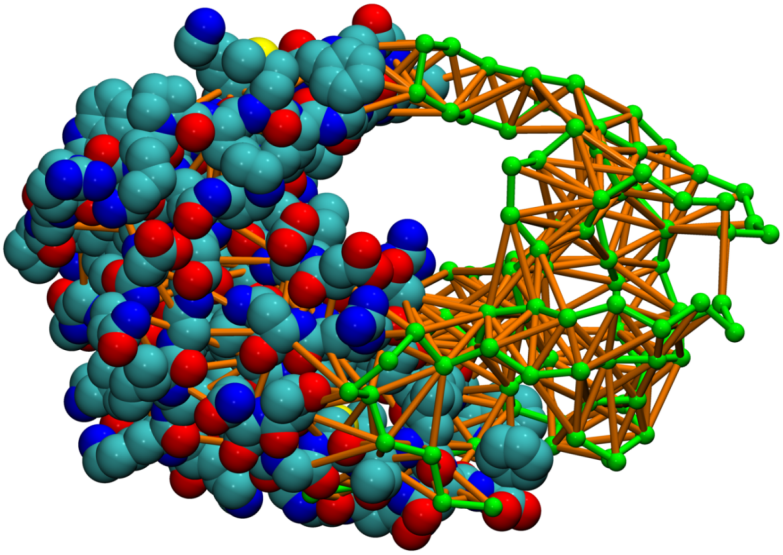
Reduced Molecular Dynamics
version 2.3
Users' Manual

Paweł Zieliński
Filip Leonarski
Adam Górecki
Marcin Szypowski
Maciej Długosz
Joanna Trylska



Contents

1 Introduction to the RedMD Package

RedMD is a free open source package for molecular dynamics simulations and normal mode analysis of coarse-grained models of proteins, nucleic acids and nucleoprotein complexes. The RedMD software was designed as a user-friendly, modular and scalable package to simulate internal motions of large biomolecular systems. The molecular interactions are currently based on one or two-bead mappings, i.e., each amino acid or nucleotide is represented as one or two interacting centers (pseudo-atoms or beads). Interactions between beads are modeled with analytic potential functions, such as harmonic, double-well, Morse or Lennard-Jones.

RedMD functionality is incorporated in a set of programs that allow the user to prepare molecular structures and generate topologies and verify input data, perform simulations. Simulations can be performed in the microcanonical and canonical ensembles (with the Berendsen thermostat or momentum scaling). Langevin and Brownian dynamics algorithms are also implemented. Internal motions can be also studied with normal mode analysis using either the elastic or the Gaussian network model. Algorithms to perform a minimization and the Monte Carlo conformational search in the dihedral angle space are also implemented.

RedMD is intended to be extensible. The source code is structured, therefore adding new methods and routines is straightforward.

The RedMD source code is written entirely in C and C++ languages and parallelized with OpenMP and MPI technologies. The code was tested on various LINUX platforms.

Please direct your comments and questions about the RedMD package to:
fleon@cent.uw.edu.pl or joanna@cent.uw.edu.pl
Centre of New Technologies, University of Warsaw
Żwirki i Wigury 93
02-089 Warsaw, Poland
phone: +48 22 5540 843

The source code of the RedMD package can be found at:

http://bionano.cent.uw.edu.pl/software

2 Parallelization

RedMD utilizes the force distribution model with the Open Multi-Processing Application Program Interface (OpenMP API, http://openmp.org) for shared-memory architectures and the Message Passing Interface standard (MPI, http://www.mcs.anl.gov/research/projects/mpi) for distributed-memory platforms (Figure 1). For most simulated problems algorithms implemented in RedMD provide almost linear scaling. Examples of parallelization efficiency are presented in Figure 2.

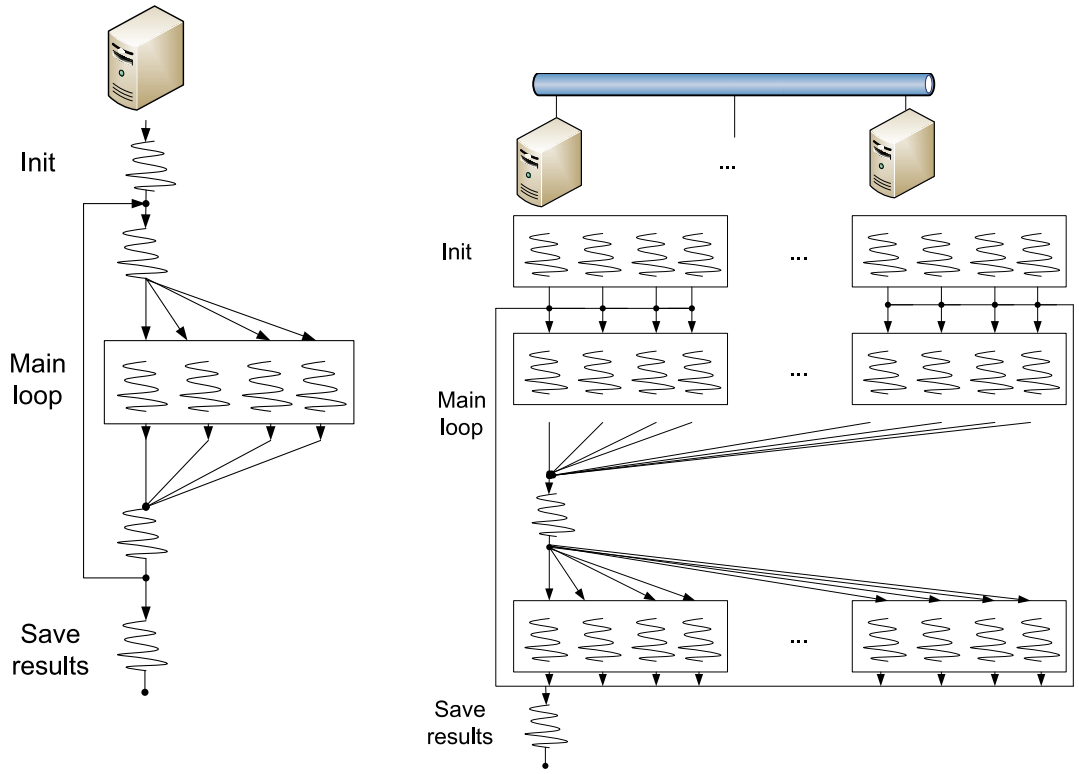


Figure 1: RedMD execution on a shared-memory architecture using OpenMP (left) and using MPI on a distributed memory architecture (right)

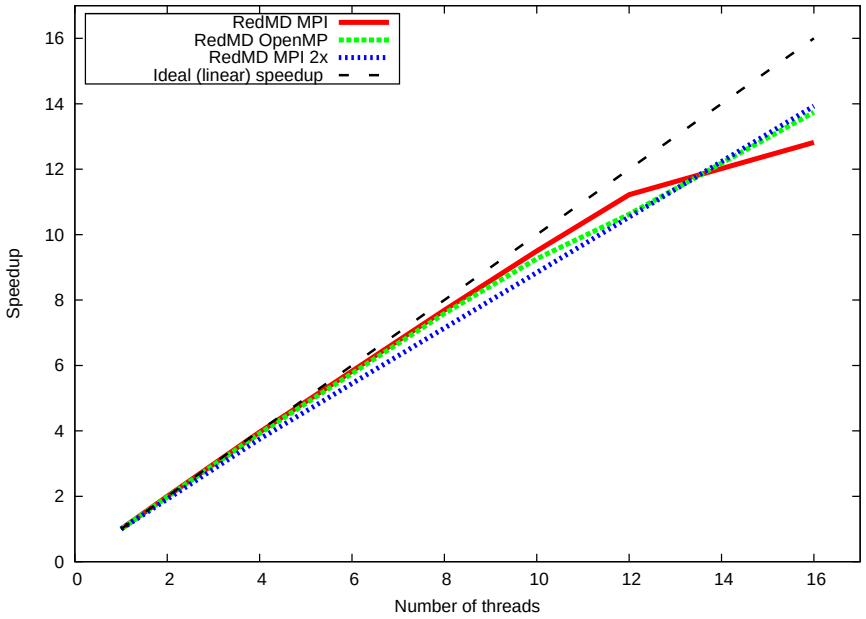


Figure 2: Speedup of nonbonded forces calculations for the 100-steps NVE dynamics simulation with the Berendsen thermostat performed for the 70S bacterial ribosome within the framework of the ribosome force field (see further in this manual). Speedup is defined as the ratio of the code execution time when using one thread to the code execution time when using N threads.

3 Installation

3.1 Requirements

Currently Linux is the only supported platforms. We tested the compilation with an open source GNU C Compiler (`gcc`) and the Intel C Compiler (`icc`). `libxml2` library is needed for parsing XML format utilized by the RedMD. For proper compilation one has to ensure that `libxml2` library is accompanied by a set of development tools and utilities. If the library is installed with a Linux package management system, one should install a package called `libxml2-dev` (Linux distributions: Debian, Ubuntu, etc.) or `libxml2-devel` (Linux distributions: RedHat, OpenSUSE, etc.).

The RedMD source code can be downloaded from `http://bionano.icm.edu.pl/software` in a form of a GNU gzip compressed `tgz` archive. We provide a configuration script and files. A standard Unix-style `make` environment is suitable to compile and build RedMD on most machines.

3.2 Compiling the code for serial and parallel execution

First, a `.tgz` archive file should be unpacked in a directory of choice (for example `/opt`):

```
cp RedMD.tgz /opt
cd /opt
gunzip -c RedMD.tgz | tar xvf -
cd ./RedMD
```

Next, from the RedMD directory the following configuration script should be executed:

```
./configure --disable-openmp --disable-mpi --prefix=/opt/RedMD
```

To compile RedMD with the OpenMP support:

```
./configure --enable-openmp --prefix=/opt/RedMD
```

To compile RedMD with the MPI support, after appropriate tuning of environment variables `CC` and `CXX`:

```
./configure --enable-mpi --prefix=/opt/RedMD
```

(list of possible options and instructions available during configuration can be displayed with the `--help` option)

The final step, building and installing of binaries, is performed using the commands:

```
make
```

and then

```
make install
```

Compiled executables are stored in the `/bin` area within the RedMD main directory.

3.3 RedMD Tools

Normal mode analysis tools: `RedENM` (Elastic Network Model) and `RedGNM` (Gaussian Network Model) and the trajectory analysis tool: `RedMD_analyze` require separate compilation. The source code and appropriate `Makefile` are stored in the `src/tools/` directory in the RedMD main directory. Before compilation it might be necessary to edit the `Makefile` and set the compiler flags and options. To compile `RedENM`, `RedGNM` and `RedMD_Analyze` modules the user should run the `make` command from the `tools` directory.
ARPACK (http://www.caam.rice.edu/software/ARPACK/) and LAPACK (http://www.netlib.org/lapack/) libraries are needed to build `RedENM` and `RedGNM` executables.

4 Units

RedMD uses the following system of units:

	internal	**input and output files input parameters**
Length	$\mathring{A} = 10^{-10}m$	$\mathring{A}$
Mass	$u = 1.6605402(10) \cdot 10^{-27}kg$	u
Charge	$e = 1.602 \cdot 10^{-19}C$	e
Time	$ps = 10^{-12}s$	ps
Temperature	K	K
Energy	$10\frac{J}{mol}$	$\frac{kcal}{mol}$

5 Force Fields

The force fields currently implemented in the RedMD package are based either on one- or two-bead mapping. In case of one-bead mapping, each amino acid or nucleotide is represented as a single spherical particle (bead) centered either on a C_α carbon or P phosphorous atom, with a mass corresponding to the total mass of a given amino acid or nucleotide. In case of two-bead mapping, each amino acid consists of two beads: a pseudo-atom CA (placed in the position of the C_α carbon) and a pseudo-atom CB (representing a side chain).

5.1 Generic Elastic Network Model

Generic Elastic Network Model shown in Figure 3 is based on the work of M. Tirion [1] and uses the following potential energy function, E:

$$E = \sum_{\{m,n\}\in\{C_\alpha,P\}} \frac{1}{2} \sum_{i<j}^{N} k_{mn} \left(r_{i,j} - r_{i,j}^{o}\right)^2 \tag{1}$$

where:

$$r_{mn} < R_{mn}^{cut-off} \tag{2}$$

Atoms C_α and P (indices m and n) are connected through harmonic bonds with force constants k_{mn} and equilibrium lengths r^o, provided that their mutual distance ($r_{m,n}$) in the initial structure is smaller than the predefined cut-off ($R_{mn}^{cut-off}$). Equilibrium bond lengths are computed for each pair of connected atoms based on the initial structure. Default parameters are shown in Table 1.

bond type	force constant $[\frac{kcal}{molA^2}]$	cut-off [Å]
C_α-C_α	1.0	8.0
C_α-P	1.0	15.0
P-P	1.0	20.0

Table 1: Default parameters for the generic Elastic Network Model

5.2 Ribosome Model

The extension of the Elastic Network Model for the ribosome is based on our earlier study [2]. This is a one-bead model with interactions between pseudo-atoms modeled according to the following potential energy function:

$$E = E_{1-2} + E_{1-3} + E_{1-4} + E^{nonbonded} + E^{bp} \tag{3}$$

The first three terms (E_{1-2}, E_{1-3}, E_{1-4}) account for the pseudo-bond, pseudo-angle and pseudo-dihedral interactions between the two, three, and four successive beads in the chain (see Figure 4).

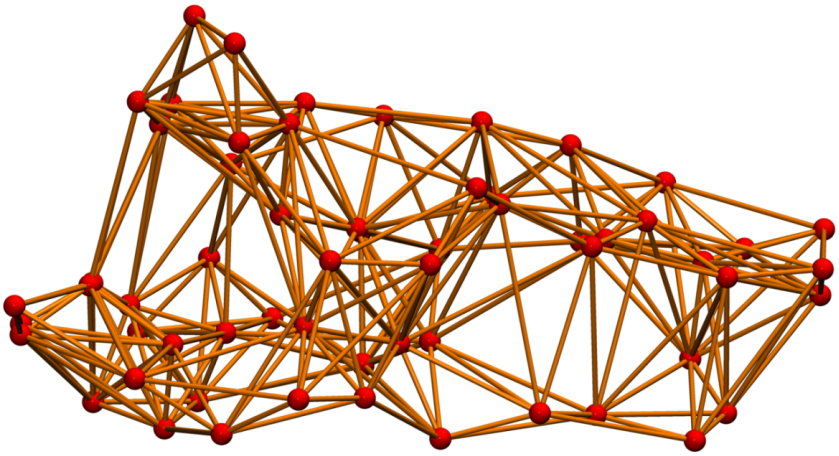


Figure 3: Interactions (orange lines) between pseudo-atoms (shown as beads) in the generic elastic network model.

They are modeled as harmonic potentials:

$$E_{1-2} = \sum_{i,i+1} \frac{1}{2} k_{1-2} (r_{i,i+1} - r^o_{i,i+1})^2 \tag{4}$$

$$E_{1-3} = \sum_{i,i+2} \frac{1}{2} k_{1-3} (r_{i,i+2} - r^o_{i,i+2})^2 \tag{5}$$

$$E_{1-4} = \sum_{i,i+3} \frac{1}{2} k_{1-4} (r_{i,i+3} - r^o_{i,i+3})^2 \tag{6}$$

where r^o_{i-j+1} $(j = 1, 2, 3)$ are equilibrium distances between beads taken from the initial structure, and force constants k_{1-j+1} depend on the type of the bead and pseudo-bond, pseudo-angle, and pseudo-dihedral interaction. The nonbonded energy term, $E^{nonbonded}$ is described with two Morse potential formulas (denoted as local and nonlocal) of a similar form:

$$A_{P,C_\alpha}(r^o)[1 - \exp(-\alpha(r - r^o))]^2 \tag{7}$$

but with different parameters.

Local C_α-C_α and P-P terms: applied within a specified cut-off distance ($R^{cut-off}$). Local interactions are structure specific as r^o is taken from the starting geometry.

Nonlocal C_α-C_α and P-P terms: applied between $R^{cut-off}$ and $R^{cut-off}_{max}$. Nonlocal terms are not structure specific as r^o depends only on the bead type and not on the initial configuration. In both cases, in order to account for decreasing of interactions with distance, the $A_{P,C_\alpha}(r^o)$ coefficient is given with an analytical formula:

$$A_{P,C_\alpha}(r^o) = const_1 \exp(\frac{r^o}{const_2}) \tag{8}$$

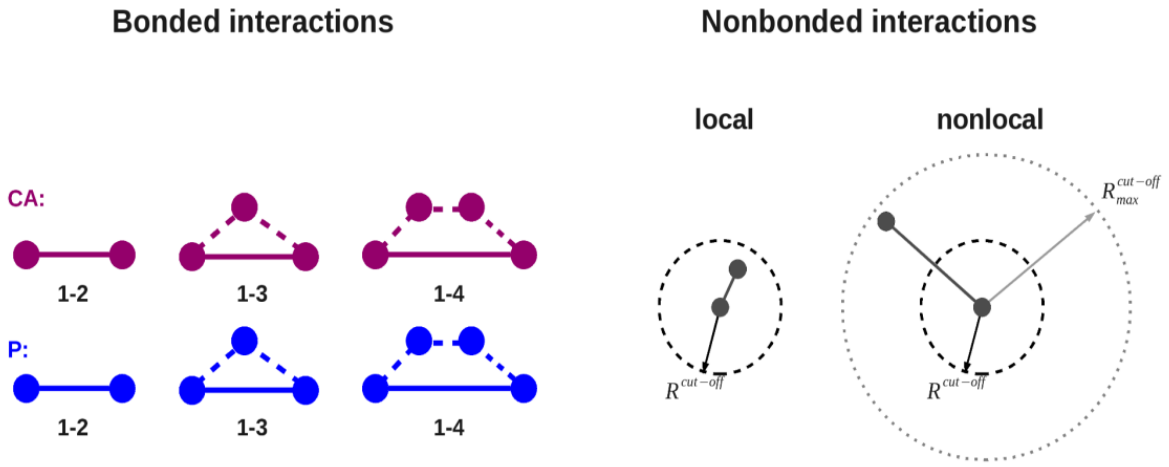


Figure 4: A diagram showing the types of interactions accounted for in the ribosome model force field.

Parameters for C_α-P interactions are computed based on geometric averaging and for example the cut-off distance, $R^{cut-off}_{C_\alpha - P}$ is calculated with the formula:

$$R^{cut-off}_{C_\alpha - P} = \sqrt{R^{cut-off}_{C_\alpha} R^{cut-off}_{P}} \tag{9}$$

The secondary structure of the RNA (interactions between paired bases, the E^{bp} term) is accounted for with a harmonic potential:

$$E^{bp}(r) = \frac{1}{2} k_{bp} (r - r^o_{bp})^2 \tag{10}$$

with the equilibrium distance r^o_{bp} taken from the starting structure. The parameters were derived from the pair radial distribution function calculated from the crystal structure of the ribosome. The Boltzmann inversion was applied and the parameters were iteratively adjusted to corroborate with experimental β-factors and amplitudes of movements seen in cryo-electron microscopy studies of the ribosome. Default parameters are presented in Table 2 and details of the parameterization procedure are described in [2].

