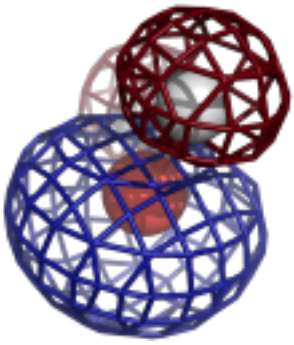
VOLUME ONE



COMPUTATIONAL CRYSTALLOGRAPHY NEWSLETTER

JULY MMX

IOTBX.PDB SECONDARY-STRUCTURE SPOTFINDER

Table of Contents

Editor

Nigel W. Moriarty, NWMoriarty@LBL.Gov

Contributors

P. D. Adams, P. V. Afonine, V. B. Chen, N. Echols, J. J. Headd, R. W. Grosse-Kunstleve, N. W. Moriarty, D. C. Richardson, J. S. Richardson, N. K. Sauter, T. C. Terwilliger, A. Urzhumtsev

PHENIX News

New releases

A new feature in the *PHENIX* GUI is designed to compare the refined structures of similar proteins. Structure comparison uses the protein sequences to display the differences between each protein chain loaded in both a table format and graphically in *COOT*. Several features of the protein structure can be compared including rotamers and secondary structure. NCS chains can be overlaid in *COOT*, edited and saved in the original orientation.

The Java kinemage viewer *KiNG* (Protein Science 2009, 18:2403-2409) has been incorporated into *PHENIX*, so that the *KiNG* jar files and supporting scripts will be part of the distribution package, while relying on the Java virtual machine that is standard on the user's Mac or Linux platform. Without any other setup, the *KiNG* program in *PHENIX* would allow viewing of macromolecular structures directly from PDB files, and viewing of kinemages (such as multi-criterion kinemages from MolProbity). We are currently working to incorporate validation kinemage creation directly within the *PHENIX* GUI, allowing even more seamless and complete user evaluation of model quality during the refinement process. Later developments might also provide kinemage displays to provide help in evaluating other aspects of refinement and model building progress.

In a similar vein to `phenix.superpose_pdbs`, a new program has been added to *PHENIX* that is specifically designed for superposing ligands. Superposing protein models is generally done using the C_α positions. Ligands require different algorithms to create atomic correspondences, some of which has been implemented in *eLBOW*

The Computational Crystallography Newsletter (CCN) is a regularly distributed electronically via email and the PHENIX website, www.phenix-online.org/newsletter. Feature articles, meeting announcements and reports, information on research or other items of interest to computational crystallographers or crystallographic software users should be submitted to the editor at any time for consideration. Submission of text by email or word-processing files using the CCN templates is requested.

(Moriarty et. al., Acta Cryst., D65, 2009, 1074–1080). *eLBOW*'s methods including graph matching are used to match the atoms from a single molecule or group of residues to another single molecule. The resulting correspondences can be used in several ways. The molecules can be aligned using the least-squares residual of the distances between matched atoms; the exact position of the corresponding atoms can be transferred; or the PDB attributes such as atom name and residue name can be transferred. A file containing a protein and ligand can be used as input and multiple instances of the ligand can be handled. The program can be accessed by running `phenix.superpose_ligands`.

A tool developed especially for our industrial consortium members and designed to decrease the time to fit a ligand by using a previously positioned ligand in the same or similar protein model has been added to *PHENIX*. Guided Ligand Replacement (*GLR*) matches the protein models, determines the location of the fit ligand, performs an atomic match of the ligands and inserts the overlaid ligand into the apo protein model. As a final step, a real-space refinement is performed on the ligand and the surrounding protein. GLR attempts to match all instances of the guiding ligand in the asymmetric unit.

New features

RESOLVE in *PHENIX* is now entirely C++ code. This allows the use of the `cctbx` routines in *RESOLVE* and speeds up *RESOLVE* model-building by a factor of nearly two. It also allows caching of resolve libraries, further speeding up *RESOLVE* in *PHENIX*.

Ligand fitting in *PHENIX* now incorporates real-space refinement, yielding greatly improved fits of ligand to density.

Loop libraries have been created for rapid fitting of short loops with `phenix.fit_loops`.

The ligand geometry restraints editor, *REEL*, has a new feature that allows searching of the Chemical Components Dictionary available at http://www.wwpdb.org/ccd.html and also distributed with PHENIX. The user can search using various ligand attributes including name and chemical formula. The results can be viewed in the molecular viewing window.

Current PHENIX development

Work is currently underway to improve the support for carbohydrates in macromolecular models. The GlycoCT format (Herget et al., Carbohydr. Res., 2008, 343:2162-2171) which can specify a carbohydrate sequence has been chosen as the base format for file based input. Other formats will be supported. A GUI input is also being developed as well as tools to handle models already containing the carbohydrates. It is expected that ligand fitting will be expanded to improve the fitting of carbohydrates focusing on the saccharide chains that commonly occur in protein models. Branching will be supported from the inception.

Crystallographic meetings and workshops

2010 Australasian Crystallography School, 17-24 July, 2010

The 2010 Australasian Crystallography School is being held at the University of Queensland, Brisbane, Australia from the 17th to the 24th of July. Pavel Afonine will be teaching a module on macro-molecular refinement.

GRC – Diffraction Method in Crystallography, 18-23 July 2010

Several of the *PHENIX* developers will be attending the Gordon Conference entitled "Diffraction Methods in Crystallography" to be held at Bates College, Lewiston, Maine from the 18th to the 23rd of July. There will be posters on the various aspects of *PHENIX* including the graphical user interface, validation and ligands. Drop by during the poster sessions to speak to the developers.

ACA – 2010 Annual Meeting, 24-29 July, 2010

The annual meeting of the American Crystallographic Association will be held in Chicago, Illinois from the 24th to the 29th of July.

22nd PHENIX Workshop, 11-13 October, 2010

The semi-annual *PHENIX* workshop is being held in Cambridge, England from the 11th to the 13th of October. Developers will be presenting the latest programs and feature releases on Monday the

11th. All interested parties in the area are invited to attend the Monday sessions.

5th PHENIX User's Workshop, 14 October, 2010

Following the *PHENIX* Workshop in Cambridge U.K., a user's workshop will provide teaching and hands-on experience with *PHENIX* for the novice and expert.

Paris Workshop, 15 October, 2010

A contingent of *PHENIX* developers will be at Institut Pasteur, Paris, France on the 15th of October to participate in a workshop for the local users. The workshop is being organized by Claudine Mayer and Deshmukh Gopaul and is funded by the "Groupe Thematique Biologie" of the French Crystallography Association.

Expert advice

Geometry restraint ESD

Many users enquire about the details of the geometry restraints used in macromolecular refinement programs. Refmac, *COOT* and *PHENIX* use a CIF format; an example of the restraints for water appears at right.

```
data_comp_list
loop_
_chem_comp.id
_chem_comp.three_letter_code
_chem_comp.name
_chem_comp.group
_chem_comp.number_atoms_all
_chem_comp.number_atoms_nh
_chem_comp.desc_level
HOH         HOH 'water           ' ligand 3 1 .
data_comp_HOH
loop_
_chem_comp_atom.comp_id
_chem_comp_atom.atom_id
_chem_comp_atom.type_symbol
_chem_comp_atom.type_energy
_chem_comp_atom.partial_charge
_chem_comp_atom.x
_chem_comp_atom.y
_chem_comp_atom.z
HOH O  O OH2  .   -0.2400    0.0000   -0.3394
HOH H1 H HOH2 .    0.8400    0.0000   -0.3394
HOH H2 H HOH2 .   -0.6000    0.0000    0.6788
loop_
_chem_comp_bond.comp_id
_chem_comp_bond.atom_id_1
_chem_comp_bond.atom_id_2
_chem_comp_bond.type
_chem_comp_bond.value_dist
_chem_comp_bond.value_dist_esd
HOH  H1       O     single        1.080 0.020
HOH  H2       O     single        1.080 0.020
loop_
_chem_comp_angle.comp_id
_chem_comp_angle.atom_id_1
_chem_comp_angle.atom_id_2
_chem_comp_angle.atom_id_3
_chem_comp_angle.value_angle
_chem_comp_angle.value_angle_esd
HOH  H2       O      H1           109.47 3.000
```

Much of the information in the file is straightforward including atom names, elements and Cartesian coordinates. Even the bonding information is mostly transparent. The atoms involved, the type of bond and the ideal bond lengths are easily discerned. The last number on the line, however, is not so simple. The ESD is an Estimated Standard Deviation that allows a refinement program to estimate the gradients of the force on the two atoms in the bond using a parabola. The units of the ESD are the same as the ideal value so for bond lengths the unit is Ångstrom. The larger the ESD value, the more flexible the restraint will be in the refinement. Conversely, reducing the number will restrain the internal coordinate to remain closer to the ideal value.

There is a limit to the amount one can reduce the ESD of a restraint. In the extreme, setting the ESD value to zero will remove the restraint from the refinement. Also, reducing the ESD to a value that is orders of magnitude smaller than the other ESD values in the same class will greatly reduce the weight of the other restraints relative to the highly restrained value and may even adversely affect the weight of the x-ray term.

Generally, it is best to have the ESD values approximately the same as the expected accuracy of the experiment. In the case of the bond lengths, an ESD of 0.01 Ångstrom could be considered a reasonable lower bound. An angle ESD of about 1 degree and torsion ESD of 5 degrees are also reasonable as a lower limit.

FAQ

How do I create restraints for my ligand in PHENIX?

There are a number of ways. If you know the ligand code corresponding to the ligand in the PDB databases such as the Chemical Components, then you can use *eLBOW* thus:

```
phenix.elbow --chemical-component=ATP
```

The resulting files, `ATP.pdb` and `ATP.cif`, will have the data such as atom names consistent with the Chemical Components database.

cctbx PDB handling tools

Ralf W. Grosse-Kunstleve[a] and 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: RWGrosse-Kunstleve@LBL.Gov

Introduction

The PDB format is the predominant working format for atomic parameters (coordinates, occupancies, displacement parameters, etc.) in macromolecular crystallography. Many small-molecule programs also support this format. The PDB format specifications are available at http://www.pdb.org/. Technically, the format is very simple, therefore a vast number of parsers exist in scientific packages. This article is about the PDB handling tools included in the Computational Crystallography Toolbox (`cctbx`, http://cctbx.sourceforge.net/), the open-source component of the *PHENIX* project (http://www.phenix-online.org/).

The evolution of the `cctbx` PDB handling tools has gone through three main stages spread out over several years. A simple parser implemented in Python has been available for a long time. In many cases Python's runtime performance is sufficient for interactive processing of PDB files, but can be limiting for large files, or for repeatedly traversing the entire PDB database. This has prompted us to implement a fast C++ parser that is described in Grosse-Kunstleve et al. (2006). However, initially the fast `cctbx` PDB handling tools only supported "read-only" access. Writing of PDB files was supported only at a very basic level. This shortcoming has been removed and the current cctbx version provides comprehensive tools for reading, manipulating, and writing PDB files. These tools are available from both Python and C++, under the `iotbx.pdb` module.

This article presents an overview of the main types in the `iotbx.pdb` module, considerations that lead to the design, and related important nomenclature. It is not a tutorial for using the `iotbx.pdb` facilities. For this, refer to http://cctbx.sourceforge.net/sbgrid2008/tutorial.html. See also http://cci.lbl.gov/hybrid_36/ which describes `iotbx.pdb` facilities for handling very large models.

Real-world PDB files

The simplicity of the PDB format is only superficial and, in the general case, stops after the initial parsing level. The structure of the PDB file implies a *hierarchy* of objects. A first approximation is this hierarchical view is:

```
model
  chain
    residue
      atom
```

This is only an approximation because of a feature that is easily overlooked at first: the "`altloc`" (official PDB nomenclature) column 17 of PDB ATOM records, specifying "alternative location" identifiers for atoms in alternative conformations. As it turned out, about 90% of the development time invested into `iotbx.pdb` was in some form related to alternative conformations. Our goal was to provide robust tools that work even for the most unusual (but valid) cases, since this is a vital characteristic of any automated system. The main difficulties encountered while pursuing this goal were:

- Chains with conformers that have different sequences
- Chains with duplicate `resseq+icode` (residue sequence number + insertion code)
- Conformers interleaved or separated

These difficulties are best illustrated with examples. An old version of PDB entry 2IZQ (from Aug 2008) includes a chain with conformers that have different sequences, the residue with `resseq` 11 in chain A, atoms 220 through 283:

```
HEADER    ANTIBIOTIC                              26-JUL-06   2IZQ
ATOM    220  N  ATRP A  11      20.498  12.832  34.558  0.50  6.03           N
................ATRP A  11 ..................................................
ATOM    243  HH2ATRP A  11      15.522   9.077  38.323  0.50 10.40           H
ATOM    244  N  CPHE A  11      20.226  13.044  34.556  0.15  6.35           N
................CPHE A  11 ..................................................
ATOM    254  CZ CPHE A  11      16.789   9.396  34.594  0.15 10.98           C
ATOM    255  N  BTYR A  11      20.553  12.751  34.549  0.35  5.21           N
................BTYR A  11 ..................................................
ATOM    261  CD1BTYR A  11      18.548  10.134  34.268  0.35  9.45           C
ATOM    262  HB2CPHE A  11      21.221  10.536  34.146  0.15  7.21           H
ATOM    263  CD2BTYR A  11      18.463  10.012  36.681  0.35  9.08           C
ATOM    264  HB3CPHE A  11      21.198  10.093  35.647  0.15  7.21           H
ATOM    265  CE1BTYR A  11      17.195   9.960  34.223  0.35 10.76           C
ATOM    266  HD1CPHE A  11      19.394   9.937  32.837  0.15 10.53           H
ATOM    267  CE2BTYR A  11      17.100   9.826  36.693  0.35 11.29           C
ATOM    268  HD2CPHE A  11      18.873  10.410  36.828  0.15  9.24           H
ATOM    269  CZ BTYR A  11      16.546   9.812  35.432  0.35 11.90           C
ATOM    270  HE1CPHE A  11      17.206   9.172  32.650  0.15 12.52           H
ATOM    271  OH BTYR A  11      15.178   9.650  35.313  0.35 19.29           O
ATOM    272  HE2CPHE A  11      16.661   9.708  36.588  0.15 11.13           H
ATOM    273  HZ CPHE A  11      15.908   9.110  34.509  0.15 13.18           H
ATOM    274  H  BTYR A  11      20.634  12.539  33.720  0.35  6.25           H
................BTYR A  11 ..................................................
ATOM    282  HH BTYR A  11      14.978   9.587  34.520  0.35 28.94           H
```

The original atom numbering does not have gaps. Here we have omitted blocks of atoms with constant `resname` and `resseq+icode` to save space.

As of Jun 8 2010, there are 74 files with mixed residue names in the PDB, i.e. only about 0.1% of the files. However, these files are perfectly valid and a PDB processing library is suitable as a component of an automated system only if it handles them sensibly.

An old version of PDB entry 1ZEH (Aug 2008) includes a chain with consecutive duplicate `resseq+icode`, atoms 878 through 894:

```
HEADER    HORMONE                                 01-MAY-98   1ZEH
HETATM  878  C1 ACRS     5      12.880  14.021   1.197  0.50 33.23           C
HETATM  879  C1 BCRS     5      12.880  14.007   1.210  0.50 34.27           C
................ACRS     5 ..................................................
HETATM  892  O1 ACRS     5      11.973  14.116   2.233  0.50 34.24           O
HETATM  893  O1 BCRS     5      11.973  14.107   2.248  0.50 35.28           O
HETATM  894  O   HOH      5      -0.924  19.122  -8.629  1.00 11.73           O
HETATM  895  O   HOH      6     -19.752  11.918   3.524  1.00 13.44           O
HETATM  896  O   HOH      7      -1.169  17.936  -6.103  1.00 12.89           O
```

To a human inspecting the old 1ZEH entry, it is of course immediately obvious that the water with `resseq` 5 is not related to the previous residue with `resseq` 5. However, arriving at this conclusion with an automatic procedure is not entirely straightforward. The human brings in the knowledge that water atoms without hydrogen are not covalently connected to other atoms. This is very detailed, specialized knowledge. Introducing such heuristics into an automatic procedure is likely to lead to surprises in some situations and is best avoided, if possible.

In the PDB archive, alternative conformers of a residue always appear consecutively. However, as mentioned in the introduction, the PDB format is also the predominant working format. Some programs produce files with conformers separated in this way (this file was provided to us by a user):

```
ATOM   1716  N  ALEU   190      28.628   4.549  20.230  0.70  3.78           N
ATOM   1717  CA ALEU   190      27.606   5.007  19.274  0.70  3.71           C
ATOM   1718  CB ALEU   190      26.715   3.852  18.800  0.70  4.15           C
ATOM   1719  CG ALEU   190      25.758   4.277  17.672  0.70  4.34           C
ATOM   1829  N  BLEU   190      28.428   4.746  20.343  0.30  5.13           N
ATOM   1830  CA BLEU   190      27.378   5.229  19.418  0.30  4.89           C
ATOM   1831  CB BLEU   190      26.539   4.062  18.892  0.30  4.88           C
ATOM   1832  CG BLEU   190      25.427   4.359  17.878  0.30  5.95           C
ATOM   1724  N  ATHR   191      27.350   7.274  20.124  0.70  3.35           N
ATOM   1725  CA ATHR   191      26.814   8.243  21.048  0.70  3.27           C
ATOM   1726  CB ATHR   191      27.925   9.229  21.468  0.70  3.73           C
ATOM   1727  OG1ATHR   191      28.519   9.718  20.259  0.70  5.22           O
ATOM   1728  CG2ATHR   191      28.924   8.567  22.345  0.70  4.21           C
ATOM   1729  C  ATHR   191      25.587   8.983  20.559  0.70  3.53           C
ATOM   1730  O  ATHR   191      24.872   9.566  21.383  0.70  3.93           O
                        ... residues 191 through 203 not shown
ATOM   1828  O  AGLY   203       8.948  14.861  23.401  0.70  5.84           O
ATOM   1833  CD1BLEU   190      26.014   4.711  16.521  0.30  6.21           C
ATOM   1835  C  BLEU   190      26.506   6.219  20.135  0.30  4.99           C
ATOM   1836  O  BLEU   190      25.418   5.939  20.669  0.30  5.91           O
ATOM   1721  CD2ALEU   190      24.674   3.225  17.536  0.70  5.31           C
ATOM   1722  C  ALEU   190      26.781   6.055  20.023  0.70  3.36           C
ATOM   1723  O  ALEU   190      25.693   5.796  20.563  0.70  3.68           O
ATOM   8722  C  DLEU   190      26.781   6.055  20.023  0.70  3.36           C
ATOM   8723  O  DLEU   190      25.693   5.796  20.563  0.70  3.68           O
ATOM   9722  C  CLEU   190      26.781   6.055  20.023  0.70  3.36           C
ATOM   9723  O  CLEU   190      25.693   5.796  20.563  0.70  3.68           O
```

In this file, conformers A and B of residue 190 appear consecutively, but conformers C and D appear only after conformers A and B of all residues 191 through 203. While this is not the most intuitive way of ordering the residues in a file, it is still considered valid because the original intention is clear. Since it was our goal to develop a comprehensive library suitable for automatically processing files produced by any popular program, we found it important to correctly handle non-consecutive conformers.

iotbx.pdb.hierarchy

When developing the procedure capable of handling the variety of real-world situations shown above, we strived to keep the underlying rule-set as simple as possible and to avoid highly specific heuristics (e.g. "water is never covalently bound"). Complex rules imply complex implementations, are difficult to explain and understand, tend to lead to surprises, and are therefore likely to be rejected by the community. With this and the real-world situations in mind, we arrived at the following *primary* organization of the PDB hierarchy:

```
Primary PDB hierarchy:

  model(s)
    id
    chain(s)
      id
      residue_group(s)
        resseq
        icode
        atom_group(s)
          resname
          altloc
          atom(s)
```

In this presentation the "(s)" indicates a list of objects of the given type. i.e. a hierarchy contains a list of models, each model has an "`id`" (a simple string) and holds a list of chains, etc.

Comparing with the "first approximation hierarchy" above, the `residue` type is replaced with two new types: `residue_group` and `atom_group`. These types had to be introduced to cover all the real-world cases shown above. The `residue_group` and `atom_group` types are new and unusual. Before we go into the details of these types, it will be helpful to consider the bigger picture by introducing the alternative *secondary* view of the PDB hierarchy:

```
Secondary view of PDB hierarchy:

  model(s)
    id
    chain(s)
      id
      conformer(s)
        altloc
        residue(s)
          resname
          resseq
          icode
          atom(s)
```

This organization is probably more intuitive at first. The first two levels (`model`, `chain`) are exactly the same as in the primary hierarchy. Each chain holds a list of `conformer` objects, which are characterized by the `altloc` character from column 17 in the PDB ATOM records. A `conformer` is understood to be a *complete copy* of a chain, but usually two conformers *share* some or even most atoms. A `residue` is characterized by a unique `resname` and the `resseq+icode`.

The secondary view of the hierarchy evolved in the context of generating geometry restraints for refinement (e.g. bond, angle, and dihedral restraints), where this organization is most useful. It is also the organization introduced in Grosse-Kunstleve et al. (2006), where it was actually the primary organization. While working with the conformer-residue organization, we found that it is difficult to manipulate a hierarchy (e.g. add or delete atoms) in obvious ways. Finally, while developing the automatic generation of constrained occupancy groups for alternative conformations, the conformer-residue organization proved to be unworkable. The main difficulty is that the relative order of residues with alternative conformations is lost in the conformer-residue organization; it is only given indirectly by interleaved residues with shared atoms -- if they exist. As convenient as the `conformer-residue` organization is for the generation of restraints, it is a hindrance for other purposes.

The names for the new types in the primary hierarchy were chosen not to collide with the secondary view. We could have used "`conformer`" and "`residue`" again, just reversed, but there would be the big surprise that one residue has different `resnames`. To send the signal "this is not what you usually think of as a residue", we decided to use `residue_group` as the type name. A residue group holds a list of `atom_group` objects. All atoms in an atom group have the same `resname` and `altloc`. Therefore "`resname_altloc_group`" would have been another plausible name, but we favored `atom_group` since it is more concise and better conveys what is the main content.

Detection of residue groups and atom groups

When constructing the primary hierarchy given a PDB file, the processing algorithm has to detect models, chains, residue groups, atom groups and atoms. Most steps are fairly straightforward, but none of the steps is actually completely trivial. For example, what to do if TER or ENDMDL cards are missing? What if residue sequence numbers are not consecutive? The ribosome community widely uses `segid` instead of chain `id` (even though the `segid` column is officially deprecated by the PDB). What if a file

contains both chain `id` and `segid`? Documenting all our answers in full detail is beyond the scope of this article (and would be more distracting than helpful anyway because the source code is openly available). Therefore we concentrate on the most important rules for the detection of residue groups and atom groups.

Then central conflict we had to resolve was:

- In order to handle non-consecutive conformers, we have to use the `resseq+icode` to find and group related residues.
- However, we cannot use the `resseq+icode` alone as a guide, because we also want to handle chains with duplicate `resseq+icode` (consecutive or non-consecutive).

This lead to a two stage procedure. In the first stage:

- A residue group is given by a block of consecutive ATOM or HETATM records with identical `resseq+icode` columns (SIGATM, ANISOU, and SIGUIJ records may be interleaved), e.g.

```
ATOM    234  H   ATRP A  11      20.540  12.567  33.741  0.50  7.24           H
ATOM    235  HA ATRP A  11      20.771  12.306  36.485  0.50  6.28           H
ATOM    244  N   CPHE A  11      20.226  13.044  34.556  0.15  6.35           N
ATOM    245  CA CPHE A  11      20.950  12.135  35.430  0.15  5.92           C
```

However, there is an important pre-condition:

- Unless a sub-block of ATOM records with identical `resname+resseq+icode` columns contains a main-conformer atom (almost "blank `altloc`"), e.g.

```
HEADER    ANTIBIOTIC RESISTANCE                   07-MAY-97   1AJQ
CRYST1   52.120   65.080   76.300 100.20 111.44 105.81 P 1           1
HETATM 6097 CA    CA     1       5.676  34.115  52.446  1.00 18.50          CA
HETATM 6100  C2  SPA     1      11.860  36.159  33.853  1.00 14.30           C
HETATM 6107  C6  SPA     1      13.085  36.522  34.644  1.00 17.34           C
```

In this case the sub-block is assigned to a separate residue group.

Within each residue group, all atoms are grouped by `altloc+resname` and assigned to atom groups. The order of the atoms in a residue group does not affect the assignment to atom groups.

After all atom records are assigned to residue groups and atom groups,

- residue groups with identical `resseq+icode`
- that do not contain main-conformer atoms

are merged in the second processing stage. While two residue groups are merged, atom groups with identical `altloc+resname` from the two sources are also merged.

Note that the grouping steps in this process may change the order of the atoms. However, our implementation preserves the relative order of the atoms as much as possible. I.e. in the hierarchy, `resseq+icode` appear in the original "first seen" order, and similarly for `altloc+resname` within a residue group.

Construction of secondary view of hierarchy

The secondary view of the hierarchy is constructed trivially from the primary hierarchy objects. For each chain, the complete list of `altloc` characters is determined in a first pass. In a second pass, a conformer object is created for each `altloc`, and a second loop over the chain assigns the atoms to each conformer. Main-conformer atoms are assigned to all conformers, atoms in alternative

conformations only to the corresponding conformer. Because of the difficulties alluded to earlier, the conformer and residue objects in the secondary view are "read-only". I.e. all manipulations such as addition or removal of residues, have to be performed on the primary hierarchy. Re-constructing the secondary view after the primary hierarchy has been changed is very fast (fractions of seconds even for the largest files, e.g. 0.22 s for PDB entry 1HTQ with almost one million atoms).

The `iotbx/examples/pdb_hierarchy.py` script shows how to construct the primary hierarchy from a PDB file, and how to obtain the secondary view.

Nomenclature related to alternative conformations

The `iotbx.pdb` module uses the following nomenclature to describe various aspects of alternative conformations:

- ***Main conf. atom*** : an atom with
  - a blank `altloc` character
  - and no other atom with the same `name+resname` (but different `altloc`) in the residue group. This second condition is needed for cases like this:

```
HEADER    HYDROLASE                               12-MAR-04   1SO7
ATOM   2460  CG1 VAL A 325     -23.284  97.713  15.815  0.66 21.74           C
ATOM   2461  CG1AVAL A 325     -23.010  97.616  18.295  0.66 22.88           C
ATOM   2462  CG2BVAL A 325     -24.819  96.373  17.146  0.66 22.57           C
```

- ***Alt. conf. atom*** : not main conf.
- ***Conformer*** : a complete chain with main conf. atoms and alt. conf. atoms with a specific `altloc`.

  Note for completeness: it is also possible to obtain conformers of an individual residue group. However, conceptually this is best viewed as a shortcut for first obtaining the conformers of a chain, and then finding the residue of interest in each conformer, ignoring all other residues.

- ***Residue*** : complete residue in a conformer with main conf. atoms and alt. conf. atoms.

Residues are classified as follows:

- ***pure main conf.*** : all main conf. atoms
- ***pure alt. conf.*** : all alt. conf. atoms
- ***proper alt. conf.*** : both main conf. atoms and alt. conf atoms, all alt conf. atoms have a non-blank `altloc` column
- ***improper alt. conf.*** : both main conf. atoms and alt. conf atoms, one or more alt conf. atoms have a blank `altloc` column (as shown in the 1SO7 fragment above).

Errors and warnings

The tools in `iotbx.pdb` are designed to be as tolerant as reasonably possible when processing input PDB files. E.g., as of Jun 8 2010, all 65802 files in the PDB archive can be processed without generating exceptions. However, to assist users and developers in creating PDB files without ambiguities, the hierarchy object provides methods for flagging likely problems as errors and warnings. It is up to the application how to react to the diagnostics.

The `phenix.pdb.hierarchy` command can be used to quickly obtain a summary of the hierarchy in a PDB file and some diagnostics. This example command highlights all main features:

```
phenix.pdb.hierarchy pdb1jxw.ent.gz
```

Output:

```
file pdb1jxw.ent.gz
  total number of:
    models:       1
    chains:       2
    alt. conf.:   5
    residues:    49
    atoms:      786
    anisou:       0
  number of atom element+charge types: 5
  histogram of atom element+charge frequency:
    " H  " 383
    " C  " 261
    " O  "  77
    " N  "  59
    " S  "   6
  residue name classes:
    "common_amino_acid" 48
    "other"              1
  number of chain ids: 2
  histogram of chain id frequency:
    " " 1
    "A" 1
  number of alt. conf. ids: 3
  histogram of alt. conf. id frequency:
    "A" 2
    "B" 2
    "C" 1
  residue alt. conf. situations:
    pure main conf.:     32
    pure alt. conf.:      3
    proper alt. conf.:   14
    improper alt. conf.:  0
  chains with mix of proper and improper alt. conf.: 0
  number of residue names: 16
  histogram of residue name frequency:
    "CYS" 6
    "THR" 6
    "ALA" 5
    "ILE" 5
    "PRO" 5
    "GLY" 4
    "ASN" 3
    "SER" 3
    "ARG" 2
    "LEU" 2
    "TYR" 2
    "VAL" 2
    "ASP" 1
    "EOH" 1    other
    "GLU" 1
    "PHE" 1
  ### WARNING: consecutive residue_groups with same resid ###
  number of consecutive residue groups with same resid: 2
    residue group:
      "ATOM    378  N   PRO A  22 .*.     N  "
      ... 12 atoms not shown
      "ATOM    391  HD3APRO A  22 .*.     H  "
    next residue group:
      "ATOM    392  CB BSER A  22 .*.     C  "
      ... 6 atoms not shown
      "ATOM    399  HB3CSER A  22 .*.     H  "
    ----------------------------------------
    residue group:
      "ATOM    432  N   LEU A  25 .*.     N  "
      ... 18 atoms not shown
      "ATOM    439  CD1CLEU A  25 .*.     C  "
    next residue group:
      "ATOM    452  CG1BILE A  25 .*.     C  "
      ... 13 atoms not shown
      "ATOM    466 HD13CILE A  25 .*.     H  "
```

The diagnostics are mainly intended for PDB working files, but we have tested the `iotbx.pdb` module by processing the entire PDB archive. (This took about 3700 CPU seconds, but using 40 CPUs the last job finished after only 277 seconds. Disk and network I/O is rate-limiting in this case, not the performance of the PDB handling library.) The results are summarized in the following table:

```
Total number of .ent files: 65802 (2010 Jun 8)

56873   WARNING: duplicate chain id
    3   WARNING: consecutive residue_groups with same resid
    2   ERROR:   duplicate atom labels
    1   ERROR:   improper alt. conf.
    0   ERROR:   duplicate model id
    0   ERROR:   residue group with multiple resnames using same altloc
```

About 86% of the PDB entries re-use the same chain id for multiple chains (which are either separated by other chains or TER cards). In many cases, the re-used chain id is the blank character, which is clearly a minor issue. However, we flag this situation to assist people in producing new PDB files with unambiguous chain ids.

The next item in the list, "consecutive residue_groups with same resid" (where `resid=resseq+icode`), was mentioned before. This situation is best avoided to minimize the chances of mis-interpreting the PDB files.

"duplicate atom labels" pose a serious practical problem (therefore flagged as "ERROR") since it is impossible to uniquely select atoms with duplicate labels, for example via an atom selection syntax as used in many programs (*CNS*, *PyMOL*, *PHENIX*, *VMD*, etc.) or via PDB records in the connectivity annotation section (LINK, SSBOND, CISPEP). Atom serial numbers are not suitable for this purpose since many programs do not preserve them.

Only one PDB entry (1JRT) includes "improper alt. conf." as introduced in the previous section.

There are no PDB entries with two other potential problems diagnosed by the iotbx.pdb module. MODEL ids are unique throughout the PDB archive (as an aside: and all MODEL records have a matching ENDMDL record). Finally, in all 74 files with mixed residue names (i.e. conformers with different sequences), there is exactly one residue name for a given altloc.

Acknowledgments

We gratefully acknowledge the financial support of NIH/NIGMS under grant number P01GM063210. Our work was supported in part by the US Department of Energy under Contract No. DE-AC02-05CH11231.

References

Grosse-Kunstleve, R.W., Zwart, P.H., Afonine, P.V., Ioerger, T.R., Adams, P.D. (2006). Newsletter of the IUCr Commission on Crystallographic Computing, 7, 92-105.

Secondary structure restraints in phenix.refine

Nathaniel Echols[a], Jeffrey J. Headd[a], Pavel Afonine[a], and 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: NEchols@LBL.Gov

Introduction

We have implemented simple, automatic restraints for common secondary structure elements in proteins and nucleic acids, which can help reduce over-fitting at low-to-moderate resolution and preserve the secondary structure geometry that can be distorted due to the low resolution. Internally, these are simply distance restraints between hydrogen-bonding atoms in helices, sheets, or nucleic acid base pairs, with or without explicit hydrogens. The current method has the advantage of being fast, easily reduced to relatively simple input parameters, and useful for improving poor-quality regions of structure where the bonding may not be automatically recognized.

General description and syntax

A new command for identifying secondary structure and generating the necessary parameters, `phenix.secondary_structure_restraints`, was introduced in version 1.6.1, and most of the functionality is also accessible through `phenix.refine` itself. In the absence of user-defined atom selections, or if the parameter `find_automatically=True` is set, the program will look first in the header of the input PDB file(s) and parse any HELIX and SHEET records (*Figure 1*). Note that when

```
HELIX    8   8 ASP A  181  ARG A  191  1                                  11
HELIX    9   9 SER A  192  ASP A  194  5                                   3
HELIX   10  10 SER A  195  GLN A  209  1                                  15
SHEET    1   A 5 ARG A  13   ASP A  14   0
SHEET    2   A 5 LEU A  27   SER A  30  -1  O  ARG A  29   N  ARG A  13
SHEET    3   A 5 VAL A 156  HIS A 159   1  O  VAL A 156  N  PHE A  28
SHEET    4   A 5 ASP A  51   ASP A  54   1  N  ALA A  51   O  LEU A 157
SHEET    5   A 5 ASP A  74   LEU A  77   1  O  HIS A  74   N  VAL A  52
```

Figure 1. Beta PDB syntax for secondary structure records (excerpted from PDB ID 1ywf; Grundner et al. 2005).

running from `phenix.refine`, a different scope requires the use `secondary_structure.input.find_automatically=True`. If none are found, the open-source program *KSDSSP* (UCSF Computer Graphics Laboratory; Kabsch & Sander 1983) is used to generate these records based on the input geometry. Because the PDB format uses fixed columns (Bernstein et al. 1977, Berman et al. 2000) and is not easily edited, the records are converted to an intermediate format using the same parameter syntax and atom selection language as other programs in *PHENIX* (*Figure 2*; Adams et al. 2010). Example of use:

```
phenix.secondary_structure_restraints model.pdb > ss.eff
```

Although the PDB format specification provides for ten types of alpha helix, only the three most commonly found in natural proteins are processed: alpha (forming bonds between the carbonyl oxygen of residue n and the amide nitrogen of residue n+4), 3_{10} (n+3), and pi (n+5). As the alpha form is by far the most common, this is the default helix type. Beta strands have only two types, parallel and antiparallel, which can coexist in a single sheet (*Figure 3*). The parameter blocks for sheets are considerably more complicated because of the order-dependency of individual strands, and the requirement of explicit annotation of the start and end of hydrogen bonding between strand pairs. The parameters may be freely edited to add or remove secondary structure groups, but we recommend minimal changes to the sheets, due to the complexity of the definitions.

From within `phenix.refine` (Afonine et al. 2005), use of the restraints may be activated with the parameter `main.secondary_structure_restraints=True`. If no helix or sheet atom selections are included in the input parameters, the automatic identification procedure will be run; otherwise, the existing parameters are used without modification. Once secondary structure is assigned it is maintained for the rest of the refinement, but future versions will probably add the option to re-annotate the structure after each macro-cycle. The log file (and console output) will contain information about the overall secondary structure content, if any.

Hydrogen bond parameterization

Hydrogen bonds are modeled as simple harmonic restraints; we have not attempted to use a more physically rigorous hydrogen-bonding potential (for example, Fabiola et al. 2002). For maximum flexibility, either explicit or implicit hydrogen bonds are supported; the latter use a longer distance restraint between heavy atoms. The methods for converting atom selections into hydrogen bonds relies on several assumptions:

- The order of residues in the input PDB file exactly corresponds to the order in the protein itself
- Each secondary structure element is continuous with no missing residues
- All residues are complete, with no missing atoms

Unless the PDB file has undergone significant manual editing, these conditions are unlikely to be violated. The main exception is in the handling of hydrogen atoms, which are handled inconsistently by different crystallography applications. The program will first examine the PDB file to determine whether hydrogens are present, in which case it defaults to explicit hydrogen bonds. However, if newly built residues are missing hydrogen atoms, they will not be properly restrained. This can be remedied by running

```
refinement.secondary_structure {
  helix {
    selection = "chain 'A' and resseq 181:191"
  }
  helix {
    selection = "chain 'A' and resseq 192:194"
    helix_type = alpha pi *3_10 unknown
  }
  helix {
    selection = "chain 'A' and resseq 195:209"
  }
  sheet {
    first_strand = "chain 'A' and resseq 13:14"
    strand {
      selection = "chain 'A' and resseq 27:30"
      sense = parallel *antiparallel unknown
      bond_start_current = "chain 'A' and resseq 29"
      bond_start_previous = "chain 'A' and resseq 13"
    }
    strand {
      selection = "chain 'A' and resseq 156:159"
      sense = *parallel antiparallel unknown
      bond_start_current = "chain 'A' and resseq 156"
      bond_start_previous = "chain 'A' and resseq 28"
    }
    strand {
      selection = "chain 'A' and resseq 51:54"
      sense = *parallel antiparallel unknown
      bond_start_current = "chain 'A' and resseq 51"
      bond_start_previous = "chain 'A' and resseq 157"
    }
    strand {
      selection = "chain 'A' and resseq 74:77"
      sense = *parallel antiparallel unknown
      bond_start_current = "chain 'A' and resseq 74"
      bond_start_previous = "chain 'A' and resseq 52"
    }
  }
}
```

Figure 2. Equivalent `phenix.refine` parameters to Figure 1.

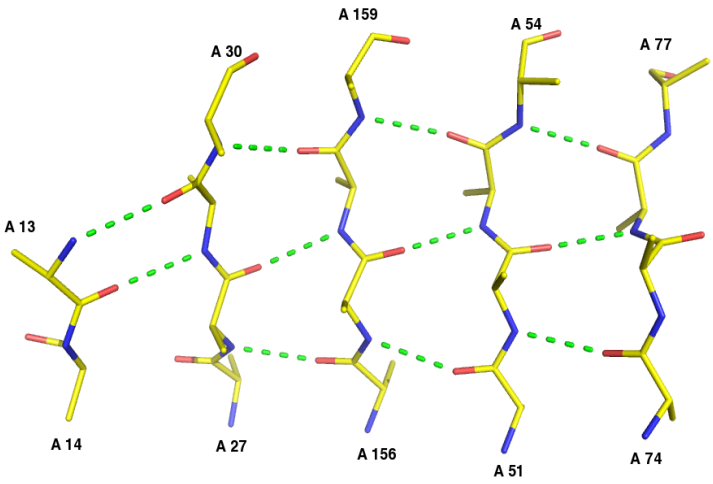


Figure 3. Beta sheet specified by the parameters in Figure 2, as rendered in *PyMOL*. (Sidechains have been omitted for clarity.) Note that not all residues annotated as belonging to strands actually form hydrogen bonds; the actual bonding pattern is dictated by the `bond_start_current` and `bond_start_previous` parameters for each strand.

`phenix.ready_set` prior to refinement to completely add hydrogens to the structure. You can also force `phenix.refine` to ignore any hydrogens present and use the implicit bonds by passing the parameter `substitute_n_for_h=True`.

All settings related to the configuration of hydrogen bonding are located in the `h_bond_restraints` block[1]. Currently, the distances used are as follows (N-O distances taken from Fabiola et al. 2002):

- explicit (H-O): 1.975 Å
- H-O outlier cutoff: 2.5 Å
- implicit (N-O): 2.9 Å
- N-O outlier cutoff: 3.5 Å

Both explicit and implicit bonds default to a sigma (standard deviation - essentially an inverse weight) of 0.05, and a slack of 0. Increasing the sigma reduces the strength of the bond restraints; increasing the slack allows it to move freely within a small range (+/- slack in either direction) before the restraint is applied. Initial experiments do not indicate any advantage to using a non-zero slack, but lower sigma values (e.g. 0.02) are beneficial in some cases, especially where explicit hydrogens are used. You may also override the global defaults and set separate sigma and slack values for each secondary structure group.

Once the hydrogen bonds have been identified, a short summary will be printed out to the log file/console:

```
109 hydrogen bonds defined.
Distribution of hydrogen bond lengths without filtering:
  2.7259 - 2.8381: 7
  2.8381 - 2.9504: 38
  2.9504 - 3.0626: 38
  3.0626 - 3.1748: 11
  3.1748 - 3.2870: 7
  3.2870 - 3.3993: 5
  3.3993 - 3.5115: 0
  3.5115 - 3.6237: 0
  3.6237 - 3.7360: 1
  3.7360 - 3.8482: 2
```

If annotation errors are present, the calculated bond lengths may cover a much wider range. For this reason, all outliers above a specified cutoff are filtered out before building the geometry restraints. Passing the parameter `remove_outliers=False` will override this, and in some instances may actually be beneficial, but we recommend visually inspecting the hydrogen bonds first to confirm that all are chemically appropriate. (This can be done in *PyMOL*; see instructions below.) After filtering, the output shown above will be repeated for the final bond list.

Nucleic acid base pairing

Version 1.6.2 adds partial support for hydrogen bond restraints between RNA base pairs (*Figure 4*) while more complete is in later nightly builds. Like *CNS* (Brünger et al. 1998), these are specified individually rather than as contiguous ranges. The automatic detection and hydrogen bond parameterization follows a similar procedure to that used for proteins, except that the all-atom contact analysis program *PROBE* (Word et al. 1999) is used to identify explicit hydrogen bonds, and a simplified list of base pairs is derived from these results. Only canonical Watson-Crick base pairs and GU pairs are supported at this time. Future versions will incorporate other known base pairings, classified according

[1] These parameters are used by both phenix.secondary_structure_restraints and phenix.refine; for the latter, the full scope name is actually refinement.secondary_structure.h_bond_restraints, but individual parameters passed as command line arguments should be recognized automatically.

to geometry as suggested by Leontis et al. (2002), versus the older Saenger (1984) Roman numeral nomenclature, which is no longer complete.

For users who require a more complete set of hydrogen bonds, we recommend the server provided by the Noller Lab at UCSC (http://rna.ucsc.edu/pdbrestraints/; Laurberg et al. 2008), which analyzes a PDB file and produces output in the custom bond format for `phenix.refine`. Note that we do not currently have any equivalent to "stacking restraints" available in *PHENIX*; additionally, although individual bases already have tight planarity restraints, the base pairs are not forced to be planar.

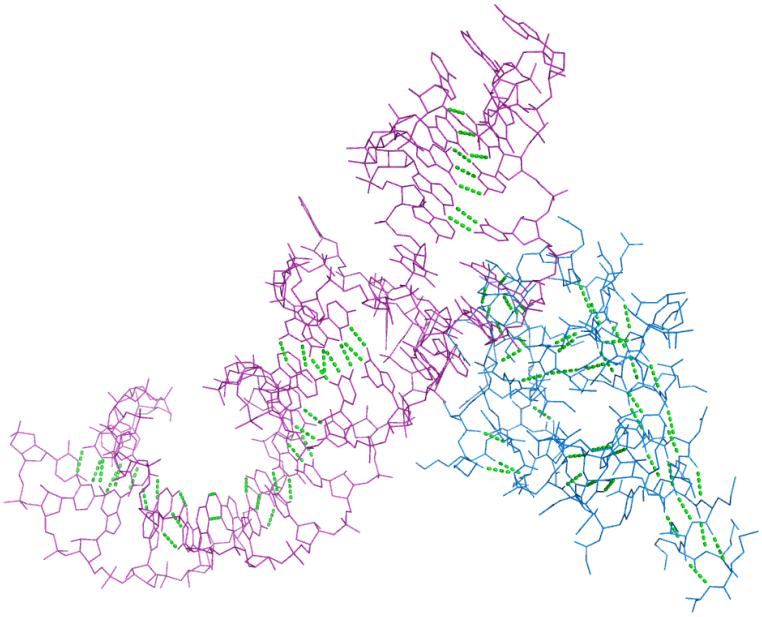


Figure 4. Automatically detected hydrogen bonds in a mixed protein-RNA structure, the signal recognition particle (PDB ID 1hq1; Batey et al. 2001).

Known limitations

The primary obstacle to generating these restraints is the difficulty of obtaining correct and complete annotations. The records in the PDB are often misleading and/or wrong; unfortunately, the annotations provided by *KSDSSP* are also occasionally incorrect. Common errors include two or more independent but adjacent helices being annotated as a single alpha helix, despite 90º bends or a 3_{10} helix in between. Both *KSDSSP* and *PROBE* are also very sensitive to input geometry, which means that poorly refined and/or manually built structures will not have all secondary structure elements detected automatically. For these situations, careful manual annotation is essential to maximize the usefulness of the added restraints. Work is underway on a graphical editor for picking secondary structures in a model (see also http://pymolwiki.org/index.php/ResDe for another approach).

At low resolution, a more serious problem is the lack of additional restraints on backbone conformation outside of regular secondary structure elements. When refining against data significantly worse than 3.0Å, Ramachandran outliers frequently comprise several percent of protein residues (0.2% is the limit recommended by Molprobity). Although `phenix.refine` does not currently offer Ramachandran restraints, version 1.6.2 introduces the separate option of restraining the model conformation to that of a high-resolution reference structure.

Additional caveats:

- No attempt is made to reconcile secondary structure restraints with NCS restraints. As always, care should be taken to exclude regions of genuine difference from NCS atom selections.
- Only bond length is restrained; angles may move freely (although in practice, other geometry restraints should limit the range of angles compatible with the specified bond length).
- Support for PDB files containing non-blank insertion codes is currently limited, especially if an insertion code occurs in the middle of a secondary structure element.
- The distribution of oxygen-nitrogen distances in beta sheets appears to be very slightly bimodal, probably due to the difference in geometry between parallel and antiparallel sheets. Increasing the slack may compensate for this, but it isn't clear whether this is necessary. Future versions may use

more intelligent distance parameters.
- Secondary structures which cross symmetry-related elements are not supported; this may occasionally happen with palindromic DNA or RNA helices. However, the custom bond syntax for `phenix.refine` may be used to manually specify hydrogen bonds.
- For extremely large structures (e.g. ribosomes), where the number of chains exceeds the number of available single-character codes, we encourage the use of two-character codes, which are also supported by Coot. However, if you prefer to use the deprecated segID, the parameters `preserve_protein_segid=True` and `preserve_nucleic_acid_segid=True` will include the segIDs in the output atom selections when identifying new secondary structure elements. (You do not need these parameters to run `phenix.refine` with the atom selections containing segIDs.)
- Detection of nucleic acid base pairs will only be run if RNA or DNA chains are identified in the PDB file. Some highly-modified RNA molecules such as tRNA may fail this procedure; for these cases, the parameter `force_nucleic_acids=True` provides a workaround.

Other uses

Two other output formats are available for the secondary structure restraints, albeit with more limited support: *PyMOL* commands for visualization (DeLano 2002), and distance restraints for *REFMAC* (Murshudov et al. 1997). A PDB file is required as input, with pre-defined atom selections optional. The same procedures described above for outlier filtering and atom selection are applied, so the final output should be identical to what would be obtained running phenix.refine with the same parameters.

- *PyMOL* format:

```
phenix.secondary_structure_restraints model.pdb \
      [restraints.eff] format=pymol > h_bonds.pml
```

- Example of output:

```
dist (chain 'A' and resi 9 and name N), (chain 'A' and resi 6 and name O)
```

- *REFMAC* format:

```
phenix.secondary_structure_restraints model.pdb \
      [restraints.eff] format=refmac > restraints.com
```

- Example of output:

```
exte dist first chain A residue 37 atom O \
      second chain A residue 41 atom N value 2.900 sigma 0.05
```

Restraints for *REFMAC* can be included as part of the input keywords; for *PyMOL*, once the PDB file is loaded distance objects can be created by selecting "Run. . ." from the "File" menu and selecting the `.pml` script, or by typing (for example) "`@/path/to/script/h_bonds.pml`" at the `PyMOL>` prompt. (For clarity, you may want to enter the command "`hide labels`" after running the script.)

Additional resources

- *KSDSSP* (included in *PHENIX* distribution):
  http://www.cgl.ucsf.edu/Overview/software.html#ksdssp
- ResDe (custom bond parameter editor for *PyMOL*):
  http://pymolwiki.org/index.php/ResDe
- Base pairing restraints generator (Noller Lab):
  http://rna.ucsc.edu/pdbrestraints/

Acknowledgements

We thank Peter Grey, Bradley Hintze, Dirk Kostrewa, and Francis Reyes for scientific discussions, Anton Vila-Sanjurjo for suggestions and code, Timm Maier for testing and feedback, and Francis Reyes and Allyn Schoeffler for sharing unpublished data. We gratefully acknowledge the *PHENIX* Industrial Consortium and NIH/NIGMS (grant P01GM063210) for financial support.

References

Adams PD, Afonine PV, Bunkóczi G, Chen VB, Davis IW, Echols N, Headd JJ, Hung LW, Kapral GJ, Grosse-Kunstleve RW, McCoy AJ, Moriarty NW, Oeffner R, Read RJ, Richardson DC, Richardson JS, Terwilliger TC, Zwart PH. PHENIX: a comprehensive Python-based system for macromolecular structure solution. *Acta Crystallogr D Biol Crystallogr*. **66**:213-21. PMID: 20124702

Afonine PV, Grosse-Kunstleve RW, Adams PD (2005). *CCP4 Newsletter.* **42**, contribution 8.

Batey RT, Sagar MB, Doudna JA. (2001) Structural and energetic analysis of RNA recognition by a universally conserved protein from the signal recognition particle. *J Mol Biol.* **307**:229-46. PMID: 11243816

Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. (2000) The Protein Data Bank. *Nucleic Acids Res.* **28**:235-42. PMID: 10592235

Bernstein FC, Koetzle TF, Williams GJB, Meyer EF Jr, Brice MD, Rodgers JR, Kennard O, Shimanouchi T, Tasumi M. (1977) The Protein Data Bank: a computer-based archival file for macromolecular structures. *J Mol. Biol.* **112**:535-42. PMID: 875032

Brünger AT, Adams PD, Clore GM, DeLano WL, Gros P, Grosse-Kunstleve RW, Jiang JS, Kuszewski J, Nilges M, Pannu NS, Read RJ, Rice LM, Simonson T, Warren GL. (1998) Crystallography & NMR system: A new software suite for macromolecular structure determination. *Acta Crystallogr D Biol Crystallogr*. **54**:905-21. PMID: 9757107

DeLano, W.L. The PyMOL Molecular Graphics System. (2008) DeLano Scientific LLC, Palo Alto, CA, USA. http://pymol.org

Fabiola F, Bertram R, Korostelev A, Chapman MS. (2002) An improved hydrogen bond potential: impact on medium resolution protein structures. *Protein Sci.* **11**:1415-23. PMID: 12021440

Grundner C, Ng HL, Alber T. (2005) Mycobacterium tuberculosis protein tyrosine phosphatase PtpB structure reveals a diverged fold and a buried active site. *Structure* **13**:1625-34. PMID: 16271885

Kabsch W, Sander C. (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* **22**:2577-637. PMID: 6667333

Laurberg M, Asahara H, Korostelev A, Zhu J, Trakhanov S, Noller HF. (2008) Structural basis for translation termination on the 70S ribosome. *Nature* **454**:852-7. PMID: 18596689

Leontis NB, Stombaugh J, Westof E. (2002) The non-Watson-Crick base pairs and their associated isostericity matrices. *Nucleic Acids Res.* **30**:3497-3531. PMID: 12177293

Murshudov GN, Vagin AA, Dodson EJ. (1997) Refinement of macromolecular structures by the maximum-likelihood method. *Acta Crystallogr D Biol Crystallogr.* **53**:240-55. PMID: 15299926

Saenger W. (1984) *Principles of Nucleic Acid Structure.* Springer-Verlag, New York, NY.

Word JM, Lovell SC, LaBean TH, Taylor HC, Zalis ME, Presley BK, Richardson JS, Richardson DC. (1999) Visualizing and quantifying molecular goodness-of-fit: small-probe contact dots with explicit hydrogen atoms. *J Mol Biol.* **285**:1711-33. PMID: 9917407

cctbx Spotfinder: a faster software pipeline for crystal positioning

Nicholas K. Sauter[a]

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

Correspondence email: NKSauter@LBL.Gov

Synopsis

This article documents the program `distl.signal_strength`, used to locate candidate Bragg spots on X-ray diffraction images for macromolecular crystallography. While standalone analysis requires about 3 seconds/image on typical Linux systems, an order of magnitude increase in the overall throughput can be achieved using concurrent multiprocessing within a client/server implementation. The program is thus suitable for the rapid location of the best-diffracting positions within the crystal sample using a low-dose X-ray probe.

Introduction

The development of microfocused X-ray beams at synchrotron facility endstations has made it possible to obtain higher-signal, lower mosaicity datasets from small crystal samples (Fischetti *et al.*, 2009). With small crystal dimensions of order 5 μm (matched to the diameter of present microbeams), it may be necessary to examine numerous specimens in order to assemble a complete dataset, making it highly desirable to automate the process of centering each sample in the microbeam. Various approaches have been proposed (Pothineni *et al.*, 2006; Song *et al.*, 2007), and it has been possible to automatically center the sample-holding loop (or other device) using videomicroscopy. However, it has been more difficult to visually identify small crystals within the loop, thus requiring the more robust approach of scanning the sample (translating it with respect to the microbeam), so as to collect a low-dose diffraction image at each translational position. It has been noted that such X-ray based autocentering may take up to 10 minutes for the smallest crystals (Song *et al.*, 2007), which presumably require very fine rastering to locate. This outcome could be improved dramatically by the use of a photon-counting pixel array detector such as the Pilatus-6M that supports a framing rate of 10 Hz. An accompanying challenge is to write software that can quantify the measured signal at a reasonably high turnaround rate, so that the diffraction analysis can keep pace with the rapid data acquisition needed for fine-grained coverage of the sample.

The standalone spotfinder program and client/server pair described below are meant to provide a toolbox for rapid diffraction analysis within the beamline computing environment. The software architecture is the same as described for the cctbx package (http://cctbx.sf.net; Grosse-Kunstleve *et al.*, 2002), with a high-level Python scripting interface that is meant to encourage the reuse and adaptation of code as the problems change. One example is that new detector hardware types can be readily supported as they are introduced.

Installation

The spotfinder program is bundled and distributed with the packages *LABELIT* and *PHENIX*, and can therefore be downloaded from either Web site (http://cci.lbl.gov/labelit/ or http://www.phenix-online.org/). Newest-version *PHENIX* installers are released on an almost daily basis, while *LABELIT* releases currently cycle every few months. Follow the installation instructions, and source the appropriate setup file (given in the message printed by the installer) to place the command-line dispatchers on path.

Command line reference

Use the command `distl.signal_strength` to produce a quick summary of the diffraction characteristics from a single diffraction exposure:

Usage:

```
distl.signal_strength image_filename [parameter=value [parameter2=value2
...]]
```

Example:

```
distl.signal_strength lysozyme_001.img distl.res.outer=2.0
```

Optional command line parameters change the program operation as follows (Default values are shown):

`distl.res.outer=None`

If specified, this is the outer (high) resolution limit to be used for the diffraction analysis, in Ångstroms. This option is useful for two reasons. Firstly, if a large number of images from the same crystal specimen are to be examined (in order to locate the best-diffracting position), then placing a uniform limit on the resolution range permits the number of observed Bragg spots to act as a good proxy of the diffraction strength. Secondly, areas on the image that are outside the outer resolution limit are not analyzed, potentially producing a substantial speedup in program throughput. [Note: areas on the stored image that are not part of the active detector are already excluded from analysis. Examples are the corner areas on circular image plates (such as the Mar), the inactive pixel stripes between fiber-optic tapers on CCD detectors (such as the ADSC), and the inactive pixel stripes between modules of the Pilatus detector.] Camera parameters such as the sample-to-detector distance and swing-arm angle are taken from the file header, to calculate the resolution at each pixel.

`distl.res.inner=None`

If specified, this is the inner (low) resolution limit to be used for the diffraction analysis, in Ångstroms. Excluding the low-resolution Bragg spots may be a potential workaround if there is severe beam leakage around the beam stop masquerading as a Bragg signal. However, the built-in heuristics that filter out unusually-large signal areas and very intense signals will normally exclude such artifacts anyway. The inner resolution cutoff has no effect on overall program throughput, as the filter is applied after the image is analyzed. This program option is only recommended for unusual situations.

`distl.minimum_signal_height=[1.5 for CCDs; 2.5 for pixel-array detectors]`

This parameter determines whether a given pixel is classified as background or signal. Active pixels are classified within non-overlapping 100×100-pixel tiles. For each tile, the best-fit plane is chosen to represent the background level. Pixel heights are distributed above and below this plane with a normal distribution whose standard deviation σ is readily calculated. Pixels higher than `minimum_signal_height`×σ above the background are classified as signal. Two successive rounds (now with 50×50-pixel tiles) are employed to remove the signal pixels from the background level. With CCD detectors the longstanding default of 1.5σ has been a highly successful compromise between increased sensitivity (favoring a lower cutoff) and better noise discrimination (favoring a higher cutoff; any cutoff lower than 1.25σ produces numerous false Bragg spots). For pixel-array detectors, which lack any significant point-spread function, 2.5σ gives better results and is now the pixel-array default. Images from very long unit cells (viruses, ribosomes) potentially have close spots that run into each other, without being separated down to the baseline. The heuristic rules in `distl` will reject these close signals, and in severe cases this will interfere with the ability to index the lattice. The solution is to raise the cutoff level to as high as 5σ, thus correctly separating the close spots. It is therefore important to consider raising the `minimum_signal_height` if the crystals have a large unit cell. At present there is no automatic way to do this; the program has no way to know the cell length ahead of time. In the future it will be beneficial to add some preliminary noise analysis to provide more sophisticated guidance for spot detection.

```
distl.minimum_spot_area=None [5 for pixel-array detectors, 10 otherwise]
```

If specified, this is the minimum number of contiguous pixels that is considered to be a spot. For image plate and CCD detectors the hard-coded default is 10 pixels/spot. With pixel-array technology, spots are much narrower due to the sharp point-spread function, but they are still generally distributed over a few pixels. The pixel-array default is therefore set to 5 pixels/spot. Synchrotron beamlines experimenting with new detectors or optics should test this parameter to find the optimum setting!

```
verbose=False
```

If True, the program will output detailed statistics on all Bragg spots and show signal pixel locations. This is potentially useful for beamline commissioning.

```
pdf_output=None
```

If specified, this is the filename (*.pdf) for an optional PDF-format graphical output showing the image and associated Bragg signals. This visual output is the most direct way to understand the "big picture" of what the program results mean. Comparing one's own visual impression of the diffraction image with the marked up spotfinder results will reveal whether the program has missed real Bragg spots (false negatives) or overinterpreted bad signals (false positives). The contrast level is pre-set internally. Pink stripes are used to color-code the inactive areas between detector modules (for the Pilatus), and red squares (again for the Pilatus) are inactive pixels, which should be the same on every image for a given instrument. Within Bragg spots, color circles indicate that the spot has been tagged as a "good Bragg spot candidate" (see below). Pink circles tag the position of the maximum pixel, and red circles show the spot center of mass.

```
--help
```

Produces an informative program synopsis.

Program operation and output

Computational steps performed by the program have already been described in several publications (Zhang *et al.*, 2006; Sauter *et al.*, 2004; Sauter & Zwart, 2009; Sauter & Poon, 2010). Typical results written to *stdout* are as follows:

```
                    File : 0_clmtr_edge_403.cbf
              Spot Total :    177
    In-Resolution Total :    170
  Good Bragg Candidates :    157
             Ice Rings :      1
    Method 1 Resolution :   2.55
    Method 2 Resolution :   2.08
      Maximum unit cell :  108.9
%Saturation, Top 50 Peaks :   0.09
In-Resolution Ovrld Spots :      0

Bin population cutoff for method 2 resolution: 20%

Number of focus spots on image #403 within the input resolution range: 170
Total integrated signal, pixel-ADC units above local background (just the good Bragg candidates) 147322
Signals range from 37.2 to 11773.1 with mean integrated signal 997.2
Saturations range from 0.0% to 0.7% with mean saturation 0.0%
```

Three types of spot count are listed:

- "Spot Total" is the number of separate spots that rise above the minimum thresholds for signal height and spot area. A graph of average pixel intensity vs. resolution is examined to prefilter likely ice rings.
- "In-Resolution Total" is the subset remaining after high- and low-resolution filters are applied. The command-line distl.res filter is applied here (if given), as well as the "Method 2 Resolution" filter described below.
- "Good Bragg Candidates" are the spots remaining after the application of several spot-quality heuristics:
    - A histogram of spot count vs. resolution is used to implement a second filter for ice rings.
    - Spots with ill-defined profiles having more than two signal maxima are not counted.
    - Outliers in intensity, area, eccentricity and skewness are thrown away.
    - Spots with too-close nearest neighbors are filtered.

