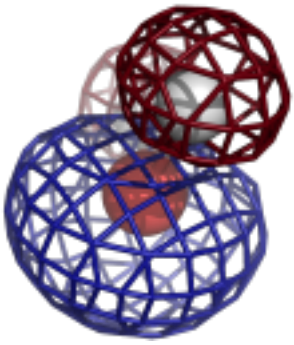
VOLUME SIX



COMPUTATIONAL CRYSTALLOGRAPHY NEWSLETTER

JANUARY MMXV

CIS PEPTIDES, SAD, CARBOHYDRATES, MATTPROB, NBO

Table of Contents

Editor

Nigel W. Moriarty, NWMoriarty@LBL.Gov

PHENIX News

New programs

Omegalyze *cis*-peptide validation

To help users avoid modeling unwarranted non-*trans* peptides in regions of poor data, automated identification of *cis*-peptides and peptides non-planar by >30° is now available. `phenix.omegalyze` provides text feedback and a listing of omega dihedral values. Running the command:

```
phenix.omegalyze nontrans_only=False
```

includes *trans* residues in the output. Running the command:

```
phenix.omegalyze kinemage=True
```

provides multicriterion kinemage markup for *cis*-peptides. For more details see this issue's "Avoiding excess *cis* peptides at low resolution or high B" fitting tip.

Real resolution of a dataset

Traditionally, the resolution of a diffraction dataset, d_{min} or d_{high}, is defined as the resolution of the highest-resolution reflection that belongs to this set. This value is a measure of the details that can be distinguished in the corresponding Fourier maps. Defined this way, it is meaningful if and only if the dataset is near 100% complete in the Ewald sphere $d \geq d_{high}$. If the dataset is incomplete, i.e. if there are unmeasured (missing) reflections in this sphere, then the actual resolution of the dataset may be different from the resolution of the highest-resolution reflection. Moreover, the resolution may vary in space and may be

different along different directions, with the minimum and maximum values along the directions not necessarily coinciding with the coordinate axes. There were previous efforts to give a better estimation of an 'effective' resolution taking into account the overall completeness of the data set (Weiss, 2001). Recently Urzhumtseva *et al*. (2013) proposed a mathematically strict definition of the data resolution and suggested a practical algorithm to calculate it. *Phenix* versions starting from dev-1935 have this algorithm implemented and available as the `phenix.resolution` command. The command takes a reflection data file in any of commonly used formats. It outputs three numbers: data resolution d_{high} calculated in the traditional way, and lowest and highest 'effective' resolutions of the data set calculated along all possible directions. For example, application of the `phenix.resolution` command to the PDB data set 4b44 results in values d_{high} = 2.30Å, d_{eff_min} = 2.23Å, d_{eff_max} = 3.07Å, showing a very large anisotropy of the measured set of reflections.

References

Urzhumtseva, L., Klaholz, B.P., Urzhumtsev, A. (2013) "On effective and optical resolution of diffraction data sets". *Acta Cryst.,* **D69**, 1921-1934.

Weiss, M.S. (2001). "Global indicators of X-ray data quality". *J.Appl.Cryst*., **34**, 130-135.

New features

New rotamer distributions

Use of sidechain rotamers derived from the new Top8000 dataset will now be the default system used both in refinement and validation. They are tuned so that the average number of rotamer outliers should be about the same; some individual outliers will differ, but are more accurately identified.

Carbohydrate linking

After a extensive testing of the Protein DataBank entries, automatic linking of carbohydrates is now the default for all Phenix programs using the PDB interpretation module. This includes N-linked sugars and glycosidic bonds.

Crystallographic meetings and workshops

Phenix Spring Workshop, March 2-5, 2015

Location: Berkeley, California. Presentations will be webcast and a User's Meeting for local students and postdocs will be held on Thursday.

West Coast Protein Crystallography Workshop XXII, March 15-18, 2015

Location: Monterey, California. A number of Phenix developers will be in attendance.

Expert advice

Fitting Tip #9: Avoiding excess *cis* peptides at low resolution or high B

Christopher Williams & Jane Richardson, *Duke University*

Even truly excellent pieces of software can do you in if their assumptions do not match your situation, so you should remain on the lookout for outstanding oddities. As a perhaps unexpected case, overall distribution patterns can go badly wrong when each fitting choice is made one at a time, independent of the rest.

The case in point here is *cis* peptides, both the classic x-Pro cases and especially the real but extremely rare *cis*-nonPro peptides (e.g. figure 1a). Tristan Croll, author of a paper now in press (Croll 2015) documents that the occurrence of *cis*-nonPro peptides has increased dramatically at ≥2.5Å resolution in recent years (such as the example in figure 1b). This includes otherwise well-done structures with >100 *cis*-nonPro, almost certainly unrealized by their depositors, in spite of both the warning message in Coot when fitting changes a peptide to *cis* and also the wwPDB's list of all *cis* peptides in the file header at deposition. The wwPDB is not strident about *cis* peptides and the warning in Coot is temporary, perhaps leading users to believe it has been reverted back to *trans*.

Tristan Croll contacted us, assigning some of the blame to MolProbity for not flagging *cis*-

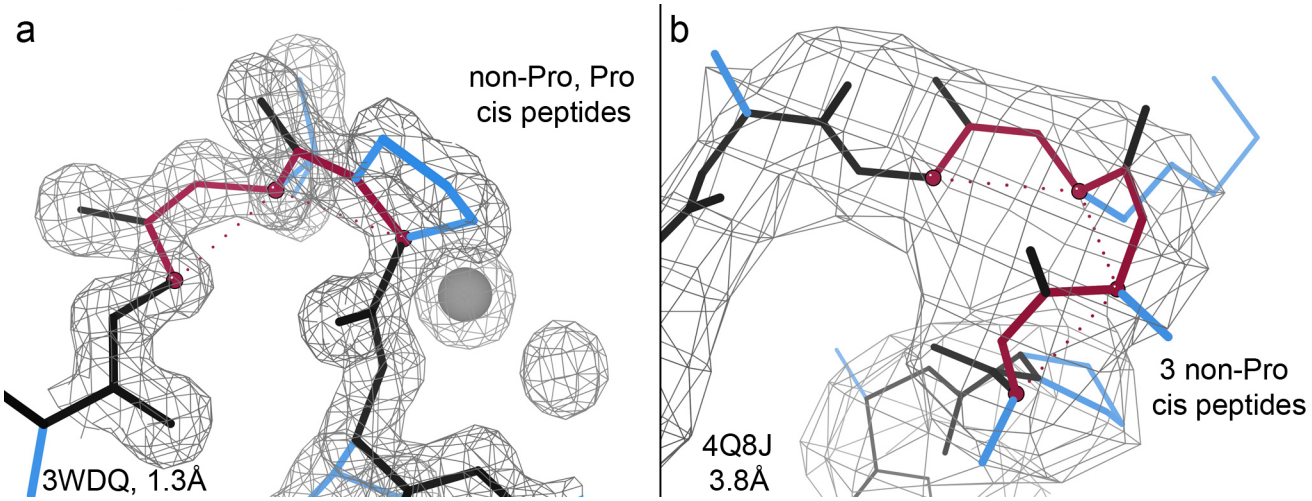


Figure 1: a) Genuine cis peptides in clear density; b) 3 unjustified cis-nonPro at low resolution.

nonPro, since people now trust that service too much for identifying all their model problems. In response, we will now report on peptide geometry. Determining the conformational category is of course simple, but devising an interface suitable at all resolution ranges was quite tricky. When we revisited the crystallographic evidence, we found that there are also an increasing number of clearly unjustified *cis* peptides at very high resolution, essentially all on loops with very poor electron density (e.g., figure 2).

Peptides preceding Pro are *cis* a bit less than 5% of the time in folded proteins, where the *cis*-*trans* barrier is substantial and packing of the Pro ring can stabilize or require a *cis* conformation. In contrast, genuine *cis*-nonPro peptides occur at a frequency of only about 0.03-0.05%; their reproducible stabilization is much harder to achieve and they are typically found at functional sites such as the classic *cis*-Gly in dihydrofolate reductase (Kraut 1982). This means some recent low-resolution structures have too many *cis*-nonPro peptides by as much as two orders of magnitude.

Even at 2Å resolution it is very difficult to be confident in the fit of a *cis*-nonPro and there is no way to justify such an assignment at >2.5Å — unless perhaps its occurrence is clear in a homologous protein at high resolution, given that conservation is strong for functionally important cases (Lorenzen 2005). We feel that eventually model-building routines should never fit *cis* or twisted nonPro peptides at low resolution or, in general, into less than very high-quality electron density. Avoidance would be very much easier to achieve than later correction. Even the much commoner *cis*-Pro are increasingly over-represented and should be subject to some limitation. As already noted (Croll 2015), unpenalized fit of *cis* peptides provides extra degrees of freedom that can allow an incorrect fit to be apparently outlier-free, potentially hiding serious problems such as

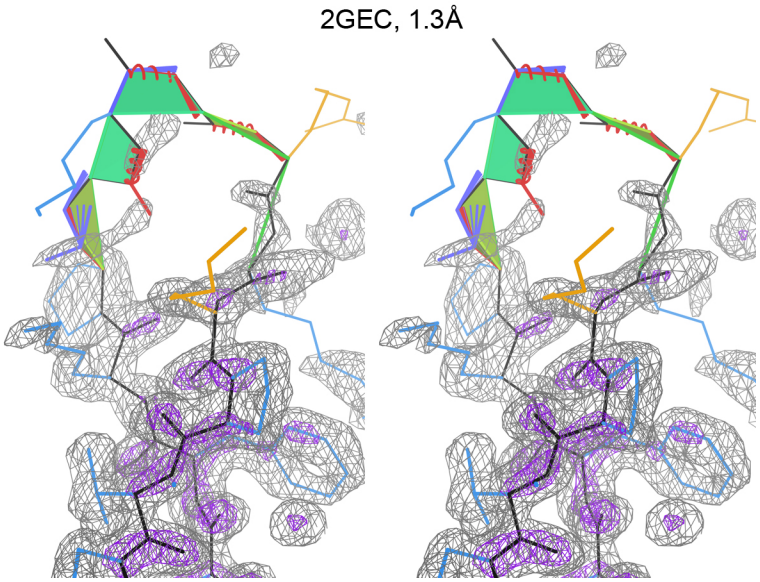


Figure 2: Cis & twisted peptides in poor density at 1.3Å.

shifted sequence register. In addition, the extra compactness of *cis* residues means they will always be systematically over-used at low resolution, since they can keep more atoms inside the contracted electron density.

The punchline for practicing crystallographers is, specifically, to check on *cis*-peptide frequencies reported in MolProbity, Coot, PDB, or elsewhere, and not to fit a *cis*-nonPro into unclear electron density. More generally, any very rare (so probably high energy) conformation should be considered suspect in high B-factor surface locations. And, finally, there will always be some new exception not yet dealt with by automated software, so you still need to actually look at your structure!

