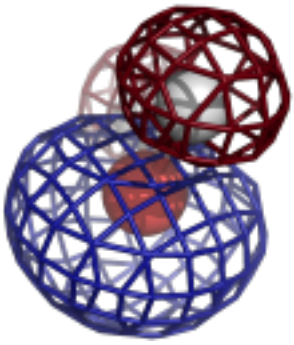
VOLUME TEN



COMPUTATIONAL CRYSTALLOGRAPHY NEWSLETTER

JANUARY MMXIX

MARCO, MAP-TO-MODEL V2

Table of Contents

Editor
Nigel W. Moriarty, NWMoriarty@LBL.Gov

Phenix News

Announcements

Workshop at the meeting of the American Crystallographic Association, Northern Kentucky Convention Center, Saturday, July 20, 2019

A workshop will be held at the next ACA annual meeting in Kentucky. The title of the all-day program – "Introduction to PHENIX for Electron Cryo-Microscopists" – indicates the target audience. Updates to schedules and the cost are available from the ACA homepage. The course is limited to 50 participants so book early.

Expert advice

Fitting Tip #17 – Asn and Gln are remarkably different

Jane Richardson, David Richardson and Christopher Williams, Duke University

Expectations of similarity

With the same amide functional group and only one carbon difference in sidechain length, Asn and Gln are usually considered one of the most similar pairs of amino acids. Looking at their 2D schematic diagrams (figure 1) or at their chemical makeup seems to confirm that idea, also reinforced by classic lists of conservative amino-acid replacements. But if one looks at what they each can or can't

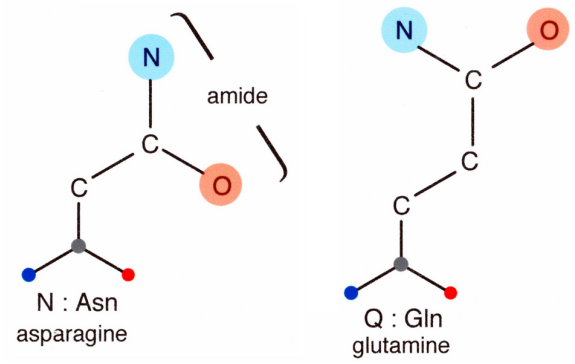


Figure 1: Schematics of Asn and Gln amino acids.

do in the context of protein 3D structures, that one-bond difference makes a huge change to their capabilities and personalities.

Multi-dimensional ϕ,ψ,χ plots

Asn has only two degrees of freedom, and has both donor and acceptor groups very close to the backbone, where they can form many distinct sidechain-backbone hydrogen bonds. This leads to a number of tight and unusual clusters in the multi-dimensional **ϕ,ψ,χ** space. Glutamine has more degrees of freedom but awkward constraints from the extra tetrahedral group and can actually H-bond back to the mainchain in only a few of its possible conformations.

As one way of showing those differences, figure 2 compares a diagonal view for the most informative 3D projections of the **ϕ,ψ,χ** plots for Asn and Gln. It uses data from the Top8000 database at 70% sequence identity (from the RCSB PDB clusters), quality-filtered at both chain and residue levels, including amide flips (Hintze, 2016). There are about 54,000 Asn and 37,000 Gln residues.

Each panel of figure 2 shows a 3D plot of **ϕ,ψ**, and **χ2** for Asn or of **ϕ,ψ**, and **χ3** for Gln, as divided down the vertical columns by the three **m,p,t** bins of **χ1** for Asn or of **χ2** for Gln. The viewpoint is rotated about 45° left from a pure Ramachandran-plot **ϕ,ψ** projection, to enable spotting 4D clusters vs spreads of the terminal amide orientations. Colored stars mark positions of local structure motifs discussed here.

From figure 2, the simplest overall observation is that the datapoint distribution for Asn is much more diverse and complex than that for Gln. The main reason is that the Gln amides are farther from the backbone so their orientations can spread freely across more of their range. The second overall observation is that datapoints are quite dense within 90° of zero and absent or sparse within 90° of 180° (Lovell 1999). Near 180° the NH2 group clashes with backbone or Cβ hydrogen atoms.

Glutamine characteristics

Especially when Gln **χ2** is trans, the **χ3** distributions are broad smears across their allowed range away from 180°. There are some exceptions where a small cluster appears at 180° offset from the central **χ3** maximum, usually seen at far left on the panels in figure 2. This data has been amide-flip corrected by MolProbity's H addition process that uses both H-bonding and clashes in the context of entire H-bond networks and seldom declares flips incorrectly (Word 1999). But it has a threshold of score difference below which it will keep the original orientation; that means there will still be some incorrect flip states remaining, which account for most points in the 180°-translated, fainter (~10%) patterns seen in figure 2. The same thing happens for Asn. After our realization that such cases are rather frequent, we plan to add a prior-probability term to the flip-correction process.

Glutamine does have some preferred patterns of amide-to-backbone H-bonds, but they nearly all occur in otherwise-favorable rotamers and so do not form tight, well-separated clusters. An especially notable Gln motif forms the helix "cap box" H-bond from the Gln OE1 to the backbone NH of the N-cap residue that starts an α-helix, as shown in figure 3 for a helix in λ repressor. This case has a Thr N-cap, although Asn is much commoner, as described below. The Gln cap-box conformation happens to work in exactly the commonest of all Gln rotamers (**mt**-30), so it just accounts for a modest part of the center of that strong, elongated cluster (hotpink * in the Gln **t** panel of figure 2).

Asparagine characteristics

Asparagine, in contrast to Gln, has many isolated clusters in unusual positions that represent distinct local structural motifs, most with specific sidechain-backbone H-bonding. Figure 4 shows the motifs for two distinct datapoint clusters of Asn sidechains on

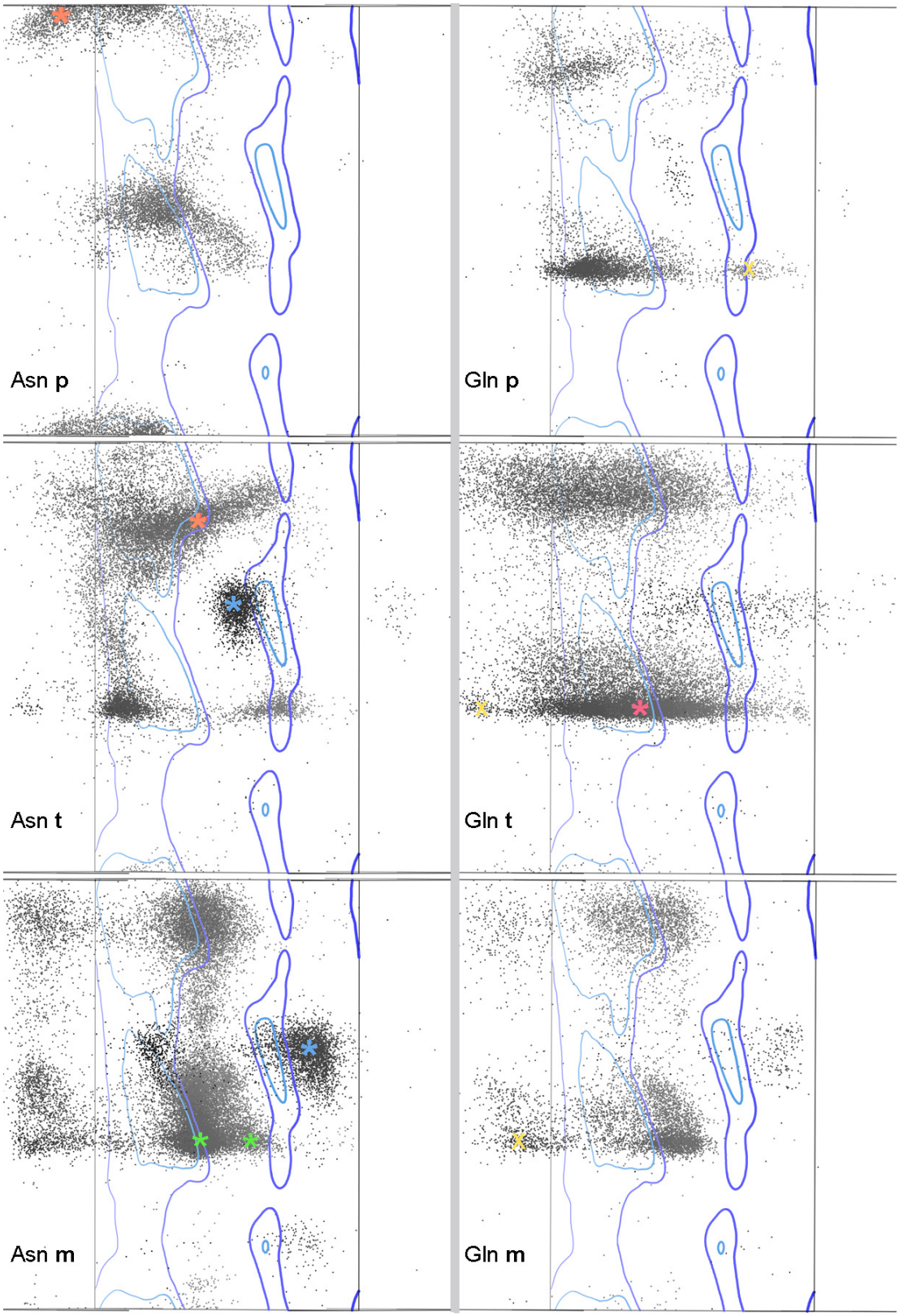


Figure 2: Diagonal views into the multidimensional ϕ,ψ,χ datapoint distributions for Asn, grouped by by $\chi 1$ bins, and for Gln, grouped by $\chi 2$ bins and labeled as **p, t, m** for ~60°, ~180°, and ~ -60°. View is turned left by 45° from straight-on ϕ,ψ, and stars mark local structural motifs discussed in the text.

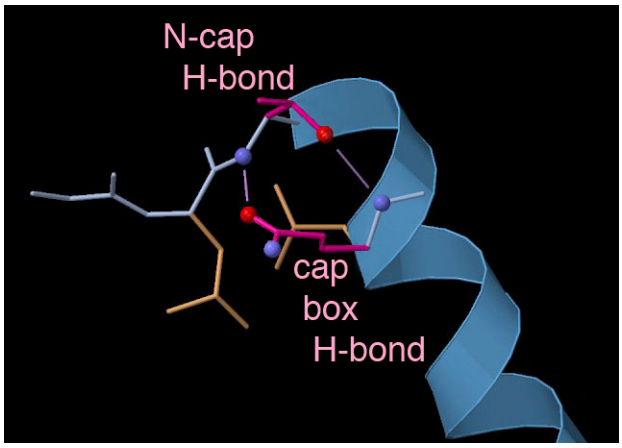


Figure 3: A helical "cap-box" Gln

regular α-helix (a relatively rare location for Asn), clearly separated in χ2 angle by an energy barrier. At left, the α**m**-80 rotamer enables the Asn NH2 to H-bond with the i-4 CO of the preceding helix turn, opening the backbone a bit, but still allowing the normal helical H-bonds as well. At right, the much more common α**m**120 rotamer places the entire sidechain in good vdW contact with the outer surface of the preceding turn of regular helix. That conformation also places the Asn OE1 in its favored position in vdW contact with the following peptide (Lovell 1999). The peaks for those rotamers are marked with green stars in the Asn **m** panel of figure 2.

Many of the local Asn motifs are enabled by the odd fact that the Asn sidechain is a very good mimic for a residue-unit of backbone, as illustrated in figure 5. At top is a short piece of extended backbone, with a sidechain (blue) going down and back. Below, on the left half, the main chain and side chain switch places; the backbone now goes down and back and the Asn sidechain (blue) mimics extended backbone with χ2 = 0° and is set up to make the previous CO H-bond equivalently (Richardson 1989). This similarity of the Asn sidechain to backbone (looking in the N-terminal direction from a given Cα) is also probably what lets Asn be the only non-Gly residue good at adopting +ϕ backbone conformations – the Cα of an Asn has two nearly identical substituents and thus is only

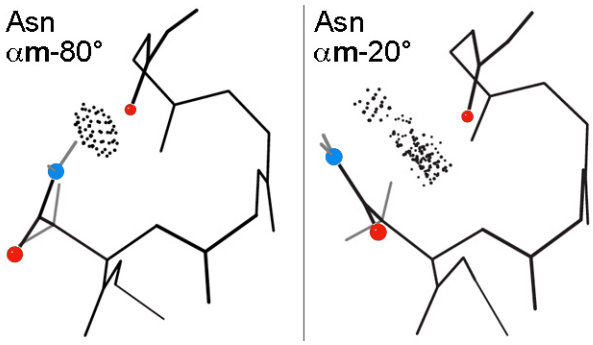


Figure 4: Two distinct conformations of Asn on α-helix

weakly asymmetric, so that normal Rα is not much better than Lα. The strong Lα peaks for Asn are marked with blue stars in the Asn **t** and **m** panels of figure 2, each with one rotamer. Asn with χ1 = **p** cannot adopt either Rα or Lα (see figure 2)

There are many local motifs in which the Asn sidechain mimics backbone. One such case is a "pseudo-turn", where the Asn takes the place of the first peptide in a tight turn, using Asn's most common local H-bond to backbone: Oδ1 to the i+2 backbone NH (Richardson 1981).

An important second case of such mimicry is at helix N-caps, where Asn is especially good at competing with backbone for its usual i+4 helical H-bond. As shown in figure 6, there are two possible rotamers that can make that

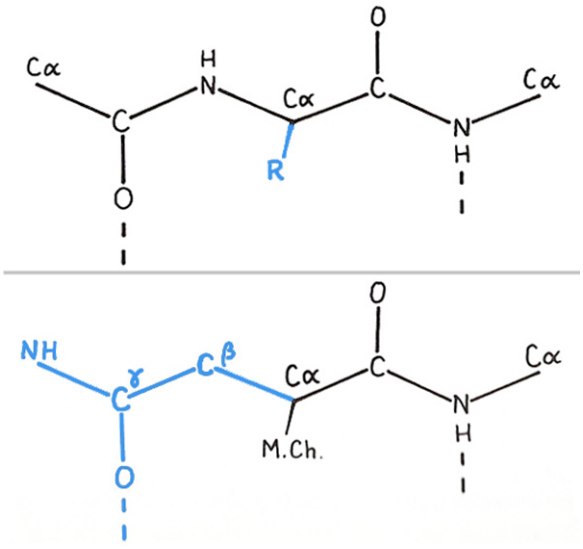


Figure 5: How an Asn sidechain mimics backbone.

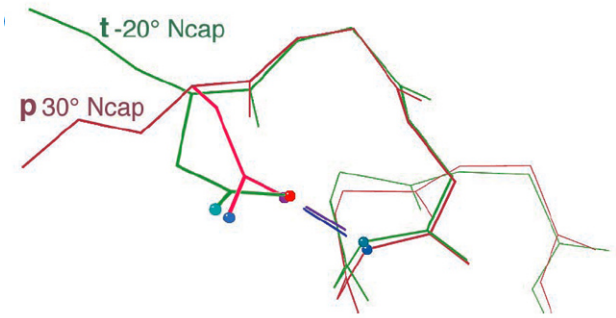


Figure 6: Two versions of Asn N-cap on α-helix.

H-bond, but **p**30 is the more common, since it is a closer geometric mimic and puts the displaced backbone in β rather than polyproline II conformation. Asn is not only the commonest N-cap residue, it is by a large factor the most specific, very strongly preferring the N-cap position and disfavoring the surrounding N-1 or N+1 to 3 positions (Richardson 1988). The sequence placement of an Asn, then, exerts a strong influence on where a helix starts, and on the direction from which the chain enters.

Yet another Asn backbone-mimic motif is to provide one more H-bond (sidechain-backbone) past β-sheet backbone H-bonding between two β-strands, either parallel or antiparallel. Usually only one such H-bond is formed, but figure 7 shows a case where the Asn amide forms two H-bonds to the opposite β-strand and a third to a separate part of the chain.

The bottom line

Glutamine is rather a "plain vanilla" sidechain, with Ramachandran plot and positional preference closest to the average of all residues. Asparagine, in contrast, has very distinct and opinionated conformational possibilities, both because it has H-bond donors and acceptors close to the backbone and because it can mimic a backbone residue.

When modeling Asn or Gln residues into a density map or evaluating them later, if you

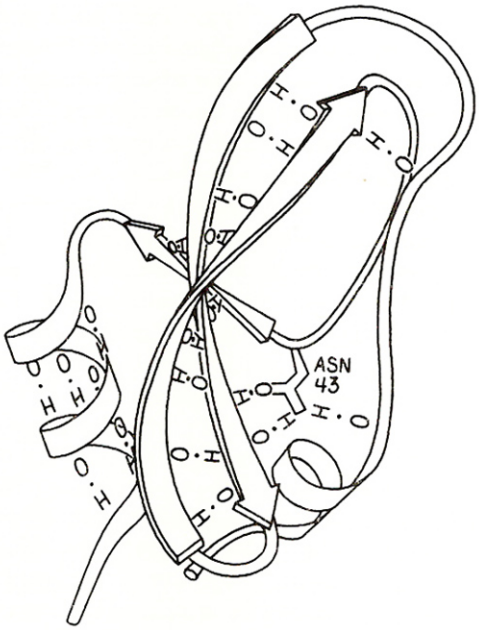


Figure 6: An extra Asn H-bond past the end of β-sheet.

have assigned a conformation with the final χ angle closer to 180° than to 0°, try the flipped alternative. If the orientation closer to 0° looks at least nearly as good by other criteria, then use that (more probable) alternative.

When modeling Asn sidechains, look for approximation to one of its distinctive local motifs such as N-caps, pseudo-turns, Lα backbone, interactions with the previous helix turn, H-bonding across the end of a β-strand pair, etc. If your Asn and its neighborhood are close to the arrangement of a typical Asn motif, try restraining the appropriate rotamer and H-bonds. In loops, look for any plausible sidechain-backbone or sidechain-sidechain H-bonding opportunities accessible with small changes.

If thinking about mutations or evolutionary relationships, don't consider Asn and Gln as conservative replacements for each other unless you know their function is either complete solvent exposure or very long-range amide H-bonding. Asn is more often the best replacement for a Gly than for a Gln.

References:

Richardson JS, Richardson DC (1988) Amino Acid Preferences for Specific Locations at the Ends of Alpha Helices, *Science* **240**: 1648-1652

Richardson JS, Richardson DC (1989) Principles and Patterns of Protein Conformation, chapter 1 in Prediction of Protein Structure and the Principles of Protein Conformation, ed. G. Fasman, Plenum Press, 1-98

Lovell SC, J.M. Word, Richardson JS, Richardson DC (1999) Asparagine and Glutamine Rotamers: B -Factor Cutoff and Correction of Amide Flips Yield Distinct Clustering, *Proc. Natl. Acad. Sci. USA*, **96**: 400-405

Word JM, Lovell SC, Richardson JS, Richardson DC (1999) Asparagine and Glutamine: Using Hydrogen Atom Contacts in the Choice of Side-chain Amide Orientation, *J Mol Biol*, **285**: 1735-1747

Hintze BJ, Lewis SM, Richardson JS, Richardson DC (2016) MolProbity's ultimate rotamer-library distributions for model validation, *Proteins: Struc Func Bioinf* **84**: 1177-1189

FAQ

Are the defaults the best for refinements?

Of course, the answer is no. The defaults have been chosen to provide the best results in the shortest time for the majority of models and data.

One of the first options to change is `optimize_xyz_weight`. Setting this option to true will optimize the weight used between the geometry and data terms of the refinement target function. Details can be found in Afonine et al., 2011.

One can also increase the number of refinement macros cycles using `number_of_macro_cycles` to ensure convergence. Ten is an adequate number.

References:

Afonine, P. V., Echols, N., Grosse-Kunstleve, R. W., Moriarty, N. W. & Adams, P. D. (2011). *Comput. Crystallogr. Newsl.* **2**, 99–103.

MARCO: The Machine Recognition of Crystallization Outcomes

Andrew E. Bruno[a], Patrick Charbonneau[b], Janet Newman[c], Edward H. Snell[d], David R. So[e], Vincent Vanhoucke[e], Christopher J. Watkins[f], Shawn Williams[g] and Julie Wilson[h]

[a]*Center for Computational Research, University at Buffalo, Buffalo, New York, USA*
[b]*Department of Chemistry and Department of Physics, Duke University, Durham, N. Carolina, USA*
[c]*Collaborative Crystallisation Centre, CSIRO, Parkville, Victoria, Australia*
[d]*Hauptman-Woodward Medical Research Institute and SUNY Buffalo, Department of Materials, Design, and Innovation, Buffalo, New York, USA*
[e]*Google Brain, Google Inc., Mountain View, California, USA*
[f]*IM&T Scientific Computing, CSIRO, Clayton South, Victoria, Australia*
[g]*Platform Technology and Sciences, GlaxoSmithKline Inc., Collegeville, Pennsylvania, USA*
[h]*Department of Mathematics, University of York, York, United Kingdom*

Correspondence email: vanhoucke@google.com

Introduction

Robust identification of the results from robotic crystallisation systems is vital to research in both industry and academia. Various research groups have approached the problem of crystal recognition using automated image analysis. As large, reliably annotated training can increase success rates, the largest training set of images previously compiled, comprising ~150,000 images, has been heavily used in that context. The resulting methods, however, often require time-consuming preprocessing stages, such as image segmentation and feature extraction. The machine-learning algorithms used for classification have thus far been specific to particular experimental setups and imaging systems.

The MARCO initiative

The macromolecular crystallization images collated by the Machine Recognition of Crystallization Outcomes (MARCO) consortium includes roughly half a million annotated images over different technical setups and imaging systems from five academic institutions and pharmaceutical companies (Figure 1. Images available from https://marco.ccr.buffalo.edu/). In contrast to the carefully curated datasets used previously, the MARCO dataset includes images with very different fields of view, problems with focus or illumination as well as those with dispensing errors. The scoring protocols of different institutions varied and, in order to homogenize the MARCO dataset, annotations were simplified to a four-class system: Crystals, Precipitate, Clear and Other. In collaboration with researchers from Google Brain, state-of-the-art deep learning algorithms were then applied to the MARCO dataset for classification. These algorithms employ Convolution Neural Networks (CNN), which require minimal preprocessing and are particularly suited to image analysis. Using a single model with all data sources combined, the trained CNN was able to correctly label 94.5% of the independent test images, regardless of their experimental origin. The algorithm and results are described in PloS one (https://arxiv.org/pdf/1803.10342.pdf) and an open source version of classifier is available at https://github.com/tensorflow/models/tree/master/research/marco.

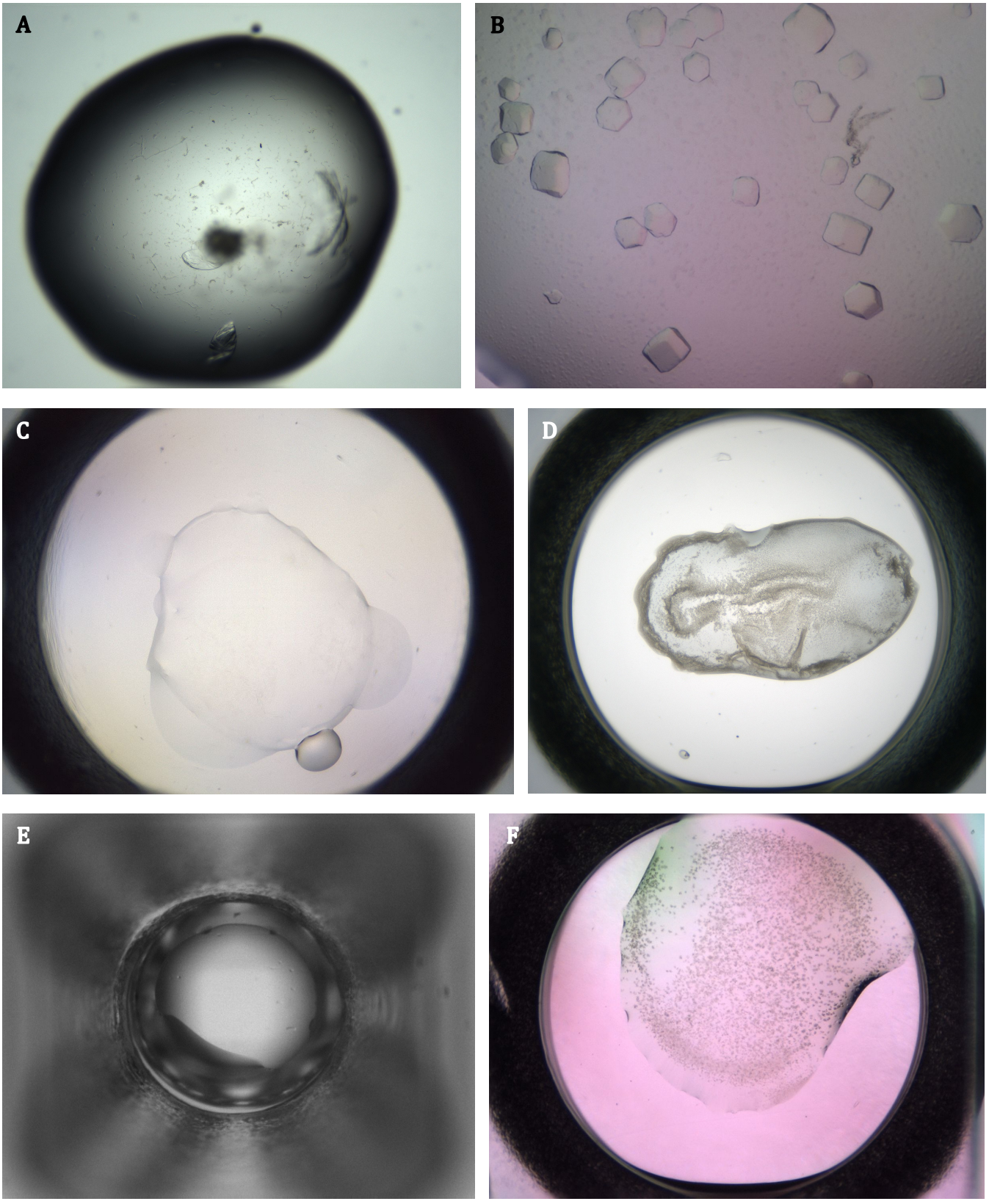


Figure 1: Images in the MARCO dataset show different experimental set-ups with various fields of view and resolutions from five industrial and academic partners: (A) Collaborative Crystallisation Centre; (B) and (F) GlaxoSmithKline; (C) Merck & Co; (D) Bristol-Myers Squibb; (E) Hauptman-Woodward Medical Research Institute.

Building a model the way you do: Map-to-model version 2

Tom Terwilliger

Los Alamos National Laboratory, Los Alamos NM 87545
New Mexico Consortium, 100 Entrada Dr, Los Alamos, NM 87544

Wouldn't it be nice if Phenix could trace the density in a cryo-EM map the way you do: find the clearest density, trace the chain at a high contour level, then dial down the contours until connections appear and trace the remainder of the chain? Seems so easy that even a computer could do it this way.

Well now it does! Version 2 of map-to-model uses a super-quick new algorithm for model-building the mimics how you would do it yourself.

Before tracing the chain, map-to-model finds helices and strands in the map. These secondary structure elements are often very accurate, so they are going to be used as fixed parts of the model to be built.

Next, map-to-model traces the chain just as you would using the new tool called trace-and-build. The trace-and-build tool chooses good density marked by the helices and strands and finds other segments of density that are very clear. Then it tries to join pairs of good segments of density by finding the highest contour level that just allows a connection. If the connection doesn't branch and isn't already used, the pair of segments is joined to make a single longer segment. This process of chain tracing is continued until no clear connections exist.

Figure 1 shows the chain tracing obtained from the small rotavirus map provided in the Phenix distribution as a model-building example. The map-to-model algorithm finds one long chain and a few short fragments. When you run map-to-model in the Phenix GUI, this chain tracing with the density and path of each chain displayed for you automatically.

Once the path of a chain is identified, a protein model is built using that path as a guide. Imagine you are building a model and you have traced the chain. What is the next thing you are going to look for? Surely you'll look for side chains marking the C_β positions. Once again map-to-model does this just the way you would. It looks for density coming off the path of the main-chain and marks likely C_β positions. Then it uses the new tool called `refine_ca_model` to find a set of C_α and C_β positions that are spaced 3.8Å apart and that match the likely C_β positions as closely as possible. At this point the helices and strands that were identified at the beginning of the procedure are spliced into the chains, creating a mosaic model and using the fixed secondary structure elements wherever they are present. With the C_α and C_β positions for a chain identified, map-to-model uses the tool Pulchra (Rotkiewicz & Skolnick, 2008) to generate an all-atom model that is refined against the map with the Phenix real-space-refine tool.

The last step in model-building is to figure out what part of the sequence in your sequence file is associated with each segment in the model. This is done in map-to-model with the new tool called `sequence_from_map`. At each position in the model, the density in the map

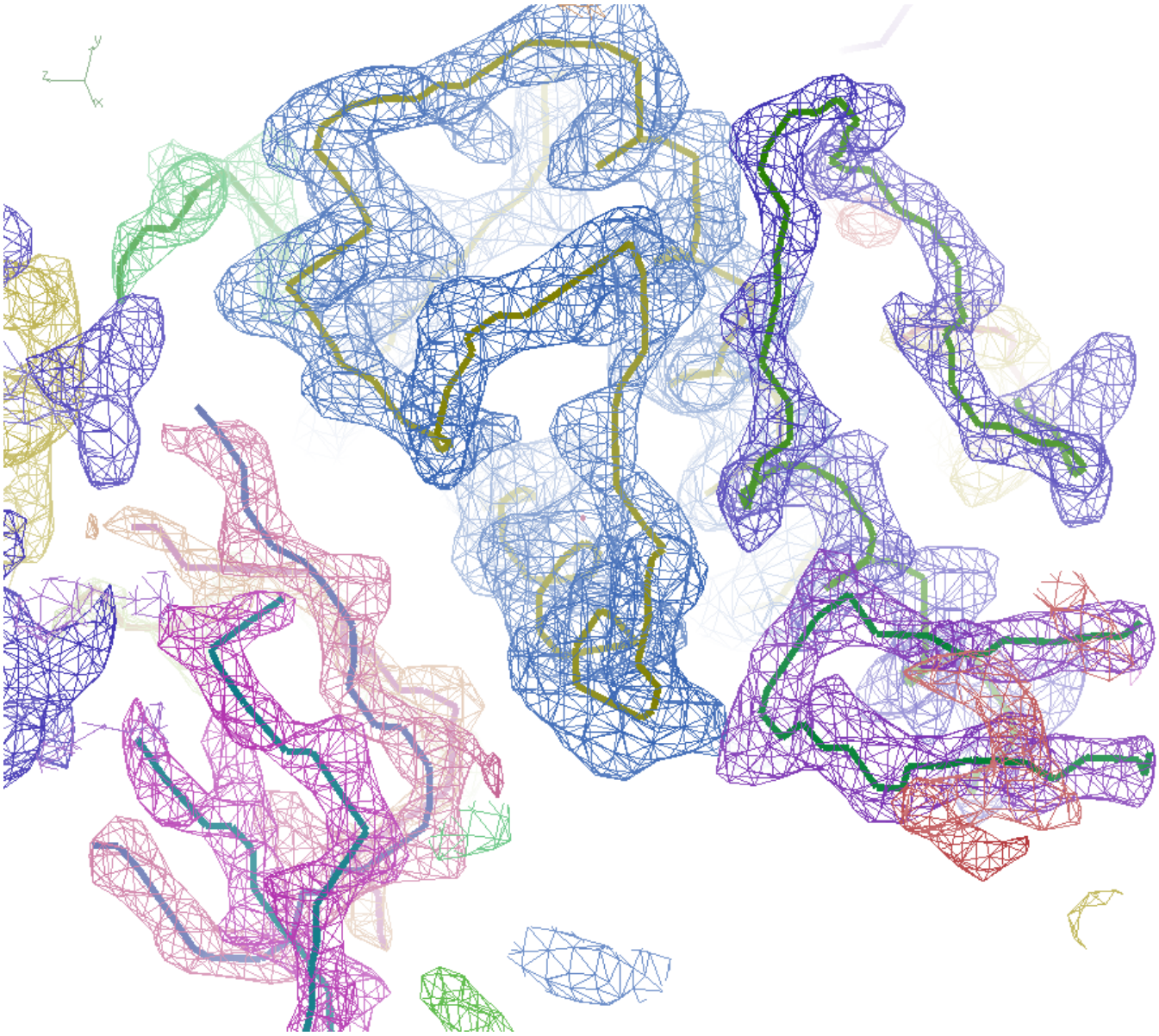


Figure 1: Example of chain tracing.

at the side-chain position is compared with expected density for each rotamer of each possible amino acid, and a relative probability for each amino acid at each position is calculated. A pseudo-sequence is then created using the most likely amino acids at each position in the model. This map-based sequence is then aligned to the supplied sequence and the best alignment is chosen and the corresponding amino acids are used at each position in the model.

Figure 2 shows part of the model created in this way for the small rotavirus map used in figure 1. The entire process takes about 5 minutes on a 4-processor machine for this small structure. The model is not perfect (it has some insertions/deletions) but it is very close to the known structure of this rotavirus protein.

Once you have built a quick model with map-to-model, you can go back and improve it. If you can see that some segments really should

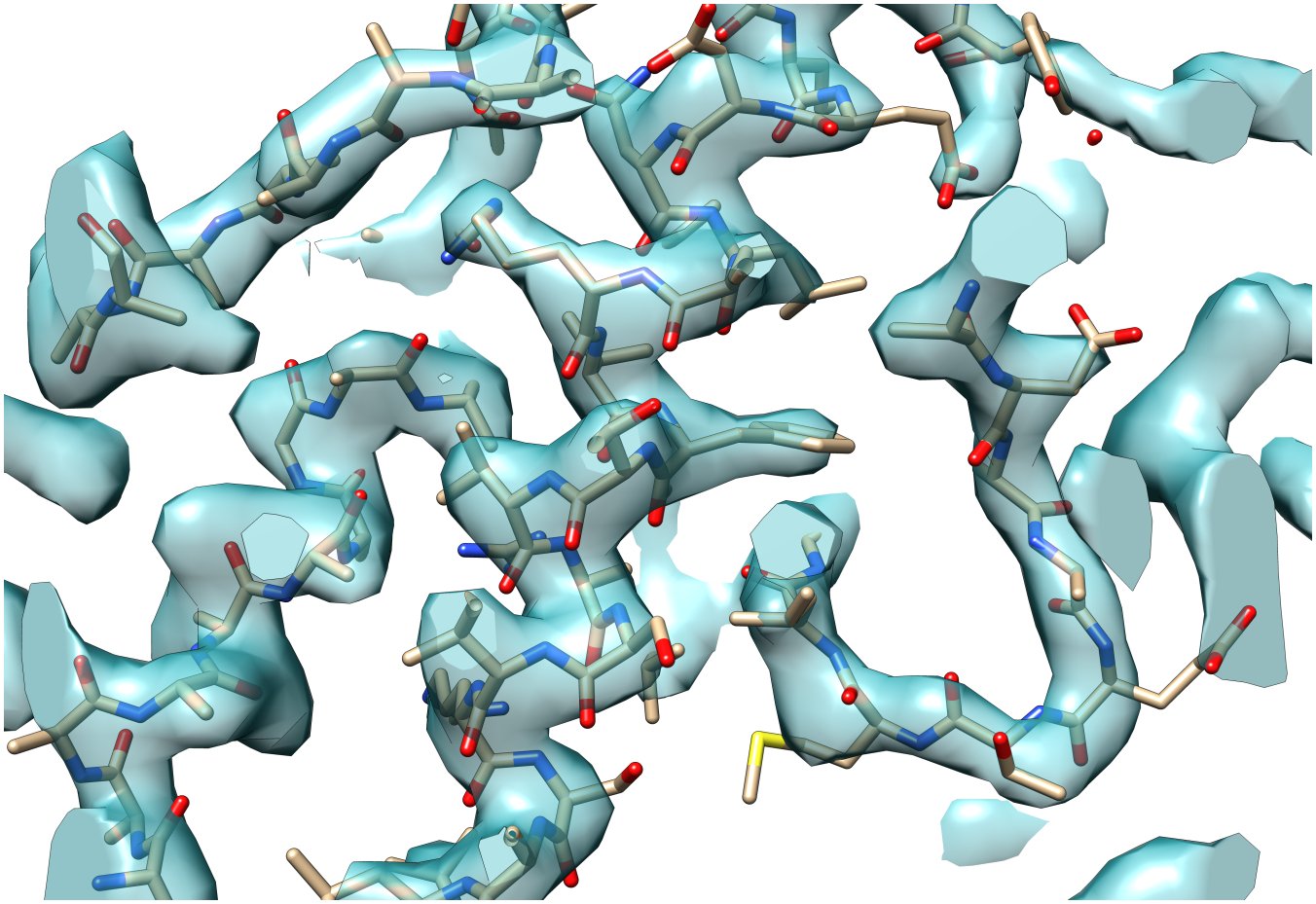


Figure 2: Model created from tracing in figure 1.

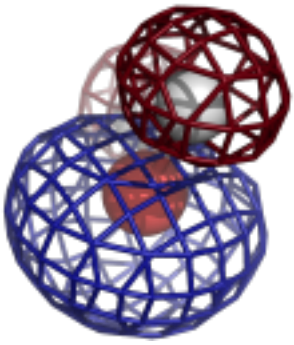
be joined, you can feed just those segments back into map-to-model and tell it to connect them. Or if you want to get rid of some of the sequence errors, you can feed your model back into map-to-model and tell it to run fix-insertions-deletions. It will use the sequence to try and identify where insertions and deletions are present and it will rebuild those segments with the appropriate number of residues.

Give the new map-to-model a try and let us know of anything that you would like improved!

Reference:

Rotkiewicz, P., J. Skolnick. J. (2008). Fast procedure for reconstruction of full-atom protein models from reduced representations. Comput Chem 29, 1460-5.

VOLUME TEN



COMPUTATIONAL CRYSTALLOGRAPHY NEWSLETTER

JULY MMXIX

BETA PEPTIDE LINKS, XFEL GUI, NUMBA, H-BONDS

Table of Contents

Editor

Nigel W. Moriarty, NWMoriarty@LBL.Gov

Phenix News

Announcements

New Phenix Release

Phenix 1.16 was released prior to the recent change in submission policy by the Protein Data Bank to only accept models solve using X-ray diffraction in the mmCIF format (Adams, P. D. *et al.*, 2019, *Acta Crystallogr. Sect. Struct. Biol.* **75**, 451–454). Changes include a new GUI designed for deposition, *mmtbx.prepare_pdb_deposition,* to create the mmCIF files for deposition into the PDB.

Also, a new tool (CLI and GUI) for getting a validation report from the PDB, *phenix.get_pdb_validation_report*.

Other changes in the addition of sequence checking to Comprehensive Validation for Cryo-EM.

One fundamental change is the inclusion of Amber functionality, by default, in the Phenix installer. This was facilitated by the move to using conda as the installation package manager. A publication is in preparation while the documentation is an ideal source of information.

A new tool, *phenix.hbond*, is available in the nightly and discussed on page 18 of this newletter.

Downloads available at phenix-online.org

Expert advice

Fitting Tip #18 – A subversive kind of misfit "water"

Jane Richardson and Christopher Williams, Duke University

It is common knowledge that a density peak fit as a crystallographic water may not actually be a water molecule. In a previous Fitting Tip (Headd & Richardson 2013) we surveyed examples of four such cases and their separable diagnoses, mostly by the atom type with which they clash: an unidentified ion, part of an unidentified ligand, the start of an unidentified alternate conformation, or a noise peak. Since then, we have documented several other clashing-water situations. Here we show the new case with the most seriously bad impact on the neighboring structure: a water fit into a peak that is really a sidechain atom.

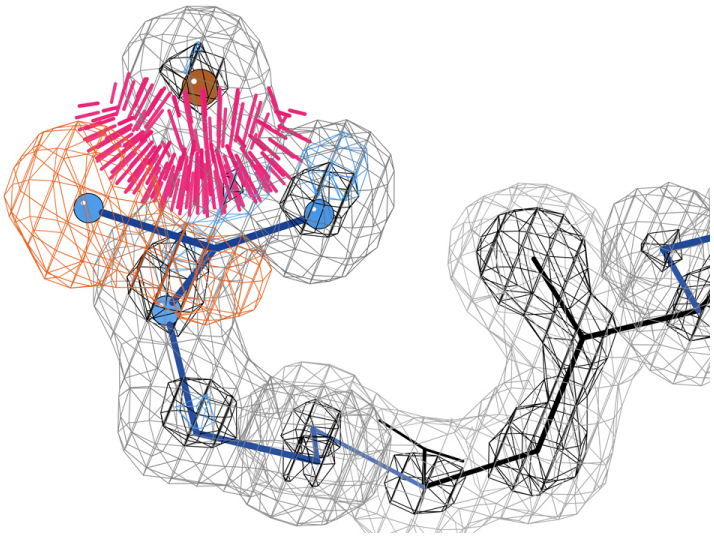


Figure 1: A water (reddish sphere) incorrectly displacing the Nh2 atom of Arg 59 in the 1qLw esterase structure (Bourne 2000). Hotpink spikes flag all-atom clashes >0.4Å, and orange contours represent difference density at -3.5σ. Gray contours are 2mFo-DFc electron density at 1.2σ and black ones at 3σ.

A sidechain can be fit incorrectly for the initial model, usually because of unclear electron density or from a molecular-replacement model with a different rotamer. That produces a difference peak for a real atom that is left outside the model.

