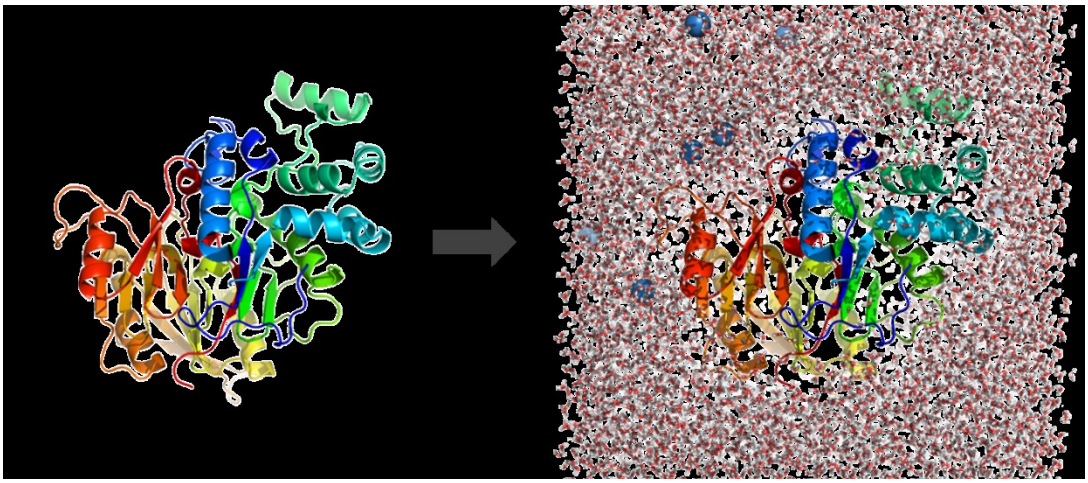
MDBuilder User's Manual

Release 1.0



October 5, 2017

『CONTENTS』

Chapter 1

Introduction

1.1 Basics of MDBuilder

MDBuilder is a free and open-source tool that allows users to build the initial configuration feeding different popular molecular dynamics (MD) simulation packages. It is written in the Python language. NumPy, the *de facto* standard library for numerical computation in Python is also utilized to enable all of its functionality. A graphical user interface (GUI) incorporated into the popular molecular visualization software PyMOL is provided.

Supported Packages:

- AMBER (version >= 10, including CHAMBER)
- NAMD
- ACEMD
- OpenMM

The general scheme of the operation flow in MDBuilder is presented in Fig. 1. Here the primary steps are summarized from the top to bottom in the figure. First of all, information from force-field-related topology and parameter files as well as the PDB-format input file is read to fill certain internal data structures. Next, MDBuilder employs required information to construct the biomolecular structures compatible with the CHARMM force field. Then, it can build initial structures by adding hydrogen atoms, wrapping each side of the protein with a layer of water, adding counter-ions, and preparing lists of force field terms and their associated parameters. At last, the resulting data is generated in two types of files that contain information required for performing energy minimization or MD simulation.

1.2 Citation

Hui Liu, Ye Jin, Youyong Li, Tingjun Hou.

MDBuilder: a program for the preparation of biomolecular simulations.

In submission

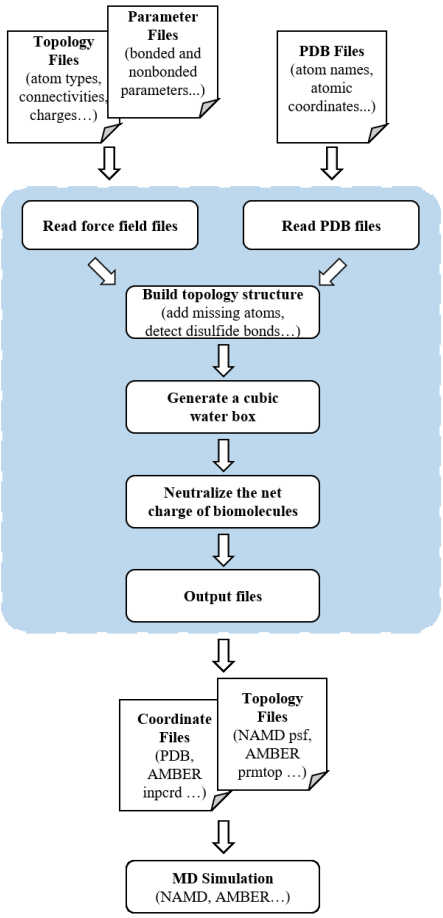


Fig. 1. The flowchart of the preparation of an initial configuration for a typical MD simulation. Procedures with light blue background are conducted by MDBuilder.

