Gromacs

23 de enero 2023

Simulación de moléculas orgánicas utilizando la paquetería de Dinámica Molecular GROMACS

Para realizar una Dinámica Molecular Clásica empleando una paquetería específica para moléculas, se requiere seguir los mismo pasos que realizamos en nuestro código escrito en python:

1. Construcción del sistema
2. Asignación del potencial de interacción
3. Optimización de geometría
4. Equilibrio del sistema
5. Simulación del sistema

Construcción del sistema

Construir el sistema significa generar un archivo con las coordenadas 3D de las moléculas que tenemos interés en simular, necesitamos incluir la molécula de interés en su solvente, iones, etc. Para construir estos sistemas hay varias opciones pero en general estas opciones las podemos dividir en dos: la primera es obtener las coordenadas de una base de datos en el que se publican coordenadas a partir de datos experimentales, como la cristalografía de rayos X o la Resonancia Magnética Nuclear. La segunda opción es utilizar herramientas computacionales para generar las coordenadas del sistema.

En este ejercicio utilizamos la segunda opción y utilizamos tres programas para generar las coordenadas iniciales de nuestro sistema.

- **Openbabel** es una caja de herramientas de química, y puede convertir, analizar o almacenar estructuras químicas de un formato a otro. Típicamente Openbabel se utiliza desde una línea de comandos y es capaz de manejar datos para modelado molecular, química, materiales, bioquímica, etc. (http://openbabel.org/wiki/M

- **Avogadro** es un editor y visualizador de moléculas, fue diseñado para generar archivos de entrada de varios programas de química computacional, bioinformática, ciencia de materiales, etc.

- **Gromacs** es el software cuya función principal es realizar la simulación de dinámica molecular que revisaremos en esta sección. A su vez, la paquetería de Gromacs incluye una gran cantidad de utilerías para construir un sistema y analizar la trayectoria de nuestra simulación. Como en esta sección estamos describiendo cómo construir un sistema, es importante mencionar que dentro de gromacs, utilizaremos las utilierías *editconf, insert-molecules y solvate* para armar nuestro sistema.

- Abrimos el programa **Avogadro** y construimos nuestra molécula de interés. Optimizamos su geometría utilizando los parámetros que vienen por default en el programa. Guardamos la molécula que generamos en el formato *mol2*.

- Utilizamos el servidor **Swissparam** para **parametrizar** nuestra molécula con el campo de fuerza de CHARMM. (https://www.swissparam.ch/). Parametrizar quiere decir generar un archivo de topología en el que estarán enlistadas las coordenadas de la molécula junto con todos los parámetros que se requieren para realizar una simulación de acuerdo con el potencial y el campo de fuerza deseado. Estos parámetros deben ser compatibles y consistentes en toda la simulación. De este servidor necesitamos dos archivos: (1) el archivo con la extensión *.itp* contiene la lista de parámetros para nuestra molécula y (2) el archivo con la extensión .pdb contiene las coordenadas con los nombres de los átomos consistentes con el campo de fuerza de CHARMM

  Ya que tenemos el archivo de coordenadas con los nombres adecuados del campo de fuerza, en este caso CHARMM vamos a realizar la **construcción del sistema** con las utilerías de gromacs:

  1. Con el comando *editconf* editamos la configuración de nuestra molécula, la acomodamos en una *caja*.

```
> gmx editconf -f mol.pdb -bt cubic -d 2.5 -o mol_caja.pdb
```

2. Con el comando *insert-molecules* vamos a inserta moléculas adicionales dentro de la caja que generamos con el comando anterior. El número de moléculas se especifica con -nmol, la molécula que queremos insertar tiene que estar en formato *pdb* y se especifica con -ci.

```
> gmx insert-molecules -f mol_caja.pdb -ci mol.pdb -nmol 10
    -o 10mol_caja.pdb
```

3. Una vez que tenemos en la caja el número de moléculas que queremos simular agregamos el solvente con el comando *solvate*, en este caso utilizaremos agua. Con -cp especificamos las coordenadas de nuestro sistema, con -cs se especifica la caja de solvente que se utilizará para llenar nuestra caja de solvente y con -o es se especifica el nombre del archivo de salida.

```
> gmx solvate -cp 10mol_caja.pdb  -cs spc216
     -o sistema_solv.pdb
```

Asignación del potencial de interacción

Ya que tenemos nuestro sistema construido, vamos a generar el archivo de topología. En este archivo vamos a combinar los parámetros de todas las moléculas presentes en nuestro sistema. De hecho, nosotros generamos el archivo de parámetros de nuestra molécula utilizando el servidor swissparam, ese archivo contiene la información de las interacciones de esa molécula.