5.3 Nucleosome Model

The details of the force field describing the interactions in the one-bead nucleosome model can be found in the work of Voltz et al. [3]. In this model applied to the nucleosome the potential energy function is constructed as:

$$E = E_{1-2} + E_{1-3} + E_{1-4} + E_{1-5} + E^{nonbonded} + E^{bp} \tag{11}$$

where harmonic terms:

$$E_{1-2} = \sum_{i,i+1} \frac{1}{2} k_{1-2} (r_{i,i+1} - r^o_{1-2})^2 \tag{12}$$

parameter	C_α beads	P beads
k_{1-2} $[\frac{kcal}{\mathring{molA}^2}]$	50.0	3.0
k_{1-3} $[\frac{kcal}{\mathring{molA}^2}]$	5.0	2.5
k_{1-4} $[\frac{kcal}{\mathring{molA}^2}]$	3.0	0.5
k_{bp} $[\frac{kcal}{\mathring{molA}^2}]$	n/a	0.6
α $[\frac{1}{\mathring{A}}]$	0.707	0.707
r_o $[\mathring{A}]$	9.5	17.6
$A_{P,C_\alpha}(r^o)$ $[\frac{kcal}{\mathring{molA}}]$	$4.0\cdot\exp(-\frac{r^o}{2.8})$	$2.0\cdot\exp(-\frac{r^o}{6.0})$
$R^{cut-off}$ $[\mathring{A}]$	12.0	20.0
$R_{max}^{cut-off}$ $[\mathring{A}]$	35.0	35.0

Table 2: Default parameters for the Ribosome model [2]

$$E_{1-3} = \sum_{i,i+2} \frac{1}{2} k_{1-3}(r_{i,i+2} - r^o_{1-3})^2 \tag{13}$$

$$E_{1-4} = \sum_{i,i+3} \frac{1}{2} k_{1-4}(r_{i,i+3} - r^o_{1-4})^2 \tag{14}$$

$$E_{1-5} = \sum_{i,i+4} \frac{1}{2} k_{1-5}(r_{i,i+4} - r^o_{1-5})^2 \tag{15}$$

describe the pseudo-bonded interactions between an ith pseudo-atom and its jth subsequent neighbour with r^o_{1-j+1} being the equilibrium distance between neighbours and k_{1-j+1} the force constant ($j = 1, 2, 3, 4$ for proteins and $j = 1, 2, 3$ for DNA). For proteins, the bonded C_α-C_α interactions are treated differently within α-helices and random coils (or loops). In α-helices, all pairs (up to the 4th nearest neighbour) are treated with harmonic potentials. In coils/loops interactions 1-4 and 1-5 are treated as nonbonded.

Nonbonded interactions, $E^{nonbonded}$, are divided into two kinds (according to the interaction distance), local and nonlocal, both described with Morse functions analogously to the Ribosome model presented above.

Interactions between the paired bases in DNA, E^{bp} are modelled with a harmonic potential.

Parameters of C_α-P interactions are derived on the basis of geometric averaging. The entire force field was parametrized based on all-atomic molecular dynamics simulations of the nucleosome particle [3]. Default parameter values are given in Table 3.

parameter	C_α beads (helices / coils)	P beads
k_{1-2} $[\frac{kcal}{mol\mathring{A}^2}]$	153.0 / 120.0	5.0
k_{1-3} $[\frac{kcal}{mol\mathring{A}^2}]$	20.0 / 0.9	1.5
k_{1-4} $[\frac{kcal}{mol\mathring{A}^2}]$	9.3 / n/a	1.0
k_{1-5} $[\frac{kcal}{mol\mathring{A}^2}]$	9.0 / n/a	n/a
r^o_{1-2} $[\mathring{A}]$	3.8 / 3.8	6.8
r^o_{1-3} $[\mathring{A}]$	5.5 / 6.4	13.0
r^o_{1-4} $[\mathring{A}]$	5.1 / n/a	18.0
r^o_{1-5} $[\mathring{A}]$	6.2 / n/a	n/a
k_{bp} $[\frac{kcal}{mol\mathring{A}^2}]$	n/a / n/a	0.6
r^o_{bp} $[\mathring{A}]$	n/a / n/a	19.3
α $[\frac{1}{\mathring{A}}]$	0.707	0.707
r_o $[\mathring{A}]$	10.0	26.3
$A_{P,C_\alpha}(r^o)$ $[\frac{kcal}{mol\mathring{A}}]$	$7.0 \cdot \exp(-\frac{r^o}{2.8})$	$\exp(-\frac{r^o}{6.0})$
$R^{cut-off}$ $[\mathring{A}]$	9.0	24.0
$R^{cut-off}_{max}$ $[\mathring{A}]$	30.0	30.0

Table 3: Default parameters of the Nucleosome model [3].

5.4 HIV-1 Protease Model

The one-bead force field for the HIV-1 protease was described in [4, 5]. The potential energy function, E consists of the following terms:

$$E = E_{1-2} + E_\theta + E_\phi + E^{nonbonded} \tag{16}$$

The pseudo-bond E_{1-2} term is given as a sum of harmonic terms:

$$E_{1-2} = \sum_{i,i+1} \frac{1}{2} k_{1-2} (r_{i,i+1} - r^o_{i,i+1})^2 \tag{17}$$

where the equilibrium bond length $r^o_{i,i+1}$ is taken from the configuration used to generate the force field. The functional form of the pseudo-bond angle term, E_θ is:

$$E_\theta = \sum_{\theta_i} \frac{1}{2} k_\alpha (\theta - \theta_\alpha)^2 - \sum_{\theta_i} \frac{1}{3} \frac{2k_\alpha + k_\beta}{\theta_\beta - \theta_\alpha} (\theta - \theta_\alpha)^3 + \sum_{\theta_i} \frac{1}{4} \frac{k_\alpha + k_\beta}{(\theta_\beta - \theta_\alpha)^2} (\theta - \theta_\alpha)^4 \tag{18}$$

where force constants k_α and k_β are associated with the two minima of the θ angle: $\theta_\alpha = 90[deg]$ - contracted α-helix like conformations and θ_β - extended β-sheet like conformations. This bistable potential provides for two minima and allows for a configurational change leading to a 20Å-wide opening of HIV-1 protease flaps covering the active site (see Figure 5). The dihedral term, E_ϕ is

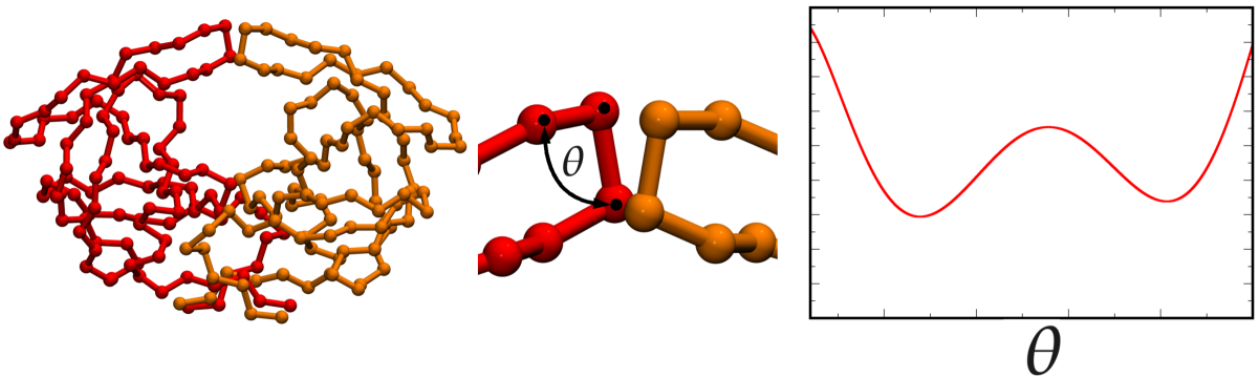


Figure 5: Left: HIV-1 protease backbone denoting two symmetric monomers. Middle: The pseudo-bond angle Θ. Right: Graphical representation of the functional form of the pseudo-bond angle potential showing two minima.

given with a sum of harmonic terms:

$$E_\phi = \sum_{\phi_i} \frac{1}{2} k_\phi (\phi_i - \phi^o_i)^2 \tag{19}$$

with equilibrium angle values ϕ^o_i taken from the initial structure.

Nonbonded interactions are treated analogously as in case of the Ribosome model.
Default parameters are given in Tables 4 and 5.

parameter	value
k_{1-2} $[\frac{kcal}{mol\mathring{A}^2}]$	50.0
k_{α} $[\frac{kcal}{mol \cdot rad^2}]$	38.0.0
k_{β} $[\frac{kcal}{mol \cdot rad^2}]$	17.0
k_{ϕ} $[\frac{kcal}{mol \cdot rad^2}]$	5.0
α $[\frac{1}{\mathring{A}}]$	0.707
r_o $[\mathring{A}]$	9.5
$A_{C_\alpha}(r^o)$ $[\frac{kcal}{mol\mathring{A}}]$	$6.0 \cdot \exp(-\frac{r^o}{2.8})$
$\mathrm{R}^{cut-off}$ $[\mathring{A}]$	8.0
$\mathrm{R}_{max}^{cut-off}$ $[\mathring{A}]$	20.0

Table 4: Default parameters applied in the HIV-1 Protease model [4, 5]

X	Gly	X	130
Pro	Gly	X	150
Ile, Leu	Gly	Thr	150
X	Ala	X	130
Gly	Ala	Asp, Thr	120
Ile, Leu	Ala	Lys, Glu	135
X	Asn	X	120
Met, Ile, Leu	Asn	Val, Leu	110
Ile, Leu	Asn	Lys, Glu	135
X	Gln	X	120
Pro, Tyr	Gln	Arg, Ile, Val	115
Gly	Gln	Leu	135
X	Glu	X	120
Lys, Leu, Ile, Met	Glu	Ala, Glu, Ile	115
X	His	X	130
Gly	Glu	Lys	120
X	Ile	X	120
Cys, Ile, Gly	Ile	Gly, Glu	110
Gly, Gly, Ile, Pro, Thr	Gly	Leu, Lys, Met, Asn, Lys	135
X	Leu	X	120
Asn, Leu, Trp	Leu	Thr, Asp	100
Glu, Pro	Leu	Val, Asn, Ser	135
X	Met	X	120
Asn, Ser	Met	Glu	140
X	Phe	X	120
Ile	Phe	Gly	135
X	Ser	X	120
Leu	Ser	Met	115
X	Thr	X	120
Val, Ile, Pro	Thr	Asp, Gly, Ile, Leu, Pro	115
Leu	Thr	Ala, Cys	135
X	Trp	X	120
Lys	Trp	Lys, Arg	135
X	Val	X	120
Arg, Pro, Leu	Val	Lys, Asn, Thr	135
X	Asp	X	130
X	Cys, Phe, Tyr	X	120
X	Arg, Pro	X	115

Table 5: Equilibrium values of θ_β angle ([deg]) for all the triplets of amino acids (any amino acid is denoted as X) in the HIV-1 Protease model [4, 5].

5.5 REACH Model

The REACH (Realistic Extension Algorithm via Covariance Matrix) model of Moritsugu and Smith [6] is a one-bead model that uses the following potential energy function (E):

$$E = \frac{1}{2}\sum_{i<j}^{N} k_{ij}(d_{ij} - d_{ij}^{o})^2 \tag{20}$$

where N is the number of pseudo-atoms, d_{ij} (d_{ij}^{o}) is the distance between the dynamical (equilibrium) coordinates of pseudo-atoms i and j, and k_{ij} is the force constant for the pseudo-bond between i and j. The REACH model assumes that the k_{ij} parameters are constant for virtual bonds 1–2, 1–3 and 1–4 (corresponding to k_{12}, k_{13}, k_{14}).

To model the nonbonded interactions a single exponential decay is used:

$$k_{nb}(r) = a \cdot e^{-b \cdot r} \tag{21}$$

with r the equilibrium bond length. The parametrization procedure based on the variance-covariance matrix obtained from an atomistic molecular dynamics simulation is described in detail in reference [6]. Table 6 gives default parameters of the REACH model; these parameters are applicable to proteins.

parameter	value
k_{1-2} $[\frac{kJ}{\mathring{A}^2}]$	712.0
k_{1-3} $[\frac{kJ}{\mathring{A}^2}]$	6.92
k_{1-4} $[\frac{kJ}{\mathring{A}^2}]$	32.0
a_{intra} $[\frac{kJ}{\mathring{A}^2}]$	2560.0
b_{intra} $[\frac{1}{\mathring{A}}]$	0.8
a_{inter} $[\frac{kJ}{\mathring{A}^2}]$	1630.0
b_{inter} $[\frac{1}{\mathring{A}}]$	0.772

Table 6: Default parameters of the REACH model [6]. Indices *intra* and *inter* denote intra- and inter-chain connections.

5.6 Nucleic Acid Models

The one–bead per nucleotide model of DNA was parametrized by Trovato and Tozzini based on the Boltmzann Inversion procedure of B-DNA molecules [7]. It can be applied towards simulation of linear and circular B-DNA structures, including plasmids. It was later extended towards RNA by

Leonarski et al.[8], allowing simulation of the dynamics of RNA helices, as well as tertiary structure prediction of RNAs with known secondary structure. This two tasks, presented in Figure 7 require different models, so two sets of RNA parameters are presented in Table 10. These models use the following potential energy function (E):

$$E = E_b + E_{ang} + E_{dih} + E_{bp} + E_{nb} \tag{22}$$

The first three terms in the above equation are intrastrand harmonic potentials

$$E_b(l) = \frac{1}{2}\sum_{bonds} k_b(l - l_0)^2 \qquad E_{ang}(\theta) = \frac{1}{2}\sum_{angles} k_\theta(\theta - \theta_0)^2 \qquad E_{dih}(\varphi) = \frac{1}{2}\sum_{dihedrals} k_\varphi(\varphi - \varphi_0)^2 \tag{23}$$

where l_0 is equilibrium distance, α_0 and φ_0 are respectively equilibrium planar and dihedral angle, k_b, k_θ, k_φ are bonded interaction force constants. For the RNA model the following term replaces the dihedral function:

$$E_{dih}(\varphi) = \sum_{dihedrals} k_\varphi(1 - \cos(\varphi - \varphi_0)) \tag{24}$$

where k_φ is half of the energy difference between the minimum and maximum and φ_0 is an equilibrium dihedral angle.

Default parameters are given in Table 7. Please note that papers describing nucleic acid force fields adopt a different convention for dihedral angle sign than RedMD. In the RedMD convention dihedral angles in helical conformation should have a negative value of dihedral angle φ_0. In the original papers [7, 8] the φ_0 equilibrium angles have positive values. Tables 7 and 10 present values prepared for use in the RedMD software.

Each Watson-Crick hydrogen bonding interaction is represented with three terms (Figure 6):

$$E_{bp} = E_{bp,0} + E_{bp,1} + E_{bp,2} \tag{25}$$

In the DNA model interactions within each of the above pseudopairs are described with a Morse with a barrier potential formula:

$$E_{bp,q}(r) = E_{0,q}\left[\left(1 - e^{-\alpha_q(r - r_{0,q})}\right)^2 - c_q\right]\mathrm{sw}_q(r) \tag{26}$$

where $\mathrm{sw}_q(r)$ denotes the switch function of the following form:

$$\mathrm{sw}_q(r) = \frac{1}{2}\varepsilon_q\left(1 - \tanh(\lambda_q(r - r_{1,q}))\right) \tag{27}$$

In the RNA model the Watson-Crick hydrogen bonding is implemented using a harmonic potential formula:

$$E_{bp,q}(r) = \frac{1}{2}k_{bp,q}(r - r_0)^2 \tag{28}$$

The last type of potential term is a nonbonded two-body potential connecting all pseudo-atom

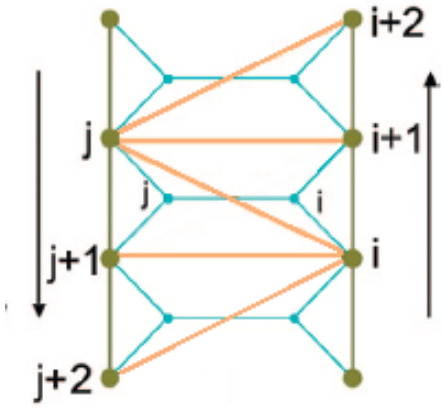


Figure 6: Schematic representation of the DNA double helix. Arrows indicate the order of residues along a strand. Orange lines denote interactions within the Watson-Crick base pairs. Each base pair is represented as a sum of three terms u_q, given with Equation 26. The interaction between a base pair formed by residues i^{th} and j^{th} is computed as a sum of terms u_0 (between residues i^{th} and j^{th}), u_1 (between residues i and $j+1$ or $i+1$ and j) and u_2 (between residues i and $j+2$ or $i+2$ and j). Parameters of these interactions are given in Table 8.

pairs that are not involved in any other interaction.
In the DNA model this term is implemented using a Morse with barrier: (Table 9)

$$E_{nb}(r) = E_{0,nb}\left[\left(1 - e^{-\alpha_{nb}(r-r_{0,nb})}\right)^2 - c_{nb}\right] \cdot (1 + 2\mathrm{sw}_2^{nb}(r)) \cdot \mathrm{sw}_1^{nb}(r) + 2A_{nb} \cdot \mathrm{sw}_2^{nb}(r) \tag{29}$$

where the switch functions are defined as:

$$\mathrm{sw}_q^{nb}(r) = \frac{1}{2}\varepsilon_q^{nb}\left(1 - \tanh(\lambda_q^{nb}(r - r_{q,nb}))\right) \tag{30}$$

The RNA model, on the other hand, uses a simple electrostatic term to describe the nonbonded interactions:

$$E_{nb}(r) = \frac{1}{4\pi\epsilon_0}\frac{q_i q_j}{\epsilon r} \tag{31}$$

where q_i and q_j are formal charges assigned to nucleotide pseudo-atoms ($q_i = q_j = -1$), ϵ_o is the electric permittivity of free space and ϵ is a relative electric permittivity; the parameter can be varied to account for shielding and other different effects present in the solution.

Two alternative forms of the nonbonded potential for the DNA model, presented in other papers [8, 9], are also implemented in the RedMD software: piecewise and double–well potential.

Piecewise potential gives a similar well and barrier behavior as in Eq. 29, however, it is based on a different philosophy. This functional form makes the parameters of the potential directly related to the shape of the function, by providing explicitly the potential well depth, barrier height,

well minimum position, barrier position and width. This potential is described in detail in the work of Leonarski et al. [8] The functional form is the following:

$$E_{nb}(r) = \begin{cases} E^0_{nb}[(\exp(-\alpha(r-r_0))-1)^2-1] & \text{if } r < r_0 \ , \\ \text{or} & \\ E^0_{nb} + (k(r-r_0)^2) & \text{if } r < r_0 \ , \\ (E^{bar}_{nb} - E^0_{nb})\exp(-\sigma_1(r-r_1)^2) + U^{bar}_0 & \text{if } r_0 < r < r_1 \ , \\ E^{bar}_{nb} & \text{if } r_1 < r < r_2 \ , \\ E^{bar}_{nb}\exp(-\sigma_2(r-r_2)^2) & \text{otherwise} \ , \end{cases} \tag{32}$$

where U_{nb} denotes the potential well depth, U^{bar}_{nb} is the energy barrier height, r_0 is the energy minimum position, r_1 is the barrier starting position, r_2 is the barrier end position, α (Morse implementation) or k (harmonic implementation) control the slope on the left side of the minimum, σ_1 and σ_2 control, respectively, the left and right slope of the barrier. To ensure the continuity, the $r_0 - r_1 > 3\frac{1}{\sqrt{2}\sigma_1}$ condition should be satisfied. For $r < r_0$ the user can choose if a Morse or harmonic implementation of the repulsive part are better in the investigated case.

The second alternative is the double–well potential. It describes well the situation, where the potential energy landscape suggests two energy minima, for example corresponding to two grooves in a nucleic acid helix. The implemented formula finds a minimum value of two Morse potentials, at the same time preventing a derivative discontinuity at a function intersection. The following functional form, described in the work of Trovato et al. [9], is implemented in the RedMD:

$$E_{nb}(r) = \frac{1}{2}\left[E_{m,1}(r) + E_{m,2}(r) - \Delta - \sqrt{(E_{m,1}(r) - E_{m,2}(r) + \Delta)^2 + \varepsilon^2} + \left(\Delta + \sqrt{\Delta^2 + \varepsilon^2}\right)\right] \tag{33}$$

where $E_{m,1}(r)$ and $E_{m,2}(r)$ are Morse functions defined as:

$$E_{m,q}(r) = E_{0,q}\left[\left(1 - \exp\left[-\alpha_q(r - r_{0,q})\right]\right)^2\right) - 1\right] \tag{34}$$

$E_{0,q}$, α_q and $r_{0,q}$ are respectively potential energy well depth, parameter controlling well width and minimum position, Δ is a parameter introducing an energy difference between tails of the two Morse functions and ε is a small non-zero number responsible for ensuring that derivative value does not have an infinite value at the function intersection.

All of the above potentials can be used in nucleic acid simulation. Technical details of including the potentials in the simulation file are described in Section 7.2.3.

