A Primer on the oxDNA Model of DNA (Sengar, et al 2021)

A Primer on the oxDNA Model & Simulation Software
Topics:

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
Example Pseudoknot
• In unbound (A) and bound (B) states,
• for sequences 3′-AGC TTC CAT G-5’/3′-AAG CTC ATG G-5’.

What is the 3D Structure of the Pseudoknot?

Use oxDNA to calculate this!
What can oxDNA do?

Capture biophysical processes

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

Insight into dynamic processes

1. hybridization
2. strand exchange
3. hairpin formation

Quantitative reproduction of phenomena not fitted

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

Response to mechanical stress

1. stretching
2. twisting
3. bending
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
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.
The form of internucleotide potential:

\[ V_{\text{oxDNA}} = \sum_{\langle ij \rangle} (V_{\text{b. b.}} + V_{\text{stack}} + V_{\text{exc}}) + \sum_{i,j \notin \langle ij \rangle} (V_{\text{HB}} + V_{\text{cr.st.}} + V_{\text{exc}} + V_{\text{cx.st.}}) \]

For Consecutive bases \( i,j \)

For Nonconsecutive bases \( i,j \)
Potential Energy Equation of oxDNA Simulation Model:

\[ V_0 = \sum_{\langle ij \rangle} (V_{b.b.} + V_{stack} + V_{\text{exc}}) + \sum_{i,j \notin \langle ij \rangle} (V_{HB} + V_{cr.st.} + V_{\text{exc}} + V_{\text{coax}}) \]

with an additional screened electrostatic repulsion term.

Summations in Potential Energy Equation:
- The first sum of the equation is taken over all pairs of nucleotides that are nearest neighbors on the same strand and
- The second sum comprises all remaining pairs.

The terms of the Potential Energy Equation represent:
- **Backbone connectivity** \((V_{b.b.})\)
  - Is a spring potential mimicking the covalent bonds along the strand.
- **Excluded volume** \((V_{\text{exc}} \text{ and } V'_{\text{exc}})\)
  - A function of the distance between repulsion sites.
- **Hydrogen bonding between complementary bases** \((V_{HB})\),
- **Stacking between adjacent bases on a strand** \((V_{\text{stack}})\),
- **Cross-stacking** \((V_{cr.st.})\) across the duplex axis and
- **Coaxial stacking** \((V_{\text{coax}})\) across a nicked backbone.

How the terms are calculated:
- The excluded volume and backbone interactions are a function of the distance between repulsion sites.
- The backbone potential is a spring potential mimicking the covalent bonds along the strand.
- All other interactions depend on the relative orientations of the nucleotides and the distance between the hydrogen-bonding and stacking interaction sites.
Details of oxDNA Model

\[ V_{HB} = f_1(\delta r_{HB}, \epsilon_{NB}, a_{HB}, \delta r_{HB}^0, \delta r_{HB}^{c,low}, \delta r_{HB}^{c,high}, \delta r_{HB}^0, \delta r_{HB}^{high}) \times f_4(\theta_1, a_{HB,1}, \theta_{HB,1}, \Delta \theta_{HB,1}, \theta_{HB,2}, \Delta \theta_{HB,2}) \times f_4(\theta_3, a_{HB,3}, \theta_{HB,3}, \Delta \theta_{HB,3}, \theta_{HB,4}, \Delta \theta_{HB,4}) \times f_4(\theta_7, a_{HB,7}, \theta_{HB,7}, \Delta \theta_{HB,7}, \theta_{HB,8}, \Delta \theta_{HB,8}). \]

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

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

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

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

- Quadratic smoothing terms for truncation:

\[ V_{smooth}(x, b, x^c) = b(x^c - x)^2. \] (2.6)

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

\[ V_{Morse}(r, \epsilon, r^0, a) = \epsilon(1 - \exp(-(r - r^0)a))^2. \] (2.2)

- Quadratic terms (used for modulation):

\[ V_{mod}(\theta, a, \theta^0) = 1 - a(\theta - \theta^0)^2. \] (2.5)

\[1\text{Thomas E Ouldridge (2012). “Coarse-grained modelling of DNA and DNA self-assembly”. In: URL: http://www.lavoisier.fr/livre/notice.asp?id=2XKW32A0ARXOWE.}\]
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)
“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
Comparison of structure in oxDNA1.0 and oxDNA1.5 vs oxDNA2.0.

• In the earlier version of the model, all interaction sites are co-linear;
• In oxDNA2.0, offsetting the backbone site allows for major and minor grooving.
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.

