Automated TLS group determination in Phenix

Pavel Afonine

Computation Crystallography Initiative
Physical Biosciences Division
Lawrence Berkeley National Laboratory, Berkeley CA, USA

October, 2010
Atomic Displacement Parameters (ADP or “B-factors”)

- ADPs model relatively small atomic motions (within harmonic approximation)
- Hierarchy and anisotropy of atomic displacements

Total ADP:

\[ U_{\text{TOTAL}} = U_{\text{CRYST}} + U_{\text{GROUP}} + U_{\text{LOCAL}} \]
Atomic Displacement Parameters (ADP or “B-factors”)

- **Total ADP** \[ U_{\text{TOTAL}} = U_{\text{CRYST}} + U_{\text{GROUP}} + U_{\text{LOCAL}} \]

- \( U_{\text{CRYST}} \) – overall anisotropic scale.
- \( U_{\text{TLS}} \) – rigid body displacements of molecules, domains, secondary structure elements.
- \( U_{\text{LOCAL}} \) – local vibration of individual atoms.
- \( U_{\text{LIB}} \) – librational motion of side chain around bond vector.
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 mutl-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.ensemble. Another new release integrates MR and Rosetta in PHENIX. For more details, see the short communications for phenix.ensemble on page 8 and phenix.mr_rosetta on page 10.

New features
Afonine et al (2010). On atomic displacement parameters and their parameterization in Phenix
• Using TLS in refinement requires partitioning a model into TLS groups. This is typically done by
  - visual model inspection and deciding which domains may be considered as rigid
  - using TLSMD method

TLS contribution of an individual atom participating in a TLS group can be computed from $T$, $L$ and $S$ matrices:

$$U_{TLS} = T + ALA^t + AS + S^tA^t$$

(20 TLS parameters per TLS group)
Methodology (TLSMD)

- Split a model into 1, 2, 3, …, N contiguous segments.
  - If \( n \) is the number of residues in the chain, \( m \) is minimum number of residues in a segment, then the number of segments is
    \[
    s(n,m) = \frac{n(n+1)}{2} - \sum_{i=2}^{m} (n+1-i)
    \]
  - For example if \( n=100 \) and \( m=3 \) (minimal number of residues so there is more “observations” than parameters), then we get 4853 possible partitions.

- Compute residual for all segment partitions:
  \[
  R = w \sum_{\text{atoms in segment}} W \left( \sum_{i=1}^{6} \left( U_{ATOM}^i - U_{TLS}^i \right)^2 \right)
  \]
Methodology
How to pick up the \textit{right} number of TLS groups?
How to pick up the right number of TLS groups?

One can try all 20 or choose an arbitrary break point.
PHENIX approach to finding TLS groups

• Design Goals:
  – Have it as integrated part of PHENIX system:
    – No need to run external software or use web servers (=send your data somewhere, which your policy may even not allow you to do).
    – Use it interactively as part of refinement (update TLS group assignment as model improves during refinement).
  – Make it fast
  – Eliminate subjective decisions (procedure should give THE UNIQUE answer and not an array of possible choices leaving the room for subjective decisions).
PHENIX approach to finding TLS groups

**Step 1:** For each chain find all secondary structure and unstructured elements

- *Number of elements defines maximum possible number of TLS groups*
- A secondary structure element can’t be split into multiple TLS groups. Large unstructured elements, can be split into smaller pieces.

Chain

\[
\begin{array}{cccccccc}
S & U & S & S & U & S & & U & S \\
\end{array}
\]

*S* – Secondary structure element

*U* – Unstructured stretch of residues (loop)

**Step 2:** Find all possible contiguous combinations

\[
\begin{array}{cccc}
S & U & S & \\
S & U & S & \\
S & U & S & \\
\end{array}
\]

\[N\text{ELEMENTS} : N\text{POSSIBLE PARTITIONS} = 3:3, 4:7, 5:15, 6:31, \ldots, 10:511, \ldots\]
**Step 3:** For each partition fit TLS groups and compute the residual

- \( R_1 \)
- \( R_2 \)
- \( R_3 \)

**Step 4:** Find the best fit among the groups of equal number of partitions. In this example, if \( R_3 < R_2 \):

- \( R_1 \)
- \( R_3 \)

**Step 5:** Find the best partition…

- Challenge: we can’t directly compare \( R_1 \) and \( R_3 \) because they are computed using different number of TLS groups (different number of parameters)
Step 5 (continued): Find the best partition…

