LatFit - Manual

Martin Mann - University Freiburg

http://www.bioinf.uni-freiburg.de/

June 24, 2008

LatPack Tools Package
Version 1.5.0

1 Description

LatFit allows the conversion of a protein's full atom structure representation in Protein Data Bank (PDB) format into a coarse grained lattice model representation. This is done by

1. fitting the C_α-atoms to neighbored lattice positions for representation of the protein backbone.
2. If side chains are modelled, C_β or the center of mass of the residues are fitted to neighbored positions of the corresponding C_α atoms.

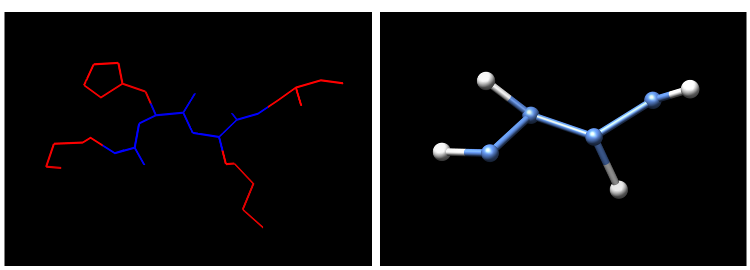

Figure 1: A full atom representation of amino acids and the corresponding side chain lattice model representation. The backbone is given in blue.

LatFit implements a common heuristic method that was successfully applied in literature [1, 2, 3]. It does not ensure to find the optimal fit onto the lattice but a reasonable good one. For instance, using the Face Centered Cubic (FCC) lattice an approximation of the backbone with a dRMSD of 1.7 Angstroms is achieved. dRMSD denotes the distance root mean square deviation of the corresponding C_α atoms.

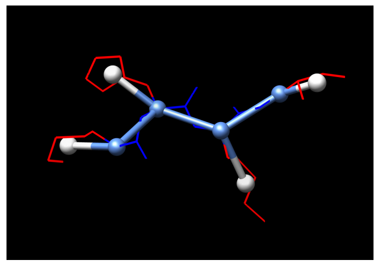

Figure 2: Mapping of the full atom and lattice model structure from Fig. 1.

2 Method

2.1 Fitting

Given:

$P = P_1, \dots, P_n$	:	3D coordinates of the original atoms to approximate
N	:	the neighboring vectors of the lattice model to use
r^X, r^Y, r^Z	:	rotation of the lattice according to X,Y, and Z-axis
k	:	number of best substructures to store per iteration

Result:

$L = L_1, \dots, L_n$	:	3D coordinates of the best fit onto the lattice

Method:

The approximation follows a greedy structure-elongating approach:

```
N' ← N rotated by the angles r^X, r^Y, r^Z                      ▷ lattice rotation
B ← {k best fits of P_1}                ▷ best structure fits of last iteration
                           ▷ initialized with the k best fits of first monomer
C ← ∅                                  ▷ structures generated in current iteration
for i = 2 ... n do
    for all L ∈ B do                                       ▷ L has length (i − 1)
        for all v⃗ ∈ N' do
            if L_(i−1) + v⃗ ∉ L_1, ..., L_(i−1) then                 ▷ selfavoidingness
                C ← C ∪ {(L_1, ..., L_(i−1), L_(i−1) + v⃗)}         ▷ store elongation
            end if
        end for
    end for
    B ← best k fits of C according to RMSD to P_1, ..., P_i
```

13: $C \leftarrow \emptyset$ ▷ reset structure storage
14: **end for**
15: report best fit $L \in B$ according to RMSD to P

Note: Due to the greedy storing of the k best structures only, it may occur that none of the k best of the last iteration can be extended in a selfavoiding way (Line 6 gives 'false'). Therefore, B would get empty and the approximation stops without finding a selfavoiding fit of the whole structure P. This problem can usually be solved by increasing k but to the cost of additional computations and runtime.

