

Jan 15, 2024

Protein PEGylation Protocol - Full Martini Coarse Grained v2.0 Protocol (MARTINI, INSANE, Gromacs simulations)

DOI

<https://dx.doi.org/10.17504/protocols.io.eq2lybjqlx9/v1>

compt.biology.agutierrez¹, Carmen Ili Gangas¹

¹Universidad De La Frontera



compt.biology.agutierrez Gutierrez

Universidad de La Frontera

Create & collaborate more with a free account

Edit and publish protocols, collaborate in communities, share insights through comments, and track progress with run records.

Create free account

OPEN  ACCESS



DOI: <https://dx.doi.org/10.17504/protocols.io.eq2lybjqlx9/v1>

Protocol Citation: compt.biology.agutierrez , Carmen Ili Gangas 2024. Protein PEGylation Protocol - Full Martini Coarse Grained v2.0 Protocol (MARTINI, INSANE, Gromacs simulations). **protocols.io** <https://dx.doi.org/10.17504/protocols.io.eq2lybjqlx9/v1>

License: This is an open access protocol distributed under the terms of the **Creative Commons Attribution License**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Protocol status: In development

We are still developing and optimizing this protocol



Created: January 12, 2024

Last Modified: January 15, 2024

Protocol Integer ID: 93471

Keywords: GROMACS, gromacs, martini coarse grained, PEGylation, PEGylate, CG, Molecular Dynamics, MD, molecular dynamics simulation, simulating molecular system, renowned molecular dynamics software suite, molecular dynamics technique, protein pegylation protocol, protein pegylation, application of molecular dynamics technique, molecular system, context of protein pegylation, systems from atomistic model, atomistic model, grained system, pegylation, protein stability, chemical binding of the peg polymer, protein, atomic system, gromacs simulation, simulation, simulations at extended time scale, various biomolecular properties with significant application, initial configurations for coarse, various biomolecular property, peg polymer, chemical binding, level biological system, grained format, complexity of an atomic system, atom, computational load, simplifying interaction, large number of particle, biological system

Disclaimer

All software used in this tutorial is not of my authorship and belongs to each person/company that presents right in front of its authorship.

Abstract

The construction of coarse-grained systems from atomistic models is a key strategy in simulating molecular-level biological systems. This process involves reducing the complexity of an atomic system, represented by a large number of particles, to a coarser or simplified level, enabling the study of phenomena at larger time scales. To carry out this conversion, the tool `insane.py` can be employed, designed to generate topologies and initial configurations for coarse-grained systems.

Firstly, `insane.py` is used to translate an atomistic model into a coarse-grained format, defining the relationship between groups of atoms and simplifying interactions. This step is essential to reduce computational load and enable simulations at extended time scales.

Subsequently, GROMACS, a renowned molecular dynamics software suite, is utilized to perform molecular dynamics simulations on the newly generated coarse-grained system. This process allows for the study of the dynamics and behavior of the system under realistic conditions.

In the context of protein PEGylation, a common methodology to enhance protein stability and solubility, a basic approach is followed. PEGylation involves the covalent attachment of polyethylene glycol (PEG) to proteins, providing unique properties to improve drug efficacy and reduce immunogenicity. The basic methodology involves the appropriate selection of conjugation sites on the protein and the chemical binding of the PEG polymer, often through amide-type reactions.

In summary, the construction of coarse-grained systems, the use of tools like `insane.py`, and the application of molecular dynamics techniques with GROMACS are pivotal in simulating molecular systems. Additionally, protein PEGylation offers an effective strategy to enhance various biomolecular properties with significant applications in drug development.

INTRODUCTION

- 1 The field of improving proteins via PEGylation remains a vibrant area of research, yet a profound understanding of the interactions between polymers and proteins is still in its infancy. To shed light on these interactions and enable predictive insights, Molecular Dynamics (MD) simulations offer a powerful tool, particularly when focused on specific protein-polymer systems at the molecular level. In this protocol, we provide detailed instructions on simulating PEGylated proteins, leveraging the capabilities of the latest iteration of the Martini Coarse-Grained (CG) force field.

The Martini CG force field is renowned for its ability to efficiently represent large biomolecular systems by grouping several atoms into a single interaction site. This coarse-grained approach significantly reduces computational demands while maintaining a remarkable level of accuracy. Specifically designed for simulating complex biological processes, the Martini force field captures the essential features of biomolecules, making it an ideal choice for studying intricate protein-polymer interactions.

This protocol focuses on the utilization of Martini CG force field in Molecular Dynamics simulations, providing researchers with the means to obtain nearly atomistic information. Unlike fully atomistic simulations, MD simulations with CG allow for the exploration of complex biological systems over extended temporal and spatial scales. This feature is particularly valuable when investigating the dynamic behavior of PEGylated proteins and their interactions with surrounding environments.

By presenting this protocol, we aim to empower researchers to delve deeper into the molecular dynamics of protein-PEG interactions, fostering a more comprehensive understanding of how PEGylation influences protein behavior at the molecular level. Through the integration of Martini CG force field in this protocol, we provide a sophisticated yet accessible approach to unraveling the complexities inherent in these biologically relevant systems.

MATERIALS

2 Skills Need it

Basics information skills needed:

- *Install standardized software.*
- *Run short commands in terminal*
- *Construct short scripts.*

Computational Biology Skills:

- *Understand protein format pdb and gro.*
- *Understand atom types in all atoms and coarse grained structures.*
- *Understand the biophysics of atoms in a solvent.*

3 **Hardware Requirements**

1° Personal computer. Could be a workstation or laptop.

2° 4 GB of ram is recommended. (As many laptops/workstation has more than that almost every computer will be fine)

4 **Software Requirements**

1° **Python ≥ 3.11.5.**

The easiest way to get started with Python is by downloading the Anaconda distribution. (Lower version can function anyway, but its already tested on this one)



Note

PYTHON INSTALLATION

Verify Python version in Terminal

Command

new command name

```
python --version
```

If its not the right version or if you don't have installed python (which would be rare because this Protocol its constructed in Ubuntu), you can install python using this 2 ways:

Install python directly to your system

Command

new command name

```
sudo apt-get update -y  
sudo apt-get install python3  
sudo apt-get upgrade -y
```

Install Python in an anaconda environment.

