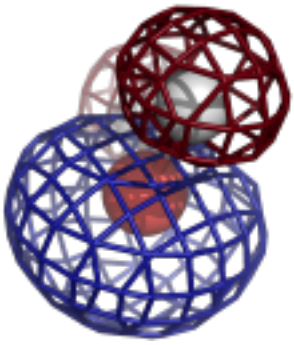
VOLUME FIVE



COMPUTATIONAL CRYSTALLOGRAPHY NEWSLETTER

JANUARY MMXIV

BIOMT, XFEL, IMAGECIF, BIG DATA, DIVCON

Table of Contents

Editor

Nigel W. Moriarty, NWMoriarty@LBL.Gov

PHENIX News

New programs

phenix.composite_omit_map

Nat Echols & Pavel Afonine

A new program for omit map calculation (Bhat, 1988) is now available. The implementation features two alternatives for de-biasing the phases:

- An iterative procedure that generates an F(model) map, and repeatedly computes a composite omit map which then provides starting phases for the next cycle. This is done over boxes encompassing the entire unit cell. Because no refinement is required, the procedure is extremely fast and is the default mode of operation.

- A more conventional refinement-based procedure, similar to the composite omit map implementation in CNS (Hodel et al. 1992, Brunger et al. 1997). Simulated annealing is available as an option; the omitted atoms will be harmonically restrained to prevent the structure from collapsing into the omit regions. This method is significantly slower but can be parallelized over multi-core computers or supported queuing systems.

Although the program is primarily intended for composite omit maps covering the entire unit cell, the refinement procedure can also be used to generate a simple omit map, for instance showing de-biased density for a ligand. This functionality is essentially identical to running phenix.refine with custom parameters, but via a simplified interface.

These features mostly supersede the equivalent modes in the AutoBuild wizard,

with the exception of the "iterative build" omit maps (Terwilliger et al. 2009).

References

Bhat, T. N. (1988) J. Appl. Cryst., 21, 279-281.

Brunger, A. T., Adams, P. D., Clore, G. M., DeLano, W. L., Gros, P., Grosse-Kunstleve, R. W., Jiang, J.-S., Kuszewski, J., Nilges, M., & Pannu, N. S. (1998). Acta Crystallographica D. 54, 905–921.

Hodel, A., Kim, S.-H., & Brunger, A. T. (1992). Acta Crystallographica A. 48, 851–858.

Terwilliger, T. C., Grosse-Kunstleve, R. W., Afonine, P. V., Moriarty, N. W., Adams, P. D., Read R. J., Zwart P. H., & Hung L.-W. (2009). Acta Crystallographica D. 64, 515–524.

New features

Optimizing phenix.autosol and phenix.hyss for weak SAD data

Tom Terwilliger, Randy Read, Gábor Bunkóczi and Nat Echols

You might notice some big differences in HySS and AutoSol in the new 2014 versions of Phenix (starting with dev-1610 and the soon-to-be-released Phenix-1.8.5). The big changes are parallelization, the use of Phaser completion in HySS, iterative Phaser completion using maps and models in AutoSol, and on-the-fly optimization of parameters in AutoSol for cases with weak anomalous signal. Now HySS and AutoSol can solve structures with very weak SAD phase information that previous versions could not solve.

A new powerful and parallel Hybrid Substructure Search (HySS)

The first thing you might notice in HySS is that it can use as many processors as you have on your computer. This can make for a really quick direct methods search for your anomalously-scattering substructure.

You might notice next that HySS now automatically tries Phaser completion to find a solution if the direct methods approach does not give a clear solution right away. Phaser completion uses the likelihood function to create an LLG map that is used to find additional sites. This is really great because Phaser completion in HySS can be much more powerful than direct methods in HySS. Phaser completion takes a lot longer than direct methods completion but it is now quite feasible, particularly if you have several processors on your computer.

The next thing you might notice in HySS is that it automatically tries several resolution cutoffs for the searches if the first try does not give a convincing solution. Also HySS will start out with a few Patterson seeds and then try more if that doesn't give a clear solution.

HySS now considers a solution convincing if it finds the same solution several times, starting with different initial Patterson peaks as seeds. The more sites in the solution, the fewer duplicates need to be found to have a convincing solution.

Putting all these together, the new HySS is much faster than the old HySS and it can solve substructures that the old HySS could not touch.

An AutoSol optimized for weak SAD data

The new AutoSol is specifically engineered to be able to solve structures at low or high resolution with a very weak anomalous signal.

One feature you may notice right away is that the new AutoSol will try to optimize several choices on the fly. AutoSol will use the Bayesian estimates of map quality and the R-value in density modification to decide which choices lead to the best phasing. AutoSol will try using sharpened data for substructure identification as well as unscaled data as input to AutoSol and pick the one leading to the best map. AutoSol will also try several smoothing radii for identification of the solvent boundary and pick the one that gives the best density modification R-value.

You'll also notice that AutoSol uses the new parallel HySS and that it can find substructures with SAD data that are very weak or that only have signal to low resolution. You can use any number of processors on your machine in the HySS step (so far the parallelization is only for HySS, not

the other steps in AutoSol, but those are planned as well).

The biggest change in AutoSol is that it now uses iterative Phaser LLG completion to improve the anomalously-scattering substructure for SAD phasing. The key idea is to use the density-modified map (and later, the model built by AutoSol) to iterate the identification of the substructure. This feature is amazingly powerful in cases where only some of the sites can be identified at the start by HySS and by initial Phaser completion. Phaser LLG completion is more powerful if an estimate of part of the structure (from the map or from a model) is available.

The new AutoSol may take a little longer than the old one due to the heavy-atom iteration, but you may find that it gives a much improved map and model. Give it a try!

Crystallographic meetings and workshops

Biophysical Society 58th Annual Meeting, February 15-19, 2014

An IYCr2014 symposium entitled "Celebrating 100 Years of Crystallography: X-Rays Are Photons Too" will be of interest to all crystallographers. This year the annual Biophysics101 session will be on tips for biophysicists who want to use the crystallographic technique. Check the website for details as the date approaches.

Keystone Symposium, "Frontiers in Structural Biology," March 30-April 4, 2014

This meeting will be held in Snowbird, Utah.

44th Mid-Atlantic Macromolecular Crystallography Meeting and 11th Annual SER-CAT Symposium, April 23-26, 2014

Keynote speakers include Bi-Cheng Wang and Wayne Hendrickson.

American Crystallographic Association (ACA) Annual Meeting, May 24-28, 2014

This year the meeting is in Albuquerque, New Mexico.

WK.01 Joint Neutron and X-ray Structure Refinement using Joint Refine in PHENIX

A workshop in association with the ACA Annual Meeting. Go to the ACA website http://www.amercrystalassn.org/2014-wk.01 for details.

Macromolecular Crystallography School - MCS2014, May 26-31, 2014

A school for students and researchers. Go to http://www.xtal.iqfr.csic.es/MCS2014/index.html for details.

2014 (Pacific) Northwest Crystallography Workshop, June 20-22, 2014

The conference organizer is P. Andrew Karplus, Department of Biochemistry and Biophysics, School of Life Sciences, Oregon State University, Corvallis, OR 97331, NWCW2014@science.oregonstate.edu.

Andrew is pleased to announce that the 2014 (Pacific) Northwest Crystallography Workshop will be held this summer at the Linus Pauling Science Center, Oregon State University. To avoid conflicts with other meetings we are restoring the meeting to its traditional month of June. In particular, it will be held Friday evening through Sunday noon the weekend of the June 20th through the 22nd.

Registration and abstract submission will begin on February 1st. Detailed information can be found at http://oregonstate.edu/conferences/event/NWCW2014/. Talks will be selected from the abstracts submitted for posters. To encourage an informal atmosphere, we will give preference to students and post-docs.

Corvallis is located between the mountains and the sea with each in easy driving distance. It is surrounded by wine country and other recreational opportunities. It is just off the I-5 corridor and can be accessed from either the Portland or Eugene airports.

Gordon Research Conference on Diffraction Methods in Structural Biology: Towards Integrative Structural Biology, Gordon Research Seminar, Bates College, Lewiston, ME, July 26-27, 2014

A seminar series preceding the GRC meeting.

Gordon Research Conference on Diffraction Methods in Structural Biology: Towards Integrative Structural Biology, Gordon Research Conference, Bates College, Lewiston, ME, July 27-August 1, 2014

A very interesting and important meeting for protein crystallographers.

23rd International Union of Crystallography (IUCr) Congress, August 5-12, 2014

This year the congress will be in Montreal, Canada.

5th Murnau conference on Structural Biology – Focus Topic: Signal Transduction Sept 10-13, 2014

Location: Murnau am Staffelsee, Germany www.murnauconference.de/2014/index.html

Expert advice

Fitting Tip #7 – Getting the Pucker Right in RNA Structures

Swati Jain, Gary Kapral, David Richardson and Jane Richardson, *Duke University*

Building good RNA backbone models into electron density is quite a challenge, but there is now a very effective trick to get one of the difficult parts right – the ribose ring pucker. Ribose rings in RNA are known, from small-molecule or high-resolution larger RNA structures, to adopt just two main conformations: C3'-endo and C2'-endo, each with a tightly defined range for the δ dihedral angle (figure 1). However, to distinguish between these two conformations from the electron density alone at more typical 2.5-3.5 Å resolution is virtually impossible. As a result, most RNA structures contain pucker errors, usually accompanied by steric clashes and geometry outliers. Therefore, it is very desirable to model the pucker correctly as early in the structure building process as feasible.

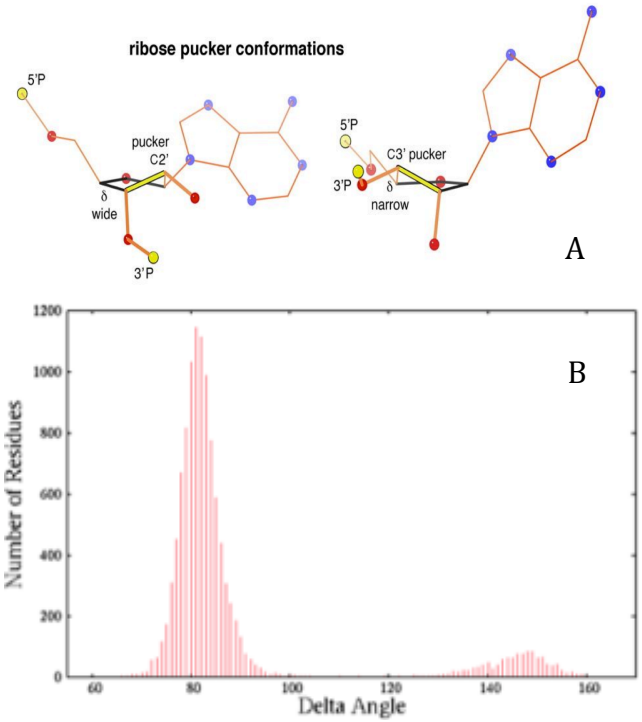


Figure 1: (A) The two main ribose pucker conformations found in RNA structures, C2'-endo (left) and C3'-endo (right). (B) Occurrence frequency plotted against the δ dihedral for the RNA11 dataset after applying clash and B-factor < 60 filters on the RNA backbone. The two occupied δ ranges are: C3'-endo: 60°-105°, C2'-endo: 125°-165°.

The Pperp Test

The trick to identify the correct ribose pucker is the 3' Pperp test, which has the great advantage of working from the two most clearly seen RNA structural features in the electron density: the phosphate and the base. This test involves judging the relative position of the 3'P (the phosphate P atom in the 3' direction) to the plane of the base, or better, to the vector extension of the glycosidic bond (C1'-N1/N9). This test is similar to the zp distance used in the program 3DNA (Lu 2003) to distinguish between A form and B form DNA helices, but is more general and can be applied to any RNA residue, even in loops, bulges, or junctions. When the ribose ring has a C3'-endo pucker, the perpendicular distance

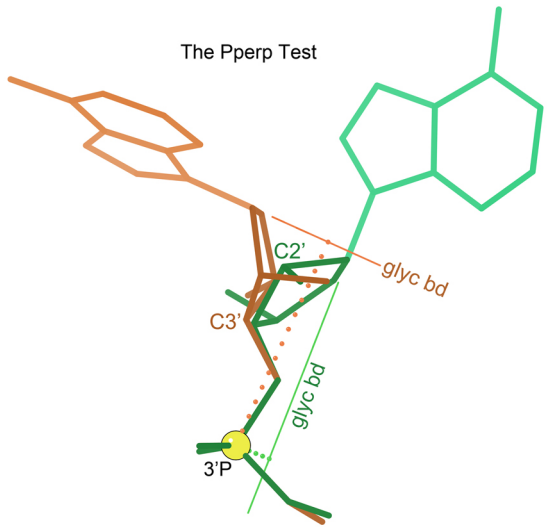


Figure 2: Judging the length of the perpendicular (dotted lines) dropped from the 3'P to the vector extension of the glycosidic bond (C1'-N9/N1). That distance is long for C3'-endo pucker (orange) and short for C2'-endo pucker (green): 4.6Å vs 1.1Å for this pair of examples.

is longer than when the ribose ring has a C2'-endo pucker (figure 2). The specific protocol to apply the Pperp test is to drop a perpendicular from the 3'P to the vector extension of the glycosidic bond (C1'-N1/N9), and measure whether its length is > 2.9Å (for C3'-endo pucker) or < 2.9Å (for C2'-endo pucker). However, the long vs. short perpendicular can be judged quite well by eye. This test is based on the strong empirical correlation between the length of the perpendicular and the pucker of the ribose ring, which becomes almost entirely clean as additional residue-level quality filters are applied (figure 3). This allows one to use a distance of 2.9Å as the cutoff distance to distinguish between the two puckers and then make a correction if the indicated pucker does not match the modeled δ value. In addition, the values of some bond angles and of the other backbone dihedral angles (besides δ) are pucker-specific, and the ability to apply the Pperp diagnosis in Phenix and then use pucker-specific values in the target function significantly improves RNA refinement behavior as well as validation statistics for the final structure.

