest/search'
queryText = """<?xml version="1.0" encoding="UTF-8"?>
<orgPdbQuery>
<version>B0907</version>
<queryType>org.pdb.query.simple.ExpTypeQuery</queryType>
<description>Experimental Method Search : Experimental Method=SOLID-
STATE NMR</description>
<mvStructure.expMethod.value>SOLID-STATE NMR</
mvStructure.expMethod.value>
</orgPdbQuery>"""
print("query:\n", queryText)
print("querying PDB...\n")
req = urllib.request.Request(url, data=queryText.encode())
f = urllib.request.urlopen(req)
result = f.read().decode()

if result:
    print("Found number of PDB entries:", result.count('\n'))
else: print("Failed to retrieve results")
```

Running this produces some simple output; we can also get the referenced structures

```
pdbcodes = result.split()  
print(pdbcodes)  
  
url = 'http://www.rcsb.org/pdb/files/%s.pdb' % pdbcodes[0]  
file = urllib.request.urlopen(url)  
pdbcontents = file.readlines()  
  
['1CEK', '1EQ8', '1M8M', '1MAG', '1MP6', '1MZT', '1NH4', '1NYJ', '1PI7',  
'1PI8', '1PJD', '1PJE', '1PJF', '1Q7O', '1RVS', '1XSW', '1ZN5', '1ZY6',  
'2C0X', '2CZP', '2E8D', '2H3O', '2H95', '2JSV', '2JU6', '2JZZ', '2K0P',  
'2KAD', '2KB7', '2KHT', '2KIB', '2KJ3', '2KLR', '2KQ4', '2KQT', '2KRJ',  
'2KSJ', '2KWD', '2KYV', '2L0J', '2L3Z', '2LBU', '2LEG', '2LGI', '2LJ2',  
'2LME', '2LMN', '2LMO', '2LMP', '2LMQ', '2LNL', '2LNQ', '2LNY', '2LPZ',  
'2LTQ', '2LU5', '2M02', '2M3B', '2M3G', '2M4J', '2M5K', '2M5M', '2M5N',  
'2M67', '2MC7', '2MCU', '2MCV', '2MCW', '2MCX', '2MEX', '2MJZ', '2MME',  
'2MMU', '2MPX', '2MPZ', '2MS7', '2MSG', '2MTZ', '2MVX', '2MXU', '2NOA',  
'2N0R', '2N1E', '2N1F', '2N28', '2N3D', '2N70', '2N7H', '2NNT', '2RLZ',
```

Much more complicated queries of the PDB are straightforward

SEARCH service

This interface exposes the RCSB PDB [advanced search](#) interface as an XML Web Service.

To use this service, POST a XML representation of an advanced search to </pdb/rest/search>.

XML representation of advanced search

Every [advanced search](#) can be represented by XML. To view an example representation, simply execute an advanced search query and then click on the [Result tab](#). One of the links on the top of the page is [Query Details](#). Clicking on this link will provide the XML representation of the query.

Every query is described by two data items:

- **queryType:** the name of the class that is implementing the query
- **arguments:** depending on the type of query that is being executed one or more differently named arguments need to be provided.

<http://www.rcsb.org/pdb/software/rest.do#search>

You can easily get example code to help you build your queries

Sample XML Queries:

Author Name

The Query is displayed in the Textbox below

```
<orgPdbQuery>
<queryType>org.pdb.query.simple.AdvancedAuthorQuery</queryType>
<description>Author Search: audit_author.name=Miller, J.S OR
(citation_author.name=Miller, J.S AND
citation_author.citation_id=primary)</description>
<searchType>0</searchType>
<audit_author.name>Miller, J.S</audit_author.name>
</orgPdbQuery>
```

You can even search for a specific sequence without leaving Python

```
<orgPdbQuery>
<queryType>org.pdb.query.simple.SequenceQuery</queryType>
<description>Sequence Search (Structure:Chain = 4HHB:A, Expectation
Value = 10.0, Search Tool = BLAST)</description>
<structureId>4HHB</structureId>
<chainId>A</chainId>
<sequence>VLSPADKTNVKAAGKVGAHAGEYGAELERMFLSFPTTKTYFPFHDLSHGSAQVKGHGKKV
ADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLISHCLLVTLAHLPAEFTPASLDKFLASVST
VLTSKYR</sequence>
<eCutOff>10.0</eCutOff>
<searchTool>blast</searchTool>
</orgPdbQuery>
```

So, finding structures in the PDB is doable, but what if we want to go deeper and DO something with them?

