LocalMove: computing on-lattice fits for biopolymers

Y. Ponty, R. Istrate, E. Porcelli and P. Clote*

Department of Biology, Boston College, Chestnut Hill, MA 02467, USA

ABSTRACT

Given an input Protein Data Bank file (PDB) for a protein or RNA molecule, `LocalMove` is a web server that determines an on-lattice representation for the input biomolecule. The web server implements a Markov Chain Monte-Carlo algorithm with simulated annealing to compute an approximate fit for either the coarse-grain model or backbone model on either the cubic or face-centered cubic lattice. `LocalMove` returns a PDB file as output, as well as dynamic movie of 3D images of intermediate conformations during the computation. The `LocalMove` server is publicly available at http://bioinformatics.bc.edu/clotelab/localmove/.

INTRODUCTION

Predicting the structure of biopolymers is one of the most important and well-studied computational problems of the 20th century—a problem, that despite enormous advances, remains only partially solved. In an effort to minimize the number of conformations to be explored, coarse-grain lattice models (beads on a string) have been studied by many authors (1–4), while coarse-grain off-lattice models have been used in discrete molecular dynamics (5). In this article, we present the `LocalMove` web server, which implements a Markov Chain Monte-Carlo (MCMC) algorithm to compute an approximate cubic or face-centered cubic lattice fit of either the coarse-grain or backbone model for an input Protein Data Bank (PDB) (6) file for a protein or RNA molecule.

Finding a self-avoiding walk on the cubic lattice that minimizes the coordinate root mean square deviation (given sequences $p_1, \ldots, p_n$ and $q_1, \ldots, q_n$ of 3D points, the coordinate root mean square deviation, denoted **rms** or cRMS, is $\sqrt{\sum_{i=1}^{n}(p_i - q_i)^2/n}$ with the original PDB file, after normalization to ensure unit distance between successive monomers, is known to be NP-complete (7). Thus various heuristic approaches (8–13) have been proposed to approximately solve this problem, including Hopfield nets,

self-consistent field optimization, integer programming (the application of integer programming (13) provides an optimal, not just approximate, solution, however with exponential run time), etc. Unfortunately, none of these methods is publicly available, so that `LocalMove` is the only publicly available tool for on-lattice fit of biopolymers, allowing users to postprocess certain threading energies (aka knowledge-based potentials) for structure classification and prediction.

The method `LocalMove`, presented in this article, performs a *Monte-Carlo* exploration of the on-lattice conformational landscape through a sequence of *local moves*, which generalize the single-monomer *end* and *corner moves*, and the 2-monomer *crankshaft* moves used in ref. (14) for the cubic lattice. At each step, a measure of similarity, distance root mean square deviation [dRMS (distance root mean square deviation (dRMS) between two conformations $P = p_1, \ldots, p_n$ and $Q' = q_1, \ldots, q_n$ is defined by

$$\text{dRMS}(P, Q) = \sqrt{\frac{\sum_{1 \leq i < j \leq n}(d_{i,j} - e_{i,j})^2}{\binom{n}{2}}},$$

where $D(P) = (d_{i,j})$ and $D(Q) = (e_{i,j})$ are the corresponding distance matrices, where $d_{i,j} = ||p_i - p_j||$ and $e_{i,j} = ||q_i - q_j||)$] is evaluated and the candidate move is either accepted or stochastically rejected, according to the Metropolis criterion. Different levels of representation are supported by `LocalMove`, scaling from the coarse-grain monomer model (C_α for amino acids, $C_{1'}$ for RNA nucleotides or alternatively nucleotide centers of mass), to all backbone atoms. `LocalMove` supports the cubic and face-centered cubic (FCC) lattices. Various termination conditions can be defined for the walk.

There appears to be little data on the quality, in terms of coordinate root mean square deviation, cRMS, of on-lattice fits, an exception being the data of Reva *et al.* (15) for approximate *cubic lattice* fits of C_α-atom traces of proteins from a small representative sample. See Tables 1 and 2 for a comparison of `LocalMove` with the method of Reva *et al.* (15,16) on the only published data of on-lattice fits that we could find.

*To whom correspondence should be addressed. Tel: +1 617 552 1332; Fax: +1 617 552 2011; Email: clote@bc.edu

MATERIALS AND METHODS

`LocalMove` addresses the problem of finding the best on-lattice fit for the coarse-grain model or backbone model for proteins and RNA, with a number of parameter choices for the user. Lattice type can be either the cubic or FCC lattice, described later.

