Hybrid Model Overview

Jonah Procyk

May 2020

Intro

This document is intended to introduce you to the new models introduced in the anm-oxdna repository https://github.com/sulcgroup/anm-oxdna. There are 6 new types of interactions and these are:

1. DNANM: Simulates DNA and Proteins in system, Uses oxDNA2 model for DNA and classic ANM for proteins
    - Available Options: Interaction Types: (MC, MD), GPU Version: Yes
2. RNANM: Simulates RNA and Proteins in system, Uses oxRNA2 model for RNA and classic ANM for proteins
    - Available Options: Interaction Types: (MC, MD), GPU Version: Yes
3. DNACT: Simulates DNA and Proteins in system, Uses oxDNA2 model for DNA and ANMT for proteins
    - Available Options: Interaction Types: (MC, MD), GPU Version: Yes
4. RNACT: Simulates DNA and Proteins in system, Uses oxDNA2 model for RNA and ANMT for proteins
    - Available Options: Interaction Types: (MC, MD), GPU Version: Yes
5. AC: Simulates Proteins alone, Uses classic ANM for proteins
    - Available Options: Interaction Types: (MC, MD), GPU Version: No
6. ACT: Simulates Proteins alone, Uses ANMT for proteins
    - Available Options: Interaction Types: (MC, MD), GPU Version: No

All hybrid Models (DNANM, RNANM, DNACT, or RNACT) are capable of simulating systems consisting of just proteins, just DNA/RNA, or proteins and DNA/RNA.

Prepping a Simulation

Four files are required for any of the new Interactions: the Input file, Parameter file, Topology file, and the Configuration (or dat) file. The one unique file for every simulation involving the protein model is the Parameter File. If no protein is in the simulation, simply set 'parfile=none' in the input file.

The Recommended Method to preparing the simulation files is to use our scripts available in the /ANMUtils directory in https://github.com/sulcgroup/anm-oxdna. To see examples of our scripts in preparing simulation files make sure to check out the /examples directory in /ANMUtils. An overview of the scripts is in the next section of this document.

Each of the following subsections will cover each file type and their specific options.

Topology File

The general format of the topology file is preserved with this model with a few caveats. The most important consideration is that the protein strands must all be declared together. They must be either before or after the declaration of all DNA or RNA particles. **To declare a Protein strand the strand id must be negative and starts from -1 incrementing -1 per strand.** The base and neighbor ids are untouched, with the one letter code for each Amino Acid supported in the base field. Protein neighbors aren't limited to their N-terminus and C-terminus neighbors. To account for this, **the index of each amino acid bonded to the particle being declared must be listed after the N-terminus and C-terminus neighbors**. Only particles with a higher index than the one being declared are included, much like the parameter file. One major change is the **header** line of the file. Usually its just two numbers with the number of strands and total particles. In any hybrid model, there are five numbers needed. **The first two are the familiar number of strands and total particles. The next three are: the number of dna/rna particles, the number of protein particles, and the number of dna/rna strands respectively.** If either dna or protein is absent from the simulation, it will run regardless as long as there is a 0 the corresponding fields.
ex. 1 14 base strand of DNA header
1 14 14 0 1
ex. 1 104 residue protein header
1 104 0 104 0
Below is a hypothetical 2 DNA 3 Protein hybrid Topology file:
2 5 2 3 1
-1 V -1 1 2
-1 A 0 2
-1 T 1 -1
1 A -1 1
1 G 0 -1

Input File

There are three different options for running any hybrid simulation (CUDA, MD CPU, MC CPU).
All three require the interaction type be set to their respective Interaction name ex. "DNANM". The CPU backend requires no extra options except for the one required key "parfile =" which must be set to the Parameter File or 'none'.

For the CUDA hybrid Interactions, the interaction type should be set to the Interaction name again ex. "DNACT" AND the backend option should be set to CUDA. Additional parameters in the input file required to use CUDA. In additon not all CUDA options are currently supported. Currently you must use a Verlet List as well as an edge based approach. The required options are listed below:

backend = CUDA
CUDA_Sort_Every = 0
CUDA_list = verlet
use_edge = 1
edge_n_forces = 1

An additional option you have with the CUDA Backend is a backend precision of mixed. This is faster than backend precision double while being almost as precise. To enable this use the following:

"backend_precision = mixed"

Parameter File

The Parameter file has the following format. The first line is the number of protein particles in the system. All lines under that follow this format:

particle_i_index particle_j_index potential_type equilibrium_distance spring_constant"

The particle indexes are the same as those in the topology file. When listing a bond, index i is always less than index j. Keeping this notation ensures each bond is accounted for only once.

The only potential type currently supported is the spring potential. The character **s** needs to be in this field.

The equilibrium distance is the distance between the α-Carbons at index i and index j. This is calculated from the PDB coordinates. Must be in float format and non-negative.

The spring constant will be the same for every bond in the system in the classic ANM. Must be in the form of a float. Using the HANM or mANM will have unique spring constants. The spring constants calculation is provided in our scripts.

The ANMT model has 4 extra angle parameters at the end of each declared bond between nearest neighbors (backbone neighbors only).

Configuration File

Protein Particles have all the same fields as their dna counterparts. The major difference is that orientation doesn't matter to the protein particles in the classic ANM. To account for this, the orientation vectors are set to any orthogonal pair of vectors. Also the Angular Momentum does nothing to our particles, so those will not be updated during the simulation. For convenience, all angular momentum components can be set to 0.

For the ANMT model this isn't true. The orientation vectors must be specified and Angular Momentum makes a difference.

Script Overview

The scripts contains a library of python classes and functions to help convert from PDB files into our Model. It also includes a Modeller implementation for the modelling of missing residues in PDB structures.

The scripts contain 5 classes for building specific structures:

1. ANM - The classic ANM model, the other classes derive from here
2. ANMT - ANM with bending/torisional modulation
3. HANM - Fits B-factors to experimental observed
4. mANM - Multiscale ANM
5. peptide - ANMT implmentation of a Peptide

Only ANM and ANMT are shown in the Examples, as well as a Modeller Reconstruction of RNA Polymerase.
The Scripts will be updated in future commits to this repository.

Need Help?

Feel free to open an issue or feature request!