**Command****new command name**

```
mkdir -p ~/miniconda3
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-
x86_64.sh -O ~/miniconda3/miniconda.sh
bash ~/miniconda3/miniconda.sh -b -u -p ~/miniconda3
rm -rf ~/miniconda3/miniconda.sh
~/miniconda3/bin/conda init bash
```

After installing Conda, you can create your environment as you need.

Command**new command name**

```
conda create --name myenv python=3.11.5 -y
conda activate myenv
```

2° Polyply v1.0

This is the core software, with this one we create a variety of Polymer structures bounded to differences structures. For more information, please visit "[Marrik Lab - Polyply v1.0](#)", in GitHub for more information.

**Note****POLYPLY INSTALLATION**

Polymer modeling software.

Install Polyply from PyPi using:

Command

new command name

```
pip install polyply
```

Alternatively install it from GitHub using:

Command

new command name

```
pip install  
git+https://github.com/fgrunewald/polyply_1.0.git#polyply_1.0
```

3° Martini Coarse Grained

The Martini force field is a coarse-grain (CG) force field suited for molecular dynamics simulations of biomolecular systems. Could be either, installed or downloaded and executed as python scripts. Last version in this link to GitHub : "**Martinize**", for more information.



Note

MARTINI INSTALLATION

Protein conversion from all atom to coarse grained.

Downloaded and Executed version

Command

new command name

```
wget https://github.com/cgmartini/martinize.py
```

Installation version

Command

new command name

```
pip install vermouth  
# or  
pip install git+https://github.com/marrink-lab/vermouth-  
martinize.git#vermouth
```

4° Insane system constructor (Membrane Construction)

Usually used as lipid system constructor. Wide lipid structures and relatively easy to use, just need 1 command. In this tutorial, we're going to use it as system constructor, add water and ions molecules. Please, visit : "[Insane, Lipidomics Tutorial](#)", for more information.



Note

INSANE INSTALLATION*Membrane and system construction.*

Downloaded and Executed version

Command

new command name

```
wget http://www.cgmartini.nl/images/tools/insane/insane.py
```

MARTINIZE & INSANE PROTOCOL

5 FILES USED IN THIS METHODOLOGY

-  1rex.pdb 102.4KB

6 MARTINI STRUCTURE CONVERSION

From all atom to coarse grained.

In this tutorial, and as a recommendation, we should use always martinize2 or vermouth method to martinize proteins. Just because it's easier to specify some characteristics useful for molecular dynamics like "***elastic networks***" and "***position restraints***".

Martinize Proteins

Command

new command name

```
martinize2 -f 1REX.pdb -o topol.top -x 1REX_cg.gro -ff martini22 -  
elastic -ef 100 -el 0.4 -eu 0.9 -pf 1000 -p Backbone -mutate HIS:HSD
```

Note

Flags Explanation

- ef = Elastic force constant, used to preserve protein secondary and tertiary structure. It should be used to test normal all atom conformations to maintain similarities between all atom and coarse grained structures. Several values should be test and research information must be used to choose the better one for your system.
- el = Elastic lower bond cut off. As the -ef flag, must be tested or compared with crystallographic structures.
- eu = Elastic upper bond cut off. As the -ef flag, must be tested or compared with crystallographic structures.
- pf = Position Restraints. Used to avoid atom movement to equilibrate the system. Should be used with -p flag to select what kind of bead you want to restraint. Backbone its the most common choice.
- mutate = Mutate one residue to other one. In general, martini has some problems recognizing Histidines as HIS, so its recomendable to always mutate HIS:HSD.

Output Files. Files that we are going to use later.

-  1rex_cg.gro 26.4KB
-  molecule_0.itp 39.3KB

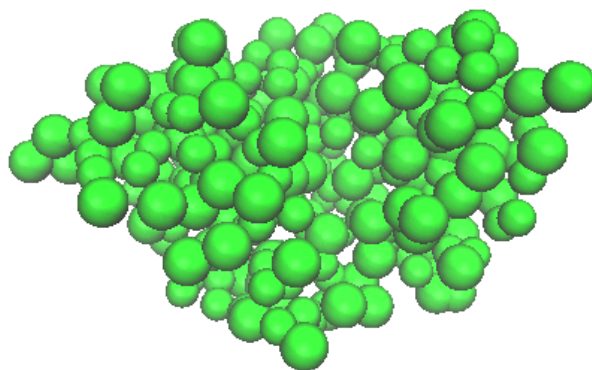
7 VISUALIZE OUR : MARTINI CONVERSION RESULTS!

The common result of converting an atomistic model to a coarse grained model is a structure made up of spheres or beads in a ratio of 4 to 1 atom per bead.

Note

ATOMISTIC MODEL OF LYSOZYME



Note**COARSE GRAINED MODEL OF LYSOZYME**

1REX_cg.gro

8 INSANE SYSTEM CONSTRUCTION

Coarse grained system generation. Preparing files for molecular dynamics.



There are different methods to create a complete system for running molecular dynamics. In general, any of these methods is useful, but the simplest one to avoid having to modify files or use too many commands is to use insane.py to generate the system, adding water molecules and ions. This can be used either to add a membrane to a protein (initially it is intended for this), but in this case we will only add a biological system.

!!!Recommendation!!!

Insane.py script use python=2, so its recommended create a conda environment with this version of python. See Step 4.

Insane System

Command

```
python insane.py -f lrex_cg.gro -o system.gro -p topol.top -salt 0.15  
-d 1 -pbc cubic -sol W:90 -sol WF:10 -charge +8
```

Note

Flags Explanation

- o = Output system, contains protein, water and ions.
- p = topologies file. Contains itp file path and molecules contained in system.gro file.
- salt = Molar concentration of salts. Recommended 0.15
- sol = W and WF types of water. If you don't intended to use Polarizable water, normal insane water (W) and antifreeze insane water (WF) need to be added. Recommended concentration W:WF are 9:1
- d = Distances from periodic images.
- charge = charge of the system. The system must be equilibrated so select the correct charge of your system.

Output Files. Files that we are going to use later.