Limiting resolution is estimated by the two methods defined in Zhang *et al.* (2006). Method 2 is used to choose spots for *LABELIT* autoindexing. It relies on the ability to produce a histogram of spot count vs. resolution, and therefore requires a minimum number of spots (usually 25). The resolution cutoff is determined by noting the falloff in bin population at higher diffraction angles. The maximum unit cell is estimated by observing nearest-neighbor distances and assuming that the closest spacing corresponds to the largest unit cell length.

"In-resolution" spots are used to produce several signal strength metrics: "% Saturation", "In-Resolution Overloaded Spots", "Signals range" and "Saturations range".

A final "Total integrated signal" metric is computed on the "Good Bragg Candidates". This value is the summed signal height (corrected for background) over all signal pixels. ADC units are analog-to-digital units, as recorded in the raw image file.

Performance assessment for crystal positioning

To position the crystal optimally in the beam, we aim to translate the sample to the position that maximizes the total number of good Bragg spots, or average intensity of Bragg spots, within a given resolution range. Low-dose X-rays can be used to perform this raster scan over the sample, as described (Song *et al.*, 2007). Very high throughput is achieved with shutterless exposures using a Pilatus-6M detector. However, the resulting turnaround time of about 0.2 seconds/image places very high performance requirements on the computational pipeline; which typically takes about 3 seconds to process one image. The magnitude of the challenge can be assessed by profiling the spotfinding code, as is done here with a 64-bit, 2.9 GHz Xeon machine running Fedora Core 8. The processor was equipped with 32 GB RAM and 16 CPU cores, although only one core was used by the single-threaded spotfinder process:

Computational step	Typical time/image	Comments
Load the dynamic libraries	0.50 sec	Client/server removes this overhead
Read file from NFS disk	0.70 sec	Local file I/O takes only 0.04 sec
Uncompress CBF image to memory	0.27 sec	cctbx-optimized code takes only 0.09 sec
Classify pixels: background/signal	0.63 sec	
First ice ring filter	0.23 sec	
Find all spots	0.14 sec	
Second ice ring filter	0.15 sec	
Additional heuristics for good spots	0.15 sec	
Total time:	2.77 sec	

Client-server architecture

The above-listed performance data show that an order-of-magnitude improvement is needed to keep pace with Pilatus-6M acquisition. We therefore move to a client-server architecture that eliminates the need to reload the dynamic libraries for each image (since the server process is persistent), and runs with multiple processes, so that numerous images may be processed concurrently within separate cores. At present, 16-core CPUs are available for under $6000, so they are within reach of beamline operating budgets. With this scheme it is easy to achieve the desired 0.2 second/image total throughput.

Server:

```
distl.mp_spotfinder_server_read_file [parameter=value ... ]
```

The program operates as a multithreaded server. Allowed parameters are:

- `distl.port=8125` (Required) The server will listen for requests on this port.
- `distl.processors=1` (Required) The total number of processes to be forked to listen for requests.
- `distl.minimum_spot_area=` same as above, the minimum spot area in pixels.
- `distl.minimum_signal_height=` same as above, the minimum signal height.
- `distl.res.outer=` same as above, the outer resolution limit in Ångstroms.

Client:

```
distl.thin_client <filepath> <host> <port>
```

No keyword parameters are allowed. The client simply takes the filepath (must be a valid filepath on the server machine), host name (usually "localhost") and port number, and outputs the spot analysis to *stdout*.

Notes:

The server can be killed from the command line by Ctrl-C, or via the client by sending the message:

```
distl.thin_client EXIT <host> <port>
```

While the server is supposed to queue requests, too many at once can cause some requests to drop out, in a manner that is not fully characterized. Therefore in the example given (`<path to sources>/spotfinder/servers/thin_client.csh`), a `/bin/sleep` command is used to time the requests at a reasonable pace given the particular server host and number of processes. In application, it is the responsibility of the caller (the beamline data collection process) to avoid overloading the server with too many simultaneous requests.

LABELIT contains a separate program (`labelit.distl`) to perform a similar spot finding function for use in autoindexing. That implementation is not suitable for multiprocessing because it writes the spot list to disk in the current working directory under a constant file name. The file can be erased with the command `labelit.reset`.

In contrast, `distl.signal_strength` and `distl.mp_spotfinder_server_read_file` perform all work in memory without any file output.

Results

Low-dose exposures were used to probe samples on a 5 × 6-position grid (Diamond, ADSC detector) or a

5 × 5-position grid (SSRL, Pilatus-6M). Images were visually inspected to produce a subjective rank for each exposure, taking into account factors such as the limiting resolution and strength of the Bragg spots. Separately, the images were analyzed with the automated spotfinder process, using an up-front resolution limit (`distl.res.outer`) of 3.0 Å.

An automated ranking scheme was developed that is consistent with the subjective rank from visual inspection. The 30 or 25 images from each sample are ranked by two criteria, the "Total Integrated Signal" in pixel-ADC units, and the count of "Good Bragg Candidates", and the rank scores based on these two criteria are averaged with equal weight. If two images receive the same average score, the tie is broken by giving a higher priority to "Total Integrated Signal". No score is developed unless there is a numerical score for "Method 2 Resolution" (although the resolution isn't actually included in the ranking); therefore images with too few spots are not scored.

Application programming interface

The spotfinder software can be accessed directly through Python code. Although the necessary interface is not formally documented, example usage is given by the program itself, within the `<path to sources>/spotfinder/` directory:

`./command_line/signal_strength.py`: Illustrates the use of command-line parameters.

`./applications/signal_strength.py`: Illustrates the core function calls, as well as the detailed parsing of the resulting spot list.

Acknowledgments

Numerous contributions from synchrotron groups helped to shape this software. Test results were contributed by Michael Soltis, Ana González and Penjit (Boom) Moorhead (Stanford Synchrotron Radiation Lightsource); Craig Ogata, Mark Hilgart and Sudhir Pothineni (GM/CA-CAT, Advanced Photon Source); John Skinner and Annie Heroux (National Synchrotron Light Source); and Alan Ashtun, Katherine McAuley, Graeme Winter and Mark Williams (Diamond Light Source, Ltd., UK). Herbert Bernstein (Dowling College) offered his expertise toward the optimization of CBF decompression, and Chris Nielson (Area Detector Systems Corp.) engaged in valuable discussions. Ralf Grosse-Kunstleve at LBNL helped to implement the overall software architecture. The financial support of NIH/NIGMS under grant number R01GM077071 is gratefully acknowledged. The work was partly supported by the US Department of Energy under Contract No. DE-AC02-05CH11231.

References

Fischetti, R.F., Xu, S., Yoder, D.W., Becker, M., Nagarajan, V., Sanishvili, R., Hilgart, M.C., Stepanov, S., Makarov, O. & Smith, J.L. (2009). Mini-beam collimator enables microcrystallography experiments on standard beamlines. *J. Synchrotron Rad.* **16**, 217-225.

Grosse-Kunstleve, R.W., Sauter, N.K., Moriarty, N.W. & Adams, P.D. (2002). The Computational Crystallography Toolbox: crystallographic algorithms in a reusable software framework. *J. Appl. Cryst.* **35**, 126-136.

Poon, B.K., Grosse-Kunstleve, R.W., Zwart, P.H. & Sauter, N.K. (2010). Detection and correction of underassigned rotational symmetry prior to structure deposition. *Acta Cryst.* D**66**, 503-513.

Pothineni, S.B., Strutz, T. & Lamzin, V.S. (2006). Automated detection and centring of cryocooled protein crystals. *Acta Cryst.* D**62**, 1358-1368.

Sauter, N.K., Grosse-Kunstleve, R.W. & Adams, P.D. (2004). Robust indexing for automatic data collection. *J. Appl. Cryst.* **37,** 399-409.

Sauter, N.K. & Zwart, P.H. (2009). Autoindexing the diffraction patterns from crystals with a pseudotranslation. *Acta Cryst.* D**65**, 553-559.

Sauter, N.K. & Poon, B.K. (2010). Autoindexing with outlier rejection and identification of superimposed lattices. *J. Appl. Cryst.* **43**, 611-616.

Song, J., Mathew, D., Jacob, S.A., Corbett, L., Moorhead, P. & Soltis, S.M. (2007). Diffraction-based automated crystal centering. *J. Synchrotron Rad.* **14**, 191-195.

Zhang, Z., Sauter, N.K., van den Bedem, H., Snell, G., and Deacon, A.M. (2006). Automated diffraction image analysis and spot searching for high-throughput crystal screening. *J. Appl. Cryst.* **39**, 112-119.

Atomic Displacement Parameters (ADPs), their parameterization and refinement in PHENIX

Pavel V. Afonine,[a] Alexandre Urzhumtsev,[b,c] Ralf W. Grosse-Kunstleve[a] and Paul D. Adams[a,d]

[a]*Lawrence Berkeley National Laboratory, Berkeley, CA 94720*
[b]*IGBMC, CNRS-INSERM-UdS, 1 rue Laurent Fries, B.P.10142, 67404 Illkirch, France*
[c]*Université Nancy; Département de Physique - Nancy 1, B.P. 239, Faculté des Sciences et des Technologies, 54506 Vandoeuvre-lès-Nancy, France.*
[d]*Department of Bioengineering, University of California at Berkeley, Berkeley, CA 94720*

Correspondence email: PAfonine@LBL.Gov

Introduction

This article describes the parameterization of, and refinement procedures for, Atomic Displacement Parameters (ADPs, or *B*-factors), as they are implemented in the crystallographic structure refinement program `phenix.refine` (Afonine *et al.*, 2005). The algorithms and parameterizations are implemented using an object oriented library approach and thus can be re-used in different contexts.

1. Atomic Displacement Parameters

Diffraction experiments produce data representing time- and space-averaged images of the crystal structure: time-averaged because atoms are in continuous thermal motions around mean positions, and space-averaged because there are often small differences between symmetry copies of the asymmetric unit in a crystal, especially in the case of macromolecular crystals. The dynamic displacements and the static spatial disorder lead to vanishing high-resolution data in reciprocal space and blurring of the diffracting density (electron or nuclear) in real-space. Ignoring the displacements when modeling the diffraction experiment would lead to a poor fit of the calculated data to the observed data. Modeling of the *small* dynamic displacements as isotropic or anisotropic harmonic displacements has been a standard practice from the earliest days of crystallography. *Larger* displacements (beyond harmonic approximation) can be modeled by using an anharmonic model (Gram-Charlier expansion; Johnson & Levy, 1974; Kendal & Stuart, 1958; not available in `phenix.refine`) or "alternative conformations".

The atomic displacement is a superposition of a number of contributions (Dunitz & White, 1973; Prince & Finger, 1973; Johnson, 1980; Sheriff & Hendrickson, 1987; Winn *et al.*, 2001), such as:

- Local atomic vibration
- Motion due to a rotational degree of freedom (e.g. libration around a torsion bond)
- Loop or domain movement
- Whole molecule movement
- Crystal lattice vibrations

More detailed models can be envisioned, but in practice many of today's refinement programs use an approximation and separate the total ADP, $\mathbf{U}_{\mathrm{TOTAL}}$, into three components:

$$\mathbf{U}_{\mathrm{TOTAL}} = \mathbf{U}_{\mathrm{CRYST}} + \mathbf{U}_{\mathrm{GROUP}} + \mathbf{U}_{\mathrm{LOCAL}} \qquad (1)$$

While the individual members of (1) represent different kinds of displacements, each of them can be modeled with different accuracy or using different models. For example, local atomic vibration, $\mathbf{U}_{\mathrm{LOCAL}}$, can be modeled using a less detailed, isotropic, model that uses only one parameter per atom. A more detailed (and accurate) anisotropic parameterization uses six parameters but requires more experimental observations to be practical. Group atomic displacement, $\mathbf{U}_{\mathrm{GROUP}}$, can be modeled using the TLS parameterization ($\mathbf{U}_{\mathrm{TLS}}$) or just one parameter per group of atoms (Fig. 1).

There are a relatively large number of conventions and notations used in connection with ADPs (**B**, **U**, $\mathbf{U}^{*}$, $\mathbf{U}_{\mathrm{CART}}$, $\mathbf{U}_{\mathrm{CIF}}$, etc); see Grosse-Kunstleve & Adams (2002) for a comprehensive review. In what follows we consistently use **U** assuming that the appropriate convention is used based on the context.

2. U_{CRYST}

$\mathbf{U}_{CRYST}$ (a symmetric 3x3 matrix) models the common displacement of the crystal (lattice vibrations) as a whole and some additional experimental anisotropic effects (Sheriff & Hendrickson, 1987; Usón *et al.*, 1999). This contribution is exactly the same for all atoms and thus it possible to treat this effect directly while performing overall anisotropic scaling (Afonine *et al.*, 2005) to compute the total model structure factors

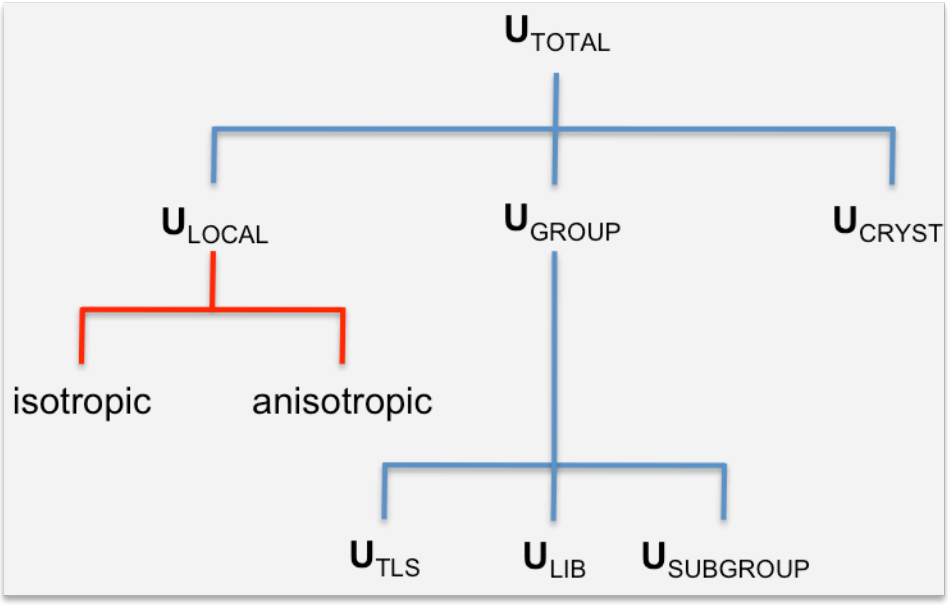


Figure 1. Hierarchy of contributions to atomic displacement parameter.

$$\mathbf{F}_{model} = k_{overall} \exp\left(-\frac{\mathbf{h}^t \mathbf{U}_{cryst} \mathbf{h}}{4}\right)\left(\mathbf{F}_{calc} + k_{sol} \exp\left(-\frac{B_{sol} s^2}{4}\right)\mathbf{F}_{mask}\right) \quad (2)$$

$\mathbf{U}_{CRYST}$ is forced to obey the crystal symmetry constraints. `phenix.refine` reports refined elements of $\mathbf{U}_{CRYST}$ matrix expressed in a Cartesian basis and uses the $\mathbf{B}_{CART}$ notation (for details, see Grosse-Kunstleve & Adams; 2002).

3. U_{GROUP}

$\mathbf{U}_{GROUP}$ is intended to model the contribution to $\mathbf{U}_{TOTAL}$ arising from concerted motions of multiple atoms (group motions). It allows for the combination of group motion at different levels (for example, *whole molecule + chain + residue*) and for the use of models of different degrees of sophistication (or accuracy), such as general TLS, TLS for a fixed axis (a librational ADP; $\mathbf{U}_{LIB}$), and a simple group isotropic model with one single parameter. In its most general form, $\mathbf{U}_{GROUP}$ can be $\mathbf{U}_{TLS}$ + $\mathbf{U}_{LIB}$ + $\mathbf{U}_{SUBGROUP}$, where, for example, $\mathbf{U}_{TLS}$ would model the motion of the whole molecule or a large domain, $\mathbf{U}_{SUBGROUP}$ would model the displacement of a smaller group such as a chain using a simpler one-parameter model and $\mathbf{U}_{LIB}$ would model a side chain libration around a torsion bond using a simplified TLS model (Stuart & Phillips, 1985). Depending on the context (model and data quality), not all these components can be realized. For example, $\mathbf{U}_{GROUP}$ may be just $\mathbf{U}_{TLS}$ or $\mathbf{U}_{SUBGROUP}$. Nested TLS parameterization (when a smaller TLS group can be enclosed within a larger TLS group) is not yet implemented in `phenix.refine`.

3.1. $U_{SUBGROUP}$

$\mathbf{U}_{SUBGROUP}$ represents one refinable isotropic ADP per group of selected atoms, with any number of groups. In `phenix.refine` there are two pre-defined selections for $\mathbf{U}_{SUBGROUP}$ to refine one or two (side+main chain atoms) $\mathbf{U}_{SUBGROUP}$ per residue. An arbitrarily selected set of atoms can also be a group. There is no general rule for choosing one parameterization over another; the choice depends on the resolution (data-to-parameter ratio) and is usually made by systematic testing using R_{work} and R_{free} as the criteria. No restraints are applied to $\mathbf{U}_{SUBGROUP}$, but the value can be constrained between predefined minimum and maximum values.

3.2. U_{LIB}

This is a special case of the TLS parameterization for a rigid-body motion that occurs around a fixed axis (see for example: Dunitz & White, 1973; Stuart & Phillips, 1985). An example of such motion could be a libration of a flexible amino-acid side chain around a bond vector (such as the C_α-C_β torsion bond. Currently this approach is being implemented in `phenix.refine`.

3.3. U_{TLS}

If the TLS model is used for the rigid-body motion of a group of atoms then

$$\mathbf{U}_{TLS} = \mathbf{T} + \mathbf{A}\mathbf{L}\mathbf{A}^t + \mathbf{A}\mathbf{S} + \mathbf{S}^t\mathbf{A}^t \quad (3)$$

with 20 refinable **T** (translation), **L** (libration) and **S** (screw-rotation) matrix elements per group (Schomaker & Trueblood; 1968).

The choice of TLS groups is often subjective and may be based on visual inspection of the molecule in an attempt to identify distinct and potentially independent fragments. A more rigorous approach is implemented in the TLSMD algorithm (Painter & Merritt, 2006a, 2006b)). The TLSMD algorithm identifies TLS groups by splitting a whole molecule into smaller pieces followed by fitting of TLS parameters to the previously refined atomic *B*-factors for each piece. Therefore, it is very important that the input ADP values for the TLSMD procedure are minimally biased by the restraints used in previous refinements, and meaningful in general (not reset to an arbitrary constant value, for example). Ideally, one may need to perform a round of unrestrained ADP refinement just for the purpose of TLS group identification with TLSMD. Since an unrestrained refinement may not be stable (it might produce non-physical values or a large spread of refined *B*-factors, especially at lower resolution), it might be better to perform a group *B*-factor refinement only, using one or two refinable parameters per residue. Such a refinement will not make use of any ADP restraints and therefore yields refined *B*-factors that attempt to fit the data most closely within the limits of the group definition used. Futhermore, currently the TLSMD procedure does not generate a unique definitive answer for the TLS group selection but rather gives a list of possible choices and the researcher has to make the decision to test one or more TLS group selections; this still leads to an element of subjectivity in the definition of TLS groups.

3.4. U_{LOCAL}

$\mathbf{U}_{LOCAL}$ models harmonic atomic vibrations occurring around the mean atomic position. Depending on the experimental data quality, this can be a simple isotropic model where the atomic vibrations have equal amplitude in all directions, or it can be more complex where atomic vibration is assumed to be anisotropic. Ideally, if all the group (or global) contributions to the total atomic displacement are subtracted, leaving only the local atomic vibration $\mathbf{U}_{LOCAL}$, then it should obey Hirshfeld's rigid bond postulate (Hirshfeld, 1976) providing a basis for the use of similarity

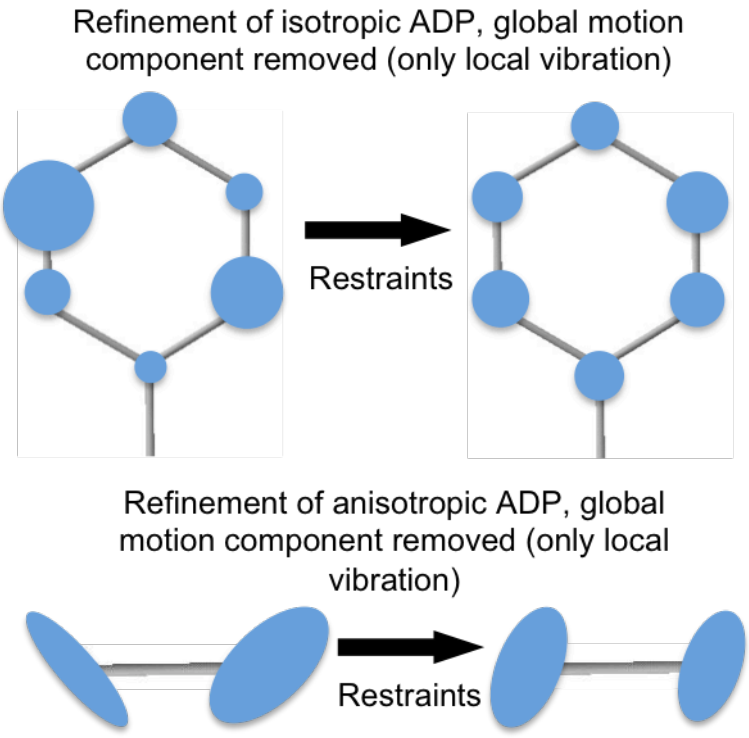


Figure 2. Illustration of similarity restraints.

restraints (Fig. 2). For isotropic ADPs:

$$T_{\text{adp}} = \sum_{\substack{\text{pairs of}\\ \text{bonded}\\ \text{atoms (i,j)}}} \left(U_{\text{local},i} - U_{\text{local},j}\right)^2 \qquad (4)$$

and for anisotropic ADPs:

$$T_{\text{adp}} = \sum_{\substack{\text{pairs of}\\ \text{bonded}\\ \text{atoms (i,j)}}} \sum_{k=1}^{6} \left(U^k_{\text{local},i} - U^k_{\text{local},j}\right)^2 \qquad (5)$$

which is similar to (4) where the inner sum spans over all six ADP matrix elements. Despite its simplicity this formula has proved useful in many cases. However, more sophisticated approaches have been suggested in order to better handle a number of special cases for which (5) is suboptimal (Hendrickson, 1985; Schneider, 1996; Sheldrick & Schneider, 1997). Since `phenix.refine` allows for a mixture of atoms with isotropic or anisotropic ADPs, it may happen (at least theoretically) that an isotropic atom is bonded to one having an anisotropic ADP. Currently, in this case only the isotropic component of each of the two atoms is participating in the restraint.

4. Practice: mixing contributions into U_{TOTAL}, restraints

4.1. $U_{TOTAL} = U_{CRYST} + U_{LOCAL}$

In practice, it is still customary to have one isotropic refinable parameter per atom at typical 'macromolecular' resolutions (~1.7-3.0Å), that is $\mathbf{U}_{TOTAL} = \mathbf{U}_{CRYST} + \mathbf{U}_{LOCAL}$, where the $\mathbf{U}_{GROUP}$ contribution is accumulated into other atomic parameters, including $\mathbf{U}_{CRYST}$ and $\mathbf{U}_{LOCAL}$.

Since in this case $\mathbf{U}_{LOCAL}$ contains other contributions to displacements ($\mathbf{U}_{GROUP}$) this in turn invalidates the use of restraints (4-5) (see Fig. 3 for an illustration). Although Fig. 3 contains some elements of dramatization and the actual deviations from similarity may not be as large as shown (especially for covalently bonded atoms), still in this case forcing the *B*-factors to be nearly identical is less valid. A possible work-around is to use a knowledge-based type of restraint introduced by Tronrud (1996). However, in `phenix.refine` we have implemented a 'softer' algorithm for similarity restraints, which is based on the following assumptions:

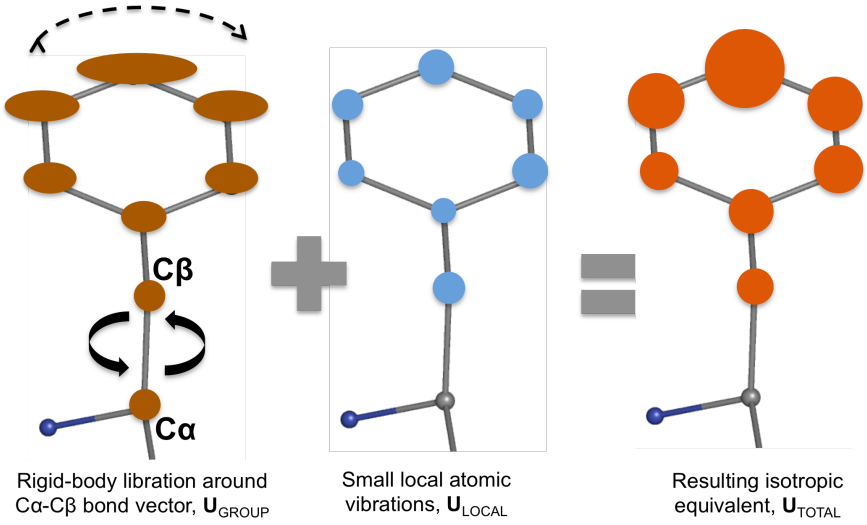


Figure 3. While local atomic vibrations ($\mathbf{U}_{LOCAL}$ obey similarity restraints, the total ADP ($\mathbf{U}_{TOTAL}$) may not obey similarity restrains because the contribution ($\mathbf{U}_{GROUP}$) arising from rigid-body motion of a whole fragment may add different displacements to each atom.

- A bond is almost rigid, therefore the ADPs of bonded atoms ($\mathbf{U}_{LOCAL}$ component) are similar (Hirshfeld, 1976);
- ADP values of spatially close (including non-bonded) atoms are similar (Schneider, 1996);

- The difference between the ADP values of atoms close in space is related to the absolute values of the ADP values. Atoms with higher ADP values are allowed larger differences (Ian Tickle, CCP4 Bulletin Board, letter from March 14, 2003).

Considering the above assumptions, we obtain

$$T_{\text{adp}} = \sum_{i=1}^{N_{\text{atoms}}} \left[\sum_{j=1}^{M_{\text{atoms}}} \frac{1}{r_{ij}^{p}} \frac{\left(U_{\text{local},i} - U_{\text{local},j}\right)^2}{\left(U_{\text{local},i} + U_{\text{local},j}\right)^{q}} \right] \qquad (6)$$

Here N_{atoms} is the total number of atoms in the model, the inner sum is extended over all M_{atoms} in the sphere of radius R around atom i, r_{ij} is the distance between two atoms i and j, $U_{\text{local},i}$ and $U_{\text{local},j}$ are the corresponding isotropic ADP values, p and q are empirical constants. By default, R, p and q are fixed at empirically derived values: 5.0Å, 1.69 and 1.03, respectively, but they can also be changed by the user. The function reduces to formula (4) if $p = q = 0$, and the radius R is set to be approximately equal to the upper limit of a typical bond length.

At high enough resolution, approximately better than 1.6-1.7Å, $\mathbf{U}_{\text{GROUP}} = \mathbf{U}_{\text{TLS}}$ is not used (currently is not implemented in `phenix.refine`) leaving $\mathbf{U}_{\text{TOTAL}} = \mathbf{U}_{\text{CRYST}} + \mathbf{U}_{\text{LOCAL}}$ where the $\mathbf{U}_{\text{LOCAL}}$ can be anisotropic.

4.2. $U_{TOTAL} = U_{CRYST} + U_{GROUP} + U_{LOCAL}$

This is the most advanced formulation of ADP parameterization available in `phenix.refine`. In this case currently $\mathbf{U}_{\text{LOCAL}}$ can only be refined isotopically for those atoms that participate in a TLS group. The NCS restraints (if available and used) as well as (4) are applied to $\mathbf{U}_{\text{LOCAL}}$ only and not to the whole ADP, $\mathbf{U}_{\text{TOTAL}}$.

`phenix.refine` allows any combination of the above ADP refinement strategies to be applied to any selected part of the structure. The only exception (which is a technical limitation and might be changed in the future) is that an atom cannot be in a TLS group and simultaneously have an anisotropic $\mathbf{U}_{\text{LOCAL}}$.

It is important to note that the positive definiteness of $\mathbf{U}_{\text{TOTAL}}$ is ensured at all times, while the individual components may or may not be positive definite (except $\mathbf{U}_{\text{CRYST}}$, see below). For example, in combined TLS and individual ADP refinement ($\mathbf{U}_{\text{TOTAL}} = \mathbf{U}_{\text{CRYST}} + \mathbf{U}_{\text{GROUP}} + \mathbf{U}_{\text{LOCAL}}$), $\mathbf{U}_{\text{LOCAL}}$ are not forced to be non-zero or even positive and $\mathbf{U}_{\text{TLS}}$ are not forced to be positive definite either, while $\mathbf{U}_{\text{TOTAL}}$ is assured to be positive definite (by using eigenvalue filtering). The benefit of such an approach is that it can compensate for non-optimal choices of TLS groups (discussed above) by a corresponding adjustment of $\mathbf{U}_{\text{LOCAL}}$ (which in *such cases* becomes a compensation factor and has little physical meaning), while the resulting overall *B*-factor, $\mathbf{U}_{\text{TOTAL}}$, is still positive definite. $\mathbf{U}_{\text{CRYST}}$ is an exception, and the physical correctness and compatibility with the crystal symmetry are enforced at

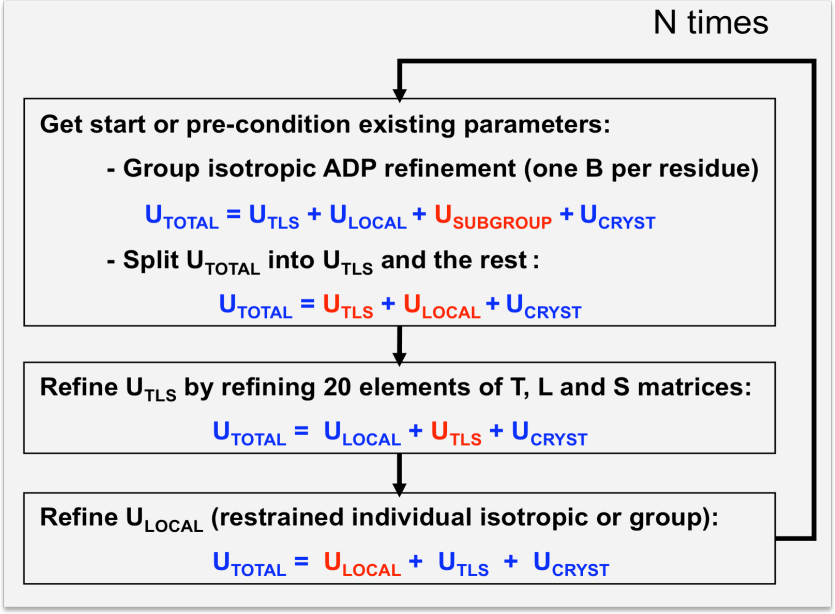


Figure 4. Protocol of combined TLS and individual ADP refinement.

the bulk-solvent correction and anisotropic scaling step.

The contributions $\mathbf{U}_{CRYST}$, $\mathbf{U}_{GROUP}$ and $\mathbf{U}_{LOCAL}$ in (1) are highly correlated making it generally impractical to separate them. In addition, when refined simultaneously the elements of the **T**, **L** and **S** matrices appear to be correlated as well. This can generate a number of numerical issues, such as: refinement of TLS matrices to non-physical values, refinement becoming stalled, and strong dependence of the final refined values on the starting values. To overcome these issues we developed a complex algorithm for efficient refinement of the terms in $\mathbf{U}_{TOTAL}$; the details are outlined at Fig. 4. The procedure begins with obtaining sensible initial values for TLS parameters and $\mathbf{U}_{LOCAL}$. This is performed in three stages: 1) pre-conditioning of *B*-factors by performing a preliminary group ADP refinement, 2) extracting **T** matrix from these refined values as described by Afonine & Urzhumtsev (2007) and 3) least-squares fit of the **T**, **L** and **S** matrices to $\mathbf{U}_{TOTAL}$ starting with the **T** matrix obtained from the previous step and zero or previously refined **L** and **S** matrices. The fitted TLS matrices are now the starting point for the TLS refinement against crystallographic data in the second step. Finally, the individual *B*-factors are refined keeping all other components fixed. This is repeated several times (3 by default).

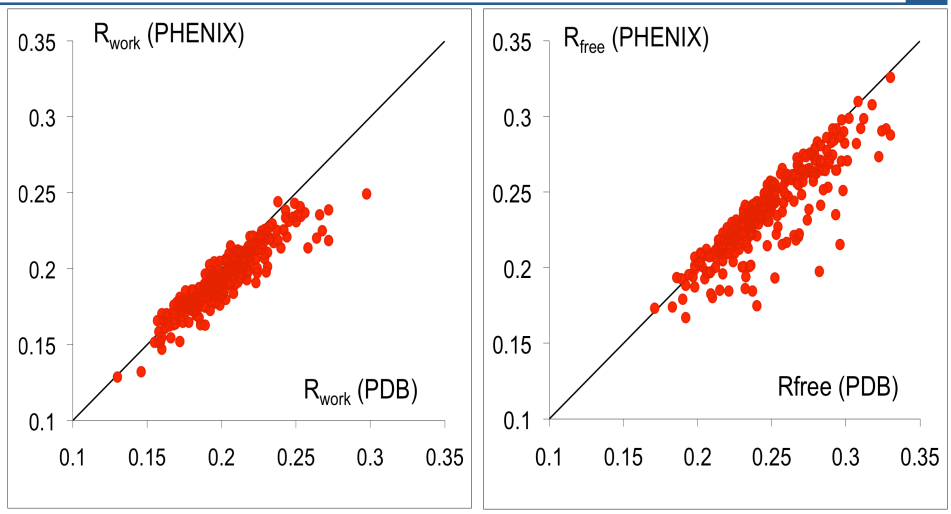


Figure 5. Result of automatic re-refinement of 355 TLS containing structures in PDB.

This combined ADP refinement protocol was tested on 355 structures from PDB that contained TLS records. The structures were selected using the strict criteria: correct TLS selections and *R*-factors reproducible within 0.5%. For the test purposes the original *B*-factors in these structures were all re-set to the model average value. The re-refinement of ADP values using the protocol described above resulted in comparable or better (than reported) final *R*-factors (Fig. 5). There was no case where refinement showed any numerical problems.

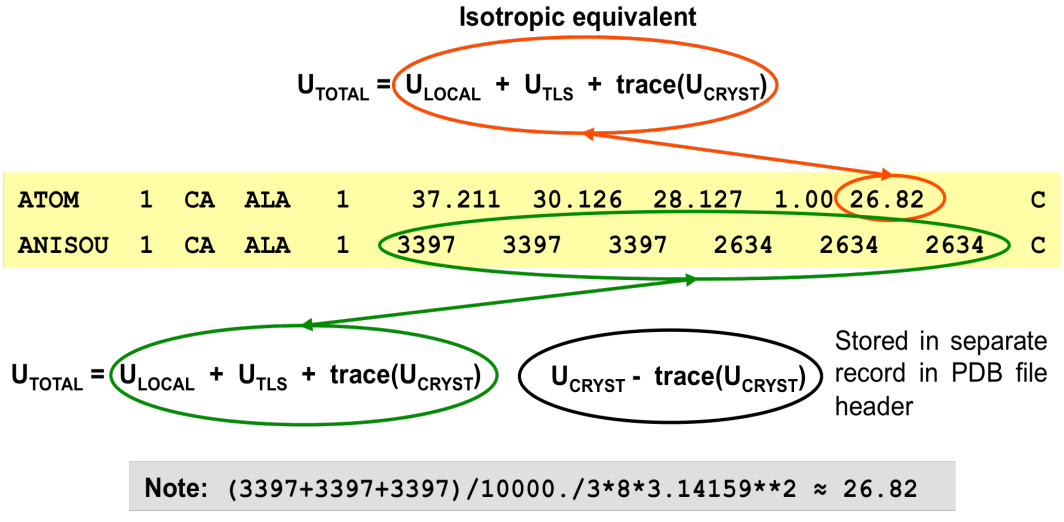


Figure 6. A scheme showing how the refined ADPs are stored in a PDB file.

5. Output of ADP information in PDB files

It should be noted that each atom participating in a TLS group receives an anisotropic ADP (ANISOU card in PDB) (Fig. 6). `phenix.refine` always outputs the total ADP for each atom to the PDB file. An exception is made for $\mathbf{U}_{\mathrm{CRYST}}$, where, similarly to CNS (Brunger, 2007), only the trace of this matrix (or the component that keeps the ADP values positive definite) is added into the output ADP for ATOM and ANISOU records, while its anisotropic component is separated and is reported in the PDB file header. To analyze the separate contributions $\mathbf{U}_{\mathrm{TLS}}$ and $\mathbf{U}_{\mathrm{LOCAL}}$ given $\mathbf{U}_{\mathrm{TOTAL}}$, `phenix.tls` (Afonine, unpublished) is a tool within *PHENIX* that is specifically designed for this.

6. Examples

In this section we illustrate the use of refinement strategies with different choices of ADP parameterization.

All refinements included five macro-cycles of individual coordinates, ADP and occupancy refinement (see the `phenix.refine` documentation for details of occupancy refinement), ordered solvent (water) update (adding, removing, refinement; data resolution permitting). If NCS is available, then both `phenix.refine` runs were performed: with and without using NCS, where the NCS groups were selected by `phenix.refine` automatically. The X-ray target weight was automatically optimized in all refinement runs. The only variation between refinement runs was the ADP parameterization:

- individual (I),
- group with one isotropic ADP per residue (G1),
- group with two isotropic ADP per residue (G2; one per main and one per side chain atoms);
- TLS only (T);
- TLS in combination with G1 (T+G1);
- TLS in combination with G2 (T+G2);
- TLS in combination with I (T+I).

In the examples below the best re-refinement result achieved in `phenix.refine` is shown in bold in the corresponding table.

Example 1: PDB code 1BL8, resolution 3.2Å

This structure was originally refined to R_{work} = 28.0 and R_{free} = 29.0% (Doyle *et al.*, 1998). Chen *et al.* (2007) re-refined this model using a Normal Mode parameterization in refinement, reducing the *R*-factors to 27.2 and 27.2%, respectively (leaving no gap between R_{work} and R_{free}). At this resolution automated water model updates were not possible.

R_{work} / R_{free} (%)	Refinement strategy							
	I	I+NCS	G1+NCS	G2+NCS	T+NCS	T+G1+NCS	T+G2+NCS	**T+I+NCS**
	25.6/32.0	24.7/28.5	27.0/30.0	26.4/30.8	25.0/28.2	24.8/28.0	24.0/28.1	**23.3/27.0**

Example 2: PDB code 2PFD, resolution 3.42Å

This structure was originally refined to R_{work} = 24.0 and R_{free} = 24.9% using a Normal Mode parameterization (Poon *et al.*, 2007). At this resolution automated water model updates were not possible.

R_{work} / R_{free} (%)	Refinement strategy							
	I	I+NCS	G1+NCS	G2+NCS	T+NCS	T+G1+NCS	T+G2+NCS	**T+I+NCS**
	21.6/28.1	20.3/26.5	21.5/28.9	22.1/29.0	21.2/27.5	21.0/27.6	21.0/28.1	**20.2/25.8**

Example 3: PDB code 1DQV, resolution 3.2Å

This structure was originally refined to R_{work} = 29.3 and R_{free} = 34.8% using the CNS program (Sutton *et al*., 1999). At this resolution automated water model updates were not possible.

R_{work} / R_{free} (%)	Refinement strategy							
	I	I+NCS	G1	G2	T	T+G1	T+G2	**T+I**
	21.6/27.9	-	25.3/30.5	24.4/29.8	22.6/27.3	21.9/26.9	21.2/27.3	**19.7/25.3**

Example 4: PDB code 1B7G, resolution 2.05Å

The PDB file header indicates R_{work} = 22.6 and R_{free} = 29.3% (Isupov et al., 1999). Winn *et al*. (2001) applied combined TLS and individual isotropic refinement which resulted in R_{work} = 22.0 and R_{free} = 25.7%.

R_{work} / R_{free} (%)	Refinement strategy						
	I	G1	G2	T	T+G1	T+G2	**T+I**
	21.4/25.2	23.1/25.8	22.9/26.4	19.7/22.5	19.1/22.5	18.9/22.0	**17.3/21.2**
	+NCS						
	21.9/25.7	22.3/26.4	23.7/26.4	20.5/22.1	20.5/22.7	19.6/21.9	17.7/21.3
	No water update, no NCS						
	20.5/26.0	22.1/26.8	22.0/27.8	19.5/24.0	18.4/22.3	19.2/24.1	16.7/21.7

Acknowledgements

This work was supported in part by the US Department of Energy under Contract No. DE-AC03-76SF00098 and NIH/NIGMS grant 1P01GM063210.

References

Afonine, P.V., Grosse-Kunstleve, R.W. & Adams, P.D. (2005). *CCP4 Newsl*. **42**, contribution 8.

Brunger, A.T. (2007). *Nature Protocols*. **2**, 2728-2733.

Chen, X., Poon, B.K., Dousis, A., Wang, Q., & Ma, J. (2007). *Structure*. **15**, 955-962.

Doyle, D.A, Cabral, J.M., Pfuetzner, R.A., Kuo, A., Gulbis, J.M., Cohen, S.L., Chait, B.T. & MacKinnon, R. (1998). *Science*. **280**, 69-77.

Dunitz, J.D. & White, D.N.J. (1973). *Acta Cryst.* A**29**, 93-94.

Isupov, M. N., Fleming, T. M., Dalby, A. R., Crowhurst, G. S., Bourne, P. C. & Littlechild, J. A. (1999). *J. Mol. Biol.* **291**, 651-660.

Ni, F., Poon, B.K., Wang, Q. & Ma, J. (2009). *Acta Cryst*. (2009). D**65**, 633-643.

Painter, J. & Merritt, E.A. (2006). *Acta Cryst*. D**62**, 439-450.

Painter, J. & Merritt, E.A. (2006). *J. Appl. Cryst*. **39**, 109-111.

Poon, B.K., Chen, X., Lu, M., Vyas, N.K., Quiocho, F.A., Wang, Q., & Ma, J. (2007). *PNAS*. **104**, 7869-7874.

Schomaker, V. & Trueblood, K.N. *Acta Cryst.* (1968). B**24**, 63-76.

Stuart, D. I. & Phillips, D.C. (1985). *Methods in Enzymology*. **115**, 117-142.

Sutton, R.B., James A. Ernst, J.A. & Brunger, A.T. (1999). *J. Cell Biol.* **147**: 589-598.

Non-periodic torsion angle targets in *PHENIX*

Jeffrey J. Headd,[a] Nigel W. Moriarty,[a] Ralf W. Grosse-Kunstleve,[a] and 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: JJHeadd@LBL.Gov

Torsion restraints in the *PHENIX* monomer library have traditionally been parameterized with a target angle (`value_angle`), a standard deviation (`value_angle_esd`), and a periodicity (`period`) as originally specified by Vagin et al. (2004). An example mmCIF definition for the χ angles in Trp residues is shown in Fig. 1A. This periodic parameterization for χ angles proves to be insufficient to recapitulate all favored rotamer positions (Lovell et al., 2000) for Asn, Asp, His, Phe, Trp, and Tyr. For example, Fig. 2A depicts the results of geometry minimization in *PHENIX* of an ideal Tyr **m0** rotamer, which results in an **m-90** rotamer. This result occurs because the χ_2 torsion for Tyr is defined as 90° with a periodicity of 2, which excludes the 0° target. In most cases, simply increasing the periodicity of a given torsion angle is counterproductive as this will open up non-rotameric conformations in addition to the desired values.

A.

```
_chem_comp_tor.comp_id
_chem_comp_tor.id
_chem_comp_tor.atom_id_1
_chem_comp_tor.atom_id_2
_chem_comp_tor.atom_id_3
_chem_comp_tor.atom_id_4
_chem_comp_tor.value_angle
_chem_comp_tor.value_angle_esd
_chem_comp_tor.period
 TRP chi1 N  CA CB CG  180.000 15.000 3
 TRP chi2 CA CB CG CD1  90.000 20.000 2
```

B.

```
_chem_comp_tor.comp_id
_chem_comp_tor.id
_chem_comp_tor.atom_id_1
_chem_comp_tor.atom_id_2
_chem_comp_tor.atom_id_3
_chem_comp_tor.atom_id_4
_chem_comp_tor.value_angle
_chem_comp_tor.alt_value_angle
_chem_comp_tor.value_angle_esd
_chem_comp_tor.period
TRP chi1 N  CA CB CG 180.000 . 15.000 3
TRP chi2 CA CB CG CD1 90.000 0 20.000 2
```

Figure 1. Examples of the torsions section of a restraints file in CIF format for TRP.

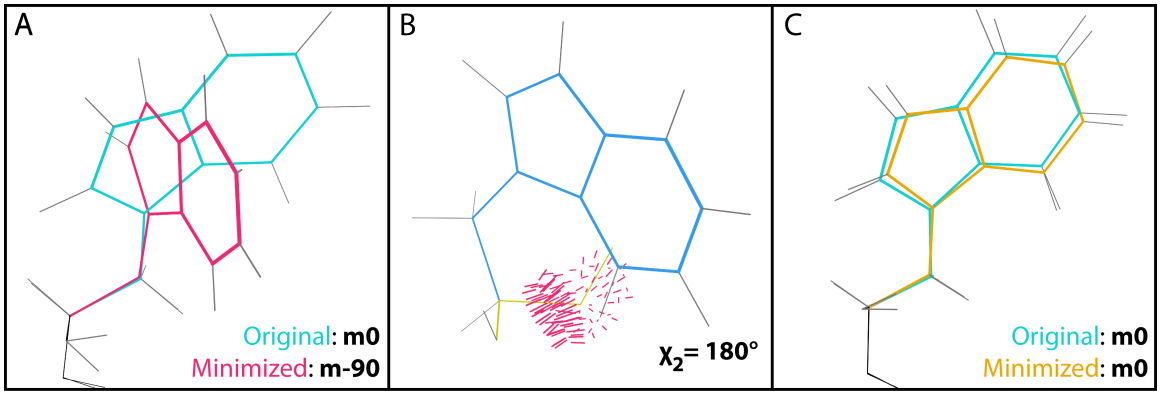


Figure 2. Geometry of various rotomers for TRP. Images generated using KiNG (Chen et al, 2009).

As shown in Fig. 2B, increasing the periodicity from 2 to 4 to allow χ_2 = 0° for Tyr opens up χ_2 = 180°, which would allow a rotamer outlier with an impossible steric clash to score favorably in the geometry term.

To recapitulate all favorable rotamer positions without introducing false-minima, PHENIX now includes an `alt_angle_value` parameter in the `tor` definitions for non-periodic parameterization of torsion angles. Fig. 1B illustrates the usage of the `alt_angle_value` parameter in the mmCIF parameterization for Tyr. This parameter takes either a single target value, or multiple target values that are comma-separated. The current `value_angle_esd` is used to apply the same ESD to the new torsion definitions as is used for the original periodic `value_angle` definition. ESD flexibility for the alternate parameter will be implemented in the future. Both `value_angle` and `alt_angle_value` may be used together for flexible implementation of torsion definitions. To allow for mixing of `tor` definitions with and withot `alt_angle_ideal` values, a '.' may be used for the `alt_angle_ideal` parameter for instances where it should be ignored by *PHENIX*.

All previously inaccessible rotamers for the above mentioned residues have been corrected in the geostd (http://geostd.sourceforge.net) in PHENIX using alt_angle_value parameters. Fig. 2C illustrates the proper recapitulation of the Tyr rotamer m0 following geometry minimization using the new parameterization. Tyr χ1 as shown in Fig. 1B demonstrates such an entry in practice.

These non-periodic torsion angle definitions may be included in any mmCIF definition for use in PHENIX, including amino acids, nucleotides, and ligands. One potential use for these parameters is for specifying sugar puckers in saccharides. Internally, eLBOW (Moriarty et al., 2009) currently uses non-periodic torsions to generate and control the puckers of saturated rings. The values of the dihedrals of a six-membered puckered ring in the chair conformer are approximately ±55 degrees. A torsion definition using periods would require a value of 60 degrees and a period of three. This is not an ideal situation for either the ideal value or the fact that 180 degrees is a minimum on the potential surface. Furthermore, the boat conformer has two torsions that are approximately zero requiring a period of six. This makes 120 degrees a possible but non-physical minimum. Testing of the non-periodic torsions for carbohydrates is currently underway and will soon be available via eLBOW.

References:

Chen, V. B., Davis, I. W. and Richardson, D. C. (2009) KiNG (Kinemage, Next Generation): A versatile interactive molecular and scientific visualization program. *Protein Science*, **18**, 2403-2409.

Lovell, S. C., Word, J. M., Richardson, J. S. and Richardson, D. C. (2000) The Penultimate Rotamer Library. *Proteins: Struct Function and Genetics*, **40**, 389-408.

Moriarty, N. W., Grosse-Kunstleve, R. W. & Adams, P. D. (2009). Acta Cryst. **D65**, 1074–1080.

Vagin, A. A., Steiner, R. A., Lebedev, A. A., Potterton, L., McNicholas, S., Long, F. and Murshudov, G. N. (2004) REFMAC5 dictionary: organization of prior chemical knowledge and guidelines for its use. *Acta Cryst.* **D60**, 2184-2195.

Model-building updates and new features

Tom Terwilliger[a]

[a]*Los Alamos National Laboratory, Los Alamos, NM 87545*

Correspondence email: terwilliger@LANL.Gov

A. Rapid phase improvement and model-building with phenix.phase_and_build (NEW)

PHENIX now has a new and very rapid method for improving the quality of your map and building a model. This `phenix.phase_and_build` approach uses all the tools described in this section. The approach is to carry out an iterative process of building a model as rapidly as possible and using this model in density modification to improve the map. This approach is related to the older `phenix.autobuild` approach. The difference is that in `phenix.autobuild` much effort was spent on building the best possible model at each stage before carrying out density modification, while in `phenix.phase_and_build` speed of model-building is optimized. The result is that `phenix.phase_and_build` is 10 times faster than `phenix.autobuild`, yet it produces nearly as good a model in the end. The `phenix.phase_and_build` approach will also find NCS from your starting map and apply it during density modification. You can run `phenix.phase_and_build` with:

```
phenix.phase_and_build my_experimental_data.mtz my_sequence.dat
```

You can also add a starting model or a starting map. This means that you can run it once, get a new model and map, then run it again to improve your model and map further.

When you run `phenix.phase_and_build` it will write out a `phase_and_build_params.eff` parameter file that can be used to re-run `phenix.phase_and_build` (just as for essentially all *PHENIX* methods). In addition, `phenix.phase_and_build` will write out the parameters files for the intermediate methods used as part of `phenix.phase_and_build` to the temporary directory used in building. You can

- Run NCS identification

  ```
  phenix.find_ncs temp_dir/find_ncs_params.eff
  ```

- Run first cycle of density modification

  ```
  phenix.autobuild temp_dir/AutoBuild_run_1_/autobuild.eff
  ```

- Run most recent model-building

  ```
  phenix.build_one_model temp_dir/build_one_model_params.eff
  ```

- Run sequence assignment and filling short gaps

  ```
  phenix.assign_sequence temp_dir/assign_sequence_params.eff
  ```

- Run loop fitting

  ```
  phenix.fit_loops temp_dir/fit_loops_params.eff
  ```

This gives you control of all the steps in map improvement and model-building in addition to letting you run them all together with `phenix.phase_and_build`.

The `phenix.autosol` wizard now uses `phenix.phase_and_build` by default for model-building. This means that now the models produced by `phenix.autosol` are quite good but are still obtained quickly.

B. Working with NCS in PHENIX with phenix.find_ncs: Identification of NCS from heavy-atom sites, from a model or from a map

The `find_ncs` method contains all the algorithms available in *PHENIX* for finding NCS, including a new algorithm for finding NCS directly from a map. The approaches used in `find_ncs` are:

1. Finding NCS from a map (NEW):

```
phenix.find_ncs my_map.mtz
```

The `find_ncs_from_density` algorithm first identifies potential locations of centers of macromolecules in the density map by finding maxima of the local RMS density. Then it cuts out a sphere of density centered at a trial center and carries out an FFT-based rotation-translation search to find all occurrences of similar density in the asymmetric unit of your map. The region over which NCS-related correlation is high is identified and the operators are written out as a "`find_ncs.ncs_spec`" file that can be read by the *PHENIX* wizards and a "`find_ncs.phenix_refine`" file that can be read by `phenix.refine`. You can run the algorithm as a whole or you can run each part separately with `phenix.guess_molecular_centers` and `phenix.find_ncs_from_density`.

2. Finding NCS from heavy-atom sites or a model

```
phenix.find_ncs ha.pdb my_map.mtz

phenix.find_ncs my_model.pdb
```

If you have a heavy-atom pdb file and a map file, then `phenix.find_ncs` will identify subsets of your heavy-atom sites that are related by non-crystallographic symmetry, and it will check whether this NCS is actually reflected in your map. It will write out the resulting NCS as `ncs_spec` and `phenix_refine` files. If you have a model with several chains that are related by NCS symmetry, `phenix.find_ncs` will find the NCS operators from the coordinates in your model.

3. Reading NCS from a my_ncs.ncs_spec file

```
phenix.find_ncs my_ncs.ncs_spec
```

will read a `ncs_spec` file written by a *PHENIX* method, and write out the NCS in formats suitable for `phenix.refine` or the wizards. If you supply also a map MTZ file, it will check for this NCS in your map.

4. Creating NCS-related copies

You can also apply NCS operators from a `my_ncs.ncs_spec` file to a single copy of your protein to create all the NCS-related copies with:

```
phenix.apply_ncs my_ncs.ncs_spec my_model_one_ncs_copy.pdb
```

C. Fitting loops with loop libraries (NEW) and by tracing chains with phenix.fit_loops

```
phenix.fit_loops my_model_with_gaps.pdb my_map.mtz my_sequence.dat
```

The `phenix.fit_loops` approach now has two main algorithms. One is to fit short gaps using a loop library derived from high-resolution structures in the PDB, and the other is to build loops directly by iterative extension with tripeptides. The loop-library approach (specified with `loop_lib=True`) is very fast, and is currently applicable for short gaps of up to 3 residues. The iterative extension approach is slower, but can be used for longer gaps (typically up to 10-15 residues).

D. Rapid building of a single model with phenix.build_one_model (NEW)

PHENIX now has a tool that you can use to quickly build a single model from a map and sequence file, or to extend an existing model. You can build a new model with resolve model-building with:

```
phenix.build_one_model my_map.mtz my_sequence.dat
```

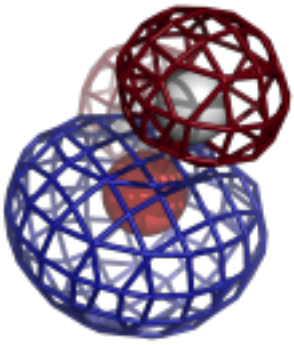
If you supply a PDB file, then the ends of each chain in your model will be extended, if possible, based on your map.

E. Sequence assignment and short gap filling with phenix.assign_sequence (NEW)

You can now carry out an improved sequence assignment of a model that you have already built with `phenix.assign_sequence`. Further, once the sequence has been assigned, this method will use the sequence and proximity to identify chains that should be connected, and it will connect those that have the appropriate relationships using the new loop libraries available in `phenix.fit_loops`. The result is that you may be able to obtain a more complete model with more chains assigned to sequence than previously. You can run it with:

```
phenix.assign_sequence my_model.pdb my_sequence.dat my_map.mtz
```

VOLUME TWO



COMPUTATIONAL CRYSTALLOGRAPHY NEWSLETTER

JANUARY MMXI

LABELIT SYMMETRY KING ROSETTA TLS TWINNING

Table of Contents

Editor

Nigel W. Moriarty, NWMoriarty@LBL.Gov

Contributors

P. D. Adams, P. V. Afonine, D. Baker, L. J. Bourhis, G. Bunkóczi, V. B. Chen, F. DiMaio, N. Echols, J. J. Headd, R. W. Grosse-Kunstleve, V. Y. Lunin, N. W. Moriarty, R. J. Read, D. C. Richardson, J. S. Richardson, N. K. Sauter, T. C. Terwilliger, A. Urzhumtsev, C. Williams

PHENIX News

New releases

A new tool for automated partitioning a model into TLS groups, `phenix.find_tls_groups`, is now available. This tool is available in the GUI and command-line interfaces and can take advantage of additional available CPU to generate the atom selection for a refinement run. The automatically defined TLS groups can be readily visualised and edited in the GUI. This tool and all others mentioned here are available in *PHENIX* version 1.7.

Visualisation of mutil-criteria kinemage graphics is now available in *PHENIX* and is discussed in the short communications on page 6.

Generation of ensembles for Molecular Replacement (MR) is the goal of new release called `phenix.ensembler`. Another new release integrates MR and Rosetta in *PHENIX*. For more details, see the short communications for `phenix.ensembler` on page 8 and `phenix.mr_rosetta` on page 10.

New features

Reference model restraints

Reference model restraints are used to steer refinement in cases where the working data

The Computational Crystallography Newsletter (CCN) is a regularly distributed electronically via email and the PHENIX website, www.phenix-online.org/newsletter. Feature articles, meeting announcements and reports, information on research or other items of interest to computational crystallographers or crystallographic software users can be submitted to the editor at any time for consideration. Submission of text by email or word-processing files using the CCN templates is requested.

set is low resolution, but there is a known related structure solved at higher resolution. The higher resolution reference model is used to generate a set of dihedral restraints that are applied to each matching dihedral in the working model. Sequence alignment is handled automatically in cases of sequence dissimilarity, including handling for deletions and insertions. Alternatively, selections may be used to hand-specify the desired reference group in a parameter file. To use (also available in GUI) add the following to the input.

```
main.reference_model_restraints=True
reference_model.file=my_reference.pdb
```

To specify reference group(s),

```
refinement.reference_model.reference_group
{
reference = chain A and resseq 130:134
selection = chain B and resseq 120:124
}
```

For a full list of reference model options, please see the `phenix.refine` documentation.

Augmented base-pairing restraints for RNA

RNA base-pairing restraints have been augmented to include many non-Watson-Crick pairings, including all 28 Saenger types. Saenger nomenclature is now used by default to specify applicable base-pairs. Base-pairs are determined automatically from the input model, but may also be specified by hand in a parameter file. To specify a base-pair by type in a parameter file:

```
nucleic_acids {
  base_pair {
    base1 = "chain \"A\" and resseq 54"
    base2 = "chain \"A\" and resseq 72"
    saenger_class = "XIX"
  }
}
```

For a full list of secondary structure options, please see the on-line documentation for `phenix.refine`.

Crystallographic meetings and workshops

20th West Coast Protein Crystallography Workshop, 20-23 March, 2011

The bi-annual WCPCW is being held at a new location, Monterey Plaza Hotel in Monterey, CA on the 20th to the 23rd of March. *PHENIX* developers will be in attendance.

RapiData, 3-8 April, 2011

RapiData, the annual data collection and structure-solving course is being held from the 3rd to the 8th of April. *PHENIX* developers will be in attendance.

International Conference on Structural Genomics May 10-14, 2011

The International Structural Genomics Organisation (ISGO) is holding a conference in Toronto, Canada from the 10th to the 14th of May. The "*PHENIX* Crystallography Software Workshop" is an all day event on the first day.

American Crystallographic Association, 28 May – 2 June, 2011

A workshop, entitled "Introduction to *PHENIX* for beginning to advanced crystallographers" is planned for the 2011 Meeting of the American Crystallographic Association in New Orleans, Louisiana. The workshop is being held on the 28th May with further information available at www.amercrystalassn.org.

XXII Congress and General Assembly of the International Union of Crystallography (IUCr), 22-30 August 2011

The 22nd Congress and General Assembly of the IUCr will be held in Madrid, Spain on the 22nd to the 30th of August.

PHENIX User's Workshop, 17 March, 2011

A *PHENIX* user's workshop is being planned in Berkeley, California on the 17th of March for local area students, postdocs and other interested parties. Please contact Nat Echols at NEchols@lbl.gov for further information.

Expert advice

Fitting Tips

Vincent Chen, Christopher Williams and Jane Richardson, Duke University

Now that model building is highly automated, for example in *PHENIX*, crystallographers get much less experience with the fun of fitting good maps and perhaps could use tips about what to look for in the difficult cases where the automated methods don't suffice requiring that they do it themselves. In this series, we will try to pass along our own hard-earned rules of thumb. Most of these are what we call "systematic errors", because they happen repeatedly in a similar pattern caused by a misleading appearance in the electron density or a misleading assumption about what should be true in the model. That means they occur fairly often in certain circumstances, but fortunately there is usually a recipe for what needs to be done to fix them.

Our initial fitting tip is one especially relevant at low resolution (around 3Å and worse), first noticed recently when helping to rebuild ribosome structures (in collaboration with Jamie Cate). We encountered places where an extended β strand, instead of the usual alternation of peptide direction, showed three carbonyls in a row that all pointed the same general direction. Figure 1 shows an example of a three-stranded antiparallel β sheet with two such problems (left panel, with the CO triples labeled in red, the all-atom clashes as pink spikes and Ramachandran outliers as green lines). This prevents good H-bonding, produces clashes, often places a sidechain on the wrong side of the sheet and, usually, leads to Ramachandran outliers.

