
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**

Jun 06, 2020

Contents

1	Installation notes	2
1.1	Dependencies for the foldamers package	2
1.2	Recommended installation steps	3
2	Coarse grained model utilities	4
2.1	Using the ‘CGModel()’ class to build coarse grained heteropolymers	4
2.2	‘basic_cgmodel’: a simple function to build coarse grained homopolymers	5
3	Thermodynamic analysis tools for coarse grained modeling	6
3.1	Using MBAR to compute expectation values for structural & thermodynamic properties	6
3.2	Evaluating thermodynamic properties with ‘pymbar’	7
3.3	Tools to evaluate thermodynamic properties with ‘pymbar’	8
3.4	Calculating the heat capacity with pymbar	9
4	Ensemble building tools	10
4.1	Using MSMBuilder to generate conformational ensembles	10
4.2	Native structure-based ensemble generation tools	11
4.3	Energy-based ensemble generation tools	12
4.4	Writing and reading ensemble data from the ‘foldamers’ database	13
5	Utilities for the ‘foldamers’ package	14
5.1	Random structure generation in ‘foldamers’	14
5.2	Reading and writing to output within the ‘foldamers’ package	16
6	Parameter analysis tools for coarse grained modeling	17

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 1

Installation notes

The `foldamers` package will eventually be available for installation via `Anaconda`. This will make resolution of software conflicts much easier. However, at present, because the package has not been made public, `Anaconda` installation is not yet possible, and software conflicts must be resolved by the individual.

Here we provide installation instructions which have been tested on multiple platforms.

1.1 Dependencies for the `foldamers` package

Due to conflicts among dependencies for the `foldamers` package with other Python versions, **we recommend using Python version 3.6.**

The following is a list of software dependencies for the `foldamers` package, with recommended version numbers in parentheses:

- 1) `cg_openmm` (version 0.0)
Dependencies for the ‘`cg_openmm`’ software package: 2) `OpenMM` (version 7.3.1) 3) `Yank` (version 0.24.1)
- 4) `pymbar` (version 3.0.3)
- 5) `MDTraj` (version 1.9.3)
- 6) `MSMBuilder` (version 3.8)
- 7) `scikit-learn` (version 0.18.1)

- 8) `sklearn-genetic` (version 0.2)
- 9) `kHelios`

1.2 Recommended installation steps

We recommend installation of [Anaconda](#) prior to installation of the ‘foldamers’ package, as this makes resolution of conflicts between dependencies much easier.

We direct users that have not installed [Anaconda](#) to the [Download page](#) in order to select the appropriate version for your platform (Windows, Linux, Mac). (It shouldn’t matter which version of Anaconda is installed.)

The following installation steps are recommended for users that have already installed [Anaconda](#) on their system:

- 1) Install ‘sklearn-genetic’ (used for genetic algorithm optimization of model parameters):

```
pip install sklearn-genetic
```

- 2) Create an Anaconda environment for Python version 3.6 (the most stable Python version for ‘foldamers’):

```
conda create -n foldamers python=3.6 mdtraj=1.9.3 openmm=7.3.1_  
↳pymbar=3.0.3 msmbuilder=3.8 scikit-learn=0.18.1 yank=0.21.2
```

- 3) Activate this new environment:

```
conda activate foldamers
```

- 4) Clone and install a fresh copy of ‘cg_openmm’:

```
git clone https://github.com/shirtsgroup/cg_openmm.git  
cd cg_openmm  
python setup.py install
```

- 5) Clone and install a fresh copy of ‘foldamers’ to your system:

```
git clone https://github.com/shirtsgroup/foldamers.git  
cd foldamers  
python setup.py install
```

Coarse grained model utilities

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

2.1 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.

2.2 '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.

Thermodynamic analysis tools for coarse grained modeling

The ‘foldamers’ package contains wide-ranging tools to analyze simulation data in order to estimate the thermodynamic properties of a system. In particular, the package leverages tools from [pymbar](#) to enable analysis of and estimate expectation values for sampled and unsampled thermodynamic states.

3.1 Using MBAR to compute expectation values for structural & thermodynamic properties

Shown below is the main ‘foldamers’ function used to re-weight simulation results with ‘pymbar’.

3.2 Evaluating thermodynamic properties with ‘pymbar’

Shown below ‘pymbar’-based tools/functions to evaluate thermodynamic properties within the ‘foldamers’ package.

3.3 Tools to evaluate thermodynamic properties with 'pymbar'

Shown below are other 'pymbar'-based tools/functions that aid evaluation of thermodynamic properties within the 'foldamers' package.

3.4 Calculating the heat capacity with pymbar

Shown below is the primary function used to evaluate the heat capacity with pymbar:

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

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.

4.1 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).

4.2 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.

4.3 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.

4.4 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’).

Utilities for the ‘foldamers’ package

The ‘foldamers’ package contains a number of utilities for reading and writing data, as well as random structure generation.

5.1 Random structure generation in ‘foldamers’

Random structures are often needed as a starting point for simulation of a new coarse grained model. Shown below are the main tools that allows the user to build a random structure:

Shown below are other tools that support the task of random structure generation:

5.2 Reading and writing to output within the ‘foldamers’ package

Shown below are ‘foldamers’ tools for writing data and structures to output files.

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.