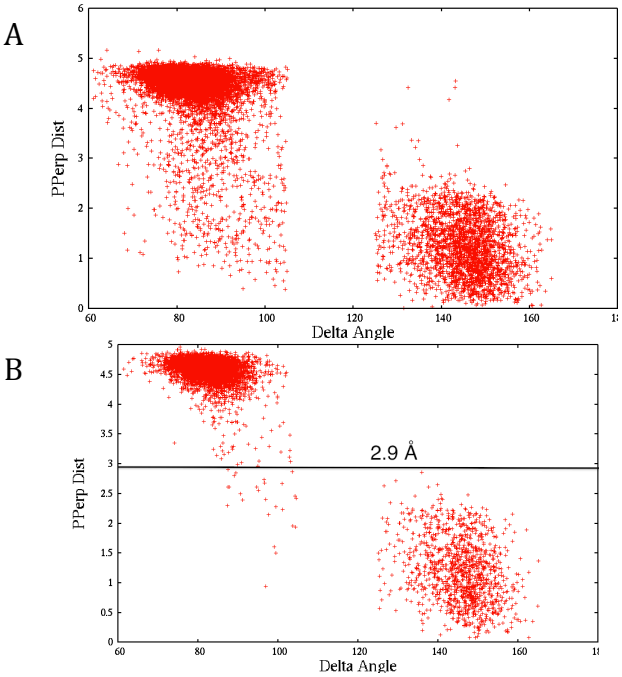


Figure 3: Pperp distance vs δ angle, plotted in the valid δ ranges for C3'-endo and C2'-endo pucker. Data is from the RNA11 dataset: (A) no residue-level filters; (B) with clash, B-factor < 60, and ε filters.

Correcting Pucker Outliers

If a residue fails the Pperp test, i.e. the Pperp distance and the δ angle indicate different puckers, the residue is flagged as a pucker outlier. Such problems can be corrected using several available tools. RNA Rotator in KiNG (Chen 2009) allows the user to change the backbone dihedral angles manually and choose a different backbone conformation from a list of known conformers. RNABC (Wang 2008) and the RCrane plugin in Coot (Keating 2012) rebuild the ribose atoms keeping the base and phosphate fixed. The most powerful option, however, is a new automated tool called ERRASER (Chou 2012a) that locally rebuilds the full RNA backbone, using Phenix refinement and a step-wise assembly procedure in Rosetta that accounts for fit to the electron density in its scoring function. It is scriptable in Phenix if Rosetta is also installed (Chou 2012b).

Common Mistakes to Avoid

The most common problem with RNA sugar pucker is to model a C2'-endo pucker as a C3'-endo, presumably because more than 80% of

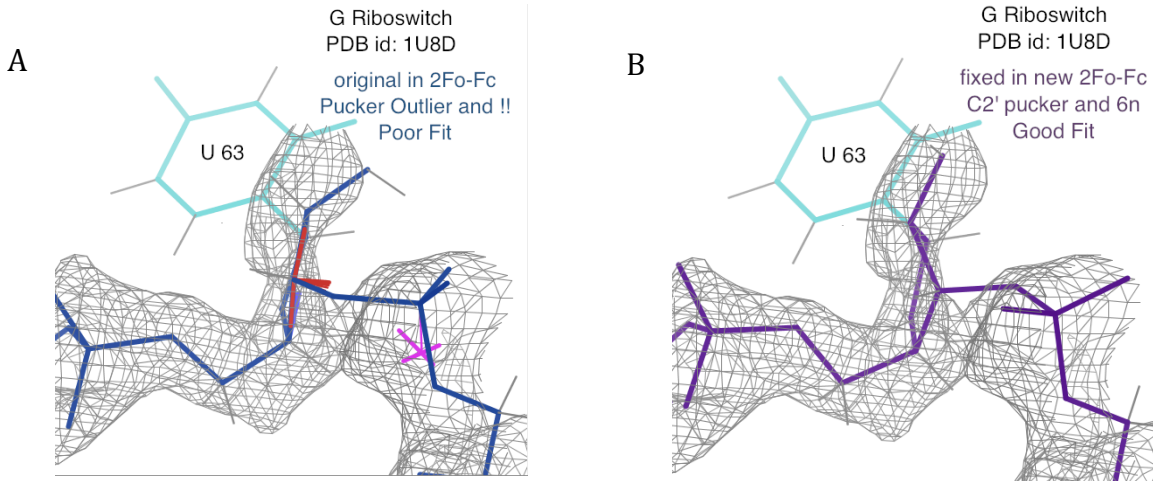


Figure 4: Residue U 63 of the G Riboswitch (PDB id: 1U8D) in the electron density. (A) Original model, with outlier flags for pucker (magenta cross), backbone conformer (**!!**), and bond angles (red and blue fans). (B) Fixed residue, with C2'-endo pucker, a **6n** backbone conformer, and better fit to the ribose density. (For both, the base and the C5' at lower right are in good density, but lie behind the clipping plane.)

the residues in RNA are C3'-endo pucker, and is thus the default expectation. An example of such an error is in the earliest G riboswitch structure (PDB id: 1U8D; Batey 2004). The resolution is 1.95Å, and it can actually be seen that the modeled pucker for U 63 does not fit the density well (figure 4A) - but pucker correction was difficult back then. Residue 63 is also an outlier for bond angles, ε, and RNA backbone conformer (Richardson 2008). Figure 4B shows our rebuild of the structure using C2'-endo pucker, which also fixes the bond-angle and ε outliers, moves the backbone to a recognized **6n** conformer, and fits the density better. A later dataset at 1.32Å (PDB id: 4FE5) confirms this and other changes.

A less common source of pucker error can result from trying too hard to force corresponding residues in different structures/chains into the same conformation. An example of such a pucker outlier occurs in the structure of a ternary complex of human exo-ribonuclease protein, histone mRNA stem loop, and stem-loop binding protein (PDB id: 4HXH (Tan 2013)). Of the two RNA chains in the asymmetric subunit, chain A is bound to both the proteins but chain D is bound only to the exo-nuclease. Residue 12 in chain A is at the protein-RNA interface of the stem-loop with the stem-loop binding protein, and it has C2'-endo pucker. As a result, residue 12 in chain D was also modeled with C2'-endo pucker. This residue is flagged as an outlier by the Pperp test (figure 5A). We rebuilt it as a C3'-endo pucker, which gets rid of the backbone clash as well as the pucker outlier (figure 5B). The corrected structure is deposited in the PDB as 4L8R.

Conclusion

It is healthy to keep in mind that a pucker outlier could possibly be real and the structure strained, but that will be extremely uncommon and there should be very strong evidence for it. Getting the ribose pucker right is highly beneficial, because the difference between the C3'-endo and C2'-endo flings around the base and the backbone dramatically (figures 1A and 2). If the pucker is fit incorrectly, then refinement is forced into large local distortions. Now the Pperp test has made pucker diagnosis quite easy for manual as well as automated use.

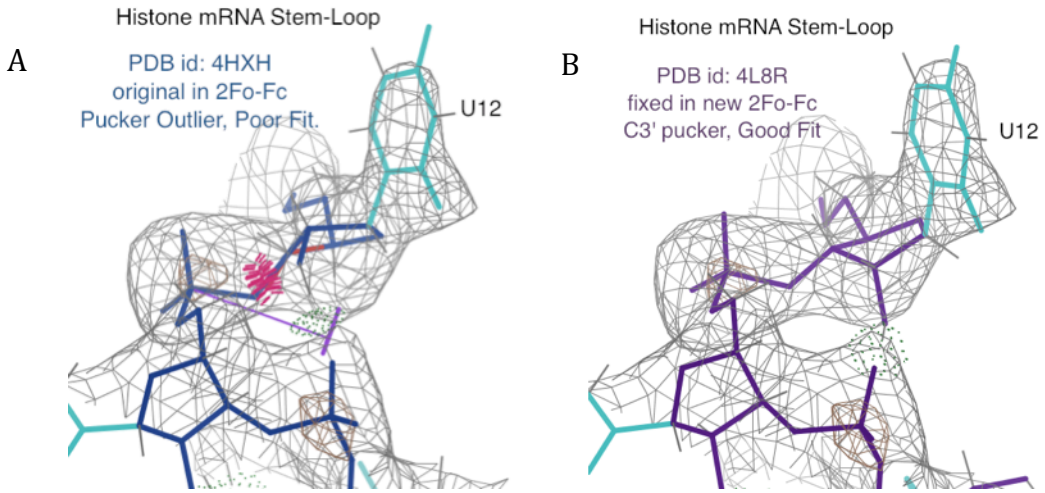


Figure 5: Residue 12 in Chain D of the histone mRNA stem-loop (1HXH) in its electron density. (A) The original residue is an outlier for pucker (purple cross), bond angle (red fan), and steric clashes (pink spikes). (B) The rebuilt model keeps the H-bond, has C3'-endo ribose pucker, no outliers, and fits the density better.

References

Batey RT, Gilbert SD, Montange RK (2004) Structure of a natural guanine-responsive riboswitch complexed with the metabolite hypoxanthine, *Nature* **432**: 411-415

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

Chou F-C, Sripakdeevong P, Dibrov SM, Hermann T, Das R (2012a) Correcting pervasive errors in RNA crystallography through enumerative structure prediction, *Nature Meth* **10**: 74-76

Chou F-C, Richardson J, and Das R (2012b) ERRASER, a powerful new system for correcting RNA models, *Comp Cryst Newsletter* **3**: 35-36

Keating KS, Pyle AM (2012) RCrane: semi-automated RNA model building, *Acta Crystallogr* **D 68**: 985-995

Lu XJ, Olson WK (2003) 3DNA: a software package for the analysis, rebuilding and visualization of three-dimensional nucleic acid structures, *Nucleic Acids Res* **31**: 5108-5121

Richardson JS, Schneider B, Murray LW, Kapral GJ, Immormino RM, *et al.* (2008) RNA backbone: Consensus all-angle conformers and modular string nomenclature (an RNA Ontology Consortium contribution), *RNA* **14**: 465-481

Tan D, Marzluff WF, Dominski Z, Tong L (2013) Structure of histone mRNA stem-loop, human stem-loop binding protein, and 3'hExo ternary complex, *Science* **339**: 318-321

Wang X, Kapral G, Murray L, Richardson D, Richardson J, Snoeyink J (2008) RNABC: Forward kinematics to reduce all-atom steric clashes in RNA backbone, *J Math Biol* **56**:253-278.

Phenix tools for interpretation of BIOMT and MTRIX records of PDB files

Youval Dar[a], Pavel V. Afonine[a], Paul D. Adams[a,b]

[a]*Lawrence Berkeley National Laboratory, Berkeley, CA 94720*

[b]*Department of Bioengineering, University of California at Berkeley, Berkeley, CA 94720*

Correspondence email: ydar@lbl.gov

Introduction

Currently, *Phenix* (Adams et al 2010) tools that require the atomic model as input expect the model to contain the atoms corresponding to the entire asymmetric unit. If non-crystallographic symmetry (NCS) is present and all NCS copies are assumed to be identical (strict NCS) then one NCS copy can be considered as independent. Furthermore, the entire contents of the asymmetric unit (ASU) can be generated by applying appropriate NCS transformations to the independent copy. These transformations are rotation matrices and translation vectors. Protein Data Bank (PDB) (Berman et al 2000, Bernstein et al 1977) model files store these transformations in MTRIX records in REMARK 350 section of the file header. As a first step towards handling strict NCS in *Phenix*, particularly as constraints for structure refinement in *phenix.refine* (Afonine et al 2012) we have implemented tools that can read PDB file with single NCS copy and corresponding MTRIX records, generate the entire ASU and output it as expanded PDB file or use corresponding objects internally. The corresponding command line tool is *phenix.pdb.mtrix_reconstruction* and the underlying implementation is located in *multimer_reconstruction.py* file of *iotbx.pdb* module.

When the biologically functional macromolecular assembly (biological unit) is known, the information that provides the generation of the biological unit is recorded in BIOMT records in REMARK 350 section of PDB file header. Similarly to MTRIX, these are rotation matrices and translation vectors that are to be applied to all chains recorded in the PDB file. Note that the biological unit may be a copy or multiple copies of the ASU or portions of the ASU. Handling of these records is coded in multimer_reconstruction.py as well. The command line tool *phenix.pdb.biomt_reconstruction* is designed to convert the content of PDB file into an expanded set of atoms representing the biological unit.

To exercise the new tools we surveyed the entire PDB and applied them to entries that contain non-trivial (non-unit) MTRIX and BIOMT records. As part of the PDB survey we checked that matrices in MTRIX and BIOMT records are proper rotation matrices (Rotation matrix R is a proper rotation when *Transpose(R) = Inverse(R), Determinant(R) = 1*). We also checked for general formatting issues of these records and more. A summary is presented in table 1.

BIOMT and MTRIX records in PDB (as of Oct. 2013)

Records processing issues include:

- The number of BIOMT matrices is different than the serial number of matrices or have missing records. In many cases this can happen if the identity matrix is listed several times with the

Table 1: PDB survey summary

Number of records surveyed:	93,252
MTRIX	
- Files with no MTRIX records	88,522
- Files with non-trivial matrix MTRIX records	2,202
- Files with non-trivial matrix which also have structure factor (SF) records	1,736
- Non-trivial files with SF records that contains only one NCS independent part	157
- Non-trivial files with SF records that have processing issues	19
BIOMT	
- Files with trivial BIOMT records (only identity matrix)	47,262
- BIOMT records with more than the identity matrix	11,960
- Files with BIOMT records processing issues (see details below)	26,207

same serial number. For example, the PDB file *4dzi* has two biomolecules and the instruction in the file, REMARKS 300 and 350, call for applying the identity matrix on two different chains, for the different biomolecules.

- Missing information in CIF file, such as files where the string "******" replaces some sigma F_meas_sigma values, causing *phenix.cif_as_mtz* to fail (*3m8l*, *2btv*, *2w0c*, *3dpr*).
- Unknown chemical element type in present in PDB file, "HETATM xxx UNK UNX ... " (*2zah, 2wtl, 4fp4*)
- Empty miller_array because R-free-flags are zero in mtz file (*2uu7, 2uwa, 2bnl*)
- "CifBuilderError: Miller arrays _refln.F_calc_1 and _refln.phase_calc are of different sizes " (*1wbi, 2xts, 3nap*)
- "CifBuilderError: Space group is incompatible with unit cell parameters" (*1x33, 2bny*)
- Analysis could not be continued because more than half of the data have values below 1e-16" (*1qez*)
- Symmetry issues, can't convert cif to mtz (*2xvt*)
- CifParserError: lexer error 1 : Unexpected character (*3zll*)
- CifParserError: error 4 : Unexpected token, at offset 11 near release, : unexpected input (*4bl4*)
- Improper transformation, tested using *Transpose(R) approximately equal Inverse(R)* and *Determinant(R)=1* with *eps*. In table 1 we used *eps=0.01*.