- Randomly generate a pool of partitions for each candidate, fit TLS matrices compute, average residuals, and compute score:

\[ \text{Score} = \frac{(R_{1_{\text{average}}} - R_1)}{(R_{1_{\text{average}}} + R_1)} \times 100 \]

Do the same for the next candidate:

The final solution is the one that has the highest score.
### PHENIX approach to finding TLS groups: examples

#### GroEl structure (one chain):

<table>
<thead>
<tr>
<th>No. of groups</th>
<th>Targets best</th>
<th>Targets rand.pick</th>
<th>diff.</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>680.7</td>
<td>869.2</td>
<td>188.5</td>
<td>12.2</td>
</tr>
<tr>
<td>3</td>
<td>297.1</td>
<td>665.7</td>
<td>368.6</td>
<td>38.3</td>
</tr>
<tr>
<td>4</td>
<td>260.4</td>
<td>448.3</td>
<td>187.8</td>
<td>26.5</td>
</tr>
<tr>
<td>5</td>
<td>206.2</td>
<td>342.1</td>
<td>135.8</td>
<td>24.8</td>
</tr>
<tr>
<td>6</td>
<td>188.4</td>
<td>264.7</td>
<td>76.3</td>
<td>16.8</td>
</tr>
<tr>
<td>7</td>
<td>182.3</td>
<td>251.2</td>
<td>68.9</td>
<td>15.9</td>
</tr>
<tr>
<td>8</td>
<td>176.9</td>
<td>229.5</td>
<td>52.5</td>
<td>12.9</td>
</tr>
<tr>
<td>9</td>
<td>173.1</td>
<td>207.3</td>
<td>34.1</td>
<td>9.0</td>
</tr>
<tr>
<td>10</td>
<td>170.2</td>
<td>196.8</td>
<td>26.6</td>
<td>7.2</td>
</tr>
<tr>
<td>11</td>
<td>167.8</td>
<td>183.1</td>
<td>15.2</td>
<td>4.3</td>
</tr>
<tr>
<td>12</td>
<td>165.6</td>
<td>179.0</td>
<td>13.4</td>
<td>3.9</td>
</tr>
<tr>
<td>13</td>
<td>163.9</td>
<td>170.8</td>
<td>6.9</td>
<td>2.1</td>
</tr>
</tbody>
</table>
PHENIX approach to finding TLS groups: usage

Command line:

```
phenix.find_tls_groups model.pdb nproc=N
```

where nproc is the number of available CPUs
TLS groups for refinement automatically

TLS refinement groups:

TLS groups:
- chain 'A' and (resseq 295:394)
- chain 'A' and (resseq 395:434)
- chain 'A' and (resseq 435:569)

Edit selected: chain 'A' and (resseq 295:394)

(*) http://skuld.bmsc.washington.edu/~tlsmd

Number of processors for phenix.find_tls_groups: 2

Object: phenix_001.pdb  Selection: chain 'A' and (resseq 295:394)

Mouse: Rotate view

827 atoms selected
Examples:

- GroEL structure (3668 residues, 26957 atoms, 7 chains):
  
  PHENIX: 135 seconds
  TLSMD: 3630 seconds
  (plus lots of clicking to upload/download the files and making arbitrary decisions)

- Lysozime structure:
  9.5 seconds with one CPU
  2.5 seconds using 10 CPUs
Automatic TLS

Why it is faster:

a) Use isotropic TLS model,

b) Solve optimization problem analytically (no minimizer used)

c) A secondary structure element cannot belong to more than one TLS group
ADP refinement: what goes into PDB

*phenix.refine outputs TOTAL B-factor (iso- and anisotropic):*

\[ U_{TOTAL} = U_{ATOM} + U_{TLS} + \ldots \]

Isotropic equivalent

Atom records are self-consistent:

- Straightforward visualization (color by B-factors, or anisotropic ellipsoids)
- Straightforward computation of other statistics (R-factors, etc.) – no need to use external helper programs for any conversions.
ADP refinement: example

Synaptotagmin refinement at 3.2 Å

Original refinement (PDB code: 1DQV)
$R$-free = 34 %
$R$ = 29 %

PHENIX – Isotropic restrained ADP
$R$-free = 28 %
$R$ = 23 %

PHENIX – TLS + Isotropic ADP
$R$-free = 25 %
$R$ = 20 %

9% improvement in both $R_{work}$ and $R_{free}$!

TLS groups determined automatically…