Coarse-grained OpenMM Documentation

Release 0.0.1

Shirts research group

Garrett A. Meek
Lenny T. Fobe
Michael R. Shirts

Dept. of Chemical and Biological Engineering
University of Colorado Boulder

May 12, 2019

CONTENTS

This documentation is generated automatically using Sphinx, which reads all docstring-formatted comments from Python functions in the 'cg_openmm' repository. (See cg_openmm/doc for Sphinx source files.)

CHAPTER

ONE

OPENMM SIMULATION() PROTOCOLS FOR COARSE GRAINED MODELING

This page details the functions in cg_openmm/src/cg_mm_tools/cg_openmm.py.

cg_mm_tools.cg_openmm.**build_mm_force**(*sigma*, *epsilon*, *charge*, *num_beads*, *cutoff=Quantity(value=1, unit=nanometer)*)

Build an OpenMM 'Force' for the non-bonded interactions in our model.

sigma: Non-bonded bead Lennard-Jones interaction distances, (float * simtk.unit.distance)

epsilon: Non-bonded bead Lennard-Jones interaction strength, (float * simtk.unit.energy)

charge: Charge for all beads (float * simtk.unit.charge)

cutoff: Cutoff distance for nonbonded interactions (float * simtk.unit.distance)

num_beads: Total number of beads in our coarse grained model (integer)

cg_mm_tools.cg_openmm.**build_mm_simulation**(*topology*, *system*, *positions*, *temperature=Quantity(value=300.0, unit=kelvin)*, *simulation_time_step=Quantity(value=0.002, unit=picosecond)*, *total_simulation_time=Quantity(value=1.0, unit=picosecond)*, *output_pdb='output.pdb'*, *output_data='output.dat'*, *print_frequency=100*)

Construct an OpenMM simulation object for our coarse grained model.

topology: OpenMM topology object

system: OpenMM system object

positions: Array containing the positions of all beads in the coarse grained model (np.array('num_beads' x 3 , (float * simtk.unit.distance))

temperature: Simulation temperature (float * simtk.unit.temperature)

simulation_time_step: Simulation integration time step (float * simtk.unit.time)

total_simulation_time: Total simulation time (float * simtk.unit.time)

output_data: Name of output file where we will write the data from this simulation (string)

print_frequency: Number of simulation steps to skip when writing data to 'output_data' (integer)

cg_mm_tools.cg_openmm.**build_mm_system**(*box_size*, *mass*, *num_beads*, *sigma*, *epsilon*, *charge*)

Construct an OpenMM system for our coarse grained model

box_size: Simulation box length (float * simtk.unit.length)

mass: Coarse grained particle mass (float * simtk.unit.length)

num_beads: Total number of beads in our coarse grained model (int)

sigma: Non-bonded bead Lennard-Jones interaction distances, (float * simtk.unit.distance)

epsilon: Non-bonded bead Lennard-Jones interaction strength, (float * simtk.unit.energy)

charge: Charge for all beads (float * simtk.unit.charge)

cg_mm_tools.cg_openmm.**build_mm_topology**(*polymer_length*, *backbone_length*, *sidechain_length*)

Construct an OpenMM topology for our coarse grained model

polymer_length: Number of monomers in our coarse grained model (integer)

backbone_length: Number of backbone beads on individual monomers in our coarse grained model, (integer)

sidechain_length: Number of sidechain beads on individual monomers in our coarse grained model, (integer)

cg_mm_tools.cg_openmm.**distance**(*positions_1*, *positions_2*)

Construct a matrix of the distances between all particles.

positions_1: Positions for a particle (np.array(length = 3))

positions_2: Positions for a particle (np.array(length = 3))

distance (float * unit)

cg_mm_tools.cg_openmm.**get_box_vectors**(*box_size*)

Assign all side lengths for simulation box.

box_size: Simulation box length (float * simtk.unit.length)

cg_mm_tools.cg_openmm.**set_box_vectors**(*system*, *box_size*)

Build a simulation box.

system: OpenMM system object

box_size: Simulation box length (float * simtk.unit.length)

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

C

INDEX