The misleading feature that causes this mistake is that in the 2.5 to 3Å resolution range the carbonyls start to lose definition in the map and a strand becomes a rather smooth, round tube. Therefore the electron density no longer provides the clues to peptide orientation that both people and software rely on at better resolution.

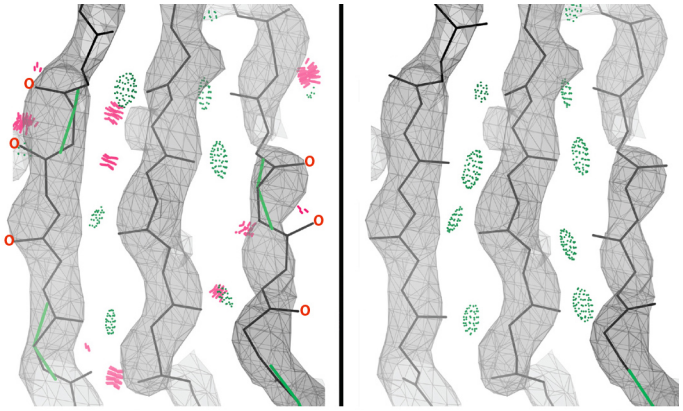


Figure 1: Model on left has a misfit β strand containing three consecutive parallel carbonyls and the corrected model on the right.

The fixing procedure begins with a near-180° flip of the peptide containing the central of the three COs, followed by geometry regularization, refitting of the affected sidechain and optimization of backbone β-sheet H-bonding. The result, as in the right panel of figure 1, should correct the clashes, Ramachandran outliers, improve H-bonding (pillows of green dots) and fit the density at least a bit better.

An alpha-test version of an automated diagnostic for these "three CO" misfit cases identified hundreds of examples in low-resolution PDB structures. See and try out the illustrated case in PDB file `3I1N` (or `3i1n` in lower case for clarity) chain G (L6), residues 95-113. However, note that in future structures for the E coli 70S ribosome many such issues will have been corrected.

FAQ

There are two methods to add links between atoms in *PHENIX*. Which one should I use?

Both methods have their place. Either can be used in most situations, however, one is designed for adding one or two links with little setup while the other is more extensible to larger numbers of links and also reusable in other protein models.

```
refinement.geometry_restraints.edits {
  zn_selection = chain X and resname ZN and resid 200 and name ZN
  his117_selection = chain X and resname HIS and resid 117 and name NE2
  asp130_selection = chain X and resname ASP and resid 130 and name OD1
  bond {
     action = *add
     atom_selection_1 = $zn_selection
     atom_selection_2 = $his117_selection
     distance_ideal = 2.1
     sigma = 0.02
     slack = None
  }
  bond {
     action = *add
     atom_selection_1 = $zn_selection
     atom_selection_2 = $asp130_selection
     distance_ideal = 2.1
     sigma = 0.02
     slack = None
  }
  angle {
     action = *add
     atom_selection_1 = $his117_selection
     atom_selection_2 = $zn_selection
     atom_selection_3 = $asp130_selection
     angle_ideal = 109.47
     sigma = 5
  }
}
```

Figure 2: Example of the "`edits`" syntax for linking atoms with bonds and angles in `phenix.refine`.

The simplest method is the "`edits`" scheme specific to *PHENIX*. The format uses the `phil` syntax (cctbx.sf.net/libtbx_phil.html) that drives the command interface for *PHENIX*. The basic concept involved is selecting atoms and performing an action. The selection syntax is the same as the selection syntax in other portions of *PHENIX* and is shown in figure 2.

The first lines create selections of three atoms that can be used in the subsequent actions. Note that the selections must parse to a single atom but can be either all the alternative locations of an atom or a specific alternative location. The use of variables such `$zn_selection` reduces clutter and errors in the bond and angle scopes but is not required. The spelt-out selection can be used inside the bond or angle entries but care should be taken with multiple entries containing the same atoms.

The most common action is adding a geometric entity and is performed using the `*add` syntax. The ideal value of the geometry entities and the `sigma` value are needed by the refinement. Additional values include a

```
data_link_NGA-THR
#
loop_
_chem_link_bond.link_id
_chem_link_bond.atom_1_comp_id
_chem_link_bond.atom_id_1
_chem_link_bond.atom_2_comp_id
_chem_link_bond.atom_id_2
_chem_link_bond.type
_chem_link_bond.value_dist
_chem_link_bond.value_dist_esd
 NGA-THR  1 C1  2 OG1   single   1.439  0.020
loop_
_chem_link_angle.link_id
_chem_link_angle.atom_1_comp_id
_chem_link_angle.atom_id_1
_chem_link_angle.atom_2_comp_id
_chem_link_angle.atom_id_2
_chem_link_angle.atom_3_comp_id
_chem_link_angle.atom_id_3
_chem_link_angle.value_angle
_chem_link_angle.value_angle_esd
 NGA-THR  1 C1  2 OG1 2 CB    108.700  3.000
 NGA-THR  1 O5  1 C1  2 OG1   112.300  3.000
#
```

Figure 3: Example of the `cif_link` syntax for linking atoms with bonds and angles in phenix.refine.

`slack` value that enables a flat-bottomed potential well and a symmetry operator for linking symmetry-related atoms. The list of geometric restraints that can be added in this fashion currently includes bonds, angles, dihedrals and planes.

The second technique is the standard link mechanism from the Monomer Library (www.ccp4.ac.uk/html/mon_lib.html). Two files are required to perform the same function as `edits` but this approach is more efficient for defining many links.

The `data_link` file defines the geometric restraints involved in the linking between two "residues". In the example in figure 3, the saccharide specified by the code `NGA` is specified as bonding to the oxygen of a threonine (`THR`). The first line of the file specifies the id of the link. Subsequent lines specify the bond between the `C1` of the `NGA` and the `OG1` of `THR` along with an ideal value and estimated standard deviation (ESD) or

sigma value. Two angles are also specified below the bonds.

The second file is used to specify the residues that are to be linked. Figure 4 shows the syntax required by `phenix.refine` to apply the link, `NGA-THR`, to the two selected residues. Note that the ordering of the selection is important.

```
refinement.pdb_interpretation.apply_cif_link {
  data_link = NGA-THR
  residue_selection_1 = chain X and resname NGA and resid 900
  residue_selection_2 = chain X and resname THR and resid 42
}
```

Figure 4: Example of the `cif_link` syntax for linking atoms with bonds and angles in `phenix.refine`.

Using the latter method a library of the links could be developed and then simply applied to the desired residues in a refinement. This is a powerful feature and the Monomer Library contains a number of useful pre-defined links such as saccharide linking to protein and glycosidic bonds that can be accessed using the "`apply_cif_link`" demonstrated in figure 4.

Currently, different modules in *PHENIX* use one or the other technique to communicate information. Metal coordination uses the edits to addition restraints to `phenix.refine`. *eLBOW* uses the `cif_link` method to link ligands to a protein residue.

Multi-criterion kinemage graphics in PHENIX

Jeffrey J. Headd[a], Vincent B. Chen[c], Nathaniel Echols[a], Nigel W. Moriarty[a], David C. Richardson[c], Jane S. Richardson[c] and 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*
[c]*Department of Biochemistry, Duke University Medical Center, Durham, NC 27710*

Correspondence email: JJHeadd@LBL.Gov

An important component of crystallographic structure determination is validation of model quality. The MolProbity (Chen *et al.*, 2010) web server provides a variety of metrics to evaluate model quality and presents the analysis in both tabular and graphical forms. The KiNG structure viewer (Chen *et al.*, 2009) used by MolProbity displays structural analysis in the visual form of a multi-criterion kinemage, allowing the user to quickly identify both interesting *structural* features and model-building errors with criterion-specific graphical cues. *PHENIX* has previously incorporated MolProbity validation methods, such as rotamer outliers and steric clashes, in tabular form (Adams *et al.*, 2010), but has not provided the same level of graphical validation available from MolProbity. To complement tabular analysis, *PHENIX* now features multi-criterion kinemage graphics. Multi-criterion kinemages generated in *PHENIX* contain the same visual validation graphics as presented by MolProbity, which includes graphics for rotamer and Ramachandran outliers, steric clashes, angle and bond-length deviations, Cβ deviations and ribose pucker outliers for nucleic acids. Figure 1 depicts an example multi-criterion kinemage displayed in KiNG for ubiquitin (pdbID: 1UBQ). See figure 2 for a complete depiction of all available graphical validation metrics.

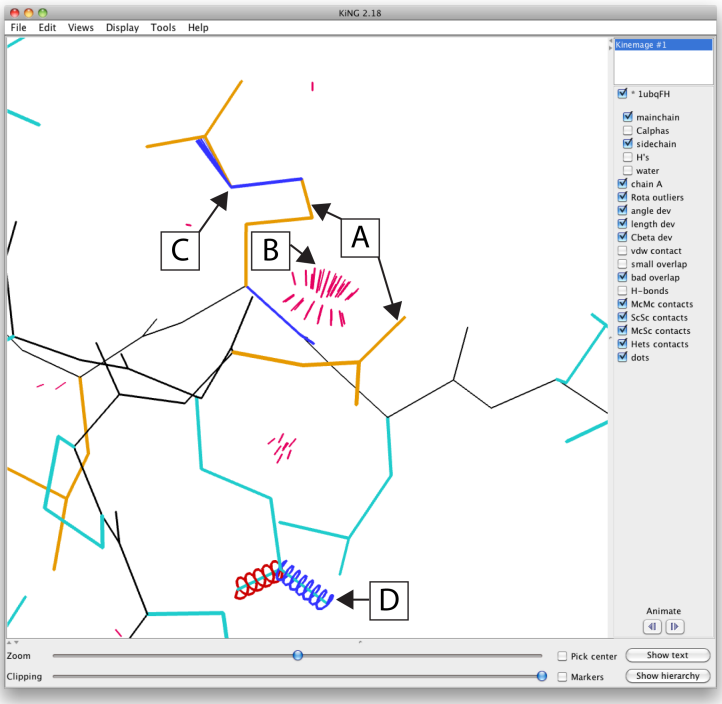


Figure 1: Close-up within a sample multi-criterion kinemage for ubiquitin (pdbID: 1UBQ). (A) Sidechain rotamer outliers are shown in gold as seen here for Arg A 72 and Asp A 39. (B) Significant steric overlaps (> 0.4Å) are shown by hot pink spikes. (C) Angle deviations greater that 4σ are flagged, blue for angles that are too small and red for angles that are too large. (D) Bond-length deviations greater than 4σ are flagged, blue for bonds that are too short and red for bonds too long.

As discussed in the July 2009 issue of the Computational Crystallography Newsletter, *PHENIX* now includes KiNG as a core component, so displaying kinemage graphics is natively available. When using the `phenix.refine` GUI, a multi-criterion kinemage is generated by default. At the completion of a refinement run, the *MolProbity* → *Summary* tab will have a button labeled "show validation in KiNG", as seen in figure 3. Clicking this button will launch KiNG and load the multi-criterion kinemage for interactive use. The .kin file is also available in the refinement project directory.

Multi-criterion kinemages are also available via the command line tools. To generate a kinemage on the command line, run:

```
phenix.kinemage 1ubq.pdb
```

This command will automatically add hydrogens if not present (crucial for correct contact analysis) and generate a multi-criterion kinemage in a file named `1ubq.kin`. To specify the output file name and/or location, run:

```
phenix.kinemage 1ubq.pdb out_file=/path/to/1ubq_multi.kin
```

Once generated, the multi-criterion kinemage may be viewed in KiNG by running:

```
phenix.king 1ubq_multi.kin
```

Serendipitously, restraints generated for novel ligands, modified amino acids and nucleic acids and other non-standard molecules with *eLBOW* (Moriarty *et al.*, 2009) can be used to augment structure validation. The kinemage generation methods in *PHENIX* use these definitions for both graphical rendering and model validation, providing more facile functionality over the current state-of-the-art in MolProbity for novel molecular handling. Kinemage generation in the `phenix.refine` GUI handles custom restraints incorporation automatically. For command line generation, a restraints file (`.cif`) may be included by running:

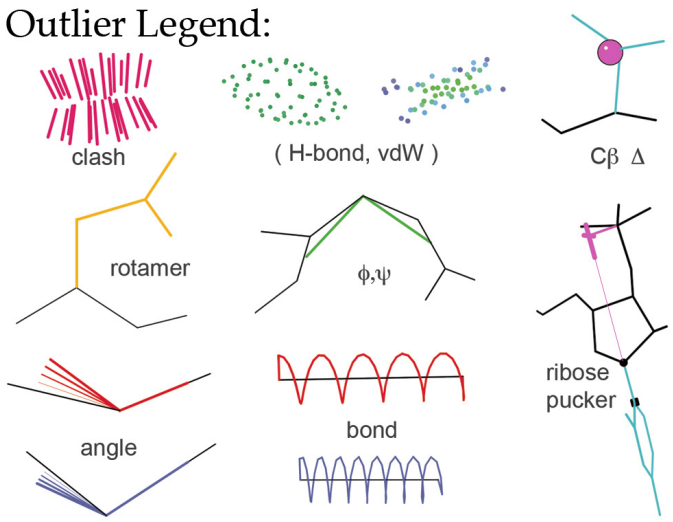


Figure 2: Legend of all outlier symbols used in multi-criterion kinemages generated by PHENIX. Taken from Chen *et al.*, 2010.

```
phenix.kinemage 1ubq.pdb cif=ligands.cif
```

The flexibility of the kinemage format and many display features of KiNG provide great opportunity for development of new crystallographic-specific validation graphics in *PHENIX*. KiNG is capable of displaying parallel coordinates, for example (Chen *et al.*, 2009), which will be useful for displaying trends across all validation criteria for tracking model improvement throughout the building and refinement process. Development is ongoing and future releases of *PHENIX* will feature expanded validation techniques to aid in structure solution.

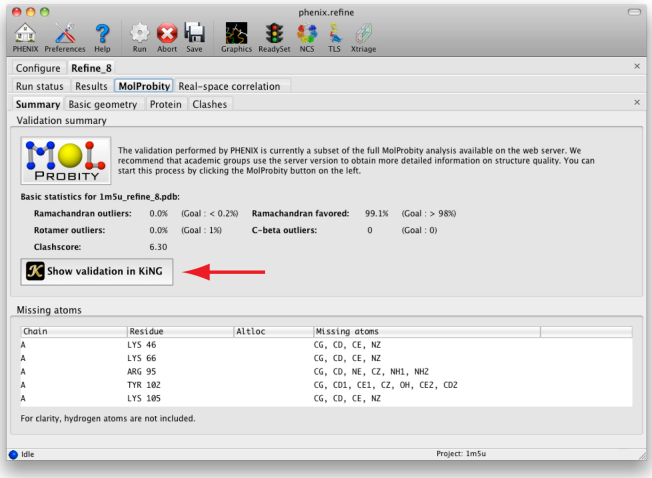


Figure 3: A multi-criterion kinemage is available in the `phenix.refine` GUI by clicking on the "Show validation in KiNG" button in the *MolProbity* tab. A multi-criterion kinemage is also generated in the comprehensive validation GUI in the same manner (not shown).

References

Adams P.D., Afonine P.V., Bunkóczi 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. and Zwart P.H.. (2010) PHENIX: a comprehensive Python-based system for macromolecular structure solution. Acta Cryst. D66:213-221.

Chen V.B., Davis I.W. and Richardson D.C. (2009) KiNG (Kinemage, Next Generation): A versatile interactive molecular and scientific visualization program. Protein Science 18:2403-2409.

Chen V.B., Arendall III, W.B., Headd J.J., Keedy D.A., Immormino R.M., Kapral G.J., Murray L.W., Richardson J.S. and Richardson D.C. (2010) MolProbity: all-atom structure validation for macromolecular crystallography. Acta Cryst. D66:12-21.

Moriarty N.W., Grosse-Kunstleve R.W. and Adams P.D. (2009) electronic Ligand Builder and Optimization Workbench (eLBOW): a tool for ligand coordinate and restraint generation. Acta Cryst. D65:1074-1080.

phenix.ensembler: a tool for multiple superposition

Gábor Bunkóczi and Randy J. Read

Department of Haematology, University of Cambridge, CIMR, Wellcome Trust/MRC Building, Hills Road, CB2 0XY, Cambridge, UK

Correspondence email: gb360@cam.ac.uk

In molecular replacement, a collection of superposed simple models ("ensemble" models) can give superior signal over any of the individual components. For example, hen egg white lysozyme can be solved with either mouse digestive lysozyme (2FBD, 40% identical) or apo bovine α-lactalbumin (1F6R, 40% identical) with both of them yield similar quality solutions (Table 1). However, when the two models are combined, solution quality (measured both in terms of translation function Z-score or final log-likelihood gain) increases (Table 1).

Table 1. Comparison of model quality between a two-member ensemble and its components.

Model	TFZ	Refined LLG
mouse digestive lysozyme	9.7	91.66
apo bovine α-lactalbumin	11.4	95.88
ensemble	12.6	146.49

In an ideal ensemble, individual components capture possible conformations for the flexible parts of the structure, while emphasizing the rigid core. Therefore, to create an ensemble, models have to be selected based on their conformational variability, while their rigid core has to be identified and superposed.

`phenix.ensembler` was written to automate the creation of models as much as possible as well as offer some help in deciding what models to include. It takes a series of PDB files as arguments (potentially unedited models from the PDB as identified by a homology search) and optional alignment files.

1. The program reads all PDB files and analyses all chains. All non-protein chains are discarded; multiple copies of the same chain are retained.
2. Equivalent residue positions in protein chains are aligned using one of the following methods (residue alignment):
   a. `ssm`: uses the secondary-structure matching (Krissinel & Henrick, 2004) algorithm (default).
   b. `muscle`: automatically create a multiple alignment using `phenix.muscle` (Edgar, 2004). Results are possibly less accurate than those from `ssm`, but are applicable for any protein chains, even those without any secondary structure.
   c. `alignments`: reads alignments provided on the command line, thereby giving full control over residue alignment.
   d. `resid`: aligns residues with identical residue number and insertion code.
3. Equivalent atoms in residues are aligned based on atom name (atom alignment) and atoms participating in the superposition are selected (default: C_α only).
4. Equivalent positions are then superposed using a multiple superposition algorithm. This gives better results than a series of pair-wise superpositions if there are significant differences between the structures. There are two algorithms implemented:
   a. `gapless` (Diamond, 1992). This is a fast algorithm, but it can only use sites that are present in all structures superposed and therefore may discard a significant number of sites when superposing several protein chains with distant homology (default).

b. `gapped` (Wang & Snoeyink, 2008). This algorithm can take alignment gaps into account in superposition and therefore can make use of more sites, giving more precise results.

5. Superposition is then alternated with weighting until convergence. Currently, there are two weighting schemes implemented:
   a. `unit`. This gives an unweighted superposition.
   b. `robust_resistant`. This assigns a weight to each superposed position based on the root mean-square deviation between all structures according to a robust-resistant weight function (default). Tolerance of the weighting can be controlled by the weighting.robust_resistant.critical parameter, with lower values down-weighting deviating positions more progressively. This approach proved very efficient in correcting for incorrect site alignment and identifying identical regions (Figure 1).
6. Superposed structures are analysed and clusters are determined based on structural similarity (controlled by the `configuration.clustering` parameter). Resulting clusters can be used to classify protein chains according to conformational variability. Depending on the number of protein chains, some experimentation may be necessary with the clustering distance parameter, so that an optimum is found between the two extremes (each structure forms a separate cluster *vs.* all structure belong to the same cluster). A representative from each cluster may then be chosen for inclusion in the ensemble.
7. After optionally sorting according to sequence identity, chain length, weighted and unweighted r.m.s.d., protein chains are transformed and written out.

The resulting ensemble model can be used directly in molecular replacement.

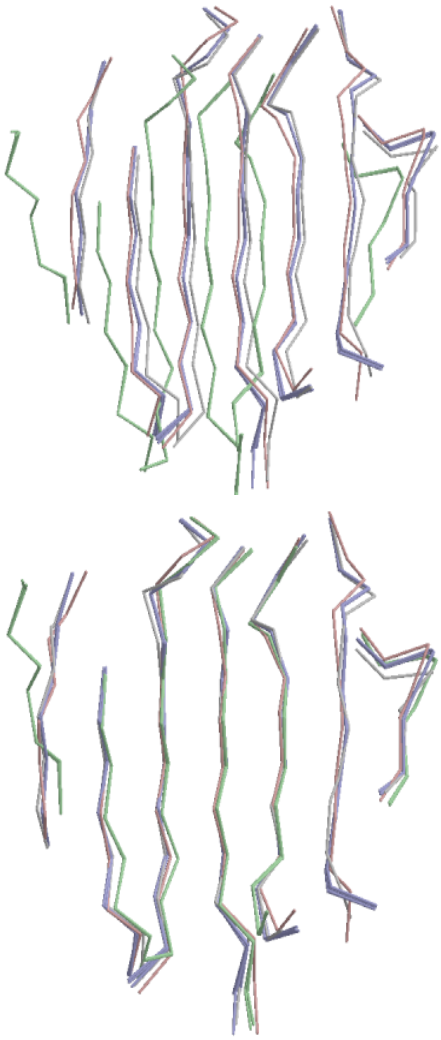


Figure 1. Correcting residue misalignment with weighting in superposition. Upper is the unweighted superposition using an alignment from *CLUSTALW* (Larkin et. al., 2007). The lower is the weighted superposition with the same alignment, resulting in much better structural agreement.

References

Diamond, R. (1992). On the multiple simultaneous superposition of molecular structures by rigid body transformations. *Protein Sci.* **1**, 1279-1287.

Edgar, R.C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.* **32**, 1792-1797.

Krissinel, E. & Henrick, K. (2004). Secondary-structure matching, a new tool for fast protein structure alignment in three dimensions. *Acta Cryst.* **D60**, 2256-2268.

Larkin, M. A, Blackshields, G., Brown, N. P., Chenna, R., McGettigan, P. A., McWilliam, H., Valentin, F., Wallace, I. M., Wilm, A., Lopez, R., Thompson, J. D., Gibson, T. J. & Higgins, D. G. (2007). Clustal W and Clustal X version 2.0. *Bioinformatics* **23**, 2947-2948.

Wang, X. & Snoeyink J. (2008). Defining and computing optimum RMSD for gapped and weighted multiple-structure alignment. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **5**, 525-533.

phenix.mr_rosetta: A new tool for difficult molecular replacement problems

Tom Terwilliger[a], Randy Read[b], Frank DiMaio[c] and David Baker[c]
[a]Los Alamos National Laboratory, Los Alamos NM 87545
[b]University of Cambridge, Department of Haematology , Cambridge, CB2 0XY, UK
[c]University of Washington, Department of Biochemistry, Seattle, WA, 98195, USA

Correspondence email: terwilliger@lanl.gov

What is phenix.mr_rosetta?

The *PHENIX* development team is working with the Baker laboratory at the University of Washington to combine the power of Rosetta structure modeling with *PHENIX* automated molecular replacement (MR), model-building, density modification and refinement. The basic idea is to find MR solutions with `phenix.automr`, rebuild them with Rosetta, including electron density map information, then rebuild those models with `phenix.autobuild`. The combination of Rosetta rebuilding and phenix rebuilding is the key part of this method. MR solutions are found with `phenix.automr` (Phaser), scored with LLG (optionally following Rosetta relaxation), the best solutions are picked and rebuilt with Rosetta including map information, the resulting models are scored with Rosetta, rescored with LLG, and the top models are rebuilt with `phenix.autobuild`.

What is phenix.mr_rosetta good for?

`phenix.mr_rosetta` can be very useful for cases where the search model used in molecular replacement is slightly too distant to rebuild successfully with `phenix.autobuild`. It can also be useful in cases where the model is too distant to even find a molecular replacement solution and pre-refinement with Rosetta can yield an improved search model.

How do I run mr_rosetta?

You can run `phenix.mr_rosetta` in a very automated way, or as a tool to find molecular replacement solutions and to systematically improve them. To run `phenix.mr_rosetta` you need to have both *PHENIX* (any recent version) and Rosetta (development version as of this writing, or version 3.2 or later once available), installed.

The basic inputs for `phenix.mr_rosetta` are pretty simple: (1) a data file with F, SIGF and freeR flags, (2a) a search model and an alignment file, or (2b) an hhpred file with a list of alignments and PDB file names and (3) a pair of fragments files that you create and download from the Robetta server.

You can get the `hhpred` file with alignments by pasting your sequence into the server at toolkit.tuebingen.mpg.de/hhpred. This takes about 10 minutes. You can get the fragments files by pasting your sequence into the Robetta server at robetta.bakerlab.org/fragmentsubmit.jsp. This takes a few hours to run, depending on the length of your sequence.

Once you have these files, you simply edit a simple script file for `phenix.mr_rosetta` that specifies these files (and other parameters if you wish). A typical command-line run of `phenix.mr_rosetta` is shown at the right.

```
phenix.mr_rosetta \
   seq_file=seq.dat \
   data=coords1.mtz \
   search_models=coords1.pdb \
   already_placed=True \
   fragment_files = test3.gz \
   fragment_files = test9.gz \
   rosetta_models=20 \
   ncs_copies=2 \
   space_group=p212121  \
   use_all_plausible_sg=False \
   nproc=200 \
   group_run_command=qsub
```

Does phenix.mr_rosetta require a cluster to run?

`phenix.mr_rosetta` does require building a number of Rosetta models with each model taking from 10-60 minutes to build with a single processor. In many cases, `phenix.mr_rosetta` can succeed with

as few as 20 models in each cycle. This means that a computer with 4 processors can be quite sufficient to run `phenix.mr_rosetta` and can finish in a day or so. In very challenging cases, as many as 2000 models may need to be built (the best models are picked and the density from the top 20% of models is averaged) making a cluster the best option. `phenix.mr_rosetta` can run on a Sun Grid Engine (SGE) cluster and on a Condor cluster. It may also run on other clusters. To run on a cluster you simply specify the command that you use to submit jobs ("`qsub`" for a SGE cluster for example).

Advanced uses of `phenix.mr_rosetta`

Once you have used `phenix.mr_rosetta` a few times, you will find that you can control where it starts and what it does in quite some detail. You can choose a particular solution that it is working on and have it build Rosetta models for that solution, then write out a table of results that you can examine. This way you can combine your intuition with the scoring that `phenix.mr_rosetta` uses to optimize your search.

You can also pre-refine your search model. This just means running Rosetta modeling on your search model without including any information from the crystallographic experiment. This can be very useful because Rosetta modeling can improve your search model and allow molecular replacement to succeed in cases where it might otherwise fail completely.

Where can I read more?

You can see all about running `phenix.mr_rosetta` in the *PHENIX* documentation.

Fuzzy space group symbols: H3 and H32

Ralf W. Grosse-Kunstleve[a], Nathaniel Echols[a] and 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: RWGrosse-Kunstleve@LBL.Gov

Introduction

Most applications in *PHENIX* (Adams et al., 2010) have to process space group symbols and most use the *sgtbx* space group toolbox for this purpose (Grosse-Kunstleve et al., 2002). The *sgtbx* space-group symbol interpreter accepts a large variety of inputs, for example "19", "P212121", or "P2(1)2(1)2(1)". Unfortunately it is not straightforward to accommodate the "H3" and "H32" symbols commonly used in macromolecular crystallography for the rhombohedral space groups *R*3 (No. 146) and *R*32 (No. 155), respectively. Currently these symbols appear in about 2.5% (1764 out of 69655) files in the PDB (www.pdb.org). This article explains the difficulties and how *PHENIX* handles rhombohedral space group symbols.

Conflicting definitions for H3 and H32

The main reference work for information about space groups and space-group symbols is *International Tables for Crystallography Volume A* (ITA) (Hahn, 1983-2006). Table 1.2 of ITA defines "symbols for the conventional centring types", which are *P*, *C*, *A*, *B*, *I*, *F*, *R* and *H*. The definition of the *H* symbol is not commonly used or known; the fractional coordinates of the lattice points within the unit cell are defined as 0,0,0; $\frac{2}{3},\frac{1}{3},0$; $\frac{1}{3},\frac{2}{3},0$. For comparison, the lattice points associated with the widely used *R* symbol are 0,0,0; $\frac{2}{3},\frac{1}{3},\frac{2}{3}$; $\frac{1}{3},\frac{2}{3},\frac{1}{3}$. The definition of the *H* symbol dates back to the precursor of ITA, *Internationale Tabellen zur Bestimmung von Kristallstrukturen* (Hermann, 1935) in which it is used for the description of 18 trigonal and 27 hexagonal space groups in "triple cell" settings (cci.lbl.gov/cctbx/multiple_cell.html). These settings were replaced by primitive or *R*-centered settings in *International Tables for Crystallography Volume I* (Henry & Lonsdale, 1952). In modern editions of the International Tables, the triple cell settings only appear in a column of ITA Table 4.3.1. According to this table, the symbol "H3" designates the triple-cell setting of space group 143, conventionally known as *P*3; the symbol "H32" (formatted $H3_2$) designates the triple-cell setting of space group 145, conventionally known as $P3_2$.

In general the macromolecular field uses the standard Hermann-Mauguin space-group symbols defined by ITA, but the PDB introduced a conflicting de-facto standard for "H3" and "H32". The root of the conflict is probably to be sought in the well-known ambiguities of Hermann-Mauguin symbols. A Hermann-Mauguin symbol uniquely identifies the space group type (one of the 230 crystallographic space group types), but not the setting. In the case of the seven rhombohedral space groups, the same Hermann-Mauguin symbol is used for a setting with a *R*-centered hexagonal basis system (e.g. "*R*3 (hexagonal axes)" in ITA) and a primitive setting with a rhombohedral basis system (e.g. "*R*3 (rhombohedral axes)"). Figure 1 shows how the basis systems are related. The ITA notation for the information about the setting is long compared to the Hermann-Mauguin symbol. This has lead many authors of crystallographic software and reference tables to introduce a more

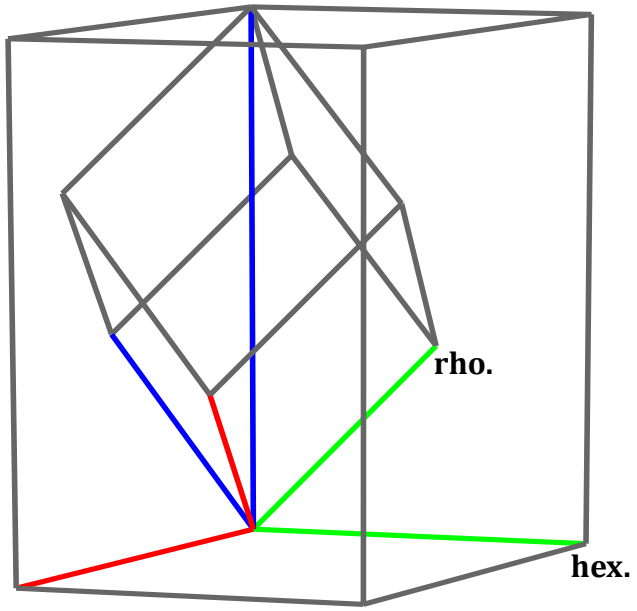


Figure 1: Basis systems of *R*3 settings with "**hex**agonal axes" and "**rho**mbohedral axes" (ITA nomenclature). The basis vectors are colored **a**=red, **b**=green, **c**=blue. For more information refer to ITA chapter 5.

compact notation. For example, the symbols "R3:h" and "R3:r" appear in the International Tables for Crystallography Volume B (Shmueli, 2001) Table A1.4.2.7 and the IUCr symCIF dictionary (Brown, 2005). The PDB has gone one step further by re-interpreting the first character of the "R3" and "R32" Hermann-Mauguin symbols as information about the setting. This confusion of centering type (ITA Table 1.2) and setting information is compounded by the conflict with the *H* centering type symbol.

The widely used *SCALEPACK* software (www.hkl-xray.com) has adopted a similar approach as the PDB, but in exactly the opposite way. According to the *SCALEPACK* manual, the symbols "R3" and "R32" correspond to the PDB symbols "H3" and "H32", respectively, and vice versa. Table 1 summarizes the symbols used for space groups 146 and 155 in different contexts.

Table 1: Summary of symbols used for space groups 146 and 155 in different contexts.

Space-group No.	Int. Tab. Vol. A	Int. Tab. Vol. B	PDB	SCALEPACK
146	R3 (hexagonal axes)	R3:h	H3	R3
	R3 (rhombohedral axes)	R3:r	R3	H3
155	R32 (hexagonal axes)	R32:h	H32	R32
	R32 (rhombohedral axes)	R32:r	R32	H32

Handling of H3 and H32 in PHENIX

An obvious way to disambiguate the rhombohedral space group symbols in Table 1 is to take the unit cell parameters into account. *PHENIX* makes use of this approach in some contexts, but it is not always practical. In many situations space group symbols and unit cell parameters are processed independently and combined only in advanced stages of involved procedures. A simple example is the validation of space group symbols in the *PHENIX* Graphical User Interface (GUI). The symbols are validated and standardized the moment the user presses the enter key or moves the input focus. It would be far more complicated to defer the validation until unit cell parameters are available. Potentially the GUI manages multiple unit cells and space group symbols and it may therefore not even be straightforward to formalize the relationships.

Until recently, the *PHENIX* GUI did not recognize the "H3" and "H32" symbols. Starting with *PHENIX* installer *dev-603*, the *sgtbx* library used by the GUI supports these symbols, following the de-facto PDB standard. We had been hesitant to take this step because it is the only non-conformance of the *sgtbx* library with the ITA standards. Eventually a considerable stream of negative feedback convinced us to value practicality higher than the principle of full ITA compliance. Since the new support for the PDB symbols is implemented in the *sgtbx* library, all applications using this library (`phenix.refine`, `phenix.xtriage`, `phenix.phaser`, to name just a few) also support these symbols now.

In the contexts of reading and writing PDB and CCP4 MTZ files, *PHENIX* included support for the PDB symbols for many years already. The interpretation of PDB `CRYST1` records is implemented in the `iotbx.pdb` module (Grosse-Kunstleve & Adams, 2010). In this context the rhombohedral space group symbols are used essentially only to infer the space group type. The choice of basis system, hexagonal vs. rhombohedral, is determined by inspection of the unit cell parameters. Internally *PHENIX* uses the symbols "R3:H", "R3:R", "R32:H" and "R32:R" (they appear, for example, in the GUI), but when formatting `CRYST1` records for output the PDB symbols are used instead.

Space group information from MTZ files is processed similarly, but in most cases the space group symbol included in the MTZ file is not actually used since the symmetry is usually defined unambiguously via lists of symmetry operations. When generating output MTZ files, *PHENIX* uses a copy of a CCP4 symmetry library file (`lib/data/symop.lib` in CCP4) to obtain the CCP4 space group symbol by matching the symmetry operations in the *sgtbx* space group object with the operations listed in the CCP4 library file. For the rhombohedral space groups, this mechanism produces the PDB symbols.

Starting with *PHENIX* installer *dev-603*, symmetry information read from merged *SCALEPACK* files is also subject to disambiguation via unit cell parameters.

Conclusion

The significant amount of programming effort and user frustration caused by the "H3" and "H32" space-group symbols is a good example of how time can be lost by not adopting long established standards. This is not meant to suggest revising the PDB as this would certainly only increase the confusion. Rather, it is a salient reminder to the entire crystallographic methods developer community to avoid ad-hoc approaches when possible. Ever more automated systems require highly reliable components and unambiguous semantics. The effort spent early on ensuring reliability and clarity is usually rewarded many times over as time passes.

References

Adams, P. D. et al. (2010). Acta Cryst. **D66**, 213-221.

Brown, I. D. (2005). ftp://ftp.iucr.org/cifdics/cif_sym_1.0.1.dic

Grosse-Kunstleve, R. W., Adams, P. D. (2010). Computational Crystallography Newsletter **1**, 4-11.

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

Hahn, T. (1983-2006). International Tables for Crystallography, Vol. A. Dordrecht: Kluwer.

Hermann, C. (1935). Internationale Tabellen zur Bestimmung von Kristallstrukturen. Berlin: Gebrüder Bornträger.

Henry, N. F. M., Lonsdale, K. (1952). International Tables for X-ray Crystallography, Vol. I. Birmingham: Kynoch Press.

Shmueli, U. (2001). International Tables for Crystallography, Vol. B. Dordrecht: Kluwer.

Visualizing the raw diffraction pattern with *LABELIT*

Nicholas K. Sauter[a]

[a]*Physical Biosciences Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720*

Correspondence email: NKSauter@LBL.Gov

Introduction

This article focuses on creating publication-quality pictures that illustrate diffraction data. While numerous tools are available for the routine conversion of raw data files into common image formats, other plots require specialized markup and can only be produced by custom-written software. A familiar example is the need to label each Bragg spot in the diffraction pattern with its proper Miller index. Also, synchrotron beamlines with very fast detectors such as the Pilatus-6M have emphasized the advantage of collecting data with thin rotation slices, which improve the signal-to-noise ratio but also leave the image sparsely populated with Bragg spots. This raises the need for a visualization tool where several consecutive images are summed together to give a more recognizable lattice. Another mechanism to conveniently examine the lattice is to plot the signal that corresponds to plane sections through reciprocal space. Old-style precession cameras would generate this type of photograph experimentally, but for modern rotation geometry it is necessary to synthesize such images with pixels taken from different shots over the whole data set. Code for all of these applications is available within the *LABELIT* package, which is distributed with *PHENIX* and available for download at www.phenix-online.org.

All examples discussed here are executed through the command line, as *LABELIT* has not yet been incorporated into the *PHENIX* graphical interface. Figures are documented in a special subdirectory within the source code: `labelit/publications/ccn_visualization` and can be reproduced by following the instructions contained therein. General documentation for *LABELIT* is at cci.lbl.gov/labelit.

Prerequisite indexing with *labelit.index*

Since the desired illustrations depend on knowledge of the Miller indices, we first perform an autoindexing step to identify the principle axes and unit cell dimensions of the crystal. It is assumed that the *PHENIX* package (any release subsequent to 1 Jan 2011) is installed and added to the path by sourcing the appropriate setup file (`phenix_env` or `phenix_env.sh`). As currently implemented, the program *MOSFLM* must also be placed on path under the alias `ipmosflm`; this is used during the autoindexing step to obtain a firm estimate of the resolution limits.

Indexing is done in a new current working directory (`cwd`), with the raw data frames placed either in `cwd` or any other directory:

```
cwd> labelit.index <data_path>/file_template_1_###.img 1 90
```

In this example the diffraction pattern is indexed from two 1° rotation images (#1 and #90) spaced widely apart to achieve the highest accuracy. Details are found in the *LABELIT* documentation. Of interest below, any lattice with higher than triclinic symmetry can be described in multiple Bravais settings:

```
LABELIT Indexing results:
Solution  Metric fit  rmsd  #spots  crystal_system  unit_cell
:)    5     0.197 dg  0.100    494  orthorhombic oP  84.6 123.3 174.2 90.0 90.0 90.0
:)    4     0.197 dg  0.105    496    monoclinic mP  84.6 123.4 174.3 90.0 90.0 90.0
:)    3     0.197 dg  0.099    493    monoclinic mP  84.6 174.2 123.3 90.0 90.0 90.0
:)    2     0.052 dg  0.081    496    monoclinic mP 123.3  84.6 174.0 90.0 90.2 90.0
:)    1     0.000 dg  0.082    498     triclinic aP  84.6 123.3 174.0 90.2 90.0 90.0
MOSFLM Integration results:
Solution  SpaceGroup Beam x   y  distance  Resolution Mosaicity RMS
:)    5         P222  94.08  94.11  179.99       2.20    0.050       0.040
      1           P1  94.08  94.09  180.02       2.14    0.050       0.027
```

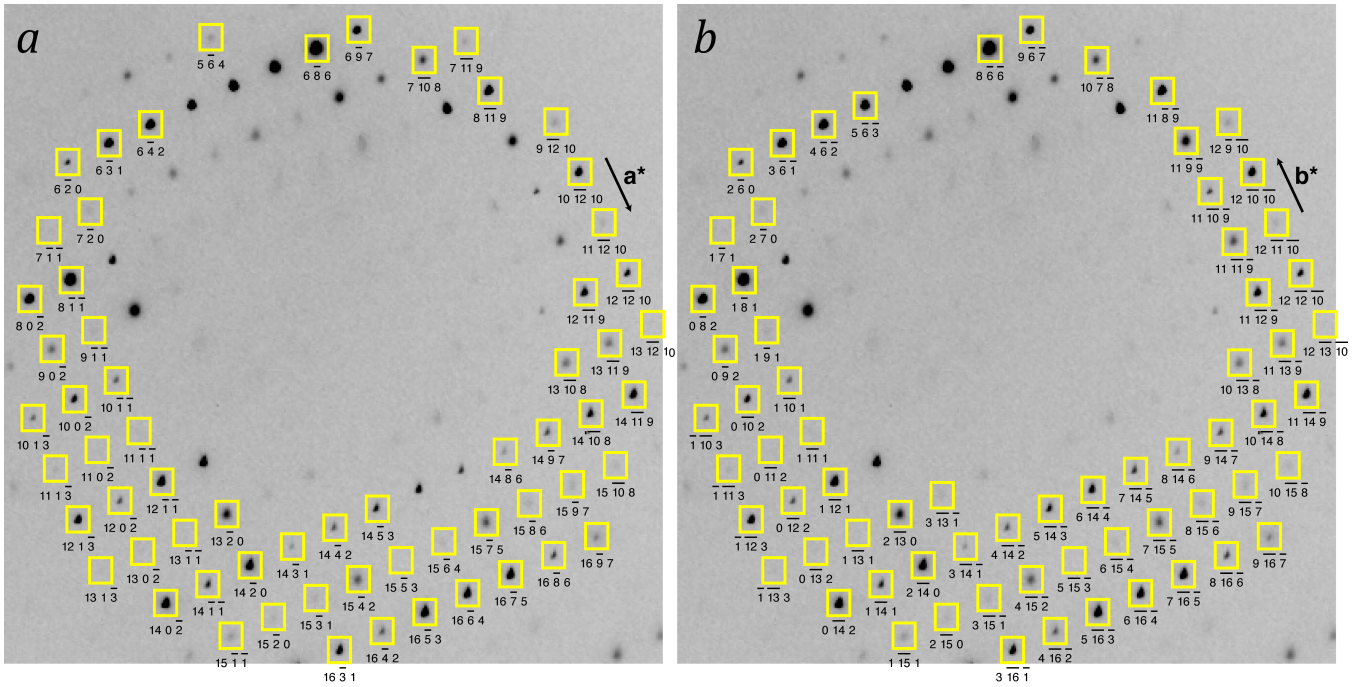


Figure 1. Detail of a rotation image from the 2qyv dataset, illustrated with `labelit.image`. The Bravais lattice is orthorhombic in model (*a*) and monoclinic in model (*b*), corresponding to the above-listed "*LABELIT* solution" numbers 5 and 2, respectively. While describing the same physical data, the two models differ in the orientations of the reciprocal cell axes (**a***, **b***, **c***) and the resultant Miller index labels attached to each Bragg spot. As the two lattice solutions are refined separately, slightly different subsets of Bragg spots are predicted by the two models (yellow boxes), but this distinction is normally erased in the subsequent steps of postrefinement and data integration.

As the printout shows, this orthorhombic lattice may be viewed either in the orthorhombic setting, or alternately in the triclinic or three different monoclinic settings. These are the indexing results from a dataset used by the Joint Center for Structural Genomics (JCSG) to solve the structure of Protein Data Bank entry 2qyv. The JCSG data repository, available for download from www.jcsg.org (Elsliger *et al.*, 2010), is used for the examples in figures 1, 3 & 4.

To create the subsequent illustrations, the indexing results must be kept in their cached location in the files `cwd/DISTL_pickle`, `cwd/LABELIT_pickle` and `cwd/LABELIT_possible`. Therefore, if multiple datasets are to be indexed, separate working directories should be created. Cached information in `cwd` can be deleted with:

```
cwd> labelit.reset
```

PDF-format rendering of diffraction images: *labelit.image*

The first product of interest is a simple picture of a rotation photograph, with Bragg spots labeled as shown in figure 1.

As with other programs in the *PHENIX* family, keyword input for `labelit.image` may be provided either at the command line or from an "effective parameter file" listing the desired keywords in a structured format. Relevant keywords can be listed out with:

```
cwd> labelit.image
cwd> labelit.image help # detailed descriptions for each keyword
```

Output from the undecorated `labelit.image` command may be used as a template to create the

effective parameter file `param.eff`:

```
bravais_choice = 5
image_number = 1
window_fraction = 0.4
window_center_x = 0.5
window_center_y = 0.5
image_brightness = 1.0
pdf_output{
  file = output.pdf
  box_size = 500
}
markup{
  bragg_spot{
    box = True
    linewidth = 0.04
    profile_shrink = 0
    color = yellow
  }
  miller_index{
    legend = False
    font_size = 10
    color = black
    vertical_offset = 10
  }
  inliers = False
}
```

The final image is computed with the command

```
cwd> labelit.image param.eff
```

or by conveniently supplying scoped command-line keywords for all non-default values:

```
cwd> labelit.image bravais_choice=5 \
                   image_number=1 \
                   pdf_output.file=output.pdf \
                   markup.miller_index.legend=True
```

Keywords and default values are as follows:

`bravais_choice=None`

This required keyword identifies the integer Bravais setting number to use for labeling the Bragg spots, as enumerated in the "*LABELIT* solution" column of the *LABELIT* output. The list of possible settings can be viewed again with the command `labelit.stats_index`. Fig. 1 illustrates how the output varies with different Bravais choices; the same physical data are displayed but differently numbered Miller indices are attached to each spot. The correct Bravais choice is not necessarily known at the time of indexing. In the case shown (PBD entry 2qyv) the published symmetry happens to be $P2_12_12$, corresponding to `bravais_choice=5` (figure 1a).

`image_number=None`

The integer image sequence number to use in the illustration (required). It is not necessary to supply the full file name, as the directory and file template are already cached by `labelit.index`.

```
window_fraction=1.0
```

Fractional length of the full image x and y dimensions to be used for illustration. A window_fraction of 0.5 would render a 1500 × 1500 square section of a 3000 × 3000 pixel raw image. To zoom in on an image detail, pick progressively smaller values.

```
window_center_x=0.5
window_center_y=0.5
```

Fractional offset on the full image to be used as the center of the section to be illustrated. The center of the raw image is at the coordinates (`window_center_x = 0.5`, `window_center_y = 0.5`), with x and y being the slow and fast directions on the image, respectively. On the printed page, slow is vertical and fast is horizontal, with the origin in the upper left corner.

```
image_brightness=1.0
```

Factor used to multiply the pixel values to produce a customized brightness. By default, a brightness scale is automatically calculated for each image, such that the 90th-percentile pixel is shown as saturated (black). An image_brightness > 1.0 makes pixels more saturated (darker).

```
pdf_output.file=None
```

Required file name for the output illustration. The top of the printed page will show the image file name and relevant information taken from the file header, while the `labelit.index` results and `labelit.image` command will be summarized at the bottom.

```
pdf_output.box_size=500
```

Number of points (unit of length, 1/72 inch) for the square edge of the illustration on the printed page.

```
markup.bragg_spot.box=True
```

Boolean value to toggle the boxes that locate the predicted position of each Bragg spot.

```
markup.bragg_spot.linewidth=0.04
```

Width of the printed lines used to outline each Bragg spot (in mm). Adjust this value to improve the clarity of the illustration if the Bragg spots are too congested.

```
markup.bragg_spot.profile_shrink=0
```

Number of pixels to shrink the box edge for outlining Bragg spots. By default, the rectangular box is sized to contain the average profile of the bright spots used by `labelit.index` for indexing, plus a two-pixel margin on each side. Use this keyword to improve clarity if necessary.

```
markup.bragg_spot.color=yellow
```

Color used for the Bragg spot boxes, as defined in the PDF-generating package *Reportlab*.

```
markup.miller_index.legend=True
```

Boolean value to toggle the inlining of Miller index HKL values underneath each Bragg spot.

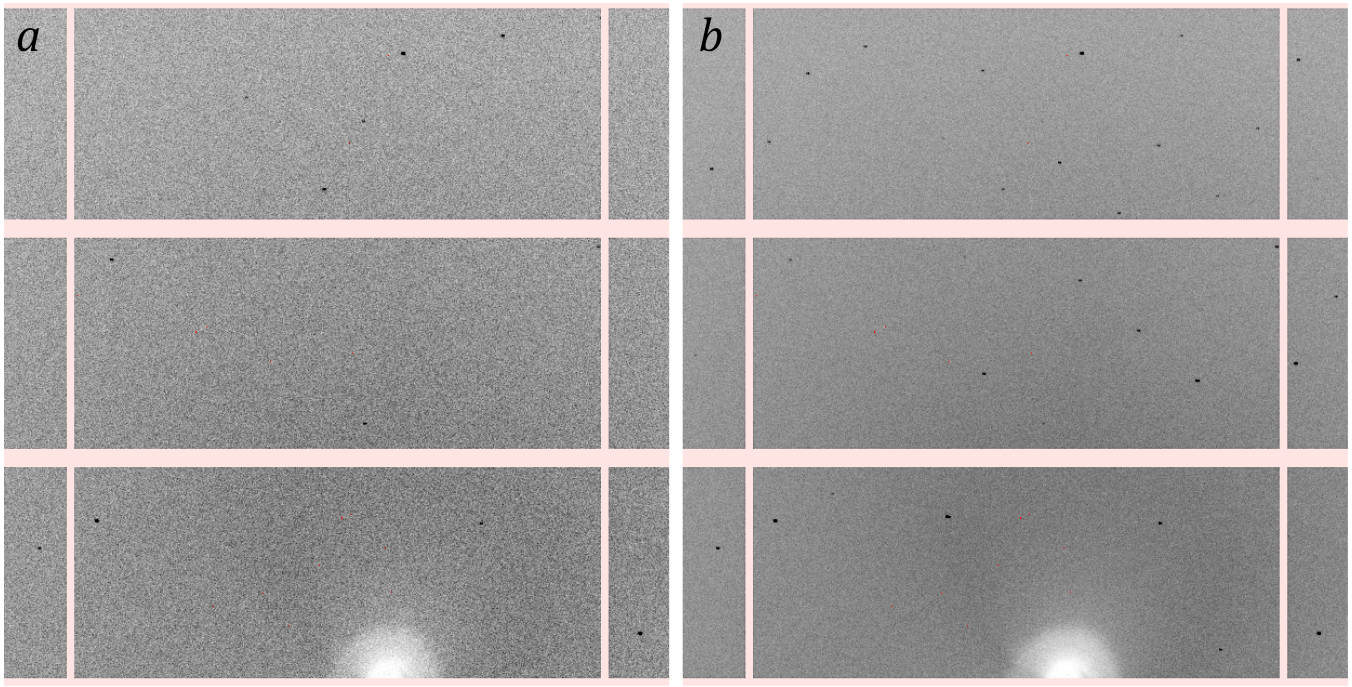


Figure 2. Detail from a cubic-lattice diffraction pattern taken from a single image (*a*, Δφ=0.2°) and a stack of five images (*b*, Δφ=1.0°). The rectangular areas are 195 × 487-pixel modules on a Pilatus-6M detector.

```
markup.miller_index.font_size=10
markup.miller_index.color=black
markup.miller_vertical_offset=10
```

For the inlined HKL values, font size in points, ink color as defined in *Reportlab*, and vertical offset in the downward direction in points, so that the HKL value does not overlap the spot.

```
markup.inliers=False
```

Boolean value to toggle dots locating the bright spots used by `labelit.index` for indexing. Red dots indicate the spot center of mass, pink dots show the maximum pixel.

Summation of consecutive thin-sliced images

Diffraction photographs of still crystals, or those with an extremely small rotation angle, will record a correspondingly thin slice through reciprocal space. Such photographs may exhibit few Bragg spots, especially if the unit cell is small and those that are captured will represent partial slices through the rocking curve, not full intensities. A sparse diffraction pattern from a 0.2° rotation photograph is shown in figure 2a.

For the purpose of illustrating the diffraction, it can be advantageous to stack consecutive rotation shots on top of each other, thus summing the partial intensities and filling out the layer slices so that the lattice pattern is more readily apparent. Such a construction is shown in figure 2b. The picture was created by supplying an `image_range` keyword:

```
cwd> labelit.image image_range=1,5
```

Keywords `image_range` and `image_number` are mutually exclusive and cannot be supplied together.

It is hoped that the availability of this method for stacking images will encourage crystallographers to

make it a common practice to acquire very finely sliced rotation images, which is now practical with the introduction of fast pixel array detectors such as the Pilatus-6M. Thin-sliced data (Pflugrath, 1999) have several advantages including improved signal-to-noise and the ability to model the Bragg spots with three-dimensional profiles. If it is easy to stack images for routine viewing, this removes the objection that thinly-sliced images are difficult to examine visually.

Note: it is possible to go immediately from raw images to PDF-format pictures *without* indexing first, if the object is to simply render the image without any markup. A separate command is provided for this purpose:

```
cwd> labelit.pdf <image template (/home/data/lysozyme_###.img)>
```

Keyword options are:

```
image_number = 1
image_range = 1,5
window_fraction = 0.4
window_center_x = 0.5
window_center_y = 0.5
image_brightness = 1.0
pdf_output{
  file = output.pdf
  box_size = 500
}
```

Synthesis of pseudo-precession photographs: *labelit.precession_photo*

While molecular structure is ideally explored with perfect crystals that give sharp Bragg peaks, it has been the imperfections that have posed large challenges over the years. The visual examination of images certainly plays an important role in diagnosing specific types of disorder (Nave, 1999). Aside from mosaicity, or isotropic disorder that gives rise to wide rocking curves (the diffraction of a Bragg spot over a large rotation angle), recent papers have highlighted specific types of long-range disorder that produce recognizable signatures at the "non-Bragg" positions of the diffraction pattern. For example, incommensurate modulation (Borgstahl *et al.*, 2009), a periodic distortion of the crystal lattice, generates discrete satellite Bragg spots; while lattice translocation disorder (Tsai *et al.*, 2009), the slight displacement of successive crystal layers, creates a pattern of streaks on specific spots.

Historically, the availability of Buerger precession cameras (*e.g.*, Blundell & Johnson, 1976) made it easy to examine specific planar layers of the reciprocal lattice, after painstaking alignment of the principal crystallographic axes (**a***, **b***, **c***) relative to the camera reference frame. In certain cases (Bragg & Howells, 1954), differences in the Bragg spot shape could be described as a function of Miller index. Achieving this type of convenient plot with modern rotation data requires a software calculation for two reasons: first, each image represents a curved surface of reciprocal space, not a plane; and secondly, the crystal axes are now rarely prealigned with the camera.

Assuming that images are available from a wide enough rotational range, the requisite planar section can be synthesized. However, it is best to keep in mind that there are implicit limitations. We assume, for example, that the crystal is rigidly fixed to the goniometer rotor, so its orientation is exactly known for each source image from the dataset. Deviations from this ideal will degrade the synthesized image, particularly at higher scattering angles. Also, our implementation does not apply scaling corrections. Thus, factors such as accumulated radiation dose that change the sample over time will cause symmetry-related reflections to appear unequal in intensity. Geometric approximations are unavoidable: in order to create a mapping between the raw image and reciprocal space coordinates, it is assumed that each raw image represents the center of its rotation range (for example, a 1° rotation image covering φ=[0°,1°] is uniformly assigned the value φ=0.5°). Moreover, image pixels far from the

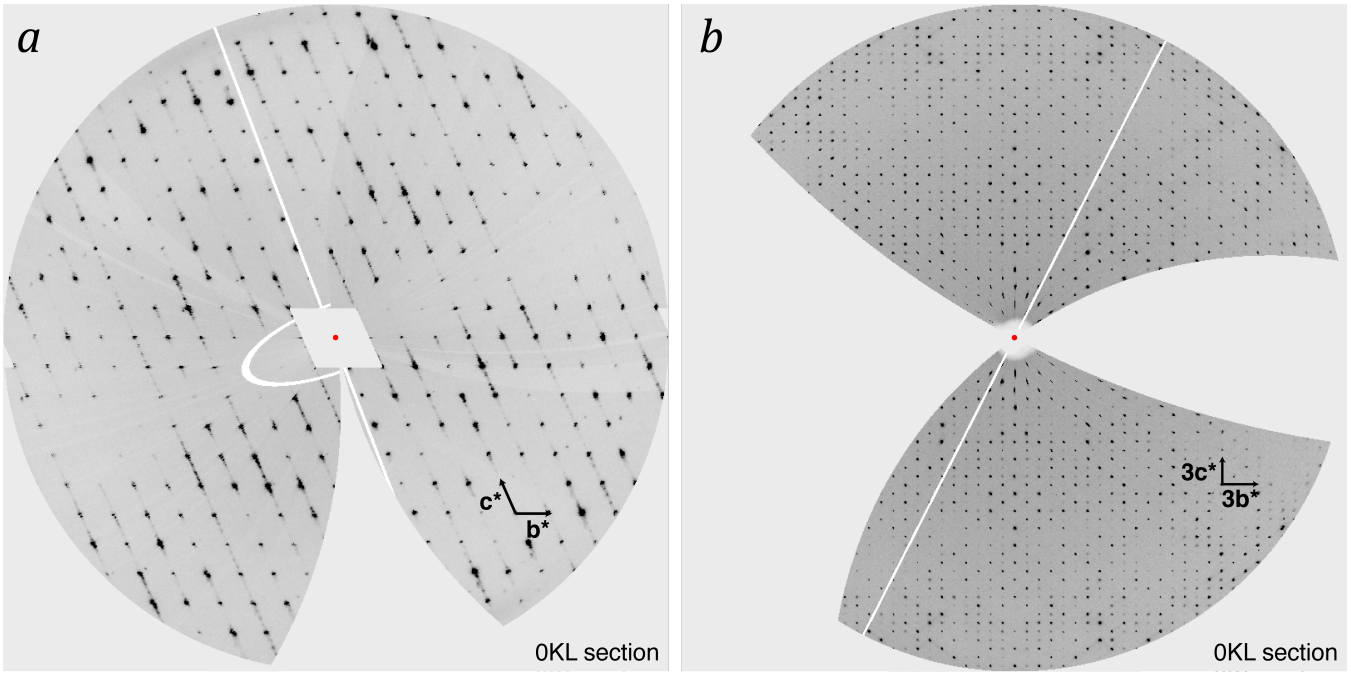


Figure 3. Reciprocal space sections illustrated with `labelit.precession_photo` for structures 1vk8 (*a*) and 2qyv (*b*). The origin of the streaks extending along the **c*** axis in (*a*) was not examined in the original publication (Dermoun *et al.*, 2010); but is presumably associated with lattice disorder.

rotation axis map to a larger rotational path through reciprocal space and appear as large quadrilaterals on the synthesized image. Clearly, the best sampling is obtained with fine rotational slicing (Pflugrath, 1999) and small image pixels.

Despite these caveats, `labelit.precession_photo` can be used (figure 3) to clearly illustrate phenomena that we recently cited: streaky Bragg spots of unknown origin associated with PDB code 1vk8 (Sauter & Poon, 2010) and a pattern of alternating weak and strong Bragg spots due to pseudotranslational symmetry in PDB structure 2qyv (Sauter & Zwart, 2009). The command line keywords for `labelit.precession_photo` are handled exactly as described above for `labelit.image`:

```
bravais_choice=None
image_range=None
pdf_output.file=None
pdf_output.box_size=500
```

Identical to the parameters described for `labelit.image`. A "None" value indicates required input. Here it is advantageous to specify an `image_range` covering the entire dataset, so that the coordinate grid of the synthesized image is filled in to the largest extent.

```
pixel_width=600
```

The width of the synthesized coordinate grid in pixels, which is then fit into the `pdf_output.box_size` dimension expressed in points.

```
resolution_outer=3.0
```

The high resolution limit of the requested plot, expressed in Ångstroms.

`intensity_full_scale=256`

Intensity value on the raw image that is treated as fully saturated (black).

`plot_section="H,K,0"`

Determine which principle axes are in the plane of the printed page; either **a***, **b*** (`H,K,0`); **b***, **c*** (`0,K,L`); or **a***, **c*** (`H,0,L`). Also, upper- and lower-layers can be sectioned, *i.e.*, "`H,K,1`"; "`H,K,-1`"; etc.

`layer_width=0`

The width of the reciprocal space section to be illustrated, given in fractional Miller index units. For example, if `plot_section` is "`H,K,0`" and `layer_width` is 0.05, then all image pixels mapping to reciprocal space coordinates between `H,K,-0.025` and `H,K,0.025` are plotted, with overlapping pixels being averaged. By default `layer_width=0`, corresponding to a section thickness of one pixel.

`apply_symmetry=None`

Point-group symmetry used to average the data. This option is not recommended by default, but is provided to respond to user comments that `labelit.precession_photo` printouts do not always look like true precession photographs. The full reciprocal space layer is not always covered. This is because the experimental rotation range is often less than the full 180° required for full coverage. Can point-group symmetry be applied to get an illustration that *looks* like a full precession photograph? Caution must be exercised, as such an operation could erase distinctions that might be important! As noted above the synthesized image normally reflects differences due to radiation damage, as well as variations in other factors such as the incident beam flux and the length of the absorption path. Moreover, the true symmetry of the diffraction may not be as high as implied by the Bravais choice, as with a monoclinic crystal with β angle close to 90°, which can be plotted within an orthorhombic cell. With these warnings in mind, the user can choose to impose point-group symmetry on the diffraction pattern if desired, with the `apply_symmetry` keyword.