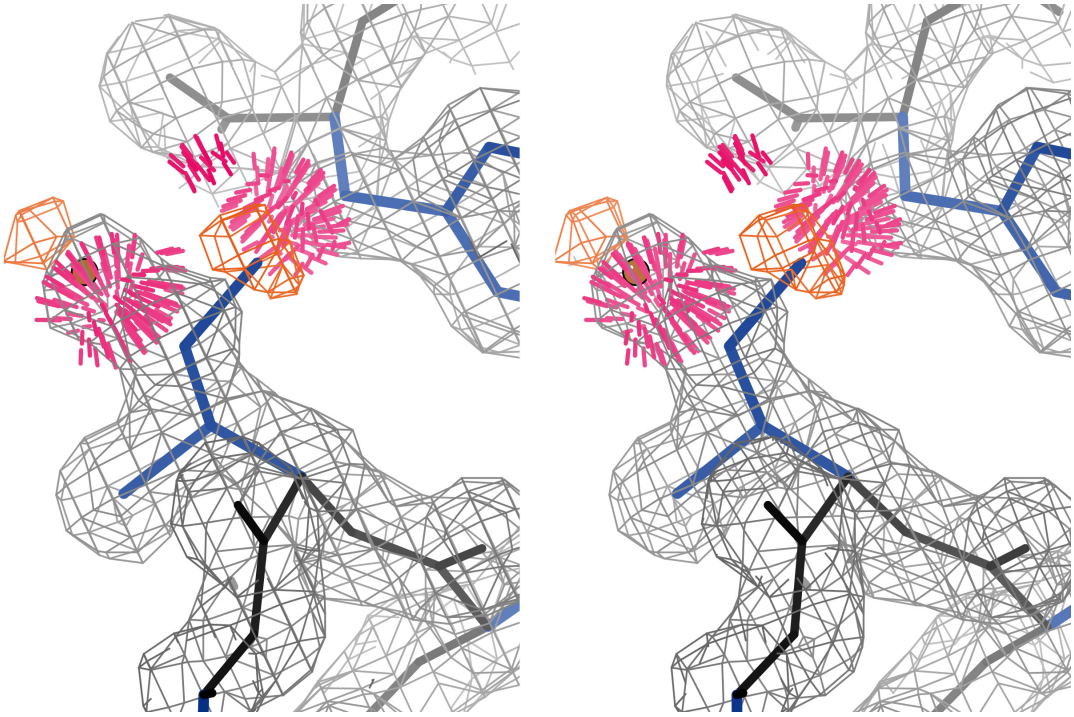


Figure 2: Stereo of a water misfit into the density for the Cd1 atom of Ile 195 in 3js8 (Sagermann 2010). The displaced Cd1 has two bad clashes with other residues. The water (reddish ball) is displaced outward in the density by its unfavorable interaction with Cg, producing another small difference peak.

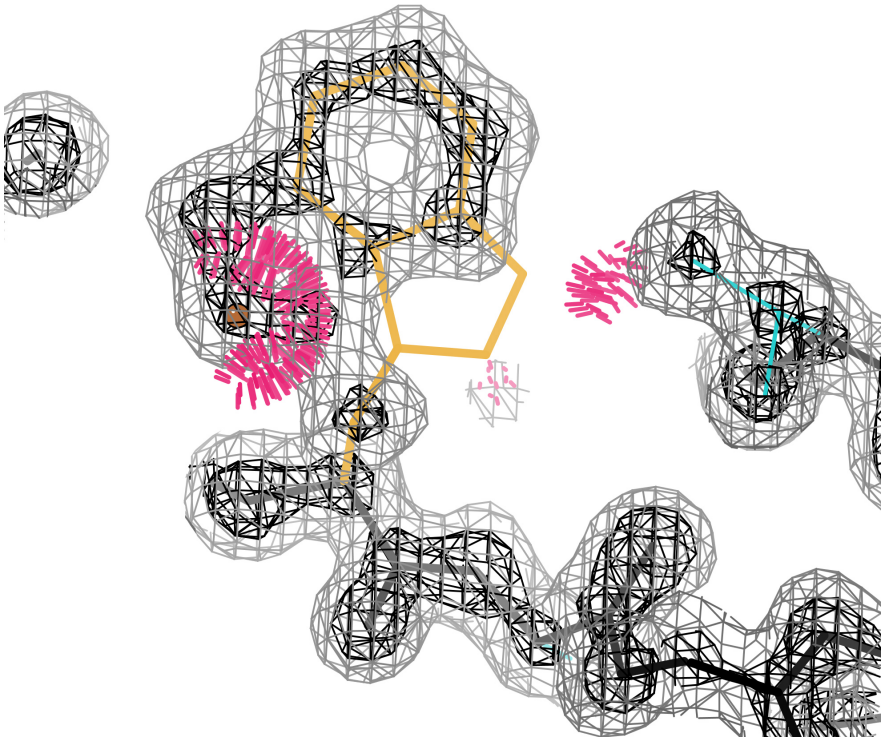


Figure 3: A water (reddish ball) trying to fill the unoccupied electron density created by fitting a Trp sidechain backwards. The water has very large clashes with four atoms of the model, which is a rotamer outlier (flagged in gold), and the incorrect atoms of the Trp also clash, with nearby sidechains. Trp B 170 in 1qw9 at 1.2Å (Hoevel 2003).

Automated or manual water picking will then often place a water in that difference peak. Refinement cannot by itself recover from this type of mistake, but an informed look at severely clashing "waters" can diagnose a correction

Arginine

Arginine, with four χ angles, is prone to an approximately but not correctly placed guanidinium group. Figure 1 shows Arg 59 in 1qLw at 1.09Å, with a water modeled into the peak that actually represents the Nh2 atom. Of course, the water has huge clashes with Ce, Nh1 and Nh2, and the misfit guanidinium produces large negative difference density on the modeled atoms. Arg 32 in 1bkr has a similar problem.

Isoleucine

For isoleucine, it is the Cd1 atom that is displaced by a water, and it usually moves into a different rotamer with Cd entirely out of density and clashing with other residues. The 3js8 cholesterol oxidase structure at 1.54Å has four such cases (Ile 195, 443, 459, and 463), each with a clash overlap >1Å between the water and the Hg12 atom. Figure 2 shows the Ile 195 example, with the large water clash, plus difference density and additional clashes for the displaced Cd1.

Tryptophan

This problem can occasionally occur even for a tryptophan fit backward and non-rotameric, where a water is placed in the density for the Cd1 atom of the sidechain's 5-membered ring. Figure 3 shows Trp 170 in chain B of the 1qLw arabinfuranosidase structure, with the rotamer-outlier sidechain (in gold) obviously backward, with only its 6-membered ring in density. The water has huge clashes with 4 atoms of the model, but it does manage to fill some of the otherwise-unoccupied density. Trp 170 is fit correctly in chain A of 1qw9, but we have twice seen this same startling pattern of a backward Trp in undeposited initial models.

This same structure also has examples of water displacing an atom in leucine and in methionine. Leu A 243 has the water in place of the Cd1 branch, pushing the sidechain aside enough to create a Cβdeviation outlier. In Met A 377 the water occupies the sulfur density of what should be the major alternate conformer. Evidently the

modeling of this structure involved early and aggressive water placement.

The bottom line

A water fit in the density of a protein atom causes especially dire consequence for the residue it displaces and clashes with. This happens infrequently, and mostly at about 2Å or higher resolution, but is important and rather easy to avoid. Use the Phenix GUI or the MolProbity multi-chart to search for bad clashes between protein atoms and modeled HOHs, and look at them, where the cases described here are blindingly obvious in Coot or kinemage graphics. Also consult Headd 2013 on other types of water problems to watch for. We are currently working on a tool that will make that process even easier by identifying water clashes and putting them into probable categories to guide their fixup.

References:

Bourne PC, Isupov MN, Littlechild JA (2000) The atomic resolution structure of a novel bacterial esterase, *Struture* **8**: 143-151 [1qLw]

Headd J, Richardson J (2013) "Fitting Tip #5: What's with water?", *Comp Cryst Newsletter* **4**: 2-5

Hoevel K, Shallom D, Niefind K, Belakhov V, Shoham G, Bassov T, Shoham Y, Schomberg D (2003) Crystal structure of a family 51 alpha-L-arabinofuranosidase in complex with 4-nitrophenyl-Ara, *Embo J* **22**: 4922-4932 [1qw9]

Sagermann M, Ohtaki A, Newton K, Doukyu N (2010) Structural characterization of the organic solvent-stable cholesterol oxidase from Chromobacterium sp. DS-1, *J Struct Biol* **170**: 32-40

FAQ

Can I submit my X-ray model to the Protein Data Bank in PDB format?

The answer is no. The PDB has moved away from PDB format in favour of the mmCIF format. Read more at Adams, P. D., Afonine, P. V., Baskaran, K., Berman, H. M., Berrisford, J., Bricogne, G., Brown, D. G., Burley, S. K., Chen, M., Feng, Z., Flensburg, C., Gutmanas, A., Hoch, J. C., Ikegawa, Y., Kengaku, Y., Krissinel, E., Kurisu, G., Liang, Y., Liebschner, D., Mak, L., Markley, J. L., Moriarty, N. W., Murshudov, G. N., Noble, M., Peisach, E., Persikova, I., Poon, B. K., Sobolev, O. V., Ulrich, E. L., Velankar, S., Vonrhein, C., Westbrook, J., Wojdyr, M., Yokochi, M. & Young, J. Y. (2019). *Acta Crystallogr. Sect. Struct. Biol.* **75**, 451–454.

Automatic β-peptide linking in Phenix

Nigel W. Moriarty

Introduction

As a general rule, amino acids polymerise using α–peptide linkages. This is the case for the standard biological amino acids – their amino groups are bonded to the Cα[1] atom (see figure 1A). For β-peptides, the amino group is bound to the Cβ atom (see figure 1B). A concise Wikipedia entry ("Beta-Peptide" 2018), discusses the details including that β-alanine (shown in figure 1B) is the only naturally occurring of the β-peptides.

On 10 June 2019, the Protein Data Bank (Burley et al. 2019) had 40 entries that contain β-alanine (3-letter code BAL) as a polymer and 13 entries as a free ligand. To automatically refine these entries, restraints for the entity are required as well as linking parameters. For the latter, links between the amino acids should ideally contain bonds, angles, dihedrals and planes as needed. For α–peptide linkages, there is one bond, four angles, three dihedral angles and two planes. A single link object can be used on each α–peptide bond with modifications for *cis* conformations. Proline (3-letter code PRO[1]) requires a different set of *cis* and *trans* links that, essentially, replacing the H hydrogen atom with the Cδ carbon atom and adjusting the values appropriately. This proline-specific link is applied to the peptide bond between the proline and the preceding amino acid.

β-peptides require two link records – one for linking to the preceding amino acid and another to link to the following peptide. Modifying the standard peptide links to accommodate the changes was done to produce the skeleton of the links. To obtain suitable values for the bond lengths and angles, a simple LBFGS-B minimization using the SciPy library (Jones, Oliphant, Peterson, et al. 2001) was performed using the highest resolution structure – 4Z0W. The 1.1Å structure contains two chains with four instances of BAL in each. The rmsZ of the link parameters was used as the target.

The resulting values are shown in table 1 and have been added to the GeoStd (Moriarty and Adams, n.d.) shipped with Phenix version 1.16. The mechanism for apply peptide links will use the appropriate link in each situation.

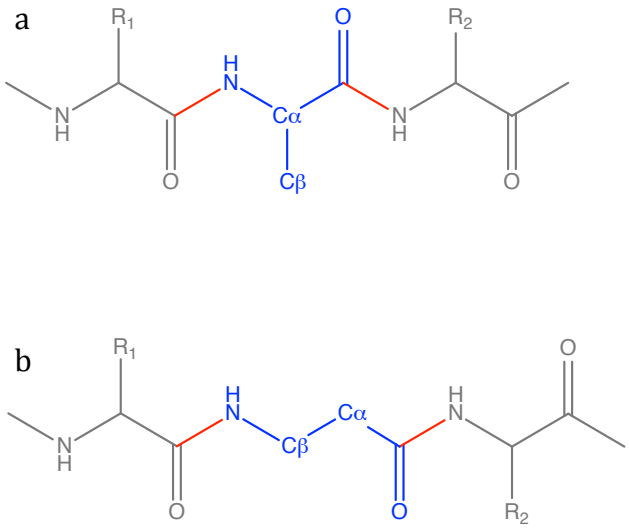


Figure 1: (a) Diagram of α-peptides. That is, the amino groups are bonded to the Cα carbon atom. The middle amino acid (blue) is linked via the red bonds to the preceding "C" carbon atom and the successive "N" nitrogen atom. (b) β-alanine (blue) polymerised with two α-peptides via similar links as in (a).

[1] Greek letters are not subscripted to aid readability and clarity.

[2] Human readable codes (Moriarty, 2016, *CCN*, 26-27) are the norm for this publication but the context makes it clear that the code for proline – PRO – does not contain a zero.

Table 1: Ideal bond lengths and bond angles for pre- and post-β-peptide links.

Pre - β	
Bond (Å)	
C–N	1.335
Angles (°)	
O–C–N	122.7
Cα–C–N	115.7
C–N–Cβ	122.7

Post - β	
Bond (Å)	
C–N	1.346
Angles (°)	
O–C–N	121.3
Cα–C–N	115.9
C–N–Cα	120.8

References

"Beta-Peptide." 2018. Wikipedia. September 6, 2018. https://en.wikipedia.org/w/index.php?title=Beta-peptide&oldid=858336632.

Burley, Stephen K., Helen M. Berman, Charmi Bhikadiya, Chunxiao Bi, Li Chen, Luigi Di Costanzo, Cole Christie, et al. 2019. "Protein Data Bank: The Single Global Archive for 3D Macromolecular Structure Data." *Nucleic Acids Research* 47 (D1): D520–28. https://doi.org/10.1093/nar/gky949.

Jones, E., T. Oliphant, P Peterson, and others. 2001. *SciPy: Open Source Scientific Tools for Python*. www.scipy.org.

Moriarty, Nigel W., and Paul D. Adams. n.d. *GeoStd*. http://sourceforge.net/projects/geostd.

phenix.hbond: a new tool for annotation hydrogen bonds

Pavel V. Afonine
Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

Hydrogen bonds (H-bonds) are non-covalent integrations that are of paramount importance to form and stabilize protein and nucleic acid structure. Secondary structure elements such as helices, sheets and interacting base pairs are held together by H-bonds. In the context of structure solution, the information about H-bonds can be used for validation and refinement. Validation typically focuses on the geometry of H-bonds, such as donor-acceptor distance and angles, as well as overall count of H bonds per structure that is expected to match prior knowledge derived from high-resolution models. Ordered solvent molecules are often validated based upon having plausible hydrogen bond interactions with the macromolecule or/and other solvent molecules. In refinement, restrains on H-bond parameters (length and angles) are particularly important at low resolution when the experimental data isn't sufficient to maintain correct secondary structure geometry (Headd *et al*., 2012). This sets the scene to introduce a new *Phenix* tool called *phenix.hbond* that is designed to annotate hydrogen bonds in atomic models. There are a number of conventions and rules that are used to identify H-bonds, for example see Torshin *et al.* (2002) and Steiner (2002). *phenix.hbond* uses geometric parameters shown in figure 1. Running *phenix.hbond* requires atomic model in PDB or mmCIF format with all hydrogen atoms added, as well as ligand restraint files if the model contains unknown to the library items. Optionally, thresholds for H-bond parameters (figure 1) can be provided that will overwrite the defaults. The program generates two output files. One is a PyMol script that can be used to visualize H-bonds as dashed lines connecting corresponding atoms that form hydrogen bond. The other file defines H-bond restraints as *restraints edits* (*Phenix* parameter file) that are suitable to use in *Phenix* refinement. Output to the log includes a list of all H bonds found that match criteria in figure 1, as well as various statistics such as histograms of H-bond lengths and angles.

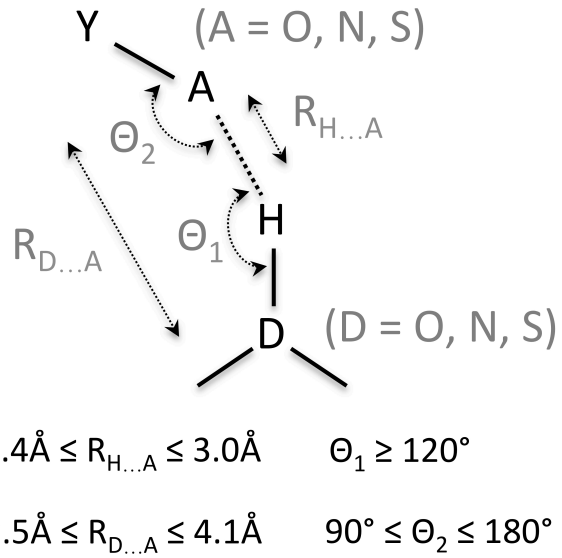


Figure 1. Hydrogen bond geometry definition used in *phenix.hbond*. $R_{D...A}$ distance is not used with default settings, but can be enabled if needed.

While there is no particular reason why this should not work for all bio-macromolecules, currently *phenix.hbond* is only optimized and tested to work with proteins, which is the limitation that will be removed in future.

Literature

Ivan Y. Torshin, Irene T. Weber, Robert W. Harrison. Protein Engineering, Design and Selection, Volume 15, Issue 5, May 2002, Pages 359–363.

Steiner, T. (2002). Angew. Chem. Int. Ed. 41. 48-76.

Headd JJ, Echols N, Afonine PV, Grosse-Kunstleve RW, Chen VB, Moriarty NW, Richardson DC, Richardson JS, Adams PD. Acta Cryst. D68, 381-390 (2012).

Bytes and Bobs : Accelerating python code with Numba.

Petrus H. Zwart[a,b]

[a] *Molecular Biophysics and Integrated Bioimaging Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720*

[b] *Center for Advanced Mathematics in Energy Research Applications, Lawrence Berkeley National Laboratory, Berkeley, CA 94720*

Correspondence email: PHZwart@lbl.gov

This is a lightweight introduction to something I encountered and found useful and interesting. Although the material presented here might be standard knowledge for some of you, it certainly wasn't for me. I provide these insights here in the hope that could be of use to some. The article below is by no means complete, exhaustive or unbiased.

Introduction

Coding in python is great, but one of the major downsides is that is can be rather slow, especially when iterating over large arrays. Have a look at the following example where we compute the sum of a large number of values in an array using a simple python for-loop (panel 1):

```
#Panel 1
import numpy as np
import time

def tst_python(x):
    result = 0
    for xx in x:
        result += xx
    return result

N=int(1e6)
x = np.random.random(int(N))
t0 = time.time()
rp = tst_python(x)
t1 = time.time()
time_python = t1-t0

print time_python, 'seconds'

>
0.206596851349 seconds
```

```
#Panel 2
from scitbx.array_family import
flex
import numpy as np
import time

def tst_flex(x):
    return flex.sum(x)

N=int(1e6)
x = np.random.random( int(N) )
x_as_flex = flex.double( x )
t0 = time.time()
rp = tst_flex(x_as_flex)
t1 = time.time()
time_flex = t1-t0

print time_flex, 'seconds'

>
0.000848054885864 seconds
```

Although a 0.20 second seems decent enough, the cctbx build-in methods available from the scitbx.array_family speed this up dramatically (panel 2).

We see that there is a speedup of a factor of 250 over the plain python code. As the reader might recall, this is accomplished in the following way:

1. Writing a dedicated C++ function that performs the numerical operation, a summation in this case.
2. Writing a C++ wrapper using the boost–python tools that exposes this functionality to python.
3. Recompiling a portion of the CCTBX library.

Although these steps are by no means hard, they can be daunting and cumbersome, especially when you haven't done this for a while. The boost-python mechanism has been the driving force in providing algorithms at acceptable speeds within the CCTBX and PHENIX software frameworks [1], and in the hand of seasoned CCTBX and PHENIX developers, is a marvelous tool to provide code at the highest performance levels.

Numba

The main drawback of boost-python however, especially for the casual, frustrated, or time-constrained developer [2] is the need to dive back into C++ to get stuff done. An alternative approach is however available: the *numba* toolkit. Numba is a just-in-time compiler that translates "python functions into optimized machine code at runtime using the industry standard LLVM compiler" [3].

Although I am sure that the computer science behind the LLVM compiler and its python interface is fascinating (see for instance [4,5]), it is more productive to focus on how to use it and what to expect.

The use of numba is relatively straightforward. By adding a specific numba decorator (*@numba.jit*) to a function, an optimized function is compiled at runtime that is almost just as fast as compiled C++ code. An illustrative example is provided in panel 3.

As you can see, the first function call upon execution is about a factor of 4 slower as compared to native python code. The second identical function call within the same python script (with a fresh set of random numbers) runs in 0.0015 seconds. This is only a factor 2 slower than the optimized C++ code, and is similar to a `numpy.sum()` function call (data not shown).

The bulk of the time upon first execution is spend in the compilation of the numba-decorated python code. At the second function call, this compilation is no longer needed resulting in a very nice performance.

Note that the keyword 'cache=True' can reduce some of the compilation time required at the first function call: only 0.1 seconds was need for the initial compilation when the script was re-executed. Note that the compilation time is independent of the argument provided to the function: if the first function call to `tst_numba` is executed on an array with length 3, the timings are the same (data not shown).

The runtimes are summarized in Figure 1 for all 5 cases.

Outlook

Numba supports numpy data types, which makes it very easy to use. Debugging numba functions is relatively straightforward, the documentation and examples on the numba website are fairly instructive. A possible drawback of numba is that it not yet supports all object-oriented features of python, forcing one to write a separate, dedicated numba functions for numerical task that can be called from within a class.

As the toy examples indicate, a C++ implementation seems to get the fastest code possible, albeit at the cost of having to deal with boost python. The use of numba allows one to code in native python, using numpy objects, but without a potential boost-python struggle. If the numba function coded up is called repeatedly, such as a target function and its derivatives in a minimizer, initial compilation costs are small price to pay to strike a balance between run-time efficiency and developer time.

Besides the illustrated accelerations on basic python code, numba also features GPU support for CUDA systems and for AMD ROC GPUS [3].

Numba can be installed with pip (including the required compiler) thus:

```
cctbx.python -m pip install numba
```

or via conda.

```
#Panel 3
import numba
import numpy as np
import time

@numba.jit(nopython=True,cache=True)
def tst_numba(x):
    result = 0
    for xx in x:
        result += xx
    return result

N=1e6
x = np.random.random(int(N))

t0 = time.time()
rn = tst_numba(x)
t1 = time.time()
time_numba = t1-t0
print time_numba, 'seconds'

x = np.random.random(int(N))
t0 = time.time()
rn = tst_numba(x)
t1 = time.time()
time_numba = t1-t0
print time_numba, 'seconds'

> First execution
0.822153091431 seconds
0.00154995918274 seconds

> Second execution
0.108898162842 seconds
0.00156705284119 seconds
```

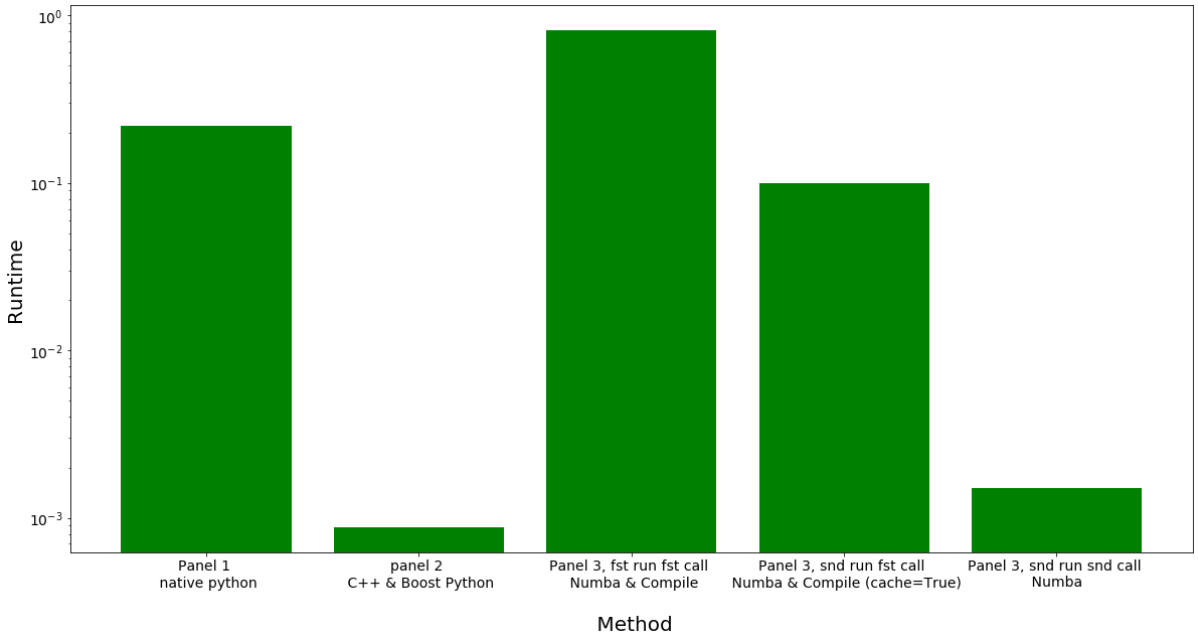
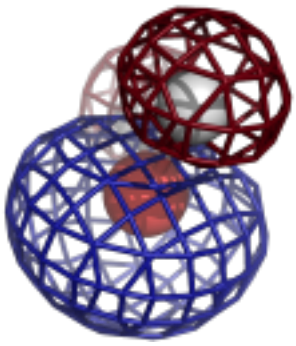


Figure 1: Runtimes for the summation of 1 million random numbers. The vertical axis is on a logarithmic scale.

References

1. https://www.boost.org/doc/libs/1_70_0/libs/python/doc/html/article.html
2. http://biosciences.lbl.gov/profiles/peter-zwart/
3. https://numba.pydata.org/index.html
4. https://llvm.org/
5. https://en.wikipedia.org/wiki/LLVM

VOLUME ELEVEN



COMPUTATIONAL CRYSTALLOGRAPHY NEWSLETTER

JANUARY MMXX

ENSEMBLE REFINEMENT

Table of Contents

Editor

Nigel W. Moriarty, NWMoriarty@LBL.Gov

Phenix News

Announcements

New Phenix Release

Highlights for the 1.17 version of Phenix include:

- Improved handling of SHELX data in *phenix.reflection_file_converter*
- *eLBOW* can output files for Amber and supports the Orca QM package
- *dials.image_viewer* is used for viewing diffraction images
- Updated map smoothing
- Fix inconsistency in clashscore values in *phenix.validation_cryoem* when hydrogen atoms are in the model

Please note that this new publication should be used to cite the use of Phenix:

Macromolecular structure determination using X-rays, neutrons and electrons: recent developments in Phenix. Liebschner D, Afonine PV, Baker ML, Bunkóczi G, Chen VB, Croll TI, Hintze B, Hung LW, Jain S, McCoy AJ, Moriarty NW, Oeffner RD, Poon BK, Prisant MG, Read RJ, Richardson JS, Richardson DC, Sammito MD, Sobolev OV, Stockwell DH, Terwilliger TC, Urzhumtsev AG, Videau LL, Williams CJ, Adams PD: Acta Cryst. (2019). D75, 861-877.

A new tool, *phenix.homology*, is available in the nightly and discussed on page 5 of this newsletter.

Downloads, documentation and changes are available at phenix-online.org

Expert advice

Fitting Tip #19 – Remember to use the information from NCS copies

Christopher Williams and Jane Richardson, Duke University

Multiple copies of the molecule in a crystal asymmetric unit or in a cryoEM 3D reconstruction are both an advantage and a disadvantage. Although NCS makes the structure very much larger, a big advantage is that if density is uninterpretable for the individual copies, you can often fit a single model to the averaged map. For less extreme cases, in Phenix you can torsionally restrain the copies to match one another, with a "top-out" function to loosen the restraints where they diverge strongly (Headd 2012). However, in our experience a large fraction of structures with NCS are built and refined as independent copies that seldom seem to have been compared with each other afterward. That final comparison step is a very powerful advantage that should not be skipped (even if top-out restrained), since comparison can separate real differences from modeling-error differences and can straightforwardly correct the errors that differ.

Individual problem areas

At high to mid resolution, differences between unsymmetrized NCS copies are most commonly for sidechain conformations or for differently disordered loops or chain ends. At low resolution (2.5-4Å), in addition one often sees significant problems in the backbone even in relatively well-ordered parts of the molecule. An example is shown in Figure 1, where each chain fits correctly the region of the other's error. At low resolution, these validation displays benefit greatly from the peptide-orientation analysis of CaBLAM, which integrates information across five residues. Traditional outliers signal problems if they are present, but their absence does not mean all is well, because the broad density is compatible with models that refine away outliers without fixing the underlying problems. Once identified, many of those problems can be fixed in Coot (Emsley 2010) with peptide-flip or crankshaft moves, such as a CaBLAM inside secondary structure, or two CaBLAM outliers in a row (where the central CO's peptide's should be rotated).

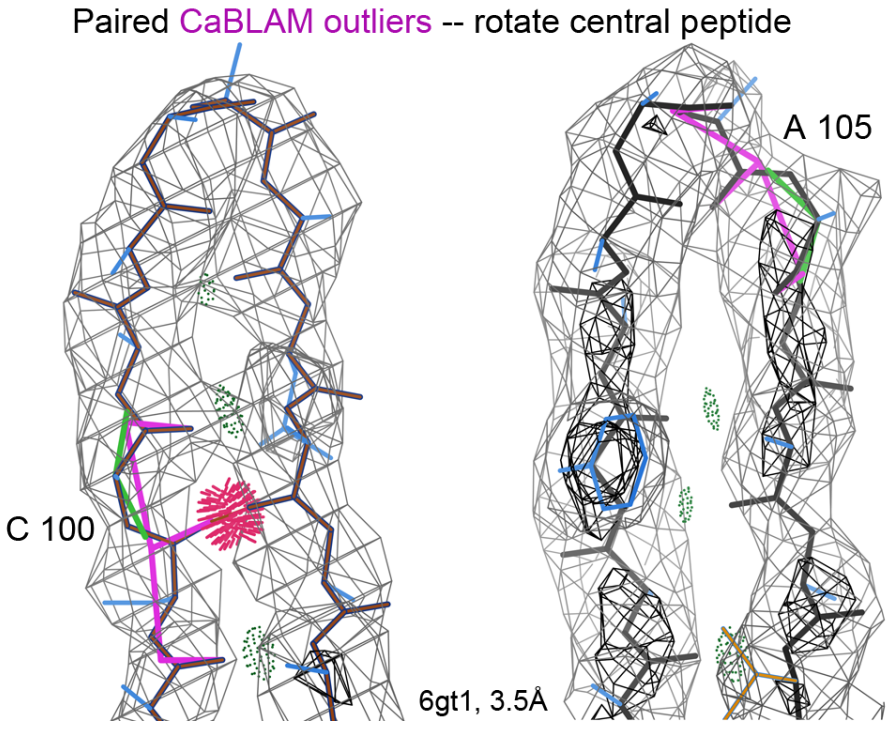


Figure 1: Models of NCS copies often have outlier problems in different places, as for this β-hairpin in chain A vs C of the 6gt1 Nek7 kinase at 3.5Å resolution (Byrne 2018). Hotpink spikes flag all-atom clashes >0.4Å, green lines are Ramachandran outliers, and the magenta dihedrals between backbone COs flag CaBLAM outliers. Gray contours are 2mFo-DFc electron density at 1.2σ and black ones at 3σ, and pillows of dark green dots are hydrogen bonds.

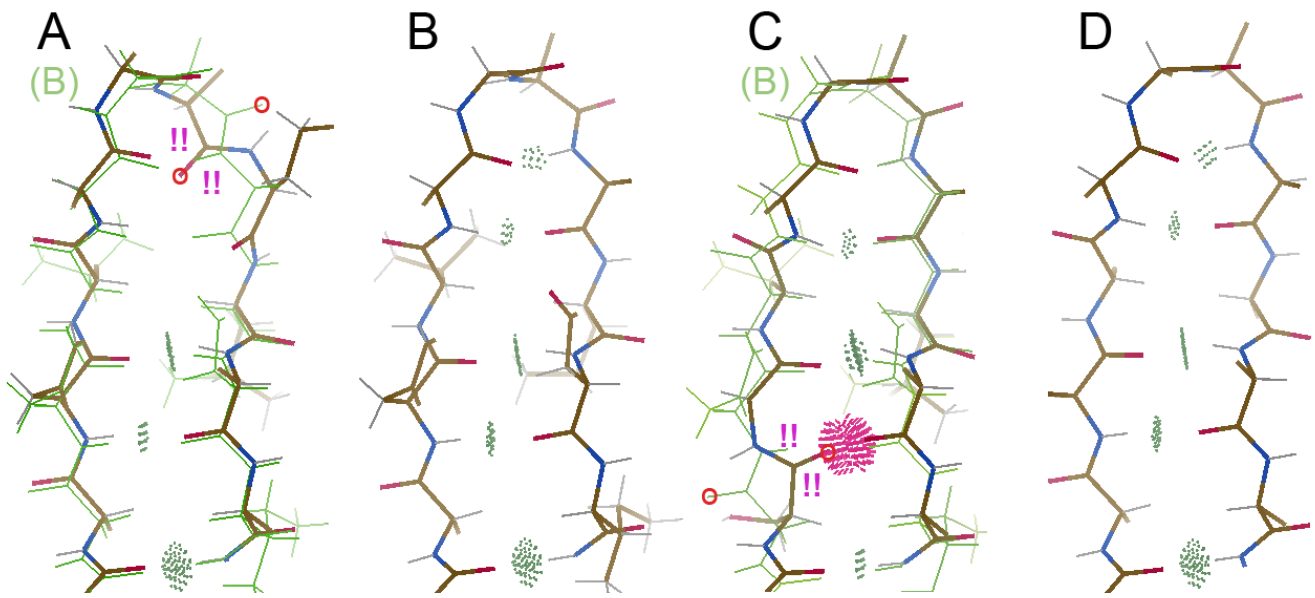


Figure 2: The same β-hairpin from 6gt1, shown for each of the 4 chains. The B chain has the best electron density and D the worst (its clean conformation was apparently copied from B). Parts A and C show a green ghost of chain B that guides how each of the errors can be fixed. The carbonyl O atoms that need rotation are marked with a red O in original and corrected positions, and the original CaBLAM outliers are represented by **!!** in magenta.

Chain C is especially easy, with two CaBLAM outliers and a huge clash flagging a classic error that happens at resolutions where bumps for the CO groups have disappeared: in a beta-strand, three COs in a row are incorrectly pointed in the same direction rather than alternating (Chen 2011). Chain A has a more complex problem at what should be a tight turn, but also rotating the central CO (105) is the first move, followed by fixing a bad-geometry adjacent group that clashes with the corrected CO and then real-space refinement of the surrounding stretch (say, 102-107). When a peptide orientation is seriously wrong, it also distorts sidechain positioning and makes sequence misalignment more likely.

[Note: KiNG (Chen 2009) is used for Figure 1, since Coot does not yet flag CaBLAM outliers. Within Phenix, the cryoEM validation GUI lists CaBLAM outliers, each with a link to center there in Coot. On the MolProbity web site (Prisant 2020), the CaBLAM outliers are included in the 3D multi-kin displayed online in the modified NGL Viewer (Rose 2015).]

Compare the NCS copies

Individual corrections are the hard way to figure this out, however. Coot not only lets you cycle through the NCS copies to find the best one, it also lets you show a "ghost" of the best one superimposed on the problem copy. That ghost both guides and validates the correction process.

In this case, locally, chain B has no outliers and a fit to its density that is very reasonable for this resolution. Chains A and C each differ from the consensus conformation in a different way, and their evidently avoidable outliers mean that those differences are surely fitting errors, not genuine differences.

The bottom line

Most of the time, the quality of your model can be significantly improved if you use the information from comparing multiple NCS copies. If resolution or density quality is really poor, it is better to fit one model to an average map. But in most cases it is better to take advantage of having an ensemble (at least a proxy for uncertainty, and sometimes for mobility). But to get the benefit from this extra information requires a step near the end where you explicitly compare the differences between models. If they differ locally with no outlier flags, then probably the local differences are real, and perhaps of interest. If all copies of a local region have problems, it's worth a try at fixing the erros, but it's hard to know what's the right answer. In the frequent and most useful case where a local region differs in both conformation and validation, then a clean copy shows how to correct a problem copy and make your overall structure better. This is an important, productive step at either high or low resolution.

References:

Byrne MJ, Cunnison RF, Bhatia C, Bayliss RW (2018) Nek7 bound to purine inhibitor, unpublished [6gt1]

Chen VB, Davis IW, Richardson DC (2009) KiNG (Kinemage, Next Generation): A versatile interactive molecular and scientific visualization program, *Protein Sci* **18**: 2403-2409

Chen V, Williams C, Richardson J (2011) "Fitting Tip #1: Not 3 parallel COs in a row", *Comp Cryst Newsletter* **2**: 3

Emsley P, Lohkamp B, Scott WG, Cowtan K (2010) Features and development of Coot, *Acta Cryst* **D66**:486-501

Headd JJ, Echols N, Afonine PA, Grosse-Kunstleve RW, Chen VB, Moriarty NW, Richardson DC, Richardson JS, Adams PD (2012) Use of knowledge-based restraints in *phenix.refine* to improve macromolecular refinement at low resolution, *Acta Cryst* **D68**: 381-390

Prisant MG, Williams CJ, Chen VB, Richardson JS, Richardson DC (2020) New tools in MolProbity validation: CaBLAM for cryoEM backbone, UnDowser to rethink "waters", and NGL Viewer to recapture online 3D graphics, *Protein Sci* **29**: 315-329 and bioRxiv 79516

Rose AS, Hildebrand PW (2015) NGL Viewer: A web application for molecular visualization, *Nucleic Acids Res* **43**: W576-W579

FAQ

Can I control the automatic linking?

When a model with poor geometry is provided to a Phenix program, the automatic linking may generate an unwanted link. This is because distance between the two entities plays a roll in the algorithm. The stop an unwanted link use

```
exclude_from_automatic_linking {
    selection_1 = None
    selection_2 = None
  }
```

to select the two entities to not create a link.

phenix.homology: finding high-resolution matches for low-resolution models at a chain level

Yanting Xu[a,b], Li-Wei Hung[b], Oleg V. Sobolev[b] and Pavel V. Afonine[b]
[a]*International Center for Quantum and Molecular Structures, Shanghai University, Shanghai 200444, China*
[b]*Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA*

Correspondence email: pafonine@LBL.Gov

With the rise of cryo-EM, low-resolution model building and refinement becames more frequent exercise. Modeling against low-resolution data is generally tedious and error-prone because the corresponding maps lack atomic and often secondary-structure level of details while containing noise thus allowing multiple interpretations. It is not uncommon, however, that the Protein Data Bank (PDB, Burley *et al*., 2019) contains an atomic model of a structure that is sequence-similar to the structure being studied obtained using a higher resolution data. In such cases the higher resolution model can be used to help low-resolution model building, refinement and validation (Headd *et al.*, 2012; van Beusekom *et al.*, 2018).

Here we present a new *Phenix* (Liebschner *et al*., 2019) tool, *phenix.homology*, that, given a low-resolution protein model or the corresponding sequence, can search the PDB for a set of highest-resolution models within a specified threshold of sequence identity. *phenix.homology* operates at a chain level: it searches for higher-resolution matches for each individual chain in the low-resolution model. *phenix.homology* makes use of BLAST sequence alignment tool (Altschul *et al.*, 1997) and *iotbx.bioinformatics* module of CCTBX (Grosse-Kunstleve *et al*., 2002).

Examples of usage scenarios:

– Given a low-resolution model (provided as model or sequence files) find *n* highest resolution models that satisfy the resolution and sequence identity criteria:

```
phenix.homology <model.pdb or sequence.txt> high_res=2 identity=95 n=3
```

– Find high-resolution matches for all low-resolution models in the PDB that satisfy specified resolution and sequence identity criteria:

```
phenix.homology low_res=3 high_res=2 identity=95 all_database=True
```

This command will iterate over all protein models of resolution 3Å or worse in the PDB. For each chain in the models, it will search for matching chains in all PDB entries of resolution 2Å or better and that have sequence identity above 95%.

There is a nuance about how sequence identity percentage is calculated depending on the presence and position of gaps in reference or matching models. Figure 1 illustrates five scenarios: (a) perfect match, (b) gap in the middle of reference model, (c,d) gap at either end of reference or matching model, (e) gap in the middle of matching model.

The parameter *piece_matching* controls how matching is done. If *piece_matching* is set to True, then leading and trailing gaps (Fig. 1c) are ignored in matching models. If *piece_matching* is set to False (the default), the leading and trailing gaps in matching models are accounted for in the matching. Gaps in the middle of the chain (Fig. 1b,e) are always accounted for regardless of the *piece_matching* setting. In case of matching chain being longer than reference chain, BLAST would only return aligned segment and report the identity (Fig 1d).

Another nuance is related to whether the sequence identity is considered for the whole model or per chain. This is governed by *chain_matching* parameter of *phenix.homology*. If *chain_matching* is set to True (default setting) then a match will be considered if at least one chain satisfies the sequence similarity criterion. If *chain_matching* is False then a match will not be considered if there is at least one chain out of the whole protein does not satisfy similarity criterion.

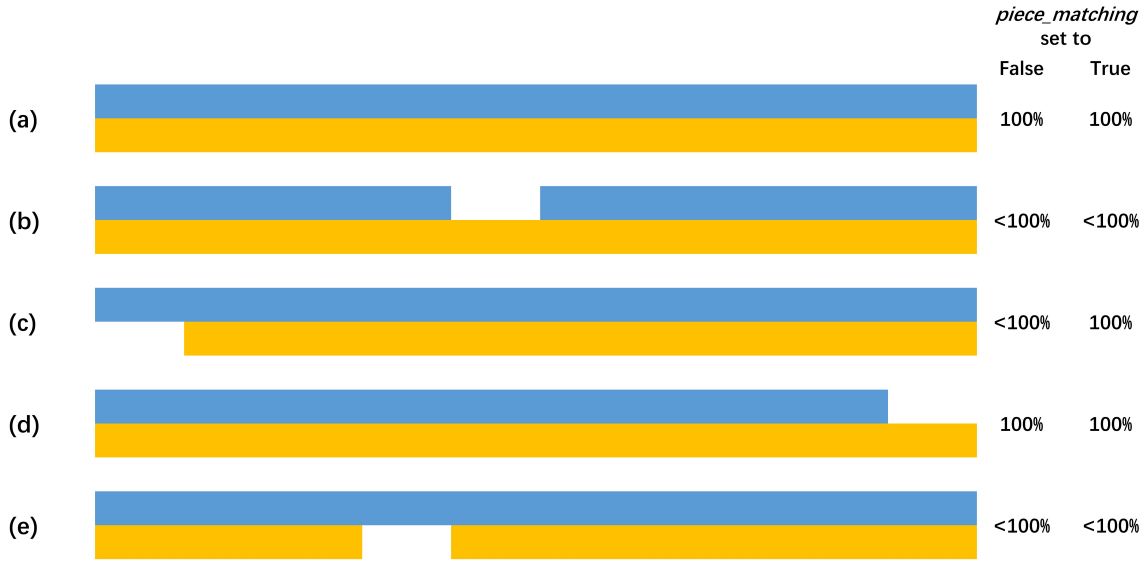


Figure 1: Example of matching scenarios: low resolution (blue), high resolution (yellow).

The tool is also accessible at the Python level:

```
from phenix.programs import homology
parameters = homology.get_default_params()
result = homology.run(sequence_string, parameters)
```

where the result contains matching high-resolution structure information: PDB code, chain ID, resolution and corresponding sequence identity.

References

Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J. H., Zhang, Z., Miller, W. & Lipman, D. J. (1997). *Nucleic Acids Res* **25**, 3389-3402.

