Volume 14 January 2023

COMPUTATIONAL CRYSTALLOGRAPHIC NEWSLETTER

JUPYTER, POLYMERISATION, ALT. LOC.

Table of Contents

Editor

Nigel W. Moriarty: NWMoriarty@LBL.Gov

Phenix News

Announcements

New Phenix Release Imminent

The latest version of Phenix – 1.21 – will be released as soon as it has been tested. 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.

New Programs

Quantum Mechanical Restraints (QMR)

The recent addition of MOPAC aids in the use of a new procedure added to Phenix that will calculate restraints of a ligand *in situ* using Quantum Mechanical minimised geometries. The details can be found in the following citation: *Acta Cryst.* (2023) D**79**, https://doi.org/10.1107/S2059798323000025

West Coast Structural Biology Workshop

March 19-22, 2023 Asilomar Conference Center

Members of the Phenix team will be in attendance.

Crystallographic meetings and workshops

Macromolecular Crystallography School Madrid 2017 (MCS2017), May 5–10, 2017

Members of the Phenix team will be in attendance.

Expert Advice

Fitting Tip #23 – Alternate conformations always want to spread

Jane Richardson, Duke University, Nigel W. Moriarty, Lawrence Berkeley National Laboratory, and Daniel Keedy, CUNY Advanced Science Research Center

Introduction

In macromolecules at very high resolution, one sees increasingly many regions that have two, or even more, alternate conformations, with at least some of their atoms clearly distinct. They come in all sizes, from a single atom to many consecutive residues, and in a wide range of separation distance or conformational difference. They can be quite simple, as for the two proline puckers in Figure 1A, or quite confusing, as for the 4 hierarchical Lys alternates in Figure 1B with two distinct sidechains on each of two shifted backbones. If we want them to represent a physically possible ensemble, each individual local alternate must be internally consistent (free of geometry outliers) and consistent with the neighboring structure (free of all-atom clashes with its surroundings). Ensuring this is a far from trivial task. A more straightforward, but seldom done, task is to extend the ends of an alternate far enough not to produce geometry outliers.

Internal consistency

Either for sidechain or for backbone, alternates very often crisscross back and forth, as seen in Figure 2 below. That makes it tricky to be sure each alternate goes through the right subset of the atom peaks to be correctly bonded and with an acceptable rotamer conformation. If the two alternates have significantly different occupancies, their peak densities will be different for well separated atoms and add up for atoms much closer than the resolution. That is quite noticeable along the backbone in Figure 2, where the front carbonyl O is much stronger than the back one. Alt A (white) should be assigned to the higher-occupancy

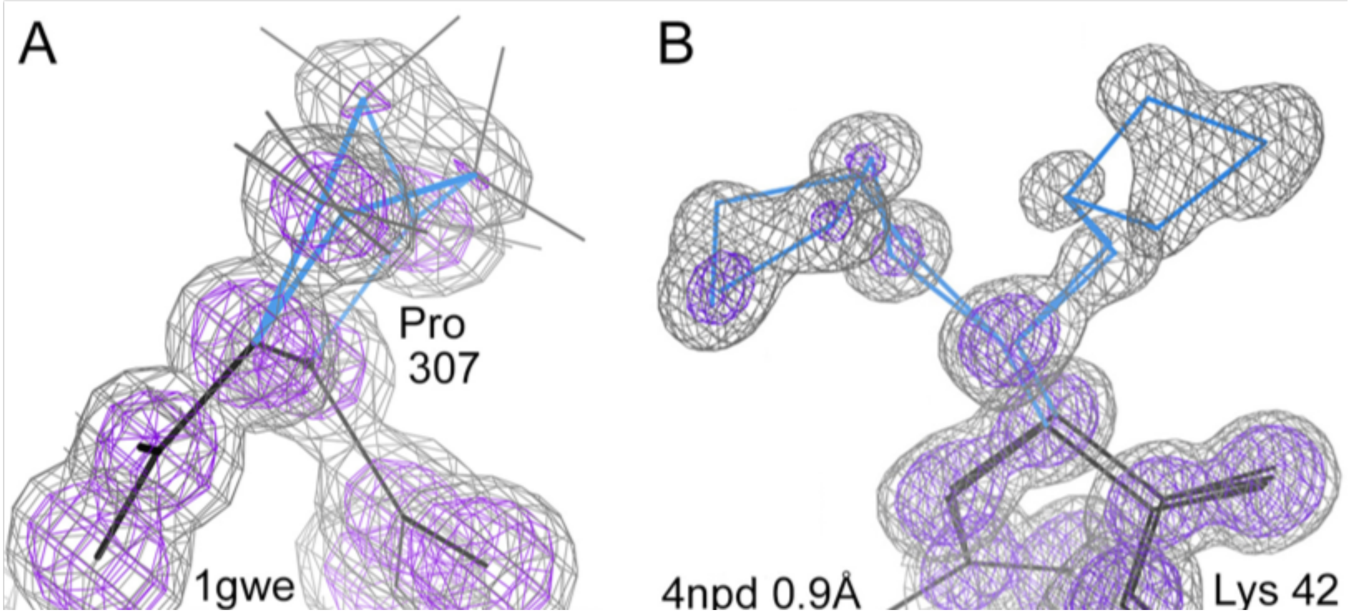


Figure 1: Some very high-resolution alternate conformations. A) A simple alternate for proline ring pucker, involving just two moving atoms, in 1gwe at 0.88Å (Murshudov 2002). B) Four hierarchical alternates of a lysine, two on each of two backbone alternates, in 4npd (Deis 2014). Contours are at 1.2σ (gray) and 3σ (purple).

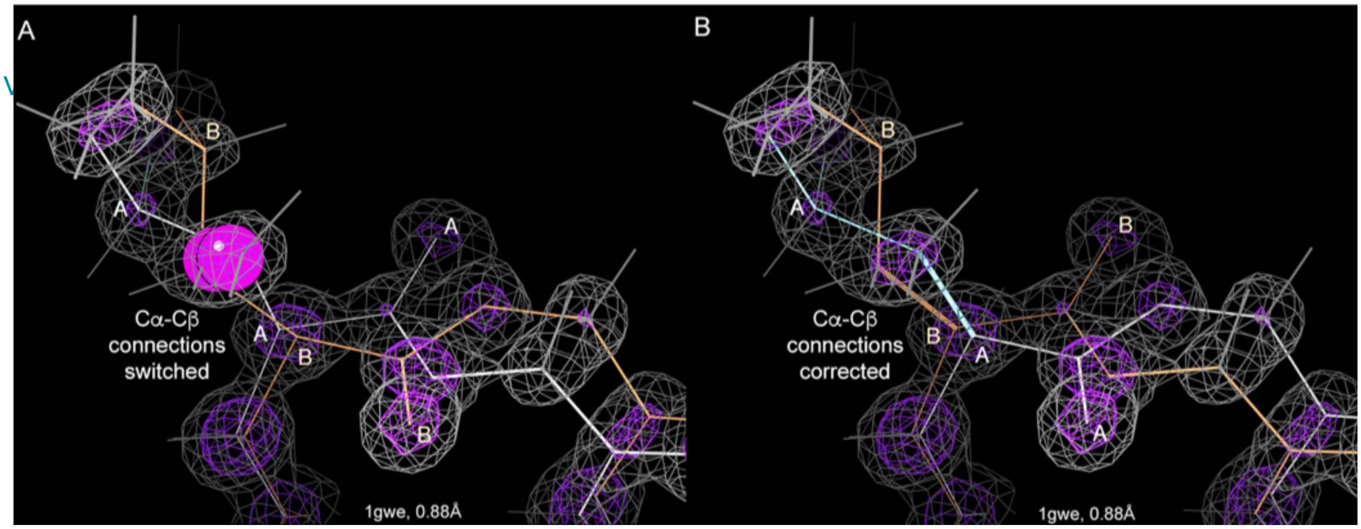


Figure 2: An example of internally inconsistent modeling of alternate conformations. A) Original PDB entry for Leu 105 of 1gwe, where Cβ deviation geometry outliers (magenta balls) flag the switched connectivities between the two alternates for the Cα-Cβ bonds. B) Alternate identities for backbone and Cβ atoms have been corrected, producing internally correct alternates and good geometry.

peaks, which is done wrong in the PDB entry (Fig. 2a). For the Cγ atoms on the sidechain, however, alt A is assigned correctly to the denser peak. This is one clue to the fact that the Cα and Cβ atoms in this residue have been connected across alternates rather than within them.