Two choices must be made when selecting the point-group symmetry. First, should Friedel symmetry, $HKL = \overline{H}\overline{K}\overline{L}$, be imposed or not? Second, for certain crystal systems (such as tetragonal) there are alternate Laue groups to select. The Laue group cannot be selected automatically based on indexing alone, as it is necessary to compare symmetry-equivalent intensities after integration and scaling. To make the full matrix of choices clear, the user should type the undecorated command:

```
cwd> labelit.precession_photo
```

which outputs a table enumerating the point group choices for each possible Bravais setting, for example:

Bravais_choice	Lattice	Laue-group	Reflection-symmetry	Friedel-only
9	tP	4/mmm	422	-1
9	tP	4/m	4	-1
8	oC	mmm	222	-1
7	mC	2/m	2	-1
6	mC	2/m	2	-1
5	oP	mmm	222	-1
4	mP	2/m	2	-1
3	mP	2/m	2	-1
2	mP	2/m	2	-1
1	aP	-1	1	-1

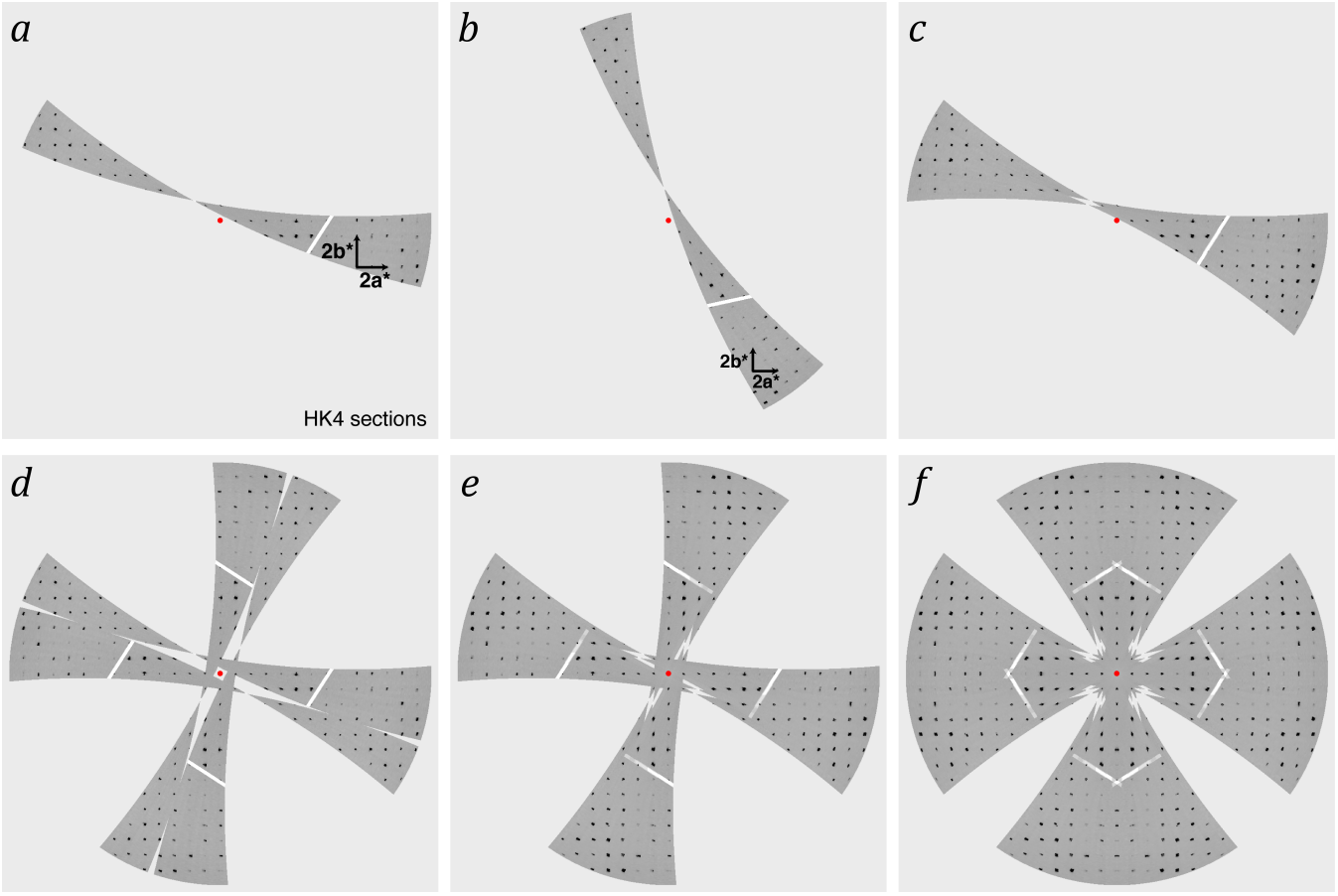


Figure 4. L=4 sections from the 1o3u diffraction pattern (Eriandsen *et al.*, 2004). The space group of the structure is $P4_12_12$, and the data are plotted either in the primitive tetragonal setting (*a*, *c*-*f*) or the *C*-centered orthorhombic setting (*b*). Point group symmetries imposed on the data are $\bar{1}$ (*c*), 4 (*d*), 4/m (*e*), and 4/mmm (*f*).

Various combinations of `bravais_choice` and `apply_symmetry` produce drastically different pictures, as illustrated in figure 4. Each panel plots the same 20° of rotation data from a tetragonal diffraction pattern. The `bravais_choice` keyword changes the axes on which the data are plotted without altering the intensities that are displayed, as seen by comparing panels (*a*, tP) and (*b*, oC). In contrast, the `apply_symmetry` option has the effect of increasing the reciprocal space coverage and/or averaging symmetry-redundant measurements of the displayed reciprocal space coordinates. For example, the application of Friedel symmetry (*c*) brings data from the L=–4 section on to the L=4 layer. Application of 4-fold symmetry (*d*) produces a clover-leaf pattern around the L-axis, while the combination of both Friedel and 4-fold symmetry (*e*) combines both effects. Finally, 4/mmm symmetry (*f*) yields two mirror planes at H=0 and K=0.

PNG- and GIF-format output

For completeness, we mention that *LABELIT* can also generate PNG-format images of the diffraction pattern:

```
cwd> labelit.png <image file> <output file.png> [-large]
cwd> labelit.overlay_distl <image file> <output file.png> [-large]
cwd> labelit.overlay_index <image file> <output file.png> [-large]
cwd> labelit.overlay_mosflm <image file> <output file.png> [-large]
```

The command `labelit.png` generates an undecorated image, while `labelit.overlay_distl` colors the subsets of bright spots either used (green) or not used (blue) for indexing. Commands

`labelit.overlay_index` and `labelit.overlay_mosflm` add markup of the predicted lattice for the highest Bravais choice, as refined by either *LABELIT* or *MOSFLM*, respectively. The only allowed keyword is `–large`, which imposes a one-to-one mapping between raw data pixels and pixels on the generated picture, otherwise the raw data pixels are binned in 2 × 2 squares.

An animated GIF-format movie can be generated of the entire dataset (this does not require indexing):

```
cwd> labelit.dataset_animation <template> <first image> <last image> <out>
cwd> labelit.dataset_animation /home/user/mydata/lyso_###.img 1 90 out.gif
```

Extensibility of Python code

Developers should be aware that the features discussed here could easily be extended by simple scripting in Python language. The applications discussed above are built on standard `cctbx` components for handling of detector formats (`iotbx.detectors`) and command-line keywords (`libtbx.phil`). Third party extensions are used for standard image formats (*Python Image Library*) and generation of PDF output (*Reportlab*).

Acknowledgments

Comments from software users were instrumental in developing the finished product. In particular, input from Tillman Heinisch (Universität Basel) and Jason Porta (University of Nebraska Medical Center) contributed significantly to `labelit.precession_photo`. The financial support of the National Institutes of Health / National Institute of General Medical Sciences under grant number R01-GM077071 is gratefully acknowledged. Operation of LBNL is partly supported by the US Department of Energy under Contract No. DE-AC02-05CH11231.

References

Blundell TL, Johnson, LN (1976). *Protein Crystallography*. London, Academic Press, Ltd.

Bragg WL, Howells ER (1954). X-ray diffraction by imidazole methaemoglobin. *Acta Crystallogr.* **7**, 409.

Dermoun Z, Foulon A, Miller MD, Harrington DJ, Deacon AM, Sebban-Kreuzer C, Roche P, Lafitte D, Bornet O, Wilson IA, Dolla A (2010). TM0486 from the hyperthermophilic anaerobe *Thermotoga maritima* is a thiamin-binding protein involved in response of the cell to oxidative conditions. *J. Mol. Biol.* **400**, 463-476.

Elsliger MA, Deacon AM, Godzik A, Lesley SA, Wooley J, Wüthrich K, Wilson IA (2010). The JCSG high-throughput structural biology pipeline. *Acta Crystallogr.* F**66**, 1137-1142.

Eriandsen H, Canaves JM, Elsliger MA von Delft F, Brinen LS, Dai X, Deacon AM, Floyd R, Godzik A, Grittini C, Grzechnik SK, Jaroszewski L, Klock HE, Koesema E, Kovarik JS, Kreusch A, Kuhn P, Lesley SA, McMullan D, McPhillips TM, Miller MD, Morse A, Moy K, Ouyang J, Page R, Robb A, Quijano K, Schwarzenbacher R, Spraggon G, Stevens RC, van den Bedem H, Velasquez J, Vincent J, Wang X, West B, Wolf G, Hodgson KO, Wooley J, Wilson IA (2004). Crystal structure of an HEPN domain (TM0613) from *Thermotoga maritima* at 1.75 Å resolution. *Proteins* **54**, 806-809.

Plugrath JW (1999). The finer things in X-ray diffraction data collection. *Acta Crystallogr.* D**55**, 1718-1725.

Sauter NK, Poon BK (2010). Autoindexing with outlier rejection and identification of superimposed lattices. *J. Appl. Crystallogr.* **43**, 611-616.

Sauter NK, Zwart PH (2009). Autoindexing the diffraction patterns from crystals with a pseudotranslation. *Acta Crystallogr.* D**65**, 553-559.

Electron density illustrations

Ralf W. Grosse-Kunstleve[a] and Luc J. Bourhis[b]
[a]*Lawrence Berkeley National Laboratory, Berkeley, CA 94720, U.S.A.*
[b]*Bruker AXS SAS, Champs-sur Marme, 77447 Marne-la-Vallée Cedex 2, France*

Correspondence email: RWGrosse-Kunstleve@LBL.Gov

Introduction

Crystallographers across all specializations invariably inspect electron density maps. This small article illustrates the effects of selected fundamental factors that shape a map, primarily to explain why macromolecular maps are often not scaled to absolute units (such as $e/Å^3$). Instead, "sigma scaling" is used, which is to compute the signal-to-noise ratio. Technically, this is to divide the density values in the map by their standard deviation (commonly called "sigma"), with the result that the standard deviation of the scaled values is one. Useful contour levels for graphical display are easily predictable for sigma-scaled maps, typically in the range 0.5-2. By contrast, the range of absolute electron density values is far more dependent on these factors:

- Isotropic Displacement Parameters (known as U_{iso} or B_{iso})
- High Resolution Limit
- Grid Resolution Factor
- Omission of the Fourier coefficient F_{000}

Here we show using a series of plots that these factors lead to a wide spread of absolute electron density values. We also take a brief look at how these parameters determine whether neighboring atoms are resolved in a density map.

B_{iso} dependence

For brevity, in the following we refer to isotropic displacement parameters as B_{iso} (see for example Grosse-Kunstleve & Adams (2002) and references therein). The B_{iso} dependence of electron density values is best illustrated via the Analytical Fourier Transform (AFT) since this eliminates the influences of the other factors in the list above. The formula for the AFT of a scatterer with displacement parameter B_{iso} and an N-Gaussian approximation to the X-ray scattering factor was given by Agarwal (1978) (see also Afonine & Urzhumtsev (2004) and references therein):

$$\rho(r) = \sum_{i=1}^{N} a_i \left(\frac{4\pi}{b_i + B_{iso}}\right)^{3/2} e^{-\frac{4\pi^2 r^2}{b_i + B_{iso}}} \qquad (1)$$

Here $\rho(r)$ is the electron density at a distance r (in Å) from the atomic center. a_i and b_i are the coefficients of the N-Gaussian approximation to the scattering factor $f(s)$ at the diffraction angle measurement $s = \sin\theta\,/\lambda$:

$$f(s) = \sum_{i=1}^{N} a_i e^{-b_i s^2} \qquad (2)$$

In the examples that follow, we work with a 5-Gaussian approximation (Grosse-Kunstleve et al., 2004) to the X-ray scattering factor of carbon. The corresponding Gaussian approximations of the International Tables for Crystallography Volume C (1992) and Waasmaier & Kirfel (1995) could not be used for the present purpose because they both involve a constant term, which is equivalent to a Gaussian term with $b_i = 0$. If the isotropic displacement parameter approaches zero, this leads to numerical instabilities when computing the AFT, as is apparent from equation (1). However, it should be noted that the International Tables, Wassmaier & Kirfel and 5-Gaussian approximations yield nearly identical results for B_{iso} values that are significantly different from zero.

Figure 1 shows AFT results with selected B_{iso} values ranging from 0.0 to 1.0 Å^2 in figure 1 (a) and powers

of 2

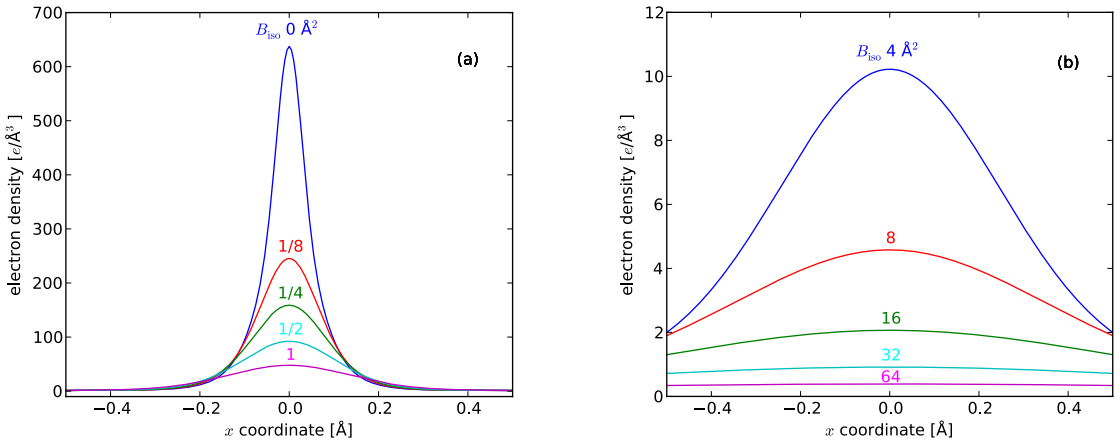


Figure 1: AFT results with selected B_{iso} values. The x coordinate is equivalent to the radial distance r of equation (1).

ranging from 4 to 64 Å^2 in figure 1 (b). The plots show that the absolute values of the electron densities are highly dependent on the displacement parameter. The electron density integrated over the unit cell is equal to the six carbon atom electrons for all values of B_{iso}, but the plots show how this charge is spread over the unit cell as B_{iso} increases, or conversely how it is concentrated around the carbon site as B_{iso} decreases. As a result, there cannot be any universally meaningful electron density contour level when the electron density is on an absolute scale.

High resolution limit

Figure 2 compares AFT and Discrete Fourier Transform (DFT) results with B_{iso}=0 Å^2. The unit cell used in the calculation is a cube with edge length 5 Å, but this value is not critical. The grid resolution factor for the DFTs is 1/4, which is used to determine the approximate map grid spacing by multiplication with the high resolution limit, for example $0.5\ \text{Å} \cdot 1/4 = 0.125\ \text{Å}$. The grid is constrained to uneven values to accommodate a symmetric range of Miller indices, for exampling ranging from -20 through 20 in each dimension, which leads to a slightly smaller grid spacing (5 Å / 41 = 0.122 Å).

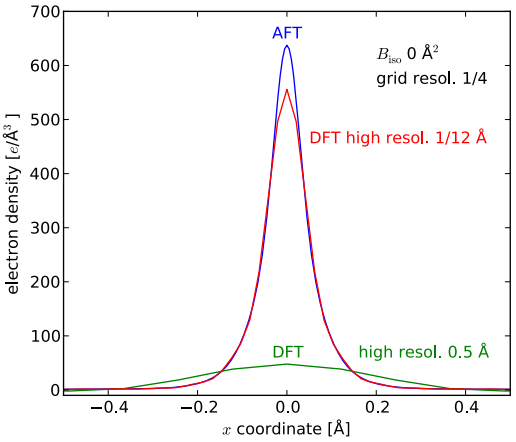


Figure 2: AFT and DFT results with B_{iso}=0 Å^2.

The blue plot is the AFT as before in figure 1. The red plot is the DFT of F_{calc} Fourier coefficients computed up to a high resolution of 1/12 Å (0.083 Å), which is the nominal limit of the 5-Gaussian approximation. The red plot is shown here mainly to demonstrate that the AFT is approximated well at an extremely high DFT resolution. The green plot is the DFT with Fourier coefficients up to a high resolution of 0.5 Å, which is still a very high resolution for macromolecular structures. The green plot illustrates that the omission of Fourier coefficients beyond the 0.5 Å resolution limit has a large impact on the electron density values if B_{iso}=0.

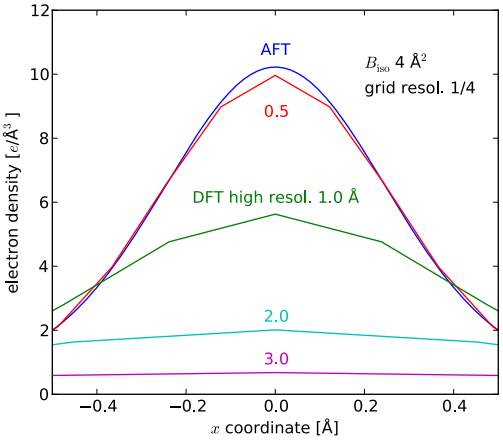


Figure 3: AFT and DFT results with B_{iso}=4 Å^2.

Figure 3 is similar to figure 2, but B_{iso}=4 Å^2 is used, a value in the typical range for small molecule crystal structures. In

this case the AFT and DFT electron density values are in close agreement if the high resolution limit is 0.5 Å. The plots with lower resolution limits (1.0, 2.0 and 3.0 A) demonstrate the still very strong influence of the resolution limit on the absolute electron density values given a non-zero B_{iso}.

Figure 4 is similar to the previous figure but B_{iso}=32 Å^2 is used, a value more typical for macromolecular crystal structures. Comparison of figures 3 and 4 illustrates that the discrepancy of the AFT and DFTs depends less strongly on the resolution limit if the value of the displacement parameter increases.

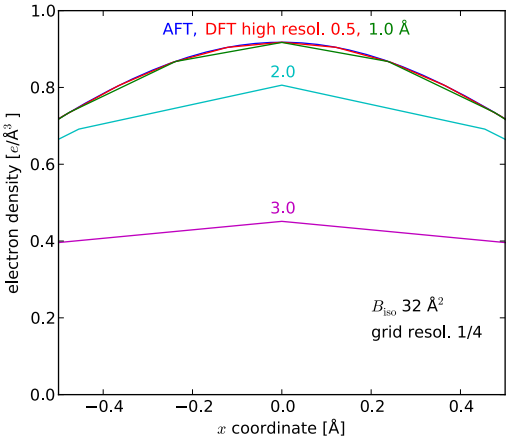


Figure 4: AFT and DFT results with B_{iso}=32 Å^2.

Grid resolution factor

Figure 5 illustrates the influence of the grid resolution factor on the appearance of the electron density. B_{iso}=4 Å^2 and a high resolution limit of 1.0 Å is used in the plots. The resolution factors are 1/2, 1/3, 1/4 and 1/8. It can be seen that the absolute electron density values are hardly affected by the resolution factor. Smaller resolution factors lead to smoother densities, but imply increased memory and runtime requirements. In most situations the factor 1/3 or 1/4 is a useful compromise.

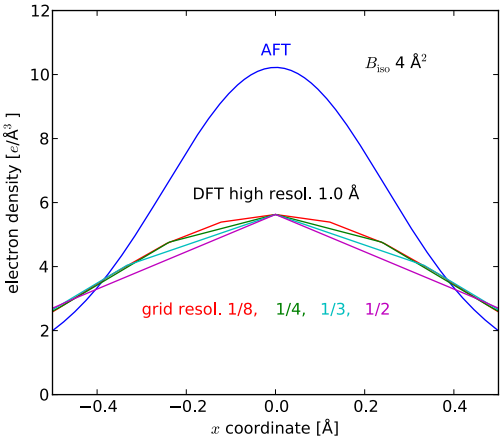


Figure 5: DFT results with a selection of grid resolution factors.

Omission of the Fourier coefficient F_{000}

Figure 6 illustrates the effect of omitting the F_{000} structure factor in the DFT, as is common practice. The contribution to the electron density is the constant $\langle\rho\rangle = F_{000}/V$, where V is the volume of the unit cell. The plots show the fractions contributed by $\langle\rho\rangle$ to electron density maxima at carbon positions of the model with PDB code 1ab1 (www.pdb.org); the values plotted are average fractions over all carbon positions. The DFTs were computed using F_{calc} with high resolution limits 0.5, 1.0, 2.0, 3.0 & 4.0 Å and carbon B_{iso} values 0, 4, 32, 64 and 128 Å^2. Similar calculations with other smaller and larger protein models showed that the plots in Figure 6 are typical. The $\langle\rho\rangle$ fraction of the electron density is a function of both the resolution and the displacement parameters. For very high resolution and small B_{iso} the $\langle\rho\rangle$ fraction is negligible but can grow to be significant at low resolutions and large B_{iso}. The $\langle\rho\rangle$ fraction in typical protein structures ranges from about 10% to 30%.

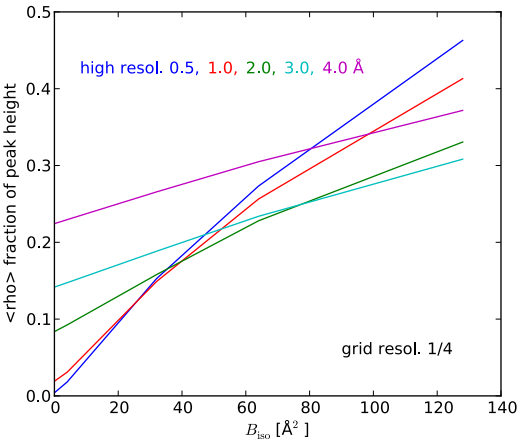


Figure 6: $\langle\boldsymbol{\rho}\rangle = \boldsymbol{F_{000}/V}$ contributions to maxima at carbon positions.

Two-atom examples

Theoretically, features separated by more than half the high-resolution limit of the Fourier coefficients can be distinguished in a DFT map. For example, two point scatterers separated by 1.5 Å should appear as separate maxima in a map if the Fourier coefficients extend to 3 Å or higher. Figure 7 illustrates the practical limits for resolving two neighboring carbon atoms with selected B_{iso} values and data resolution limits. The Gaussian shapes of both the X-ray scattering factor and the isotropic displacements lead to a

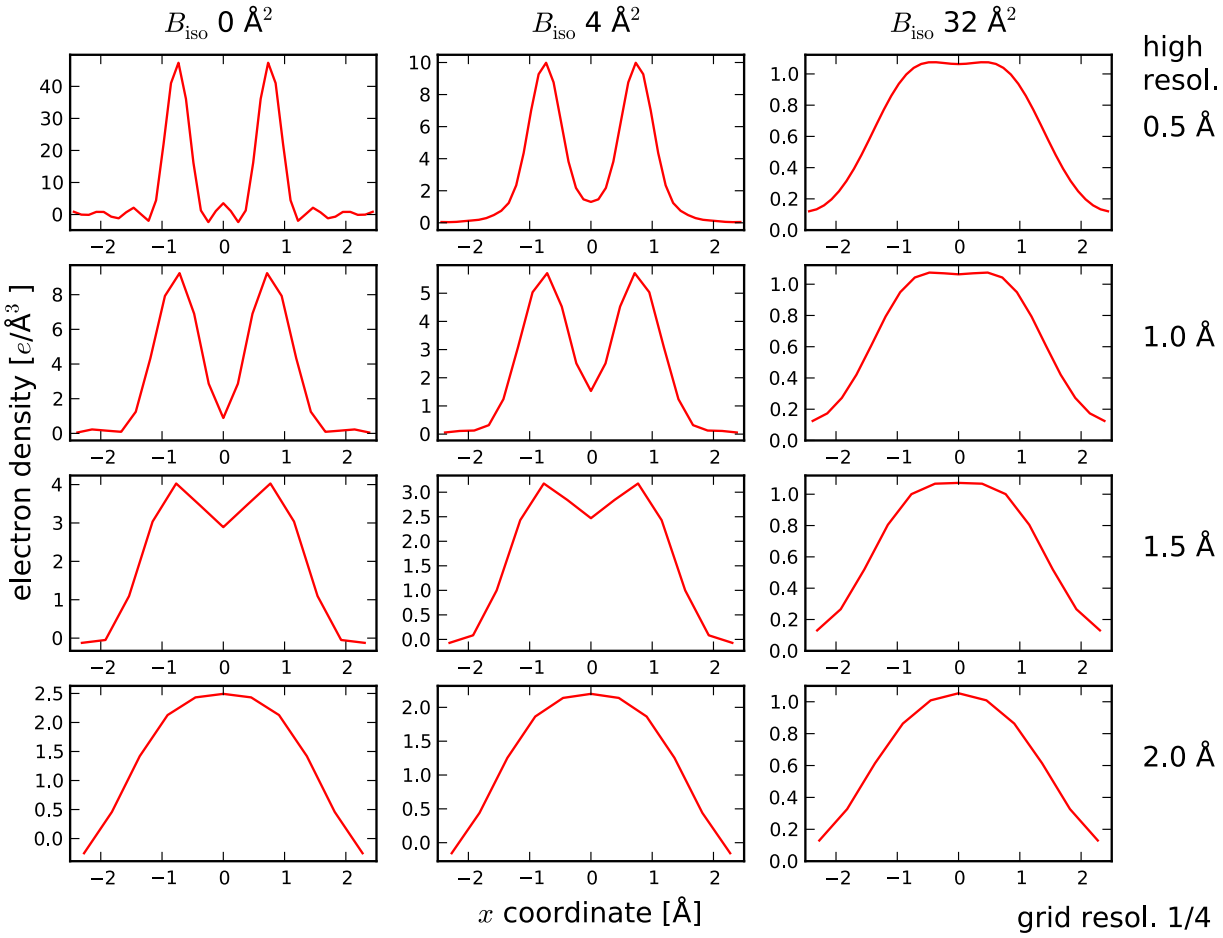


Figure 7: Examples of electron densities around two carbon atoms separated by 1.5 Å.

blurring of the electron density. Two carbon atoms separated by 1.5 Å are difficult to distinguish at 1.5 Å resolution even with B_{iso}=0 Å^2. Increasing B_{iso} values quickly reduce the effective resolution further.

Figure 7 also provides further examples of the spread of absolute electron density values; note the different scales of the y-axes in the matrix of plots.

References

Afonine, P.V. & Urzhumtsev, A.G. (2004). Acta Cryst. A**60**, 19-32.

Agarwal, R.C. (1978). Acta Cryst. A**34**, 791-809.

Grosse-Kunstleve, R.W., Adams, P.D. (2002). J. Appl. Cryst., **35**, 477-480.

Grosse-Kunstleve, R.W., Sauter, N.K., Adams, P.D. (2004). Newsletter of the IUCr Commission on Crystallographic Computing, **3**, 22-31.

International Tables for Crystallography Volume C (1992). Edited by A.J.C. Wilson, Kluwer Academic Publishers, Dordrecht/Boston/London.

Waasmaier, D., Kirfel, A. (1995). Acta Cryst., A**51**, 416-431.

Maximum likelihood refinement for twinned structures

Vladimir Y. Lunin
Institute of Mathematical Problems of Biology, Russian Academy of Sciences, Pushchino, Moscow region, 142290 Russia
Correspondence email: lunin@impb.psn.ru

Synopsis

A Rice-type probability distribution with properly adjusted parameters is a reasonable approximation for observed structure factor magnitudes when merohedral (or pseudo-merohedral) twinning is present. This allows to extend easy the usual maximum likelihood refinement technique for twinned structures.

Introduction

One of the most attractive ideas in crystallographic refinement is to enhance refinement power by maximisation of a likelihood function instead of the conventional minimisation of the least squares (LSQ) criterion. A special type of this likelihood function, which was used primarily for the evaluation of model quality (Lunin, 1982; Lunin & Urzhumtsev, 1984; Read, 1986; Lunin & Skovoroda, 1995; Urzhumtsev *et al*., 1996), was shown to be a good new goal function for the refinement of atomic models (Pannu & Read, 1996; Bricogne & Irwin, 1996; Murshudov et al., 1997; Adams et al., 1997). Now Maximum Likelihood (ML) refinement is an essential feature of such mainstream program complexes as phenix.refine (Afonine *et al*., 2005), REFMAC (Murshudov *et al*., 1997), Phaser (Read, 2001) and others. In this paper a simple way to extend this idea for refinement of twinned structures is discussed. In recent decades, twinning has been shown to be an important feature of macromolecular crystals (Yeates, 1997; Yeates & Fam, 1999; Dauter, 2003; Parsons, 2003; Lebedev et al., 2006). Twinned crystals are composed of separate differently orientated crystal specimens. If certain conditions for unit-cell parameters and orientation of the specimens are met (merohedral twinning), the reciprocal-space lattices of different domains coincide and the measured intensity of a diffracted beam becomes the sum of two (or more) different intensities that come from different specimens. The basic idea of ML refinement, namely, to maximise the probability to reproduce the observed values after allowed random corrections of the model have been done (Lunin, Afonine & Urzhumtsev, 2002), can be easily extended to take twinning into account. The problem is how to calculate the likelihood function practically. In this paper, a simple approximation for the likelihood function in the case of twinned intensities is suggested and tested. The found approximation allows conserving the usual "shape" of ML criterion and requires small corrections only in computational procedures.

1. Glossary

In this paper we distinguish four kinds of values:

- "true" or "theoretical" values that correspond to real structure and that are the goal of our study;
- "observed" values (intensities, or structure factor magnitudes) that were obtained in an experiment; they differ from the true values by experimental errors;
- "calculated" values that were obtained with the use of some model; these value differ from the true ones due to errors present in the model and approximations used in calculations;
- random values (structure factors, or errors) appear when we consider a value calcuated with the use of randomly generated additives.

2. Maximum likelihood refinement for twinned structures

2.1. Merohedral twinning

The phenomena of merohedral twinning may appear when the exact or approximate (pseudo-merohedral twinning) symmetry of crystal lattice exceeds the symmetry of crystal content. If $\mathbf{R}$ is such "extra" rotation symmetry operation and two specimens of the crystal linked by this rotation are

present in the X-ray beam simultaneously, then the diffraction patterns from two specimens overlap. If $I^{true}(\mathbf{s})$ is the intensity of $\mathbf{s}=(h,k,l)$ the indexed reflection for the first specimen, then theoretically, the measured intensity should be equal to

$$J^{theor}(\mathbf{s}) = (1-\kappa) I^{true}(\mathbf{s}) + \kappa I^{true}(\mathbf{R}^T\mathbf{s}) \tag{1}$$

Here $1-\kappa$ and κ are relative volumes of two specimens. The twinning fraction κ may be estimated by different methods (Yeates, 1988; Dauter, 2003; Lunin *et al*., 2007), which are outside of the scope of this paper.

2.2. Conventional least squares refinement

If some preliminary model of the studied structure exists a conventional least square refinement of the model parameters $\mathbf{q}$ could be performed by minimisation of discrepancy

$$Q_{LSQ}(\mathbf{q}) = \sum_{\mathbf{s}} \left(H^{calc}(\mathbf{s};\mathbf{q}) - H^{obs}(\mathbf{s}) \right)^2 \Rightarrow \min \tag{2}$$

with

$$H^{obs}(\mathbf{s}) = \sqrt{J^{obs}(\mathbf{s})}, \tag{3}$$

$$H^{calc}(\mathbf{s}) = \sqrt{(1-\kappa)\left|\mathbf{F}^{calc}(\mathbf{s};\mathbf{q})\right|^2 + \kappa\left|\mathbf{F}^{calc}(\mathbf{R}^T\mathbf{s};\mathbf{q})\right|^2}, \tag{4}$$

where $J^{obs}(\mathbf{s})$ are experimentally measured intensities from the twinned crystal.

A weakness of criterion (2) is that the model can contain irremovable errors so that no combination of flexible model parameters makes $\mathbf{F}^{calc}$ equal to $\mathbf{F}^{true}$ (and correspondingly H^{calc} equal to H^{true}). As a simple example, a portion of the atoms may be absent in the current model and their absence can not be compensated by moving of the atoms present in the model. Furthermore the target values H^{obs} in (2) differ from H^{true} by experimental errors. Furthermore, these errors cannot be corrected by changing of model parameters either. To some extent, these shortcomings may be overcome in the framework of maximum likelihood approach.

2.3. Maximum likelihood refinement

Maximum likelihood approach draws into refinement additional information present in the form of some statistical pattern for irremovable errors. For example one can consider measurement errors

$$\delta(\mathbf{s}) = J^{obs}(\mathbf{s}) - J^{true}(\mathbf{s}) \tag{5}$$

as independent random variables normally distributed with zero mean and variance $\sigma^2(\mathbf{s})$ (estimated in the experiment for every reflection separately). Similarly one can assume the any absences of atoms in the current model be uniformly distributed in the unit cell. After statistical properties of irremovable errors has been modeled the question may be asked "How large is the probability to get the calculated values equal to the observed values after random corrections have been introduce following the defined statistical pattern of irremovable errors?" This probability is called as “statistical likelihood” and it may be calculated for the independent observations as the product

$$L = \prod_{\mathbf{s}} Probability\left\{H^{cor}(\mathbf{s};\mathbf{q}) = H^{obs}(\mathbf{s})\right\}, \tag{6}$$

where $H^{cor}(\mathbf{s})$ stands for the random value that is the result of improvement of calculated values by random corrections. The likelihood may be adopted as a measure of goodness of the current structure model: the larger the likelihood value (6) the more reasonable the model. The choice of the model parameters resulting in the largest likelihood lies in the basis of maximum likelihood approach in mathematical statistics.

2.4. Calculation of the likelihood

One of the key problems in applications of the ML approach is the calculation of probability distributions for "randomly corrected" values, *i.e.* for $H^{cor}(\mathbf{s})$ values in our case. If the irremovable errors are restricted to the model incompleteness and measurement errors one can define random value $H^{cor}(\mathbf{s})$ as

$$H^{cor}(\mathbf{s};\mathbf{q}) = \sqrt{(1-\kappa)\left|\mathbf{F}^{cor}(\mathbf{s};\mathbf{q})\right|^2 + \kappa\left|\mathbf{F}^{cor}(\mathbf{R}^T\mathbf{s};\mathbf{q})\right|^2 + \delta(\mathbf{s})}, \tag{7}$$

$$\mathbf{F}^{cor}(\mathbf{s};\mathbf{q}) = \mathbf{F}^{part}(\mathbf{s};\mathbf{q}) + \mathbf{U}^{lost}(\mathbf{s}). \tag{8}$$

Here $\mathbf{q}$ is a set of current model parameters; $\mathbf{F}^{part}(\mathbf{s};\mathbf{q})$ are usual (deterministic) structure factors corresponding to the current partial model; $\delta(\mathbf{s})$ is a random error distributed with the normal distribution with zero mean and variance $\sigma^2(\mathbf{s})$; $\mathbf{U}^{lost}(\mathbf{s})$ are random structure factors calculated from randomly generated atomic positions.

A reasonable approximation to probability distribution of random variable $F^{cor}(\mathbf{s})$ is known (see e.g., Srinivasan & Parthasarathy, 1976) and for a general type non-centrosymmetric reflection is

$$P_F(F;\mathbf{s}) = \frac{2F}{\Sigma_Q(s)}\exp\left[-\frac{F^2 + \left(F^{part}(\mathbf{s})\right)^2}{\Sigma_Q(s)}\right] I_0\left(\frac{2FF^{part}(\mathbf{s})}{\Sigma_Q(s)}\right) \tag{9}$$

with

$$\Sigma_Q(s) = \sum_{j=1}^{N_{lost}} f_j^2(s), \tag{10}$$

where $f_j(s)$ are atomic scattering factors for the missing atoms and I_0 is the modified Bessel function. The distribution of the form (9) is often known as the Rice distribution.

The distribution for $F^{cor}(\mathbf{s})$ may have a more general form with two parameters $\alpha(\mathbf{s})$and $\beta(\mathbf{s})$ (Lunin 1982; Lunin & Urzhumtsev, 1984; Uzhumtsev *et al.*, 1996)

$$P_F(F;\mathbf{s}) = \frac{2F}{\beta(\mathbf{s})}\exp\left[-\frac{F^2 + \alpha^2(\mathbf{s})\left(F^{part}(\mathbf{s})\right)^2}{\beta(\mathbf{s})}\right] I_0\left(\frac{2\alpha(\mathbf{s})FF^{part}(\mathbf{s})}{\beta(\mathbf{s})}\right) \tag{11}$$

If more sources of errors are taken into account, *e.g.* atomic scattering factors are not known exactly, common scaling factor should be applied to $H^{cor}(\mathbf{s})$ before comparing with $H^{obs}(\mathbf{s})$, some errors are present in positions of atoms included in a rigid body block. Non-uniform prior coordinate distributions for the missing atoms may be involved in this scheme as well (Afonine *et al.*, unpublished).

2.5. Rice-type approximation for twinned intensity distribution

To get the final distribution that could be used to calculate the likelihood (6) one should calculated the triple convolution of probability distributions for $\left|\mathbf{F}^{cor}(\mathbf{s})\right|^2$, $\left|\mathbf{F}^{cor}(\mathbf{R}^T\mathbf{s})\right|^2$ and $\delta(\mathbf{s})$. This is not a simple task and the result cannot be presented in a simple analytical form. To avoid this problem one can skip the problem of convolution and suppose that the random values $H^{cor}(\mathbf{s})$ obey directly the distribution (11) with adequately designed $\alpha(\mathbf{s})$and $\beta(\mathbf{s})$ parameters. A reason for this hypothesis is that distribution (11) originates from Gaussian two-dimensional distribution. It is simply the marginal distribution of the distance-coordinate when the two-dimensional Gaussian distribution is written in polar coordinates. Due to the Central Limit Theorem (of theory of probabilities) Gaussian distribution appears in many different circumstances so that one can hope that its derivative (11) is suitable for $H^{cor}(\mathbf{s})$ values as well.

To check this hypothesis a series of test was performed (See section 3 below for the details). In these tests simulated sets of $H^{obs}(\mathbf{s})$ values were checked against theoretical distributions (11) with properly adjusted parameters. It was found that correspondence is very good for relatively strong reflections and is reasonable for weak ones. More definitely, the similarity of empirical and theoretical curves depended mostly on relative value of intensity of reflection in comparison with the part of the true intensity corresponding to the missing atoms. To characterise a reflection power we used the ratio $\lambda = H^{calc}(\mathbf{s})\big/\sqrt{\Sigma_Q(s)}$, where $\Sigma_Q(s)$ is mean intensity corresponding to the lost atoms defined by (10). Figure 1 shows empirical and theoretical probability distribution for different values of this ratio. A more accurate comparison was made with the use of χ^2 criterion. For large enough values of the ratio ($\lambda > 1$) these tests give no reason to reject the null hypothesis that the distribution of $H^{cor}(\mathbf{s})$ values is consistent with probability distribution (11). At the same time, for weak reflection the confidence level was small enough (see test 9 in table 1) and the approximation becomes less

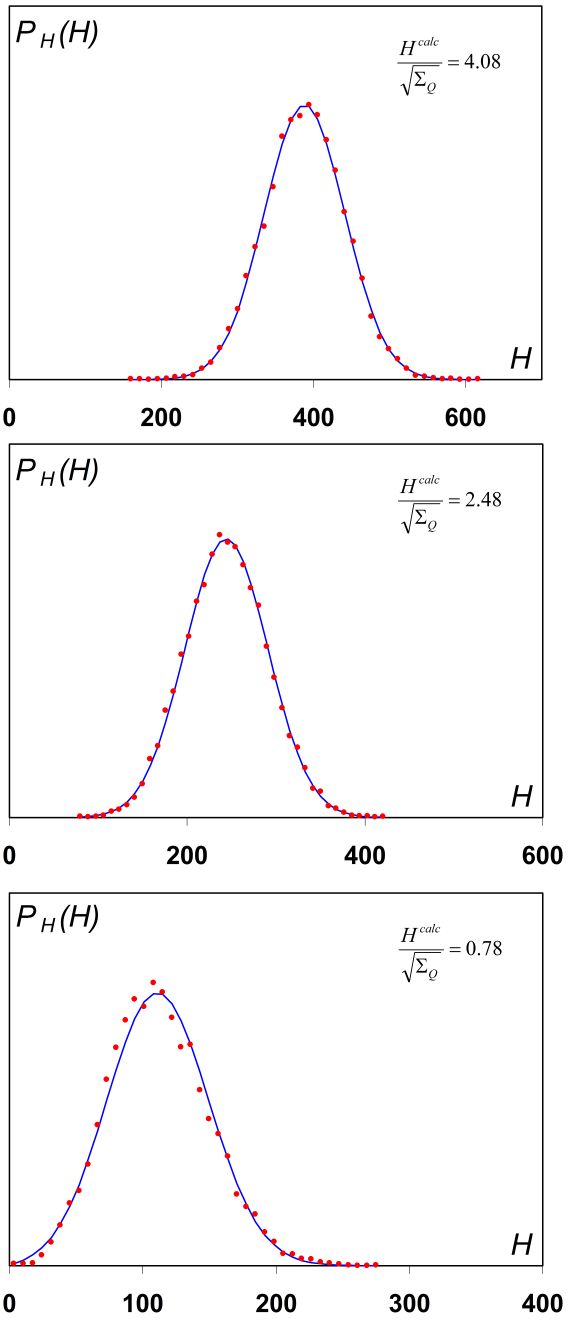


Figure 1. Theoretical distributions (11) with optimally defined parameters α/β (solid line) and distributions for simulated H^{obs} values (dots) for reflections with different $H^{calc}\big/\sqrt{\Sigma_Q}$ ratio. (See section 3 for details).

accurate (figure 1c). It is noteworthy that the tests did not reject the hypothesis when the twinning fraction was set as $\kappa = 0$. This means that in the usual ML refinement (without twinning) experimental measurement errors may be "absorbed" by α / β values.

A similar approach may be used if one has more then one twinning operation present. The only difference is that the calculated values $H^{calc}(\mathbf{s})$ are a mixture of more then two intensities.

2.6. Maximum likelihood refinement for twinned crystals

If the parameters α and β are defined and $\mathbf{q}$ is the set of flexible parameters of the partial model (*i.e.* parameters that are allowed to be changed in the refinement) then the likelihood function (6) takes the form

$$L(\mathbf{q}) = \prod_{\mathbf{s}} \frac{2H^{obs}}{\beta(\mathbf{s})} \exp\left[-\frac{\left(H^{obs}(\mathbf{s})\right)^2 + \alpha^2(\mathbf{s})\left(H^{calc}(\mathbf{s};\mathbf{q})\right)^2}{\beta(\mathbf{s})} \right] I_0\left(\frac{2\alpha(\mathbf{s}) H^{obs}(\mathbf{s}) H^{calc}(\mathbf{s};\mathbf{q})}{\beta(\mathbf{s})} \right) \quad (12)$$

(This formula supposes the non-centosymmetric reflection of a general type only be used in the calculation; see section 1.7 below for a more general case).

The calculation of logarithm of (12), changing the sign and omitting the terms that do not depend on the flexible model parameters, reduces the problem of maximisation of (12) to the problem of minimisation of the standard ML criterion

$$Q_{ML}(\mathbf{q}) = \sum_{\mathbf{s}} \left\{ \frac{\alpha^2(\mathbf{s})\left(H^{calc}(\mathbf{s};\mathbf{q})\right)^2}{\beta(\mathbf{s})} - \ln I_0\left(\frac{2\alpha(\mathbf{s}) H^{obs}(\mathbf{s}) H^{calc}(\mathbf{s};\mathbf{q})}{\beta(\mathbf{s})} \right) \right\} \Rightarrow \min \quad (13)$$

with $H^{obs}(\mathbf{s})$ and $H^{calc}(\mathbf{s})$ values defined as (3-4). The only difference with the conventional ML-criterion is that $F^{calc}(\mathbf{s})$ values as replaced by $H^{calc}(\mathbf{s})$.

2.7. Maxumum likelihood estimates of α and β parameters

The minimisation (13) supposes that the parameters $\alpha(\mathbf{s})$ and $\beta(\mathbf{s})$ are known for all reflections. To define these parameters providing model parameters are known, the maximum likelihood principle may be applied again (Lunin & Urzhumtsev, 1984; Read, 1986; Lunin&Skovoroda, 1995). Let us suppose that for a thin spherical shell in the reciprocal space parameters $\alpha(\mathbf{s})$ and $\beta(\mathbf{s})$ are the same for all reflection in the shell, i.e. $\alpha(\mathbf{s}) = \alpha$, $\beta(\mathbf{s}) = \beta$ and the product in (12) is calculated over reflection from this zone only. Consider now that the model parameters $\mathbf{q}$ in (12) be known and fixed, then the likelihood function depends on two parameters α and β only and these parameters may be defined by maximisation of the likelihood again. As before, this maximisation can be reduced to minimisation of

$$Q_{\alpha\beta}(\alpha,\beta) = K \ln\beta + \frac{1}{\beta}\sum_{\mathbf{s}}\left(H^{obs}(\mathbf{s})\right)^2 + \frac{\alpha^2}{\beta}\sum_{\mathbf{s}}\left(H^{calc}(\mathbf{s})\right)^2 - \sum_{\mathbf{s}} \ln I_0\left(\frac{2\alpha H^{obs}(\mathbf{s}) H^{calc}(\mathbf{s})}{\beta} \right) \quad (14)$$

where K is the number of reflections included in the sums. Now only the terms depending on α and β are included in minimisation criterion. A technique of minimisation of this function is described in

(Lunin & Skovoroda, 1995).

As it was shown before for usual refinement (Lunin, 1982; Lunin & Skovoroda, 1995) parameters α and β and model parameters $\mathbf{q}$ are highly biased if defined from the same set of reflections. So that α and β parameters should be defined on the base of test reflections only (*i.e.* ones excluded from refinement). The same observation is valid in the presence of twinning as well. Furthermore, some special attention should be paid when splitting the full set of reflections into work and test parts. The pair of reflections linked by the twinning law (*i.e.* $\mathbf{s}$ and $\mathbf{R}^T\mathbf{s}$) should have similar type ("work" or "test").

2.8. Remark on centrosymmetric and special reflections

The distribution (11) may be used as is for a general type non-centrosymmetric reflection. To go to a more common case we should correct distribution (11) for non-centrosymmetric reflection as

$$P_H\left(H;\mathbf{s}\right)=\frac{2H}{\varepsilon\left(\mathbf{s}\right)\beta\left(\mathbf{s}\right)}\exp\left[-\frac{H^2+\alpha^2\left(\mathbf{s}\right)\left(H^{calc}\left(\mathbf{s}\right)\right)^2}{\varepsilon\left(\mathbf{s}\right)\beta\left(\mathbf{s}\right)}\right]I_0\left(\frac{2\alpha\left(\mathbf{s}\right)HH^{calc}\left(\mathbf{s}\right)}{\varepsilon\left(\mathbf{s}\right)\beta\left(\mathbf{s}\right)}\right) \quad (15)$$

Here the parameter $\varepsilon\left(\mathbf{s}\right)$ depends only on the reflection indices and on the particular space group $\Gamma=\left\{\left(\mathbf{G}_\nu,\mathbf{t}_\nu\right)\right\}_{\nu=1}^{n}$ and may be calculated as the number of reciprocal-space symmetries $\mathbf{G}_\nu^T$ that when applied to the vector $\mathbf{s}$ leave it invariable, i.e. $\mathbf{G}_\nu^T\mathbf{s}=\mathbf{s}$. For centrosymmetric reflection the corresponding probability distribution has the form

$$P_H\left(H;\mathbf{s}\right)=\sqrt{\frac{2}{\pi\varepsilon\left(\mathbf{s}\right)\beta\left(\mathbf{s}\right)}}\exp\left[-\frac{H^2+\alpha^2\left(\mathbf{s}\right)\left(H^{calc}\left(\mathbf{s}\right)\right)^2}{2\varepsilon\left(\mathbf{s}\right)\beta\left(\mathbf{s}\right)}\right]\cosh\left(\frac{\alpha\left(\mathbf{s}\right)HH^{calc}\left(\mathbf{s}\right)}{\varepsilon\left(\mathbf{s}\right)\beta\left(\mathbf{s}\right)}\right) \quad (16)$$

The changes in (12-14) are straightforward. We do not include the statistical weight $\varepsilon\left(\mathbf{s}\right)$ into $\beta\left(\mathbf{s}\right)$ parameter as it would destroy the hypothesis that all $\beta\left(\mathbf{s}\right)$ are the same in a thin spherical shell in the reciprocal space, which is used to define α and β parameters provided the current model is fixed (see section 2.7).

3. Testing the null hypothesis

3.1. Tests and results

A series of tests were performed to check to what extent the Rice-type distribution (11) might approximate the distribution of measured twinned intensity for a particular reflection. In every such test a sample of "experimental" magnitudes $\left\{H_j^{obs}\right\}_{j=1}^{N}$ was generated by a Monte Carlo type procedure for some fixed reflection $\mathbf{s}$ taking into account three sources of uncertainties:

- "missing" atoms, i.e. ones present in the real structure, but absent in the current partial model;
- an experimental error when measuring intensities;
- an arbitrary scale for the measured intensities.

The parameters α^{opt} and β^{opt} in the theoretical distribution

$$P^{theor}\left(H;\alpha,\beta\right)=\frac{2H}{\beta}\exp\left[-\frac{H^2+\alpha^2\left(H^{calc}\right)^2}{\beta}\right]I_0\left(\frac{2\alpha HH^{calc}}{\beta}\right) \qquad (17)$$

that matches best the "observed" magnitudes were defined by maximisation of the likelihood

$$L\left(\alpha,\beta,\right)=\prod_{j=1}^{N}P^{theor}\left(H_j^{obs};\alpha,\beta\right)\Rightarrow \max \qquad (18)$$

(see details below). Figure 1 allows visual comparison of optimally fitted theoretical distributions with the empirical ones calculated from generated H_j^{obs} sets. These graphs reveal quite reasonable correspondence and back the idea that distribution (11) is suitable to describe deviations in experimental magnitudes when a twinning and experimental errors are present. To estimate the correspondence quantitatively the "null hypotheses" that $\left\{H_j^{obs}\right\}_{j=1}^{N}$ are random numbers generated following to $P^{theor}\left(H;\alpha^{opt},\beta^{opt}\right)$ probability distribution was checked under χ^2 criterion. Table 1 provides the obtained χ^2 value as well as the significance value defined as the probability to get the obtained or even worse (larger) value of χ^2 in the case when $\left\{H_j^{obs}\right\}_{j=1}^{N}$ are really obtained with the tested distribution (i.e. the null hypothesis is true). The null hypotheses should be rejected if the corresponding confidence level is small. Table 1 shows that for strong reflections one has no reasons to reject the null hypothesis. This is not the case for weak reflections. Nevertheless, the differences in empirical and theoretical distributions concern mostly "tails" of distribution. In ML refinement one tries to adjust the H^{calc} values to maxima of corresponding distributions, so that inaccuracies in "tails" of distributions for weak reflections should not affect the results of refinement too greatly.

Table 1. Results from a number of tests explained in the text. H^{calc} and Σ_Q are defined by equations (4) and (10) correspondingly.

test number	κ	$\frac{\sigma_I}{\left(H^{calc}\right)^2}$	$\frac{H^{calc}}{\sqrt{\Sigma_Q}}$	theoretical		defined		# bins	χ^2	confidence p	Fig.
				α	β	α^{opt}	β^{opt}				
1	0.	0.	5.04	1.	8701.	1.0	8690.	29	26.3	0.61	
2	0.	0.	2.37	1.	8675.	0.998	8721.	29	34.2	0.23	
3	0.	0.	0.50	1.	8601.	0.678	9828.	29	20.6	0.87	
4	0.	0.05	5.04			1.001	8897.	29	36.0	0.17	
5	0.	0.05	2.37			0.998	8832.	29	27.6	0.54	
6	0.	0.05	0.50			0.682	9816.	29	21.6	0.83	
7	0.36	0.05	4.08			1.011	5647.	29	24.1	0.73	1a
8	0.36	0.05	2.48			1.039	4435.	29	24.2	0.72	1b
9	0.36	0.05	0.78			1.435	3087	29	114.	$4.2\text{x}10^{-12}$	1c

3.2. Test parameters

Every particular test was defined by a set of parameters:

- the indexes of two twinned reflections $\mathbf{s}=(h_1,k_1,l_1)$ and $\mathbf{R}^T\mathbf{s}=(h_2,k_2,l_2)$ and magnitudes and phases of the corresponding structure factors calculated for a partial model; in our test the partial model was obtained by exclusion of 10% of atoms randomly from 1C5E (Yang *et al.*, 2000) model;
- the twin fraction value κ; (this value was defined as 0.36 for 1C5E crystal);
- the space group ($P2_1$ for 1C5E);
- the number $N_{missing}$ of atoms that are present in the whole structure, but are absent in the partial model; the positions of these atoms were considered as independent random variables uniformly distributed in the unit cell; $N_{missing}$ was set to 209 in our tests (10% of the whole number of atoms in 1C5E model);
- the accuracy of measuring of intensities σ_I; the measuring error δ was considered as normally distributed random variable with zero mean and variance σ_I^2; σ_I was set to zero or to $0.05*\left(H^{calc}\right)^2$ in our test;
- the scale factor $Scale$ converting magnitudes into some relative scale; It was found that this coefficient influence on optimal α and β values, bit does not influence on the quality of approximation; the results in table 1 correspond to value $Scale=1$;
- start value for random number generator, number of bins for χ^2 statistics, *etc.*

As a result simulated magnitudes H^{obs} were calculated as

$$H^{obs}=Scale*\sqrt{I^{twin}+\delta}\,, \tag{19}$$

$$I^{twin}=(1-\kappa)\left|\mathbf{F}^{part}(h_1,k_1,l_1)+\mathbf{U}(h_1,k_1,l_1)\right|^2+\kappa\left|\mathbf{F}^{part}(h_2,k_2,l_2)+\mathbf{U}(h_2,k_2,l_2)\right|^2 \tag{20}$$

where $\mathbf{U}(h,k,l)$ is a structure factor calculated from randomly generated coordinates of $N_{missing}$ atoms and δ is generated randomly with Gaussian distribution $N(0,\sigma_I^2)$. A characteristic value

$$\lambda=\frac{H^{calc}}{\sqrt{\Sigma_Q}} \tag{21}$$

with Σ_Q define by (10) was calculated for every test.

3.3. Special case: no twinning, no measurement errors

If the twinning and measurement errors are absent close formulas (9-10) exist for the probability distribution and parameters α and β. This may be used to check the developed numerical procedure (see section 4 below) for estimation of α and β parameters as well as to check the extent the generated $\left\{H_j^{obs}\right\}_{j=1}^N$ values are consistent with the theoretical distribution. Results of three such tests carried out for different λ-ratio occupy the first three lines in table 1. It is worthy of noting that though for weak reflection (test 3) the α^{opt} and β^{opt} parameters are different from the theoretical ones but the consistency of sample data with the found distribution is not worse then when using the distribution with theoretical parameter values. The number of bins, χ^2 value and confidence p are 31, 26.0 and 0.72 correspondingly in the last case.

3.4. Conventional maximum likelihood: no twinning, measurement errors present

Tests 4-6 were performed for the case when twinning is absent, but experimental errors are present. In this case $H^{calc}(\mathbf{s})$ coincides with $F^{part}(\mathbf{s})$. The goal of these tests was to estimate to what extent distributions (11) may be used in conventional ML-refinement without explicit corrections for measurement errors. The results support an idea that in refinement (13) these errors are taken into account implicitly be means of changing of the α and β parameters.

3.5. Common case

Tests 7-9 included both a twinning and measurement errors. For strong and medium reflections the distributions (11) describe well sample data $H^{obs}(\mathbf{s})$. For weak reflections ($\lambda < 1$) results are worse (test 9). Nevertheless the visual comparison of graphs in figure 1c shows that at "qualitative" level the difference is not too dramatic and these distributions could be used for weak reflections as well.

4. Maximum likelihood estimates of the distribution parameters

A problem we meet in these tests is how to define the parameters of distribution (17) that makes it the most consistent with sample data $\left\{H_j^{obs}\right\}_{j=1}^{N}$. Formally, this problem is a particular case of a more general problem studied in (Lunin & Skovoroda, 1995) where sample data were supposed to have different probability distributions (with different H_j^{calc} values) linked through the common α and β parameters. Nevertheless, this particular case is a "singular point" of the more general approach as some key parameter

$$\Omega = \left\langle \left(H^{obs}\right)^2 \left(H^{calc}\right)^2 \right\rangle - \left\langle \left(H^{obs}\right)^2 \right\rangle \left\langle \left(H^{calc}\right)^2 \right\rangle \qquad (22)$$

is equal to zero if all H_j^{calc} are the same. This requires a separate study of the problem.

4.1. Normalisation of variables

First we rewrite the problem (17-18) using a more convenient for analysis notation. Let the raw moments of the sample data $\left\{H_j^{obs}\right\}_{j=1}^{N}$ be

$$B_1 = \left\langle H^{obs} \right\rangle = \frac{1}{N}\sum_{j=1}^{N} H_j^{obs}, \quad B_2 = \left\langle \left(H^{obs}\right)^2 \right\rangle, \quad B_4 = \left\langle \left(H^{obs}\right)^4 \right\rangle, \qquad (23)$$

where $\langle . \rangle$ means the arithmetic mean, *i.e.* for any function $\psi(H)$ we define

$$\left\langle \psi\left(H^{obs}\right) \right\rangle = \frac{1}{N}\sum_{j=1}^{N} \psi\left(H_j^{obs}\right) \quad . \qquad (24)$$

If at least two observation in the set $\left\{H_j^{obs}\right\}_{j=1}^{N}$ are different, then

$$B_2 - B_1^2 = \left\langle \left(H^{obs} - \left\langle H^{obs} \right\rangle\right)^2 \right\rangle > 0 \,, \qquad (25)$$

so that

$$\frac{B_1^2}{B_2} < 1 \quad , \quad \frac{B_1}{\sqrt{B_2}} < 1 \quad . \tag{26}$$

Let normalised variables be

$$u = \sqrt{\frac{B_2}{\beta}} \, , \quad t = \frac{\alpha H^{calc}}{\sqrt{\beta}} \, , \quad z^{obs} = \frac{H^{obs}}{\sqrt{B_2}} \quad , \tag{27}$$

then the maximisation of the likelihood (17-18) is equivalent to minimisation of the function

$$Q(u,t) = -2\ln u + u^2 + t^2 - \left\langle \ln I_0\left(2utz^{obs}\right)\right\rangle , \qquad u \geq 0, t \geq 0 \, . \tag{28}$$

(The statistical weight $\varepsilon(\mathbf{s})$ is included into β parameter in this case).

4.2. The minimum of Q(u,t) at the border of the allowed region

Let study first the behavior of function $Q(u,t)$ at the border of the allowed region $u \geq 0, t \geq 0$. This function tends to infinity if $u \to 0$. Using an asymptotic expansion

$$\ln I_0(x) = x - \frac{1}{2}\ln x + \ldots \quad , x \to +\infty \tag{29}$$

we get for large values of the product ut

$$Q(u,t) = \left(t - \frac{B_1}{\sqrt{B_2}}u\right)^2 + \left(1 - \frac{B_1^2}{B_2}\right)u^2 + \ldots \quad , \quad ut \to +\infty \, , \tag{30}$$

so that $Q(u,t)$ function grows to infinity if u or t parameter grows. If $t = 0$, then the function

$$Q(u,0) = -2\ln u + u^2 \tag{31}$$

has the unique minimum at $u = 1$. As a result, the minimum at the border of the allowed region (including infinity) is attained for $(u,t) = (1,0)$.