Burley, S. K., Berman, H. M., Bhikadiya, C., Bi, C., Chen, L., Di Costanzo, L., Christie, C., Dalenberg, K., Duarte, J. M., Dutta, S., Feng, Z., Ghosh, S., Goodsell, D. S., Green, R. K., Guranović, V., Guzenko, D., Hudson, B. P., Kalro, T., Liang, Y., Lowe, R., Namkoong, H., Peisach, E., Periskova, I., Prlić, A., Randle, C., Rose, A., Rose, P., Sala, R., Sekharan, M., Shao, C., Tan, L., Tao, Y.-P., Valasatava, Y., Voigt, M., Westbrook, J., Woo, J., Yang, H., Young, J., Zhuravleva, M. & Zardecki, C. (2019). Nucleic Acids Res. 47, D464–D474.

Grosse-Kunstleve, R. W., Sauter, N. K., Moriarty, N. W. & Adams, P. D. (2002). J. Appl. Cryst. 35, 126–136.

Headd, J. J., Echols, N., Afonine, P. V., Grosse-Kunstleve, R. W., Chen, V. B., Moriarty, N. W., Richardson, D. C., Richardson, J. S. & Adams, P. D. (2012). *Acta Crystallogr D Biol Crystallogr* **68**, 381-390.

Liebschner, D., P. V. Afonine, M. L. Baker, G. Bunkóczi, V. B. Chen, T. I. Croll, B. Hintze, L.-W. Hung, S. Jain, A. J. McCoy, N. W. Moriarty, R. D. Oeffner, B. K. Poon, M. G. Prisant, R. J. Read, J. S. Richardson, D. C. Richardson, M. D. Sammito, O. V. Sobolev, D. H. Stockwell, T. C. Terwilliger, A. G. Urzhumtsev, L. L. Videau, C. J. Williams, and P. D. Adams. (2019). Acta Cryst. **D75**, 861-877

van Beusekom, B., Joosten, K., Hekkelman, M. L., Joosten, R. P. & Perrakis, A. (2018). *Iucrj* **5**, 585-594.

Lessons from using the Cambridge Structure Database: I – Bond number specification

Nigel W. Moriarty[a*]

[a]Molecular Biosciences and Integrated Bioimaging, Lawrence Berkeley National Laboratory, Berkeley, CA 94720

[*]Correspondence e-mail: nwmoriarty@lbl.gov

Preface

As a user of the Cambridge Structural Database (CSD), I had to learn to use the available interfaces. Several lessons were learned either by trial and error, or by noticing discrepancies in published works when attempting to replicate the searches. Clearly, an expert user of the CSD would be aware of these details but an example for the novice can be very useful both for teaching and as a cautionary tale.

Introduction

One of the techniques used to obtain accurate internal coordinate values of chemical entities involves the interrogation of experimentally determined small molecule structure databases such as the Cambridge Structure Database (CSD, Groom *et al.*, 2016) and the Crystallography Open Database (COD, Gražulis *et al.*, 2009). The former has several powerful interfaces including Conquest (Bruno *et al.*, 2002), a structure based search tool.

Notwithstanding the ample documentation, using these tools has a learning curve that can be challenging. Human nature also plays a role. Many are loathed to actually read the documentation preferring to jump into using the program. Conquest is a flexible and intuitive interface that searches the CSD for matches to a structure fragment that can be drawn in a window. Seems simple enough but there are pitfalls that can easily shallow the unsuspecting.

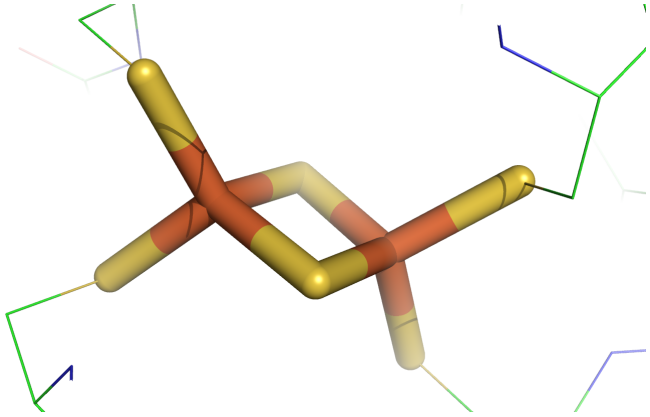
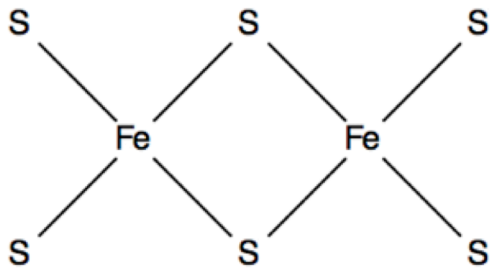


Figure 1: The Fe_2S_2 cluster (FES) from PDB entry 3wcq. The coordinated sulfur atoms from four cysteine amino acids are included in the "sticks" representation.

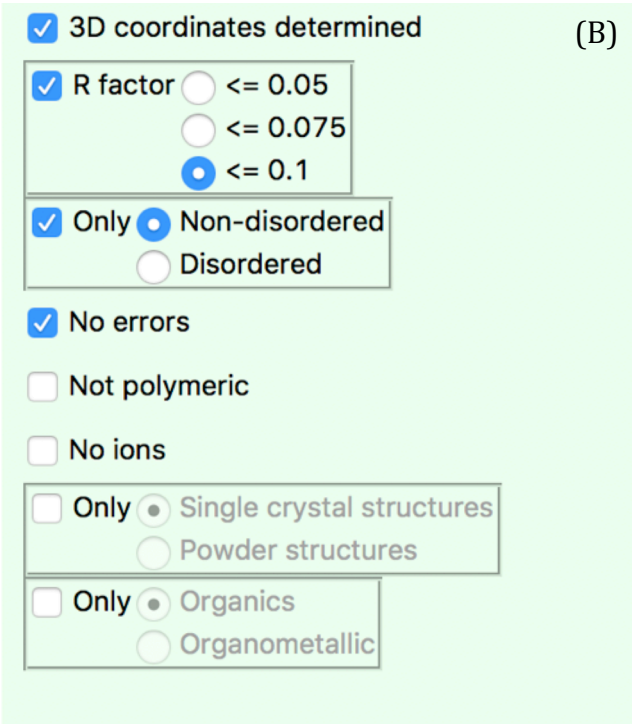
Bond number specification

The iron-sulfur cluster entry in the Chemical Component Library (CCL, Westbrook *et al.*, 2015) designated FES is a rhomboidal Fe_2S_2 entity (see figure 1) that has inaccurate geometry information in both the CCL and Monomer Library (Vagin *et al.*, 2004). This statement is based on a recent study (Moriarty & Adams, 2019) of another iron-sulfur cluster, SF4, that had similar issues. Another indicator is the high-resolution FES structure from the Protein Data Bank (PDB, Burley *et al.*, 2019) structure 3wcq shown in figure 1. This PDB entry has very different geometry values for the FES compared to the CCL and Monomer Library. Note that FES

(A)



(B)



(C)

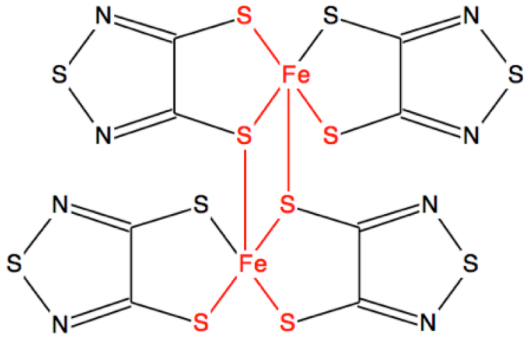


Figure 2: (A) Simple Conquest search fragment for the Fe_2S_2 cluster FES. Image taken from Conquest Draw window. (B) Conquest filter settings. (C) Example of unreasonable CSD entity with matching atoms highlighted in red.

coordinates with four cysteine sulfur atoms (included in the "sticks" representation of PyMOL (DeLano, 2002)) – two to each iron atom to form a tetra-coordinated centre. Note also that the two sulfur atoms in FES are only coordinated to the iron atoms in the FES.

Using the Conquest chemical fragment interface, a simple search can be constructed as shown in figure 2a. Searching with the filters in figure 2b results in 229 results. Verification of the results quickly shows that this search was deficient. The first result, AFAVOS, is shown in figure 2c. Focusing on the red atoms and bonds it is clear that the iron atoms are penta-coordinated and the sulfur atoms are bonded to atoms external to the group. The reason is that the number of bonds that an atom in the search fragment can be bonded to is "unspecified" – meaning any number is accepted as a hit.

One option in Conquest is to specify the exact number of bonded atoms for each atom. Once done, the search structure has T*n* where *n* is the number of bonded atoms associated with each atom as shown in figure 3. This structure returns 18 hits. Stepping though the results shows that all 18 are much closer to the FES entity topology.

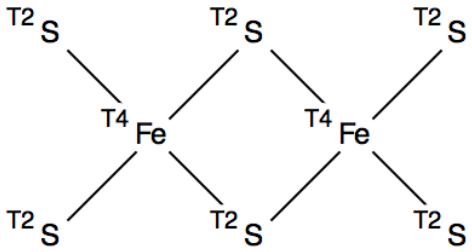


Figure 3: Conquest search fragment for the Fe_2S_2 cluster FES with bond numbers set.

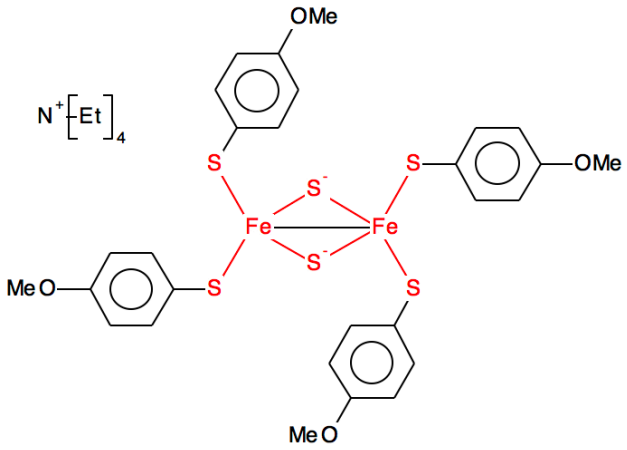


Figure 4: Search result from search fragment in figure 2A with an additional bond between the two iron atoms.

One nuance arises when considering the entry BORTOT from the first search shown in figure 4. The inclusion of the bond between the two iron atoms would appear to be an annotation of the depositor. Removing the number of bonded atoms from the iron atoms (see figure 5a) returns 24 hits. One of the hits is SIWYOM shown in figure 5b. This is clearly not an instance that can provide geometry details of FES so the stricter search is a better choice.

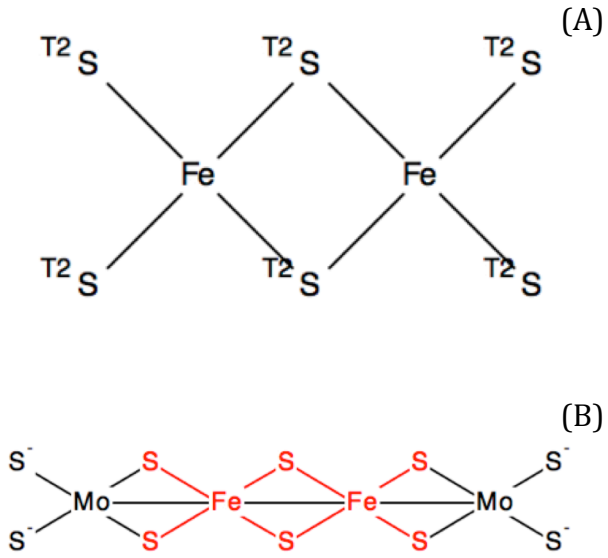


Figure 5: (A) Conquest search fragment for the Fe_2S_2 cluster FES with bond numbers set for non-iron atoms. (B) Unreasonable hit resulting from search in panel (A).

Conclusions

Always verify that the results from a structure search are reasonable. The first simple search attempt in this example resulted in many unreasonable hits for FES. Using the "number of bonded atoms" option is a powerful tool for filtering results.

References

Bruno, I. J., Cole, J. C., Edgington, P. R., Kessler, M., Macrae, C. F., McCabe, P., Pearson, J. & Taylor, R. (2002). *Acta Crystallogr. B*. **58**, 389–397.

Burley, S. K., Berman, H. M., Bhikadiya, C., Bi, C., Chen, L., Costanzo, L. D., Christie, C., Duarte, J. M., Dutta, S., Feng, Z., Ghosh, S., Goodsell, D. S., Green, R. K., Guranovic, V., Guzenko, D., Hudson, B. P., Liang, Y., Lowe, R., Peisach, E., Periskova, I., Randle, C., Rose, A., Sekharan, M., Shao, C., Tao, Y.-P., Valasatava, Y., Voigt, M., Westbrook, J., Young, J., Zardecki, C., Zhuravleva, M., Kurisu, G., Nakamura, H., Kengaku, Y., Cho, H., Sato, J., Kim, J. Y., Ikegawa, Y., Nakagawa, A., Yamashita, R., Kudou, T., Bekker, G.-J., Suzuki, H., Iwata, T., Yokochi, M., Kobayashi, N., Fujiwara, T., Velankar, S., Kleywegt, G. J., Anyango, S., Armstrong, D. R., Berrisford, J. M., Conroy, M. J., Dana, J. M., Deshpande, M., Gane, P., Gáborová, R., Gupta, D., Gutmanas, A., Koča, J., Mak, L., Mir, S., Mukhopadhyay, A., Nadzirin, N., Nair, S., Patwardhan, A., Paysan-Lafosse, T., Pravda, L., Salih, O., Sehnal, D., Varadi, M., Vařeková, R., Markley, J. L., Hoch, J. C., Romero, P. R., Baskaran, K., Maziuk, D., Ulrich, E. L., Wedell, J. R., Yao, H., Livny, M. & Ioannidis, Y. E. (2019). *Nucleic Acids Res.* **47**, D520–D528.

DeLano, W. L. (2002). PyMOL 0.99.

Gražulis, S., Chateigner, D., Downs, R. T., Yokochi, A. F. T., Quirós, M., Lutterotti, L., Manakova, E., Butkus, J., Moeck, P. & Le Bail, A. (2009). *J. Appl. Crystallogr.* **42**, 726–729.

Groom, C. R., Bruno, I. J., Lightfoot, M. P. & Ward, S. C. (2016). *Acta Crystallogr. Sect. B Struct. Sci. Cryst. Eng. Mater.* **72**, 171–179.

Moriarty, N. W. & Adams, P. D. (2019). *Acta Crystallogr. Sect. Struct. Biol.* **75**, 16–20.

Vagin, A. A., Steiner, R. A., Lebedev, A. A., Potterton, L., McNicholas, S., Long, F. & Murshudov, G. N. (2004). *Acta Crystallogr. D Biol. Crystallogr.* **60**, 2184–2195.

Westbrook, J. D., Shao, C., Feng, Z., Zhuravleva, M., Velankar, S. & Young, J. (2015). *Bioinformatics*. **31**, 1274–1278.

Ensemble refinement produces consistent R-free values but smaller ensemble sizes than previously reported

Stephanie A. Wankowicz[a,b] and James S. Fraser[a,b,c,d]

a - Biophysics Graduate Program, University of California San Francisco, San Francisco, CA, USA.
b - Department of Bioengineering and Therapeutic Sciences, University of California San Francisco, San Francisco, CA, USA.
c - Quantitative Biosciences Institute, University of California San Francisco, San Francisco, CA, USA.
d - Molecular Biophysics and Integrated Bioimaging Division, Lawrence Berkeley National Laboratory, Berkeley, CA
Correspondence email: jfraser@fraserlab.com

Introduction

Ensemble refinement combines molecular dynamics (MD) simulations with crystallographic data to provide a model of atomic fluctuations that are present in the crystal lattice. As implemented in *phenix.ensemble_refinement*, MD simulations are performed where the model is restrained by a time-averaged X-ray restraint (Burnley et al. 2012). Because the agreement with observed structure factors is calculated by averaging of several recent snapshots of the MD simulation, ensemble refinement differs significantly from traditional refinement where a single structure is used to calculate the agreement. To attempt to control for crystalline disorder, a Translation/Libration/Screw (TLS) model model is fitted prior to the simulation, leaving the simulation to fit the residual difference density. After the simulation is run, a procedure reduces the ensemble size down from all snapshots acquired during the period to a minimal set that will reproduce the R-free within a tolerated value. In the original paper describing *phenix.ensemble_refinement*, this yielded 39-600 ensemble members in the 20 PDB depositions that were subjected to refinement. The structural diversity across these ensemble members is a representation of the residual conformational heterogeneity after accounting for the disorder modeled by the TLS model.

We set out to run ensemble refinement on a large number of publicly available X-ray crystallography structures. Although some parameter names and default values had apparently changed since the original paper, the online documentation provided a guide to reasonable values (Phenix documentation: ensemble_refinement.html). For our analysis, all structures had a resolution between 1-2.5 Angstroms. Using Phenix version 1.15, we pursued the following workflow (code is available on github[1]).

1. Download existing model and structure factor files
2. Run *phenix.ready_set*
3. Re-refinement of model using *phenix.refine*
4. Ensemble refinement over a grid search of parameters
5. Selection of best model based on R_{free}

All input parameters for our analysis are available (https://ucsf.app.box.com/folder/95195345802).

Non-default inputs

- `wxray_coupled_tbath_offset: grid search of 2.5, 5, 10`

Errors

About 10% of the structures failed during refinement. There were numerous reasons for these failures including, a lack of appropriate

[1]https://github.com/stephaniewanko/Fraser_Lab/tree/master/phenix_pipeline

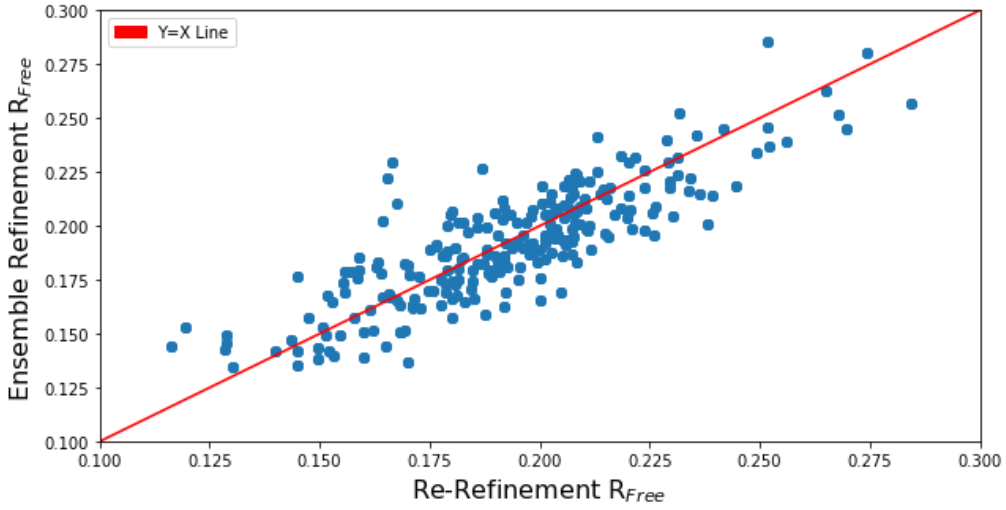


Figure 1. *R_{free} values from re-refinement and ensemble refinement are correlated (R^2=0.91). In 418 (57.6%) structures, the ensemble refinement R_{free} value was lower than the refinement R_{free} value. In 307 (42.3%) structures, the ensemble refinement R_{free} value was higher than the refinement R_{free} value.*

intensities or amplitude information, poor maps, and issues with ligands.

Conclusion

There were two major differences between our analysis and the original Burnley 2012 paper (Burnley et al. 2012). First, in the Burnley paper all 20 structures had reduced R_{free} values when subjected to ensemble refinement. In our study, overall, ensemble refinement R_{free} was comparable to re-refinement R_{free}, with 57.6% of structures having an improved R_{free} with ensemble refinement compared to traditional refinement, as shown in figure 1. This may be due to non-optimal parameter selection or insufficient model preparation. Second, it was unclear why we were getting such smaller ensemble size compared to the 2012 paper. We were expecting many ensemble sizes to be greater than 100; however, all of our structures returned ensembles <100, as demonstrated in figure 2. Although the ensembles obviously contain more diversity than single structures, we were curious as to the underlying cause of the greatly reduced ensemble size. To further investigate, we tested our ensemble refinement pipeline on the 20 PDB models originally analyzed in Burnley 2012 paper.

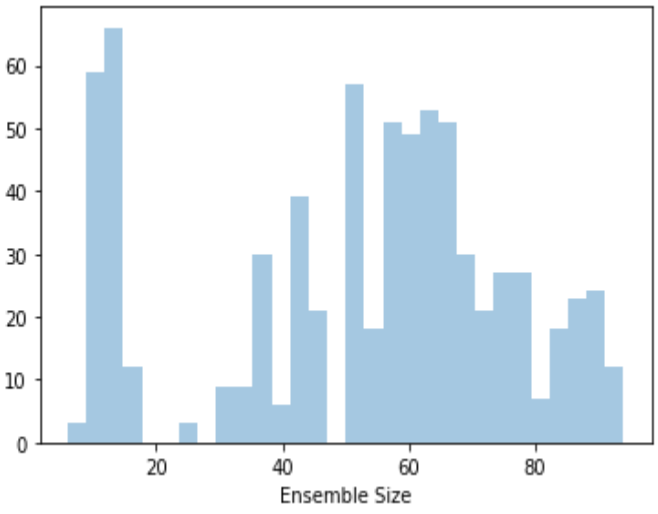


Figure 2. *Most structures have smaller ensemble sizes (the number of models in the ensemble output) than we expected based on the results in Burnley 2012.*

Recreating Burnley 2012 Paper

To recreate the results from the Burnley 2012 paper, we followed the same pipeline outlined above. Of note, while we automatically re-refined the models coming from the PDB, we did not perform any manual refinement, which

Table 1. *Input R values from Burnley 2012 compared to our input to our recreation.*

PDB	Resol.	Original Ensemble Size	R_{work} Burnley 2012	R_{free} Burnley 2012	R_{work} Recreation	R_{free} Recreation	Recreation Lowest R_{free} Ensemble Size
1kzk	1.1	600	0.125	0.153	0.155	0.179	100
3k0m	1.3	250	0.104	0.129	0.127	0.144	167
3k0n	1.4	209	0.115	0.133	0.117	0.143	167
2pc0	1.4	250	0.145	0.188	0.231	0.252	125
1uoy	1.5	167	0.104	0.137	0.136	0.165	125
3ca7	1.5	40	0.149	0.184	0.237	0.292	56
2r8q	1.5	200	0.132	0.162	0.164	0.188	125
3ql0	1.6	70	0.204	0.254	0.217	0.256	50
1x6p	1.6	400	0.121	0.149	0.141	0.163	134
1f2f	1.7	143	0.128	0.168	0.170	0.210	84
3ql3	1.8	80	0.160	0.208	0.171	0.207	56
1ytt	1.8	84	0.139	0.174	0.179	0.206	63
3gwh	2.0	39	0.160	0.200	0.198	0.230	67
1bv1	2.0	78	0.149	0.182	0.188	0.240	84
1iep	2.1	200	0.183	0.238	0.207	0.256	63
2xfa	2.1	100	0.171	0.217	0.226	0.261	60
3odu	2.5	50	0.208	0.269	0.247	0.297	32
1m52	2.6	50	0.161	0.211	0.198	0.240	32
3cm8	2.9	67	0.194	0.235	0.231	0.264	39
3rze	3.1	72	0.210	0.280	0.250	0.289	32

left us with input structures with slightly higher R_{free}/R_{work} compared to the Burnley 2012 paper (table 1). We extended our grid search to include three parameters suggested by the Phenix documentation (pTLS, wxray_coupled_tbath_offset, tx).

- pTLS defines the fraction of atoms included in the TLS fitting procedure. This is intended to model static crystalline lattice disorder and varying this parameter results in movement being absorbed by the TLS B-factors rather than by atomic fluctuations.
- wxray_coupled_tbath_offset controls the X-ray weight. This helps ensures that the simulation runs at the target temperature.
- tx dictates the structure factor memory relaxation time. This governs the time period for which a particular conformation retains its influence. The higher the number, the more a particular conformation affects the average.

Additionally, we added harmonic restraints for all ligands in each structure. Of note, while Burnley 2012 paper reported only one ensemble structure per PDB, we had 36 ensemble structures (corresponding to a 3 x 3 x 4 grid search of the parameters pTLS, tx, wxray_coupled_tbath_offset) and choose one select ensemble structure based on the criteria of lowest R_{free}. This test was run on Phenix version dev-3584 (a mid 2019 version).

Non-default inputs

- `wxray_coupled_tbath_offset: grid search of 2.5, 5, 10`
- `pTLS: grid search of 0.6,0.8,1.0`
- `tx: 0.5, 0.8, 1, 1.5`

Outputs

As shown in table 1, in almost all cases, the ensemble sizes were lower than what was found in the Burnley 2012 paper. Figure 3 illustrates that we found that R_{free} correlated with ensemble size (R^2 =-0.61). Similarly, resolution was slightly correlated with ensemble size (R^2=-0.48). Overall, the recreated R_{free} were highly correlated with the R_{free} from the Burnley 2012 paper (R^2=0.892). We could not identify any pattern between the parameter values correlated with R_{free} and the optimal parameter value as judged by R_{free} was idiosyncratic for each structure.

As we wanted to use ensemble refinement to assess dynamics, we want to see if different parameter values (pTLS, wxray_coupled_tbath_offset, tx) change the RMSF. We examined all structures, but focus our analysis below on C-ABL kinase domain in complex with STI-571 (PDB: 1IEP).

While the RMSF values of C-ABL kinase domain in complex with STI-571 (PDB: 1IEP) were highly correlated (>0.8) across all parameter values, highlighted in figure 4, there were only some notable deviations in magnitude for the pTLS parameter values, demonstrated in figure 5.

Because of the lack of correlation between the parameters and R_{free} values, and the relative consistency of the RMSF calculations, we evaluated each PDB independently and chose parameters that yielded the lowest R_{free}. At least one of the 20 PDBs had an optimal ensemble using each of the wxray_coupled_tbath_offset and pTLS parameter values. For the tx parameter only 3 out of the 5 values were used in optimal ensembles (0.8, 1.0, 1.5).

Conclusions

Overall, we were still getting much smaller ensemble sizes compared to the Burnley 2012 paper. However, our R_{free} values correlated very well with the R_{free} values from the paper giving

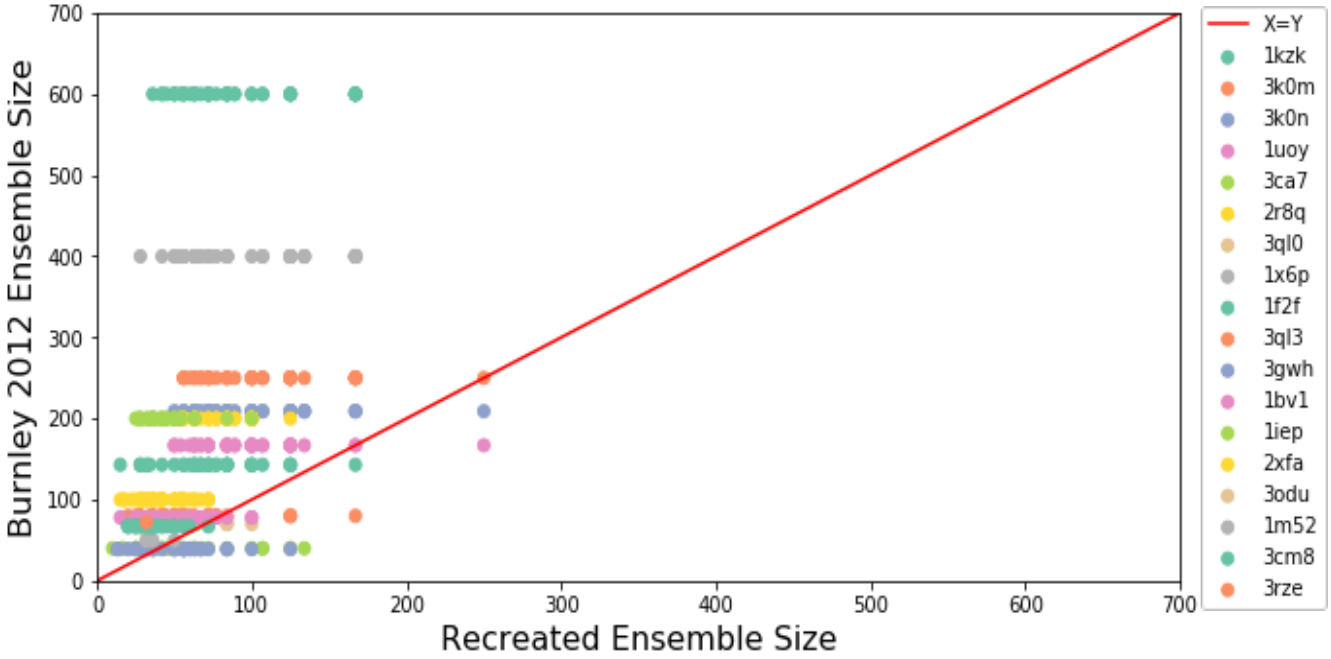


Figure 3. *Burnley 2012 paper ensemble size compared to our recreated ensemble size. In almost all cases, the ensemble sizes were lower than what was found in the Burnley 2012 paper.*

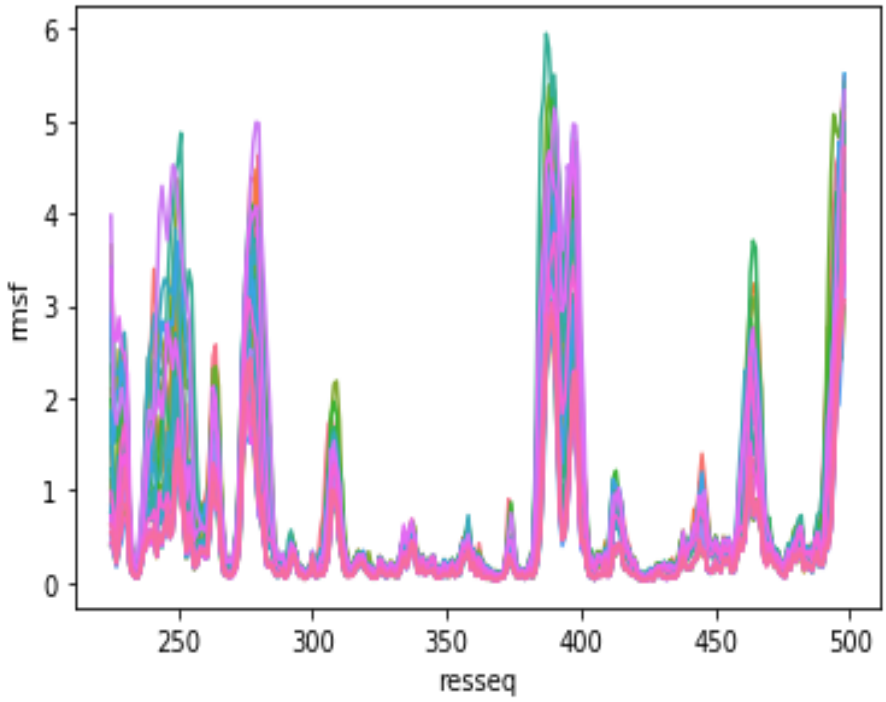


Figure 4. *RMSF of C-ABL kinase domain in complex with STI-571(PDB: 1IEP) across all 45 parameter values.*

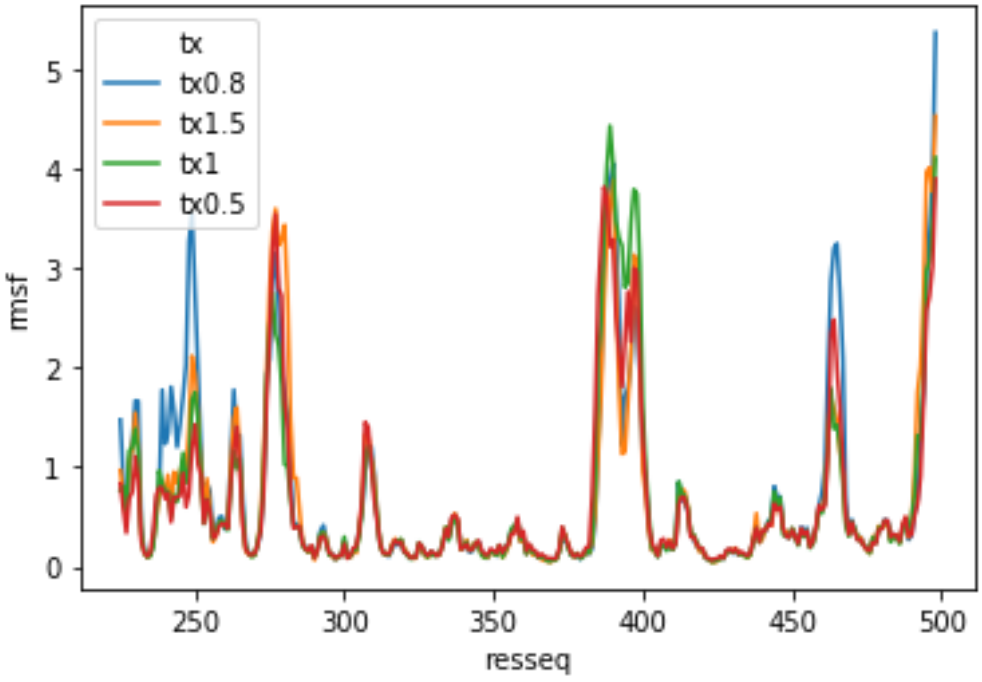


Figure 5. *RMSF of C-ABL kinase domain in complex with STI-571(PDB: 1IEP) across all tx parameter values.*

us confidence in the underlying procedure. The wxray_coupled_tbath_offset, pTLS, or tx parameter values were not correlated with R_{free}, R_{work}, or the ensemble size. In terms of RMSF changes, the only parameter that produced a major difference was pTLS, as expected. pTLS determines the percentage of atoms included in

the TLS model, which predicts the local positional displacement of atoms in a crystal structure with the underlying assumption that the atoms included are members of a rigid body. In our results, lower pTLS values (fewer atoms included in the pTLS model) have higher RMSF on average. It is unclear to us if choosing a model based on the best R_{free} will result in accurate results for protein conformational heterogeneity, especially when comparing two protein structures with different pTLS values.

Investigating the ensemble size difference

To try to resolve the discrepancy in the ensemble sizes from the original 2012 paper to our recreation of their results, we used the optimal parameter values from the test above for each PDB and tested four other keyword changes that we predicted might give us results closer to the Burnley 2012 paper.

1) Using the Phenix version released most closely to the Burnley 2012 paper (version 1.8.2, the first release to contain the *phenix.ensemble_refinement* command).
2) Removing the use of the conformation dependent restraint library.
3) Re-setting the ensemble R_{free} tolerance parameter to 0.001
4) Re-setting the ensemble reduction feature to false

Testing Phenix version 1.8.2

The Burnley 2012 paper was run using a different Phenix version than we used with our recreation. We ran ensemble refinement on Phenix version 1.8.2, which corresponds to the public release of the method after the Burnley 2012 paper.

Non-default inputs (Phenix version 1.8.2)

- `wxray_coupled_tbath_offset, pTLS, tx parameter values corresponding to the optimal` R_{free} `for each individual PDB from the previous tests.`

Errors

Only five out of the 20 structures ran ensemble refinement successfully. There were multiple reasons for failures. These included a pTLS error with chain breaks and errors reading in parameters fed into ensemble refinement.

Conclusions

Many structures failed to run ensemble refinement. However, for the five structures that finished, the ensemble sizes were still smaller than expected based on the Burnley 2012 paper and were highly correlated with our previous recreation, as seen in figure 6 and 7. Figure 8 demonstrated that we continue to observe a good correlation between the original and updated R_{free} values (R^2=0.95).

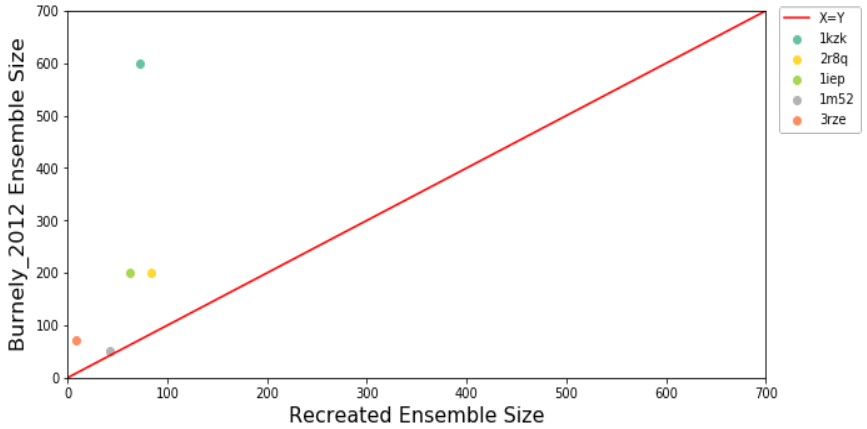


Figure 6. *Recreated ensemble sizes are smaller compared to the ensemble sizes in Burnley 2012 (R^2=0.57).*

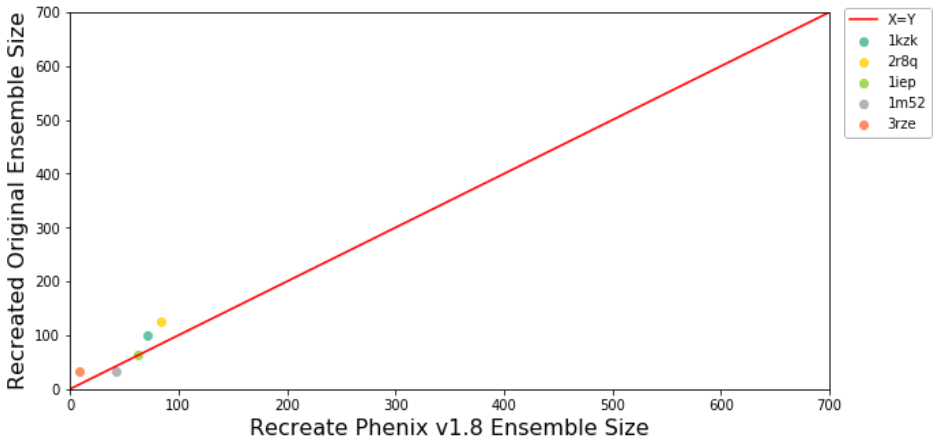


Figure 7. *Recreated ensemble sizes with Phenix version are similar to the initially recreated ensemble sizes (R^2=0.88).*

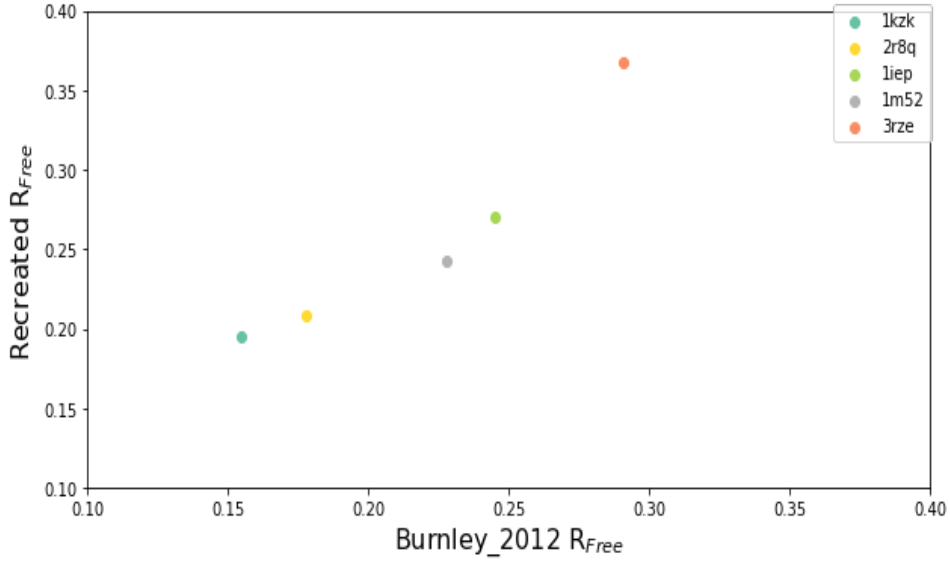


Figure 8. *Recreated R_{free} were highly correlated with the* Burnley 2012 *paper (R^2=0.95).*

Reverting R_{free} Tolerance to 0.001

The last step of ensemble refinement takes all of the snapshots saved from the MD simulation and selects the lowest number of models that together have an R_{free} within the percentage of the full ensemble R_{free}. This percentage is defined by the R_{free} tolerance parameter. The current version of Phenix (1.16), defaults this parameter to 0.0025 but in the Burnely 2012 paper, it was set to 0.001. Therefore, we tested if we could increase the ensemble size by changing this parameter back to what was used in the paper using Phenix version 1.16.