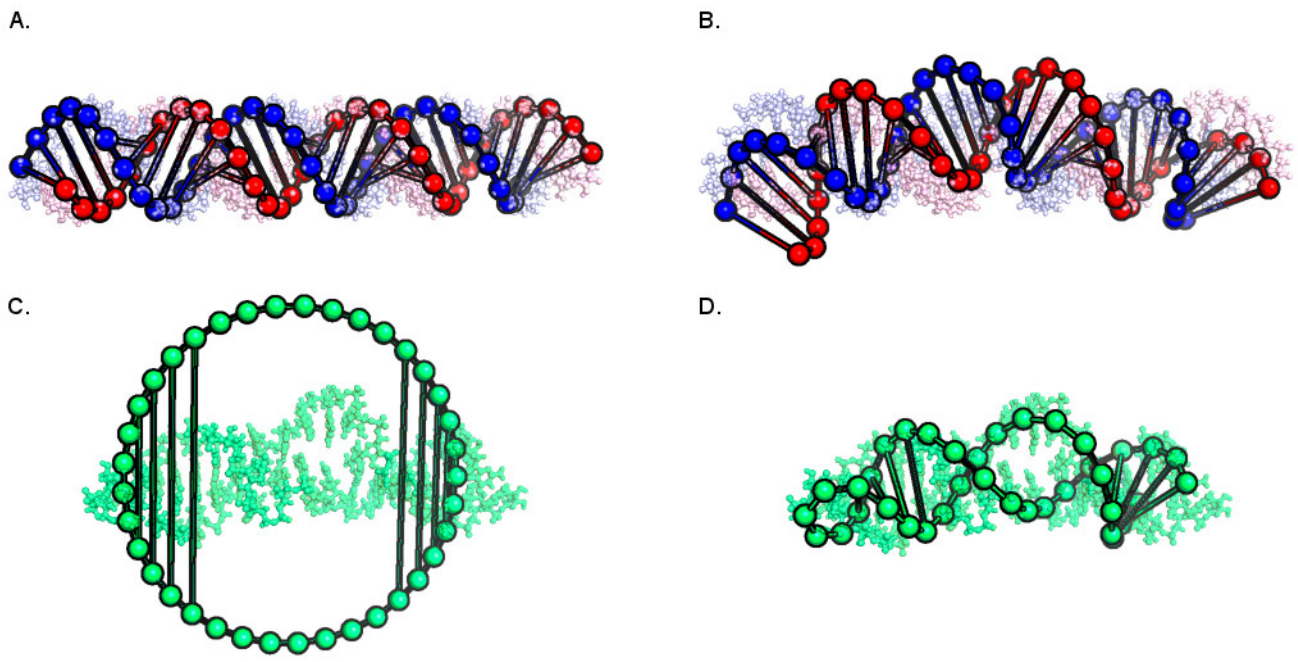


Figure 7: Subfigures **A** and **B** show conformational states from a RedMD simulation of a 35-bp. helix. Subfigures **C** and **D** show an RNA hairpin structure being simulated from an unfolded circular state (**C**) to a folded one (**D**).

parameter	value
k_b $[\frac{kcal}{mol\text{Å}^2}]$	12.5
k_α $[\frac{kcal}{mol\text{Å}^2}]$	25
k_φ $[\frac{kcal}{mol\text{Å}^2}]$	10
l_0 [Å]	6.66
α_0 [deg]	149
φ_0 [deg]	-19.1

Table 7: Default parameters for harmonic (bonded) potentials in the DNA model [7].

parameter	q=0	q=1	q=2
$E_{0,q}$ $[\frac{kcal}{mol\text{Å}^2}]$	2.8	2.8	4.8
α_q $[\text{Å}^{-1}]$	0.8	0.8	0.4
$r_{0,q}$ [Å]	19.25	17.93	15.3
$r_{1,q}$ [Å]	21.5	20.18	17.8
λ_q $[\text{Å}^{-1}]$	3.7	3.7	2
ε_q $[\frac{kcal}{mol}]$	1	1	1
c_q	0.4	0.4	0.18

Table 8: Default parameters for Watson-Crick interactions in the DNA model [7].

parameter	value
$E_{0',nb}$ $[\frac{kcal}{mol\text{Å}^2}]$	75
α_{nb} $[\text{Å}^{-1}]$	0.014
$r_{0,nb}$ [Å]	12
$r_{1,nb}$ [Å]	15.7
$r_{2,nb}$ [Å]	16.5
λ_1^{nb} $[\text{Å}^{-1}]$	1
λ_2^{nb} $[\text{Å}^{-1}]$	3
ε_1^{nb} $[\frac{kcal}{mol}]$	1
ε_2^{nb} $[\frac{kcal}{mol}]$	0.77
c_{nb}	0.007
A_{nb}	0.4

Table 9: Default parameters for nonbonded interactions in the DNA model [7].

FF parameter		Equilibrium dynamics	Structure prediction
r_0	[Å]	5.92	5.85
k_b	$[\frac{\text{kcal}}{\text{mol Å}^2}]$	2.56	1.98
α_0	[deg]	152.4	152.6
k_α	$[\frac{\text{kcal}}{\text{mol deg}^2}]$	24.4	68.6
φ_0	[deg]	−17.8	−17.1
k_φ	$[\frac{\text{kcal}}{\text{mol deg}^2}]$	14.0	11.2
$r_{0,0}$	[Å]	18.97	18.29
$k_{bp,0}$	$[\frac{\text{kcal}}{\text{mol Å}^2}]$	1.44	7.88
$r_{0,1}$	[Å]	17.61	17.86
$k_{bp,1}$	$[\frac{\text{kcal}}{\text{mol Å}^2}]$	4.71	11.4
$r_{0,2}$	[Å]	16.22	15.86
$k_{bp,2}$	$[\frac{\text{kcal}}{\text{mol Å}^2}]$	2.84	10.2
ϵ	[ ]	76.0	34.3

Table 10: FF parameters for the RNA model optimized using the evolutionary algorithm[8].

5.7 βA-40 Model

This two-bead model was developed by Mukherjee and Bagchi [10, 11] to study the dynamics of Alzheimer β-amyloid peptide. Every amino acid is mimicked by two atoms – a backbone atom (C_α) and a side atom (C_β), except for glycine which is represented by a single backbone atom. The total potential energy of the protein is divided into the following terms

$$E = E_B + E_\theta + E_T + E_{LJ} + E_{helix} \tag{35}$$

All constants mentioned in the following brief description of the potential terms are gathered in Tables 11 and 12. E_B stands for the energy of bond vibrations and is equal to

$$E_B = \frac{1}{2}K_r \sum_{i=2}^{N}(r_{i,i-1} - r_0)^2 + \frac{1}{2}K_r^s \sum_{i=1}^{N}(r_{i,i}^s - r_0^s)^2. \tag{36}$$

where N is the number of amino acids in the protein, r_0 and $r_{i,i-1}$ are the equilibrium and current bond lengths between two C_α atoms, while r_0^s and $r_{i,i}^s$ between C_α and C_β atoms. The values of r_0^s were taken from TABLE I of reference [12]. E_θ is the bending potential energy modeled with harmonic terms

$$E_\theta = \frac{1}{2}K_\theta \sum_{i=2}^{N-1}(\theta_{i-1,i,i+1} - \theta_0)^2 + \frac{1}{2}K_\theta \sum_{i=2}^{N}(\theta_{i-1,i,i}^s - \theta_0^s(i))^2 + \frac{1}{2}K_\theta \sum_{i=1}^{N-1}(\theta_{i,i,i+1}^s - \theta_0^s(i))^2 \tag{37}$$

where $\theta_{i-1,i,i+1}$ and $\theta_{i-1,i,i}^s$ are the angles between three C_α atoms and between two C_α and one C_β side atom, respectively. The torsional potential takes the form,

$$E_T = \epsilon_T \sum_\theta \frac{1}{2}[1 + cos(3\phi)], \tag{38}$$

where the sum runs over the dihedral angles in the molecule. Nonbonded interactions, i.e. the interactions between atoms not linked with the bonded or bending potentials, are modeled with Lennard-Jones potentials

$$E_{LJ} = 4\sum_{i,j}\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] \tag{39}$$

where $\sigma_{ij} = \frac{1}{2}(\sigma_{ii} + \sigma_{jj})$, and σ_{ii} is the size of the i-th atom. $\epsilon_{ij} = \sqrt{\epsilon_{ii}\epsilon_{jj}}$ and ϵ_{ii} is the interaction parameter derived from amino acid hydropathies H_i in the following way

$$\epsilon_{ii} = \epsilon_{min} + (\epsilon_{max} - \epsilon_{min})\frac{H_i - H_{min}}{H_{max} - H_{min}}. \tag{40}$$

Atomic sizes ($\sigma_{C\alpha}$) and their interaction parameters ($\epsilon_{C\alpha}$) are set constant and independent of the amino acid type. The force field also includes the helix potential between the C_α atoms

$$E_{helix} = \frac{1}{2}\sum_{i=4}^{N-5}\left[K_i^{1-3}(r_{i,i+2} - r_h)^2\right] + \frac{1}{2}\sum_{i=4}^{N-6}\left[K_i^{1-4}(r_{i,i+3} - r_h)^2\right] \tag{41}$$

where $K_i^{1-3} = \frac{1}{3}[K_i + K_{i+1} + K_{i+2}]$ and $K_i^{1-4} = \frac{1}{4}[K_i + K_{i+1} + K_{i+2} + K_{i+3}]$, with the condition that $K_i^{1-4}, K_i^{1-3} \geq 0$.

parameter	value
K_r $[\frac{kJ}{mol\text{Å}^2}]$	43.0
K_r^s $[\frac{kJ}{mol\text{Å}^2}]$	8.6
r_0 [Å]	3.81
K_θ $[\frac{kJ}{molrad^2}]$	10.0
θ_0 [°]	96
H_{min}	-4.5
H_{max}	4.5
ϵ_{min} $[\frac{kJ}{mol}]$	75
ϵ_{max} $[\frac{kJ}{mol}]$	75
r_h [Å]	5.5
$\epsilon_{C\alpha}$ $[\frac{kJ}{mol}]$	0.05
$\sigma_{C\alpha}$[Å]	1.8

Table 11: Default parameters for interactions in the β40A model.

5.8 Protein Electrostatics - Coulomb Sum with Distance Dependent Dielectric Constant

In RedMD one can extend all protein-applicable force fields by adding electrostatic Coulomb interactions with a distance dependent dielectric constant

$$E_{coul} = \frac{1}{4\pi\epsilon_0} \sum_{i<j}^{N} \frac{q_i q_j}{\epsilon(r_{ij}) r_{ij}} \tag{42}$$

with

$$\epsilon(r_{ij}) = \epsilon \cdot r_{ij} \tag{43}$$

where ϵ_0 is vacuum permittivity, q_i and q_j are formal charges assigned to pseudo-atoms corresponding to titratable amino acids (see Table 13) and ϵ is a constant parameter (with a default value of 4). The potential form of Eq. 42 is based on the assumption that the electrostatic forces are effectively 'screened' in real systems. This effect is approximated by introducing a dielectric term that increases with the distance between interacting charges. The effective range of electrostatic interactions is defined with two cut-off distances (i.e. electrostatic terms are evaluated for a

residue	H_i	σ_{ii}	$\theta_0^s(i)$	r_0^s	K_i
ALA	1.8	4.6	121.9	0.77	17.20
VAL	4.2	5.8	121.7	1.49	6.71
LEU	3.8	6.3	118.1	2.08	13.59
ILE	4.5	6.2	118.9	1.83	10.15
CYS	2.5	5.0	113.7	1.38	5.50
MET	1.9	6.2	113.1	2.34	13.07
PRO	1.6	5.6	81.9	1.42	-37.15
PHE	2.8	6.8	118.2	2.97	7.91
TYR	-1.3	6.9	110.0	3.36	8.08
TRP	-0.9	7.2	118.4	3.58	8.77
ASP	-3.5	5.6	121.2	1.99	9.80
ASN	-3.5	5.7	117.9	1.98	6.02
GLN	-3.5	6.1	118.0	2.58	10.49
HIS	-3.2	6.2	118.2	2.78	7.57
GLU	-3.5	6.1	118.2	2.63	14.45
SER	-0.8	4.8	117.9	1.28	8.60
THR	-0.7	5.6	117.1	1.43	5.85
ARG	-4.5	6.8	121.4	3.72	13.59
LYS	-3.9	6.3	122.0	2.94	12.73
GLY	-	-	-	-	0.00

Table 12: Default values of parameters used in the β40A Model for different amino acids.

given pair of charges if the distance between them falls between the two cut-off values), similarly as in case of nonlocal interactions.

5.9 External Fields

Apart from molecular force fields RedMD allows one to use external fields. Currently, the *containing* or *bounding sphere* external field is implemented. It can be applied when simulating many molecules and has the following form:

$$\vec{F} = A \cdot (R_o - r)^{-n} \hat{e} \quad \text{for} \quad r > R_{cut} \tag{44}$$

This field assures that the molecules of the system do not move away from the system's center of mass further than the containing sphere radius, R_o. $\vec{F}$ is a radial force acting on a given molecule in the region between the containing sphere radius (R_o) and the user-specified cut-off (R_{cut}). Parameters A and n ($n \geq 0$) allow the user to modulate the amplitude and radial dependence of $\vec{F}$.

5.10 User provided force fields

RedMD applicability is not limited to a described set of predefined force fields, but allows the user to extend this set by providing their own definition of the functional form. For the nonbonded or bonded two–body distance dependent potential the user can include in the code the analytical

residue	$q^{protonated}$ [e]	$q^{deprotonated}$ [e]
ASP	0.0	-1.0[*]
GLU	0.0	-1.0[*]
LYS	1.0[*]	0.0
ARG	1.0[*]	0.0
HIS	1.0[*]	0.0
TYR	0.0[*]	-1.0
N-terminal	1.0[*]	0.0
C-terminal	0.0	-1.0[*]

Table 13: Formal charges assigned to pseudo-atoms. Only pseudo-atoms corresponding to titratable amino acids bear non-zero charges. Default values are denoted with [*].

expression (see page 7.2.3 for technical implementation details). Additional nonbonded potential can be also provided in a tabularized form.

6 Methods

In this section we give a brief description of molecular dynamics algorithms implemented in the RedMD package. The technique of molecular dynamics simulations of biomolecules is described in detail for example in [13, 14].

6.1 Integration Algorithms

Two variants of the Verlet integrator [15] for solving the Newton's equation of motion

$$\frac{d\vec{p}}{dt} = \vec{F} \tag{45}$$

are currently implemented: *velocity Verlet* [16] and *leapfrog* [17]. Both variants possess favorable numerical properties like symplectiness and time-reversibility [18] and are appropriate to produce the microcanonical ensemble.

6.1.1 Velocity Verlet

Velocity Verlet algorithm transforms one point in phase space to the next

$$\{\vec{r}_i(t), \vec{p}_i(t)\} \rightarrow \{\vec{r}_i(t+\Delta t), \vec{p}_i(t+\Delta t)\} \tag{46}$$

where index i runs over system's pseudo-atoms with masses m_i, $\vec{r}$ and $\vec{p}$ are the position and momentum vectors and Δt is the integration step. The velocity Verlet integration scheme can be divided into the following steps:

`calculate positions at time` $t+\Delta t$ `and momenta at time` $t+\frac{1}{2}\Delta t$ `for all pseudo-atoms in the system:`

$$\vec{r}_i(t+\Delta t) = \vec{r}_i(t) + \frac{\vec{p}_i(t)}{m_i}\Delta t + \frac{1}{2}\frac{\vec{F}_i(t)}{m_i}\Delta t^2$$

$$\vec{p}_i(t + \frac{1}{2}\Delta t) = \vec{p}_i(t) + \frac{1}{2}\vec{F}_i(t)\Delta t$$

calculate forces acting on pseudo-atoms at time $t + \Delta t$, $\vec{F}_i(t + \Delta t)$

calculate momenta at time $t + \Delta t$ for all pseudo-atoms in the system:

$$\vec{p}_i(t + \Delta t) = \vec{p}_i(t + \frac{1}{2}\Delta t) + \frac{1}{2}\vec{F}_i(t + \Delta t)\Delta t$$

6.1.2 Leapfrog

The leapfrog propagation scheme transforms one point in phase space to the next as:

$$\{\vec{p}_i(t-\frac{1}{2}\Delta t), \vec{r}_i(t)\} \rightarrow \{\vec{p}_i(t+\frac{1}{2}\Delta t), \vec{r}_i(t+\Delta t)\} \tag{47}$$

To generate molecular dynamics trajectory using the *leapfrog* algorithm the following scheme is used:

`calculate momenta at time` $t-\frac{1}{2}\Delta t$ `for all pseudo-atoms in the system using Taylor expansion:`

$$\vec{p}_i(t-\frac{1}{2}\Delta t) = \vec{p}_i(t) - \frac{1}{2}\vec{F}_i(t)\Delta t$$

and then

`calculate momenta at time` $t+\frac{1}{2}\Delta t$ `for all pseudo-atoms in the system:`

$$\vec{p}_i(t+\frac{1}{2}\Delta t) = \vec{p}_i(t-\frac{1}{2}\Delta t) + \vec{F}_i(t)\Delta t$$

`calculate positions at time` $t+\Delta t$ `for all pseudo-atoms in the system:`

$$\vec{r}_i(t+\Delta t) = \vec{r}_i(t) + \frac{\vec{p}_i(t+\frac{1}{2}\Delta t)}{m_i}\Delta t$$

`calculate forces acting on pseudo-atoms at time` $t+\Delta t$, $\vec{F}_i(t+\Delta t)$

6.2 Temperature Control

Two algorithms for performing deterministic, constant temperature molecular dynamics simulations are implemented in RedMD: *scaling of momentum* and *weak coupling thermostat* proposed by Berendsen [19].

6.2.1 Simple Momentum Scaling

Kinetic temperature T is fixed at the desired value T_o by scaling the momentum vector at each integration step so that:

$$\vec{p}(t+\Delta t) \leftarrow \alpha\vec{p}(t) \tag{48}$$

with α defined as:

$$\alpha = \sqrt{\frac{T_o}{T}} \tag{49}$$

and with instantaneous kinetic temperature T computed from the equipartition theorem. One should note that the above algorithm is the most artificial approach for temperature scaling [20].

6.2.2 Berendsen Thermostat

The Berendsen approach for temperature scaling mimics the diffusive process with frictional coefficient γ defined as:

$$\gamma = \frac{1}{2\tau}\left(1 - \frac{T_o}{T}\right) \tag{50}$$

where T is the instantaneous kinetic temperature and τ is the time constant of the coupling to a heat bath (thermostat). As a result, at each integration step the momentum vector is scaled with the factor:

$$\alpha = \sqrt{1 - \frac{\Delta t}{\tau}\left(1 - \frac{T_o}{T}\right)} \tag{51}$$

If the factor τ is large compared to the time step (Δt), a microcanonical ensemble is realized; when τ equals Δt a simple momentum scaling described above is approached. Effectively, the flexible parameter τ controls the rate at which the target temperature is reached. We suggest the value of τ so chosen, that $\frac{\Delta t}{\tau} \sim 0.001$.

6.3 Langevin Dynamics

RedMD considers the simplest form of the Langevin equation [21, 22]:

$$\frac{d\vec{p}_i}{dt} = \vec{F}_i - \gamma\vec{p}_i + \vec{R}_i(t) \tag{52}$$

where index i runs over pseudo-atoms with masses m_i, γ is the damping constant (or collision parameter), and $\vec{R}_i(t)$ is a random vector with the Gaussian distribution function whose average value is zero and whose variance-covariance is:

$$< \vec{R}_i(t)\vec{R}_i(t') >= 2\gamma k_B T_o m_i \delta(t - t') \tag{53}$$

where T_o is the desired simulation temperature.