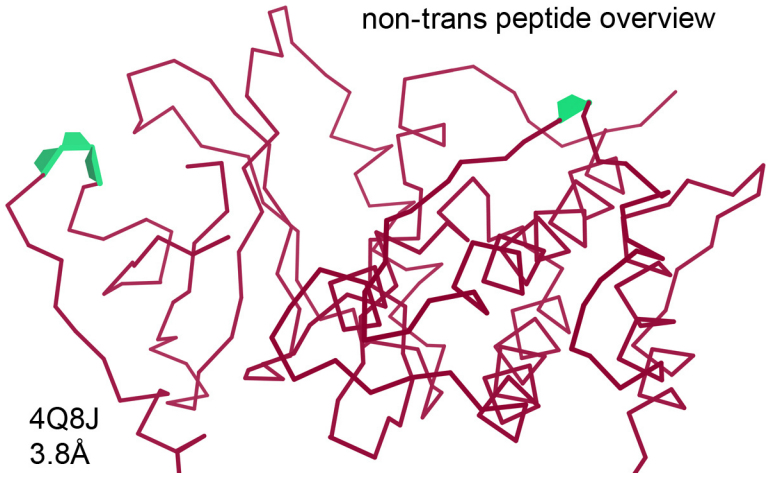


Figure 3: LoRx cis-peptide markup in an overview.

Technical notes for Phenix: Omegalyze methods

Our goals for a *cis*-peptide validation report were to provide an overall count, a list, a visualization of *cis*-peptides and to draw attention to their location in a protein structure (figure 3), without pre-judging the correctness of each individual case. Genuine *cis*-nonPro peptides are just as rare as genuine Ramachandran outliers and each case merits examination, before either acceptance as valid and interesting, or correction, or reluctant acceptance as likely wrong but uncorrectable.

An additional category of peptide conformation was needed to capture peptides with omega dihedrals far from either of the plausible planar conformations. Following usage in the PDB header, we designated peptides more than 30° away from either planar *trans* or planar *cis* as "Twisted". Although a few very convincing cases >30° are seen at high resolution (Berkholz 2012), twisted peptides are presumed to be modeling errors, requiring very strong experimental and biological evidence for justification.

Each omega peptide dihedral spans two residues, so a decision must be made for which residue number to report it. In the omegalyze assessment, as in ramalyze validation, each omega is associated with the residue immediately following, in order to preserve the unique importance of *cis*-prolines. The `phenix.omegalyze` functionality in `cctbx` was built with the same inheritance and many of the same methods as `phenix.ramalyze` to simplify its interface and maintenance.

The *cis*-peptide validation is so far available in Phenix only through the command line:

```
phenix.omegalyze file.pdb
```

Integration with the Phenix GUI is planned for implementation soon.

The default output is text, printed to `stdout` in the form as shown in schema 1. The output shown is from chain B of 2CN3.pdb, a xyloglucanase at 1.95Å, which happens to include all 3 main categories of peptide conformation, some clearly correct in the density (e.g. *cis*-Pro 353, 406) and some clearly incorrect (e.g. *cis*-Gly299, twisted His 275, Glu378); an example of each is illustrated in figure 4 a & b. Omegalyze output is in four colon-delimited columns, with additional summary lines at the end. The columns are:

1) residue identifier
2) residue type, either Pro or general

```
residue:type:omega:conformation
 B 162  GLU:General:-1.10:Cis
 B 270  GLY:General:-19.62:Cis
 B 275  HIS:General:132.01:Twisted
 B 294  PRO:Pro:5.66:Cis
 B 299  GLY:General:18.68:Cis
 B 349  ASN:General:-5.01:Cis
 B 353  PRO:Pro:-1.70:Cis
 B 377  PRO:Pro:7.46:Cis
 B 378  GLU:General:-78.58:Twisted
 B 402  PRO:Pro:9.87:Cis
 B 406  PRO:Pro:-2.29:Cis
SUMMARY: 5 cis prolines out of 45 PRO
SUMMARY: 0 twisted prolines out of 45 PRO
SUMMARY: 4 other cis residues out of 682 nonPRO
SUMMARY: 2 other twisted residues out of 682 nonPRO
```

Schema 1: Output example from omegalyse

3) omega value
4) category of peptide conformation: either *Cis*, *Trans*, or Twisted.

By default, *trans* conformation residues are not displayed. The summary lines that follow the residue-by-residue output provide whole-model counts for *cis* and twisted peptides.

Omegalyze also produces 3D validation markup for *cis*-peptides, available in the multi-criterion kinemage generated through the command line:

```
phenix.kinemage file.pdb
```

As seen in the figures here, *cis* and twisted peptides are marked with green planes that fill the space between the C_α trace and the full mainchain trace at the site of the peptide of interest. For *cis*-peptides, this results in a green trapezoid shape. For twisted peptides, the trapezoid likewise becomes twisted, indicating the severity of the twist by the angle

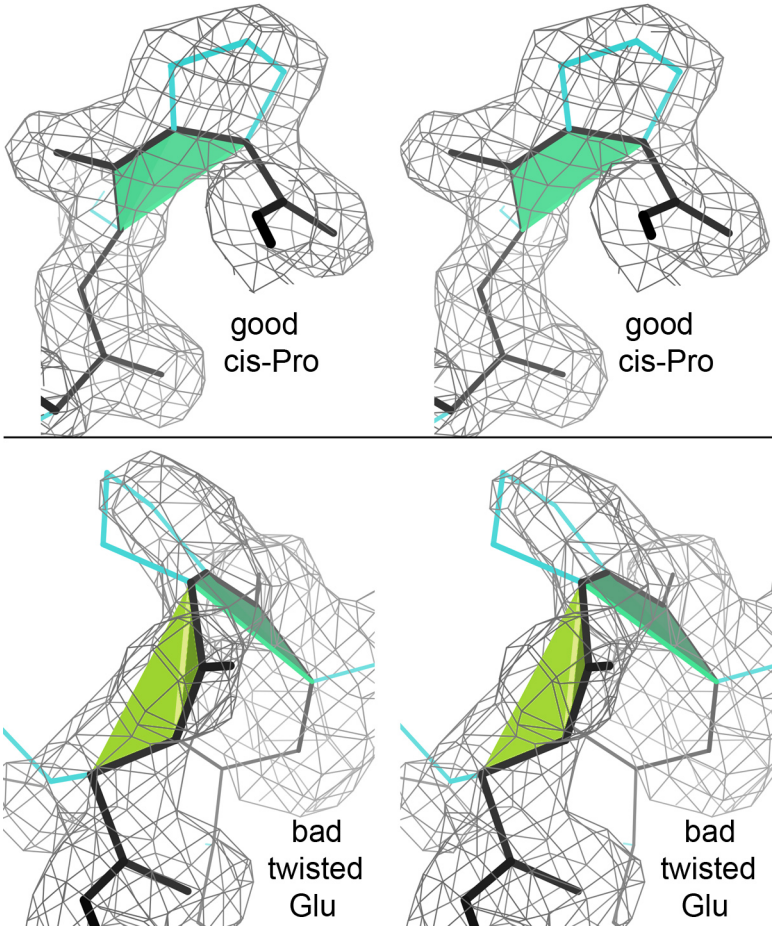


Figure 4: 2CN3: correct cis-Pro 406, incorrect twisted Glu 378.

between its two component triangles. *Cis*-peptides are colored in sea green: while the probable-outlier twisted peptides are colored in a more obnoxious lime green. These colors are intentionally similar to, but distinct from, the green used to mark Ramachandran outliers, our main method of backbone assessment. Selecting a vertex of the markup will display the calculated omega and the assigned category of the peptide.

Outside of Phenix proper, the omegalyze assessments will appear on the Richardson lab's MolProbity webserver, starting in version 4.2. They are part of the new LoRx mode for validation at low resolution along with CaBLAM (Williams 2014), but since *cis* peptides can be an issue at any resolution their diagnosis will always be done.

These various forms of omega validation will help alert users to the presence of excessive *cis* peptides in their models.

References

Berkholz DS, Driggers CD, Shapovalov MV, Dunbrack RL Jr, Karplus, A (2012) Non-planarity of peptide bonds: a common and conformation-dependent feature of proteins, *Proc Natl Acad Sci USA* **109**: 449-453.

Croll TI (2015) The rate of *cis-trans* conformation errors is increasing in low-resolution crystal structures, *Acta Cryst* **D**, in press.

Filman DJ, Matthews DA, Bolin JT, Kraut J (1982) Crystal structures of *Escherichia coli* and *Lactobacillus casei* dihydrofolate reductase refined at 1.7 Å resolution. I. General features and binding of methotrexate, *J Biol Chem* **257**: 13650-13662.

Lorenzen S, Peters B, Goede A, Preissner R, Frommel C (2005) Conservation of *cis* prolyl bonds in proteins during evolution, *Proteins* **58**: 589–595

Williams CJ, Hintze BJ, Richardson DC, Richardson JS (2013) CaBLAM identification and scoring of disguised secondary structure at low resolution, *Comp Cryst Newsletter* **4**: 9-10

FAQ

Tips for coordinated metal refinement

It is not uncommon to observe pronounced residual (difference) map features around metal ions. These features may originate from a number of possible reasons, such as: a) under-refined metal parameters, b) non-optimal metal parametrization, c) Fourier map artifacts, d) partial or/and shared occupancy, e) incorrect metal identity.

Provided that the metal identity is correctly assigned, refinement hints below may be helpful:

1. Ensure charge is in the model input file. In case of PDB file it is defined in rightmost part of ATOM record, for example:

```
HETATM 3241  SN   SN C   3      5.000   5.000   5.000  0.25 41.55          SN4+
```

2. Refine occupancy of the metal.
3. If it is a heavy metal (has substantially more electrons than typical macromolecular atoms C, O and N), refine anisotropic ADP of metal.
4. If it is anomalous scatterer refine f' and f''.
5. Run refinement until convergence. Usually it takes more than default 3 macro-cycles, about 5-10 macro-cycles.

If the residual map features are Fourier map artifacts then there isn't much one can do about it.

More details at

www.phenix-online.org/presentations/faq.pdf

Plan a SAD experiment, scale SAD data, and analyze your anomalous signal

Thomas C Terwilliger

Los Alamos National Laboratory, Los Alamos, NM

Correspondence email: terwilliger@lanl.gov

There are three new tools available to you in Phenix for planning and analyzing your SAD experiment. These tools are designed to help you decide how accurately you need to measure your data in order to solve the substructure, to scale your unmerged SAD data and to evaluate the signal you actually obtained in that SAD data.

Planning a SAD experiment with phenix.plan_sad_experiment

`phenix.plan_sad_experiment` is a tool for estimating the anomalous signal that you might get from your SAD experiment and for predicting whether this signal would be sufficient to solve the structure.

