GOMC Software Requirements
AIChE 2019

Multiple software packages are needed to prepare input files for GOMC, such as PDB and PSF files. It is strongly recommended to download and install the following software and confirm they installed correctly. The complete installation of these software will not take more than 30 minutes. If you have any difficulties installing any of these software please contact GOMC team (Mohammad Soroush or Younes Nejahi).

Software Installation Overview:

1. Homebrew (Only macOS)

Homebrew is the easiest and most flexible way to install the UNIX tools that Apple didn't include with macOS. Homebrew should be installed before any of the other software packages, since it

will make their installation significantly easier. To install Homebrew, please open Terminal and enter the following command:

```
$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Please enter your password when prompted, and after the installation is done you should see a successful message.
You may want to check for any updates by entering the following command:

```
$ brew update
```

2. Git

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. We will be using Git to download GOMC software.
Before installation we want to make sure if Git is already installed. Please enter the following command:

```
$ git --version
```

If you see an output similar to the following line, it means you already have Git installed and do not have to install it.

```
$ git version 2.16.3
```

2.1. Install on macOS

The easiest way to install Git package on macOS is to install "Command Line Tools" in macOS. This method will also install some other packages that we are going to need. To install Command Line Tools, please open Terminal and enter the following command:

```
$ xcode-select --install
```

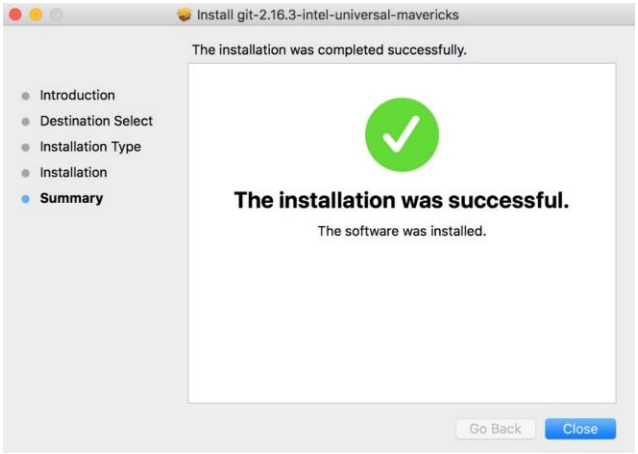
We can use Homebrew to install Git on macOS. Please open Terminal and enter the following command:

```
$ brew install git
```

Another option to install Git is using macOS installer package which can be found here. Once you open the page you should be able to see Download link along with latest version number:

Clicking download will take you to another page and will automatically download a .dmg file. Opening the file should bring you three files.

Please open the .pkg file and follow the instructions (If you receive an error while opening the file, please right click on the .pkg file and select Open. That should give you an option to open the package installer)



If the installation was successful, you should see the above message.

2.2. Install on GNU/Linux (CentOS)

The easiest way to install Git and have it ready to use is to use CentOS's default repositories. Use **yum**, CentOS's native package manager, to search for and install the latest git package available in CentOS's repositories:

```
$ sudo yum install git
```

2.3. Install on GNU/Linux (Fedora)

Similar to CentOS, you can easily install Git on Fedora using the default repositories.

```
$ sudo dnf install git
```

2.4. Install on GNU/Linux (Ubuntu)

We can install Git without having to add any repositories.

```
$ sudo apt-get install git
```

3. gcc and g++

gcc and g++ are the C and C++ compilers of GNU Compiler Collection (GCC). It is required to install these two compilers to compile GOMC.
Before installation we want to make sure if Git is already installed. Please enter the following commands:

```
$ gcc --version

$ g++ --version
```

If it is already installed you should get a version number and you may skip this step. An example output would be similar to the following lines:

```
gcc (GCC) 4.4.7 20120313 (Red Hat 4.4.7-11)
```

3.1. Install on macOS

The easiest way to install these two packages on macOS is to install “Command Line Tools” in macOS. If you installed Git using this method you don’t need to install it again. To install Command Line Tools, please open Terminal and enter the following command:

```
$ xcode-select --install
```

3.2. Install on GNU/Linux (CentOS)

To install on CentOS, you can simply enter the following command to install both gcc and g++:

```
$ sudo yum install gcc gcc-c++
```

3.3. Install on GNU/Linux (Fedora)

To install on Fedora, you can simply enter the following command to install both packages:

```
$ sudo dnf install gcc gcc-c++
```

3.4. Install on GNU/Linux (Ubuntu)

To install on Ubuntu, you can simply enter the following commands to install both packages:

```
$ sudo apt-get update

$ sudo apt-get dist-upgrade

$ sudo apt-get install build-essential
```

4. CMake:

CMake is an open-source, cross-platform family of tools designed to build, test and package software. CMake is used to control the software compilation process using simple platform and compiler independent configuration files, and generate native makefiles and workspaces that can be used in the compiler environment of your choice.

Before installation we want to make sure if cmake is already installed. Please enter the following command:

```
$ cmake --version
```

If you see an output similar to the following line, it means you already have cmake installed but you may need to upgrade your cmake.

```
$ cmake version 3.11.1
```

4.1. Install on macOS:

We can use Homebrew to install CMake on macOS. Please open Terminal and enter the following command:

```
$ brew install cmake
```

If you already have CMake, you can run the following command to upgrade it.

```
$ brew upgrade cmake
```

4.2. Install on GNU/Linux:

The easiest way to install CMake and have it ready to use is to use CentOS's default repositories. Use **yum**:

```
$ sudo yum install cmake
```

If you already have CMake, you can run the following command to upgrade it.

```
$ sudo yum upgrade cmake
```

4.3. Install on GNU/Linux (Fedora)

Similar to CentOS, you can easily install CMake on Fedora using the default repositories.

```
$ sudo dnf install cmake
```

If you already have CMake, you can run the following command to upgrade it.

```
$ sudo dnf upgrade cmake
```

4.4. Install on GNU/Linux (Ubuntu)

We can install CMake without having to add any repositories.

```
$ sudo apt-get install cmake
```

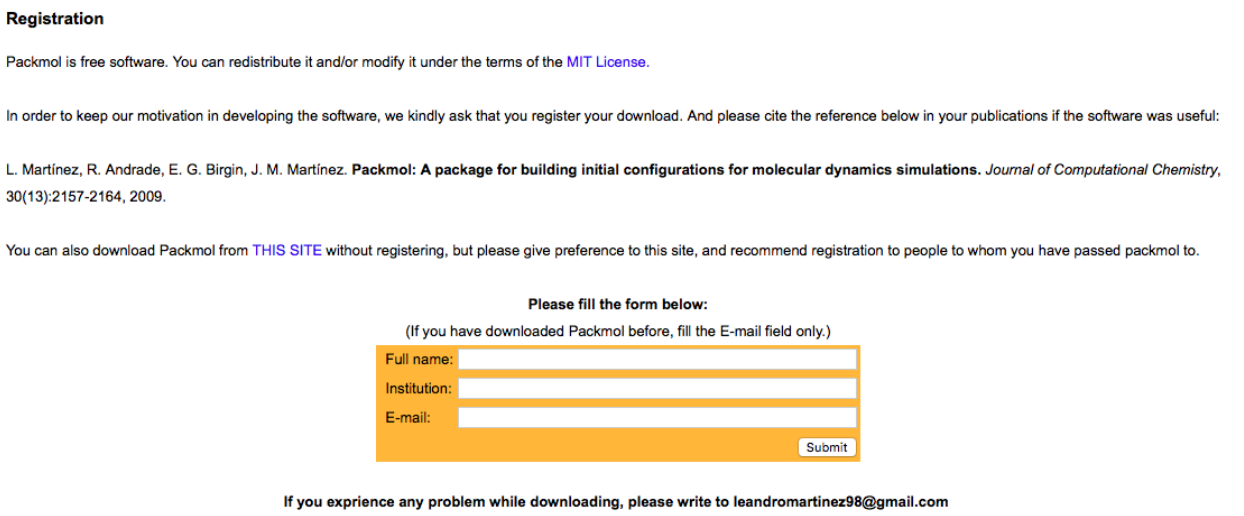
If you already have CMake, you can run the following command to upgrade it.

