

The oxDNA Coarse-Grained Model of DNA

An Introduction to the Model and Software Framework

Oliver Henrich
Email: oliver.henrich@strath.ac.uk

Department of Physics, University of Strathclyde, Glasgow, UK

30th April 2024

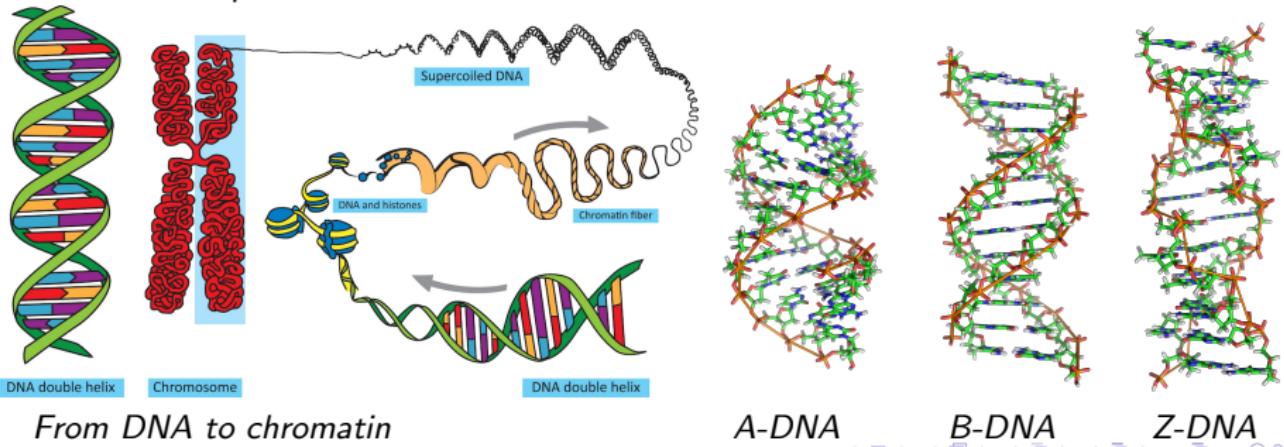
Outline

1. oxDNA Model
2. oxDNA Software
3. Practical Exercises

GitHub tutorial repo at https://github.com/ohenrich/oxDNA_tutorial

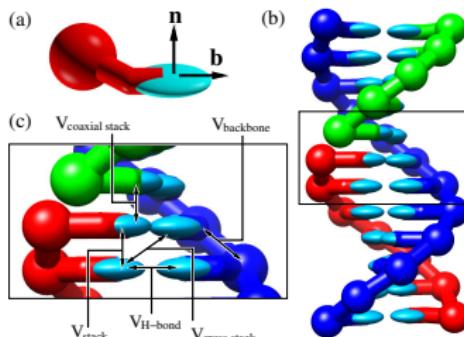
1. oxDNA Model DNA Facts

- Double-stranded DNA is $\approx 2\text{nm}$ wide, but 2m long in one cell (or $100\text{AU} = 0.58$ light days in all cells)
- Human genome contains 3×10^9 base pairs (bps)
- DNA loop around a nucleosome core particle contains 147 bps
- Smallest loop in chromatin fibre consists of 5×10^4 bps
- Atomistic simulation of DNA can model 3,000 bps (probably less) and typically resolve times on the μs -scale
- Coarse-grained models** target much **larger time and length scales** in the range of ms and Mbps



1. oxDNA Model Overview

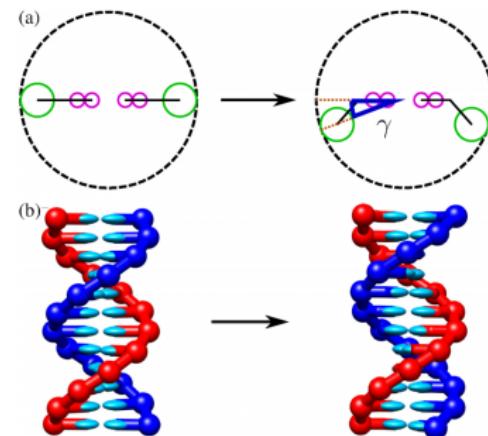
- Each nucleotide is described as **rigid body**
- 3 **interaction** sites for backbone, stacking and hydrogen-bonding
- 7 **effective interactions** between nucleotides
 - Bonded interaction for backbone connectivity
 - 6 pair interactions for excluded volume, stacking, cross-stacking, coaxial stacking, hydrogen-bonding and electrostatic interaction
- **oxDNA: 13 DOF per nucleotide**
3 positions, 3 translational momenta, 3 angular momenta, 1 unit quaternion (4 components)
- **Atomistic simulation** (pyrimidine base plus sugar-phosphate group):
around **200 DOF per nucleotide**
34 atoms per nucleotide, each with 3 positions and 3 momenta