2.2 Lattice Rotation

To find the best lattice approximation of a full atom protein structure P not only one lattice orientation has to be considered. Different rotations of the lattice along the X, Y, and Z-axis have to be generated during the search for an optimal fit. For each rotation tuple (r^X, r^Y, r^Z), the best possible fit can be obtained using the method introduced in Sec. 2.1.

A systematic search can be done that divides evenly a given rotation range $[0, m \cdot \pi]$ into s values for each of the rotation angles r^X, r^Y, and r^Z, with $m > 0$ as a user defined maximal rotation factor. Afterwards, all s^3 different rotation combinations are used to find the best structure approximation. This yields the best fit L of structure P onto a lattice with the corresponding best rotation angles r_b^X, r_b^Y, and r_b^Z.

The symmetry of the lattice gives directly a maximal rotation factor m necessary. For instance in the cubic lattice, a rotation of 90° is symmetric according to all axes. Therefore, m can be limited to 0.5 to avoid unnecessary calls of the fitting procedure for symmetric rotation angles. The same hold for the cubic and face centered cubic lattice.

2.3 Refinement

The best structure found via systematic search is usually not the best possible due to the discretized rotation steps and the large size of the interval searched. Here, a refinement of this structure can help.

Therefore, a small interval around each of the so far best rotation angles $r_b^i \in \{r_b^X, r_b^Y, r_b^Z\}$ is defined. For a user given refinement rotation factor $m_r > 0$ the intervals are $[r_b^i - (m_r \cdot \pi), r_b^i + (m_r \cdot \pi)]$. Once again, a systematic search is performed by an even division of the interval in s_r values. This results in additional s_r^3 calls of the fitting procedure.

3 Available Lattices

Several lattice models can be used to fit a structure onto. For side chain models, a combination of two different lattices can be used (see parameter **-scLat**).

The currently supported lattice models and the corresponding neighboring vectors are:

ID	Name	Neighborhood vectors	#
SQR	Square	$\{\pm(1,0,0), \pm(0,1,0)\}$	4
CUB	Cubic	$\{\pm(1,0,0), \pm(0,1,0), \pm(0,0,1)\}$	6
FCC	Face Centered Cubic	$\left\{ \begin{smallmatrix} \pm(1,1,0), \pm(1,0,1), \pm(0,1,1), \\ \pm(1,-1,0), \pm(1,0,-1), \pm(0,1,-1) \end{smallmatrix} \right\}$	12

4 Program Parameters

PDB Input

-pdbFile The full atom protein representation to fit onto a lattice in Protein Data Bank (PDB) format. If this parameter is not given or set to 'STDIN', the structure is read from the standard input.
Note: **-fitSideChain** is used, reading from 'STDIN' is not possible due to a necessary double parsing of the PDB input.

-pdbAtom The atom identifier that has to be fitted as the backbone monomer of the lattice structure. Usually, 'CA' for C_α-atoms is used. If 'CoM' is given, the center of mass of the amino acid side chain is fitted. If **-fitSideChain** is used, the given atom is fitted as the side chain monomer of the structure and C_α for the backbone.

-pdbAtomAlt If the PDB file contains alternatives for atoms to fit, an alternative identifier has to be given to allow a unique fit.

-pdbChain In case the PDB file contains several amino acid chains, one chain to process has to be specified.

-pdbChainGaps Some PDB files show imcomplete atom position information such that no information is given for some amino acids. Such *gaps* in the sequence are usually rejected by the program that tries to fit an entire chain. If such a non-consecutive chain with gaps should be fitted instead, the parameter **-pdbChainGaps** has to be given.

Lattice Settings

-lat The lattice onto that the backbone has to be fitted. The available list of lattice identifiers is given in Sec. 3.

-bondLength The distance in Angstroms between two neighbored C_α-atoms in the lattice. Used to scale all neighboring vectors of the lattice (Sec. 3) to this length. A common value used in literature is '3.8'.

Side Chain Settings

-fitSideChain If present, a fit of two monomers per amino acid is done. One for backbone and one representing the side chain. The C_α-atom ('CA') is used to fit the backbone monomer. The atom specified with **-pdbAtom** is used for the side chain monomer fit.
Note: The fit of a Glycine amino acid includes a side chain monomer as well, even it has none in real proteins! Here, both monomers (backbone and side chain) approximate the C_α-atom position.

-scLat Per default the same neighboring vectors as for the backbone fit (**-lat**) are used for the neighboring of backbone (C_α) and side chain monomers. Using **-scLat**, a different set of allowed neighboring vectors (=lattice) can be specified. The length of these vectors are calculated in relation to the backbone vector lengths (see **-bondLength**). The available list of lattice identifiers is given in Sec. 3.

-scContrib Allows for a weight of the side chain fit according to the backbone approximation. This factor is multiplied to each side chain monomer RMSD that is added to the overall RMSD. Therefore, higher the value yield a better fit of the side chain monomer compared to the C_α atom.
Note: a value $\neq 1.0$ makes the reported RMSD meaningless due to the scaling of the side chain distances!

-fitDirVec If present, a fit of direction vectors instead of side chain atoms is performed. A direction vector is given by $\vec{d} = k \cdot (\textbf{pdbAtom} - C_\alpha)$, whereby k is a calculated scaling factor to set the length of $\vec{d}$ to **dirVecLength**.

-dirVecLength The length of the direction vector to fit, if **-fitDirVec** is specified.

Fitting Parameters (see Method Sec. 2)

-rotMax Factor that limits the maximal rotation angles in radian measure. The rotations are done within $[0.0, \textbf{rotMax} \cdot \pi]$ for each dimension X,Y, and Z.

-rotSteps Number of discrete rotation steps done to find a good fit. Therefore, the interval $[0.0, \textbf{rotMax} \cdot \pi]$ is divided into **rotSteps** equal intervals.

-nKeep Number of best structures to store that are extended in the next iteration.

-refRotSteps Determines if a refinement of the best structure of the "global" screening should be done or not. If set to 0 no refinement is done. Otherwise, the best approximation should be improved. This is done via a fine grained rotation around the best angles r^X, r^Y, r^Z so far. The rotation is done in the intervals $[r^i - \Delta r, r^i + \Delta r]$ around each rotation angle with $i \in \{X, Y, Z\}$ and $\Delta r = \mathbf{refRotMax} \cdot \pi$.

-refRotMax Factor that defines the rotation intervals around the best rotation angles so far in which a refinement should be done (see **-refRotSteps**).

Output

-outMode The format in that the output should be written. Possible formats are:

ID	Format Description
CML	Chemical Markup Language (XML)
PDB	Protein Data Base format
XYZ	Coordinate output (plain text)

-outFile If not specified or set to 'STDOUT' the final structure output is done to standard output. Otherwise it is written to the specified file.

-outAllBest Every time a better fit than the last best found is achieved the corresponding structure is printed.

-outLatPoint Prints the non-rotated lattice structure with discrete lattice positions instead of the rotated.
Note: Also the neighbor vector scaling via **-bondLength** is ignored!

-outOrigData If present, adding the atom positions of the original protein structure (**-pdbFile**) to the output.

Miscellaneous

-v Give verbose output during computation.

-s Give only minimal output during computation.

-help Prints the available program parameters.

5 Contact

Martin Mann
Bioinformatics Group
University Freiburg, Germany

http://www.bioinf.uni-freiburg.de/

References

[1] B. H. Park and M. Levitt: **The Complexity and Accuracy of Discrete State Models of Protein Structure**, *Journal of Molecular Biology* 1995, **294**:493-507

[2] J. Miao, J. Klein-Seetharaman and H. Meirovitch: **The Optimal Fraction of Hydrophobic Residues Required to Ensure Protein Collapse**, *Journal of Molecular Biology* 2003, **344**:797-811

[3] A. Godzik, A. Kolinski and J. Skolnick: **Lattice Representations of Globular Proteins: How Good Are They?**, *Journal of Computational Chemistry* 1993, **14(10)**:1194-1202