foldamers Documentation

Release 0.0

Garrett A. Meek
Theodore L. Fobe
Connor M. Vogel

Research group of Professor Michael R. Shirts

Dept. of Chemical and Biological Engineering
University of Colorado Boulder

Sep 09, 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

COARSE GRAINED MODEL UTILITIES

The foldamers package uses "CGModel()" objects to define and store information about the properties of coarse grained models.

1.1 'basic_cgmodel': a simple function to build coarse grained homopolymers

Shown below is the 'basic_cgmodel' function, which requires a minimal set of input arguments to build a coarse grained holopolymer model.

1.2 Using the 'CGModel()' class to build coarse grained heteropolymers

Shown below is a detailed description of the 'CGModel()' class object, as well as some of examples demonstrating how to use its functions and attributes.

CHAPTER

TWO

ENSEMBLE BUILDING TOOLS

The foldamers package contains several tools for building conformational ensembles. The MDTraj and MSMBuilder packages are leveraged to perform structural analyses in order to identify poses that are structurally similar.

2.1 Using MSMBuilder to generate conformational ensembles

The foldamers package allows the user to apply K-means clustering tools from MSMBuilder in order to search for ensembles of poses that are structurally similar. The centroid configurations for individual clusters are used as a reference, and ensembles are defined by including all structures that fall below an RMSD positions threshold (<2 Angstroms).

ensembles.cluster.**concatenate_trajectories**(*pdb_file_list*, *combined_pdb_file=None*)

ensembles.cluster.**align_structures**(*reference_traj*, *target_traj*)

ensembles.cluster.**get_cluster_centroid_positions**(*pdb_file*, *cgmodel*, *n_clusters=None*)

2.2 Native structure-based ensemble generation tools

The foldamers package allows the user to build "native" and "nonnative" structural ensembles, and to evaluate their energetic differences with the Z-score. These tools require identification of a "native" structure.

2.3 Energy-based ensemble generation tools

The foldamers package allows the user to build structural ensembles that exhibit similar energies. Shown below are tools that enable energy-based ensemble generation.

2.4 Writing and reading ensemble data from the 'foldamers' database

The foldamers package is designed to store the low-energy poses from simulation runs of new (previously un-modelled) coarse grained representations. At present, the package does not enable storage of heteropolymers, in order to minimize the size of the database. For homopolymers, the syntax for assigning directory names for coarse grained model data is as follows:

directory_name = str("foldamers/ensembles/" + str(polymer_length) + "_" + str(backbone_length) + "_" + str(sidechain_length) "_" + str(sidechain_positions) + "_" + str(bb_bb_bond_length) + "_" + str(sc_bb_bond_length) + "_" + str(sc_sc_bond_length))

For example, the directory name for a model with 20 monomers, all of which contain one backbone bead and one sidechain bead, and whose bond lengths are all 7.5 Angstroms, would be: "foldamers/ensembles/20_1_1_0_7.5_7.5_7.5".

The following functions are used to read and write ensemble data to the foldamers database (located in 'foldamers/ensembles').

CHAPTER

THREE

PARAMETER ANALYSIS TOOLS FOR COARSE GRAINED MODELING

The 'foldamers' package allows wide-ranging parameter analyses for a coarse grained model. In particular, the package contains tools to analyze quantities that reflect secondary structure, including: 1) the fraction of native contacts, 2) the orientational ordering parameter 'P2', and 3) using kHelios, helical order parameters such as the pitch.

3.1 How to

Shown below are functions/tools used in order to calculate the heat capacity with pymbar.

CHAPTER

FOUR

THERMODYNAMIC ANALYSIS TOOLS FOR COARSE GRAINED MODELING

This page details the functions and classes in src/thermo

4.1 Tools to calculate the heat capacity with pymbar

Shown below are functions/tools used in order to calculate the heat capacity with pymbar.

CHAPTER

FIVE

UTILITIES FOR THE 'FOLDAMERS' PACKAGE

This page details the functions and classes in src/util.

5.1 Input/Output options (src/utilities/iotools.py)

Shown below is a detailed description of the input/output options for the foldamers package.

utilities.iotools.**write_bonds**(*CGModel*, *pdb_object*)

Writes the bonds from an input CGModel class object to the file object 'pdb_object', using PDB 'CONECT' syntax.

CGModel: Coarse grained model class object

pdb_object: File object to which we will write the bond list

utilities.iotools.**write_cg_pdb**(*cgmodel*, *file_name*)

Writes the positions from an input CGModel class object to the file 'filename'. Used to test the compatibility of coarse grained model parameters with the OpenMM PDBFile() functions, which are needed to write coordinates to a PDB file during MD simulations.

CGModel: Coarse grained model class object

filename: Path to the file where we will write PDB coordinates.

utilities.iotools.**write_pdbfile_without_topology**(*CGModel*, *filename*, *energy=None*)

Writes the positions from an input CGModel class object to the file 'filename'.

CGModel: Coarse grained model class object

filename: Path to the file where we will write PDB coordinates.

energy: Energy to write to the PDB file, default = None

5.2 Utilities and random functions (src/utilities/util.py)

CHAPTER

SIX

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX