RedMDStream: Parameterization and simulation toolbox for coarse–grained molecular dynamics models

Supporting material

Filip Leonarski[*] [†], **Joanna Trylska**[*]

[*]Centre of New Technologies, University of Warsaw, Banacha 2c, 02-097 Warsaw, Poland;
[†]Department of Chemistry, University of Warsaw, Pasteura 1, 02-093 Warsaw, Poland
e-mail: F.Leonarski@cent.uw.edu.pl, joanna@cent.uw.edu.pl

S1 RedMDStream parallelization benchmarks

RedMDStream is adapted to run the most time consuming tasks in a parallel environment. These tasks are the optimizations of the CG model using meta-heuristic algorithms: the evolutionary algorithm and particle swarm optimization. The user may choose between the OpenMP parallelization, suitable for multi–core machines, and the Message Parsing Interface (MPI) based parallelization, suitable for high performance clusters. The benchmarks are provided in Fig. S1.

OpenMP, as seen in Fig. S1a, should be used only on machines with low number of cores (2–8). The MPI version is, on the other hand, well suited for large clusters. As shown in Figs. S1b–c, the performance growth is linear with the increasing number of processors up to 512 cores. The decrease of performance seen in Fig. S1b for 768 cores corresponds to a situation where the number of cores becomes close to the number of MD simulations performed in each iteration of the optimization algorithm (96 members x 10 reruns = 960 simulations). Increasing the generation size could help to achieve better performance. However, our other studies show that increasing the number of iterations rather than the generation size gives better results.

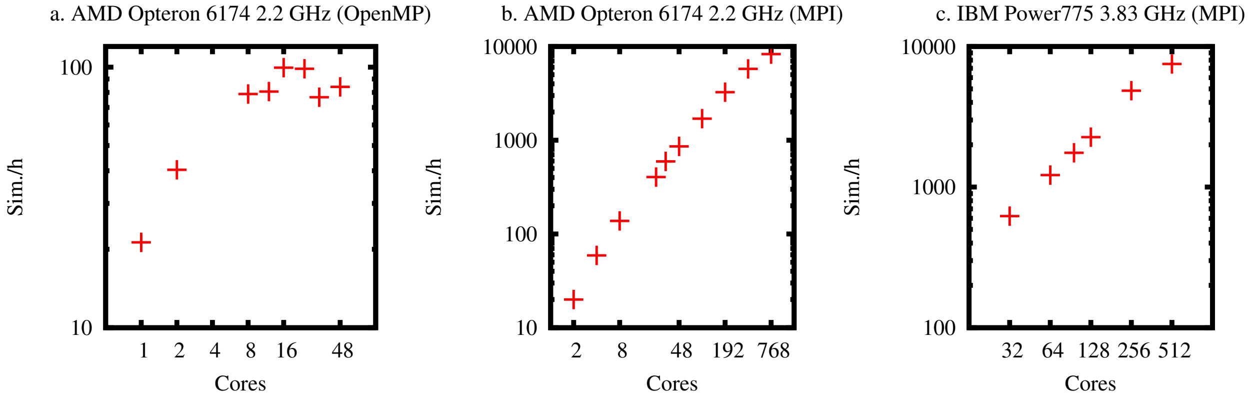
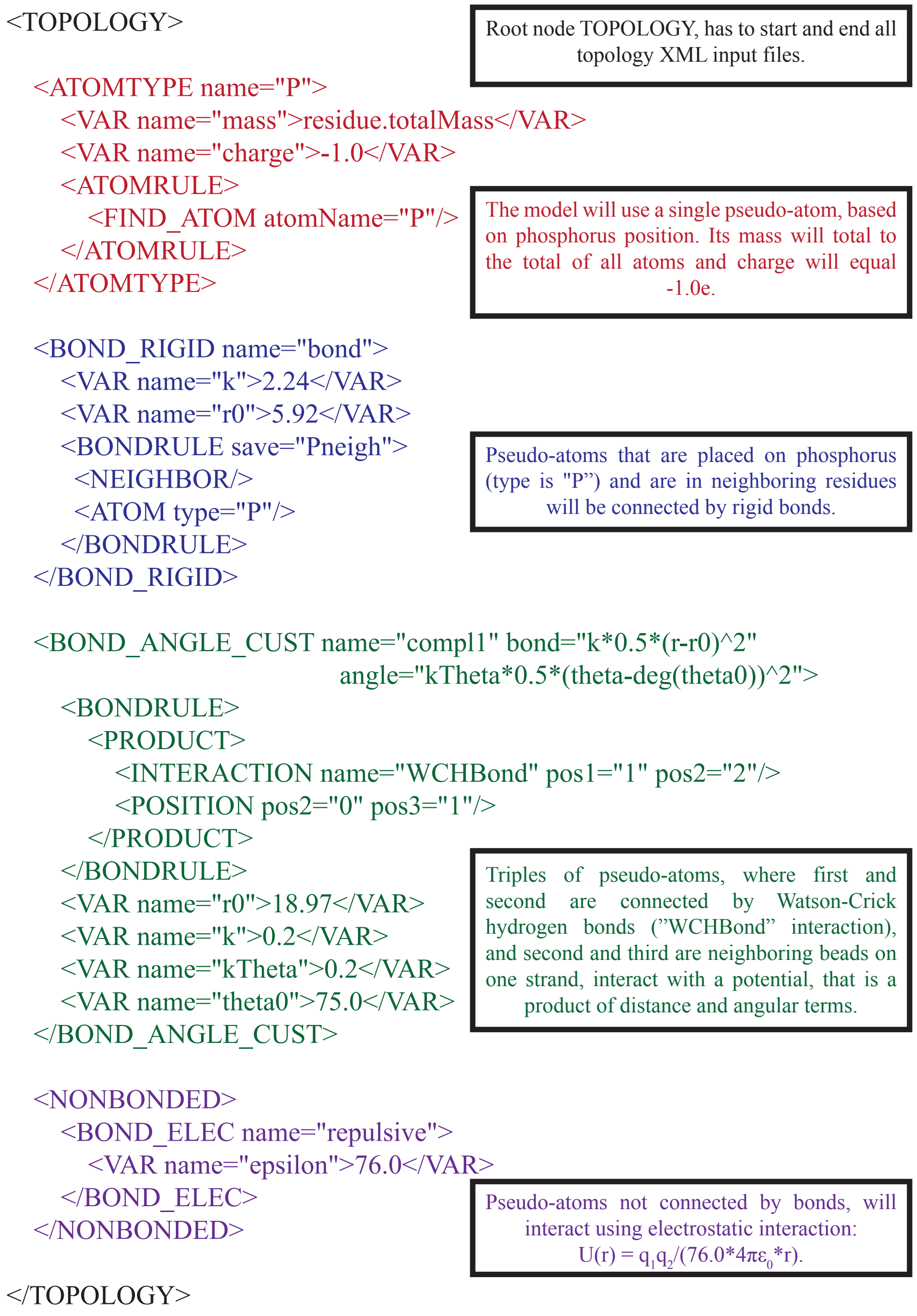


Figure S1: RedMDStream parallelization tests for the optimization of the example CG model of the microROSE fragment. Evolutionary algorithm optimization with the generation size of 96, each consisting of 10 independent 5 ns simulations. For clarity, a logarithmic scale was used on both the x and y axes. **a.** OpenMP on the AMD Opteron 6100 cluster, 48 cores/node. **b.** MPI on the AMD Opteron 6100 cluster, 48 cores/node. **c.** MPI on the IBM Power775 cluster, 32 cores/node.

