ART-SM Backmapping Tutorial

Christian Pfaendner

August 2023

1 Theoretical Background

1.1 Backmapping

Switching between coarse-grain (CG) and atomistic resolution (illustrated in Figure 1) is desirable to take full advantage of the strengths of each modeling approach. Namely, the increased sampling and lower computational cost of CG simulations and the higher accuracy of atomistic simulations. While coarse-graining is straight forward, backmapping remains a challenging task as structural information has to be reintroduced.

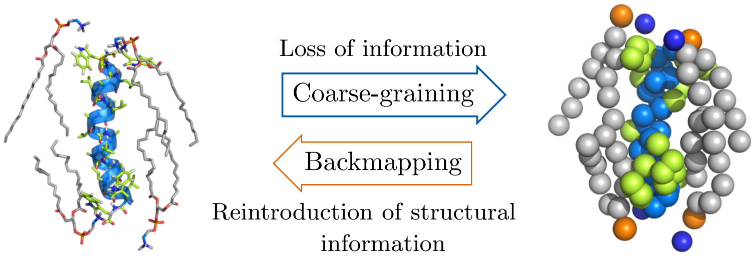

Figure 1: Coarse-graining vs backmapping. Model WALP23 peptide embedded in a DMPC lipid bilayer. Here, only 4 lipids in the surroundings of the peptide are shown for illustration purposes. Left: All atom (CHARMM36) representation with lipid nonpolar hydrogens hidden. Right: Martini representation of the same system following the 4-to-1 heavy atom to bead mapping scheme.

1.2 ART-SM Workflow

ART-SM consists of two primary steps (see Figure 2): First, a database of fragment pairs is generated from atomistic simulations and corresponding mapping files, which specify the 3D structure of atomistic molecules and their mapping to CG resolution. Second, the constructed database is used to recover atomistic details from a corresponding CG structure. More details on the workflow and the methodology can be found in our paper xy

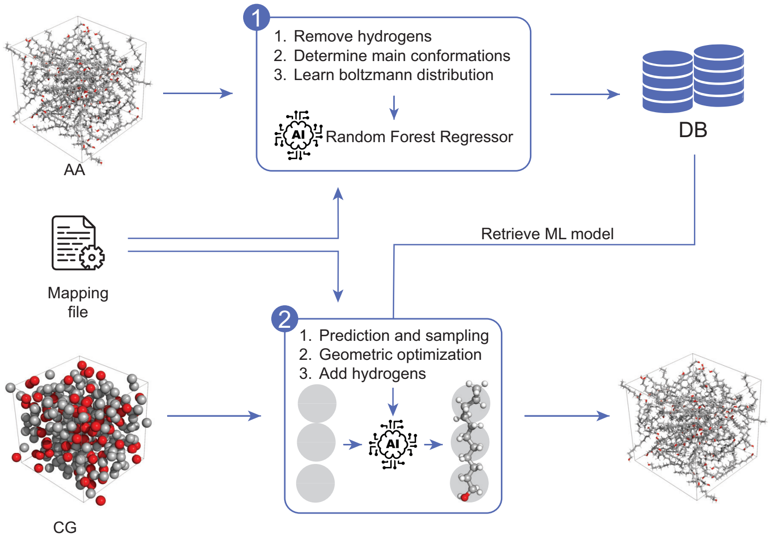
1
1. Remove hydrogens
2. Determine main conformations
3. Learn boltzmann distribution
AI
Random Forest Regressor
DB
AA
Mapping file
Retrieve ML model
2
1. Prediction and sampling
2. Geometric optimization
3. Add hydrogens
AI
CG

Figure 2: Workflow of ART-SM. Box1: A database of fragment pairs is generated by first, identifying the main conformations of the individual fragments and their connections based on the dihedral angles of heavy atoms. Second, a random forest regressor is trained to learn the probabilities of each main conformation dependent on the distances of fragments. Box 2: Given a mapping file, the coarse-grained structure is backmapped by selecting proper main conformations for fragments and their connections based on the distance of the coarse-grained beads. Afterwards, the geometry of each molecule is optimized to connect fragments. Note that all previous steps are performed separately for each fragment pair. Finally, hydrogens are added to the backmapped structure.

2 Installation

ART-SM is available on PyPI and can be easily installed with:

```
pip install artsm
```

Alternatively you can clone the GitHub repository and install it locally:

```
git clone TODO
cd ARTSM
pip install .
```

If the dependencies are not automatically installed, additionally execute:

```
pip install -r requirements.txt
```

3 Practical example: Backmapping of heptan-1-ol

In this example we backmap a simulation box containing 190 heptan-1-ol molecules in Martini resolution to all atom (CHARMM36) representation. However, ART-SM is restricted neither to any specific force field nor to a certain level of coarse-graining. This being said, the coarser your model the harder it gets to automatically identify correct main conformations and their probabilities.

3.1 Data

All the data for this tutorial is available from our GitHub repository and can be obtained with:

```
git clone TODO
```

Change into the `tutorial/data` folder and look at the files in it:

```
cd tutorial/data
ls -l *
```

The folder contains:

- `heptanol.pdb` All atom structure file containing 190 molecules of heptan-1-ol
- `heptanol.xtc` Simulation trajectory file containing 190 all atom molecules of heptan-1-ol
- `cg.pdb` Coarse-grained Martini3 structure file of heptanol.pdb
- `topol.yaml` Mapping file of heptan-1-ol from Martini 3 to all atom resolution
- `config_global.yaml` and `config_heptanol.yaml` Configuration files used for the database construction

Information on the file formats can be found here: pdb, xtc, and yaml
Note that we use the Parsers from MDAnalysis for input and output operations. Thus, all the file formats supported by MDanalysis can be used for ART-SM as well.