1.3 License

MDBuilder is freely available under the GPL License:

MDBuilder: preparation of biomolecular simulations

Copyright (C) 2016-2017 Zhejiang University

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Chapter 2

Installation

2.1 Prerequisites

- Python 2.7
- NumPy
- PyMOL

Please download the installation files and follow the corresponding instructions on their websites.

2.2 Download and Install

After installing the above required programs, you can download the MDBuilder source code from https://github.com/HuiLiuCode/MDBuilder. To install MDBuilder, simply extract the downloaded source file to a scratch directory. PyMOL can install plugins into the correct directory automatically with the help of the built-in plugin installation tool, via "Menu Bar -> Plugin -> Plugin Manager -> Install New Plugin-> Choose file...".

Chapter 3
Applications

3.1 Usage

3.1.1 I/O Page

The I/O Page is primarily utilized to handle the files in MDBuilder, acting as the controller of the data input/output stream. In order to run the preparation of the MD simulations, MDBuilder requires three files as follows. An initial pdb file will typically be obtained through the RSCB Protein Data Bank, and the parameter and topology files for a given class of molecule may be obtained from http://mackerell.umaryland.edu/CHARMM_ff_params.html.

Load a local file via pressing the "Browse" button.

- **PDB File:** describing Cartesian coordinates for each atom in the system (pdb-format files). A PDB file could be either specified from a local directory via the "Browse" button, or automatically downloaded from the RSCB Protein Data Bank by typing the four-character PDB accession code via the "Download" button.
- **Topology File:** describing atom names, atom types, charges, and how the atoms are connected in a molecule based on the CHARMM force field (inp-format files).

  No CMAP: indicating whether the specified topology file includes CMAP correction to proteins. If the file contains CMAP term, make sure the "No CMAP" button is in its unselected state. CMAP is an energy correction map based on quantum mechanical calculations and is one of the latest additions to the CHARMM force field. It improves protein backbone behavior and thus yields more accurate dynamic properties for the protein.
- **Parameter File:** describing numerical constants for lists of the bonds, angles and dihedrals, and atom types to evaluate forces and energies based on the CHARMM force field (prm-format files). The parameter file is closely tied to the topology file, and the two are typically distributed together and given matching names.

The resulting data is generated in two types of files that contain information required for performing energy minimization or MD simulation. After choosing a file format, set up the valid directory path and enter an appropriate file name for each output file via pressing the "SaveAs" button.

- **Topology File:** describing atom and residue names, connectivities between the atoms (lists of bonds, angles, and dihedral angles), force field parameters (numerical constants for bonded and nonbonded

interactions), and additional information needed to carry out a simulation. MDBuilder supports saving your system in two formats: pdb and AMBER inpcrd.

- **Coordinate File:** describing atomic Cartesian coordinates, and periodic box size. MDBuilder supports saving your system in two formats: NAMD psf and AMBER prmtop.

3.1.2 Preparation Page

The Preparation Page can generate a topology structure by merging the atomic coordinates of the initial structure contained in the PDB and CHARMM force field files together.

- **Rename Group**

  Segment Names: specified lists of names of the PDB sequence segments to which each atom belongs. The default name starts with "S". Remember to separate these names with at least one blank. Make sure the number of names matches that of the PDB sequence segments. For example, enter the "X1 X2" string to rename the first and second segment to X1 and X2, respectively.

  Use Default Rules: indicating whether to use system-defined rules to rename the atoms or residues.

  Specify a Rule File: indicating whether to use user-defined rules to rename the atoms or residues. Load the definition file via the "Browse" button. A text file (e.g. user_rename_rule.in) should be created that looks like this:

```
# user_rename_rule.in
aliasres HIS HSE
aliasres HOH TIP3
aliasres WAT TIP3
aliasres ZN ZN2
aliasres NA SOD
aliasres CS CES
aliasres CL CLA
aliasres K POT
aliasatom ILE:CD1 CD
aliasatom HOH:O OH2
aliasatom WAT:O OH2
aliasatom :OXT OT2
aliasatom NA:NA SOD
aliasatom CS:CS CES
aliasatom CL:CL CLA
aliasatom POT:POT POT
```

- **Disulfide Bond Group**

  Automatically Detect with a Cutoff (Å): indicating whether to automatically locate possible disulfide bonds by checking the distance between SG (gamma sulfur) atoms in cysteines (CYS). The default cutoff is 2.1 Å; that means if a distance between the SG atoms of a pair of cysteines is found to be less than 2.1 Å, a disulfide bond is assumed.