RedMD uses Brooks-Brünger-Karplus (BBK) integration scheme [23] appropriate for the small-γ regime:

`calculate momenta at time` $t + \frac{1}{2}\Delta t$ `for all pseudo-atoms in the system:`

$$\vec{p}_i(t + \frac{1}{2}\Delta t) = \vec{p}_i(t) + \frac{\Delta t}{2}\left(\vec{F}_i(t) - \gamma m_i \vec{p}_i(t) + \vec{R}_i(t)\right)$$

`calculate positions at time` $t + \Delta t$ `for all pseudo-atoms in the system:`

$$\vec{r}_i(t + \Delta t) = \vec{r}_i(t) + \Delta t \frac{\vec{p}_i(t + \frac{1}{2}\Delta t)}{m_i}$$

`calculate random vector using independent Gaussian numbers` Z_i `with zero mean and variance`

one:

$$\vec{R}_i(t+\Delta t) = \sqrt{\frac{2\gamma m_i k_B T_o}{\Delta t}} Z_i$$

calculate new forces at time $t+\Delta t$, $\vec{F}_i(t+\Delta t)$

calculate momenta at time $t+\Delta t$ for all pseudo-atoms in the system:

$$\vec{p}_i(t+\Delta t) = \vec{p}_i(t+\frac{1}{2}\Delta t) + \frac{\Delta t}{2}\left(\vec{F}_i(t+\Delta t) - \gamma\vec{p}_i(t+\Delta t) + \vec{R}_i(t+\Delta t)\right)$$

6.4 Brownian Dynamics

In Brownian dynamics simulations pseudo-atoms are assigned a diffusion coefficient D computed as:

$$D = \frac{k_{BT}}{6\pi\eta R_H} \tag{54}$$

where η is solvent viscosity and R_H is the hydrodynamic radius of a pseudo-atom. The basic Brownian dynamics propagation scheme derived by Ermak and McCammon [24] is used:

calculate random number $\vec{R}_i$ with the Gaussian distribution function whose average value is zero and the variance is 1

multiply a random vector by a factor $\sqrt{2D\Delta t}$

calculate force $\vec{F}_i(t)$ acting on each pseudo-atom

calculate positions of pseudo-atoms at time $t+\Delta t$:

$$\vec{r}_i(t+\Delta t) = \vec{r}_i + \frac{D}{k_b T_o}\vec{F}_i\Delta t + \vec{R}_i$$

6.5 Minimization - L-BFGS

The optimization algorithm implemented in RedMD is the limited memory BFGS (L-BFGS) [25]. L-BFGS belongs to a class of quasi-Newton methods, i.e. the Hessian matrix is not explicitly computed or stored during optimization. This feature makes L-BFGS particularly well suited for optimization problems with a large number of dimensions. To approximate the Hessian matrix (and avoid its explicit computation and storing), the L-BFGS method uses the Broyden-Fletcher-Goldfarb-Shanno update.

6.6 Monte Carlo conformational search in the dihedral angle space

RedMD uses the algorithm of Li and Scheraga [26] for solving the multiple-minima problem in protein folding. The algorithm consists of three steps.

```
step 1:  a random change of molecule's configuration in one or more randomly selected
dihedral angles (from all the variable dihedral angles) is performed

step 2:  the random conformation generated in step 1 is subjected to minimization
using the L-BFGS algorithm to reach the nearest local minimum

step 3:  this local minimum is examined by the Metropolis criterion [27] and compared
with the previously accepted local minimum to update or reject the current configuration
```

6.7 Holonomic Constraints

During molecular dynamics simulations it is possible to "freeze" the highest-frequency motions in order to use a larger integration step. To satisfy the bond length constraints (applicable only to C_α-C_α bonds), two algorithms are implemented: SHAKE [28] and its variant termed RATTLE [29]. In the current RedMD version, it is possible to use the SHAKE algorithm with the leapfrog integrator, in Brownian dynamics simulations and constant temperature simulations with the Berendsen thermostat. The RATTLE algorithm is integrated with the velocity Verlet integrator.

7 Input & Output Files

In this section we describe the input files necessary to perform simulations and the output files that are created upon RedMD execution.

7.1 Simulation Control File

This section gives a description of the main RedMD control file used to set up molecular dynamics calculations. The main control file is a plain text format file containing different keywords which define the course of RedMD calculations. The syntax is the following:

```
keyword value
```

and each line must be terminated with an `EOL` character. The program reads only two words from each line and interprets the first one as the keyword name and the second as its value. The rest of the line is omitted and treated as a comment:

```
BD yes           perform Brownian dynamics simulation
dt 0.01           using time step of 0.01[ps]
number_of_steps 10000          perform 10000 integration steps
```

Lines beginning with ∗, // or # are also ignored (e.g. lines 1, 3 and 5 of the following example):

```
# perform Brownian dynamics simulation
BD yes
* using time step of 0.01[ps]
dt 0.01
// perform 10000 integration steps
number_of_steps 10000
```

See section 9 for a full list of acceptable keywords and section 10 for exemplary molecular dynamics input files.

7.2 XML-based Topology & Structure Files

In order to store information on molecular systems, RedMD uses the XML (Extensible Markup Language) (http://www.w3.org/XML). The topology (`*.txml`) and structure (`*.sxml`) files conform to all the rules of a well-formed XML documents. By using the XML format, we benefit from representing the topology and structure data as trees and utilize useful features of XML such as inheritance and subtyping.

7.2.1 Topology

The RedMD topology (`*.txml`) fully defines the geometry of the molecule. The file contains information on pseudo-atoms, their positions, connectivity within linear chains and, optionally, the

secondary base pairing connections between nucleotide bases. An example of the topology file created with RedMD for a fictitious model compound is presented below:

```
<?xml version="1.0"?>
 <STRUCTURE>
  <COMMENT>
  Generated with: ../../RedMD binary/bin/RedMD extractPDB test.pdb
  </COMMENT>
    <MOLECULE molId="mol1">
     <ATOM id="1" name="CA" x="48.28" y="55.85" z="-24.70" resName="ALA" chainID="A" resSeq="1"/>
      <BOND idAtom1="1" idAtom2="2"/>
     <ATOM id="2" name="CA" x="46.55" y="58.58" z="-22.62" resName="ARG" chainID="A" resSeq="2"/>
      <BOND idAtom1="2" idAtom2="3"/>
     <ATOM id="3" name="CA" x="46.17" y="57.64" z="-18.91" resName="THR" chainID="A" resSeq="3"/>
      <BOND idAtom1="3" idAtom2="4"/>
     <ATOM id="4" name="CA" x="44.09" y="60.39" z="-17.22" resName="LYS" chainID="A" resSeq="4"/>
      <BOND idAtom1="4" idAtom2="5"/>
     <ATOM id="5" name="CA" x="40.26" y="60.45" z="-16.83" resName="GLN" chainID="A" resSeq="5"/>
      <BOND idAtom1="5" idAtom2="6"/>
   </MOLECULE>
 </STRUCTURE>
```

The first line of the presented file contains the XML declaration `<?xml version="1.0"?>` that gives the version of the applied XML. According to the rules of an XML document, the `*.txml` file has exactly one root element (or root node). Therefore, topology data are enclosed between the root start-tag given with `<STRUCTURE>` and the root end-tag `</STRUCTURE>`. Comments are enclosed between tags `<COMMENT>` and `</COMMENT>`. Non-empty element (or node) `MOLECULE` (tags `<MOLECULE>` and `</MOLECULE>`) with the attribute `molId` whose value given in double quotes (””) is `mol1`, denotes individual molecules and contains nested empty elements `<ATOM />` and `<BOND />`. The attributes of empty elements `<ATOM />` are:

unique atom indexes, `id`
names of atoms, `name`
positions of atoms, `x`, `y`, `z`
names of corresponding residues, `resName`
a chain to which a given atom belongs, `chainID`
a position of a residue in the sequence, `resSeq`

Additional attributes of `ATOM` elements are: `serial`, `altLoc`, `iCode`, `occupancy`, `tempFactor`, consistently with Protein Data Bank annotations [30].

Attributes `idAtom1` and `idAtom2` of empty elements `<BOND />` describe the connectivity within molecular chains.

Topology files may also contain information on paired nucleotide bases (this information can be extracted from PDBML files, see Section 8), as shown below for a model molecule consisting of two continuous linear chains connected via hydrogen bonds between base pairs. `GROUP` elements are introduced for storing other XML elements: 1-2 connections (`label=''backbone''`) and paired bases (`label=''basePairs''`)

```
<?xml version="1.0"?>
 <STRUCTURE>
 <COMMENT>
 Generated with: ../../RedMD binary/bin/RedMD extractPDBXML test.xml.gz
 </COMMENT>
   <MOLECULE molId="mol1">
     <GROUP label_entity_id="1" label_asym_id="A" chainID="A">
       <GROUP label="atoms">
        <ATOM id="1" name="P" x="-50.76" y="76.72" z="327.18" resSeq="2" resName="U"/>
        <ATOM id="2" name="P" x="-50.82" y="73.75" z="332.17" resSeq="3" resName="U"/>
        <ATOM id="1518" name="P" x="18.80" y="53.59" z="308.17" resSeq="1519" resName="U"/>
        <ATOM id="1519" name="P" x="20.26" y="52.40" z="301.46" resSeq="1520" resName="U"/>
       </GROUP>
       <GROUP label="backbone">
        <BOND idAtom1="1" idAtom2="2"/>
        <BOND idAtom1="1518" idAtom2="1519"/>
       </GROUP>
     </GROUP>
     <GROUP label_entity_id="2" label_asym_id="B" chainID="B">
      <GROUP label="atoms">
       <ATOM id="1520" name="P" x="0.01" y="161.51" z="290.81" resSeq="1" resName="G"/>
       <ATOM id="1521" name="P" x="5.95" y="163.40" z="287.33" resSeq="2" resName="C"/>
       <ATOM id="1591" name="P" x="13.66" y="151.00" z="294.84" resSeq="72" resName="C"/>
       <ATOM id="1592" name="P" x="15.03" y="156.15" z="296.53" resSeq="73" resName="A"/>
      </GROUP>
      <GROUP label="backbone">
       <BOND idAtom1="1520" idAtom2="1521"/>
       <BOND idAtom1="1590" idAtom2="1591"/>
       <BOND idAtom1="1591" idAtom2="1592"/>
      </GROUP>
    </GROUP>
    <GROUP label="basePairs">
       <BOND idAtom1="1520" idAtom2="1591"/>
       <BOND idAtom1="1521" idAtom2="1590"/>
    </GROUP>
  </MOLECULE>
</STRUCTURE>
```

7.2.2 Structure

The RedMD structure (`*.sxml`) files contain information on the topology and all the interactions between psudo-atoms in the studied system.

The structure files are created automatically within the framework of a particular force field (see Sections 5 and 8.2), however they can be later modified according to one needs taking into consideration rules presented below.
An exemplary structure (`*.sxml`) file is given below:

```
<?xml version="1.0"?>
 <STRUCTURE>
 <COMMENT>
  Generated with: ../../RedMD binary/bin/RedMD genModel MODEL test.txml
 </COMMENT>
  <NONBONDED>
   <NLMORSE cutoff="20.00">
    <PAIR type1="CA" type2="CA" alpha="0.707000" E0="0.35" l0="9.50"/>
   </NLMORSE>
  </NONBONDED>
  <MOLECULE molId="mol1">
   <ATOM id="1" name="CA" x="48.28" y="55.85" z="-24.70" resName="ALA" chainID="A" resSeq="1" m="71"/>
   <ATOM id="2" name="CA" x="46.55" y="58.58" z="-22.62" resName="ARG" chainID="A" resSeq="2" m="156"/>
   <ATOM id="3" name="CA" x="46.17" y="57.64" z="-18.91" resName="THR" chainID="A" resSeq="3" m="101"/>
   <ATOM id="4" name="CA" x="44.09" y="60.39" z="-17.22" resName="LYS" chainID="A" resSeq="4" m="128"/>
   <ATOM id="5" name="CA" x="40.26" y="60.45" z="-16.83" resName="GLN" chainID="A" resSeq="5" m="128"/>
   <ATOM id="6" name="CA" x="39.78" y="57.35" z="-19.07" resName="THR" chainID="A" resSeq="6" m="101"/>
   <ATOM id="7" name="CA" x="41.78" y="58.06" z="-18.07" resName="ALA" chainID="A" resSeq="7" m="71"/>
   <BOND idAtom1="1" idAtom2="2" k="50.00" l0="3.84"/>
   <BOND idAtom1="2" idAtom2="3" k="50.00" l0="3.85"/>
   <BOND idAtom1="3" idAtom2="4" k="50.00" l0="3.83"/>
   <BOND idAtom1="4" idAtom2="5" k="50.00" l0="3.84"/>
   <BOND idAtom1="5" idAtom2="6" k="50.00" l0="3.85"/>
   <BOND idAtom1="6" idAtom2="7" k="50.00" l0="3.85"/>
   <DIH_HARM idAtom1="1" idAtom2="2" idAtom3="3" idAtom4="4" alpha0="173.98" kAlpha="5.00"/>
   <DIH_HARM idAtom1="2" idAtom2="3" idAtom3="4" idAtom4="5" alpha0="-90.65" kAlpha="5.00"/>
   <DIH_HARM idAtom1="3" idAtom2="4" idAtom3="5" idAtom4="6" alpha0="7.55" kAlpha="5.00"/>
   <MORSE idAtom1="2" idAtom2="6" alpha="0.70" l0="7.74" E0="0.38"/>
   <MORSE idAtom1="2" idAtom2="7" alpha="0.70" l0="10.63" E0="0.58"/>
  </MOLECULE>
 </STRUCTURE>
```

The presented `*.sxml` file contains exactly one root element delimited with tags `<STRUCTURE>` `</STRUCTURE>`. The `NONBONDED` element stores information on nonlocal interactions. Here, the `type1` and `type2` attribute values enclosed within the element `PAIR` point to CA pseudo-atoms. Their generic nonbonded interactions are modelled with the Morse potential within the cut-off distance of 20$\mathring{A}$ (`<NLMORSE cutoff="20.00">`). The parameters of the Morse function are defined with attribute values `alpha`, `E0` and `l0` of the `PAIR` element.
The `MOLECULE` element has an attribute `molId` with an unique value `mol1`. This uniqueness is important in case of linking together `*.sxml` files of two different molecules. The elements nested within `<MOLECULE></MOLECULE>` tags are:

`ATOM` - pseudo-atoms, with attributes specifying indexes (`id`), names (`name`), positions (`x`,`y`,`z`), residues (`resName`), chains (`chainID`), sequence within chains (`resSeq`) and masses (`m`)
`BOND` - harmonic bonds between pseudo-atoms, with attributes identifying connected pseudo-atoms (`idAtom1`, `idAtom2`), force constants (`k`) and equilibrium values (`l0`) for harmonic bond lengths
`DIH_HARM` - dihedral angles modelled with harmonic potentials, with attributes identifying involved atoms (`idAtom1`, `idAtom2`, `idAtom3`, `idAtom4`), force constants (`kAlpha`) and equilibrium values

(alpha0)
MORSE - local nonbonded interactions described with Morse functions, between atoms given with attributes idAtom1 and idAtom2, with the shape of Morse functions defined with attributes alpha, l0 and E0

Below we give a few more examples of the content of the *.sxml file:

The harmonic bond between atoms 1 and 2 with the force constant k and equilibrium length l0, constrained with the SHAKE algorithm is defined in the following lines:

```
<ATOM id="1" name="CA" x="51.84" y="59.78" z="-6.81" resName="PRO" chainID="A" resSeq="1" m="97"/>
<ATOM id="2" name="CA" x="51.57" y="59.56" z="-2.99" resName="GLN" chainID="A" resSeq="2" m="128"/>
<BOND idAtom1="1" idAtom2="2" k="50.00" l0="3.83" shake="true"/>
```

Local nonbonded interaction is described with the Morse term. Effectively, the local and non-local nonbonded interactions are modeled with a function (see Equations 7 and 8):

$$E(l) = E_o(1 - e^{-\alpha(l-l_o)})^2 \tag{55}$$

where E_o is the dissociation constant. The Morse potential interaction between atoms 10 and 22 is represented as:

```
<ATOM id="10" name="CA" x="40.90" y="53.61" z="-6.53" resName="LEU" chainID="A" resSeq="10" m="113"/>
<ATOM id="22" name="CA" x="40.64" y="49.60" z="-10.01" resName="ALA" chainID="A" resSeq="22" m="71"/>
<MORSE idAtom1="10" idAtom2="22" alpha="0.70" l0="5.32" E0="0.91"/>
```

Pseudo-bond angle terms of the bonded interactions used in the HIV-1 Protease model (see Eq. 18) and described with the function:

$$E(\theta) = k_\theta 1(\theta - \theta_0)^2 + k_\theta 2(\theta - \theta_0)^3 + k_\theta 3(\theta - \theta_0)^4 \tag{56}$$

are realized in the *.sxml file as:

```
<ANGLE_QUAR idAtom1="1" idAtom2="2" idAtom3="3" theta0="90.0" kTheta1="38.0" kTheta2="-77.6" kTheta3="90.6"/>
```

The bending potential of the following form

$$E(\theta) = k_\theta(\theta - \theta_0)^2, \tag{57}$$

the torsional potential

$$E(\phi) = A\sum_{\varphi}[1 + cos(N\varphi)], \tag{58}$$

the helix potential

$$E(l) = k(l - l_0)^2, \tag{59}$$

and the nonlocal Lennard-Jones potential (5.7), are encoded in an excerpt of the sxml output given below:

```
<NONBONDED>
    <BETA_A40_LJ/>
</NONBONDED>
<ATOM id="1" name="CA" x="0" y="0" z="0" resName="GLY" chainID="A" resSeq="1" m="1" eps="0.012" size="1.8" HR="1"/>
<TORSION idAtom1="1" idAtom2="2" idAtom3="3" idAtom4="4" N="3.00" A="0.119"/>
<HELIX idAtom1="4" idAtom2="7" k="2.054" l0="5.50"/>
<BENDING idAtom1="1" idAtom2="2" idAtom3="3" k="1.194" theta0="96.00"/>
```

The `eps` attribute appearing in the `ATOM` tag is the interaction parameter and `size` defines the size of the atom. They are used to calculate σ_{ij} and ϵ_{ij} in the nonlocal Lennard-Jones potential (5.7). The `RH` attribute in the `ATOM` tag is the atom's hydrodynamic radius used in Brownian dynamics as described in Section 9.5.

7.2.3 Structure XML nonbonded tags reference

RedMD provides a range of nonbonded potentials that complement ones predefined in the previously discussed force fields and also allows to simply add the new ones. In order to do that, after creating a final structure `*.sxml` file, one has to identify a nonbonded section, which is located between `<NONBONDED>` and `</NONBONDED>` tags. Afterwards one can add or alter a nonbonded potential list according to needs. Reference of tags used to define the potential is presented below, followed by a list of general parameter that could be applied to all/most potentials.