Non-default inputs (Phenix version 1.16)

- `Wxray_coupled_tbath_offset, pTLS, tx parameter values corresponding to the best` R_{free} `for each individual structure.`
- `ensemble_reduction_rfree_tolerance = 0.001`

Conclusions

Reducing the R_{free} tolerance parameter back to where it was initially set did increase our recreated ensemble size (median increase: 26 models), see figure 9. However, for many of these structures the

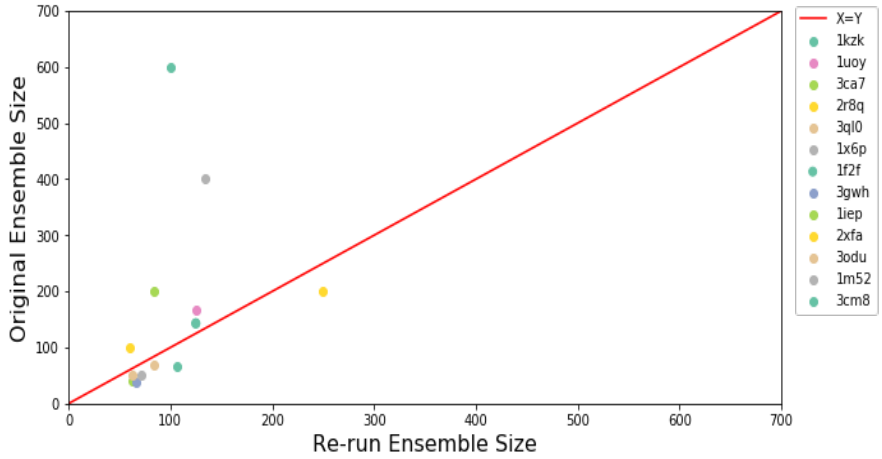


Figure 9. Burnley 2012 paper *ensemble size compared to our recreated ensemble size with an R_{free} Tolerance of 0.001 (R^2=0.307).*

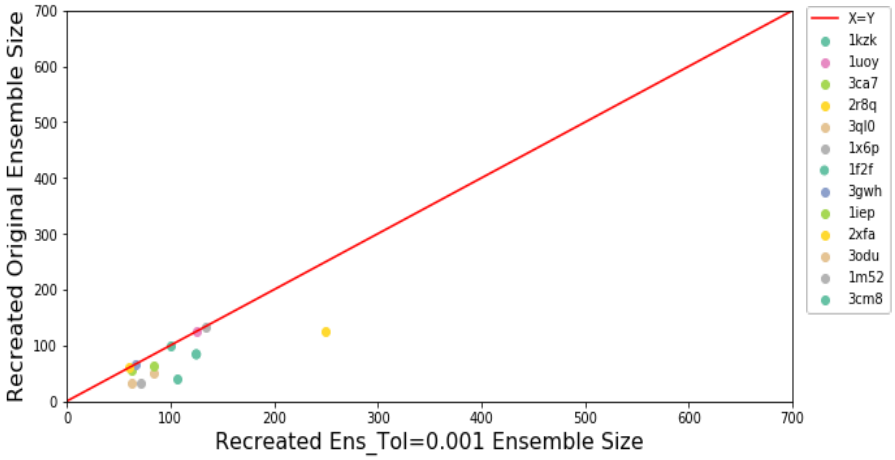


Figure 10. *Recreated ensemble sizes with ensemble_rfree_tolerence parameter =0.001 are mostly larger than the initially recreated ensemble sizes(R^2=0.71).*

number of models were still far below what was observed in the Burnley 2012 paper, as seen in figure 10. Figure 11 shows that the R_{free} correlation was still observed between the recreation on the Burnley 2012 paper.

Removing Conformational Dependent Library (CDL)

In the Burnley 2012 paper, the default restraints were Engh and Huber, but more recent versions of phenix use the conformation dependent library (CDL). One hypothesis is that the older restraints would bias ensembles to have energetically reasonable angles and bond lengths compared to the modern CDL restraints, leading the ensemble sizes to decrease under CDL restraints. Therefore, we set the cdl restraint library to false. Of note there are three other library (omega_cdl, rdl, and hpdl) that are also available as parameters but are set as false as the default.

Non-default inputs (Phenix version 1.16)

- `Wxray_coupled_tbath_offset, pTLS, tx parameter values corresponding to the best` R_{free} `for each individual structure.`
- `restraints_library_cdl = False`

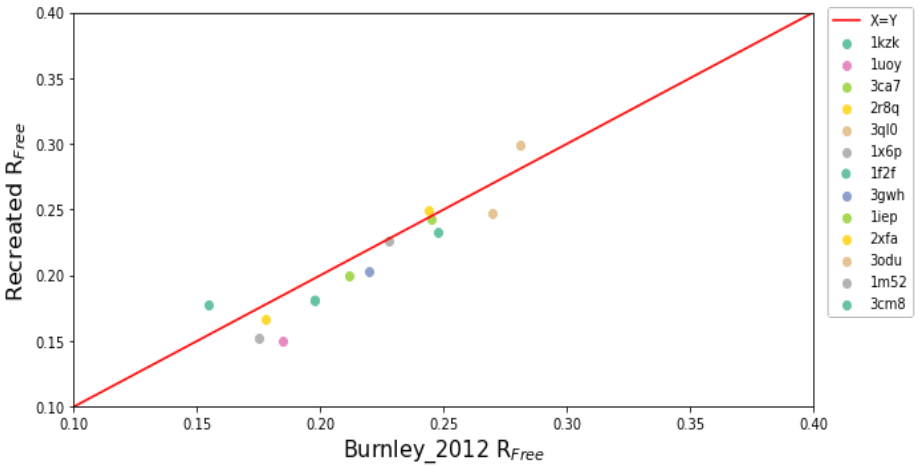


***Figure 11**. Recreated R_{free} with an R_{free} tolerance of 0.001 were highly correlated (R^2 =0.9) with the R_{free} Burnley 2012 paper*

Conclusions

By turning the CDL restraints off, we did not observe an increase in ensemble sizes, see figure 12. For one structure (PDB:3GWH), the size of the ensemble did increase, see figure 13. We suspect this is due to the input model having a high number of geometry outliers. By turning off CDL, we may have further increased the geometry problem, resulting in a larger ensemble size. There was still a high correlation between the original and recreated R_{free} (R^2=0.96) as shown in figure 14.

Testing Ensemble Reduction

When the ensemble reduction parameter is turned to false, ensemble refinement outputs all of the models in the ensemble rather than selecting down a smaller number of models to match the R_{free} tolerance value. By turning this value off, we hypothesized that the size of the ensembles would all be 500, since that is the number of models created in ensemble refinement (based on default parameters).

Non-default inputs

```
wxray_coupled_tbath_offset, pTLS, tx
parameter values corresponding to
the best Rfree for each individual
structure.
ensemble_reduction = False
```

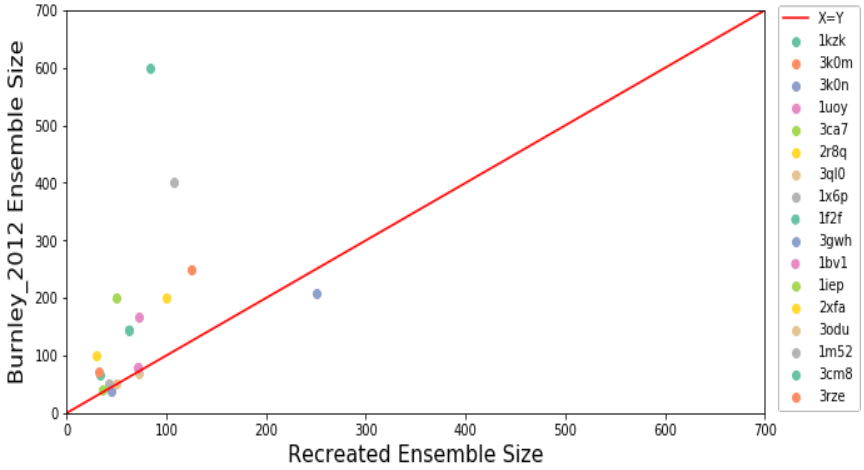


***Figure 12.** Recreated ensemble sizes are smaller compared to the ensemble sizes in Burnley 2012 (R^2=0.4).*

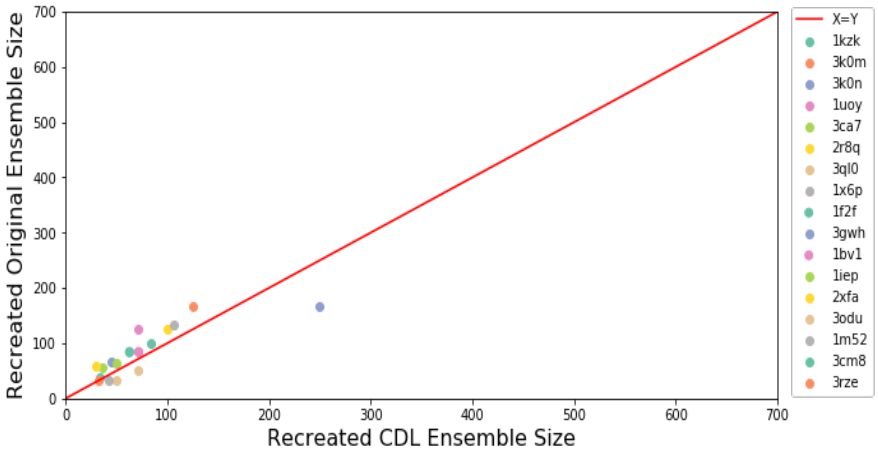


Figure 13. *Recreated ensemble sizes with CDL parameter=False are similar to the initially recreated ensemble sizes(R^2=0.82).*

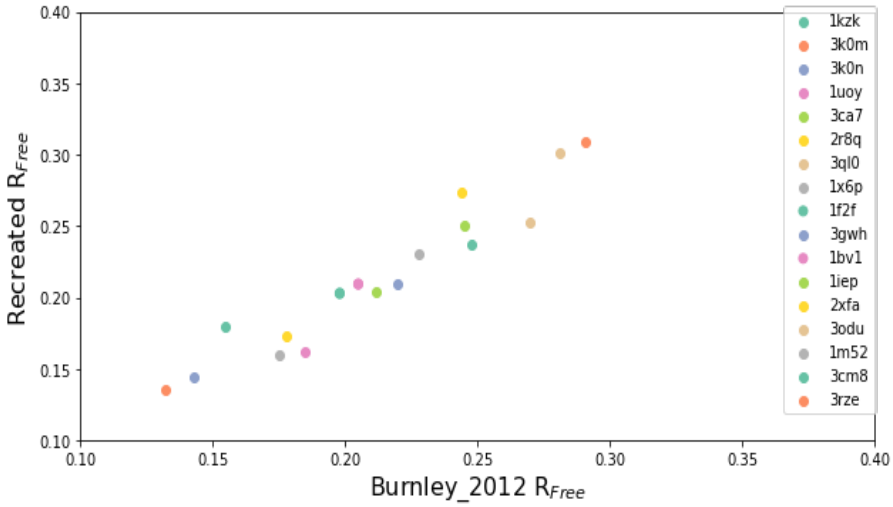


Figure 14. *Recreated R_{free} were highly correlated with the* Burnley 2012 *paper (R^2=0.96).*

Conclusion

While turning off the ensemble reduction parameter did increase the ensemble size, we only observed two ensemble size. There was not a trend of the ensemble size observed in Burnley 2012 and our recreated ensemble size (R^2=-0.03), see figure 15. There was still a high correlation between the original and recreated R_{free} (R^2=0.95) as shown in figure 16.

Using specific TLS selections from the 2012 paper

After discussing our results with the original authors, we realized that the authors used specific TLS selections and other values for the pTLS, tx, and wxray_coupled_tbath_offset parameters. We then set the following parameters from their log files in ensemble refinement Phenix version 1.16. Of note, there were additional parameters that were different between the two versions that we were not able to change.

Parameters changed:

- `pTLS`
- `tx`
- `wxray_coupled_tbath_offset`

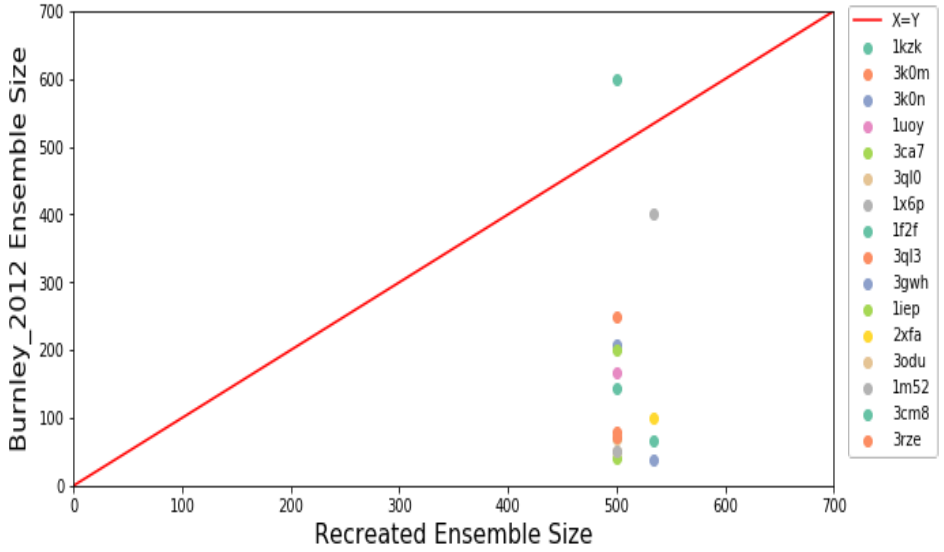


Figure 15. *Recreated ensemble sizes are larger compared to the ensemble sizes in Burnley 2012 (R^2=-0.03).*

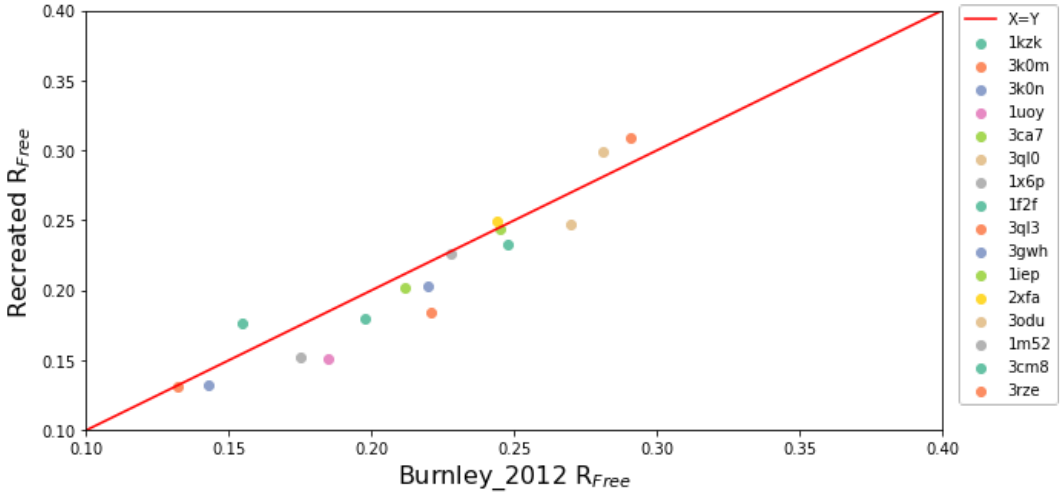


Figure 16. *Recreated R_{free} were highly correlated with the* Burnley 2012 *paper (R^2=0.95).*

- `pTLS selections`
- `harmonic restraints`

Conclusion

Altering these parameters more closely resembled our original recreation both in ensemble size and Rfree and does not underly the larger ensembles in the 2012 paper, as shown in figure 17.

Overall conclusions

While we were not fully able to recreate the ensembles in the Burnley 2012 analysis, we are confident that ensemble refinement is stable and outputs interesting representations of conformational heterogeneity. Metrics that can be used to assess those representations, such as R values or RMSF are not greatly affected by the changes to the method. We would advise future users of the ensemble refinement methods that you may observe lower number of models in each ensemble compared to the Burnley 2012 paper. The only parameter change that seemed to increase the ensemble size was changing

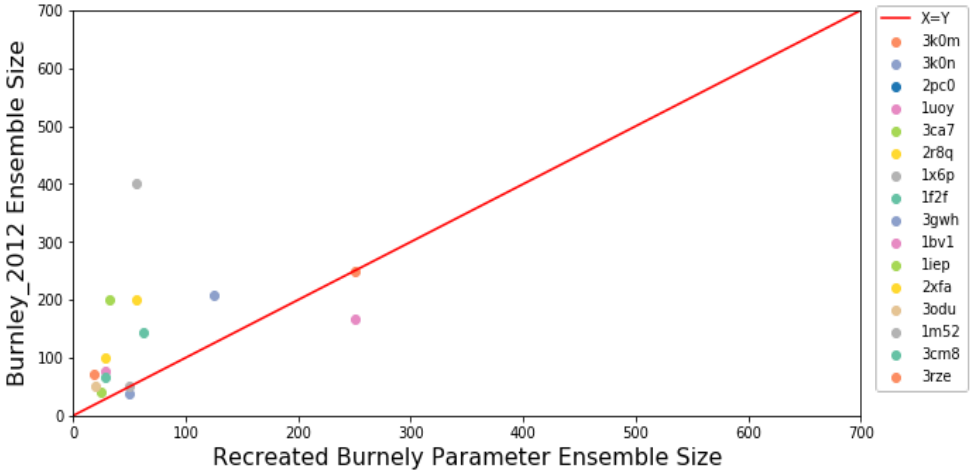


Figure 17. *Recreated ensemble sizes with the Burnley 2012 parameters not correlated to the Burnley 2012 parameters (R^2=0.41).*

ensemble_reduction to false but these values did not correlate with the ensemble sizes observed in Burnley 2012 paper. As advised on the phenix website, we suggest that users perform a grid search over the parameters of tx, wxray_coupled_tbath_offset, and pTLS. Additionally, adding harmonic restraints on all non-water HETATMS is important. All input parameters for our analysis are available (https://ucsf.app.box.com/folder/95195345802) . Moving forward, we would like to encourage publishing the exact parameters used in any refinement procedure for reproducibility.

References

Burnley, B. Tom, Pavel V. Afonine, Paul D. Adams, and Piet Gros. 2012. "Modelling Dynamics in Protein Crystal Structures by Ensemble Refinement." *eLife*.

dtmin - a Domain Tunable Python Minimizer

Duncan H. Stockwell[a*], Randy J. Read[a] and Airlie J. McCoy[a]
[a]*Department of Haematology, Cambridge Institute for Medical Research, Clinical School of Medicine, University of Cambridge, CB2 0XY, UK*
*Correspondence email: dhs37@cam.ac.uk

Introduction

Phaser's minimizer, written in C++, has been developed to meet the needs of optimizing likelihood target functions for molecular replacement (MR) and single-wavelength anomalous diffraction (SAD) phasing. Optimizing likelihood targets, by minimizing the minus-log-likelihood gain (see (McCoy, 2004) for review), is intrinsic to *Phaser's* anisotropy correction, translational non-crystallographic symmetry epsilon factor correction, rigid body refinement of posed ensembles, gyre refinement of oriented ensembles, single atom molecular replacement, SAD substructure content analysis, SAD substructure refinement, and log-likelihood pruning (see (McCoy *et al.*, 2009) for documentation). The minimization takes initial parameter value estimates (e.g. approximate global minima obtained from grid search methods) and improves them to give optimal estimates of structure factor phases and map coefficients throughout the MR and SAD phasing pathways. *Phaser* is currently being reconfigured as *phasertng*, to support directed acyclic graph data structures; and *phasertng* is central to our development of *phaser.voyager*, which will leverage the directed acyclic graph data structure to allow dynamic decision making in phasing strategies. As part of this project, we have developed a python minimizer library, *dtmin*, that has the same advanced functionality as the *phaser* minimizer and made it available in the *cctbx* library. The advanced functionality includes the ability to change the subset of refined parameters each refinement 'macrocycle' (see below), bounding parameters, logarithmic reparameterization, outlier rejection, use of large shift estimation, and a mechanism for debugging derivative calculations ('study_parameters'). This article documents how to use *dtmin* and briefly compares it to the *cctbx* minimizer *scitbx.lbfgs*.

Terms

Target Function: the function to be minimized. There are various names for this in the literature and in different fields, such as 'cost' function, or 'loss' function. In crystallography this is typically of the form:

$$\text{Target} = -\sum_i \log\left(P(x_i|\,\theta_1,\ldots,\theta_J)\right) + \sum_j k_j\left(\theta_j-\theta_{0_j}\right)^2$$

where $P(x_i|\theta_1,\ldots,\theta_J)$ is the likelihood of the ith data value given the model parameters, and $k_j\left(\theta_j-\theta_{0_j}\right)^2$ is a restraint term for the jth model parameter θ_j. The term 'restraint' is used in crystallography and molecular dynamics; in other literature it is known as a 'penalty function'.

Gradient: the array of first derivatives, with respect to the parameters, of the function to be minimized. The gradient may be calculated analytically or using finite differences; which method is faster depends critically on the function and its parameterization. The target function is often an intermediate in the calculation of the gradient.

Hessian: the matrix of second derivatives of the function to be minimized. As with the gradient function, this may be calculated analytically or with finite differences. Some minimization methods do not require all elements of the matrix to be calculated. Whether the Hessian is more computationally intensive to compute than the gradient depends on the complexity of the second derivatives and the number of elements in the Hessian that are calculated.

Bounds: range of values that a parameter is permitted to take. For example, σ_A is restricted to values between 0 and 1. Bounds need not only be the mathematically allowed values, they can be used to restrict the range of values for good convergence. For example, B-factors may be restricted to values between -20Å^2 and 500Å^2, although mathematically they may take any value. Bounds are optional for each parameter.

Reparameterization: Some parameters have more quadratic behavior if reparameterized for refinement, e.g. B-factors. Shifts are calculated in terms of the reparameterized variable. Although many reparameterizations are possible, in practice we have found that the most effective reparameterization is logarithmic. When performing logarithmic reparameterization, an 'offset' is applied to the parameter before taking the logarithm. The offset supports reparameterization of negative parameter values while different values should be tested to optimize the convergence. *dtmin* has a logarithmic reparameterization available by flagging each parameter true/false and supplying an 'offset' value in the case of true.

Outliers: data points that should be excluded from the target value calculation. This may be because they take disallowed values, will bias the refinement, or are extremely unlikely to be good estimates of the true values. Data points can also be excluded to speed up the target value calculation. Good minimization should always include a robust method for outlier rejection.

Protocol: The protocol specifies which subset of the parameters are refined during each *macrocycle* of refinement. For stable refinement, it is often useful to refine the parameters in steps, first minimizing values whose initial values are likely to be far from convergence and then adding less significant parameters in later macrocycles. Within each macrocycle, there are *microcycles* each of which consists of finding a direction and performing a line-search. See figure 1.

Large Shifts: The maximum distance that you think each parameter should reasonably be able to move in one *microcycle*. They must be specified for each parameter and are used to stabilize the minimizer by damping parameters shifts that try to move parameters more than their 'large shift' in the first step of a line search. For example, 0.1 is a large shift for a fractional coordinate shift, but negligible for an atomic B-factor; appropriate 'large shift' values for these two parameters could be 0.01 and 10. Large shift values can also be used for rough curvature estimates by taking the reciprocal of the 'large shift' squared, effectively putting different parameter types on a common scale. Different large shift values will have a significant effect on the behavior of the minimizer and should be carefully optimized.

General properties

dtmin minimizes a real valued function of *n* real parameters using an iterative line-search based method. Given some starting values for the parameters, a direction in the *n*-dimensional parameter space is chosen and a 1D minimization procedure (the 'line-search') carried out along this direction. In general, the direction chosen for a line search differs with minimization method. For example, in the method of 'steepest descent', the negative gradient of the function at the starting point is taken as the line-search direction. This method, although simple, has slow convergence (requires a great many iterations) for functions that have 'valleys' rather than 'holes'. *dtmin* has implementations of Newton's method and the BFGS (Broyden–Fletcher–Goldfarb–Shanno) method for finding line-search directions. Both Newton's method and the BFGS algorithm make use of function gradients and the Hessian. While using the Hessian typically decreases the number of iterations to convergence over methods relying solely on gradient evaluations, Hessian evaluation increases the computational cost of each iteration over purely gradient driven methods. In some cases, the computational cost may be very high. Gradient driven methods can be implemented by setting the Hessian to the identity matrix

(equivalent to the method of 'steepest descent'), or to a constant estimate of the Hessian at the minimum, such as from "large shift" values for each parameter (see below). Alternatively, the computational cost of Hessian evaluation can be reduced by only providing the diagonals of the Hessian matrix (the 'curvatures').

In *dtmin*, termination occurs when one of the following criteria is met: the gradient values are all exactly zero, meaning we have found a true minimum (this is unlikely both due to numerical imprecision and because other termination criteria will be met first); every parameter is bounded, we are at the bounds and the step direction wants to push all the parameters over their bounds; none of the parameters are shifted by a significant amount, where significance is calibrated by scales derived from the diagonals of the Hessian; the function does not decrease by a significant amount, where significance is calibrated by the current function value and the numerical precision; or the maximum number of microcycles has been reached.

Architecture

The *dtmin* is architected as two main classes. The first, the 'minimizer' class, performs the overall iteration to convergence in its 'run' method and should not be modified. The second, the user implemented 'refinement' class, is specific to your problem – this is where the implementation for your target function goes! It is derived from a single RefineBase class, which in turn is derived from four base classes (Compulsory, Optional, Logging and Auxiliary; see below). Only the functions in Compulsory must be implemented by the user for the minimizer to run.

Minimizer:

The minimizer controls the minimization strategy, shown in Figure 1. It calls functions defined in the RefineBase class. To run the minimizer, derive your implementation-specific Refine Class from RefineBase and pass it to the 'run' function of the Minimizer class, along with the protocol for refinement (which parameters to refine in each macrocycle), the maximum number of microcycles per macrocycle (typically around 50), the minimizer to use (either 'newton' or 'bfgs') and whether or not to output the study_parameters gradient and curvatures debugging information (this will be 'False' for production code).

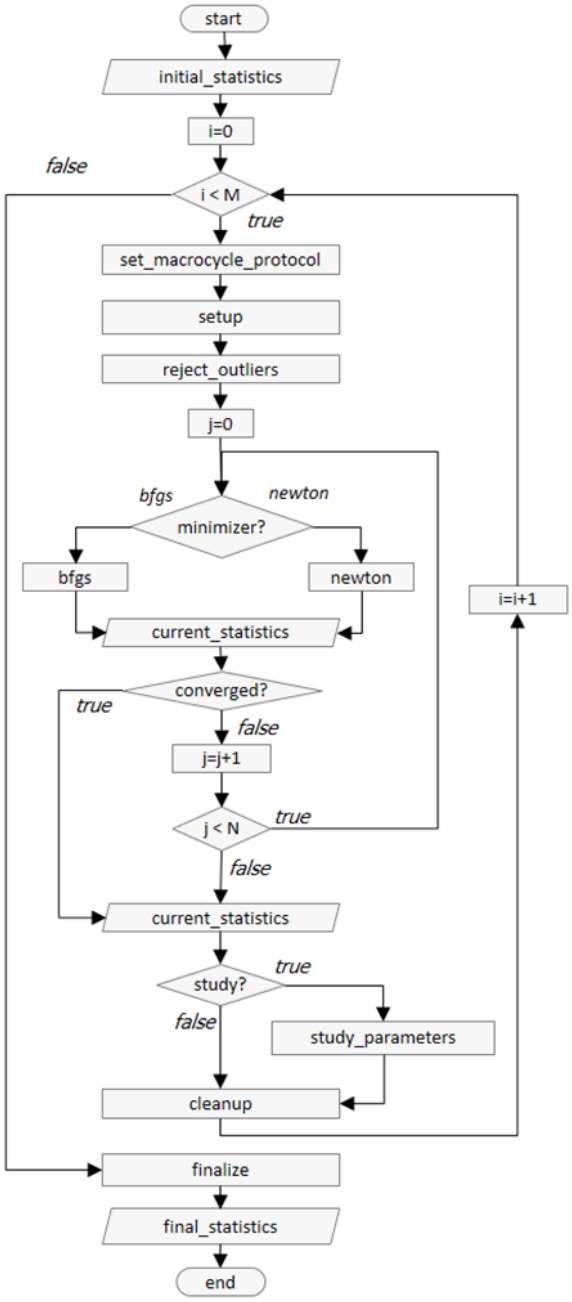


Figure 1: Flow diagram of the overall minimization strategy of the Minimizer class, where *i* is the macrocycle index and M is the number of macrocycles, *j* is the microcycle index and N the maximum number of microcycles per macrocycle.

Initial parameters: The starting values for the refinement are set in the initialization of the Refine Class. No functions are provided for initial setup.

Study parameters: The study_parameters function is used in development for checking values of the implemented gradients and curvatures against values computed using finite differences. The function outputs values obtained from the calls to target_gradient_hessian and values obtained from finite difference first derivatives, and finite difference second derivatives calculated from function values or from analytic gradients. The results can be interrogated either by inspection or using a 'mathematica' (Wolfram Research Inc., 2019) notebook available for download from phaserwiki (McCoy *et al.*, 2009). Note that the function target_gradient_hessian itself may have been implemented to return gradients and Hessians derived from finite difference methods, in which case the study_parameters protocol is superfluous.

Output: Output from the minimizer is controlled by the level of output requested. Default output from the minimizer reports the path through the minimization and can be used to optimize the path through refinement. If the functions in the Logging class are implemented (see below), intermediate statistics can be reported during the minimization.

Result: After the minimizer's run method is called, the Refine object is left in the minimized state.

RefineBase:

Your function to minimize should be defined in a class derived from RefineBase. RefineBase is, in turn, derived from four classes:

Compulsory: Compulsory functions must be implemented and will throw an exception when called by the minimizer if they are not.

Optional: Optional functions have default implementations, although overriding the defaults is recommended for optimal performance of the minimizer.

Logging: Logging functions produce no output by default and should be customized to generate output before, during and after minimization.

Auxiliary: Helper functions required for minimization, which should not be modified by the user.

See Table 1 for a summary of the functions in classes Compulsory, Optional and Logging. We have provided two example implementations, which are described below, to help guide users in how to implement the necessary functions for their problem. The function to set the protocol at the beginning of the macrocycle (set_macrocycle_parameters) is used to define the set of parameters that are refined by the macrocycle. The RefineBase member variable 'nmp', which is the number of parameters that are refined by the macrocycle, must be set by this function. If parameter selection is required by the user's refinement protocol, it will be necessary to keep an internal record of the list of parameters and the selection within the Refine object. Note that changing the set of function parameters refined between macrocycles, via set_macrocycle_protocol, will require a concomitant change in the functions that return or accept arrays depending on the set of function parameters refined between macrocycles (Table 1).

Example 1

The twisted Gaussian function was used as a development test case for *scitbx.lbfgs* and is implemented in the script *scitbx/lbfgs/dev/twisted_gaussian.py*. This script performs minimizations starting from 100 random start points, both with and without the use of curvatures to prime the Hessian approximation. As a reference implementation of the *dtmin,* we provide the script *scitbx/dtmin/twisted_gaussian.py*, which implements the same twisted Gaussian function minimizations. Although these two scripts minimize the same target function, differences in implementation will give different results. The

Table 1: Description of the functions, their arguments and return types and a summary of each function. Lists with an asterisk (*) are required to be the length of the number of refined parameters in the macrocycle (nmp) which is set in the function set_macrocycle_protocol. The array (Hessian matrix) with double asterisk (**) is required to be of length (nmp squared). Functions that return or accept arrays depending on nmp are indicated with a dagger (†).

Function	Arguments	Return type	Description	Default
Compulsory Functions				
target	None	Float	Target function (lower is better). Includes restraint terms (if any).	Raise NotImplementedError
get_macrocycle_parameters†	None	List of Float*	Get the parameters being refined this macrocycle	Raise NotImplementedError
set_macrocycle_parameters†	List of Float*	None	Set the current values of the macrocycle parameters	Raise NotImplementedError
macrocycle_large_shifts†	None	List of Float*	Array of large shift values for the parameters being refined this macrocycle	Raise NotImplementedError
set_macrocycle_protocol	List of String	None	Sets up the refine object for the current macrocycle.	Raise NotImplementedError
macrocycle_parameter_names †	None	List of String*	Names of the parameters being refined this macrocycle	Raise NotImplementedError
Optional Functions				
target_gradient†	None	Float, List of Float*	Target and gradient for the function to be minimized	Finite difference gradient
target_gradient_hessian†	None	Float, List of Float*, Array of Float**, Bool	Target, Gradient and Hessian of the parameters being refined in the current macrocycle. Also a bool to indicate whether the Hessian is diagonal or not so that the minimizer can do a simplified inverse calculation	Finite difference gradient. Hessian whose diagonals are the reciprocal of the square of the large shifts of the parameters
bounds†	None	List of 'Bounds' objects*	Bounds (minimum and/or maximum or no bounds) of each parameter being refined this macrocycle.	No bounds
reparameterize†	None	List of 'Reparams' objects*	Flags and offset (if true) for reparameterization of the parameters being refined this macrocycle	No reparameterization
reject_outliers	None	None	Flag data points for exclusion from the target function calculations this macrocycle	No outlier rejection
setup	None	None	Any preparation of the Refine Class prior to minimization	None
cleanup	None	None	Any reconfiguration of Refine Class between macrocycles	None
finalize	None	None	Any finalization of the Refine Class before exit	None
maximum_distance_special†	Float	List of Float*, List of Float*, List of Float*, List of Bool*, Float	Specialist function for restricting the line-search distance when there are correlated parameters	None
Logging Functions				
initial_statistics	None	None	Report initial statistics	None
current_statistics	None	None	Report current statistics	None
final_statistics	None	None	Report final statistics	None

example script is implemented to minimize the function with respect to all parameters in each macrocycle. Therefore, the set_macrocycle_protocol function does not perform any parameter selection.

The major architectural difference between *scitbx.lbfgs* and *dtmin* is that *dtmin* performs the overall refinement to convergence (with a configurable strategy), whereas in *scitbx.lbfgs,* the overall refinement strategy is not implemented in the library. The *scitbx* implementation of the twisted Gaussian minimization consists of the definition of the target function, gradients and curvatures, and then calls to steps in the *lbfgs* minimizer, such as requests_f_and_g() and requests_diag() (see scitbx documentation for more details). The *dtmin* implementation of the twisted Gaussian consists of a definition of the

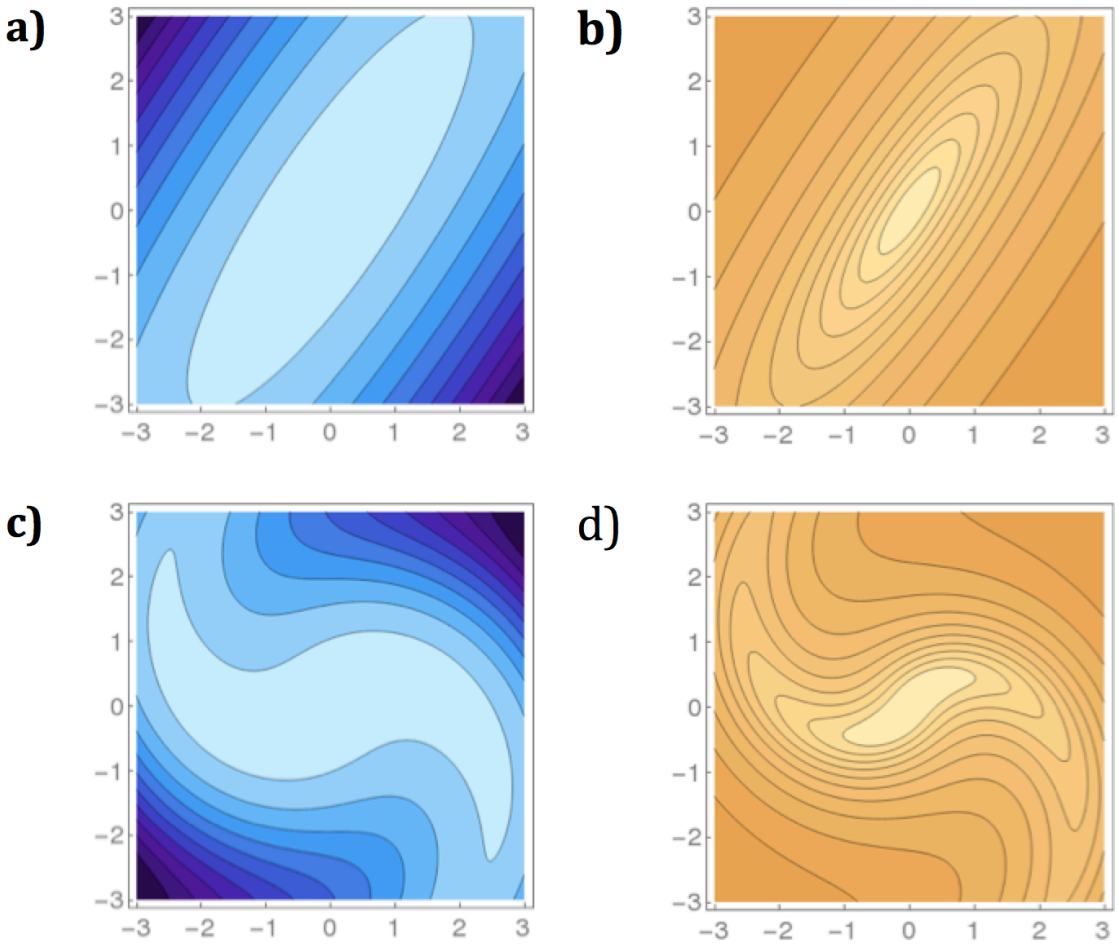


Figure 2: 'Twisted Gaussian' plotted with linear contour levels in (a) and (c) and with log-log scaled contour levels to accentuate the visualization of the gradient near the minimum in (b) and (d). (a) and (b) The functional form is the negative of the log of a bivariate Gaussian with covariate matrix entries s11=1.0, s12= 1.2, s22=2.0 and the twist parameter *t*=0, i.e. 'untwisted' Gaussian. (c) and (d) as in (a) and (b) with the twist parameter *t*=0.5.

refinement class (RefineTG, derived from RefineBase as described above) and one call to the Minimizer class 'run' function to perform the minimization to convergence. In the example, *dtmin* is configured to run with two macrocycles.

Figure 2 shows the twisted Gaussian function (details below) with two different degrees of twist, one with no twist (untwisted) and the other with a non-zero twist parameter (*t=0.5*). Figures 3 and 4 show the minimization paths taken by different minimizers in the case of the untwisted and twisted Gaussian, respectively, starting at three different positions. For the untwisted Gaussian, the path to the minimum takes fewest steps when the full Hessian is used and can be reached in one step with either the Newton (3j) or BFGS (3h) minimizer. Note that the minimization path is the same for these two because the function is quadratic. For the twisted Gaussian, the BFGS minimizer takes fewer steps than the Newton minimizer, although for one starting position (2,2) and the Hessian primed with the curvatures (4g), the minimizer does not reach the minimum in the two macrocycles of the protocol. In figure (4b) the starting position (2,-2) fails to refine using *scitbx.lbfgs* because the calculation generates a negative element in the diagonals of the inverse Hessian; the comparable test with the *dtmin* BFGS algorithm in figure (4g) succeeds because *dtmin* uses a heuristic involving the large shift values to repair negative curvatures. Newton

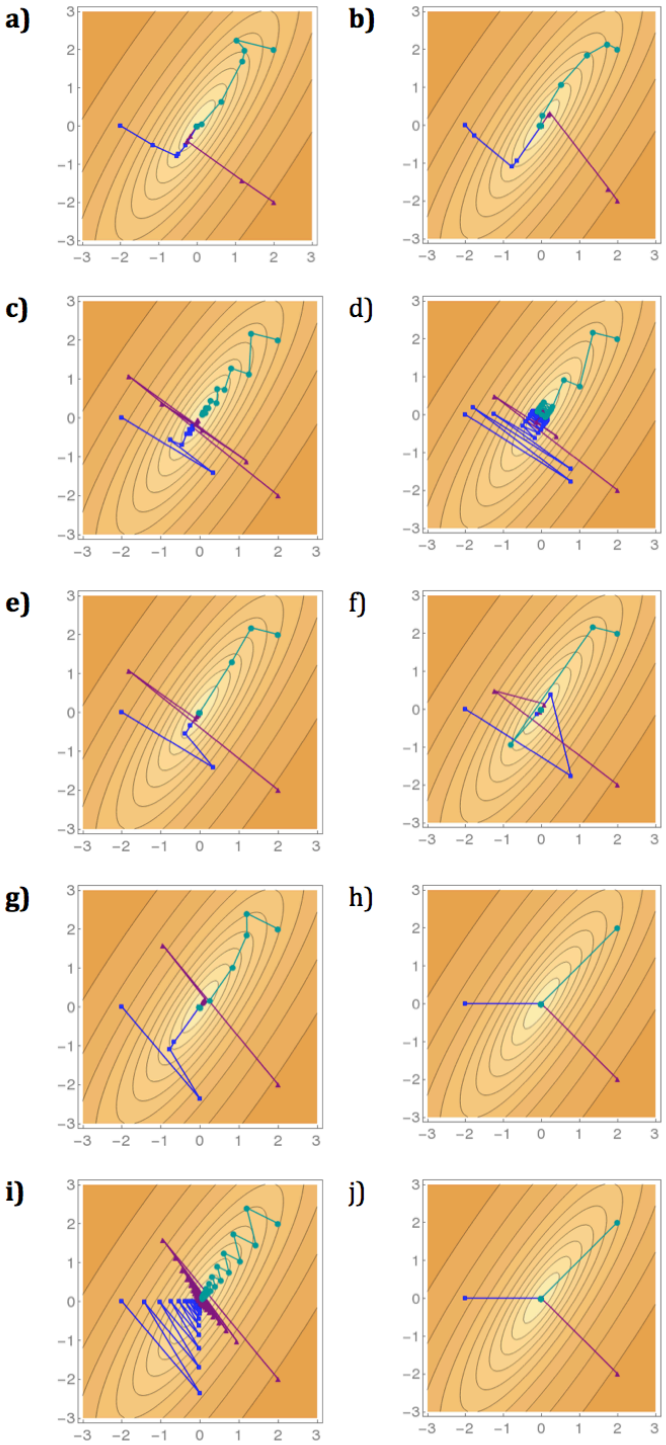


Figure 3: Path to minimum taken by different minimization methods for the 'untwisted' Gaussian in Figure 2 starting at three different positions: (2,2) in cyan, (2,-2) in purple and (-2,0) in blue. (a) and (b). a) *scitbx.lbfgs* minimizer without curvatures b) *scitbx.lbfgs* minimizer with curvatures c) *dtmin* Newton minimizer with the Hessian set to the identity matrix d) *dtmin* Newton minimizer with the diagonals of the Hessian set to the reciprocal of the square of the large shift value for each parameter e) *dtmin* BFGS minimizer with the Hessian set to the identify matrix f) *dtmin* BFGS minimizer with the diagonals of the Hessian set to the reciprocal of the square of the large shift value for each parameter g) dtmin BFGS minimizer with the diagonals of the Hessian set to analytical curvatures h) *dtmin* BFGS minimizer with the full Hessian i) *dtmin* Newton minimizer with the diagonals of the Hessian set to analytical curvatures j) *dtmin* Newton minimizer with the full Hessian.