```
$ sudo apt-get upgrade cmake
```

5. PACKMOL:

PACKMOL creates an configuration of molecules for molecular dynamics or Monte Carlo simulations by packing molecules in defined regions of space. The packing is done in a way to minimize steric overlap while ensuring that all molecules are placed in the user defined volume. PACKMOL is written in Fortran, which requires gfortran to be compiled. Please make sure that gfortran compiler has been installed. If you have not installed gfortran, please refer to gfortran section. Detailed information on how to use PACKMOL can be find in their user manual.

Please fill out the PACKMOL form and download the latest version of PACKMOl.



Registration

Packmol is free software. You can redistribute it and/or modify it under the terms of the MIT License.

In order to keep our motivation in developing the software, we kindly ask that you register your download. And please cite the reference below in your publications if the software was useful:

L. Martínez, R. Andrade, E. G. Birgin, J. M. Martínez. **Packmol: A package for building initial configurations for molecular dynamics simulations.** *Journal of Computational Chemistry*, 30(13):2157-2164, 2009.

You can also download Packmol from THIS SITE without registering, but please give preference to this site, and recommend registration to people to whom you have passed packmol to.

Please fill the form below:

(If you have downloaded Packmol before, fill the E-mail field only.)

Full name:

Institution:

E-mail:

Submit

If you exprience any problem while downloading, please write to leandromartinez98@gmail.com

Detailed instruction on compiling the PACKMOL can be found on How to compile Packmol in user manual.

5.1. Install on macOS or GNU/Linux:

Once you have downloaded the "packmol.tar.gz" file from the home-page, you need to expand the files and compile the package. This is done by copying and executing the following command in your terminal:

```
$ tar -xvzf packmol.tar.gz
```

This will create a directory called "packmol", inside which you can find the source code. You can build the executable by copying the following command in your terminal:

```
$ cd packmol/.

$ make
```

If no error was reported the packmol executable was built. If you received the following error

```
$ make: /usr/bin/gfortran: No such file or directory
```

run the following command in your terminal to set the path for Fortran compiler and compile it again:

```
$ ./configure gfortran

$ make
```

6. VMD:

VMD is designed for modeling, visualization, and analysis of biological systems such as proteins, nucleic acids, lipid bilayer assemblies, etc. It may be used to for small organic molecules. VMD can read standard Protein Data Bank (PDB) files and display the structure. VMD can be used to animate and analyze the trajectory of a molecular dynamics (MD) and Monte Carlo (MC) simulations. Additionally, the molefracture plugin in VMD can be used to build PDB files for user generated molecules. VMD can also act as a graphical front end for an external MD or MC program by displaying and animating a molecule undergoing simulation on a remote computer.

VMD supports Linux (RHEL 6.7 and later) and Apple macOS-X (10.4.7 or later). To download the VMD, first you need to create an account on the UIUC website. To download the latest version of VMD (1.9.3) for macOS, please click here. To download the latest version of VMD (1.9.3) for Linux, please click here. After registering and agreeing to VMD license, VMD will download automatically. VMD documentation can be found here. The VMD installation guide can be found by clicking here.

6.1. Install on macOS:

To install the pre-compiled macOS X bundle version of VMD, open the VMD disk image (vmd193macx86nocuda.dmg) and drag the VMD application (vmd 1.9.3.app) into your Desktop. Open your terminal, navigate to your Desktop by copying the following command into your terminal:

```
$ cd ~/Desktop/.
```

Move and rename the VMD application to /usr/local/bin/. directory by copying the following command into your terminal: Note that this command requires administration access.

```
$ sudo mv VMD\ 1.9.3.app/    /usr/local/bin/VMD_1.9.3
```

change your directory to the location of VMD by copying the following command into your terminal:

```
$ cd  /usr/local/bin/VMD_1.9.3/Contents/vmd/.
```

Rename the VMD executable file by copying the following command into your terminal:

```
$ mv vmd_MACOSXX86  vmd
```

add the path to vmd permanently by copying the following command into your terminal:

```
$ echo 'export PATH=$PATH:/usr/local/bin/VMD_1.9.3/Contents/vmd' >> ~/.bash_profile
```

and then source the bash_profile file by copying the following command into your terminal.

```
$ source ~/.bash_profile
```

You can open the VMD now by typing vmd in your terminal.

6.2. Install on GNU/Linux:

To install the pre-compiled Unix version of VMD, then only three steps remain to be done after you uncompress the distribution by copying the following command into your terminal:

```
$ tar -xvzf vmd-1.9.3.bin.LINUXAMD64.opengl.tar
```

change your directory to vmd-1.9.3 by copying the following command into your terminal:

```
$ cd vmd-1.9.3/.
```

Generate the Makefile by copying and running the following command in your terminal

```
$ ./configure  .
```

After configuration is complete, cd to the “src” directory

```
$ cd src/.
```

and type the following command to install VMD.

```
$ make install
```

This will put the code in these two directories `"/usr/local/bin"` and `"/usr/local/lib/vmd”.` After this, you just type vmd to begin, provided that vmd is in your path.

7. Alchemlyb

alchemlyb is a library for doing alchemical free energy calculations more easily. It includes functions for parsing data from formats common to existing MD and MC engines, including GOMC, subsampling these data, and fitting these data with an estimator to obtain free energies.

These functions are simple in usage and pure in scope, and can be chained together to build customized analyses of data. Since alchemlyb is written in python3, in order to install alchemlyb, you must install python3. To install python3 using anaconda, please refer to anaconda website.

After installing python3, alchemlyb can be installed using pip:

```
$ pip install alchemlyb
```

You can also install the alchemlyb from source file. Once git is installed, alchemlyb can be installed from source file by cloning the alchemlyb repository:

```
$ git clone https://github.com/alchemistry/alchemlyb.git

$ cd alchemlyb

$ pip install .
```

For more information about alchemlyb installation and application, please refer to alchemlyb documentation.

8. Alchemical-analysis

Alchemical-analysis is an open tool written in python2, implementing some recommended practices for analyzing alchemical free energy calculations. It analyze alchemical free energy calculations conducted in GROMACS, AMBER, SIRE, and GOMC using recommended best practices from Klimovich *et al.*, JCAMD 29:397-411 (2015). This tool handles analysis via a slate of free energy methods, including BAR, MBAR, TI, and the Zwanzig relationship (exponential averaging) among others, and provides a good deal of analysis of computed free energies and convergence in order to help you assess the quality of your results. Since the developer of alchemical-analysis are in the process of migrating all functionality from alchemical-analysis to alchemlyb, a parser for GOMC free energy output is implemented in a separate GitHub repository.

In order to install alchemical-analysis, you must install python2. To install python2 using anaconda, please refer to anaconda website. After installing python2, alchemical-analysis can be installed from source file. Once git is installed, alchemical-analysis can be installed from source file by cloning the alchemichal-analysis repository:

```
$ git clone https://github.com/msoroush/alchemical-analysis.git

$ cd alchemical-analysis

$ sudo python setup.py install
```

9. GOMC

Once git, CMake, and GNU Compiler Collection (GCC) are installed, we can clone GOMC from GitHub. Navigate your terminal, where you wish to install GOMC. Copy the following command to your terminal to clone GOMC.

```
$ git clone https://github.com/GOMC-WSU/GOMC.git
```

Change your directory to GOMC and give execution permission to the "metamake.sh" file by Copy the following command to your terminal:

```
$ cd GOMC

$ chmod  u+x  metamake.sh
```

This document is created when new features such as free energy calculation and multi-particle move has not been merge to the master branch. Hence, we need to switch to "FreeEnergy" branch by running the following command in your terminal:

```
 $ git checkout FreeEnergy
```

Compile the GOMC by running the following command in your terminal:

```
$ ./metamake.sh
```

This command will create a "bin" directory and generate all the executable files in that directory.

Congratulation! You are ready to use GOMC.