- BioPython (conda install biopython) provides lots of functionality, including some for working with the PDB:

```
from Bio.PDB.PDBParser import PDBParser
from Bio.PDB import PDBList
p = PDBParser(PERMISSIVE = 1) #Ignore many common problems
pdb1 = PDBList()
structure_id = '1HXW'
filename = pdb1.retrieve_pdb_file(structure_id, file_format='pdb')
structure = p.get_structure(structure_id, filename)

firstmodel = structure[0]
chain_A = firstmodel['A']
res10 = chain_A[10]
print(res10.get_resname())
```

BioPython has lots of useful functionality for working with the contents of PDBs

```
a.get_name()          # atom name (spaces stripped, e.g. "CA")
a.get_id()            # id (equals atom name)
a.get_coord()         # atomic coordinates
a.get_bfactor()       # B factor
a.get_occupancy()     # occupancy
a.get_altloc()        # alternative location specific
a.get_sigatm()        # std. dev. of atomic parameters
a.get_siguij()        # std. dev. of anisotropic B factor
a.get_anisou()         # anisotropic B factor
a.get_fullname()      # atom name (with spaces, e.g. ".CA.")
```

For example we can print the coordinates of all CA atoms with B factor greater than 10

```
for model in structure.get_list():
    for chain in model.get_list():
        for residue in chain.get_list():
            if residue.has_id("CA"):
                ca=residue["CA"]
                if ca.get_bfactor()>10.0:
                    print(ca.get_coord())
```

Lots of other stuff is also available in BioPython

- Handles common file formats and can extract contents into Python
 - Interfacing with databases, searching, handling formats: BLAST, FASTA, GenBank, PubMed, Medline, SCOP, ...
 - Easy handling of sequences -- comparisons, translation, complementation
 - Working with sequence alignments
 - Clustering/data classification tools
 - Lots of other stuff...
 - BUT if you want to mutate residues in a structure, you need a MODELING tool, e.g. Modeller, Rosetta, etc.

<http://biopython.org/DIST/docs/tutorial/Tutorial.pdf>



Search docs

Installation

Help! How do I get started?

MDTraj Examples

What's New?

Frequently Asked Questions

IPython Notebook Viewer

Discussion Forums

Trajectories

Atom Selection DSL

Analysis Functions

File Objects

OpenMM Reporters

MDTraj Utils

Command-line trajectory conversion:

`mdconvert`

MDTraj HDF5 Format Specification

Building the documentation

MDTraj Programming Style

Docs » MDTraj

Edit on GitHub

MDTraj

Read, write and analyze MD trajectories with only a few lines of Python code.

MDTraj is a python library that allows users to manipulate [molecular dynamics \(MD\)](#) trajectories. Extensive [trajectory analysis](#) routines are implemented. With MDTraj, you can

- Read and write from **every MD format imaginable** (`pdb`, `xtc`, `trr`, `dcd`, `binpos`, `netcdf`, `mcrd`, `prmtop`, ...)
- Run **blazingly fast** RMSD calculations (4x the speed of the original [Theobald QCP](#)).
- Use tons of [analysis](#) functions like bonds/angles/dihedrals, hydrogen bonding identification, secondary structure assignment, NMR observables.
- **Lightweight API**, with a focus on **speed** and vectorized operations.

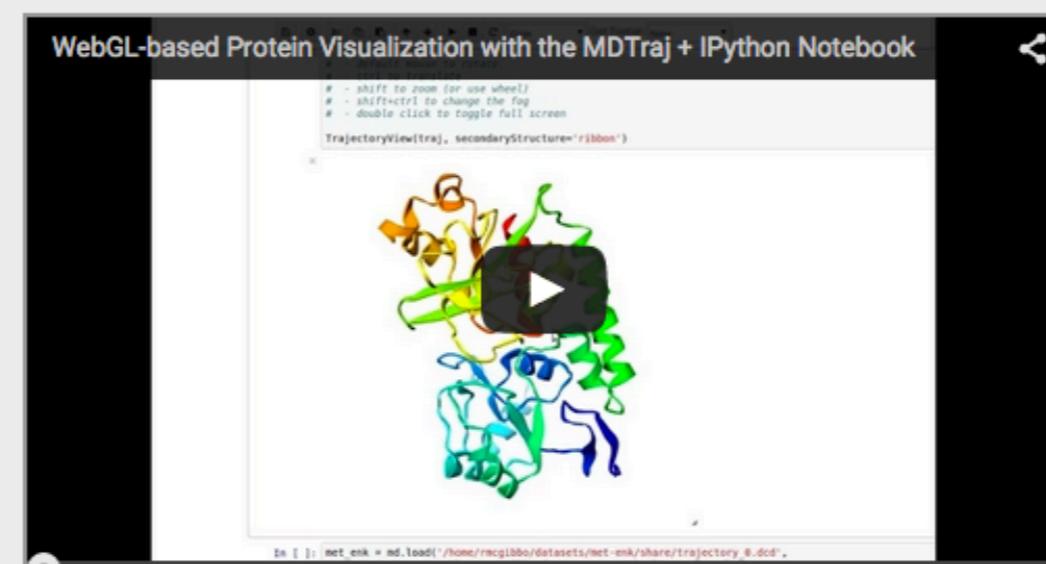
The library also ships with a flexible command-line application for converting trajectories between formats. When you install MDTraj, the script will be installed under the name `mdconvert`.

[Download the Code](#)

[See it in Action](#)

[Get Help](#)

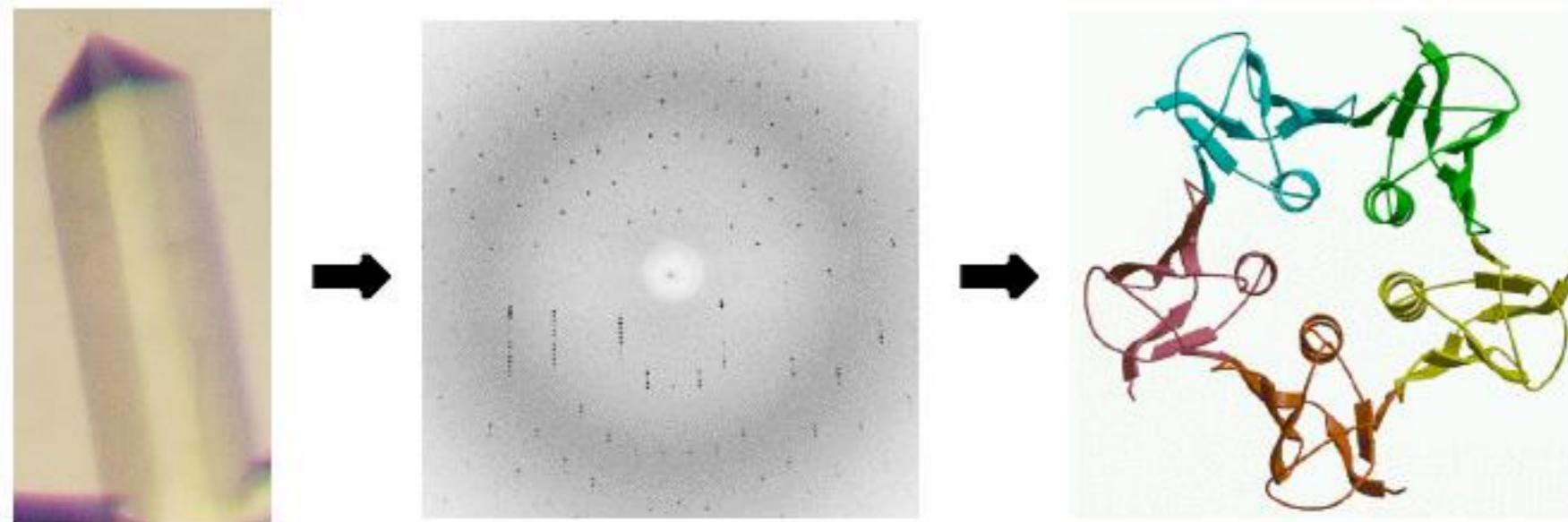
[Get Involved](#)



OEBio is OpenEye's toolkit for working with proteins, etc.

- Basics for working with biopolymers, including structure and sequence alignments, etc.
- See also Spruce for protein prep
- <https://docs.eyesopen.com/toolkits/python/ochemtk/index.html#oebio-theory>

Crystallography tries to turn diffraction data into structures



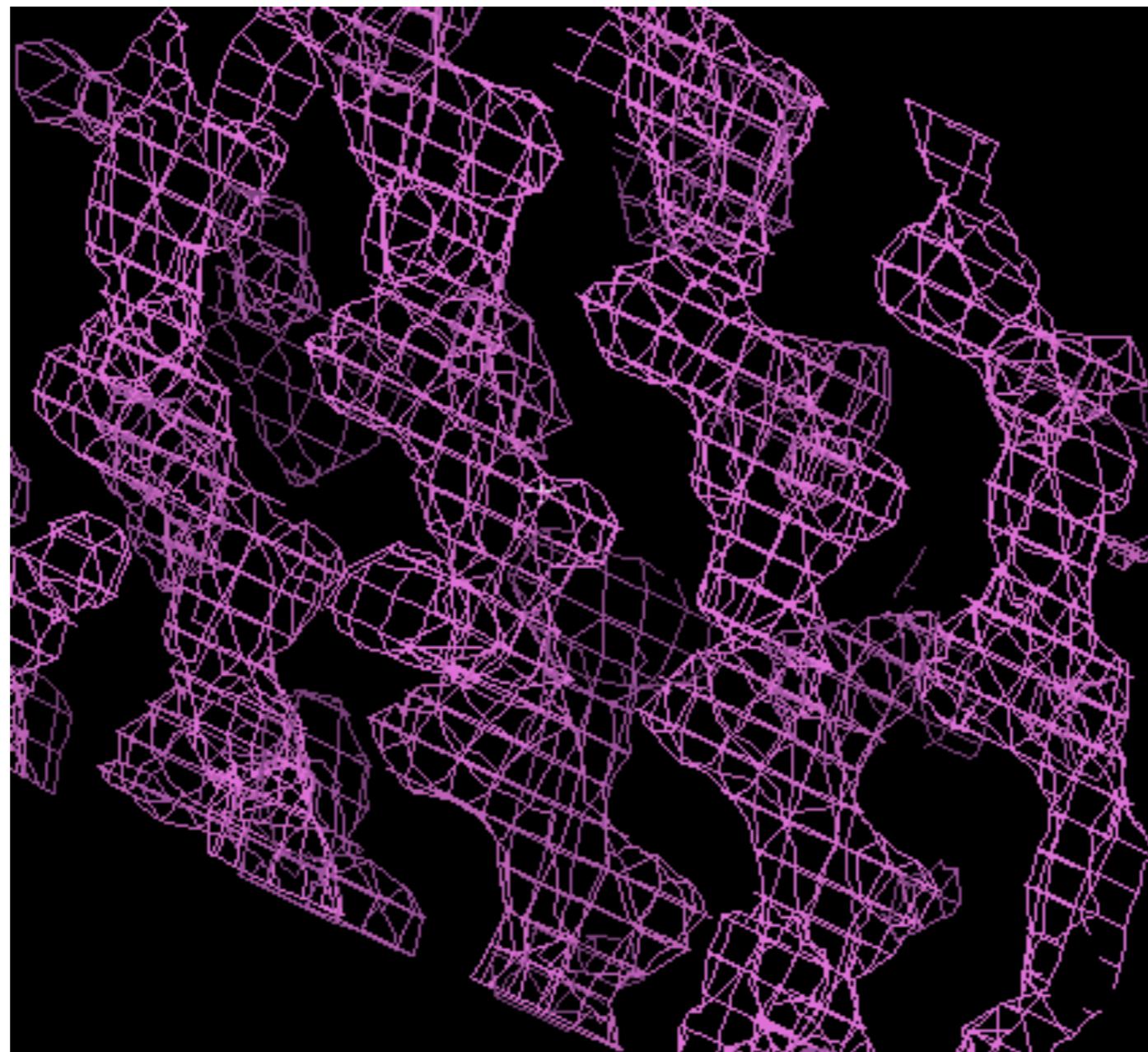
Really, there is not quite enough data -- each atom has about 3-4 parameters, and we have about one observation per parameter

This means we fit a model to the data

The data actually collected is a diffraction pattern (or set of them) which gives electron scattering amplitudes

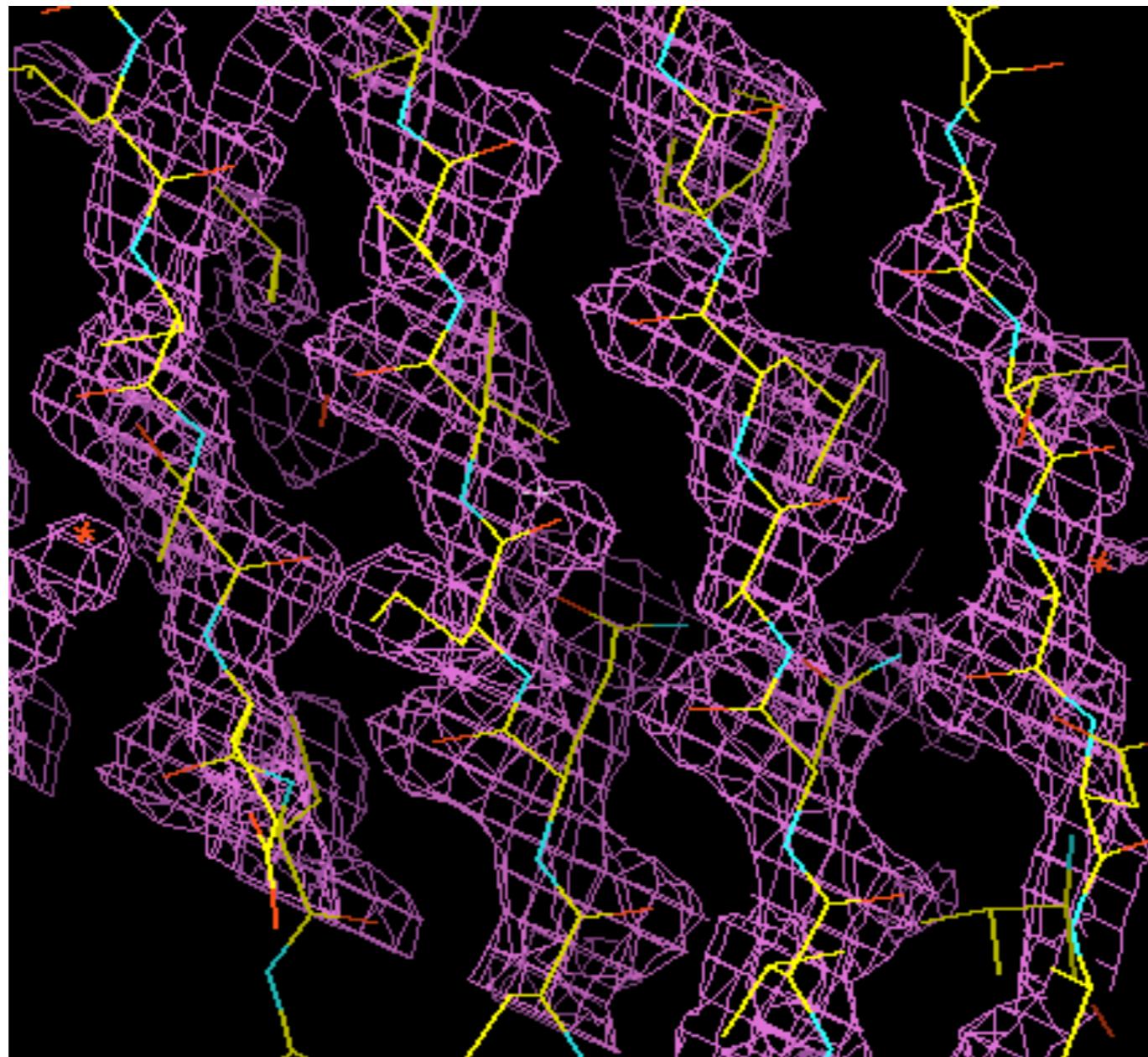
- The refinement process
 - Takes scattering amplitudes
 - Turns it into the electron density everywhere
 - Fits a model to the electron density
- There is a "phase problem" to get from the diffraction pattern to the electron density

But at some point, we have to fit a model
to the "data" -- the electron density