You supply `phenix.plan_sad_experiment` with a sequence file, the anomalously scattering atom you plan to use for the experiment and the wavelength for data collection. `phenix.plan_sad_experiment` will estimate the necessary I/σ_I of your dataset to provide enough anomalous signal to solve the structure.

`phenix.plan_sad_experiment` will try various values of I/σ_I for your dataset at each of several resolutions. For each I/σ_I it will estimate the half-dataset anomalous correlation that would result along with the likely true correlation between your anomalous differences and those that would be calculated from a final model of your structure (cc^*_{ano}). From this anomalous correlation (cc^*_{ano}), `phenix.plan_sad_experiment` will estimate the anomalous signal (related to cc^*_{ano} by the square root of the number of reflections divided by the square root of the number of sites). Then `phenix.plan_sad_experiment` will choose a value of I/σ_I that gives an anomalous signal of about 15 (if achievable with the maximum I/σ_I you specify).

The way that `phenix.plan_sad_experiment` and `phenix.anomalous_signal` estimate the probability that you can solve your dataset is to compare the anomalous signal in this dataset with the anomalous signal in other datasets at the same resolution. Then the fraction of similar datasets that can be solved by HySS is used as the probability that the anomalous substructure for your dataset will also be found.

Similarly, the mean figure of merit for datasets with an estimated anomalous correlation (cc^*_{ano}) similar to that for your data is used as an estimate of the figure of merit that you would obtain if the substructure is found for your crystal.

Scale unmerged anomalous data or multiple datasets with phenix.scale_and_merge

`phenix.scale_and_merge` is a tool for scaling unmerged anomalous data or multiple data files and creating a scaled dataset and two scaled half-datasets.

You supply `phenix.scale_and_merge` with a directory containing data files or the name of a single unmerged data file. You can optionally also specify a pair of labels that identifies datasets that are to be kept together. For example if you collected your data as pairs of data files with inverse beam geometry, you might have called the members of a pair `data_1_0_w1.HKL` and `data_1_0_w2.HKL`, related by w1 and w2.

`phenix.scale_and_merge` will first check the cell dimensions of all the datasets. Normally it will choose the largest set of similar crystals (you can have it keep all datasets with `only_similar_datasets=False`). It will also check the anisotropy in all the data files and calculate the average anisotropy (to be applied by default to all data files before scaling).

`phenix.scale_and_merge` will then scale all the data together to create an overall scaled dataset. This will be done in several steps. First `phenix.scale_and_merge` will split your data files into smaller files if your data files contain duplicate measurements of the same indices. The intensities in each data file will be adjusted for anisotropy to match the average anisotropy of all the data files. In this way all the data files are matched but the overall character of the data is not changed. Next, `phenix.scale_and_merge` will scale each individual file with local scaling.

Once all the individual data files have been scaled with local scaling, `phenix.scale_and_merge` will merge all the scaled files together. Merging of the individual datasets is done twice and then optionally two additional times to optimize anomalous differences.

In the first merging the individual datasets are simply averaged with weights based on the sigma for each reflection. Then the merged dataset is used as a reference and each individual dataset is compared to it. This allows an estimation of dataset variances (estimates of systematic differences between datasets and the mean). The dataset variance plus the individual variances are then to be used as an estimate of the total variance for each reflection. The second merging uses these total variances in weighting rather than the original sigma values.

If anomalous differences are optimized (default with `optimize_anomalous=True`), merging is carried out another time in order to optimize the weighting of anomalous differences in the merging step. For each unique reflection in the asymmetric unit of each dataset, the *I+* and *I-* are used to calculate anomalous differences. The anomalous differences from each individual dataset are then compared with the anomalous differences from the merged dataset to estimate individual dataset anomalous difference variances. Then the anomalous differences from each individual dataset are averaged, with weights based on the original sigmas and the dataset variances. These merged anomalous differences are then used to replace the anomalous differences in the merged dataset above (for example, a reflection in the merged dataset above that has *I+*, σ_{I+}, *I-* and σ_{I-} would get new values of *I+* and *I-* that have a difference equal to the appropriate merged anomalous difference, but the same mean as before.) The correlation of the anomalous differences in the original merged dataset and after optimization is printed (this should be high, for example 0.80).

The original datasets are then split into two parts for creation of two half-datasets. These half-datasets are useful for estimating the quality of the data and are used in `phenix.anomalous_signal` for this purpose.

The splitting into half-datasets is done in one of four ways with the method chosen based on the number of anomalous differences available for comparison using each method. With each method the data in each half-dataset are scaled just as the entire dataset was scaled. The preferred method is to split by files. Half of the data files are used to create each half-dataset. The next preferred method is to split with the first half of each dataset in one half-dataset and the second half in the other. The third preferred method is splitting alternate reflections with each unique index (after mapping to the asymmetric unit) into the two half-datasets. The last

method is to randomly assign reflections to the two half-datasets. The reason for this hierarchical approach is that reflections measured close in time, within the same dataset are better matched than those measured further in time or within different datasets. These approaches for splitting the data files attempt to pair anomalous differences measured close in time and in the same dataset.

Analyzing the anomalous signal in a SAD dataset with phenix.anomalous_signal

Once you have scaled your data with `phenix.scale_and_merge`, you can use `phenix.anomalous_signal` to analysis the anomalous signal in your data and to predict whether this signal is sufficient to solve the structure.

You supply `phenix.anomalous_signal` with scaled anomalous data, two half-dataset files with scaled anomalous data, the number of sites or a sequence file and name of the anomalously scattering atom. `phenix.anomalous_signal` will calculate the anomalous signal in your dataset from (1) the half-dataset anomalous correlation, (2) the skew of the anomalous difference Patterson map and (3) the estimated measurement error in your data.

`phenix.anomalous_signal` will then estimate the probability that you can solve this dataset using likelihood-based HySS (standard run) and will estimate the figure of merit of phasing that you should obtain.

The way that `phenix.plan_sad_experiment` and `phenix.anomalous_signal` estimate the probability that you can solve this dataset is to compare the anomalous signal in this dataset with the anomalous signal in other datasets at the same resolution. Then the fraction of similar datasets that can be solved by HySS is used as the probability that your dataset will also be solved.

With these new tools you should now be able to plan and carry out your SAD experiment even more effectively than before!

Validation of carbohydrate structures in CCP4 6.5

Jon Agirre and Kevin Cowtan
Department of Chemistry, The University of York, YO10 5DD (UK)
Correspondence email: kevin.cowtan@york.ac.uk

Introduction

Pyranose and furanose sugars, as most other cyclic compounds, have strong conformational preferences that are dictated by a minimization of angle, torsional and steric strains. For most of the biologically relevant pyranoses, the preferred conformation is either a $^4C_1$ or a $^1C_4$ chair, and any transitions to higher-energy conformations (*e.g.* half-chair or envelope) are usually a consequence of external factors such as the neighboring presence of catalytic residues from a carbohydrate-active enzyme.

While the set of geometric restraints that crystallographic refinement software impose is usually descriptive enough to reproduce a realistic geometry for amino acids modeled at medium to low resolution, cyclic sugars may end up in a high-energy conformation that, in the absence of clear density supporting it, should be treated as an outlier. Even with the addition of harmonic torsion restraints, any subtle mistakes in the specification of bonding distances – linkages between sugars need to be explicitly declared – or a wrong three-letter code selection (*e.g.* using 'GLC' for β-D-glucopyranose, together with the restraints designated for it) can result in distortion.

Conformational analysis

The method proposed by Cremer and Pople (1975) has been chosen as primary conformational analysis tool. The algorithm, which is applicable to rings of any cardinality, calculates a minimal set of puckering coordinates that describe each conformation. A total puckering amplitude term (*Q*) is also calculated

$$Q = \sqrt{\sum_{j=1}^{N} (\vec{R}_j \cdot \vec{n})^2} = \sqrt{\sum_{j=1}^{N} Z_j^2}$$

for *N* atoms, with $\vec{R}_j$ being the positional vector of atom *j* in a coordinate system with the origin in the ring's geometrical center, and $\vec{n}$ being the unit vector normal to the ring's mean plane. Therefore, Z_j accounts for the vertical displacement of atom *j*

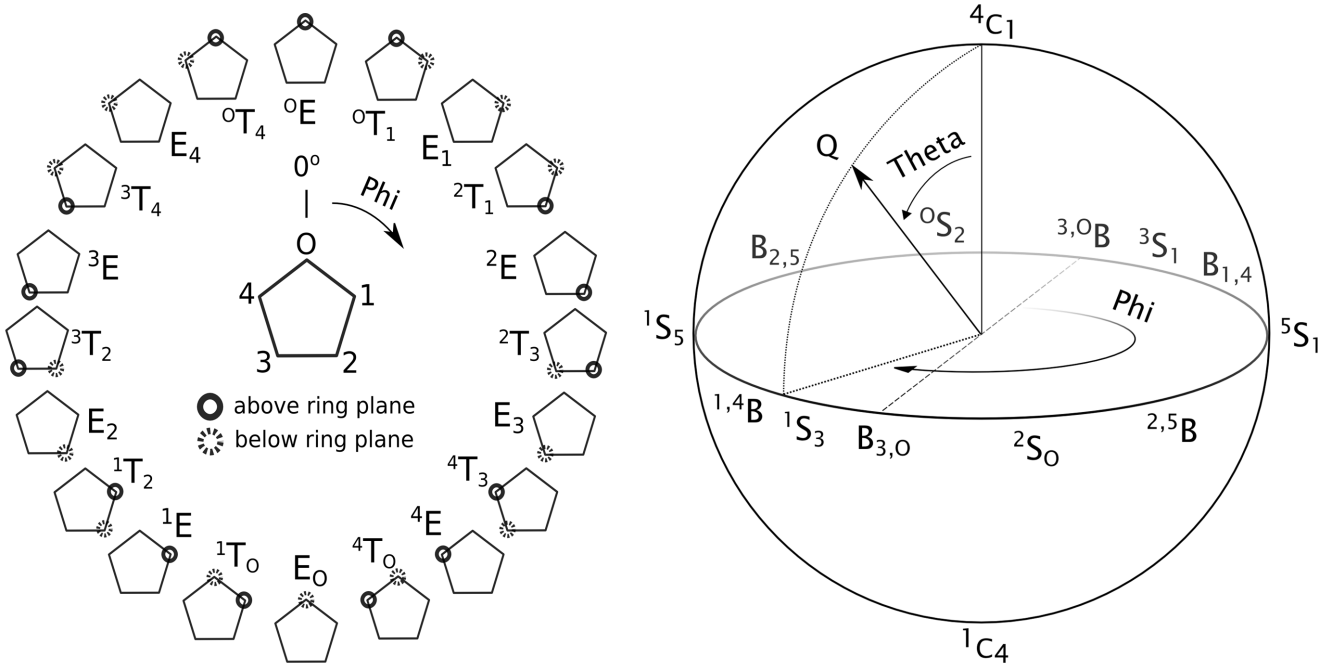


Figure 1: Correspondence between the Cremer-Pople angles for furanoses (Φ, image on the left) and pyranoses (Φ, Θ) to the conformation codes defined by IUPAC.