  Specify a Bond File: indicating whether to manually specify disulfide bridge definitions. Load the definition file via the "Browse" button. It is suggested to check for and refer to the presence of disulfide bonds (SSBOND) by looking at the header section of the original PDB file, as this information is not read from the PDB file.

3.1.3 Solvation Page

- **Solvents Group**

  Solvent Model: TIP3P water model, which is the most common choice. Other popular choices, such as SPC/E, TIP4P, and TIP5P, haven't been implemented in MDBuilder yet.

  Segment Name: the residue name of the water molecule in the coordinate file. The default name is WAT.

  Coordinate File: specified the coordinate file that contains a pre-equilibrated box filling with TIP3P explicit water molecules.

- **Box Parameters Group**

  Buffer Distance (Å): the minimum buffering distance between any atoms originally presented in the solute and the edges of the periodic box in angstrom. The default distance is 9.0 Å. If you use too big a number, you end up with a big water box and waste a lot of unnecessary computing time on uninteresting water molecules. However, if you use too small a water box, during the simulation, the molecule may undergo conformation changes and part of it may stick outside of the box.

  Overlap Cutoff (Å): the cutoff distance between the solute and the water molecules. The default value is 2.4 Å.

3.1.4 Ionization Page

- **Ions Group**

  Cation Model: a specific positive ion model that is applied to neutralize the system. The supported models are 'Na+', 'K+', 'Mg2+', 'Ca2+', and 'Zn2+'.

  Cation Number: the number of positive ions that is added to the system. If the "Automatically neutralize" button is activated, this counter is disabled.

  Anion Model: a specific negative ion model that is applied to neutralize the system. 'Cl-' is the only supported model.

  Anion Number: the number of negative ions that is added to the system. If the "Automatically neutralize" button is activated, this counter is disabled.

  Segment Name: the residue name of the ion molecule in the coordinate file. The default name is ION.

  Calculate the Total Charge: calculating the total charge of the system via clicking this button.

  Automatically Neutralize: indicating whether to automatically neutralize the net charge of the system.

- **Choose a Method to Place the Ions:** adding counter-ions commonly by two different ways: by electrostatic potential or randomly. The previous approach is to compute the electrostatic potentials of the system and place counter-ions one by one at the grid point where the potential is the lowest or highest.

- **Parameters Group**

  Ion-to-Ion Cutoff (Å): the cutoff distance between any ion molecule and other ion molecules in angstrom. The default value is 5.0 Å.

  Ion-to-Solvent Cutoff (Å): the cutoff distance between any ion molecule and water molecules in angstrom. The default value is 5.0 Å.

  Salt Concentration (mol/L): the concentration of counter-ions in solution. The default value is 0.0 mol/L.

3.2 Example

1. **Launch MDBuilder via "Menu Bar -> Plugin -> MDBuilder 1.0".**
2. **Set up the configuration in the "I/O" page.**

   The first step is to load the molecule. A pdb file is provided with the tutorial.

   Choose an appropriate CHARMM force field (in this case, .../MDBuilder/top_all22_prot.inp and .../MDBuilder/par_all27_prot_lipid.prm).

   Specify the output file format, and choose a valid location where the output file will be saved using a "SaveAs" dialog. See Fig. 2.