(a) *oxDNA nucleotide*
 (b) *Duplex*
 (c) *Interactions*

1. oxDNA Model Nucleotide Geometry

- Each nucleotide has a centre of mass (COM) r_{COM} , a base vector \mathbf{b} , base normal \mathbf{n} and a third vector $\mathbf{y} = \mathbf{n} \times \mathbf{b}$
- The **backbone interaction** site is at
 $r_{back} = r_{COM} - 0.4 \mathbf{b}$ (oxDNA1)
 $r_{back} = r_{COM} - 0.34 \mathbf{b} + 0.3408 \mathbf{y}$ (oxDNA2)
- The **stacking interaction** site is at
 $r_{stack} = r_{COM} + 0.34 \mathbf{b}$
- The **hydrogen-bonding interaction** site is at
 $r_{base} = r_{COM} + 0.4 \mathbf{b}$



- (a) *oxDNA1 and oxDNA2 nucleotides: the base vector \mathbf{b} is horizontally oriented from left to right, whereas the base normal \mathbf{n} points away from the observer.*
- (b) *The angled backbone interaction sites leads to the correct geometry with major and minor grooves.*

1. oxDNA Model Angles and Vectors

Relative distance vectors are defined between two nucleotides i and j

- backbone interaction sites

$$\mathbf{r}_{back,ij} = \mathbf{r}_{back,i} - \mathbf{r}_{back,j}$$

- stacking interaction sites

$$\mathbf{r}_{stack,ij} = \mathbf{r}_{stack,i} - \mathbf{r}_{stack,j}$$

- hydrogen-bonding interaction sites

$$\mathbf{r}_{base,ij} = \mathbf{r}_{base,i} - \mathbf{r}_{base,j}$$

- mixed sites

$$\mathbf{r}_{back-base,ij} = \mathbf{r}_{back,i} - \mathbf{r}_{base,j}$$

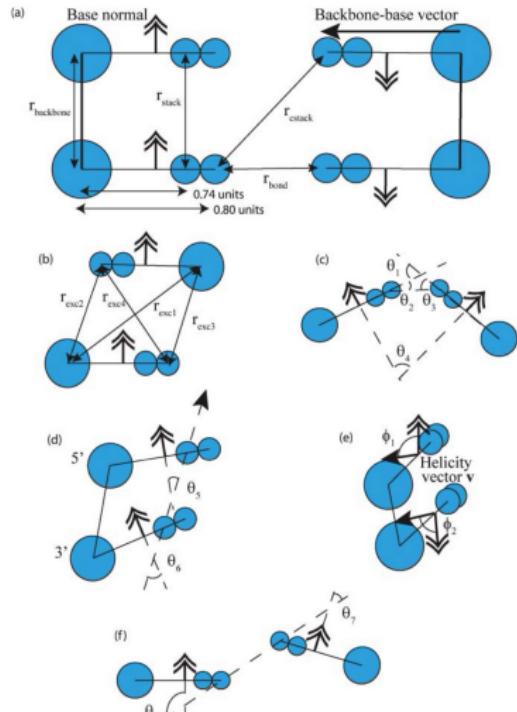
$$\mathbf{r}_{base-back,ij} = \mathbf{r}_{base,i} - \mathbf{r}_{back,j}$$

Relative angles are defined using the above vectors, the base vector \mathbf{b} and base normal \mathbf{n}

$$\cos(\theta_1) = -\hat{\mathbf{b}}_i \cdot \hat{\mathbf{b}}_j$$

$$\cos(\theta_2) = -\hat{\mathbf{b}}_i \cdot \hat{\mathbf{r}}_{base,ij}$$

$$\cos(\theta_3) = \hat{\mathbf{b}}_j \cdot \hat{\mathbf{r}}_{base,ij}$$

$$\vdots \quad \vdots \quad \vdots$$


oxDNA1 vectors and angles

1. oxDNA Model Potential Forms

Elementary potentials are used, which take distances or angles as arguments.

- FENE springs for backbone connectivity

$$V_{FENE}(r, \epsilon, r^0, \Delta) = -\frac{\epsilon}{2} \ln \left(1 - \frac{(r-r^0)^2}{\Delta^2} \right)$$

- Morse potential for stacking and hydrogen-bonding

$$V_{Morse}(r, \epsilon, r^0, a) = \epsilon (1 - \exp(-a(r - r^0)))^2$$

- Harmonic potential for cross-stacking and coaxial-stacking

$$V_{harm}(r, k, r^0) = \frac{k}{2}(r - r^0)^2$$

- Lennard-Jones potential for excluded volume

$$V_{LJ}(r, \epsilon, \sigma) = 4\epsilon \left(\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6 \right)$$

- Quadratic terms for angular modulations

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

- Quadratic smoothing terms for truncation

$$V_{smooth}(x, b, x^c) = b(x - x^c)^2$$

- Debye-Hückel potential for electrostatics

$$V_{DH}(r, \lambda) = \frac{q_{eff}}{4\pi\epsilon_0\epsilon_r} \exp(-r/\lambda)/r$$

1. oxDNA Model Modulation Factors

The above potentials are used directly or in angular and radial modulation factors $f_{1,\dots,6}$.