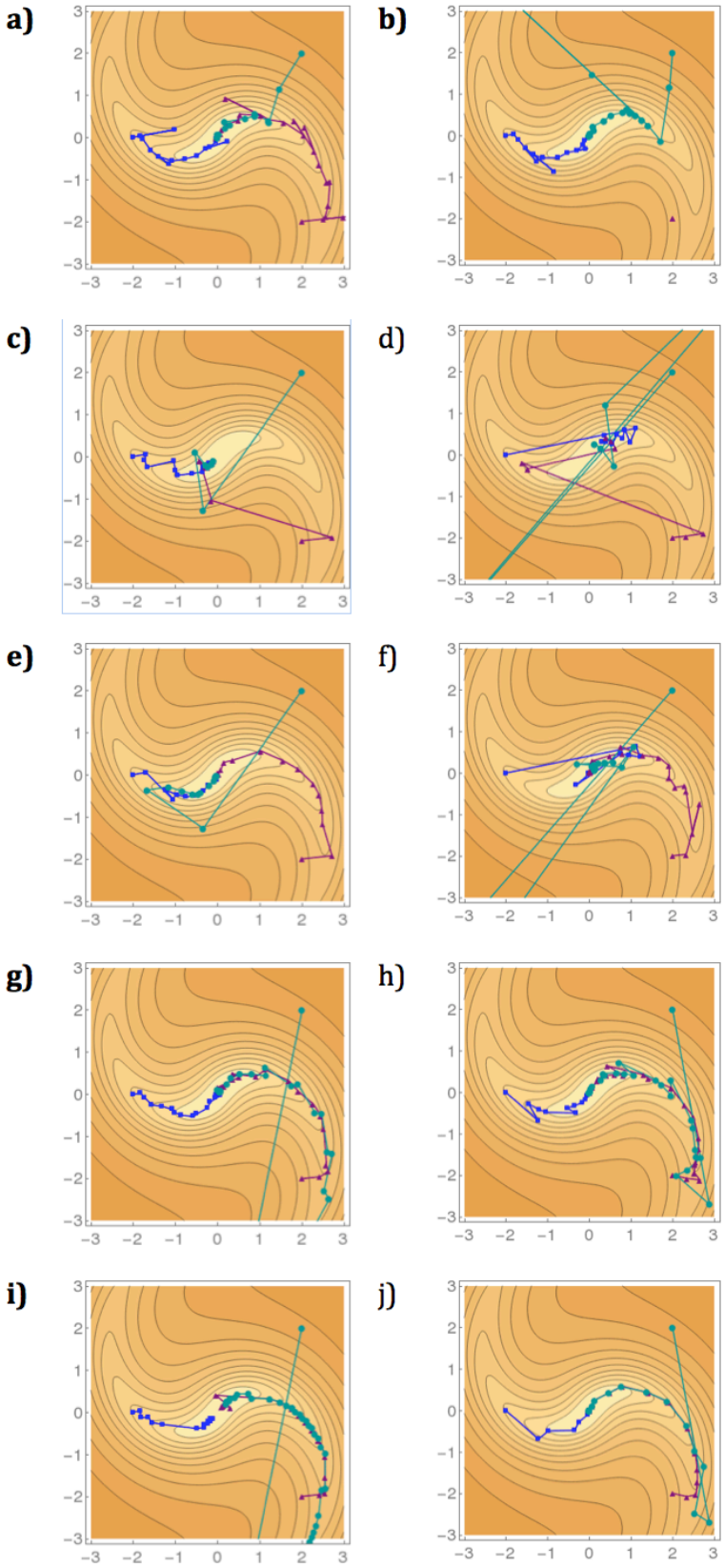


Figure 4: Path to minimum taken by different minimization methods for the 'twisted Gaussian in Figure 2 (c) and (d) starting at three different positions: (2,2) in cyan, (2,-2) in purple and (-2,0) in blue. Panels as in Figure 3.

minimization with the Hessian set to the identity matrix is equivalent to steepest descent. The appropriate minimizer and minimization protocol to use in any given case will depend on the properties of the function to be minimized and its parameterization.

The plots in Figures 2, 3 and 4 can be generated using scripts available at phaserwiki.

The functional form of the twisted Gaussian is as follows:

$$\mathrm{TG}(x,y) = -\log\left(\frac{1}{\sqrt{4\pi(s_{11}s_{22}-s_{12}^2)}}\exp\left(-\frac{s_{22}x_t^2-2s_{12}x_ty_t+s_{11}y_t^2}{2(s_{11}s_{22}-s_{12}^2)}\right)\right)$$

$$\text{where } \begin{bmatrix}x_t\\y_t\end{bmatrix} = \begin{bmatrix}\cos(\theta) & -\sin(\theta)\\ \sin(\theta) & \cos(\theta)\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix}, \text{and } \theta = t\sqrt{x^2+y^2}$$

In Figures 3 and 4 the large shift values for both *x* and *y* were two while two macrocycles of minimization were performed, both for all parameters (*i.e. x* and *y*).

Example 2

A second example script is provided in *scitbx/dtmin/regression/tst_dtmin_basic.py*. This script is a minimal template script that minimizes a quadratic and is intended for copying and editing. The architecture of the file is shown in Figure 5.

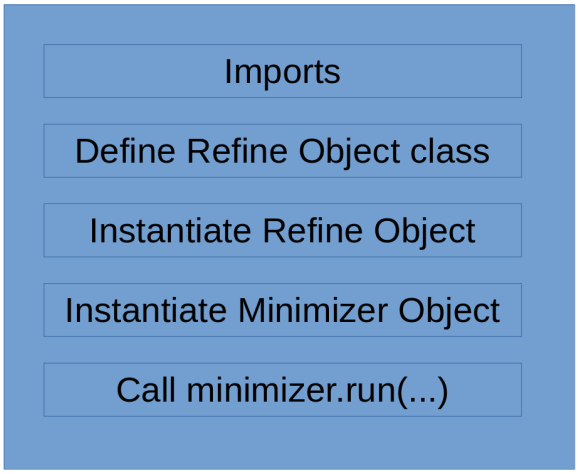


Figure 5: Architecture of a minimal python script for application of mintbx to a minimization problem. At the top of the RefineBase and Minimizer classes are imported. The function to be minimized is implemented in a 'Refine' class that inherits from RefineBase; compulsory and optional functions are described in Table 1. The derived 'Refine' class object is instantiated and passed to the Minimizer object, along with parameters to control the refinement protocol. The minimization is performed with a call to the Minimizer's run() function, and the 'Refine' object left in the minimized state.

References

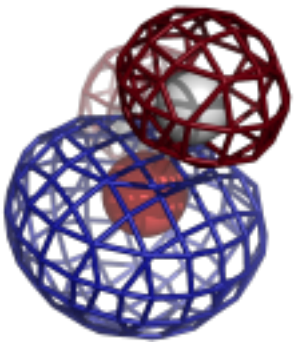
McCoy, A. J. (2004). *Acta Crystallogr. D*. **60**, 2169–2183.

McCoy, A. J., Read, R. J., Bunkóczi, G. & Oeffner, R. D. (2009).

Phaserwiki, http://www.phaser.cimr.cam.ac.uk.

Wolfram Research Inc. (2019). https://www.wolfram.com/mathematica.

VOLUME TWELVE



COMPUTATIONAL CRYSTALLOGRAPHY NEWSLETTER

JULY MMXXI

PDB@50, cis-PRO, top2018, chiral validation

Table of Contents

Editor

Nigel W. Moriarty, NWMoriarty@LBL.Gov

Editor's Note

Some of you may be aware of the recent announcement that the Protein Data Bank (PDB) is extending the length of the codes for the PDB entries from four to eight, and for Chemical Component Dictionary (CCD) entries from three to four. Read the news release here. Some may also be aware of an Editor's Note from July 2015 promoting the use of "human readable" formatting for codes. In short, it suggested using only the appropriate case for the letter o, i and L (see table A for examples).

Table A: Examples of human readable PDB codes compared with standard representations.

Standard		Human readable	
Uppercase	Lowercase	Uppercase	Lowercase
1OI0	1oi0	1oi0	1oi0
1IJJ	1ijj	1iJJ	1ijj
4OCL	4ocl	4oCL	4ocL
5SS2	5ss2	5ss2	5ss2

A small addendum to the original specification appears in the last line. The uppercase letter S (nineteenth letter of the alphabet) can be confused with the numeral 5 (five).

So, the appropriate case for the four letter is o, i, L and s. Interestingly, this provides a mnemonic – Lois – that is easy to remember and provides the correct case for each letter.

Alternatively, one could always use lowercase for the codes with the one exception for "L", which should always be uppercase. This approach has the added advantage of

lowering the ambiguity of seeing a code and guessing whether it is uppercase or lowercase.

Phenix News

Announcements

New Phenix Release Imminent

Developers are working on a Python3.7 version of Phenix. This version will contain many new features. In the meantime, nightly builds are available by contacting the download email.

Please note that the latest publication should be used to cite the use of Phenix:

Macromolecular structure determination using X-rays, neutrons and electrons: recent developments in Phenix. Liebschner D, Afonine PV, Baker ML, Bunkóczi G, Chen VB, Croll TI, Hintze B, Hung LW, Jain S, McCoy AJ, Moriarty NW, Oeffner RD, Poon BK, Prisant MG, Read RJ, Richardson JS, Richardson DC, Sammito MD, Sobolev OV, Stockwell DH, Terwilliger TC, Urzhumtsev AG, Videau LL, Williams CJ, Adams PD: Acta Cryst. (2019). D75, 861-877.

Downloads, documentation and changes are available at phenix-online.org

Expert advice

Fitting Tip #21 – What are chiral outliers and what can I do about them?

Jane Richardson and Christopher Williams, Duke University

Chirality (or handedness) is an important and pervasive feature of biology. Your hands, of course, are handed, and so are macromolecules. Proteins are made of chiral L-amino acids – a property that is manifested at larger scale in righthanded α-helices and twisted β-sheets. Nucleic acids are made of handed nucleotide components with each form of DNA or RNA double helix having a specific handedness.

Handedness reversals are very rarely seen in macromolecular structures because they are disallowed by the geometry libraries used in model-building software and are very difficult for refinement to change. Backward chirality at the Cα is already detected by extremely large Cβ deviations (although that measure does not test other chiral centers). However, we have recently encountered a few chirality outliers in deposited or in-process models, and have implemented chirality validation in Phenix and MolProbity (Prisant 2020). If any chiral, pseudo-chiral or tetrahedral-geometry outliers occur in a model, they are now noted in the summary report and are listed individually in a Phenix validation GUI table or in a MolProbity text report. They are flagged in yellow on the 3D structure in the MolProbity "multi-kin" kinemage graphics, as seen in the icon above and in most of the figures below.

In pure geometry, the choice of chirality is a binary, plus-or-minus property, but to allow for molecular flexibility and for convenient programming it is usually measured by "chiral volume": volume of the tetrahedron enclosed by the central atom and the three attached atoms of highest chemical priority (for biological macromolecules, the lowest priority atom is almost always a hydrogen). Therefore, intermediate changes in the chiral volume measure can usefully detect serious

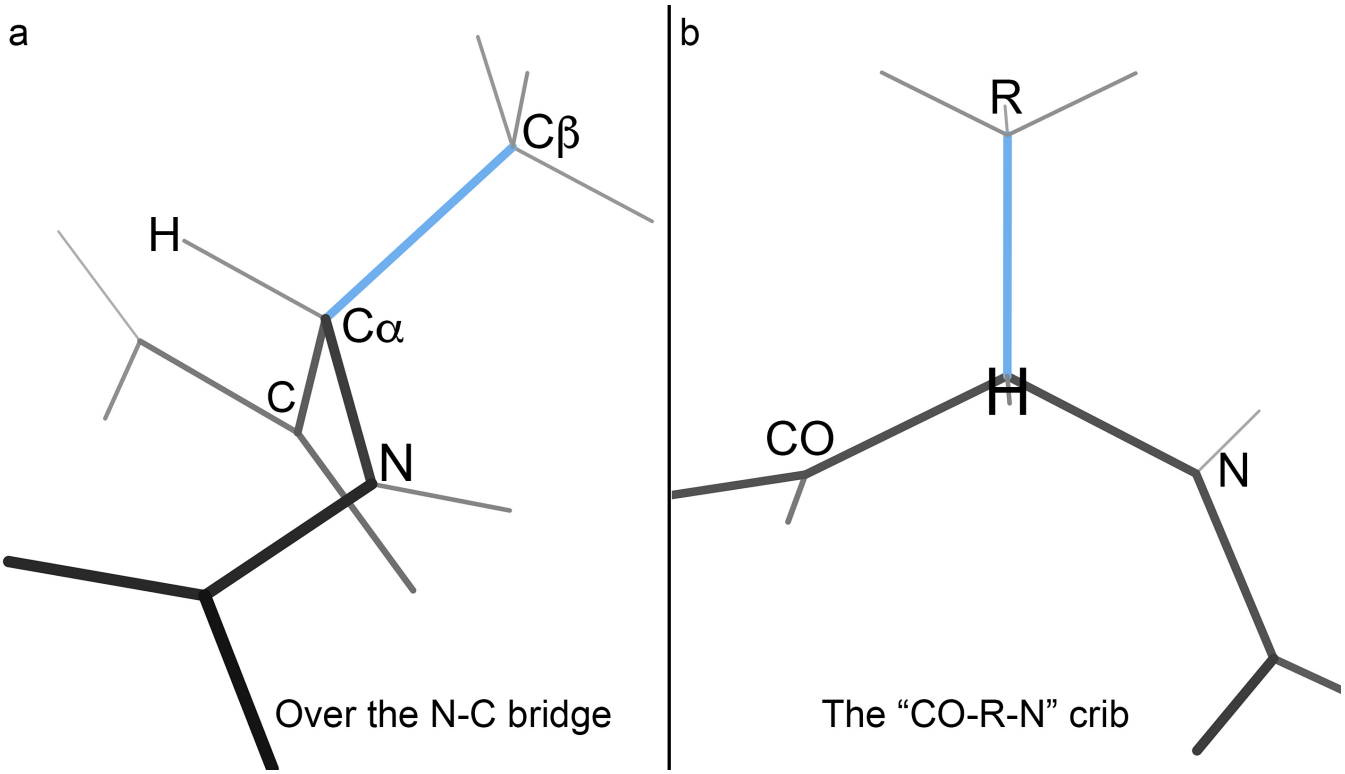


Figure 1: Mnemonics for identifying the normal L-amino acid handedness at a Cα. a) Turn the model or graphics to look from the N-terminal direction across the backbone "bridge" from N to C; the sidechain should be on your right b) Turn to look down on the Cα from the Hα; the substituents should read CO, R (the sidechain), N in the clockwise direction.

distortions of tetrahedral geometry, which are more common than chirality errors.

Definitions

A true chiral atom makes covalent bonds to four distinct atom types or branches. Common cases are:

• The protein Cα atom, bonded to the backbone N, backbone carbonyl C, sidechain Cβ, and H (Hα). Figure 1 illustrates two different mnemonics to help you distinguish normal L-amino acids from the chiral opposite D-amino acids.

• The Cβ atom of Ile or Thr, bonded to the Cα, the Hβ, the Cγ1 or Oγ1 of the long or heavy sidechain branch, and the Cγ2 methyl of the shorter branch.

• The nucleic acid C1' atom, bonded to C2' and O4'of the sugar ring, the C1' H, and the N1/N9 of the base (the other positions of substituents on the puckered sugar ring are also chiral - the C3', C5' and for RNA the C2').

• Carbohydrates are even richer in chiral centers, as are many enzyme substrates and other ligands.

A pseudo-chiral atom makes tetrahedral bonds to two distinct and two identical atoms or branches. The two identical ones are distinguished in name only, by an arbitrary consensus label (usually a number, such as Hb2 vs Hb3). Examples are:

• The Cβ of Val, with bonds to the Cα, the Hβ and the two identical Cγ methyls, which by pre-established chemical convention are labeled as Cg1 for the right-arm position and Cg2 for the left arm. Confusingly, that convention makes Val χ1 values differ in

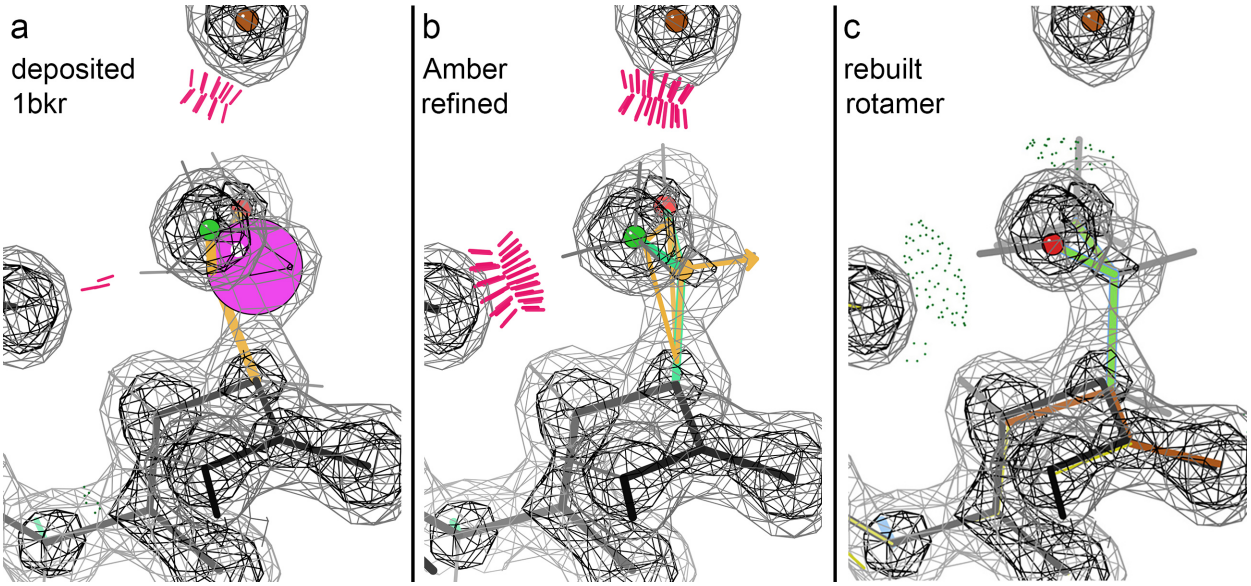


Figure 2: A real chiral outlier at a Thr. a) 1bkr as deposited, with a backward-fit rotamer and huge Cβ deviation (magenta ball, with ideal position at its center and observed position on its surface). b) Amber refinement moves all 3 sidechain non-H atoms into some density peak at the cost of reversed chirality at Cβ. c) The correct, outlier-free answer is just a different rotamer, with Oγ (red ball) in the higher peak and H-bonded.

backbone relationship from those for Thr and Ile.

• The nucleic acid P atom, with bonds to the backbone O5' and O3' and to the identical O1P and O2P atoms.

• Complexly connected het groups such as FeS clusters.

Categories of outlier cases

Chiral outliers are reported in three categories: chiral outliers, tetrahedral geometry outliers, and pseudo-chiral outliers, covering all chiral centers or tetrahedral centers defined in the Phenix geostd or monomer_library dictionaries.

True reversals of chirality can occur in software systems that do not include chirality among their geometrical restraints. This is true, for instance, in the otherwise-excellent Amber force-field refinement available in Phenix (Moriarty 2020). Figure 2a illustrates Thr 101 in 1bkr at 1.1Å resolution (Banuelos 1998), where the backward-fit sidechain places Oγ and methyl Cγ in the wrong density peaks and the Cβ far out of density; this produces clashes, a rotamer outlier, very bad covalent angles and a huge Cβ deviation (magenta sphere). Pure downhill refinement with the Amber force field moved all 3 non-H atoms into their density peaks by allowing the chirality around Cβ to reverse (Figure 2b), flagged as a yellow chirality outlier in current validation. The correct fix is to refit the sidechain rotamer, as shown in panel c, now with 2 H-bonds, no outliers and good density fit even before further refinement.

Apparent chiral outliers can occur because the group is misnamed in its 3-letter code. If an alanine D-amino-acid is called ALA rather than DAL, or a normal ALA is called DAL as in Figure 3, then MolProbity will also produce a graphical markup with an arrow to where the central tetrahedral atom of a DAL would be positioned, relative to the C, N, and Cβ atoms. In this case the fix is simple: just assign the correct residue name.

Tetrahedral-geometry outliers have chiral volumes more than 4σ different from the ideal chiral volume of the group involved. For

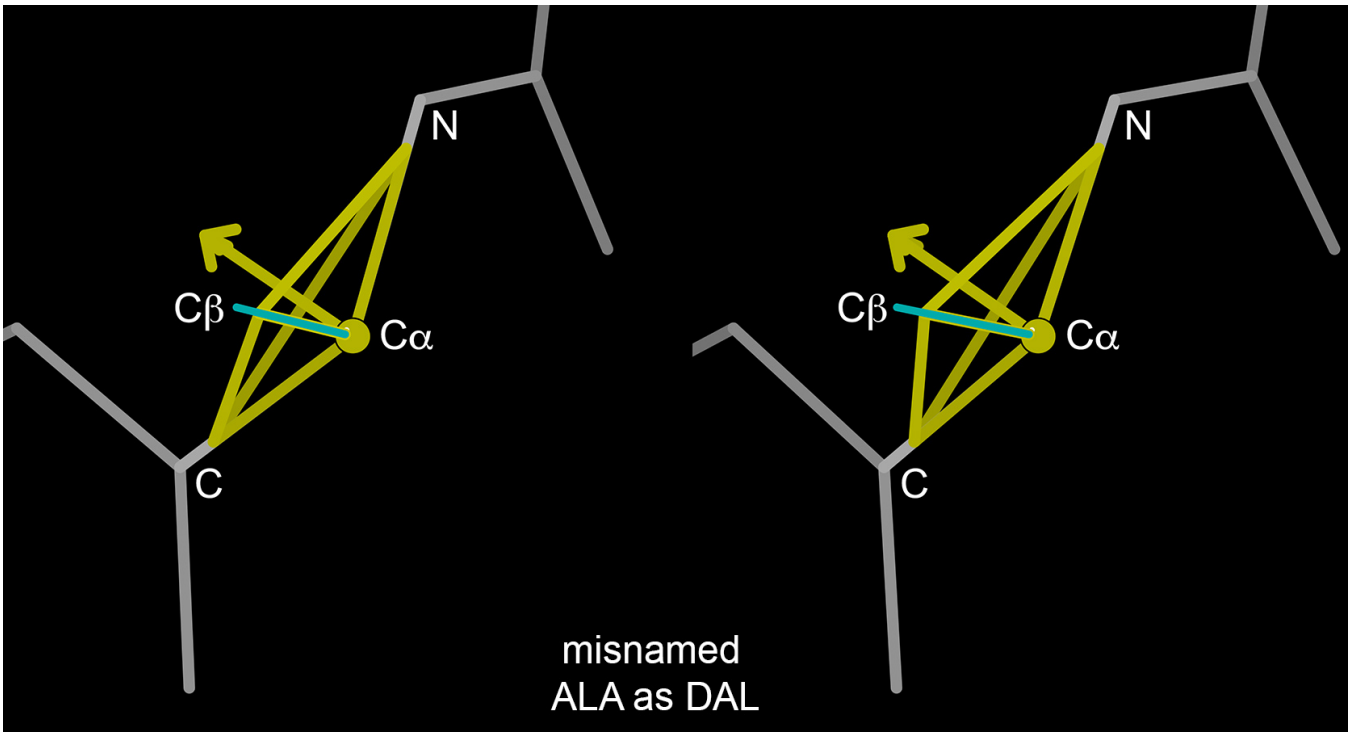


Figure 3: An apparent chiral outlier (yellow markup) caused by incorrect naming of the residue's 3-letter code. An actual ALA L-amino acid residue in a helix has been named as DAL (D-amino acid). In an actual DAL, the Cα would lie at the end of the arrow.

instance, Figure 4 shows Leu 995 in 3ogv at 1.4Å resolution, which is so far from tetrahedral that it is nearly planar. It is flagged with a similar yellow markup, but without the arrow. Bond-length and bond-angle outliers also show that there is a problem, but the chiral outlier more clearly indicates the problem: The Cβ should move left somewhat and the Cγ should move right, to fit the density better and provide stronger chirality, which could then be refined successfully. The Leu rotamer was presumably fit backward originally (with the Cγ back and left rather than forward and right), and the density pulled it almost flat during refinement.

Pseudo-chiral outliers are always caused by failure to name, or number, the two identical substituents in accordance with standard conventions. An easily understandable case would be switching Cg1 and Cg2 labels in a valine sidechain but with Cγ and Cβ atoms in the correct places and density peaks. An example is shown in Figure 5.

Atom naming issues do not matter in many ways, but they cause problems with identifying dihedral angles, superimposing related structures, and similar functionalities. Since they are trivial to fix, that should always be done. Look up the wwPDB naming conventions (by 3-letter code) for the particular group, and follow them.

These and the other chiral categories are reported individually by MolProbity in a chirals.txt report such as shown in Figure 6 for a file deliberately messed up to include all three types of chiral outliers.

Discussion

This chiral outlier validation does not flag naming errors among multiple H atoms (only between heavier atoms or between them and

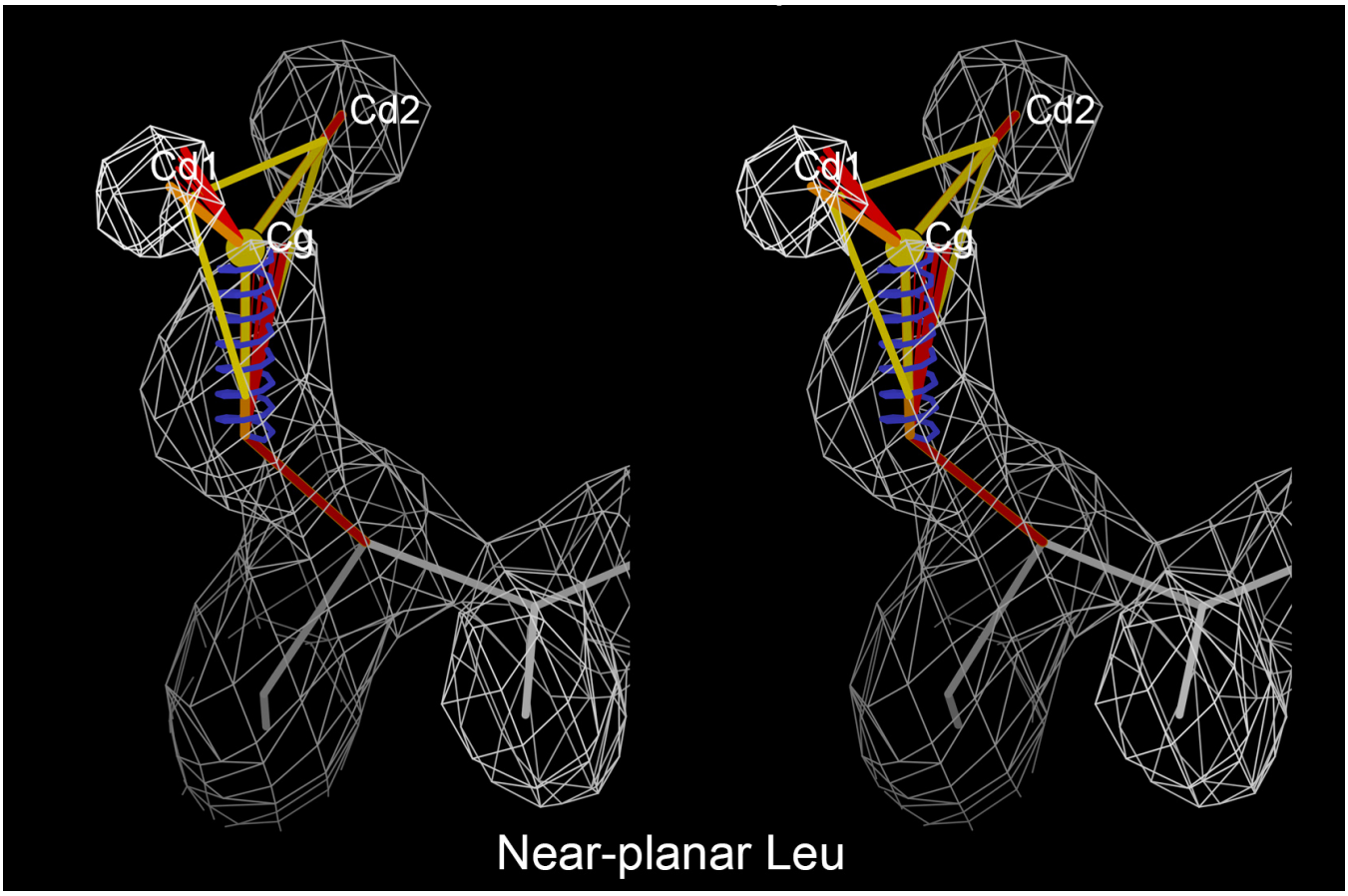


Figure 4: An extreme tetrahedral-geometry outlier at a Leu Cg. Presumably, the original fit in a 180°-opposite non-rotamer fights in refinement with fit to the density, producing this nearly flat tetrahedral group. This distortion also shows very large bond-angle outliers (red fans).

an H). Modern software should provide accurate H names, but you might encounter such errors in older files, since the conventions changed very thoroughly when the wwPDB moved from version 2.7 to v3.0 format nearly 10 years ago. For example, the hydrogens on a methylene were previously numbered 1 and 2, but are now 2 and 3. Specifically, at Cβ the continuing heavier-atom

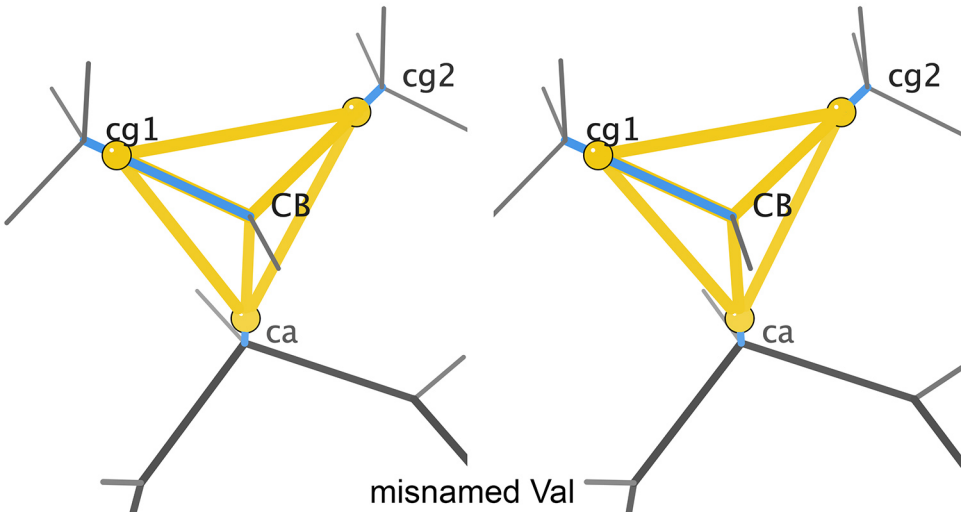


Figure 5: A pseudo-chiral atom-naming problem. The two branches of a Val sidechain are identical methyl groups, but the atoms need unique names. By pre-existing chemical convention, the righthand arm should be labeled as branch 1, but here the names cg1 and cg2 are assigned backward (cg1 on the left-hand branch) and therefore are flagged as a pseudo-chiral naming error.

branch (Cγ) is now considered #1, and Hb2 and Hb3 are named successively in the clockwise direction looking out the sidechain. If you ever need it, MolProbity still includes a utility for converting v2.7 to v3.0 format.

```
SUMMARY: 3 total outliers at 242 tetrahedral centers (1.24%)
SUMMARY: 1 handedness outliers at 218 chiral centers (0.46%)
SUMMARY: 1 tetrahedral geometry outliers
SUMMARY: 1 pseudochiral naming errors


Handedness swaps
-----------------------------------------------------------------------------
 A:  15 ::DAL:CA:25.12
-----------------------------------------------------------------------------

Tetrahedral geometry outliers
-----------------------------------------------------------------------------
 A:  25 ::ILE:CB:5.38
-----------------------------------------------------------------------------

Probable atom naming errors around pseudochiral centers
  e.g. CG1 and CG2 around Valine CB
-----------------------------------------------------------------------------
 A:   9 ::VAL:CB:26.59
-----------------------------------------------------------------------------
```

Figure 6: A chirals.txt report for an artificially constructed file with all 3 types of chiral outliers: handedness swaps, tetrahedral geometry, and pseudo-chiral naming.

Chiral problems in ligands, modified residues, and especially carbohydrates can happen in good, modern structures. Figure 7 shows a true chiral outlier at the C15 branchpoint of the YG 37 base in the anti-codon loop of the 1ehz tRNA at 1.93Å resolution. It is in weak, patchy density, but it would be preferable to model the correct enantiomer. In complex carbohydrates, a chiral or pseudo-chiral outlier may often mean that either the wrong sugar or the wrong linkage type has been modeled. In Phenix, the carbohydrate libraries were recently re-analyzed and updated. The wwPDB now has much better carbohydrate validation available at deposition and has recently finished remediating carbohydrates in previous deposits.

As macromolecular structural biologists, we should all be grateful for the voluminous libraries of chemical and conformational

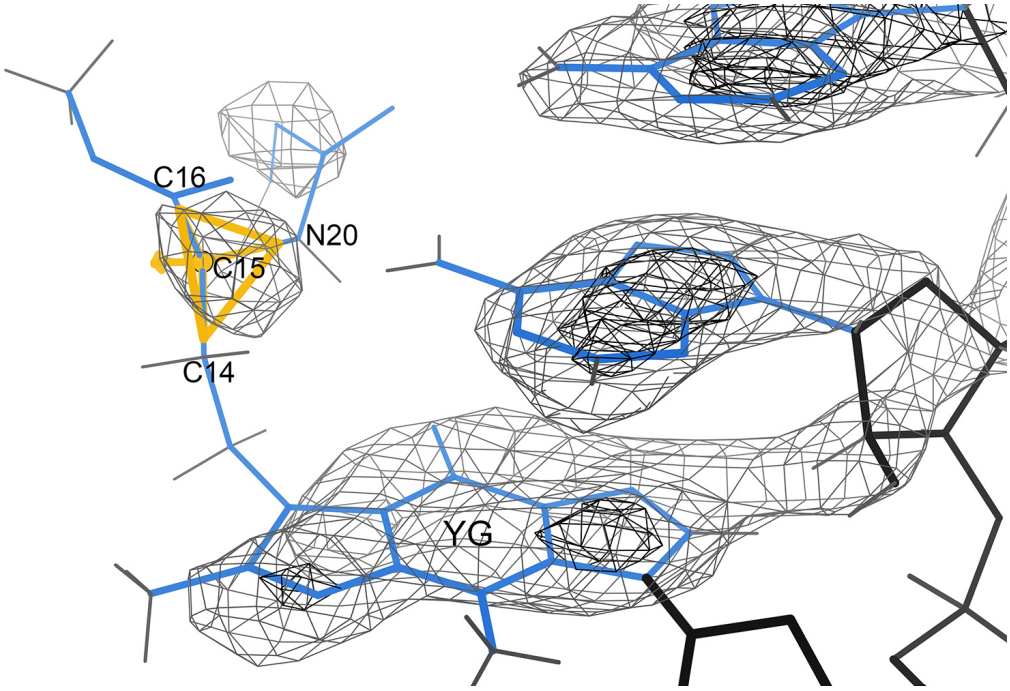


Figure 7: A true chiral outlier (yellow markup) at the C15 branchpoint in the YG 37 modified base of the 1.93Å 1ehz tRNA structure (Shi & Moore 2000). The local electron density suggests substantial disorder, but the two branches are different lengths with different atom types, so that in the other handedness there would be different possibilities for H-bonding with the neighboring bases.

constraints and especially to the work of chemistry, computation, and small-molecule crystallography that made those libraries possible. They are not infallible, but only rarely is the problem their fault.

The bottom line

Chiral outliers, and even pseudo-chiral outliers, occur very rarely when using modern software but are well worth flagging, understanding, and fixing when they do. Naming will of course be fixed by the annotators when you deposit your structure, but it is cleaner and more polite to do it yourself. Tetrahedral-geometry outliers are much more common and serious, and are flagged by the same chiral volume formalism. They almost always signal that the local group has been fit in the wrong conformation, which definitely should be rebuilt before final refinement.

References:

Banuelos S, Saraste M, Carugo KD (1998) Structural comparisons of calponin homology domains: implications for actin binding, *Structure* **6**: 1419-1431

Maksimainen M, Hakulinen N, Kallio JM, Timoharju T, Turunen O, Rouvinen J (2011) Crystal structures of Trichoderna reesii beta-galactosidase reveal conformational changes in the active site, *J Struct Biol* **174**: 156-163

Prisant MG, Williams CJ, Chen VB, Richardson JS, Richardson DC (2020) New tools in MolProbity validation: CaBLAM for cryoEM backbone, UnDowser to rethink "waters", and NGL Viewer to recapture online graphics *Prot Science* **29**: 315-329

Moriarty NW, Janowski PA, Swails JM, Nguyen H, Richardson JS, Case DA, Adams PD (2020) Improved chemistry restraints for crystallographic refinement by integrating the Amber force field into Phenix, *Acta Crystallogr* **D76**: 51-62

Shi H, Moore PB (2000) The crystal structure of yeast phenylalanine tRNA at 1.93Å: a classic structure revisited, *RNA* **6**: 1091-1105

FAQ

Can I have an angle restraint involving symmetry atoms?

The short answer is no. The main reason is that applying a symmetry operator is ambiguous in the case of three atoms compared to two atoms needed for a bond.

Updates from the Worldwide PDB: Celebrating PDB50 and PDBx/mmCIF news

Christine Zardecki
RCSB Protein Data Bank

Correspondence email: Zardecki@rcsb.rutgers.edu

Celebrating the 50th Anniversary of the Protein Data Bank

In 1971, the structural biology community established the single worldwide archive for macromolecular structure data–the Protein Data Bank (PDB). From its inception, the PDB has embraced a culture of open access, leading to its widespread use by the research community. PDB data are used by hundreds of data resources and millions of users exploring fundamental biology, energy, and biomedicine.

To commemorate and celebrate 50 years of the PDB, the wwPDB is organizing multiple events in 2021 (wwpdb.org/pdb50):

- The inaugural PDB50 event was held virtually in May 2021.
- Transactions Symposium 2021: Function Follows Form: Celebrating the 50th Anniversary of the Protein Data Bank (July 30-31, 2021).
  This virtual event is part of the Annual Meeting of the American Crystallographic Association.
- Bringing Molecular Structure to Life: 50 Years of the PDB (October 20-22, 2021) Virtual EMBL Conference
- Royal Society of Chemistry PDB Workshop (Nov 16 and 18, virtual)
- Learning from 50 years of the Protein Data Bank: A satellite symposium of the Biophysical Society of Japan (Nov 25-27, 2021)

Visit wwpdb.org/pdb50 for updates and related materials.

PDBx/mmCIF News

PDB users and related software developers should be aware of upcoming developments and plans related to the distribution of PDB data. Announcements are made at wwpdb.org.

Modifications to Support for SHEET and Ligand SITE records in June 2021

In 2014, PDBx/mmCIF became the PDB's archive format and the legacy PDB file format was frozen. In addition to PDBx/mmCIF files for all entries, wwPDB produces PDB format-formatted files for entries that can be represented in this legacy file format (e.g., entries with over 99,999 atoms or with multi-character chain IDs are only available in PDBx/mmCIF)

As the size and complexity of PDB structures increases, additional limitations of the legacy PDB

format are becoming apparent and need to be addressed.

Defining complex SHEET records

Restrictions in the SHEET record fields in legacy the PDB file format do not allow for the generation of complex beta sheet topology. Complex beta sheet topologies include instances where beta strands are part of multiple beta sheets and other cases where the definition of the strands within a beta sheet cannot be presented in a linear description. For example, in PDB entry 5wln a large beta barrel structure is created from multiple copies of a single protein; within the beta sheet forming the barrel are instances of a single beta strand making contacts on one side with multiple other strands, even from different chains.

This limitation, however, is not an issue in the PDBx/mmCIF formatted file, where these complex beta sheet topology can be captured in *_struct_sheet*, *_struct_sheet_order*, *_struct_sheet_range*, and *_struct_sheet_hbond*.

Starting June 8th, 2021, legacy PDB format files will no longer be generated for PDB entries where the SHEET topology cannot be generated. For these structures, wwPDB will continue to provide secondary structure information with helix and sheet information in the PDBx/mmCIF formatted file.

Deprecation of _struct_site (SITE) records

wwPDB regularly reviews the software used during OneDep biocuration. The *_struct_site* and *_struct_site_gen* categories in PDBx/mmCIF (SITE records in the legacy PDB file format) are generated by in-house software and based purely upon distance calculations, and therefore may not reflect biological functional sites.

Starting in June 2021, the in-house legacy software which produces *_struct_site* and *_struct_site_gen* records will be retired and wwPDB will no longer generate these categories for newly-deposited PDB entries. Existing entries will be unaffected.

Consistent Format for Validation and Coordinate Data

wwPDB validation reports are now provided in PDBx/mmCIF format for all new depositions in OneDep. This change makes validation data more interoperable with the PDB archival format. Data are more logically and better organized in the PDBx/mmCIF reports, and therefore more "database-friendly" than the report in XML format. PDBx/mmCIF-format validation reports for newly released and modified entries will be distributed through the PDB and EMDB Core Archives.

The new PDBx/mmCIF reports are easier to interpret. They contain a high-level summary and offer easier access to residue-level information. Data are provided at multiple levels: entity, chain-specific, and even at the individual residues. For example, it is more straightforward to obtain the total number of clashes. The corresponding validation dictionary is available at mmcif.wwpdb.org/dictionaries/mmcif_pdbx_vrpt.dic/Index. Examples of PDBx/mmCIF validation reports for X-ray, 3DEM, and NMR are publicly available at GitHub.