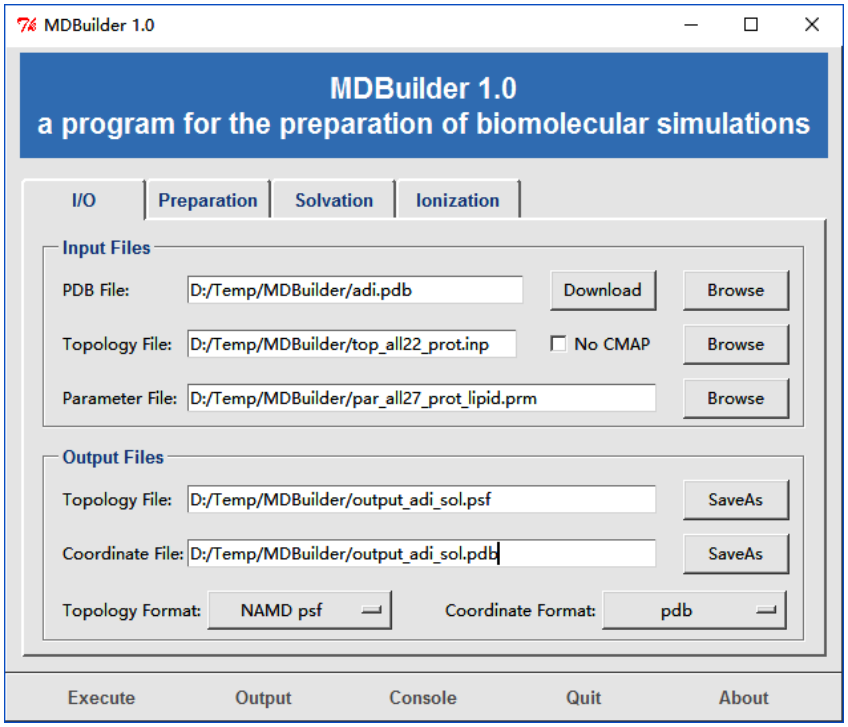


Fig. 2. The configuration in the "I/O" page.

3. **Set up the configuration in the "Preparation" page.**

   Use the default segment name of "S1" due to only one segment in the protein.

   Load a user-defined rule file containing definitions to rename the atoms and residues (in this case, .../MDBuilder/user_rename_rule.in).

   S-S bonds to pairing cysteines are automatically detected with a cutoff of 2.5 Å. See Fig. 3.

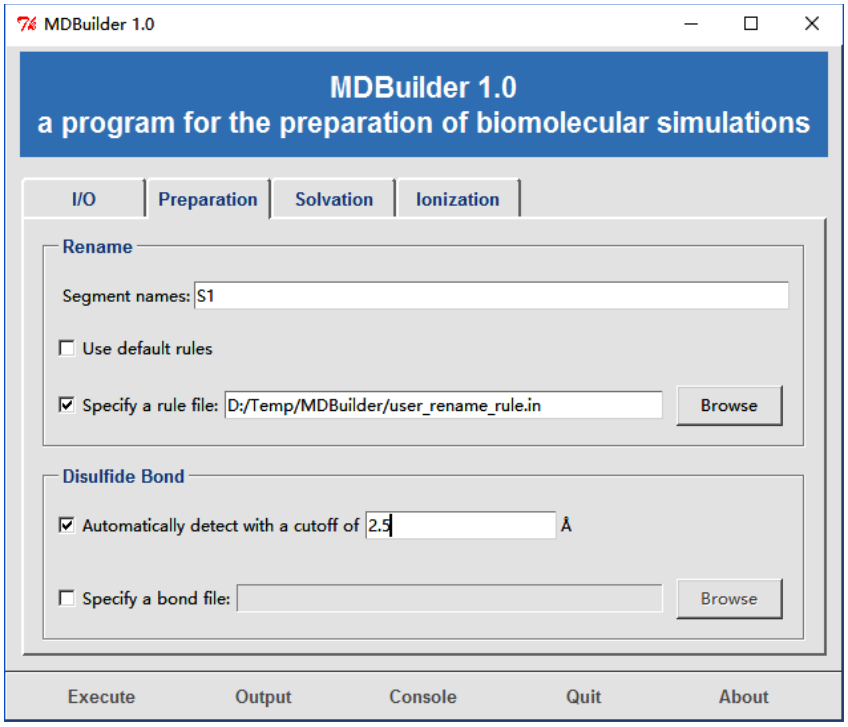


Fig. 3. The configuration in the "Preparation" page.

4. **Click the "Execute" button and wait.**

   If no errors occur, a message window containing the message “Successfully completed” pops up. Optionally click the "Console" button to take a little insight to what goes on behind the screen and some detailed information of the system in runtime. Additionally, visually inspect the model using PyMOL. See Fig. 4.