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

$$f_2(r) = \begin{cases} V_{harm}(r, k, r^0) - V_{harm}(r^c, k, r^0) & \text{if } r^{low} < r, r^{high}, \\ k V_{smooth}(r, b^{low}, r^{c,low}) & \text{if } r^{c,low} < r < r^{low}, \\ k V_{smooth}(r, b^{high}, r^{c,high}) & \text{if } r^{high} < r < r^{c,high}, \\ 0 & \text{otherwise} \end{cases}$$

$$f_3(r) = \begin{cases} V_{LJ}(r, \epsilon, \sigma) & \text{if } r < r^*, \\ \epsilon V_{smooth}(r, b, r^c) & \text{if } r^* < r < r^c, \\ 0 & \text{otherwise} \end{cases}$$

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

$$f_5(x) = \begin{cases} 1 & \text{if } x > 0, \\ V_{mod}(x, a, 0) & \text{if } x^* < x < 0, \\ V_{smooth}(x, b, x^c) & \text{if } x^c < x < x^*, \\ 0 & \text{otherwise} \end{cases}$$

$$f_6(\theta) = \begin{cases} V_{smooth}(\theta, b, \theta^c) & \text{if } \theta \geq \theta^c, \\ 0 & \text{otherwise} \end{cases}$$

1. oxDNA Model Interactions

The **oxDNA2 potential** consists of **1 bonded** and **6 pair interactions**.

- **Backbone connectivity** (bonded): $V_{backbone} = V_{FENE}(.)$

- **Excluded volume** (pair)

$$V_{excv} = f_3(r_{back-back}, \dots) + f_3(r_{back-base}, \dots) + f_3(r_{base-back}, \dots) + f_3(r_{base-base}, \dots)$$

- **Stacking** (pair): $V_{stack} = f_1(\cdot) \times f_4(\cdot) \times f_4(\cdot) \times f_4(\cdot) \times f_5(\cdot) \times f_5(\cdot)$

- **Hydrogen-bonding** (pair): $V_{HB} = f_1(\cdot) \times f_4(\cdot) \times f_4(\cdot) \times f_4(\cdot) \times f_4(\cdot) \times f_4(\cdot)$

- **Cross-stacking** (pair)

$$V_{x-stack} = f_2(\cdot) \times f_4(\cdot) \times f_4(\cdot) \times f_4(\cdot) \times \{f_4(\cdot) + f_4(\cdot)\} \times \{f_4(\cdot) + f_4(\cdot)\} \times \{f_4(\cdot) + f_4(\cdot)\}$$

- **Coaxial stacking** (pair)

$$V_{coaxial-stack} = f_2(\cdot) \times f_4(\cdot) \times \{f_4(\cdot) + f_6(\cdot)\} \times \{f_4(\cdot) + f_4(\cdot)\} \times \{f_4(\cdot) + f_4(\cdot)\}$$

- **Electrostatic** (pair): $V_{elec} = V_{DH}(\cdot)$

The complete potential contains sums over consecutive nucleotides on the same strand and all other pairs.

$$V = \sum_{nearest\ neighbours} (V_{backbone} + V'_{excv} + V_{stack})$$

$$+ \sum_{other\ pairs} (V_{excv} + V_{HB} + V_{x-stack} + V_{coaxial-stack} + V_{elec})$$

1. oxDNA Model Summary

- The oxDNA model uses a **top-down coarse-graining approach** with rather complex **bespoke interactions**.
- The oxDNA potential comprises **one bonded interaction** and **six pair interactions**. As strands denature, there is no residual memory of other conformations as is often the case with 3+body interactions.
- The **thermodynamic properties** of oxDNA are basically those of the **SantaLucia nearest-neighbour model**, thought to be an **exact empirical fit** experiments.
- **Uniquely among coarse-grained models** at this level of detail, oxDNA is able to describe the **thermodynamics of duplex formation** and provide an **accurate average representation** of the structure and mechanics of **both single-stranded and double stranded DNA and its assemblies**.
- oxDNA2 features **sequence-specific hydrogen-bonding and stacking interaction strengths**. But there is **no intrinsic sequence-specific curvature or elasticity** (\Rightarrow oxDNA3).

- [1] T. Ouldridge, A. Louis, and J. Doye, **Structural, Mechanical, and Thermodynamic Properties of a Coarse-Grained DNA Model**, *J. Chem. Phys.* **134**, 085101 (2011).
- [2] B. Snodin, et al., **Introducing Improved Structural Properties and Salt Dependence into a Coarse-Grained Model of DNA**, *J. Chem. Phys.* **142**, 234901 (2015).
- [3] A. Sengar, et al., **A primer on the oxDNA model of DNA: When to use it, how to simulate it and how to interpret the results**, *Front. Mol. Biosci.* **8**, 693710 (2021).