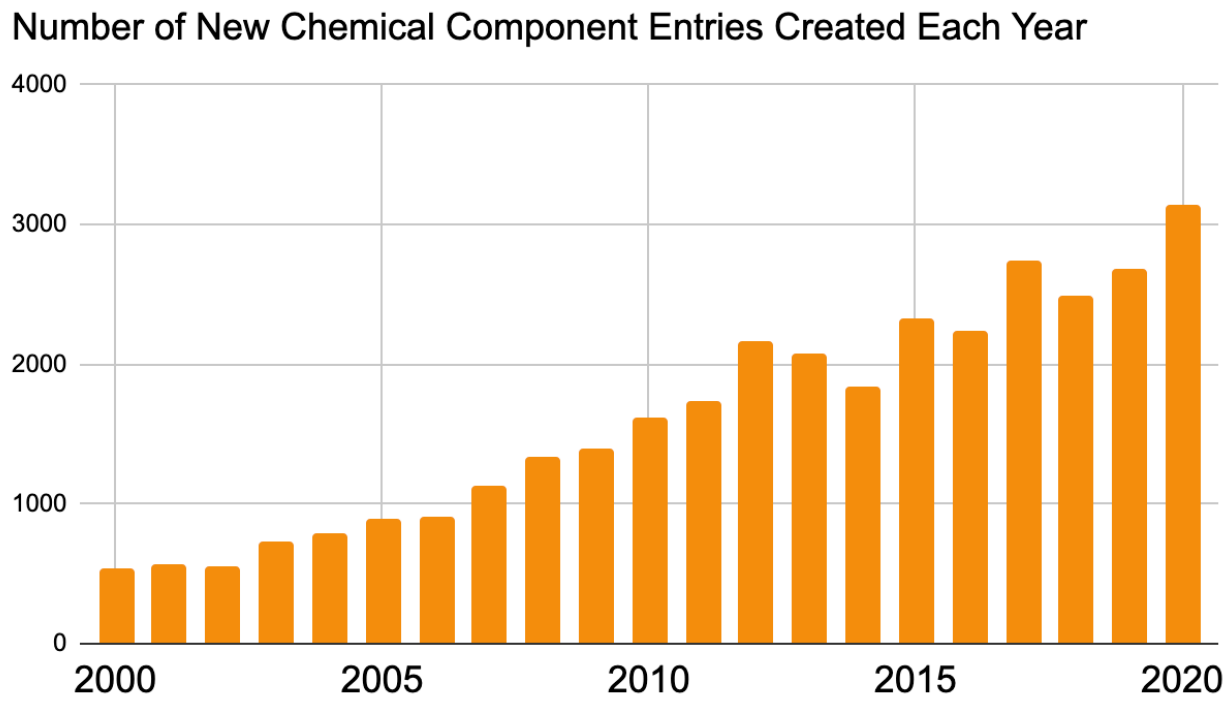
PDBx/mmCIF validation reports will be provided for the full PDB and EMDB archives once archival validation recalculation is performed.

wwPDB strongly recommends all PDB users and software developers adopt this format for future applications.

Future Planning: Entries with Extended PDB and CCD ID Codes will be Distributed in PDBx/mmCIF Format only

wwPDB, in collaboration with the PDBx/mmCIF Working Group, has set plans to extend the length of ID codes for PDB and Chemical Component Dictionary (CCD) ID entries in the future. Entries containing these extended IDs will not be supported by the legacy PDB file format.

CCD entries are currently identified by unique three-character alphanumeric codes. At current growth rates, we anticipate running out of



available new codes in the next three to four years. At this point, the wwPDB will issue four-character alphanumeric codes for CCD IDs in the OneDep system. Due to constraints of the legacy PDB file format, entries containing these new, four character ID codes will only be distributed in PDBx/mmCIF format. The wwPDB will begin implementation of extended CCD ID codes in 2022.

In addition, wwPDB also plans to extend PDB ID length to eight characters prefixed by 'PDB', e.g., pdb_00001abc. Each PDB ID has a corresponding Digital Object Identifier (DOI), often required for manuscript submission to journals and described in publications by the structure authors. Both extended PDB IDs and corresponding PDB DOIs, along with existing four character PDB IDs, will be included in the PDBx/mmCIF formatted files for all new entries by Fall 2021.

For example, PDB entry 1ABC will also have the extended PDB ID (pdb_00001abc) and the corresponding PDB DOI (10.2210/pdb1abc/pdb) listed in the _database_2 PDBx/mmCIF category.

```
loop_
_database_2.database_id
_database_2.database_code
_database_2.pdbx_database_accession
_database_2.pdbx_DOI
PDB 1abc pdb_00001abc
10.2210/pdb1abc/pdb
WWPDB D_1xxxxxxxxx ? ?
```

Once four-character PDB IDs are all consumed, newly-deposited PDB entries will only be issued extended PDB ID codes, and entries will only be distributed in PDBx/mmCIF format.

wwPDB is asking PDB users and related software developers to review code and begin to remove such limitations for the future.

Lessons from using the Cambridge Structure Database: III – Outlier rejection

Nigel W. Moriarty[a*]

[a]Molecular Biosciences and Integrated Bioimaging, Lawrence Berkeley National Laboratory, Berkeley, CA 94720

[*]Correspondence e-mail: nwmoriarty@lbl.gov

Preface

Continuing the series about lessons from using the Cambridge Structural Database (CSD), this work delves deeper into the nuances of data handling. More information about goals in the previous installment (Moriarty, 2020, 2021).

Introduction

The Cambridge Structural Database (CSD, Groom *et al.*, 2016) contains a wealth of small molecules that can be mined for geometry information. The tools in the CSD suite – Conquest (Bruno *et al.*, 2002), a structure based search tool, and Mercury (Macrae *et al.*, 2006, 2008), a data analysis tool – are flexible and highly featured making them ideal for their designated tasks.

The CSD is a curated data set leading to reliable entries. However, it is almost impossible to have consistent results from a particular search. Reasons for this may be user "error" as addressed in the first two editions of this series. More precise specification of the search structure will return the group of structures desired.

Another reason may be atypical or aberrant data in the specific entry. This could be an error that is corrupting the database or simply an example where a more nuanced search is required. Either way, it is more difficult to identify so filtering out these entries is desirable.

Outlier rejection

Thankfully, there is a technique that can help with database anomalies and user errors. Outlier rejection is the identification of outliers and removing them the analysed data. This is an active field of research with many techniques with various applications and effectiveness. In fact, the Mercury program has outlier identification.

One of the first signs of problems is an unusually large standard deviation of the bond lengths and/or bond angles. Theoretically, one can step though each entry (using Mercury) to "eyeball" for any issues but this gets tedious very quickly.

An example

Arginine is a charged essential amino acid containing the guanidinium moiety. Figure 1 shows the guanidinium group terminating the side chain with a positively charged central carbon atom and an electronic resonance bond structure. Note that the generally planar structure includes the central charged carbon atom, the three bound nitrogen atoms and the bound hydrogen atoms. A recent CSD structure search of the guanidinium group was reported as part of study (Moriarty *et al.*, 2020) into the planarity of the guanidinium group. The entries returned were analysed in a spreadsheet to enable more detailed study of features of the data. It should be noted that Mercury performs similar tasks but it also

allows easy export in helpful formats for spreadsheet programs.

The standard deviations of several geometric features were considered too large so the tedious task and stepping though each entry was undertaken. Several incongruous entries were identified including GUACET shown in figure 2 produced by Mercury. The hydrogen atoms are not in the plane of the moiety. This is hypothesis not to happen but is evident in more than one case. Regardless of whether the planarity of the hydrogen atoms is correct or an error, the other geometric features are affected. This makes these entries outliers to the hypothesised geometry of the guanidinium.

One could remove them in this "eyeball" fashion but using an outlier rejection technique a uniform, defensible and efficient process. The selected technique was Tukey's fences (Beyer, 1981) which removed all the examples discovered in the step through and a couple more that were also questionable. Based on the quadrature method, it was easy to program and gave similar results to the outlier identification in Mercury.

Conclusions

It has been a theme of this series to "Always verify that the results from a structure search are reasonable." This installment provided insights into removing the "unreasonable."

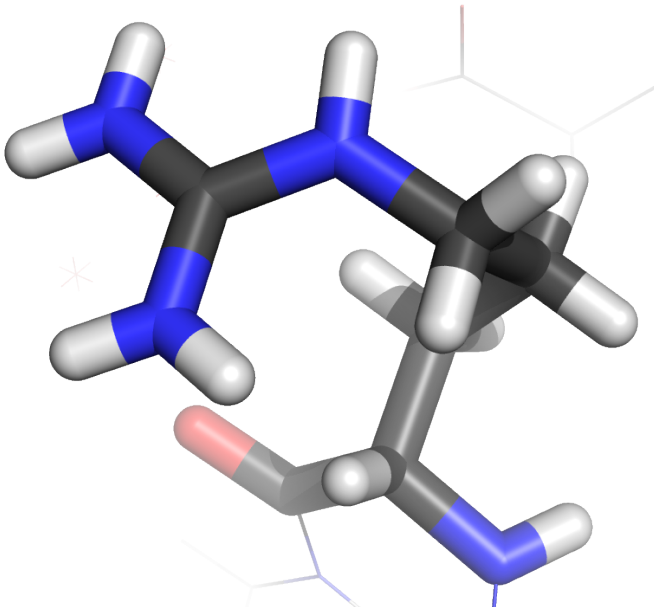


Figure 1: The arginine amino acid with the charged, planar guanidinium group in the upper left.

Coda

However, the inconsistencies of the entries removed are based on the hypothesis that the hydrogen atoms are planar. There is one entry, HOWHIK, that has out-of-plane hydrogen atoms but there is a SO_4^+ molecule that is attracting them to a far less non-planar positions than the example in figure 2. Clearly, the hydrogen atoms are affected by the nearby charge. This is an example of a more nuanced understanding of the guanidinium. Is it possible that the hydrogen atoms are more flexible? If so, by how much?

References

Beyer, H. (1981). *Biom. J.* **23**, 413–414.

Bruno, I. J., Cole, J. C., Edgington, P. R., Kessler, M., Macrae, C. F., McCabe, P., Pearson, J. & Taylor, R. (2002). *Acta Crystallogr. B*. **58**, 389–397.

Groom, C. R., Bruno, I. J., Lightfoot, M. P. & Ward, S. C. (2016). *Acta Crystallogr. Sect. B Struct. Sci. Cryst. Eng. Mater.* **72**, 171–179.

Macrae, C. F., Bruno, I. J., Chisholm, J. A., Edgington, P. R., McCabe, P., Pidcock, E., Rodriguez-Monge, L., Taylor, R., Streek, J. van de & Wood, P. A. (2008). *J. Appl. Crystallogr.* **41**, 466–470.

Macrae, C. F., Edgington, P. R., McCabe, P., Pidcock, E., Shields, G. P., Taylor, R., Towler, M. & Streek, J. van de (2006). *J. Appl. Crystallogr.* **39**, 453–457.

Moriarty, N. W. (2020). *Comput. Crystallogr. Newsl.* **11**, 7–10.

Moriarty, N. W. (2021). *Comput. Crystallogr. Newsl.* **12**, 6–8.

Moriarty, N. W., Liebschner, D., Tronrud, D. E. & Adams, P. D. (2020). *Acta Crystallogr. Sect. Struct. Biol.* **76**, 1159–1166.

The effect of adding a single peptide bond class

Nigel W. Moriarty[1] and Paul D. Adams[1,2]

[1]Molecular Biophysics and Integrated Bioimaging Division, Lawrence Berkeley National Laboratory, Berkeley, CA

[2]Department of Bioengineering, University of California at Berkeley, Berkeley, CA

Introduction

Comprehensive restraints for refinement of protein structure were introduced by Engh & Huber (1991) for the standard amino acids. Gleaned from the Cambridge Structural Database (CSD, Groom *et al.*, 2016), the group of restraints (EH91) became the standard for crystallographic refinement forming the basis of the Monomer Library (Vagin *et al.*, 2004) used in *REFMAC* (Murshudov *et al.*, 2011) and BUSTER (Bricogne *et al.*, 2011) while also being available in the *Phenix* suite of programs (Liebschner *et al.*, 2019).

Briefly, the EH91 restraints provided ideal bond lengths and angles for each of the designated standard amino acids at that time. Generally, each geometry restraint's ideal was based on the identity of amino acid. Programmatically, the three-letter code of each amino acid was the key to a dictionary of bond and angle ideal values. That is, there is a single value for each bond and angle based on the amino acid that was used for each instance of that amino acid type in the macromolecule. This paradigm can be called a Single Value Library (SVL) as the bond and angle ideal values are set once.

Engh & Huber updated the restraints (2001) for the International Tables of Crystallography, Volume F, that has been designated EH99 elsewhere. One of the major changes in the EH99 restraints from the EH91 restraints was the recognition that *cis*-proline has different ideal values for most of the bonds and angles compared to the *trans* form. The largest difference is found in the linking angle C–N–Cα that increases from 122.6° in the *trans* form restraints (which were previously used for *cis*-PRO) to 127.0° in the *cis* (Fig. 1). The estimated standard deviation (e.s.d.) was reduced from 5° to 2.4°. This is quite a large change, effectively doubling the contribution of the restraint to the final target. Other PRO restraints were changed that are purely in the amino acid entity as shown in Fig. 1 alone with others not shown. This results in the PRO restraints being based on the peptide bond form in addition to the identity of the amino acid; a small step away from the SVL paradigm. Interestingly, in neither set of restraints do the sums of the angles around the nitrogen atom add to 360°. The EH91 restraints have a sum of 359.6° compared to 359.1° for EH99; arguably, negligible compared to the e.s.d values.

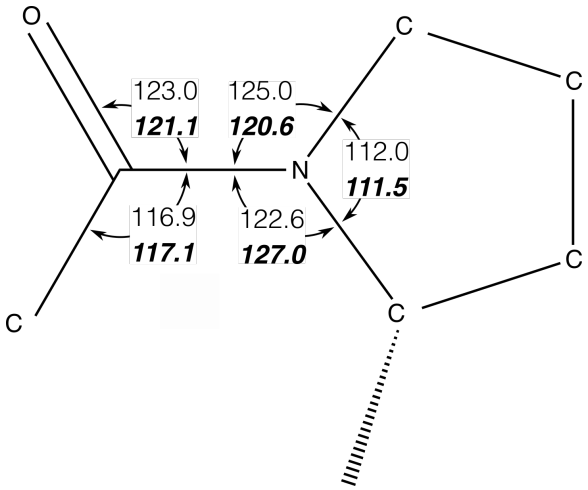


Figure 1: Diagram of a selected set of ideal angle values. EH99 values for *cis*-PRO are shown in bold italics with the EH91 values included for comparison.

More recent studies have investigated the influence of other factors on the ideal geometric values. One such study on the Conformation Dependent Library (CDL, Berkholz *et al.*, 2009) showed that the backbone geometry bond and angle ideal values depend on the ψ/ϕ angles of the backbone. The efficacy of the CDL was investigated (Moriarty *et al.*, 2014) by re-refining a large number of the entries available in the Protein Data Bank (PDB, Burley *et al.*, 2019) leading to the adoption of the CDL as the default (Moriarty *et al.*, 2016) in all *Phenix* packages. One caveat is that the CDL v1.2 is only for *trans*-peptides.

Despite the popularity of the EH91 restraints, the EH99 restraints were not implemented in the Monomer Library being absent in version 5.41. One can infer that no comprehension investigation of the influence of the EH99 *cis*-PRO restraints on protein refinement has been performed.

Approximately 5% of prolines are *cis*-PRO making the investigation into the addition of two sets of restraints for PRO nuanced.

Methods

To compare refinements using the EH99 *cis*-PRO restraints against EH91 restraints, the EH99 restraints were implemented in *Phenix* for use in all programs. Technically, the generic mechanism using the *cif_link* and *cif_mod* in the Monomer Library could have been used to add the EH99 *cis*-PRO restraints to *Phenix*, however, because of the CDL implementation there was an opportunity to implement a more flexible algorithm by using the CDL infrastructure.

To test the restraint libraries, structures were selected from the PDB using the following criteria. Entries must have untwinned experimental data available that are at least 90% complete. Each entry's R_{free} was limited to a maximum of 35%, R_{work} to 30% and the ΔR (R_{free}-R_{work}) to a minimum of 1.5%. Entries containing nucleic acids were excluded.

Each model was then subjected to 10 macrocyles of refinement using the default strategy in *phenix.refine* for reciprocal space coordinate refinement. Other options applied to both EH99 and EH91 refinements included optimization of the weight between the experimental data and the geometry restraints. This protocol was performed in parallel. The quality of the resulting models was assessed numerically using MolProbity (Williams *et al.*, 2018) available in *Phenix*. To avoid typographical ambiguity, PDB codes are given here with lower case for all letters except L (e.g., 1nLs). Post-refinement filtering removed refined models that exceeded a *clashscore* of 12.

Results & Discussion

As previously stated, the *cis* peptide link occurs in approximately 5% of prolines. This implies that the change will not be reflected in global measures like the R factors. This is indeed true. The same is true for many of the other validation metrics reported by Molprobity.

One metric reported by Molprobity and PDB alike is the root mean squared deviation (r.m.s.d.) values for the bond and angle restraints in the refined model compared to the ideal values of the restraints. Another similar metric is the r.m.s.Z values that use the e.s.d. values of the restraints to calculate the number of standard deviations from the mean – the Z-score.

Both the r.m.s.d. and r.m.s.Z values will be largely unaffected by the modified *cis*-PRO

restraints if the entire model is compared. Focusing the scope of the metrics has been demonstrated to provide validation of new restraints for iron-sulfur clusters (Moriarty & Adams, 2019) and arginine (Moriarty *et al.*, 2020). The latter has a detailed discussion of the nuances of validating single amino acid restraints as well as applying the metrics to other internal coordinates such as torsion angles.

For this case, the focus is ever tighter – just the *cis*-PRO instances in the models. Figure 2 shows the comparison of the r.m.s.d. values for the EH91 restraints denoted "122" to indicate the approximate ideal angle for C–N–Cα and EH99 denoted "127" for the new angle ideal value. The results for the entire models are shown as dashed and dotted lines but have negligible differences. Notwithstanding, the *cis*-PRO restraints (denoted "PRO 122" and "PRO 127") have significant differences. All bond r.m.s.d. values are similar at resolution worse than 2Å. At better than 2Å, the *cis*-PRO entities have smaller r.m.s.d. values. This change is not based on the new *cis*-PRO restraints as both the old and new restraints are very similar.

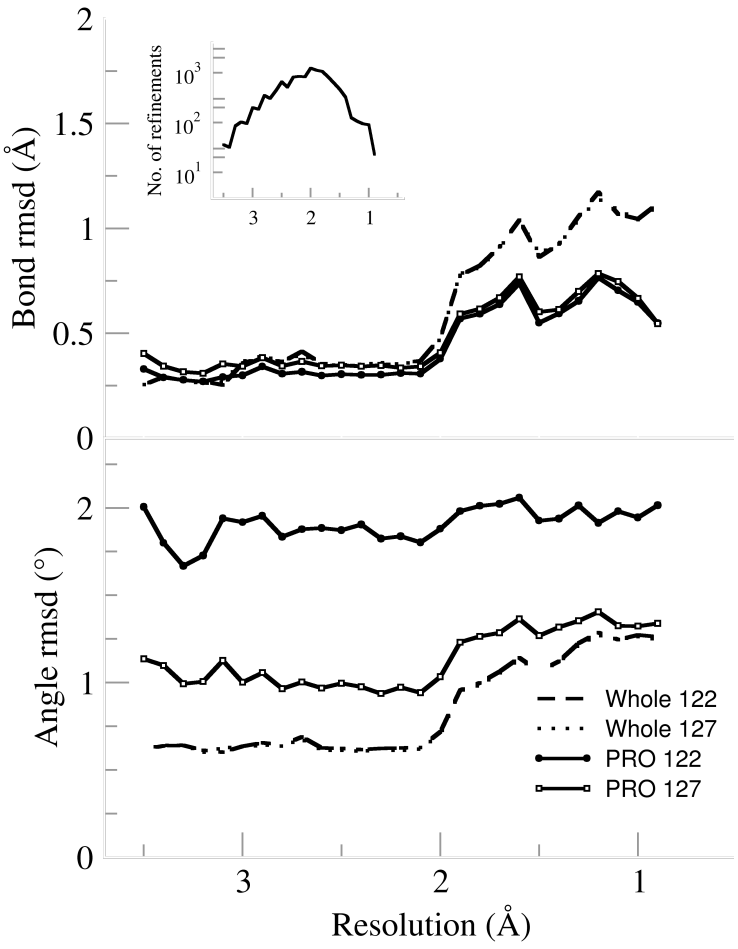


Figure 2: Bond and angle r.m.s.d. values averaged in 0.1Å bins. The r.m.s.d. values for the whole model are shown in dashed and dotted lines, while for the *cis*-PRO r.m.s.d. values are solid lines. Refinements with original EH91 restraints are denoted by solid circle markers (*cis*-PRO only) and EH99 restraints are denoted with open circle markers (*cis*-PRO only). Inset shows the number of refinements in each resolution bin.

Understandably, because the angle ideal values have a far greater change than the bond ideal values. the r.m.s.d. values for the angles are affected to a much greater extent. Not difference is detectable in the values for the whole models but the r.m.s.d. values for just the *cis*-PRO differ by approximately 1°. The EH91 values are uniformly approximately 2° across all resolutions. This uniformity indicates that the EH91 restraints are not suitable as the geometries do not approach the ideal values as the experimental data has less information (low resolution). For the EH99 (PRO 127), the r.m.s.d. values are approximately 1° at low resolution increasing to 1.3° at higher

resolution reflecting the experimental data information. This trend is also inline with the values for entire model indicating a more balanced set of restraints.

Figure 3 shows the r.m.s.Z results in a similar format as Fig. 2. Similarly, the bond values have very little differentiating between the two sets of restraints. By contrast, the angle r.m.s.Z values for the angles are informative. At higher resolutions, the EH99 (PRO 127) restraints result in similar r.m.s.Z values for both the whole models and the *cis*-PRO indicating a balance. Tellingly, the r.m.s.Z values for the EH91 refinements are approximately 0.1 larger at all resolutions even though the e.s.d. for the angle was reduced by half in the EH99 restraints. This implies that the larger e.s.d. was necessary in the earlier restraints to cover the correct ideal angle value.

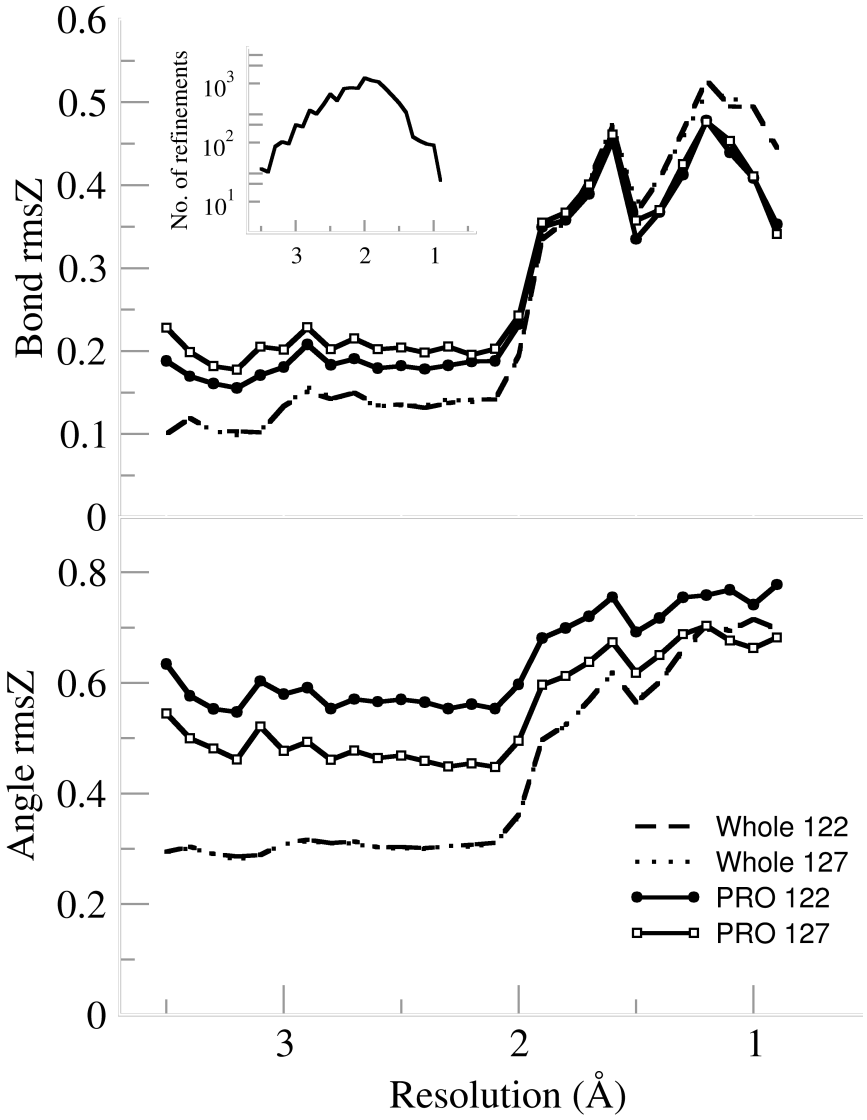


Figure 3: Bond and angle r.m.s.Z values averaged in 0.1Å bins. The r.m.s.Z values for the whole model are shown in dashed and dotted lines, while for the *cis*-PRO r.m.s.Z values are solid lines. Refinements with original EH91 restraints are denoted by solid circle markers (*cis*-PRO only) and EH99 restraints are denoted with open circle markers (*cis*-PRO only). Inset shows the number of refinements in each resolution bin.

A more focused view of the behavior of the restraints for the *cis*-PRO entities appears in Figure 4. The graph is a comparison of the deviations of the refined C–N–Cα angle values from the ideal specified by the restraints. Error bars are placed at the standard error of measurement values. As expected from the results shown in both the r.m.s.d. and r.m.s.Z figures, the r.m.s.d. values of the specific angle using the EH99 restraints are smaller than the earlier released restraints. Specifically, the EH99 values are less than 2° while the EH91 values hover

around 5°. This is an affirmation that the latter restraints are an improvement. A counter argument is the increase in r.m.s.d. values at low resolution for both sets of restraints. There must be other forces (restraints) at play.

Conclusion

The subtle differences between the overall results using the EH91 and EH99 restraints hide the large improvement of the *cis*-PRO entities. The metrics indicate that the *cis*-PRO entities have better geometries (lower r.m.s.d. values) using the EH99 restraints. Even though only 5% of PRO are *cis*-peptides, clearly, any improvement in the restraints will help generate more accurate models but there appears to be room for improvement in the area of *cis*-PRO restraints.

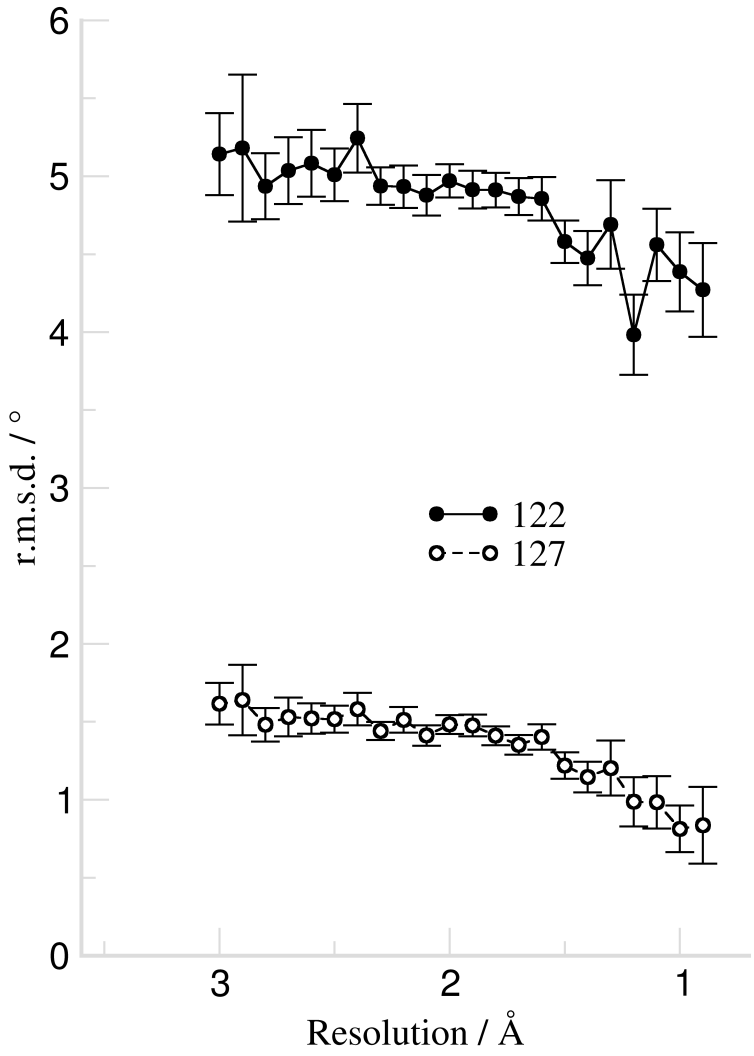


Figure 4: Deviations of the C–N–Cα angle values from the ideal value in 0.1Å bins.

References

Berkholz, D. S., Shapovalov, M. V., Dunbrack, Jr., R. L. & Karplus, P. A. (2009). *Structure*. **17**, 1316–1325.

Bricogne, G., Blanc, E., Brandi, M., Flensburg, C., Keller, P., Paciorek, W., Roversi, P., Sharff, A., Smart, O. S., Vonrhein, C. & Womack, T. O. (2011). BUSTER Cambridge, United Kingdom: Global Phasing Ltd.

Burley, S. K., Berman, H. M., Bhikadiya, C., Bi, C., Chen, L., Costanzo, L. D., Christie, C., Duarte, J. M., Dutta, S., Feng, Z., Ghosh, S., Goodsell, D. S., Green, R. K., Guranovic, V., Guzenko, D., Hudson, B. P., Liang, Y., Lowe, R., Peisach, E., Periskova, I., Randle, C., Rose, A., Sekharan, M., Shao, C., Tao, Y.-P., Valasatava, Y., Voigt, M., Westbrook, J., Young, J., Zardecki, C., Zhuravleva, M., Kurisu, G., Nakamura, H., Kengaku, Y., Cho, H., Sato, J., Kim, J. Y., Ikegawa, Y., Nakagawa, A., Yamashita, R., Kudou, T., Bekker, G.-J., Suzuki, H., Iwata, T., Yokochi, M., Kobayashi, N., Fujiwara, T., Velankar, S., Kleywegt, G. J., Anyango, S., Armstrong, D. R.,

Berrisford, J. M., Conroy, M. J., Dana, J. M., Deshpande, M., Gane, P., Gáborová, R., Gupta, D., Gutmanas, A., Koča, J., Mak, L., Mir, S., Mukhopadhyay, A., Nadzirin, N., Nair, S., Patwardhan, A., Paysan-Lafosse, T., Pravda, L., Salih, O., Sehnal, D., Varadi, M., Vařeková, R., Markley, J. L., Hoch, J. C., Romero, P. R., Baskaran, K., Maziuk, D., Ulrich, E. L., Wedell, J. R., Yao, H., Livny, M. & Ioannidis, Y. E. (2019). *Nucleic Acids Res.* **47**, D520–D528.

Engh, R. & Huber, R. (1991). *Acta Crystallogr. Sect. A*. **47**, 392–400.

Engh, R. & Huber, R. (2001). *International Tables for Crystallography*, Vol. *F*, edited by M. Rossmann & E. Arnold, pp. 382–392. Dordrecht: Kluwer Academic Publishers.

Groom, C. R., Bruno, I. J., Lightfoot, M. P. & Ward, S. C. (2016). *Acta Crystallogr. Sect. B Struct. Sci. Cryst. Eng. Mater.* **72**, 171–179.

Liebschner, D., Afonine, P. V., Baker, M. L., Bunkóczi, G., Chen, V. B., Croll, T. I., Hintze, B., Hung, L.-W., Jain, S., McCoy, A. J., Moriarty, N. W., Oeffner, R. D., Poon, B. K., Prisant, M. G., Read, R. J., Richardson, J. S., Richardson, D. C., Sammito, M. D., Sobolev, O. V., Stockwell, D. H., Terwilliger, T. C., Urzhumtsev, A. G., Videau, L. L., Williams, C. J. & Adams, P. D. (2019). *Acta Crystallogr. Sect. Struct. Biol.* **75**, 861–877.

Moriarty, N. W. & Adams, P. D. (2019). *Acta Crystallogr. Sect. Struct. Biol.* **75**, 16–20.

Moriarty, N. W., Liebschner, D., Tronrud, D. E. & Adams, P. D. (2020). *Acta Crystallogr. Sect. Struct. Biol.* **76**, 1159–1166.

Moriarty, N. W., Tronrud, D. E., Adams, P. D. & Karplus, P. A. (2014). *FEBS J.* **281**, 4061–4071.

Moriarty, N. W., Tronrud, D. E., Adams, P. D. & Karplus, P. A. (2016). *Acta Crystallogr. Sect. -Biol. Crystallogr.* **72**, 176–179.

Murshudov, G. N., Skubak, P., Lebedev, A. A., Pannu, N. S., Steiner, R. A., Nicholls, R. A., Winn, M. D., Long, F. & Vagin, A. A. (2011). *Acta Crystallogr. Sect. -Biol. Crystallogr.* **67**, 355–367.

Vagin, A. A., Steiner, R. A., Lebedev, A. A., Potterton, L., McNicholas, S., Long, F. & Murshudov, G. N. (2004). *Acta Crystallogr. D Biol. Crystallogr.* **60**, 2184–2195.

Williams, C. J., Headd, J. J., Moriarty, N. W., Prisant, M. G., Videau, L. L., Deis, L. N., Verma, V., Keedy, D. A., Hintze, B. J., Chen, V. B., Jain, S., Lewis, S. M., Arendall, W. B., Snoeyink, J., Adams, P. D., Lovell, S. C., Richardson, J. S. & Richardson, D. C. (2018). *Protein Sci.* **27**, 293–315.

The top2018 pre-filtered dataset of high-quality protein residues

Christopher J. Williams, David C. Richardson and Jane S. Richardson
Department of Biochemistry, Duke University, Durham, NC 27710

Correspondence email: dcrjsr@kinemage.biochem.duke.edu

Introduction

This article announces the recent release on Zenodo of a large, high-quality reference dataset of PDB-format coordinate files from which all residues with low model certainty have been removed. Each file is a single protein chain while the total set of files were selected for low redundancy, high resolution, good MolProbity score and other chain-level criteria. Residue-level validation is even more important than overall validation, but only recently has it become feasible to distribute reference datasets in this pre-filtered form.

Our laboratory has emphasized the importance of residue-level as well as chain-level quality filtering of reference datasets as a foundation for model validation and for further bioinformatic structural studies. We began such work in the late 1990s when we introduced our flagship validation of all-atom contact analysis based on the Top100 dataset of reference protein chains, which in our own use we filtered at the residue level on any atomic B-factor >40 (Word 1999). We made available the list for those 100 chains and for all our subsequent, increasingly larger reference datasets (8000 chains by 2013), but had to leave the application of B cutoffs to the user. After deposition of structure factors became required, our validations used explicit electron-density filters for map value and correlation coefficient at each atom, as well as B-factor, all-atom clash and covalent-geometry filters, but we still found no feasible mechanism for distributing all the coordinate files with residue-filter annotations.

Our residue-level quality filtering process relies on extensive infrastructure, especially our developer team's integration into the Phenix software project (Liebschner 2019). We also now manage the filtering information with a Neo4j graphical database (Yoon 2017; Webber 2020). We have switched to using a graphical database to store our reference data because sequence connectivity is modeled natively there (but cumbersome in relational databases), as are the cyclic graphs that define local structural motifs.

The recent breakthrough in our ability to distribute coordinate files in a residue-filtered mode has been enabled by two things. First is our realization that making residue-level quality filtering easily available is worth giving up user flexibility in setting filter thresholds. Second, even more important, is the Zenodo online service that hosts open access to very large, DOI-identified datasets (Sicilia 2017). We have now taken advantage of that venue to distribute our current residue-level pre-filtered datasets. This development allows other researchers to make full and proper use of our curated reference data without needing the expertise, infrastructure and effort required to perform residue-level quality-filtering themselves.

Here we outline the production of this high-quality Top2018 (~15,000-chain) protein dataset and announce the availability of two residue-level pre-filtered versions suitable for general use with little or no further modification. One set is residue-filtered on

mainchain criteria and the other on both mainchain and sidechain criteria. Each set is available at 30%, 50%, 70% and 90% sequence-identity levels. The filtered-out residues leave gaps in the chain, but the remaining high-reliability fragments are surprisingly long –mostly 20-30 residues or more.

Chain selection

We assembled a set of high-quality, low-redundancy protein chains. Chains were selected for consideration from the Protein Data Bank on the following criteria:

- Chain is protein
- Sequence length ≥ 38 residues
- Parent structure solved with x-ray crystallography
- Parent structure solved at better than 2.0Å resolution
- Parent structure has deposited structure factors
- Parent structure deposited on or before December 31, 2018

These chains were analyzed with our validation statistics and chains that failed the following criteria were removed from consideration:

- MolProbity score < 2.0
- <3% of residues have Cβ deviations
- <2% of residues have covalent bond length outliers
- <2% of residues have covalent bond angle outliers

The remaining chains were treated within their PDB-defined sequence-identity clusters, which are calculated weekly with MMseqs2 (Steinegger & Soedling 2018). From each cluster, we selected the chain with the best (lowest) average of resolution and MolProbity score as the best-quality representative of that cluster.

The PDB provides homology clustering at several different levels of stringency. We prepared sets of chains at the 90%, 70%, 50% and 30% sequence-identity levels. (90% is the most permissive, allowing as much as 90% sequence homology between the representatives from different clusters. 30% is the most restrictive, grouping chains into fewer clusters with greater differences between clusters.)

Residue-level filtering

While the selected chains are of good overall quality, this does not guarantee that all residues in them are modeled at high quality with high confidence (Figure 1). Therefore, we applied a residue-level filtering process. Two different residue-filtered sets were created, one filtered just on the mainchain and one filtered on the full residue, including the sidechains. The mainchain filtering considered the atoms N, Cα, C, O and Cβ. Cβ is included with the mainchain atoms since its ideal position is determined solely from other mainchain atom positions. The full-residue filtering considered both mainchain and sidechain heavy atoms. Attached Hydrogen atoms were considered for all-atom contact analysis. Hydrogen atoms were not considered in fit-to-map analyses, as their signal in the map is generally weak or absent.

For a residue to be included in the final dataset, all atoms under consideration had to meet the following criteria:

- B-factor < 40
- Real-space correlation coefficient (rscc) > 0.7

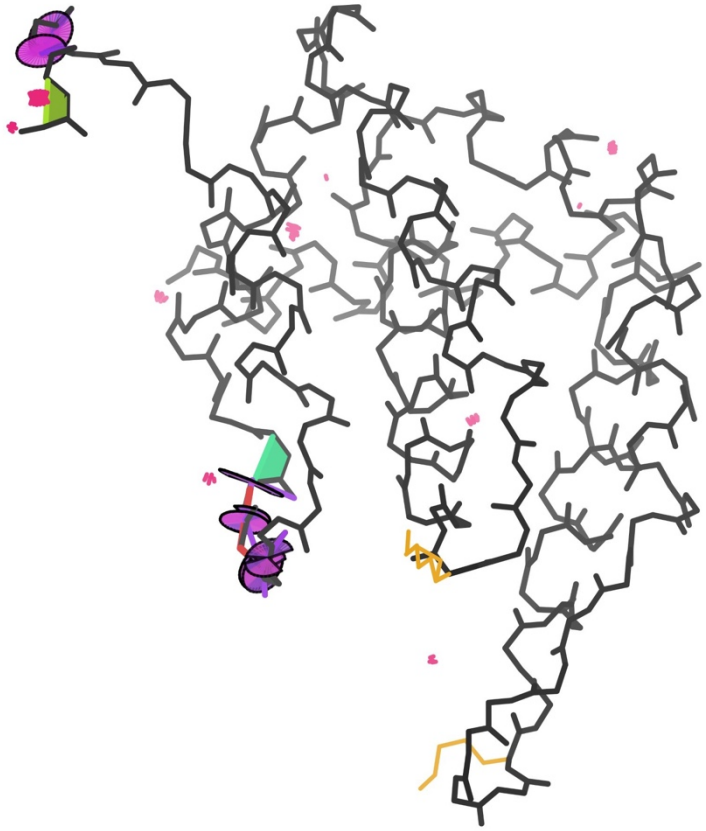


Figure 1: *3D distribution of quality in 5Lp0*

The structure 5Lp0 demonstrates a typical distribution of structure quality for models included in the Top2018. Most of the model is reliable and free from outliers, but two short regions contain a concentration of significant outliers. These problematic regions should not be blindly accepted with the rest of the structure.

- 2mFo-DFc map value at atom position > 1.2 sigma
- No covalent geometry outliers involving those atoms
- No steric clashes involving those atoms
- No alternate modeling conformations for those atoms

All atoms from residues that failed any of these criteria were removed from the PDB files.

The fit-to-map criteria (B-factor, rscc and map value) were obtained using:

```
phenix.real_space_correlation
detail=atom
```

Fit-to-map assessment could not be performed for some structures due to bad MTRIX records or other data issues. Chains from those structures were discarded. The B-factor, rscc and map cutoffs were those developed during production of a rotamer library using our previous, top8000 dataset (Hintze, 2016).

Chains that were < 60% complete after residue filtering were discarded from the final dataset. This serves as a final check on overall structure quality and reduces the amount of chain fragmentation in the included chains.

Only protein residues were filtered. Individual filtering of ligands, ions and waters is beyond

```
USER  DOC Lines marked with USER  DEL list residues pruned by
USER  DOC quality filtering.
USER  DOC Format is chain:resseq:icode:reason_for_pruning
USER  DOC Reasons for pruning are abbreviated as 1-letter codes: bcmgoa
USER  DOC b=bfactor, c=real space correlation, m=2Fo-Fc mapvalue
USER  DOC g=geometry outlier, o=steric overlap, a=alternate conformations
USER  DOC Lines marked USER  INC list the uninterrupted fragments of structure
USER  DOC still included after pruning by quality filtering
USER  DOC Format is chain1:resseq1:icode1:chain2:resseq2:icode2:fragment_length
USER  DOC where 1 is the first and 2 the last residue of the fragment
USER  DOC Line marked with USER  PCT gives statistics for structure completeness
USER  DEL: A:   2: :bcm---
USER  DEL: A:   3: :bcm--a
USER  DEL: A:   4: :----oa
USER  INC: A:   5: : A:  38: :34
USER  DEL: A:  39: :----o-
USER  INC: A:  40: : A:  41: :2
USER  DEL: A:  42: :----o-
USER  DEL: A:  43: :----o-
USER  INC: A:  44: : A:  53: :10
USER  DEL: A:  54: :-----a
USER  DEL: A:  55: :-----a
USER  INC: A:  56: : A:  63: :8
USER  DEL: A:  64: :-----a
USER  INC: A:  65: : A: 114: :50
USER  DEL: A: 115: :b-----
USER  DEL: A: 116: :bc----
USER  PCT:5 fragments:104 residues pass:115 total residues:90.4 % pass
```

Figure 2: *In-file documentation of the residue-level quality filtering.*

Each filtered .pdb file ends with USER DEL records that document the residues that were removed and the reasons for removal as a 6-letter string, USER INC records for the residue stretches that remain and an explanation of the formatting for these records.

the current scope of this dataset. Ligands, ions and waters are included in these files in the interest of completeness, but no guarantee of their quality is implied.

In-file Documentation

The results of residue-level filtering are documented in each resulting .pdb file in USER records appended to the end of the file (Figure 2). These records report the residues that were removed and the reasons for their removal (as a string of 6 single-letter codes), the residues that remain and the lengths of the sequence fragments they form and the overall completeness statistics for the filtered file. See the self-documentation in these USER records for full details.