-  system.gro 92.4KB



-  topol.top 192B

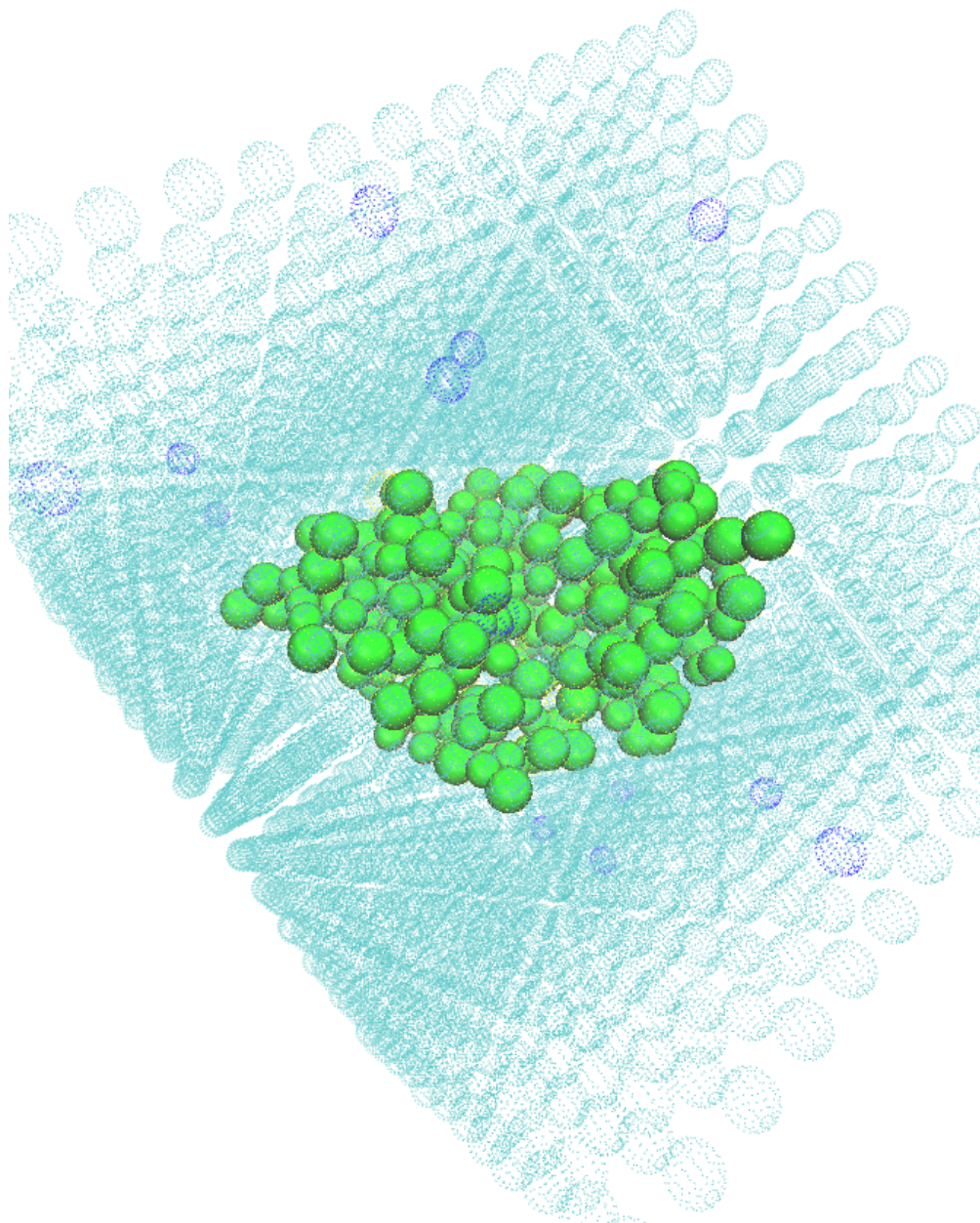
9 VISUALIZE OUR : INSANE CONSTRUCTION RESULTS!

Following the same idea as before, the result of insane is the construction of a box composed of our protein in question, chosen solvent corresponding to water molecules and antifreeze water and finally ions both sodium and chlorine molecules.

Note

COARSE GRAINED SYSTEM :

- PROTEIN --> Green beads
- SOLVENT--> Cyan
- IONS--> Blue



Obviously, the system will still consist of beads that to the human eye are **NOT** connected, however, given the parameter files such as itp files, we know that computationally, they **ARE**. So **DO NOT BE AFRAID OF THE PROCESS**, later on they will be connected and the expected structure will be displayed.

GROMACS FILE PREPARATIONS

10 FILES USED IN THIS SECTION

GENERATED IN A PREVIOUS SECTION :

-  system.gro 92.4KB → Insane Output
-  topol.top 192B → Insane Output
-  molecule_0.itp 39.3KB → Martini Output

USED IN THIS SECTION :

-  minimization.mdp 1.2KB
-  NPT_1.mdp 1.2KB
-  Production.mdp 1.3KB
-  AMINOACIDS.itp 12.3KB
-  IONS.itp 1.2KB
-  martiniv2.itp 46.7KB
-  SOLVENT.itp 429B

GENERATED IN THIS SECTION :

-  index.ndx 44KB
-  1.sh 1.8KB

11 FILES MODIFICATION PRIOR MOLECULAR DYNAMICS : ITP and TOPOL files.

ITP and Topol files modification.

Prior run MD, we will create a folder called LYSOZYME_MD. In this folder we will create 2 new folders, with the following names :

- ITP
- MDP



And copy the files generated in Insane (system.gro & topol.top) and Martini (molecule_0.itp) steps into LYZOSYME_MD folder. ***This its just to have everything organized, its nor necessary but its useful.***

<Topol.top>

This file is created by insane with default filepaths, so it is necessary to correct them to match the filepath of our files.

Note

Topol file modifications.Default **topol.top** file

```
#include "martini.itp"

[ system ]
; name
Insanely solvated protein.

[ molecules ]
; name number
Protein      1
W            1596
WF           177
NA+          16
CL-          24
```

Corrected **topol.top** file

```
#include "../ITP/martini2.itp"
#include "../ITP/AMINOACIDS.itp"
#include "../ITP/molecule_0.itp"
#include "../ITP/SOLVENT.itp"
#include "../ITP/IONS.itp"

[ system ]
; name
Insanely solvated protein.

[ molecules ]
; name number
molecule_0  1
W            1596
WF           177
NA+          16
CL-          24
```

<molecule_0.itp>



This file is created by martini in step 6, but its useful modify some arguments in it to control some characteristics.