S2 RedMDStream input XML format example



```
<TOPOLOGY>

  <ATOMTYPE name="P">
    <VAR name="mass">residue.totalMass</VAR>
    <VAR name="charge">-1.0</VAR>
    <ATOMRULE>
      <FIND_ATOM atomName="P"/>
    </ATOMRULE>
  </ATOMTYPE>

  <BOND_RIGID name="bond">
    <VAR name="k">2.24</VAR>
    <VAR name="r0">5.92</VAR>
    <BONDRULE save="Pneigh">
     <NEIGHBOR/>
     <ATOM type="P"/>
    </BONDRULE>
  </BOND_RIGID>

  <BOND_ANGLE_CUST name="compl1" bond="k*0.5*(r-r0)^2"
                              angle="kTheta*0.5*(theta-deg(theta0))^2">
    <BONDRULE>
      <PRODUCT>
        <INTERACTION name="WCHBond" pos1="1" pos2="2"/>
        <POSITION pos2="0" pos3="1"/>
      </PRODUCT>
    </BONDRULE>
    <VAR name="r0">18.97</VAR>
    <VAR name="k">0.2</VAR>
    <VAR name="kTheta">0.2</VAR>
    <VAR name="theta0">75.0</VAR>
  </BOND_ANGLE_CUST>

  <NONBONDED>
    <BOND_ELEC name="repulsive">
      <VAR name="epsilon">76.0</VAR>
    </BOND_ELEC>
  </NONBONDED>

</TOPOLOGY>
```

Root node TOPOLOGY, has to start and end all topology XML input files.

The model will use a single pseudo-atom, based on phosphorus position. Its mass will total to the total of all atoms and charge will equal -1.0e.

Pseudo-atoms that are placed on phosphorus (type is "P”) and are in neighboring residues will be connected by rigid bonds.

Triples of pseudo-atoms, where first and second are connected by Watson-Crick hydrogen bonds (”WCHBond” interaction), and second and third are neighboring beads on one strand, interact with a potential, that is a product of distance and angular terms.

Pseudo-atoms not connected by bonds, will interact using electrostatic interaction: $U(r) = q_1 q_2/(76.0*4\pi\varepsilon_0*r)$.

Figure S2: Example of an XML file describing the CG MD model topology in the RedMDStream software. For brevity and clarity of the figure, this example does not describe a particular force field but provides a general outline of the file limiting the number of potential energy terms. The file is constructed according to the XML standard, based on the concept of the tags enclosed in < and > brackets.

S3 Example CG model

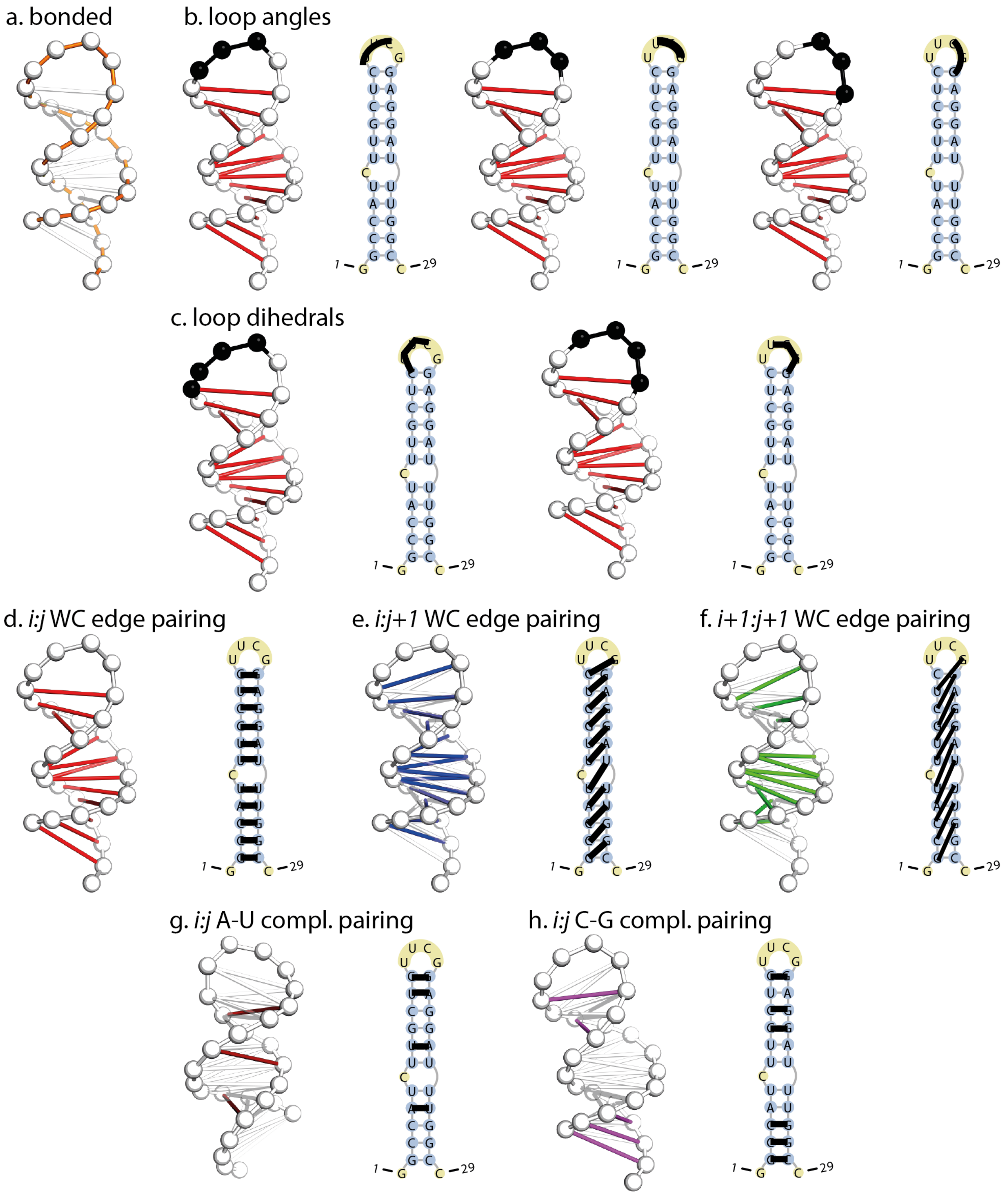
Table S1: Potential energy function used for different variants of the CG model. Potential energy function types: harmonic potential for the distance is $V(r) = \frac{1}{2}k_r(r - r_0)^2$ and for the angle is $V(\theta) = \frac{1}{2}k_\theta(\theta - \theta_0)^2$, cosine is used for the following dihedral $V(\alpha) = k_\alpha(1 - \cos(\alpha - \alpha_0))$, and repulsive stands for the Coulombic repulsive nonbonded potential $V(r) = \frac{1}{4\pi\varepsilon_0\varepsilon}\frac{q_1 q_2}{r}$. Input files for the example are included in the Supporting Material file named microrose.RedMDStream.zip.

		Variant 1	Variant 2	Variant 3
Intrastrand bond	see Fig. S3a	harmonic	harmonic	harmonic
Intrastrand angle (helix)	[a]	harmonic	harmonic	harmonic
Intrastrand angle (loop)	see Fig. S3b	harmonic	harmonic	harmonic
Intrastrand dihedral (helix)	[b]	cosine	cosine	cosine
Intrastrand dihedral (loop)	see Fig. S3c	cosine	cosine	cosine
Watson–Crick edge base pairing (1) (*i:j*)[c]	see Fig. S3d	harmonic	harmonic	—
Watson–Crick edge base pairing (*i:j+1*)	see Fig. S3e	—	harmonic	harmonic
Watson–Crick edge base pairing (*i+1:j+1*)	see Fig. S3f	—	harmonic	harmonic
Complementary A–U pairing (*i:j*)	see Fig. S3g	—	—	harmonic
Complementary C–G pairing (*i:j*)	see Fig. S3h	—	—	harmonic
Nonbonded		repulsive	repulsive	repulsive

[a] All intrastrand angles, with the exception of those designated as a loop (Fig S3b).

[b] All intrastrand dihedrals, with the exception of those designated as a loop (Fig S3c).

[c] Includes complementary A–U and C–G pairing, as well as wobble U–U pairs present in the structure (1).



a. bonded

b. loop angles

c. loop dihedrals

d. *i:j* WC edge pairing

e. *i:j+1* WC edge pairing

f. *i+1:j+1* WC edge pairing

g. *i:j* A-U compl. pairing

h. *i:j* C-G compl. pairing

Figure S3: Graphical presentation of different force field terms. Please use in connection with Table S1. WC stands for Watson-Crick.

S4 Reference trajectory

The following protocol was used to obtain a full–atomistic trajectory of the microROSE RNA thermometer hairpin (2). This trajectory was the source of the reference data (RMSD, RMSF, distance distributions) for the CG MD model optimization.

The starting structure was taken from the PDB database (PDB ID:2GIO) (2, 3). The structure was immersed in a box of water molecules to form a truncated octahedron, with 15 Å minimal distance between the solute and box edges. 28 K^+ ions were added to neutralize the RNA charge in the vicinity of the solute according to an electrostatic potential grid. Finally 28 K^+ and Cl^- ions were placed randomly inside the solvent to obtain a 100 mM ionic strength.

Full-atomistic MD simulation was performed using the Amber ff99 FF (4) with the bsc0 (5) and χ-OL (6) corrections and NAMD 2.9 (7) simulation engine. The SHAKE (8) algorithm was used to constrain water bonds, allowing for a 2 fs time step. The NVT ensemble was used for heating and equilibration and NPT ensemble for the production simulation. Constant temperature was obtained with the Langevin thermostat, with a damping constant set to 5 ps^{-1}. Constant pressure was controlled by the Langevin piston. Periodic boundary conditions were applied with the Particle Mesh Ewald method for long-range interactions with a grid spacing of about 1 Å and a 10 Å short-range cut-off for nonbonded interactions.

The simulation consisted of the following steps:

1. Local minimization for 10000 steps with restraints on all heavy atoms in the RNA.

2. Heating phase from 30 K to 310 K in 15 steps, every 20 K, each for 5 ps (75 ps in total) with restraints on all heavy atoms in the RNA.

3. Equilibration for 360 ps in the NVT ensemble with heavy atom restraints loosening in 6 steps.

4. Equilibration for 10 ns in the NPT ensemble without restraints.

5. Production trajectory of 100 ns.

Three independent trajectories were obtained with the same protocol. Trajectory analysis was performed using the RedMDStream application. The reference RMSF and distribution data presented in Figs. S4 and S7–S10 were obtained by averaging the outcome of the three trajectories. RMSD in Fig. S6 is presented for the trajectory No. 1, although there is no major difference with the other two.

S5 Optimization protocol

To obtain the best parameter set, the following optimization protocol was used with RedMD-Stream.

Each CG force field variant was optimized 3 times with the particle swarm optimization (PSO) method and 3 times with the evolutionary algorithm method. Optimization was carried out for at least 25 iterations. Each iteration involved evaluation of 96[1] different CG MD models (force fields). In this example the parameters of the functional terms presented in Table S1 were subject to optimization. After evaluation of all the 96 models, a new set of 96 models was created according to the rules of the optimization algorithms. PSO was run with default options. For the evolutionary algorithm, elitism with recalculation was turned on (16 members), blending crossing-over, tournament mating and Gaussian mutations. See RedMDStream reference manual for description of these options.

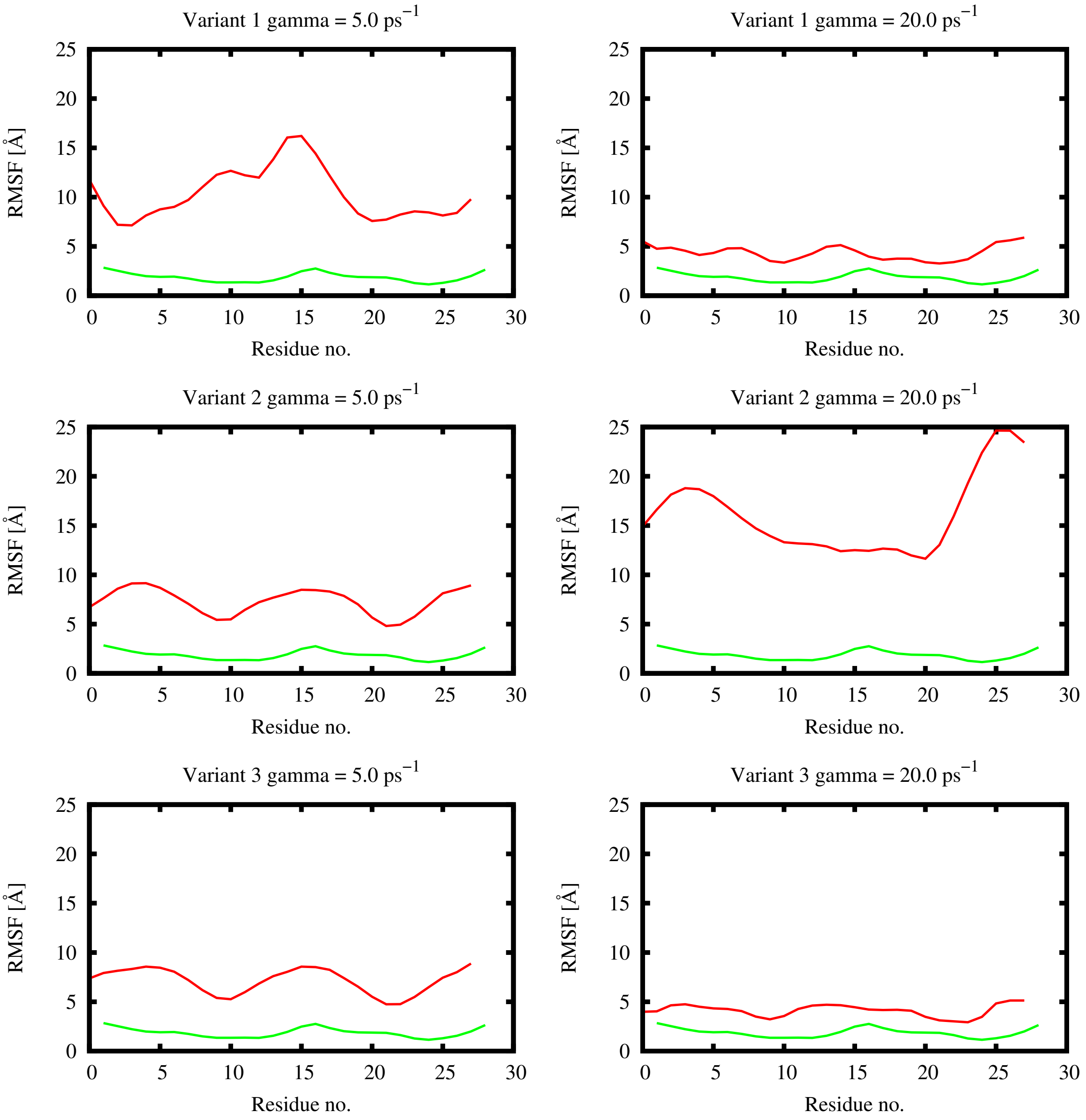
For each of the 96 models, in one iteration of the optimization procedure 10 CG MD runs were performed to account for random effects. A single simulation consisted of structure minimization, followed by a 5 ns Langevin dynamics run with a 10 fs time step. Langevin dumping constant was set either to 5 ps^{-1} or 20 ps^{-1}. The optimization criteria according to which the CG force fields were scored are listed in Table S2. The criteria were calculated for each of 10 CG MD repeats of the simulation and then averaged.