Only about 2.4% of all PDB files contain MTRIX information where the rotation matrices are not the identity matrix. Only in 157 files a single NCS copy is present (see Table 2) along with MTRIX records necessary to generate the entire ASU contents. Out of the 157 files with a single NCS copy 8 had processing issues.

There are about 13% of all PDB files for which valid BIOMT records are available. For those files the biological assembly reconstruction function can be used to obtain the complete set of biological assembly atoms coordinates.

If a PDB file contains only one NCS copy, then non-trivial MTRIX records must be provided so the complete ASU content can be generated. R-factors calculated from such expanded file are expected to match the published values (in REMARK 3 record) within some reasonably small tolerance (usually a few percentage points of relative difference).

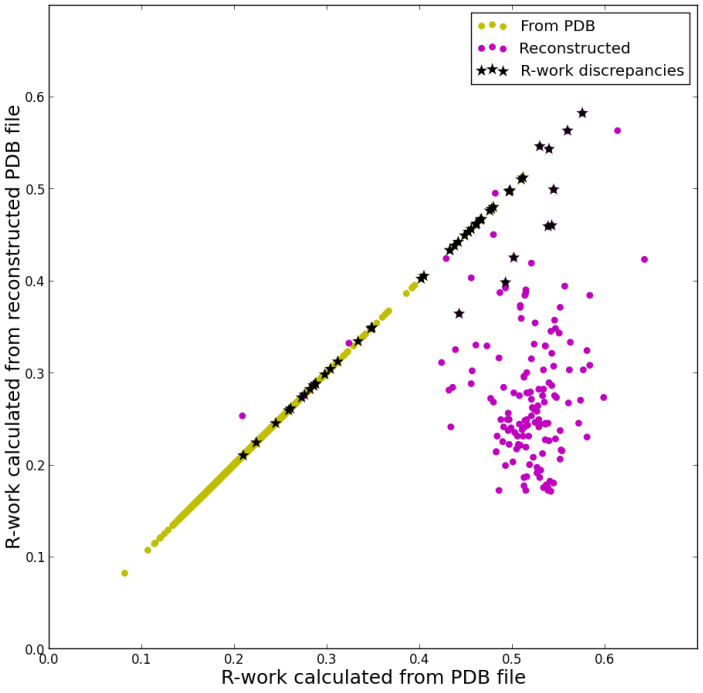


Figure 1: Scatter plot of the R-work calculated using the expanded PDB file (using MTRIX records) (blue) vs. R-work calculated using original PDB file (yellow). In black are 48 files for which the calculated R-work, either from the PDB file (38) or from the reconstructed file (10), differs in more than 50% than the R-work value published in the PDB file records. 1717 files were used for this out of 1736 for which the structure factors are available (19 files were excluded due to processing issues).

If a PDB file contains the entire ASU of atoms, then a single NCS copy is not readily available. R-factors calculated from such PDB file taken as is are expected to closely match published values.

To assert the above statements true and exercise our new tools we selected all PDB entries that have non-trivial MTRIX records and have experimental structure factors available (1736 total). The MTRIX records have a flag indicating whether a file contains only one NCS copy or the entire ASU. For each entry, with only one NCS copy, we calculated two R-factors: using the PDB file as is and after applying MTRIX records. The reconstruction function does not apply the MTRIX records when a file contains the entire ASU.

Table 2: 157 PDB file containing a single NCS (non crystallographic symmetry) copy. PDB columns contain the file name and reported R-work value. ASU is the R-work calculated from the PDB file, and NCS is the value calculated from the expanded NCS. Entries in the table are sorted (from large to small) by R-factor discrepancy between published and calculated from reconstructed ASU content values. PDB files with processing errors are listed at the end.

PDB ID	PDB	ASU	NCS	PDB ID	PDB	ASU	NCS	PDB ID	PDB	ASU	NCS	PDB ID	PDB	ASU	NCS
3p0s	0.209	0.560	0.563	2gtl	0.288	0.521	0.315	3zfe	0.240	0.518	0.231	2xbo	0.247	0.508	0.244
2wws	0.230	0.530	0.546	2g33	0.360	0.563	0.333	4fts	0.259	0.561	0.267	1h8t	0.246	0.497	0.249
2xpj	0.308	0.576	0.582	2zzq	0.242	0.535	0.268	4fsj	0.262	0.574	0.270	5msf	0.188	0.530	0.186
2bfu	0.230	0.545	0.499	2ws9	0.275	0.516	0.249	3raa	0.294	0.543	0.286	4jgz	0.286	0.456	0.288
3n7x	0.278	0.540	0.543	1c8n	0.253	0.502	0.278	2c4q	0.190	0.541	0.182	3vbo	0.227	0.490	0.225
2iz9	0.200	0.543	0.460	2w4z	0.296	0.521	0.271	1qjy	0.233	0.491	0.241	2ztn	0.305	0.577	0.303
1dwn	0.288	0.539	0.459	1x9t	0.306	0.529	0.282	4hl8	0.352	0.510	0.359	1b35	0.228	0.540	0.226
1ei7	0.195	0.443	0.364	3r0r	0.225	0.488	0.249	3tn9	0.265	0.527	0.258	6msf	0.195	0.531	0.194
2wzr	0.230	0.493	0.398	4gb3	0.330	0.525	0.354	3es5	0.248	0.514	0.241	4aed	0.279	0.518	0.278
2vf1	0.273	0.502	0.425	1vsz	0.380	0.456	0.403	1js9	0.238	0.484	0.231	3vbs	0.273	0.477	0.272
1m1c	0.266	0.493	0.392	2g34	0.365	0.551	0.343	4gbt	0.256	0.522	0.262	3vbr	0.257	0.496	0.256
1llc	0.374	0.482	0.495	2c4y	0.192	0.542	0.171	4fte	0.239	0.539	0.245	3vbf	0.236	0.496	0.237
1dzl	0.280	0.557	0.394	1z7s	0.224	0.501	0.203	2x5i	0.288	0.534	0.282	3ra2	0.244	0.517	0.243
2vf9	0.311	0.521	0.419	1ddl	0.151	0.515	0.172	2izn	0.197	0.527	0.191	3kz4	0.328	0.536	0.329
3ntt	0.252	0.547	0.348	2bq5	0.293	0.599	0.273	1zba	0.183	0.513	0.177	1za7	0.245	0.536	0.244
4ar2	0.280	0.509	0.371	2c4z	0.191	0.539	0.172	1r2j	0.247	0.434	0.241	1vb2	0.261	0.524	0.260
4gmp	0.269	0.546	0.357	3fbm	0.294	0.546	0.275	1k5m	0.216	0.507	0.222	1rhq	0.273	0.274	0.274
3vdd	0.230	0.486	0.316	2gh8	0.249	0.512	0.231	2w4y	0.278	0.436	0.284	4iv3	0.235	0.503	0.235
3bcc	0.289	0.509	0.373	1qjx	0.231	0.495	0.249	2qqp	0.285	0.520	0.279	3vbh	0.217	0.505	0.217
3s4g	0.380	0.516	0.300	1f8v	0.219	0.552	0.237	4jgy	0.227	0.497	0.222	3nou	0.390	0.515	0.390
3lob	0.347	0.643	0.423	2iz8	0.192	0.539	0.175	4ftb	0.240	0.572	0.245	3not	0.387	0.515	0.387
3qpr	0.461	0.487	0.387	2bs1	0.245	0.547	0.228	4ang	0.308	0.534	0.303	3nop	0.384	0.514	0.384
1tdi	0.228	0.513	0.296	4aqq	0.291	0.533	0.275	3zfg	0.249	0.530	0.244	2xgk	0.303	0.562	0.303
1ohg	0.373	0.545	0.307	1qju	0.206	0.509	0.221	3zff	0.243	0.511	0.238	2wbh	0.275	0.508	0.275
3e8k	0.365	0.424	0.311	1x36	0.245	0.581	0.230	3cji	0.258	0.521	0.253	2qij	0.331	0.524	0.331
2qzv	0.615	0.614	0.563	1w39	0.245	0.506	0.231	2c51	0.185	0.545	0.180	2c50	0.219	0.515	0.219
2e0z	0.268	0.553	0.216	1x35	0.268	0.432	0.281	1vak	0.211	0.552	0.206	2buk	0.273	0.548	0.273
1wcd	0.219	0.480	0.268	1pgw	0.212	0.493	0.199	1uf2	0.303	0.584	0.308	1wce	0.371	0.552	0.371
4bcu	0.155	0.519	0.200	2wff	0.462	0.480	0.450	1ohf	0.219	0.483	0.214	1tnv	0.384	0.584	0.384
1vcr	0.379	0.429	0.424	2vq0	0.257	0.536	0.245	3ux1	0.283	0.516	0.278	3m8l	--	--	--
1ng0	0.281	0.581	0.324	2fz2	0.276	0.528	0.264	4g0r	0.216	0.533	0.212	2btv	--	--	--
3dar	0.212	0.209	0.253	2fz1	0.309	0.543	0.321	3s6p	0.237	0.529	0.241	2zah	--	--	--
4f5x	0.293	0.536	0.329	1x9p	0.307	0.513	0.295	3ra4	0.212	0.523	0.208	3dpr	--	--	--
4g93	0.379	0.542	0.345	3vbu	0.272	0.491	0.284	2bu1	0.219	0.554	0.215	2w0c	--	--	--
1bcc	0.270	0.457	0.302	3hag	0.277	0.540	0.289	1laj	0.218	0.483	0.214	2bny	--	--	--
2izw	0.294	0.439	0.325	4gh4	0.167	0.537	0.178	1a34	0.179	0.534	0.175	1x33	--	--	--
1f2n	0.218	0.529	0.249	3chx	0.342	0.324	0.332	7msf	0.200	0.527	0.197	3nap	--	--	--
1ny7	0.202	0.486	0.172	1pgl	0.196	0.513	0.186	4iv1	0.190	0.516	0.187				
3oah	0.275	0.525	0.246	1a37	0.320	0.461	0.330	3ra9	0.243	0.499	0.240				
1vb4	0.255	0.536	0.227	1lp3	0.338	0.473	0.329	3ra8	0.245	0.514	0.248				

Figure 1 shows the effect of the reconstruction function of the ASU content from MTRIX records on R-work. As expected, for Protein Databank files containing the complete asymmetric unit, *phenix.pdb.mtrix_reconstruction* does not change the input model or the R-work (points on the diagonal). For most of the 136 entries where the PDB files contain only one NCS copy (Out of the 157, 8 had processing errors), R factors drastically drop, as expected. Table 2 provides a summary of R values for all 157 files with a single NCS copy. It includes reported value, calculated from PDB file as is, and calculated after applying the MTRIX transformation.

R-work values reported in the PDB file and for the model calculated using *mmtbx.f_model.manager* are not exactly the same. In figure 2 we can see that the values for the expanded files are at least as good as those calculated for files with complete ASU. We can also see that there can be significant difference between the reported and the calculated values.

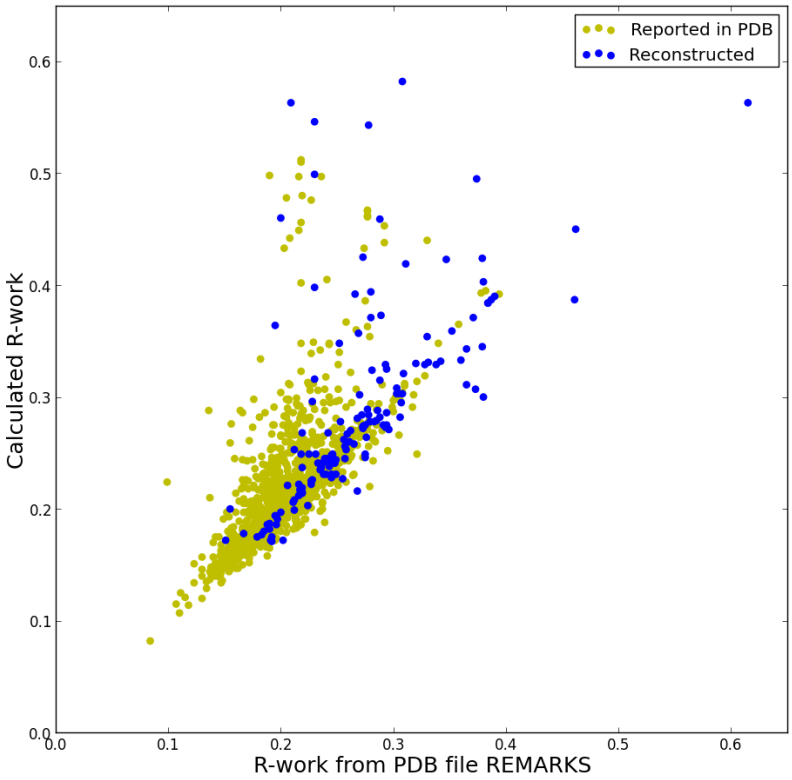


Figure 2: R-work reported in PDB file vs R-work calculated. In yellow are points were the complete ASU were in the PDB file. In blue are the values for the expanded files.

References

Adams PD, Afonine PV, Bunkoczi G, Chen VB, Davis IW, et al. 2010. PHENIX: a comprehensive Python-based system for macromolecular structure solution. *Acta Crystallogr D* 66: 213-21

Afonine PV, Grosse-Kunstleve RW, Echols N, Headd JJ, Moriarty NW, et al. 2012. Towards automated crystallographic structure refinement with phenix.refine. *Acta crystallographica. Section D, Biological crystallography* 68: 352-67

Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, et al. 2000. The Protein Data Bank. *Nucleic Acids Res* 28: 235-42

Bernstein FC, Koetzle TF, Williams GJ, Meyer EF, Jr., Brice MD, et al. 1977. The Protein Data Bank: a computer-based archival file for macromolecular structures. *J Mol Biol* 112: 535-42

Kaufmann B, Bowman, V.D., Li, Y., Szelei, J., Waddell, P.J., Tijssen, P., Rossmann, M.G. 2010. PDB ID: 3n7x. pp. Crystal structure of Penaeus stylirostris densovirus capsid. http://www.rcsb.org/pdb/explore/explore.do?structureId=3n7x: J.Virol.

Kaufmann B, El-Far, M., Plevka, P., Bowman, V.D., Li, Y., Tijssen, P., Rossmann, M.G. 2011. PDB ID: 3p0s. pp. rystal structure of Bombyx mori densovirus 1 capsid. http://www.rcsb.org/pdb/explore/explore.do?structureId=3p0s: J.Virol.