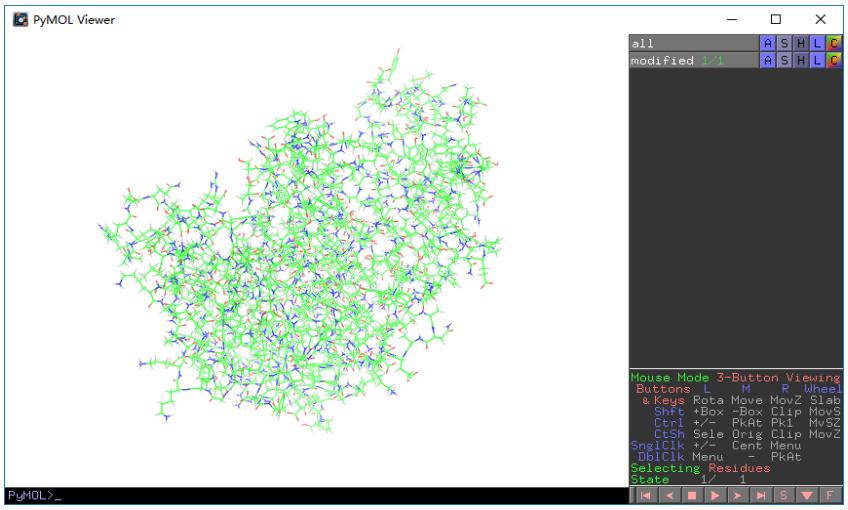


Fig. 4. The structure of the “dry” protein.

5. **Set up the configuration in the "Solvation" page.**

The detailed configuration looks like Fig. 5. The coordinate file containing TIP3P explicit water molecules (tip3p.crd) is provided with the source code.

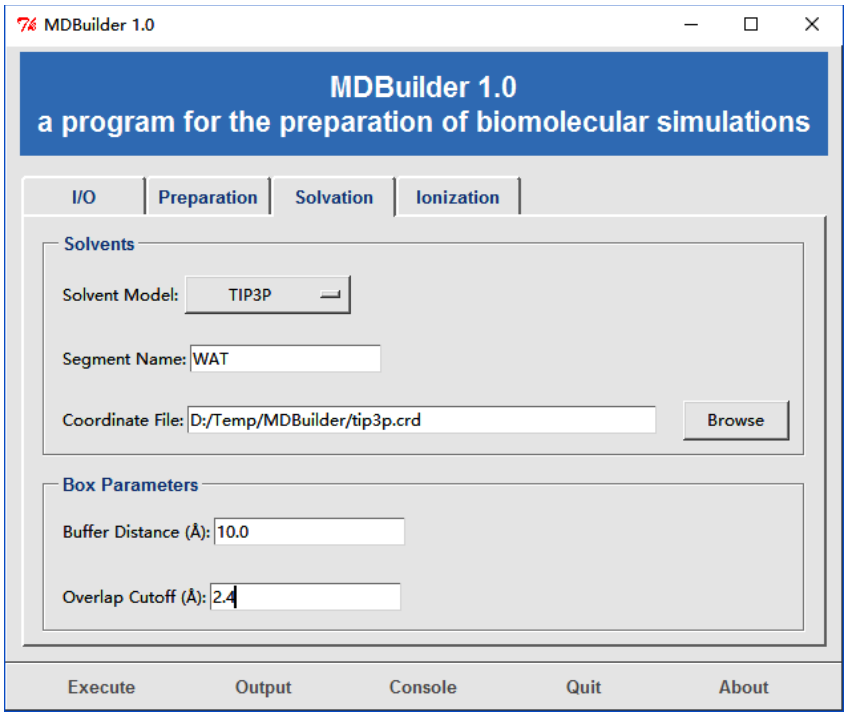


Fig. 5. The configuration in the "Solvation" page.

6. **Click the "Execute" button and wait.**

   If no errors occur, a message window containing the message “Successfully completed” pops up. Optionally check the output information via the "Console" button. It is a good idea to visually inspect the model to make sure everything looks reasonable using PyMOL. See Fig. 6.