`LocalMove` applies the Monte-Carlo algorithm (17,18), where *energy* is defined as follows. Given a conformation $P = p_1, \ldots, p_n$, where each $p_i \in \mathbf{R}^3$, define the *distance matrix* $D(P) = (d_{i,j})$, where $d_{i,j}$ is the Euclidean distance between p_i and p_j. Define the dRMS between two conformations $P = p_1, \ldots, p_n$ and $Q' = q_1, \ldots, q_n$ by

$$\text{dRMS}\,(P, Q) = \sqrt{\sum_{1 \leq i < j \leq n} (d_{i,j} - e_{i,j})^2 \Big/ \binom{n}{2}}$$

where $D(P) = (d_{i,j})$ and $D(Q) = (e_{i,j})$ are the corresponding distance matrices. To determine approximate on-lattice fit, define the energy $E(C)$ of a given lattice conformation by dRMS(C, P_0), where P_0 is the normalized conformation of monomers C_α or $C_{1'}$ in the coarse-grain model, or backbone atoms, as depicted in Figure 1 in the backbone model. The off-lattice conformation P_0 is normalized so that distance between successive atoms is 1.

In `LocalMove`, if C' denotes the temporary conformation obtained by replacing a k-monomer segment in the current conformation C, then C' becomes the next configuration, provided that C' is a self-avoiding walk and either $E(C') \leq E(C)$ or a random real z is less than $e^{-(E(C')-E(C))/RT}$, i.e. the Metropolis criterion holds. Details and parameter choices for the user are suggested below. Algorithmic details, computational experiments for various parameters and extensive benchmarking will appear in a companion methods paper in preparation.

Models

Backbone representation. For protein, on-lattice models have historically considered the coarse-grain representation where each residue is represented by a single point, yielding the C_α-trace. For proteins, this level of granularity

seems reasonable, since the average distance between consecutive C_α carbons in proteins extracted from the Nucleic Acid Database (NDB) (19) yields an average of 3.8 Å with a low SD of 0.04 Å. In the case of RNA, a coarse-grain model is less able to capture the essence of an RNA conformation, since the average distance between successive $C_{4'}$ atoms is 6.1 Å with a SD of 0.46 Å. In the case of RNA, the backbone model thus appears to be a better representative of the conformation than is the coarse-grain model.

While it is beyond the scope of the current article to answer the question of choosing the best representation of biopolymers backbone for general on-lattice applications, we tried to offer the user the choice of a suitable representation. Namely, our algorithm extracts a subset of the atoms in the model/chain of interest, and performs its search for the best fit of this selection. The different levels of representation currently supported by `LocalMove` are:

	Proteins	RNA
Full backbone	N-C_α-C	P-$O_{5'}$-$C_{5'}$-$C_{4'}$-$C_{3'}$-$O_{3'}$
Coarse-grain	C_α or μ	$C_{1'}$, N, P or μ

where in the RNA coarse-grain model, the user can select among the carbon $C_{1'}$- or nitrogen N-atom, both adjacent to the glycosidic bond, the backbone phosphorus or the center of mass of the nucleotide, denoted by μ.

Lattices. `LocalMove` supports the cubic and FCC lattice. The latter, well-known to crystallographers as one of the *Bravais* lattices, has contact number 12, meaning that each lattice point has 12 immediate neighbors; see Figure 2. Covell and Jernigan have shown that the FCC lattice is the most appropriate 3D lattice for fitting protein C_α-atoms as a self-avoiding walk; i.e. cRMS values are smaller for the FCC than for the cubic, body-centered cubic and tetrahedral lattices.