Importance of Residue Filtering

A key fact that motivated preparation of these datasets is that good average model quality across a whole structure is nevertheless compatible with extremely bad model quality in locally disordered regions with poor density. Familiar cases of this are mobile, unresolved sidechains on a protein's surface compared to well-packed sidechains in a protein's core and unseen backbone at chain termini or in disordered loops.

The CCTBX community may remember the crisis of overabundant *cis*-non-prolines some years ago (Croll, 2015). This phenomenon was pronounced at lower resolutions, but is present in poorly-resolved regions of even very high-resolution structures. Residue-level filtering guards against the inclusion of incorrectly-modeled *cis*-nonPro peptides, on both a statistical and an individual level.

Before filtering, the 70% homology set of the top2018 contained 1959 *cis*-nonPro out of

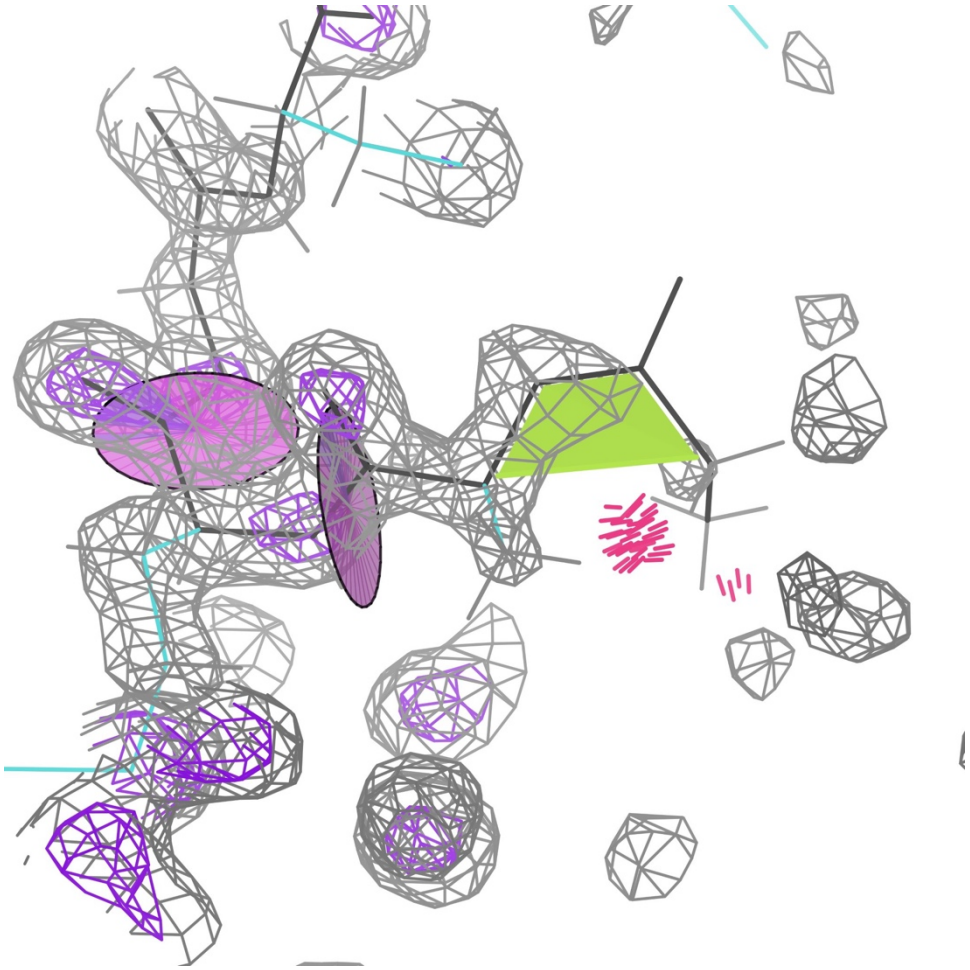


Figure 3: *Erroneous cis-non-proline in 5Lp0*

5Lp0 contains a *cis*-alanine modeled at its N-terminus. The sparse electron density at the terminus is misleading, creating a temptation to model this conformation, but providing no justification for it. Since there is nothing to hold them in place, *cis* conformations at termini are always modeling errors. This residue fails our fit-to-map criteria and has a steric clash. It is therefore removed from the file during filtering.

3,324,246 evaluable peptide bonds, for an occurrence rate of 0.048% or about 1 in 2000 (a rate often reported before any data-quality controls). After filtering, there remain 776 *cis*-nonPro out of 2,652,118, for an occurrence rate of 0.029% or about 1 in 3500. This lower rate agrees with recent observations of valid *cis*-nonPro occurrence (Williams 2018b).

More importantly than these general statistics, residue-level filtering removes many obviously incorrect *cis*-nonPro peptides from the dataset. These include some known, systematic patterns of incorrect *cis* modeling, such as building *cis*-peptides into the truncated density at chain termini (Figure 3). *Cis*-peptides are particularly valuable to filter out, as they tend to be modeled into regions of low certainty (Figure 4). The lack of strong electron density in such regions *allows* this and other modeling errors to occur. It is vital to the health of a statistical reference dataset, homology model, or fragment library to remove these regions of poor and/or unsupported model, as we do in this dataset.

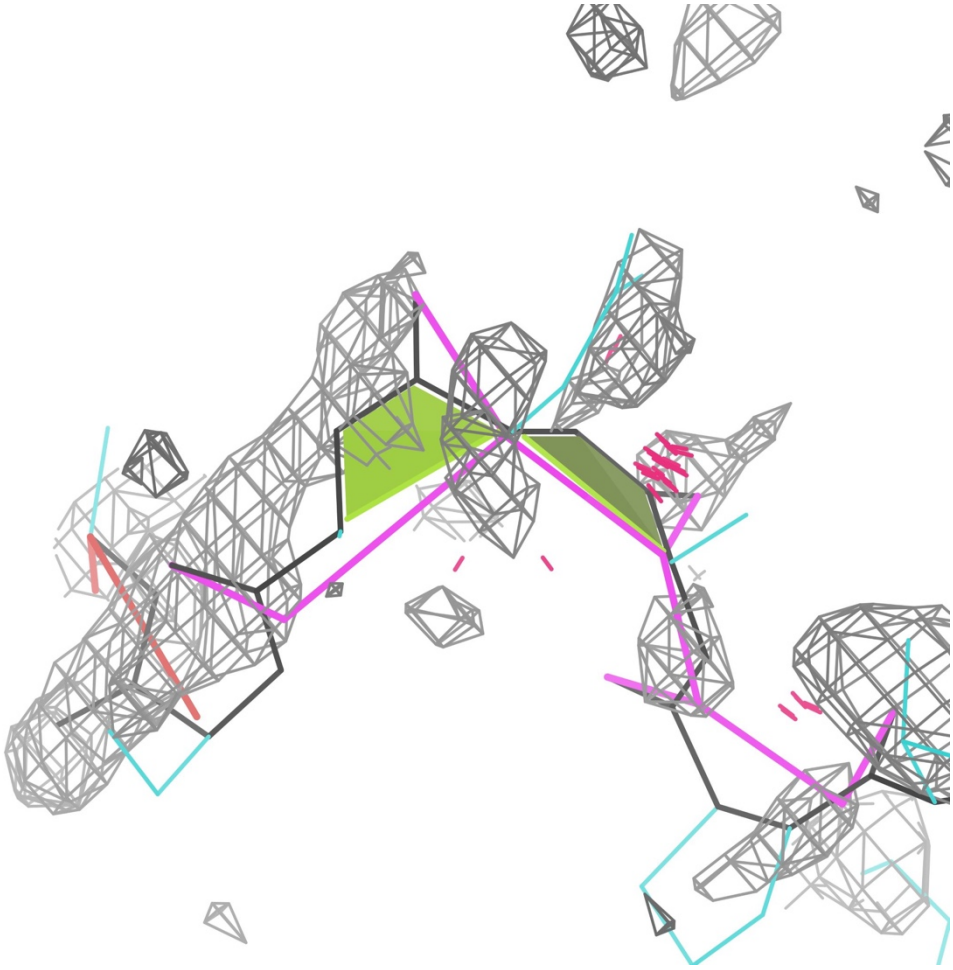


Figure 4: *Double* cis-*non-Pro in 4rm4*

Residue-level filtering also catches and removes this badly-resolved region of 4rm4, including residues 170-172, which form two successive unsupported *cis*-nonPro peptides. This region is clearly not a reasonable interpretation of even this minimal density and should not be allowed to influence future models or statistics. Multiple successive *cis*-nonPro are also a recognized systematic error never seen in genuine cases. Magenta lines show CaBLAM outliers (Williams 2018a) and a cluster of hotpink spikes shows a steric clash ≥4.0Å.

Residue-level filtering thus ensures that the population of *cis*-nonPro peptides is not statistically or locally overrepresented due to modelling errors. The *cis*-nonPro that remain in the dataset (Figure 5) do so based on a reasonable standard of map and model quality and can be used in fragment-based methods or the like with confidence (although we would still advise reasonable statistical weighting).

Conclusions

The full-coordinate, residue-filtered reference datasets described here omit all residues that fail the quality filters, so that they contain only coordinates for residues which are almost certainly correct. The full-residue quality-filtered reference dataset can be used to prepare protein sidechain rotamer libraries (Lovell 2000; Hintze 2016) or to study macromolecular structural motifs that span multiple residues and involve backbone-

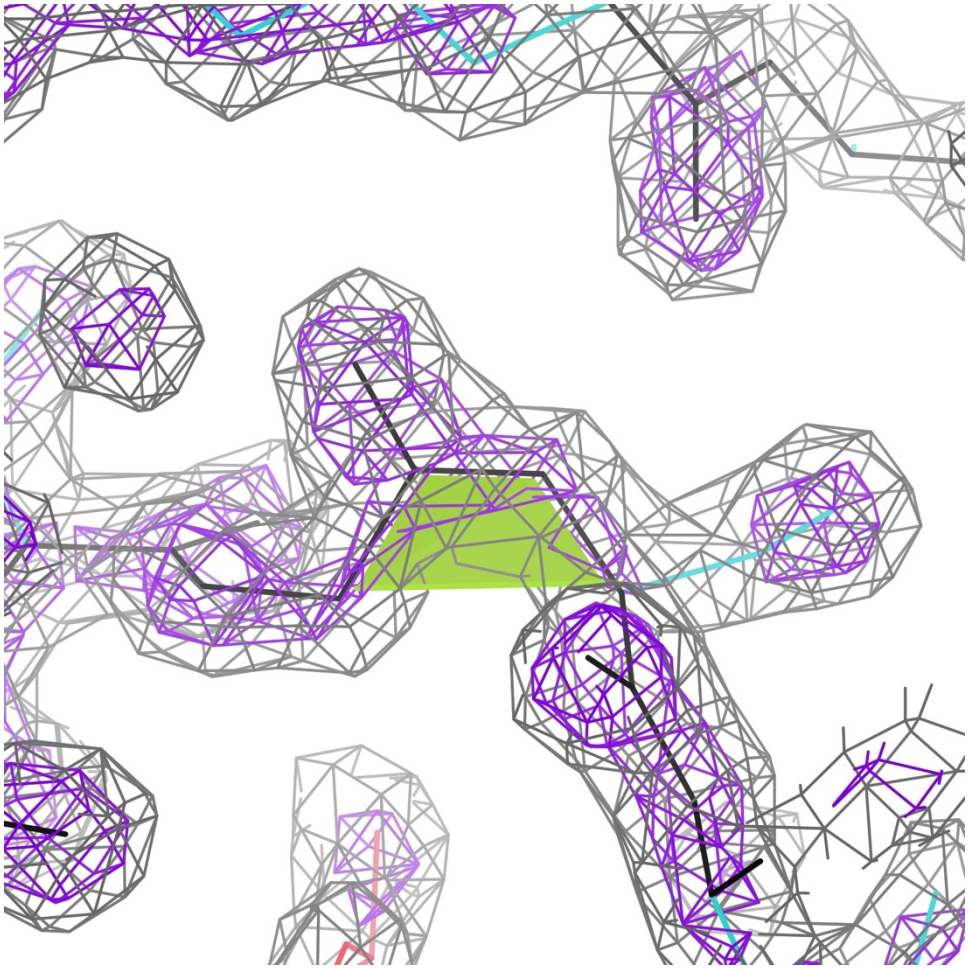


Figure 5: *Supported* cis-*nonPro in 6btf*

This *cis*-serine, residue 275 of 6bft, passes our quality criteria and is included in the structure after filtering. The 1.6Å density is persuasive and well fit by the model. Residues like this that pass our filtering are not guaranteed to be correct, but are guaranteed to meet acceptable quality standards.

sidechain interactions (Videau 2004; Richardson, 2013). The mainchain residue-filtered reference dataset can be used to define Ramachandran distributions (Lovell2003; Read2011) and to prepare curated fragment libraries for model-building or for protein design (Leaver-Fay 2013; Williams2015; Williams2018). In contrast, these gapped, residue-filtered datasets are not suitable for applications that require the full local context, such as Voronoi analyses or all-atom contacts.

Availability

These datasets are available on the Zenodo data repository, each at four levels of sequence redundancy.

The mainchain-filtered set is here: https://doi.org/10.5281/zenodo.4626149.

The full-residue-filtered set is here: https://doi.org/10.5281/zenodo.5115232.

Zenodo supports versioning and these links will resolve to the latest version of each dataset.

References

Croll TI (2015). The rate of *cis–trans* conformation errors is increasing in low-resolution crystal structures. *Acta Crystallographica Section D: Biological Crystallography*, *71*(3), 706-709.

Hintze BJ, Lewis SM, Richardson JS, Richardson DC (2016). Molprobity's ultimate rotamer-library distributions for model validation. *Proteins: Structure, Function, and Bioinformatics*, *84*(9), 1177-1189.

Leaver-Fay A, O'Meara MJ, Tyka M, Jacak R, Song Y, Kellogg EH, Thompson J, Davis IW, Pache RA, Lysko S, Gray JJ, Kortemme T, Richardson JS, Havranek JJ, Snoeyink J, Baker D, Kuhlman B (2013). Chapter Six - Scientific Benchmarks for Guiding Macromolecular Energy Function Improvement. *Methods in Enzymology*, 523. 109-143.

Liebschner D, Afonine PV, Baker ML, Bunkoczi G, Chen VB, Croll TI. Hintze BJ, Hung L-W, Jain S, McCoy AJ, Moriarty NW, Oeffner RD, Poon BK, Prisant MG, Read RJ, Richardson JS, Richardson DC, Sammito MD, Sobolev OV, Stockwell DH, Terwilliger TC, Urzhumtsev AG, Videau LL, Williams CJ, Adams PD (2019). Macromolecular structure determination using X-rays, neutrons, and electrons: Recent developments in Phenix, *Acta Cryst* **D75**: 861-877

Lovell SC, Word JM, Richardson JS, Richardson DC (2000). The penultimate rotamer library. *Proteins*, 40(3):389-408.

Read RJ, Adams PD, Arendall WB III, Brunger AT, Emsley P, Joosten RP, Kleywegt GJ, Krissine EB, Lutteke T, Otwinowski Z, Perrakis A, Richardson JS, Sheffler WH, Smith JL, Tickle IJ, Vriend G, Zwart PH (2011). A New Generation of Crystallographic Validation Tools for the Protein Data Bank. *Structure*, 19(10) 12 October 1395-1412.

Sicilia MA, García-Barriocanal E, Sánchez-Alonso S (2017). Community Curation in Open Dataset Repositories: Insights from Zenodo. *Procedia Computer Science*. 106: 54-60.

Steinegger M, Soedling J (2018) Clustering huge protein datasets in linear time, *Nature Communications*, doi: 10.1038/s41467-018-04964-5.

Videau LL, Arendall WB III, Richardson JS (2004). The Cis Pro Touch-Turn: a Rare Motif Preferred at Functional Sites. *Proteins*, 56, 298-309.

Richardson JS, Keedy DA, Richardson DC (2013) "The Plot thickens: more data, more dimensions, more uses", pp. 46-61 in Biomolecular Forms and Functions: A Celebration of 50 Years of the Ramachandran Map, ed. Bansal M, Srinivasan N, World Scientific Publishing, Singapore.

Webber J, Van Bruggen R (2020) Graph Databases for Dummies, Neo4j Special Edition. John Wiley & Sons, ISBN: 978-1-119-74602-7.

Williams CJ "Using C-Alpha Geometry to Describe Protein Secondary Structure and Motifs" (2015) *Duke University*. ProQuest Dissertations Publishing.

Williams CJ, Headd JJ, Moriarty NW, Prisant MG, Videau LL, Deis LN, Verma V, Keedy DA, Hintze BJ, Chen VB, Jain S, Lewis SM, Arendall WB, Snoeyink J, Adams PD, Lovell SC, Richardson JS, Richardson DC (2018a). MolProbity: More and better reference data for improved all-atom structure validation. *Protein Science*, *27*(1), 293-315.

Williams CJ, Videau LL, Hintze BJ, Richardson JS, Richardson DC (2018b) *Cis*-nonPro peptides: Genuine occurrences and their functional roles, *bioRxiv*, doi: 10.1101/324517

Word JM, Lovell SC, LaBean TH, Zalis ME, Presley BK, Richardson JS, Richardson DC (1999) "Visualizing and Quantitating Molecular Goodness-of-Fit: Small-probe Contact Dots with Explicit Hydrogen Atoms", *J Mol Biol* **285**: 1711-1733.

Yoon BH, Kim SK, Kim SY (2017) Use of Graph Database for the Integration of Heterogeneous Biological Data. *Genomics Inform*, 15(1) 19–27.

COMPUTATIONAL CRYSTALLOGRAPHIC NEWSLETTER

-VE IONS, MULTI-COMPONENT SF, RNA2023

Table of Contents

Editor

Nigel W. Moriarty: NWMoriarty@LBL.Gov

Phenix News

Announcements

New Phenix Release Imminent

The latest version of Phenix – 1.21 – has been released. It will be the last release using Python2. All future version will be Python3 starting with 3.7 or 3.9 depending on OS.

Several new modules have been included in the installation including the quantum chemistry code MOPAC allowing QM calculations with the latest methods with further installation.

Downloads, documentation and changes are available at phenix-online.org. Changes include

- Full support for structure determination with AlphaFold models in Phenix GUI
  - PredictAndBuild X-ray and Cryo-EM structure solution from data and sequences
  - Phenix AlphaFold server
  - Video tutorials for prediction, X-ray structure solution and Cryo-EM map interpretation
- Cryo-EM tools support ChimeraX visualization
- Cryo-EM density modification and anisotropic scaling display local resolution
- Tutorials available for automated structure determination with PredictAndBuild
- New em_placement and emplace_local tools
  - likelihood-based docking of models into cryo-EM maps
- MOPAC v22 is now distributed with Phenix
- Quantum Mechanical Restraints (QMR) to calculate ligand restraints *in situ*
  - Available in phenix.refine and separate command-line tool, *mmtbx.quantum_interface*
  - Higher level QM available via 3rd-party Orca package

- Approximately 27k of 37k restraints (QM calculated and validated) deployed in the GeoStd
- Automated tests for programs in the Phenix GUI

Crystallographic meetings and workshops

Crystallographic & Cryo-EM Structure Solution with Phenix workshop at 74th American Crystallography Association meeting July 7, 2024

Members of the Phenix team will be conducting this workshop in coordination with the ACA.

Expert Advice

Fitting Tip #24 – Negative Ions are not just charge opposites of Positive Ions

Jane Richardson and Michael Prisant, Duke University

The contrast

Positive and negative ions are more different than one might expect. Many positive ions can form metals in the solid state and negative ions do not. Here we study them as isolated, charged atoms in the context of solvated macromolecules. There, positive ions make quite short, direct interactions: ionic bonds with negatively charged atoms or polar bonds either with unprotonated partially charged atoms or with lone pairs on waters (electron donors, not H-bond donors and not drawn as lines here). In contrast, negative ions are bound by much longer, indirect interactions: multiple H-bonds from donor H atoms such as NH or water H (shown as pillows of green dots at the resulting vdW-radius overlaps).

Positive Ions

Most of us are more familiar with the properties and binding sites of positive ions because they are much more common in protein and nucleic acid structures than negative ions.

Zinc is the simplest to recognize and most reproducible. It usually has 4 strongly bound and tetrahedrally directed ligands – routinely they are unprotonated His N or Cys S (as in the Zn finger shown in figure 1), sometimes an O, but never a water. Most often Zinc plays a structural role and is fully occupied, so that its high density is obvious.

Calcium, Ca+2 in proteins (e.g. 1w0n, or similar Dy+3 of Fig. 4) almost always has 6 to 8 oxygen ligands, just arranged around it with no specific coordination geometry. It likes the negative charge of sidechain carboxyls and binds one or both O atoms.

Potassium, like many + ions, has essential biological roles and occurs in proteins that store or transport it such as ion channels, where it successively binds at rings of 4 backbone CO groups. In figure 1, a K+ sits symmetrically between two layers of O6 G atoms (red) in an RNA G-quadruplex.

Magnesium has strongly octahedral coordination, although that geometry is not directly visible at lower resolution or at the solvent surface. Most typically it has 1-2 macromolecular ligands and the rest waters, but can bind well even with all waters, as in the figure. Mg+2 is very common in RNA, since it is a major counterion that stabilizes RNA folding.

Iron, in various (+ve) oxidation states and geometries, is essential and common, as are Mn, Co, Ni and Cu to a lesser extent.

Negative Ions

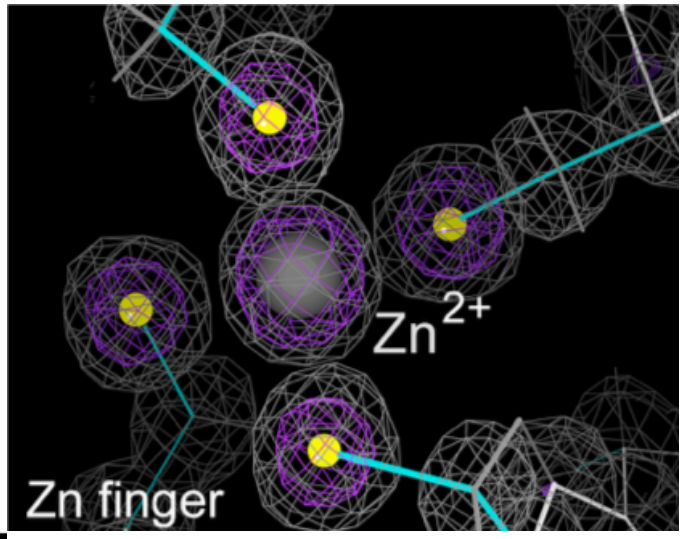
Negative ions seen in macromolecules are the halides: in order of atomic number, Fluoride F- (9), Chloride Cl (17), Bromide Br (35) and Iodide I (53). As noted, they make H-bonds rather than

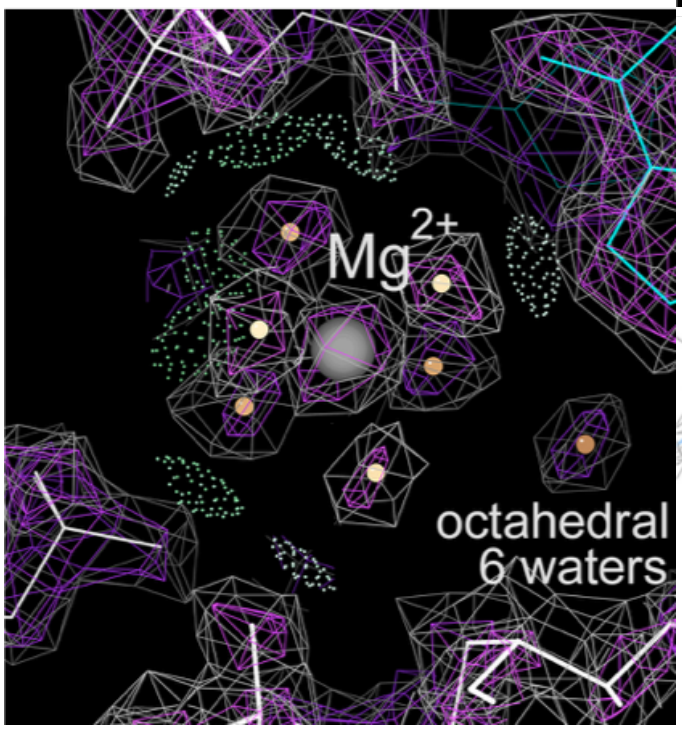
short polar bonds and prefer 6 octahedral H-bond donor ligands. But some of the octahedral directions may instead touch in good van der Walls interactions, the angles may sometimes be distorted and only some ligands will be visible when at the molecular surface or at lower resolution.

Chloride is the most common halide in the PDB. Figure 2 top is a Cl- from a lysozyme crystal soaked in NaCl. It has one backbone NH and 4 water ligands in octahedral geometry. The 6th direction is a good vdW contact (smaller, darker green dots) to the face of the peptide just below it. Fig. 2 center shows the context of that Cl-, including a Na+ and 2 Cl- at the surface each with only one Asn NH2 and one water ligand visible. Fig. 2 bottom shows a clear I- Iodide, which is very nearly indistinguishable from the Cl- in Fig. 2 top. The distance from ion to ligands ranges from 3.1 to 3.5Å in both cases, the map density at the ion is similar (12σ vs 10σ) and each is about twice the density at nearby oxygens. The Iodide may not be at full occupancy, of course.

The bottom line

The bottom line for model fitting is how to recognize ion sites, tell them from waters and





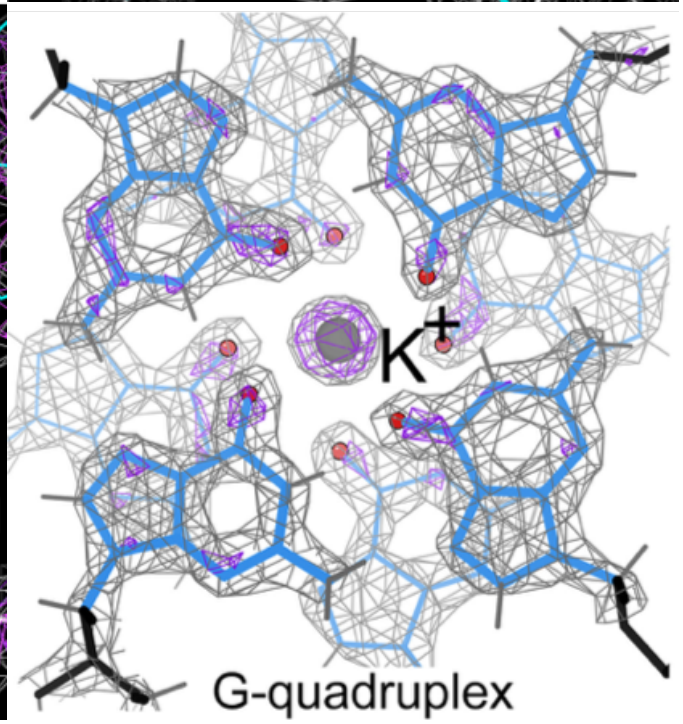


Figure 1: Positive ions, in clear high-resolution examples. (Upper) A Zn+2 ion with 4 tetrahedral Cys S ligands, in 3t7L at 1.1Å (Chaikuad 2011). (Lower left) Mg+2 ion A5103, with 6 octahedral water ligands, in the 8a3d human ribosome at 1.67Å by cryoEM (Faille 2023). (Lower right) A K+ ion in 6e8u at 1.55Å, sandwiched between layers of an RNA-aptamer G-quadruplex (Trachman 2019).

discriminate them from each other. [Also, of course, you will very often see these same atoms as part of other small-molecule ligands; that aspect is not treated here.] The above rules and examples can help you make responsible, probable assignments for individual bound ion atoms. You should always be able to tell positive and negative ions apart, since they have non-overlapping ranges of ion-to-ligand distances. Distances to positive ions vary from 1.9Å for phosphate O to Magnesium up to 2.8 Å for some ligands of Potassium. In contrast, distances to negative ions are H-bonds, with heavy-atom distances that vary from 3.1 to 3.5 Å. Only the transition-metal positive ions have tetrahedral coordination. The lighter Na+, Mg+2. K+ and Ca+2 each have somewhat different ligand and distance preferences, so that set of positive ions can often be distinguished at high to mid resolutions. The negative ions, however, look remarkably like one another and only a full-occupancy iodine is unambiguous just from the map.

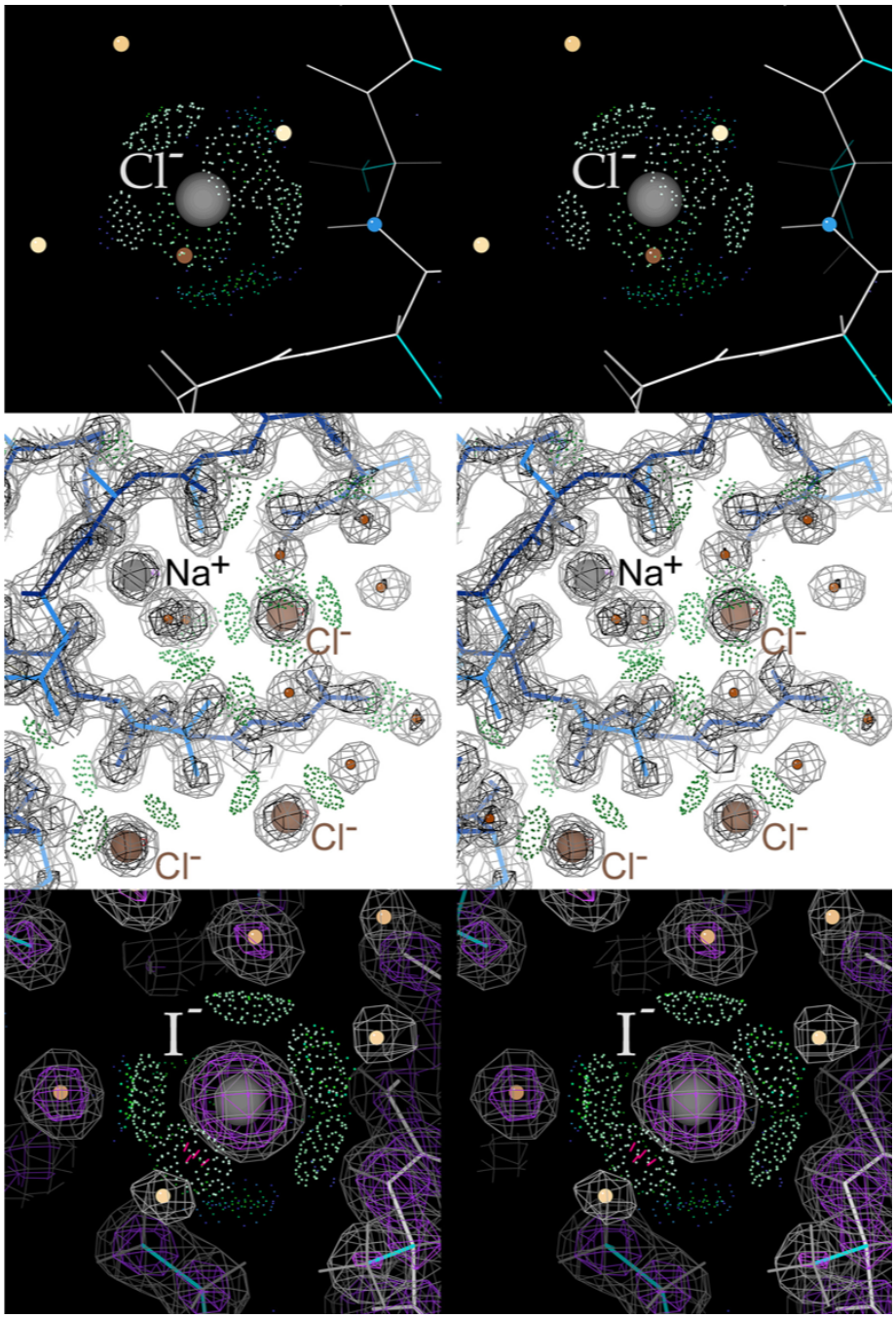


Figure 2: Negative ions in stereo at high resolution. Top: An octahedral Chloride in 7bmt (Koelmel 2012). Center: The context in 7bmt , with a Na+ and two surface Cl-. Bottom: An octahedral Iodide in 2ciw (Kuhnel 2006).

You will need more information than just the density map and coordination at resolutions lower than 2 or 2.5Å, in poorly ordered regions and always if the ion identity and presence really matter for the point of your structure. Even if you know a certain ion is essential and you put it in the crystallization medium, it might still not actually be there in the conformation you crystallized. The paper for the 4enc F- riboswitch shown in figure 3 (Ren 2012) is a great example of using many extra tests to pin down a difficult and important case.

Be sure to check the density value and B factor of the ion peak relative to its surroundings. Look at any density peak significantly higher than the density of good peaks for your macromolecular atoms and at any atoms with an unreasonably low B-factor. See whether the sequence annotation for your molecule shows any common ion binding motifs, such as a Zn finger, an EF hand, or a G-quadruplex. If your molecule is known to have a functional ion site, a distant structural or accidental second site for the same ion is not unusual (as in Fig. 3 of Fitting Tip #22). If you have more than two Cys close together, try a Zn (or perhaps Fe) site as well as disulfides (as in Fig. S4 of Lawson 2021).

Convenient other resources are now also available. The CheckMyMetal web service (Gucwa 2023) at https://cmm.minorlab.org is very useful to assess and distinguish among positive ions and it can now even model and briefly refine a potential replacement ion for you. Be aware that it treats an ion with no direct macromolecular ligands as "unattached" and that it lists but does not handle negative ions (not surprising, since they are not metals), not listing or visualizing any interactions at all for them, neither their H-bonds nor even if they are directly bonded to a positive ion. That failure to show any interactions for negative ions is also true for the NGL Viewer used at the RCSB and PDBe sites. MolProbity's KiNG viewer (Chen 2009) has the opposite problem: it explicitly shows the H-bonds to negative ions but not the ionic bonds to positive ions, as seen in the figures here.

In a large structure it is not feasible to look at all modeled water peaks to check for possible ions or other problems. However, the UnDowser function in MolProbity (Prisant 2020) and in Phenix helps by listing all clashing waters, their clash partners and the B-factor comparison for each clash. The paper shows examples of 14 different types of errors. Of the waters that clash with non-positive polar atoms, many turn out to be positive ions, but UnDowser almost never flags a negative ion since they have near-normal H-bond distances. Another

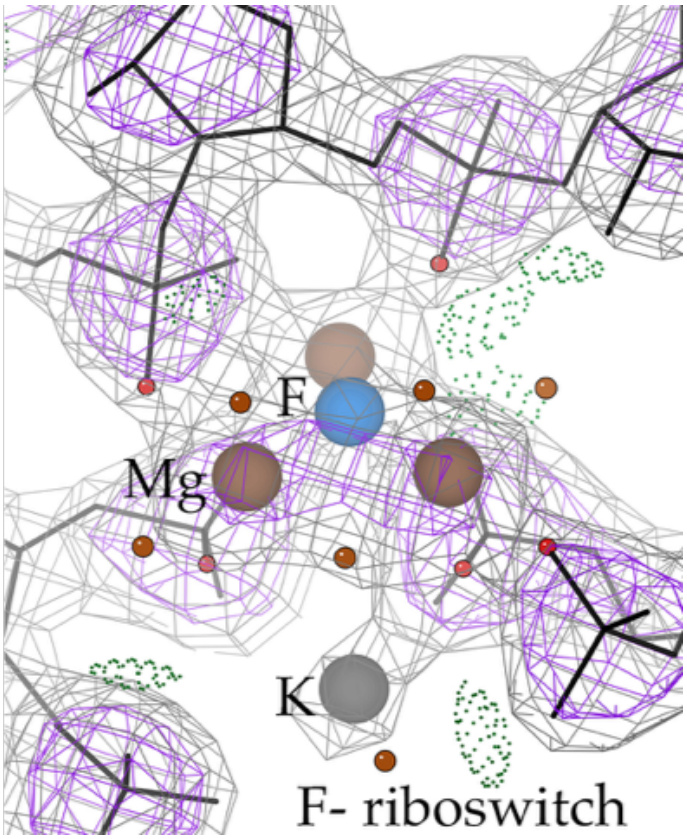


Figure 3: The Fluoride riboswitch, in the bound conformation, from 4enc at 2,27Å (Ren 2012). Contours are at 1.2 and 3s, but peaks for the waters can be seen at lower contour levels.

helpful resource is the ion identification step done in *phenix_refine* after automated water placement (Echols 2014). It is complementary to UnDowser in considering changes of ion identity as well as water-to-ion, but not trying at all to diagnose other water problems. It is less good at the lighter positive ions and is justifiably more conservative, as a fully automated procedure. However, it can fairly often find negative ions since it considers coordination geometry, checks high density values and is greatly aided by the use of anomalous data when that is available. We will hope to combine the strengths of these two methods more thoroughly in the future.

An addendum – Three Ion-related superpowers of macromolecules

I. The Fluoride riboswitch shows that, amazingly, a highly negative RNA molecule can bind a small

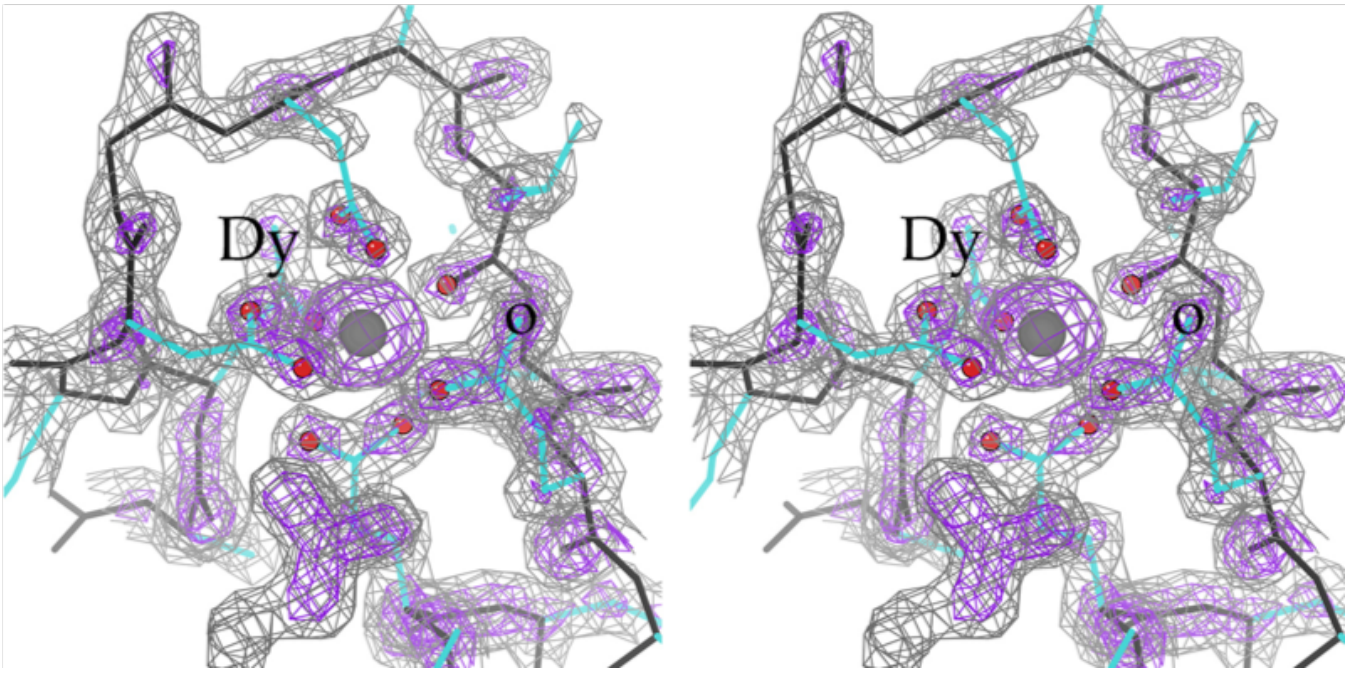


Figure 4: H. quercus lanmodulin, binding Dysprosium Dy+3 with an EF hand helix-loop-helix motif. 8fnr at 1.8Å.

negative ion strongly and selectively (Chloride is known not to bind). It does this by positioning Mg+2 ions next to pairs of phosphates that are distant in sequence (top and bottom RNA strands in Fig. 3) and presumably also distant in space for the unbound conformation of the riboswitch. When F- is present in sufficient concentration, it brings those places together, with three Mg+2 (brown) and a K+ (gray) surrounding the single F- ion (blue), as shown in Figure 3. That change puts the overall riboswitch into the conformation that controls the expression of proteins of Fluoride metabolism.

II. The 15 lanthanides, or "rare-earth", elements differ in useful electronic, magnetic, or catalytic properties, but all are +3 ions and are extremely similar in the physical and chemical properties that allow industrial separation. Current protocols use strong solvents and are very inefficient, requiring on the order of 100 passes over a column. There is a natural family of proteins called lanmodulins that bind lanthanides and it has recently been found that a lanmodulin from the *H. quercus* bacterium in oak tree buds is especially specific (Mattocks 2023). Its crystal structure (Figure 4) showed that the ions were bound at 4 EF-hand motifs with large carboxylate-rich loops. Given waste from rare-earth magnets, it can separate the Neodymium (atomic # 60) from the Dysprosium (66) with 98% purity and 99% specificity in a single column pass. This feat is accomplished because Dy+3 coordinates only 9 oxygens (red balls) while Nd+3 coordinates 10 and allows a dimer to form, further increasing the affinity difference. That extra O is the second branch of the Glu 91 carboxylate, marked with a black o. It is 3.56Å from the Dy ion, while the 9 coordinating O atoms average 2.42Å +/- 0.1 away. But with the larger-radius Nd, that COO could swing to coordinate both oxygens, which would also form a second salt-link H-bond to the Arg in the neighboring molecule (unoccupied guanidinium density below the cluster), helping the dimer form in solution as well as in the crystal.

III. Over the years, JSR has noticed examples of well-occupied single-atom ions next to a protein or RNA but with only water ligands and no direct macromolecular interactions (Fig. 1 bottom). Why would they bind there rather than just staying in the bulk solution to interact with waters there? She now has an answer through MGP's

connections to chemical physics (Berkowitz 2021; Johnson 2003). It seems that waters form clusters around negative ions in solution, but the ion is pushed to one edge of the water cluster. The reason is that the ion wants octahedral coordination and the waters want tetrahedral coordination. A similar mismatch should also happen for positive ions with octahedral coordination. The superpower of macromolecules is to make individual water molecules happy in positions which form most of an octahedral binding site that an ion likes better than staying in solution. These all-water sites are not at all rare for RNA Mg sites –- there are three in the first 30 listed Mg in the 8a3d ribosome and two clear examples in the small 6eru aptamer.

