oxDNA Workshop

Reem Mokhtar & Tong Niu

Duke University

September 8, 2015

Overview

1. "Coarse-graining DNA for simulations of DNA nanotechnology.",2013
2. "Introducing improved structural properties and salt dependence into a coarse-grained model of DNA.",2015
3. Usage

"Coarse-graining DNA for simulations of DNA nanotechnology.",2013

What can oxDNA do?

Capture biophysical processes

1. Structural
2. Thermodynamic
3. Mechanical
4. Can be simulated on diffusive time-scales

Quantitative reproduction of phenomena not fitted

1. Kinetics of toehold mediated strand exchange
2. Overstretching force of duplex DNA

Insight into dynamic processes

1. hybridization
2. strand exchange
3. hairpin formation

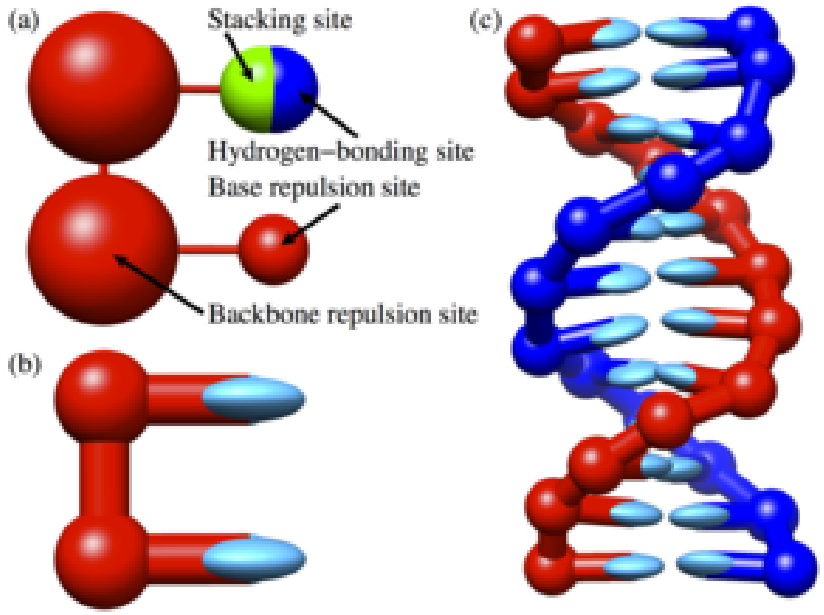
Response to mechanical stress

1. stretching
2. twisting
3. bending

oxDNA Model

The model treats DNA as:

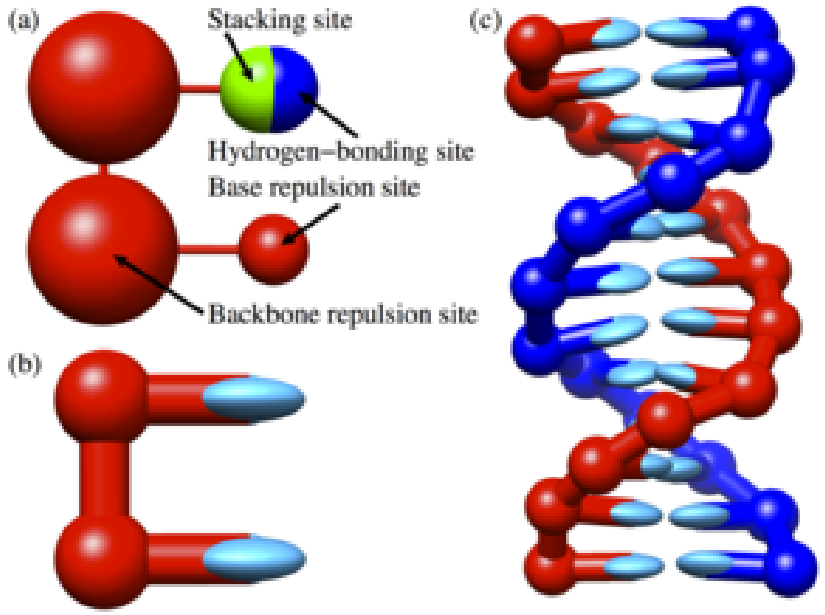
1. A string of rigid nucleotides
2. Which interact through potentials
3. Which depend on the position and orientation of the nucleotides