from the mean ring plane. In the case of pyranoside rings, the puckering coordinates are most conveniently expressed in angular form ($\Phi_{[0,2\Pi]}, \Theta_{[0,\Pi]}$) by solving the following set of equations

$$Q \sin\Theta \cos\Phi = \sqrt{\frac{1}{3}} \sum_{j=1}^{6} Z_j \cos\left[\frac{4\Pi(j-1)}{6}\right]$$

$$Q \sin\Theta \sin\Phi = \sqrt{\frac{1}{3}} \sum_{j=1}^{6} Z_j \sin\left[\frac{4\Pi(j-1)}{6}\right]$$

$$Q \cos\Theta = \sqrt{\frac{1}{6}} \sum_{j=1}^{6} (-1)^{j-1} Z_j$$

so that they can be graphically represented on the surface of a sphere of radius Q. This sphere, with lowest energy $^4C_1$ and $^1C_4$ ($\Theta = 0$ and $\Theta = \Pi$) chair conformations on the North and South pole respectively, is able to depict every conformational itinerary followed by pyranose sugars in their transition from their low-energy chair conformation to a more distorted boat or skew-boat intermediate ($\Theta = {}^{\Pi}/_{4}$) during catalysis (Davies *et al.*, 2011). For convenience, additional vertical displacements akin to the Z_j ones are calculated for those atoms implicated in the anomer and handedness detection.

A similar calculation is performed for furanose rings, but producing just Q and Φ .

Characteristics

Privateer-validate relies on a small database of three-letter codes for which the anomer, handedness and lowest-energy conformation have been calculated. By comparison to these values, the program is able to determine, for instance, if a modeled carbohydrate has been distorted from its initial conformation. When run within CCP4i2 (currently in alpha test phase), an HTML report is displayed with the IUPAC-compliant conformation code, the Cremer-Pople parameters and diagnostics for each sugar. The equivalence between Cremer-Pople angles and conformations can be visualized in Figure 1.

In addition to chemical correctness checks, a real space correlation coefficient is calculated for each sugar against an mFo-DFc map computed omitting all sugar models from the phase calculation. The resulting map coefficients can also be output to an MTZ file for later use.

Whenever glycosylation is present in the input structure, the program will produce linear descriptions of the detected trees. Here is an example of the nomenclature used:

$$\text{Man}\leftarrow\alpha_2\text{–Man}\leftarrow\alpha_6\text{–Man}\leftarrow\alpha_6\text{–(Man}\leftarrow\alpha_3\text{–)Man}\leftarrow\beta_4\text{–GlcNac}\leftarrow\beta_4\text{–GlcNac} \leftarrow \beta\text{–ASN}_{523}$$

Coot script files (Emsley *et al.*, 2010) are also produced with a guided tour of the detected issues. These scripts can be used manually outside CCP4i2 or by simply selecting 'Manual model rebuilding' within the aforementioned graphical interface. The omit mFo-DFc map is presented in pink color while 2mFo-DFc density is displayed in blue. Each button contains a description of the issue or issues detected by privateer-validate, as it can be seen in Figure 2.

The produced startup scripts also activate torsion angle restraints by default; using them in combination with the 'sphere refinement' function (hotkey: 'R') makes most of the issues exposed by privateer-validate easily fixable in Coot. Torsion angle restraints may be subsequently required by refinement software in order to avoid further distortions.

Availability

The Privateer software package can be obtained as part of the CCP4 distribution (http://www.ccp4.ac.uk). The validation software presented here serves as the prelude to a sugar detection and modeling tool that will be distributed in the forthcoming weeks as an update to CCP4 6.5. Privateer uses the Clipper libraries (Cowtan, 2003) and is distributed under the terms of the GNU Lesser General Public License.

Acknowledgements

The authors would like to thank Professors Eleanor J Dodson, Keith S Wilson and especially Gideon J Davies for many stimulating discussions.

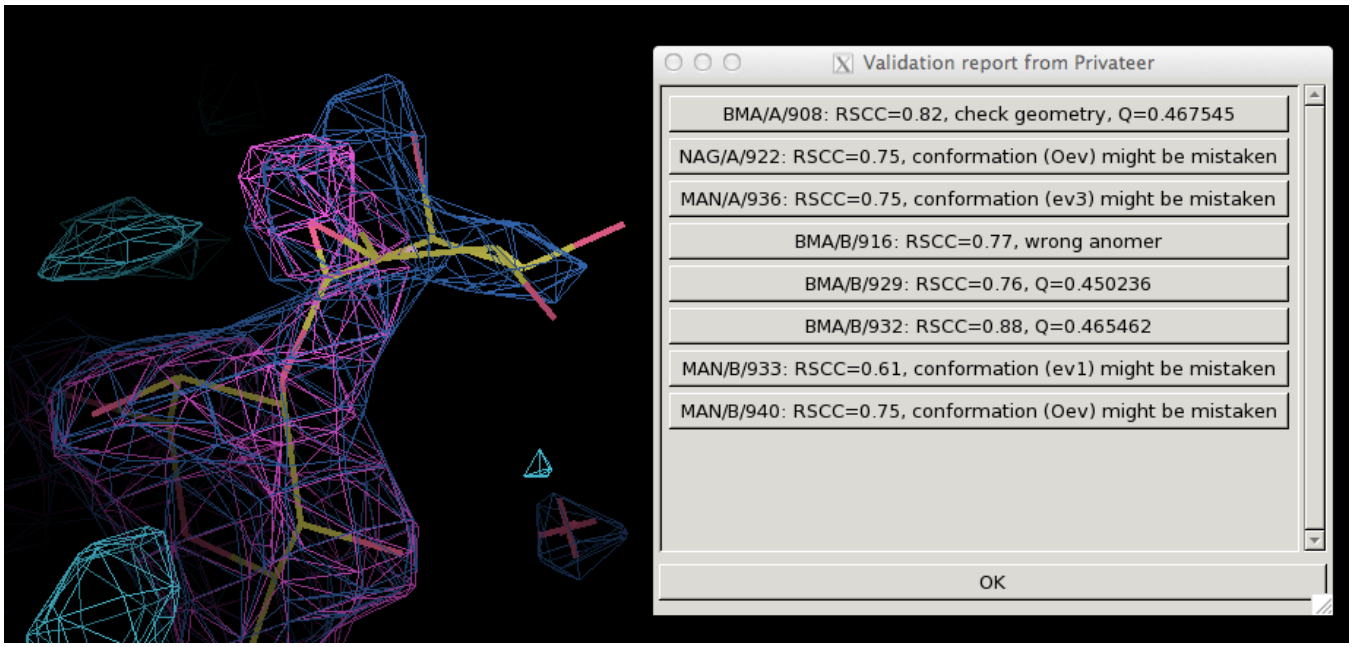


Figure 2: Validation of a glycoprotein (PDB code: 4IID). A number of terminal sugars display a high-energy conformation (envelope) as a consequence of the weak density they have been modeled in and the absence of torsion restraints in the original refinement.

References

Cowtan, K. (2003). The Clipper C++ libraries for x-ray crystallography. IUCr Computing Commission Newsletter, 2:4–9.

Cremer, D. t. and Pople, J. (1975). General definition of ring puckering coordinates. Journal of the American Chemical Society, 97(6):1354–1358.

Davies, G. J., Planas, A., and Rovira, C. (2011). Conformational analyses of the reaction coordinate of glycosidases. Accounts of chemical research, 45(2):308– 316.

Emsley, P., Lohkamp, B., Scott, W., and Cowtan, K. (2010). Features and development of coot. Acta Crystallographica Section D: Biological Crystallography, 66(4):486–501.

Disulfide bond restraints

Oleg V. Sobolev,[a] Nigel W. Moriarty,[a] Pavel V. Afonine,[a] Bradley J. Hintze,[c] 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, Durham, NC 27710 USA*

Correspondence email: NWMoriarty@LBL.Gov

Starting from *Phenix* dev-1810 version, restraints across disulfide links have been implemented on one angle and two dihedrals, including handedness-dependent values for χ3 around the central SS bond. The angle is for C_β-S_γ-S_γ atoms on either side across the link, and the central dihedral restraint applies to C_β-S_γ-S_γ-C_β atoms. Current values in the Monomer Library were target 103.8° and esd 1.8° for the bond angle, and target 90.0° and esd 10.0° for the SS χ3 dihedral, but were not in use as restraints.

Small-molecule crystal structures documented both left-handed (Peterson *et al.* 1960) and right-handed (Oughton & Harrison 1959) conformations of the bridge, and there is an early protein survey in Richardson (1981). More recently, high-resolution protein structures show multiple rotamers and handedness-dependent SS dihedral values that systematically deviate from 90°. A survey of 1677 quality-filtered disulfides in the Top8000 dataset (Richardson *et al.* 2013) gives mean SS χ3 dihedral values of +93° ±11° and -86° ±9°, as shown in figure 1a. The χ2 values (which also span the link) have a broader but still useful distribution (figure 1b), with means +79° ±17°, +183° ±29°, and -73 ±17°. The C_β-S_γ-S_γ angle mean is 104.2° ±2.1°.

The restraint values have been updated accordingly, using the alternative-value syntax available in *Phenix* for the individual, non-periodic peaks of χ2 and χ3. The esd for χ3 was set at 10° and for χ2 at 20°.

A re-refinement of the `1ejg` crambin with these new targets reduced the χ3 deviations of the refined model from [15°, 9°, 3°] for its three disulfides to [12°, 5°, 1°]. These reductions are due to the new target values being closer to the high-res structure values. The bond and angle rmsds as well as the R factors are not significantly different.

These restraint value changes were introduced in dev-1950.