- **Morse potential**:

$$E_{nb}(r) = E_0 \left[\left(1 - \exp\left[-\alpha(r - r_0)\right]\right)^2 \right) - 1 \right] \tag{60}$$

  XML tag corresponding to the potential is as following (in blue are parameters present in equation shown above):

  `<NLMORSE alpha="`α `E0="`E_0 `l0="`r_0`"/>`

  The Morse potential can be defined separately for different atom pairs:
  `<NLMORSE>`
  `<PAIR type1="CA" type2="P" alpha="`α_{CA-P} `E0="`$E_{0\,CA-P}$ `l0="`$r_{0\,CA-P}$`"/>`
  `<PAIR type1="P" type2="P" alpha="`α_{P-P} `E0="`$E_{0\,P-P}$ `l0="`$r_{0\,P-P}$`"/>`
  `<PAIR type1="CA" type2="CA" alpha="`α_{CA-CA} `E0="`$E_{0\,CA-CA}$ `l0="`$r_{0\,CA-CA}$`"/>`
  `<NLMORSE/>`

- **Coulombic electrostatic potential with distance dependent dielectric constant**:

$$E_{nb}(r) = \frac{1}{4\pi\epsilon_0} \frac{q_i q_j}{(\epsilon r_{ij}) r_{ij}} \tag{61}$$

  XML tag corresponding to the potential is as following (in blue are parameters present in equation shown above):

  `<COULOMB epsilon="`ϵ`"/>`

- **Piecewise potential**:

$$E_{nb}(r) = \begin{cases} E^0_{nb}[(\exp(-\alpha(r-r_0))-1)^2-1] & \text{if } r < r_0 \ , \\ \text{or} & \\ E^0_{nb} + (k(r-r_0)^2) & \text{if } r < r_0 \ , \\ (E^{bar}_{nb} - E^0_{nb})\exp(-\sigma_1(r-r_1)^2) + U^{bar}_0 & \text{if } r_0 < r < r_1 \ , \\ E^{bar}_{nb} & \text{if } r_1 < r < r_2 \ , \\ E^{bar}_{nb}\exp(-\sigma_2(r-r_2)^2) & \text{otherwise} \ , \end{cases} \tag{62}$$

XML tag corresponding to the potential is as following (in blue are parameters present in equation shown above). Piecewise potential with a Morse repulsive part:

```
<DNA_NLPIECEWISE harmonic="false" wellDepth="−E^0_nb" wellPosition="r_0" wellAlpha="α"
barrierPosition="r_1" barrierHeight="E^bar_nb" barrierWidth="r_2−r_1" barrierSig1="σ_1"
barrierSig2="σ_2"/>
```

Harmonic repulsive part:

```
<DNA_NLPIECEWISE harmonic="true" wellDepth="−E^0_nb" wellPosition="r_0" wellAlpha="k"
barrierPosition="r_1" barrierHeight="E^bar_nb" barrierWidth="r_2−r_1" barrierSig1="σ_1"
barrierSig2="σ_2"/>
```

- **Double well potential**

$$E_{nb}(r) = \frac{1}{2}\left[E_{m\,1}(r) + E_{m\,2}(r) - \Delta - \sqrt{(E_{m\,1}(r) - E_{m\,2}(r) + \Delta)^2 + \varepsilon^2} + \left(\Delta + \sqrt{\Delta^2 + \varepsilon^2}\right)\right] \tag{63}$$

where $E_{m\,1}(r)$ and $E_{m\,2}(r)$ are Morse functions defined as:

$$E_{m\,q}(r) = E_{0\,q}\left[\left(1 - \exp\left[-\alpha_q(r - r_{0\,q})\right]\right)^2\right) - 1\right] \tag{64}$$

XML tag corresponding to the potential is as following (in blue are parameters present in equation shown above):

```
<NB_DOUBLEWELL E1="E_01" E2="E_02" R1="r_01" R2="r_02" Alpha1="α_1" Alpha2="α_2" Delta="Δ"
Epsilon="ε"/>
```

- **Containing sphere force**:

$$\vec{F} = A \cdot (R_o - r)^{-n}\hat{e} \quad \text{for} \quad r > R_{cut} \tag{65}$$

XML tag corresponding to the potential is as following (in blue are parameters present in equation shown above):

```
<B_SPHERE cutoff="R_cut" A="A" n="n"/>
```

- **Tabulated potential** Any potential created by user can be used in RedMD, if presented in a text file with a following format:

```
1.0     5.0     -2.0
1.1     4.0     -1.0
1.2     3.0     -0.0
...
```

with the first column containing distance [Å], second containing energy [kcal/mol] and the last one derivative value [kcal/mol/Å].

Distances have to be equally spaced and in ascending order, program reads only the first two distances, computes their difference and assumes all the other spacings are the same. Please be careful, as file accordance with this rule is not checked by RedMD and no warning will be produced, if file is not equally spaced or not in ascending order. For values below the lowest distance and above the highest no extrapolation is performed, but respectively the lowest and the highest distance energy/derivative is simply taken.

If the table file name is `tab.pot`, the following tag is used:

```
<NB_TABULATED potFilename="tab.pot"/>
```

- User defined potential User can easily add own potential with an analytical formula form in the RedMD source code, by providing both energy and derivative definition. These user potentials are found in the following files, for the nonbonded potential:
`src/potential/nonbonded/nb_user1.c`,
`src/potential/nonbonded/nb_user2.c`,
`src/potential/nonbonded/nb_user3.c`
and for the 2-body bonding potential:
`src/potential/bonded/dist_user1.c`,
`src/potential/bonded/dist_user2.c`,
`src/potential/bonded/dist_user3.c`.
Instructions are included in the source code files. After modifications of the mentioned files one has to recompile the whole application for changes to take effect.

Tags corresponding to first user defined nondonded potential is as following (in blue are variables name that can be used in the source code):

```
<NB_USER1 p0="par0" ...  p9="par9" E0="Upar0" ...  E9="Upar9" set1="
set1" ...  set5="set5"/>
```

Respectively for the other potentials: `<NB_USER2 ... />`, `<NB_USER3 ... />` and `<DIST_USER1 ... />`, `<DIST_USER2 ... />`, `<DIST_USER3 .../>` for the local potential. Parameters E0 to E9 are converted from kcal/mol to internal energy units) and parameters set1 to set5 can be set only with values `"True"` and `"False"`.

In addition to numerical function parameters the following parameters can be used to set nonbonded potential behaviour:

- `pairListDist="x"` RedMD maintains a list of pairs of atoms that interact by a nonbonded interaction. Only atoms that satisfy a distance condition (are closer to each other than x Å) are included in the list.
- `pairListFreq="n"` This pair list is updated every n steps.
- `cutoff="x"` For atoms distanced by more than x Åthe potential energy and force are considered zero.
- `tabularized="true" tab_start="x" tab_step="y" tab_count="n"` With this option potential energy and force values are precomputed before start of the simulation and stored in a table. This table is calculated for distances from x Å for n entries, every y Å. The table is accessed during the simulation to find through interpolation energy and force values. Such approach can save simulation time for the most expensive potential. At the moment function is available for DNA Morse with barrier, DNA piecewise, double–well and user defined potential. Applying it for other potentials will result in an error.

An example of `.sxml` with only Coulomb electrostatic potential is presented below:

```
<NONBONDED>
 <COULOMB cutoff="25.000" pairlistDist="50.000" pairlistFreq="10" epsilon="4.000"/>
</NONBONDED>

<MOLECULE molId="mol1">
 <ATOM id="1" name="CA" x="6.283" y="15.041" z="-6.838" resName="PRO" chainID="A" resSeq="1" m="97" q="1.000"/>
 <ATOM id="2" name="CA" x="6.570" y="15.342" z="-3.027" resName="GLN" chainID="A" resSeq="2" m="128" q="0.000"/>
 <ATOM id="3" name="CA" x="3.074" y="13.685" z="-2.377" resName="ILE" chainID="A" resSeq="3" m="113" q="0.000"/>
 <ATOM id="4" name="CA" x="2.006" y="13.535" z="1.253" resName="THR" chainID="A" resSeq="4" m="101" q="0.000"/>
 <ATOM id="5" name="CA" x="-0.014" y="10.377" z="1.861" resName="LEU" chainID="A" resSeq="5" m="113" q="0.000"/>
 <ATOM id="6" name="CA" x="-2.933" y="12.675" z="2.724" resName="TRP" chainID="A" resSeq="6" m="186" q="0.000"/>
 <ATOM id="7" name="CA" x="-5.990" y="12.202" z="0.413" resName="GLN" chainID="A" resSeq="7" m="128" q="0.000"/>
 <ATOM id="8" name="CA" x="-4.954" y="8.582" z="-0.506" resName="ARG" chainID="A" resSeq="8" m="156" q="1.000"/>
 <ATOM id="9" name="CA" x="-2.591" y="9.757" z="-3.216" resName="PRO" chainID="A" resSeq="9" m="97" q="0.000"/>
 <ATOM id="10" name="CA" x="-4.176" y="8.983" z="-6.695" resName="LEU" chainID="A" resSeq="10" m="113" q="0.000"/>
 <ATOM id="11" name="CA" x="-2.296" y="10.095" z="-9.803" resName="VAL" chainID="A" resSeq="11" m="99" q="0.000"/>
 <ATOM id="12" name="CA" x="-2.957" y="9.655" z="-13.574" resName="THR" chainID="A" resSeq="12" m="101" q="0.000"/>
 <ATOM id="13" name="CA" x="-0.652" y="7.090" z="-15.010" resName="ILE" chainID="A" resSeq="13" m="113" q="0.000"/>
```

More than one potential can be defined in a single SXML file. Here an example of `.sxml` with non–local Morse and containing sphere potentials is presented:

```
<STRUCTURE>
  <COMMENT>
</COMMENT>
  <NONBONDED>
    <NLMORSE cutoff="20.000000">
      <PAIR type1="CA" type2="CA" alpha="0.707000" E0="0.207081" l0="9.500000"/>
    </NLMORSE>
    <B_SPHERE cutoff="50.00" pairlistFreq="2" A="102310.0" n="1" R0="140.00"/>
  </NONBONDED>
  <MOLECULE molId="mol1">
    <ATOM id="1" name="CA" x="102.023" y="17.727" z="94.965" resName="MET" chainID="X" resSeq="1" m="131"/>
    <ATOM id="2" name="CA" x="101.965" y="14.128" z="93.350" resName="ARG" chainID="X" resSeq="2" m="156"/>
    <ATOM id="3" name="CA" x="100.683" y="14.757" z="89.688" resName="GLU" chainID="X" resSeq="3" m="129"/>
    <ATOM id="4" name="CA" x="102.824" y="13.700" z="86.607" resName="CYS" chainID="X" resSeq="4" m="103"/>
  </MOLECULE>
  <MOLECULE molId="mol2">
    <ATOM id="1" name="CA" x="227.773" y="7.376" z="-27.785" resName="ILE" chainID="X" resSeq="865" m="113"/>
    <ATOM id="2" name="CA" x="229.867" y="10.038" z="-25.940" resName="LYS" chainID="X" resSeq="866" m="128"/>
    <ATOM id="3" name="CA" x="228.382" y="11.430" z="-22.632" resName="VAL" chainID="X" resSeq="867" m="99"/>
    <ATOM id="4" name="CA" x="229.867" y="14.146" z="-20.346" resName="MET" chainID="X" resSeq="868" m="131"/>
  </MOLECULE>
</STRUCTURE>
```

NOTE: For clarity, in the `*.sxml` examples presented above, the number of decimal places may be limited to avoid line breaking.

7.3 General Rules for Creating `*.txml` and `*.sxml` Files

The topology `*.txml` and structure `*.sxml` files must conform to the following rules:

- non-empty elements are delimited by both a start-tag and an end-tag: `<MOLECULE></MOLECULE>`
- empty elements are marked with an empty-element tag: `<ATOM />` (or alternatively `<ATOM></ATOM>`)
- all attribute values are quoted with either single (') or double (") quotes
- tags may be nested but must not overlap; each non-root element must be completely contained in another element
- element names are case-sensitive

7.4 RedMD Trajectories

Molecular dynamics trajectories resulting from the RedMD execution can be written as formatted PDB, XYZ or VEL files or using the binary CHARMM/X-PLOR DCD format [31, 32]. The trajectories can be visualized using for example the VMD package [33].

The PDB trajectory file contains Cartesian coordinates of pseudo-atoms written at different simulation time steps (snapshots). Each snapshot frame begins with the term `MODEL` and ends with the term `ENDMDL`:

```
MODEL
ATOM          CA ILE A 215        20.704 -18.842 -16.868 1.00 1.00
ATOM          CA LYS A 216        19.667 -19.054 -13.280 1.00 1.00
ATOM          CA TRP A 217        21.998 -11.747 -11.960 1.00 1.00
ATOM          CA LYS A 218        24.333 -14.609 -12.874 1.00 1.00
ENDMDL
MODEL
ATOM          CA ILE A 215        20.227 -18.861 -17.416 1.00 1.00
ATOM          CA LYS A 216        19.449 -18.820 -13.764 1.00 1.00
ATOM          CA TRP A 217        21.527 -11.718 -12.444 1.00 1.00
ATOM          CA LYS A 218        24.014 -14.521 -13.180 1.00 1.00
ENDMDL
```

The XYZ trajectory files contain Cartesian coordinates (x, y, z) of pseudo-atoms as a function of time, separated with lines containing the information on the total number of pseudo-atoms and the time at which the snapshot was saved. No information on residues is given. This format is widely supported by many modeling and visualization programs although no formal specification has been published. Our XYZ format is the following:

```
4
1A36.xyz time [ps]:  0.000000
CA 20.703808 -18.841698 -16.867678
CA 19.666808 -19.053698 -13.279678
CA 21.997808 -11.746698 -11.959678
CA 24.332808 -14.608698 -12.873678
4
1A36.xyz time [ps]:  10.000000
CA 20.226640 -18.860777 -17.415889
CA 19.448818 -18.820134 -13.764272
CA 21.527288 -11.717516 -12.443908
CA 24.013920 -14.520548 -13.180344
```

VEL trajectory files have the same structure as XYZ files but instead of coordinates the pseudo-atom velocities (v_x, v_y, v_z) are given.

Unformatted, binary DCD trajectory files are often more convenient due to their smaller size. DCD files generated with RedMD are structured as follows (FORTRAN UNFORMATTED, with Fortran data type descriptions):

HDR	NSET	ISTRT	NSAVC	5-ZEROS	NATOM-NFREAT	DELTA	9-ZEROS
'CORD'	#files	step 1	step interval	zeroes	(zero)	time step	(zeroes)
C*4	INT	INT	INT	5INT	INT	DOUBLE	9INT
NTITLE		TITLE					
INT (=2)		C*MAXTITL (=32)					
NATOM							
#atoms							
INT							
X(I),	I=1,NATOM		(DOUBLE)				
Y(I),	I=1,NATOM						
Z(I),	I=1,NATOM						

7.5 Energy Output

During molecular dynamics simulation, the energy of the molecule is written to a text file (with output frequency chosen by the user). This file can be used for analysis, as well as verification purposes:

```
#  STEP  TOTAL    KINETIC  POTENTIAL  HARM  MORSE    NLMORSE
   0     -1105.0  120.0    -1225.1    0.0   -1172.3  -52.8
   100   -1104.7  61.7     -1166.4    39.4  -1151.8  -54.0
```

Apart from the total (TOTAL), kinetic (KINETIC) and potential (POTENTIAL) energy, additional terms are written:
HARM - the total energy of harmonic terms
MORSE - the total energy of local nonbonded interactions given with Morse terms
NLMORSE - the total energy of nonlocal nonbonded interactions given with Morse terms

Energy terms that appear in the energy output file depend on the applied force field. All energy terms are given in units of $\frac{kcal}{mol}$.

7.6 Restart Files

Three files are needed to restart (continue) the simulation: a file with coordinates (`*.coord`) from the previous run, a file with momenta (`*.mom`) from the previous run, and a control file for restarting/continuing the simulation. These files are created when appropriate keywords are specified in the simulation control file (see Section 9). The coordinate and momenta files are formatted as follows:

```
1000
20.226640200933019 -8.860777042544177 -17.415889026259169
19.448817510503815 -8.820134477308015 -13.764272284521876
21.527288226728508 -11.717516343615166 -12.443908340068297
24.013920197728364 -14.520548188109931 -13.180343645962846
```

In the first line the number of performed simulation steps is given. The following lines contain coordinates (or velocities) of pseudo-atoms with each line corresponding to one pseudo-atom.

The third restart file – the RedMD control file – contains the keywords needed to continue/restart the simulation with an additional keyword `step` (see Section 9) which gives the number of simulation steps performed so far. Provided that the preceding simulation was not terminated unexpectedly, its value should match the values given in the `*.coord` and `*.mom` files.

During execution, RedMD creates also temporary copies of restart files (`.tmpcoord`, `.tmpmom`, and `.tmpinp`) that can be used to restart the simulation in case the restart data get lost or corrupted.

8 RedMD Programs

This section describes all programs included in the RedMD package (see Figure 8) that allow the user to:

- build a model of a biomolecule according to the chosen force field (i.e. perform the reduction of degrees of freedom and generate the topology `*.txml` and structure `*.sxml` files based on the input PDB or XML PDB data),
- verify and if necessary edit the model,
- conduct the simulation.

A detailed description of how to run each program is also given.

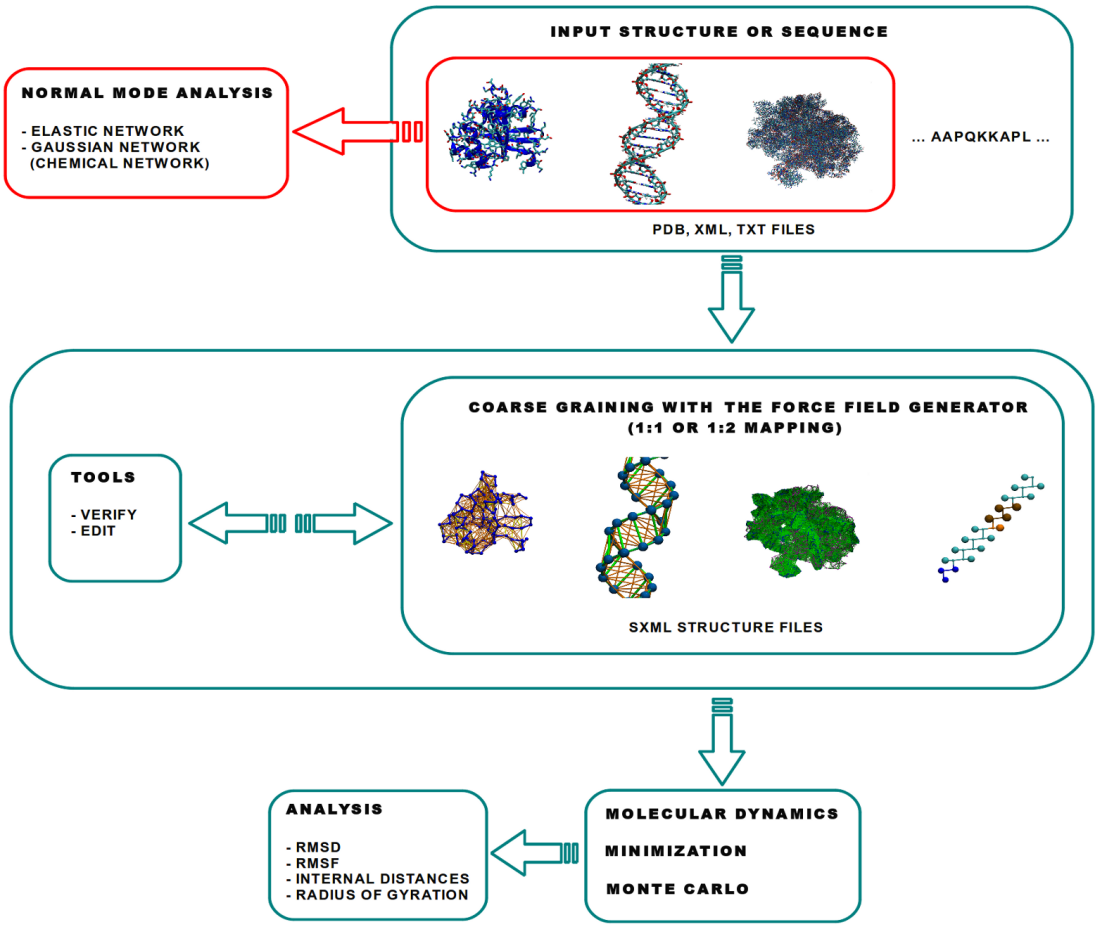


Figure 8: Flowchart illustrating the usage of the RedMD package.

8.1 Topology Extraction Tools

The `RedMD_extractPDB` and `RedMD_extractPDBML` programs can be used to create the topology (`*.txml`) of the reduced model of the molecule.

RedMD_extractPDB - extracts information on chains from an input PDB file (`name.pdb`) and produces the topology (`*.txml`) file. Only C_α (CA) and P atoms are extracted.

Usage:

```
RedMD_extractPDB [OPTION...] name.pdb
```

where possible options are:
`-o, --outxml=` *name* - name of the output `*.txml` file. If none of these options is present, the topology is redirected to the standard output
`-V, --version` - print program version and quit
`-h, --help` - print help and quit
`--usage` - print short usage description and quit

Supported PDB files adhere to the following format:

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"ATOM "	
7 - 11	Integer	serial	Atom serial number
13 - 16	Atom	name	Atom name
17	Character	altLoc	Alternate location indicator
18 - 20	Residue name	resName	Residue name
22	Character	chainID	Chain identifier
23 - 26	Integer	resSeq	Residue sequence number
27	AChar	iCode	Code for insertion of residues
31 - 38	Real(8.3)	x	X coordinate
39 - 46	Real(8.3)	y	Y coordinate
47 - 54	Real(8.3)	z	Z coordinate
55 - 60	Real(6.2)	occupancy	Occupancy
61 - 66	Real(6.2)	tempFactor	Temperature factor
77 - 78	String(2)	element	Element symbol, right-justified
79 - 80	String(2)	charge	Charge on the atom

RedMD_extractPDBXML - extracts information on chains and if applicable also on base pairs from an input XML PDB (PDBML) file or gzipped XML PDB file (`name.xml, input.name.gz`) and produces the topology (`*.txml`) file. Only C_α (CA) and P atoms are extracted. The XML versions of files can be obtained from the PDB repository, `http://pdbml.rcsb.org`.