It is nonobvious where the atoms should be -- and this is good data

The actual model IS consistent with this data, but the model is not the data



This makes it a lot more obvious what's here

This all results in a model

The model is not the data

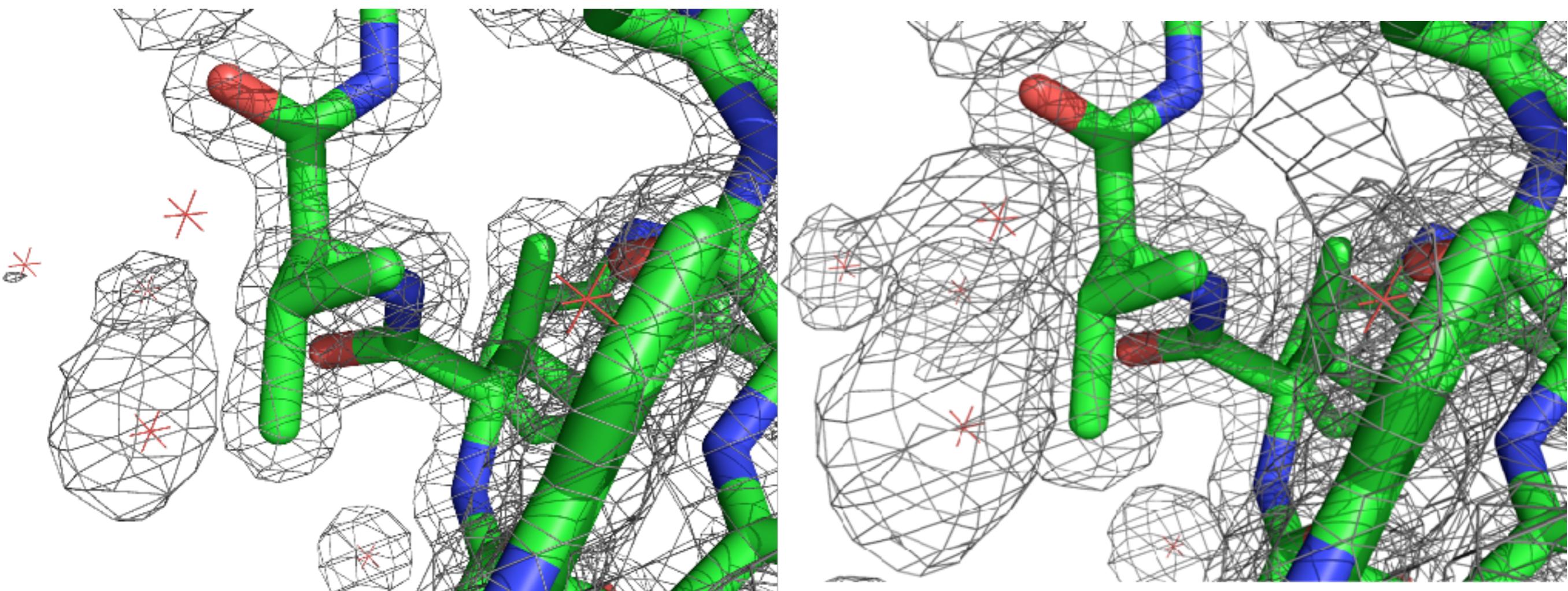
The electron density is (closer to) the
data

So, one should visualize the electron
density to check the model

There are two common ways of visualizing the electron density

- "Difference map" $F_o - F_c$
 - Shows difference between calculated and observed structure factors
- The map itself, usually as $2F_o - F_c = 2|F_o| - |F_c|$
 - Shows the actual observations, with the difference from the calculated values also appearing to a limited extent

But, there is a value for the electron density everywhere, so we have to select contour levels



PDB files don't actually contain electron density, though modern ones contain calculated structure factors

- Calculated structure factors yield the electron density
- Usually, see the Electron Density Server to get electron densities



Welcome to the Electron Density Server at Uppsala University

Enter a PDB code (4 characters):

Or enter a search string:

<http://eds.bmc.uu.se/eds/>

From EDS, you choose a type of map and a format

Electron-density map generation for 1w2i

Map format : O Type : 2mFo-DFc Generate map

(Note: this may take a few seconds, or many minutes, depending on the size of your map.)

Let's do a difference map for visualization in PyMol

Electron-density map generation for 1w2i

Map format : CCP4 Type : mFo-DFc Generate map

(Note: this may take a few seconds, or many minutes, depending on the size of your map.)