2. oxDNA Software Overview

We will use the following software

- LAMMPS version of oxDNA

The implementation of oxDNA1, oxDNA2 and oxAxRNA2 in the LAMMPS code via the CG-DNA package

- Standalone version of oxDNA

The original re-implementation of oxDNA1, which was extended to oxDNA2 and oxAxRNA2

- tacoxDNA

A suite of tools and converters

- oxView

A visualisation and data manipulation toolkit

2. oxDNA Software LAMMPS Version

- Large-scale **Atomic/Molecular Massively Parallel Simulator**
- Available from the LAMMPS website at <https://www.lammps.org>
Distributed under GPL v2 by Sandia National Laboratories
Latest stable release 2nd August 2023, updated 2nd March 2024 (initial release 1995)
- Popular
 - 405,000 downloads between September 2004 and June 2021
 - 1,600 forks on GitHub, ca. 100 direct contributors
- **Versatile**
 - Very **advanced molecular dynamics capabilities**
 - Code distributed over **ca. 100 standard and USER packages**
 - Supported on **CPU-, multicore- and GPU-architectures**, but not all packages offer all options
 - **REPLICA**: collection of multi-replica methods, e.g. parallel tempering
 - **PLUMED**: free energy library, enhanced sampling
 - **COLVARs**: collective variables library, advanced sampling methods like metadynamics, umbrella sampling, adaptive biasing force
- **Extendable**
 - Object-oriented C++ class structure
 - Top-level classes that are visible everywhere in the code
 - Virtual parent classes derived from top-level classes
 - Extensive use of polymorphism

2. oxDNA Software LAMMPS Version

Building the LAMMPS version with standard make

- Requires C/C++ compiler that supports the C++11 standard
- Change to /src in your LAMMPS directory
- Load ASSPHERE, MOLECULE and CG-DNA packages (minimal requirement)

```
make yes-asphere yes-molecule yes-cg-dna
```

- Check modules are loaded and clean

```
make ps
```

```
Installed YES: package ASSPHERE
Installed YES: package CG-DNA
Installed YES: package MOLECULE
```

- Compile the serial and/or parallel version using the default Makefiles in /src/MAKE

```
make [-j4] serial
make [-j4] mpi
```

- More Makefile configurations are in /src/MAKE/MACHINES

[4] O. Henrich, Y. A. Gutiérrez Fosado, T. Curk, and T. E. Ouldridge, Coarse-Grained Simulation of DNA Using LAMMPS: An Implementation of the OxDNA Model and Its Applications, *Eur. Phys. J. E* **41**, (2018).

[5] LAMMPS CG-DNA Documentation <https://docs.lammps.org/PDF/CG-DNA.pdf>

2. oxDNA Software Standalone Version

- **oxDNA** code includes oxDNA1, oxDNA2 and oxAxNA
- Available from <https://github.com/lorenzo-rovigatti/oxDNA>
Distributed under GPL v3
Latest stable release 3.6.1 (12th March 2024)
- 26 forks on GitHub, half a dozen contributors
- Very **advanced Monte Carlo capabilities** like **Virtual-Move Monte Carlo (VMMC)**
- Supported on **single-core CPU- and single GPU-architectures**
- Very extensive suite of **oxDNA Analysis Tools (OAT)** of bespoke postprocessing and analysis scripts

2. oxDNA Software Standalone Version

Building the serial standalone version for CPU-architectures with standard CMake

- Requires
 - C/C++ compiler that supports the C++14 standard
 - CMake version ≥ 3.5
 - optionally CUDA toolkit version ≥ 10
- Change to the oxDNA top-level directory

```
cd oxDNA
```

- Create a build directory and change to it

```
mkdir build  
cd build
```

- Create Makefiles, specify additional options

```
cmake ..
```

- Compile the serial version

```
make [-j4]
```

[6] oxDNA Documentation <https://lorenzo-rovigatti.github.io/oxDNA>

[7] oxDNA Website <https://dna.physics.ox.ac.uk>

2. oxDNA Software Standalone Version

Build with oxDNA Analysis Tools (OAT)

- Create Makefiles with specific options

```
cmake .. -DPython=ON -DOxpySystemInstall=On
```

- It might be necessary to specify explicit paths.

```
cmake .. -DPython=ON -DOxpySystemInstall=On  
-DPYTHON_INCLUDE_DIRS=/path/to/python/include/dir  
-DPYTHON_EXECUTABLE=/path/to/python/binary
```

- Compile and install

```
make [-j4]  
make install
```

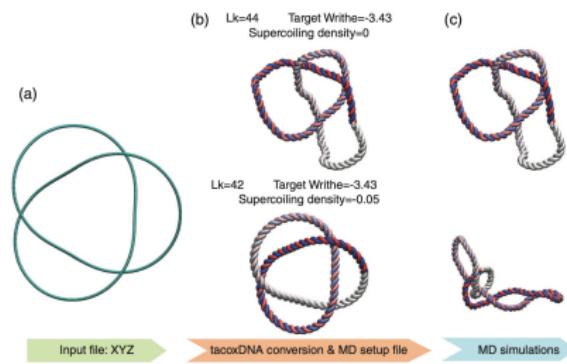
Note: A full installation of Anaconda3 is highly recommended!