Algorithm

Simulated annealing. `LocalMove` implements the MCMC algorithm, as well as simulated annealing,
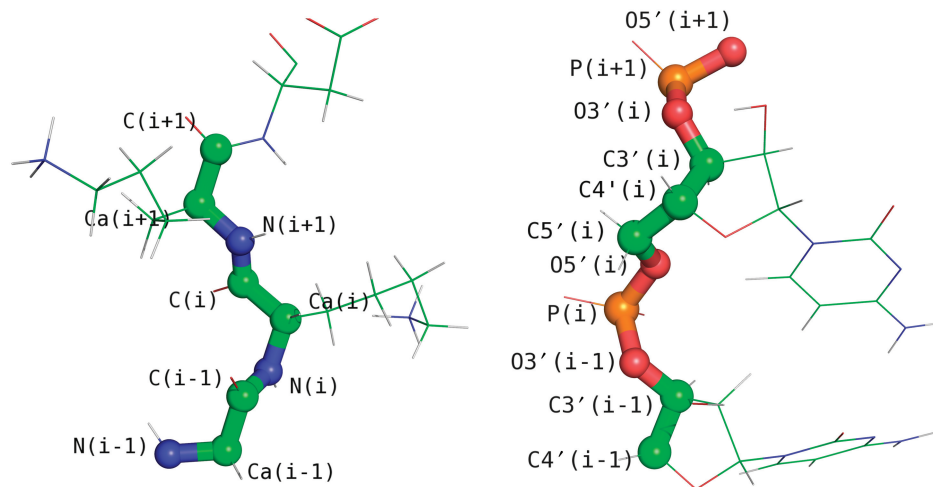

Figure 1. Backbones of protein (left) and RNA (right). Note that residue resp. nucleotide positions increase from the bottom of the figure towards the top.

and the user can set an initial temperature, terminal threshold temperature and temperature scaling factor c (i.e. temperature is periodically decreased by $T = c \cdot T$). Alternatively, a greedy descent (no Metropolis step) and a Fixed Metropolis probability strategy are implemented.

Three strategies are implemented in `LocalMove` to choose an initial self-avoiding configuration: Random, a random 3D self-avoiding walk is generated; Straight line; Rounded (greedy). By rounding, we mean a greedy, iterative procedure to place the next monomer (or atom) of a growing chain on the closest lattice point to the previous monomer (or atom), while guaranteeing a self-avoiding walk. If this strategy does not produce a self-avoiding walk, which sometimes happens, then `LocalMove` chooses a random self-avoiding walk as the initial on-lattice conformation.

`LocalMove` performs *local k-monomer moves*, generalizing the move set of Šali *et al.* (14). Given a current self-avoiding walk $p_1, \ldots, p_n$, `LocalMove` randomly chooses positions i, j, and replaces the intermediate k-monomer walk $p_{i+1}, \ldots, p_{j-1}$, where $k = j - i - 1$, by a different k-monomer walk $p'_{i+1}, \ldots, p'_{j-1}$ having the same vector difference. Three types of strategies are proposed regarding *self-avoidance*: strict, where the self-avoidance of the resulting walk is tested in linear time and the move is rejected if the test is failed; local, where only a subset of points adjacent to the insertion point are tested; none, where self-avoidance is not enforced, depending on the option. The relevant parameters handled by `LocalMove` for such moves are the local move size, the self-avoidance strategy and the strategy for picking a new local move at random.

`LocalMove` simulations can be stopped for some of the different following reasons: either a limit temperature is bypassed during the simulated annealing; a distance threshold is reached; the maximal number of steps have been performed or the simulation is *stalled* for too long, leaving few hope for improvement. In the latter, the required improvement over a user-defined period of time can be either relative or absolute.

Additional features

In addition to the features described earlier, our webserver gives its user the possibility to follow in realtime the lattice fitting process. After the beginning of the lattice fitting process, the user's browser is redirected to a webpage featuring an experiment player based on the popular JMol. Additionally, an email is sent to the user, featuring an unique identifier for the ongoing experiment. By entering this identifier at any time during or after completion of the experiment, the user can access its results or follow its progress. Results are kept until about 1 week after the end of the experiment, and are then deleted. Even if such is the case, the user is proposed to repeat the experiment, using the same parameters or is allowed to modify them in a prefilled version of the webserver form. This allows for a quick and easy modification of an already run experiment.

Finally, `movies` can be generated automatically after the lattice fitting process is over. To that purpose, snapshots of the molecule are rendered using `PyMol` each 500 steps of the Monte-Carlo algorithm, and assembled using `FFMpeg`.