Note

molecule_0 file modifications.

; At line 302 appears the position restraints option. This is what we set before as "Position Restraints" in step 6.

Default **molecule_0.itp** file (just the first 4 lines starting from line 302 as an example)

```
[ position_restraints ]
#ifdef POSRES
  1 1 1000.0 1000.0 1000.0
  4 1 1000.0 1000.0 1000.0
```

Modified **molecule_0.itp** file (just the first 4 lines starting from line 568 as an example)

```
[ position_restraints ]
#ifdef POSRES
  1 1 POSRES_FC POSRES_FC POSRES_FC
  4 1 POSRES_FC POSRES_FC POSRES_FC
```

; At line 568 appears the "Rubber band" option. This is what we set before as "Elastic Networks" in step 6.

Default **molecule_0.itp** file (just the first 4 lines starting from line 568 as an example)

```
; Rubber band
  1 84 6 0.89714 100.0
  1 85 6 0.64849 100.0
  1 89 6 0.52919 100.0
```

Modified **molecule_0.itp** file (just the first 4 lines starting from line 568 as an example)

```
#ifdef RUBBER_BANDS
#ifdef RUBBER_FC
#define RUBBER_FC 10
#endif
  1 84 6 0.89714 RUBBER_FC*1.00
  1 85 6 0.64849 RUBBER_FC*1.00
  1 89 6 0.52919 RUBBER_FC*1.00
```

Output file after modification:

-  topol.top 326B
-  molecule_0.itp 45.1KB

Position Restraints Modification : This modification will lead us to control the Position Restraints value from MDP files. While we enter into Equilibration steps in molecular dynamics, we want to decrease the Position_Restraints gradually, until reach the lowest Position Restraints (value of your preferences) before Production Run.

RUBBER_BANDS Modification : This modification will lead us to activate Rubber bands (our elastic networks values) from MDP files.

12 FILES MODIFICATION PRIOR MOLECULAR DYNAMICS :Input Parameters files.

Input parameters explanation and modification.

Input files can be downloaded from Martini Coarse Grained official website → [Martini Coarse Grained](#). You can modify some values depending in what do you want and your expertise in biophysics. .

<Standard Input Parameters File Attached in section 10>

Minimization Parameters: The solvated, electroneutral system is now assembled. Before we can begin dynamics, we must ensure that the system has no steric clashes or inappropriate geometry. The structure is relaxed through a process called energy minimization (EM).

Equilibration Parameters: Equilibration is often conducted in two phases, NVT and NPT phases. The first phase is conducted under an *NVT* ensemble (constant Number of particles, Volume, and Temperature). This ensemble is also referred to as "isothermal-isochoric" or "canonical." And, equilibration of pressure is conducted under an *NPT* ensemble, wherein the Number of particles, Pressure, and Temperature are all constant.

In coarse grained system, its not need it an *NVT* equilibration, so just perform the quantity needed *NPT equilibrations modifying Position restraints forces*. For this tutorial we will use an approach in which we will increase the time in the "dt" option from 1 to 15 femtoseconds, increasing 5 femtoseconds for each mdp balance file, so 4 NPT equilibration processes will be performed.

In addition to the above, the positional constraint values will gradually decrease from 1000 to 125 kJ/mol nm², decreasing by half to 125 kJ/mol nm².

Production Parameters: Upon completion of the two equilibration phases, the system is now well-equilibrated at the desired temperature and pressure. We are now ready to release the position restraints and run production MD for data collection. The process is just like we have seen before, as we will make use of the checkpoint file (which in this case now contains preserve pressure coupling information) to grompp.

Almost every parameter in mdp file can be modified. If you download mdp files from martini website, unless you really know what to change in this parameters, as an advice "**don't change the parameters**". These defaults parameters are already tested.

13 FILES GENERATION PRIOR MOLECULAR DYNAMICS : index files.

Index File explanation and generation.

The GROMACS index file (usually called index.ndx) contains some user definable sets of atoms. The file can be read by most analysis Gromacs programs. One of the most easiest files generated by Gromacs, just need 1 command.

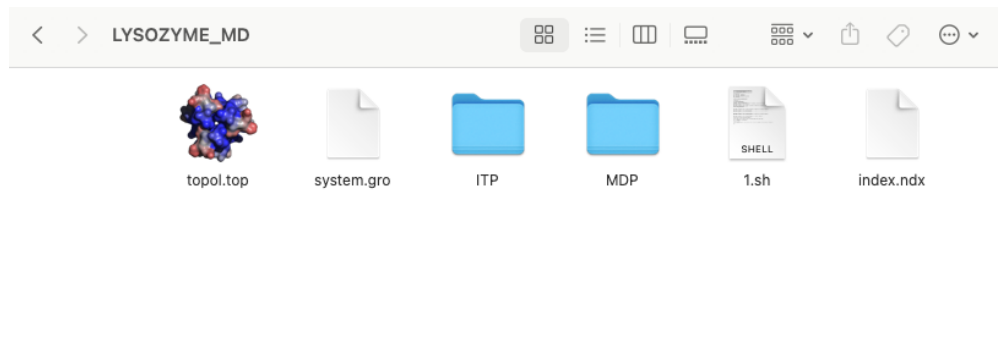
Command

new command name

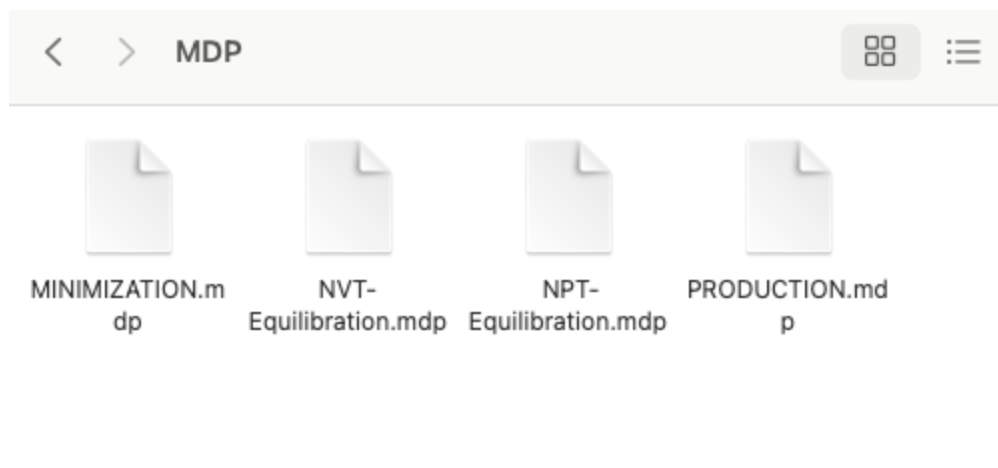
```
echo q | gmx make_ndx -f system.gro -o index.ndx
```

Unless you need to create a special index group, just use this short command to generate your index file.