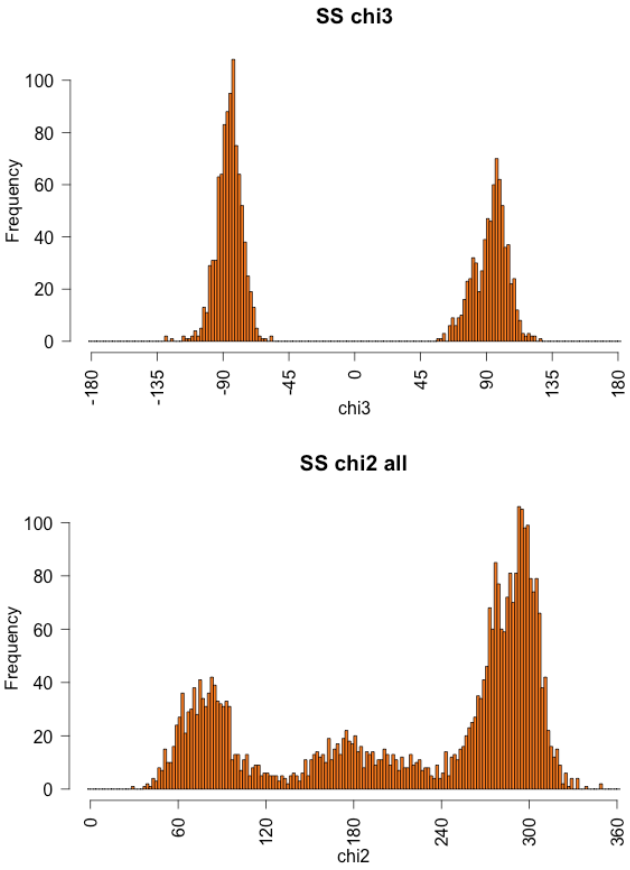


Figure 1: Histograms of the χ2 and χ3 values taken from the 1677 quality-filtered disulfide bridges.

References

Oughton BM, Harrison PM. *The crystal structure of hexagonal L-cystine*, Acta Cryst (1959) **12**: 396-404

Peterson J, Steinrauf LK, Jensen LH. *Direct determination of the structure of L-cystine dihydrobromide.* Acta Cryst (1960) **13**:104-109

Richardson JS. *Anatomy and Taxonomy of Protein Structures*, Adv Prot Chem (1981) **34**: 167-339

Richardson JS, Keedy DA, Richardson DC. *The Plot thickens: more data, more dimensions, more uses* (2013), pp. 46-61 in Biomolecular Forms and Functions: A Celebration of 50 Years of the Ramachandran Map, ed. Bansal M, Srinivasan N, World Scientific Publishing, Singapore, ISBN 978-981-4449-13-27

Improved Probabilistic Estimates of Biocrystal Solvent Content

Christian X. Weichenberger[a] and Bernhard Rupp[b,c]

[a]*Center for Biomedicine, European Academy of Bozen/Bolzano (EURAC), Viale Druso 1, I-39100 Bozen/Bolzano, Italy*
[b]*Department of Forensic Crystallography, k.-k. Hofkristallamt, 991 Audrey Place, Vista, CA 92084, USA*
[c]*Department of Genetic Epidemiology, Medical University Innsbruck, Schöpfstrasse 41, A-6020 Innsbruck, Austria*

Correspondence email: br@hofkristallamt.org, Christian.Weichenberger@eurac.edu

Introduction

When copies of biomolecular chains assemble into a crystal lattice, the molecules occupy a certain fraction of the available space, and the remainder is filled with solvent. As early as in 1968 Matthews (Matthews, 1968) has addressed the question concerning the distribution of fractional solvent in the crystallographic asymmetric unit. In his work, he defined the quantity V_M, nowadays known as the Matthews coefficient, as the fraction of the asymmetric unit volume V_A and the molecular weight M, $V_M = V_A/M$ and derived the equation for the solvent content $V_S = 1 - 1.230/V_M$. Analysis of 116 protein crystal structures has shown that solvent content ranged from 27% (almost spherical packing density) to 65%, with the most common value of approximately 43%. In the very early stages of structure determination, prior knowledge of the solvent content distribution allows estimating the number of molecules that can be present in the asymmetric unit. As suspected by Matthews (Matthews, 1976) and as demonstrated by Kantardjieff and Rupp in 2003 (Kantardjieff & Rupp, 2003), a distinct correlation of higher experimental resolution with lower solvent content exists. This dependency is accounted for in the Matthews probability (MP) calculator, *MATTPROB*, a web applet publicly available at www.ruppweb.org/mattprob to compute the oligomerization probabilities given the experimental resolution, unit cell parameters, crystal space group, and the macromolecule's weight. In this short communication we summarize the results from an update (Weichenberger & Rupp, 2014) ten years after the initial publication of the MP calculator (Kantardjieff & Rupp, 2003), describe the web interface, and present an alternative, non-parametric approach to compute the probabilities, which has become the default mode in the web interface.

Ten years of probabilistic solvent content estimates: an update

Motivated by Matthews' early studies on protein solvent content (Matthews, 1976), about ten years ago Kantardjieff and Rupp (Kantardjieff & Rupp, 2003) carried out a systematic analysis on more than 15,000 Protein Data Bank (PDB) (Bernstein *et al.*, 1977, Berman, 2008) entries determined by X-ray crystallography available at that time. The amount of protein structures was sufficient to statistically investigate the correlation between solvent content and molecular weight and experimental resolution. A weak tendency was recognized that V_M correlates to molecular weight, but experimental resolution was a much clearer discriminator of V_M: Protein structures that are packed more tightly tend to diffract better. This insight gave rise to the computation of Matthews probabilities based on the distribution of V_M conditional on experimental resolution. The approach relies arranging the resolution range of 1.2 Å to 3.5 Å for protein crystals in 13 bins, and separate, non-binned treatment of nucleic acid chains and protein/nucleic acid complexes. A bin is here defined as the resolution range from 0 Å (highest possible resolution) down to any of the 13 points for proteins or the full range of resolutions in the two non-binned cases. For each such bin, the distribution of V_M is parameterized by a modified extreme value function, which at the heart is a Gumbel (Gumbel, 1941) probability density function with additional scaling parameters. This empirical fit function serves as the analytical probability density function for probing the possible oligomerization number m of a query molecule with molecular weight M by reporting the probabilities of V_M as a function of $m \times M$, which by definition of V_M corresponds to values of V_M/m. The advantage of introducing resolution as a parameter in probability calculation has been demonstrated through examples where a resolution-agnostic

Table 1: Number of PDB homo-oligomers (column labeled "Nr.") by oligomerization number *n*, furnished with probability of occurrence P(*n*). From our dataset of 50,190 homo-oligomers we identified 30 distinct oligomerization numbers by grouping identical SEQRES records. Less than half of the entries are monomers but more than three quarters of the oligomers are monomers and dimers. These numbers may give a guide when estimating the oligomerization number from solvent content. Unsurprisingly, odd homo-oligomerization numbers are less frequent when compared to their even neighbors. For example, there are only four 11-mers, compared to hundreds of 10-mers and 12-mers.

n	Nr.	P(*n*)	*n*	Nr.	P(*n*)	*n*	Nr.	P(*n*)	*n*	Nr.	P(*n*)	*n*	Nr.	P(*n*)
1	21984	0.44	7	69	1.37×10^{-3}	13	6	1.20×10^{-4}	22	4	7.97×10^{-5}	44	2	3.98×10^{-5}
2	17143	0.34	8	783	1.56×10^{-2}	14	41	8.17×10^{-4}	24	44	8.77×10^{-4}	45	1	1.99×10^{-5}
3	2169	4.32×10^{-2}	9	51	1.02×10^{-3}	15	17	3.39×10^{-4}	28	11	2.19×10^{-4}	48	4	7.97×10^{-5}
4	5440	1.08×10^{-1}	10	199	3.96×10^{-3}	16	62	1.24×10^{-3}	30	6	1.20×10^{-4}	54	1	1.99×10^{-5}
5	373	7.43×10^{-3}	11	4	7.97×10^{-5}	18	14	2.79×10^{-4}	32	2	3.98×10^{-5}	56	1	1.99×10^{-5}
6	1390	2.77×10^{-2}	12	321	6.40×10^{-3}	20	45	8.97×10^{-4}	36	1	1.99×10^{-5}	60	2	3.98×10^{-5}

computation would have lead to a different favored oligomerization number.

It is important to note that the calculation of MPs assumes the Bayesian argument that the observed resolution represents an empirical lower limit of the true diffraction potential of the crystal: it has diffracted to at least the reported resolution, but in theory could have diffracted better. This is reflected in the definition of the resolution bins described above as they collect data from all structures with at least the resolution specified by the bin. Furthermore, the calculator reports probabilities, thus chances to find a different oligomerization state other than that associated with the highest probability are real. This happens if the crystal's V_M is different from the mode of the distribution.

In our follow-up work we have reexamined the statements from the 2003 publication and addressed several other questions that arose in the literature during the past decade. We followed the same data mining protocol as presented in (Kantardjieff & Rupp, 2003): from the initial set of 77,481 crystal structures we removed highly redundant entries and found 60,218 protein structures, 998 nucleic acid structures and 2,414 structures of protein/nucleic acid complexes. Of the 50,190 protein structures consisting of homo-oligomers we did not find any dependency of solvent content on oligomerization number, confirming previous findings reported by (Chruszcz *et al.*, 2008). A comprehensive list of oligomerization numbers and occurrences in PDB is given in Table 1.

We emphasize the importance of an accurate estimate of molecular weight *M*, as this becomes a sensitive parameter when investigating structures with an expected high number of oligomers. For this reason, in the web interface we have removed the potentially misleading species-dependent option to compute the molecular weight from sequence length (i.e., number of residues). Instead, links to compute the actual molecular weight from sequence are provided.

In principle the prior probability P(*n*) (*cf.* Table 1) could be used to always bias the resolution dependent prediction of oligomerization states *m* as P(*m*|n,res). However, very often a strong biological prior exists based on experimental knowledge that will lead to a corrected posterior estimate of the solvent content. Therefore, our calculator explicitly allows including such a strong biological prior by selecting a probable oligomerization state. We believe that an informed decision by the user to include a strong biological prior should override the automatic imposition of a generic P(*n*) prior[1].

[1] The effect of simply weighting the MP by P(*n*) can be tested by selecting the 2013 parameter set and sending the GET string to the server with *matt_prob_linux_pn* instead of default *matt_prob_linux*.