The other, even better clue is the severe Cβ deviation outliers (large magenta balls on the Cβ atoms); they combine the effects of multiple bad bond angles at the Cα (Lovell 2003). Wrong connections are also often flagged by bond length outliers unless refinement restraints are too tight. Wrong connections change dihedral angles and thus sometimes produce rotamer or Ramachandran outliers, but not in this case. In Fig 2b the alternate identities of the atoms have been corrected for all of the backbone and for Cβ, making each alternate internally consistent and removing the geometry outliers.

Consistency with surroundings

Consistency with the surroundings means that the individual alternates have no severe all-atom clashes with neighboring parts of the model (other parts of the macromolecules, het groups, or waters), and also that their polar atoms successfully make H-bonds with the nearby partners. As seen in Figure 3a, the backbone of 1gwe Leu 105 fails at that, since its alt A carbonyl O clashes with the nearby water that is much too close to H-bond. The close pair of waters are only 1.2Å apart, but have been modeled as HOH 181 and 182 instead of as alternates. If the waters are assigned as alternates, with the stronger one as alt A, and the backbone alternates are switched as in Figure 2B, then it all works consistently. Not only is there no clash in Figure 3b, but the alt A water can H-bond with the alt A backbone N (white) and the alt B water with the alt B backbone carbonyl O (peach) at excellent distances.

Most often, consistent interactions with the surroundings necessitates defining alternates for neighboring structure where the differences are too small to have been fitted originally, but are necessary to make a consistent ensemble. The small differences needed can often be confirmed and fitted using difference density peaks. In Figure 4A below, alt B of 1gwe Asp 410 is somewhat too close to its own backbone carbonyl O, and difference peaks suggest alternate orientations of the peptide, provided in the refit of Figure 4B.

Even more complicated interactions of alternate conformations arise, such as interacting cooperative networks of alternates, as seen in the Spa-N protein of 4npd (Deis 2014), or clashing alternates across a symmetry axis that cannot be

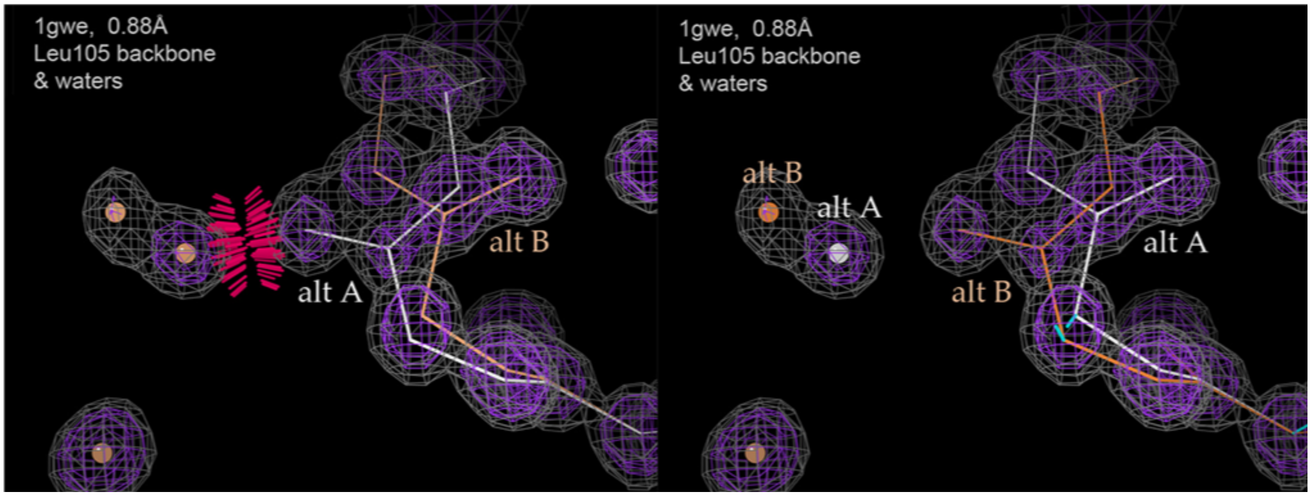


Figure 3: A case of backbone alternates inconsistent with neighboring waters. A) 1gwe Leu 105 alt A backbone clashing with a non-alternate water. B) With alternates assigned for the two close water peaks and the backbone alternates switched as in Fig, 2B above, now the protein alternates are consistent with the neighboring waters, forming two H-bonds rather than a clash.

expressed correctly without changing the spacegroup, such as shown in Figure 5. 1gwe is a homo-tetramer, but not quite perfectly symmetric, as demonstrated by the thorough overlap of Tyr 378 in what has to be labeled as the same alternate from neighboring molecules, but must actually be instantiated in each individual contact as one of each alternate. Note also that the two clashes at lower right mean alternates should be defined for that peptide, as also suggested by the very elongated density for the carbonyl O.

Don't stop too soon at alternate ends

Default treatment in nearly all software allows definition of alternates either for a sidechain ending at Cβ, which leaves the Cα as a single atom,