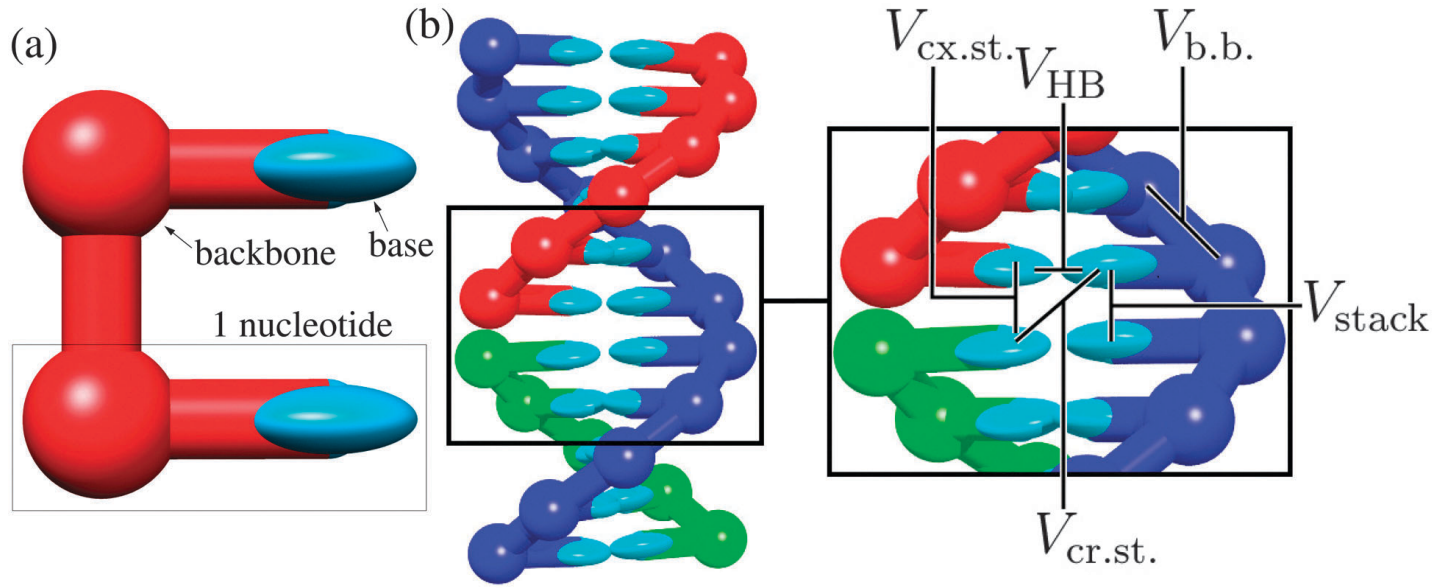
oxDNA Model

The interactions are:

1. Sugar-phosphate backbone connectivity,
2. Excluded volume,
3. Hydrogen bonding,
4. Nearest-neighbour stacking,
5. Cross-stacking between base-pair steps in a duplex,
6. Coaxial stacking.



