

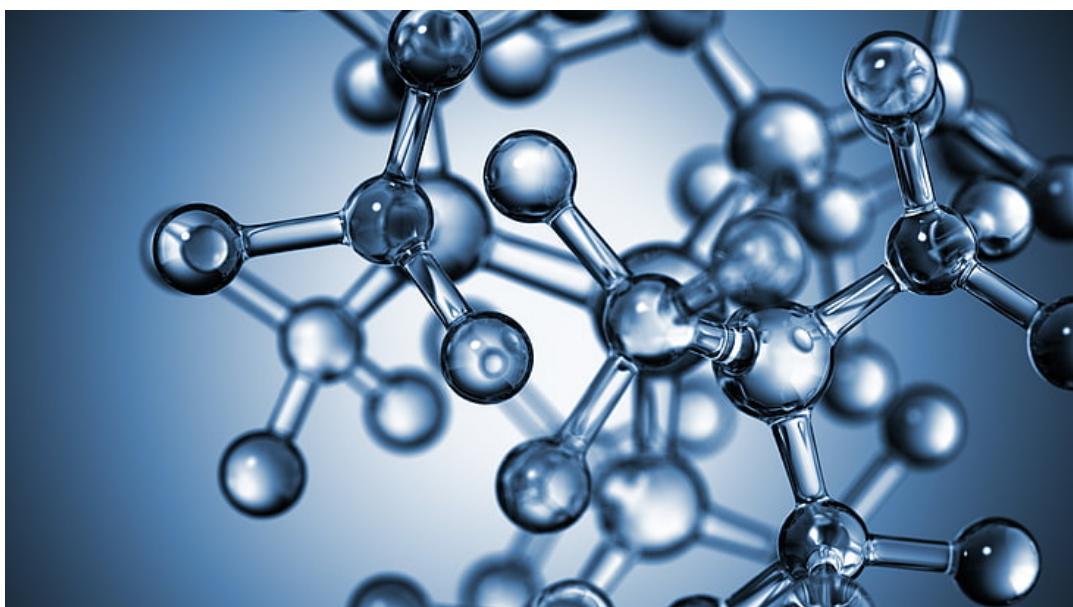


UNIVERSIDAD
DE MURCIA

Entrega Modelado Molecular: Simulaciones Dinámica Molecular

Análisis del sistema tripéptido A-R-Y en agua

Máster en Bioinformática - Facultad de Biología



Francisco Javier López Carbonell

17/03/2024

Índice

1. Resumen general	1
2. Introducción	2
3. Objetivos	5
3.1. Objetivo General	5
3.2. Objetivos Específicos	5
4. Metodología	6
4.1. Construcción del entorno	6
4.2. PyMOL	6
4.3. GROMACS	7
4.3.1. Topología y solvatación del sistema	7
4.3.2. Equilibrado	9
4.3.3. Ejecución	12
4.3.4. Análisis	15
4.4. RStudio: uso del paquete ggplot2	16
5. Resultados y discusión	17
5.1. Simulación básica a distinta temperatura	17
5.1.1. Evolución del radio de giro	17
5.1.2. Evolución de distancias de enlace	18
5.1.3. Evolución de ángulos de enlace	19
5.1.4. Evolución de ángulos dihedros	21
5.1.5. Evolución de temperatura	23
5.1.6. Evolución de velocidades	24
5.1.7. Energía total del sistema	27
5.1.8. Energía cinética del sistema	28
5.2. Simulación extendida de A-R-Y	30
5.2.1. Distribución de temperatura	30
5.2.2. Distribuciones de velocidad	31
5.2.3. Diagrama de Ramachandran	32
5.2.4. Conformaciones de mayor probabilidad	34
6. Conclusiones	35
7. Referencias	36

Índice de figuras

1.	Fórmula general de un aminoácido	2
2.	Fórmula alanina	2
3.	Fórmula arginina	3
4.	Fórmula tirosina	3
5.	Estructura ARY en <i>PyMOL</i>	6
6.	Solvatación ARY en <i>PyMOL</i>	9
7.	Composición fichero equiNVTmdp	10
8.	Fichero equiNVTmdp modificado 400K	10
9.	Sistema neutralizado con átomo cloruro	11
10.	Fichero slurm equilibrado	12
11.	Fichero runNVTmdp modificado	13
12.	Fichero cartoon.pdb en PyMOL	15
13.	Radio de giro del sistema ARY	17
14.	Distancia de enlace NE-CZ-ARG-3	18
15.	Distancia de enlace CB-CZ-TYR-4	19
16.	Ángulo de enlace NE-CZ-NH1-ARG-3	20
17.	Ángulo de enlace CA-CB-CZ-TYR-4	20
18.	Ángulos de torsión phi y psi	21
19.	Ángulo dihedro phi ALA-2	22
20.	Ángulo dihedro psi ALA-2	22
21.	Ángulo dihedro phi TYR-4	22
22.	Ángulo dihedro psi TYR-4	22
23.	Temperatura del sistema ARY	23
24.	Evolución átomo 1	24
25.	Distribución átomo 1	24
26.	Evolución átomo 2	25
27.	Distribución átomo 2	25
28.	Evolución átomo 3	25
29.	Distribución átomo 3	25
30.	Evolución átomo 4	26
31.	Distribución átomo 4	26
32.	Evolución átomo 5	26
33.	Distribución átomo 5	26
34.	Energía total del sistema ARY	27
35.	Energiá total media a 298K	27
36.	Energiá total media a 400K	27
37.	Energiá cinética del sistema ARY	28
38.	E. cinética media a 298K	28
39.	E. cinética media a 400K	28
40.	Distribución temperatura simulación extendida	30
41.	Componente x átomo 1	31