El archivo de topología (*topol.top*) engloba el campo de fuerza general que vamos a utilizar y los archivos de parámetros de las moléculas de interés. Además, tiene un índice de las moléculas que constituyen el sistema. No puede haber errores o typos en este archivo. El número y el orden de las moléculas del índice final debe ser correcto, de acuerdo con el archivo de coordenadas del sistema. El archivo tiene la siguiente forma:

```
; Incluir el campo de fuerza
#include "charmm22.ff/forcefield.itp"

; Incluir los archivos de topologia de cada molecula
#include "mol.itp"

; Incluir la topologia del agua
#include "charmm22.ff/tip3p.itp"

; Incluir la topologia de los iones
#include "charmm22.ff/ions.itp"

[ system ]
; Nombre del sistema
Ejercicio 1

[ molecules ]
; Compuesto        #de moleculas
LIG     10
SOL     3739
```

Los errores más comunes en este archivo son: que el número de moléculas no sea el adecuado, que no encuentre los archivos *.itp* o que no estén en el orden adecuado.

Optimización de Geometría

La construcción del sistema la hicimos de forma *aleatoria*, pusimos una molécula, definimos la caja del sistema, metimos otras moléculas de forma aleatoria y llenamos de agua también de forma aleatoria. Es posible que algunas de las moléculas de nuestro sistema hayan quedado lo suficientemente cerca como para generar algún error a la hora de hacer la simulación. De tal forma vamos a realizar una optimización de la geometría del sistema utilizando el programa Gromacs antes de iniciar la dinámica molecular.

Gromacs se ejecuta desde la terminal y los dos comandos principales para hacer una dinámica molecular son grompp y mdrun.

- **grompp** es el procesador de gromacs, este comando lee la topología de una molécula, lee las coordenadas del sistema y revisa que el archivo de topología empate con las coordenadas del sistema. Posteriormente grompp lee las instrucciones del archivo *simul.mdp*, en este archivo le vamos a indicar a gromacs el conjunto de algoritmos que vamos a utilizar para hacer la simulación u optimización del sistema.

```
> gmx grompp -f min.mdp -c ../constr/sistema.pdb
    -p topol.top -o min -maxwarn 1
```

- **mdrun** es la maquinaria principal de gromacs. Evidentemente, mdrun realiza la Dinámica Molecular pero también puede hacer optimizaciones, dinámicas estocásticas, cálculos de energías, etc.

```
> gmx mdrun -deffnm min -v
```

```
; Estos parámetros indican qué hará, cuándo detenerse y qué guarda
integrator  = steep         ; Algoritmo de minimización
                        (steep = steepest descent minimization)
emtol       = 1000.0        ; Detiene la minimization cuando
                        la máxima fuerza sea < 1000.0 kJ/mol/nm
emstep      = 0.01          ; El tamaño del paso de la minimización
nsteps      = 50000         ; El número máximo de pasos

; Parámetros para describir las interacciones y los vecinos

nstlist         = 1         ; frecuencia con la que se actualiza
                              la lista de vecinos
cutoff-scheme   = Verlet    ; Algoritmo para apagar la
                              interacción en la frontera
ns_type         = grid      ; Método de busca de
                              vecinos para la lista
coulombtype     = PME       ; Tratamiento de interacciones
```

```
                                    electrostáticas de largo alcance
rcoulomb        = 1.0           ; A qué distancia se cortan las
                                    interacciones electrostáticas
rvdw            = 1.0           ; A qué distancia se corta las
                                    interacciones de Van der Waals
pbc             = xyz           ; Usa condiciones periódicas a la
                                    frontera en las 3D
```

Equilibrio del sistema

Es indispensable que antes de calcular las propiedades que nos interesa sobre nuestro sistema verifiquemos que se encuentra en condiciones de equilibrio. De lo contrario, estaríamos calculando propiedades que evidentemente van a cambiar significativamente conforme avance la simulación.

Antes de hacer la simulación de producción realizaremos dos simulaciones que servirán para equilibrar el sistema, la primera pequeña simulación de equilibrio la realizaremos en un **ensamble NVT**, es decir, vamos a mantener el número de moléculas, el volumen y la temperatura constantes. Esta simulación tiene el objetivo de equilibrar la temperatura del sistema al valor de nuestro interés.

Posteriormente realizaremos la segunda pequeña dinámica para equilibrar la densidad de nuestro sistema, de tal forma haremos la simulación en un **ensamble NPT**, vamos a mantener el número de moléculas la presión y la temperatura constantes. Al no mantener el volumen constante, el tamaño de la caja de nuestro sistema va a variar con el tiempo hasta llegar a un valor de equlibrio.

Los pasos para hacer estas simulaciones de equilibrio son equivalentes a los pasos de la minimización. Primero combinamos el archivo de coordenadas, el de topología y las instrucciones para realizar la dinámica con el comando *grompp* y posteriormente ejecutamos las dinámicas con los comandos *mdrun*, esto para cada equilibrio:

Equilibrio NVT

```
> gmx grompp -f nvt.mdp -c min.gro -p topol.top -o nvt -maxwarn 1
```

```
> gmx mdrun -deffnm nvt -v
```

El archivo *nvt.mdp* contiene los pasos y los parámetros que utilizará gromacs para realizar la dinámica de equilibrio:

```
title                    = Equilibrio NVT

; Parámetros de Simulación
integrator               = md        ; integrador leap-frog
nsteps                   = 50000     ; 2 * 50000 = 100 ps
dt                       = 0.002     ; 2 fs

; Parámetros de frecuencia de escritura de archivos
nstxout                  = 500       ; coordenadas, cada 1.0 ps
nstvout                  = 500       ; velocidades
nstenergy                = 500       ; energías
nstlog                   = 500       ; output

; Parámetros de interacciones de enlace
continuation             = no        ; primera dinámica
constraint_algorithm     = lincs     ; restricción de enlaces
constraints              = h-bonds   ; restricción de vibración H
lincs_iter               = 1         ; presición de LINCS
lincs_order              = 4         ; presición de LINCS

; interacciones de no enlace
cutoff-scheme       = Verlet    ; Corte de las interacciones
ns_type             = grid      ; algoritmo búsqueda de vecinos
nstlist             = 10        ; frecuencia de búsqueda de vecinos
rcoulomb            = 1.0       ; corte de electrostaticas (nm)
rvdw                = 1.0       ; corte de van der Waals (nm)
```

```
DispCorr            = EnerPres  ; algoritmo corte vdW

; Electrostáticas
coulombtype             = PME       ; Particle Mesh Ewald, largo
pme_order               = 4         ; interpolación cúbica
fourierspacing          = 0.16      ; el tamaño del grid

; Termostato
tcoupl                  = V-rescale ; Termostato de Berendsen
tc-grps                 = system    ; en todo el sistema igual
tau_t                   = 0.1       ; cada cuánto escala las Temps
ref_t                   = 300       ; Temperatura de referencia

; Barostato
pcoupl                  = no        ; no mantenemos P cte

; Condiciones periódicas a la frontera
pbc                     = xyz       ; 3-D PBC

; Generación de velocidades iniciales
gen_vel                 = yes       ; distribución de Maxwell
gen_temp                = 300       ; temperatura de la distribution
gen_seed                = -1        ; semilla aleatoria
```

Equilibrio NPT

```
> gmx grompp -f npt.mdp -c nvt.gro -p topol.top -o npt -maxwarn 1
```

```
> gmx mdrun -deffnm npt -v
```

```
title                   = Equilibrio NPT
```

```
; Parámetros de Simulación
integrator              = md        ; integrador leap-frog
nsteps                  = 50000     ; 2 * 50000 = 100 ps
dt                      = 0.002     ; 2 fs

; Parámetros de frecuencia de escritura de archivos
nstxout                 = 500       ; coordenadas, cada 1.0 ps
nstvout                 = 500       ; velocidades
nstenergy               = 500       ; energías
nstlog                  = 500       ; output

; Parámetros de interacciones de enlace
continuation            = yes        ; primera dinámica
constraint_algorithm    = lincs     ; restricción de enlaces
constraints             = h-bonds   ; restricción de vibración H
lincs_iter              = 1         ; presición de LINCS
lincs_order             = 4         ; presición de LINCS

; interacciones de no enlace
cutoff-scheme       = Verlet    ; Corte de las interacciones
ns_type             = grid      ; algoritmo búsqueda de vecinos
nstlist             = 10        ; frecuencia de búsqueda de vecinos
rcoulomb            = 1.0       ; corte de electrostaticas (nm)
rvdw                = 1.0       ; corte de van der Waals (nm)
DispCorr            = EnerPres  ; algoritmo corte vdW

; Electrostáticas
coulombtype             = PME       ; Particle Mesh Ewald, largo
pme_order               = 4         ; interpolación cúbica
fourierspacing          = 0.16      ; el tamaño del grid

; Termostato
tcoupl                  = V-rescale ; Termostato de Berendsen
tc-grps                 = system    ; en todo el sistema igual
tau_t                   = 0.1       ; cada cuánto escala las Temps
ref_t                   = 300       ; Temperatura de referencia
```

```
; Barostato
pcoupl            = Parrinello-Rahman     ; Barostato
pcoupltype        = isotropic             ; en 3 direcciones
tau_p             = 2.0                 ; cada cuánto tiempo
ref_p             = 1.0                 ; presión en bar
compressibility   = 4.5e-5              ; compresibilidad isotérmica
                                        ; del agua, bar^-1

refcoord_scaling  = com                 ; reescalar el centro de masa

; Condiciones periódicas a la frontera
pbc                     = xyz       ; 3-D PBC

; Generación de velocidades iniciales
gen_vel                 = yes       ; las velocidades vienen
                                    ; de la simulación anterior
```

Producción de Simulación Molecular

Una vez que tenemos nuestro sistema en equilibrio podemos considerar la dinámica molecular para calcular propiedades de nuestro sistema. Lo más común es simular un sistema en un ensamble **NPT**. Los parámetros de simulación serán muy parecidos a la pequeá simulación de equilibrio a presión constante. Es importante revisar que no guardemos coordenadas y velocidades muy seguido o generaremos trayectorias enormes.

```
> gmx grompp -f prod.mdp -c npt.gro -p topol.top -o prod -maxwarn 1
```

```
> gmx mdrun -deffnm prod -v
```

```
; Parámetros de Simulación
integrator              = md        ; integrador leap-frog
nsteps                  = 500000     ; 2 * 500000 = 1 ns
```

```
dt                      = 0.002     ; 2 fs

; Parámetros de frecuencia de escritura de archivos
nstxout                 = 5000       ; coordenadas, cada 1.0 ps
nstvout                 = 0       ; velocidades
nstenergy               = 5000       ; energías
nstlog                  = 5000       ; output


; Parámetros de interacciones de enlace
continuation            = yes        ; primera dinámica
constraint_algorithm    = lincs     ; restricción de enlaces
constraints             = h-bonds   ; restricción de vibración H
lincs_iter              = 1         ; presición de LINCS
lincs_order             = 4         ; presición de LINCS

; interacciones de no enlace
cutoff-scheme       = Verlet    ; Corte de las interacciones
ns_type             = grid      ; algoritmo búsqueda de vecinos
nstlist             = 10        ; frecuencia de búsqueda de vecinos
rcoulomb            = 1.0       ; corte de electrostaticas (nm)
rvdw                = 1.0       ; corte de van der Waals (nm)
DispCorr            = EnerPres  ; algoritmo corte vdW

; Electrostáticas
coulombtype             = PME       ; Particle Mesh Ewald, largo
pme_order               = 4         ; interpolación cúbica
fourierspacing          = 0.16      ; el tamaño del grid

; Termostato
tcoupl                  = V-rescale ; Termostato de Berendsen
tc-grps                 = system    ; en todo el sistema igual
tau_t                   = 0.1       ; cada cuánto escala las Temps
ref_t                   = 300       ; Temperatura de referencia

; Barostato
```

```
pcoupl              = Parrinello-Rahman     ; Barostato
pcoupltype          = isotropic             ; en 3 direcciones
tau_p               = 2.0                 ; cada cuánto tiempo
ref_p               = 1.0                 ; presión en bar
compressibility     = 4.5e-5              ; compresibilidad isotérmica
                                          ; del agua, bar^-1

refcoord_scaling    = com                 ; reescalar el centro de masa

; Condiciones periódicas a la frontera
pbc                     = xyz       ; 3-D PBC

; Generación de velocidades iniciales
gen_vel                 = yes       ; las velocidades vienen
                                    ; de la simulación anterior
```