Table S2: Optimization criteria. The reference conformation for criterion 1 was the NMR structure deposited in PDB. Criteria 2–11 are referenced to a 100 ns full–atomistic MD simulation of microROSE in the Amber f99 force field (see main text and Section S4). Distributions in criteria 4–11 are determined by taking the average of the Hodgkin index and Kolmogorov-Smirnov score. Weights are arbitrary and based on our previous parameterization tests for other systems. Please refer to RedMDStream manual for details on how these criteria are calculated.

No.	Optimization criterion	Weight	Range
1	Root mean square deviation	6%	3.0 - 10.0 Å
2	Root mean square fluctuation	6%	0.0 - 10.0 Å
3	All–atom distance distribution difference	16%	0.0 - 1.0
4	Intrastrand neighbor distance distribution difference	8%	0.0 - 1.0
5	Intrastrand neighbor angle distribution difference	12%	0.0 - 1.0
6	Intrastrand neighbor angle distribution difference (loop only, see Fig. S3b)	8%	0.0 - 1.0
7	Intrastrand neighbor dihedral distribution difference	12%	0.0 - 1.0
8	Intrastrand neighbor dihedral distribution difference (loop only, see Fig. S3c)	8%	0.0 - 1.0
9	*i:j* WC edge pair distance (see Fig. S3d)	8%	0.0 - 1.0
10	*i:j+1* WC edge pair distance (see Fig. S3e)	8%	0.0 - 1.0
11	*i+1:j+1* WC edge pair distance (see Fig. S3f)	8%	0.0 - 1.0

[1] This number is termed *generation size* in the evolutionary algorithm nomenclature.

S6 Results


Variant 1 gamma = 5.0 ps^{-1}

Variant 1 gamma = 20.0 ps^{-1}

Variant 2 gamma = 5.0 ps^{-1}

Variant 2 gamma = 20.0 ps^{-1}

Variant 3 gamma = 5.0 ps^{-1}

Variant 3 gamma = 20.0 ps^{-1}

Figure S4: Root mean square fluctuations of the P beads of the best model in each tested CG model variant on a 5 ns timescale. Gamma is the Langevin damping constant. **Red line** for the coarse–grained simulation, **green** for the Amber ff99+bsc0+chiOL reference 100 ns simulation.

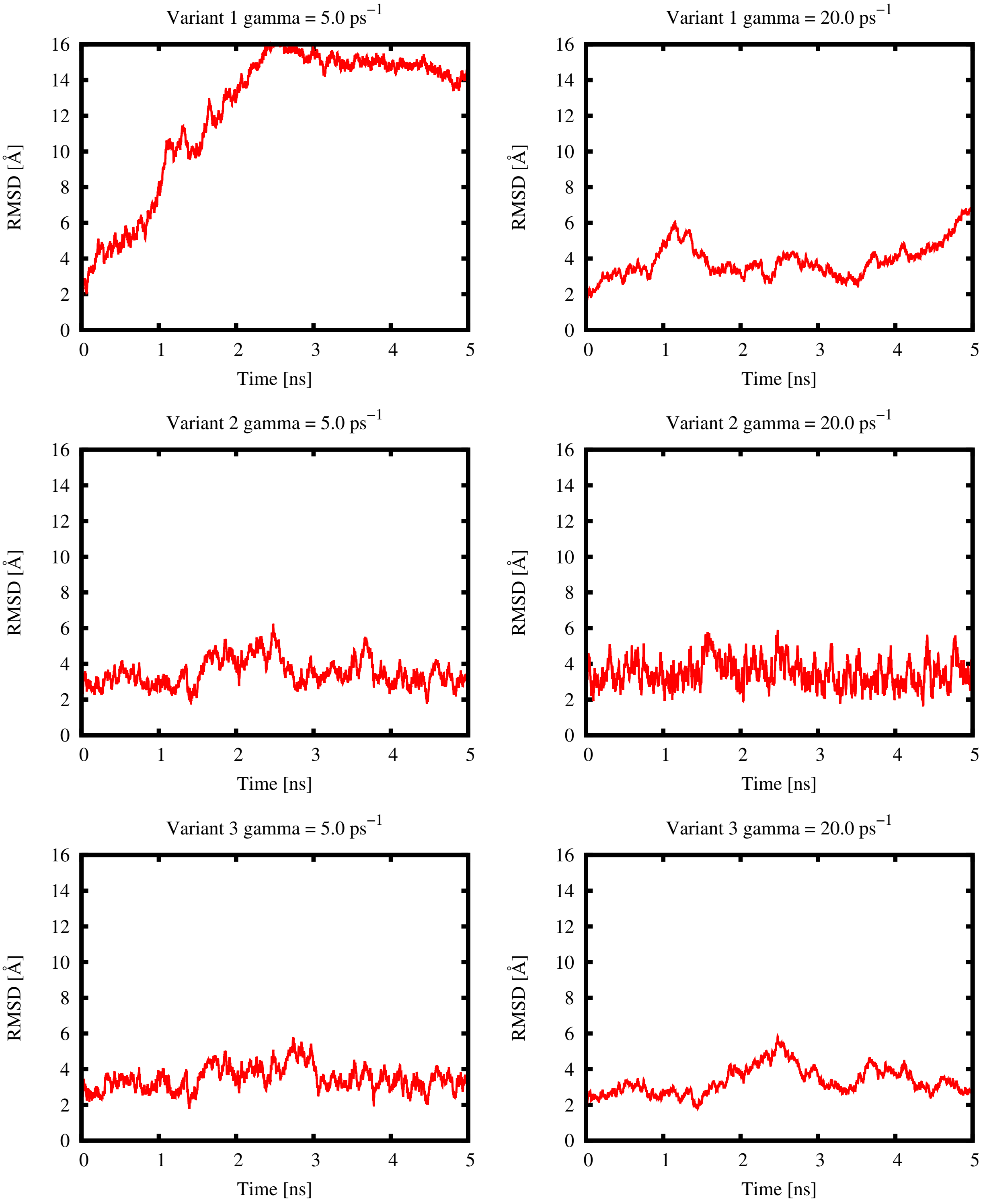


Figure S5: Root mean square deviation of the P beads of the best model in each tested CG variant on a 5 ns timescale. Gamma is the Langevin damping constant.