---

Differences between oxDNA2 and oxDNA

In the oxDNA2.0 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,
7. Electrostatic interactions via Debye-Huckel-like term (DH)

\[
V_{\text{oxDNA2}} = \sum \text{nearest neighbours} \left( V_{\text{backbone}}^* + V_{\text{stack}}^* + V_{\text{exc}}^* \right)
+ \sum \text{other pairs} \left( V_{\text{HB}}^* + V_{\text{cross stack}}^* + V_{\text{exc}}^* + V_{\text{coax stack}}^* + V_{\text{DH}}^* \right),
\]

(12)
Structure and interactions of the oxDNA2.0 model

Three strands forming a nicked duplex as represented by oxDNA2.0, with the central section of the complex illustrating key interactions from Potential Energy Equation highlighted.

Individual nucleotides have an orientation described by a vector normal to the plane of the base (labelled n), and a vector indicating the direction of the hydrogen bonding interface (labelled b).
Snapshots of poly (dT) molecules used in the equilibration time tests:

(A) and (B): poly (dT) with 20 nucleotides;

(C) and (D) poly (dT) with 100 nucleotides.

Non-representative initial states of poly (dT) molecules (left),

Representative configurations obtained post-equilibration (right).
Equilibration plots for the ploy (dT) molecules with various number of bases, obtained as averages over 20 independent simulations. For both MD and VMMC simulations: the potential $V_0$ is plotted as a function of simulation progress, in units of reduced time (3.03 ps) for MD and attempted steps per particle for VMMC.
Plot of correlation of inter-segment vectors vs distance (number of base pairs along the DNA):

- 500 dsDNA (blue curves)
- 100 ssDNA (red curves).

Log Plot of correlation of inter-segment vectors vs distance (number of base pairs along the DNA):
The response of ssDNA to tension, and the role of stacking therein.

(A) Force-extension plots for 100-nucleotide poly (dA) using three models:
- no stacking (black);
- average stacking strength (oxDNA1.0, blue) and
- sequence-dependent stacking (oxDNA1.5, red).

Stronger stacking leads to an increased force at larger extensions, and extremely strong stacking results in a plateau-like feature as stacking is disrupted.

(B) Stacking probability for
- average stacking strength (oxDNA1.0, blue) and
- sequence-dependent stacking (oxDNA1.5, red)
as a function of applied force. Adjacent nucleotides are defined as stacked if the stacking energy between the pair is more than -0.1 units.
Free-energy profile of an 8-base-pair duplex (3′-ACTGACGT-5′ and 3′-ACGTCAGT-5′) at 312K in a simulation volume of side length 15 units.
Energy per nucleotide vs simulation time step for (a) 4mer, (b) 5mer, (c) 6mer duplexes at 320 K.

Sudden transitions from low to high energy are indicative of rare-event melting.
Computational time for equilibration of poly (dT) molecules of various lengths.

<table>
<thead>
<tr>
<th>Strand length</th>
<th>Total runtime (MD) in seconds</th>
<th>Total runtime (VMMC) in seconds</th>
<th>Equilibration time as a fraction of total runtime (MD)</th>
<th>Equilibration time as a fraction of total runtime (VMMC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>409.67</td>
<td>14.804</td>
<td>0.0118</td>
<td>0.0244</td>
</tr>
<tr>
<td>100</td>
<td>1804.18</td>
<td>147.57</td>
<td>0.0128</td>
<td>0.0287</td>
</tr>
<tr>
<td>1,000</td>
<td>32493.0</td>
<td>4187.03</td>
<td>0.0134</td>
<td>0.0299</td>
</tr>
</tbody>
</table>

Simulations were performed using a single core Intel(R) Core(TM) i5-4300U CPU @ 1.90GHz
Performance of the oligomer benchmark as time per time step for various system sizes:

Shown are results of

- the oxDNA standalone code on a single CPU
- NVIDIA V100 GPU and
- the LAMMPS implementation of the oxDNA2 model at different CPU counts
Performance of the pointer benchmark as time per time step at various salt concentrations:

Shown are results of

- the oxDNA standalone code on a single CPU
- NVIDIA V100 GPU and
- the LAMMPS implementation of the oxDNA2 model at different CPU counts
Features of the oxDNA2.0 Software

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
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**
   - general information (timestep, energy, box side)
   - orientation, positions of each nucleotide

2. **Topology:** backbone-backbone bonds between nucleotides in the same strand

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

\[ t = T \]
\[ b = L_z L_y L_z \]
\[ E = E_{tot} U K \]
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
...
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**

```plaintext
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

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

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]
```
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
Input File

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

\[
\text{[time (steps * dt)] [potential energy] [kinetic energy] [total energy]}
\]

**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.
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.dat
Energy: energy_seq_dep.dat
Trajectory: trajectory_seq_dep.dat
Last Trajectory File: last_conf_seq_dep.dat
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