RESULTS

Preliminary results are given in Tables 1 and 2, to compare `LocalMove` (greedy strategy, rounded initial conformation, self-avoiding walk test for intermediate conformations) with the method of Reva *et al.* (15) (optimal parameters $A = 10$, $T \approx 0.1$ – see p. 7 of ref. (15)). Although the method of Reva *et al.* is clearly superior for cubic lattice fits, it is not publicly available. In contrast, `LocalMove` provides acceptable approximate

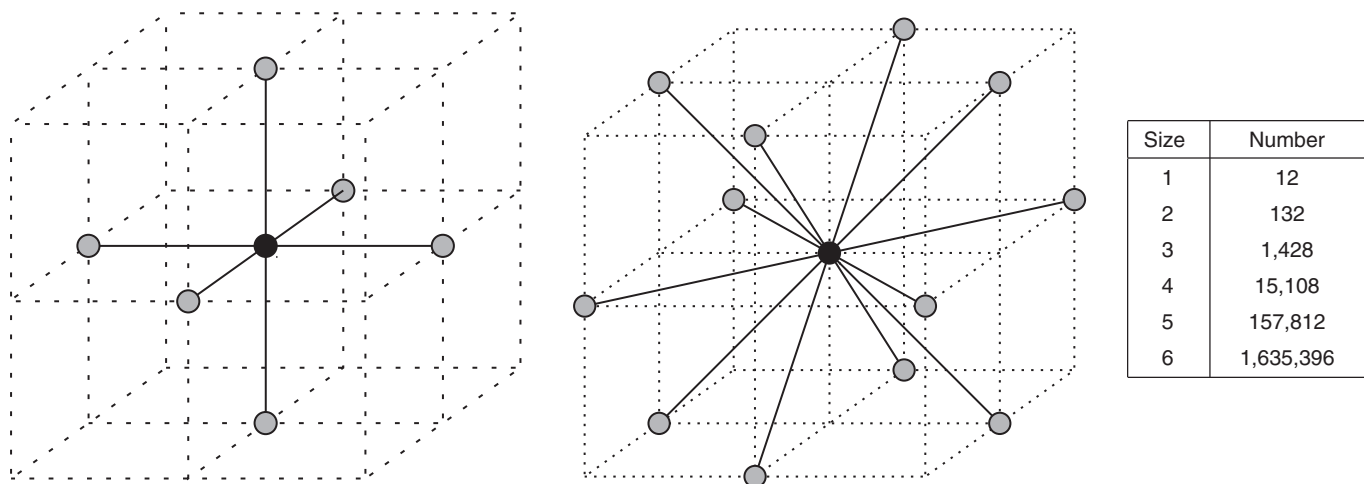| Size	Number
1	12
2	132
3	1,428
4	15,108
5	157,812
6	1,635,396

Figure 2. Neighbors of a point under various lattice models: (left) 3D cubic lattice, (middle) 3D FCC, (right) numbers of self-avoiding walks of various sizes on FCC. The FCC lattice can be represented as the set of all integral coordinates (x, y, z), such that $(x + y + z) \bmod 2 = 0$. If $p = (x, y, z)$ and $q = (a, b, c)$, then p, q are immediate neighbors if $|x - a| + |y - b| + |z - c| = 0 \bmod 2$, and $|x - a|, |y - b|, |z - c| \leq 1$. Note that immediate neighbors on the FCC lattice are at Euclidean distance $\sqrt{2}$ from each other, hence comparisons with PDB data are made after normalization that ensures unit distance between successive monomers.

lattice representations for cubic and FCC lattices, for various coarse grain and backbone models of both protein and RNA.

Table 1 and 2 respectively list the best scores and average scores for cubic lattice fits of 17 protein chains of various sizes. Scores for the method of Reva *et al.* (15) are values of RMS in lattice units, while those of `LocalMove` are values of cRMS in lattice units—i.e. PDB files are scaled to have distance 1.0 between successive monomers (or atoms)

when superimposing structures. RMS, as measured in ref. (15), is approximately the same as cRMS; however, there is a technical difference, explained as follows. In Reva's method, a cubic orthonormal lattice is projected onto the C_α-trace of a protein, self-consistent field is approximated, followed by dynamic programming. It is unclear from ref. (15), whether the (stochastic) cubic orthonormal lattice is defined from any three randomly chosen orthogonal basis vectors emanating from origin $(0, 0, 0)$,

Table 1. Comparison of best scores out of 100 runs. Scores are RMS for the optimized method of Reva *et al.* (15) ($A = 10$, $T \approx 0.1$, shells 1, 2), while remaining scores are cRMS using various strategies `LocalMove` (See text for distinction between RMS and cRMS.)