More information regarding installation and known issues is available in the
[6] oxDNA documentation <https://lorenzo-rovigatti.github.io/oxDNA/install.html>

2. oxDNA Software tacoxDNA Tools and Converters

Available

- as webserver at <http://tacoxdna.sissa.it>
- as standalone Python code from <https://github.com/lorenzo-rovigatti/tacoxDNA>

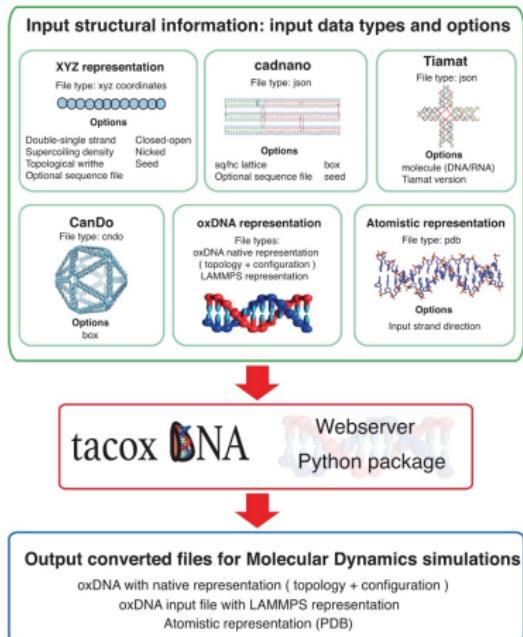


[8] A. Suma, et al., TacoxDNA: A User-Friendly Web Server for Simulations of Complex DNA Structures, from Single Strands to Origami, *J. Comput. Chem.* **40**, 2586 (2019).

2. oxDNA Software tacoxDNA Tools and Converters

A variety of format conversions is supported.
The native oxDNA format can also be used as intermediate.

- LAMMPS format \Leftrightarrow native oxDNA format
- PDB format \Leftrightarrow native oxDNA format
- XYZ format \Rightarrow native oxDNA format
- cadnano \Rightarrow native oxDNA format
- CanDo \Rightarrow native oxDNA format
- Tiamat \Rightarrow native oxDNA format



[8] A. Suma, et al., TacoxDNA: A User-Friendly Web Server for Simulations of Complex DNA Structures, from Single Strands to Origami, *J. Comput. Chem.* **40**, 2586 (2019).

2. oxDNA Software oxView Visualisation and Manipulation Toolkit

oxView is a webbrowser-based visualiser

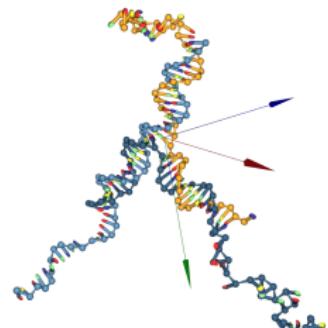
- Can load structures with over 1 million nucleotides
- Create videos from simulation trajectories
- Allow users to perform basic edits to DNA and RNA designs

Available from <https://github.com/sulcgroup/oxdna-viewer>

- Navigate down to the README.md and click Try it!
or
- Load <https://sulcgroup.github.io/oxdna-viewer>
or
- Run locally on localhost:8000 by starting a python server in your oxView directory

```
python -m http.server 8000
```

and open <http://localhost:8000> in your browser



[9] E. Poppleton, et al., Design, Optimization and Analysis of Large DNA and RNA Nanostructures through Interactive Visualization, Editing and Molecular Simulation, *Nucleic Acids Res.* **48**, e72 (2020).

[10] J. Bohlin, et al., Design and Simulation of DNA, RNA and Hybrid Protein–Nucleic Acid Nanostructures with OxView, *Nat. Protoc.* **17**, 1762 (2022).

3. Practical Exercises (0) Compiling the Source Codes and Download

- Clone the LAMMPS code from GitHub and compile (C/C++11 compiler, MPI)

```
git clone https://github.com/lammps/lammps.git
cd /lammps/src
make yes-asphere yes-molecule yes-cg-dna
make ps
make [-j4] mpi
```

- Clone the oxDNA standalone code from GitHub and compile (C/C++14 compiler, CMake, Python ≥ 3.9 , but Anaconda3 recommended)

```
git clone https://github.com/lorenzo-rovigatti/oxDNA.git
cd /oxDNA
mkdir build
cd build
cmake .. -DPython=ON [-DOxpySystemInstall=On]
make [-j4]
make install
```

- Clone tacoxDNA (Python3) and oxView (browser, Chrome recommended)

```
git clone https://github.com/lorenzo-rovigatti/tacoxDNA.git
git clone https://github.com/sulcgroup/oxdna-viewer.git
```

This is only necessary if you want to use these tools in a standalone fashion, e.g. because you cannot connect to the internet.

3. Practical Exercises (1) Creating an Initial Configuration

- Clone the oxDNA tutorial directory from GitHub.

```
git clone https://github.com/ohenrich/oxDNA_tutorial.git
```

- Navigate to the first exercise.

```
cd exercises/1_initial_config
```

- Inspect the file sequence.txt.

```
vi sequence.txt
```

```
DOUBLE AAAAAACGCGAAA...
```

The keyword DOUBLE indicates that we will create double-stranded DNA. Omitting the keyword produces a single-stranded configuration.

- Check syntax of configuration generator.

```
python generate-sa.py
```

```
Usage: generate-sa.py <box size> <file with sequences>
```

- Generate an initial configuration in a box size 100.

```
python generate-sa.py 100 sequence.txt
```

```
Found duplex of 63 bases
```

```
nstrands, nnuc1 = 2 126
```

```
Adding duplex of 63 bases
```

```
done line 1 / 1, now at 126/126
```

```
ALL DONE. just generated 'generated.dat' and 'generated.top'
```

3. Practical Exercises (1) Creating an Initial Configuration

- Inspect the oxDNA standalone format: topology file

```
vi generated.top
```

```
126 2          # total no. of nucleotides and strands
1 A -1 1      # strand ID, nucleotide type, 3' partner, 5' partner
1 A 0 2
1 A 1 3
1 A 2 4
...
```

- Inspect the oxDNA standalone format: configuration file

```
vi generated.dat
```

```
t = 0          # timestep
b = 100.0 100.0 100.0    # box dimensions
E = 0. 0. 0.      # energy
81.10076700045578 45.544279111797685 26.695973276367443
 0.6836324286755338 0.6187606869041726 -0.3870166854349663
 0.4842717557006389 0.012139188631672909 0.8748334165599674 0.0
 0.0 0.0 0.0 0.0 0.0
81.55953447082155 45.3431814998625 26.89033700942101
 0.233605215000666 0.9618090481628894 -0.1426602901878769
 0.4842717557006389 0.012139188631672909 0.8748334165599674 0.0
 0.0 0.0 0.0 0.0 0.0
...
# position, base vector, base normal, velocity, angular momentum
```

- Rename the files

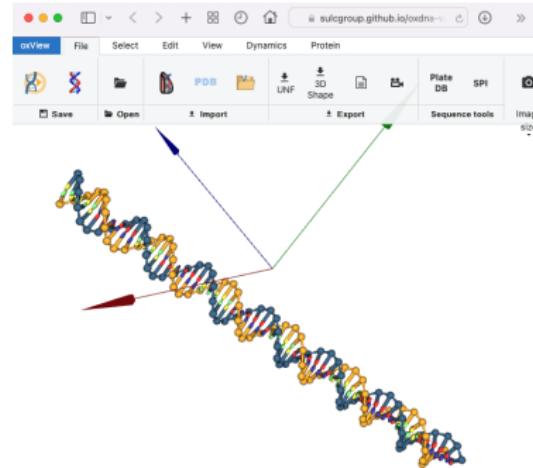
```
mv generated.top dsDNA_init.top
mv generated.dat dsDNA_init.dat
```

3. Practical Exercises (1) Creating an Initial Configuration

- Download the topology file `dsDNA_init.top` and configuration file `dsDNA_init.dat` into a working directory on your local host.

```
scp username@hostname:/path/to/oxDNA_tutorial/exercises/ \
1_initial_config/dsDNA_init* /your/working/directory/
```

- Open oxView in your browser,
<https://sulcgroup.github.io/oxdna-viewer>
 click on the 'Open' tab,
 navigate to your local working directory
 where you saved the two files and load
 them.
- Inspect and edit the visualisation, e.g. by
 rotating and translating the dsDNA
 molecule.



3. Practical Exercises (2) Equilibration Using The Standalone Code

In this exercise we want to equilibrate the initial configuration using the Virtual-Move Monte Carlo (VMMC) algorithm provided in the standalone code.

- Copy the topology and configuration file as well as the oxDNA executable into the directory for the second exercise and navigate to this directory.

```
cp /exercises/1_initial_config/dsDNA_init.* \
/exercises/2_MC_equilibration
cp /oxDNA/build/bin/oxDNA /exercises/2_MC_equilibration
```

- Perform a VMMC run with the oxDNA standalone code on a single CPU using the input input file.

```
./oxDNA input
```

MC step	Potential energy per nucleotide	Acceptance ratio for translational moves	Acceptance ratio for rotational moves	Acceptance ratio for volume moves
0	-1.679223	0.000	0.000	0.000
100	-1.608630	0.751	0.197	0.000
200	-1.608018	0.746	0.199	0.000
300	-1.587952	0.743	0.200	0.000
...

The columns correspond to MC step, potential energy per nucleotide, acceptance ratio for translational moves, acceptance ratio for rotational moves, acceptance ratio for volume moves.