14 DISPOSITION OF FILES IN LYSOZYME_MD FOLDER



LYSOZYME_MD Folder Arrangement.



MDP Folder Arrangement.



ITP Folder Arrangement.

- **MDP** : MDP folder will have every MDP file we will use during the simulation.
- **ITP** : ITP folder will have every itp file include in topol.top file.



- **Some Files** : Single files in the LYZOSYME_MD folder.
- **1.sh** : sh script for Dynamics simulation optimization.

GROMACS MOLECULAR DYNAMICS

15 MOLECULAR DYNAMICS GROMACS v2019 - SIMULATION

General commands and usage.

GROMACS stands out as essential software for simulating molecular dynamics (MD) in coarse-grained structures, simplifying representations and enabling efficient large-scale studies. Its balanced approach between accuracy and computational efficiency makes it crucial in biomolecular research.

To run a molecular dynamics you need to execute 2 commands per phase. We understand that we will have x amount of phases as x amount of mdp files. Therefore, two commands per phase are required :

<grompp command>

The first, the grompp command, will load data from both the mdp, itp, system.gro, index.ndx and topol.top files.

<mdrun command>

The second, mdrun command, executes the parameters loaded to Gromacs with the grompp command. In this command you specify how many cores to use, whether to use gpu and the name of the output files.

As an example, here is the command to run the energy minimization phase :

Command

new command name

```
gmx grompp -f minimization.mdp -o minimization.tpr -c system.gro -r
system.gro -p topol.top -n index.ndx -maxwarn 3
gmx mdrun -deffnm minimization -v -nt 6
```

Finally, to pass from the energy minimization stage to NVT equilibration, the files generated previously in the mdrun command (should be minimization.gro) must be used

to generate the new NVT files.

Command

new command name

```
gmx grompp -f NVT-Equilibration.mdp -o NVT-Equilibration.tpr -c  
minimization.gro -r system.gro -p topol.top -n index.ndx -maxwarn 3  
gmx mdrun -deffnm NVT-Equilibration -v -nt 6
```

16 VISUALIZE OUR : GROMACS MOLECULAR DYNAMICS RESULTS!

Two main results will be obtained. The files production.gro and production.xtc will correspond to the direct results of the molecular dynamics and the files run-connect.pdb and final.xtc, which will correspond to the correction of the direct results of the molecular dynamics, visible to the human eye as structures.

Both production files will have the normal format of any coarse grained file, so they are not viewable or understandable by the human eye. So the corrected files will be the ones we will use from now on.

Note

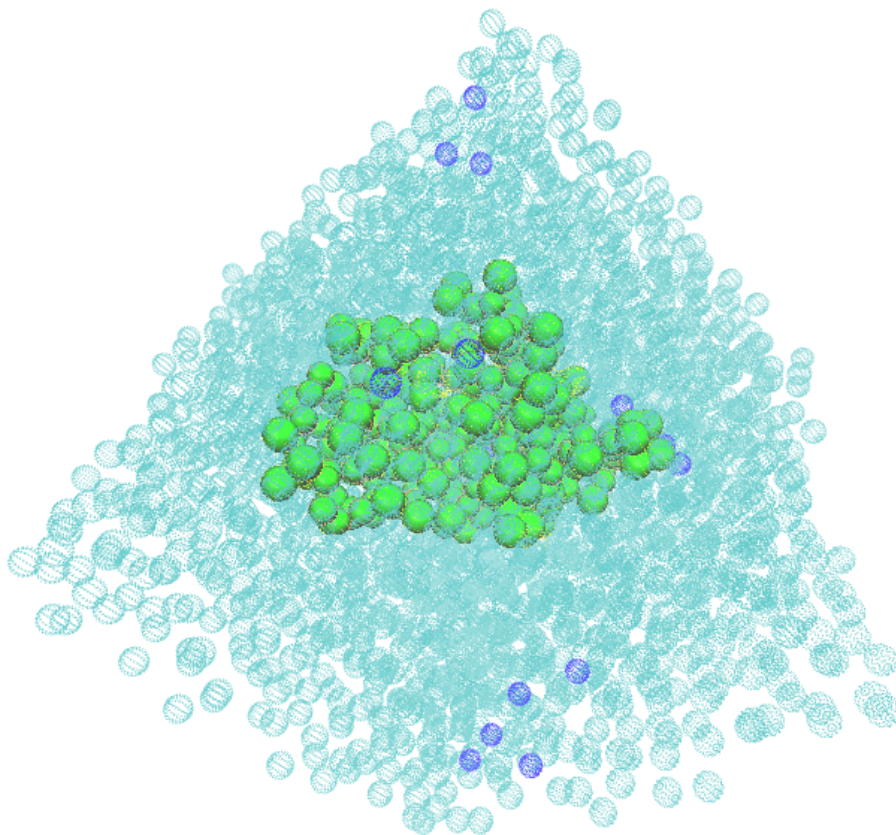
MD simulation example.

12 Frames

Protein in VDW Representation

Water and IONS in Dotted Representation

Coarse Grained **NOT** corrected.