Análisis de una Dinámica Molecular utilizando la utilería de Gromacs

Visualización

Es importante visualizar si la simulación se ve adecuada. Esto lo logramos cargando el archivo de coordenadas (gro) de salida de la simulación junto con el archivo de la trayectoria (xtc) de salida en vmd.

```
> vmd archivo.gro archivo.xtc
```

Debemos revisar varias propiedades al correr una optimización, dinámica de equilibrio:

La energía del sistema

particularmente durante una optimización, debemos revisar si la energía del sistema va disminuyendo y si llegamos a ver una energía que ya no cambie mucho con cada paso. Además, durante los equilibrios, la energía del sistema debe también ir llegando a una meseta. Finalmente, durante la producción de una dinámica molecular, esperamos que la energía del sistema sea constante.

Gromacs tiene su utilería de análisis. El comando **gmx energy** extrae propiedades del sistema que queremos revisar, es un programa interactivo y le pregunta al usuario qué componentes de la energía total o cuál de las propiedades del sistema quiere extraer.

```
> gmx energy -f min.edr -o E_potencial
```

La salida de este comando es una pantalla donde aparece cada propiedad que fue escrita en el archivo *min.edr* y que podríamos estar interesados en graficar.

```
Select the terms you want from the following list by
selecting either (part of) the name or the number or a combination.
End your selection with an empty line or a zero.
-------------------------------------------------------------------
  1  Bond             2  Angle            3  Proper-Dih.      4  LJ-14
  5  Coulomb-14       6  LJ-(SR)           7  Coulomb-(SR)     8  Coul.-recip.
  9  Potential       10  Pressure         11  Vir-XX          12  Vir-XY
 13  Vir-XZ          14  Vir-YX           15  Vir-YY          16  Vir-YZ
 17  Vir-ZX          18  Vir-ZY           19  Vir-ZZ          20  Pres-XX
 21  Pres-XY         22  Pres-XZ          23  Pres-YX         24  Pres-YY
 25  Pres-YZ         26  Pres-ZX          27  Pres-ZY         28  Pres-ZZ
 29  #Surf*SurfTen   30  T-rest
```

Para extraer los datos de la energía potencial a lo largo de la minimización seleccionamos 9 y luego damos dos *enters*.

```
> 9
>
```

El archivo de salida tendrá el nombre *E_potencial.xvg*. Este archivo, tiene un pequeño script para graficar los datos con **xmgrace**. Así que escribimos en la terminal

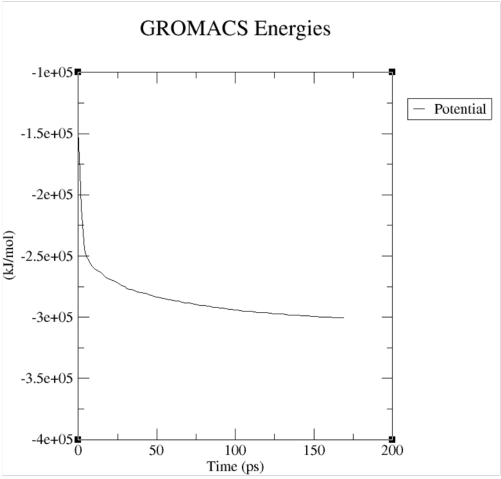
```
> xmgrace E_potencial.xvg
```

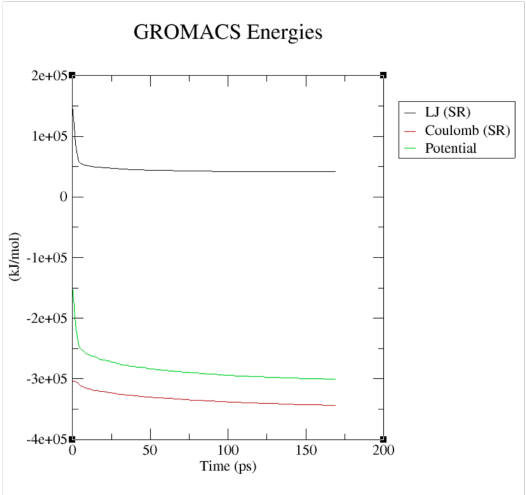
Para extraer datos de la simulación e imprimirlos en columnas consecutivas del mismo archivo, por ejemplo, analizar diferentes contribuciones a la energía potencial o agregar alguna los datos de alguna propiedad de interés, se pueden seleccionar múltiples opciones de la lista:

```
> 9
> 5
> 1
```