pdbID	Size	Reva	Greedy RMSD 4	greedy RMS 4	Fixed RMSD 4	Anneal RMSD 4
1epg	53	0.573	0.731	0.795	0.819 229	0.63 175
2ovo	56	0.612	0.634	0.895	0.901 008	0.556 876
1acb:I	63	0.630	0.751	1.059	1.034 372	0.620 289
2ctx	71	0.666	0.842	1.164	1.065 386	0.723 368
1fkf	107	0.658	0.658	0.767	1.15 722	0.686 337
3sic	107	0.654	0.720	0.877	1.169 239	0.655 475
1cdp	108	0.653	0.644	0.798	0.761 084	0.757 177
2trx	108	0.678	0.682	0.919	0.87 416	0.868 124
1hmd	113	0.628	0.908	1.494	0.797 958	0.79 189
1ppa	121	0.670	0.817	1.747	0.924 893	0.917 508
1rat	124	0.698	0.955	1.186	1.493 441	1.26 029
2aza	129	0.703	0.740	0.869	1.743 398	0.699 028
1ifb	131	0.736	0.801	0.980	1.185 644	0.733 953
1myg	153	0.683	1.273	1.347	0.871 547	0.870 891
2fcr	173	0.693	0.711	1.036	0.972 334	0.975 169
1fdl	218	0.714	0.791	1.169	1.348 602	0.891 129
7tim:A	247	0.718	0.886	1.108	1.110 148	1.106 702
RMSDc	122	0.669	0.797	1.071	1.072	0.809
Time	–	–	76.240	66.830	70.39	128.18

Four strategies of `LocalMove` are displayed, in order from left to right: greedy method to minimize dRMS, greedy method to minimize RMS, Monte Carlo with fixed probability of 20% in Metropolis step to minimize dRMS and Monte Carlo with simulated annealing to minimize dRMS. For each strategy of `LocalMove`, the maximum number of monomers moved is 4, and the intitial self-avoiding walk is determined by rounding if possible. In the simulated annealing, initial temperature $T = 10$, stopping temperature $T = 0.1$, temperature scaling factor $c = 0.95$, (artificial) Boltzmann constant $k = 4.699 \times 10^{-5}$. Reva *et al.* (15) study the effect of parameters A, T and number of shells on the accuracy and time of their method. Accuracy in this table is given for $A = 10$, $T \approx 0.1$ taking first and second shells, for which Reva *et al.* report a run time of ~30 s. Average `LocalMove` run time in seconds for each of the four strategies is respectively 76.24, 66.83, 70.39 and 128.18. (Shorter run times with less accuracy found when minimizing RMSDc instead of RMSDd, and when maximum number of monomers moved is 3, rather than 4.)

Table 2. Comparison of average scores out of 100 runs, for method of Reva *et al.* (15) and the four strategies of `LocalMove`, as explained in Table 1

pdbID size Reva	Greedy RMSD 4	Greedy RMS 4	Anneal RMSD 4	Fixed RMSD 4
1epg 53 0.682	1.435	1.358	0.807	0.873
2ovo 56 0.691	0.713	0.824	0.675	0.942
1acb:I 63 0.707	0.767	0.951	0.744	1.071
2ctx 71 0.762	0.798	0.897	0.960	1.094
1fkf 107 0.784	0.852	0.996	0.807	1.195
3sic 107 0.757	0.761	0.900	0.768	1.193
1cdp 108 0.699	0.953	1.086	0.770	0.807
2trx 108 0.744	0.694	0.801	0.895	0.899
1hmd 113 0.709	0.946	1.117	0.799	0.832
1ppa 121 0.722	0.855	0.856	0.955	0.960
1rat 124 0.773	0.986	1.192	1.587	1.507
2aza 129 0.789	1.012	1.771	0.846	1.767
1ifb 131 0.802	1.108	1.203	0.824	1.200
1myg 153 0.724	0.772	0.962	0.874	0.891
2fcr 173 0.749	1.160	1.516	0.981	0.995
1fdl 214 0.863	0.921	1.082	1.087	1.358
7tim:A 247 0.761	0.954	1.193	1.111	1.117
Average 122 0.748	0.923	1.100	0.911	1.100

or whether the origin is randomly chosen as well. In contrast, given the C_α traces $p_1, \ldots, p_n$ and $q_1, \ldots, q_n$, BioPython computes cRMS by superimposing the centers of mass, then computing optimal rotation matrix to return the C_α-trace $r_1, \ldots, r_n$ obtained by $q_1, \ldots, q_n$ by the computed translation and rotation. The value of cRMS is then