42.	Componente y átomo 1	31
43.	Componente z átomo 1	31
44.	módulo átomo 1	31
45.	Diagrama de Ramachandran teórico	32
46.	Diagrama de Ramachandran ALA-2	33
47.	Diagrama de Ramachandran ARG-3	33
48.	Diagrama de Ramachandran TYR-4	34

1. Resumen general

El presente informe corresponde a la realización de simulaciones de dinámica molecular mediante el software libre *GROMACS* ("GROningen MAchine for Chemical Simulations"). A su vez, contaremos con el apoyo de *PyMOL*, una herramienta complementaria para visualizar y analizar tanto los archivos procedentes de "Protein Data Bank", como los resultados de simulaciones de dinámica molecular realizadas con *GROMACS*.

En concreto, hemos realizado simulaciones de un tripéptido específico, en este caso constituido por Alanina (A), Arginina (R) y Tirosina (Y) en agua a dos temperaturas distintas (298K y 400K).

Para ello hemos realizado una fase de preparación de nuestro sistema. En concreto, hemos generado archivos de topología, diseñado la forma de nuestro sistema (caja cúbica), y solvatado nuestro sistema, además de equilibrar el sistema y neutralizar residuos cargados en condiciones ambientales (positivamente en nuestro caso). A continuación, hemos analizado distintos parámetros de ambas simulaciones que nos ha permitido conocer como nuestro sistema ha evolucionado en el tiempo.

Por último, hemos realizado una simulación más profesional, aumentando el tiempo de equilibrado y ampliando el intervalo de resultado por parte de GROMACS y estudiando diagramas de Ramachandran para encontrar conformaciones de alta probabilidad.

2. Introducción

Los aminoácidos son los monómeros que componen las proteínas. Existen 20 aminoácidos, compartiendo todos ellos una estructura básica común que consiste en: un átomo central de carbono, también llamado carbono alfa, unido a un grupo amino (NH_2), un grupo carboxilo (COOH) y a un átomo de hidrógeno. Cada aminoácido también tiene otro átomo o grupo de átomos unidos al átomo central, conocido como el grupo R, que determina la identidad del aminoácido. Abajo se presenta esta estructura básica de cualquier aminoácido:

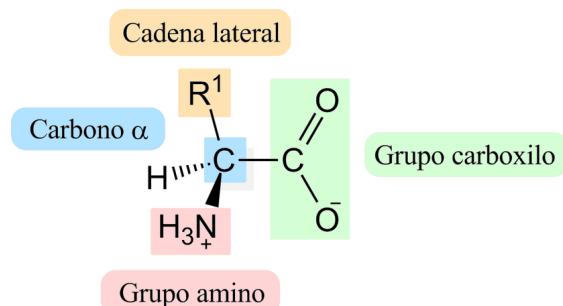


Figura 1: Fórmula general de un aminoácido

El carbono-alfa recibe su nombre debido a su posición adyacente al carbono en el grupo carboxilo dentro de la estructura de un aminoácido. Por otro lado, el componente que distingue los diferentes aminoácidos (R) se conoce como cadena lateral o radical. Estas cadenas laterales pueden variar entre aminoácidos, lo que los hace polares, apolares, con carga eléctrica o neutros.

En esta práctica, se trabajarán con tres aminoácidos específicos: alanina, arginina y tirosina (ARY).

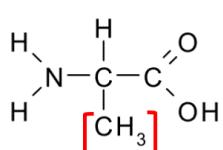
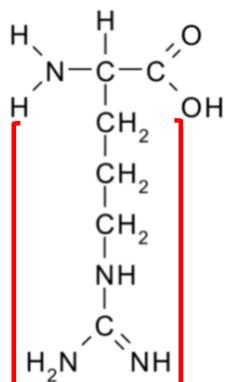


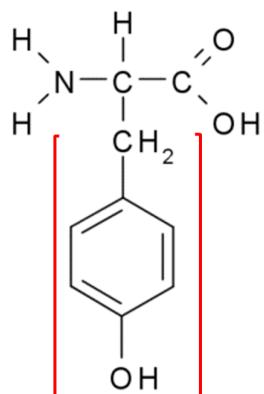
Figura 2: Fórmula alanina

Alanina: se puede observar que la alanina contiene un grupo radical metilo, lo que indica que es uno de los aminoácidos más simples. El carbono-alfa de la alanina está unido a un grupo metilo (-CH₃), lo que la clasifica como un aminoácido no polar o alifático.