3.2 Mapping file

In general the mapping file (`topol.yaml` in our example) specifies the 3D structure of all molecules in the simulation system and their mapping to coarse-grained resolution. It is used for the database construction as well as for the backmapping step and is of the form:

```
Molecule1:
    smiles: str
    adj_atoms:
        atom1:
            - atom2
            - atom3
        atom2:
            - atom4
        atom3: ...
    mapping:
        bead1:
            - atom1
            - atom2
            - ...
```

```
                bead2: ...
            atom_order:
                - atom1
                - ...
        Molecule2:
        ...
```

- The first identation level defines an unique identifier for each molecule. It has to match the residue name field in the atomistic and CG pdb file.

- `smiles` Represents the molecules structure using the simplified molecular input entry system (SMILES). It is crucial to unambiguously specify the stereochemistry of the molecules, including cis/trans isomery and enantiomers. For example, specifying hept-2-en-1-ol as CCCCC=CCO neglects the cis/trans isomery at the double bond and it should therefore be defined as CCCC/C=C/CO for cis or CCCC/C=C\CO for trans conformation.

- `adj_atoms` Defines the connectivity of the molecules via a neighboring list. For instance:

  ```
                  atom1:
                      - atom2
                      - atom3
  ```

  specifies that atom1 is connected to atom2 and atom3 via a single bond. To indicate higher bond orders, provide them, separated by a hyphen, directly after the atom name. For instance `atom2-3` indicates a triple bond to `atom2`. Naming has to be consistent with the atom name field in the atomistic pdb file.

- `mapping` Describes the mapping between atomistic and CG resolution, e.g.:

  ```
                  bead1:
                      - atom1
                      - atom2
  ```

  states that the CG particle `bead1` represents the atoms `atom1` and `atom2` of the atomistic structure. `bead` and `atoms` has to match the atom name field in the CG and atomistic pdb files.

Since manually defining mapping files is an error-prone and tedious task, they can be automatically generated given an atomistic and corresponding coarse-grained structure in the same conformation. This means that for the Martini model each CG bead must be at the same position as the center of mass of the corresponding atoms at atomistic resolution.

```
        cd ../mapping
        artsm-mapping -a aa.pdb -m cg.pdb -s HEP CCCCCCCO -o hep.yaml
```

- `-a` Atomistic structure

- `-m` Congruent CG structure

- `-s` Residue identifier for each molecule followed by its SMILES string. If your system contains more than one type of molecule specify this option multiply times, e.g. `-s HEP CCCCCCCO -s PRP CCCO -s UND CCCCCCCCCCCO`
- `-o` Output mapping file

Note that providing a single molecule is not necessary. Instead the pdb files can contain multiple molecules, also of different types.

3.3 Database construction

Constructing a database is necessary for backmapping CG structures to atomistic resolution later on. However, if you are working with fragment pairs that are already contained in our pre-trained database `predef_database/release.db`, this step can be skipped. Either have a look at the molecules that were used to construct our database in some link to supplementary? or skip to Section 3.4 and try to run it with your CG system. The database can be easily constructed with:

```
cd ..
mkdir database
artsm-build_db -s data/heptanol.pdb -x data/heptanol.xtc -t
   data/topol.yaml -d database/database.db --time_step 5000
```

- `-d` Output database. Additionally, a database `release.db` will be created within the same output folder. It contains the same database except that the extracted data from the simulations are not included.
- `--time_step` [ps] Conformations of subsequent time steps are dependent on each other. To obtain independent data, it is good practice to select a frame only every [–time_step] ps for the database construction process. The correct time step `n` for your simulation depends on the simulation parameters, the molecules involved, and other factors. As a point of reference, we conservatively estimated that a time step of 500 ps is sufficient for undecan-1-ol. Here, `heptanol.xtc` was preprocessed and contains snapshots that are 5 ns apart in the original simulation to save storage.

Instead of command line flags, the arguments `-s`, `-x`, and `-t` time step should also be possible. Update after implementation can be also provided via a config file:

```
mkdir database
artsm-build_db -c data/config_heptanol.yaml -d database/
   database.db --time_step 5000
```

This works well for a single simulation. However, the actual goal is to build a large database from multiple different simulations and reuse it later for a wide variety of CG systems. Therefore you can also supply a global config file that contains paths to config files of individual simulations. It has the form:

```
simulation1: "path/to/config/of/simulation1"
simulation2: "path/to/config/of/simulation2"
...
```

In our simple example the global config file contains only the path to the config file of the heptan-1-ol simulation (have a look at `data/global_config.yaml`). To generate the database use the `-g` flag:

```
mkdir database
artsm-build_db -g data/global_config.yaml -d database/
   database.db --time_step 5000
```

Since we use data from only one simulation, all three approaches result in the same database.

3.4 Backmapping

Once we have built our database it is time to backmap the coarse-grained structure to atomistic resolution:

```
mkdir backmap
artsm-backmap -d database/release.db -s data/cg.pdb -t data/
   topol.yaml -o backmap/heptanol_backmapped.pdb --hydrogen
```

- `-d` Database
- `-s` CG structure
- `-t` Mapping file
- `-o` Output file containing the backmapped structure
- `--hydrogens` Hydrogens are added to the backmapped atomistic structure. Absence of this flag will output only heavy atoms.

The backmapped structure is a good starting structure for subsequent molecular dynamics simulations. We recommend to perform a subsequent energy minimization and short position constraint simulation on the backmapped structure.