Kleywegt GJ, Jones TA. 1997. Model building and refinement practice. *Method Enzymol* 277: 208-30

Sagurthi SR, Rajaram, V., Savithri, H.S., Murthy, M.R.N. PDB ID: 2wws. pp. PHYSALIS MOTTLE VIRUS: NATURAL EMPTY CAPSID. http://www.rcsb.org/pdb/explore/explore.do?structureId=2wws: To be Published

Sagurthi SR, Rajaram, V., Savithri, H.S., Murthy, M.R.N. PDB ID: 2xpj pp. Crystal structure of Physalis Mottle Virus with intact ordered RNA. http://www.rcsb.org/pdb/explore/explore.do?structureId=2xpj: To be Published

Coping with BIG DATA image formats: integration of CBF, NeXus and HDF5

Herbert J. Bernstein[a], Jonathan M. Sloan[b], Graeme Winter[b], Tobias S. Richter[b], NeXus International Advisory Committee[c], Committee on the Maintenance of the CIF Standard[d]

[a]*Department of Mathematics and Computer Science, Dowling College, Oakdale, NY 11769*
[b]*Diamond Light Source, Harwell Science and Innovation Campus, OX11 0DE (UK)*
[c]*http://wiki.NeXusformat.org/NIAC* *
[d]*http://www.iucr.org/resources/cif/comcifs* †

Correspondence email: yayahjb@gmail.com

Introduction

This is an update to a July 2013 report of the same title (Bernstein, Sloan *et al.* 2013).

The BIG DATA demands of the new generation of X-ray pixel array detectors necessitate the use of new storage technologies as we meet the limitations of existing file systems. In addition, the modular nature of these detectors provides the possibility of more complex detector arrays, which in turn requires a complex description of the detector geometry (for example, see the companion article "XFEL Detectors and ImageCIF", this issue). Taken together these give an opportunity to combine the best of CBF/imgCIF (the Crystallographic Binary File), NeXus (a common data framework for neutron, X-ray and muon science) and HDF5 (Hierarchical Data Format, version 5, the high-performance data format used by NeXus) for the management of such data at synchrotrons. Discussions are in progress between COMCIFS (the IUCr Committee for the Maintenance of the CIF Standard) and NIAC (the NeXus International Advisory Committee) on an integrated ontology. A proof-of-concept API based on CBFlib and the HDF5 API is being developed in a collaboration among Dowling College, Brookhaven National Laboratory and Diamond Light Source. A preliminary mapping and a combined API are under development. Releases of CBFlib since CBFlib 0.9.2.12 can store arbitrary CBF files in HDF5 and recover them, support use of all CBFlib compressions in HDF5 files and can convert sets of miniCBF files to a single NeXus file.

Data Rates, Formats and High Performance X-ray Detectors

CCD X-ray detectors provide images at a moderate data rate of one every few to several seconds (see http://www.adsc-xray.com/Q4techspecs.html). Current higher performance X-ray detectors, such as the DECTRIS Pilatus, are capable of collecting six-megapixel images at 10 - 25 frames per second (Trueb *et al.* 2012), while the newest Pilatus3 6M instruments can operate at 100 frames per second (https://www.dectris.com/pilatus3_specifications.html). The coming next generation of high performance X-ray detectors for MX such as the DECTRIS Eiger will be capable of collecting 16+ megapixel images at more than 125 frames per second (Willmott 2011) (Johnson *et al.* 2012). The ADSC DMPAD is also expected to produce 900 fine-sliced images in steps of two-tenths of a degree at 125 frames per second (Hamlin, Hontz and Nielsen 2012). The Cornell-SLAC pixel array detector (CSPAD) for XFELs produces 120 2.3 megapixel frames per second using 2 bytes per pixel (Hart, Boutet *et al.* 2012). Note that gain-corrected CSPAD images use 8 bytes per pixel. Table 1 shows typical sustained data rates for detectors used for MX at NSLS, DLS, etc. compared to uncompressed XFEL rates (likely to decrease

* The members of NIAC are: Mark Könnecke, Paul Scherrer Institut, Switzerland (Chair), Frederick Akeroyd, Rutherford Appleton Laboratory, UK (ISIS Representative, Technical Committee Chair), Herbert J. Bernstein, ImgCIF, Bjorn Clausen, Los Alamos National Laboratory, USA, Stephen Cottrell, Rutherford Appleton Laboratory, UK (Muon Representative), Jens-Uwe Hoffmann, Helmholtz Zentrum Berlin, Germany, Pete Jemian, Advanced Photon Source, USA (Documentation Release Manager), David Männicke, Australian Nuclear Science and Technology Organisation, Australia, Raymond Osborn, Argonne National Laboratory, USA, Peter Peterson, Spallation Neutron Source, USA, Tobias Richter, Diamond Light Source, UK (Executive Secretary), Armando Sole, European Synchrotron Radiation Facility, France, Jiro Suzuki, KEK, Japan, Benjamin Watts, Swiss Light Source, Switzerland, Eugen Wintersberger, DESY, Germany, Joachim Wuttke, FRM II and JCNS, Germany.

† The members of COMCIFS are: James R. Hester (Chair), Herbert J. Bernstein, John C. Bollinger, Brian McMahon (Coordinating Secretary), John Westbrook.

Table 1. Typical Sustained Data Rates

Detector	Raw Image Size (MB)	Frame Rate (Hz)	Compressed Rate (Gb/sec)	USB Disk Data Rate (%)
ADSC Q315 (2x2 binned)	18	0.37	.013	7
Pilatus 2 6M	24	10	.48	240
Pilatus 2 Fast 6M	24	25	1.2	600
CSPAD	4.6	120	4.4	2208
Pilatus 3 6M	24	100	4.8	2400
Eiger 16M	72	125	18	9000

with suitable compression) and expected rates from Eiger, expressed in terms of the typical data rate for an inexpensive USB disk of 25 MB/sec = 200 Mb/sec. A data management system designed for very large numbers of files as well as for very large data volumes and data rates is needed (Fig. 1). Efficient recording of metadata coordinated with the data is also needed and database access to information about images and experimental runs is needed. For MX, these data rates, data volumes, numbers of distinct images and numbers of distinct experiments argue for a very organized, high performance infrastructure. HDF5 and NeXus provide the necessary organization of the raw data and CBF provides the necessary organization of the associated metadata for subsequent processing as well as contributing useful compression algorithms.

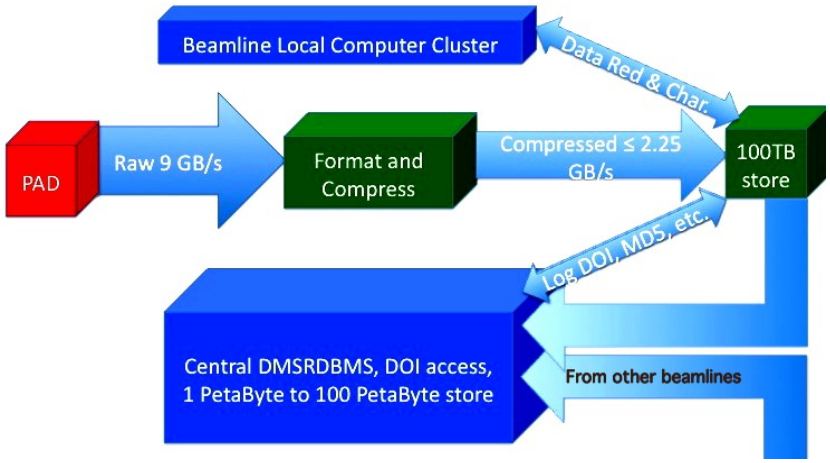


Figure 1. Major data flows from the beamline advanced pixel array detectors (PADs) to the data management system relational database (DMSRDBMS). In order to be manageable, the raw 9 gigabyte per second data flow from each PAD needs to be compressed locally by at least 4:1, before going into a beamline 100 terabyte store for beamline local computer cluster access for up to a week for data reduction and characterization. The required bandwidth of the pipes from the beamlines to the DMSRDBMS depends on the compression used. If no further compression is used, 2.25 gigabyte (18 gigabit) per second per beamline network connections are required. If a combined lossless/lossy compression is used, then 225 - 450 megabyte (1.8 to 3.6 gigabit) per second per beamline network connections will suffice. The flows for transfers to user home institutions are not shown.

Today for MX alone Diamond Light Source employs one Pilatus 2M, three Pilatus 6M fast and one Pilatus 3 6M, giving a combined data rate of over 1 GB/sec and over 200 files/sec, creating the need to manage hundreds of thousands of images each day. For the Advanced Beamlines for Biological Investigations with X-rays (ABBIX) that are being built for NSLS-II (Hendrickson 2012), just two of the beam lines, the Frontier Macromolecular Crystallography (FMX) beamline and the Automated Macromolecular Crystallography (AMX) beamline (Schneider *et al*. 2012), are expected to produce an aggregate of more than 94 terabytes per operational half day, 660 terabytes per week or 38 petabytes per year. The anticipated beamline flux is 10^{13} photons per second for FMX and $2x10^{13}$ photons per second for AMX, approximately 50 times the NSLS X25 and X29 fluxes. One subtle effect of these high fluxes is that there will be more photons per pixel in images, making them more difficult to compress.

A final issue, in addition to the actual recording of data, is that of automated processing. At Diamond Light Source and elsewhere there has been a push towards the automated analysis of diffraction data, as interactively processing diffraction data at the current rate of typically 20 data sets per hour

per beamline is unsustainable. This however places an increased strain on the file system, as typically the same data are read as many as six times in order to be processed, resulting in over 1000 file access operations per second. With the storage of many frames per file, as planned for NeXus, the rate of file operations would decrease substantially.

HDF5 and NeXus

The Hierarchical Data Format Version 5 (HDF5) is a self-describing file format with a robust, well-documented API routinely handling multi-gigabyte files of data. It has a diverse user community covering a wide range of disciplines and is fully supported (Dougherty *et al.* 2009). HDF5 is particularly well suited to the management of very large volumes of complex scientific data and has been adopted as the primary data format in a wide range of disciplines (http://www.hdfgroup.org/HDF5/users5.html) and provides the "inner workings" of important frameworks, such as NetCDF (Rew *et al.* 2004) and NeXus. To avoid confusion we use the term *format* to describe the logical organization of data on a storage medium. An *ontology* is a dictionary of terms that may include descriptions of the relationships between terms. An ontology can be realized in one or more formats. We are therefore dealing with the HDF5 *format*, the NeXus *ontology*, a CBF *format* and an imgCIF *ontology*. The HDF5 *format*, XML *format* and NeXus *ontology* together form the NeXus data transfer *framework*. The CBF *format*, CIF *format* and imgCIF *ontology* form the imgCIF data transfer *framework*.

HDF5 is tree-oriented, which is a very powerful and useful characteristic allowing file-system-like nesting of groups of data within groups of data, in order for information to be easily, reliably and efficiently searched. However, tables are more useful for loading information into a relational database management system (Codd 1970).

NeXus (Filges 2001) (Könnecke 2006) is a tree-oriented ontology for use wth HDF5 (and XML and HDF4) of importance in managing neutron and X-ray data. NeXus adds rules for storing data in files and a dictionary of documented names to HDF-5 in order to make HDF-5 applicable to the problem domain of synchrotron, neutron and muon scattering. NeXus is a convenient thin layer over HDF5 that is widely used at many physics research centers, including at synchrotrons. Together NeXus and HDF5 provide a portable, extensible and efficient framework for the storage and management of data.

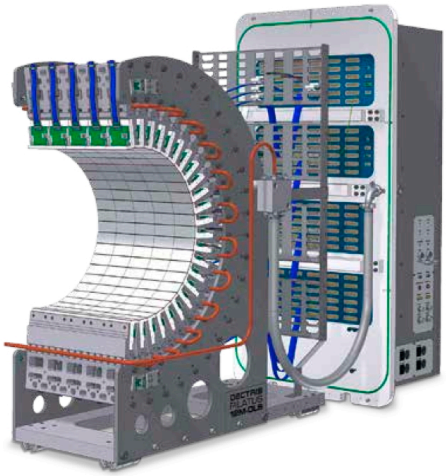


Figure 2. Curved DECTRIS detector for DLS beamline I23, an example of a detector with a complex geometry best described using the imgCIF/CBF ontology. Prior to this work, NeXus could not support a detector such as this. Now it will be possible.

Jan 2013 DECTRIS Eiger Workshop and Followup

The attendees at the January 2013 DECTRIS Workshop agreed on the use of an HDF5-based NeXus framework for the DECTRIS Eiger pixel array detector. The workshop charged Herbert J. Bernstein with following up on mapping additional terms to the new format. Tobias Richter, Jonathan Sloan and Herbert J. Bernstein have worked on a CBF-NeXus concordance and supporting software based on CBFlib and HDF5 with the cooperation of Bob Sweet, Graeme Winter and Mark Koennecke. Discussions with NIAC were held and then discussions with COMCIFS were held prior to ECM 28 in August 2013. There was general agreement that it was a good idea to have CIF and NeXus interoperate. COMCIFS and NIAC have agreed to start on a single crystal monochromatic macromolecular crystallography experiment NeXus application definition. An application definition in NeXus is a specification of the required metadata and data for that application. Significant progress has been made on the application definition and a draft will be available in Spring 2014.

Mapping from NeXus to CBF

All NeXus base classes now have proposed slots in CIF categories. Handling of the DECTRIS-proposed Eiger HDF5 format is in the concordance. This concordance will require some relaxation of current NeXus name practices. "CBF_" prefixes are being used as an interim solution. Most of these prefixes are expected to be removed in the final version.

For this project, organizing data and metadata according to the conventions of the IUCr Crystallographic Information File (Hall *et al.* 1991) using imgCIF (Bernstein, Hammersley, 2005) and its open source supporting software CBFlib (Ellis, Bernstein 2005) provides a database-friendly tabular structure. The imgCIF ontology provides the metadata needed for the analysis of diffraction images and is supported by all the major detector manufacturers. This aspect is particularly important for instruments with complex geometries, e.g., the Pilatus 12M being constructed by DECTRIS for the long wavelength beamline I23 at Diamond Light Source (Fig. 2).

The embedding of CIF tables in HDF5 files was demonstrated at the "HDF5 as hyperspectral data analysis format" workshop in January 2010 (Götz *et al.* 2010). The workshop recommendation was, in part, "Adopt as much as possible from imgCIF and sasCIF".