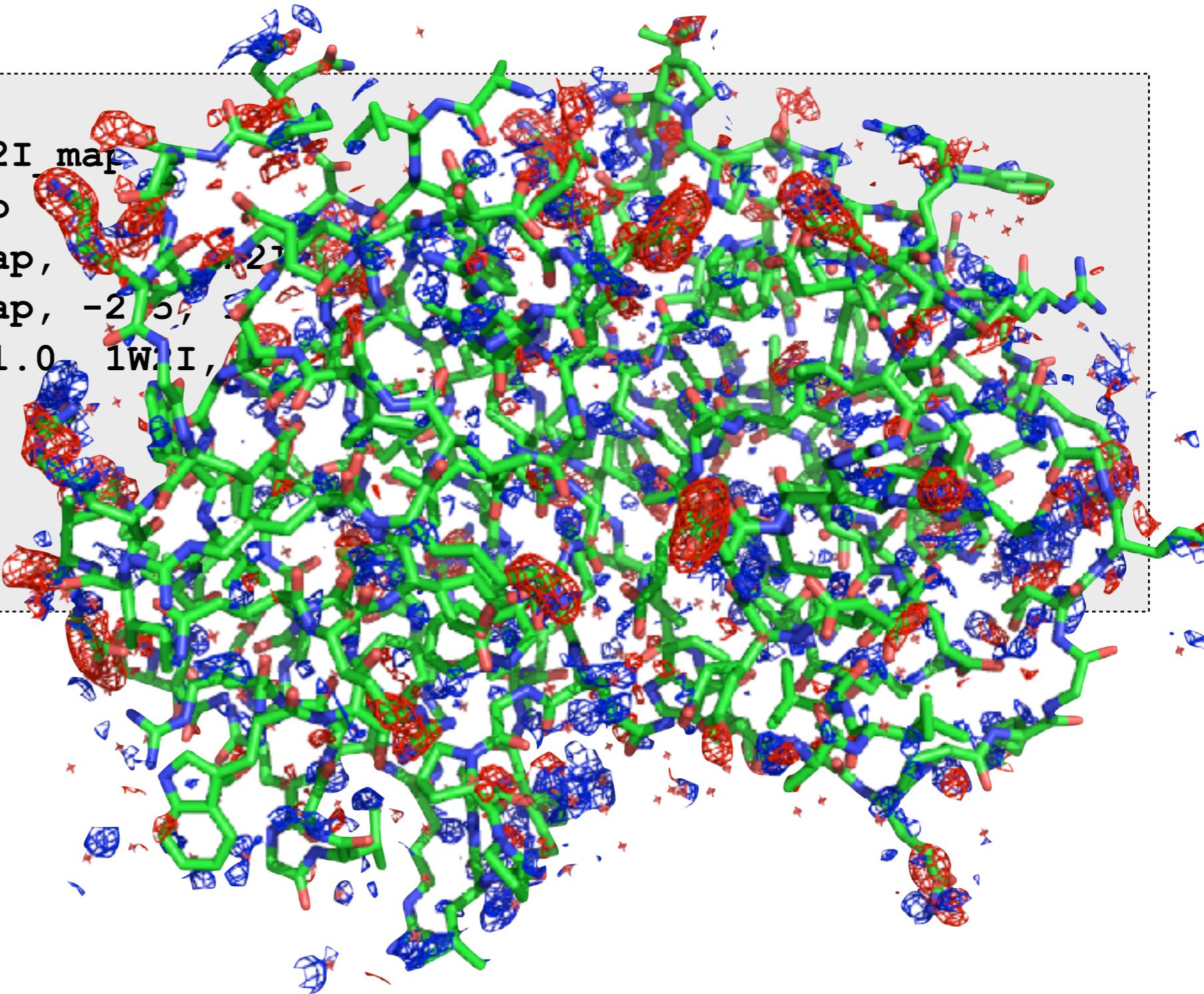
Next, I download the map to my computer, extract it, and get ready to load into PyMol

Here is your gzipped map : [1w2i_diff ccp4.gz](#)

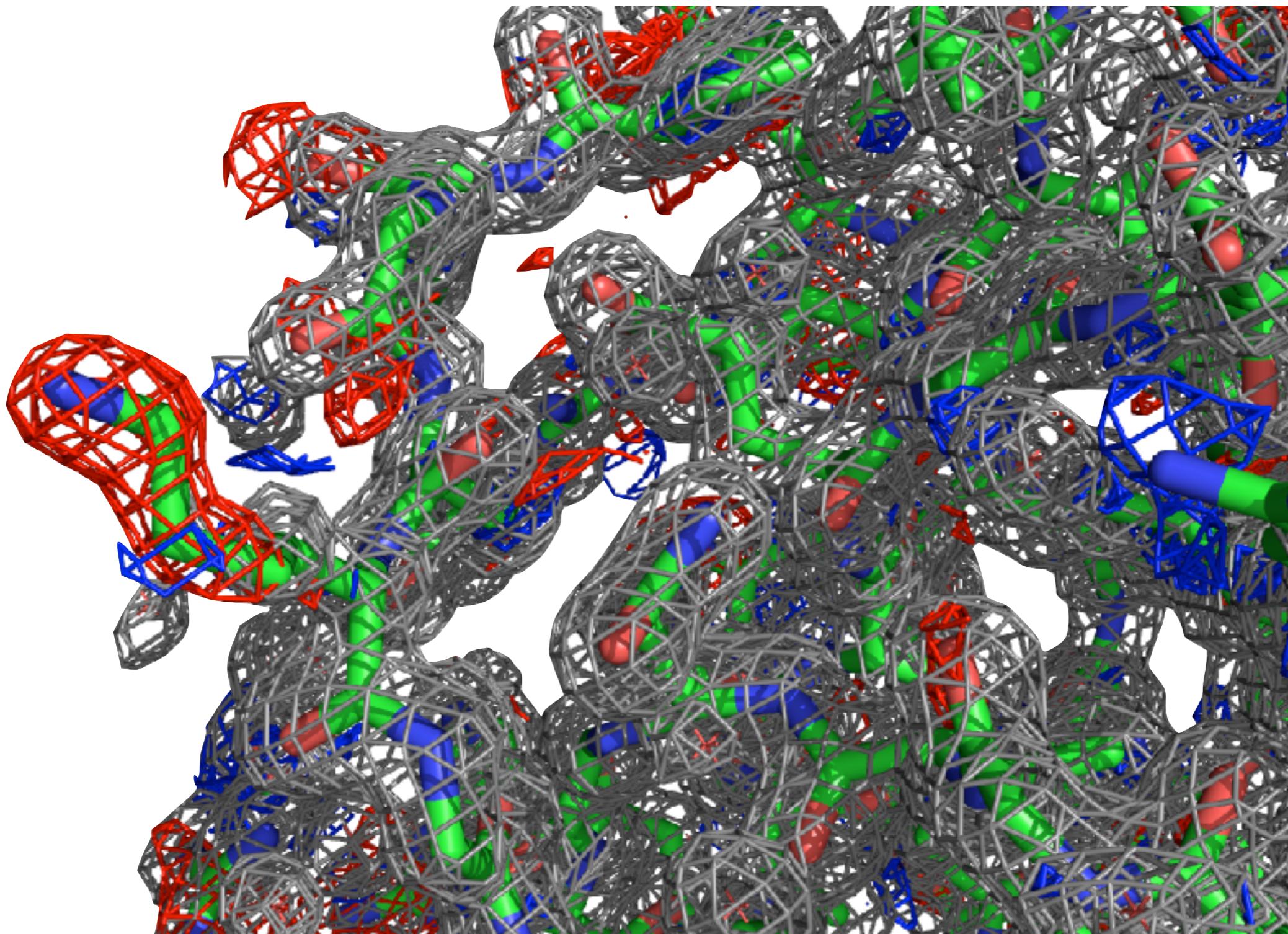
- Check out this PyMol tutorial:
 - http://pymolwiki.org/index.php/Display_CCP4_Maps
- Note: Difference maps usually contoured at +/- 2.5 sigma

In PyMol:

```
load 1W2I.pdb
load 1w2i_diff ccp4, 1W2I_map
load 1w2i ccp4, 1W2I_2fo
isomesh map_pos, 1W2I_map, -2.0, 2.0
isomesh map_neg, 1W2I_map, -2.0, 2.0
isomesh map, 1W2I_2fo, 1.0
color gray, map
color blue, map_pos
color red, map_neg
show sticks
```



Let's focus on an area with large differences and look at the actual density there as well



In some sense, the structure is not the result of crystallography

- The data: The diffraction pattern
- The results: The electron density
- The author's interpretation: The structure

To sum up:

You want to look at the electron density to help
assess the model

PyMol and other viewers can help you do so