Note

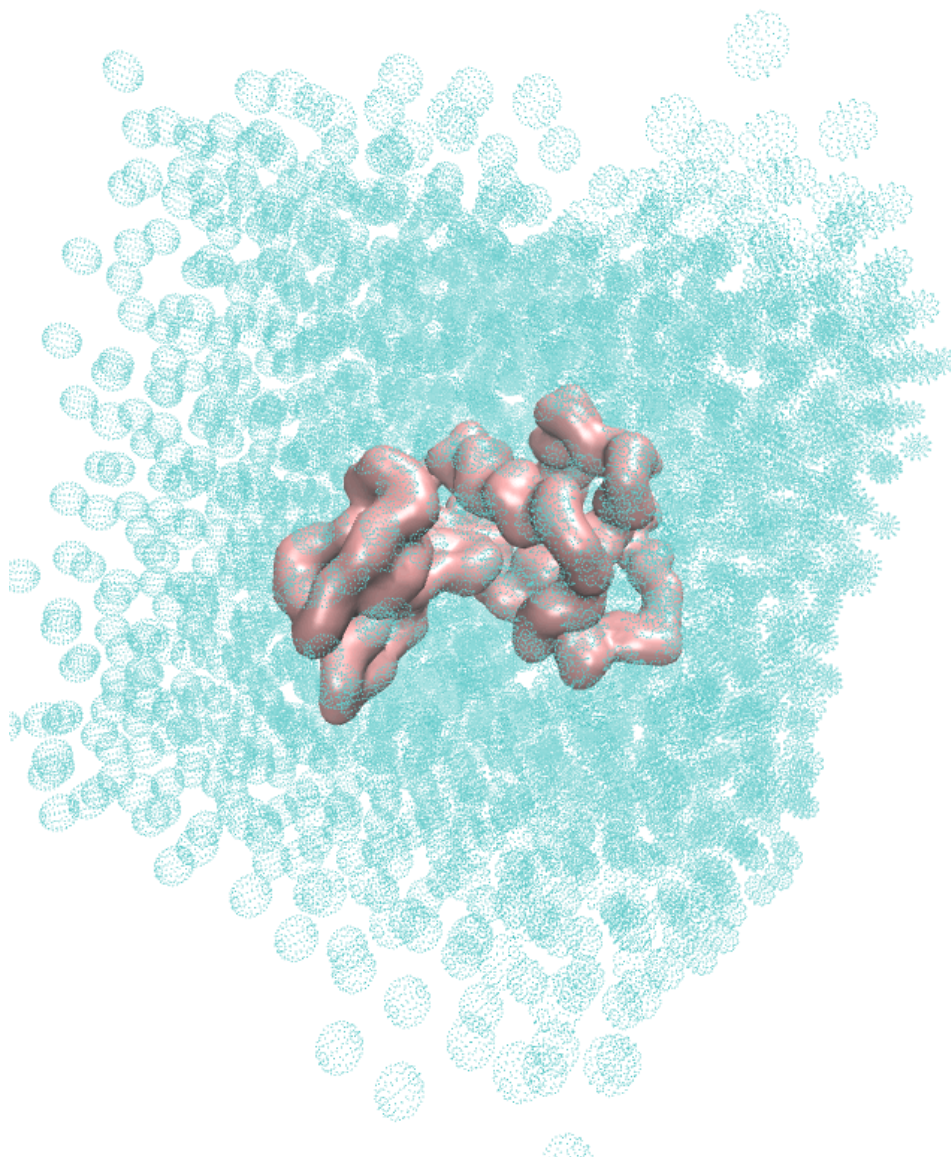
MD simulation example.

12 Frames

Protein in QuickSurfaces Representation

Water and IONS in Dotted Representation

Coarse Grained corrected.



17 MOLECULAR DYNAMICS GROMACS v2019 - GENERAL ANALYSIS

The analysis of molecular dynamics (MD) simulations in GROMACS is crucial for understanding atomic-level interactions in biomolecular structures. GROMACS provides detailed trajectories, and the analysis reveals valuable insights into dynamics and stability. Aspects such as conformational variability and binding site identification are explored, contributing to the understanding of biological processes. Advanced analytical techniques allow for precise examination of energetics. In summary, GROMACS facilitates a detailed analysis that drives advancements in biomolecular research.

This are the most common analysis for Gromacs simulations. There are others that it can be found in **Gromacs-Command**. These are attached a simple script for Gromacs analysis just with 3 command RMSD, RMSF and Gyrate. As was said, there are others analysis you would need to use but theses ones function like example of using.

Note

```
#!/bin/sh
# SCRIPT EXAMPLE FOR MOLECULAR DYNAMICS ANALYSIS.

InitFrame=Production.gro
FinRun=final.xtc
Index=index.ndx

mkdir results ; cd results

# RMSD Analysis
printf "\n3,\n3" | gmx rms -s ../Production.tpr -f ../$FinRun -n ../$Index -o
rmsd_Lysozyme.xvg -tu ns

# RMSF Analysis
echo 3 | gmx rmsf -s ../Production.tpr -f ../$FinRun -n ../$Index -o rmsf_Lysozyme.xvg -res

# GYRATE Analysis
echo 3 | gmx gyrate -f ../$FinRun -s ../Production.tpr -n ../$Index -o gyrate_Lysozyme.xvg
```

PEGylation Method - Polyply v1.0 Methodology

18 FILES USED IN THIS METHODOLOGY

GENERATED IN PREVIOUS SECTIONS :

-  run-conect.pdb 27.4KB → Clean, just the protein and connects.
-  molecule_0.itp 21.6KB → Clean, no <Position Restraints> and <Rubber Bands> options.

GENERATED IN THIS SECTION :

-  MEE.itp 59B
-  OH_end.itp 76B
-  PEO.itp 529B
-  combined_links.ff 335B
-  topol.top 326B
-  sequences.json 20.5KB

19 FILES CREATION PRIOR POLYPLY USAGE - itp and combined_links.ff

Polyply v1.0 is a specialized software designed for the efficient development of polymers within a coarse-grained force field framework. This innovative tool streamlines the process of generating polymer structures, including polymer melts, amorphous blends, and liquid-liquid phase-separated systems. Its primary function revolves around simplifying the creation of input files (itp files) and initial configurations for diverse polymer scenarios. Polyply proves particularly valuable for researchers and scientists engaged in molecular dynamics simulations, offering a user-friendly interface and facilitating the exploration of polymer behaviors at a larger scale.

The polymers as PEG are composed by 3 type of bead.