It has long been known that proteins are tool users –- clear for the cofactors, including waters, that they co-opt to help in enzyme catalysis. The riboswitch is an effective but somewhat heavy-handed use of charge, while the lanmodulin specificity and the all-water ion sites are quite subtle uses of geometrical detail.

References

Berkowitz M (2021) Molecular simulations of aqueous electrolytes: Role of explicit inclusion of charge transfer into force fields, *J Phys Chem B* **125**: 13069-76

Chaikuad A, Williams E, Guo K, Sanvitale C ... Bullock A, SGC (2011) Crystal structure of the FYVE domain of endofin (ZFYVE16) at 1.1Å resolution, to be published [3t7L Zn2+]

Chen VB, Davis IW, Richardson DC (2009) KiNG (Kinemage, Next Generation): A versatile interactive molecular and scientific visualization program, *Protein Sci* **18**: 2403-2409

Faille A, Dent KC, Pellegrino S, Jaako P, Warren AJ (2023) The chemical landscape of the human ribosome at 1.67Å resolution, *bioRxiv*, doi: https://doi.org/10.1101/2023.02.28.530191 [8a3d Mg2+]

Gucwa M, Lenkiewicz J, Zheng H, Cymborowski M, Cooper DR, Murzyn K, Minor W (2023) CMM – An enhanced platform for interactive validation of metal binding sites, *Protein Sci* **32**: e4525

Koelmel W, Kuper J, Kisker C (2021) Cesium based phasing of macromolecules: a general ease to use approach for solving the phase problem, *Sci Rep* **11**: 17038-17038 [7bmt Na+, 3 Cl-]

Lawson CL, Kryshhtafoviych A, Adams Pd, Afonine PV, Baker ML ... Chiu W (2021) CryoEM model validation recommendations based on outcomes of the 2019 EMDataResource challenge, *Nat Meth* **18**: 156-164

Mattocks JA, Jung JJ, Lin CY, Dong Z … Cotruvo JA Jr (2023) Enhanced rare-earth separation with a metal-sensitive lanmodulin dimer, *Nature* **618**: 87-93 [8fnr Dy3+]

Prisant MG, Williams CJ, Chen VB, Richardson JS, Richardson DC (2020) New tools in MolProbity validation: CaBLAM for cryoEM backbone, UnDowser to rethink "waters" and NGL Viewer to recapture online 3D graphics, *Protein Sci* **29**: 315-329

Ren A, Rajashankar KR, Patel DJ (2012) Fluoride encapsulation by Mg2+ ions and phosphates in a fluoride riboswitch, *Nature* **486**: 85-89 [4enc F-]

Robertson WH, Johnson M (2003) Molecular aspects of halide ion hydration: The cluster approach, *Annu Rev Phys Chem* **54**: 172-213

Trachman RJ III, Aufour A, Jeng SCY, Abdolahzadeh A ...Ferre-D-Amare AR (2019) Structure and functional reselection of the Mango-III fluorogenic RNA aptamer, *Nat Chem Biol* **16**: 472-479 [6e8u K+]

FAQ

Can I use MOPAC with the current Phenix installer?

Yes, but if you downloaded the Python2 version (the default) the environment variable $PHENIX_MOPAC needs to be set to point to the user installed copy of MOPAC. Better to install the Python3 version (at the bottom of the download page) to have a recent version of MOPAC.

Multi-Component Structure Factor Modeling in Presence of Twinning

Alexandre G. Urzhumtsev[1,2], Paul D. Adams[3,4], Pavel V. Afonine[3,#]

[1]*Centre for Integrative Biology, Institut de Génétique et de Biologie Moléculaire et Cellulaire, CNRS–INSERM-UdS, 1 rue Laurent Fries, BP 10142, 67404 Illkirch, France*

[2]*Université de Lorraine, Faculté des Sciences et Technologies, BP 239, 54506 Vandoeuvre-les-Nancy, France*

[3]*Molecular Biophysics & Integrated Bioimaging Division, Lawrence Berkeley National Laboratory, Berkeley, CA, USA*

[4]*Department of Bioengineering, University of California Berkeley, Berkeley, CA, USA*

#PAfonine@lbl.gov

Introduction

In macromolecular crystals the region surrounding macromolecules is occupied by disordered solvent which contributes to the structure factors. The flat-mask bulk solvent model developed by Jiang & Brunger (1994), followed by several improvements, *e.g.*, Fokine & Urzhumtsev (2002) and Afonine *et al.* (2013), considers this region filled uniformly with the same type of solvent. The total model structure factors are then defined as a scaled sum of atomic model contribution, $\mathrm{F}_{calc}(\mathrm{s})$ and the bulk-solvent:

$$\mathrm{F}_{model}(\mathrm{s}) = k_{total}(\mathrm{s})[\mathrm{F}_{calc}(\mathrm{s}) + k_{mask}(\mathrm{s})\mathrm{F}_{mask}(\mathrm{s})] \qquad (1)$$

Here $\mathrm{F}_{mask}(\mathrm{s})$ are the Fourier coefficients calculated from the flat solvent mask and $k_{mask}(\mathrm{s})$ are corresponding resolution-dependent scale factors. The uniform character of the bulk-solvent has been challenged in the past. Indeed, bulk-solvent region may have isolated sub-regions inside macromolecules or at their interfaces that can be empty, partially occupied, or occupied by disordered chemical entities that are chemically different than the bulk-solvent itself (Liu *et al.*, 2008; Matthews & Liu, 2009; Lunin *et al.*, 2001; Sonntag *et al.*, 2011). To account for the eventually non-uniform features of the bulk-solvent, we have proposed an approach that allows multi-part solvent treatment (mosaic solvent model; manuscript in preparation) in a computationally efficient manner (Afonine *et al.*, 2023). In this approach the bulk-solvent contribution is considered as a scaled sum of contributions $\mathrm{F}_n(\mathrm{s})$ arising from N different solvent components

$$\mathrm{F}_{model}(\mathrm{s}) = k_{total}(\mathrm{s})\left[\mathrm{F}_{calc}(\mathrm{s}) + \sum_{n=1}^{N} k_n(\mathrm{s})\mathrm{F}_{mask}(\mathrm{s})\right] \qquad (2)$$

Here we describe the extension of the algorithm described in Afonine *et al.* (2023) to account for the case of twinned crystals. This new procedure essentially assembles several parts from previously published works with some specific adjustments needed to account for twinning.

Method

To simplify expressions in what follows, we introduce $\mathrm{F}_0(\mathrm{s}) = \mathrm{F}_{calc}(\mathrm{s})$ with $k_0 = 1$ and rewrite (2) as

$$\mathrm{F}_{model}(\mathrm{s}) = k_{total}(\mathrm{s}) \sum_{n=1}^{N} k_n(\mathrm{s}) \mathrm{F}_n(\mathrm{s}) \tag{3}$$

where the coefficients $k_{total}(\mathrm{s})$ and $k_n(\mathrm{s})$ are the variables to determine. In presence of twinning, intensities of the model structure factors are calculated as

$$I_{model}(\mathrm{s}) = \sum_{\mu=1}^{M} \alpha_\mu \left| \mathrm{F}_{model}(T_\mu \mathrm{s}) \right|^2 \tag{4}$$

Here T_μ is the twin operator represented by a 3x3 matrix generating a reflection of the same resolution, $|T_\mu \mathrm{s}| = |\mathrm{s}|$, and the coefficients $0 < \alpha_\mu \leq 1$ describe twinning fractions such that

$$\sum_{\mu=1}^{M} \alpha_\mu = 1 \tag{5}$$

If $M > 1$, we order them by magnitude $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_N$. In the absence of twinning, $M = 1$ and $\alpha_1 = 1$. Below, we consider respective coefficients α_μ as known (see, for example, §2.4 in Afonine *et al.*, 2013).

We search for the coefficients $k_{total}(\mathrm{s})$ and $k_n(\mathrm{s})$ giving the best fit of the model to experimental structure factor intensities minimizing the least-squares function

$$LS = \frac{1}{4} \sum_{\mu=1}^{M} [I_{model}(\mathrm{s}) - I_{obs}(\mathrm{s})]^2 \tag{6}$$

Coefficients $k_{total}(\mathrm{s})$ and $k_n(\mathrm{s})$ are correlated and their values are found iteratively. The initial values of $k_{total}(\mathrm{s})$ and $k_n(\mathrm{s})$ are defined using (1) as described in Afonine *et al.* (2013) with all $k_n(\mathrm{s}) = k_{mask}(\mathrm{s})$, *i.e.*, the considering then bulk solvent mask as a whole. For further iterations, when (approximate) values of coefficients $k_n, n = 0,\ldots,N$ are known, the common scale factor $k_{total}(\mathrm{s})$ is recalculated exactly as in Afonine *et al.* (2013) and we do not repeat this description here. The only difference is that $I_{model}(\mathrm{s})$ is now calculated as a sum (4) of contribution from multiple components, $M > 1$ while Afonine *et al.* (2013) supposes .

Knowing an approximate $k_{total}(\mathrm{s})$ value, individual values $k_n(\mathrm{s})$ for different components are calculated using one of four algorithms described in Afonine *et al.* (2023), with 2nd and 4th algorithms (Alg2 and Alg4, correspondingly) being preferable. However, since Alg4 operates with structure factor amplitudes, the only applicable algorithm in the case of twinning is Alg2 that we generalize below for the case of twinning.

First, we introduce real-valued coefficients

$$G_{nm}(\mathrm{s}) = G_{mn}(\mathrm{s}) = \frac{1}{2}[\tilde{\mathrm{F}}_n(\mathrm{s})\tilde{\mathrm{F}}^*_m(\mathrm{s}) + \tilde{\mathrm{F}}_m(\mathrm{s})\tilde{\mathrm{F}}^*_n(\mathrm{s})] = \tilde{F}_n(\mathrm{s})\tilde{F}_m(\mathrm{s})\,.\cos[\phi_n(\mathrm{s}) - \phi_m(\mathrm{s})] \tag{7}$$

and their twinned combinations

$$\tilde{G}_{nm}(\mathrm{s}) = \sum_{\mu=1}^{M} \alpha_\mu G_{nm}(T_\mu \mathrm{s}) \tag{8}$$

In presence of twinning, function (6) can be reduced to a polynomial of the fourth order

$$LS = \frac{1}{4}\left[\sum_{n=0}^{N}\sum_{m=0}^{N}\sum_{j=0}^{N}\sum_{l=0}^{N} k_n k_m k_j k_l \left(\sum_{\mathrm{s}} \tilde{G}_{jl}(\mathrm{s})\tilde{G}_{nm}(\mathrm{s})\right)\right] - \frac{1}{2}\left[\sum_{n=0}^{N}\sum_{m=0}^{N} k_n k_m \left(\sum_{\mathrm{s}} \tilde{G}_{nm}(\mathrm{s}) I_{obs}(\mathrm{s})\right)\right] + \frac{1}{4}\sum_{\mathrm{s}} [I_{obs}(\mathrm{s})]^2 \tag{9}$$

(for derivation, see formula (7) in Afonine *et al.*, 2023). This function can be minimized using, for example, L-BFGS (Liu & Nocedal, 1989). This procedure requires partial derivatives of (9)

generalizing expression (9) from Afonine *et al.* (2023) and naturally coinciding with it in the absence of

$$\frac{\delta LS}{\delta k_j} = \frac{1}{4}\sum_{m=0}^{N}\sum_{m=0}^{N}\sum_{l=0}^{N} k_l k_m k_n \left(\sum_{\mathrm{s}} \tilde{G}_{jl}(\mathrm{s})\tilde{G}_{nm}(\mathrm{s})\right) - \sum_{n=0}^{N} k_n \left(\sum_{\mathrm{s}} \tilde{G}_{jn}(\mathrm{s}) I_{obs}(\mathrm{s})\right) \tag{10}$$

twinning.

Acknowledgment

PVA and PDA thank the NIH (grants R01GM071939, P01GM063210 and R24GM141254) and the PHENIX Industrial Consortium for support of the PHENIX project. This work was supported in part by the US Department of Energy under Contract No. DE-AC02-05CH11231. AU acknowledge Instruct-ERIC and the French Infrastructure for Integrated Structural Biology FRISBI [ANR-10-INBS-05].

References

Afonine, P. V., Grosse-Kunstleve, R. W., Adams, P. D. & Urzhumtsev, A. (2013). *Acta Cryst.* D**69**, 625–634.

Afonine, P. V., Adams, P. D. & Urzhumtsev, A. (2021).

https://www.biorxiv.org/content/10.1101/2021.12.09.471976v1

Afonine, P. V., Adams, P. D. & Urzhumtsev, A. (2023). *Acta Cryst.* A**79**, 345-352.

Fokine, A. & Urzhumtsev, A. (2002). *Acta Cryst*. D**58**, 1387–1392.

Jiang, J. S. & Brünger, A. T. (1994). *J. Mol. Biol*. **243**, 100–115.

Liu, D. C. & Nocedal, J. (1989). *Math. Program*. **45**, 503–528.

Liu, L., Quillin, M.L. & Matthews, B.W. (2008). *Proc. Natl. Acad. Sci.*

Lunin, V.Yu., Lunina, N.L., Ritter, S., Frey, I., Keul, J., Diederichs, K., Podjarny, A., Urzhumtsev, A.G. & Baumstark, M. (2001). *Acta Cryst.* D**57**, 108-121.

Matthews, B.W. & Liu, L. (2009). *Protein Sci*. **18**, 494–502.

Sonntag, Y., Musgaard, M., Olesen, C., Schiøtt, B., Møller, J.V., Nissen, P. & Thøgersen L. (2011). *Nat Commun.* **2**, 304.

Urzhumtsev, A. & Podjarny, A. D. (1995). Jnt CCP4/ESF–EACMB Newsl. Protein Crystallogr. **31**, 12–16.

The RNA2023 Residue-Filtered RNA Dataset. Negative Ions Are Not Just Charge Opposites of Positive Ions; Plus a Digression on 3 Ion-Related Superpowers of Macromolecules

Christopher J Williams and Jane S Richardson

Duke University, Durham, NC, 27710

Correspondence email: christopher.sci.williams@gmail.com

Introduction

Any learning, human or machine, requires high-quality input data. The Richardson lab uses high-quality, filtered datasets of protein or RNA residues to develop structure validations. These datasets have allowed us to set validation targets for Ramachandran and CaBLAM distribution and to define clusters of sidechain rotamers and RNA backbone conformations.

Over the years, we have created and shared several lists of high-quality protein and RNA chains for use as training sets. However, we previously had to leave the task of residue-level filtering to the individual users of these datasets due to file-size limits. Thanks to high-volume data distribution through Zenodo, the release of the Top2018 protein residue dataset (Williams, 2022) represented the Richardson lab making our residue-level filtering broadly available for proteins. Here we present the RNA2023 dataset, an equivalent, residue-filtered dataset for RNA.

Chain selection

Our selection of candidate chains was based on the 3.150 version of the http://rna.bgsu.edu/rna3dhub/nrlist list (Leontis and Zirbel, 2012). This list groups RNA structures into classes based on sequence and structure, then selects the best representative of each class based on resolution, RSR, RSCC, Rfree, clashes, and completeness. This selection process is philosophically similar to our selection process for protein chains in previous work.

We applied an additional resolution cutoff of 1.9Å or better to this list. We made a single exception to this cutoff – 6ugg, a 1.95Å tRNA structure – as we felt including a high-quality, uncomplexed tRNA representative was important.

We added two additional cryoEM ribosome structures: 8a3d, a human ribosome at 1.67Å and 8b0x, an *E.coli* ribosome at 1.55Å. 8a3d contributed three RNA chains to the candidate list: 28S rRNA, 5S rRNA, and 5.8S rRNA. 8b0x contributed five RNA chains: 16S rRNA, 23S rRNA, 5S rRNA, an mRNA, and a tRNA. These structures were solved via cryoEM, and represent the first cryo structures to be included in one of our high-quality datasets. They contribute not only a large number of residues but also, more importantly, a great diversity of local conformations.

Finally, after the residue filtering process, we removed any chains that did not have any surviving suites, i.e. the sugar-to-sugar backbone region of two sequential residues. In the protein datasets, we set a completeness cutoff of 60% after filtering. RNA chains are frequently short enough and suites are long enough that a percentage-based completeness cutoff is overly punishing. Possession of at least one complete suite – the basic unit of RNA backbone geometry (Murray, 2003) – was therefore used instead as our criterion for meaningful contribution to the dataset.

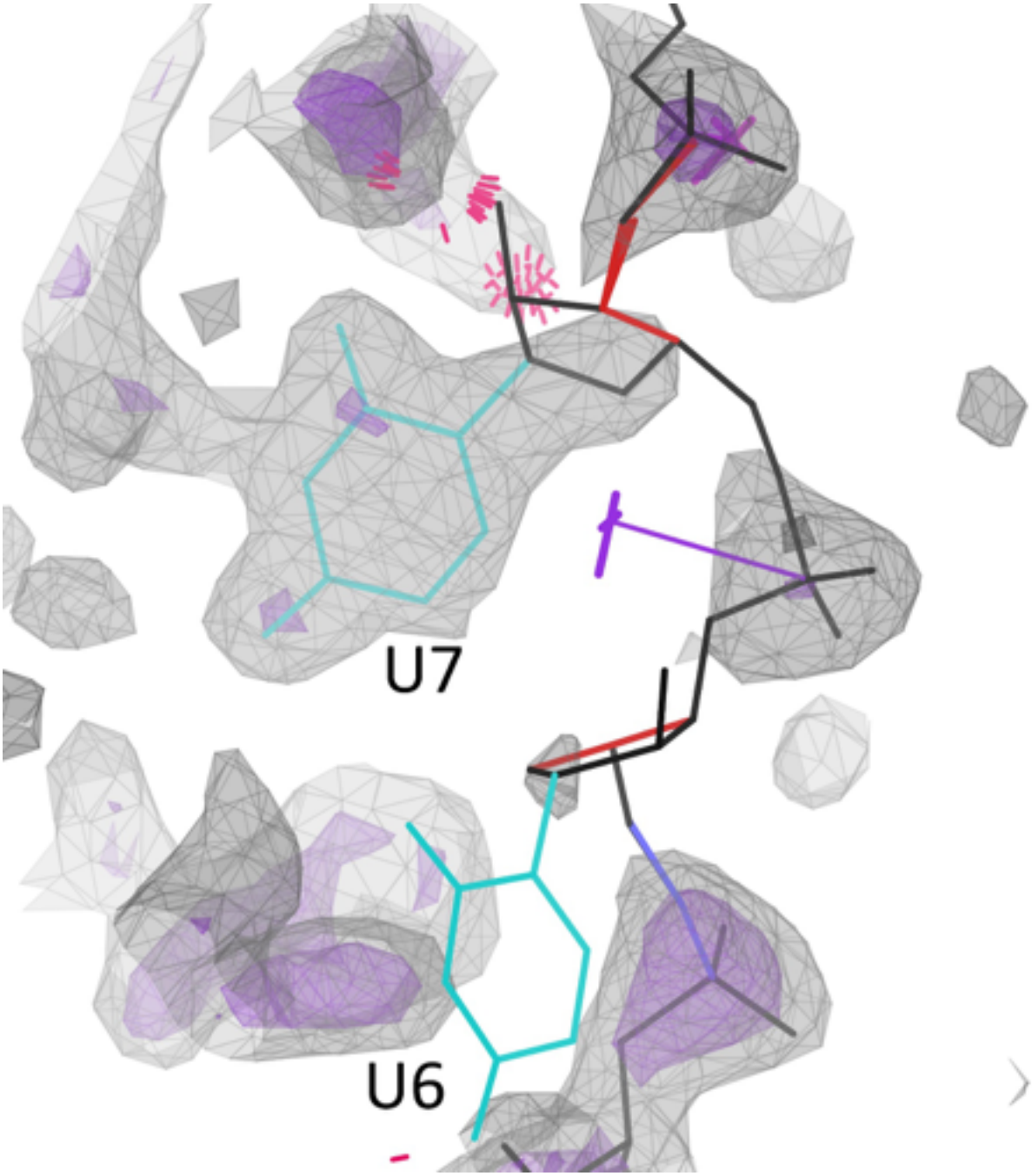


Figure 1: 3boy, chain D, residues 6-7, with electron density map at 1.2σ (gray) and 3.0σ (purple). 3boy is a 1.7Å x-ray structure of generally good quality, and its chain D, a 22-residue mRNA, is one of the candidates for this dataset. This region, however, is poorly resolved and as a result, poorly modeled. Residue 7 fails every one of our residue filtering criteria, except for occupancy. These residues should not be accepted simply because their parent structure is good overall, and their serious problems illustrate the importance of residue-level filtering.

Residue criteria

As in the protein case, model quality and confidence vary across an RNA structure. Structures of good overall quality may have regions of low quality or low confidence (Figure

1) that should not be included in a training or reference dataset.

We require that all RNA residues meet the following validation criteria:

- No steric overlaps (clashes) >= 0.5Å
- No sugar pucker outlier (Jain, 2015)
- No covalent geometry (bond or angle) outliers (>=4σ)

All RNA residues from x-ray crystallography structures must meet the following map criteria:

- 2Fo-Fc map value for P atom >= 2.4σ
- 2Fo-Fc map value, averaged for two lowest atoms, >= 1.2σ
- RSCC value, averaged for two lowest atoms, >= 0.7
- Occupancy = 1.0

All RNA residues from EM structures must meet the following map criteria:

- Residue inclusion fraction within depositor-recommended contour level >= 0.95 for 8a3d, or = 1.0 for 8b0x
- Whole-residue RSCC >= 0.7
- Occupancy = 1.0

Residues that failed any of these criteria were removed from the structure files.

We also prepared an alternative "nosuiteout" dataset that additionally removes all "!!" suite conformation outliers (Richardson, 2008).

Steric overlaps were identified using Reduce and Probe. Sugar puckers, RNA covalent geometry, and RNA suites (where applicable), were validated using `phenix.rna_validate`. We used `phenix.real_space_correlation detail=atom` to calculate 2Fo-Fc map values, RSCC, and occupancies for x-ray structures. We used `phenix.map_model_cc` to calculate RSCC scores for EM models. Residues' inclusion fractions were taken from the validation.xml files provided by the PDB.

Dataset contents

132 unique structures contributed chains to the dataset. The standard dataset contains 151 chains and 6217 complete suites. 4293 suites are the dominant 1a conformation; 1924 suites are non-1a. The nosuiteout dataset contains 149 chains and 5567 complete suites. 4127 suites are 1a; 1427 suites are non-1a, non-!!.

New residue-filtering challenges

Residue-level filtering for RNA presents new challenges relative to our previous protein work.

B-factor was frequently used in our previous datasets as a primary filtering criterion due to its ready availability in all PDB files. However, B-factor is handled differently among different refinements and carries inconsistent meaning across different resolutions. The inconsistency in B-factor became critical in the preparation of this dataset, where we did not find a B-factor cutoff that was sufficiently selective but not unreasonably punishing across the candidate chains. We therefore chose to drop B-factor as a filtering criterion and to depend instead on direct map-model metrics such as RSCC. We expect this change in filtering philosophy will persist into our future protein datasets.

This dataset is the first time we have included EM structures. Model validation criteria (clashes, sugar puckers, etc.) have consistent expectations across methods. However, cryoEM density maps differ from x-ray electron density maps in physical interaction and mathematical protocols and require new criteria, especially because their numerical density values are inherently relative even for the zero point.

Here we use the atom-fraction "residue_inclusion" value taken from the validation.xml files available on the rscbPDB. This measure is calculated for each residue and

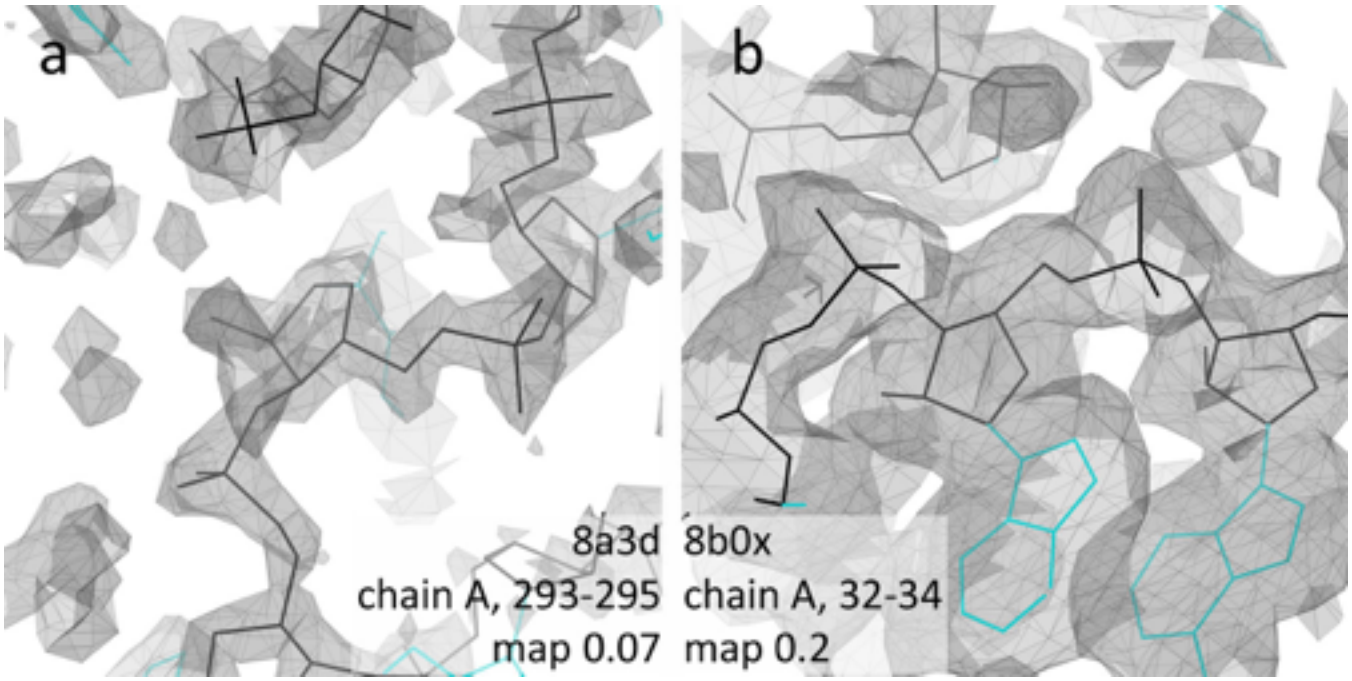


Figure 2: Good local regions with 1.0 residue inclusion fraction, from the human 8a3d and *E.coli* cryoEM ribosomes structures, shown with a single map contour at each structure's depositor-recommended contour level. A) 8a3d, centered on the ribose of 28S G294, with the highly restrictive contour at map density value 0.07. Inspection of example regions convinced us that a 0.95 residue inclusion score would keep only highly reliable residues. b) 8b0x centered on 16S A33, at the quite lenient contour at map density value 0.2. Here, we cannot be more selective than an inclusion fraction of 1.0.

indicates the fraction of that residue's non-hydrogen atoms that fall within a depositor-defined map level (contour_level_primary_map, in the same file). This is roughly equivalent to our x-ray criterion which asks whether the residue falls within the 1.2σ density envelope. (A few residues do not have residue inclusion values. All of these had some other modeling problem that would have eliminated them from the dataset. Any residue for which a residue inclusion fraction cannot be calculated can safely be assumed to fail reasonable filtering criteria.)

Using the depositor's recommendation to define the critical map level presents challenges. There are many legitimate factors a depositor may consider in setting this cutoff, which do not all correspond to our interest in determining local map quality. Figure 2 shows the differences in map character between the two ribosome structures at the depositor-recommended contour levels. Based on visual inspection of example regions near full inclusion at that contour, we chose 0.95 as the required residue inclusion fraction for 8a3d, and 1.0 as the required fraction for 8b0x.

Our use of the residue inclusion fraction represents an early and accessible measure of local map quality. Like B-factor, it requires no special knowledge or software to access, being available from PDB downloads. And like B-factor, it is handled inconsistently across different research groups and different structures. We expect our map-based filtering criteria to evolve in future datasets as we develop or discover better methods.

Dataset access

The rna2023 dataset is available on Zenodo at https://doi.org/10.5281/zenodo.8103013 This link will resolve to the latest version if updates are made.

There are two versions of filtering available. The default dataset includes residues with "!!" suite evaluations. These are conformational outliers relative to known suites. However, as the number of solved RNA structures increases, we are discovering new valid suite conformations. For most purposes, it is therefore appropriate to allow "outlier" RNA backbone conformations, if fit to map and other validation metrics support the model. For specialist purposes where it is desirable to consider only known suite conformations, we supply an alternative "nosuiteout" version of filtering where residues with "!!" conformations have been removed. Each of these filtering versions is available in PDB and mmCIF formats.

The Zenodo repository also includes a file with PDB metadata such as resolution, R values, and deposition title, a chain list file documenting the completeness statistics for member chains, and suitename tables with precomputed suitename identities for all member residues.

References

Jain, S., Richardson, D. C., & Richardson, J. S. (2015). Computational methods for RNA structure validation and improvement. In *Methods in Enzymology* (Vol. 558, pp. 181-212). Academic Press.

Leontis, N. B., & Zirbel, C. L. (2012). Nonredundant 3D Structure Datasets for RNA Knowledge Extraction and Benchmarking. In RNA 3D Structure Analysis and Prediction N. Leontis & E. Westhof (Eds.), (Vol. 27, pp. 281–298). Springer Berlin Heidelberg. doi:10.1007/978-3-642-25740-7_13

Murray, L. J., Arendall III, W. B., Richardson, D. C., & Richardson, J. S. (2003). RNA backbone is rotameric. *Proceedings of the National Academy of Sciences*, *100*(24), 13904-13909.

Richardson, J. S., Schneider, B., Murray, L. W., Kapral, G. J., Immormino, R. M., Headd, J. J., ... & Berman, H. M. (2008). RNA backbone: consensus all-angle conformers and modular string nomenclature (an RNA Ontology Consortium contribution). *RNA*, *14*(3), 465-481.

Williams, C. J., Richardson, D. C., & Richardson, J. S. (2022). The importance of residue-level filtering and the Top2018 best-parts dataset of high-quality protein residues. *Protein Science*, *31*(1), 290-300.

Histidine Protonation Dependent Library (HPDL) for Updating Restraints of the Imidazole Moiety

Nigel W. Moriarty

Molecular Biophysics and Integrated Bioimaging, Lawrence Berkeley National Laboratory, Berkeley California 94720, United States

Correspondence email: NWMoriarty@LBL.Gov

Abstract

Histidine can be protonated with a hydrogen atom on either or both of the two nitrogen atoms of the imidazole moiety. The protonation state leads to a change to the geometry of the histidine side-chain: specifically and largely the angles in the ring. Updating the restraints based on histidine protonation leads to an improvement of agreement of the refined geometry and restraints at high resolution.

Introduction

Protonation of histidine can take on three different forms – one hydrogen atom on either or both of the nitrogen atoms in the heterogeneous ring of the imidazole moiety. As (Malinska *et al.*, 2015) noted:

> This variability is of particular importance in protein structures, although it is also particularly recalcitrant to X-ray crystallographic characterization because of the limited resolution and the inability to detect H atoms that blight the method in routine applications.

This assessment led to the development of a set of ideal values for each protonation state of the histidine side-chain based on a search of the Cambridge Structural Database (Groom *et al.*, 2016). While not specifically stated or implemented, these ideal values can be used as ideal values for bond and angle restraints for specific protonation states of histidine used in a model refinement.

Incidentally, Malinska *et al.* also developed a method for predicting the protonation by relaxing the histidine side-chain restraints and using the (approximately unrestrained) refined bond and angle values in two functionals. Unfortunately, it does not work well at "routine" resolutions. In the 1Å example given, only just over 60% of the protonations could be determined.

For context, a recent investigation into the restraints for the arginine amino acid (Moriarty *et al.*, 2020) revealed that changing the ideal values and estimated standard deviation (e.s.d.) of each restraint can have a notable effect on the refinement results particularly if focused on the specific amino acid. In particular, the change in the overall angle root mean squared deviation (r.m.s.d.) values were negligible but the arginine specific r.m.s.d. improved by 0.25° at better than 2Å resolution with a much smaller improvement at lower resolutions. While muted these improvements prompted a deeper and successful investigation into the merits of the new arginine restraints revealing that the torsion restraint needed adjustment. Interestingly, this nuance was missed in the earlier arginine restraints paper (Malinska *et al.*, 2016). Another notable result is that bond restraints (and their changes) have much less influence.

Methods

In a similar fashion to the implementation of the Conformation Dependent Library (CDL, Karplus, 1996; Berkholz *et al.*, 2010, 2009; Moriarty, Adams et al., 2014; Tronrud *et al.*, 2010; Moriarty, Tronrud *et al.*, 2014; Moriarty et al., 2016), the Histidine Protonation Dependent Library (HPDL) adjusts the bond and angle restraints directly in the restraint objects in memory. This is computationally efficient and because the user just has to choose the option, there are no additional files required. The option is hpdl=True.

Interestingly, even though Malinska et al. developed new ideal values for different protonation states of histidine, the update angle restraints for the hydrogen atoms were absent from the paper. Even though the refinement program used in the study – REFMAC (Murshudov *et al.*, 2011) – routinely does not write hydrogen atoms in the final result model, it can use hydrogen atoms internally using the restraints in the Monomer Library (Vagin *et al.*, 2004). If this was the case, the hydrogen atoms would have been refined incorrectly and there would have been scant evidence of the error. The simplest solution for the current work was to bisect the external angle of each nitrogen atom such that the hydrogen atom is restrained to the plane of the ring. As a guide, the internal angle of a protonated nitrogen atom is larger than the un-protonated nitrogen atom.

To test the HPDL restraints, models from the PDB with two different sets of restraints. The first set ('standard') uses the restraints for histidine from the Monomer Library, which is the standard restraints library for the refinement of macromolecules in Phenix (Liebschner *et al.*, 2019). The second set of refinements used the HPDL restraints.

All refinements were performed using phenix.refine (Afonine *et al.*, 2012). Coordinate and experimental data files were obtained from the PDB that met the following criteria: resolution better than 3.05Å, data completeness >90%, data are not twinned, Rwork<30%, Rfree<35% and Rfree-Rwork>1.5%. For entries with resolutions of better than 1.05Å, the Rfree-Rwork criterion was changed to >0.5%. By using these criteria, we excluded suspicious entries and low-resolution data, allowing automatic refinement strategies with default options. Hydrogen atoms were added to the models using Phenix ReadySet!. Ligand restraints were generated by Phenix eLBOW (Moriarty *et al.*, 2009). Each model was then subjected to ten macrocyles of refinement using the default strategy in phenix.refine for the refinement of coordinates, atomic displacement parameters (ADP) and occupancies. Nondefault refinement options included optimization of the weight between the experimental data and the geometry restraints. In addition, anisotropic ADPs were used for non-H protein atoms at resolutions better than 1.55Å and for water oxygen atoms at resolutions better than 1.25Å. The quality of the resulting models was assessed numerically using MolProbity (Williams *et al.*, 2018) in Phenix. To filter out problematic structures, refined models with a clashscore of greater than 12 were not included in the analysis. The results were grouped into resolution bins of width 0.1Å. Resolution bins with less than 10 refined structures were not taken into account. This led to a total of 40,694 protein structures refined with conventional and modified arginine restraints.

Results

The quintessential comparison of the two sets of restraints is shown in Figure 1. The overall r.m.s.d. values for the overall protein models are identical except for the 0.8Å bin which differs by 0.04° in 25 models. Figure 2 has a zoomed in depiction of Figure 1. For the histidine specific comparison, the 0.8Å bin is improved by 0.11°. The r.m.s.d. values differ by about 0.05° for the two next two lower resolution bins. Neither of the comparisons are significantly different. The bond r.m.s.d. Value differences are even less significant.

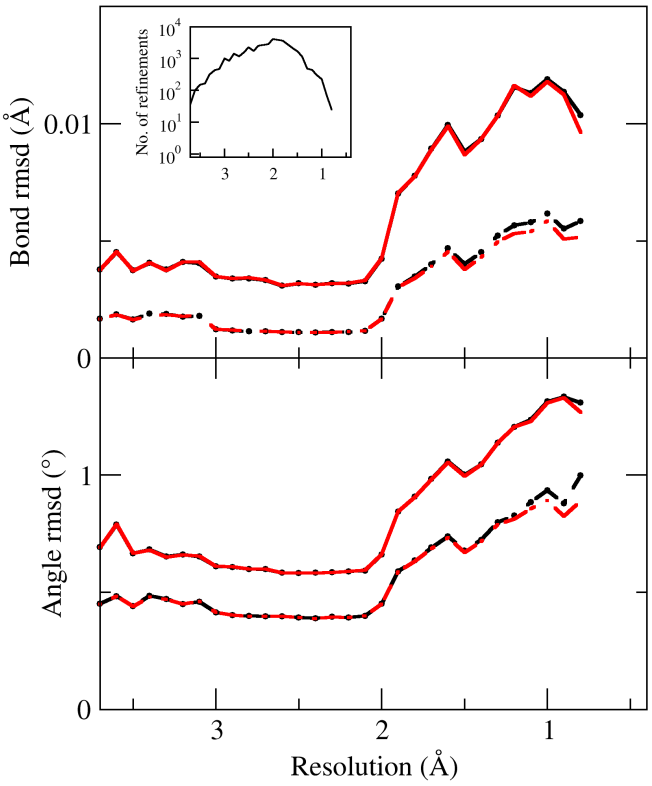


Figure 1: Comparison of bond and angle r.m.s.d. values for whole protein (solid lines) and histidine (dashed lines) for two sets of refinements using the standard restraints (black lines) and HPDL restraints (red lines) in 0.1Å bins. Insert shows the number of refinements in each resolution bin.

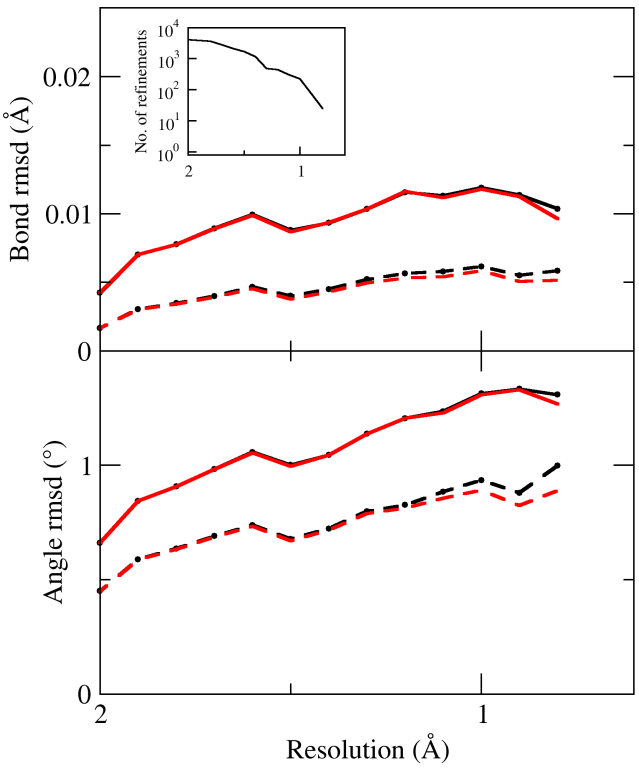


Figure 2: Same as Fig1 but zoomed into the high-resolution range.

Conclusions

The histidine protonation dependent library (HPDL) improves the r.m.s.d. values by an insignificant amount.

References

Afonine, P. V., Grosse-Kunstleve, R. W., Echols, N., Headd, J. J., Moriarty, N. W., Mustyakimov, M., Terwilliger, T. C., Urzhumtsev, A., Zwart, P. H. & Adams, P. D. (2012). Acta Crystallogr. Sect. D-Biol. Crystallogr. 68, 352–367.

Berkholz, D. S., Krenesky, P. B., Davidson, J. R. & Karplus, P. A. (2010). Nucleic Acids Research 38, D320–D325.

Berkholz, D. S., Shapovalov, M. V., Dunbrack, Jr., R. L. & Karplus, P. A. (2009). Structure 17, 1316–1325.

Groom, C. R., Bruno, I. J., Lightfoot, M. P. & Ward, S. C. (2016). Acta Cryst B, Acta Cryst Sect B, Acta Crystallogr B, Acta Crystallogr Sect B, Acta Crystallogr B Struct Crystallogr Cryst Chem, Acta Crystallogr Sect B Struct Crystallogr Cryst Chem 72, 171–179.

Karplus, P. A. (1996). Protein Science 5, 1406–1420.

Liebschner, D., Afonine, P. V., Baker, M. L., Bunkóczi, G., Chen, V. B., Croll, T. I., Hintze, B., Hung, L.-W., Jain, S., McCoy, A. J., Moriarty, N. W., Oeffner, R. D., Poon, B. K., Prisant, M. G., Read, R. J., Richardson, J. S., Richardson, D. C., Sammito, M. D., Sobolev, O. V., Stockwell, D. H., Terwilliger, T. C., Urzhumtsev, A. G., Videau, L. L., Williams, C. J. & Adams, P. D. (2019). Acta Cryst D 75, 861–877.

Malinska, M., Dauter, M. & Dauter, Z. (2016). Protein Sci 25, 1753–1756.

Malinska, M., Dauter, M., Kowiel, M., Jaskolski, M. & Dauter, Z. (2015). Acta Crystallogr. D Biol. Crystallogr. 71, 1444–1454.

Moriarty, N. W., Adams, P. D. & Karplus, P. A. (2014). Computational Crystallography Newsletter 5, 42–49.

Moriarty, N. W., Grosse-Kunstleve, R. W. & Adams, P. D. (2009). Acta Crystallogr. Sect. D-Biol. Crystallogr. 65, 1074–1080.

Moriarty, N. W., Liebschner, D., Tronrud, D. E. & Adams, P. D. (2020). Acta Cryst D 76, 1159–1166.

Moriarty, N. W., Tronrud, D. E., Adams, P. D. & Karplus, P. A. (2014). FEBS Journal 281, 4061–4071.

Moriarty, N. W., Tronrud, D. E., Adams, P. D. & Karplus, P. A. (2016). Acta Crystallographica Section D-Biological Crystallography 72, 176–179.

Murshudov, G. N., Skubak, P., Lebedev, A. A., Pannu, N. S., Steiner, R. A., Nicholls, R. A., Winn, M. D., Long, F. & Vagin, A. A. (2011). Acta Crystallogr. Sect. D-Biol. Crystallogr. 67, 355–367.

Tronrud, D. E., Berkholz, D. S. & Karplus, P. A. (2010). Acta Crystallographica Section D-Biological Crystallography