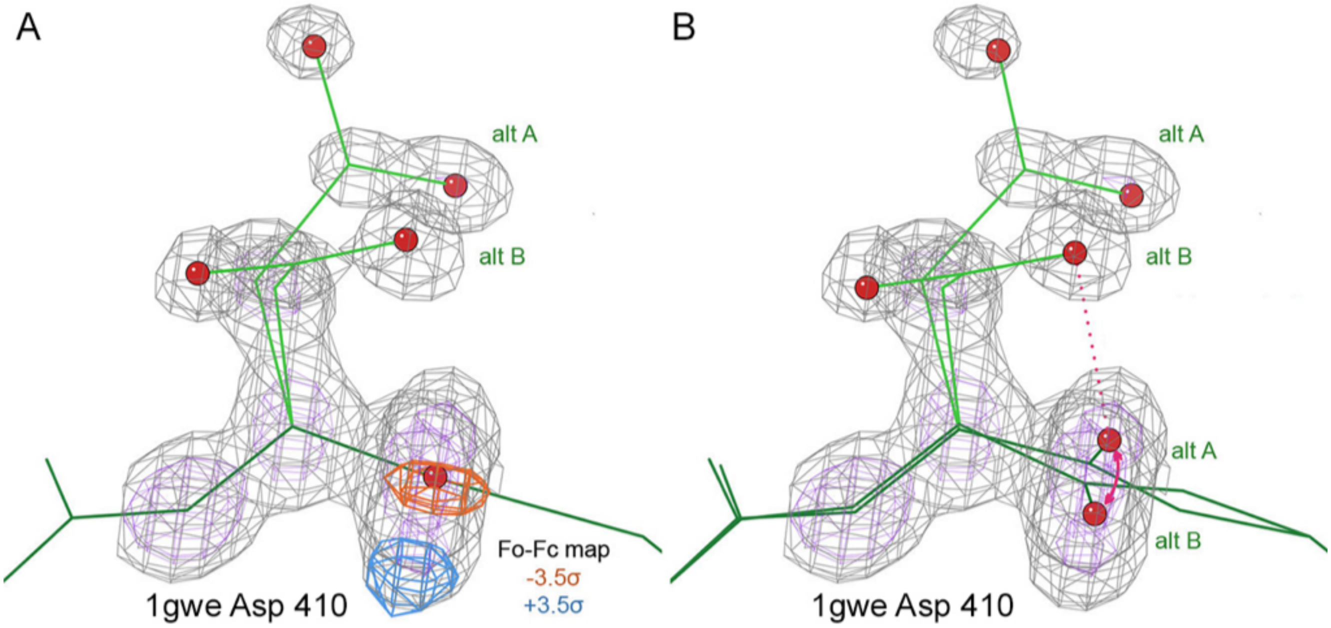


Figure 4: Sidechain alternates that need a backbone shift as well. A) Alternates for the sidechain of Asp 410 in 1gwe, with + (blue) and – (orange) difference density peaks that suggest the need for a small orientation shift of the peptide backbone. B) Providing those backbone alternates fits the 2Fo-Fc density better and removes the difference peaks.

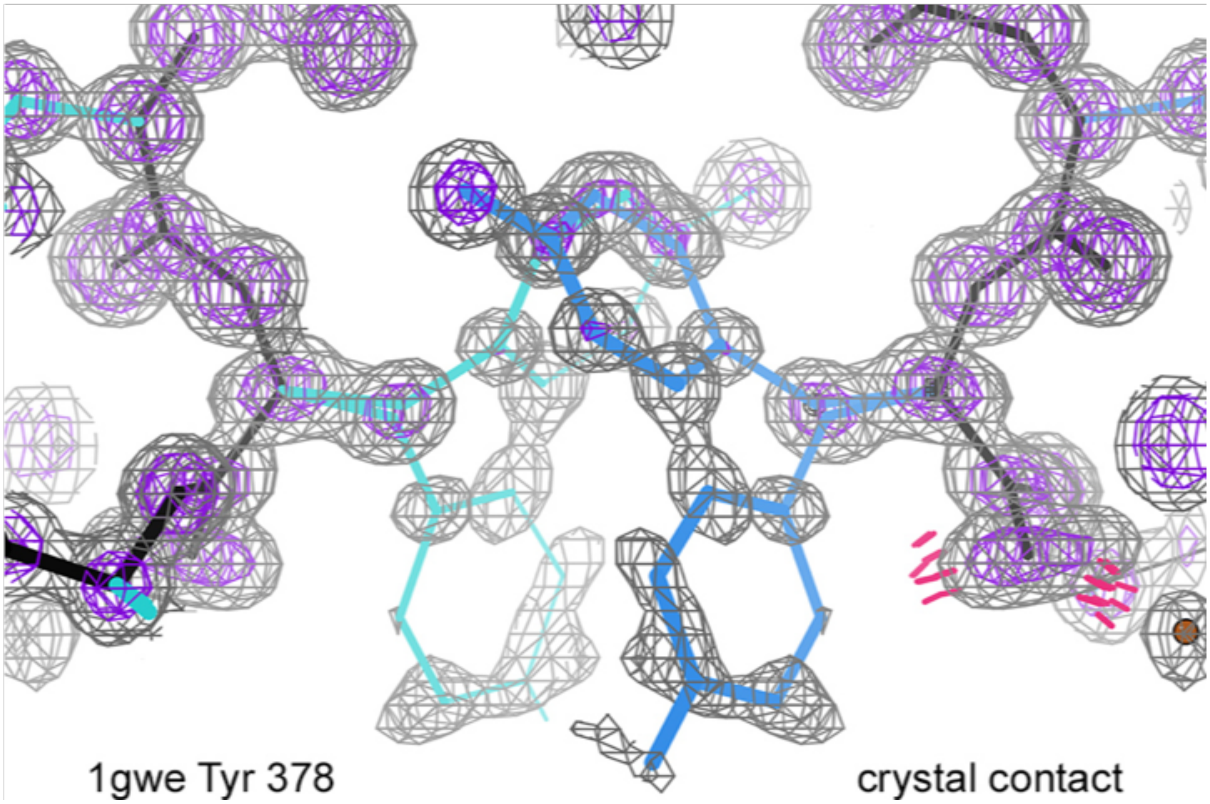


Figure 5: An alternate-conformation relationship that breaks the space-group definition across a 2-fold symmetry axis. The overlap of the alt A Tyr sidechains is so interpenetrated that the viewer thinks they are covalently bonded and does not show their clashes.

or for full residues ending at the near side of the peptide bonds, which leaves several atoms in the planar peptide as singletons. Refinement cannot possibly fix the resulting bond angle outliers, which would require separate positions of at least that next atom. These reasonable-sounding and easy-to-program definitions guarantee bad geometry unless the last alternate atom pairs are very close together (less than 0.2Å or preferably 0.15Å apart). Examples of this effect are shown in Figure 6A for a protein case and in Figure 6B for a nucleic acid case. The resulting distortions are often quite extreme, mostly in bond angles, but occasionally in bond lengths, Cβ deviations, or omega values.

Fortunately, this problem is fairly straightforward to fix. Make copies of the next atom or atoms to extend the definition of alternates, and let refinement make the geometry acceptable. Phenix has a utility that makes that process very easy. Some version of this should always be done when validation shows geometry outliers at the ends of alternates.

Backrubs for good backbone in single-residue alternates

When the problem is a sidechain alternate ended at Cβ, the bond-angle outliers often show as Cβ deviations, as in the example of Figure 7A, since those see combined effects of all the bond angles around the Cα atom. Such a problem clearly requires definition of alternates for the Cα atom, but unless the Cβ alternates are very close together, it actually requires spreading the alternates along the backbone up to single Cα atoms at n-1 and n+1. That can be done, and the smooth motions defined, by a backrub motion (Davis 2006) between the two conformations. The backrub is a hinge motion of the entire residue around the axis between Cα atoms at n-1 and n+1,