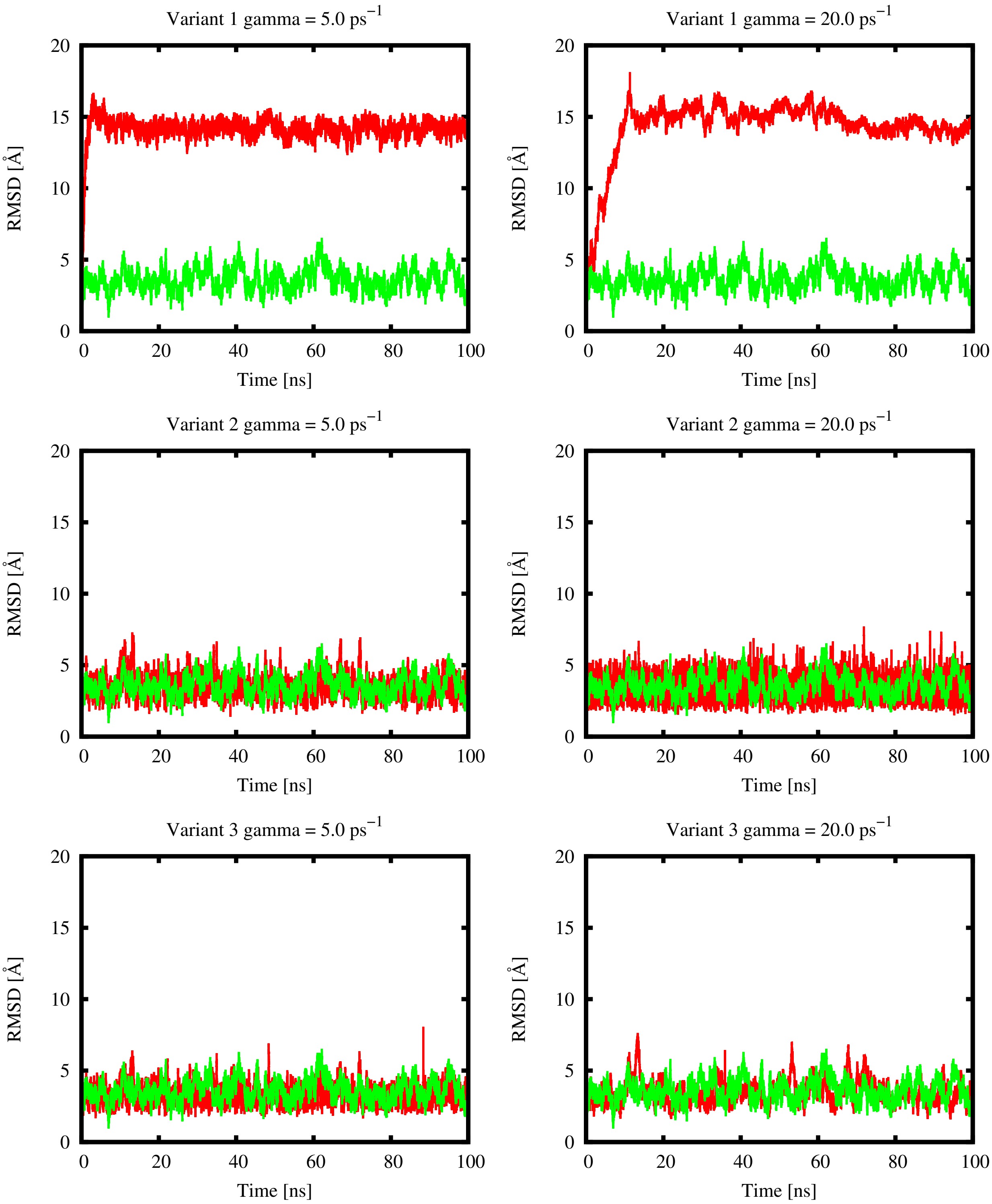


Figure S6: Root mean square deviation of the P beads of the best model in each tested CG variant on a 100 ns timescale. Gamma is the Langevin damping constant. **Red line** for the coarse–grained simulation, **green** for the Amber ff99+bsc0+chiOL reference 100 ns simulation.

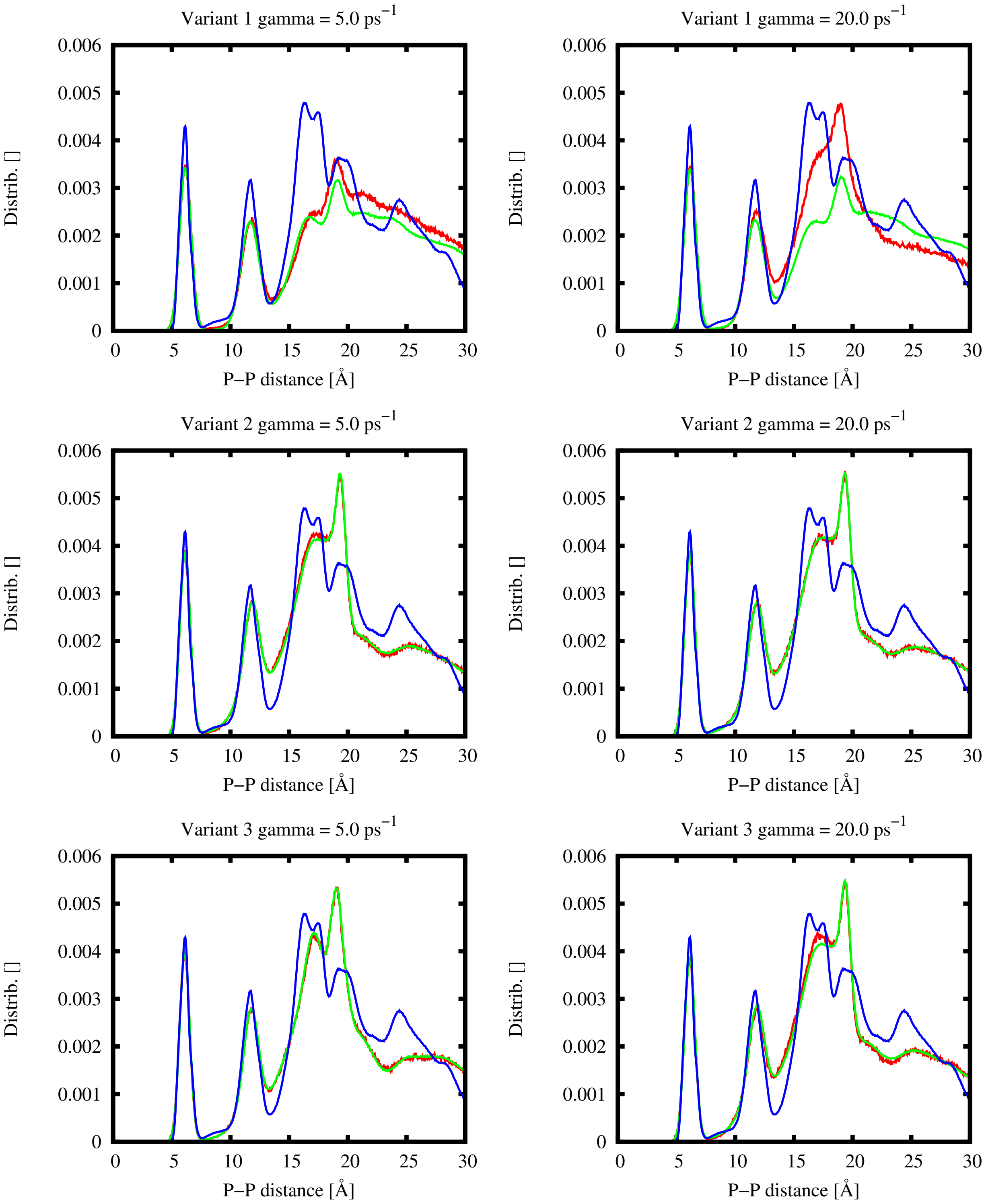


Figure S7: All–bead distance distributions for the best model in each tested CG variant. **Red line** 5 ns timescale, **green** 100 ns timescale, **blue** Amber ff99+bsc0+chiOL reference 100 ns trajectory. Gamma is the Langevin damping constant.