4.3. Stationary point equation

If a minimum is attained at some inner point, then at this point one has

$$\begin{cases} \dfrac{\partial Q}{\partial u} = -\dfrac{2}{u} + 2u - t\Lambda(ut) = 0 \\ \dfrac{\partial Q}{\partial t} = 2t - u\Lambda(ut) = 0 \end{cases} \tag{32}$$

where the function $\Lambda(x)$ is defined as

$$\Lambda(x) = \left\langle 2z^{obs}\frac{I_1\left(2z^{obs}x\right)}{I_0\left(2z^{obs}x\right)}\right\rangle \quad . \tag{33}$$

Multiplying the first equation in (32) by u, the second by t and taking the difference we get

$$u^2 - t^2 = 1 \quad , \quad u = \sqrt{1+t^2} \tag{34}$$

The second equation in (32) may be used now to get the necessary condition for the point of the minimum

$$\Psi(t) = 2t - \sqrt{1+t^2}\,\Lambda\left(t\sqrt{1+t^2}\right) = 0 \quad . \tag{35}$$

4.4. Uniqueness of solution of Ψ(t)=0

We have $\Psi(0) = 0$ and using the asymptotic

$$\frac{I_1(x)}{I_0(x)} = 1 - \frac{1}{2x} + \ldots \quad , \quad x \to +\infty \tag{36}$$

we get

$$\Psi(t) = 2\left(1 - \frac{B_1}{\sqrt{B_2}}\right)t + \ldots \quad , \quad t \to +\infty \quad . \tag{37}$$

so that (due to (26)) $\Psi(t)$ increases for large t. As a result we have two alternatives possible

- $\Psi(t)$ increases starting from $t = 0$, so that $t = 0$ is the only solution of the equation (35); this means that the global minimum of $Q(u,t)$ is attained at $(1,0)$;
- $\Psi(t)$ decreases first, then grows to infinity, so that it exists a second solution of equation (35) corresponding to the global minimum.

To resolve this alternative it is necessary to study the vicinity of the point $t = 0$. Using the asymptotic

$$\frac{I_1(x)}{I_0(x)} = \frac{1}{2}x - \frac{1}{16}x^3 + \ldots \quad , \quad x \to 0 \tag{38}$$

we get

$$\Psi(t) = \left(\frac{B_4}{B_2^2} - 2\right)t^3 + \ldots \quad , \quad t \to 0 \quad . \tag{39}$$

It follows from the last formula that

- if $B_4 > 2B_2^2$, then $t = 0$ is the only solution for equation $\Psi(t) = 0$ and the global minimum of $Q(u,t)$ is attained at $u^{opt} = 1, t^{opt} = 0$; correspondingly $\alpha^{opt} = 0, \beta^{opt} = B_2 = \left\langle \left(H^{obs}\right)^2 \right\rangle$;
- if $B_4 < 2B_2^2$, then the equation $\Psi(t) = 0$ has non-zero solution t^{opt} and the global minimum of $Q(u,t)$ is attained at $\left(\sqrt{1+\left(t^{opt}\right)^2}, t^{opt}\right)$; parameters $\alpha^{opt}, \beta^{opt}$ are defined in this case as

$$\alpha^{opt} = \frac{\sqrt{B_2}}{H^{calc}} \frac{t^{opt}}{\sqrt{1+\left(t^{opt}\right)^2}} \quad , \quad \beta^{opt} = \frac{B_2}{1+\left(t^{opt}\right)^2} \quad . \tag{40}$$

4.5. Solution of the equation Ψ(t)=0

Due to non-monotonic behavior of function $\Psi(t)$ we used a zero-order method (without of the use of derivatives) to solve it. Nevertheless any other method can be used as well with necessary precautions.

5. Switching in the likelihood function

Let denote $J_j^{obs} = \left(H_j^{obs}\right)^2$, then the switching condition $B_4 < 2B_2^2$ may be written as

$$\left\langle \left(J^{obs} - \left\langle J^{obs} \right\rangle \right)^2 \right\rangle < \left\langle J^{obs} \right\rangle^2 \tag{41}$$

or

$$\sqrt{\text{variance}\left(J^{obs}\right)} < \text{mean}\left(J^{obs}\right) \quad . \tag{42}$$

The obtained condition means that the likelihood approach suggest to use a simple (Wilson) distribution

$$P^{theor}(H) = \frac{2H}{\left\langle \left(H^{obs}\right)^2 \right\rangle} \exp\left[-\frac{H^2}{\left\langle \left(H^{obs}\right)^2 \right\rangle} \right] \tag{43}$$

to describe the observed values instead of a more complicated distribution (17), if deviations of intensities J^{obs} from the mean are too large (more then the mean intensity value). To some extent we can interpret this as the likelihood function indicates that the calculated H^{calc} from a partial model is too unreliable and should not be used. Such switching is a rather typical feature of likelihood functions used for refinement or evaluation of atomic models in crystallography (Lunin, 1982; Lunin & Skovoroda, 1995; Lunin *et al.*, 2002).

Acknowledgments

This work was supported in part by grant 10-04-00254-a of Russian Foundation for Basic Research.

References

Adams, P. D., Pannu, N. S., Read, R. J. & Brünger, A. T. (1997). Cross-validated maximum likelihood enhances crystallographic simulated annealing refinement. *Proc Natl Acad. Sci. USA*, **94**, 5018-5023.

Afonine, P.V., Grosse-Kunstleve, R.W. & Adams, P.D. (2005). The Phenix refinement framework. *CCP4 Newsl.* **42**, contribution 8.

Bricogne, G. & Irwin, J. (1996). Maximum-likelihood structure refinement: theory and implementation within BUSTER + TNT. *Proceedings of the CCP4 Study Weekend*, Daresbury Laboratory, Warrington, England, 85-92.

Dauter, Z. (2003). Twinned crystals and anomalous phasing. *Acta Cryst.* D**59**, 2004-2016.

Lebedev, A.A., Vagin, A.A. & Murshudov, G.N. (2006). Intensity statistics in twinned crystals with

examples from the PDB. *Acta Cryst*. D**62**, 83-95.

Lunin, V.Y. (1982). The use of maximum likelihood approach to estimate phase errors in protein crystallography. *Preprint* (Russian), Pushchino, Russia. http://www.impb.ru/pdf/LCM_1982_3r.pdf

Lunin, V.Y. & Urzhumtsev, A.G. (1984). Improvement of Protein Phases by Coarse Model Modification. *Acta Cryst*., A**40**, 269-277.

Lunin, V.Y. & Skovoroda, T.P. (1995). R-free Likelihood-Based Estimates of Errors for Phases Calculated from Atomic Models. *Acta Cryst*. A**51**, 880-887.

Lunin, V.Y., Afonine P.V. & Urzhumtsev, A.G. (2002). Likelihood-based refinement. I. Irremovable model errors. *Acta Cryst*. A**58**, 270-282.

Lunin, V.Y., Lunina, N.L. & and Baumstark, M.W. (2007). Estimates of the twinning fraction for macromolecular crystals using statistical models accounting for experimental errors. *Acta Cryst*. , D**63**, 1129–1138

Murshudov, G.N., Vagin, A.A. & Dodson,E.J. (1997) Refinement of Macromolecular Structures by the Maximum-Likelihood Method. *Acta Cryst*. D**53**, 240-255.

Pannu, N. S. & Read, R. J. (1996). Improved structure refinement through maximum likelihood. *Acta Cryst*. A**52**, 659-668.

Parsons, S. (2003). Introduction to twinning. *Acta Cryst*. D**59**, 1995-2003.

Read, R. J. (1986). Improved Fourier coefficients for maps using phases from partial structures with errors. *Acta Cryst*. A**42**, 140-149.

Read, R.J. (2001). Pushing the boundaries of molecular replacement with maximum likelihood. *Acta Cryst*. D**57**, 1373-1382.

Srinivasan, R. & Parthasarathy, S. (1976). *Some statistical applications in X-ray crystallography*. Pergamon Press.

Urzhumtsev, A.G., Skovoroda, T.P. & Lunin, V.Y. (1996). A procedure compatible with X-PLOR for the calculation of electron-density maps weighted using an R-free-likelihood-based approach. *J.Appl.Cryst*., **29**, 741-744.

Yang, F., Dauter, Z. & Wlodawer, A. (2000). Effects of crystal twinning on the ability to solve a macromolecular structure using multiwavelength anomalous diffraction. *Acta Cryst*. D**56**, 959-964.

Yeates, T. O. (1988). Simple statistics for intensity data from twinned specimens. *Acta Cryst*. A**44**, 142-144.

Yeates, T. O. (1997). Detecting and overcoming crystal twinning. *Methods Enzymol*. **276**, 344-358.

Yeates, T.O. & Fam, B.C. (1999). Protein crystals and their evil twins. *Structure*, **7**, R25-R29.

TLS for dummies

Alexandre Urzhumtsev[a,b], Pavel V. Afonine[c] and Paul D. Adams[c,d]

[a]*IGBMC, CNRS-INSERM-UdS, 1 rue Laurent Fries, B.P.10142, 67404 Illkirch, France*
[b]*Physics Department, University of Nancy, B.P. 239, Faculté des Sciences et des Technologies, 54506 Vandoeuvre-lès-Nancy, France*
[c]*Lawrence Berkeley National Laboratory, One Cyclotron Road, BLDG 64R0121, Berkeley, CA 94720 USA.*
[d]*Department of Bioengineering, University of California Berkeley, Berkeley, CA 94720 USA.*

Correspondence email: sacha@igbmc.fr

Abstract

TLS model of a rigid-body harmonic displacement introduced in crystallography by Schomaker & Trueblood (1968) became a regular tool in macromolecular studies and is a part of most of modern refinement packages. There are a very large number of publications relevant to *TLS* and explaining its different aspects. However, these publications typically lack the details essential for understanding how the *TLS* model actually works, or contain too much of the formal mathematical details that are difficult to comprehend for readers without advanced mathematical background. In these notes we do not present any new development of the *TLS* model. Instead, we consider many simple examples that illustrate important features of the model. Using these examples, a general case is studied resulting in the widely known formulae. Simplified formulae are given for several special cases that may occur in macromolecular modeling and refinement. We believe that these notes may be useful for individuals who want to understand the basics of *TLS* modeling and not just use it as a "black box", as well as for crystallographic software developers wanting to implement some specific features described here.

Table of contents

1. Introduction

Crystallographic modeling of uncertainties in atomic positions uses different kinds of parameters. Some of them describe the same phenomenon with different accuracy; an example is isotropic or anisotropic atomic displacement parameters (*ADP*). Other parameters describe these uncertainties at different levels, such as individual atomic vibrations inside an atomic group or a thermal motion of this group as a whole (see Afonine et al., 2010 and references therein). In this sense, one should not consider an atomic group motion only as a way to reduce the number of parameters to work at low resolutions but as a way to better describe uncertainties in atomic positions.

Modeling a rigid group motion is based on the fact that any displacement of a rigid body may be considered as a superposition of a rotation around a given axis and a translation (see, for example, Goldstein, 1950). Eventually, these two motions may be correlated. When a rigid group oscillates, that is moves around its mean position, the term 'libration' is used instead of 'rotation' as it has been introduced in crystallography by Cruickshank (1956b). In what follows we stay within a harmonic model approximation that assumes small atomic displacements.

While some procedures to model harmonic rigid-group displacement and refine corresponding parameters have been suggested previously (see for example Pawley, 1963, 1964) nowadays the *TLS* model of a rigid-body harmonic displacement (Schomaker & Trueblood, 1968) is the mostly used one. Here *T*, *L* and *S* stand for 3 matrices describing translation, libration and their correlation (screw-rotation), respectively. It has been demonstrated that *TLS* modeling may provide reasonable results even for larger-scale vibrations (see for example Painter & Merritt, 2005). This ability to cover a broader vibrational range is considered a powerful feature of this modeling, although the results obtained at such inappropriate conditions should be interpreted with a caution.

There is a lot of literature about *TLS* modeling such as Johnson (1970, 1980), Scheringer (1973), Dunitz (1979), Stuart & Phillips (1985), Howlin et al. (1989), Tickle & Moss (1999), Winn et al. (2001), Painter & Merritt (2005, 2006a), Coppens (2006), Zucker et al. (2010) and references therein. The goal of this article is to give some technical details and practical computation schemes that are not available in the referenced above articles. Also, some specific cases are discussed (such as a *TLS* modeling with a fixed axis) that can be directly used in crystallographic structure refinement as an alternative to a traditional group ADP refinement. Differently from many articles on the subject, we try to keep all derivations and formulae at the basic level of the mathematics permitting most of readers to understand and reproduce them easily. We progress by short sections from easier specific cases to more complex and general ones.

2. Description of motion

2.1. Atomic displacement parameter (*ADP*)

A crystallographic atomic model represents not only time- and space-averaged positions $\mathbf{r}_n$ of atoms but also the uncertainties in these positions. These uncertainties result in blurring of atomic peaks in the Fourier maps and are characterised by atomic displacement parameters, *ADP*, also known as *B*-factors, isotropic or anisotropic. To simplify the analysis we suppose that all unit cells of the crystal have exactly the same structure and all uncertainties in atomic positions come from harmonic atomic motion only (as opposed to anharmonic large-scale motion resulting in distinct alternative conformations typically modeled using occupancies).

More formally, let's suppose that there is a Cartesian coordinate system with the origin **O** and the three orthonormal basis vectors $\mathbf{i}, \mathbf{j}, \mathbf{k}$ (the vectors are orthogonal to each other and are of a unit length). A position of an atom *n* at a moment *t* is defined by the coordinates (x_n, y_n, z_n) of $\mathbf{r}_n$ in this coordinate system (*e.g.* the PDB coordinates) and by the coordinates $(q_{nx}(t), q_{ny}(t), q_{nz}(t))$ of an instant deviation

$\mathbf{q}_n(t)$ from $\mathbf{r}_n$. The electron density in each point $\mathbf{r}_n + \mathbf{q}_n$ is proportional to the frequency (probability) $p_n(\mathbf{q}_n)$ characterizing the occurrence of the atom in this point. When (q_{nx}, q_{ny}, q_{nz}) are small, the logarithm of this probability, considered as a continuous smooth function, can be expanded into the Taylor series on these coordinates. When this expansion is done around a peak of the distribution, *i.e.* the most frequent position, the linear terms of this expansion are equal to zero as they correspond to the gradient of this function. (*Remark*: For a development around the mean position the linear term of the Taylor series does not necessarily vanish. For an atom in a single conformation it vanishes because this mean position can be expected to be close to or coincide with the most frequent value. For an atom in several alternative conformations such an expansion is done around each peak independently resulting in individual *ADP*s for each conformer). In a harmonic approximation, we have this expansion up to quadratic terms

$$p_n(\mathbf{q}) \approx \tilde{p}_n(\mathbf{q}) = \tag{2.1}$$

$$= \kappa_n \cdot \exp\left(\alpha_{nxx} q_x^2 + \alpha_{nyy} q_y^2 + \alpha_{nzz} q_z^2 + 2\alpha_{nxy} q_x q_y + 2\alpha_{nxz} q_x q_z + 2\alpha_{nyz} q_y q_z\right)$$

Non-harmonic approximations are discussed for example in Trueblood *et al*. (1996) Coppens (2006) and references therein. Expression (2.1) can be put into a standard form where the coefficients of the quadratic function of the atomic coordinates are presented by a symmetric matrix U_n^{-1}

$$\tilde{p}_n(\mathbf{q}) = (2\pi)^{-3/2} (\det U_n)^{-1/2} \exp\left(-\frac{1}{2}\mathbf{q}^\tau U_n^{-1} \mathbf{q}\right) \tag{2.2}$$

(see for example Trueblood *et al*., 1996). Here and in what follows the vector of instant deviation $\mathbf{q}$ is presented by a column vector of its coordinates $\begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix}$ and τ stands for transposition, $(q_x, q_y, q_z) = \begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix}^\tau$. The matrix U_n is defined as (see for example Cruickshank, 1956a and references therein)

$$U_n = \left\langle \mathbf{q}_n \mathbf{q}_n^\tau \right\rangle = \left\langle \begin{pmatrix} q_{nx} \\ q_{ny} \\ q_{nz} \end{pmatrix} \begin{pmatrix} q_{nx} & q_{ny} & q_{nz} \end{pmatrix} \right\rangle = \begin{pmatrix} \langle q_{nx}^2 \rangle & \langle q_{nx} q_{ny} \rangle & \langle q_{nx} q_{nz} \rangle \\ \langle q_{nx} q_{ny} \rangle & \langle q_{ny}^2 \rangle & \langle q_{ny} q_{nz} \rangle \\ \langle q_{nx} q_{nz} \rangle & \langle q_{ny} q_{nz} \rangle & \langle q_{nz}^2 \rangle \end{pmatrix} \tag{2.3}$$

In (2.3) the symbol $\langle\ \rangle$ means the time average and index *n* in U_n means that each atom is related to its matrix of displacements. This matrix is non-negative definite, *i.e.* $\left(\mathbf{q}^\tau U_n \mathbf{q}\right) \geq 0$ for any vector $\mathbf{q}$.

If a group of atoms moves together oscillating as a rigid body then atomic displacements $\mathbf{q}_n$ for different atoms are not independent but are expressed through some common parameters and the corresponding U_n also can be expressed as a combination of several common matrices. Here we are

going to show corresponding relations and their derivation. Obviously, such a description of an atomic group as a rigid body is only an approximation; this problem and ways to take next-level details into account were discussed for example by Johnson (1970b), Scheringer (1978b,c), Schomaker & Trueblood (1984), Dunitz *et al*. (1988), Bürgi (1989), Moore (2009) and others.

Obviously, a physical phenomenon – the probability of atomic displacement $\mathbf{q}_n$ in the crystal – is invariant with respect to the mathematical description used to describe it. This means that if we change the coordinate system, the coordinates of $\mathbf{q}_n$ change, so the matrix U_n should change accordingly, following some rules. Appendices A1 and A2 remind these basic rules, without going into advanced definitions of quadratic forms, tensors and other mathematical entities.

2.2. Rotation axes and their parameterisation

To define a displacement of a rigid body with respect to its original position, one needs to know a translation vector, a rotation axis and a rotation angle. An axis may be characterised by a unit vector ***l*** = (l_x, l_y, l_z) along it and by a point **w** = (w_x, w_y, w_z) that belongs to the axis.

In order to define a position of the rotation axis parallel, for example, to the coordinate axis **k** (figure 2.1a), two parameters are sufficient. For example these may be Cartesian coordinates of the point **w** = $(w_x, w_y, 0)$ in which this axis crosses the plane **Oij**. Alternatively, these may be polar coordinates R and α of this point. Note that with such a choice **Ow** by construction is normal to **k** and therefore **w** is the closest to the origin **O** among all points of the axis.

For an arbitrary axis ***l*** that passes through the origin **O** (figure 2.1b) two parameters are sufficient to define its direction. For example these may be two polar angles: the angle ψ between ***l*** and the **k** axis and the angle φ between **i** and the projection of ***l*** into the plane **Oij**.

To fully define an axis in an arbitrary position and not crossing the origin **O** (figure 2.1c), two parameters are required for its orientation, for example ψ and φ as above and two more for its position (***l*** defines unambiguously the plane normal to it and containing the origin **O**; in this plane

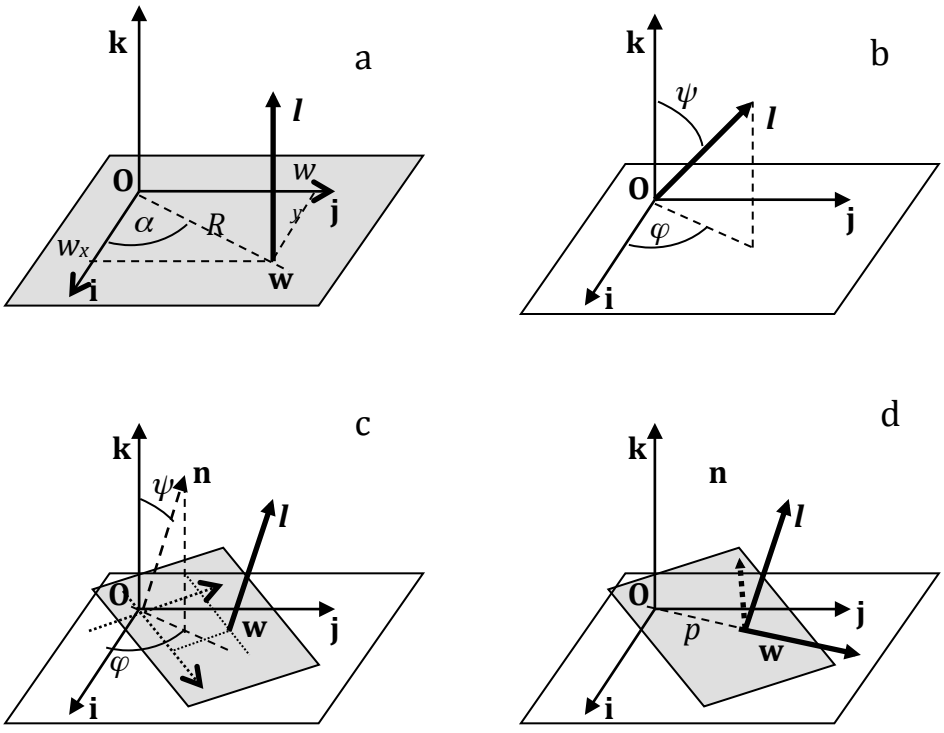


Figure 2.1. Schematic illustration of the definition of a rotation axis ***l***. (a) Axis parallel to **k**; its position is defined by 2 parameters, either by w_x and w_y, or by R and α. (b) Axis in an arbitrary orientation crossing the origin; its orientation is defined by 2 angles. (c) Axis in a general position; **n** is the vector parallel to ***l*** and crossing the origin; the plane normal to **n** and ***l*** is in grey. Similar to (b), two parameters are sufficient to define orientation of **n** and ***l***; to define the position of ***l*** two more parameters are required as in (a); they are coordinates of the intersection **w** of ***l*** with the 'grey' plane. (d) Normalized **Ow** and ***l*** are considered as the rotated **i** and **k**; p is the distance |**Ow**|.

the two coordinates, Cartesian or polar, of the point **w** in which ***l*** crosses the plane define its position). Obviously other points **w** may be chosen as discussed later.

Alternatively, the sufficiency of 4 parameters can be understood as following. Let **w** be a point on the rotation axis ***l*** such that the vectors ***l*** and **Ow** are orthogonal; in other words **w** is in the plane normal to ***l*** and crossing the origin (figure 2.1d). Then three Euler angles are sufficient to describe the orientation of **Ow** and ***l*** (for example a rotation of the base vectors so that rotated **i** coincides with **w** and rotated **k** coincides with ***l***) and the forth parameter is the shift p of the ***l*** axis along the direction **Ow**.

2.3. Rotation: linear approximation

When considering a libration of a body around a given axis, a displacement of each point may be expanded in series on the rotation angle δ. Eventually, high-order series may be considered; see for example, Johnson & Levy (1974), Johnson (1980), Coppens (2006) and references therein. Tickle & Moss (1999) mentioned that "*The harmonic model is applicable only if the motion is purely translational, but provided the libration amplitudes are not too large it is a good approximation*". Cruickshank (1956c) gave a value of 8° (0.14 radians) for the oscillation amplitude as a limit of this approximation.

In particular, working with small libration amplitudes means that a displacement towards the rotation axis is of a next order of magnitude than the displacement normal to the axis. For example, when the body is rotated around the coordinate axis **k** by an angle δ, the point **r** with the coordinates (1,0,0) gets the coordinates ($\cos\delta$, $\sin\delta$, 0). Replacing the exact displacement ($\cos\delta$-1, $\sin\delta$, 0) by its *linear approximation* **v** we neglect all terms starting from δ^2 in the Taylor series for cosine and sine, and the approximate coordinates of the shift are

$$(\cos\delta\,, \sin\delta, 0) - (1,0,0) \approx (0, \delta, 0) = \mathbf{v} \qquad (2.4)$$

More generally, for small angles δ any point positioned at the distance $R = 1$ from the rotation axis is displaced by a distance $d \approx R\delta = \delta$ (in radians). In these notes we use this parameter d instead of δ to define the rotation amplitude.

Also within a linear approximation, the point $(x_n,0,0)$ as well as $(x_n,0,z_n)$ are shifted by the vector $(0, x_n d, 0)$. Similarly, the point $(0,y_n,z_n)$ is shifted by $(-y_n d, 0, 0)$ and a general-position point $\mathbf{r}_n = (x_n,y_n,z_n)$ is shifted by

$$\mathbf{q}_n \approx \mathbf{v}_n = (-y_n d, x_n d, 0) = d\left[\mathbf{k} \times \mathbf{r}_n\right] \qquad (2.5)$$

where × is a vector cross product.

For a rotation around an arbitrary axis ***l*** crossing the origin, a linear approximation $\mathbf{v}_n$ to the displacement $\mathbf{q}_n$ of a point $\mathbf{r}_n$ may be expressed as

$$\mathbf{v}_n = d\left[\boldsymbol{l} \times \mathbf{r}_n\right] \qquad (2.6)$$

The latter is often presented in an equivalent form

$$\mathbf{v}_n = dA_n\boldsymbol{l} \qquad (2.7)$$

with the matrix

$$A_n = \begin{pmatrix} 0 & z_n & -y_n \\ -z_n & 0 & x_n \\ y_n & -x_n & 0 \end{pmatrix} \qquad (2.8)$$

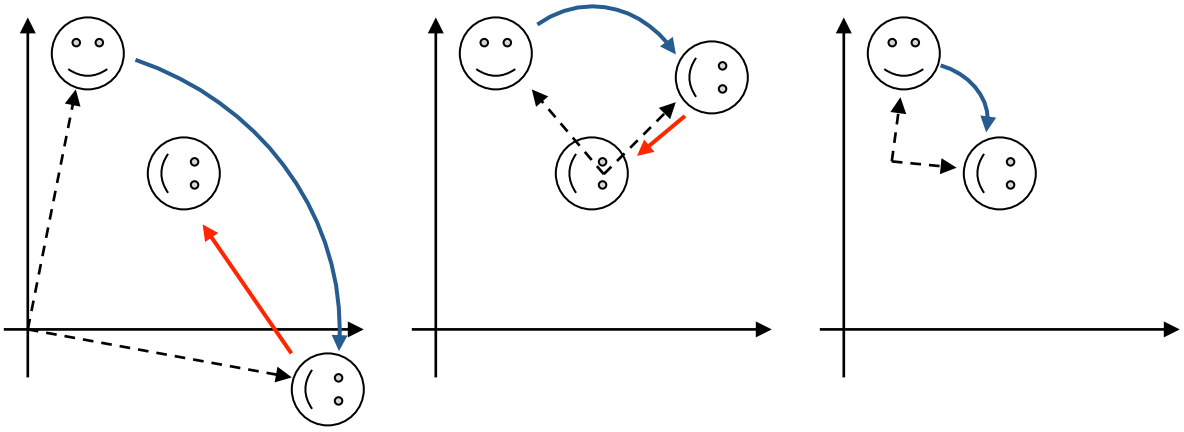


Figure 2.2. Three examples of possible combinations of a rotation (blue arrow) followed by a translation (red arrow) corresponding to the same transformation. The dashed lines show the rotation radius and indicate the rotation axis, always perpendicular to the plane of the page. Left image: the axis crosses the origin. Central image: the axis crosses the final object; corresponding translation is smaller than at the left image. Right image: the same transformation is presented as a pure rotation.

expressed through the Cartesian coordinates (x_n, y_n, z_n) of the point $\mathbf{r}_n$ in the coordinate system defined above (section 2.1). If $\boldsymbol{l}$ passes through a point $\mathbf{w} = (w_x, w_y, w_z)$ and not through the origin $\mathbf{O}$ (see for example section 2.2), vector $\mathbf{v}_n$ becomes

$$\mathbf{v}_n = d[\boldsymbol{l} \times (\mathbf{r}_n - \mathbf{w})] = d[\boldsymbol{l} \times \mathbf{r}_n] - d[\boldsymbol{l} \times \mathbf{w}] = dA_n \boldsymbol{l} - d\boldsymbol{l} \times \mathbf{w} \qquad (2.9)$$

Here $\boldsymbol{l} \times \mathbf{w}$ is a given vector, the same for all points of the oscillating rigid body; on the contrary $dA_n \boldsymbol{l}$ is point-dependent in the same fashion as A_n.

One can note that (2.9) presents a rotation around an axis in a general position as a rotation around an axis at the origin followed by a translation $-d\boldsymbol{l} \times \mathbf{w}$ common for all points. Inversely, a rotation around an axis at the origin (or at any other point) followed by a translation normal to this axis can be always presented as a pure rotation by the same angle. The position of this rotation axis is unique. Figure 2.2 inspired by figure 2 of the *TLSView Manual* (Merritt, pymmlib.sourceforge.net/tlsview/tlsview.html) illustrates this fact.

2.4. Choice of the point at the axis

Obviously, the shift (2.9) is independent of the choice of the point at the rotation axis $\boldsymbol{l}$. If we substitute a point $\mathbf{w}$ by another point $\mathbf{w}' = (w'_x, w'_y, w'_z)$ at the same axis, this gives:

$$\mathbf{v}'_n = d[\boldsymbol{l} \times (\mathbf{r}_n - \mathbf{w}')] = d[\boldsymbol{l} \times (\mathbf{r}_n - \mathbf{w} + \mathbf{w} - \mathbf{w}')] =$$

$$= d[\boldsymbol{l} \times (\mathbf{r}_n - \mathbf{w})] + d[\boldsymbol{l} \times (\mathbf{w} - \mathbf{w}')] = d[\boldsymbol{l} \times (\mathbf{r}_n - \mathbf{w})] = \mathbf{v}_n \qquad (2.10)$$

since $\mathbf{w} - \mathbf{w}'$ is collinear to $\boldsymbol{l}$.

2.5. Non-linear effects

The (omitted) second-order term in the development of the libration-based shift (2.5) into the Taylor series on the rotation angle corresponds to the displacement toward the rotation axis. This term is

responsible for two effects. The first one is the curvature of electron density for individual atoms that appears as "*banana-shaped contours*" (Howlin *et al.*, 1989); some authors use the term "*boomerang shape*". To the best of our knowledge such effects were not reported in practical macromolecular studies.

The second effect is a shrinking of an apparent bond length, the distance between the centers of two covalently linked atoms as we see them in the electron density maps (obviously, of a high enough resolution), in comparison with the actual bond length. This problem was discussed by Cruickshank (1956c, 1961), Busing & Levy (1964), Schomaker & Trueblood (1968), Scheringer (1972a,b), Stuart & Phillips (1985), Dunitz *et al*. (1988). Similarly, Scheringer (1978a) and Haestier *et al*. (2008) discusses a modification of apparent bond angles. Cruickshank (1956c) and Haneef *et al*. (1985) estimated the bond-length correcting value as 0.010-0.015 Å. Howlin *et al*. (1989) showed some larger values, up to 0.06 Å. Interestingly, Burns *et al.* in 1967 wrote that "*... it has become fairly common practice at the end of a molecular crystal structure determination to analyze the anisotropic temperature parameters on the assumption that the molecule is rigid. Often the purpose is no more than the correction of bond lengths...*". Such examples can be found in Becka & Cruickshank (1961), Birnbaum (1972), Downs *et al*. (1992), Steiner & Saenger (1993), Dunitz (1999).

In these notes we stay within a linear approximation (2.5-2.9).

3. Special case 1: rigid body translation

The simplest case is a pure translational motion, *i.e.* an oscillation of a rigid body without rotation compound. For such a displacement all points of the body are shifted by the same vector $\mathbf{q}_n = \mathbf{u}$. For this particular motion we introduce a special notation for the matrix U_n, the same for all points of the group:

$$U_n = T = \begin{pmatrix} \langle u_x^2 \rangle & \langle u_x u_y \rangle & \langle u_x u_z \rangle \\ \langle u_x u_y \rangle & \langle u_y^2 \rangle & \langle u_y u_z \rangle \\ \langle u_x u_z \rangle & \langle u_y u_z \rangle & \langle u_z^2 \rangle \end{pmatrix} \qquad (3.1)$$

By definition, T is symmetric and therefore is defined by 6 elements, 3 at the diagonal and 3 off-diagonal.

Also by definition T is non-negative definite, therefore it has three non-negative eigenvalues corresponding to three mutually orthogonal eigenvectors. In the basis composed of these normalised eigenvectors $\mathbf{i}_t$, $\mathbf{j}_t$, $\mathbf{k}_t$, matrix T becomes

$$T_t = \begin{pmatrix} \langle t_x^2 \rangle & 0 & 0 \\ 0 & \langle t_y^2 \rangle & 0 \\ 0 & 0 & \langle t_z^2 \rangle \end{pmatrix} \qquad (3.2)$$

with the eigenvalues at the diagonal. They are variances of the displacement along these three new axes. Zero off-diagonal elements mean that these displacements are non-correlated with each other and that they can be used as three new parameters of the problem. Note that for an isotropic translational displacement with $\langle t_x^2 \rangle = \langle t_y^2 \rangle = \langle t_z^2 \rangle$ the matrix T is diagonal in any basis.

Since both the basis $(\mathbf{i}, \mathbf{j}, \mathbf{k})$ and the basis $(\mathbf{i}_t, \mathbf{j}_t, \mathbf{k}_t)$ are orthonormal, the transformation between them can be only a rotation. The corresponding transformation matrix R_t (see Appendix A1) can be defined by three rotation angles (see for example Urzhumtsev & Urzhumtseva (1997) for various

parameterisations used in crystallography for rotation matrices). Together with $\langle t_x^2 \rangle, \langle t_y^2 \rangle, \langle t_z^2 \rangle$ this makes the total number of parameters to be equal to 6 as above.

It follows from Appendix A2, coordinates (t_x, t_y, t_z) of a shift **u** in the basis $(\mathbf{i}_t, \mathbf{j}_t, \mathbf{k}_t)$ are linked to its coordinates (u_x, u_y, u_z) in the basis $(\mathbf{i}, \mathbf{j}, \mathbf{k})$ by relation

$$\begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} = R_t \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \tag{3.3}$$

and the matrix T in the initial base is

$$T = R_t \, T_t \, R_t^{\tau} \tag{3.4}$$

4. Special case 2: rotation axis parallel to k

4.1. Rotation around k axis

Now we consider a pure rotation of a rigid body with no translation component. For simplicity, we start from a rotation by a small angle δ around the axis **k**. As previously, if d_z is a random value for a shift of a point at a distance equal to 1 from **k** (section 2.3), its probability distribution defines a corresponding shift $\mathbf{q}_n \approx \mathbf{v}_n = (-y_n d_z, x_n d_z, 0)$ of a point $\mathbf{r}_n = (x_n, y_n, z_n)$ giving its matrix U_n (2.3) as

$$U_n = \begin{pmatrix} \langle q_{nx}^2 \rangle & \langle q_{nx} q_{ny} \rangle & \langle q_{nx} q_{nz} \rangle \\ \langle q_{nx} q_{ny} \rangle & \langle q_{ny}^2 \rangle & \langle q_{ny} q_{nz} \rangle \\ \langle q_{nx} q_{nz} \rangle & \langle q_{ny} q_{nz} \rangle & \langle q_{nz}^2 \rangle \end{pmatrix} = \begin{pmatrix} y_n^2 \langle d_z^2 \rangle & -x_n y_n \langle d_z^2 \rangle & 0 \\ -x_n y_n \langle d_z^2 \rangle & x_n^2 \langle d_z^2 \rangle & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$= \langle d_z^2 \rangle \begin{pmatrix} y_n^2 & -x_n y_n & 0 \\ -x_n y_n & x_n^2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{4.1}$$

The same results can be also obtained as

$$U_n = \langle \mathbf{q}_n \mathbf{q}_n^{\tau} \rangle = \langle (d_z A_n \mathbf{k})(d_z A_n \mathbf{k})^{\tau} \rangle = \langle d_z^2 A_n \mathbf{k} \mathbf{k}^{\tau} A_n^{\tau} \rangle = \langle d_z^2 \rangle A_n \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} A_n^{\tau} \tag{4.2}$$

For any atom in the rigid group the elements of the matrix in the right hand of expression (4.1) are actual atomic coordinates (as found in PDB file, for example; a better choice will be discussed below) and the random displacement of the rigid group is presented by a common factor $\langle d_z^2 \rangle$ that shows the amplitude of librations.

4.2. Roation axis parallel to k

When the rotation axis is parallel to **k** and passes through a point $\mathbf{w}^k = \left(w_x^k, w_y^k, 0\right)$ different from the origin, the corresponding shift (section 2.3) is

$$\mathbf{q}_n \approx \mathbf{v}_n = d_z A_n \mathbf{k} - d_z \mathbf{k} \times \mathbf{w} = d_z A_n \mathbf{k} - d_z W_k \mathbf{k} = d_z A_n \mathbf{k} + d_z W_k^\tau \mathbf{k} \tag{4.3}$$

with

$$W_k = \begin{pmatrix} 0 & 0 & -w_y^k \\ 0 & 0 & w_x^k \\ w_y^k & -w_x^k & 0 \end{pmatrix} \tag{4.4}$$

Here matrix W_k is introduced similarly to matrix A_n in (2.6)-(2.7), section 2.3. Similarly to (4.2), expression (4.3) leads to

$$U_n = \tag{4.5}$$

$$= \left\langle d^2 \right\rangle W_k^\tau \left(\mathbf{k}\mathbf{k}^\tau\right) W_k + \left\langle d^2 \right\rangle A_n \left(\mathbf{k}\mathbf{k}^\tau\right) A_n^\tau + \left\langle d^2 \right\rangle A_n \left(\mathbf{k}\mathbf{k}^\tau\right) W_k + \left\langle d^2 \right\rangle W_k^\tau \left(\mathbf{k}\mathbf{k}^\tau\right) A_n^\tau$$

Here

$$\mathbf{k}\mathbf{k}^\tau = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.6}$$

and

$$\mathbf{k}\mathbf{k}^\tau W_k = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ w_y^k & -w_x^k & 0 \end{pmatrix}, \qquad W_k^\tau \mathbf{k}\mathbf{k}^\tau = \begin{pmatrix} 0 & 0 & w_y^k \\ 0 & 0 & -w_x^k \\ 0 & 0 & 0 \end{pmatrix} \tag{4.7}$$

$$W_k^\tau \mathbf{k}\mathbf{k}^\tau W_k = \begin{pmatrix} \left(w_y^k\right)^2 & -w_x^k w_y^k & 0 \\ -w_x^k w_y^k & \left(w_x^k\right)^2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{4.8}$$

The first term in (4.5) is independent of the point $\mathbf{w}^k$ and corresponds to an apparent translation even when it was no translation in the initial description of the motion (see section 2.3 for similar examples).

5. Special case 3: rotation around k correlated with translation

5.1. Several examples

When in addition to the rotation around **k** (section 4.1) the body is undergoing a translation as described in section 3, the total distribution of the displacement of each atom depends on the correlation between rotation and translation, as well as on the direction of the translation. We start from a couple of simple illustrations. The displacement distribution can be represented by a surface on which the points have the same probability distribution; for harmonic oscillations this surface is called a

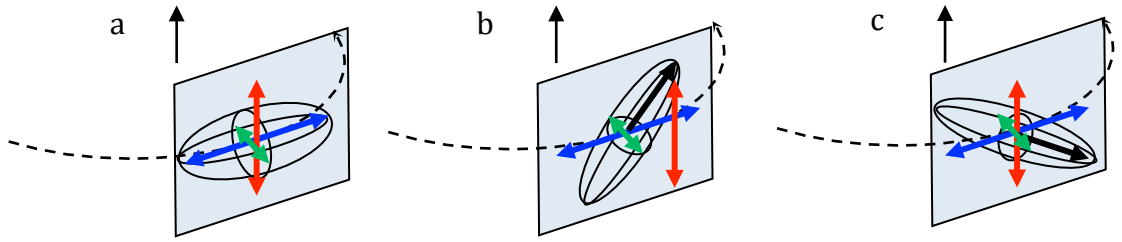


Figure 5.1. Schematic illustration of a libration around the vertical axis (an arrow at the center of circle) and a random translation. The motions are uncorrelated in the rotation plane (horizontal). Ellipses show surfaces of atomic displacement with the same probability. (a) The rotational displacement (blue) and translational displacement along the rotation axis (red) are uncorrelated; main elliptic axis is horizontal. (b) The displacements are positively correlated; the main elliptic axis (black bold) is in the plane formed by blue and red arrows and follows a right-hand helix. (c) The displacements are negatively correlated; similar to (b) but the main elliptic axis follows a left-hand helix. See Section 5.1 for more detail.

thermal ellipsoid.

Let's suppose that the translational displacement is isotropic. When rotation and translation are not correlated, the thermal motion ellipsoid in our example has one axis parallel to the axis **k** and two other axes normal to it (figure 5.1a).

When the displacement $(-y_n d_z, x_n d_z, 0)$ of a point (x_n, y_n, z_n) due to rotation is coupled (correlated) with its shift $(0,0,s_z d_z)$ *along the rotation axis*, the total linear displacement $(-y_n d_z, x_n d_z, s_z d_z)$ approximates an arch of a helix. The parameter s_z defines the slope of the trajectory with respect to the axis (figures 5.1b, c).

A superposition of a rotation and a correlated displacement in the direction *normal to the rotation axis* generates an apparent rotation axis shifted with respect to the original one, as discussed above in section 2.3. As an example (figure 5.2), let's consider again a rotation around the **k** axis, that generates

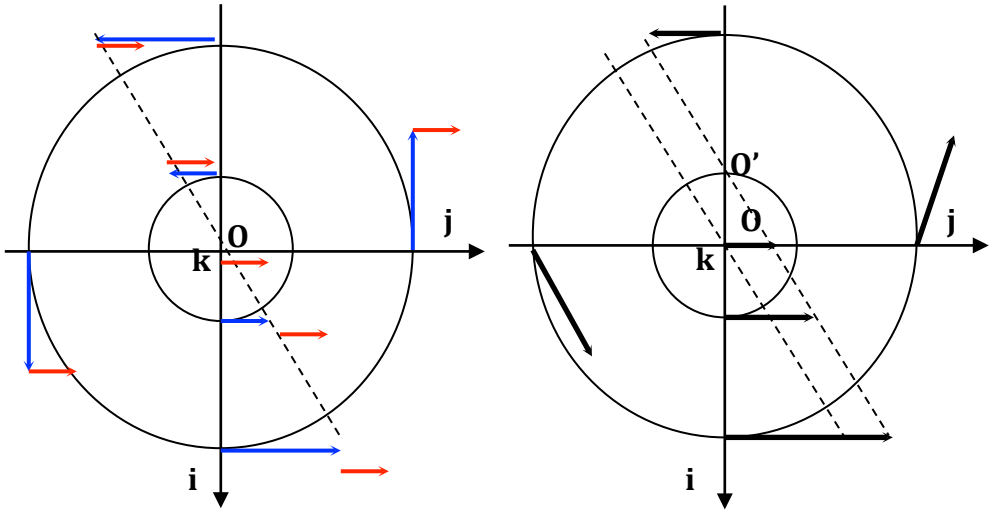


Figure 5.2. Schematic illustration of a correlation of a rotation around an axis **k** normal to the projection shown and a translation in the direction **j**. (a) displacements due to rotation (linear approximation, blue arrows) and translation (red arrows) are shown for several points; (b) total displacement (black arrows) and the shift of the rotation axis to its apparent position **O'**. See section 5.1 for more detail.

the shift $(-y_n d_z, x_n d_z, 0)$ for the points (x_n, y_n, z_n). If additionally we add a translation $(0, s_y d_z, 0)$, where s_y is some number, the total shift becomes $(-y_n d_z, (x_n + s_y) d_z, 0)$. This corresponds to a rotation around the axis parallel to **k** and crossing the point $(-s_y, 0, 0)$.

5.2. Screw axes along k

Let us analyse a screw rotation around an axis **k** more formally. As discussed in the previous section, the corresponding displacement of a point (x_n, y_n, z_n) is

$$\mathbf{q}_n = (-y_n d_z, x_n d_z, s_z d_z) = d_z A_n \mathbf{k} + s_z d_z \mathbf{k} \qquad (5.1)$$

The component $s_z d_z \mathbf{k}$ is the same for all points of the rigid body and it is a translation component of the group. Accordingly (2.3) the matrix U_n is

$$U_n = s_z^2 \langle d_z^2 \rangle (\mathbf{k}\mathbf{k}^\tau) + A_n \langle d_z^2 \rangle (\mathbf{k}\mathbf{k}^\tau) A_n^\tau + A_n (s_z \langle d_z^2 \rangle \mathbf{k}\mathbf{k}^\tau) + (s_z \langle d_z^2 \rangle \mathbf{k}\mathbf{k}^\tau) A_n^\tau \qquad (5.2)$$

One may note a similarity between (5.2) and (4.5). The first term in (5.2) is independent of the atomic coordinates and stands for the translation of the group. The second term depends, through matrix A_n, on the atomic coordinates quadratically and corresponds to the group rotation. Two last terms in (5.2) depend on atomic coordinates linearly and are due to the screw component. Following Schomaker & Trueblood (1968) we associate these terms with the matrices T, L and S that we define here as

$$T = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & s_z^2 \langle d_z^2 \rangle \end{pmatrix} \qquad (5.3)$$

$$L = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \langle d_z^2 \rangle \end{pmatrix} \qquad (5.4)$$

$$S = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & s_z \langle d_z^2 \rangle \end{pmatrix} \qquad (5.5)$$

With these matrices,

$$U_n = T + A_n L A_n^\tau + A_n S + (A_n S)^\tau \qquad (5.6)$$

5.3. *TLS* presentation

Now let's generalise the examples of sections 5.1 and 5.2. We keep the same notation and use **u** for translational displacement (section 3) and $\mathbf{v}_n$ for the displacement due to libration, always in a linear approximation (section 2.3).

For an atom *n* presented by its Cartesian coordinates (x_n, y_n, z_n) in the same basis as above the total

displacement vector

$$\mathbf{q}_n = \mathbf{u} + \mathbf{v}_n = \mathbf{u} + d_z A_n \mathbf{k} \tag{5.7}$$

has the coordinates

$$\begin{pmatrix} q_{nx} \\ q_{ny} \\ q_{nz} \end{pmatrix} = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} + \begin{pmatrix} -y_n d_z \\ x_n d_z \\ 0 \end{pmatrix} = \begin{pmatrix} u_x - y_n d_z \\ u_y + x_n d_z \\ u_z \end{pmatrix} \tag{5.8}$$

where d_z, as previously, defines the linear approximation $(0, d_z, 0)$ to the displacement of the point (1,0,0). Note that it follows from section 5.2, translation vector $\mathbf{u}$ may include the screw component and therefore be correlated with the rotation. Components of the matrix U_n (2.3) in the same coordinate system (**i**, **j**, **k**) are:

$$U_{nxx} = \left\langle \left(u_x - y_n d_z\right)^2 \right\rangle = \left\langle u_x^2 + y_n^2 d_z^2 - 2 y_n u_x d_z \right\rangle = \left\langle u_x^2 \right\rangle + y_n^2 \left\langle d_z^2 \right\rangle - 2 y_n \left\langle u_x d_z \right\rangle$$

$$U_{nyy} = \left\langle \left(u_y + x_n d_z\right)^2 \right\rangle = \left\langle u_y^2 + x_n^2 d_z^2 + 2 x_n u_y d_z \right\rangle = \left\langle u_y^2 \right\rangle + x_n^2 \left\langle d_z^2 \right\rangle + 2 x_n \left\langle u_y d_z \right\rangle$$

$$U_{nzz} = \left\langle u_z^2 \right\rangle$$

$$U_{nxy} = U_{nyx} = \left\langle \left(u_x - y_n d_z\right)\left(u_y + x_n d_z\right) \right\rangle = \left\langle u_x u_y - y_n u_y d_z + x_n u_x d_z - x_n y_n d_z^2 \right\rangle$$

$$= \left\langle u_x u_y \right\rangle - y_n \left\langle u_y d_z \right\rangle + x \left\langle u_x d_z \right\rangle - x_n y_n \left\langle d_z^2 \right\rangle$$

$$U_{nxz} = U_{nzx} = \left\langle \left(u_x - y_n d_z\right) u_z \right\rangle = \left\langle u_x u_z - y_n u_z d_z \right\rangle = \left\langle u_x u_z \right\rangle - y_n \left\langle u_z d_z \right\rangle$$

$$U_{nyz} = U_{nzy} = \left\langle \left(u_y + x_n d_z\right) u_z \right\rangle = \left\langle u_y u_z + x_n u_z d_z \right\rangle = \left\langle u_y u_z \right\rangle + x_n \left\langle u_z d_z \right\rangle \tag{5.9}$$

Similarly to section 5.2, we can obtain the *TLS* presentation

$$U_n = T + A_n L A_n^\tau + A_n S + S^\tau A_n^\tau \tag{5.10}$$

similar to (5.6). Currently,

$$T = U_T = \begin{pmatrix} \left\langle u_x^2 \right\rangle & \left\langle u_x u_y \right\rangle & \left\langle u_x u_z \right\rangle \\ \left\langle u_x u_y \right\rangle & \left\langle u_y^2 \right\rangle & \left\langle u_y u_z \right\rangle \\ \left\langle u_x u_z \right\rangle & \left\langle u_y u_z \right\rangle & \left\langle u_z^2 \right\rangle \end{pmatrix} \tag{5.11}$$

$$L = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \left\langle d_z^2 \right\rangle \end{pmatrix} \tag{5.12}$$

$$S = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \langle u_x d_z \rangle & \langle u_y d_z \rangle & \langle u_z d_z \rangle \end{pmatrix} \qquad (5.13)$$

and

$$A_n = \begin{pmatrix} 0 & z_n & -y_n \\ -z_n & 0 & x_n \\ y_n & -x_n & 0 \end{pmatrix} \qquad (5.14)$$

In presentation (5.10-5.14) matrices *T*, *L* and *S* are the same for all points of the rigid body while A_n is expressed through the point coordinates. This presentation can be obtained either by decomposition of (5.8) or by applying (2.3) to (5.7).

Expressions (5.11-5.13) show the 10 parameters common for the rigid group: 6, 1 and 3 parameters associated with *T*, *L* and *S*, respectively. Together with the matrices A_n, they fully define U_n for all atoms in the rigid group.

5.4. Origin shift

Obviously, matrices U_n should not depend on the choice of the origin of the coordinate system while matrices A_n do. This means that *T*, *L*, *S* or at least some of them, vary with the origin. Inversely, this means that some combinations of *T*, *L* or *S* may correspond to the same U_n but expressed in coordinate systems with different origins. We will demonstrate this relation that is very important for further analysis.

In a new coordinate system with the origin shifted by vector $\mathbf{p} = (p_x, p_y, p_z)$

$$\mathbf{O}' = \mathbf{O} + \mathbf{p} \qquad (5.15)$$

the new coordinates are $x_n - p_x, y_n - p_y, z_n - p_z$ defining the matrix

$$A'_n = \begin{pmatrix} 0 & z_n - p_z & -(y_n - p_y) \\ -(z_n - p_z) & 0 & x_n - p_x \\ y_n - p_y & -(x_n - p_x) & 0 \end{pmatrix} =$$

$$= \begin{pmatrix} 0 & z_n & -y_n \\ -z_n & 0 & x \\ y_n & -x_n & 0 \end{pmatrix} - \begin{pmatrix} 0 & p_z & -p_y \\ -p_z & 0 & p_x \\ p_y & -p_x & 0 \end{pmatrix} = A_n - P \qquad (5.16)$$

(we define the shift vector in the opposite way as Tickle & Moss (1999) do). Matrix *P* in (5.16) is the same for all points of the group. Accordingly to (5.10) matrix U_n becomes

$$U_n = T + (A'_n + P) L (A'_n + P)^{\tau} + (A'_n + P)\ S + S^{\tau} (A'_n + P)^{\tau}$$

$$= T + \left(A'_n L A'^{\tau}_n + A'_n L P^{\tau} + P L A'^{\tau}_n + P L P^{\tau} \right) + \left(A'_n S + P S \right) + \left(S^{\tau} A'^{\tau}_n + S^{\tau} P^{\tau} \right)$$

$$= \left(T + P L P^{\tau} + P S + S^{\tau} P^{\tau} \right) + A'_n L A'^{\tau}_n + A'_n \left(S + L P^{\tau} \right) + \left(S^{\tau} + P L^{\tau} \right) A'^{\tau}_n$$

$$= T' + A'_n L' A'^{\tau}_n + A'_n S' + S'^{\tau} A'^{\tau}_n = U'_n \qquad (5.17)$$

(at the last transition we substituted $P L A'^{\tau}_n$ by $P L^{\tau} A'^{\tau}_n$ using the symmetry $L = L^{\tau}$). Comparison with (5.10) defines new matrices T', L' and S' as

$$T' = T + \left(P L P^{\tau} + P S + S^{\tau} P^{\tau} \right)$$

$$L' = L \qquad (5.18)$$

$$S' = S + L P^{\tau}$$

It is very important that the expressions (5.18) were obtained for a general case of matrices *T*, *L* and *S* in (5.10) with no use of their specific form (5.11-5.13). A simplest example illustrating (5.18) is presented below.

Let's suppose that a rigid group oscillates around the axis **k**. In this case the only non-zero matrix is *L* (5.6) while *T* and *S* are zero. For the point **M** = (0,0,0) its $A_n = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ giving zero matrix U_n (a point sits at the rotation axis). Now let's choose another coordinate system shifting the origin by **p** = (1,0,0). The new coordinates of the point **M** are (-1,0,0), its new matrix $A'_n = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$ and matrix $P = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}$. New matrices are

$$S' = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \langle d_z^2 \rangle \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \langle d_z^2 \rangle & 0 \end{pmatrix}$$

$$A'_n S' = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \langle d_z^2 \rangle & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -\langle d_z^2 \rangle & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$T' = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \langle d_z^2 \rangle \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \langle d_z^2 \rangle & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$A'_n L' A'^{\tau}_n = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \langle d_z^2 \rangle \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \langle d_z^2 \rangle & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

New matrix $U'_n = T' + A'_n L' A'^{\tau}_n + A'_n S' + S'^{\tau} A'^{\tau}_n$ is a zero matrix, as it should be.

5.5. Search for the apparent rotation axis

Applying (5.18) to the special case (5.11-5.13) of a rotation around **k** gives

$$LP^{\tau} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -p_y \langle d_z^2 \rangle & p_x \langle d_z^2 \rangle & 0 \end{pmatrix} \tag{5.19}$$

$$S + LP^{\tau} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \langle u_x d_z \rangle - p_y \langle d_z^2 \rangle & \langle u_y d_z \rangle + p_x \langle d_z^2 \rangle & \langle u_z d_z \rangle \end{pmatrix} \tag{5.20}$$

$$PLP^{\tau} = \begin{pmatrix} p_y^2 \langle d_z^2 \rangle & -p_x p_y \langle d_z^2 \rangle & 0 \\ -p_x p_y \langle d_z^2 \rangle & p_x^2 \langle d_z^2 \rangle & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{5.21}$$

$$PS = \begin{pmatrix} -p_y \langle u_x d_z \rangle & -p_y \langle u_y d_z \rangle & -p_y \langle u_z d_z \rangle \\ p_x \langle u_x d_z \rangle & p_x \langle u_y d_z \rangle & p_x \langle u_z d_z \rangle \\ 0 & 0 & 0 \end{pmatrix} \tag{5.22}$$

$$T + PLP^{\tau} + PS + S^{\tau} P^{\tau} = \tag{5.23}$$

$$= \begin{pmatrix} p_y^2 \langle d_z^2 \rangle - 2p_y \langle u_x d_z \rangle + \langle u_x^2 \rangle & \left\{ \begin{matrix} p_x \langle u_x d_z \rangle - p_y \langle u_y d_z \rangle \\ p_x p_y \langle d_z^2 \rangle - \langle u_x u_y \rangle \end{matrix} \right\} & -p_y \langle u_z d_z \rangle + \langle u_x u_z \rangle \\ \left\{ \begin{matrix} p_x \langle u_x d_z \rangle - p_y \langle u_y d_z \rangle \\ -p_x p_y \langle d_z^2 \rangle + \langle u_x u_y \rangle \end{matrix} \right\} & p_x^2 \langle d_z^2 \rangle + 2p_x \langle u_y d_z \rangle + \langle u_y^2 \rangle & p_x \langle u_z d_z \rangle + \langle u_y u_z \rangle \\ -p_y \langle u_z d_z \rangle + \langle u_x u_z \rangle & p_x \langle u_z d_z \rangle + \langle u_y u_z \rangle & \langle u_z^2 \rangle \end{pmatrix}$$

Expression (5.23) shows existence of a special origin

$$p_y = \frac{\langle u_x d_z \rangle}{\langle d_z^2 \rangle}, \quad p_x = -\frac{\langle u_y d_z \rangle}{\langle d_z^2 \rangle} \qquad . \qquad (5.24)$$

that minimises the diagonal elements of T' making this matrix equal to

$$T' = \begin{pmatrix} \langle u_x^2 \rangle - \frac{\langle u_x d_z \rangle^2}{\langle d_z^2 \rangle} & \langle u_x u_y \rangle - \frac{\langle u_x d \rangle \langle u_y d_z \rangle}{\langle d_z^2 \rangle} & \langle u_x u_z \rangle - \frac{\langle u_x d_z \rangle \langle u_z d_z \rangle}{\langle d_z^2 \rangle} \\ \langle u_x u_y \rangle - \frac{\langle u_x d_z \rangle \langle u_y d_z \rangle}{\langle d_z^2 \rangle} & \langle u_y^2 \rangle - \frac{\langle u_y d_z \rangle^2}{\langle d_z^2 \rangle} & \langle u_y u_z \rangle - \frac{\langle u_y d_z \rangle \langle u_z d_z \rangle}{\langle d_z^2 \rangle} \\ \langle u_x u_z \rangle - \frac{\langle u_x d_z \rangle \langle u_z d_z \rangle}{\langle d_z^2 \rangle} & \langle u_y u_z \rangle - \frac{\langle u_y d_z \rangle \langle u_z d_z \rangle}{\langle d_z^2 \rangle} & \langle u_z^2 \rangle \end{pmatrix} \qquad (5.25)$$

The minimisation of all diagonal elements T_{xx}, T_{yy}, T_{zz} at a time can be reformulated as the minimisation of the trace of the matrix, $tr(T) = T_{xx} + T_{yy} + T_{zz} \rightarrow \min\limits_{p_x, p_y, p_z}$, which stays non-negative due to the Cauchy-Schwarz inequality. In fact, this minimisation means that the new origin is at the apparent rotation axis, as discussed in the examples of sections 2.3, 4.2 and 5.1 (see also Schomaker & Trueblood, 1968; Pawley, 1970; Tickle & Moss, 1999). This is also confirmed by (5.20) that becomes

$$S' = S + LH^{\tau} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \langle u_z d_z \rangle \end{pmatrix} \qquad (5.26)$$

showing no correlation between rotation-translation displacements in the plane normal to the rotation axis (we remind the reader that in this example case it is the axis $\mathbf{k}$). The results above are independent of the choice of p_z (a shift along the rotation axis).

Relocation of the translation component by including it into the displacement of the rotation axis is opposite to an operation discussed in section 3: an apparent translation component for the axes different from the coordinate ones. Comparison of (5.24) with (4.4) shows their similarity in determination of the position of the rotation axis.

5.6. Parameters with a physical meaning

Following from section 3, one may define the elements of *T* through its eigenvalues (uncorrelated translations) $\langle t_x^2 \rangle, \langle t_y^2 \rangle, \langle t_z^2 \rangle$ and the rotation matrix R_t. One may note also that due to relation (3.3)

$$\begin{pmatrix} \langle u_x d_z \rangle \\ \langle u_y d_z \rangle \\ \langle u_z d_z \rangle \end{pmatrix} = R_t \begin{pmatrix} \langle t_x d_z \rangle \\ \langle t_y d_z \rangle \\ \langle t_z d_z \rangle \end{pmatrix} \qquad (5.27)$$

where $\langle t_x d_z \rangle, \langle t_y d_z \rangle, \langle t_z d_z \rangle$ describe correlation of mutually uncorrelated random displacements (t_x, t_y,

t_z) with the displacement due to libration around the axis **k** and the matrix R_t is the same as before.

As a next step, one may introduce the correlations

$$-1 \le c_x = \frac{\langle t_x d_z \rangle}{\sqrt{\langle t_x^2 \rangle}\sqrt{\langle d_z^2 \rangle}} \le 1\,,\ -1 \le c_y = \frac{\langle t_y d_z \rangle}{\sqrt{\langle t_y^2 \rangle}\sqrt{\langle d_z^2 \rangle}} \le 1,\ -1 \le c_z = \frac{\langle t_z d_z \rangle}{\sqrt{\langle t_z^2 \rangle}\sqrt{\langle d_z^2 \rangle}} \le 1 \tag{5.28}$$

as independent parameters instead of $\langle u_x d_z \rangle, \langle u_y d_z \rangle, \langle u_z d_z \rangle$, resulting in another set of parameters: $\langle t_x^2 \rangle, \langle t_y^2 \rangle, \langle t_z^2 \rangle, \langle d_z^2 \rangle$, c_x, c_y, c_z and 3 mutually orthogonal directions of uncorrelated translations described through R_t by three Euler angles; this makes ten parameters in total as previously. Such a choice has an advantage that all corresponding parameters have a clear physical meaning.

6. Special case 4: three rotation axes parallel to ijk

6.1. Uncorrelated pure rotations

Now we will extend the analysis done in section 4.2. When a rotation axis is parallel to **i** or **j** instead of **k**, the resulting matrix has always the form (4.5) in which W_k and $\mathbf{k}\mathbf{k}^\tau$ (4.4 and 4.6) are replaced by

$$W_i = \begin{pmatrix} 0 & w_z^i & -w_y^i \\ -w_z^i & 0 & 0 \\ w_y^i & 0 & 0 \end{pmatrix} \quad \text{and} \quad \mathbf{i}\mathbf{i}^\tau = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{6.1}$$

for rotations around **i** or by

$$W_j = \begin{pmatrix} 0 & w_z^j & 0 \\ -w_z^j & 0 & w_x^j \\ 0 & -w_x^j & 0 \end{pmatrix} \quad \text{and} \quad \mathbf{j}\mathbf{j}^\tau = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{6.2}$$

for rotations around **j**, respectively.