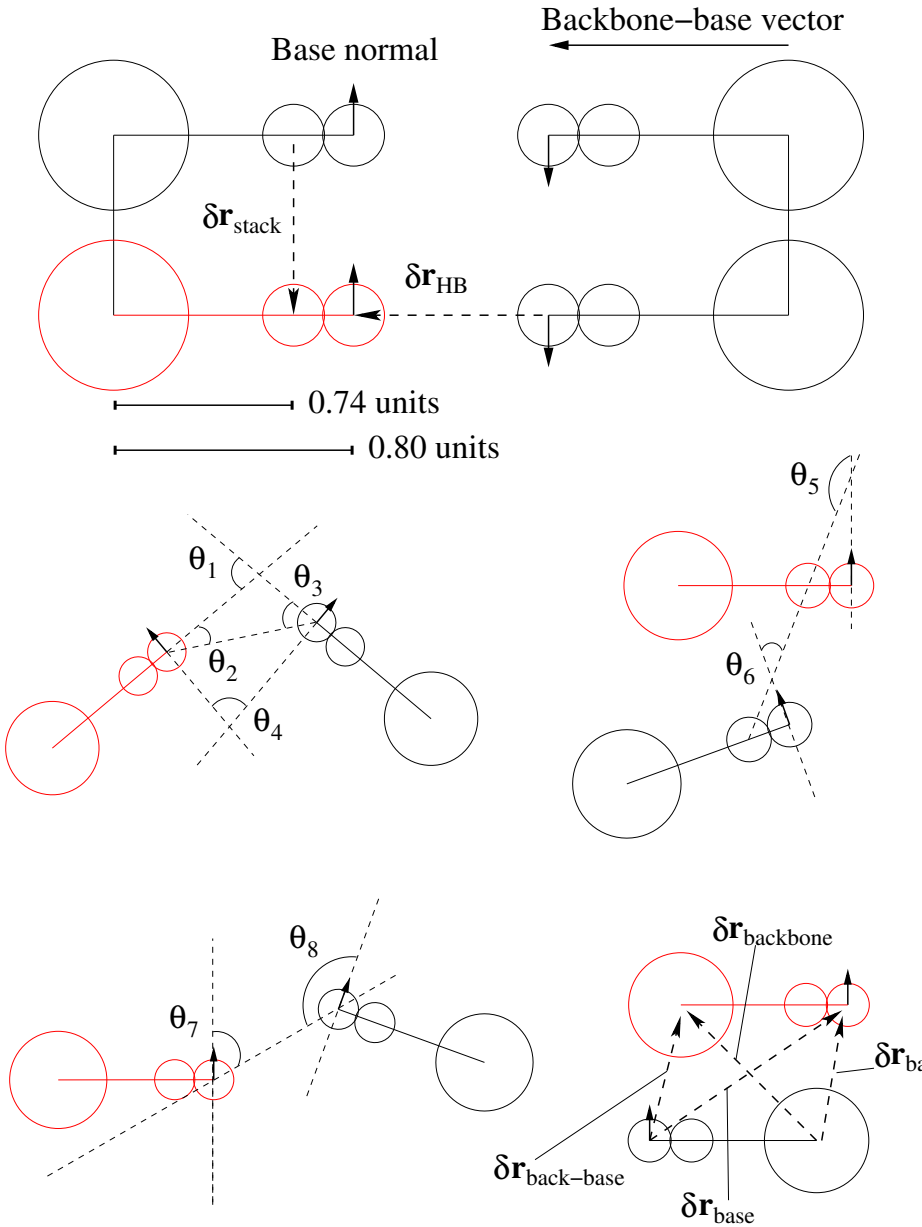
oxDNA Model



The form of internucleotide potential:

$$V_{\text{oxDNA}} = \sum_{<ij>} (V_{\text{b. b.}} + V_{\text{stack}} + V_{\text{exc}'}) + \sum_{i,j \notin <ij>} (V_{\text{HB}} + V_{\text{cr.st.}} + V_{\text{exc}} + V_{\text{cx.st.}})$$

oxDNA Model[1]