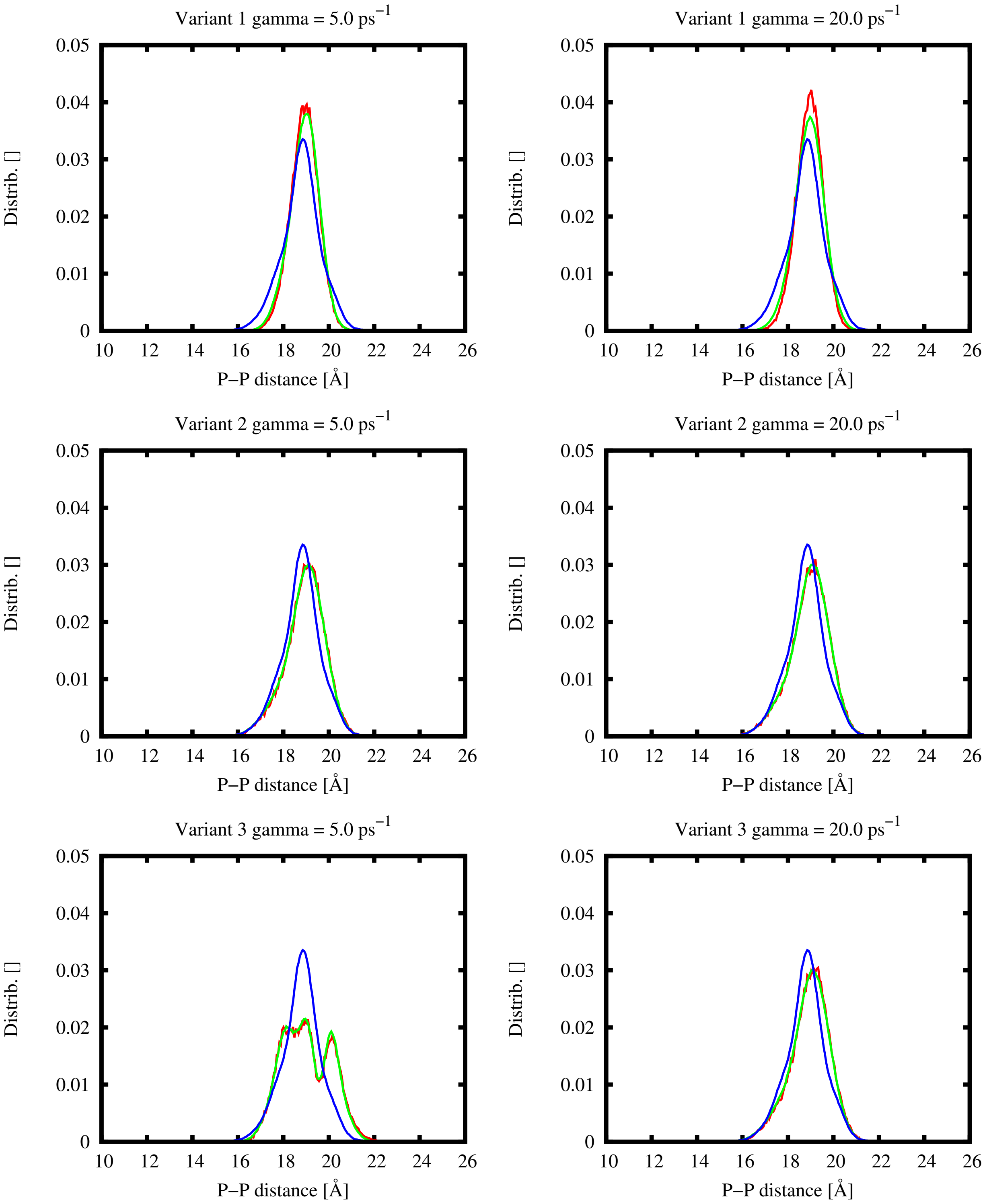


Figure S8: *i:j* distance distributions for the complementarily paired beads (see Fig. S3d) for the best model in each tested CG variant. **Red line** 5 ns timescale, **green** 100 ns timescale, **blue** Amber ff99+bsc0+chiOL reference 100 ns trajectory. Gamma is the Langevin damping constant.

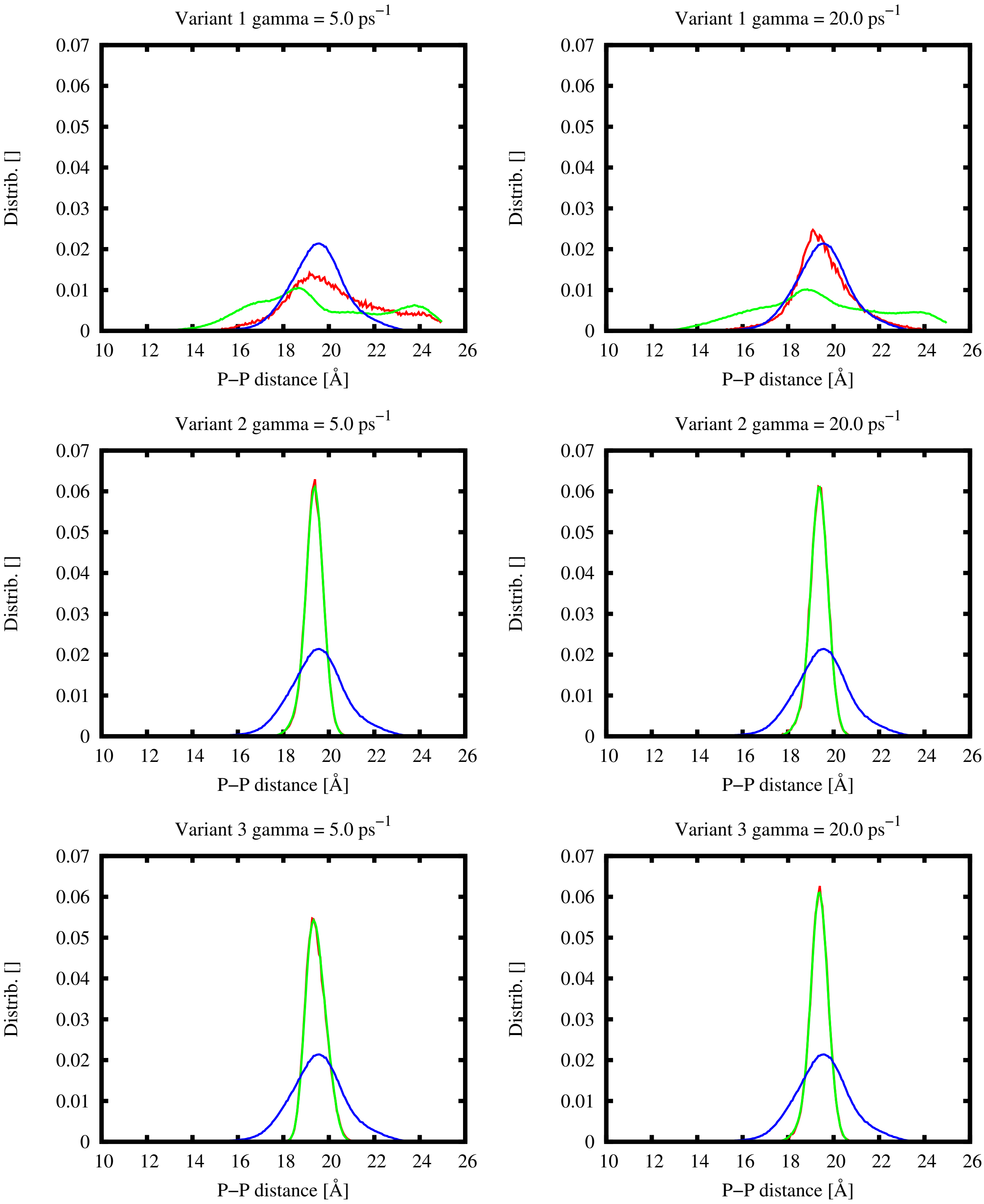


Figure S9: *i:j+1* distance distributions for the complementarily paired beads (see Fig. S3e) for the best model in each CG variant. **Red line** 5 ns timescale, **green** 100 ns timescale, **blue** Amber ff99+bsc0+chiOL reference 100 ns trajectory. Gamma is the Langevin damping constant.