- Linker bead → MEE.itp
- Polymer bead → PEO.itp
- OH bead → OH_end.itp

If you need more information of what kind of bead are related or why are need it, please visit : **PEGylation Proteins Book**. They published a book with a methodology in Martini 3 version, methodology that don't work with Polyply v1.0 for martini 2.

The PEO.itp file can be downloaded from **Martini Coarse Grained** Website, and its has this format



```
[ moleculetype ]
PEO 1
;
[ atoms ]
1 EO 1 PEO EO 1 0.000 45
[ bonds ]
; back bone bonds
1 2 1 0.37 7000
;
[ angles ]
1 2 3 2 135.00 50
1 2 3 10 135.00 75
;
[ dihedrals ]
1 2 3 4 1 180.00 1.96 1
1 2 3 4 1 0 0.18 2
1 2 3 4 1 0 0.33 3
1 2 3 4 1 0 0.12 4
```

The MEE.itp and OH_end.itp files can be found on [Martini PEGylation Proteins Book](#) , and its has this format

```
; MEE.itp file
[ moleculetype ]
MEE 1
[ atoms ]
1 N0 1 MEE MEE 1 0.000 72
```

```
; OH_end.itp
[ moleculetype ]
; name nexcl.
OHend 1
[ atoms ]
1 P1 1 OHend OH 1 0.000 36
```

<Attached Files >

<Attached Files >

The combined_links.ff file need to be written the first time. This is because there is no single combined_links.ff file, but depends on the structure of the protein you want to PEGylate and the amount of PEG molecules you plan to add to the system.

For this tutorial we select **aminoacid number 1**, the first Lysine, other relevant information is Lysozyme has only **287 beads** in CG format.

```
; combined_links.ff file created for a PEGylated polymer with 5
beads.
[ link ]
[ molmeta ]
by_atom_id true
[ bonds ]
3 288 1 0.41 2000 ; R-MEE
288 289 1 0.39 5000 ; MEE-PEG
293 294 1 0.28 7000 ; PEG-OH
[ angles ]
288 3 2 2 150 15 ; MEE-Qd-C3
289 288 3 2 170 50 ; EO-MEE-Qd
292 293 294 2 150 15 ; EO-EO-OH
```

In order not to confuse the reader and since the combined_links.ff file is one of the most relevant files for protein gluing, a notes section is added where each section of the file is explained. In order not to confuse the reader.

Note

Explanation of the combined_link.ff format for 5 beads PEG polymer.

The format of bonds parameters :

```
[ bonds ]
3 288 1 0.41 2000 ; R-MEE
288 289 1 0.39 5000 ; MEE-PEG
293 294 1 0.28 7000 ; PEG-OH
```

R-MEE :

- 3 : Number of the bead to which you want to attach the MEE bead that will be the linker between the amino acid and the PEGylate. In this case LYS 1 has 3 beads, and its selected the bead named SC2, atom number 3 for PEGylation
- 288 : Bead number that will correspond to the bead linker or MEE. This number always corresponds to the total number of natural beads of the +1 protein. In this case the protein has a total of 287 beads, so the number corresponding to the MEE bead will be 288.
- 1 : Func. type of bonds.
- 0.41 2000 : Will be a standard for Aminoacid-MEE connection

MEE-PEG :

- 288 : Bead number that correspond to MEE bead.
- 289 : Bead number that correspond to first PEG bead
- 0.39 5000 : Will be a standard for MEE-PEG connection

PEG-OH :

- 293 : Bead number that correspond to last PEG bead.
- 294 : Bead number that correspond final OH bead.
- 0.28 7000 : Will be a standard for PEG-OH connection

The format of angles parameters (follow the same format that the bond format)

[angles]

288 3 2 2 150 15 ; MEE-SC3-SC2

289 288 3 2 170 50 ; EO-MEE-SC3

292 293 294 2 150 15 ; EO-EO-OH

MEE-Qd-C3

- 288 3 2 : Connection of 3 beads. In this case correspond to MEE-SC3-SC2.
- 2 : Func. type of bonds.
- 150 15 : Will be a standard for MEE-SC3-SC2 connection

PEG-MEE-Qd

- 289 288 3 : Connection of 3 beads. In this case correspond to PEG-MEE-SC3.
- 2 : Func. type of bonds.
- 170 50 : Will be a standard for PEG-MEE-SC3 connection

PEG-PEG-OH

- 292 293 294 : Connection of 3 beads. In this case correspond to PEG-PEG-OH.
- 2 : Func. type of bonds.
- 150 15 : Will be a standard for PEG-PEG-O connection

20 POLYPLY USAGE METHODOLOGY

To generate any protein bound to x number of polymers, 3 commands must be used. All these commands are included in the POLYPLY software. These commands are : polyply gen_seq, polyply gen_params & polyply gen_coords. The first one, polyply gen_seq is in charge of creating the sequences.json file, which corresponds to a file that only takes graphic input in the form of a json file. The second, polyply gen_params, generates an itp file with the parameters of the amino acid sequence plus the PEG molecules we want to add in the place we set earlier. Finally, the last command, polyply gen_coords, will use the coordinates of our coarse grained protein and the previously generated itp file to create a .gro file with the coordinates of our new PEGylated protein.



POLYPLY GEN_SEQ

Recommendation : Before use polyply gen_seq, copy from molecule_0 all lines related with position restraints and rubber bands to other blank text document. Then erase from molecule_0 those lines. Polyply gen_seq always give error when you use it when position restraints and rubber bands.