$$
\begin{aligned}
V_{\mathrm{HB}} &= f_1(\delta r_{\mathrm{HB}}, \epsilon_{\mathrm{NB}}, a_{\mathrm{HB}}, \delta r^0_{\mathrm{HB}}, \delta r^{c,low}_{\mathrm{HB}}, \delta r^{c,high}_{\mathrm{HB}}, \delta r^{low}_{\mathrm{HB}}, \delta r^{high}_{\mathrm{HB}}) \\
&\times f_4(\theta_1, a_{\mathrm{HB},1}, \theta^0_{\mathrm{HB},1}, \Delta\theta^\star_{\mathrm{HB},1})\ f_4(\theta_2, a_{\mathrm{HB},2}, \theta^0_{\mathrm{HB},2}, \Delta\theta^\star_{\mathrm{HB},2}) \\
&\times f_4(\theta_3, a_{\mathrm{HB},3}, \theta^0_{\mathrm{HB},3}, \Delta\theta^\star_{\mathrm{HB},3})\ f_4(\theta_4, a_{\mathrm{HB},4}, \theta^0_{\mathrm{HB},4}, \Delta\theta^\star_{\mathrm{HB},4}) \\
&\times f_4(\theta_7, a_{\mathrm{HB},7}, \theta^0_{\mathrm{HB},7}, \Delta\theta^\star_{\mathrm{HB},7})\ f_4(\theta_8, a_{\mathrm{HB},8}, \theta^0_{\mathrm{HB},8}, \Delta\theta^\star_{\mathrm{HB},8}).
\end{aligned}
$$

- The radial part of the stacking and hydrogen-bonding potentials:

$$
f_1(r) = \begin{cases}
V_{\mathrm{Morse}}(r, \epsilon, r^0, a) - V_{\mathrm{Morse}}(r^c, \epsilon, r^0, a)) & \text{if } r^{low} < r < r^{high}, \\
\epsilon V_{\mathrm{smooth}}(r, b^{low}, r^{c,low}) & \text{if } r^{c,low} < r < r^{low}, \\
\epsilon V_{\mathrm{smooth}}(r, b^{high}, r^{c,high}) & \text{if } r^{high} < r < r^{c,high}, \\
0 & \text{otherwise.}
\end{cases} \tag{2.7}
$$

- The angular modulation factor used in stacking, hydrogen-bonding, cross-stacking and coaxial stacking:

$$
f_4(\theta) = \begin{cases}
V_{\mathrm{mod}}(\theta, a, \theta^0) & \text{if } \theta^0 - \Delta\theta^\star < \theta < \theta^0 + \Delta\theta^\star, \\
V_{\mathrm{smooth}}(\theta, b, \theta^0 - \Delta\theta^c) & \text{if } \theta^0 - \Delta\theta^c < \theta < \theta^0 - \Delta\theta^\star, \\
V_{\mathrm{smooth}}(\theta, b, \theta^0 + \Delta\theta^c) & \text{if } \theta^0 + \Delta\theta^\star < \theta < \theta^0 + \Delta\theta^c, \\
0 & \text{otherwise.}
\end{cases} \tag{2.10}
$$

- Quadratic smoothing terms for truncation:

$$
V_{\mathrm{smooth}}(x, b, x^c) = b(x^c - x)^2. \tag{2.6}
$$

- Morse potential (used for stacking and H-bonding):