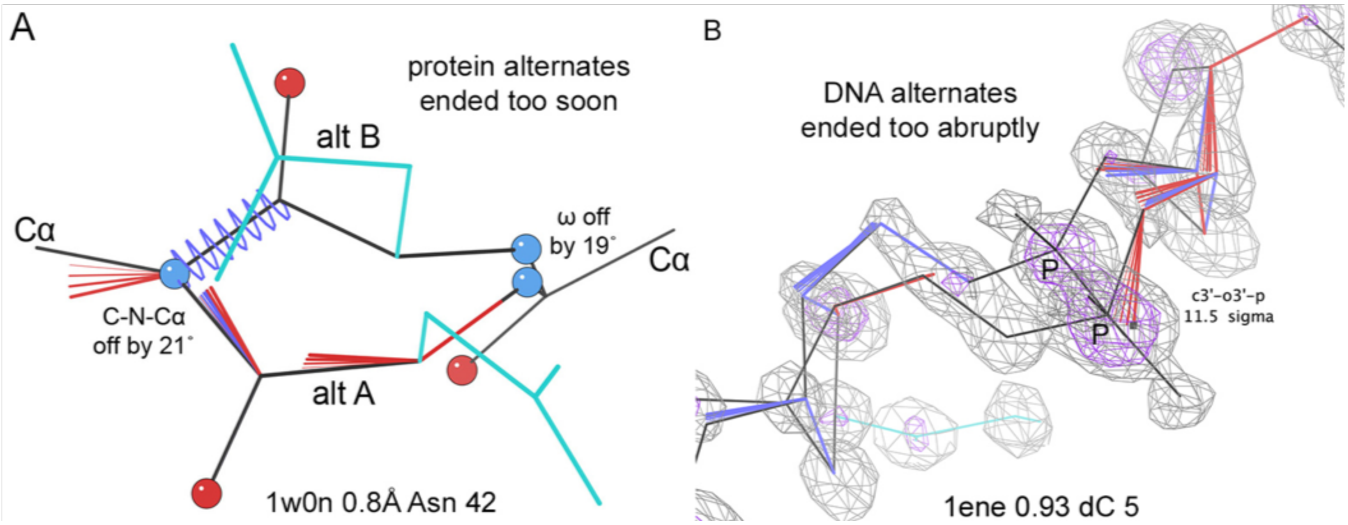


Figure 6: Two examples of the very common problem of serious geometry outliers where alternate conformations are ended too soon. A) Widely separated sidechain alternates of Asn 42 in 1w0n (Jamal 2004), with bond angle outliers up to 21°. B) Backbone DNA alternates at dC 5 of 1ene (Chiu 2000), with bond angle outliers up to 11.5σ (red and blue fans).

with minor rotations of the individual peptides to maintain their orientations and H-bonding, as seen for the correction in Figure 7B. The new backbone conformations instantiate the subtler changes that are strongly implied by the two clearly-resolved sets of sidechain-atom positions. Backrubs can be done in Coot (Emsley 2010) or in KiNG (Chen 2009) graphics and modeling programs, or can be approximated by duplicating the intervening atoms, shifting one slightly in the right direction, and refining.

Alternates must often be added when "waters" clash with nonpolar atoms

Density peaks fit as HOH but that clash with polar atoms are often actually ions or larger het groups. Those that clash with nonpolar atoms have sometimes displaced an atom of the macromolecule, but at high resolution they are often part of an unmodeled alternate conformation (Prisant 2020). In Figure 3d of that paper, an HOH that clashes with both S atoms and a Cβ of disulfide 91-186 in 3ajd at 1.27Å (Kuratani 2010) is actually one sulfur of an unmodeled alternate conformation of the disulfide. In Figure 1 of Headd 2013, reproduced in Figure 8 below, two waters in pear-shaped density clash with the nonpolar Cβ H atoms of Asp 9 in 1eb6 at 1.0Å. Panel 8B shows that they are actually the Oδ atoms of an alternate sidechain conformation in the most favored **m-20** rotamer of Asp, with the pear shapes pointing to the Cγ between them. A small backrub motion makes the fit perfect.

The bottom line

Each individual alternate conformation needs to be validated for rotamer and Ramachandran outliers, and especially for bond length and angle outliers, Cβ deviation outliers, and all-atom clashes. Then those problems need to be fixed, which sometimes involves correcting connectivity among the alternates, but which usually requires adding more alternates or more alternate atoms. This Fit Tip explains methods that help in making those corrections, such as defining atoms as duplicated even when their peaks are not separated, or using backrub motions for smooth transitions along the backbone. Almost all these things involve "spread" of alternates: new alternates, more extensive alternates, or more alternate atoms.

Although currently most alternate conformations in crystal structures are modeled manually, the software qFit (Riley 2021) seeks to automate this process, yielding multi-conformer models with a parsimonious set of 1-4 conformations for each residue. qFit remains in active development, and will benefit from ongoing codification of the lessons reported here, based on detailed inspection and remodeling of specific local examples.

[Note that although many of the examples shown here are from fairly early structures, these

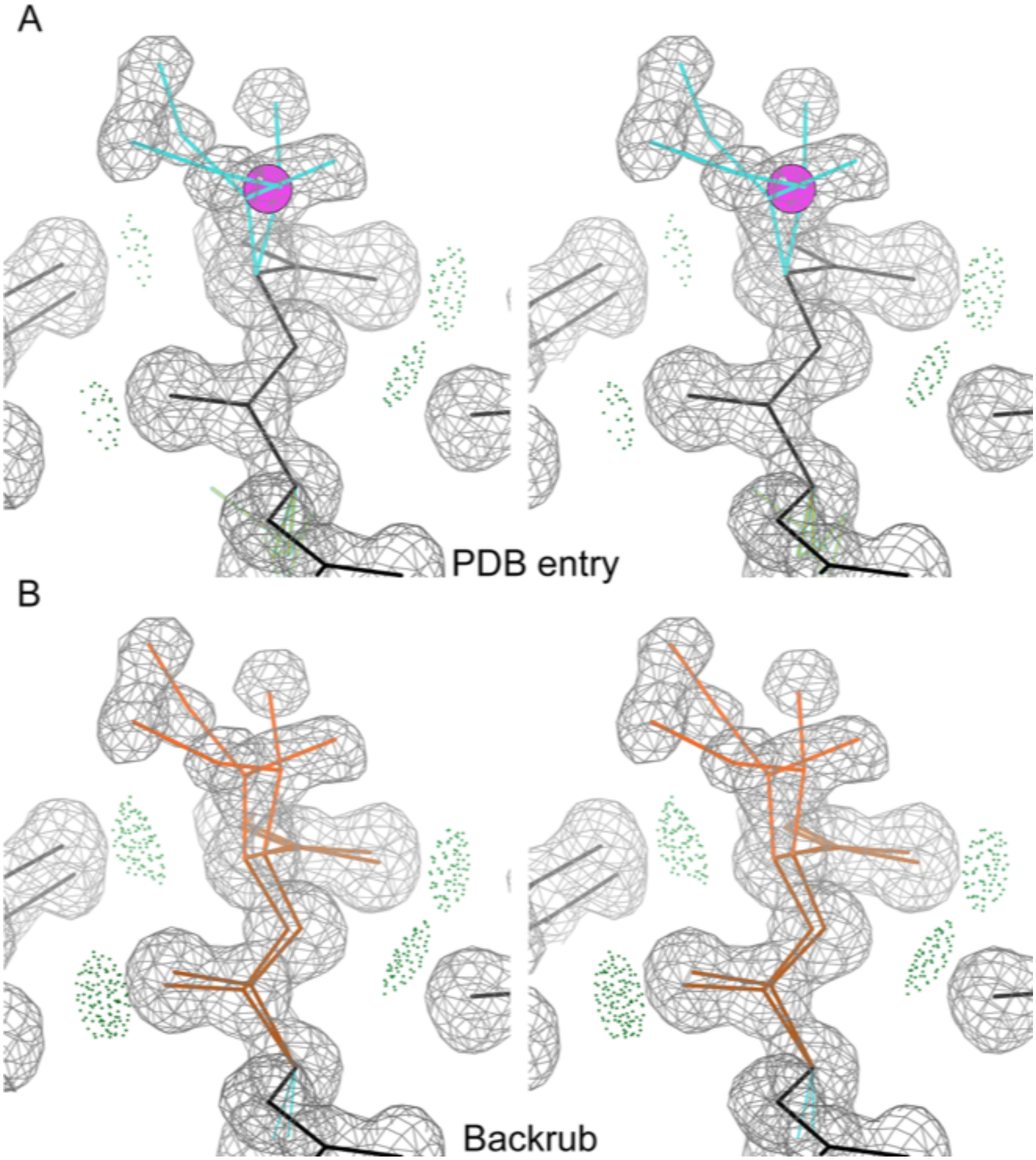


Figure 7: Stereo images of using a backrub motion to correct serious geometry outliers from not propagating sidechain alternates into the backbone. A) Clear alternate rotamers for the sidechain of Ile 47 in 1n9b at 0.9Å (Nukaga 2003), with a bad Cβ deviation outlier (magenta ball). B) Producing valid geometry by spreading the definition of alternates into the backbone and separating them with a small backrub motion between Cαs n-1 to n+1.