$$\frac{\sqrt{\sum_{i=1}^{n} ||p_i - r_i||^2}}{n}.$$

To illustrate the stability of our approach, we ran `LocalMove` on all RNA models/chains found in the NDB (Figure 3). Namely, we fit the backbone atoms $O_{5'}, P, O_{3'}, C_{5'}, C_{4'}, C_{3'}$ on the FCC lattice. We rescaled the resulting models and superimposed them with the original NDB backbone data, normalized so that adjacent atoms were at distance $\sqrt{2}$, the distance between adjacent lattice points in the FCC lattice. Superimposition was performed using Biopython http://biopython.org/. After removal of 17 spurious values, the cRMS values obtained when superimposing the 1735 on-lattice RNA models/chains on (normalized) backbone data from the NDB, we obtain mean cRMS is 0.554169 with SD of 0.145392. Similarly, we obtained `LocalMove` fits of backbone atoms (N, C_α, C) of monochain proteins from PDBselect25](20), a nonredundant protein database, where pairwise sequence identity is at most 25%. When on-lattice fits were superimposed on original (normalized) backbone off-lattice data from PDBselect25, the cRMS had mean of 0.612181 and SD of 0.161009.

For these experiments with both NDB and PDBselect25, `LocalMove` was run for one million steps, using the greedy (Monte Carlo with zero probability for Metropolis moves) strategy with at most three-monomer moves. In this case, the greedy strategy attempts to minimize dRMS; i.e. `LocalMove` accepts a randomly proposed k-monomer move, for $k \leq 3$, provided that the dRMS score of the proposed move is lower. The initial on-lattice structure determined by *rounding*. We allow early termination when

relative improvement is <0.01%; i.e. after every 15 000 steps, if the relative difference between best score and that of an ancestor 15 000 steps prior to current step is less than 0.0001, (recall that score means dRMS) then computation terminates. Technically, this means that we compute whether $s_0 - s_1/s_0 < 0.0001$, where s_0 denotes the ancestor score 15 000 steps before and s_1 denotes the current move.

DISCUSSION

In this article, we present a new web server, `LocalMove`, capable of determining approximate on-lattice fits of protein and RNA 3D conformations on the cubic and the FCC lattice (Figure 4). `LocalMove` returns the PDB file of the approximate on-lattice fit, and interactively displays a dynamic movie of 3D images of intermediate conformations during the computation. In Tables 1 and 2, we benchmark `LocalMove` against what appears to be the only publicly available data set for previous on-lattice fits. To the best of our knowledge, no other method is publicly available to compute on-lattice fits of protein and RNA molecules. Reva's method and most of the earlier methods handle only the cubic lattice, known not to be optimal for biopolymer folding, while `LocalMove` handles cubic and FCC lattices with a variety of coarse grain and backbone models for both protein and RNA.

We believe that the new server, `LocalMove` (Figure 5), as well as our previous 3D RNA motif detection server, `DIAL`, described in Ferrè *et al.* (21), will contribute to better detection and classification of RNA motifs, essential ultimately for predicting tertiary structure, catalytic sites and function of RNA.
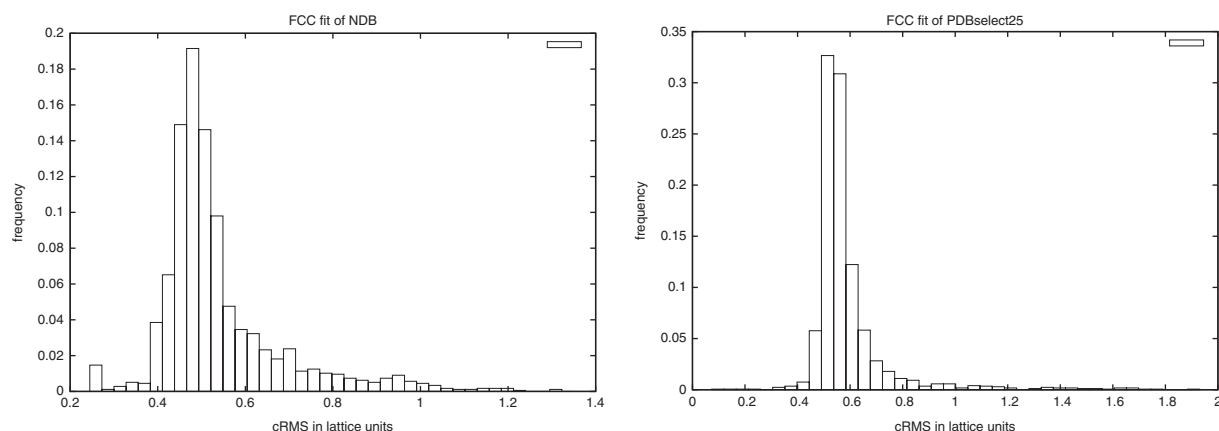
ACKNOWLEDGEMENTS

