Computational Biochemistry

Release 0.1

Folorunsho Bright Omage, Prof João Batista Teixeira da Rocha

Feb 08, 2023

CONTENTS

Departamento de Bioquímica e Biologia Molecular (UFSM) Programas de Pós-Graduação em Ciências Biológicas: Bioquímica Toxicológica - PPGBTox Programa de Pós-Graduação em Educação em Ciências: Química da Vida e Saúde - PPGECQVS Centro de Ciências Naturais e Exatas - CCNE 97105-900 Santa Maria RS Brasil

Computational Biochemistry is a field of biochemistry that uses computational technologies in *biochemical research*.

Check out the usage section for further information, including how to installation the project.

Note: This project is under active development.

CHAPTER

ONE

CONTENTS

1.1 Rigid Docking

1.1.1 Installation

To use Lumache, first install it using pip:

```
(.venv) $ pip install lumache
```

1.1.2 Example 1: Joao Vitor

To retrieve a list of random ingredients, you can use the `lumache.get_random_ingredients()` function:

The `kind` parameter should be either `"meat"`, `"fish"`, or `"veggies"`. Otherwise, `lumache.get_random_ingredients()` will raise an exception.

For example:

```
>>> import lumache
>>> lumache.get_random_ingredients()
['shells', 'gorgonzola', 'parsley']
```

1.2 Flexible Docking

1.2.1 Installation

To use Lumache, first install it using pip:

```
(.venv) $ pip install lumache
```

1.2.2 Creating recipes

To retrieve a list of random ingredients, you can use the `lumache.get_random_ingredients()` function:

The `kind` parameter should be either `"meat"`, `"fish"`, or `"veggies"`. Otherwise, `lumache.get_random_ingredients()` will raise an exception.

For example:

```
>>> import lumache
>>> lumache.get_random_ingredients()
['shells', 'gorgonzola', 'parsley']
```

1.3 Protein-protein Docking

1.3.1 Installation

To use Lumache, first install it using pip:

```
(.venv) $ pip install lumache
```

1.3.2 Example 1: Tamie Duarte

Tamie docking p00ppp;llllll To use Lumache, first install it using pip:

```
(.venv) $ pip install lumache
```

1.3.3 Creating recipes

To retrieve a list of random ingredients, you can use the `lumache.get_random_ingredients()` function:

The `kind` parameter should be either `"meat"`, `"fish"`, or `"veggies"`. Otherwise, `lumache.get_random_ingredients()` will raise an exception.

For example:

```
>>> import lumache
>>> lumache.get_random_ingredients()
['shells', 'gorgonzola', 'parsley']
```

1.4 Molecular Dynamics Simulation

1.4.1 Installation

To use Lumache, first install it using pip:

```
(.venv) $ pip install lumache
```

1.4.2 Protocol

In this method, the interactions between the atoms and molecules are modeled using force fields, and the time evolution of the system is computed using the Newton's equations of motion. The Amber 20 package provides a suite of tools for performing MD simulations, including the generation of input files, the simulation itself, and the analysis of simulation results. Here is a step-by-step guide to performing an MD simulation using Amber 20:

1.4.3 Preparation of input files:

1. Preparation of a molecular structure file in a format readable by Amber, such as PDB or AMBER format. As with the protein and ligand
2. Preparation of a topology file, which defines the molecular structure and the force field parameters to be used.

Tip: In this task, the `antechamber` and `parmchk2` tools from AmberTools will be utilized to generate parameters for the General AMBER force field 2 (GAFF2). This force field is specifically designed for pharmaceutical molecules and works well with traditional AMBER force fields for proteins. The AM1-BCC2 charge method will be used to assign partial charges.