Tables are easily embedded into trees. Going in the other direction is more difficult. There is serious effort required to make general trees into tables suitable for use in a relational database management system, involving a process known as "normalization"(Codd 1972). See Fig. 3.

One of the tasks of this project is to extend the imgCIF ontology to ensure workable database access to metadata in the HDF5 tree that has not already been normalized into CIF categories. For example, Digital Object Identifiers (DOIs) and SHA2 or SHA3 checksums from multiple experiments will need to be brought forward into a common table for post-experiment forensic validation.

CBF and Database Access

The Crystallographic Binary File (CBF) format is a complementary format to the Crystallographic Information File (CIF), supporting efficient storage of large quantities of experimental data in

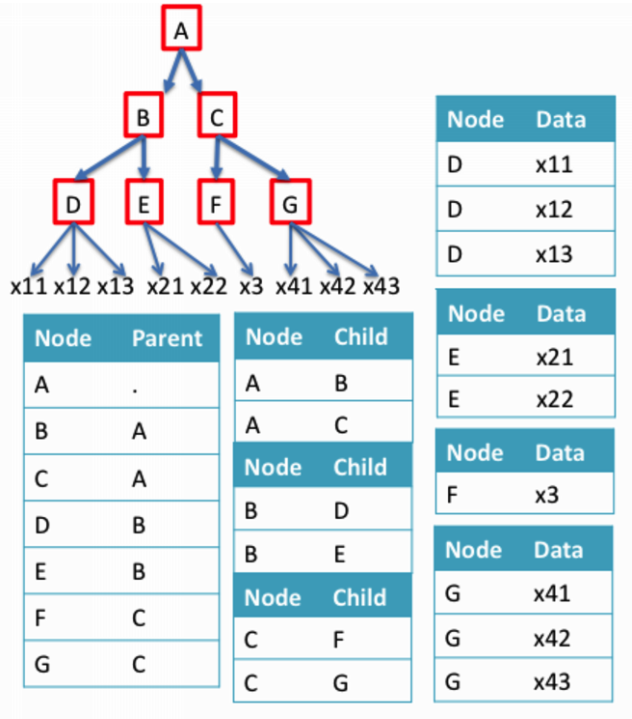


Figure 3. Example of a tree mapped into tables for database access identifying the links between parents and children in both directions as well as data. This is about the simplest example we can provide of a tree that demonstrates the need for "normalization" in the conversion from trees to tables.

a self-describing binary format with a sophisticated description of the experimental geometry. For large PAD images, the raw binary CBF format is heavily used both within laboratories and for interchange among collaborating groups. When dealing with large numbers of independent experiments producing large numbers of CBF/imgCIF image files, HDF5 provides the virtual file system needed to manage the massive data flow. While it is feasible to simply encapsulate the CBF/imgCIF image files as opaque objects within an HDF5-based data-management system, active management of the data can be done more efficiently when the imgCIF tags are made visible in the HDF5 tree, a capability demonstrated in 2010.

With the anticipated throughput of NSLS II beamlines and the current capabilities of MX beamlines equipped with pixel array detectors, the management of data and the possibility of interrogating the data files for experimental information becomes critical.

Figure 3. Tests on fifty image files from DLS with low to moderate pixel density. Total size 1.2 gigabytes. Relative times are shown. Multiply by 3 for the number of cores needed to keep up with the compression workload. The fifty files are from a set of 900 images recorded by Graeme Winter at Diamond Light Source beamline I02 as part of routine development work, and come from a crystal of DNA (TCGGCGCCGA) bound to a large ligand (lambda-[Ru(TAP)2(dppz)]2+). The aggregate of fifty was chosen to produce an uncompressed file small enough (1.2 GB) to be acceptable at user sites. Larger aggregates could be used for sites able to accommodate larger files.

Compression Method	Compression Ratio	Relative Time
external bzip2 compression	20.4:1	5.6
HDF5 CBFlib canonical compression	15.7:1	3.9
HDF5 CBFlib nibble offset compression	11.5:1	2.9
HDF5 CBFlib packed V2 compression	11.0:1	2.8
HDF5 zip compression	9.7:1	2.4
external LZ4 compression (C1 one pass)	8.7:1	2.2
HDF5 CBFlib packed compression	8.6:1	2.2
external LZ4 compression (C0 two passes)	5.2:1	1.3
HDF5 CBFlib byte offset compression	4.0:1	1.0

Compression

There are long-standing issues about compression in crystallography. High-speed, high-compression-ratio compression is a critical issue for the next generation of detectors. Some compressions raise license issues. Some popular ones are slow or inefficient or both. Some can be handled in processing programs such as XDS if license and programming language issues can be addressed. Low pixel density fine-slicing with clean backgrounds makes some compressions more effective.

CBFlib provides useful compressions. See Table 2. A plugin module has been written to allow HDF5 to read and write CBFlib compressions. Starting with CBFlib release 0.9.2.11, that module is included. HDF5 1.8.11 and later is required. For general documentation on HDF5 dynamically loaded filters, see

http://www.hdfgroup.org/HDF5/doc/Advanced/DynamicallyLoadedFilters/HDF5DynamicallyLoadedFilters.pdf

The filter has been registered with the HDF5 group as 32006and cbf.h includes the symbolic name for the filter CBF_H5Z_FILTER_CBF. The source and header of the CBFlib filter plugin are cbf_hdf5_filter.c and cbf_hdf5_filter.h, respectively, in the CBFlib kit. To use the filter in C applications, you will need to include cbf_hdf5_filter.h in the application and have the cbflib.so library in the search path used by HDF5 1.8.11. Each compressed image in the HDF5 file is in the same format as the MIME-headed compressed images in the corresponding CBF, so the Fortran image search logic used in XDS can be used directly on these files.

Where to Find Software and Documentation

Draft imgCIF/CBF version 1.7 dictionary that now includes information on going from CBF to NeXus: https://www.sites.google.com/site/nexuscbf/home/cbf-dictionary

PDF summary of the concordance: https://www.sites.google.com/site/nexuscbf/mapping-draft

CBFlib kit: http://downloads.sf.net/cbflib/CBFlib-0.9.3.3.tar.gz

that includes both Jonathan Sloan's utilities to convert sets of minicbfs into a single NeXus file and a plugin filter that supports the full set of CBFlib compressions in HDF5.

Conclusion

The essential first steps in the integration of CBF, NeXus and HDF5 have been taken. There is work still to be done in applying this work at beam lines and in data processing software. Collaborators are most welcome.

Acknowledgements

Our thanks to James Hester, Chair of COMCIFSand Mark Koennecke, Chair of NIAC, for supporting and encouraging this effort and to all participants in COMCIFS and NIAC for helpful commentsand also to James Hester for contributing the section on ontologies and formats, a critical issue in this

work. Our thanks go to Nick Sauter and Aaron Brewster for extending this work to CSPAD. Our thanks for years of supporting efforts, at the BNL PXRR Group: Robert M. Sweet, Dieter Schneider, Howard Robinson, John Skinner, Matt Cowan, Leonid Flaks, Richard Buono; at DLS: Alun Ashton, Bill Pulford; and at the Dowling College ARCiB Lab Group: Mojgan Asadi, Kostandina Bardhi, Maria Karakasheva, Ming Li, Limone Rosa

Our thanks to DECTRIS, BIOIHDFand the HDF Group

Our thanks to Frances C. Bernstein

This work was funded in part by NIGMS, DOE, NSF, PaNdata ODI (EU 7th Framework Programme)

References

Bernstein HJ and Hammersley AP (2005). “Specification of the Crystallographic Binary File (CBF/imgCIF)". In: S. R. Hall and B. McMahon, Eds., International Tables For Crystallography, Chap. 2.3, pp. 37 - 43, International Union of Crystallography, Springer, Dordrecht, NL.

Bernstein HJ, Sloan JM, Winter G, Richter TS, NeXus International Advisory Committee and Committee on the Maintenance of the CIF Standard (2013). "Coping with BIG DATA Image Formats: Integration of CBF, NeXus and HDF5." poster, American Crystallographic Association, 2013 Annual Meeting. Honolulu, HI.

Codd EF (1970). “A relational model of data for large shared data banks”. Communications of the ACM, Vol. 13, No. 6, pp. 377 - 387.

Codd EF (1972). Courant Computer Science Symposium 6, Chap. “Further Normalization of the Data Base Relational Model”, pp. 33 - 64. Prentice-Hall.

Dougherty MT, Folk MJ, Bernstein HJ, Bernstein FC, Eliceiri KW, Benger W, Zadok E and Best C (2009). “Unifying Biological Image Formats with HDF5”. Communications of the ACM, Vol. 52, No. 10, pp. 42 – 47.

Ellis PJ and Bernstein HJ (2005). Definition and Exchange of Crystallographic Data, International Tables For Crystallography, Chap. “CBFlib: an ANSI C library for manipulating image data”, pp. 544 -- 556. International Union of Crystallography, Springer, Dordrecht, NL.

Filges U (2001). “The new NeXus API based on HDF5". In: VITESS Workshop Berlin, 25 - 27 June 2001.

Götz A, Solé V, Madonna C and Maydew AF (2010). “ELISA VEDAC Workshop Report, Workshop Title: HDF5 as hyperspectral data exchange and analysis format, Grenoble, January 11th - January 13th, 2010”. http://vedac.esrf.eu/public-discussions/hdf5-workshop/workshop-report.

Hall SR, Allen FH and Brown ID (1991). “The Crystallographic Information File (CIF): a new standard archive file for crystallography”. Acta Crystallographica Section A: Foundations of Crystallography, Vol. 47, No. 6, pp. 655 – 685.

Hamlin RC, Hontz T and Nielsen C (2012). “The New Dual Mode Pixel Array Detector" in: Meeting of the American Crystallographic Association, Boston, MA, 28 July - 1 August 2012, American Crystallographic Association, Abstract 11.01.1151.

Hart P, Boutet S, Carini G, Dubrovin M, Duda B, Fritz D, Haller G, Herbst R, Herrmann S, Kenney C, Kurita N, Lemke H, Messerschmidt M, Nordby M, Pines J, Schafer D, Swift M, Weaver M, Williams G, Zhu D, Van Bakel N and Morse J (2012). "The CSPAD megapixel x-ray camera at LCLS." Proceedings of SPIE 8504: 85040C-85040C-85011.

Hendrickson WA (2012). “NSLS-II - Status of the Life Sciences Program”. Tech. Rep., Brookhaven National Laboratory, X6A Science Advisory Committee, February 2012. http://protein.nsls.bnl.gov/mediawiki/images/e/e3/Hendrickson_2012.pdf.

Johnson I, Bergamaschi A, Buitenhuis J, Dinapoli R, Greiffenberg D, Henrich B, Ikonen T, Meier G, Menzel A, Mozzanica A, Radicci V, Satapathy DK, Schmitt B and Shi X (2012). “Capturing dynamics with Eiger, a fast-framing X-ray detector”. Journal of Synchrotron Radiation, Vol. 19, No. 6, pp. 19, 1001 -1005.

Könnecke, M (2006). “The State of the NeXus data Format”, Physica B: Condensed Matter Vol. 385–386, Part 2, 15 November 2006, 1343–1345, Proceedings of the Eighth International Conference on Neutron Scattering.

Rew R, Ucar B and Hartnett E (2004). “Merging netCDF and HDF5”. In: 20th Int. Conf. on Interactive Information and Processing Systems.

Schneider DK, Sweet RM and Skinner J (2012). “Projection of MX ABBIX needs at AMX and FMX for Data Acquisition, Data Processing, Software, Data Archiving and Networking". February 2012. Private Communication.

Trueb P, Sobott BA, Schnyder R, Loeliger T, Schneebeli M, Kobas M, Rassool RP, Peake DL and Broennimann C (2012). “Improved count rate corrections for highest data quality with PILATUS detectors”. Journal of Synchrotron Radiation, Vol. 19, No. 3, pp. 347 - 351.

Willmott P (2011). An Introduction to Synchrotron Radiation: Techniques and Applications. John Wiley and Sons, Chichester, UK. page 6.

XFEL Detectors and ImageCIF

Aaron S. Brewster[a], Johan Hattne[a], James M. Parkhurst[b], David G. Waterman[c], Herbert J. Bernstein[d], Graeme Winter[b], and Nicholas K. Sauter[a]

[a]*Lawrence Berkeley National Laboratory, Berkeley, CA 94720*
[b]*Diamond Light Source, Harwell Science and Innovation Campus, OX11 0DE (UK)*
[c]*CCP4, Research Complex at Harwell, STFC Rutherford Appleton Laboratory, OX11 0FA (UK)*
[d]*Department of Mathematics and Computer Science, Dowling College, Oakdale, NY 11769*

Correspondence email: asbrewster@lbl.gov

Introduction

Serial femtosecond crystallography performed using X-ray free electrons lasers (XFELs) creates a challenging task for modern detectors (Chapman *et al.* 2011, Boutet *et al.* 2012, Kern *et al.* 2013). Pulses containing 10^{12} photons, 40-50 femtoseconds (fs) long are generated using a linear accelerator and impact a liquid jet of protein microcrystals at a rate of 120 pulses per second. Still image diffraction patterns from thousands of crystals can be collected in a matter of minutes. The Cornell-SLAC pixel array detector (CSPAD) is a unique detector designed to operate at these rates and record data from exposures on the fs time scale (Hart *et al.* 2012). The CSPAD is modularized into 32 sensors arranged in a roughly square pattern. This creates unique challenges for representing the data in such a way that the geometric layout of the experiment is accurately recorded. To this end, we have adopted and extended the ImageCIF/CBF file specification (Bernstein & Hammersley 2005) to record CSPAD diffraction data, adding sufficient parameters to the ImageCIF dictionary to lay out the XFEL experiment at SLAC, including fully specifying the detector geometry. The new ImageCIF parameters described below help us handle the detector geometry by expressing it easily refineable terms. This extensibility and the ability to parameterize the entire experiment explicitly made ImageCIF/CBF the best option for representing this data. The CSPAD CBF format is natively understood by *cbflib* (Ellis & Bernstein 2001), a software package developed specifically for reading and writing CBF files. We have also incorporated the format into *cctbx* (Grosse-Kunstleve *et al.* 2002, Sauter *et al.* 2013), using a new multi-tile detector model defined by the module *dxtbx* (the diffraction experiment toolbox)(Waterman *et al.* 2013, Parkhurst *et al.* in preparation). These software packages allow us to refine the experimental geometry against measured data, leading to better indexing rates and more accurate integration of the reflection signal (Hattne *et al.* submitted).