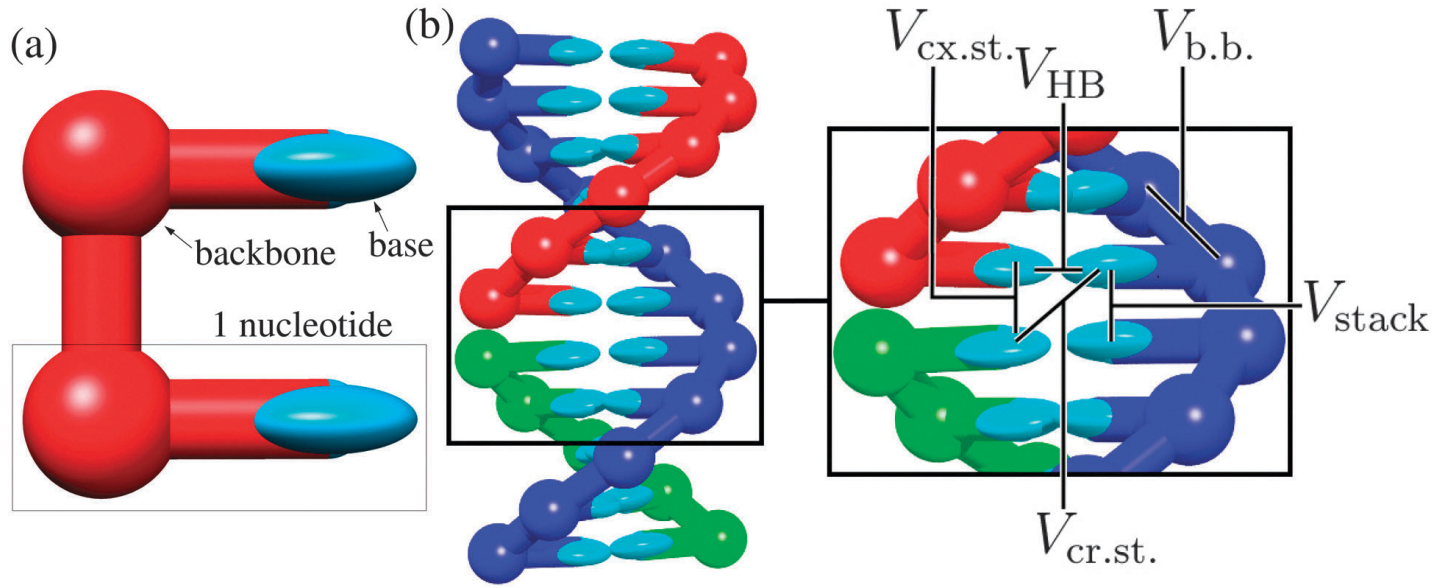
$$
V_{\mathrm{Morse}}(r, \epsilon, r^0, a) = \epsilon\big(1 - \exp\left(-(r - r^0)a\right)\big)^2. \tag{2.2}
$$

- Quadratic terms (used for modulation):

$$
V_{\mathrm{mod}}(\theta, a, \theta^0) = 1 - a(\theta - \theta^0)^2. \tag{2.5}
$$

[1] Thomas E Ouldridge (2012). "Coarse-grained modelling of DNA and DNA self-assembly". In: URL: http://www.lavoisier.fr/livre/notice.asp?id=2XKW32AOARXOWE.

oxDNA Model



Structural Properties

pitch, base-pair rise and radius, propeller twist, and distinction between major and minor grooves (new! see example in oxDNA folder MAJOR_MINOR_GROOVING)

Features

1. Molecular and Brownian dynamics
2. Monte Carlo simulations
3. Regular Metropolis Monte Carlo simulations
4. Additional interactions: oxDNA can be extended to simulate additional pairwise potentials.
5. External forces: In order to favor motif formation or to mimic different external environments, different kind of forces can be applied to nucleotides or points in space.
6. Standalone single- and double-strand generator
7. Output converter
8. Cadnano converter

"Introducing improved structural properties and salt dependence into a coarse-grained model of DNA.",2015

1. "Coarse-graining DNA for simulations of DNA nanotechnology.",2013
2. "Introducing improved structural properties and salt dependence into a coarse-grained model of DNA.",2015
   - Differences between oxDNA2 and oxDNA
3. Usage

Differences between oxDNA2 and oxDNA

oxDNA

1. The interactions for the different bases are identical except for the hydrogen-bonding term (only Watson-Crick base pairs).
2. For thermodynamics, uses SantaLucia[2] for melting of short duplexes

oxDNA2

1. Different widths for major and minor grooves.
2. Including term for salt-dependent interaction
3. Differentiating between AA and TT stacking interaction (instead of fitting using SantaLucia). Evidence that contiguous AA are much stiffer than TT.