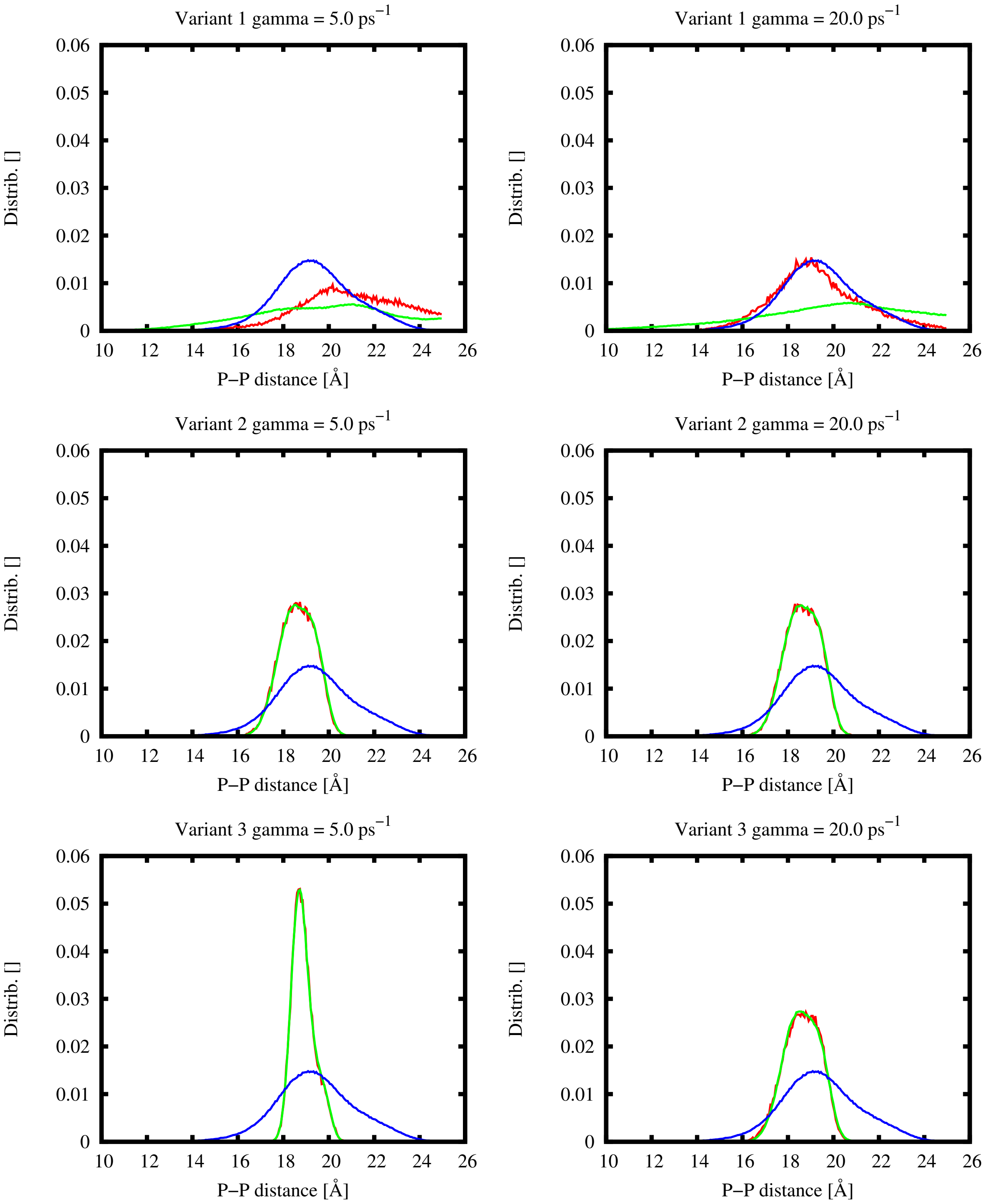


Figure S10: *i+1:j+1* distance distributions for the complementarily paired beads (see Fig. S3f) for the best model in each CG variant. **Red line** 5 ns timescale, **green** 100 ns timescale, **blue** Amber ff99+bsc0+chiOL reference 100 ns trajectory. Gamma is the Langevin damping constant.

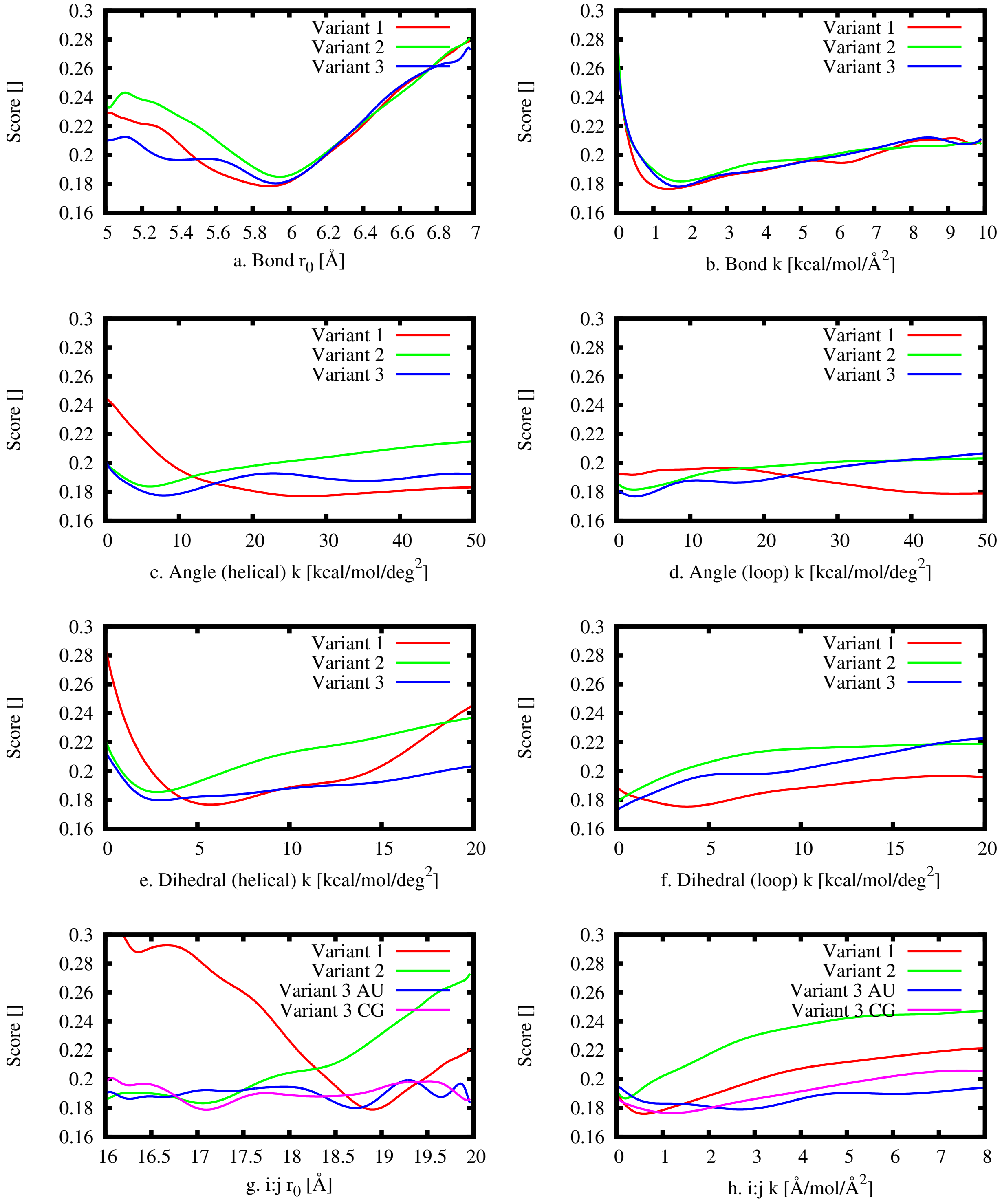


Figure S11: The relationship between the force field parameter and optimization score for different variants of the CG model. Since Variant 3 differentiates between AU and CG base pairs, the parameters characterizing these interactions are provided separately. Due to large noise, a Bezier smoothening was applied so the graphs provide only a general overview of the function's landscape.

S7 Note on metaheuristic optimization of CG MD parameters

MD simulations depend on the starting positions and velocities assigned to system's particles and thus are chaotic by nature. A small change in initial conditions, e.g. in random initial assignment of atoms' velocities, gives different trajectories of the same system. No two simulations with different random number generator seed are the same and this is reflected in the score. No two trajectories of the same system with the same setup return the same numerical score. From our experience the differences of about 10% in subsequent scores, i.e. from trajectories starting with different initial velocities, should be expected. Also, this uncertainty is at the same level regardless if 10 or 100 independent runs of exactly the same system were executed. So we see no convergence upon increasing the number of independent trajectories that are taken into account in the statistics. However, we observe a relationship between the simulation length and uncertainty. Increasing the simulation time scale from 5 ns to 100 ns resulted in a 4–fold decrease in the score standard deviation among 10 runs, mostly because of more consistent distributions between subsequent runs.