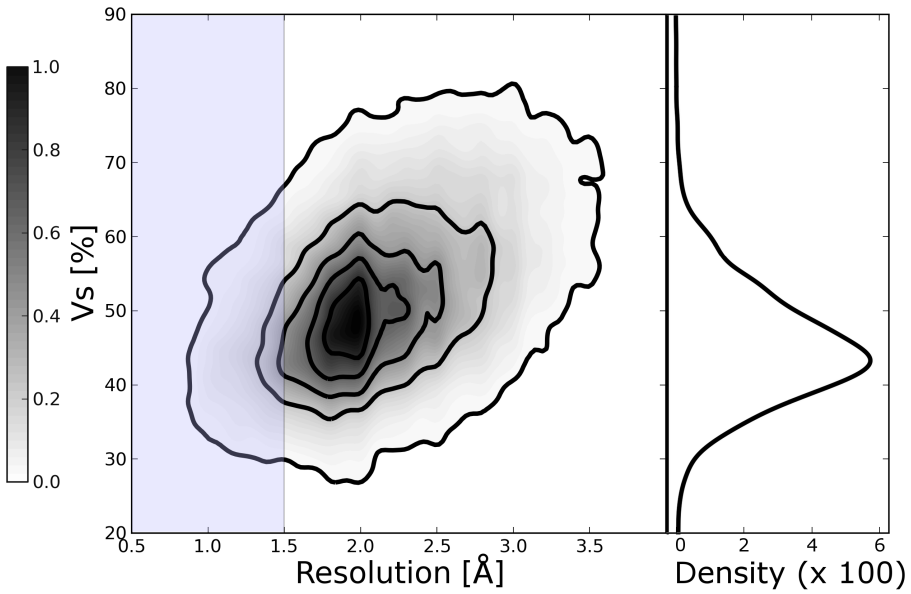


Figure 1: Parameter-free kernel density estimate of 60,218 pairs of solvent content V_S and resolution retrieved from PDB. In the middle part of the figure we show the two-dimensional kernel density estimate with an axis-aligned bivariate normal kernel of the full data set, normalized to have 1.0 as the maximum. A clear tendency towards lower values of V_S is visible for higher (better) resolution. The right hand side of the figure displays the one-dimensional probability density function of the highlighted region in the central figure that consists of approximately 6,200 protein structures resolved at a resolution of 1.5Å or better. We notice that the mode of this distribution is shifted towards higher crystal packing compared to similar distributions that include structures with lower resolution. This figure was generated with matplotlib (Hunter, 2007) and R (R Core Team, 2012).

We applied principal component analysis on the set of 50,190 homo-oligomeric protein structures using the observed variables resolution, molecular weight, and V_M. The analysis corroborates use of resolution as the most important and single variable to describe V_M, and minor, insignificant changes were observed when investigating the full data set of 60,218 protein structures. We repeated the parameter fitting process with the updated data sets and published the parameter set and its use in the web interface.

When comparing the distributions of V_M and V_S, we observed that V_S is a much better behaved distribution function in the sense that it is more symmetric and less tailed, indicated by a great reduction of sample skewness. This motivated us to favor V_S over V_M when constructing a parameter-free version of MP in the R programming language (R_Core_Team, 2012) by reconstructing the density function for pairs of resolution and V_S observed in the PDB with a two-dimensional kernel density estimator with an axis-aligned bivariate normal kernel (Wand, 1994, Wand & Jones, 1995). We arrive at the function $P_r(V_S) = P(V_S \mid \text{resolution} \leq r)$, i.e. the probability density function under the Bayesian assumption that the crystal has diffracted to at least resolution r, where resolution r has previously been observed in the PDB. Figure 1 presents the two-dimensional kernel estimate of P(V_S, resolution) with an example of its one-dimensional projection $P_r(V_S)$ for a fixed resolution r = 1.5Å.

The *MATTPROB* web interface has been updated with the new parameter set and the parameter-free version of the MP calculator. In Figure 2 we give a real world example taken from PDB entry 3vto, a tail-forming metal binding protein from bacteriophage Mu (Harada *et al.*, 2013) used to penetrate the host membrane during the infection process, which crystallized as a hexamer at resolution 1.44Å. Without using resolution as prior information, a pentamer would be the most likely oligomer. Using the kernel-estimated probability density function $P_{1,44}(V_S)$ with data arriving from 4,042 proteins determined at a resolution of 1.44Å or better, a hexamer is predicted as the most probable oligomer. Entering additional information that the tail protein

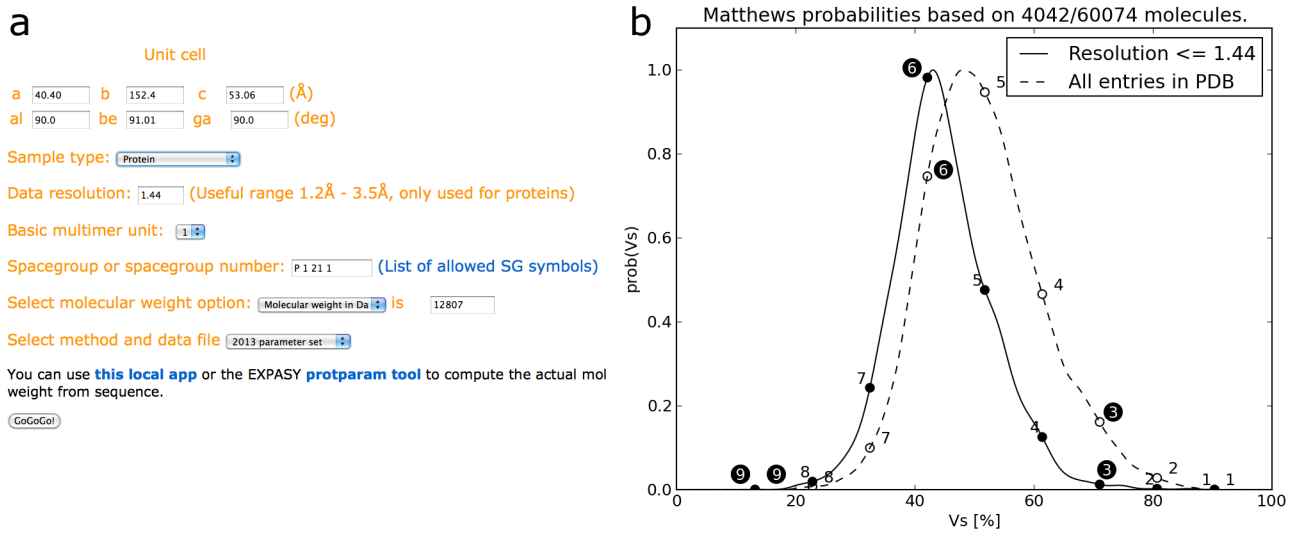


Figure 2: **Usage of the *MATTPROB* web applet. (a)** Web interface of *MATTPROB*, **www.ruppweb.org/mattprob**, filled with the unit cell parameters of PDB entry 3vto, where we have computed the molecular weight of the protein chain with 115 residues as M = 12,807 Dalton. The first row of values represents the unit cell axes lengths, and in the second row the respective angles alpha, beta, and gamma are input. The *Sample type* drop down menu offers three types of macromolecules: proteins, nucleic acids, and complexes of protein and nucleic acids. The input field *Data resolution* expects the experimental resolution in units of Ångströms for the query protein crystal. In the other two cases involving nucleic acids, resolution-dependent probability calculation is only available with the parameter-free kernel-based estimator and should be critically examined due to sparse data in PDB. In case there exists prior knowledge of functional oligomerization, the *Basic multimer unit* drop down menu allows restricting computation of probabilities to multiples of the specified chain(s). In the case of 3vto, this number could be set to three (see panel (b), black circled numbers, for the result). The Hermann-Mauguin space group symbol is input in the *Spacegroup or spacegroup number* field, followed to the right by a supporting link to a list of all allowed symbols and settings. The molecular weight M is supplied in the following line, and ultimately the MP calculation method is chosen by *Select method and data file*, where either the 2003 or 2013 parameterized version or the new 2013 kernel-based estimator can be chosen, the latter is set as the default method. Computation of MPs is initiated pressing the *GoGoGo!* button and after a few seconds the results are presented on a separated page. The underlying *MATTPROB* program using the kernel density estimator has been implemented in the Python programming language and utilizes *matplotlib* (Hunter, 2007) as its plotting backend. The parameterized versions of *MATTPROB* have been implemented using the original 2003 FORTRAN code. **(b)** MP graph. The summary graph shows possible values of V_S and associated probabilities for the resolution-dependent density function as well as for the resolution-agnostic version. The title of the plot informs about the number of PDB entries used for computing the probability density function, and it should be underlined that the higher the resolution of the query crystal, the lower the number of PDB entries found, i.e. fewer data points have been available for density estimation. Only the numbers in black are output when selecting a trimer as basic biological unit in the *Basic multimer unit* drop down menu. In the graphs, the probability density functions are always normalized to have a maximum value of 1.

assembles as a trimer then points to a hexameric structure in the ASU, excluding the possibility of a single trimer and the pentamer (Figure 2b, dark circles around predicted oligomerization number).

Example: Implementation of the *MATTPROB* kernel estimator in R

Many programming languages come with scientific libraries that support kernel density estimations. For the sake of brevity and simplicity, we demonstrate computation of a resolution dependent kernel density estimate in the statistical programming language R. From our website, www.ruppweb.org/mattprob, the three data files for protein, nucleic acids, and protein/nucleic acids complexes, respectively, can be downloaded. The R code below reads in the comma separated data file for protein crystal structures, and filters for PDB entries with resolution better or equal to 1.44 Å (variable `maxRes`). We construct a kernel density estimate

```
library("KernSmooth")
max.res   = 1.44
vs        = 51.77
dat       = read.csv("pdb_02_06_2013_pro_sorted_flagged_highest_cs.csv")
trunc.dat = dat[ dat$reso<=max.res, ]
dens      = bkde(trunc.dat$vs, gridsize=500)
i         = max(which(dens$x<=vs))
vs.x      = dens$x[i]
dens.y    = dens$y[i]
xlim      = c(0, 100)
ylim      = c(0, 0.06)
plot(dens, xlab="Vs [%]", ylab="Density", type="l", xlim=xlim, ylim=ylim)
par(new=TRUE)
plot(vs.x, dens.y, xlim=xlim, ylim=ylim, xlab="", ylab="", axes=FALSE)
```

Schema 1: Code fragment for implementing a fully functional *MATTPROB* calculator.

of the probability density of the solvent content from these truncated data using the `bkde` call from the previously imported `KernSmooth` library. The result is stored in the list `dens`, with abscissa and ordinate values in `dens$x` and `dens$y`, respectively. We then find the abscissa index `i`, which corresponds to the solvent content closest to the query variable `vs`. (The value `vs`=51.77 is retrieved via Matthews' formula given in the Introduction for a pentamer for the above example PDB entry 3vto, see also Figure 2b.) Finally, we plot both the density estimate curve and the calculated point.

With the code fragment in schema 1, the interested reader can easily implement a fully functional *MATTPROB* calculator. We would like to point out that data files for nucleic acids and protein/nucleic acids complexes have a very limited number of entries, and therefore high quality resolution-dependent predictions cannot be expected for these types of macromolecules.

Conclusions

We have reinvestigated possible variables for predicting the most probable oligomer given the crystallographic unit cell dimensions, space group, and molecular weight. We found that experimental resolution is still the most powerful discriminatory variable for predicting solvent content from known protein structures. We have updated the parameter set to reflect the state of PDB in 2013. An MP calculator based on a parameter-free kernel density estimate of the solvent content probability density has been implemented and is the default mode of Matthews probability computation in the web interface at www.ruppweb.org/mattprob. A major advantage of this method is that it is free of any previously used binning approach and can be queried with any resolution currently found in the PDB. Finally, we have sketched the central *MATTPROB* calculation function in the R programming language giving a starting point for independent implementations. Supporting data files are available for download at www.ruppweb.org/mattprob/kernel_data_tables_2013.zip.