CSPAD Detector Geometry

Three full-size CSPAD detectors are in service at the present, two at the Linac Coherent Light Source (LCLS) Coherent X-ray Imaging (CXI) instrument, and one at the LCLS X-ray Pump Probe (XPP) instrument. Each detector is comprised of 32 sensors and each sensor houses two application-specific integrated circuits (ASICs), 194x185 pixels in dimension, with a pixel size of 110 microns and a three-pixel gap between them (Figure 1). 8 sensors comprising 16 ASICs form a quadrant. The four quadrants surround a central hole, through which the undiffracted beam passes. The CXI detector quadrants are adjustable on diagonal rails, allowing the central size to grow and shrink. This allows the second detector, typically positioned 2.5 meters behind the first, to receive signal.

The 32 sensors are not orthogonalized, meaning edges between sensors are not parallel; each sensor is tilted slightly off of 90°. Further, the sensors are not co-planar with the detector, having small angles off of the planar normal. LCLS provides optical measurements to position the sensors in three-dimensional space, and these measurements have been enormously useful in specifying the detector geometry. At the CXI beamline, quadrant positions are not provided, as they are variable. Their location needs to be experimentally determined, initially by aligning the quadrants to rings from powder diffraction, and subsequently refined against single crystal diffraction. For both CXI and XPP, detector tilt and position need to be refined as well. For example, the beam itself is not always perfectly parallel to the detector rail, leading to small changes in beam center at different detector distances. All of this geometric information needs to be recorded for each still in a way understandable by developers working on indexing and integration while still

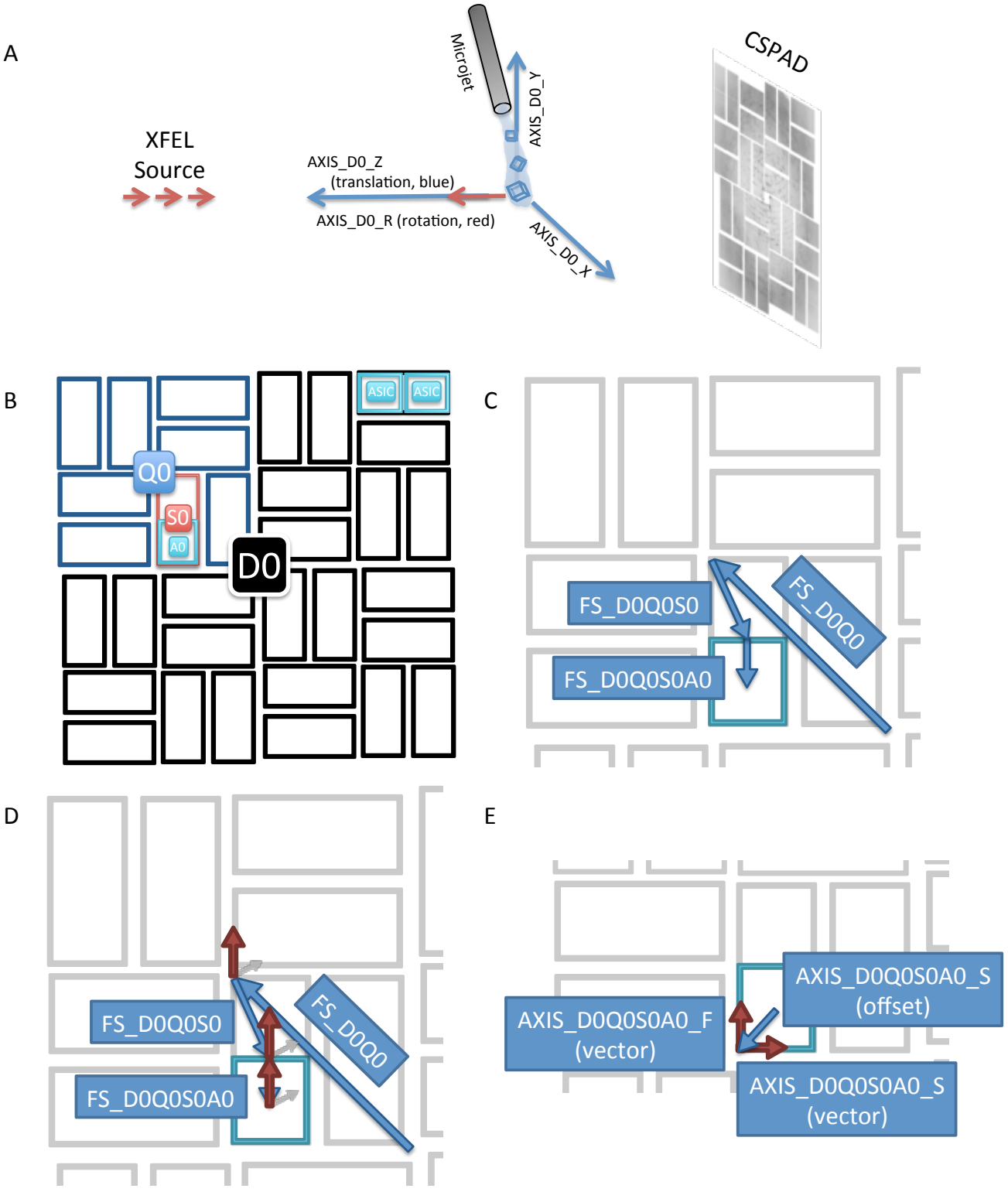


Figure 2: ImageCIF axes describing the CSPAD detector. A) XFEL experiment overview. Crystals are injected into the XFEL stream via a micro-injection system. The root ImageCIF axes for the CSPAD detector as a whole are shown. Axes AXIS_D0_X, _Y and _Z are translation axes along which the detector can be moved. Detector distance is specified as a translation along AXIS_D0_Z. A fourth axis, AXIS_D0_R, defines a rotation axis around which the detector can be rotated. B) Overview of the detector. Rectangles are 32

Caption continues on the following page.

sensors, each comprising 2 ASIC pixel array chips, as shown in the upper right hand corner. D0, Q0, S0 and A0 are highlighted, corresponding to detector zero (*i.e.*, the detector as a whole), quadrant zero in the upper left hand corner, sensor zero and ASIC zero. C) Frame shift vector offsets for the three rotation axes that position the quadrant, sensor, and ASIC centers. CBF rotation axes have three components: an offset from the base of the parent axis to its base, a vector describing the direction around which to rotate, and a rotation angle. Three axis offsets are shown (blue) that together describe the position of an ASIC relative to the center of the detector. D) Vector components (red) of the three rotation axes shown in B. Arrows are normal to the respective surface planes. Around these axes the various elements are rotated into position by specifying appropriate angles. E) Vectors (red) and offsets (blue) for the 2 fast and slow axes of an ASIC element. Here, the red vectors are translation axes that are co-planar with the ASIC chip. They relate how the pixel array is laid out in space to its in-memory arrangement. The blue arrow is the slow axis' offset from the center of the ASIC. The fast axis depends on the slow axis so its offset is zero.

being easily parameterized for refinement.

ImageCIF/CBF

ImageCIF is a specific CIF dictionary for representing diffraction data. Binary encoding of the pixel array data together with ImageCIF metadata comprises the Crystallographic Binary File format (CBF). In use by a variety of companies to record diffraction frame data, ImageCIF and CBF are internationally agreed upon standards maintained by the International Union of Crystallography. ImageCIF allows complete description of the geometry of the crystallographic experiment. For these reasons, we found it applicable to our needs.

In ImageCIF, one describes frame data in the form of a blueprint for the detector. First, the individual pixel-array elements are defined. In the case of the CSPAD, 64 elements are specified:

```
loop_
_diffrn_detector_element.id
_diffrn_detector_element.detector_id
 ELE_D0Q0S0A0 CSPAD_FRONT
 ELE_D0Q0S0A1 CSPAD_FRONT
 ELE_D0Q0S1A0 CSPAD_FRONT
 ELE_D0Q0S1A1 CSPAD_FRONT
...
 ELE_D0Q3S7A0 CSPAD_FRONT
 ELE_D0Q3S7A1 CSPAD_FRONT
```

Here, a new CIF table is defined with the 'loop_' keyword, named diffrn_detector_element. The table links elements by their IDs to a detector ID (CSPAD_FRONT). Multiple detectors can be defined in the same file; if the second detector at CXI, known as the back detector and positioned up to 2.5 meters behind the front detector, is in use, its data and metrology could be recorded in the same file. The convention we use for naming the CSPAD elements includes IDs for the detector (D), quadrant (Q), sensor (S) and ASIC (A). Thus ELE_D0Q0S1A0 is the array of pixels that represents detector 0, quadrant 0, sensor 1, ASIC 0. Later in the file, each of these elements has a separate binary encoding of their pixel data. Other tables specify gain, array dimensions and further physical properties of each element.

Once elements are defined, the geometry of the detector is laid out using two tables: axis and diffrn_scan_frame_axis. The axis table specifies lines of motion and axes of rotation for the experiment, while the diffrn_scan_frame_axis table specifies physical settings for detector components along the axes specified in the axis table. The ImageCIF axis convention specifies the origin to be the sample position, with the X-axis pointing along the axis of right-handed goniometer rotation, the Z-axis pointing to the beam source, and the Y-axis completing a right handed system (Figure 1A). In the case of many XFEL experiments, no goniometer is present, so the X-axis is simply orthogonal to the beam and gravity. Thus the first few lines of the CSPAD axis table are shown in scheme 1.

Each line defines an axis by its type: general, translation or rotation. Equipment refers to kinds of devices that move along the given axis. Other examples include goniometer axes, which XFEL experiments generally do not include. The vector specifies either the direction of translation or the axis about which rotation is performed, and the offset positions the base of the axis in space relative to the parent axis specified in the depends_on field. Finally, equipment_component is a new field we have added to the ImageCIF dictionary in collaboration with its principal maintainer, Herbert Bernstein. This field allows us to group axes together, which will be important to distinguish hierarchy level later when we construct a detector model using *dxtbx* software.

Axis positions are specified in the diffrn_scan_frame_axis table:

```
loop_
_axis.id
_axis.type
_axis.equipment
_axis.depends_on
_axis.vector[1]
_axis.vector[2]
_axis.vector[3]
_axis.offset[1]
_axis.offset[2]
_axis.offset[3]
_axis.equipment_component
 AXIS_SOURCE     general     source    .              0  0 1   .   . . .
 AXIS_GRAVITY    general     gravity   .              0 -1 0   .   . . .
 AXIS_D0_Z       translation detector  .              0  0 1   .   . . detector_arm
 AXIS_D0_Y       translation detector AXIS_D0_Z       0  1 0   .   . . detector_arm
 AXIS_D0_X       translation detector AXIS_D0_Y       1  0 0   .   . . detector_arm
 AXIS_D0_R       rotation    detector AXIS_D0_X       0  0 1   0   0 0 detector_arm
```

Scheme 1: ImageCIF 'loop' table. The first 12 lines comprise a header in which the table and each of its 11 columns are named. The first 6 axes are also shown, describing the detector as a whole and its orientation in the laboratory.

```
 FS_D0Q0         rotation    detector AXIS_D0_R       0  0 1 -50  42 0 detector_quadrant
 FS_D0Q0S0       rotation    detector FS_D0Q0         0  0 1  11 -23 0 detector_sensor
 FS_D0Q0S0A0     rotation    detector FS_D0Q0S0       0  0 1 -11   0 0 detector_asic
```

Scheme 2: These three axis entries in the loop table correspond to frameshifts describing the transition from the detector as a whole to quadrant 0, from quadrant 0 to sensor 0, and from sensor 0 to ASIC 0.

```
 AXIS_D0Q0S0A0_S translation detector FS_D0Q0S0A0     0 -1 0 -11  10 0 detector_asic
 AXIS_D0Q0S0A0_F translation detector AXIS_D0Q0S0A0_S 1  0 0   0   0 0 detector_asic
```

Scheme 3: The fast and slow axes of an ASIC. The slow axis is offset (-10, 11, 0) mm from the ASIC center and points in the -Y direction (in relation to its parent). The fast axis depends on the slow axis and points in the X direction (in relation to its parent).

```
loop_
_diffrn_scan_frame_axis.axis_id
_diffrn_scan_frame_axis.frame_id
_diffrn_scan_frame_axis.angle
_diffrn_scan_frame_axis.displacement
 AXIS_SOURCE  FRAME1 0    0
 AXIS_GRAVITY FRAME1 0    0
 AXIS_D0_X    FRAME1 0    0
 AXIS_D0_Y    FRAME1 0    0
 AXIS_D0_Z    FRAME1 0 -171
 AXIS_D0_R    FRAME1 0    0
```

While both angle and displacement can be specified, only the one or the other is meaningful, depending on if the axis is a rotation or translation axis. The detector distance is specified by translating 171 mm along AXIS_D0_Z (in the negative Z direction since Z points to the source).

Once the detector position is specified, subcomponents are laid out in the axis table (Figures 1B, 1C and 1D) as shown in scheme 2. Here, the frame shifts needed to position quadrant 0, detector 0 and asic 0 are specified. Because these are not mechanical axes, we adopt a convention of naming them FS_ for frame shift instead of AXIS_. These are rotation axes to allow sensor rotations to be specified in the diffrn_scan_frame_axis table:

```
 FS_D0Q0     FRAME1    0 0
 FS_D0Q0S0   FRAME1 89.7 0
 FS_D0Q0S0A0 FRAME1    0 0
```

We can see that sensor 0 is rotated 89.7 degrees around its rotation axis specified in the axis table, the (0, 0, 1) axis *i.e.* the Z-axis. In reality, the sensor is tilted slightly from normal. Another CBF file we have generated records the sensor 0 axis vector to be (-0.000974376302058 0.00044773585801 0.999999425062), indicating a very slight tilt from the normal (about 0.6°). ImageCIF allows us to record even this small error, improving the accuracy of the detector description.