As a consequence one cannot fine–tune the CG MD model better than allowed by this intrinsic uncertainty of the score. This also propagates to all other aspects of the methodology. Not all changes in the parameters of the metaheuristics significantly affect the quality of the algorithm. Based on our previous experiences (9) and on the RNA thermometer example, we recommend the following settings for the evolutionary algorithm optimization:

- Elitism operation, i.e., propagating the best parameters without changes, is essential. Lowering the elite set size from 16 (out of 96) to 4, 1 and 0 increased the score for the converged solution, even for very long optimization runs. See example in Fig. S12.

- Tournament mating runs converge to a better score than the ones with a fitness-based mating.

- Change of mutation and cross–over type does not significantly affect the quality of the results.

- It takes only first few generations to find a solution close to the optimal one. Further, the optimization slows down and provides only minor improvements, even if carried over hundreds of generations. Improvements in the latter stage might be smaller than the significance level. Therefore, one should either perform a quick optimization (20 generations) or a long one (>100 generations).

- Enlarging the generation size does not improve much the converged score. We have not seen any benefit from having more than 100 members in a single generation. Performing more iterations should be preferred over using larger generations. If parallel calculations are performed with MPI on N processors, the generation sizes should be a multiple of N-1 for optimal usage of the cluster time.

We recommend to run the same optimization problem with evolutionary algorithm and particle swarm optimization (PSO). The latter has less options (only three parameters + generation size + iteration number) and still provides an effective optimization. For the PSO algorithm, the values of the parameters presented in the literature (10) ($\psi_p = 0.1$,$\psi_g = 0.1$ and $\omega = 0.9$) are effective and, from our tests, changing them does not improve the algorithm. We have not seen a bias towards one or the other algorithm in terms of the score quality. Considerations on the generation size and number of iterations presented above for the evolutionary algorithm also apply to PSO. RedMDStream also provides a simplex optimization, but unless a problem has a very

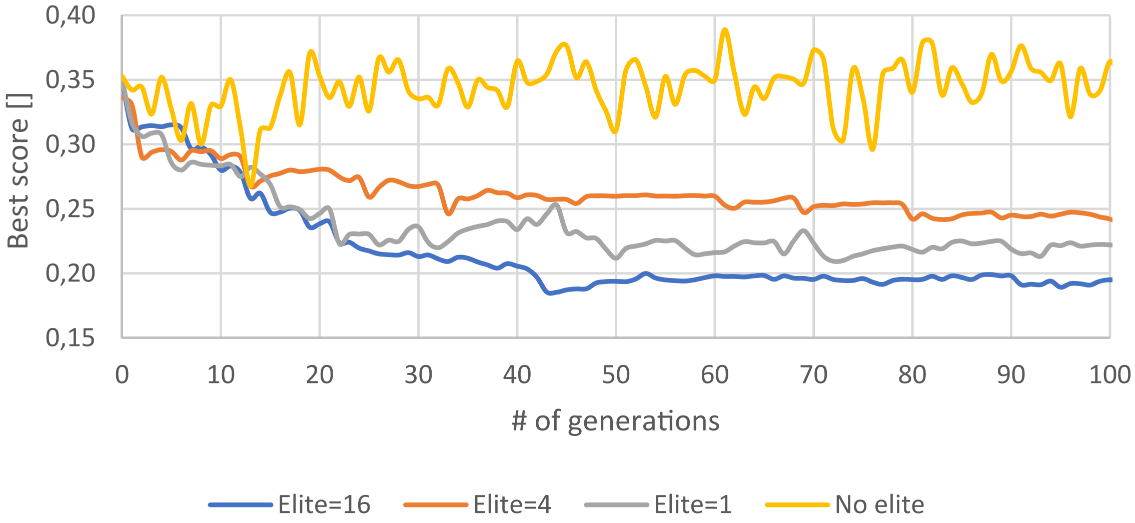


Figure S12: Effect of an elitism on the convergence of the evolutionary algorithm optimization. The best score in each generation is presented. The generation size was set to 96 and the size of an elite set (the models that are moved intact to the next generation) was subject to change. Turning off the elitism scheme, results in a nonfunctional algorithm. Good solutions are not saved for the next generations and the score does not converge to an optimal value. Larger size of an elite set gives better stability of the solution and better final score. RedMDStream allows elite solutions score to be recalculated every generation and added to already collected score values. Therefore, the elite solutions are under higher scrutiny and they might lose their place in the set, resulting in a higher score in a subsequent generation.

low uncertainty level (much lower than in the RNA thermometer example), the algorithm does not converge and behaves like a random walk, rather than a directed optimization.

The outcome of the evolutionary optimization or PSO algorithm is not a single optimal model, but rather a set of solutions that are indistinguishable by the scoring function. The user has to choose one CG model (CG force field) from this set of CG models with the same score. Testing these, equivalent from the point of view of the score, CG models under simulation conditions that were not part of the optimization scheme or selecting one of the CG models intuitively might be the best choice.

For example, in a CG MD simulations of ordered systems, like nucleic acids, the metaheuristic procedure usually gives low energy parameters for the repulsive part of the nonbonded interaction terms because the dynamic simulations do not involve a conformation in which two beads collide. However, it would be unwise to develop a general model for nucleic acids without the repulsive part. Therefore, one should either select the model with the strongest repulsion (from the ensemble of indistinguishable CG models) or appropriately set the nonbonded parameter range for the optimization (in a way to forbid too extensive weakening of the nonbonded repulsive part).

The metaheuristic optimization methods are helpful but their outcome has to be always related to the biophysical problem that one wants to solve. The selected solution (CG model) has to satisfy not only the score (note that the scoring function is defined by the user) but also the chemistry and physics of the problem.

Supporting references

[1] Leontis, N. B., and E. Westhof, 2001. Geometric nomenclature and classification of RNA base pairs. *RNA* 7:499–512.

[2] Chowdhury, S., C. Maris, ..., F. Narberhaus, 2006. Molecular basis for temperature sensing by an RNA thermometer. *EMBO J.* 25:2487–2497.

[3] Berman, H. M., T. Battistuz, ..., C. Zardecki, 2002. The Protein Data Bank. *Acta Crystallogr. D Biol. Crystallogr.* 58:899–907.

[4] Cornell, W. D., P. Cieplak, ..., P. Kollman, 1995. A Second Generation Force Field for the Simulation of Proteins, Nucleic Acids, and Organic Molecules. *J. Am. Chem. Soc.* 117:5179 – 5197.

[5] Pérez, A., I. Marchán, ..., M. Orozco, 2007. Refinement of the AMBER force field for nucleic acids: improving the description of alpha/gamma conformers. *Biophys. J.* 92:3817–29.

[6] Banáš, P., D. Hollas, ..., M. Otyepka, 2010. Performance of Molecular Mechanics Force Fields for RNA Simulations: Stability of UUCG and GNRA Hairpins. *J. Chem. Theory Comput.* 6:3836–3849.

[7] Phillips, J. C., R. Braun, ..., K. Schulten, 2005. Scalable molecular dynamics with NAMD. *J. Comput. Chem.* 26:1781–1802.

[8] Schlick, T., 2010. Molecular Modeling and Simulation: An Interdisciplinary Guide. 2nd. ed., Springer-Verlag, New York.

[9] Leonarski, F., F. Trovato, ..., J. Trylska, 2013. Evolutionary Algorithm in the Optimization of a Coarse-Grained Force Field. *J. Chem. Theory Comput.* 9:4874–4889.

[10] Haupt, R. L., and S. E. Haupt, 2004. Practical Genetic Algorithms. 2nd ed. John Wiley & Sons, Hoboken, New Jersey.