1.4.4 Using Antechamber:

Antechamber is specifically designed for the generation of molecular force field parameters and the assignment of partial charges. The tool is widely used for the preparation of molecular systems for molecular dynamics simulations, and it generates force field parameters that can accurately describe the behavior of the molecule.

```
$ antechamber -i ligand.H.pdb -fi pdb -o ligand.mol2 -fo mol2 -c bcc -nc 0 -rn ligand -
↪at gaff2
```

Antechamber uses a combination of empirical and quantum mechanical methods to determine the force field parameters, which include bond lengths, bond angles, torsion angles, and partial charges. It can handle a wide range of molecular systems, including small molecules, peptides, and proteins. The force field parameters generated by antechamber are based on the General AMBER force field (GAFF), a widely used force field in molecular dynamics simulations.

```
-antechamber is the name of the tool being executed.
-i ligand.H.pdb specifies the input file, which is in PDB format and named ligand.H.pdb.
-fi pdb specifies the format of the input file, which is PDB.
-o ligand.mol2 specifies the output file, which will be in MOL2 format and named ligand.
↪mol2.
-fo mol2 specifies the format of the output file, which is MOL2.
-c bcc specifies the charge method to be used, which is the Bond Charge Corrected (BCC)␣
↪method.
-nc 0 specifies the net charge of the molecule, which is 0 in this case.
-rn ligand specifies the root name for the output file, which is ligand.
-at gaff2 specifies the atom type to be used, which is the General AMBER force field 2␣
↪(GAFF2).
```

1.4.5 Examining force field parameters with parmchk2

While the most likely combinations of bond, angle and dihedral parameters are defined in the parameter file it is possible that our molecule might contain combinations of atom types for bonds, angles or dihedrals that have not been parameterised. If this is the case, we will have to specify any missing parameters before we can create our prmtop and inpcrd files in LEap.

We will use parmchk2 to test if all the parameters we require are available.

```
$ parmchk2 -i GWS.mol2 -f mol2 -o GWS.frcmod -s gaff2
```

3. Preparation of a parameter file, which provides information about the simulation conditions, such as temperature, pressure, and simulation time.

1.4.6 Setting up the simulation:

Load the molecular structure, topology, and parameter files into Amber. Prepare the simulation box, if necessary, by solvating the molecule in water or other solvent. Minimize the energy of the system to remove any starting configuration artifacts.

1.4.7 Running the simulation:

Run the simulation using the appropriate command. This will compute the time evolution of the system for the specified time interval. During the simulation, energy and other thermodynamic quantities are computed and logged to a trajectory file. Analysis of simulation results:

1.4.8 Load the trajectory file into Amber.

Analyze the simulation results by computing various properties, such as the potential energy, temperature, and density. Visualize the simulation results using visualization tools, such as VMD or PyMOL.

The `kind` parameter should be either `"meat"`, `"fish"`, or `"veggies"`. Otherwise, `lumache.get_random_ingredients()` will raise an exception.

For example:

```
>>> import lumache
>>> lumache.get_random_ingredients()
['shells', 'gorgonzola', 'parsley']
```

1.4.9 Example 1

Author: Folorunsho Bright Omage

To retrieve a list of random ingredients, you can use the `lumache.get_random_ingredients()` function:

The `kind` parameter should be either `"meat"`, `"fish"`, or `"veggies"`. Otherwise, `lumache.get_random_ingredients()` will raise an exception.

For example:

```
>>> import lumache
>>> lumache.get_random_ingredients()
['shells', 'gorgonzola', 'parsley']
```

1.5 Density Functional Theory

1.5.1 Installation

To use Lumache, first install it using pip:

```
(.venv) $ pip install lumache
```

1.5.2 Creating recipes

To retrieve a list of random ingredients, you can use the `lumache.get_random_ingredients()` function:

The `kind` parameter should be either `"meat"`, `"fish"`, or `"veggies"`. Otherwise, `lumache.get_random_ingredients()` will raise an exception.

For example:

```
>>> import lumache
>>> lumache.get_random_ingredients()
['shells', 'gorgonzola', 'parsley']
```