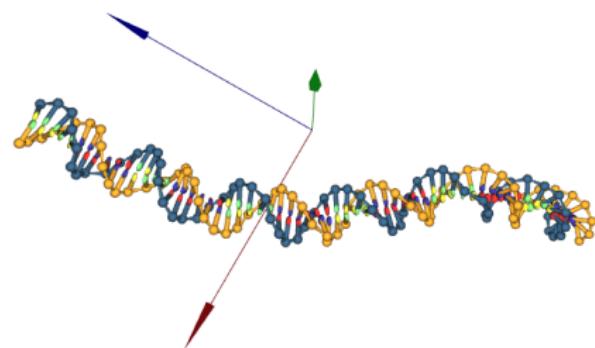
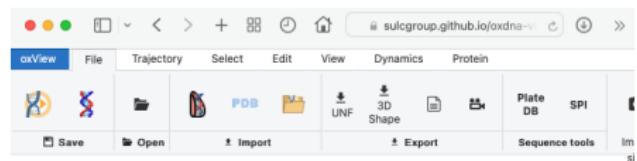
- The sequence of configurations of the VMMC run is contained in dsDNA-equ.dat. The last configuration is contained in last-config.dat.

3. Practical Exercises (2) Equilibration Using The Standalone Code

- Download the configuration file dsDNA_equ.dat into a working directory on your local host.

```
scp username@hostname:/path/to/oxDNA_tutorial/exercises/ \
2_MC_equilibration/dsDNA_equ.dat /your/working/directory
```

- In your oxView tab in your browser, click on the 'Open' tab, navigate to your working directory, load the new configuration file and the previous topology file (the topology has not changed).
- Inspect the output of the VMMC equilibrated dsDNA molecule.



3. Practical Exercises (3) Temperature Quench Using LAMMPS

The dsDNA molecule has been equilibrated at $T = 20^\circ\text{C}$. We want to perform a quench to $T = 100^\circ\text{C}$ with LAMMPS and observe how the dsDNA molecule denatures into two ssDNA molecules.

In order to do this, we need to convert the last equilibrated configuration `last_config.dat` from oxDNA standalone to LAMMPS format.

- In your oxView visualisation from exercise (II), check if the dsDNA molecule is centred around the origin. Otherwise navigate to the 'View' tab, select 'Centering → Origin' and click 'Apply'.
- Navigate to the 'File' tab and select 'Downloading the current scene as oxDNA files' from the 'Save' tab.
- Export the current topology and configuration file as `output.top` and `output.dat` to your Download directory.
- Navigate to the taxoxDNA webserver at <http://tacoxdna.sissa.it> and select the 'oxDNA → LAMMPS' tool.
- Select `output.top` and `output.dat` as oxDNA topology and configuration file and click 'Generate'.
- Click 'Download output'. A new tab opens in your browser. Save the page as `txt` file.

3. Practical Exercises (3) Temperature Quench Using LAMMPS

- Rename the LAMMPS data file to `data.dsDNA_equ.lmp` and copy it into the directory for the third exercise on your remote host.

```
scp ./data.dsDNA_equ.lmp username@hostname:/exercises/3_MD_quench
```

- On the remote host open `data.dsDNA_equ.lmp` in a text editor.

Set the number of bond types to **2 bond types** and the box boundaries to
-50.0 50.0 xlo xhi -50.0 50.0 ylo yhi -50.0 50.0 zlo zhi.

The top of your LAMMPS data file should look like this:

```
# LAMMPS data file
126 atoms
126 ellipsoids
124 bonds
4 atom types
2 bond types

# System size
-50.000000 50.000000 xlo xhi
-50.000000 50.000000 ylo yhi
-50.000000 50.000000 zlo zhi
...
...
...
```

The first edit is necessary so we can monitor the potential energy in the FENE backbone interaction. The second edit resets the boundaries with the dsDNA molecule centred in the simulation box.

3. Practical Exercises (3) Temperature Quench Using LAMMPS

- Perform an MD run with LAMMPS on 8 CPUs using the `in.dsDNA.lmp` input file.

```
mpirun -np 8 ./lmp_mpi -in in.dsDNA.lmp
LAMMPS (2 Aug 2023 - Update 3)
WARNING: Atom style hybrid defines both, per-type and per-atom
         masses; both must be set, but only per-atom masses will be
         used (../atom_vec_hybrid.cpp:132)
Reading data file ...
orthogonal box = (-50 -50 -50) to (50 50 50)
...
```

- Note: On this occasion do not worry about the warning

```
WARNING: Proc sub-domain size < neighbor skin, could lead to lost
         atoms
```

To optimise runtimes we run here on 8 CPU cores, which leads on average to only around 16 nucleotides per process. At this system size (126 nucleotides), we would normally run only on 2 or 4 MPI-tasks.

3. Practical Exercises (3) Temperature Quench Using LAMMPS

Visualising the trajectory with oxView requires transformation from the LAMMPS into the oxDNA format with tacoxDNA.

As the webserver supports currently only the transformation of data files, tacoxDNA has to be used on the command line.