Command

new command name

```
polyply gen_seq -f molecule_0.itp -from_file protein:molecule_0 \  
                -from_string linker:1:1:MEE-1.0  
polymer:5:1:PE0-1.0 end:1:1:OHend-1.0 \  
                -seq protein linker polymer end \  
                -connects 0:1:0-0 1:2:0-0 2:3:4-0 \  
                -o sequences.json -name test \  
                -label 0:"from_itp":"molecule_0-1"
```



Note

Command Explanation Line by Line

-f --> Set parameter itp file. In this case we going to use the file of our protein, molecule_0.itp
-from_file --> set as "protein" whatever called molecule_0. This name need to be the same as the [moleculetype] in molecule_0.itp.
-from_string --> set as "linker", "polymer", and "end" to MEE, PEO, and OHend beads. Its also set the number of each one that will be inserted. In this command we are going to generate 1 MEE bead, 5 PEO beads and 1 end bead.
-seq --> Set how this beads will be concatenated. Protein, then Linker, then Polymer, and finally end beads.
-connects --> Set which seq will be bound to each other and what residue from each seq will correspond to the bound. As polyply always start counting from 0, Protein is 0, Linker is 1, Polymer is 2, and end is 3.
So 0:1:0-0 has 2 meanings. **0:1** means, seq 0 and 1 will bound each others, protein and linker. **0-0** means, the residue number 0 from seq 0 (Protein) will bound to residue 0 from seq 1 (linker).
So **1:2:0-0** means : seq 1 (linker) and seq 2 (Polymer) are bounded. And residue number 0 from seq 1 its bound to residue number 0 from seq 2.
Remember that polyply start counting from 0. So if its appears as residue number 1 (in itp file) it should be set as residue number 0.
-o --> Output, sequences.json.
-name --> Could be any name, in this case we are going to use test.
-label --> label for new [moleculetype]

Output Files. Files that we are going to use later.

-  sequences.json 20.5KB

POLYPLY GEN_PARAMS

Command

new command name

```
polyply gen_params -f molecule_0.itp MEE.itp PEO.itp OH_end.itp  
combined_links.ff -seqf sequences.json -o lysoPEG.itp -name lysoPEG
```

Output Files. Files that we are going to use later.

-  lysoPEG.itp 23.4KB



POLYPLY GEN_COORDS

Recommendation

After polyply command, remember change charge of the bead from 1.0 to 0.0.

Command

new command name

```
polyply gen_coords -p topol.top -o lysoPEG.gro -c run-conect.pdb -  
name lysoPEG -dens 1
```

Output Files. Files that we are going to use later.

-  lysoPEG.gro 13.1KB

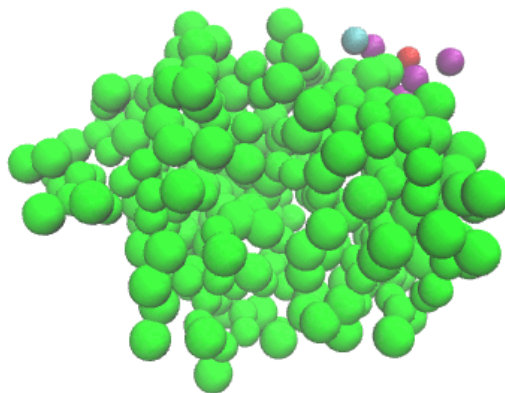
21 **VISUALIZE OUR : PEGylation Method - Polyply v1.0 Methodology RESULTS!**

Following the same idea as before, the result of PEGylation is a model composed by our protein in question, and MEE linker, PEG beads and OH end.

Note

PEGylation Procedure completed.

Protein in Green VDW Representation.
MEE linker in cyan VDW Representation.
PEG bead in violet VDW Representation.
OH end bead in red VDW Representation.
Coarse Grained **NOT** corrected.



PEGylated Protein GROMACS MOLECULAR DYNAMICS

22 As explained above, once we have our pegylated protein structure, we will proceed to the construction of the system.

For this we used the same steps previously performed, listed below:

- **Step 8** → To build the Pegylate system in a water box with ions. Make sure that you add some dimensions for box, changing the old command to this one, this will help to prevent some particles move into the void.

Command

new command name

```
python insane.py -f lyzoPEG.gro -o system.gro -p topol.top -x 10 -y 10 -z 10 -salt 0.15 -d 1 -pbc cubic -sol W:90 -sol WF:10 -charge +8
```

- **Step 11 to 14** → To organize the folder where we will run our dynamics. We are going to use the same MDP and ITP files used before, with 1 exception, our new lysoPEG.itp file.
- **Step 15** → To run molecular dynamics. We are going to use the same command as before.
- **Step 17** → Analyze molecular dynamics.
- **Step 18** → We erase Position Restraints and Rubber Bands of our molecule_0.itp file. Now those lines we must add again to our lysoPEG.itp file.

NEW FILES USED IN THIS METHODOLOGY

-  topol.top 306B
-  system.gro 401.7KB

23 VISUALIZE OUR : PEGylated Protein MD RESULTS!

As in step 16. Two main results will be obtained. The files production.gro and production.xtc will correspond to the direct results of the molecular dynamics and the



files run-conect.pdb and final.xtc, which will correspond to the correction of the direct results of the molecular dynamics, visible to the human eye as structures.

Both production files will have the normal format of any coarse grained file, so they are not viewable or understandable by the human eye. So the corrected files will be the ones we will use from now on.

Note

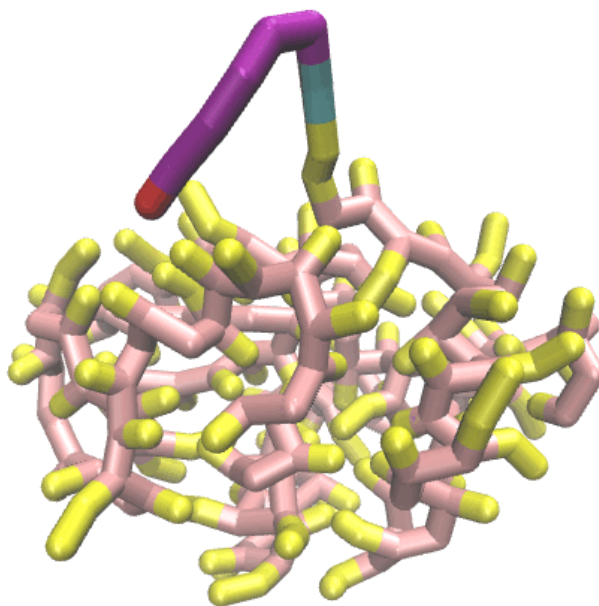
MD simulation example.

12 Frames

Protein in Licorice Representation

Water and IONS not represented for clarity.

Coarse Grained corrected.





FINAL REMARKS

- 24 Finally, the system for creating a coarse grained structure and adding any number of polymers of choice, such as in this case PEGylates, is an efficient way to understand the behavior of any possibly modified structure.

Anyway, please refer to the official pages of each program for more information.