Para graficar cada columna independientemente hay que trabajar sobre el archivo en xmgrace.

```
> xmgrace -nxy E_potencial.xvg
```





Índices

Para generar un índice con un conjunto de moléculas se utiliza el comando *make_ ndx*

```
> gmx make_ndx -f min.tpr -o indice.ndx
```

El índice que se genera tendrá el nombre que especificamos en la salida -o, en este caso indice.ndx.

```
Reading file min.tpr, VERSION 2019.1 (single precision)
Reading file min.tpr, VERSION 2019.1 (single precision)
Going to read 0 old index file(s)
Analysing residue names:
There are:    16      Other residues
There are:  6591      Water residues
Analysing residues not classified as Protein/DNA/RNA/Water and splitting into groups...

  0 System              : 20221 atoms
  1 Other               :   448 atoms
  2 LIG                 :   448 atoms
  3 Water               : 19773 atoms
  4 SOL                 : 19773 atoms
  5 non-Water           :   448 atoms

 nr : group      '!': not  'name' nr name   'splitch' nr    Enter: list groups
 'a': atom       '&': and  'del' nr         'splitres' nr   'l': list residues
 't': atom type  '|': or   'keep' nr        'splitat' nr    'h': help
 'r': residue              'res' nr         'chain' char
 "name": group             'case': case sensitive           'q': save and quit
 'ri': residue index

>
```

podríamos seleccionar, por ejemplo, todos los átomos de oxígeno del sistema y generar un grupo adicional de la pequeña lista que se despliega en la terminal:

```
> a 0
```

Análisis de propiedades estructurales de la simulación de dinámica molecular

Después de realizar una dinámica molecular será necesario revisar varias propiedades generales del sistema, por ejemplo, su energía total, potencial, cinética, temperatura, volumen y presión para ver si el sistema que simulamos se encuentra *estable*.

```
> gmx energy -f prod.edr -o archivo.xvg
```

Propiedades estructurales

Una vez que analizamos propiedades generales del sistema, comúnmente estaremos interesados en estudiar propiedades mesoscópicas o estructurales de nuestro sistema. Por ejemplo:

- **gmx hbond** es un comando que cuenta el número de puentes de hidrógeno que existe entre dos selecciones del índice.
- **gmx distance** calcula las distancias entre dos pares de átomos
- **gmx angle** calcula distribuciones de ángulos entre tres puntos seleccionados.
- **gmx density** calcula la densidad parcial a lo largo de la caja de nuestro sistema de algún conjunto de moléculas en particular.
- **gmx rms** (root mean square deviation) de las posiciones atómicas es una medida cuantitativa de similaridad entre dos estructuras. Cuantifica cuánto se movió un sistema.
- **gmx msd** calcula el desplazamiento cuadrático medio de los átomos seleccionados a partir de un sistema de referencia. Utilizando la relación de Einstein, el msd puede calcular las constantes de difusión de un conjunto de moléculas del sistema.
- **gmx coovar y anaeig** realizan un análisis de coovarianza para calcular modos principales en el movimiento del sistema.

- **gmx rdf** calcula la función de distribución radial de un conjunto de moléculas/átomos seleccionados con otros.

El formato para correr este tipo de análisis conserva casi siempre las misma forma:

```
> gmx propiedad -f trayectoria.xtc -n indice.ndx -s sistema.tpr
```

Recuerda que siempre se puede escribir **-h** en la línea de comandos de la terminal para que gmx imprima un recordatorio o ayuda sobre qué archivos requiere cada comando. Además utilizando python podríamos analizar cualquier parámetro que nos interese sobre nuestro sistema utilizando las librerías de **MDAnalysis**. Esta librería contiene toda la información para leer los diferentes archivos de coordenadas, topología o trayectorias que se generan de los programas más comunes de dinámica molecular.

Instalación de VMD

Descargar el archivo que subí a classroom y correr los siguientes pasos

```
> tar xvf vmd-1.9.4a57.bin..... etc .....tar.gz
> cd vmd-1.9.4.a57
> ./configure LINUXAMD64
> cd src
> sudo make install
```

Y listo, con eso queda instalado VMD

Instalación de GROMACS

```
> sudo apt-get install gromacs
```