foldamers Documentation

Release 0.0.1

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

Apr 10, 2019

CONTENTS

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

CHAPTER

ONE

OPENMM FUNCTIONS

This page details the functions and classes in src/openmm.py

1.1 ‘cgmodel’ class for OpenMM simulation

Shown below is a detailed description of the ‘cgmodel’ class object, which contains all simulation objects required by OpenMM.

class openmm.**cgmodel**(*box_size=Quantity(value=10.0, unit=nanometer), polymer_length=12, backbone_length=1, sidechain_length=1, sidechain_positions=[0], mass=Quantity(value=12.0, unit=dalton), sigma=Quantity(value=8.4, unit=angstrom), epsilon=Quantity(value=0.5, unit=kilocalorie/mole), bond_length=Quantity(value=1.0, unit=angstrom), bb_bond_length=Quantity(value=1.0, unit=angstrom), bs_bond_length=Quantity(value=1.0, unit=angstrom), ss_bond_length=Quantity(value=1.0, unit=angstrom), charge=Quantity(value=0.0, unit=elementary charge)*)

Construct all of the objects that OpenMM expects/requires for simulations with a coarse grained model.

box_size: Simulation box length, default = 10.00 * unit.nanometer

polymer_length: Number of monomer units (integer), default = 8

backbone_length: Number of beads in the backbone portion of each (individual) monomer (integer), default = 1

sidechain_length: Number of beads in the sidechain portion of each (individual) monomer (integer), default = 1

sidechain_positions: List of integers defining the backbone bead indices upon which we will place the sidechains, default = [0] (Place a sidechain on the backbone bead with index “0” (first backbone bead) in each (individual) monomer

mass: Mass of coarse grained beads (float * simtk.unit.mass) default = 12.0 * unit.amu

sigma: Non-bonded bead Lennard-Jones interaction distances, (float * simtk.unit.distance) default = 8.4 * unit.angstrom

epsilon: Non-bonded bead Lennard-Jones interaction strength, (float * simtk.unit.energy) default = 0.5 * unit.kilocalorie_per_mole

bond_length: Bond length for all beads that are bonded, (float * simtk.unit.distance) default = 1.0 * unit.angstrom

bb_bond_length: Bond length for all bonded backbone beads, (float * simtk.unit.distance) default = 1.0 * unit.angstrom

bs_bond_length: Bond length for all backbone-sidechain bonds, (float * simtk.unit.distance) default = 1.0 * unit.angstrom ss_bond_length: Bond length for all beads within a sidechain, (float * simtk.unit.distance) default = 1.0 * unit.angstrom

charge: Charge for all beads (float * simtk.unit.charge) default = 0.0 * unit.elementary_charge

box_size polymer_length backbone_length sidechain_length sidechain_positions mass sigma epsilon bond_length bb_bond_length bs_bond_length ss_bond_length charge num_beads topology system positions simulation

1.2 foldamers 'modules' for OpenMM simulation

`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)

`openmm.`**`build_mm_simulation`**(*topology*, *system*, *temperature*, *simulation_time_step*, *total_simulation_time*, *positions*, *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

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)

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

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)

`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) 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)

openmm.**get_box_vectors**(*box_size*)

Assign all side lengths for simulation box.

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

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

INDEX