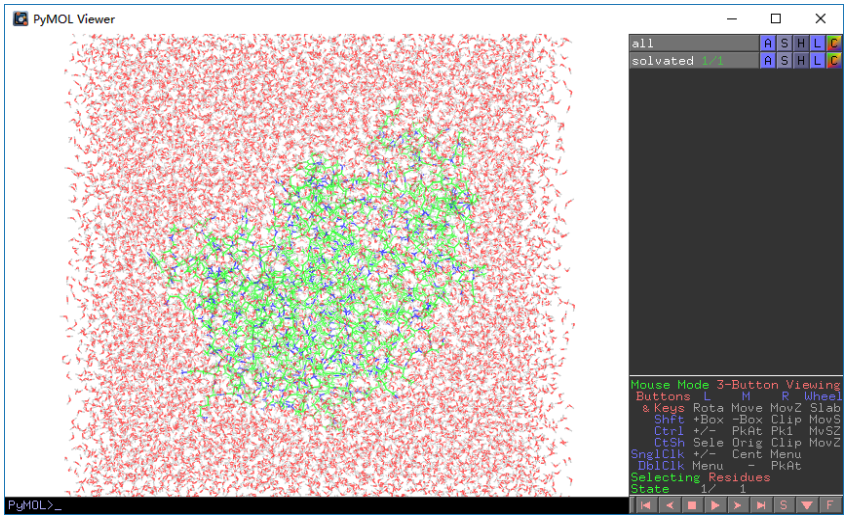


Fig. 6. The structure of the protein solvated with a cubic water box.

7. **Set up the configuration in the "Ionization" page.**

Before neutralizing the system, you can calculate the total charge via the button. In this case, the net charge is -10.0 e. For simplicity, we instruct MDBuilder to automatically neutralize the system. The detailed configuration looks like Fig. 5.

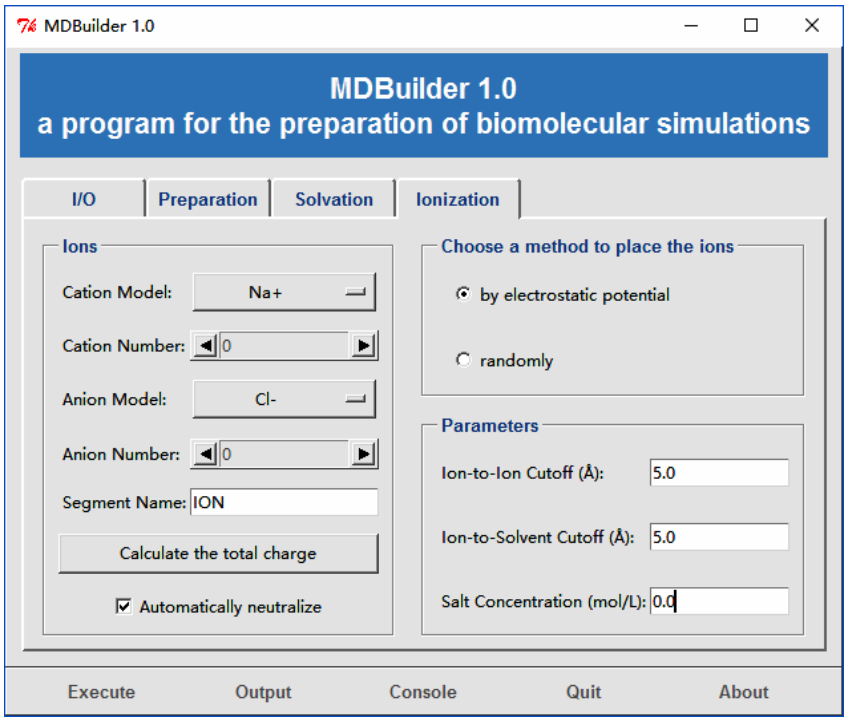


Fig. 7. The configuration in the "Ionization" page.

8. Click the "Execute" button and wait.

If nothing seems to happen on your computer for a while, just exercise a little bit of patience. Similarly, if no errors occur, a message window containing the message “Successfully completed” pops up. Optionally check the number of ions you were expecting have actually been added via the "Console" button. It is also a good idea to visually inspect the model using PyMOL. See Fig. 8.