When three rotations around the axes parallel to **i**, **j** and **k** with the corresponding amplitudes d_x, d_y and d_z are executed simultaneously the resulting shift is

$$\mathbf{q}_n \approx \mathbf{v}_n = \left(d_x A_n \mathbf{i} + d_x W_i^\tau \mathbf{i}\right) + \left(d_y A_n \mathbf{j} + d_y W_j^\tau \mathbf{j}\right) + \left(d_z A_n \mathbf{k} + d_z W_k^\tau \mathbf{k}\right) \tag{6.3}$$

For uncorrelated rotations, *i.e.* such that

$$\langle d_x d_y \rangle = \langle d_x d_z \rangle = \langle d_y d_z \rangle = 0, \tag{6.4}$$

a calculation similar to (4.3)-(4.8) gives the matrix U_n in the form (5.10) with

$$T = \tag{6.5}$$

$$= \begin{pmatrix} \left(w_z^j\right)^2\left\langle d_y^2\right\rangle + \left(w_y^k\right)^2\left\langle d_z^2\right\rangle & -w_x^k w_y^k\left\langle d_z^2\right\rangle & -w_x^j w_z^j\left\langle d_y^2\right\rangle \\ -w_x^k w_y^k\left\langle d_z^2\right\rangle & \left(w_z^i\right)^2\left\langle d_x^2\right\rangle + \left(w_x^k\right)^2\left\langle d_z^2\right\rangle & -w_y^i w_z^i\left\langle d_x^2\right\rangle \\ -w_x^j w_z^j\left\langle d_y^2\right\rangle & -w_y^i w_z^i\left\langle d_x^2\right\rangle & \left(w_y^i\right)^2\left\langle d_x^2\right\rangle + \left(w_x^j\right)^2\left\langle d_y^2\right\rangle \end{pmatrix}$$

$$L = \begin{pmatrix} \left\langle d_x^2\right\rangle & 0 & 0 \\ 0 & \left\langle d_y^2\right\rangle & 0 \\ 0 & 0 & \left\langle d_z^2\right\rangle \end{pmatrix} \tag{6.6}$$

$$S = \begin{pmatrix} 0 & w_z^i\left\langle d_x^2\right\rangle & -w_y^i\left\langle d_x^2\right\rangle \\ -w_z^j\left\langle d_y^2\right\rangle & 0 & w_x^j\left\langle d_y^2\right\rangle \\ w_y^k\left\langle d_z^2\right\rangle & -w_x^k\left\langle d_z^2\right\rangle & 0 \end{pmatrix} \tag{6.7}$$

As previously, T corresponds to the apparent translation term, the same for all points.

6.2. Screw rotations around the coordinate axes

Following from the calculations in section 5.2, one can derive that three simultaneous uncorrelated rotations around the three coordinate axes with the amplitudes d_x, d_y, d_z and screw components s_x, s_y, s_z give U_n in the form (5.10) with the matrices T, L and S:

$$T = \begin{pmatrix} s_x^2\left\langle d_x^2\right\rangle & 0 & 0 \\ 0 & s_y^2\left\langle d_y^2\right\rangle & 0 \\ 0 & 0 & s_z^2\left\langle d_z^2\right\rangle \end{pmatrix} \tag{6.8}$$

$$L = \begin{pmatrix} \left\langle d_x^2\right\rangle & 0 & 0 \\ 0 & \left\langle d_y^2\right\rangle & 0 \\ 0 & 0 & \left\langle d_z^2\right\rangle \end{pmatrix} \tag{6.9}$$

$$S = \begin{pmatrix} s_x\left\langle d_x^2\right\rangle & 0 & 0 \\ 0 & s_y\left\langle d_y^2\right\rangle & 0 \\ 0 & 0 & s_z\left\langle d_z^2\right\rangle \end{pmatrix} \tag{6.10}$$

It may be useful to compare (6.8)-(6.10) with (6.5)-(6.7).

7. Rotation around an axis in a general position

7.1. Rotation around a fixed bond

A libration of an atomic group around a given axis plays a special role in macromolecular modeling where dihedral angles are relatively flexible compared to bond angles and lengths. This may be a libration of a peptide side chain around $C_\alpha C_\beta$ bond (see figure 7.1 for illustration) or, in general, a

libration of an atomic group or a domain around a bond between two given atoms. Detailed studies of a rotation around a bond can be found in Prince & Finger (1972), Dunitz & White (1973), Sygusch (1976) and Schomaker & Trueblood (1998).

In this section, let's consider a rotation around the vector **g** between two fixed points $\mathbf{G}_1 = (G_{1x}, G_{1y}, G_{1z})$ and $\mathbf{G}_2 = (G_{2x}, G_{2y}, G_{2z})$, thus **g** being fixed as well. In figure 7.2 point $\mathbf{G}_1$ correspond to C_α and $\mathbf{G}_2$ corresponds to C_β when the peptide group is fixed. It is trivial to express a unit vector $\boldsymbol{l} = (l_x, l_y, l_z)$ along the rotation axis through the coordinates of the two chosen points:

$$\mathbf{g} = (g_x, g_y, g_z) = (G_{2x} - G_{1x}, G_{2y} - G_{1y}, G_{2z} - G_{1z}) \tag{7.1}$$

$$\boldsymbol{l} = \mathbf{g} / \|\mathbf{g}\| = \mathbf{g} / \sqrt{g_x^2 + g_y^2 + g_z^2} \tag{7.2}$$

The point **w** at the rotation axis can be taken for example as

$$\mathbf{w} = (w_x, w_y, w_z) = (G_{1x}, G_{1y}, G_{1z}) \tag{7.3}$$

We remind that the result is independent of the choice of a point at the rotation axis, see section 2.4. We remind the reader also that 4 parameters and not 6 (coordinates of the two points) are sufficient to define a rotation axis (section 2.2).

The shift $\mathbf{q}_n$ of a point (x_n, y_n, z_n) is defined now as (see section 2.3)

$$\begin{pmatrix} q_{nx} \\ q_{ny} \\ q_{nz} \end{pmatrix} = dA_n \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix} - dW \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix} = dA_{nw} \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix} \tag{7.4}$$

where

$$W = \begin{pmatrix} 0 & w_z & -w_y \\ -w_z & 0 & w_x \\ w_y & -w_x & 0 \end{pmatrix}$$

$$A_{nw} = \begin{pmatrix} 0 & z_n - w_z & -(y_n - w_y) \\ -(z_n - w_z) & 0 & x_n - w_x \\ y_n - w_y & -(x_n - w_x) & 0 \end{pmatrix} \tag{7.5}$$

and d is a random parameter describing the amplitude of libration. Similarly to (4.2) if in (7.4) we use the single-term presentation through A_{nw}, the averaging of $\mathbf{q}_n \mathbf{q}_n^\tau$ gives

$$U_n = \left\langle \mathbf{q}_n \mathbf{q}_n^\tau \right\rangle = A_{nw} L A_{nw}^\tau = A_{nw} L_d A_{nw}^\tau \tag{7.6}$$

with

$$L = \left\langle d^2 \right\rangle \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix} \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix}^{\tau} = \left\langle d^2 \right\rangle \begin{pmatrix} l_x^2 & l_x l_y & l_x l_z \\ l_x l_y & l_y^2 & l_y l_z \\ l_x l_z & l_y l_z & l_z^2 \end{pmatrix} = \left\langle d^2 \right\rangle L_d \qquad (7.7)$$

Here $\left\langle d^2 \right\rangle$ characterises the random distribution and size of the libration angle. This is the single *random* parameter *to be adjusted* to the experimental data. As mentioned above, six *fixed* parameters, 4 of them being independent, are used to define matrices L_d.

In this simplest situation using the two-terms presentation in (7.4) requires more computing than that with A_{nw}.

7.2. Coordinate system aligned with the bond

In this trivial case of a libration around a single fixed axis a direct approach (7.5-7.7) with no intermediate coordinate systems seems to be preferable for practical applications. However, the procedure descried in this section may be useful to understand more complex situations.

Using matrix A_{nw} (7.5) instead of A_n (7.4) in fact means a "hidden" change of the origin of the coordinate system. This eliminates matrices T and S which are unnecessary in this trivial case. We may further modify the coordinate system by choosing new base vectors $(\mathbf{i}_l, \mathbf{j}_l, \mathbf{k}_l)$ such that the new vector $\mathbf{k}_l = \boldsymbol{l}$. To do so, we define the angle ψ between $\boldsymbol{l}$ and the axis $\mathbf{k}$

$$\psi = \arccos(\mathbf{kl}) = \arccos(l_z) \qquad (7.8)$$

and the angle φ between the axis $\mathbf{i}$ and the projection $\mathrm{l}_{xy} = (l_x, l_y, 0)$ of $\boldsymbol{l}$ into the plane **Oij** :

$$\cos(\varphi) = l_x \Big/ \sqrt{l_x^2 + l_y^2} \quad , \quad \sin(\varphi) = l_y \Big/ \sqrt{l_x^2 + l_y^2} \qquad (7.9)$$

(for illustration see figure 2.1b). Now matrix

$$R_l = \begin{pmatrix} \cos\varphi\cos\psi & -\sin\varphi & \cos\varphi\sin\psi \\ \sin\varphi\cos\psi & \cos\varphi & \sin\varphi\sin\psi \\ -\sin\psi & 0 & \cos\psi \end{pmatrix} \qquad (7.10)$$

describes the transformation of the original coordinate system (Appendix A1) with the base vectors $(\mathbf{i}, \mathbf{j}, \mathbf{k})$ into an intermediate system with the base vectors $(\mathbf{i}_l, \mathbf{j}_l, \mathbf{k}_l)$, in particular

$$\begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix} = R_l \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \qquad (7.11)$$

and inversely

$$\begin{pmatrix}0\\0\\1\end{pmatrix} = R_l^{-1}\begin{pmatrix}l_x\\l_y\\l_z\end{pmatrix} \qquad (7.12)$$

The coordinates in the new system are calculated from the original coordinates of a point as

$$\begin{pmatrix}x'_n\\y'_n\\z'_n\end{pmatrix} = R_l^{-1}\begin{pmatrix}x_n - C_{1x}\\y_n - C_{1y}\\z_n - C_{1z}\end{pmatrix} = \begin{pmatrix}\cos\psi\cos\varphi & \cos\psi\sin\varphi & -\sin\psi\\ -\sin\varphi & \cos\varphi & 0\\ \sin\psi\cos\varphi & \sin\psi\sin\varphi & \cos\psi\end{pmatrix}\begin{pmatrix}x_n - C_{1x}\\y_n - C_{1y}\\z_n - C_{1z}\end{pmatrix} \qquad (7.13)$$

and matrix U'_n (see (4.2)) is

$$U'_n = \langle d^2\rangle A'_n\begin{pmatrix}0&0&0\\0&0&0\\0&0&1\end{pmatrix}A'^{\tau}_n = \langle d^2\rangle\begin{pmatrix}y'^2_n & -x'_n y'_n & 0\\ -x'_n y'_n & x'^2_n & 0\\ 0&0&0\end{pmatrix} \qquad (7.14)$$

resulting in (Appendix A2)

$$U_n = \langle d^2\rangle R_l\begin{pmatrix}y'^2_n & -x'_n y'_n & 0\\ -x'_n y' & x'^2_n & 0\\ 0&0&0\end{pmatrix}R_l^{\tau} \qquad (7.15)$$

7.3. Axis with the fixed direction

Now let's suppose that the two points defining the libration axis $\boldsymbol{l}$ (*e.g.* C_α and C_β atoms in figure 7.1) oscillate around their central positions. If (for simplicity) we consider the direction of the axis being fixed (for example when the distance between the atoms is much larger than their displacements; see figure 7.2 for another illustration), the displacement (7.4) with the definitions (7.5) becomes

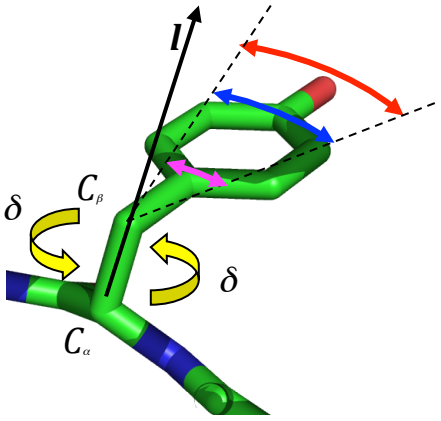


Figure 7.1. Schematic illustration of a libration around a bond. Angle δ is the parameter describing random oscillations around the bond $C_\alpha C_\beta$. A shift of each atom is proportional to its distance to this rotation axis.

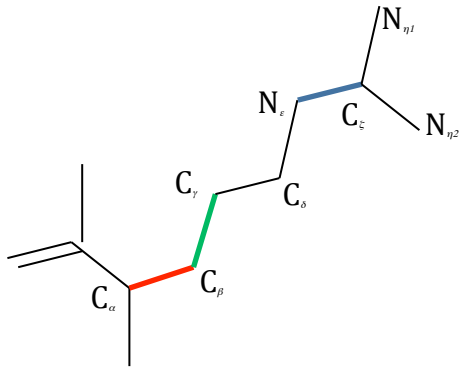


Figure 7.2. Schematic illustration of different kinds of libration axes associated with interatomic bonds. The peptide group is considered as a fixed. The rotation axis $C_\alpha C_\beta$ is a fixed axis (see Sections 7.1-7.2). The rotation axis $C_\beta C_\gamma$ changes its orientation (Section 7.5). If $N_\varepsilon C_\xi$ and $C_\alpha C_\beta$ are roughly parallel, the libration around $C_\alpha C_\beta$ translates the bond $N_\varepsilon C_\xi$ rather than changes its orientation (Sections 7.3-7.4).

$$\begin{pmatrix} q_{nx} \\ q_{ny} \\ q_{nz} \end{pmatrix} = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} + dA_n \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix} - dW \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix} = \begin{pmatrix} \hat{u}_x \\ \hat{u}_y \\ \hat{u}_z \end{pmatrix} + dA_n \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix} \qquad (7.16)$$

where we introduce a new random vector, the same for all points of the rigid group

$$\begin{pmatrix} \hat{u}_x \\ \hat{u}_y \\ \hat{u}_z \end{pmatrix} = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} - dW \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix} \qquad (7.17)$$

Differently from section 7.1, here the two-terms presentation is preferable and the averaging of $\mathbf{q}_n\mathbf{q}_n^\tau$ gives a sum

$$U_n = \left\langle \mathbf{q}_n\mathbf{q}_n^\tau \right\rangle = T + A_n L A_n^\tau + \left(A_n S + S^\tau A_n^\tau \right) \qquad (7.18)$$

always in the same form as (5.10) where the matrix A_n in the base $(\mathbf{i},\mathbf{j},\mathbf{k})$ is always (2.7) and other matrices are

$$T = \begin{pmatrix} \left\langle \hat{u}_x^2 \right\rangle & \left\langle \hat{u}_x\hat{u}_y \right\rangle & \left\langle \hat{u}_x\hat{u}_z \right\rangle \\ \left\langle \hat{u}_x\hat{u}_y \right\rangle & \left\langle \hat{u}_y^2 \right\rangle & \left\langle \hat{u}_y\hat{u}_z \right\rangle \\ \left\langle \hat{u}_x\hat{u}_z \right\rangle & \left\langle \hat{u}_y\hat{u}_z \right\rangle & \left\langle \hat{u}_z^2 \right\rangle \end{pmatrix}$$

$$L = \left\langle d^2 \right\rangle \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix} \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix}^\tau = \left\langle d^2 \right\rangle \begin{pmatrix} l_x^2 & l_x l_y & l_x l_z \\ l_x l_y & l_y^2 & l_y l_z \\ l_x l_z & l_y l_z & l_z^2 \end{pmatrix} \qquad (7.19)$$

$$S = \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix} \begin{pmatrix} \left\langle d\hat{u}_x \right\rangle \\ \left\langle d\hat{u}_y \right\rangle \\ \left\langle d\hat{u}_z \right\rangle \end{pmatrix}^\tau = \begin{pmatrix} l_x\left\langle d\hat{u}_x \right\rangle & l_x\left\langle d\hat{u}_y \right\rangle & l_x\left\langle d\hat{u}_z \right\rangle \\ l_y\left\langle d\hat{u}_x \right\rangle & l_y\left\langle d\hat{u}_y \right\rangle & l_y\left\langle d\hat{u}_z \right\rangle \\ l_z\left\langle d\hat{u}_x \right\rangle & l_z\left\langle d\hat{u}_y \right\rangle & l_z\left\langle d\hat{u}_z \right\rangle \end{pmatrix}$$

This presentation shows ten *random* parameters required to define U_n: six independent elements of the matrix *T*, libration scale $\left\langle d^2 \right\rangle$ in *L* and three parameters $\left\langle d\hat{u}_x \right\rangle, \left\langle d\hat{u}_y \right\rangle, \left\langle d\hat{u}_z \right\rangle$ for the correlations of the rotation and translation components in *S*. All other values necessary to calculate U_n (7.18) for all points are defined through the coordinates of $\mathbf{G}_1$ and $\mathbf{G}_2$ (7.1-7.2) and the coordinates of atoms "hidden" in A_n.

7.4. Axis with the fixed direction – modified coordinate systems

As above, we can switch to an equivalent set of parameters that have clearer physical interpretation. First, we diagonalise *T* as discussed in section 3.1 and get matrix R_t that describes the transition (3.3) from the common system $(\mathbf{i},\mathbf{j},\mathbf{k})$ to another Cartesian coordinate system $(\mathbf{i}_t,\mathbf{j}_t,\mathbf{k}_t)$ with the axes along the three principal axes of vibration (by its construction, this new coordinate system has nothing to do with the geometry of the rigid body but is defined by the nature of its movement). This leads to

$$\begin{pmatrix} q_{nx} \\ q_{ny} \\ q_{nz} \end{pmatrix} = \begin{pmatrix} \hat{u}_x \\ \hat{u}_y \\ \hat{u}_z \end{pmatrix} + dA_n \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix} = R_t \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} + dA_n \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix} \tag{7.20}$$

and

$$T = \left\langle R_t \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}^{\tau} R_t^{\tau} \right\rangle = R_t \begin{pmatrix} \langle t_x^2 \rangle & \langle t_x t_y \rangle & \langle t_x t_z \rangle \\ \langle t_x t_y \rangle & \langle t_y^2 \rangle & \langle t_y t_z \rangle \\ \langle t_x t_z \rangle & \langle t_y t_z \rangle & \langle t_z^2 \rangle \end{pmatrix} R_t^{\tau} = R_t \begin{pmatrix} \langle t_x^2 \rangle & 0 & 0 \\ 0 & \langle t_y^2 \rangle & 0 \\ 0 & 0 & \langle t_z^2 \rangle \end{pmatrix} R_t^{\tau}$$

$$L = \langle d^2 \rangle \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix} \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix}^{\tau} = \langle d^2 \rangle \begin{pmatrix} l_x^2 & l_x l_y & l_x l_z \\ l_x l_y & l_y^2 & l_y l_z \\ l_x l_z & l_y l_z & l_z^2 \end{pmatrix} \tag{7.21}$$

$$S = \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix} \begin{pmatrix} \langle dt_x \rangle \\ \langle dt_y \rangle \\ \langle dt_z \rangle \end{pmatrix}^{\tau} R_t^{\tau} = \begin{pmatrix} l_x \langle dt_x \rangle & l_x \langle dt_y \rangle & l_x \langle dt_z \rangle \\ l_y \langle dt_x \rangle & l_y \langle dt_y \rangle & l_y \langle dt_z \rangle \\ l_z \langle dt_x \rangle & l_z \langle dt_y \rangle & l_z \langle dt_z \rangle \end{pmatrix} R_t^{\tau}$$

We can also consider the Cartesian coordinate system $(\mathbf{i}_l, \mathbf{j}_l, \mathbf{k}_l)$ where $\mathbf{k}_l$ is aligned with the rotation axis, the corresponding matrix R_l being (7.10). Then (7.21) can be presented as

$$T = R_{lt} \begin{pmatrix} \langle t_x^2 \rangle & 0 & 0 \\ 0 & \langle t_y^2 \rangle & 0 \\ 0 & 0 & \langle t_z^2 \rangle \end{pmatrix} R_{lt}^{\tau}$$

$$L = \langle d^2 \rangle R_l \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} R_l^{\tau} \tag{7.22}$$

$$S = R_l \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \langle dt_x \rangle & \langle dt_y \rangle & \langle dt_z \rangle \end{pmatrix} R_{lt}^{\tau}$$

with

$$R_{lt} = R_t R_l^{-1} \tag{7.23}$$

Working with diagonalised matrices is more convenient and as previously shown aids in a better understanding of parameters of the *TLS* model. Also, we will see below in section 7.6 that there exists a special origin shift that diagonalises matrix *S*. Section 8.3 shows that in fact it is more convenient to start from diagonalisation of *L* and *S* and only then diagonalise *T*.

7.5. Libration axis that may change its direction

At the next level of generalisation we suppose that the direction of the rotation axis can vary. From now on let's assume that the coordinates (l_x, l_y, l_z) of the vector $\boldsymbol{l}$ be also random values. This may correspond to a general case of a group motion and not necessarily to a rotation around a covalent bond (see also an illustration in figure 7.2). If we introduce a new vector

$$\mathbf{d} = \begin{pmatrix} d_x \\ d_y \\ d_z \end{pmatrix} = d \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix} = \begin{pmatrix} dl_x \\ dl_y \\ dl_z \end{pmatrix} \tag{7.24}$$

then matrices (7.19) in (7.18) become

$$T = \begin{pmatrix} \langle \hat{u}_x^2 \rangle & \langle \hat{u}_x \hat{u}_y \rangle & \langle \hat{u}_x \hat{u}_z \rangle \\ \langle \hat{u}_x \hat{u}_y \rangle & \langle \hat{u}_y^2 \rangle & \langle \hat{u}_y \hat{u}_z \rangle \\ \langle \hat{u}_x \hat{u}_z \rangle & \langle \hat{u}_y \hat{u}_z \rangle & \langle \hat{u}_z^2 \rangle \end{pmatrix} \tag{7.25}$$

$$L = \begin{pmatrix} \langle d_x^2 \rangle & \langle d_x d_y \rangle & \langle d_x d_z \rangle \\ \langle d_x d_y \rangle & \langle d_y^2 \rangle & \langle d_y d_z \rangle \\ \langle d_x d_z \rangle & \langle d_y d_z \rangle & \langle d_z^2 \rangle \end{pmatrix} \tag{7.26}$$

$$S = \begin{pmatrix} \langle d_x \hat{u}_x \rangle & \langle d_x \hat{u}_y \rangle & \langle d_x \hat{u}_z \rangle \\ \langle d_y \hat{u}_x \rangle & \langle d_y \hat{u}_y \rangle & \langle d_y \hat{u}_z \rangle \\ \langle d_z \hat{u}_x \rangle & \langle d_z \hat{u}_y \rangle & \langle d_z \hat{u}_z \rangle \end{pmatrix} \tag{7.27}$$

Calculating explicitly the elements of U_n through elements of the composite matrices (7.25-7.27) and atomic coordinates gives (we use the symmetry of the L matrix)

$$U_{nxx} = T_{xx} + \left(z_n^2 L_{yy} + y_n^2 L_{zz} - 2y_n z_n L_{yz}\right) + 2\left(z_n S_{yx} - y_n S_{zx}\right) \tag{7.28}$$

$$U_{nyy} = T_{yy} + \left(z_n^2 L_{xx} + x_n^2 L_{zz} - 2x_n z_n L_{xz}\right) + 2\left(x_n S_{zy} - z_n S_{xy}\right)$$

$$U_{nzz} = T_{zz} + \left(y_n^2 L_{xx} + x_n^2 L_{yy} - 2x_n y_n L_{xy}\right) + 2\left(y_n S_{xz} - x_n S_{yz}\right)$$

$$U_{nxy} = T_{xy} + \left(y_n z_n L_{xz} + x_n z_n L_{yz} - x_n y_n L_{zz} - z_n^2 L_{xy}\right) + \left(x_n S_{zx} - y_n S_{zy} + z_n \left(S_{yy} - S_{xx}\right)\right)$$

$$U_{nxz} = T_{xz} + \left(y_n z_n L_{xy} + x_n y_n L_{yz} - x_n z_n L_{yy} - y_n^2 L_{xz}\right) + \left(z_n S_{yz} - x_n S_{yx} + y_n \left(S_{xx} - S_{zz}\right)\right)$$

$$U_{nyz} = T_{yz} + \left(x_n z_n L_{xy} + x_n y_n L_{xz} - y_n z_n L_{xx} - x_n^2 L_{yz}\right) + \left(y_n S_{xy} - z_n S_{xz} + x_n \left(S_{zz} - S_{yy}\right)\right)$$

(note the "+" sign at $x_n y_n L_{xz}$ term in the last equation in comparison with 1.2.11.5 in Coppens, 2006; this agrees with Table 1 in *TLSView Manual* (Merritt, pymmlib.sourceforge.net/tlsview/tlsview.html).

Clearly this general situation (7.25-7.27) is characterised by 21 parameters: 6 to describe *T*, 6 to describe *L* and 9 to describe an asymmetric matrix *S*. It is ubiquitously pointed out in the literature (see for example Schomaker & Trueblood (1968) or Coppens (2006)) that the linear combinations (7.28) of the *T*, *L* and *S* elements use only differences between S_{xx}, S_{yy}, S_{zz} and not these values themselves. Knowledge of two of these differences defines the third one. Simultaneous increasing or decreasing of $\langle d_x\hat{u}_x\rangle, \langle d_y\hat{u}_y\rangle, \langle d_z\hat{u}_z\rangle$ by the same quantity does not change U_n. Some consequences of this are discussed below in section 7.8.

7.6. Symmetrisation of S

Following from section 5.5, we will show that there exists a special origin for *S* which makes *S* symmetric (for example see Brenner (1967)). Accordingly to section 5.4, since *P* is antisymmetric (see (5.16)), $P^\tau = -P$ and *L* is symmetric, $L^\tau = L$, then

$$\left(S'\right)^\tau = S^\tau - \left(LP\right)^\tau \tag{7.29}$$

and

$$\left(S'\right)^\tau - S' = S^\tau - S + LP - \left(LP\right)^\tau \tag{7.30}$$

If with some choice of the origin matrix S' is symmetric, the last expression is equal to 0 giving

$$S^\tau - S = \left(LP\right)^\tau - LP = D \tag{7.31}$$

where *D* by construction is antisymmetric with diagonal elements equal to zero and off-diagonal elements equal to

$$\begin{aligned} D_{xy} &= -D_{yx} = p_x L_{xz} + p_y L_{yz} - p_z\left(L_{xx} + L_{yy}\right) = S_{yx} - S_{xy} \\ D_{zx} &= -D_{xz} = p_x L_{xy} - p_y\left(L_{xx} + L_{zz}\right) + p_z L_{yz} = S_{xz} - S_{zx} \\ D_{yz} &= -D_{zy} = -p_x\left(L_{yy} + L_{zz}\right) + p_y L_{xy} + p_z L_{xz} = S_{zy} - S_{yz} \end{aligned} \tag{7.32}$$

Here p_x, p_y, p_z are unknown parameters and the right-hand expressions contain corresponding elements of (7.27). The determinant of this system of linear equations, after insertion of the values from (7.26), is equal to

$$\begin{aligned} &\left(\langle d_x^2\rangle + \langle d_y^2\rangle\right)\left(\langle d_x^2\rangle\langle d_y^2\rangle - \langle d_x d_y\rangle^2\right) + \left(\langle d_x^2\rangle + \langle d_z^2\rangle\right)\left(\langle d_x^2\rangle\langle d_z^2\rangle - \langle d_x d_z\rangle^2\right) + \\ &+\left(\langle d_y^2\rangle + \langle d_z^2\rangle\right)\left(\langle d_y^2\rangle\langle d_z^2\rangle - \langle d_y d_z\rangle^2\right) + \\ &+2\left(\langle d_x^2\rangle\langle d_y^2\rangle\langle d_z^2\rangle - \langle d_x d_y\rangle\langle d_x d_z\rangle\langle d_y d_z\rangle\right) \end{aligned} \tag{7.33}$$

and is non-negative by the Cauchy-Schwarz inequality. It is equal to zero only in the hypotheticial case of a full correlation of all three motions, a case that does not happen in practice. This means that the system of equations (7.32) has always a unique solution. The corresponding point $\left(p_x, p_y, p_z\right)$ is called centre of libration (Pawley (1963), Hirshfeld *et al*. (1963) Schomaker & Trueblood (1968) and

Scheringer (1973)), "centre of diffusion" or "centre of reaction" (Brenner (1967), Tickle & Moss (1999)).

In fact, a possibility to symmetrise *S* is evident because it was demonstrated in case of a rotation around the **k** axis (section 5.4) and because the symmetry property is conserved with the rotation of the coordinate system (Appendix A4). This proves also that such a choice of the origin simultaneously minimises the trace of *T*, $tr(T) \rightarrow \min_{p_x, p_y, p_z}$, since the trace is also invariant with the coordinate system (Appendix A4).

When symmetrizing the matrix *S*, the number of the independent elements in it is reduced from 9 to 6. There is no contradiction since 3 'disappearing' parameters are now converted into *a priori unknown* coordinates of the reaction center.

7.7. T and L parameterisation

As previously, both symmetric matrices *T* and *L* can be diagonalised

$$T = R_t \begin{pmatrix} t_x^2 & 0 & 0 \\ 0 & t_y^2 & 0 \\ 0 & 0 & t_z^2 \end{pmatrix} R_t^\tau \qquad (7.34)$$

$$L = R_d \begin{pmatrix} d_{lx}^2 & 0 & 0 \\ 0 & d_{ly}^2 & 0 \\ 0 & 0 & d_{lz}^2 \end{pmatrix} R_d^\tau \qquad (7.35)$$

(see also Painter & Merritt (2006) about the diagonalisation of *L*). A difference with section 7.2 is that now R_l defined in (7.10) cannot be used anymore and is substituted by R_d that is built from the coordinates of the eigenvectors of *L*, similarly to R_t (see section 3). These eigenvectors describe three mutually orthogonal axes around which the rigid body has uncorrelated librations with the parameters d_{lx}, d_{ly}, d_{lz}, similarly to (6.6). So both *T* and *L* are characterised each by three mutually orthogonal directions and by three displacements corresponding to these directions. Other types of efficient parameterisation can be suggested (see for example Pawley (1970) or Rae (1975a, b)). In particular, the parameterisation suggested by Rae allows easy and efficient introducing of constraints on the *TLS* parameters.

7.8. S parameterisation

A non-symmetric matrix *S* is defined by nine its elements. However, only the differences between its diagonal elements are used in (7.28). The relation

$$(S_{yy} - S_{xx}) + (S_{xx} - S_{zz}) + (S_{zz} - S_{yy}) = 0 \qquad (7.36)$$

means that only 2 of these differences are sufficient to define U_n unambiguously reducing the total number of parameters to 20 = 6 + 6 + 8. Traditionally starting from Schomaker and Trueblood (1968) this constraint on the diagonal elements of *S* is presented differently. Equations (7.28) mean that the knowledge of U_n cannot unambiguously define the diagonal elements of *S* and an arbitrary constant *h* can be added to all of them simultaneously. The resulting diagonal elements cannot be too large since they should satisfy the Cauchy-Schwarz inequalities

$$S_{xx}^2 \le \left\langle \hat{u}_x^2 \right\rangle \left\langle d_x^2 \right\rangle,\ S_{yy}^2 \le \left\langle \hat{u}_y^2 \right\rangle \left\langle d_y^2 \right\rangle, S_{zz}^2 \le \left\langle \hat{u}_z^2 \right\rangle \left\langle d_z^2 \right\rangle \qquad (7.37)$$

Historically, the convention of setting the trace of the S matrix equal to 0,

$$S_{xx} + S_{yy} + S_{zz} = 0 \qquad (7.38)$$

is used to assert that (7.37) holds true.

Such a condition corresponds to (6.6) obtained for three rotation axes (we saw in section 6.7 above that the three resulting axes are always mutually orthogonal). The condition (7.38) can be also understood from the following equality:

$$\begin{aligned} tr(S) &= \left\langle d_x \hat{u}_x \right\rangle + \left\langle d_y \hat{u}_y \right\rangle + \left\langle d_z \hat{u}_z \right\rangle = \left\langle d_x \hat{u}_x + d_y \hat{u}_y + d_z \hat{u}_z \right\rangle \\ &= \left\langle dl_x \left[u_x - \left(l_y w_z - l_z w_y \right) \right] + dl_y \left[u_y - \left(l_z w_x - l_x w_z \right) \right] + dl_z \left[u_z - \left(l_x w_y - l_y w_x \right) \right] \right\rangle \\ &= \left\langle \left(dl_x \right) u_x + \left(dl_y \right) u_y + \left(dl_z \right) u_z \right\rangle + \\ &+ \left\langle d \left[l_x \left(l_y w_z - l_z w_y \right) + dl_y \left(l_z w_x - l_x w_z \right) + dl_z \left(l_x w_y - l_y w_x \right) \right] \right\rangle \\ &= \left\langle \left(dl_x \right) u_x + \left(dl_y \right) u_y + \left(dl_z \right) u_z \right\rangle \end{aligned} \qquad (7.39)$$

The second term in (7.39) disappeares because the expression in $[\ ]$ is a scalar product of two orthogonal vectors, $\boldsymbol{l}$ and $\boldsymbol{l} \times \mathbf{w}$. Therefore, condition (7.38) becomes

$$\left\langle \left(dl_x \right) u_x + \left(dl_y \right) u_y + \left(dl_z \right) u_z \right\rangle = 0, \qquad (7.40)$$

a requirement that among of all possible decompositions of U_n we choose that with no correlation between the translation and rotation. This agrees with the remarks by some authors that the difficulty of finding individual S_{xx}, S_{yy}, S_{zz} "*arises from incomplete knowledge about the correlation of atomic motions*" (Bürgi (1989), see also Scheringer (1973)).

8. General case

8.1. Several axes in a general position

Let's suppose now that a rigid body participates simultaneously in several librations, K in total, of different amplitudes (see for example Stuart & Phillips (1985)). These librations are defined by axes $\boldsymbol{l}_k$, k = 1,K, by the points $\mathbf{w}_k$ at the corresponding axis and by the elementary shifts d_k, due to rotations (section 2.3). The axes are not necessarily mutually intersecting. The displacement vector $\mathbf{v}$ due to rotations becomes (see (2.9))

$$\mathbf{v}_n = \sum_k d_k A_n \boldsymbol{l}_k - \sum_k d_k \boldsymbol{l}_k \times \mathbf{w}_k = A_n \sum_k d_k \boldsymbol{l}_k - \sum_k d_k \boldsymbol{l}_k \times \mathbf{w}_k = dA_n \hat{\boldsymbol{l}} - \hat{\mathbf{d}}_w \qquad (8.1)$$

Now, with $d = \left\| \hat{\mathbf{d}} \right\|$ and $\hat{\boldsymbol{l}} = \hat{\mathbf{d}} / d$, both

$$\hat{\mathbf{d}} = \sum_k d_k \boldsymbol{l}_k = \begin{pmatrix} \sum_k d_k l_{kx} \\ \sum_k d_k l_{ky} \\ \sum_k d_k l_{kz} \end{pmatrix} = \begin{pmatrix} \hat{d}_x \\ \hat{d}_y \\ \hat{d}_z \end{pmatrix} = d\hat{\boldsymbol{l}} \tag{8.2}$$

and

$$\hat{\mathbf{d}}_w = \sum_k d_k \boldsymbol{l}_k \times \mathbf{w}_k = \sum_k d_k W_k \boldsymbol{l}_k = \sum_k d_k \begin{pmatrix} 0 & w_{kz} & -w_{ky} \\ -w_{kz} & 0 & w_{kx} \\ w_{ky} & -w_{kx} & 0 \end{pmatrix} \begin{pmatrix} l_{kx} \\ l_{ky} \\ l_{kz} \end{pmatrix} =$$

$$= \begin{pmatrix} \sum_k d_k \left(w_{kz} l_{ky} - w_{ky} l_{kz} \right) \\ \sum_k d_k \left(w_{kx} l_{ky} - w_{kz} l_{kx} \right) \\ \sum_k d_k \left(w_{ky} l_{kx} - w_{kx} l_{ky} \right) \end{pmatrix} = \begin{pmatrix} \hat{d}_{wx} \\ \hat{d}_{wy} \\ \hat{d}_{wz} \end{pmatrix} \tag{8.3}$$

are random vectors depending on random variables d_k and on the parameters of the axes. The second term in (8.1) is common for all points and acts as an apparent (random) translation of the rigid body (see section 4.2). Expression (8.1) shows that a multiple-axes rotation may be considered as a rotation around a single random rotation axis (section 7.5). The normalised vector $\hat{\boldsymbol{l}}$, $\left\| \hat{\boldsymbol{l}} \right\| = 1$, defines the rotation axis and d is the parameter for the rotation angle as discussed in section 2.3.

8.2. General formulae

When a translational displacement complements rotations, the total displacement $\mathbf{q}_n = \mathbf{u} + \mathbf{v}_n$ of the atom n at a point $\mathbf{r}_n$ is the sum of $\mathbf{u}$ and $\mathbf{v}_n$ due to rigid body translation and libration, respectively:

$$\begin{pmatrix} q_{xn} \\ q_{yn} \\ q_{zn} \end{pmatrix} = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} + A_n \begin{pmatrix} \hat{d}_x \\ \hat{d}_y \\ \hat{d}_z \end{pmatrix} - \begin{pmatrix} \hat{d}_{wx} \\ \hat{d}_{wy} \\ \hat{d}_{wz} \end{pmatrix} = \begin{pmatrix} u_x - \hat{d}_{wx} \\ u_y - \hat{d}_{wy} \\ u_z - \hat{d}_{wz} \end{pmatrix} + A_n \begin{pmatrix} \hat{d}_x \\ \hat{d}_y \\ \hat{d}_z \end{pmatrix} = \begin{pmatrix} \hat{u}_x \\ \hat{u}_y \\ \hat{u}_z \end{pmatrix} + A_n \begin{pmatrix} \hat{d}_x \\ \hat{d}_y \\ \hat{d}_z \end{pmatrix} \tag{8.4}$$

where random values are $\hat{u}_x, \hat{u}_y, \hat{u}_z, \hat{d}_x, \hat{d}_y, \hat{d}_z$. For the atom n, the components of U_n expressed in the original Cartesian coordinate system $(\mathbf{i}, \mathbf{j}, \mathbf{k})$ as (7.19) are

$$T = \begin{pmatrix} \langle \hat{u}_x^2 \rangle & \langle \hat{u}_x \hat{u}_y \rangle & \langle \hat{u}_x \hat{u}_z \rangle \\ \langle \hat{u}_x \hat{u}_y \rangle & \langle \hat{u}_y^2 \rangle & \langle \hat{u}_y \hat{u}_z \rangle \\ \langle \hat{u}_x \hat{u}_z \rangle & \langle \hat{u}_y \hat{u}_z \rangle & \langle \hat{u}_z^2 \rangle \end{pmatrix} \tag{8.5}$$

$$L = \begin{pmatrix} \langle \hat{d}_x^2 \rangle & \langle \hat{d}_x \hat{d}_y \rangle & \langle \hat{d}_x \hat{d}_z \rangle \\ \langle \hat{d}_x \hat{d}_y \rangle & \langle \hat{d}_y^2 \rangle & \langle \hat{d}_y \hat{d}_z \rangle \\ \langle \hat{d}_x \hat{d}_z \rangle & \langle \hat{d}_y \hat{d}_z \rangle & \langle \hat{d}_z^2 \rangle \end{pmatrix} \quad (8.6)$$

$$S = \begin{pmatrix} \langle \hat{d}_x \hat{u}_x \rangle & \langle \hat{d}_x \hat{u}_y \rangle & \langle \hat{d}_x \hat{u}_z \rangle \\ \langle \hat{d}_y \hat{u}_x \rangle & \langle \hat{d}_y \hat{u}_y \rangle & \langle \hat{d}_y \hat{u}_z \rangle \\ \langle \hat{d}_z \hat{u}_x \rangle & \langle \hat{d}_z \hat{u}_y \rangle & \langle \hat{d}_z \hat{u}_z \rangle \end{pmatrix} \quad (8.7)$$

Equations (8.5)-(8.7) are literally similar to (7.25)-(7.27) derived and analysed previously.

8.3. Analysis of the *TLS* matrices

Both simple examples above and the demonstration in section 8.2 show that for all kinds of rigid-body oscillations, considered in a harmonic approximation, the set of matrices U_n for all atoms of the group can be expressed as a sum (5.10) of four terms where T, L and S are common for all atoms and each antisymmetric matrix A_n contains individual coordinates of the atom n. Matrices T and L are symmetric while S is not necessarily symmetric.

Now let's suppose that crystallographic calculations, for example as discussed below in section 9, gave us some T, L & S matrices and we want to find which rigid-body motion produces these matrices. To answer this question several steps should be performed. We remind the reader that rotation axes that do not pass through the origin as well as the rotation-translation correlation contribute to an apparent translation and that these contributions should be removed properly in order to define the pure translation component.

a) Origin shift. Shift of the origin into the reaction center p_x, p_y, p_z (section 7.6) is the solution of the system of linear equations (7.32). The new matrices T', L', S' in this new coordinate system are obtained accordingly to (5.18). In this coordinate system matrix S' is symmetric, trace of T' is the smallest possible; both these properties are retained in further rotation of the coordinate system (Appendix A4). Obtain new atomic coordinates (x'_n, y'_n, z'_n) by subtracting (p_x, p_y, p_z), $\mathbf{r}'_n = \mathbf{r}_n - \mathbf{p}$. In fact here and later we do not need atomic coordinates for the T, L, or S interpretation; the transformation is done simply for completeness.

b) Diagonalisation of L. Find 3 non-negative eigenvalues of matrix L' and three mutually orthogonal eigenvectors; rotation matrix R_d (7.35) is composed from the coordinates of these eigenvectors. Choose a new coordinate system with the new axes along the three eigenvectors; R_d is the transformation matrix to this system. Recalculate the matrices $L'' = R_d L' R_d^\tau$ (7.35), $T'' = R_d T' R_d^\tau$, $S'' = R_d S' R_d^\tau$ and new atomic coordinates as $\mathbf{r}''_n = R_d^{-1}\mathbf{r}'_n = R_d^\tau \mathbf{r}'_n$ in this new system (Appendices A1-A2). In the new coordinate system matrix L'' is diagonal with the elements $L''_{xx}, L''_{yy}, L''_{zz}$. The rotation axes are parallel to the new coordinate axes and pass through the points $(0, w_y^i, w_z^i)$, $(w_x^j, 0, w_z^j)$, $(w_x^k, w_y^k, 0)$ to be defined.

c) Position of rotation axes. Obtain estimates $\bar{d}_x = \sqrt{L''_{xx}}$, $\bar{d}_y = \sqrt{L''_{yy}}$, $\bar{d}_z = \sqrt{L''_{zz}}$ (6.6) of the

rotation parameters around the three axes defined at the previous step; calculate the positions of the rotation axes (4.16):

$$w_y^i = -\frac{S''_{xz}}{L''_{xx}},\ w_z^i = \frac{S''_{xy}}{L''_{xx}},\ w_x^j = \frac{S''_{yz}}{L''_{yy}},\ w_z^j = -\frac{S''_{yx}}{L''_{yy}},\ w_x^k = -\frac{S''_{zy}}{L''_{zz}},\ w_y^k = \frac{S''_{zx}}{L''_{zz}} \qquad (8.8)$$

d) Contribution of rotations to T due to axes displacement. Calculate contribution (6.5) of the rotations to the translation of the group due to the displacement of the axes

$$\Delta_T = \begin{pmatrix} \left(w_z^j\right)^2 L''_{yy} + \left(w_y^k\right)^2 L''_{zz} & -w_x^k w_y^k L''_{zz} & -w_x^j w_z^j L''_{yy} \\ -w_x^k w_y^k L''_{zz} & \left(w_z^i\right)^2 L''_{xx} + \left(w_x^k\right)^2 L''_{zz} & -w_y^i w_z^i L''_{xx} \\ -w_x^j w_z^j L''_{yy} & -w_y^i w_z^i L''_{xx} & \left(w_y^i\right)^2 L''_{xx} + \left(w_x^j\right)^2 L''_{yy} \end{pmatrix}; \qquad (8.9)$$

the residual translation matrix after removal this contribution is

$$T''' = T'' - \Delta_T \qquad (8.10)$$

e) Minimisation of correlation between translation and rotation. Calculate the trace $trS'' = S''_{xx} + S''_{yy} + S''_{zz}$ of S'' and get a new $\widetilde{S}$ (section 7.8) with the minimal correlation between translation and rotation (7.40)

$$\widetilde{S} = S'' - \begin{pmatrix} \frac{1}{3} trS'' & 0 & 0 \\ 0 & \frac{1}{3} trS'' & 0 \\ 0 & 0 & \frac{1}{3} trS'' \end{pmatrix} \qquad (8.11)$$

f) Contribution of screwing to T. Obtain estimates $\bar{s}_x = \frac{\widetilde{S}_{xx}}{L''_{xx}}$, $\bar{s}_y = \frac{\widetilde{S}_{yy}}{L''_{yy}}$, $\bar{s}_z = \frac{\widetilde{S}_{zz}}{L''_{zz}}$ of the screw parameters (6.9)-(6.10) following the rotation axes currently aligned with the coordinate axes; remove the contribution of the screwing from the translation matrix (6.8) :

$$\widetilde{T} = T''' - \begin{pmatrix} \bar{s}_x \widetilde{S}_{xx} & 0 & 0 \\ 0 & \bar{s}_y \widetilde{S}_{yy} & 0 \\ 0 & 0 & \bar{s}_z \widetilde{S}_{zz} \end{pmatrix} \qquad (8.12)$$

The resulting matrix $\widetilde{T}$ stands for the pure translation of the rigid group with the contribution of other movements removed.

g) Three uncorrelated translations. Find 3 non-negative eigenvalues of matrix $\widetilde{T}$ and three mutually orthogonal eigenvectors (section 3). The three eigenvectors give the directions of the uncorrelated translations and the corresponding eigenvalues are means square displacement along these axes. Do not forget that these vectors are given in the coordinate system with the axes parallel to the rotation axes and not in the original one.

9. Search for the optimal *TLS* decomposition

9.1. Optimal *TLS* decomposition and refinement with *TLS*

Summarizing the analysis above, we conclude that for all kinds of harmonic oscillation of a rigid group, the matrices U_n for all its points can be presented as (5.10) through three common matrices *T*, *L* & *S* and through matrices A_n specific for each point. Sections 3-8 above show how to calculate the contribution $U_{TLS,n}$ of a rigid group motion into atomic displacement parameters U_n of the corresponding atoms when the movement is known. A frequent task (see for example Sternberg *et al*. (1979)) is the inverse problem: given a set of matrices for a group of atoms (let's call these matrices $\overline{U}_n$, differently from U_n that are matrices calculated from the atomic parameters), present them in the *TLS* form considering that this group oscillates as a rigid body, *i.e.* find all the elements of the three composing matrices (8.5-8.7) reproducing as close as possible the whole set of $\overline{U}_n$. We may note that such a problem is important not only for their *a posteriori* analysis (see for example Holbrook & Kim (1984), Kuriyan & Weiss (1991), Stec *et al.*(1995) while some more applications can be found in the additional list of references) but also to obtain initial values of *TLS* parameters for their further refinement as discussed below.

The problem of decomposing the set of $\overline{U}_n$ into *TLS* (find *TLS* such that the corresponding U_n are close as much as possible to $\overline{U}_n$) is more complicated in a real situation when the matrices $\overline{U}_n$ contain contribution of individual (independent) atomic vibrations $U_{ind,n}$ and other contributions and errors, when rigid groups are an idealisation and when a composition of these rigid groups is *a priori* unknown. Looking for optimal values of the *TLS* parameters means to minimise some target function with respect to these parameters. Here we do not discuss how to decide which atoms belong to which rigid group. This problem was studied by, for example, Winn *et al*. (2001) and Painter & Merritt (2005, 2006a, 2006b). See also references therein for further details outside the scope of this article.

To find an optimal solution of the problem, a formal measure of the solution quality shall be introduced first. Traditionally, it might be a least-squares target for a difference between the elements of all $\overline{U}_n$ and corresponding calculated U_n (for example, Painter & Merritt (2006)):

$$\sum_{n=1,N}\left\{\left(U_{TLS,nxx}-\bar{U}_{nxx}\right)^2+\left(U_{TLS,nyy}-\bar{U}_{nyy}\right)^2+\left(U_{TLS,nzz}-\bar{U}_{nzz}\right)^2+\right. \tag{9.1}$$

$$\left.+\left(U_{TLS,nxy}-\bar{U}_{nxy}\right)^2+\left(U_{TLS,nxz}-\bar{U}_{nxz}\right)^2+\left(U_{TLS,nyz}-\bar{U}_{nyz}\right)^2\right\}\rightarrow \min$$

However, this target may be more sensible to errors for atoms that are far from the reaction center (since the matrices A_n are "larger"), does not distinguish atomic types (a proper modeling of U_n for a heavy atom may be more important considering its contribution to the electron density and structure factors) and others. Also the sum over all elements of the matrix U_n is probably non-optimal since not all of them contribute equally to the form of electron clouds. From that point of view, the target function (9.1) is an intuitive, but has no real physical background. Merritt (1999) suggested a more sophisticated density-correlation-based target expressed through anisotropic atomic displacement parameters

$$\frac{\left(\det \bar{U}^{-1} \det \bar{U}_{TSL}^{-1}\right)^{1/4}}{\left[\det\left(\bar{U}^{-1} + \bar{U}_{TSL}^{-1}\right)/8\right]} \rightarrow \max \qquad (9.2)$$

In fact, some more general function of the form

$$f\left(TLSparameters; \left\{\bar{U}_{nxx}, \bar{U}_{nxx}, \bar{U}_{nxx}, \bar{U}_{nxx}, \bar{U}_{nxx}, \bar{U}_{nxx}, n = 1, N\right\}\right) \rightarrow \min \qquad (9.3)$$

can be introduced. Here in the simplest case '*TLS parameters*' mean the elements of the matrices (8.5-8.7); some other ways of parameterisation have been discussed in the text. Formally speaking, one should control that the residual matrices $\bar{U}_n - U_{TLS,n}$ are positive definite since they suppose to correspond to $U_{ind,n}$. At the same time, practical studies show that neglecting this restriction allows to improve the *R*-factors (Afonine, phenix-online.org/newsletter/CCN_2010_07.pdf).

Another task, different from reinterpretation of $\bar{U}_n$, is to consider directly *TLS parameters* as parameters of the model during model refinement

$$f(\{atomic\ coordinates\}, \{TLS\ parameters\}, \{F_{obs}(hkl)\}) \rightarrow \min \qquad (9.4)$$

as it was implemented in *CORELS* (Sussman *et al*., 1977), *RESTRAIN* (Driessen *et al*., 1989) and later in *REFMAC* (Winn *et al*., 2001), *phenix.refine* (Afonine *et al*., 2005) and *BUSTER* (Bricogne *et al*., 2009). See also Moss et. al. (1996). Here f can be any appropriate function of model parameters and experimental diffraction data.

9.2. Practical scheme

As stated above, the formal goal is to find the parameters of the *T*, *L* and *S* matrices that minimise the target (9.3) or (9.4). One can note that the elements of $U_{TLS,n}$ are linear functions of the elements of the *TLS* matrices. When the target is a quadratic function of elements of U_n like (9.1), a close solution of the problem can be suggested solving the system of linear equations. However, in practice, even in this case iterative optimisation methods may be required where knowledge of best possible initial parameter values facilitates solving the problem.

Given a set of $\bar{U}_n$ values, the *TLS* matrices intuitively should include "as much as possible" of common atomic movement leaving the rest to individual atomic movements. To start the procedure, one can try to assign all possible common motion to the T matrix making L and S equal to zero unless they are known from previous refinement cycles. When all atomic displacement factors are isotropic with $\bar{B}_n$ instead of $\bar{U}_n$, the search for the "maximal" T is trivial. This matrix is diagonal, with all three diagonal elements equal to the minimal $\bar{B}_n$ value over all atoms of the group. When the displacement parameter is anisotropic for some of atoms of the group, the "maximal" T can be found following the algorithm described by Afonine & Urzhumtsev (2007). Then the parameters of all three TLS matrices are refined.

9.3. Once more about the origin at the reaction center

When using *TLS* parameterisation, the choice of group origin is arbitrary and does not affect the matrices $U_{TLS,n}$ calculated from obtained *TLS*. Typically the origin is taken as center of mass or center of geometry of a *TLS* group (the difference is the use or lack of use of the molecular weights in averaging

the atomic coordinates; usually this difference is insignificant) as suggested by Rae (1975b) and used for example in *RESTRAIN* (Driessen *et al.*, 1989). Citing Tickle & Moss (1999), "*...for a molecule constrained by intermolecular forces in a crystalline environment, the centre of gravity loses the special significance that it has for freely moving rigid bodies.*" The natural origin for this model is a reaction center and the idea that the body is oscillated around some bond(s) suggests that it is rather at a periphery of the body and not at its centre. Unlikely for the center of mass, the reaction center is initially unknown.

If a hypothetical rotation axis is known in advance, one can initially choose any its point as the origin instead of the center of mass. In any case, it seems useful to find the coordinates of the reaction center from diagonalisation of S (section 7.6) when it starts to be known and to reassign there the origin for further calculations (see for example *TLSANL* by Howlin *et al.*, 1993).

Appendix A. Changing the coordinate system

A1. Transformation matrix

Let (x, y, z) be coordinates of a vector $\mathbf{q}$ in some coordinate system with the basis vectors $(\mathbf{i}, \mathbf{j}, \mathbf{k})$ and (x', y', z') be coordinates of the same vector in another coordinate system with the basis vectors $(\mathbf{i}', \mathbf{j}', \mathbf{k}')$. Let (i'_x, i'_y, i'_z), (j'_x, j'_y, j'_z) and (k'_x, k'_y, k'_z) be coordinates of the vectors of the new base in the initial coordinate system. We define the transformation matrix

$$Q = \begin{pmatrix} i'_x & j'_x & k'_x \\ i'_y & j'_y & k'_y \\ i'_z & j'_z & k'_z \end{pmatrix} \qquad \text{(A1.1)}$$

It is easy to see the rule

$$\begin{pmatrix} i'_x \\ i'_y \\ i'_z \end{pmatrix} = Q \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} j'_x \\ j'_y \\ j'_z \end{pmatrix} = Q \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} k'_x \\ k'_y \\ k'_z \end{pmatrix} = Q \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \qquad \text{(A1.2)}$$

linking the initial coordinates of the base vectors $(\mathbf{i}, \mathbf{j}, \mathbf{k})$ with their coordinates in the new system. The same rule can be applied to any vector $\mathbf{q}$:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = Q \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} \qquad \text{(A1.3)}$$

To demonstrate this, it is sufficient to note that

$$\mathbf{q} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k} = x'\mathbf{i}' + y'\mathbf{j}' + z'\mathbf{k}' \qquad \text{(A1.4)}$$

That gives, accordingly to (A1.2), in the original coordinate system

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = x' Q \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + y' Q \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + z' Q \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = Q \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} \qquad \text{(A1.5)}$$

A2. Relation between coordinates

Let (q_x, q_y, q_z) and (q'_x, q'_y, q'_z) be coordinates of a vector $\mathbf{q}$ in some coordinate systems with the base vectors $(\mathbf{i}, \mathbf{j}, \mathbf{k})$ and $(\mathbf{i}', \mathbf{j}', \mathbf{k}')$, respectively. If Q is the transformation matrix as defined in Appendix A1, then accordingly to (A1.3) the coordinates of a vector in these coordinate systems are linked by relations:

$$\begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix} = Q \begin{pmatrix} q'_x \\ q'_y \\ q'_z \end{pmatrix} \quad , \quad \begin{pmatrix} q'_x \\ q'_y \\ q'_z \end{pmatrix} = Q^{-1} \begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix}, \tag{A2.1}$$

$$(q_x\ q_y\ q_z) = (q'_x\ q'_y\ q'_z) Q^{\tau} \quad , \quad (q'_x\ q'_y\ q'_z) = (q_x\ q_y\ q_z)(Q^{\tau})^{-1}$$

Let's have a quadratic function of the coordinates

$$f(\mathbf{q}) = \left(-\mathbf{q}^{\tau} U^{-1} \mathbf{q}\right) = (q_x\ q_y\ q_z) U^{-1} \begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix} \tag{A2.2}$$

such that this function is independent of the choice of the coordinate system. This means that, using (A2.1),

$$f(\mathbf{q}) = (q'_x\ q'_y\ q'_z)\left[U'\right]^{-1} \begin{pmatrix} q'_x \\ q'_y \\ q'_z \end{pmatrix} = (q_x\ q_y\ q_z) U^{-1} \begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix}$$

$$= \left[(q'_x\ q'_y\ q'_z) Q^{\tau}\right] U^{-1} \left[Q \begin{pmatrix} q'_x \\ q'_y \\ q'_z \end{pmatrix} \right] = (q'_x\ q'_y\ q'_z) \left[Q^{\tau} U^{-1} Q\right] \begin{pmatrix} q'_x \\ q'_y \\ q'_z \end{pmatrix} \tag{A2.3}$$

requiring the relation

$$U'^{-1} = Q^{\tau} U^{-1} Q \tag{A2.4}$$

or

$$U' = Q^{-1} U_n (Q^{\tau})^{-1} \tag{A2.5}$$

The last relation is satisfied for U defined as (2.3) since

$$U' = \left\langle \mathbf{q}\mathbf{q}^{\tau} \right\rangle = \left\langle \begin{pmatrix} q'_x \\ q'_y \\ q'_z \end{pmatrix} \begin{pmatrix} q'_x & q'_y & q'_z \end{pmatrix} \right\rangle$$

$$= \left\langle Q^{-1} \begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix} \begin{pmatrix} q_x & q_y & q_z \end{pmatrix} (Q^{\tau})^{-1} \right\rangle = Q^{-1} U (Q^{\tau})^{-1} \qquad \text{(A2.6)}$$

Quite often both old and new bases are composed from the unit vectors mutually orthogonal to each other. This means that the transformation matrix Q is nothing else but a rotation matrix. For example, a rotation by angle φ around the axis $\mathbf{k}$ is defined by the matrix

$$Q = R_z(\varphi) = \begin{pmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \text{(A2.7)}$$

Note also that here the inverse transformation is a rotation by angle $-\varphi$ giving

$$Q^{-1} = R_z^{-1}(\varphi) = R_z(-\varphi) = \begin{pmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} = R_z^{\tau}(\varphi) = Q^{\tau} \qquad \text{(A2.8)}$$

This property $R^{-1}(\varphi) = R^{\tau}(\varphi)$ is true for any rotation matrix given in the orthonormal coordinate system. With this property the transformations (A2.4-A2.5) become

$$U'_n = Q^{-1} U_n Q \qquad \text{(A2.9)}$$

$$U'^{-1}_n = Q^{-1} U_n^{-1} Q$$

A3. Matrices of linear operators

A transformation of vectors of the three-dimensional space, $\mathbf{q} \rightarrow \mathbf{p}$, is called linear (a linear operator) if for any vectors $\mathbf{q}_1 \rightarrow \mathbf{p}_1$, $\mathbf{q}_2 \rightarrow \mathbf{p}_2$ and for any number λ there are

$$(\mathbf{q}_1 + \mathbf{q}_2) \rightarrow \mathbf{p}_1 + \mathbf{p}_2 \qquad \text{(A3.1)}$$

$$(\lambda \mathbf{q}_1) \rightarrow \lambda \mathbf{p}_1$$

A particular example of a linear transformation is rotation.

In a given coordinate system with the base vectors $(\mathbf{i}, \mathbf{j}, \mathbf{k})$ a linear transformation can be defined by a matrix

$$R = \begin{pmatrix} i_x & j_x & k_x \\ i_y & j_y & k_y \\ i_z & j_z & k_z \end{pmatrix} \qquad \text{(A3.2)}$$

which columns are coordinates of the transformed base vectors $\mathbf{i}, \mathbf{j}, \mathbf{k}$, respectively. It is easy to see that the coordinates (p_x, p_y, p_z) of each transformed vector are related to the coordinates (q_x, q_y, q_z) of the initial vector by relation

$$\begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} = R \begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix} \qquad \text{(A3.3)}$$

Note that here (A3.3) links coordinates of different vectors in the same coordinate system while (A1.3) and (A2.1) link coordinates of the same vector but in different coordinate system.

It can be noted that the matrix of the inverse transformation $\mathbf{p} \rightarrow \mathbf{q}$, when this transformation exists, is just the inverse matrix R^{-1}. It follows from (A3.3) and (A2.1) that

$$\begin{pmatrix} p'_x \\ p'_y \\ p'_z \end{pmatrix} = Q^{-1} \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} = Q^{-1} R \begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix} = Q^{-1} R Q \begin{pmatrix} q'_x \\ q'_y \\ q'_z \end{pmatrix} \qquad \text{(A3.4)}$$

and when changing the coordinate system, the matrix of a linear operator, not necessary a rotation one, is transformed following the rule similar to (A2.10) :

$$R' = Q^{-1} R Q \qquad \text{(A3.6)}$$

Just as a remark we remind the reader that the property $R^{-1}(\varphi) = R^{\tau}(\varphi)$ of the rotation matrices is not necessary conserved for non-orthonormal coordinate systems. For example, for a rotation by $\pi/3$ in the hexagonal coordinate system

$$R_{zH}(\pi/3) = \begin{pmatrix} 1 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \; R_{zH}^{-1}(\pi/3) = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \neq R_{zH}^{\tau}(\pi/3) \qquad \text{(A3.7)}$$

A4. Properties of matrices U (trace and symmetry)

Let's consider two square matrices, A with the coefficients α_{jk} and B with the coefficients β_{jk}, $j,k=1,\ldots,K$. We start from a trivial exercise

$$tr(AB) = \sum_{j=1,K} (AB)_{jj} = \sum_{j=1,K} \sum_{k=1,K} \alpha_{jk} \beta_{kj} = \sum_{k=1,K} \sum_{j=1,K} \alpha_{jk} \beta_{kj} =$$

$$= \sum_{k=1,K} \sum_{j=1,K} \beta_{kj} \alpha_{jk} = \sum_{k=1,K} (BA)_{jj} = tr(BA) \qquad \text{(A4.1)}$$

showing a property of the trace of a product of two matrices. As a consequence, for the matrices with the

property (A2.9) or (A3.6)

$$tr(U') = tr(Q^{-1}UQ) = tr(UQQ^{-1}) = tr(U) \quad , \qquad \text{(A4.2)}$$

the trace is conserved when changing the coordinate system.

Another property used in the main text is that for such matrices the symmetry of the matrix $U^{\tau} = U$ is conserved with the rotation of the coordinate system :

$$(U')^{\tau} = (Q^{-1}UQ)^{\tau} = Q^{\tau}U^{\tau}\left(Q^{-1}\right)^{-1} = Q^{-1}UQ = U' \qquad \text{(A4.3)}$$

REFERENCES

Afonine, P.V. & Grosse-Kunstleve, R.W. & Adams, P.D. (2005). "The Phenix refinement framework". *CCP4 Newsletter on Protein Crystallography,* **42**, contribution 8.

Afonine, P.V. & Urzhumtsev, A. (2007). "On determination of *T* matrix in *TLS* modelling". *CCP4 Newsletter on Protein Crystallography,* **45**, http://www.ccp4.ac.uk/newsletters/

Afonine, P.V., Urzhumtsev, A., Grosse-Kunstleve, R.W. & Adams, P.D. (2010). "Atomic displacement parameters (ADPs), their parameterization and refinement in PHENIX".*Cryst.Comput. Newsletters*, **1**, 24-31.

Becka, L.N. & Cruickshank, D.W.J. (1961). "Coordinate errors due to rotational oscillations of molecules". *Acta Cryst.* **14**, 1092-1092.

Birnbaum, G.I. (1972). "The crystal and molecular structure of the trans-syn photodimer of methyl orotate". *Acta Cryst.* B**28**, 1248-1254.

Brenner, H. (1967). "Coupling between the translational and rotational Brownian motions of rigid particles of arbitrary shape". *J.Colloid Interface Chem.* **23**, 407-435.

Bricogne, G., Blanc, E., Brandl, M., Flensburg, C., Keller, P., Paciorek, W., Roversi, P., Smart, O., Vonrhein, C. & Womack, T.O. (2009). *BUSTER v.2.8.0.* Global Phasing Ltd, Cambridge.

Bürgi, H.B. (1989). "Interpretation of atomic displacement parameters: intramolecular translational oscillation and rigid-body motion". *Acta Cryst.* B**45**, 383-390.

Burns, D.M., Ferrier, W.G. & McMullan, J.T. (1967). "The rigid-body vibrations of molecules in crystals". *Acta Cryst.* **22**, 623-629.

Busing, W.R., Levy, H.A. (1964). "The effect of thermal motion on the estimation of bond lengths from diffraction measurements". *Acta Cryst.* **17**, 142-146.

Coppens, P. (2006). "The structure factor". *International Tables for Crystallography*, Vol. B, ed. U.Shmueli; Kluwer Academic Publishers; Dordrecht/Boston/London, 10-24.

Cruickshank, D.W.J. (1956a). "The determination of the anisotropic thermal motion of atoms in crystals". *Acta Cryst.* **9**, 747-753.

Cruickshank, D.W.J. (1956b). "The analysis of the anisotropic thermal motion of molecules in crystals". *Acta Cryst.* **9**, 754-756.

Cruickshank, D.W.J. (1956c). "Errors in bond lengths due to rotational oscillations of molecules". *Acta*

Cryst. **9**, 757-758.

Cruickshank, D.W.J. (1961). "The variation of apparent bond lengths with temperature in molecular crystals". *Acta Cryst.* **14**, 896-897.

Downs, R.T., Gibbs, G.V., Barletmehs, K.L. & Boisen, M.B., Jr. (1992). "Variation of bond lengths and volumes of silicate tetrahedra with temperature". *American Mineral.* **77**, 751-757.

Driessen, H., Haneef, M.I.J., Harris, G.W., Howlin, B., Khan, G. & Moss, D.S. (1989). "RESTRAIN: restrained structure-factor least-squares refinement program for macromolecules". *J.Appl. Cryst.,* **22,** 510-516.

Dunitz, J.D. & White, D.N.J. (1973). "Non-rigid-body thermal-motion analysis". *Acta Cryst.* A**29**, 93-94.

Dunitz, J.D. (1979). *X-ray analysis and the structure of organic molecules*, Cornell University Press, Ithaca and London.

Dunitz, J.D., Schomaker, V. & Trueblood, K.N. (1988). "Interpretation of atomic displacement parameters from diffraction studies of crystals". *J.Phys.Chem.* **92,** 856-867.

Dunitz, J.D. (1999). "A curiously short carbon-carbon double bond ?". *Chem.Communication,* 2574-2574.

Goldstein, H. (1950). *Classical Mechanics*. Cambridge, Massachusetts: Addison-Wesley.

Haestier, J., Sadki, M., Thompson, A. & Watkin, D. (2008). "Error estimates on bond-length and angle corrections from *TLS* analysis". *J.Appl. Cryst.* **41**, 531-536.

Haneef, I., Moss, D.S., Stanford, M.J. & Borkakoti, N. (1985). "Restrained structure-factor least-squares refinement of protein structures using a vector-processing computer". *Acta Cryst.* A**41**, 426-433.

Hirshfeld, F.L., Sandler, S. & Schmidt, G.M.J. (1963). "The structure of overcrowded aromatic compounds. VI. The crystal structure of benzo[c]phenanthrene and of 1,12-dimethylbenzo[c]phenanthrene". *J.Chem.Soc.*, 2108-2125.

Holbrook, S.R. & Kim, S.-H. (1984). "Local mobility of nucleic acids as determined from crystallographic data. I. RNA and B form DNA". *J.Molec.Biol.* **173**, 361-388.

Howlin, B., Moss, D.S. & Harris, G.W. (1989). "Segmented anisotropic refinement of bovine ribonuclease A by the application of the rigid-body *TLS* model". *Acta Cryst.* A**45**, 851-861.

Howlin, B., Butler, S.A., Moss, D.S., Harris, G.W. & Driessen, H.P.C. (1993). "TLSANL: *TLS* parameter-analysis program for segmented anisotropic refinement of macromolecular structures". *J.Appl. Cryst.* **26**, 622-626.

Johnson, C.K. (1970a). "The Effect of Thermal Motion on Interatomic Distances and Angles". In *Crystallographic Computing*, ed. F.R.Ahmed, Munksgaard, Copenhagen, 220-226.

Johnson, C.K. (1970b). "Generalized treatments for Thermal Motion". In *Thermal Neutron Diffraction*, ed. B.T.M.Willis, Oxford University Press: London, 132-136.

Johnson, C.K. & Levy, H.A. (1974). "Thermal-Motion Analysis Using Bragg Diffraction Data". *International Tables for X-ray* Crystallography, Vol. IV, eds. J.A.Ibers and W.C.Hamilton; Birmingham: Kynoch Press, 311-336.

Johnson, C.K. (1980). "Thermal motion analysis". In *Computing in Crystallography*, eds. R.Diamond, S.Ramaseshan, K.Venkatesan, Indian Academy of Sciences, Bangalor, India. pp.14.01-14.19

Kuriyan, J. & Weis, W.I. (1991). "Rigid protein motion as a model for crystallographic temperature

factors". *Proc.Natl.Acad.Sci. USA,* **88**, 2773-2777.

Merritt, E.A. (1999). "Comparing anisotropic displacement parameters in protein structres". *Acta Cryst.* D**55**, 1997-2004.

Moore, P.B. (2009) "On the relationship between diffraction Patterns and motion in macromolecular crystals". *Structure,* **17,** 1307-1315.

Moss, D.S., Tickle, I.J., Theis, O. & Wostrack, A. (1996). "X-ray analysis of domain motions in protein crystals". *Proceeding of the CCP4 Study Weekend. Macromolecular refinement*, eds. E.Dodson, M.Moore, A.Ralph, S.Bailey, Warrington: Daresbury Laboratory, 105-114.

Painter, J. & Merritt, E.A. (2005). "A molecular viewer for the analysis of *TLS* rigid-body motion in macromolecules". *Acta Cryst.* D**62**, 439-450.

Painter, J. & Merritt, E.A. (2006a). "Optimal description of a protein structure in terms of multiple groups undergoing *TLS* motion". *Acta Cryst.* D**61**, 465-471.

Painter, J. & Merritt, E.A. (2006b). "TLSMD web server for the generation of multi-group *TLS* models". *J.Appl. Cryst.* **39**, 109-111.

Pawley, G.S. (1963). "On the least-squares analysis of the rigid body vibrations of non-centrosymmetrical molecules". *Acta Cryst.* **16**, 1204-1208.

Pawley, G.S. (1964). "Least-squares structure refinement assuming molecular rigidity". *Acta Cryst.* **17**, 457-458.

Pawley, G.S. (1970). "Rigid-body molecular motion in crystals. The centre of libration". *Acta Cryst.* A**26**, 289-292.

Prince, E. & Finger, L.M. (1972). "Use of constraints on thermal motion in structure refinement of molecules with librating side groups". *Acta Cryst.* B**29**, 179-183.

Rae, A.D. (1975a). "Crystal structure refinement using a number of orthogonal axial systems". *Acta Cryst.* A**31**, 560-570.

Rae, A.D. (1975b). "Rigid-body motion in crystals - the application of constraints on the *TLS* model". *Acta Cryst.* A**31**, 570-574.

Scheringer, C. (1972a). "A lattice-dynamical treatment of the thermal-motion bond-length correction". *Acta Cryst.* A**29**, 616-619.

Scheringer, C. (1972b). "A lattice-dynamical bond-length correction for diatomic and triatomic molecules". *Acta Cryst.* A**29**, 619-628.

Scheringer, C. (1973). "A lattice-dynamics interpretation of molecular rigid-body vibration tensonrs". *Acta Cryst.* A**29**, 554-570.

Scheringer, C. (1978a). "The thermal-motion correction for bond angles". *Acta Cryst.* A**34**, 428-431.

Scheringer, C. (1978b). "Temperature factors for large librations of molecules. Expression in a general crystal metric and for any site symmetry". *Acta Cryst.* A**34**, 702-709.

Scheringer, C. (1978c). "Dynamic density and structure factors for rigid molecules with large librations". *Acta Cryst.* A**34**, 905-908.

Schomaker, V. & Trueblood, K.N. (1968). "On the rigid-body motion of molecules in crystals". *Acta Cryst.*

B**24**, 63-76.

Schomaker, V. & Trueblood, K.N. (1984). *Acta Cryst.* A**40**, C339.

Schomaker, V. & Trueblood, K.N. (1998). "Correlation of internal torsional motion with overall molecular motion in crystals". *Acta Cryst.* B**54**, 507-514.

Stec, B., Zhou, R. & Teeter, M.M. (1995). "Full-matrix refinement of the protein crambin at 0.83 A and 130 K". *Acta Cryst.* D**51**, 663-681.

Steiner, T. & Seanger, W. (1993). "Distribution of observed C-H bond lengths in neutron crystal structures and temperateure dependence of the mean values". *Acta Cryst.* A**49**, 379-384.

Sternberg, M.J.E., Grace, D.E.P. & Phillips, D.C. (1979). "Dynamic information from protein crystallography. An analysis of temperature factors from refinement of the hen egg-white lysozyme structure". *J.Molec.Biol.* **130**, 231-253.

Stuart, D.I. & Phillips, D.C. (1985). "Description of overall anisotropy in diffraction from macromolecular crystals". *Methods in Enzymology*, **115**, 117-142.

Sussman, J.L., Holbrook, S.R., Church, G.M. & Kim, S.-H. (1977). "A structure-factor least-squares refinement procedure for macromolecular structures using constrained and restrained parameters". *Acta Cryst.* A**33**, 800-804.

Sygusch, J. (1976). "Constrained thermal motion refinement for a rigid molecule with librating side groups". *Acta Cryst.* B**32**, 3295-3298.

Tickle, I. & Moss, D.S. (1999). "Probabilistic approach and geometric interpretation of the model and consequences". Notes from IUCr Cryst. Computing School, *http://public-1.cryst.bbk.ac.uk/~tickle/iucr99/iucrcs99.htm.*

Trueblood, K.N., Bürgi, H.-B., Burzlaff, H., Dunitz, J.D., Grammaccioli, C.M., Schulz, H.H., Shmueli, U., Abrahams, S.C. (1996). "Atomic displacement parameter nomenclature. Report of a subcomittee on atomic displacement parameter nomenclature". *Acta Cryst.* A**52**, 770-781.

Urzhumtseva, L.M. & Urzhumtsev, A.G. (1997) "Tcl/Tk based programs. II. CONVROT: program to recalculate different rotation descriptions". *J.Appl. Cryst.,* **30**, 402-410.

Winn, M.D., Isupov, M.N. & Murshudov, G.N. (2001) "Use of *TLS* parameters to model anisotropic displacements in macromolecular refinement". *Acta Cryst.,* **D57,** 122-133.

Zucker, F., Champ, P.C. & Merritt, E.A. (2010). "Validation of crystallographic models contaning *TLS* or other descriptors of anysotropy". *Acta Cryst.* D**66**, 889-900.

SOME OTHER RELEVANT ARTICLES

Aragao, D., Frazao, C., Sieker, L., Sheldrick, G.M., LeGall, J. & Carrondo, M.A. (2003). "Structure of dimeric cytochrome c3 from Desulfovibrio gigas at 1.2 A resolution". *Acta Cryst.* D**59**, 644-653.

Arnoux, B., Ducruix, A. & Prange, T. (2002). "Anisotropic behavious of the C-terminal Kunitz-type domain of the a3 chain of human type VI collagen at atomic resolution (0.9 A)". *Acta Cryst.* D**58**, 1252-1254.

Artymiuk, P.J., Blake, C.C.F., Grace, D.E.P., Oatley, S.J., Phillips, D.C., Sternberg, M.J.E. (1979). "Crystallographic studies of the dynamic properties of lysozyme". *Nature*, **280**, 563-568.

Blessing, R. (1986). "Hydrogen bonding and thermal vibrations in crystalline phosphate salts of histidine

and imidazole". *Acta Cryst.* B**42**, 613-621.

Bloch, F. (1932). "Zur Theorie des Austauschproblems und der Remanenzerscheinung der Ferromagnetika" ("On the theory of the exchange problem and the remanence phenomenon of ferromagnets"). *Zeitschrift für Physika.* **74**, 295-335.

Bürgi, H.B. & Capelli, S.C. (2000) "Dynamics of molecules in crystals from multitemperature anisotropic displacement parameters. I. Theory". *Acta Cryst.* A**56**, 403-412.

Byrom, P.G., Hoffmann, S.E. & Lucas, B.W. (1989). "MORGUE, a new powder diffraction profile refinement program with control-file factility to include structural and rigid-body thermal-motion constraints". *J.Appl. Cryst.* **22**, 629-633.

Capelli, S.C., Fortsch, M. & Bürgi, H.B. (2000) "Dynamics of molecules in crystals from multitemperature anisotropic displacement parameters. II. Application to benzene (C6D6) and urea [OC(NH)2]". *Acta Cryst.* A**56**, 413-424.

Chaudhry, C., Horwich, A.L., Brunger, A.T. & Adams, P.D. (2004). "Exploring the structural dynamics of the E.coli chaperonin GroEL using translation-libration-screw crystallographic refinement of intermediate states". *J.Molec.Biol.* **342**, 229-245.

Cochran, W. (1951a). "The structures of pyrimidines and purines. V. The electron density distribution in adenine hydrochloride". *Acta Cryst.* **4**, 81-92.

Cochran, W. (1951b). "Some Properties of the (Fo-Fc)-Synthesis". *Acta Cryst.* **4**, 408-411.

Diamond, R. (1990). "On the Use of Normal Modes in Thermal Parameter Refinement: Theory and Application to the Bovine Pancreatic Trypsin Inhibitor". *Acta Cryst.* A**46**, 425-435.

Ducros, V.M.-A., Lewis, R., Verna, C.S., Dodson, E.J., Leonard, G., Turkenburg, J.P., Murshudov, G.N., Wilkinson, A.J. & Brannigan, J.A. (2001). "Crystal structure of GerE, the ultimate transcriptional regulator of spore formation in Bacillus subtilis". *J.Molec.Biol.* **306**, 759-771.

Garcia, P., Dahaoui, S., Fertey, P., Wenger, & E., Lecomte, C. (2005). "Crystallographic investigation of temperature-induced phase transition of the tetrathiafulvalene-p-bromanil, TTF-BA charge transfer complex". *Phys.Rev.* B, **72**, 104115.

Harata, K., Abe, Y. & Muraki, M. (1999). "Crystallographic evaluation of internal motion of human alpha-lactalbumin refined by full-matrix least-squares method". *J.Molec.Biol.* **287**, 347-358.

Harata, K., Abe, Y. & Muraki, M. (1998). "Full-matrix least-squares refinement of lysozymes and analysis of anisotropic thermal motion". *Proteins Struct. Funct. Genet.* **30**, 232-243.

Harata, K. (2003). "Crystallographic analysis of the thermal motion of the inclusion complex of cyclomaltoheptaose (beta-cyclodextrin) with hexamethylentetramine". *Carbohydrate Res.* **338**, 353-359.

Harris, G.W., Pickersgill, R.W., Howlin, B. & Moss, D.S. (1992). "The segmented anisotropic refinement of monoclinic papain by the application of the rigid-body *TLS* model and comparison to bovine ribonuclease A". *Acta Cryst.* B**48**, 67-75.

Hirshfeld, F.L. & Shmueli, U. (1972). "Covariances of thermal parameters and their effect on rigid-body calculations". *Acta Cryst.* A**28**, 648-652.

Holbrook, S.R., Dickerson, R.E. & Kim, S.-H. (1985). "Anisotropic thermal parameter refinement of the DNA dodecamer CGCGAATTCGCG by the segmented rigid-body method". *Acta Cryst.* B**41**, 255-262.

Hummel, W., Raselli, A. & Burgi, H.-B. (1990). "Analysis of atomic displacement parameters and molecular motion in crystals". *Acta Cryst.* B**46**, 683-692.

Johnson, C.K. (1970). "An Introduction to Thermal Motion Analysis". In *Crystallographic Computing*, ed. F.R.Ahmed, Munksgaard, Copenhagen, 207-219.

Kidera, A. & Go, N. (1990). "Refinement of protein dynamic structure: normal mode refinement". *Proc.Natl.Acad.Sci. USA,* **87**, 3718-3722.

Kidera, A. & Go, N. (1992). "Normal mode refinement: crystallographic refinement of protein dynamic structure. I. Theory and test by simulated diffraction data". *J.Molec.Biol.* **225**, 457-475.

Kidera, A., Matsushima, M. & Go, N. (1994). "Dynamic structure of human lysozyme derived from X-ray crystallography: normal mode refinement". *Biophys.Chem.* **50**, 25-31.

Merritt, E.A. (1999). "Expanding the model: anisotropic displacement parameters in protein structure refinement". *Acta Cryst.* D**55**, 1109-1117.

Moroz, O.V., Antson, A.A., Murshudov, G.N., Maitland, N.J., Dodson, G.G., Wilson, K.S., Skibshoj, I., Lukanidin, E.M., Bronstein, I.B. (2001) "The three-dimensional structure of human S100A12". *Acta Cryst.,* **D57,** 20-29.

Murshudov, G.N., Vagin, A.A., Lebedev, A., Wilson, K.S. & Dodson, E.J. (1999) "Efficient anisotropic refinement of macromolecular structures using FFT". *Acta Cryst.,* **D55,** 247-255.

Papiz, M.Z. & Prince, S.M. (1996). "Group anisotropic thermal parameter refinement of the light-harvestingcomplex from purple bacteria *Rhodopseudomonas acidophila*". *Proceeding of the CCP4 Study Weekend. Macromolecular refinement*, eds. E.Dodson, M.Moore, A.Ralph, S.Bailey, Warrington: Daresbury Laboratory, 115-123.

Papiz, M.Z. & Prince, S.M. (2003). "The structure and thermal motion of the B800-850 LH2 complex from Rps.acidophila at 2.0 A resolution and 100 K: new structural features and functionally relevant motions". *J.Molec.Biol.* **326**, 1523-1538.

Pawley, G.S. (1965). "Refinement of azulene assuming rigid-body thermal motion". *Acta Cryst.* **18**, 560-561.

Pawley, G.S. (1966). "Further refinements of some rigid boron compounds". *Acta Cryst.* **20**, 631-638.

Pawley, G.S. (1968). "Anisotropic temperature factors and screw rotation coefficients from a lattice dynamical viewpoint". *Acta Cryst.* B**24**, 485-486.

Pawley, G.S. (1970). "The use of molecular lattice dynamical motion". *Crystallographic Computing*, eds. Ahmed, F.R., Hall, S.R., Huber, C.P., Munksgaard, Copenhagen, 243-249.

Perez, J., Faure, P. & Benoit, J.-P. (1996). "Molecular rigid-body displacements in a tetragonal lysozyme crystal confirmed by X-ray diffuse scattering". *Acta Cryst.* D**52**, 722-729.

Phillips, C., Gover, S. & Adams, M.J. (1995). "Structure of 6-phosphogluconate dehydrogenase refined at 2 A resolution". *Acta Cryst.* D**51**, 290-304.

Raaijmakers, H., Toro, I., Birkenbihl, R., Kemper, B. & Suck, D. (2001). "Conformational flexibility in T4 endonuclease VII revealed by crystallography: implications for substrate binding and cleavage". *J.Molec.Biol.* **308**, 311-323.

Sali, A., Veerapandian, B., Cooper, J.B., Moss, D.S., Hofmann, T. & Blundell, T.L. (1992) "Domain flexibility in aspartic proteinases". *Proteins Struct. Funct. Genet.* **12**, 158-170.

Sarma, G.N., Savvidis, S.N., Becker, K., Schirmer, M., Schirmer, R.H. & Karplus, P.A. (2003). "Glutathione reductase of the malarial parasite Plasmodium falciparum: crystal structure and inhibitor development". *J.Molec.Biol.* **328**, 893-907.

Schneider, T.R. (1996). "What we can learn from anisotropic temperature factors?". *Proceeding of the CCP4 Study Weekend. Macromolecular refinement*, eds. E.Dodson, M.Moore, A.Ralph, S.Bailey, Warrington: Daresbury Laboratory, 133-144.

Sternberg, M.J.E., Grace, D.E.P. & Phillips, D.C. (1979). "Dynamic information from protein crystallography. An analysis of temperature factors from refinement of the hen egg-white lysozyme structure". *J.Molec.Biol.* **130**, 231-253.

Trueblood, K.N. (1978). "Analysis of molecular motion with allowance for intramolecular torsion". *Acta Cryst.* A**34**, 950-955.

Verlinde, C.L. & De Ranter, C.J. (1989) "Furan revisited : when to avoid ab initio studies on crystal structures". *J.Molec.Struct. (Therochem.)*, **187**, 161-167.

Wilson, C. (2000) "Single crystal neutron diffraction from molecular materials". *Singapore : World Scientific Publishing*.

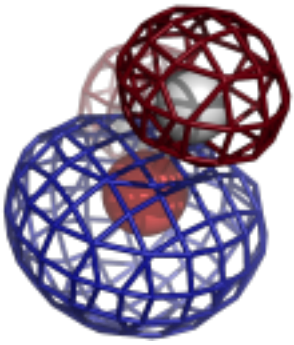
Wilson, M.A. & Brunger, A.T. (2000) "The 1.0 A crystal structure of Ca2+ -bound calmodulin: an analysis of disorder and implications for functionally relevant plasticity". *J.Molec.Biol.* **301**, 1237-1256.

Wilson, M.A. & Brunger, A.T. (2003) "Macromolecular *TLS* refinement in REFMAC at moderate resolutions". *Acta Cryst.* D**59**, 1782-1792.

Winn, M.D., Murshudov, G.N. & Papiz, M.Z. (2003) "Macromolecular *TLS* refinement in REFMAC at moderate resolutions". *Methods in Enzymology,* **374,** 300-321.

Yousef, M.S., Fabiola, F., Gattis, J.L., Somasundaram, T. & Chapman, M.S. (2002) "Refinement of the arginine kinase transition-state analogue complex at 1.2 A resolution: mechanistic insights". *Acta Cryst.* D**58**, 2009-2017.

VOLUME TWO



COMPUTATIONAL CRYSTALLOGRAPHY NEWSLETTER

JULY MMXI

DATA VIEWER, MR-ROSETTA, LYSOZYME, SPOTFINDER

Table of Contents

Editor

Nigel W. Moriarty, NWMoriarty@LBL.Gov

Contributors

P. D. Adams, P. V. Afonine, D. Baker, G. Bunkóczi, F. DiMaio, N. Echols, J. J. Headd, R. W. Grosse-Kunstleve, D. Lucent, N. W. Moriarty, J. Newman, T. S. Peat, R. J. Read, D. C. Richardson, J. S. Richardson, N. K. Sauter, T. C. Terwilliger

PHENIX News

New releases

The default behavior of KiNG has been improved when loading electron density maps. Now when KiNG is launched from the refinement and validation GUIs, maps are automatically loaded and presented using the Coot-default color scheme. This Coot-default color scheme can also be accessed when opening maps with the command-line *phenix.king* by using the `-phenix` flag. Finally, KiNG includes and applies better map presets for 2Fo-Fc, Fo-Fc and anomalous maps.

A new graphical tool for visualization of reciprocal-space data is now available in *PHENIX* and is discussed in the short communications section starting on page 88.

Open source spotfinding code has been released in the cctbx for use at beamlines. A description of this extremely fast program is in the short communications section on page 93.

New features

The recent inclusion of Rosetta in *PHENIX* via the *phenix.mr_rosetta* has had many new features added. Hints on how to using it to its full potential are included in the short communications on page 94.

Refinement in *PHENIX* continues to be improved. An article about improved target

The Computational Crystallography Newsletter (CCN) is a regularly distributed electronically via email and the PHENIX website, www.phenix-online.org/newsletter. Feature articles, meeting announcements and reports, information on research or other items of interest to computational crystallographers or crystallographic software users can be submitted to the editor at any time for consideration. Submission of text by email or word-processing files using the CCN templates is requested.

weight optimization including use of parallel processing begins on page 99.

Crystallographic meetings and workshops

PHENIX User's Workshop, 22 September, 2011

A *PHENIX* user's workshop is being planned in Durham, North Carolina on the 22nd of September for local area students, postdocs and other interested parties. Please contact Jeff Headd at JJHeadd@lbl.gov for further information.

IUCr Commission on Crystallographic Computing, Mieres 2011, Crystallographic Computing School, 16-22 August, 2011

A crystallographic computing school run by the IUCr Commission on Crystallographic Computing will be held in Oviedo, Spain from the 16th to the 22nd of August 2011. *PHENIX* developers will be giving lectures and available for questions.

Expert advice

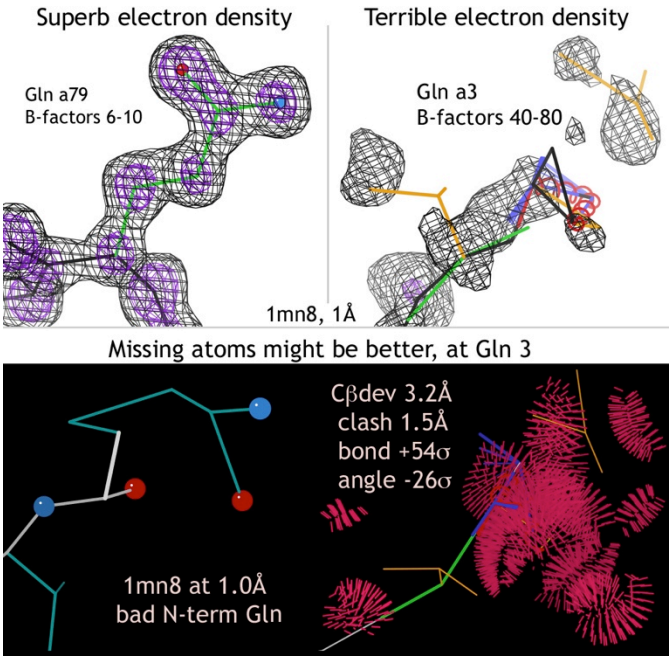
Fitting Tips

Vincent Chen, Christopher Williams and Jane Richardson, Duke University

Even very high-resolution structures are prone to a few types of systematic error that would be better avoided.

When you're lucky enough to be at 1Å resolution, the electron density is gorgeous, unambiguous and delightful in most places - like the upper left figure of Gln 79 in 1mn8 (2Fo-Fc with contours at 1.2 and 3.0 s). It is very tempting, then, to strongly down weight the geometry terms, or even turn them off altogether. That produces a good model in the well-ordered regions with low B-factors. [Note that normal weighting would also produce a good model there.]

However, even at ultra high resolution there are almost always a few disordered places with very poor density and high B-factors, such as at Gln 3 in the upper right figure. A



reasonable peptide and sidechain were probably fit here initially. Then refinement tried too hard to move atoms into what little density it could find, resulting in the violently distorted model at lower left. As shown at upper and lower right, there are bond length outliers up to 56σ, bond angle outliers up to 26σ, a 3.2Å Cβ deviation, many steric clashes with all-atom overlap up to 1.5Å, 2 bad rotamers and a Ramachandran outlier. It seems clear that no one looked at this region in the final model, because surely they would have been motivated to do something about it.

This example is an extreme case, but not an unusual problem. The tips here are:

1) Keep a non-negligible weight on the geometry term (except perhaps in a local test that won't be deposited). B-factor dependent weights would be a desirable option.
2) Don't rely on overall rmsd for bond lengths and angles - always look at map and model for the worst individual deviations and check out the chain termini.
3) Perhaps residue 3 should have been omitted as well as 1 and 2. If you do choose to fit into very poor density, enforce acceptable geometry and conformation.

FAQ

How do I make composite omit maps in *PHENIX*?

This can be achieved using the GUI by choosing the "AutoBuild – create omit map" option under the "Maps" tab in the main GUI window. It is also very easy using the command line. To make a simple omit map of the model, the following options can be used with the Autobuild command:

```
phenix.autobuild data=data.mtz model=coords.pdb composite_omit_type=simple_omit
```

Coefficients for the output omit map will be in the file resolve_composite_map.mtz in the subdirectory OMIT/ . A simulated annealing omit map can be generated by changing the type:

```
phenix.autobuild data=data.mtz model=coords.pdb composite_omit_type=simple_omit
```

The region of the omit map can be specified by adding the "omit_box_pdb" option thus:

```
phenix.autobuild data=data.mtz model=coords.pdb composite_omit_type=simple_omit
                  omit_box_pdb=target.pdb
```

Once again, coefficients for the output omit map will be in the file resolve_composite_map.mtz in the subdirectory OMIT/ . An additional map coefficients file omit_region.mtz will show you the region that has been omitted. (Note: be sure to use the weights in both resolve_composite_map.mtz and omit_region.mtz).

More information about maps can be found online.

A lightweight, versatile framework for visualizing reciprocal-space data

Nathaniel Echols[a] and 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: nathaniel.echols@gmail.com

Introduction

For diagnostic and educational purposes, it is often useful to display data from reflection files in graphical format. In macromolecular crystallography, the CCP4 program hklview [1] has been the primary tool for this, but it is limited to 2D pseudo-precession camera "slices" through reciprocal space. Ongoing work on assessing data pathologies and improving refinement and map quality in PHENIX, especially at low resolution, necessitated the development of a simple program capable of both 2D and 3D views of reflection data.

The complete program, phenix.data_viewer, is written as a standalone wxPython app, but was designed to be easily embedded in other programs and potentially re-used in other contexts. The 3D viewer relies entirely on OpenGL for rendering, using a custom set of Python OpenGL bindings in the gltbx module of CCTBX. The 2D viewer uses low-level wxPython drawing commands on a blank canvas, with the underlying native graphics API performing the actual work. Both views support saving screen captures of the canvas in PNG format. In principle these frontends could be replaced with GUI-independent output formats, for instance using the GD drawing library [2], facilitating use in web servers.

The 2D and 3D displays are nearly identical with respect to input and options, but operate independently of each other. Both have a control panel with all user-adjustable parameters, plus information on the last clicked reflection (Figures 1 and 2). We have attempted to provide the user with

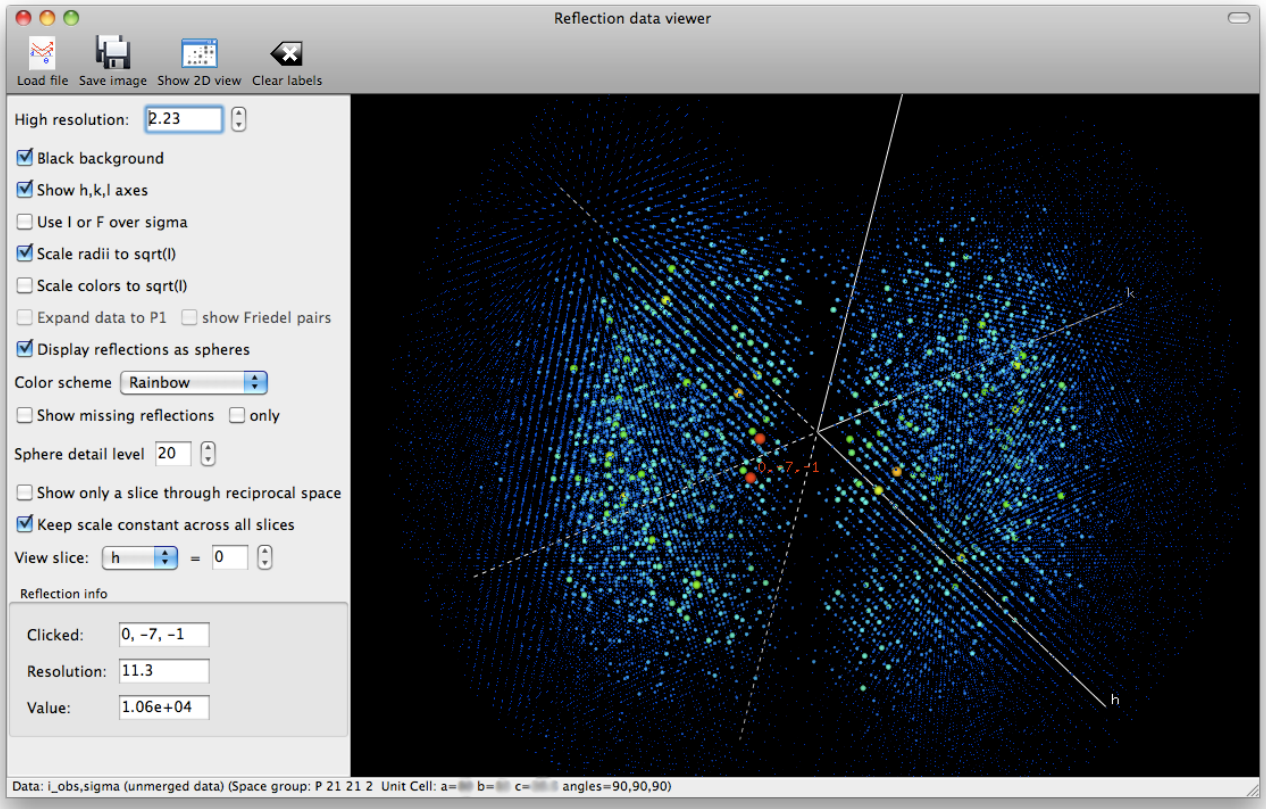


Figure 1. 3D viewer, displaying contents of a Scalepack file processed with the "no merge original index" macro. The dataset was collected as a 100-degree wedge with inverse beam geometry.

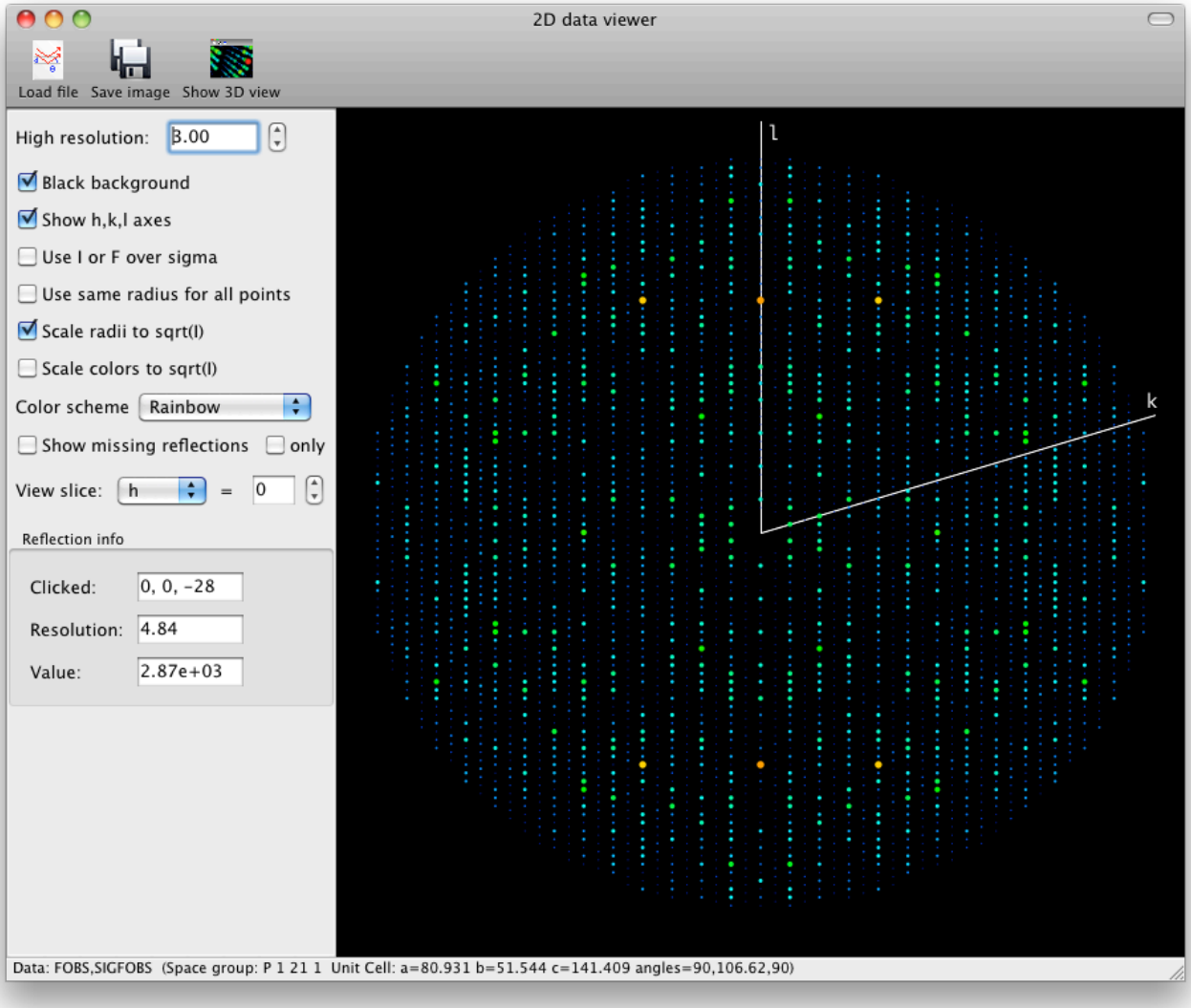
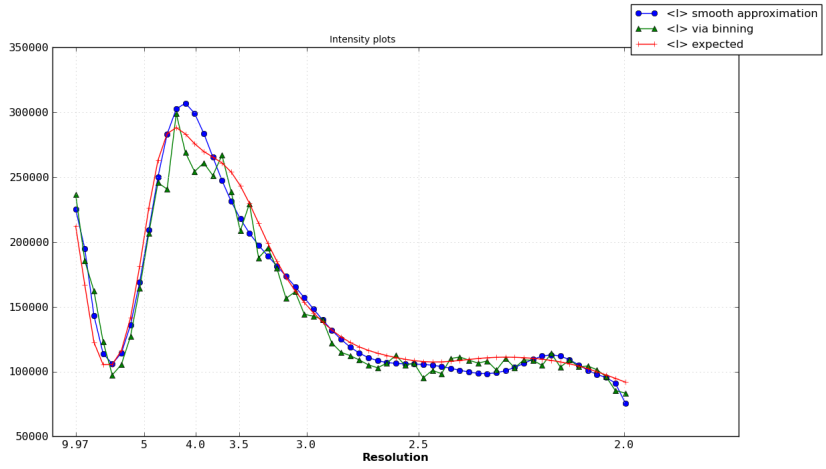


Figure 2. 2D viewer, showing the 0kl section from a dataset with pseudo-translational symmetry (PDB ID: **3ori**, truncated at 3.0Å resolution), resulting in alternating strong and weak columns of spots. The effect is especially noticeable when viewing successive sections along the *k* axis, as shown in Figure 4. Also clearly visible is the sharp dip in mean intensity around 7Å resolution characteristic of protein crystals; the Wilson plot from *phenix.xtriage* is shown below for comparison.



a large amount of control over how the data are displayed. By default, both point size and color are used to convey the relative magnitude of reflections, using a variety of scaling options. We have found this to be more intuitive than the monochromatic or grayscale rendering, especially in 3D where the number of reflections may be in the tens (or hundreds) of thousands. In addition to the reflections actually present in the input file, missing reflections may also be visualized (Figure 3), either alongside the real data or independently. This may be useful for judging errors and/or pathologies in data collection [3].

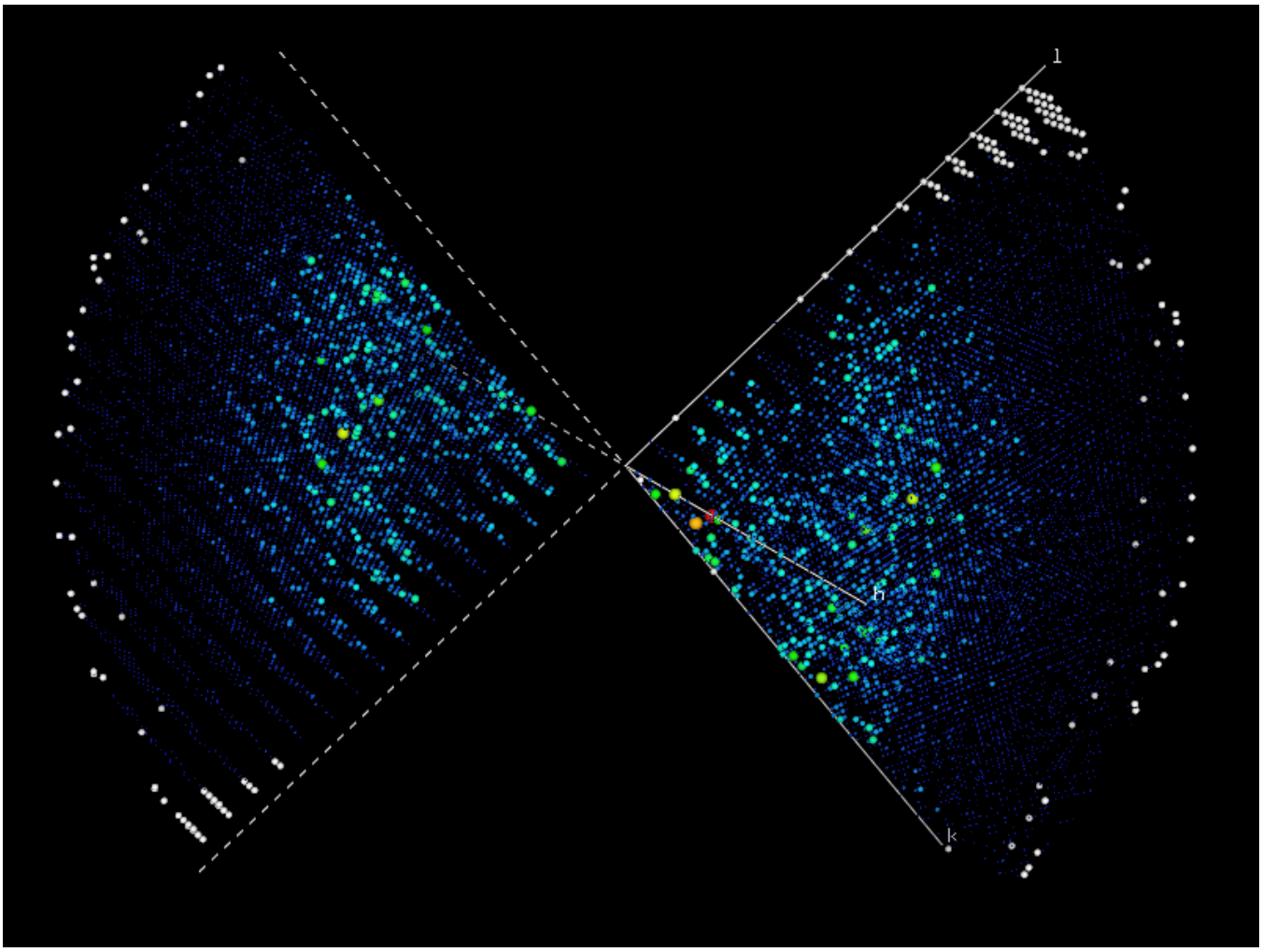


Figure 3. The same dataset shown in Figure 1 merged to contain only symmetry-unique data, with missing reflections displayed as white spheres.

Because CCTBX deals primarily with fully processed (merged and scaled) data, support for unmerged reflection files is currently uneven. By default, the 3D viewer displays only those reflections in the input file, although the controls allow expansion to P1 symmetry and generation of Friedel mates. The 2D view approximates the behavior of hklview: when unmerged data are provided, it will only display the original reflections, potentially leaving some regions empty, but automatically expands merged data to cover all of reciprocal space. Visualization of missing reflections in unmerged data is limited to the reciprocal-space asymmetric unit, which may lead to visual artifacts depending on the oscillation range (Figure 3).

Although visualizations of this sort are useful for interpreting many properties of reciprocal space, especially with regards to missing and/or pathological data (Figure 4), they are not intended to directly represent the data as they appear in the actual diffraction experiment [4]. In particular, the size of the spheres or circles representing individual reflections has no relationship to the apparent "size" of the reflection as captured on an area detector, which is actually determined by factors such as crystal mosaicity, beam divergence, etc. However, a possible future enhancement is the addition of an Ewald sphere and display of its intersection with the reflections as it would appear on an area detector, given user-defined parameters for mosaicity and other experimental properties.

Availability

phenix.data_viewer is included with all PHENIX installers starting with build 780, and can be run from

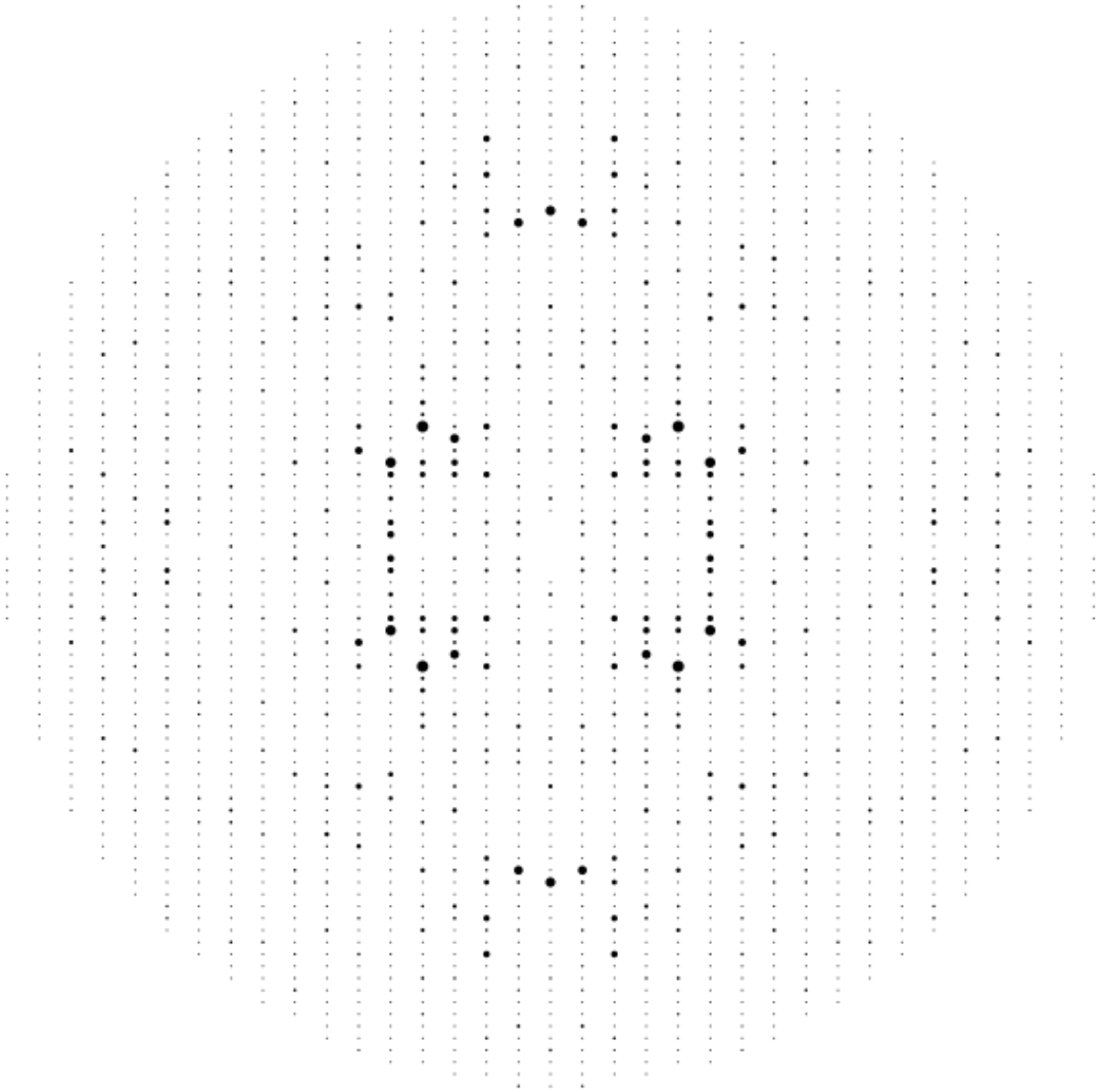


Figure 4. The h0l and h1l sections of the dataset with translational NCS shown in Figure 2. The images are displayed in black-and-white for clarity, but the effect is more striking using the default settings when viewed interactively.

the command line or from the PHENIX GUI (under "Reflection tools"). It is also available in standalone CCTBX builds, but must be compiled from source due to the wxPython and OpenGL dependencies. The code is available as unrestricted open-source under the CCTBX license, in the module `crys3d.hklview`.

Acknowledgments

We thank Jaroslaw Kalinowski for suggesting the 3D viewer concept.

Notes

1. Originally written by Phil Evans, and described at http://www.ccp4.ac.uk/html/hklview.html. Prof. Evans has called our attention to an excellent replacement for hklview called ViewHKL (available as a standalone download at http://www.ccp4.ac.uk/prerelease/) written by him and Eugene Krissinel.

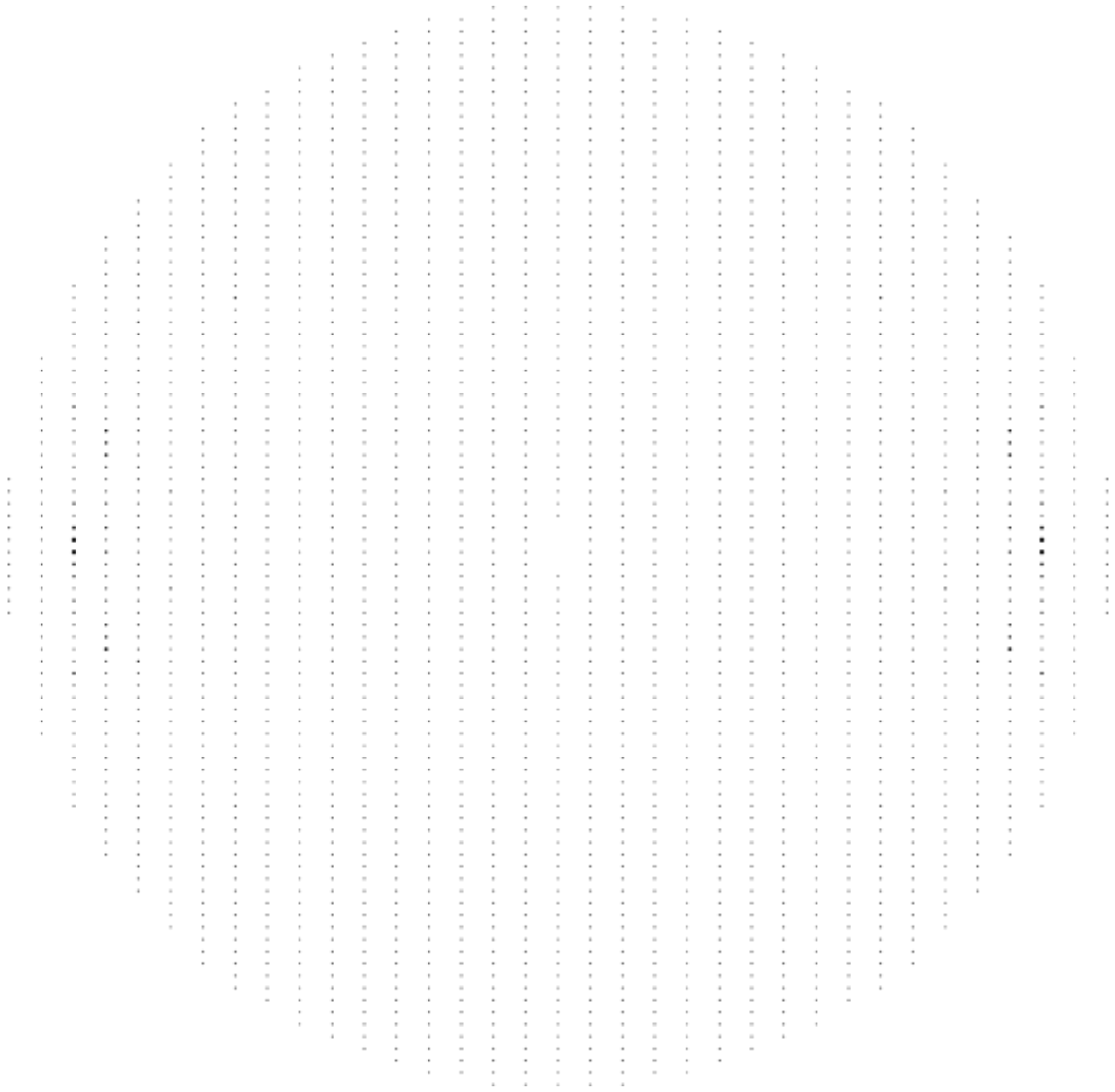


Figure 4. (continued)

Notes (continued)

2. http://newcenturycomputers.net/projects/gdmodule.html

3. See also Urzhumtsev, A. G. (1991). Acta Cryst. A47, 794-80 and Urzhumtseva & Urzhumtsev (2011) J. Appl. Cryst. vol. 44 part 4.

4. The program labelit.precession_photo, described in the previous issue of this newsletter, generates similar 2D slices of reciprocal space using the raw diffraction images directly.

An extremely fast spotfinder for real-time beamline applications

Nicholas K. Sauter
Lawrence Berkeley National Laboratory, Berkeley, CA 94720
Correspondence email: NKSauter@LBL.Gov

The Bragg spot analyzer described last year [*CCN* **1**, 18-23 (2010)] has been enhanced for high-throughput applications such as diffraction mapping and continual monitoring for radiation damage. The software provides raw measurements that can be harnessed by beamline developers for graphical display and instrument control. Recent work improves the program's output and performance.

Most significantly, the spotfinder package is now released under the cctbx open source license (see http://cctbx.sf.net), which now makes it independent of the packages *LABELIT* and *PHENIX*, and accessible to all beamlines worldwide. Succinct instructions for download and installation are posted at http://cci.lbl.gov/labelit, under the link for "Beamline Server". New cctbx code is tested, packaged and released on a near-daily basis; interested users are encouraged to either contact the authors with feature requests or join the cctbx open-source development group.

High-throughput performance is achieved by delegating the analysis of individual diffraction images to separate processors on a multicore CPU. The overall software architecture includes a "client" process (such as the beamline graphical user interface), which contacts the multiprocessing "server" whenever a Bragg spot analysis is required for a new image. The client, which is developed by the beamline group, can be implemented in any language (Java, TCL, Python, etc.) that supports the http: protocol needed to contact the server. In fact, it is straightforward to test the server with a standard Web browser, by requesting a URL that includes the file name of the diffraction image and any desired processing options. A simple mapping is used to convert the Unix command line for the underlying spotfinder program into a URL for the spotfinder server. Two separate implementations of the spotfinder server are now released, one using all-Python tools, and a second that uses the Apache httpd Web server for multiprocess control, within which a Python interpreter is provided by the mod-python package. The two servers give identical data analysis and similar performance, but there are some tradeoffs: the Python server is slightly easier to download and install, but the Apache/mod-python is superior in its ability to tune for peak performance. We observe the following general performance benchmarks under 64-bit Linux:

OS	Fedora 8	Fedora 13
CPU	Intel Xeon	AMD Opteron
Clock speed	2.93 GHz	2.20 GHz
# of processors	16 cores	48 cores
Overall throughput	8.9 frames/s	25 frames/s

These tests involved the processing of 720 Pilatus-6M images, with diffraction spots identified out to the corner of the detector.

Finally, many new features have been added to the spotfinder. Diffraction strength can now be summarized as a function of resolution bin, which should be of particular interest for monitoring Bragg spot quality over time from a given specimen. Additional quality measures have been added, such as background level, and signal-to-noise expressed as $I/\sigma(I)$. Numerous additional options are available for controlling the algorithm, all of which are documented on the Web page. The spotfinder work was funded under NIH/NIGMS grant numbers R01GM077071 and R01GM095887.

Hints for running *phenix.mr_rosetta*

Thomas C. Terwilliger[1], Frank DiMaio[2], Randy J. Read[3], David Baker[2], Gábor Bunkóczi[3], Paul D. Adams[4], Ralf W. Grosse-Kunstleve[4], Pavel V. Afonine[4], Nathaniel Echols[4]

1 Los Alamos National Laboratory, BioScience Division and Los Alamos Institutes, Los Alamos, NM 87545
2 University of Washington, Department of Biochemistry, Seattle, WA, 98195, USA
3 University of Cambridge, Department of Haematology, Cambridge Institute for Medical Research, Cambridge, CB2 0XY, UK
4 Lawrence Berkeley National Laboratory, One Cyclotron Road, Bldg 64R0121, Berkeley, CA 94720, USA

Introduction

A combination of structure-modeling tools available in *Rosetta* (Qian et al., 2007, DiMaio et al., 2009) and the molecular replacement (Read, 2001) and model-building (Terwilliger et al, 2008) tools available in *Phenix* (Adams et al., 2010) has been very useful in determining structures by molecular replacement (DiMaio et al., 2011). The approach is most appropriate for cases where the best template is somewhat too different from the target structure to be useful in conventional molecular replacement. In a previous *Phenix* newsletter we summarized the *phenix.mr_rosetta* tool and how to use it. The basic idea is that *Rosetta* modeling can be useful at two stages in molecular replacement. First, it can be useful in improving a template before using it as a search model. Second, it can be useful in improving a model that has been placed in the unit cell and where an electron density map is available. By combining *Rosetta* with *Phenix* tools, the range of models useful for molecular replacement can be expanded. Here we give some hints for getting the most out of this approach.

Downloading templates from the PDB based on an alignment file

One useful feature of *phenix.mr_rosetta* is the ability to use an alignment file that lists templates available in the PDB and alignments of those templates to the target structure. These alignment files can be obtained from the *hhpred* server (http://toolkit.tuebingen.mpg.de/hhpred; Soding, 2005). You can supply one or more alignment files to *phenix.mr_rosetta* with the keyword

```
hhr_files=my_hhr_file
```

and you can specify how many of the templates in each file (e.g. 5) are to be used with

```
number_of_models=5
```

It is a good idea to have a close look at the alignment file and choose how many models to download based on the range of sequence identities in the file. If there are a few models with high (>40%) identities, just use those. If there are many templates with similar sequence identities (and over similar parts of the target sequence) then you might want to include many of them, particularly if the sequence identity is low (<25%).