Acknowledgements

BR acknowledges support from the European Union under a FP7 Marie Curie People Action, grant PIIF-GA-2011–300025 (SAXCESS). The web site www.ruppweb.org is supported by the k-k. Hofkristallamt, Vista, CA 92084.

References

Berman, H. (2008). *Acta Crystallogr.* **A64**, 88-95.

Bernstein, F. C., Koetzle, T. F., Williams, G. J., Meyer, E. F., Jr., Brice, M. D., Rodgers, J. R., Kennard, O., Shimanouchi, T. & Tasumi, M. (1977). *J Mol Biol* **112**, 535-542.

Chruszcz, M., Potrzebowski, W., Zimmerman, M. D., Grabowski, M., Zheng, H., Lasota, P. & Minor, W. (2008). *Protein Sci* **17**, 623-632.

Gumbel, E. J. (1941). *Ann. Math. Statist.* **12**, 163-190.
Harada, K., Yamashita, E., Nakagawa, A., Miyafusa, T., Tsumoto, K., Ueno, T., Toyama, Y. & Takeda, S. (2013). *Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics* **1834**, 284-291.
Hunter, J. D. (2007). *Computing In Science and Engineering* **9**, 90-95.
Kantardjieff, K. A. & Rupp, B. (2003). *Protein Sci* **12**, 1865-1871.
Matthews, B. W. (1968). *J. Mol. Biol.* **33**, 491-497.
Matthews, B. W. (1976). *Ann Rev Phys Chem* **27**, 493-523.
R Core Team (2012). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing.
Wand, M. P. (1994). *J. Computational Graphical Statistics* **3**, 433-445.
Wand, M. P. & Jones, M. C. (1995). *Kernel Smoothing*. Boca Raton, Florida: Chapman and Hall/CRC Press.
Weichenberger, C. X. & Rupp, B. (2014). *Acta Crystallogr.* **D70**, 1579-1588.

Rapid Evaluation of Non-Bonded Overlaps in Atomic Models

Youval Dar,[a] Nigel W. Moriarty,[a] Jeffrey J. Headd[b], Jane S. Richardson[c], David Richardson[c], and Paul D. Adams[a,d]

[a]*Lawrence Berkeley National Laboratory, Berkeley, CA 94720*
[b]*Janssen Research & Development, LLC, 1400 McKean Road, Spring House, PA 19477*
[c]*Biochemistry Department, Duke University Medical Center, Durham, NC 27710*
[d]*Department of Bioengineering, University of California at Berkeley, Berkeley, CA 94720*

Correspondence email: ydar@lbl.gov

Introduction

Detecting and evaluating steric overlaps between non-bonded atoms is an important tool in the process of improving protein and nucleic acid structure models. Overlaps may arise between atoms of a single monomer or protomer, between solvent molecules or due to interactions between symmetry related atoms. *MolProbity* (Davis *et al.*, 2004), a suite of tools for structure validation, produces a *clashscore* – the number of steric overlaps of the electron cloud per 1000 atoms. In *MolProbity,* the overlaps of atoms electron density or clashes are evaluated by a rolling-probe algorithm (Word *et al.*, 1999). This is achieved by rolling a 0.5Å diameter sphere on the Van Der Waals (VdW) surface and recording a clash when VdW surface of non-bonded atoms overlap is ≥ 0.4Å. An overlap typically indicates local model stereochemistry issues. *MolProbity* performs a very comprehensive all-atom contact analysis (Chen *et al.*, 2010) and is an integral part of the model validation in the *Phenix* suite of programs (Adams *et al.*, 2010). However, there is a need to rapidly and repeatedly calculate non-bonded overlaps during structure refinement with phenix.refine (Afonine *et al.*, 2012), to report the current quality of the model at each refinement step.

Therefore, a tool has been developed in the *CCTBX* toolbox (Grosse-Kunstleve et al., 2002) that makes it possible to perform a rapid, but simplified analysis of overlapping atoms during the model refinement process. This algorithm makes some approximations compared to the more sophisticated approach used in *MolProbity*, but reports the overlaps due to crystal symmetry which are not detected currently by the rolling-probe algorithm as implemented in *MolProbity*. This CCTBX non-bonded overlaps (NBO) analysis is used to report on atomic overlaps during structure refinement, while the comprehensive structure refinement performed after refinement makes use of the more detailed *MolProbity* analysis.

In the process of model refinement *Phenix* analyzes all non-bonded interactions, including interactions between symmetry related copies. This process results in information on non-bonded interacting atoms pairs that is readily available for calculating the non-bonded overlaps. This method allows easy filtering of the overlaps into groups such as the total number of overlaps, overlaps due to symmetry operations and overlaps in the macromolecule (protein, DNA or RNA). Additional breakdowns can be easily added if the need arises using the selection mechanism available in *Phenix*.

In contrast to *MolProbity* (the rolling-probe algorithm), the *CCTBX* overlaps are evaluated from the difference between the model non-bonded atomic distance and the calculated VdW distance. The non-bonded overlaps (NBO) is therefore a count of the number of overlaps. A normalized NBO can be calculated by dividing by the number of atoms. Multiplying by 1000 places the normalized NBO on a similar scale to the *MolProbity clashscore.* However, since the methods for finding overlaps and clashes are different, the NBO count cannot be directly compared to the

MolProbity clashscore. The *CCTBX* NBO does not provide any graphical visualization and it is not intended to replace the *MolProbity clashscore* but to provide additional indicator for model quality during structure refinement.

Method and usage

Evaluation of the *CCTBX* non-bonded *clashscore* is performed as follows:

1. Create a *Geometry Restraints Manager* object (Grosse-Kunstleve *et al.*, 2002) and extract the non-bonded proxies (non-bonded atoms pairs list). While no explicit secondary structure or other parameters related to Hydrogen bonds are provided, the *Geometry Restraints Manager* identifies most of H-Bonds donor-acceptor pairs and adjust the Van der Waals distance to reflect hydrogen bonding.
2. Collect all overlapping non-bonded atoms where an overlap is defined as:

   $$R_{nonbonded} - R_{VdW} \leq 0.4\text{Å}$$

   where $R_{nonbonded}$ is the distance between the atoms and is the sum of the Van der Waals radii.
3. *Heavy Atoms – Hydrogen* overlaps are omitted when they are less than 5 covalent bonds apart.
4. Inline overlaps (eclipsing): In the situation where two bonded atoms are overlapping with the same non-bonded atom, there are two different situations. If all atoms are inline, an overlap is be considered as a single overlap even if the there is unacceptable overlap between all three atoms. If the non-bonded atom perpendicular to the bond of the bonded atoms and overlaps with both, it is considered two overlaps. This situation is illustrated in figure 1. Atoms are considered to be inline when the *abs(α)<45°*, where α is defined (see figure 1) as the angle between the X-H bond and the line from the center of the X-H bond and the second atom, Y. The value of *α* was chosen to be the inline limit based on manual inspection.
5. Eliminate double counting of clashes related to symmetry operation, since each symmetry overlap appears twice in the non-bonded overlapping atoms pairs list.

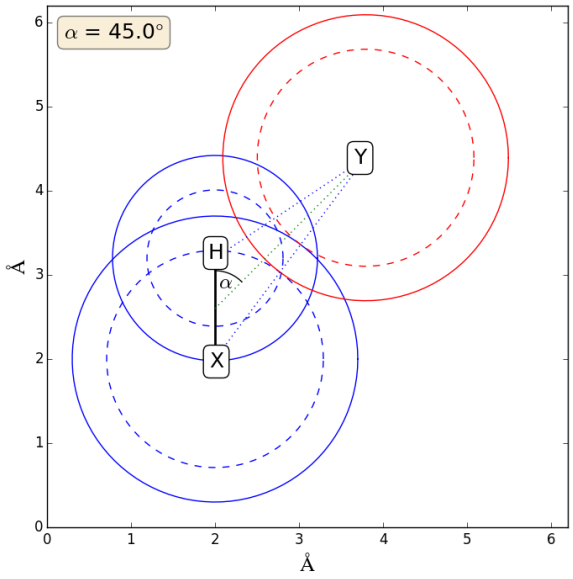


Figure 1: Collision between X-H and Y atoms. The solid circles are the VDW radii. The dash circles are the radius of allowed overlap. When the overlap exceeds the dashed circle it considered to be a clash. The angle α is the limit of inline, ecplipsed, overlap.

Currently three NBO values are provided:

1. All NBO: *number of ALL unique overlaps*.
2. Symmetry related NBO: *number of all overlaps due to symmetry*.
3. Macro molecule (protein, RNA or DNA) NBO: *number of overlaps in macro molecule, excluding symmetry related NBO*.

For comparison purposes, normalized NBO values were calculated for this work. One can obtain the normalized NBO as follows: *1000 x (number of unique clashes) / (number of atoms in the model)*. Note that this is the number of atoms in model, not necessarily the complete asymmetry unit of the structure. Note, also, that both the NBO value and the number of atoms are generally different from the number of clashes and the number of atoms used to calculate *MolProbity clashscore*. Differences include atoms with partial occupancy, solvent – solvent clashes, definition of bonded and non-bonded, and the inclusion of solvent atoms in model atom count.

When evaluating the normalized NBO, the number of atoms used to normalize is as follows:

1. All NBO in model:
   1000 x (number of ALL unique overlaps) / (number of ALL atoms in the model)
2. Symmetry related NBO:
   1000 x (number of overlaps due to symmetry) / (number of ALL atoms in the model)
3. Macromolecule NBO:
   The model considered for this include only protein, RNA and DNA
   1000 x (number of overlaps, excluding symmetry related) / (number of atoms in the macro molecule)

Command line usage

The non-bonded overlaps (NBO) can be calculated using the command-line script

> mmtbx.nonbonded_overlaps xxxx.pdb [options]

Because the NBO algorithm relies on both the VDW distance and bonding information, the presence of nonstandard residues requires a restraints CIF be supplied:

> mmtbx.nonbonded_overlaps xxxx.pdb xxxx.ligands.cif

A restraints *CIF* file can be obtain using *ReadySet!* or *eLBOW (Moriarty et al., 2009)*. A code implementation example is given in the appendix.

When used from the command line, if the model contain no Hydrogen atoms, Hydrogen are added using *phenix.reduce* with the following options:

1. Add hydrogens on OH and SH group (-oh)
2. Create NH hydrogens on HIS rings (-his)
3. Add H and rotate and flip NQH groups (-flip)
4. Fraction of std. bias towards original orientation (-pen9999)
5. Keep bond lengths as found (-keep)
6. Process adjustments for all conformations (-allalt)

PDB NBO Survey

To test the NBO and in particular analyze the symmetry-related NBO in the Protein DataBank (Berman *et al.*, 2000; Bernstein *et al.*, 1977), the NBO program was run on PDB files (based on the index of PBD structures at the Lawrence Berkeley Lab PDB mirror on Jan. 21 2015) filtered to contain a single model, no unknown residues and valid CRYST1 crystal symmetry records. This resulted in 50,646 models for investigation.