Usage:

```
RedMD_extractPDBXML [OPTION...] name.(xml;xml.gz)
```

where possible options are:
`-o, --outxml=`*name* - name of the output `*.txml` file. If none of these options is present, topology is redirected to the standard output
`-V, --version` - print program version and quit

-?, --help - print help and quit
--usage - print short usage description and quit

NOTE - Nucleosome Model: In case of the Nucleosome model, fields ATOM for all CA atoms should be replaced with fields HELI (for helices), COIL (for random coils or loops) or TAIL (unstructured proteins tails), to provide information regarding secondary structure of a molecule (see Section 5.3 of this manual and examples distributed with the RedMD package).
NOTE - Ribosome Model: In case of the Ribosome model, RedMD_extractPDBXML should be used to extract topology because only the XML PDB (PDBML) files contain information about the secondary structure of RNA giving connectivity between base pairs. The structure file for the whole ribosome (composed of the small and large subunit) can be generated based on separate structure files prepared for individual subunits. We provide a tool to join these files which assures that interactions between subunits are correctly modeled.

8.2 Force Field Generator

GenForceField

Force field generator is a program that allows the user to create the structure file (*.sxml) of a biomolecule according to the chosen parameter set and the potential energy function, based on the input topology file (*.txml) or directly from the PDB (*.pdb) or the PDBML (*.xml) file. Currently, seven force field generators are provided. In case of the βA-40 model, generator creates the structure file using as an input a plain-text sequence file.

Usage:

GenForceField [OPTIONS] [FFOPTIONS] name(.pdb|.pdb.gz|.xml|.xml.gz|.txml|.txml.gz|.sxml (RibJoin only)) [name2.sxml]

where possible options are:

-h, --help - print help and quit
-u, --usage - print short usage description and quit

--infile=*value* - the program recognizes formats of input files using their extensions. However, this option can by used to override format of the input file; possible values are:
*.txml, *.xml
-o, --outxml=*name* - name of the output *.sxml file, name.sxml

[name2.sxml] - second structure file, when the type option (see below) is set to RibJoin

--type=*value* - force field to be used, possible values are:
COULOMB - assign charges to a molecule stored in the input *.txml file
Rib - use the Ribosome model
RibJoin - join two ribosomal subunits contained in separate *.sxml files. Typically, structures

of ribosomes deposited in the PDB archive are split between two files. Each file contains a single ribosomal subunit. `RibJoin` makes it possible to prepare a single structure file for the whole ribosome, using separate structure files of ribosomal subunits generated from appropriate PDB or PDBML files.
`Nucl` - use the Nucleosome model
`HIVP` - use the HIV-1 protease model
`ENM` - use the Elastic Network model
`DNA` - use the DNA model
`BetaA40` - use the βA-40 model
`REACH` - use the REACH model

Force field specific options (FFOPTIONS) are described below.

8.2.1 Generic Elastic Network Model

Harmonic bonds options:
`--cutoffCC=`*value* - cut-off distance for creating CA-CA bonds, 8.0[$\mathring{A}$]
`--cutoffCP=`*value* - cut-off distance for creating CA-P bonds, 15.0[$\mathring{A}$]
`--cutoffPP=`*value* - cut-off distance for creating P-P bonds, 20.0[$\mathring{A}$]
`--kCC=`*value* - force constant for harmonic CA-CA bonds, 1.0[$\frac{kcal}{mol \cdot \mathring{A}^2}$]
`--kCP=`*value* - force constant for harmonic CA-P bonds, 1.0[$\frac{kcal}{mol \cdot \mathring{A}^2}$]
`--kPP=`*value* - force constant for harmonic P-P bonds, 1.0[$\frac{kcal}{mol \cdot \mathring{A}^2}$]

All of the above options can be contained within the XML-formatted file that can be passed to the force field generator with the option `--conffile=`*name*. Parameters specified in the command line take precedence over parameters stored in the XML parameter file (see Section 8.2.10).

8.2.2 Ribosome Model

Harmonic bonds options:
`--KCA_12=`*value* - force constant for harmonic 1-2 CA-CA bonds, 50.0[$\frac{kcal}{mol \cdot \mathring{A}^2}$]
`--KCA_13=`*value* - force constant for harmonic 1-3 CA-CA bonds, 5.0[$\frac{kcal}{mol \cdot \mathring{A}^2}$]
`--KCA_14=`*value* - force constant for harmonic 1-4 CA-CA bonds, 1.5[$\frac{kcal}{mol \cdot \mathring{A}^2}$]
`--KP_12=`*value* - force constant for harmonic 1-2 P-P bonds, 3.0[$\frac{kcal}{mol \cdot \mathring{A}^2}$]
`--KP_13=`*value* - force constant for harmonic 1-3 P-P bonds, 2.5[$\frac{kcal}{mol \cdot \mathring{A}^2}$]
`--KP_14=`*value* - force constant for harmonic 1-4 P-P bonds, 0.5[$\frac{kcal}{mol \cdot \mathring{A}^2}$]
`--KSEC=`*value* - force constant for harmonic bonds between paired bases, 0.6[$\frac{kcal}{mol \cdot \mathring{A}^2}$]
`--rigid=`*true/false* - the option to apply SHAKE to bonds between neighbouring (1-2) backbone pseudo-atoms

Local Morse terms options:
`--cutoffCA=`*value* - cut-off distance for creating local Morse CA-CA terms, 12[$\mathring{A}$]
`--cutoffP=`*value* - cut-off distance for creating local Morse P-P terms, 20[$\mathring{A}$]

The options presented below correspond to an analytical formula $E \cdot \exp\left(-\frac{x}{R}\right)$ which describes the shape of the local Morse terms (the dissociation constant) for different types of atoms:
`--edicE=`*value* - CA-CA term, E, 4.0[$\frac{kcal}{mol}$]
`--edicR=`*value* - CA-CA term, R, 2.8[$\mathring{A}$]
`--edipE=`*value* - P-P term, E, 2.0[$\frac{kcal}{mol}$]
`--edipR=`*value* - P-P term, R, 6.0[$\mathring{A}$]

Nonlocal Morse terms options:
`--cutoff=`*value* - cut-off distance, 35.0[$\mathring{A}$]
`--rMorseCA=`*value* - the Morse equilibrium length for CA-CA terms, 9.5[$\mathring{A}$]
`--rMorseP=`*value* - the Morse equilibrium length for P-P terms, 17.6[$\mathring{A}$]

All of the above options can be contained within the XML-formatted file that can be passed to the force field generator with the option `--conffile=`*name*. Parameters specified in the command line take precedence over parameters stored in the XML parameter file (see Section 8.2.10).

8.2.3 The RibJoin Type - Joining Ribosomal Subunits

Local Morse terms options:
`--cutoffCA=`*value* - cut-off distance for creating local Morse CA-CA terms, 12[$\mathring{A}$]
`--cutoffP=`*value* - cut-off distance for creating local Morse P-P terms, 20[$\mathring{A}$]

The options presented below correspond to an analytical formula $E \cdot \exp\left(-\frac{x}{R}\right)$ which describes the shape of the local Morse terms (the dissociation constant) for different types of atoms:
`--edicE=`*value* - CA-CA term, E, 4.0[$\frac{kcal}{mol}$]
`--edicR=`*value* - CA-CA term, R, 2.8[$\mathring{A}$]
`--edipE=`*value* - P-P term, E, 2.0[$\frac{kcal}{mol}$]
`--edipR=`*value* - P-P term, R, 6.0[$\mathring{A}$]

All of the above options can be contained within the XML-formatted file that can be passed to the force field generator with the option `--conffile=`*name*. Parameters specified in the command line take precedence over parameters stored in the XML parameter file (see Section 8.2.10).

8.2.4 Nucleosome Model

Harmonic bonds options:
`--KCA_12C=`*value* - force constant for harmonic 1-2 CA-CA bonds within coils, 120.0[$\frac{kcal}{mol \cdot \mathring{A}^2}$]
`--KCA_12H=`*value* - force constant for harmonic 1-2 CA-CA bonds within helices, 153.0[$\frac{kcal}{mol \cdot \mathring{A}^2}$]
`--KCA_13C=`*value* - force constant for harmonic 1-3 CA-CA bonds within coils, 5.0[$\frac{kcal}{mol \cdot \mathring{A}^2}$]
`--KCA_13H=`*value* - force constant for harmonic 1-3 CA-CA bonds within helices, 5.0[$\frac{kcal}{mol \cdot \mathring{A}^2}$]
`--KCA_14H=`*value* - force constant for harmonic 1-4 CA-CA bonds within helices, 1.5[$\frac{kcal}{mol \cdot \mathring{A}^2}$]
`--KCA_15H=`*value* - force constant for harmonic 1-5 CA-CA bonds within helices, 1.5[$\frac{kcal}{mol \cdot \mathring{A}^2}$]

--KP_12=*value* - force constant for harmonic 1-2 P-P bonds, 3.0[$\frac{kcal}{mol \cdot \mathring{A}^2}$]
--KP_13=*value* - force constant for harmonic 1-3 P-P bonds, 2.5[$\frac{kcal}{mol \cdot \mathring{A}^2}$]
--KP_14=*value* - force constant for harmonic 1-4 P-P bonds, 0.5[$\frac{kcal}{mol \cdot \mathring{A}^2}$]
--KSEC=*value* - force constant for harmonic bonds between paired bases, 0.6[$\frac{kcal}{mol \cdot \mathring{A}^2}$]
--RSEC=*value* - equilibrium length for harmonic bonds between paired bases, 19.3[$\mathring{A}$]
--RCA_12C=*value* - equilibrium length for harmonic 1-2 CA-CA bonds within coils, 3.8[$\mathring{A}$]
--RCA_12H=*value* - equilibrium length for harmonic 1-2 CA-CA bonds within helices, 3.8[$\mathring{A}$]
--RCA_13C=*value* - equilibrium length for harmonic 1-3 CA-CA bonds within coils, 6.4[$\mathring{A}$]
--RCA_13H=*value* - equilibrium length for harmonic 1-3 CA-CA bonds within helices, 5.5[$\mathring{A}$]
--RCA_14H=*value* - equilibrium length for harmonic 1-4 CA-CA bonds within helices, 5.1[$\mathring{A}$]
--RCA_15H=*value* - equilibrium length for harmonic 1-5 CA-CA bonds within helices, 6.2[$\mathring{A}$]
--RP_12=*value* - equilibrium length for harmonic 1-2 P-P bonds 6.8[$\mathring{A}$]
--RP_13=*value* - equilibrium length for harmonic 1-3 P-P bonds 13.0[$\mathring{A}$]
--RP_14=*value* - equilibrium length for harmonic 1-4 P-P bonds 18.0[$\mathring{A}$]

Local Morse terms options:
--cutoffCA=*value* - cut-off distance for creating local Morse CA-CA terms, 9[$\mathring{A}$]
--cutoffP=*value* - cut-off distance for creating local Morse P-P terms, 24[$\mathring{A}$]

The options presented below correspond to an analytical formula $E \cdot \exp\left(-\frac{x}{R}\right)$ which describes the shape of the local Morse terms (the dissociation constant) for different types of atoms:
--edicE=*value* - CA-CA term, E, 7.0[$\frac{kcal}{mol}$]
--edicR=*value* - CA-CA term, R, 2.8[$\mathring{A}$]
--edipE=*value* - P-P term, E, 1.0[$\frac{kcal}{mol}$]
--edipR=*value* - P-P term, R, 6.0[$\mathring{A}$]

Nonlocal Morse terms options:
--cutoff=*value* - cut-off distance, 30.0[$\mathring{A}$]
--rMorseCA=*value* - the Morse equilibrium length for CA-CA terms, 10.0[$\mathring{A}$]
--rMorseP=*value* - the Morse equilibrium length for P-P terms, 26.3[$\mathring{A}$]

All of the above options can be contained within the XML-formatted file that can be passed to the force field generator with the option --conffile=*name*. Parameters specified in the command line take precedence over parameters stored in the XML parameter file (see Section 8.2.10).

8.2.5 HIV-1 Protease Model

Harmonic bonds options:
--KCA_12=*value* - force constant for harmonic 1-2 CA-CA bonds, 50.0[$\frac{kcal}{mol \cdot \mathring{A}^2}$]
--rigid=*true/false* - option to apply SHAKE to bonds between neighbouring (1-2) backbone pseudo-atoms

Local Morse terms options:

`--cutoffCA=`*value* - cut-off distance for creating local Morse CA-CA terms, 8[$\mathring{A}$]

The options below correspond to an analytical formula $E \cdot \exp\left(-\frac{x}{R}\right)$ which describes the shape of local Morse terms (the dissociation constant) for different types of atoms:
`--edicE=`*value* - CA-CA term, E, 6.0[$\frac{kcal}{mol}$]
`--edicR=`*value* - CA-CA term, R, 2.8[$\mathring{A}$]

Nonlocal Morse terms options:
`--cutoff=`*value* - cut-off distance, 20.0[$\mathring{A}$]
`--rMorseCA=`*value* - the Morse equilibrium length for CA-CA terms, 9.5[$\mathring{A}$]

Secondary structure options:
`--ka=`*value* - alpha helices, 38.0[$\frac{kcal}{mol \cdot rad^2}$]
`--kb=`*value* - beta sheets, 17.0[$\frac{kcal}{mol \cdot rad^2}$]

All of the above options can be contained within the XML-formatted file that can be passed to the force field generator with the option `--conffile=`*name*. Parameters specified in the command line take precedence over parameters stored in the XML parameter file (see Section 8.2.10).

8.2.6 REACH Model

Harmonic bonds options:
`--cutoffCA=` *value* - cut-off distance for creating CA-CA bonds, 12.0[$\mathring{A}$]
`--aintra=` *value* - nonbonded, intra-chain interactions, formula $a \cdot e^{-b \cdot r}$, 2560.0[$\frac{kJ}{\mathring{A}^2}$]
`--bintra=` *value* - nonbonded, intra-chain interactions, formula $a \cdot e^{-b \cdot r}$, 0.8[$\frac{1}{\mathring{A}}$]
`--ainter=` *value* - nonbonded, inter-chain interactions, formula $a \cdot e^{-b \cdot r}$, 1630.0[$\frac{kJ}{\mathring{A}^2}$]
`--binter=` *value* - nonbonded, inter-chain interactions, formula $a \cdot e^{-b \cdot r}$, 0.772[$\frac{1}{\mathring{A}}$]

All of the above options can be contained within the XML-formatted file that can be passed to the force field generator with the option `--conffile=`*name*. Parameters specified in the command line take precedence over parameters stored in the XML parameter file (see Section 8.2.10).

8.2.7 DNA Model

Harmonic terms options:
`--K_12=`*value* - force constant for harmonic 1–2 bonds, 12.5[$\frac{kcal}{mol \cdot \text{Å}^2}$]
`--K_13=`*value* - force constant for harmonic 1–3 bonds (planar angles), 25[$\frac{kcal}{mol \cdot \text{Å}^2}$]
`--K_14=`*value* - force constant for harmonic 1–4 bonds (dihedral angles), 10[$\frac{kcal}{mol \cdot \text{Å}^2}$]
`--l_12=`*value* - equilibrium length for harmonic 1–2 bonds, 6.66[Å]
`--l_13=`*value* - equilibrium angle measure for harmonic 1–3 bonds, 149[deg]
`--l_14=`*value* - equilibrium dihedral angle measure for harmonic 1–4 bonds, -19.1[deg]

Local Morse terms options:
`--c_0=`*value* - c parameter for u_0 potentials, 0.4

--c_1=*value* - c parameter for u_1 potentials, 0.4
--c_2=*value* - c parameter for u_1 potentials, 0.18
--r0_0=*value* - r_0 parameter for u_0 potentials, 19.25[Å]
--r0_1=*value* - r_0 parameter for u_1 potentials, 17.93[Å]
--r0_2=*value* - r_0 parameter for u_2 potentials, 15.3[Å]
--r1_0=*value* - r_1 parameter for u_0 potentials, 21.5[Å]
--r1_1=*value* - r_1 parameter for u_1 potentials, 20.18[Å]
--r1_2=*value* - r_1 parameter for u_2 potentials, 17.8[Å]
--E_0=*value* - E parameter for u_0 potentials, 2.8$[\frac{kcal}{mol \cdot Å^2}]$
--E_1=*value* - E parameter for u_1 potentials, 2.8$[\frac{kcal}{mol \cdot Å^2}]$
--E_2=*value* - E parameter for u_2 potentials, 4.8$[\frac{kcal}{mol \cdot Å^2}]$
--alpha_0=*value* - α_0 parameter for u_0 potentials, 0.8$[Å^{-1}]$
--alpha_1=*value* - α_1 parameter for u_1 potentials, 0.8$[Å^{-1}]$
--alpha_2=*value* - α_2 parameter for u_2 potentials, 0.4$[Å^{-1}]$
--epsilon_0=*value* - ε_0 parameter for u_0 potentials, 1$[\frac{kcal}{mol}]$
--epsilon_1=*value* - ε_1 parameter for u_1 potentials, 1$[\frac{kcal}{mol}]$
--epsilon_2=*value* - ε_2 parameter for u_2 potentials, 1$[\frac{kcal}{mol}]$
--lambda_0=*value* - λ_0 parameter for u_0 potentials, 3.7$[Å^{-1}]$
--lambda_1=*value* - λ_1 parameter for u_1 potentials, 3.7$[Å^{-1}]$
--lambda_2=*value* - λ_2 parameter for u_2 potentials, 2$[Å^{-1}]$

Nonlocal Morse terms options:
--E_nb=*value* - E parameter for u_{nb} potentials, 75$[\frac{kcal}{mol \cdot Å^2}]$
--r0_nb=*value* - r_0 parameter for u_{nb} potentials, 12[Å]
--r1_nb=*value* - r_1 parameter for u_{nb} potentials, 15.7[Å]
--r2_nb=*value* - r_2 parameter for u_{nb} potentials, 16.5[Å]
--alpha_nb=*value* - α parameter for u_{nb} potentials, 0.014$[Å^{-1}]$
--lambda1_nb=*value* - λ_1 parameter for u_{nb} potentials, 1$[Å^{-1}]$
--lambda2_nb=*value* - λ_2 parameter for u_{nb} potentials, 3$[Å^{-1}]$
--epsilon1_nb=*value* - ε_1 parameter for u_{nb} potentials, 1$[\frac{kcal}{mol}]$
--epsilon2_nb=*value* - ε_2 parameter for u_{nb} potentials, 0.77$[\frac{kcal}{mol}]$
--c_nb=*value* - c parameter for u_{nb} potentials, 0.007
--A_nb=*value* - A parameter for u_{nb} potentials, 0.4

All of the above options can be contained within the XML-formatted file that can be passed to the force field generator with the option --conffile=*name*. Parameters specified in the command line take precedence over parameters stored in the XML parameter file (see Section 8.2.10).

8.2.8 βA-40 Model

--forceConst=*name* - a XML-formatted file with constants needed by the force field generator. Its root element with a start-tag <forceConstants> has subelements <sizeCA>, <distCA>, <angleCA>, <KR>, <KRS>, <RH>, <KTheta>, <epsTors>, <epsCA>, <epsMin>, <epsMax>, <HMin>, <HMax> which enclose the values of parameters $\sigma_{C\alpha}$, r_0, θ_0, K_r, K_r^s, K_θ, H_{min}, H_{max}, ϵ_{min}, ϵ_{max},

r_h, as described in Section 5.7 and Table11, as well as the <massCA> subelement storing the mass of a backbone atom. The root element also contains subelements <hydropathies>, <sizes>, <angles>, <HydrRadCA>, <HydrRadCB>, <masses> with each having its own set of subelements <A>, <V>, <L> etc., for storing the parameters of amino acids with the corresponding letter symbols. They represent H_{ii}, σ_{ii} and θ_0^s as described in Section 5.7, the hydrodynamic radii of backbone and side atoms of different amino acids used in Brownian dynamics, and finally the masses of side atoms. By default, the first three types of parameters are set to the values given in Table 12, whereas hydrodynamic radii of all amino acids' side atoms are assigned a value of 10[Å] and backbone atoms of 1[Å]. The masses of side atoms are set to $1u$.