- On the remote host clone the tacoxDNA repository from the GitHub website.

```
git clone https://github.com/lorenzo-rovigatti/tacoxDNA
```

- Navigate to your working directory with the LAMMPS trajectory.

```
cd /exercises/3_MD_quench
```

- Transform the LAMMPS trajectory into oxDNA format using the command line.

```
python /path/to/tacoxDNA/src/LAMMPS_oxDNA.py data.dsDNA.1.lmp \
          out.lammpstrj
```

```
Wrote data to 'data.dsDNA.1.lmp.oxdna' / 'data.dsDNA.1.lmp.top'
DONE
```

- Download the topology and configuration file to your local host and visualise the trajectory with oxView.

3. Practical Exercises (4) oxDNA Analysis Tools

Calculate the mean structure of a trajectory file

`oat mean -h`

```
usage: mean.py [-h] [-p num_cpus] [-o output_file] [-d deviation_file]
               [-i index_file] [-a alignment_configuration]
               trajectory
```

Computes the mean structure of a trajectory **file**

positional arguments:

trajectory The trajectory **file** you wish to analyze

optional arguments:

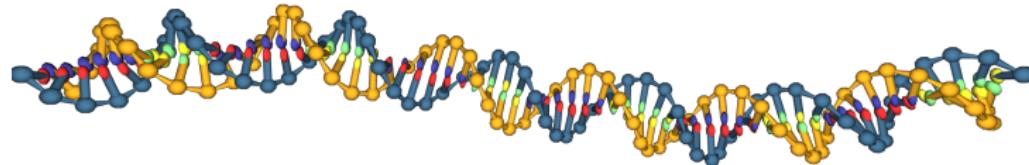
- h, --help show this **help** message **and exit**
- p num_cpus (optional) How many cores to use
- o output_file, --output output_file
 The filename to save the mean structure to
- d deviation_file, --deviations deviation_file
 Immediatley run oat deviations **from** the output
- i index_file Compute mean structure of a subset of particles **from**
 a space-separated **list in** the provided **file**
- a alignment_configuration, --align alignment_configuration

3. Practical Exercises (4) oxDNA Analysis Tools

- Calculate the mean structure of VMMC equilibration in /exercises/2_MC_equilibration

```
oat mean -p 4 -o dsDNA_mean.dat dsDNA_equ.dat
```

- Visualise with oxView



Requires dsDNA configuration as input for meaningful orientational averaging

3. Practical Exercises (4) oxDNA Analysis Tools

Calculate the ensemble of distances between nucleotides

`oat distance -h`

```
usage: distance.py [-h] [-i input [input ...]] [-o output_file] [-f
<histogram/trajectory/both>] [-p num_cpus]
```

Finds the ensemble of distances between **any** two particles **in the system**

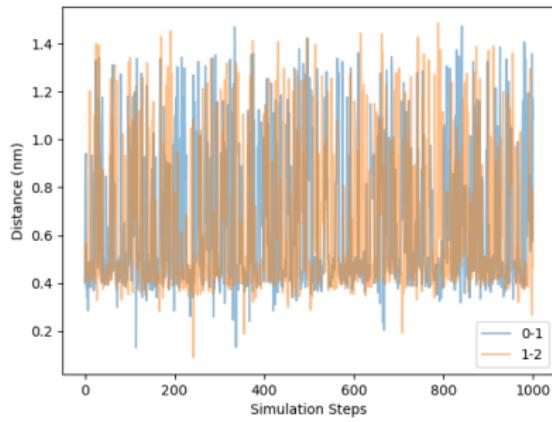
optional arguments:

- h, --help show this help message and exit
- i input [input ...], --input input [input ...]
A trajectory, and a list of particle pairs to compare. Can call -i multiple times to plot multiple datasets.
- o output_file, --output output_file
The name to save the graph file to
- f <histogram/trajectory/both>, --format <histogram/trajectory/both>
Output format for the graphs. Defaults to histogram.
Options are "histogram", "trajectory", and "both"
- p num_cpus, --parallel num_cpus
(optional) How many cores to use

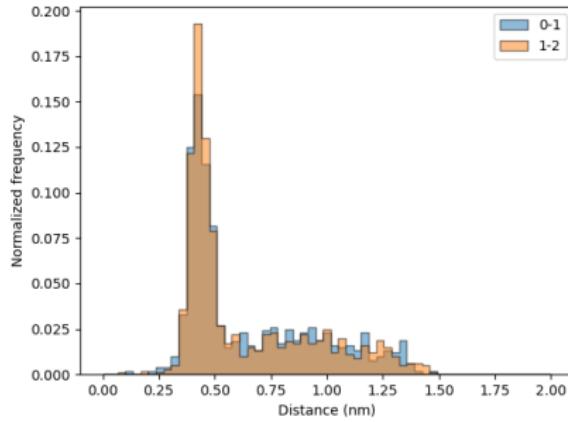
3. Practical Exercises (4) oxDNA Analysis Tools

- Calculate the ensemble of distances between the first and second and the second and third nucleotide

```
oat distance -p 4 -f both -i data.dsDNA.1.lmp.oxdna 0 1 1 2 -o
distance_dsDNA.1.png
```



Trajectory of distances



Histogram of distances