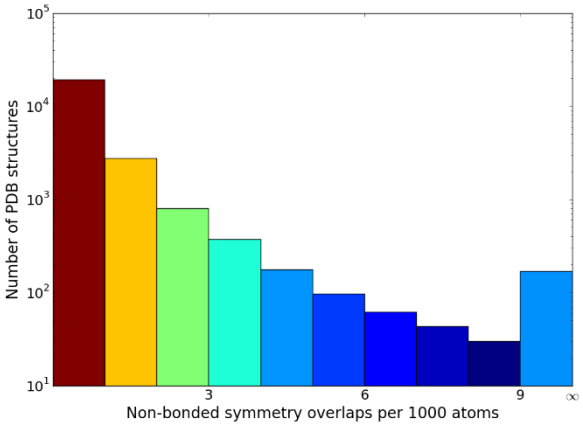


Figure 2: Macro molecule (models include only protein, RNA or DNA) normalized symmetry-related overlaps for 23,685 structures were MolProbity clashscore is less or equal to three and PDB structures with good crystal symmetry, a single model and no unknown atom pairs.

Recent work (Moriarty *et al.*, 2014) that investigated the relationship of the rmsd of the N-C_{α}-C angle in 25,976 refined structures and the starting *MolProbity clashscore* showed that structures with a *clashscore* greater than six failed to exhibit the expected reduced rmsd values for low resolution structures. In fact, if structures with *clashscore* values greater than six are included, the average rmsd for the N-C_{α}-C angle (and to a lesser extent all angles) was larger for structures in the 3.0-3.5Å range than at 2.0Å. This is likely due to regions of the model that are outside the radius of convergence of the refinement method and contribute to the higher *clashscore* value. A more conservative filtering choice for the *clashscore* of less than three removes the majority of models that have any poor quality regions. Filtering for *clashscore* less than three resulted in 23,685 entries to calculate NBO statistics (figure 2), the

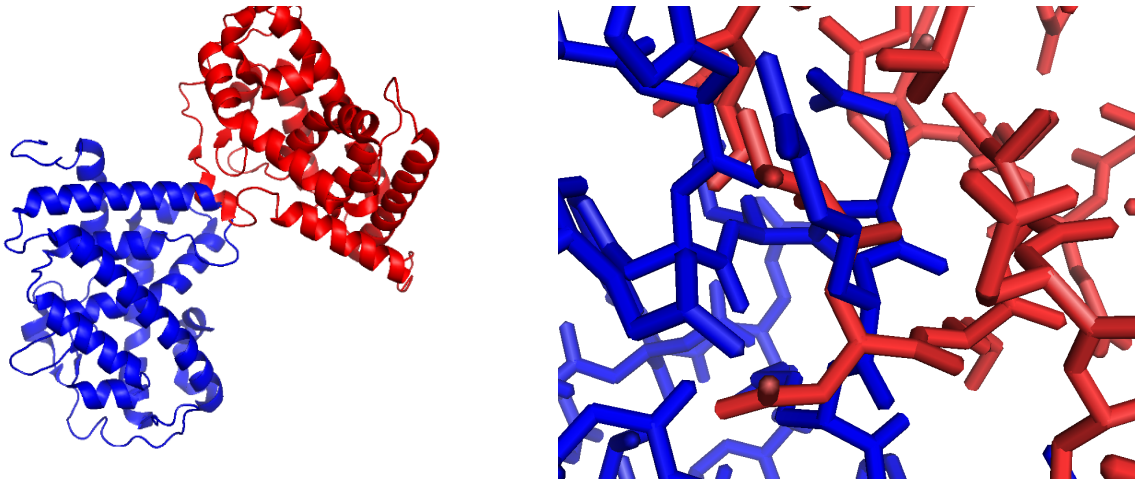


Figure 3: Overlap due to symmetry operation in 2H77 (DOI:10.2210/pdb2h77/pdb). On the right, a close up on the overlapping region.

histogram of normalized symmetry-related NBO. The symmetry-related NBO of the macromolecule counts overlaps that are not included in the *MolProbity clashscore*. If the models were uniform in quality, then the number symmetry-related overlaps should be less than the overlaps internal to the model. This is partly because the quality should be the same throughout the model and partly because there are fewer opportunities for overlaps on the model surface. Two reasons for the latter are that there are fewer atoms on a surface compared to the internal volume and the fact that less than the entire surface interacts with a symmetry surface. Therefore, it can be expected that the normalized symmetry-related NBO should be smaller than the upper limit of three chosen for the *clashscore*. However, the results of the NBO analysis (figure 2) show that there are a significant number of structures with elevated normalized symmetry-related NBO.

A striking example of an extreme symmetry-related overlap is shown in figure 3. This example, 2H77 (DOI:10.2210/pdb2h77/pdb), has a resolution of 2.33Å and was deposited in 2006.

Figure 4 explores symmetry-related overlaps over time — it shows the percent of the filtered structures deposited each year in the PDB that have a normalized symmetry NBO below the thresholds listed in the legend. The total number of deposited models is displayed in the bottom portion. It is clear that among the 50,646 suitable structures tested, there is

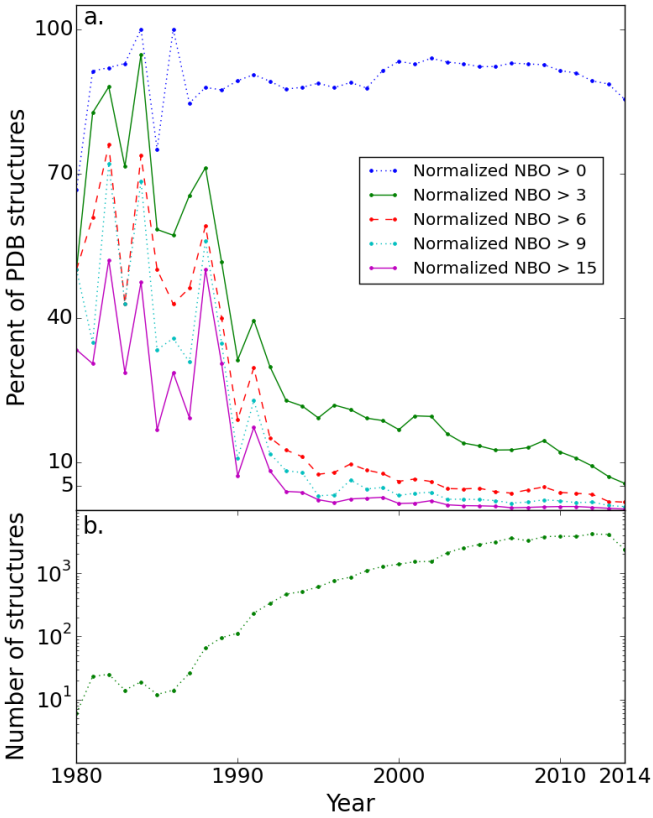


Figure 4: (a) Percent from the total structure number of structures with symmetry related NBO per 1000 atom larger than 0,3,6,9,15. (b) The number of structures with a single model, good CRYST1 records and no unknown atoms types vs. PDB deposition year. Total of 50,646 structures data points.

a notable improvement in the number of symmetry overlaps. This can be attributed to improved model validation during the PDB deposition process, improved structure refinement algorithms, and the use of advanced validation tools such as *MolProbity*. However, in 2014 85% of tested structures still had non-zero symmetry NBO values. Furthermore, considering a desirable value for a normalized total NBO to be less than three, in 2014 132 structures (5.5%) were worse than this.

Summary

A tool has been developed to rapidly evaluate non-bonded overlaps (NBO) in protein and nucleic acid structure models during structure refinement with phenix.refine. These NBO include symmetry related overlaps, and are calculated using VdW bond lengths and model distances. An analysis of the PDB revealed a significant number of structures with an elevated symmetry-related NBO, emphasizing the need to account for these kinds of interactions during structure refinement and ultimately validation. The combination of MolProbity and the *CCTBX* NBO provides Phenix with a comprehensive and complementary set of tools for efficient refinement and validation of models.

The NBO information is not currently available in the Phenix GUI but will be made available in the near future.

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 Crystallogr D* **66**, 213-221.

Afonine, P. V., Grosse-Kunstleve, R. W., Echols, N., Headd, J. J., Moriarty, N. W., Mustyakimov, M., Terwilliger, T. C., Urzhumtsev, A., Zwart, P. H. & Adams, P. D. (2012). *Acta crystallographica. Section D, Biological crystallography* **68**, 352-367.

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., Meyer, E. F., Jr., Brice, M. D., Rodgers, J. R., Kennard, O., Shimanouchi, T. & Tasumi, M. (1977). *J Mol Biol* **112**, 535-542.

Chen, V. B., Arendall, W. B., Headd, J. J., Keedy, D. A., Immormino, R. M., Kapral, G. J., Murray, L. W., Richardson, J. S. & Richardson, D. C. (2010). *Acta Crystallogr D* **66**, 12-21.

Davis, I. W., Murray, L. W., Richardson, J. S. & Richardson, D. C. (2004). *Nucleic Acids Research* **32**, W615-W619.

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

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

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

Word, J. M., Lovell, S. C., LaBean, T. H., Taylor, H. C., Zalis, M. E., Presley, B. K., Richardson, J. S. & Richardson, D. C. (1999). *Journal of Molecular Biology* **285**, 1711-1733.

Appendix

Code implementation example

```
import mmtbx.monomer_library.pdb_interpretation as pdb_inter
import cctbx.geometry_restraints.nonbonded_overlaps as nbo
from libtbx.utils import null_out
from libtbx.utils import Sorry
#
pdb_processed_file = pdb_inter.run(
   args=files,
   assume_hydrogens_all_missing=False,
   hard_minimum_nonbonded_distance=0.0,
   nonbonded_distance_threshold=None,
   substitute_non_crystallographic_unit_cell_if_necessary=False,
   log=null_out())
# test that CRYST1 records are ok
 sps = pdb_processed_file.all_chain_proxies.special_position_settings
 if not sps: raise Sorry('Bad CRYST1 records')
#
grm = pdb_processed_file.geometry_restraints_manager()
xrs = pdb_processed_file.xray_structure()
macro_mol_sel = nbo.get_macro_mol_sel(pdb_processed_file)
#
nb_overlaps = nbo.info(
   geometry_restraints_manager=grm,
   macro_molecule_selection=macro_mol_sel,
   sites_cart= xrs.sites_cart(),
   site_labels= xrs.scatterers().extract_labels(),
   hd_sel= xrs.hd_selection())
nb_overlaps.show()
```