Finally, the fast and slow axes are specified for each asic tile in the axis table (Figure 1E, scheme 3). Note that the slow axis is offset from the center

```
# read the file
image = reader(filename)

# iterate through the quadrants of the detector
detector = reader.get_detector()

quadrants = detector.hierarchy()
for quadrant in quadrants:
       # vector pointing to the center of the quadrant relative to the
       # center of the detector
       origin = quadrant.get_origin()

       # unit vectors pointing in the fast and slow directions of the
       # quadrant plane
       fast = quadrant.get_fast_axis()
       slow = quadrant.get_slow_axis()

       # these three vectors form a 3D basis for this quadrant
       <optimize 9 parameters against a set of measured data>

       quadrant.set_frame(refined_fast,
                          refined_slow,
                          refined_origin)

# apply the detector object changes to the image's internal cbf handle
image.sync_detector_to_cbf

#write the new file
image._cbf_handle.write_file(new_filename)
```

Scheme 4: Pseudo-Python code describing a possible optimization of the four quadrant positions.

of the ASIC, positioning it at the (0, 0) pixel. The fast and slow axes are unit vectors that specify the readout directions for the data stored in the CBF binary sections. These entries, together with the above information, completely describe the detector geometry.

dxtbx and CSPAD ImageCIF

Recently, we have collaborated with researchers at the Diamond Light Source in the UK to develop a new *cctbx* component, the diffraction experiment toolbox *dxtbx*. This toolbox provides Python and C++ based interfaces for generically reading crystallographic data regardless of file format. Importantly, the toolbox exposes models of the diffraction experiment through a set of four interfaces, the detector, the scan, the goniometer and the beam. The developer can sub-class from more general file reader classes and expose the detector geometry through these interfaces. For the purpose of XFEL data (still data), only the detector and beam models are useful.

We have written an appropriate generic reader for multi-tile detector data in CBF format, and ensured its compatibility with this CSPAD CBF format. The reader reads the axis list and creates a hierarchy of components using the equipment_component tag in the axis table to group axes together. Scheme 4 is an example of Python code that uses this reader to read a CSPAD CBF file and show how the hierarchy can be used to refine quadrant positions.

The hierarchical model provides powerful tools for interacting with detector geometry to accomplish tasks of importance to XFEL data collection in a straightforward manner.

Finally, XFEL sources can produce hundreds of thousands of individual diffraction patterns. Representation of each pattern as a single CBF file in hard disk storage can be detrimental to file system performance, a problem exacerbated when handling large numbers of experimental runs, each with many files. Use of HDF5 reduces the file system burden for large numbers of runs by grouping multiple images into large HDF5 files, reducing the burden for each run. Therefore, optional conversion of CBF/ImageCIF files to HDF5/NeXus in *CBFlib* is under development (Bernstein *et al.* 2013). The hierarchical geometries presented here will be preserved, with

the added benefit that metadata only needs to be recorded once per complete dataset in an single HDF5 master file, as opposed to being repeated in thousands of separate CBF image files, each containing a full header description (see also the *Computational Crystallography Newsletter* companion article in this issue, "Coping with BIG DATA image formats: integration of CBF, NeXus and HDF5").

Conclusion

Integration of XFEL intensity data requires precise knowledge of where individual pixels are in physical space. Spot centroids are used for indexing, followed by crystal unit cell and orientation refinement. Correct refinement will predict spot locations such that integration masks will capture true signal while avoiding background. The ImageCIF/CBF representation we are implementing in *cctbx.xfel* for the CSPAD detector allows for simpler refinement of detector geometry, at the detector, quadrant, sensor and ASIC levels to sub-pixel accuracy. Incorporation into *dxtbx* enables straightforward access to detector and beam models, facilitating this refinement.

Acknowledgements

This work was supported by NIH grants GM095887 and GM102520 and Director, Office of Science, Department of Energy (DOE) under contract DE-AC02-05CH11231 for data-processing methods (N.K.S.).

References

Bernstein HJ and Hammersley AP (2005). Specification of the Crystallographic Binary File (CBF/imgCIF). *International Tables For Crystallography*. H. S. R. and M. B. Dordrecht, NL, Springer. **G:** 37-43.

Bernstein HJ, Sloan JM, Winter G, Richter TS, NeXus International Advisory Committee and Committee on the Maintenance of the CIF Standard (2013). "Coping with BIG DATA Image Formats: Integration of CBF, NeXus and HDF5." *poster, American Crystallographic Association, 2013 Annual Meeting*. Honolulu, HI.

Boutet S, Lomb L, Williams GJ, Barends TR, Aquila A, Doak RB, Weierstall U, DePonte DP, Steinbrener J, Shoeman RL, Messerschmidt M, Barty A, White TA, Kassemeyer S, Kirian RA, Seibert MM, Montanez PA, Kenney C, Herbst R, Hart P, Pines J, Haller G, Gruner SM, Philipp HT, Tate MW, Hromalik M, Koerner LJ, van Bakel N, Morse J, Ghonsalves W, Arnlund D, Bogan MJ, Caleman C, Fromme R, Hampton CY, Hunter MS, Johansson LC, Katona G, Kupitz C, Liang M, Martin AV, Nass K, Redecke L, Stellato F, Timneanu N, Wang D, Zatsepin NA, Schafer D, Defever J, Neutze R, Fromme P, Spence JC, Chapman HN and Schlichting I (2012). "High-resolution protein structure determination by serial femtosecond crystallography." *Science* **337**: 362-364.

Chapman HN, Fromme P, Barty A, White TA, Kirian RA, Aquila A, Hunter MS, Schulz J, DePonte DP, Weierstall U, Doak RB, Maia FR, Martin AV, Schlichting I, Lomb L, Coppola N, Shoeman RL, Epp SW, Hartmann R, Rolles D, Rudenko A, Foucar L, Kimmel N, Weidenspointner G, Holl P, Liang M, Barthelmess M, Caleman C, Boutet S, Bogan MJ, Krzywinski J, Bostedt C, Bajt S, Gumprecht L, Rudek B, Erk B, Schmidt C, Homke A, Reich C, Pietschner D, Struder L, Hauser G, Gorke H, Ullrich J, Herrmann S, Schaller G, Schopper F, Soltau H, Kuhnel KU, Messerschmidt M, Bozek JD, Hau-Riege SP, Frank M, Hampton CY, Sierra RG, Starodub D, Williams GJ, Hajdu J, Timneanu N, Seibert MM, Andreasson J, Rocker A, Jonsson O, Svenda M, Stern S, Nass K, Andritschke R, Schroter CD, Krasniqi F, Bott M, Schmidt KE, Wang X, Grotjohann I, Holton JM, Barends TR, Neutze R, Marchesini S, Fromme R, Schorb S, Rupp D, Adolph M, Gorkhover T, Andersson I, Hirsemann H, Potdevin G, Graafsma H, Nilsson B and Spence JC (2011). "Femtosecond X-ray protein nanocrystallography." *Nature* **470**: 73-77.

Ellis P and Bernstein H (2001). "CBFlib: An API for CBF/imgCIF Crystallographic Binary Files with ASCII Support.

Grosse-Kunstleve RW, Sauter NK, Moriarty NW and Adams PD (2002). "The Computational Crystallography Toolbox: crystallographic algorithms in a reusable software framework." *Journal of applied crystallography* **35**: 126-136.

Hart P, Boutet S, Carini G, Dubrovin M, Duda B, Fritz D, Haller G, Herbst R, Herrmann S, Kenney C, Kurita N, Lemke H, Messerschmidt M, Nordby M, Pines J, Schafer D, Swift M, Weaver M, Williams G, Zhu D, Van Bakel N and Morse J (2012). "The CSPAD megapixel x-ray camera at LCLS." *Proceedings of SPIE* **8504**: 85040C-85040C-85011.

Hattne J, Echols N, Tran R, Kern J, Gildea R, Brewster A, Alonso-Mori R, Glöckner C, Hellmich J, Laksmono H, Sierra R, Lassalle-Kaiser B, Lampe A, Han G, Gul S, DiFiore D, Milathianaki D, Fry A, Miahnahri A, White W, Schafer D, Seibert M, Koglin J, Sokaras D, Weng T, Sellberg J, Latimer M, Glatzel P, Zwart P, Grosse-Kunstleve R, Bogan M, Messerschmidt M, Williams G, Boutet S, Messinger J, Zouni A, Yano J, Bergmann U, Yachandra V, Adams P and Sauter N (submitted). "The accurate processing of diffraction data from X-ray free-electron lasers."

Kern J, Alonso-Mori R, Tran R, Hattne J, Gildea RJ, Echols N, Glockner C, Hellmich J, Laksmono H, Sierra RG, Lassalle-Kaiser B, Koroidov S, Lampe A, Han G, Gul S, Difiore D, Milathianaki D, Fry AR, Miahnahri A, Schafer DW, Messerschmidt M, Seibert MM, Koglin JE, Sokaras D, Weng TC, Sellberg J, Latimer MJ, Grosse-Kunstleve RW, Zwart PH, White WE, Glatzel P, Adams PD, Bogan MJ, Williams GJ, Boutet S, Messinger J, Zouni A, Sauter NK, Yachandra VK, Bergmann U and Yano J (2013). "Simultaneous femtosecond X-ray spectroscopy and diffraction of photosystem II at room temperature." *Science* **340**: 491-495.

Parkhurst J, Brewster A, Fuentes-Montero F, Waterman D, Hattne J, Ashton A, Echols N, Evans G, Sauter N and Winter G (in preparation). "dxtbx: the diraction experiment toolbox."

Sauter NK, Hattne J, Grosse-Kunstleve RW and Echols N (2013). "New Python-based methods for data processing." *Acta crystallographica. Section D, Biological crystallography* **69**: 1274-1282.

Waterman DG, Winter G, Parkhurst JM, Fuentes-Montero L, Hattne J, Brewster A, Sauter NK and Evans G (2013). "The DIALS framework for integration software." *CCP4 Newsletter on Protein Crystallography* **49**: 13-15.

Quantum Mechanics-based Refinement in Phenix/DivCon

Oleg Y. Borbulevych[a], Nigel W. Moriarty[b], Paul D. Adams[b,c] and Lance M. Westerhoff[a]

[a] *QuantumBio Inc, 2790 West College Ave, State College, PA, 16801, USA*

[b] *Lawrence Berkeley National Laboratory, Berkeley, CA 94720*

[c] *Department of Bioengineering, University of California at Berkeley, Berkeley, CA 94720*

Correspondence email: lance@quantumbioinc.com

Introduction

Conventional macromolecular crystallographic refinement relies on *a priori* determined stereochemistry restraints and a simple restraint function to ensure the correct model geometry of the macromolecule along with any bound ligands, cofactors and metal coordination spheres. The benefit of this method is in its speed: entire structures can be refined and re-refined many times in a cycle in order to arrive at a structure the practitioner expects or trusts. However, unfortunately, as revealed in a recent survey of ligand structures deposited to PDB, on the order of 60% of all ligands within the PDB have questionable or wrong geometry (Gore *et al.*, 2011). This problem can often be traced to deficiencies or inaccuracies both in the ligand restraints and in the energy functional. In terms of the restraints, correct creation of an accurate file requires an *a priori* understanding of the ligand geometry within the active site. Traditionally, this is an error prone process, and tools such as *eLBOW* (Moriarty *et al.*, 2009) have helped immensely. However, even perfectly created restraints may not capture the influences of intermolecular covalent and non-bonded interactions, metal coordination and/or solvation. These influences are left to the energy functional to capture and compared with modern molecular and quantum mechanics methods, this energy functional is very simple in nature as it is missing electrostatics, polarization, hydrogen bonds, dispersion, *etc*.

In order to help address these deficiencies, we have integrated the semiempirical quantum mechanics (SE-QM) engine DivCon (Borbulevych et al., 2014) with the *Phenix* crystallographic package (Adams *et al.*, 2010). DivCon uses a fast, all-atom, linear-scaling, semiempirical quantum mechanics (SE-QM) method (Dixon & Merz, 1996, 1997, QuantumBio & Inc, 2011) to routinely characterize structures with thousands or even 10's (or 100's) of thousands of atoms. This *Phenix*/DivCon method - invoked using the `phenix.refine` (Afonine *et al.*, 2012) command line argument, `qblib=True` - does not rely on *a priori*-determined stereochemical restraints. Instead, it uses the same SE-QM Hamiltonians (AM1(Dewar *et al.*, 1985), PM3(Stewart, 1989)and PM6(Stewart, 2009, Hostas *et al.*, 2013)) available in advanced computational chemistry tools to calculate the gradients required to drive structure optimization and refinement. With this method, SE-QM gradients are calculated in "real-time" at each cycle of the LBFGS minimization for the entire structure or just the region(s) of interest. Since the region includes not only the ligand but the surrounding active site as well, the QM protocol also replaces link restraints and will model the geometry surrounding intermolecular covalent bonds, metal coordination spheres and so on.

Region QM Refinement

The DivCon package utilizes the linear scaling QM methodology that decreases the computational time as compared with conventional QM calculations. To optionally further speed-up the calculation, the command line argument `qblib_region_selection` has been added to carry out a region-specific QM-based refinement on the area(s) of interest within the protein/ligand complex (figure 1). All residues within a certain distance of any atom of the ligand are defined as a part of the main or core region (argument: `qblib_region_radius`). This core region is the region for which the atomic SE-QM gradients are determined and used at each step of the refinement. The second SE-QM region, referred to as a buffer region, includes any residues surrounding the core region (argument: `qblib_buffer_radius`). By using a buffer region to chemically insulate the core region, we gain a significant speed-up versus a full QM treatment, and at the same time, we limit any errors that may occur in the gradients due to capping or the artificial chemical environment surrounding the core region. In this case, the QM-gradients generated from the atoms within the buffer region

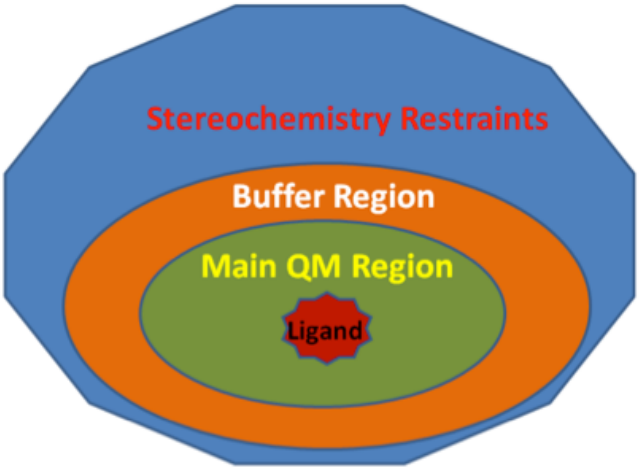


Figure 1. The schematic representation of the region QM refinement concept. The ligand and surrounding protein residues consists of the main QM region that is treated at the QM level as well as the buffer region. The rest of the protein is treated with the conventional stereochemistry restraints.