Figure 3. *(Left)* Distribution of cRMS for `LocalMove` best on-lattice fits for the *backbones* ($O_{5'}, P, O_{3'}, C_{5'}, C_{4'}, C_{3'}$) of 1735 RNA models/chains from the NDB, superimposed with the (normalized) NDB files. Statistics for cRMS: mean is 0.554169, SD is 0.145392, both measured in lattice units. *(Right)* Distribution of cRMS for `LocalMove` best on-lattice fits for the *backbone* (N, C_α, C) of 1733 (monochain) proteins from PDBselect25 (20), a nonredundant protein database (pairwise, proteins have at most 25% sequence identity), superimposed with the (normalized) original PDB files. Statistics for cRMS: mean is 0.612181, SD is 0.161009.
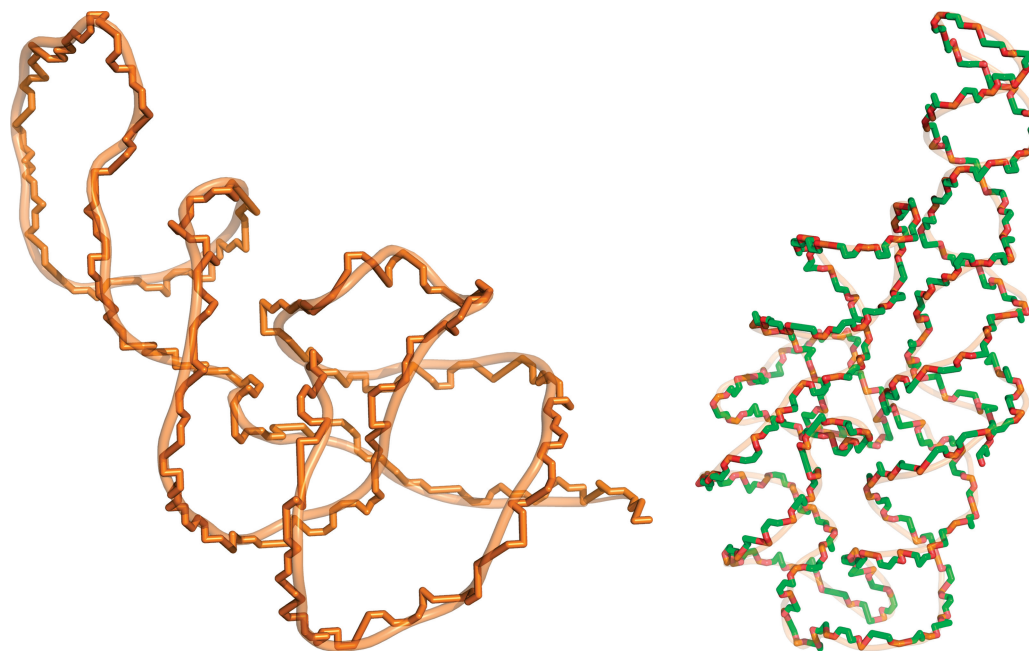
Figure 4. FCC lattice fits for the full backbone of two ribozymes models [PDB IDs 1SJ3:R (left panel) and 1GID:A (right panel)] superimposed with their original models.

Figure 5. Screen shot of LocalMove web server.

`LocalMove`, while Y.P. extended the algorithm and built the web server, requiring an interface using Python, JavaScript Jmol and mySQL. Invaluable technical help was provided by staff bioinformatics programmer, Jason Persampieri. `LocalMove` is somewhat related to the software MoCaPro, written by Sebastian Will under the direction of P.C. at Ludwig-Maximilians-Universität München. The latter software generalized the work of Šali *et al.* (14); MoCaPro is not publicly available and cannot perform the computations supported by `LocalMove`.