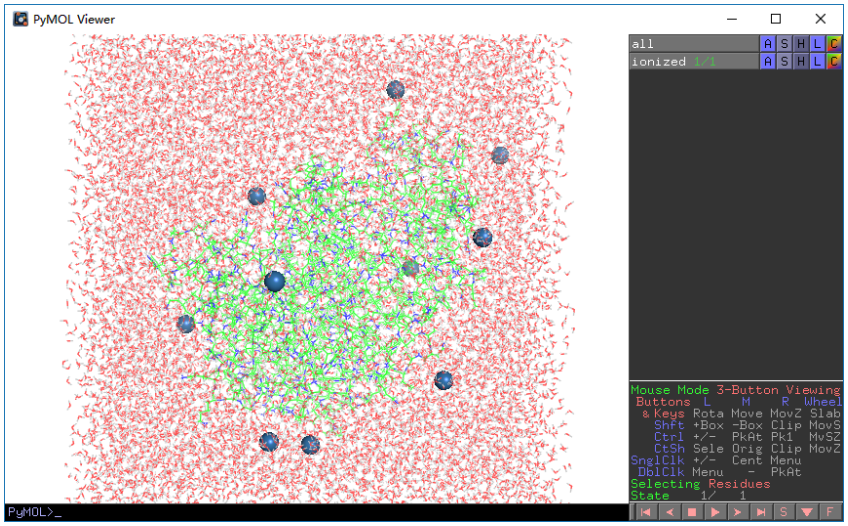


Fig. 8. The structure of the solvated protein with explicit counter-ions.

9. Click the "Output" button to generate the files.

If no errors occur, a message window containing the message "2 files were generated" pops up. Optionally check for the resulting data of the initial system, such as total mass, unit cell volume, unit cell center and density, via the "Console" button. See Fig. 9. Pay close attention to the output for any warnings or errors during the preparation.

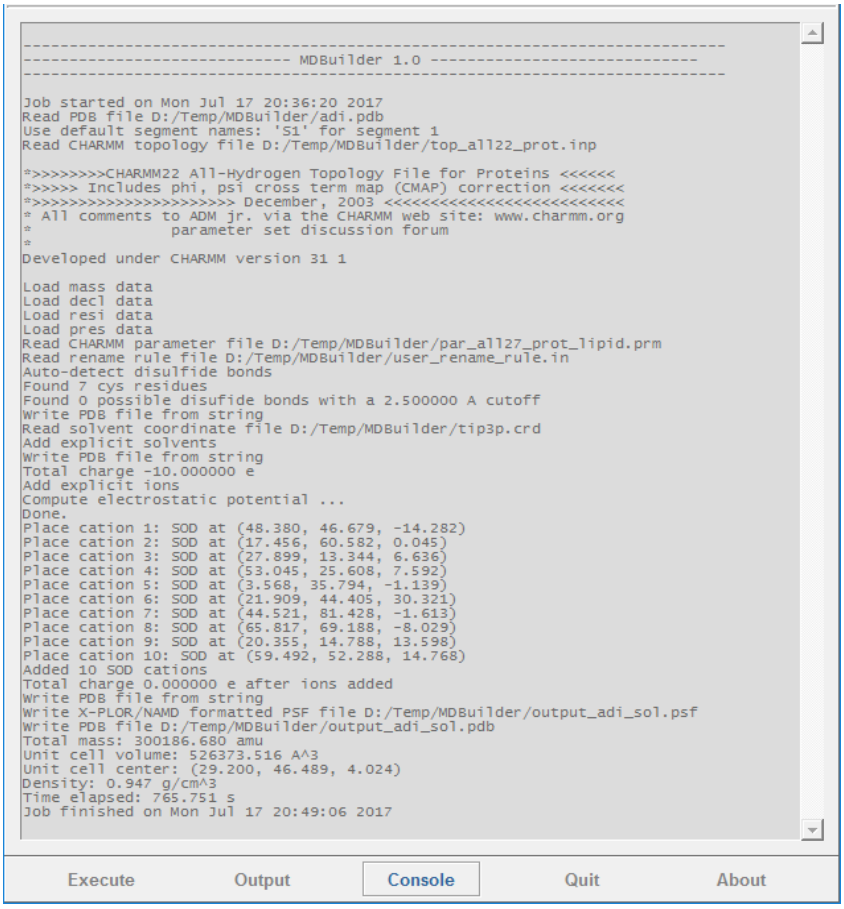


Fig. 9. The output information in the console.

10. Click the "Quit" button to quit MDBuilder.

Bibliography

Case, D.A., et al. (2005) The Amber biomolecular simulation programs. J. Comput. Chem., 26, 1668 -1688.

DeLano, W.L. (2002) The PyMOL molecular graphics system. http://pymol.org.

Phillips, J.C., et al. (2005) Scalable molecular dynamics with NAMD. J. Comput. Chem., 26, 1781 -1802.