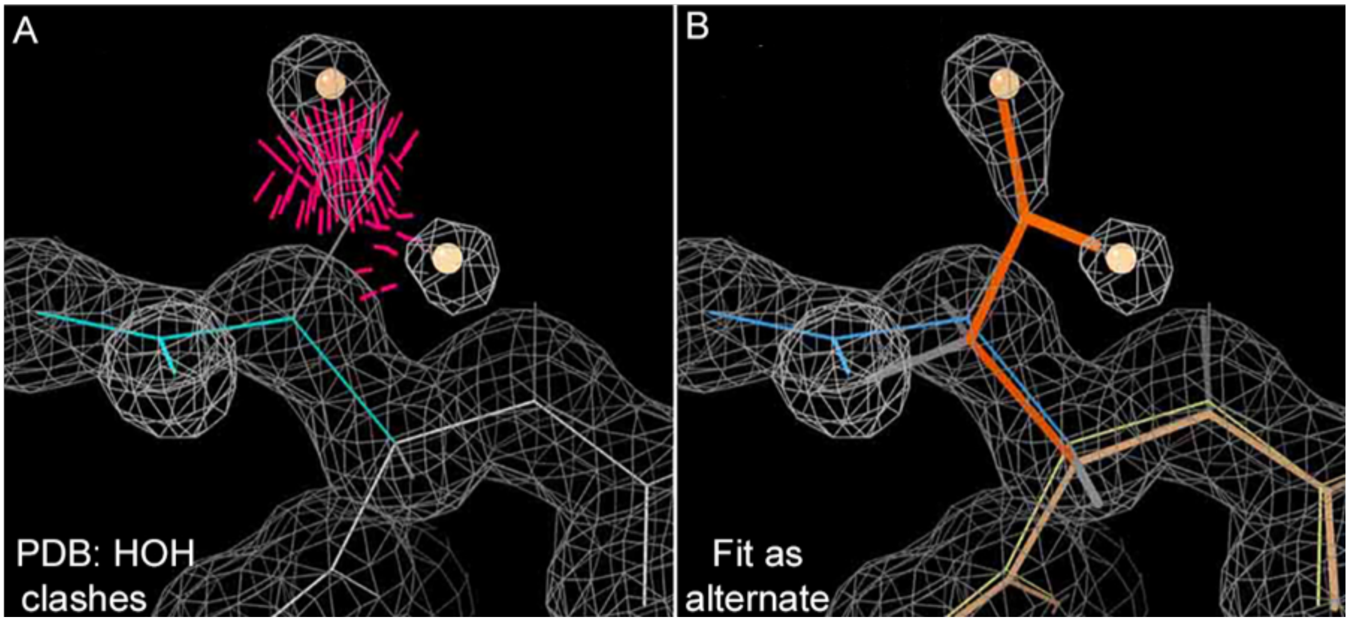


Figure 8: Stereo images of density peaks fit as HOH that clash with nonpolar atoms but are actually an unmodeled alternate conformation. A) Asp 9 and clashing waters as fit in 1eb6 (Mcauley 2001). B) The clashes are absent when those peaks are fit as an alternate conformation of the Asp.

alternate-conformation problems are still very common in current structures.]

Although currently most alternate conformations in crystal structures are modeled manually, the software qFit (Riley 2021) seeks to automate this process, yielding multi-conformer models with a parsimonious set of 1-4 conformations for each residue. qFit remains in active development, and will benefit from ongoing codification of the lessons reported here, based on detailed inspection and remodeling of specific local examples.

References:

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

Chiu TK, Dickerson RE (2000) Crystal structures of B-DNA reveal sequence-specific binding and groove-specific bending of DNA by magnesium and calcium, *J Mol Biol* **301**: 915-945

Davis IW, Arendall WB III, Richardson DC, Richardson JS (2006) The backrub motion: How protein backbone shrugs when a sidechain dances, *Structure* **14**: 265-274

Deis LN, Pemble CW, Qi Y, Hagarman A, Richardson DC, Richardson JS, Oas TG (2014) Multiscale conformational heterogeneity in staphylococcal protein a: a possible determinant of functional plasticity, *Structure* **22**: 1467-1477

Emsley P, Lohkamp B, Scott WG, Cowtan K (2010) Features and development of Coot, *Acta Crystallogr* **D68**: 486-501

Fuhrmann CN, Kelch BA, Ota N, Agard DA (2004) The 0.83Å resolution crystal structure of alpha-lytic protease reveals the detailed structure of the active site and identifies a source of conformational strain, *J Mol Biol* **338**: 999-1013

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

Jamal S, Boraston AB, Turkenburg JP, Tarbouriech N, Ducros VM-A, Davies GJ (2004) *Ab initio* structure determination and functional characterization of Cbm36: A new family of calcium-dependent carbohydrate binding modules, *Structure* **12**: 1177-1187

Kuratani M, Hirano M, Goto-Ito S, (2010) Crystal structure of *Methanococcus jannashii* Trn4 complexed with sinefungin, *J Mol Biol* **401**: 323-333

Lovell SC Davis IW, Arendall WB III, de Bakker PIW, Word JM, Prisant MG, Richardson JS, Richardson DC (2003) Structure Validation by Cα Geometry: ϕ,ψ and Cβ Deviation, *Proteins: Struct Funct Genet* **50**: 437-450

Mcauley KE, Jia-Xing Y, Dodson EJ, Lehmbeck J, Ostergaard PR, Wilson KS (2001 A quick solution: *Ab initio* structure determination of a 19 kDa metalloproteinase using Acorn, *Acta Crystallogr* **D57**: 1571-1578

Murshudov GN, Grebenko AAI, Brannigan JA, Antson AA, Barynin VV, Dodson GG, Dauter Z, Wilson KS, Melik-Adamyan WR (2002) The structures of Micrococcus lysodeikticus catalase, its ferryl intermediate (compound II) and Nadph complex, *Acta Crystallogr* **D58**: 1972-1982

Nukaga M, Mayama K, Hujer AM, Bonomo RA, Knox JR (2003) Ultrahigh resolution structure of a class A beta-lactamase: On the mechanism and specificity of the extended-spectrum SHV-2 enzyme, *J Mol Biol* **328**: 289-301

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

Riley BT, Wankowicz SA, de Oliveira SHP, van Zundert GCP, Hogan DW, Fraser JS, Keedy DA, van den Bedem H (2021) qFit 3: Protein and ligand multi-conformer modeling for X-ray crystallographic and single-particle cryo-EM density maps, *Protein Sci* **30**: 270-285

FAQ

crystallographic software users can be submitted to the editor at any time for consideration. Submission of text by email or word-processing files using the CCN templates is requested. The CCN is not a formal publication and the authors retain full copyright on their contributions. The articles reproduced here may

The wwPDB News

The computational crystallography community should be aware of the recent wwPDB news announcements.