--aminoSeq=*name* - a file with a sequence of a single-chain peptide. It has the form of a string of amino acid one letter symbols

8.2.9 Protein Electrostatics

--ASP=*value* - protonated/deprotonated ASP side chain (default: -1.0[e], deprotonated)
--GLU=*value* - protonated/deprotonated GLU side chain (default: -1.0[e], deprotonated)
--LYS=*value* - protonated/deprotonated LYS side chain (default: 1.0[e], protonated)
--TYR=*value* - protonated/deprotonated LYS side chain (default: 0.0[e], protonated)
--ARG=*value* - protonated/deprotonated LYS side chain (default: 1.0[e], protonated)
--HIS=*value* - protonated/deprotonated LYS side chain (default: 1.0[e], protonated)
--NTER=*value* - protonated/deprotonated N-terminal (default: 1.0[e], protonated)
--CTER=*value* - protonated/deprotonated C-terminal (default: -1.0[e], deprotonated)
--cutoff=*value* - exterior cut-off, Coulombic terms are not evaluated if the distance between the charges within a given pair is above this value (default: 25[Å])
--pairlistDist=*value* - cut-off for generating the list of Coulombic terms (default: 2·`cutoff`)
--pairlistFreq=*value* - the frequency of updating the list of interacting charges (default: 10)
--epsilon=*value* - dielectric constant, see Equation 43

All of the above options can be contained within the XML-formatted file that can be passed to the force field generator with the option --conffile=*name*. Parameters specified in the command line take precedence over parameters stored in the XML parameter file (see Section 8.2.10).

8.2.10 Force Field Parameter Files

XML-formatted files containing force field specific options that are passed to the `GenForceField` program. Its root element <config> has subelements corresponding to force field parameters. For example, in case of the Elastic Network model, subelements enclosing values of particular parameters are <cutoffCC>, <cutoffCP>, <cutoffPP>, <kCC>, <kCP> and <kPP>.

8.3 RNA Model Tools

Ribonucleic acid molecules are usually present *in vivo* as single chains, folded into a 3D structure stabilized by a network of Watson–Crick and other hydrogen bond interactions. Therefore, specialized tools to deal with RNA are attached to RedMD: these are the *i*nput file generator and *d*istance distribution analyzer.

8.3.1 Input File Generator

The RNAGenForceField is an alternative for the standard RedMD GenForceField providing a proper description of RNA secondary and tertiary structure, as described in Leonarski et al.[8].

Usage:

```
RNAGenForceField [OPTION...] name.pdb rna.secondary.structure.file output.sxml
```

The command line parameters have to be provided in an order described above: first, the options and second the file names. The `rna.secondary.structure.file` can be provided in two formats: a dot–bracket notation or RNAView output.

In the dot-bracket file case only canonical Watson–Crick interactions are included and pseudo–knots are excluded. The file should contain one line with the character number consistent with base numbers[1]. The dot–bracket file can be generated from a sequence using secondary structure prediction tools, for example, Vienna RNA[2] or Mfold[3].

RNAView is a tool to extract the bonding information from a full–atomistic PDB file[4]. The output from the file can be loaded to the generator. It provides more information than the dot–bracket notation, allowing for pseudo–knots and noncanonical base pairs, however it requires the knowledge of the actual RNA 3D structure.

By default the RNA equilibrium dynamics force field parameters will be used (first parameter column in Table 10).

The following options to the program are available:

`--startFromCircle` - the RNA chain is placed on a large circle, with the distance between the P-P atoms of 5 Å (see Figure 7C). This allows to start a simulation from an unfolded conformation, so this option should be used in a tertiary structure prediction simulation.
`--tertPredFF` - use the tertiary structure prediction force field (the second column in Table 10), this option overwrites all previously set parameters.
`--param` *`label val`* - set the value of the parameter described by *`label`* (bonded potential: `l0`, `k`, angle pot.: `theta0`, `kTheta`, dihedral potential: `alpha0`, `kAlpha`, harmonic hydrogen bonding potential: `IJ0_R0`, `IJ0_K`, `IJ1_R0`, `IJ1_K`, `IJ2_R0`, `IJ2_K`, nonbonded Coulomb potential: `NONB_ELEC_EPS`) to a value of *`val`* (floating point number).

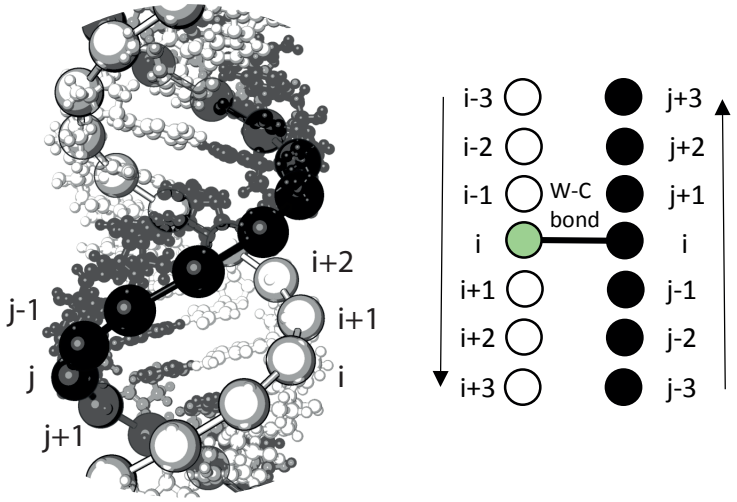


Figure 9: Description of the helix numbering, where *i:i+n* are the beads on the same strand and *i:j+n* are the distances on complementary strands, relative to the selected hydrogen bond.

8.3.2 Distance Distribution Analyzer

The `RNADistributionTool` allows to find the distributions of distances in a set of RNA structures, both from the database and MD trajectory. This program allows to select atoms for distribution calculation, aiding in the design of coarse–grained models for nucleic acids.

Usage:

```
RNADistributionTool [OPTION...] distribution.source
```

The `distribution.source` is a path towards either a single PDB file or a directory containing multiple PDB files. Distributions are printed to the standard output. The first row provides a legend.

The following options are available:

`--min` *`minvalue`* - sets a minimal value of the distance for the distribution calculations, the atoms connected by a shorter distance will be ignored. [default=0 Å]
`--max` *`maxvalue`* - sets a maximal value of the distance for the distribution calculations, the atoms connected by a longer distance will be ignored. [default=50 Å]
`--bins` *`number of bins`* - sets a number of bins used for the distribution calculation. [de-

[1]Please note that if a PDB file contains nucleotides lacking a phosphorus atom, e.g., terminal residues used by full–atomistic MD force fields, they have to be manually removed from the dot-bracket file
[2]http://www.tbi.univie.ac.at/RNA/
[3]http://mfold.rna.albany.edu/?q=mfold
[4]http://ndbserver.rutgers.edu/ndbmodule/services/download/rnaview.html

fault=100]
--dcd *fileName* - molecular dynamics trajectory is loaded for distribution calculation. With this option one has to provide a single PDB file, that will work as a reference for the DCD (it has to have consistent number and order of the P pseudo–atoms).
--use2d - use secondary structure information.
--file2d *fileName* - provides a file containing secondary structure information, either in RNAView format or dot–bracket notation, see previous subsection.
--total - calculate the total radial distribution function for P atoms.
--intrastr *max* - calculate the distance distribution of P atoms neighboring on a single strand by not more than *max* beads, separately for each distance. See *i:i+n* distances in Figure 9.
--interstr *min max* - calculate the distance distribution of P atoms on complementary strands. The distance is calculated for atoms that are positioned between *min* and *max* from the position of the Watson-Crick complementary base. See *i:j+n* distances in Figure 9.
--nbonded - calculate the distance distribution of P atoms that are not connected by bonds, angle, dihedrals or hydrogen bonds.
--angle - find distributions of angles between three neighboring P atoms on a single strand.
--dih - find distributions of dihedral angles between four neighboring P atoms on a single strand.
--norm - print the normalization information. An additional column will be added which contains a sum of the distribution value over the full range.

8.4 Previewing & Editing Tools

Three programs are provided which allow the user to preview and edit the RedMD XML files. These are: `RedMD_prevModel`, `RedMD_grepNode` and `RedMD_updateXML`.

`RedMD_prevModel` - generates a PDB file based on the input topology (`*.txml`) or structure (`*.sxml`) files (the program accepts also gzipped files). The user may choose which tag to output. The `<BOND.../>` or `<MORSE.../>` tags from an input `*.txml` and/or `*.sxml` files are represented in the output PDB file as lines beginning with the `CONECT` phrase. This program is useful for visualization and validation purposes (see Figure 10).

Usage:

`RedMD_prevModel [OPTION...]` *name.(sxml;sxml.gz;txml;txml.gz)* `[OPTION...] [<XML tag>]`

where possible options are:
`-?, --help` - print help and quit
`--usage` - print short usage description and quit
`-V, --version` - print program version and quit

Usage examples:

If you want to visualize the pseudo-atoms connected with the `BOND` tags:

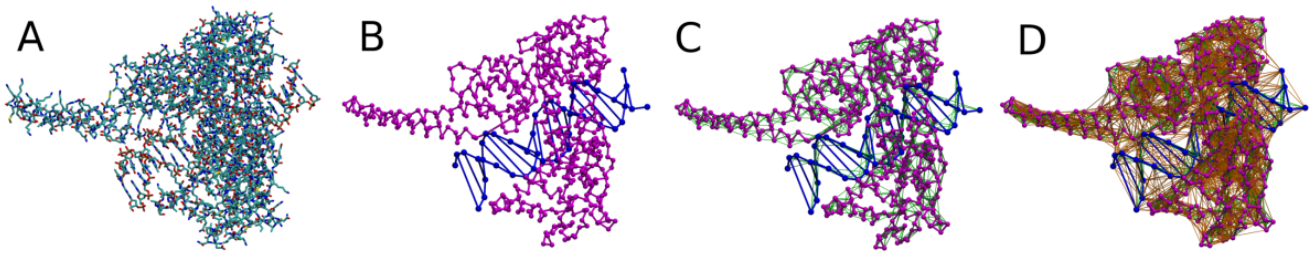


Figure 10: The topoisomerase I/DNA complex (PDB code 1A36). (A) Full atom model. (B) Topology: C_α (magenta) and P (blue) pseudo-atoms and backbone connections (1-2 bonds and base pairing). (C) Harmonic 1-2 (magenta and blue), 1-3 and 1-4 pseudo-bonds (green) characterized with the tag `BOND`. (D) Harmonic pseudo-bonds (`BOND` colors as in (C)) and local nonbonded interactions (`MORSE`, orange) between pseudo-atoms. Visualization was performed under VMD [33].

`RedMD_prevModel` *file.sxml* `BOND > file.pdb`
If you want to visualize the pseudo-atoms connected with the `MORSE` tags:
`RedMD_prevModel` *file.sxml* `MORSE > file.pdb`

`RedMD_grepNode` - allows the user to extract and print on the screen the information about the nodes present in the XML files.

Usage:

`RedMD_grepNode [OPTION...]` *name.(sxml;txml)* `[OPTION...] [<node name> [node attributes names list]]`

where possible options are:
`-i, --addindex=`*number* - index extracted nodes, starting with a given value
`-?, --help` - print help and quit
`--usage` - print short usage description and quit
`-V, --version` - print program version and quit

Usage examples:

Print coordinates of all atoms: `RedMD_grepNode` *file.sxml* `ATOM x y z`

Result:

```
16.381 10.209 59.786
15.306 15.821 56.766
17.678 21.502 54.579
12.511 25.157 52.440
18.890 25.334 50.327
24.551 22.113 48.394
```

Print parameters of all bonded interactions: RedMD_grepNode *file.sxml* BOND

Result:

```
<BOND idAtom1="1" idAtom2="2" k="50.000000" l0="3.740863"/>
<BOND idAtom1="2" idAtom2="3" k="50.000000" l0="3.798448"/>
<BOND idAtom1="3" idAtom2="4" k="50.000000" l0="3.805084"/>
<BOND idAtom1="4" idAtom2="5" k="50.000000" l0="3.794807"/>
<BOND idAtom1="5" idAtom2="6" k="50.000000" l0="3.830967"/>
<BOND idAtom1="6" idAtom2="7" k="50.000000" l0="3.731771"/>
<BOND idAtom1="7" idAtom2="8" k="50.000000" l0="3.791100"/>
<BOND idAtom1="8" idAtom2="9" k="50.000000" l0="3.756520"/>
```

RedMD_updateXML - this utility program updates the coordinates (and/or momenta) stored in the input `*.sxml` file using external PDB files. Ordering and number of atoms in external PDB files must be exactly the same (!) as in the input `*.sxml` file. Gzipped files are also accepted as an input.

Usage:

RedMD_updateXML [OPTION...] *name.(sxml;sxml.gz)*

where possible options are:
-o, --outxml=*name* - the name of the output `*.sxml` file
-p, --moment=*name* - an input file containing the momenta of atoms (PDB format)
-x, --coord=*name* - an input file containing the coordinates of atoms (PDB format)
-?, --help - print help and quit
--usage - print short usage description and quit
-V, --version - print program version and quit

8.5 Main Simulation Module

All molecular dynamics simulations are performed with the main simulation module - `RedMD`. To start a simulation, two files are needed: a properly prepared structure file (`*.sxml`) and a simulation control file which sets up the simulation requested by the user.

`RedMD`

Usage (serial and OpenMP version):

`RedMD` *input*

Usage (MPI version):

`mpirun -np N RedMD` *input*

where *input* gives the path to the simulation control file, N is the number of threads to be used. In case of MPI, the actual command (`mpirun` or `mpiexec`) depends on the system.

Additional files are needed to restart a simulation. Apart from the structure XML file (`*.sxml`), the files with final coordinates (`*.coor`) and momenta (`*.mom`) from the previous run should be provided. Appropriate paths are specified in the restart input file generated by RedMD upon user's request during the preceding run (see Section 9 and examples/ directory provided with RedMD).

8.6 Trajectory Analysis Tool

After performing the simulations, one can conduct a basic analysis of the generated trajectories using the `RedMD_analyze` tool.

`RedMD_Analyze`

The `RedMD_Analyze` tool allows the user to analyze the root mean square fluctuation (RMSF), the root mean square deviation (RMSD), the radius of gyration (Rg) or a distance between two selected atoms as a function of time. Currently, the `RedMD_Analyze` tool supports two trajectory formats: XYZ and PDB.

Usage:

`RedMD_Analyze MODE_OPTION [OPTIONS..]` *inputfile.(xyz;pdb)*

where possible options are:

Analysis options:
`--RMSD=`*initialStructureFile.(xyz; pdb)* - compute RMSD relative to a given structure (first frame)
`--RMSF=`*initialStructureFile.(xyz; pdb)* - compute RMSF realtive to a given structure (first frame)
`--Rg` - compute the radius of gyration
`--dist` - track the distance within a pair of atoms
`--freq=`*value* - a parameter defining the frequency of reading the trajectory frames; default is set to 1

Distance analysis options:
`--atom1=`*atomindex* - the index of the first atom in a pair
`--atom2=`*atomindex* - the index of the second atom in a pair

Output options:
`--outfile=`*name* - the name of the output file, by default *out.dat*
`--usage` - print short usage description and quit
`--help` - print help message and quit

9 Simulation Control File Keywords

In this section we list the valid RedMD keywords grouping them by their general function. We also give brief descriptions of their actions and modifiers.

9.1 Run Control

`number_of_processors` *number* - the number of processors to be used in case of a parallel execution; the default is the maximum number of available processors.

`number_of_steps` *number* - the number of integration steps to be performed; this parameter controls the total duration of the trajectory. The value of this keyword is additive, i.e., for the continuation/restart run, `number_of_steps` should be set as the sum of the number of steps already performed and the number of steps to be executed.

`dt` *value* - integration time step, [ps]

`xml` *name* - the path to the `*.sxml` structure file

`momentum` *name* - the path to the file with momenta for starting the simulation; if this keyword is not present the momenta are by default generated from the Maxwell distribution at temperature `T_sim`. The file should be in the plain ASCI format, with the number of lines corresponding to the total number of pseudo-atoms in the system and three columns giving p_x p_y p_z (with spaces or tabs separating the columns).

`seed` *number* - the seed for the random number generator; if this keyword is not specified the seed is generated based on the current system time.

`step` *number* - the step number to begin a restart simulation; this keyword appears automatically in the RedMD restart input file (it is meaningful only for continuation/restart runs) and, in general, its value should not be altered.

`T_sim` *value* - initial temperature, [K]

`T_end` *value* - target temperature, [K]

`freqTranRot` *value* - this keyword defines the frequency of removing the translation of the center of mass and the rotation around the center of mass.

9.2 Microcanonical Ensemble

`NVE` or `integration` *yes/no* - perform simulation in the microcanonical ensemble

`NVE_alg_type` *name* - defines the integration algorithm; possible choices are *VelocityVerlet* and *LeapFrog*

9.3 Canonical Ensemble

NVT or berendsen *yes/no* - perform simulation at constant temperature using the Berendsen algorithm to control the temperature.

tau *value* - time constant of the coupling to the heath bath in the Berendsen algorithm, [ps].

gamma *value* - fictious frictional coefficient in the Berendsen algorithm, related to τ (see Equation 50). Either gamma or tau should be explicitly set in the simulation control file, [ps^{-1}]

momentum_scaling *yes/no* - perform simulation at constant temperature using momentum scaling approach for temperature control.

9.4 Langevin Dynamics

langevin *yes/no* - perform stochastic Langevin dynamics simulation

gamma *value* - damping constant (or collision frequency), [ps^{-1}]

9.5 Brownian Dynamics

BD *yes/no* - perform stochastic Brownian dynamics simulation

hradius *value* - hydrodynamic radius of pseudo-atoms, [$\mathring{A}$]. If this keyword is not specified the program uses HR attributes of <ATOM> tags in the sxml.

9.6 Minimization

minimize *yes/no* - minimize the input structure
minimize_maxF *value* - the minimization stops if the maximal force acting on a particular atom is smaller than this value, [$\frac{kcal}{mol}$]
minimize_maxit *value* - the number of minimization steps
linesearch_maxit *value* - the number of steps performed during a line search

9.7 Monte Carlo

MC *yes/no* - perform Monte Carlo search in the dihedral angle space
number_of_steps *value* - the number of Monte Carlo steps
MC_nAngles *value* - the number of dihedral angles that are varied during a single Monte Carlo

move
`minimize_maxF` *value* - the minimization stops if the maximal force acting on a particular atom is smaller than this value, $[\frac{kcal}{mol}]$
`minimize_maxit` *value* - the number of minimization steps
`linesearch_maxit` *value* - the number of steps performed during a line search

9.8 Constraints

NOTE: To use constraints `--shake` or `-s` options MUST(!) be used while generating the structure `*.sxml` files (see Section 8.2). Additionally, if simulations are to be performed without any constraints, the structure `*.sxml` files MUST NOT(!) contain `shake` attributes for `BOND` tags (i.e., `shake=''true''` fields in the structure file should not appear, see Section 7.2.2).