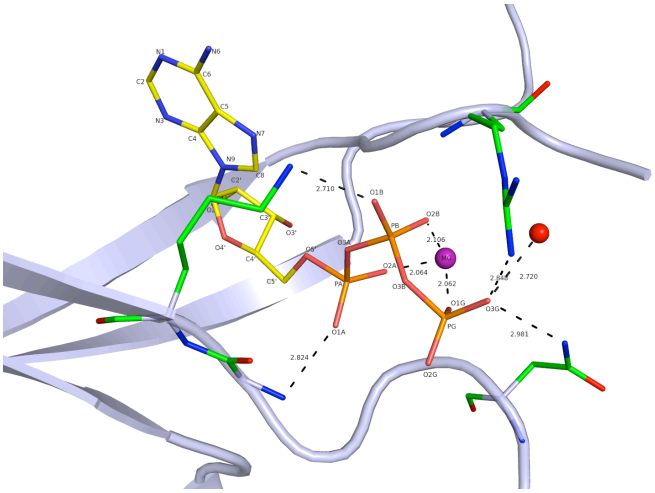


Figure 2. Intermolecular interactions of ATP (*yellow*) and in the protein structure 4AFF.

are not used in the refinement and the standard stereochemical restraints are used instead. This regional QM method can also be applied concurrently to more than one region within the complex.

For the examples presented here, the published atom placement for the ligand and for the protein is adopted at the beginning of the refinement. It should be noted that the QM method is not a "silver bullet," and the successful investigator will still need to protonate the structure and perform any initial atomic placement (by hand or through the use of experimental density-aware docking techniques). Also, while the restraints are provided to `phenix.refine` to satisfy its built-in error-trapping features, unless the `macro_cycle_to_skip` command line parameter is used to run conventional refinement prior to QM refinement, the restraints are not actually used in QM refinement. Therefore, any ligand(s) will need to be chemically correct upon initial placement. Here, just as in anywhere else, the concept of "garbage-in/garbage-out" is important.

Example #1: ATP geometry - QM refinement of PDB:4AFF

For the first example, adenosine triphosphate (ATP) is a cofactor found in numerous macromolecular structures and is therefore of significant interest to the community. PDBid:4AFF contains the ATP coordinated to the Mg^{2++} (figure 2). The default PM6 Hamiltonian (Stewart, 2009, Hostas *et al.*, 2013), which has published support for 70 elements, is used in this refinement.

In order to trigger the addition of both *Phenix* and DivCon to your `$PATH`, both the `phenix_env` and `qbenv.csh` file (`phenix_env.sh` & `qbenv.sh` for bash) will need to be sourced as per scheme 1 (assuming the csh shell is used).

Once this initial atom placement and chemical connectivity has been adopted, QuantumBio provides a `qbphenix` Perl script that can be used to prepare the PDB file and run `phenix.refine` as required. This script is configured to run either the *Phenix ReadySet!* protonation or the Protonate3D protonation found in MOE from the Chemical Computing Group, Inc. (CCG) depending upon availability. For the example in scheme 2, the *ReadySet!* tool will be used.

4AFF Re-refinement Results

The region QM refinement of 4AFF centered around ATP in *Phenix* yielded R_{work}=0.175 and R_{free}=0.181. Despite the good 2Fo-Fc density in the region of ATP we find that there are several noticeable differences in the QM refined geometry

```
% source /path/to/phenix-dev-1555-or-newer/phenix_env
% source /path/to/DivConDiscoverySuite-b####/etc/qbenv.csh
```

Scheme 1: Environment setup commands for Phenix/DivCon

```
% qbphenix --pdbFile 4AFF.pdb --sfFile 4AFF.mtz --chain A          \
     --resname ATP --resid 1117 --region-radius 4.0                \
     --buffer-radius 3.0 --protonation ReadySet
% phenix.refine 4AFF.pdb 4AFF.mtz 4AFF.cif qblib=True              \
     qblib_region_radius=4.0 qblib_buffer_radius=3.0               \
     qblib_region_selection="chain A and resname ATP and resid 1117"
```

Scheme 2. Commands to run DivCon refinement in Phenix using the 4AFF example.

Table 1. Selected geometry parameters in ATP.

Parameter	QM refined	Phenix Restraint Value	Small Molecule Crystal Structure[±]
PA-O3A-PB	126	120.5	125(1)-129.3(2)
O3A-PB-O3B	105	108.2	101.9(2)-104(1)
PB-O3B-PG	128	120.5	127(1)-130.5(2)
PA-PB-PG	86	-	84.6(3)-85.4(7)

compared with the Monomer Library (v4.3) values for ATP (table 1).

While it is expected that the SE-QM method and the simplified force field in *Phenix* would provide different results, initially this difference was flagged as significant. Tamasi *et al.* (Tamasi *et al.*, 2010) performed crystallographic studies on a number of small molecule systems with ATP and demonstrated that crystalline water molecules as well as counterions affect the molecule geometry of ATP. In 4AFF, the 3-phosphate moiety of ATP forms numerous H-bonds with protein residues as well as is involved in the coordination with magnesium and one water molecule (figure 2). When comparing these results with those observed in the QM-based refinement, we find that the geometry parameters in question are very close to the corresponding experimental data (table 1). Conventional restraint targets on the other hand deviate significantly from the experimental values listed in table 1 suggesting that in this case, *a priori* prediction of the molecular geometry can be a challenging task especially when non-bonded interactions significantly influence the ligand/co-factor geometry. In order to capture this sort of influence in conventional refinement, the practitioner would need to know enough about the bound state and manipulate the restraints in order to mimic the chemistry that is automatically captured using SE-QM.

Example #2: Covalently Bound Ligand - QM refinement of PDB:2V6N

Suicide inhibitors represent a traditional challenge for conventional refinement (Kleywegt, 2007). Such ligands are covalently bound to certain amino acids (*e.g.* SER or LYS) thus becoming a part of the polypeptide chain. This bond makes it difficult to choose restraints for the chemically modified amino acid *and* the ligand. Furthermore, structural changes of the ligand (and amino acid) due to this bond can "trickle down" through the molecule affecting adjacent bond angles/lengths/etc.

The standard approach to a bound ligand in *Phenix* is to use the `phenix.ligand_linking` program to generate the two files required by `phenix.refine` to add the bonds and angles to the restraints model. The bond values used are from QM calculations, however, the angles are not as precisely determined.

For this refinement, perhaps surprisingly, there is no significant difference between `phenix.refine` command on the non-covalently bound ligand and the covalently bound ligand (see scheme 3). In QM, there is no concept of "explicitly defined" covalent bonds. As before, *ReadySe*t! was chosen for this protonation method in order to simplify the use of the *Phenix* suite. MOE could have been used as well (`--protonation` **`MOE`**).

```
% qbphenix --pdbFile 2V6N.pdb --sfFile 2V6N.mtz --chain A          \
     --resname XP1 --resid 2307 --region-radius 3.0                \
     --buffer-radius 2.5 --protonation ReadySet
% phenix.refine 2V6N.pdb 2V6N.mtz 2V6N.cif                         \
     refinement.input.xray_data.labels=FP,SIGFP qblib=True         \
     qblib_region_radius=3.0 qblib_buffer_radius=2.5               \
     qblib_region_selection="chain A and resname XP1 and resid 2307"
```

Scheme 3. Commands to run DivCon refinement in Phenix using the 2V6N example.

Table 2. Selected bond lengths (Å) and bond angles (°) in the structure 2V6N.

Parameters	QM Refinement	*Phenix* Refinement	Original PDB
CAC XP1- CAD XP1	1.46	1.48	1.52
SG Cys145-CAC XP1	1.74	1.81	1.83
CB Cys145-SG Cys145-CAC XP1	101	101	91
SG Cys145-CAC XP1- OAH XP1	123	118	127
SG Cys145-CAC XP1- CAD XP1	115	116	111

2V6N Re-refinement Results

The structure 2V6N determined at 1.98 Å resolution has revealed the SARS coronavirus main proteinase inactivated by the covalently bound ligand 4-dimethylaminobenzoic acid (XP1). The covalent bond is made between the carbonyl carbon of the ligand and the sulfur atom of CYS 145 (table 2).

The deposited structure exhibits several geometry distortions in the region of the linkage bond between the ligand and Cys145. In particular, the linkage bond S-C (1.83 Å) is longer than the normal C_{sp2}-S bond (1.75 Å) (Allen *et al.*, 1987). Notably, the C-S-C bond angle is 91°, which is much smaller than the expected value between 99-109° depending on the chemistry (Shigeru & Joyce, 1991).

The re-refined structure using the standard *Phenix* procedure described earlier produces a linkage bond of 1.81 Å despite the ideal value being 1.78 Å. There is also a marked deviation from the ideal restraint value for the C-S-C angle. The model value is 101° while the ideal value was tetrahedral. This demonstrates that the density influences the final structure strongly.

The key advantage of the QM refinement that *no geometry restraints* for the ligand and the surrounding active site, including the linkage C-S bond, are needed. QM refinement leads to correct geometry of the ligand and the ester bond and the structure is completely fixed relative to the original PDB model. Notably, the C-S-C angle became 101° that is within the acceptable range. Figure 3 depicts the refined structure vs. the originally deposited structure.

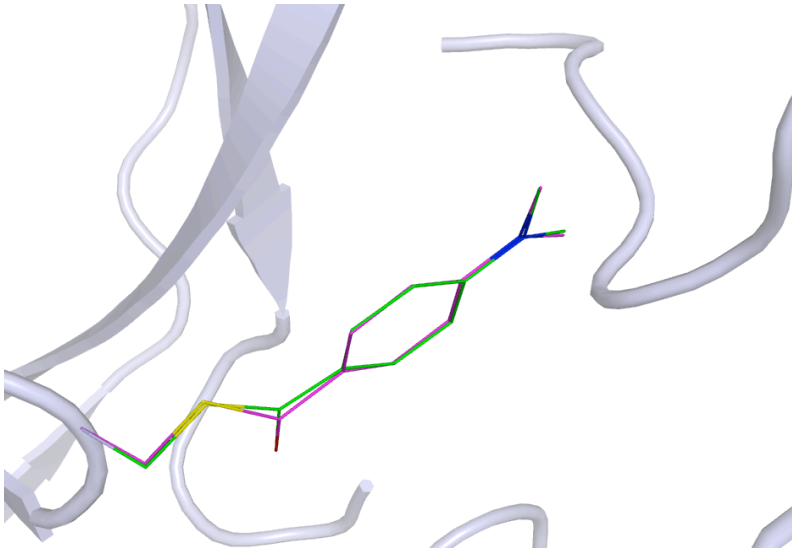


Figure 2. Superimposition of the QM re-refined PDB structure 2V6N (green) with the original PDB (magenta).

Discussion & Further information

X-ray crystallography is a crucial tool in the modern, industrial and academic structure based drug discovery toolbox, and refined crystal structures often form the basis of core discovery research projects. Unfortunately, with conventional methods based on *a priori* restraints and simple functions, it is not unusual for investigators to download a crystal structure of interest, add protons, and re-optimize the once carefully refined heavy atoms positions using any number of different molecular mechanics force fields (e.g. AMBERFF, MMFF, etc). With the *Phenix*/DivCon integration, it is hoped that this practice can become less prevalent as resulting models adopt structures that are chemically correct at the outset since these technologies are the same methods that are used in some of the most advanced computational chemistry and molecular modeling approaches available. Further, since *Phenix*/DivCon actually uses QM to treat the target:ligand complex together during the refinement, this method will be able to capture the interactions between the various species and ultimately provide greater insight into the inner workings of the active site. The following is a list of websites that provide additional information on the use and access of these technologies:

- Usage Tutorial: http://www.quantumbioinc.com/support/manual-phenixdc/tutorial
- Performance Guidelines: http://www.quantumbioinc.com/support/manual-phenixdc/guidelines
- FAQ: http://www.quantumbioinc.com/support/manual-phenixdc/phenix_divcon_faq
- Publications: http://www.quantumbioinc.com/publications?tag=xray
- Licensing Information: http://www.quantumbioinc.com/products/software_licensing

References

Adams, P. D., Afonine, P. V., Bunkoczi, G., Chen, V. B., Davis, I. W., Echols, N., Headd, J. J., Hung, L. W., Kapral, G. J., Grosse-Kunstleve, R. W., McCoy, A. J., Moriarty, N. W., Oeffner, R., Read, R. J., Richardson, D. C., Richardson, J. S., Terwilliger, T. C. & Zwart, P. H. (2010). Acta Crystallographica Section D-Biological Crystallography 66, 213-221.

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 Crystallographica Section D 68, 352-367.

Allen, F. H., Kennard, O., Watson, D. G., Brammer, L., Orpen, A. G. & Taylor, R. (1987). Journal of the Chemical Society-Perkin Transactions 2 S1-S19.

Borbulevych, O. Y., Plumley, J. A., Martin, R. M., Merz, K. M. & Westerhoff, L. M. (2014). Acta Crystallographica Section D-Biological Crystallography, in press.

Dewar, M. J. S., Zoebisch, E. G., Healy, E. F. & Stewart, J. J. P. (1985). Journal of the American Chemical Society 107, 3902-3909.

Dixon, S. L. & Merz, K. M. (1996). Journal of Chemical Physics 104, 6643-6649.

Dixon, S. L. & Merz, K. M. (1997). Journal of Chemical Physics 107, 879-893.

Gore, S., Tjelvar, S., Olsson, G. & Zhuravleva, M. (2011). Acta Crystallographica Section A: Foundations of Crystallography A67, C104.

Hostas, J., Rezac, J. & Hobza, P. (2013). Chem. Phys. Lett. 568-569, 161-166.

Kleywegt, G. J. (2007). Acta Crystallographica Section D-Biological Crystallography 63, 94-100.

Moriarty, N. W., Grosse-Kunstleve, R. W. & Adams, P. D. (2009). Acta Crystallographica Section D-Biological Crystallography 65, 1074-1080.

QuantumBio & Inc (2011). LibQB. Version 5.0, www.quantumbioinc.com.

Shigeru, O. & Joyce, D. (1991). Organic Sulfur Chemistry.

Stewart, J. J. P. (1989). Journal of Computational Chemistry 10, 209-220.

Stewart, J. J. P. (2009). Journal of Molecular Modeling 15, 765-805.

Tamasi, G., Berrettini, F., Hursthouse, M. B. & Cini, R. (2010). The Open Crystallography Journal 3, 1-13.