REFERENCES

1. Abe,H. and Go,N. (1981) Noninteracting local-structure model of folding and unfolding transition in globular proteins. II. Application to two-dimensional lattice proteins. *Biopolymers*, **20**, 1013–1031.
2. Lau,K. and Dill,K.A. (1989) A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *J. Am. Chem. Soc.*, **22**, 3986–3997.
3. Hart,W.E. and Istrail,S.C. (1996) Fast protein folding in the hydrophobic-hydrophilic model within three-eighths of optimal. *J. Comput. Biol.*, **3**, 53–96.
4. Kihara,D., Lu,H., Kolinski,A. and Skolnick,J. (2001) TOUCHSTONE: an *ab initio* protein structure prediction method that uses threading-bases tertiary restraints. *Proc. Natl Acad. Sci. USA*, **98**, 10125–10130.
5. Dokholyan,N.V., Buldyrev,S.V., Stanley,H.E. and Shakhnovich,E.I. (1998) Discrete molecular dynamics studies of the folding of a protein-like model. *Fold. Des.*, **3**, 577–587.
6. Berman,H.M., Battistuz,T., Bhat,T.N., Bluhm,W.F., Bourne,P.E., Burkhardt,K., Feng,Z., Gilliland,G.L., Iype,L., Jain,S. *et al.* (2002) The Protein Data Bank. *Acta Crystallogr. D. Biol. Crystallogr.*, **58**, 899–907.
7. Manuch,J. and Gaur,D.R. (2008) Fitting protein chains to cubic lattice is np-complete. *J. Bioinform. Comput. Biol.*, **6**, 93–106.
8. Godzik,A., Kolinski,A. and Skolnick,J. (1993) Lattice representations of globular proteins: How good are they? *J. Comput. Chem.*, **14**, 1194–1202.

9. Rabow,A.A. and Scheraga,H.A. (1993) Lattice neural network minimization. Application of neural network optimization for locating the global-minimum conformations of proteins. *J. Mol. Biol.*, **232**, 1157–1168.

10. Rykunov,D.S., Reva,B.A. and Finkelstein,A.V. (1995) Accurate general method for lattice approximation of three-dimensional structure of a chain molecule. *Proteins Struct. Funct. Genet.*, **22**, 100–109.

11. Park,B.H. and Levitt,M. (1995) The complexity and accuracy of discrete state models of protein structure. *J. Mol. Biol.*, **249**, 493–507.

12. Reva,B.A., Rykunov,D.S., Olson,A.J. and Finkelstein,A.V. (1995) Constructing lattice models of protein chains with side groups. *J. Comput. Biol.*, **2**, 527–535.

13. Huang,X. (2007) Fitting protein chains to lattice using integer programming approach. *M.Sc. thesis*. School of Computing Science, Simon Fraser University.

14. Sali,A., Shakhnovich,E. and Karplus,M. (1994) How does a protein fold? *Nature*, **369**, 248–251.

15. Reva,B., Finkelstein,A., Rykunov,D. and Olson,A. (1996) Building self-avoiding lattice models of proteins using a self-consistent field optimization. *Proteins Struct. Funct. Genet.*, **26**, 1–8.

16. Reva,B.A., Rykunov,D.S., Finkelstein,A.V. and Skolnick,J. (1998) Optimization of protein structure on lattices using a self-consistent field approach. *J. Comput. Biol.*, **5**, 531–538.

17. Metropolis,N., Rosenbluth,A., Rosenbluth,M., Teller,A. and Teller,E. (1953) Equation of state calculations by fast computing machines. *J. Chem. Phys.*, **21**, 1087–1092.

18. Clote,P. and Backofen,R. (2000) *Computational Molecular Biology: An Introduction*. John Wiley & Sons Ltd., Chichester, pp. 279.

19. Berman,H.M., Olson,W.K., Beveridge,D.L., Westbrook,J., Gelbin,A., Demeny,T., Hsieh,S.H., Srinivasan,A.R. and Schneider,B. (1992) The nucleic acid database. A comprehensive relational database of three-dimensional structures of nucleic acids. *Biophys. J.*, **63**, 751–759.

20. Hobohm,U. and Sander,C. (1994) Enlarged representative set of protein structures. *Protein Sci.*, **3**, 522.

21. Ferre,F., Ponty,Y., Lorenz,W.A. and Clote,P. (2007) DIAL: a web server for the pairwise alignment of two RNA three-dimensional structures using nucleotide, dihedral angle and base-pairing similarities. *Nucleic Acids Res.*, **35**, W659–W668.