`shake` *yes/no* - use the SHAKE algorithm to constrain bond lengths at their equilibrium values

`shake_tol` *value* - SHAKE tolerance (maximal allowable deviation from the equilibrium length), the default is 0.001, $[\mathring{A}]$

`shake_maxit` *number* - maximal number of SHAKE iterations, the default is 1000

`rattle` *yes/no* - use the RATTLE algorithm to constrain bond lengths at their equilibrium values

`rattle_tol` *value* - RATTLE tolerance (maximal allowable deviation from the equilibrium length), the default is 0.001, $[\mathring{A}]$

`rattle_maxit` *number* - maximal number of RATTLE iterations, the default is 1000

9.9 Output Control

`printfreq` *number of steps* - determines how frequently the information on the current simulation status is printed out; the default is 100

`dcd_file` *name* - the path to the output trajectory file in the DCD format

`dcdfreq` *number of steps* - determines how frequently the data is written to the DCD trajectory file; default 0

`out_file` *name* - the path to the general simulation output file

`outfreq` *number of steps* - determines how frequently the data is written to the output file; default 0

`pdb_file` *name* - the path to the output trajectory file in the PDB format

`pdbfreq` *number of steps* - determines how frequently the data is written to the PDB trajectory file; default 0

`vel_file` *name* - the path to the trajectory file containing pseudo-atoms' velocities

`velfreq` *number of steps* - determines how frequently the velocities are written to the trajectory file given with *vel_file*; default 0

`xyz_file` *name* - the path to the output trajectory file in the XYZ format

`xyz_freq` *number of steps* - determines how frequently the data is written to the XYZ trajectory file; default 0

`restart` *yes/no* - whether to create a restart file

`restart_input` *name* - the path to the file containing parameters needed to restart the simulation; default is *restart.inp*

`restart_coord` *name* - the path to the restart file with pseudo-atoms' coordinates, default is *restart.coord*

`restart_mom` *name* - the path to the restart file with pseudo-atoms' velocities, default is *restart.mom*

`restartfreq` *number of steps* - determines how frequently the restart data is saved; default 0

10 Examples

We provide a few examples that may help the RedMD users run their own simulations. They can be found in the `examples` folder in the RedMD source directory (after unpacking the RedMD `tgz` archive). Exemplary RedMD control files for setting up different kinds of simulations are also given below.

Brownian dynamics simulation

```
##################################
# RedMD simulation control file #
##################################

# give the name of the structure XML (*.sxml) file
xml molecule.sxml
# perform 100000 integration steps
number_of_steps 100000
# using the time step of 0.02ps
dt 0.02
# perform Brownian dynamics simulation at 300K
BD yes
T_end 300.0
# use SHAKE for bond lengths
# with the tolerance of 0.01Å
# and maximal number of SHAKE iterations 500
shake yes
shake_tol 0.01
shake_maxit 500
# each residue (pseudo-atom) is represented as a spherical particle
# with hydrodynamic radius of 3.2Å
hradius 3.2
# seed for the random number generator is 31415
seed 31415
# Do not remove overall translations and rotations of the system
freqTranRot 0
# print information to the standard output every 500 steps
printfreq 500
# name of the text file with energy output and
# how frequently to write to this file (every 500 steps)
out_file molecule.out
outfreq 500
# save trajectory using binary DCD format
# write to the trajectory file every 500 steps
dcd_file molecule.dcd
dcdfreq 500
```

```
# save restart/continuation data:
# restart control file is restart.inp
# file with coordinates is molecule.coord
# file with momenta is molecule.mom
# refresh restart data every 500 steps
restart yes
restart_input restart.inp
restart_coord molecule.coord
restart_mom molecule.mom
restartfreq 500
# use two nodes for computations
number_of_processors 2
```

NVT simulation with the Berendsen algorithm

```
##################################
# RedMD simulation control file #
##################################

# give the name of the structure XML (*.sxml) file
xml molecule.sxml
# perform 100000 integration steps
number_of_steps 100000
# using the time step of 0.01ps
dt 0.01
# perform the NVT molecular dynamics simulation
# using the Berendsen algorithm for temperature control
# using the coupling constant of 10ps
berendsen yes # or NVT yes
tau 10
# initially, assign atom velocities at 293K
# but couple the system to the heat bath and perform simulation at 300K
T_sim 293
T_end 300
# do not use SHAKE for bond lengths
shake no
# seed for the random number generator is 31415
seed 31415
# Remove overall translations and rotations of the system
# every 1000 steps
freqTranRot 1000
# print information to the standard output every 500 steps
printfreq 500
# name of the text file with energy output and
# the frequency for writing to this file (every 500 steps)
```

```
out_file molecule.out
outfreq 500
# save trajectory using PDB format
# write to the trajectory file every 500 steps
pdb_file molecule.pdb
pdbfreq 500
# save restart/continuation data:
# restart control file is restart.inp
# file with coordinates is molecule.coord
# file with momenta is molecule.mom
# refresh restart data every 500 steps
restart yes
restart_input restart.inp
restart_coord molecule.coord
restart_mom molecule.mom
restartfreq 500
# use two nodes for computations
number_of_processors 2
```

Langevin dynamics simulation

```
###################################
# RedMD simulation control file #
###################################

# give the name of the structure XML (*.sxml) file
xml molecule.sxml
# perform 100000 integration steps
number_of_steps 100000
# using the time step of 0.02ps
dt 0.02
# perform a Langevin dynamics simulation at 300K
langevin yes
T_end 300.0
# draw initial velocities at 300K
T_sim 300.0
# set the value of the damping constant to 2ps^-1
gamma 2
# seed for the random number generator is 31415
seed 31415
# do not remove overall translations and rotations of the system
freqTranRot 0
# print information to the standard output every 500 steps
printfreq 500
# name of the text file with energy output and
```

```
# the frequency of writing to this file (every 500 steps)
out_file molecule.out
outfreq 500
# save trajectory using XYZ format
# write to the trajectory file every 500 steps
xyz_file molecule.xyz
xyzfreq 500
# do not save restart/continuation data:
restart no
# use two nodes for computations
number_of_processors 2
```

Monte Carlo conformational search

```
##################################
# RedMD simulation control file #
##################################

# give the name of the structure XML (*.sxml) file
xml molecule.sxml
# 100000 Monte Carlo moves
number_of_steps 100000
# vary single dihedral during each MC move
MC_nAngles 1
# perform a Monte Carlo search at 300K
MC yes
T_end 300.0
# seed for the random number generator is 31415
seed 31415
# print information to the standard output every 500 steps
printfreq 500
# name of the text file with energy output and
# the frequency of writing to this file (every step)
out_file molecule.out
outfreq 1
# save trajectory using XYZ format
# write to the trajectory file every step
xyz_file molecule.xyz
xyzfreq 1
# minimization parameters
minimize_maxit 100
minimize_maxF 0.001
linesearch_maxit 10
# use two nodes for computations
number_of_processors 2
```

11 Normal Mode Analysis

We provide a set of tools to perform coarse-grained normal mode analysis of molecular systems through diagonalization of the Hessian matrix (the matrix of the second-order partial derivatives of the potential energy function). The eigenproblem for the Hessian matrix is solved either using LAPACK (http://www.netlib.org/lapack/) routines, or, if a limited set of lowest eigenvalues (thus lowest frequency molecular motions) is requested, with routines from the ARPACK package (http://www.caam.rice.edu/software/ARPACK/).

11.1 RedENM

The user can use either the Tirion's [1] elastic network model or the anisotropic network model with nonuniform force constants of elastic springs [34, 35]. A molecular system is represented on the residue level – amino acids are replaced with C_α and nucleic acids with P pseudo-atoms. Beads are assigned unit masses or, optionally, total masses of corresponding residues, i.e., a mass-weighted Hessian matrix is used. The output of RedENM consists of the full Hessian matrix in the sparse format, eigenvalues and eigenvectors, relative frequencies of molecular motions, collectivities of normal modes [36], and normalized temperature factors (or B-factors). Additionally, the XYZ-formatted trajectories are outputed to illustrate motions along each of the normal mode vectors.

11.1.1 Usage

The RedENM module is interactive. To perform normal mode analysis on an input file in the PDB format, *name.pdb*, the user needs to execute the following command:

```
RedENM name.pdb
```

and then set up the parameters from the command line, as presented below (values entered by the user are indicated in red):

```
Enter cutoff for CA:CA pairs [A]
[cutoff_CA_CA] = 12
Cutoff for CA:CA pairs set to:  12.000000 [A]

Mass weighted analysis?  (yes/no)
[mwa] = yes
Mass weighted analysis will be performed

Enter the change in the RMSD for scaling
motions along eigenvectors
[motion_scale] = 1.5
RMSD change set to:  1.500000

Enter the interactions scaling factor
typically in the range 0 - 3;
```

```
default for ENM 0.0
values different than 0.0 for ANM
[p] = 0
Interactions scaling factor set to:  0.000000
ENM (harmonic) model will be used

Computing Hessian 594x594 matrix.
Computing off-diagonal terms ...
Computing diagonal terms ...

LAPACK/ARPACK subroutines are used
to diagonalize Hessian

How many modes (eigenvalues) should be computed?
Enter a number (0 = all)
[NUM_eig] = 0

All eigenvalues will be computed
using LAPACK's DSYEV.

How many modes (trajectories) should be written
to separate *.xyz files?
[NUM_traj] = 24
Number of trajectories set to 24

How many lowest modes should be used for
B-factors calculations?
[cutoff] = 594
Number of lowest modes set to 594

Enter temperature [K] for B-factors calculations
[Temp] = 100
Temperature set to:  100.000000
```

Upon execution, the program prints on the screen the information about the current stage of calculations.

11.1.2 Output

Files resulting from executing RedENM are the following:

`H_matrix.dat` - upper triangle of the Hessian matrix, sparse format
`Masses.dat` - masses of all pseudo-atoms present in the system
`Frequencies.dat` - eigenvalues of the Hessian and relative frequencies
`Eigenvectors.dat` - eigenvectors

`Collectivity.dat` - collectivities of modes
`Reduced.pdb` - reduced representation of the molecule
`B_factors.dat` - temperature factors
`B_factors_norm.dat` - normalized temperature factors
`log` - log file
`number.xyz` - trajectories used to visualize motions along eigenvectors (names correspond to normal modes sorted according to their frequency); first 6 trivial modes (0-5) are skipped

11.2 RedGNM

RedGNM program is based on the modified Gaussian network model for proteins (Chemical Network Model) described in [37]. The total Hessian matrix is a linear combination of two Kirchoff (contact) matrices H_c and H_n:

$$H_{total} = k_{covalent} H_c + k_{noncovalent} H_n$$

H_c considers covalent bonds between neighboring residues of the same chain. If some residues in the chain are missing, a covalent bond between the last and the first residue of the gap is added. H_n contact matrix is computed using the residues that are not neighbors in the same chain but the closest distance between their nonhydrogen atoms is smaller than a user-specified cut-off. $k_{covalent}$ is set to 1.0 and only $k_{noncovalent}$ can be modified. The program optionally outputs the Hessian matrix, the user-defined number of eigenvalues and eigenvectors, as well as the sums:

$$B_k = \sum_{i=1}^{n} \frac{v_{ik}^2}{\lambda_i},$$

where v_{ik} is the k-th coordinate of the i-th eigenvector and n is the user-specified number of eigenvectors (counting from the smallest eigenvalues, excluding the first eigenvalue which is equal to zero and corresponds to translations of the molecule). More eigenvalues equal to zero indicate that the interaction network is not continuous and result in an error message. If all eigenvalues are considered, B_k is proportional to the variance of any of the k-th residue's coordinate and it is further referred to as theoretical B-factor or temperature factor. In place of the temperature factors, RedGNM can also output PDB files with squares of a given vector's coordinate, normalized so that the biggest value is equal to 99.00. These values would be proportional to the variance of any of the residue's coordinate if the molecule's oscillations along this coordinate were restricted to the given eigenvector. Thus, they can be used to visualize the mobility of molecule's regions in a particular oscillation mode.

11.2.1 Usage

To use RedGNM program one has to execute the command:

`./RedGNM` *configFile.xml*

where optional parameter configFile.xml is the name of the configuration file. If no parameter

is given, a default configuration file config/GNMConfig.xml is used. Its content is presented below.

```
<?xml version="1.0"?>
<GNMConfigData>
    <cutoff> 7.0 </cutoff>
    <HMatrix fileName="outputFiles/HMatrix.dat"/>
    <BFactors fileName="outputFiles/BFactors.dat"/>
    <eigenvalues fileName="outputFiles/Eigenvalues.dat"/>
    <eigenvectors fileName="outputFiles/Eigenvectors.dat"/>
    <reduced fileName="outputFiles/Reduced.pdb"/>
    <PDB fileName="resources/1fnm.pdb"/>
    <noncovalentParam> 0.1 </noncovalentParam>
    <numEig> 100 </numEig>
    <tempFactor fileName="outputFiles/tempFact">
        <first> 1 </first>
        <last> 10 </last>
    </tempFactor>
</GNMConfigData>
```

The root element with a start-tag `<GNMConfigData>` must contain the elements: `cutoff`, `noncovalentParam` and `numEig`. `Cutoff` and `noncovalentParam` enclose the numbers defining the cut-off and the $k_{noncovalent}$ parameter of the model, respectively. The number stored in the `<numEig>` element describes how many eigenvalues and eigenvectors will be taken into account, excluding the first translational one. If the given number is zero, or is greater or equal to the number of all non-zero eigenvalues, all of them will be considered. Another obligatory element has a start-tag `<PDB>`, with its attribute `fileName` specifying the name of the input PDB file.

Optional elements `<HMatrix>`, `<BFactors>`, `<eigenvalues>`, `<eigenvectors>`, and `<reduced>`, each with an attribute `fileName`, specify the outputs of the hessian matrix elements in a sparse matrix format, the theoretical B-factors, eigenvalues or eigenvectors, and the reduced protein representation. If an element is omitted, the corresponding data is not saved. Each PDB file with temperature factors corresponding to the movement restricted to the i-th eigenspace is saved as the tempFactor-named element's `fileName` attribute concatenated with 'i.pdb'. The range of i is defined by the values enclosed in the `<tempFactor>` subelements `first` and `last`.

References

[1] M. Tirion. Large amplitude elastic motions in proteins from a single-parameter, atomic analysis. *Phys. Rev. Lett.*, 77:1905–1908, 1996.

[2] J. Trylska, V. Tozzini, and J. A. McCammon. Exploring global motions and correlations in the ribosome. *Biophys. J.*, 89:1455–1463, 2005.

[3] K. Voltz, J. Trylska, V. Tozzini, V. Kurkal-Siebert, J. Langovski, and J. Smith. Coarse-grained force field for the nucleosome from self-consistent multiscaling. *J. Comp. Chem.*, 29:1429–1439, 2008.

[4] V. Tozzini and J. A. McCammon. A coarse grained model for the dynamics of flap opening in HIV-1 protease. *Chem. Phys. Lett.*, 413:123–128, 2005.

[5] V. Tozzini, J. Trylska, C. E. Chang, and J. A. McCammon. Flap opening dynamics in HIV-1 protease explored with a coarse-grained model. *J. Struct. Biol.*, 157:606–615, 2007.

[6] K. Moritsugu and J. C. Smith. Coarse-grained biomolecular simulation with REACH: Realistic extension algorithm via covariance Hessian. *Biophys. J.*, 93:3460–3469, 2007.

[7] F. Trovato and V. Tozzini. Supercoiling and local denaturation of plasmids with a minimalist DNA model. *J. Phys. Chem. Lett.*, 112:13197–13200, 2008.

[8] F. Leonarski, F. Trovato, V. Tozzini, A Leś, and J. Trylska. Evolutionary algorithm in the optimization of a coarse–grained force field. *J. Chem. Theory Comput.*, 9:4874–4889, 2013.

[9] F. Trovato, R. Nifosi, A. Di Fenza, and V. Tozzini. A minimalist model of protein diffusion and interactions: the green fluorescent proiten within the cytoplasm. *Macromolecules*, 46:8311–8322, 2013.

[10] A. Mukherjee and B. Bagchi. Contact pair dynamics during folding of two small proteins: Chicken villin head piece and the alzheimer protein beta-amyloid. *J. Chem. Phys.*, 120:1602–1612, 2004.

[11] A. Mukherjee and B. Bagchi. Correlation between rate of folding, energy landscape, and topology in the folding of a model protein hp-36. *J. Chem. Phys.*, 118:4733–4747, 2003.

[12] M. Levitt. A simplified representation of protein conformations for rapid simulations of protein folding. *J. Mol. Biol.*, 104:59–107, 1976.

[13] T. Schlick. *Molecular Modelling and Simulations: an Interdisciplinary Guide*. Interdisciplinary Applied Mathematics, Vol 21. Springer-Verlag, New York, NY, 2002.

[14] A. R. Leach. *Molecular Modelling. Principles and Applications*, volume second edition. Pearson Education Ltd., England, 2001.

[15] L. Verlet. Computer 'experiments' on classical fluids: I. Thermodynamical properties of Lennard-Jones molecules. *Phys. Rev.*, 159:98–103, 1967.

[16] W. C. Swope, H. C. Andersen, P. H. Berens, and K. R. Wilson. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *J. Chem. Phys.*, 76:637–649, 1982.

[17] R. W. Hockney and J. W. Eastwood. *Computer simulation using particles*. McGraw-Hill, New York, NY, 1981.

[18] M. E. Tuckerman, Berne B. J., and G. J. Martyna. Reversible multiple time scale molecular dynamics. *J. Chem. Phys.*, 97:1990–2001, 1992.

[19] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. Dinola, and J. R. Haak. Molecular dynamics with coupling to an external bath. *J. Chem. Phys.*, 81:3684–3690, 1984.

[20] S. C. Harvey, R. K. Z. Tan, and T. E. Cheatham III. The flying ice cube: Velocity rescaling in molecular dynamics leads to violation of energy equipartition. *J. Comput. Chem.*, 19:726–740, 1998.

[21] R. W. Pastor, B. R. Brooks, and A. Szabo. An analysis of the accuracy of Langevin and molecular dynamics algorithms. *Mol. Phys.*, 65:1409–1419, 1988.

[22] R. W. Pastor. *Techniques and applications of Langevin dynamics simulations*. In G. R. Luckhurst and C. A. Veracini, editors, The Molecular Dynamics of Liquid Crystals. Kluwer Academic, Dordrecht, The Netherlands, 1994.

[23] A. Brünger, C. L. Brooks III, and M. Karplus. Stochastic boundary conditions for molecular dynamics simulations of st2 water. *Chem. Phys. Lett.*, 105:495–500, 1982.

[24] D. L. Ermak and J. A. McCammon. Brownian dynamics with hydrodynamic interactions. *J. Chem. Phys.*, 69:1352–1360, 1978.

[25] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Verlag, New York, NY, 1999.

[26] Z. Li and H.A. Scheraga. Monte carlo-minimization approach to the multiple-minima problem in protein folding. *Proc. Natl. Sci. USA*, 84:6611–6615, 1987.

[27] N.A. Metropolis, A.W. Rosenbluth, A.H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1092, 1953.

[28] J. P. Ryckaert, G. Ciccotti, and H. J. C. Berendsen. Numerical integration of the cartesian equations of motion of a system with constraints: Molecular dynamics of n-alkanes. *J. Comput. Phys.*, 23:327–341, 1977.

[29] H.C. Andersen. Rattle: a velocity version of the shake algorithm for molecular dynamics calculations. *J. Comput. Phys.*, 52:24–34, 1983.

[30] F. C. Bernstein, T. F. Koetzle, G. J. B. Williams, J. E. F. Meyer, M. D. Brice, J. R. Rodgers, T. Shimanouchi O. Kennard, and M. Tasumi. The protein data bank: A computer-based archival file for macromolecular structures. *J. Mol. Biol.*, 112:535–542, 1977.

[31] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus. CHARMM: a program for macromolecular energy, minimization, and dynamics calculations. *J. Comp. Chem.*, 4(2):187–217, 1983.

[32] C. D. Schwietes, J. J. Kuszewski, N. Tjandra, and G. M. Clore. The Xplor-NIH NMR molecular structure determination package. *J. Magn. Res.*, 160:66–74, 2003.

[33] W. Humphrey, A. Dalke, and K. Schulten. VMD – Visual Molecular Dynamics. *J. Molec. Graph.*, 14:33–38, 1996.

[34] P. Doruker, A. R. Atilgan, and I. Bahar. Dynamics of proteins predicted by molecular dynamics simulations and analytical approaches: application to alpha-amylase inhibitor. *Proteins: Struct., Funct., Bioinf.*, 15:512–524, 2000.

[35] A. R. Atilgan, S. R. Durrell, R. L. Jernigan, M. C. Demirel, O. Keskin, and I. Bahar. Anisotropy of fluctuation dynamics of proteins with an elastic network model. *Biophys. J.*, 80:505–515, 2001.

[36] D. Ming and R. Brüschweiler. Reorientational contact-weighted elastic network model for the prediction of protein dynamics: Comparison with NMR relaxation. *Biophys. J.*, 90(10):3382–3388, 2006.

[37] D. A. Kondrashov, Q. Cui, and G. N. Phillips Jr. Optimization and evaluation of a coarse-grained model of protein motion using x-ray crystal data. *Biophys. J.*, 91:2760–2767, 2006.