---

[2] John SantaLucia and Donald Hicks (2004). "The thermodynamics of DNA structural motifs." In: *Annual Review of Biophysics and Biomolecular Structure* 33.1, pp. 415–440. DOI: 10.1146/annurev.biophys.32.110601.141800. URL: http://www.annualreviews.org/doi/abs/10.1146/annurev.biophys.32.110601.141800.

Differences between oxDNA2 and oxDNA

The interactions are:

1. Sugar-phosphate backbone connectivity,
2. Excluded volume,
3. Hydrogen bonding,
4. Nearest-neighbour stacking,
5. Cross-stacking between base-pair steps in a duplex,
6. Coaxial stacking.
7. Electrostatic interactions via Debye-Huckel-like term (DH)

$$
\begin{aligned}
V_{\mathrm{oxDNA2}} &= \sum_{\substack{\text{nearest}\\ \text{neighbours}}} (V^*_{\mathrm{backbone}} + V^*_{\mathrm{stack}} + V'_{\mathrm{exc}}) \\
&\quad + \sum_{\substack{\text{other}\\ \text{pairs}}} (V^*_{\mathrm{HB}} + V_{\mathrm{cross\ stack}} + V_{\mathrm{exc}} + V^*_{\mathrm{coax\ stack}} + V^*_{\mathrm{DH}}),
\end{aligned}
\tag{12}
$$

Input Files

Files

1. Configuration
    1. general information (timestep, energy, box size)
    2. orientation, positions of each nucleotide
2. Topology: backbone-backbone bonds between nucleotides in the same strand

Input Files

Files

1. Configuration
   1. general information (timestep, energy, box side)
   2. orientation, positions of each nucleotide
2. Topology: backbone-backbone bonds between nucleotides in the same strand

$$\overbrace{r_x r_y r_z}^{\text{Position}} \quad \overbrace{b_x b_y b_z}^{\text{Backbone}-\text{baseversor}} \quad \overbrace{n_x n_y n_z}^{\text{Normalversor}} \quad \overbrace{v_x v_y v_z}^{\text{Velocity}} \quad \overbrace{L_x L_y L_z}^{\text{Angularvelocity}}$$

File `relaxed.conf` under `oxDNA/EXAMPLES/CADNANO_INTERFACE/TILE`

```
t = T
b = Lz Ly Lz
E = Etot U K
```

Input Files

Files

1. Configuration
   1. general information (timestep, energy, box size)
   2. orientation, positions of each nucleotide
2. Topology: backbone-backbone bonds between nucleotides in the same strand

File `topology.dat` under `oxDNA/EXAMPLES/CADNANO_INTERFACE/TILE`

```
N Ns
S# B 3′bond 5′bond
```

Output: Energy File Layout

MD Simulations

1. time (steps * dt)
2. potential energy
3. kinetic energy
4. total energy

Note

Potential, kinetic and total energies are divided by the total number of particles.

Example MD

```
0.0000 -1.032758 0.413539 -0.619219
500.0000 -0.664228 0.359874 -0.304355
1000.0000 -0.651344 0.321058 -0.330286
...
```

Output: Energy File Layout

MC simulations

1. time (steps)
2. potential energy
3. acceptance ratio for translational moves
4. acceptance ratio for rotational moves
5. acceptance ratio for volume moves

Example MC

```
0.0000 -1.032758 0.413539 -0.619219
500.0000 -0.664228 0.359874 -0.304355
1000.0000 -0.651344 0.321058 -0.330286
...
```

Output: Energy File Layout

VMMC simulations

1. time (steps)
2. potential energy
3. acceptance ratio for translational moves
4. acceptance ratio for rotational moves
5. acceptance ratio for volume moves
6. if umbrella sampling enabled: [order parameter coordinate 1] [order parameter coordinate 1] ... [order parameter coordinate n] [current weight]