If you want to have more control over your search models, then you can download them yourself from the PDB and edit them with the *phenix.sculptor*. Alternatively you can download and edit them simultaneously with *phenix.mr_model_preparation.* Then you can specify these as search models with

```
search_models="model1.pdb model2.pdb"
```

and *phenix.mr_rosetta* will use each of these in turn as a search model.

Automatic searching for multiple NCS copies

You can control whether *phenix.mr_rosetta* checks for variable numbers of NCS copies in the asymmetric unit with the parameter

```
ncs_copies="1 2 4"
```

which will instruct *phenix.mr_rosetta* to search (in separate runs) for 1, 2, and 4 copies. You can also say,

```
ncs_copies=None
```

which will try all plausible values of ncs_copies. This can be convenient, but you might want to instead run 3 separate runs, specifying one value of ncs_copies in each. The reason this may be a good idea is that *phenix.mr_rosetta* does not stop when a satisfactory solution is found. Instead it will complete all the jobs and then report the best one. So if the job with `ncs_copies=4` takes a really long time (as it might if there are not actually 4 copies) then the whole *phenix.mr_rosetta* job would take a long time to complete.

Improving templates with *Rosetta* to use as search models in molecular replacement

One of the uses of *phenix.mr_rosetta* is to carry out homology-modeling of a template before using it as a search model. You can do this automatically during a *phenix.mr_rosetta* run with the keywords

```
run_prerefine=True
number_of_prerefine_models=1000
```

Typically you would want to generate about 1000-2000 models with *Rosetta.* Then the best model will be used in the following steps. Generating models at this stage with Rosetta does not take too long; a 150-residue protein might take about 5 minutes for each model.

Note that it is best to specify the number of `ncs_copies` if you use `run_prerefine`. If you do not, then you may end up running several parallel jobs, each of which is independently carrying out prerefinement on the same input model (to be used later with different numbers of ncs copies). Once you have run your job with one value of `ncs_copies`, you can just use the best prerefined model from that job as a search model in your other runs.

If you just want to run *Rosetta* rebuilding on a template and you don't want to do anything else, you can use a simple command to do this:

```
phenix.mr_rosetta \
  seq_file=seq.dat \
  search_models=coords1.pdb \
  run_prerefine=True \
  number_of_prerefine_models=1
```

Your prerefined model(s) will be listed in

```
MR_ROSETTA_1/GROUP_OF_PLACE_MODEL_1/RUN_FILE_1.log
```

and you can pick the best of these (most negative score, listed first).

Fragment files for *Rosetta*

If your model has gaps in it, then you will need to provide fragment files for *Rosetta* to use in filling in those gaps. If your chain has 650 residues or fewer, then this is fairly straightforward, and you can paste your sequence into the *Robetta* fragment server (http://robetta.bakerlab.org/fragmentsubmit.jsp; Chivian et al., 2003).

If your chain has more than 650 residues then you will need to break it up into segments and submit separate requests to the fragment server for each segment. Then you will get several 3-mer and 9-mer fragments files, one for each piece that you submit. You can then simply paste these together after editing all but the first to fix the residue numbers. To edit the files just use

```
phenix.offset_robetta_resid \
  <fragment_file_name> \
  <new_fragment_file_name> \
  <offset-for-residue numbers>
```

If you have multiple chain types in your structure then you will want to have a separate set of fragments files for each chain type. You can specify these with the keywords *fragment_files_chain_list*, *fragment_files_3_mer_by_chain*, and *fragment_files_9_mer_by_chain* instead of the keyword *fragment_files*.

Use *fragment_files_chain_list* to define which chain ID each of your *fragment_files_3_mer_by_chain* and *fragment_files_9_mer_by_chain* go with. Note that you only need one set of fragments files for each unique chain. So if chains A and C are the same, you just need to specify fragments for chain A.

Testing your installation of Rosetta and *phenix.mr_rosetta*

As a run of *phenix.mr_rosetta* can take a long time (hours to days or even weeks depending on how many models you search for and how many processors you have available) you may want to make sure everything is working properly before you start. You can test that both *Rosetta* and *phenix.mr_rosetta* work properly with the command

```
phenix_regression.wizards.test_command_line_rosetta_quick_tests
```

This takes about 15 minutes and will end with "OK" if everything is all right.

Running *phenix.mr_rosetta* on a cluster

You probably will want to run *phenix.mr_rosetta* on a cluster as it can take so much computational time to run. You can run on a Sun Grid Engine, Condor, or other cluster. If you run on a Sun Grid Engine (SGE) cluster, you only need to specify two keywords. The first tells *phenix.mr_rosetta* how to submit a job:

```
group_run_command=qsub
```

If your job submission is more complicated you can specify that:

```
group_run_command="/etc/run/qsub -abc"
```

You can then specify how many processers are to be used:

```
nproc=200
```

Note that *phenix.mr_rosetta* will submit individual jobs to the queue, not array jobs. This means that many jobs may be submitted.

On a condor cluster, you can specify

```
group_run_command=condor_submit
```

instead of "qsub".

On other clusters and supercomputers job submission may be more complicated. However you can control how it is done with the group_run_command keyword and with the keyword queue_commands. For example on a PBS system you

```
queue_commands='#PBS -N mr_rosetta'
queue_commands='#PBS -j oe'
queue_commands='#PBS -l walltime=03:00:00'
queue_commands='#PBS -l nodes=1:ppn=1'
```

When *phenix.mr_rosetta* actually submits a job, these commands will appear at the top of the script that is submitted (just after the definition of the shell to use), like this:

```
#!/bin/sh
#PBS -N mr_rosetta
#PBS -j oe
#PBS -l walltime=03:00:00
#PBS -l nodes=1:ppn=1
cd /home/MR_ROSETTA_3/GROUP_OF_PLACE_MODEL_1
sh /home/ MR_ROSETTA_3/GROUP_OF_PLACE_MODEL_1/RUN_FILE_1.sh
```

Finding your results with *phenix.mr_rosetta*

When *phenix.mr_rosetta* has completed you can find the best model and map by looking at the end of the

log file that has been written. You should see something like:

```
Results after repeat_mr_rosetta:

ID: 306
R/Rfree:   0.24 /   0.27
MODEL:
/net/omega/raid1/scratch1/terwill/blind_tests/all_cases/mr_rosetta_from_sta
rt/1_ag9603/MR_ROSETTA_6/ONE_REPEAT_1/RUN_1/GROUP_OF_AUTOBUILD_1/RUN_2/Auto
Build_run_1_/cycle_best_2.pdb

MAP COEFFS
/net/omega/raid1/scratch1/terwill/blind_tests/all_cases/mr_rosetta_from_sta
rt/1_ag9603/MR_ROSETTA_6/ONE_REPEAT_1/RUN_1/GROUP_OF_AUTOBUILD_1/RUN_2/Auto
Build_run_1_/cycle_best_2.mtz

Writing solutions as csv to
/net/omega/raid1/scratch1/terwill/blind_tests/all_cases/mr_rosetta_from_sta
rt/1_ag9603/MR_ROSETTA_6/repeat_results.csv
Saved overall mr_rosetta results in
/net/omega/raid1/scratch1/terwill/blind_tests/all_cases/mr_rosetta_from_sta
rt/1_ag9603/MR_ROSETTA_6/repeat_results.pkl

To see details of these results type
    phenix.mr_rosetta
mr_rosetta_solutions=/net/omega/raid1/scratch1/terwill/blind_tests/all_case
s/mr_rosetta_from_start/1_ag9603/MR_ROSETTA_6/repeat_results.pkl
display_solutions=True
```

This will be the model with the lowest R value obtained. If you want to see information about all the models (including intermediate models produced) then you can use the command that is listed at the end of this run:

```
 phenix.mr_rosetta
mr_rosetta_solutions=/net/omega/raid1/scratch1/terwill/blind_tests/all_case
s/mr_rosetta_from_start/1_ag9603/MR_ROSETTA_6/repeat_results.pkl
display_solutions=True
```

If the *phenix.mr_rosetta* run involved more than one cycle it will say " Results after repeat_mr_rosetta: " (as in the case above). In this case the list of solutions obtained with the above command will only include results from the repeat cycle.

To obtain results from earlier stages, look earlier in the log file to the place where the word "RESULTS OF AUTOBUILDING" (in capitals) first appears and then search down to the next `display_solutions` command:

```
=============================================================
RESULTS OF AUTOBUILDING:
=============================================================

ID: 222
R/Rfree:   0.25 /   0.28
MODEL:
/net/omega/raid1/scratch1/terwill/blind_tests/all_cases/mr_rosetta_from_sta
rt/1_ag9603/MR_ROSETTA_6/GROUP_OF_AUTOBUILD_1/RUN_2/AutoBuild_run_1_/cycle_
best_4.pdb
MAP COEFFS
/net/omega/raid1/scratch1/terwill/blind_tests/all_cases/mr_rosetta_from_sta
rt/1_ag9603/MR_ROSETTA_6/GROUP_OF_AUTOBUILD_1/RUN_2/AutoBuild_run_1_/cycle_
best_4.mtz

Writing solutions as csv to
```

```
/net/omega/raid1/scratch1/terwill/blind_tests/all_cases/mr_rosetta_from_sta
rt/1_ag9603/MR_ROSETTA_6/autobuild_results.csv

Saved overall mr_rosetta results in
/net/omega/raid1/scratch1/terwill/blind_tests/all_cases/mr_rosetta_from_sta
rt/1_ag9603/MR_ROSETTA_6/autobuild_results.pkl

To see details of these results type
    phenix.mr_rosetta
mr_rosetta_solutions=/net/omega/raid1/scratch1/terwill/blind_tests/all_case
s/mr_rosetta_from_start/1_ag9603/MR_ROSETTA_6/autobuild_results.pkl
display_solutions=True
```

where the appropriate command will be listed.

When you print out a list of solutions in this way, each solution is listed along with the lineage of that solution (all the solutions obtained on the path to this solution).

Restarting *phenix.mr_rosetta* if something goes wrong

As *phenix.mr_rosetta* completes each stage, it writes out a file that contains all the information needed to go on from that stage. In the examples above where the command `display_solutions=True` is used, this file is read and the information is simply printed. If you want to use this information to carry on, then you need to specify two things. First you need to name the file containing the solutions you want to use. This file is listed in your log file as in the examples above, and a file is written out after each major step. You specify it with:

```
mr_rosetta_solutions=working_solutions.pkl
```

Second you need to specify where to start. You can do this with the keyword "start_point":

```
start_point=rosetta_rebuild
```

This will start with *Rosetta* rebuilding with density (provided you have supplied solutions that include the previous step, `rescore_mr`).

Acknowledgments

We gratefully acknowledge the financial support of NIH/NIGMS under grant number P01GM063210. Our work was supported in part by the US Department of Energy under Contract No. DE-AC02-05CH11231.

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., and Zwart P.H.. (2010). Acta Cryst. D66, 213-221.

Chivian D, Kim DE, Malmstrom L, Bradley P, Robertson T, Murphy P, Strauss CEM, Bonneau R, Rohl CA, Baker D. (2003) Automated prediction of CASP-5 structures using the Robetta server. Proteins 53 Suppl 6:524-33

DiMaio F., Tyka,M.D., Baker, M.L., Chiu,W., Baker, D. (2009). Refinement of protein structures into low-resolution density maps using rosetta. Journal of Molecular Biology 392: 181-190.

DiMaio, F., Terwilliger, T.C., Read, R.J., Wlodawer, A., Oberdorfer, G., Wagner, U., Valkov, E., Alon, A., Fass, D., Axelrod, H.L., Das, D., Vorobiev, S.M., Iwai, H., Pokkuluri, P.R., Baker, D. (2011). Improving molecular replacement by density and energy guided protein structure optimization Nature 473, 540-543.

Qian, B., Raman, S., Das, R., Bradley, P., McCoy, A.J., Read, R.J., and Baker, D. (2007). High resolution structure prediction and the crystallographic phase problem. Nature 450, 259-264.

Read, R.J. (2001). Pushing the boundaries of molecular replacement with maximum likelihood. Acta Cryst. D 57, 1373-1382.

Söding J. (2005) Protein homology detection by HMM-HMM comparison. Bioinformatics 21, 951-960.

Terwilliger, T.C., Grosse-Kunstleve, R.W., Afonine, P.V., Moriarty, N.W., Zwart, P.H., Hung, L.W., Read, R.J., and Adams, P.D. (2008). Iterative model building, structure refinement and density modification with the PHENIX AutoBuild wizard. Acta Cryst. D 64, 61-69.

Improved target weight optimization in *phenix.refine*

Pavel V. Afonine[1], Nathaniel Echols[1], Ralf W. Grosse-Kunstleve[1], Nigel W. Moriarty[1] and Paul D. Adams[1,2].
[1]*Lawrence Berkeley National Laboratory, One Cyclotron Road, MS64R0121, Berkeley, CA 94720 USA*
[2]*Department of Bioengineering, University of California Berkeley, Berkeley, CA, 94720, USA.*
Correspondence email: PAfonine@lbl.gov

Abstract

Restrained refinement of individual atomic coordinates and atomic displacement parameters combines experimental observations with prior knowledge. The two contributions need to be properly weighted with respect to each other in order to obtain the best results. This article describes a new target weight determination procedure in *phenix.refine* and presents the results of systematic tests on structures with lower resolution data.

Introduction

In *phenix.refine* (Afonine et al., 2005, Adams et al., 2010) the refinement of individual atomic coordinates or individual atomic displacement parameters (ADP, also known as B-factors) involves the minimization of a refinement target function that includes prior chemical or empirical knowledge. In the case of individual coordinate refinement this target function T is defined as:

$$T = w_{xc} \cdot w_{xc\_scale} \cdot T_{data} + w_c \cdot T_{geo\_restraints} \quad (1)$$

T_{data} is the target function quantifying the fit of experimental observations (X-ray and/or neutron data) and model-based predictions, using, for example, a least-squares or maximum-likelihood function. $T_{geo\_restraints}$ quantifies the fit of current model geometry (such as bonds, angles, dihedrals and nonbonded interactions) to tabulated "ideal" geometry, for example as inferred from high-resolution diffraction experiments. The three weight factors w_{xc}, w_{xc_scale} and w_c are redundant; equation (1) could be reformulated with only one weight factor. However, the formulation with three weight factors is helpful in practice. This is also true for the analogous formulation used in ADP refinement:

$$T = w_{xu} \cdot w_{xu\_scale} \cdot T_{data} + w_u \cdot T_{ADP\_restraints} \quad (2)$$

The weight factors w_c and w_u are usually one, but can be set to zero for unrestrained refinement. The weights w_{xc} and w_{xu} are determined automatically as described by Brünger *et al.* (1989) and Adams *et al.* (1997), using the ratio of the gradient norms after removing outliers:

$$w_{xc} = \sqrt{\frac{\langle \nabla T_{geo\_restraints}{}^2 \rangle}{\langle \nabla T_{data}{}^2 \rangle}} \quad (3)$$

$$w_{xu} = \sqrt{\frac{\langle \nabla T_{ADP\_restraints}{}^2 \rangle}{\langle \nabla T_{data}{}^2 \rangle}} \quad (4)$$

w_{xc_scale} and w_{xu_scale} are empirical scale factors, usually with values between 0.5 and 1.0.

An automatic weight determination procedure based on equations (3) and (4) has been used in *phenix.refine* from the beginning of its development. The procedure is usually reliable at typical macromolecular resolutions (around 1.5-2.5 Å) but sometimes problematic at significantly lower (> 3 Å) or higher (< 1.5 Å) resolutions. Typical problems are unexpectedly high R_{free} values, large gaps between R_{free} and R_{work}, unreasonably large geometry deviations from ideality, high Molprobity clash-scores, or large differences between ADPs of bonded atoms.

Brünger (1992) described a procedure that systematically searches for the weight leading to the lowest R_{free}. Until recently, the implementation in *phenix.refine* used an array of 10-20 values for w_{xc_scale} or w_{xu_scale}, with values distributed between 0.05 and 10. A full trial refinement was performed for each weight. In our experience, using R_{free} as the only guide for determining the optimal weight can sometimes discard results that are clearly more preferable if other quality measures are also taken into account. For example, R_{free} may oscillate only slightly while R_{work}, bond and angle deviations, or clash-scores change significantly. In this article we describe an enhanced weight search procedure that makes active use of an ensemble of quality measures.

Methods

In contrast to the previously used procedure, the new procedure in *phenix.refine* examines trial

weights on an absolute scale:

$$T = w_{trial} \cdot T_{data} + T_{restraints} \qquad (5)$$

Since *phenix.refine* uses normalized targets for data and restraints, the range of plausible values is predictable. For example, the amplitude-based ML target (Lunin & Skovoroda, 1995; Afonine *et al.*, 2005) typically yields values that fall in the range between 1 and 10 (depending on resolution, data quality and model quality). The weight optimization procedure is parameterized with a spectrum of trial weights that is sufficiently large to offset such variations in the scale of T_{data}.

The new procedure executes the following steps:

1. For each trial weight, perform 25 iterations of LBFGS minimization (Liu & Nocedal, 1989) and save R_{work}, R_{free}, R_{free} - R_{work}. For coordinate refinement, also save bond and angle RMSDs and the clash-score. For ADP refinement, also save the mean difference between B-factors of bonded atoms $<\Delta B_{ij}>$.
2. Select the subset of plausible results corresponding to R_{free} values in the range $[R_{free}^{min}, R_{free}^{min} + \Delta]$, where Δ is a resolution-dependent value in the range from 0 (high resolution) to 2% (low resolution) and R_{free}^{min} is the smallest R_{free} value obtained in step 1.
3. Reduce the subset further by applying selection criteria based on the R_{free} - R_{work} difference and bond and angle RMSDs (coordinate refinement) or $<\Delta B_{ij}>$ (ADP refinement).
4. In the case of coordinate refinement, reduce the subset further based on the clashscores (c). The first step is to the select results that satisfy the condition $\bar{c}/3 < c < 3\bar{c}$. For the second step recompute the mean $\bar{c}_{new}$ for the new subset and select results in the range from the minimum of the clashscores, c_{new}^{min}, to $c_{new}^{min} + w_{cs} * \bar{c}_{new}$. Currently the default value for w_{cs} is 0.1.
5. For the remaining subset select the result that corresponds to the lowest R_{free}.

The choice of Δ values in step 2 is based on the evaluation of a large number of refinements. We selected a number of data/model pairs covering a range of resolutions. For each pair we ran multiple refinements with identical parameters, except for the random seed used in the target weight determination and the simulated annealing module. In another series of tests, we applied modest random shifts to coordinates and *B*-factors before refinement. An ensemble of similar solutions is obtained for each data/model pair. Identical solutions cannot be expected (for example, see Terwilliger *et al.*, 2007) because the refinement target function is very complex and populated with many local minima; therefore the starting point is important. In addition, the structural deviations can be a consequence of static or dynamic disorder that is difficult to model. The Δ values reflect typical R_{free} fluctuations we observed in our refinement results, ranging from small fractions of a percent for high-resolution refinements and approaching 2 percentage points in a few low-resolution cases.

The optimal value for $<\Delta B_{ij}>$ in step 3 is not clearly defined, as discussed in Afonine *et al.* (2010a). Our current working estimate is $0.1<B>$, where $<B>$ is the average *B*-factor.

The weight optimization procedure is easy to parallelize since the refinements with different trial weights are independent. Starting with PHENIX version dev-810, the `refinement.main.nproc` parameter is available to specify the number of CPUs the weight optimization procedure may use in parallel. To give one example, the command

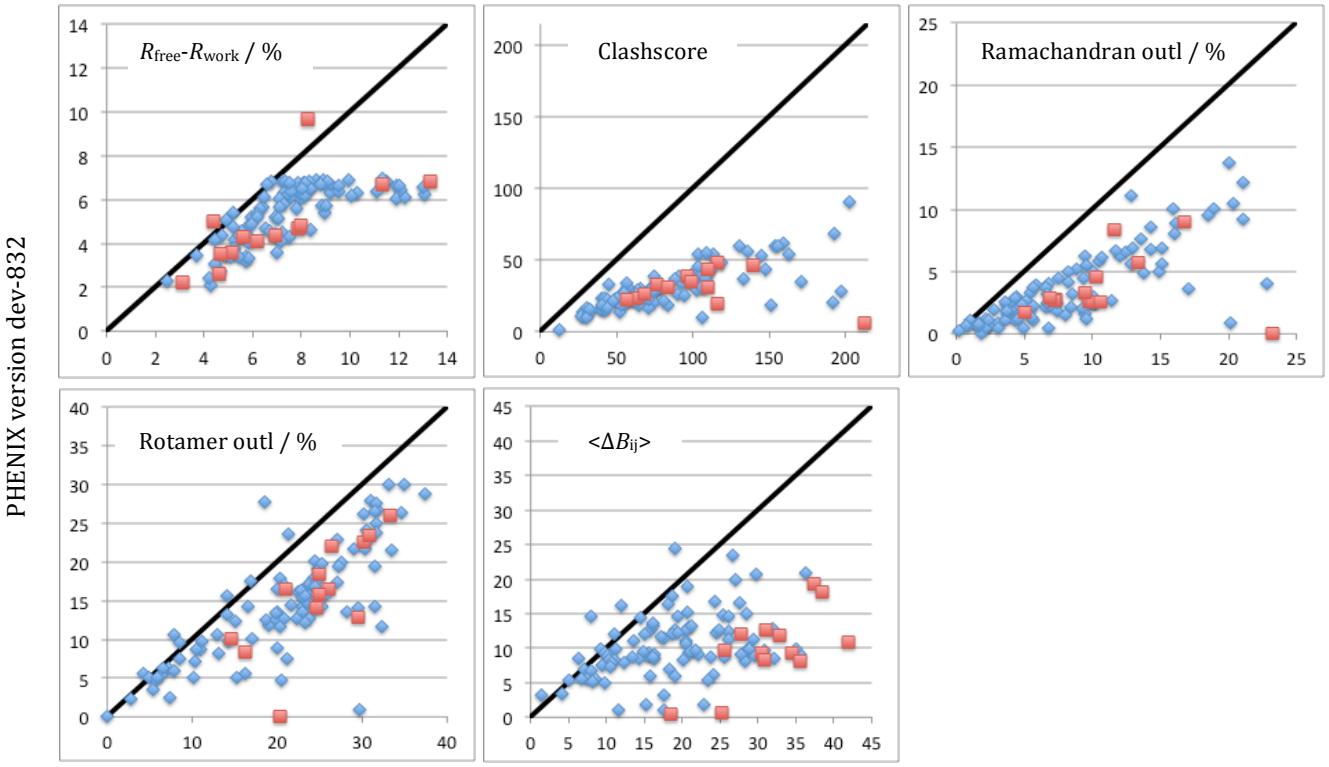
```
phenix.refine 1av1.pdb 1av1.mtz \
  optimize_xyz_weight=True \
  optimize_adp_weight=True nproc=16
```

finishes in approximately 430 seconds on a 48-core 2.2GHz AMD Opteron system. With `nproc=1` the refinement requires more than 2000 seconds on the same machine.

Results and discussion

To evaluate the new procedure we selected a set of low-resolution structures from the PDB (Bernstein *et al.*, 1977; Berman *et al.*, 2000) using on the following criteria:

- data high resolution limit between 3.5 and 4.5 Å,
- data completeness (overall and 6 Å – inf) better than 85%,
- data collected from untwinned crystals,



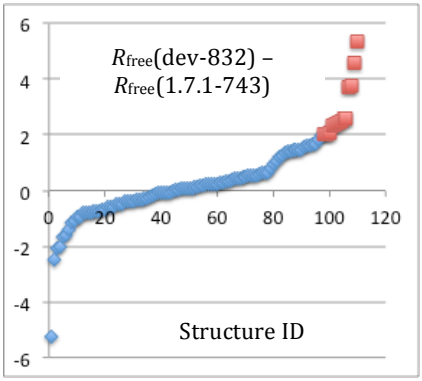
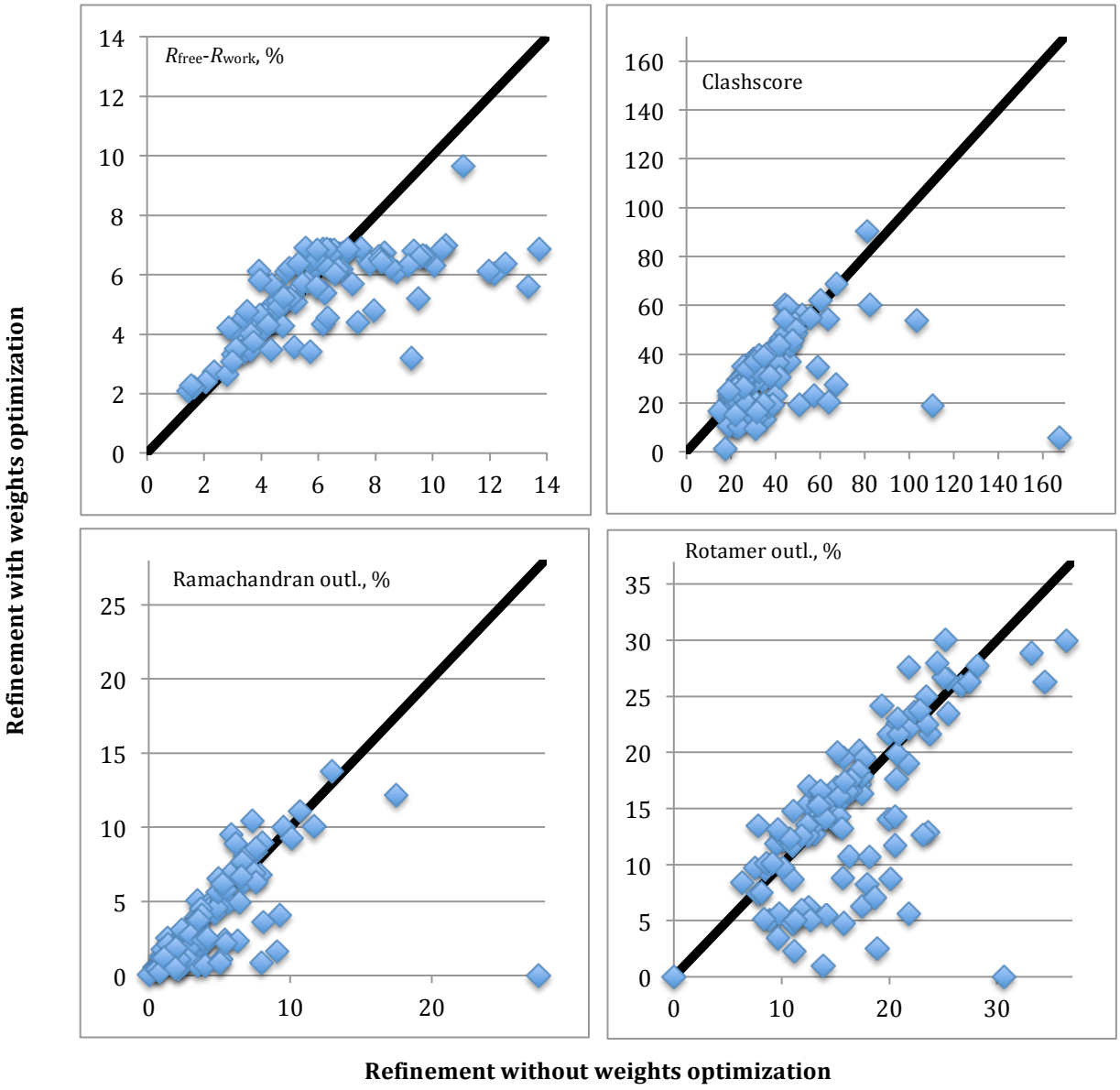


Figure 1. Comparison of refinement results between two PHENIX versions 1.7.1-743 and dev-832 (see text for details). Red squares highlight cases where the difference in R_{free} was larger than 2 percentage points (see the last plot where the results are ordered by R_{free} difference between the runs).

- data extractable from the PDB archive using *phenix.cif_as_mtz* (Afonine *et al.*, 2010b),
- models consisting only of common protein residues, ligands, heavy atoms and water,
- non-zero occupancy for all atoms,
- R_{free} flags available and a minimal gap between R_{free} and R_{work} of more than 2 percentage points.

We found 108 matching structures. Each structure was refined with 5 macro-cycles of restrained refinement of individual coordinates and ADPs (Afonine *et al.*, 2005). Most selected structures contain NCS-related molecules. NCS-related groups were determined automatically by *phenix.refine* and restrained in Cartesian space (atom-pair-wise harmonic restraints to the group average). Refinements were performed using the old weight optimization procedure as available in PHENIX (Adams *et al.*, 2002; Adams *et al.*, 2010) version 1.7.1-743 and the new procedure as included in a current development version (dev-832). The detailed results are presented in Figures 1 and 2.

Figure 1 compares refinement results (R_{free} - R_{work} difference, clash-score, <ΔB_{ij}>, percent of rotamer outliers and Ramachandran plot outliers) between *phenix.refine* runs using PHENIX versions 1.7.1-743 and dev-832. The results show clearly that the new procedure provides much improved geometry statistics and lower <ΔB_{ij}> values in most cases. The reduction of Ramachandran and rotamer outliers is especially noteworthy, since these quality measures are not directly used in the target weight optimization. The R_{free} comparison shows that the deviations are mostly within the Δ parameter range (2 percentage points), as



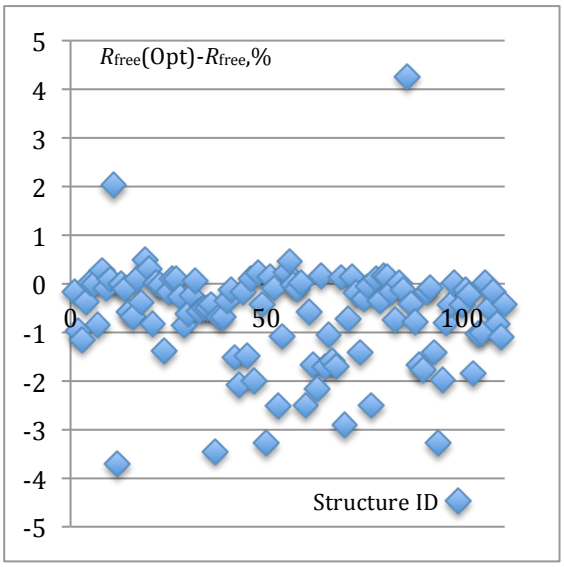


Figure 2. Results of refinement with and without weights optimization using PHENIX version dev-832.

expected. The few outliers with differences larger than 2 percentage points (red squares on Fig. 1) may be due to non-optimal NCS group selections that require further analysis, or $<\Delta B_{ij}>$ values that were forced to obey the requested limit. The large number of Ramachandran plot outliers may also indicate problems with the starting models that are beyond the anticipated convergence radius of these refinement procedures. Examining these cases in detail may lead to further improvements.

Figure 2 shows a comparison of statistics similar to Figure 1, after refinement with and without weight optimization using the current PHENIX development version (dev-832 or later).

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 Cryst. D* **66**, 213-221.

Adams, P. D., Grosse-Kunstleve, R. W., Hung, L. W., Ioerger, T. R., McCoy, A. J., Moriarty, N. W., Read, R. J., Sacchettini, J. C., Sauter, N. K. & Terwilliger, T. C. (2002). *Acta Cryst. D* **58**, 1948-1954.

Adams, P. D., Pannu, N. S., Read, R. J. & Brünger, A. T. (1997). *Proc. Natl. Acad. Sci*. **94**, 5018-5023.

Afonine, P. V., Grosse-Kunstleve, R. W. & Adams, P. D. (2005). *Acta Cryst.* D**61**, 850-855.

[a]Afonine, P.V., Urzhumtsev, A., Grosse-Kunstleve, R.W. & Adams, P.D. (2010). *Computational Crystallography Newsletter*. 1, 24-31.

[b]Afonine, P.V., Grosse-Kunstleve, R.W., Chen, V.B., Headd, J.J., Moriarty, N.W., Richardson, J.S., Richardson, D.C., Urzhumtsev, A., Zwart, P.H. & Adams, P.D. (2010). *J. Appl. Cryst.* **43**, 669-676.

Afonine, P. V., Grosse-Kunstleve, R.W. & Adams, P.D. (2005). *CCP4 Newsletter on Protein Crystallography* **42 (8)**.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N. & Bourne, P. E. (2000). *Nucleic Acids Res* **28**, 235-242.

Bernstein, F. C., Koetzle, T. F., Williams, G. J. B., Meyer, E. F., Brice, M. D., Rodgers, J. R., Kennard, O., Shimanouchi, T. & Tasumi, M. (1977). *J Mol Biol* **112**, 535-542.

Brünger, A. T. (1992). *Nature* **355**, 472-475.

Brünger, A.T., Karplus, M. & Petsko, G.A. (1989). *Acta Cryst.* A**45**, 50-61.

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

Lunin, V. Y. & Skovoroda, T. P. (1995). *Acta Cryst*. A**51**, 880-887.

Terwilliger, T. C., Grosse-Kunstleve, R. W., Afonine, P. V., Adams, P. D., Moriarty, N. W., Zwart, P. H., Read, R. J., Turk, D. & Hung, L.-W. (2007). *Acta Cryst*. D**63**, 597-610.

Mite-y Lysozyme Crystals and Structures

Janet Newman*, Del Lucent and Thomas S Peat
Molecular and Health Technologies, CSIRO, 343 Royal Parade, Parkville, VIC, 3052, Australia

*Correspondence email: janet.newman@csiro.au

Synopsis

Three different sandwich spreads were tested for their ability to crystallise lysozyme, the resultant crystals were of uniformly high quality and produced structures that fell within the envelope of the known structures of lysozyme.

Abstract

Marmite, Promite and Vegemite are three variations of yeast extract pastes, which are considered edible foodstuffs by many people around the world. These spreads all report high levels of sodium on their nutritional information labels and we were interested if this would correspond to an ability to support lysozyme crystallisation, which may be easily crystallised from sodium chloride solutions. Counter diffusion crystallisation experiments were set up with hen egg white lysozyme, using Marmite, Promite or Vegemite as the crystallant. The technique of counter diffusion was chosen as this allowed crystal growth to be observed, despite the black, opaque nature of the crystallants. Crystals grew from all three spreads and these crystals were tested for diffraction quality and structures were produced from crystals of each variety of yeast extract paste. The tested crystals grew in the familiar $P4_32_12$ tetragonal space group, with cell dimensions of approximately 79 x 79 x 38 $Å^3$.

Keywords: counter diffusion, vegemite, lysozyme, multiple structure alignment

1. Introduction

Yeasts have been used in the production of human foods for millennia (Cavalieri *et al.*, 2003, Legras *et al.*, 2007). One of the more recent incarnations of yeast food products are the yeast extract pastes that are popular spreads for toast and sandwiches in some countries. The process for making concentrated yeast extract requires the addition of sodium chloride to a yeast cell pellet in order to induce autolysis after which the resulting lysate is filtered, flavoured and concentrated (Irving, 1992, Cook, 1910), this is a modification of a process developed by Liebig, modelled in turn after his process for the extraction of the essence of meat (Brock, 1997). One of the earliest of these products to be available commercially was Marmite, which was sold by the Marmite Food Company (later Marmite Ltd) of Burton on Trent, UK in 1902 (*The Bumper Book of Marmite*, 2009). Similar products are available in Australia (Vegemite, Promite), Switzerland (Cenovis) and New Zealand (Marmite) - note that New Zealand Marmite is produced under license and has a different formulation than the British product of the same name (wikipedia.org/wiki/Marmite). Vegemite has been touted as being potentially one of the most culturally specific foods – if you eat Vegemite, then you are very likely to be Australian and *vice versa* (Rozin & Siegal, 2003). These products are renown for their salty tang and the nutritional information labels show that these products contain anywhere from 3400 mg to 4844 mg sodium per 100 g of product. Assuming that the counter-ion of the sodium is chloride and given the mass percentage of sodium in NaCl is 39.34%, this would suggest that the products contain from 8.7 g to 12.3 g of NaCl per 100 g.

Hen egg white lysozyme (HEWL) is a readily available protein that is notoriously overused as a crystallisation test protein (see for example (Lu *et al.*, 2010, Newman, 2005, Newman *et al.*, 2007, Vrikkis *et al.*, 2009)). This protein crystallises out of numerous conditions (Newman *et al.*, 2007) but is often crystallised from a sodium acetate / sodium chloride crystallant, where the acetate is buffered to around pH 4.6 and the NaCl concentration is around 5% (or equivalently, around 1 M) (Bergfors, 2009).

Counter diffusion is a crystallisation technique in which a concentrated protein solution is introduced into a capillary and the crystallant solution is allowed to migrate into the capillary. If the capillary is of sufficiently narrow bore, the

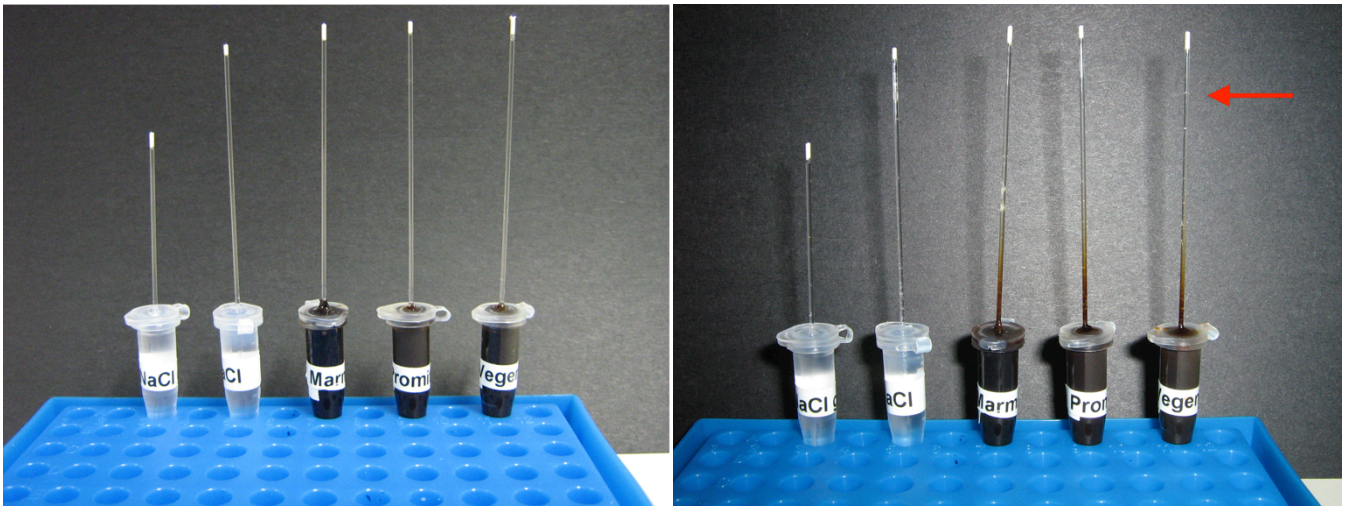


Figure 1. (a) The capillary counter diffusion experiment on setup at left, (b) shows the same experiment 16 days laterat right. The red arrow shows the position of a crystal in the Vegemite experiment. These experiments were set up at room temperature and incubated at 4C.

crystallant moves into and along the tube only by diffusion and sets up a concentration gradient along the length of the tube (Garcia, 2003, Ng *et al.*, 2003, Ng *et al.*, 2008). Because of this transient gradient, large, well formed crystals can grow even if the crystallant is of much higher concentration than would normally be used in a more standard vapour diffusion experiment

We set up counter diffusion experiments with commercial HEWL and yeast pastes for a number of reasons - primarily to build expertise with the counter diffusion technique, but also as we were rather curious whether crystallogenesis could be achieved with these salty foodstuffs. We were also interested in determining whether the structures of HEWL determined from crystals grown in the spreads would be significantly different from the large number of structures already available for this protein.

2. Materials and methods

Counter diffusion experiments were set up in two slightly different formats: in both cases all three pastes were set up along with two sodium chloride control experiments. In one variation, three vials were prepared by adding approximately one millilitre of Vegemite (Kraft Australia), Marmite (Sanitarium NZ) or Promite (Mars Food Australia) to the bottom of a push-cap vial. The pastes are hard to work with neatly - eventually a technique was developed where a spatula was used to scoop paste into a 5 ml syringe, this syringe was used to fill a 1 ml syringe, which was used to deposit the spread cleanly at the bottom of the vial. One millilitre of melted 1% agarose gel (Applichem) was layered over the pastes and allowed to solidify. Two controls were set up: the first control vial was set up by adding 1 ml of 5 M NaCl (Sigma S7653) and layering over 1 ml of the 1% agarose solution, the second by allowing 0.5 ml agarose to harden in the bottom of a vial, then adding 1 ml of the 5 M NaCl solution on top. Five 64 mm long, 0.63 mm internal diameter glass capillaries (Drummond MicroCap 1-000-0200) were filled with a solution of 40 mg/ml lysozyme (Sigma L6876) in 50 mM sodium acetate pH 4.5 and one end sealed with Haematocrit sealing compound (Brand). The open end of the capillary was inserted through the agarose gel into the paste (or salt solution). The setups were stored at room temperature. The capillaries were examined for crystal formation under a light microscope without removing them from the vials (Figure 1a)

In a second variation on the counter diffusion technique, capillaries were filled with the protein solution and one end sealed as above. The crystallant was contained in 0.65 ml Eppendorf tubes, which had lids pierced with an 18-gauge needle. The vials were filled with crystallant: for the spreads, the tubes were filled with the paste. The NaCl controls were set up with either an agarose layer in the bottom of the Eppendorf tube with 5 M NaCl solution filling the remaining volume, or with 0.5 ml of the 5 M NaCl solution

Table 1. Sample information

Macromolecule details			
Database code(*s*)	PDB code: 3N9A (Vegemite), 3N9C (Marmite) and 3N9E (Promite)		
Component molecules	Hen Egg White Lysozyme (EC number: 3.2.1.17)		
Mass (Da)	14,700		
Source organism	Gallus gallus (details: Purchased from Sigma L6876)		
Crystallization and crystal data			
	Crystal 1 – Vegemite	Crystal 2 – Marmite	Crystal 3 – Promite
Crystallization method	Free interface diffusion/counterdiffusion	Free interface diffusion/counterdiffusion	Free interface diffusion/counterdiffusion
Temperature (K)	293	293	293
Apparatus	Drummond microcaps	Drummond microcaps	Drummond microcaps
Atmosphere	1	1	1
Seeding	None	None	None
Crystallization solutions			
Macromolecule	20 ml, lysozyme (40 mg ml^{-1}), sodium acetate (pH 4.5, 50 m*M*)	20 ml, Lysozyme (40 mg ml^{-1}), sodium acetate (pH 4.5, 50 m*M*)	20 ml, lysozyme (40 mg ml^{-1}), sodium acetate (pH 4.5, 50 m*M*)
Unit-cell data			
Crystal system, space group	Tetragonal, $P4_32_12$	Tetragonal, $P4_32_12$	Tetragonal, $P4_32_12$
***a*, *b*, *c* (Å)**	79.29, 79.29, 37.89	79.41, 79.41, 38.02	79.29, 79.29, 38.00
a, b, g (°)	90, 90, 90	90, 90, 90	90, 90, 90

placed into the empty tube and the agarose gel layered over. The unsealed end of a prepared capillary was inserted through the lid into the filled (and closed) Eppendorf tube and the gap between the capillary and lid was sealed with a dab of clear nail polish (Figure 1b). These setups were stored at 4 C. In all 10 experiments, care was taken to ensure that there was protein solution all the way to the ends of the capillaries and that the unsealed end of the capillaries did not touch the vials/tubes, to ensure that the crystallant could diffuse freely into the protein solution.

Data were collected at the MX1 beamline of the Australian Synchrotron from crystals grown from the three yeast extracts from the room temperature experiments. Data were collected on the crystals *in-situ*, at room temperature (see table 1 for details). The crystals were prepared for data collection by wicking away most of the mother liquor and re-sealing the capillaries with wax. The capillaries were mounted on a magnetic cap with modelling clay and we translated the crystals several times during data collection to introduce a fresh part of the crystal to the beam. 180 frames of data were collected, with each frame being a 1 degree oscillation exposed for 1 second. The crystal to detector distance was 140 mm, leading

Table 2. Data collection and structure solution statistics. Values for the outer shell are given in parentheses.

	Diffraction set 1 (crystal 1 -Vegemite)	Diffraction set 2 (crystal 2 - Marmite)	Diffraction set 3 (crystal 3 - Promite)
Diffraction source	AS, MX1	AS, MX1	AS, MX1
X-ray beam size	0.1 mm x 0.1 mm	0.1 mm x 0.1 mm	0.1 mm x 0.1 mm
Sampling protocol	1º oscillation, 1 sec/º	1º oscillation, 1 sec/º	1º oscillation, 1 sec/º
Wavelength (Å)	0.98	0.98	0.98
Detector	ADSC Quantum 215	ADSC Quantum 215	ADSC Quantum 215
Temperature (K)	293	293	293
Resolution range (Å)	39.7–1.40 (1.48–1.40)	40.0 – 1.50 (1.58-1.50)	39.7 – 1.38 (1.45-1.38)
No. of unique reflections	23901 (2999)	19946 (2867)	25326 (3480)
No. of observed reflections	443919	208981	495066
Completeness (%)	98.0 (87.0)	100 (100)	99.4 (96.0)
Redundancy	18.6 (13.3)	10.5 (10.6)	19.5 (11.6)
$< I/s(I)>$	35.7 (6.1)	19.9 (4.6)	35.1 (4.8)
R_{merge}	5.5 (45.4)	6.7 (49.5)	5.4 (47.4)
$R_{p.i.m.}$	1.3	2.2	1.2
Data-processing software	*SCALA*	SCALA	SCALA
Phasing method	MR	MR	MR
Starting model data set	2BLX	2BLX	2BLX
Alterations to search model	none	none	none
Solution software	PHASER	PHASER	PHASER

to a maximum resolution of around 1.4 Å (see table 2 for details).

Data were indexed with MOSFLM (Leslie, 1992), merged and scaled with SCALA, molecular replacement (using the structure 2BLX from the Protein Data Bank (PDB, http://www.pdb.org) (Deshpande *et al.*, 2005)) was performed using PHASER and the models refined with REFMAC 5.6 (Collaborative Computational Project, 1994) (see table 3 for refinement details).

The refined structures of HEWL generated in this work were compared to other HEWL structures available in the PDB. A sequence search was run using the "Sequence (Blast/Fasta)" option in the PDB website where the sequence of HEWL (pdb 2BLX) was used, along with a BLAST (Altschul *et al.*, 1990) cutoff E-value of 10^{-50}. This returned 332 structures, which had identical or 1 amino acid difference to the search sequence. A python script was created which used PyMol version 1.2r3 (Delano, 2003) to align the 332 sequences to the sequence from the PDB entry 2BLX and the aligned sequences were used to generate an ensemble of aligned structures. This was represented by drawing a "sausage" with a radius corresponding to the difference seen at each main chain atom position around the structure 2BLX. Two envelopes of the ensemble were calculated: one using the root mean square deviation (rmsd) of the main chain atoms from the reference structure and a second where the maximal distance from the main chain atom was used to set the radius of the "sausage" at that atom. We superposed the structures from the

Table 3. Structure refinement and model validation. Values for the outer shell are given in parentheses.

	Diffraction set 1 (crystal 1 -Vegemite)	Diffraction set 2 (crystal 2 - Marmite)	Diffraction set 3 (crystal 3 - Promite)
Refinement software	*REFMAC* 5.6	*REFMAC* 5.6	*REFMAC* 5.6 Both the salt
Refinement on	*F*	*F*	*F*
Resolution range (Å)	56.1–1.4	56.2 – 1.50	56.1 - 1.38
No. of reflections used in refinement	22633	18886	23991
Final overall *R* factor	16.3 (23.1)	16.0 (26.6)	16.2 (26.1)
Atomic displacement model	isotropic	isotropic	isotropic
Overall average *B* factor (Å²)	19.8	19.9	20.7
No. of protein atoms	1001	1001	1001
No. of nucleic acid atoms	0	0	0
No. of ligand atoms	0	0	0
No. of solvent atoms	78	77	82
Total No. of atoms	1204	1215	1231
No. of refined parameters			
Non-crystallographic symmetry restraints	None	None	None
Final R_{work}	16.3 (23.1)	16.0 (26.6)	16.2 (26.1)
No. of reflections for R_{free}	1224 (78)	1016 (62)	1288 (71)
Final R_{free}	18.9 (27.5)	19.0 (25.4)	18.3 (30.8)
Ramachandran plot analysis			
Most favoured regions (%)	97.8	95.7	96.6
Additionally allowed regions (%)	2.2	4.3	3.4

Vegemite, Marmite and Promite crystallisations on this ensemble, to obtain a visual gauge of how similar these structures were to the myriad of other structures available for HEWL.

3. Results and discussion

A number of crystals appeared in the capillaries in all experiments within 48 hours of setup. Fewer and smaller crystals were observed in all of the room temperature experiments than in the corresponding 4 C experiments. Sixteen days after setup there were crystals along the entire length of all the capillaries at 4 C, with considerable diffusion of the paste (as seen by a brown colouration) along the sample capillaries (Figures controls and the Promite paste showed poor crystal morphology close to the open end of the capillary and better crystal morphology towards the closed end of the capillary. The cold Vegemite and Marmite experiments had fewer, larger and better formed crystals than the equivalent Promite experiment. The experiments set up at 20 C gave fewer crystals in the paste experiments than the corresponding experiments at 4 C, with no crystals observed at the “distant” – or sealed end of the capillaries. The warm salt controls showed crystal growth along the entire length of the capillaries - mostly the high-salt "sea urchin" crystal habit. The agarose layer over the pastes in the vials turned quite black and there was

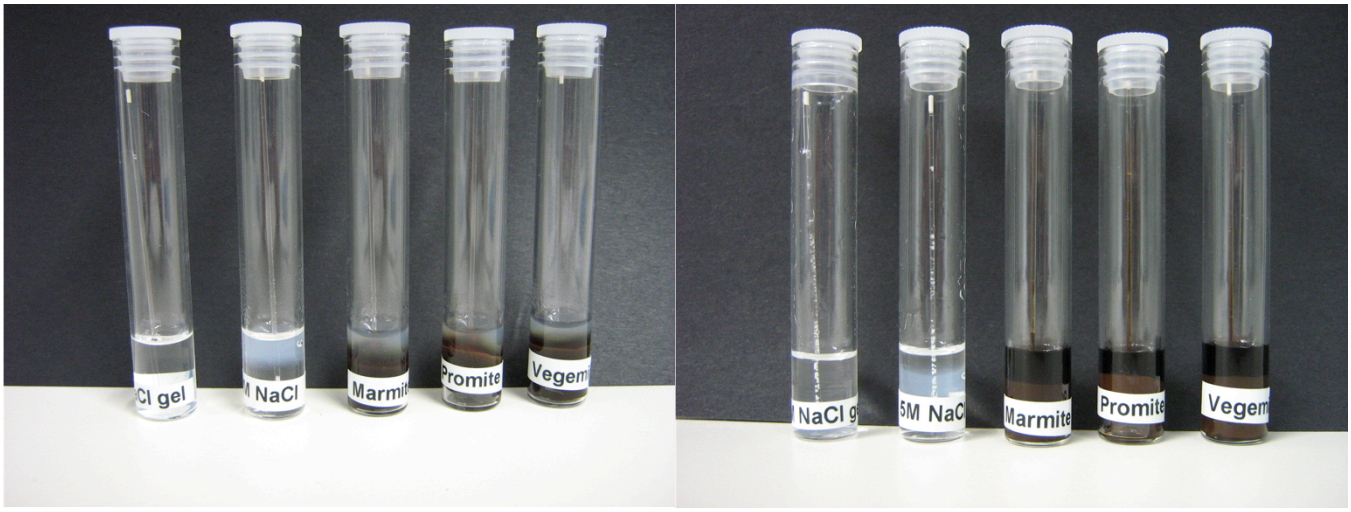


Figure 2. (a) Capillary counter diffusion experiments set up in vials immediately after setup. (b) shows the same experiments 16 days later. These experiments were incubated at 20C.

colouration from the pastes over half the length of the capillaries in the paste setups at room temperature (Figure 2b).

As stated above, both the 5 M salt controls and the Promite setups showed poor crystal morphology in the "near" end of the capillaries, compared to the experiments set up with Vegemite and Marmite. Promite has significantly more sodium (4844 mg 100 g^{-1}) than either Vegemite or Marmite (3489 mg 100 g^{-1} or 3400 mg 100 g^{-1} respectively), potentially leading to this effect. Counter diffusion is an intriguing technique, as a vast range of concentrations can be tested in the one experiment. However, our 20 C 5 M NaCl controls showed poor crystal habit, suggesting that this aspect of the technique can be overwhelmed if the crystallant concentration is too high, or if the capillary is too wide in internal diameter, or not long enough. We wanted to use glass X-ray capillaries for this experiment, but found that they were too delicate for the process – they crushed when the wide end was removed and broke when pushed into the Haematocrit sealant. The experiments where the capillary was enclosed in a glass vial were harder to visualise than the experiments were only the end of the vial was pushed into an Eppendorf tube. A refinement that we now use is to use 0.2 ml PCR tubes to contain the crystallant, as we find that these smaller tubes, when filled completely with liquid, give a very good view of the entire capillary. We saw a large amount of background scatter during data collection, presumably from the Drummond MicroCap capillary and would recommend moving to a thinner walled tube for *in-situ* diffraction studies, although the robustness of the thick MicroCaps has a lot to recommend it. Two crystals produced from the Marmite spread were somewhat less durable in the beam than the crystals produced by the other spreads, but this could be general variation of crystal quality rather than being specific to the Marmite experiment.

The three resulting structures are of very high quality overall, with strong, unambiguous density for residues 1-129. A loop region (residues 70-72), as well as the C-terminus (residues 126-129) were less well defined than the rest of the structure, suggesting that these regions are somewhat flexible. The three structures, one from each spread, where refined in a very similar manner and, unsurprisingly, the results were very similar. There are some differences between the three for side-chains where the density suggested multiple conformers were present (such as Arg73 or Asn19).

There is very little unexplained density in the maps – small blobs, but nothing large enough to be modelled by a sugar or some other cellular component. We do see some radiation damage, (as indicated by negative density around the sulfur atoms) near the disulfide bridge Cys6-Cys127 and a smaller amount associated with the disulfide bridge Cys76-Cys94.

The variation seen in the large number of lysozyme structures (332) obtained from the PDB

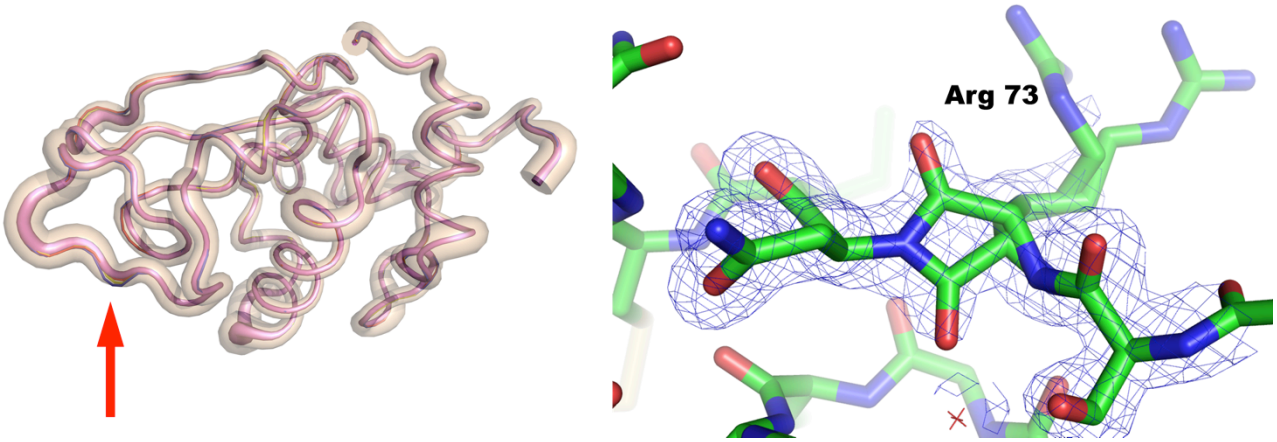


Figure 3. Backbone variation among known lysozyme structures. (a) Sausage diagram showing the maximum backbone distance (transparent tan envelope) and backbone RMSD (opaque violet envelope) for all structures in the PDB with sequence similarity E-values less than or equal to10^{-50} of 2BLX. Shown in ribbons are the backbones of the lysozyme structures crystallized in Vegemite (red), Marmite (blue), and Promite (yellow). The red arrow indicates the position of Arg73, where the three current structures deviate most from the consensus backbone trace. (b) Electron density of the main chain around Arg73 in the Vegemite structure. The density is well modelled by having two conformers for this residue, which include two very different positions for the main chain oxygen.

is remarkably small – the overall rmsd in the backbone positions was 0.55 Å. The envelope of deviation from the chosen structure (Figure 3a) shows some variation along the chain, with an unsurprisingly greater spread being seen in the loop regions, which mirrors what we observed in the three structures presented here. We find notable the lack of difference between over three hundred structures, solved and refined using different technologies by a host of different people. The three structures solved in this present work fall mostly within the smaller, rmsd envelope of all the structures, with some points of difference. The differences tend to occur where we have modelled in alternative conformers of residues; we used COOT (Emsley & Cowtan, 2004) for model building and used the option whereby a complete residue was included as an alternative conformer: older technologies tended to only include side-chain atoms in alternate conformer definitions. Figure 3b shows the region of the electron density for Arg73 in the Vegemite structure that we modelled as two conformers, with quite different positions of the main chain carbonyl oxygen. The other two structures showed similar electron density in this region and were modelled in a similar way. We believe that the deviations of the structures from the envelope of PDB structures can all be explained in this way.

The structures have been deposited in the PDB with accession codes 3N9A (Vegemite), 3N9C (Marmite) and 3N9E (Promite).

4. Conclusions

Commercial hen egg white lysozyme may be readily crystallised by the counter diffusion method from common sandwich spreads – the non sodium chloride components of the spreads appear not to have major negative effect on the resulting crystals, as there was no obvious excess density seen in the refined structures and the structures themselves aligned well with a large number of previously solved HEWL structures. We have developed a script that uses PyMol to align and visualise a large number of protein structures, this is available by emailing del.lucent@csiro.au.

Acknowledgements

We thank the beamline scientists of the Australian Synchrotron for their help in data collection. Dr. Del Lucent gets special thanks for agreeing to take and maybe even eat the excess spreads.

References

Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. (1990). *J Mol Biol* **215**, 403-410.

Bergfors, T. M. (2009). Editor. *Protein*

Crystallization Second Edition La Jolla: International University Line.
Brock, W. H. (1997). *Justus von Liebig: The Chemical Gatekeeper*. Cambridge: Cambridge University Press.
The Bumper Book of Marmite, 2009). Bath: Absolute Press.
Cavalieri, D., McGovern, P., Hartl, D., Mortimer, R. & Polsinelli, M. (2003). *Journal of Molecular Evolution* **57**, S226-S232.
Collaborative Computational Project, N. (1994). *Acta Crystallographica Section D* **50**, 760-763.
Cook, F. C. (1910). *A comparison of Beef and Yeast Extracts of Known Origin*. US Department of Agriculture.
Delano, W. (2003). *The PyMol Molecular Graphics System.* Version 0.99.
Deshpande, N., Addess, K. J., Bluhm, W. F., Merino-Ott, J. C., Townsend-Merino, W., Zhang, Q., Knezevich, C., Xie, L., Chen, L., Feng, Z., Green, R. K., Flippen-Anderson, J. L., Westbrook, J., Berman, H. M. & Bourne, P. E. (2005). *Nucleic Acids Res* **33**, D233-237.
Emsley, P. & Cowtan, K. (2004). *Acta Crystallographica Section D* **60**, 2126-2132.
Garcia, J. M. (2003). *Methods in Enzymology*, Vol. 368. *Macromolecular Crystallography, Part C,* edited by C. Carter, pp. 130-154: Elsevier.
Irving, J. (1992). *Vegemite Cook Book*. Melbourne: Ark Publishing Pty Ltd.
Legras, J.-L., Merdinoglu, D., Cornuet, J.-M. & Karst, F. (2007). *Molecular Ecology* **16**, 2091-2102.
Leslie, A. G. W. (1992). *Joint CCP4 + ESF-EAMCB Newsletter on Protein Crystallography* **26**.
Lu, Q.-Q., Yin, D.-C., Liu, Y.-M., Wang, X.-K., Yang, P.-F., Liu, Z.-T. & Shang, P. (2010). *Journal of Applied Crystallography* **43**, 473-482.
Newman, J. (2005). *Acta Crystallographica* **D61**, 490-493.
Newman, J., Xu, J. & Willis, M. C. (2007). *Acta Crystallographica* **D63**, 826-832.
Ng, J. D., Gavira, J. A. & Garcia-Ruiz, J. M. (2003). *Journal of Structural Biology* **142**, 218-231.
Ng, J. D., Stevens, R. C. & Kuhn, P. (2008). Vol. 426. *Structural Proteomics: High-Throughput Methods,* edited by B. Kobe, M. Guss & T. Huber, pp. 363-376. New York: Springer Science+Business Media LLC.
Rozin, P. & Siegal, M. (2003). *Gastronomica* **3**, 63-67.
Vrikkis, R. M., Fraser, K. J., Fujita, K., MacFarlane, D. R. & Elliott, G. D. (2009). *Journal of Biomechanical Engineering* **131**, 074514.