Deprecation of FTP File Download Protocol in the PDB Archive

The FTP protocol for file downloads has been losing popularity over the years in favor of HTTP/S. There are many advantages of HTTP/S including speed, statelessness, security (HTTPS), and better support. Importantly during the past 2-3 years the main web browsers (Chrome and Firefox) have dropped support for the FTP protocol, which has effectively discontinued the FTP protocol for non-technical users.

Given that the majority of file download activity on the internet has moved to HTTP/S, wwPDB plans to deprecate FTP download protocol on November 1st 2024.

wwPDB has traditionally supported FTP, together with HTTP/S and RSYNC. Gradual deprecation of the FTP protocol, in favor of the HTTP/HTTPS protocol will be approached while maintaining support for the RSYNC protocol which offers additional functionality compared to the other 2 protocols.

As announced previously, we have introduced DNS names that are specific to the protocols:

- files.wwpdb.org for HTTP/S
- ftp.wwpdb.org for FTP. To be deprecated on November 1st 2024. Note that from September 2023 this DNS name will not accept HTTP/S traffic.
- rsync.wwpdb.org for RSYNC

Starting September 2023, wwPDB will start enforcing use of these updated DNS names for the preparation of FTP protocol deprecation.

Future Planning: PDB entries with extended CCD or PDB IDs will be distributed in the PDBx/mmCIF format only

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

CCD ID extension

CCD entries are currently identified by unique three-character alphanumeric IDs. At current growth rates, we anticipate running out of three-character IDs before 2024. After this point, the wwPDB will issue **five-character alphanumeric accession codes for CCD IDs** in the OneDep system. To avoid confusion with current four-character PDB IDs, four-character codes will not be used. Owing to limitations of the legacy PDB file format, PDB entries containing the new five character ID codes will only be distributed in PDBx/mmCIF format.

In addition, wwPDB has reserved a set of CCD IDs: 01 - 99, DRG, INH, LIG that will never be used in the PDB. These reserved codes can be used for new ligands during structure determination so that they can be identified as new upon deposition and added to the CCD during biocuration.

PDB ID extension

wwPDB will be extending PDB ID length to eight characters prefixed by ‘pdb’, e.g., pdb_00001abc. Each PDB entry has a corresponding Digital Object Identifier (DOI), often required for manuscript submission to journals and described in publications by the structure authors. Extended PDB IDs and corresponding PDB DOIs have been included in the PDBx/mmCIF formatted atomic coordinate files for all new and re-released entries since August 2021.

For example, PDB entry issued with 4-character PDB ID, 1abc, will have the extended PDB ID (pdb_00001abc) and corresponding PDB DOI (10.2210/pdb1abc/pdb), as 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
```

For example, PDB entry issued with 8-character PDB ID, pdb_00099xyz, after all 4-character IDs are consumed:

```
loop_
_database_2.database_id
_database_2.database_code
_database_2.pdbx_database_accession
_database_2.pdbx_DOI
PDB pdb_00099xyz pdb_00099xyz 10.2210/pdb_00099xyz/pdb
```

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

Resources

wwPDB is asking users and software developers to review their code and remove any current limitations on PDB and CCD ID lengths, and to enable use of PDBx/mmCIF format files. Example files with extended PDB and/or CCD IDs are available via github to assist code revisions, see https://github.com/wwPDB/extended-wwPDB-identifier-examples. To learn about PDBx/mmCIF, please visit https://mmcif.wwpdb.org/.

Please contact info@wwpdb.org with any questions.

Automatic Linking of Nonstandard Amino Acids

Nigel W Moriarty

MBIB, Lawrence Berkeley Lab, Berkeley, CA, USA

Correspondence email: NWMoriarty@LBL.Gov

Introduction

Macromolecular software such as the Phenix package (Liebschner *et al.*, 2019) needs to be aware of basic protein structure starting with the standard amino acids and their polymerisation. The latter can be achieved by definitions read from a file. The file is read from the installation and applied automatically to simplify the user experience. To accommodate this automation, there are a few rules regarding the preparation of the file(s). It should be noted that nucleic acids have a similar set of rules governing their polymerisation.

Understanding the automation steps provides insights into the manual procedures.

Standard Amino Acids

Rule One

Because a protein is a polymerised chain of amino acids these are arranged in sequence in the input file. Using the PDB format means that there should be no TER cards in the middle of the chain.

Rule Two

The atom names need to be the standard atom names for the main chain atoms shown in figure 1. Note that the atoms names are identical to the element symbol except for the Cα atom that is named CA. The main reason is that the link applied (see Schema 1) uses the atom names to apply bonds, angles and more (not shown). Schema 1 displays the first portion of the *trans*-peptide link definition. There is a similar *cis*-peptide link that is applied if the torsion angle of the input is within 45° of the *cis* conformation. This can be changed by the user as well the ability to specify which peptide links should be restrained to the *cis* conformation. The ideal values for both *cis*- and *trans*-peptide are affected by a following PRO amino acids so these situations have another pair of links (Moriarty & Adams, 2021).

```
data_link_TRANS
loop_
_chem_link_bond.link_id
_chem_link_bond.atom_1_comp_id
_chem_link_bond.atom_id_1
_chem_link_bond.atom_2_comp_id
_chem_link_bond.atom_id_2
_chem_link_bond.type
_chem_link_bond.value_dist
_chem_link_bond.value_dist_esd
 TRANS 1 C 2 N single 1.329 0.014
loop_
_chem_link_angle.link_id
_chem_link_angle.atom_1_comp_id
_chem_link_angle.atom_id_1
_chem_link_angle.atom_2_comp_id
_chem_link_angle.atom_id_2
_chem_link_angle.atom_3_comp_id
_chem_link_angle.atom_id_3
_chem_link_angle.value_angle
_chem_link_angle.value_angle_esd
 TRANS 1 O 1 C 2 N 123.000 1.600
 TRANS 1 CA 1 C 2 N 116.200 2.000
 TRANS 1 C 2 N 2 H 124.300 3.000
 TRANS 1 C 2 N 2 CA 121.700 1.800
```

Schema 1: Portion of the *trans*-peptide link as described in the restraints library

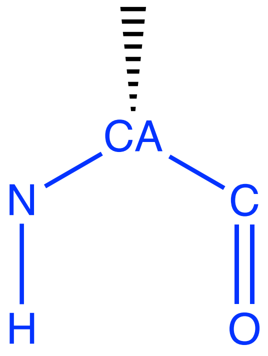


Figure 1: Atom naming convention for amino acids

Rule Three

Naturally, the coordinates for the amino acids must be reasonable. One reason is to ensure that gaps in the protein chain are modeled as

```
data_comp_list
loop_
_chem_comp.id
_chem_comp.three_letter_code
_chem_comp.name
_chem_comp.group
_chem_comp.number_atoms_all
_chem_comp.number_atoms_nh
 OMY OMY (betaR)-3-chloro-beta-hydroxy-L-tyrosine L-peptide 24 15
```

Schema 2: Truncated _chem_comp loop of the restraints file for nonstandard amino acid OMY.

such and not inadvertently distorted by peptide links. The distance criterium for linking requires the N atom be within 3Å of the preceding C atom.

Rule Four

It can be helpful, but not necessarily required, if the field for `_chem_comp.group` is set to, for example, the string "peptide". Schema 2 is a reduced example. This is, of course, the case for amino acid restraints distributed with Phenix.

Checking

Once the process of parsing and generating peptide (and other) links is performed the information is (or can optionally be) written to a `.geo` file for inspection.

Another method of checking is the addition of `LINK` records in the PDB format output and `_struct_conn` loop in the mmCIF format output. The former is added solely for viewing purposes while the latter is required for deposition to the PDB.

Nonstandard Amino Acids

Many nonstandard amino acids are derived from the standard set with only changes to the side chain. This enables to use of the same mechanism in Phenix to apply the peptide link if the rules are followed.

Rule One (ordered in input file) is clearly required so the program knows which other amino acids are linked along with Rule Three (distance limit). Rule Two (atom names) can be done easily in the restraints generation program eLBOW (Moriarty *et al.*, 2009). The first is to use the entry in the Chemical Components Library (CCL) via the `--chemical_components` option as it contains the atom names specified by the PDB. In general, the PDB has specified the atom names in the convention mentioned earlier. The second approach uses the `--template` option to graph match the atoms from a PDB file to specify the atom names in the generated restraints.

A recent push to include restraints for all nonstandard amino acids listed in the CCL will greatly reduce the need for using eLBOW in future Phenix releases. The restraints have been systematically added to the GeoStd (Manuscript in preparation).

Other polymers

A special case are entities that have peptide-like linkages but are not technically amino acids. The same rules can be used to automatically apply the peptide link to any entity pair. Naturally, the chemistry should be consistent.

Conclusions

This communication describes the mechanism used to streamline the polymerisation of amino acids – both standard and nonstandard – along with the information to apply it to other entities. Future communications will describe the more manual approaches to more complex and uncommon situations.

References

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.

Moriarty, N. W. & Adams, P. D. (2021). *Computational Crystallography Newsletter* **12**, 47–52.

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

Edit CCTBX code in live Jupyter cells with your favorite text editor

Blaine H.M. Mooers[a,b,c]

[a]*Department of Biochemistry and Molecular Biology, University of Oklahoma Health Sciences Center, Oklahoma City, OK 73104*
[b]*Stephenson Cancer Center, University of Oklahoma Health Sciences Center, Oklahoma City, OK 73104*
[c]*Laboratory of Biomolecular Structure and Function, University of Oklahoma Health Sciences Center, Oklahoma City, OK 73104*

Correspondence email: blaine-mooers@ouhsc.edu

Introduction

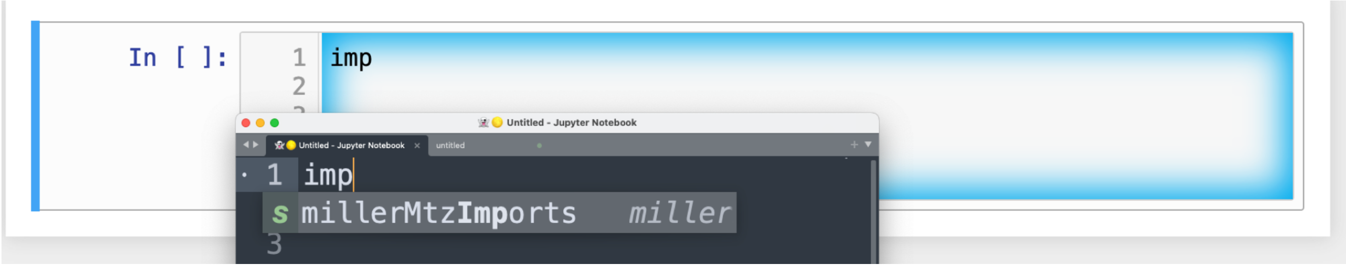
The editing tools available in web-based computational notebooks (e.g., Jupyter Notebook, JupyterLab, Google Colab Notebook) are generally inferior to the editing tools available in popular text editors and Integrated Development Environments (IDEs). By adding the GhostText extension to the web browser and a server to one of several leading text editors, it is possible to send the text from the browser through a WebSocket to the server in the text editor. Thus, it is possible to edit the contents of a computational notebook cell from inside a text editor. Changes made in the text editor instantly appear in the notebook and vice versa. By applying the power of a text editor to computational notebooks, experienced developers can continue to use familiar editing commands and tools. This article presents the features of GhostText and how to use it with leading text editors. A library of CCTBX snippets has been shared on GitHub for use with these text editors; the library can also be used to edit Python scripts.

The Problem

Like many structural biologists, I use several web-based platforms for interactive computing (e.g., Jupyter Notebook, JupyterLab, Google Colab). Usually, I use these computational notebooks for prototyping functions or for running someone's else literature programming document. This document may be a tutorial for a software package, a tutorial on a computational topic, an appendix to a book, or part of the supplemental materials to a manuscript. There are over 10.5 million Jupyter notebooks available on GitHub as of January 30, 2023 (https://github.com/parente/nbestimate), which reflects the popularity of sharing notebooks in this fashion.

While these notebooks support the use of libraries of code snippets, this support is not as advanced as what is commonly available in text editors. For example, these notebooks do not support tab triggers and tab stops with code snippets. The last two features are available in the leading text editors and are missed when moving from a text editor or IDE to a computational notebook. In addition, the support for snippets in Jupyter Notebook (this is the application: the document is called Jupyter notebook--note the lowercase *n*) and JupyterLab has to be enabled by installing third-party software. The snippet extensions are not transferable between Jupyter Notebook and JupyterLab.

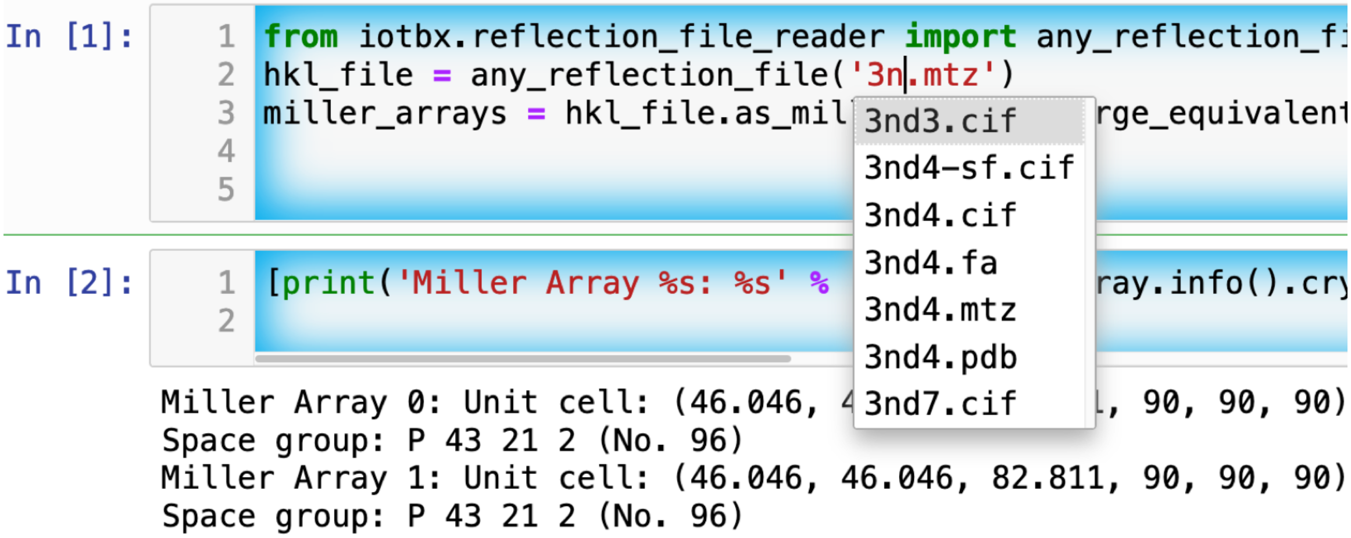
As a quick reminder, tab triggers insert chunks of computer code after entering the tab trigger name and hitting the TAB key. The tab trigger name can be as short as several letters. Many text editors and IDEs have pop-up menus that aid the selection of the correct tab trigger. Tab stops are sites within the code snippet the cursor advances to after entering TAB again. These sites often have placeholder values that can be edited. Sites with identical placeholder values can be mirrored so that a change in value at one site is propagated to the other tab stops with the same placeholder value. The absence of tab stops can increase the number of bugs introduced by the developer by overlooking parameter values in the code snippet that need to be changed to adapt the snippet to



```
In [ ]: imp
```

```
imp
```

Figure 1: Example of a tab trigger being entered in Sublime Text 3 editor and appearing in a Jupyter Notebook cell. A pop-up menu lists the available snippets. The list was narrowed to one snippet by the entry of three letters.