Example VMMC

```
0.0000 -1.032758 0.413539 -0.619219
500.0000 -0.664228 0.359874 -0.304355
...
```

Usage

`https://dna.physics.ox.ac.uk/index.php/Documentation`

To generate the topology and configuration files from a specific sequence, we use `generate-sa.py` under `oxDNA/UTILS/`:

1. Box side (ensuring no change in number of particles)
2. Input sequence

Example Sequence File

```
DOUBLE AGGGCT
CCTGTA
```

Generate

generate-sa.py 3 sequence_file

Usage

To generate the topology and configuration files from a specific sequence, we use `generate-sa.py` under `oxDNA/UTILS/`:

1. Box side
2. Input sequence

Example Sequence File

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

Generate

generate-sa.py 20. sequence_file

Usage

Input File

```
topology = generated.top
conf_file = generated.conf
trajectory_file = trajectory.dat
#log_file = log.dat
print_conf_interval = 1000000
time_scale = linear
external_forces=0
```

Generate Trajectory

```
oxDNA inputfile
```

Usage

Visualization

Under `UTILS`:

```
traj2vis xyz trajectory_file topology_file
```

Analyze bonds

Under `UTILS`:

```
output_bonds input_file trajectory_file [count]
```

Input File

Syntax

1. case-sensitive
2. options in the documentation are in form of key = value
3. arbitrary spaces
4. leading # for comments
5. pipe — sign between values separates alternative values
6. default value is right after equal sign

Generic Options

`interaction_type = DNA | DNA2 | RNA | patchy | LJ`

Simulation model

Generic Options

`sim_type = MD | MC | VMMC`

Molecular Dynamics, Monte Carlo, or Virtual Move Monte Carlo

Generic Options

`backend = CPU | CUDA`

CUDA backend is only supported by sim_type=MD.

Energy File

Energy File Layout

The energy file layout for MD simulations is
[time (steps * dt)] [potential energy] [kinetic energy] [total energy]

Note

Potential, kinetic and total energies are divided by the total number of particles.

Example: Hairpin Formation

VMMC simulations

1. Sequence-averaged (SA) model.
2. Sequence-dependent (SD) model.
3. SA model in which two base pairs are connected by mutual traps

Monte Carlo

Repeatedly randomly sample a volume in d-dimensional space to obtain an estimate of an integral at the price of a statistical error.

Monte Carlo methods

Repeated random sampling of a volume in d-dimensional space to obtain an estimate of an integral at the price of a statistical error.

General pattern for pseudocode

1. Define a domain of possible inputs.
2. Generate inputs randomly from a probability distribution over the domain.
3. Perform a deterministic computation on the inputs.
4. Aggregate the results.

VMMC

- Problem: sequential updates of particles (which leads to low acceptance rates when attractions are strong, leading to strong suppression of collective motion).
- Algorithm avoids this problem by proposing simultaneous moves of collections "clusters" of particles according to gradients of interaction energies.

Sequence-averaged (SA) model

Stacking Parameters

- For stacking energies in ssDNA, uses the thermodynamic results in Jill A Holbrook et al.,1999, with the exception of AA and TT stacking not distinguished (in oxDNA)

Files

Input:	inputMD
HB energy:	hb_energy.dat
Energy:	energy.dat
Trajectory:	trajectory.dat
Last Trajectory File:	last_conf.dat
Log file:	log.dat

Sequence-dependent (SD) model

Files

Input:	inputMD_seq_dep
HB energy:	hb_energy_seq_dep.
Energy:	energy_seq_dep.dat
Trajectory:	trajectory_seq_dep.
Last Trajectory File:	last_conf_seq_dep.d
Log file:	log_seq_dep.dat

SA model in which two base pairs are connected by mutual traps

Files

Input:	inputTRAP
HB energy:	hb_energy_trap.dat
Energy:	energy_trap.dat
Trajectory:	trajectory_trap.dat
Last Trajectory File:	last_conf_trap.dat
Log file:	log_trap.dat