```
In [1]: from iotbx.reflection_file_reader import any_reflection_fi
        hkl_file = any_reflection_file('3n.mtz')
        miller_arrays = hkl_file.as_mil              rge_equivalent

In [2]: [print('Miller Array %s: %s' %               ray.info().cry

        Miller Array 0: Unit cell: (46.046, 4        l, 90, 90, 90)
        Space group: P 43 21 2 (No. 96)
        Miller Array 1: Unit cell: (46.046, 46.046, 82.811, 90, 90, 90)
        Space group: P 43 21 2 (No. 96)
```

Figure 2: Two code cells with executed Jupyter code cells.

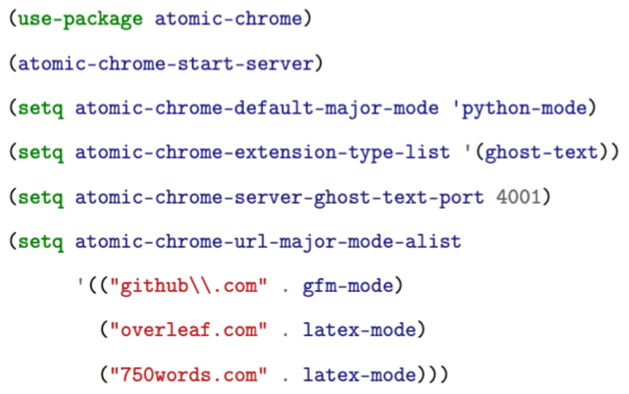
the current situation. The lack of support for tab triggers and tab stops in computational notebooks can dampen the enthusiasm of experienced developers for using computational notebooks. Of course, one solution is write an extension in JavaScript that supports tab triggers and tab stops.

An easy solution

Another approach is to send the text in the active code cell to a powerful text editor on your local computer via the browser extension known as GhostText (https://ghosttext.fregante.com/). GhostText is a Javascript program developed by Federico Brigante, a prolific JavaScript developer. Versions of the extension are available for the Google Chrome, Firefox, Edge, Opera, and Safari. The extension for the Google Chrome browser works in the Brave browser, and the extension for Firefox works in the Waterfox and Icecat browsers.

The text editor also needs to be extended with a server that enables two-way communication with the web page via a WebSocket. While edits made on the browser side of the WebSocket are immediately sent to an open page in the Text Editor and vice versa, the text editor's snippets and other editing tools only work in the text editor (Figure 1). The connection can be closed from either side of the WebSocket.

A Jupyter notebook with two code snippets from the cctbxsnips library for Sublime Text 3 editor are shown in Figure 2. The two code cells have



```
(use-package atomic-chrome)
(atomic-chrome-start-server)
(setq atomic-chrome-default-major-mode 'python-mode)
(setq atomic-chrome-extension-type-list '(ghost-text))
(setq atomic-chrome-server-ghost-text-port 4001)
(setq atomic-chrome-url-major-mode-alist
      '(("github\\.com" . gfm-mode)
        ("overleaf.com" . latex-mode)
        ("750words.com" . latex-mode)))
```

Code listing 1: Emacs lisp to configure atomic-chrome.

been run and the output from the second cell is printed in the notebook. The first cell is being edited to change the name of the mtz file that is to be read. A pop-up menu in Jupyter has appeared with a list of candidate file names.

The servers for the editors are editor specific. The following text editors are supported: Visual Studio Code, Vim, NeoVim, Emacs, Atom, and Sublime Text 3. GhostText was developed initially for Sublime Text 3, so Sublime Text can serve as a positive control even when another editor in the list is your favorite editor. For example, the server for Emacs is provided by the atomic-chrome package that is available in the Milkypostman's Emacs Lisp Package Archive (MELPA) and on GitHub (https://github.com/alpha22jp/atomic-chrome). My configuration for atomic-chrome in my Emacs initialization file (e.g., init.el) is listed in code listing 1.

The third line in Code listing 1 sets the default Emacs mode (equivalent to a programming language scope): I set it to Python for Jupyter code cells. Atomic-chrome uses text-mode by default. You can change the default mode to other programming languages that you may use in Jupyter, like Julia or R. The last three lines specify the Emacs mode to be used when text is imported from the text areas of webpages on github.com, Overleaf.com and 750words.com. Similar configuration options are available in the other text editors, or you manually change the language scope for the window with the text imported from Jupyter.

Using GhostText elsewhere

As suggested above, GhostText works beyond notebook cells, wherever a web page has a text area. For example, it works in LaTeX documents on Overleaf, a popular web service for academic writing in LaTeX documents. (Note that since December 2022, you must select the "Source (legacy)" tab rather than the default Source tab in the left-hand corner of the document for GhostText to work correctly. GhostText also works on the website 750words.com, a web service that provides a clutter-free platform for writing and saves the daily writing in the cloud. I have configured Emacs to open text from 750words in the LaTeX mode (see the second to last line in the above code listing 1), so I can insert LaTeX code snippets.

Keyboard shortcuts

Ghost Text provides a keyboard shortcut for the browser to open or close the connection to the text editor. These shortcuts keep the developer's hands on the keyboard and avoid breaks in context by moving the hand to the mouse. The shortcuts are as follows: macOS, command-shift-K; Linux, control-shift-H; and Windows, control-shift-K.

Troubleshooting

I have been using GhostText daily since mid-May 2022 with Emacs and either Google Chrome or Firefox; I have encountered three difficulties. First, other servers inside Emacs can occupy the port for GhostText and block the atomic-chrome server. I have to kill the offending server or restart Emacs. Second, saving the text in the Emacs buffer to a file can cause the text to become out of sync with the text in the editor and on the web page. The lack of updating can lead to lost work; making intermediate saves of the text on the web page is safer than a local file. Third, multiple editors with GhostText servers installed can compete for the same WebSocket. This problem is solved by closing the editor that is not in current use or configuring its server to use an alternate WebSocket.

Support for using GhostText with CCTBX

To support the use of GhostText to edit electronic notebooks containing code from the CCTBX library, we have formatted a collection of cctbx snippets for the VSCode, Atom, Sublime Text 3, Vim, NeoVim, and Emacs. Snippets are available for the UltiSnips, Snipmate, and neosnippets plugins for Vim and NeoVim. The snippets are available on GitHub (https://github.com/MooersLab/MooersLab/blob/main/README.md#cctbxsnips-for-editors) Sublime Text 3 has the most effortless setup. Emacs provides the highest degree of customization. Note that the snippet library cannot be used with the program nteract (https://nteract.io/)--an easy-to-use desktop application for editing and running Jupyter notebooks offline – because nteract does not support snippet libraries. The snippet library is not restricted to electronic notebooks and can aid the development of Python scripts in plain text files.

References

Mooers, B. H. M. (2021). Computational Crystallography Newsletter 12, 268-276.

Left blank intentionally