Arginina: el átomo de carbono-alfa de la arginina presenta una cadena lateral compuesta por $\text{CH}_2\text{-CH}_2\text{-CH}_2\text{-NH-C-NH-NH}_2$, lo cual la sitúa dentro de los aminoácidos con carga positiva y propiedades polares.

Figura 3: Fórmula arginina



Tirosina: el átomo de carbono-alfa de la tirosina está enlazado a una cadena lateral que contiene un anillo bencénico con un grupo hidroxilo (-OH), lo que la clasifica como un aminoácido polar con una cadena lateral aromática.

Figura 4: Fórmula tirosina

Los aminoácidos **neutros** son aquellos cuya cadena lateral no contiene grupos carboxilo ni amino, lo que resulta en una carga eléctrica neta de 0 a pH neutro. Esta categoría se subdivide en polares y apolares.

Los aminoácidos **polares** presentan grupos hidrófilos en sus cadenas laterales, lo que les permite formar puentes de hidrógeno con moléculas polares y los hace altamente solubles en agua. La tirosina es un ejemplo dentro de este grupo.

Por otro lado, los aminoácidos **apolares** tienen cadenas laterales hidrófobas, lo que los lleva a preferir ubicarse en el interior de las proteínas globulares cuando están en solución acuosa. La alanina pertenece a este grupo.

En contraste, la arginina es un aminoácido **básico** que contiene un grupo amino (guanidino en este caso) en su cadena lateral. Debido a su naturaleza básica, este grupo puede captar protones del medio, lo que resulta en una carga positiva para el aminoácido.

El trabajo realizado consta de un conjunto de simulaciones de dinámica molecular sobre el modelo de este tripéptido, con el objetivo de analizar los resultados a nivel de distintos parámetros. Este tripéptido (ARY) se encuentra en la carpeta llamada "tripptides", ubicada en el directorio "for-students", en el directorio "personal-report-MM".

El tripéptido considerado tiene los extremos bloqueados a través de un grupo acetilo y un grupo N-metilamina para evitar uniones peptídicas indeseables. En caso contrario, quedarían un grupo carboxilo y un grupo amino sueltos que pueden elazarse con otra molécula y dimerizarse.

Esta molécula viene dada en formato .pdb ("Protein Data Bank"), incluyendo las coordenadas de cada átomo en un espacio tridimensional. Para visualizarla a largo del trabajo, se utilizará *PyMOL* versión 2.5.0a0, que es un software especializado en la representación y manejo de moléculas. Más adelante en la sección 4.2 del informe se detallará su uso más concretamente.

El resto del estudio se llevará a cabo con *GROMACS* 2016.4, un paquete de funciones diseñado para realizar simulaciones y cálculos de dinámica molecular. Se usará para construir el sistema, equilibrarlo, hacer las simulaciones y los análisis finales de datos.

Estos dos programas están implementados en el servidor ("dayhoff.inf.um.es") usado para llevar a cabo la realización de este trabajo junto con *ggplot2* de RStudio, que será usado para mostrar los gráficos extraídos de *GROMACS*.

3. Objetivos

3.1. Objetivo General

Realizar un estudio simulado de dinámica molecular de acuerdo a las herramientas propuestas en clase de modelado molecular para un tripéptido específico.

3.2. Objetivos Específicos

- Analizar la evolución de simulaciones del tripéptido A-R-Y a 298 K y a 400 K de temperatura.
- Estudiar la energía cinética del sistema a 298 K y a 400 K de temperatura.
- Analizar la evolución de temperatura y velocidad de una simulación más eficiente, exportando resultados con un intervalo de tiempo más amplio.
- Interpretar gráficos ramachandran y e identificar conformaciones más probables.

4. Metodología

Los resultados del presente informe han sido obtenidos por medio de varios software relacionados con simulación de sistemas moleculares dinámicos y visualización de propiedades específicas procedentes de dicha simulación.

4.1. Construcción del entorno

El primer paso será crear el directorio "1-build", en el subdirectorio creado previamente denominado "personal-report-MM". En el directorio "home/personal-report-MM/for-students" se encuentra el tripéptido asignado en formato ".pdb" (ary.pdb), que copiamos en "1-build", usando el comando cp de bash. El resto de directorios que contendrán los ficheros generados en el estudio se encontrarán en "personal-report-MM".

4.2. PyMOL

PyMOL es una herramienta versátil y poderosa para visualizar y analizar estructuras moleculares en 3D, muy útil en campos como la biología estructural, la bioquímica, la química computacional y la farmacología. En nuestro caso, será de gran ayuda para visualizar archivos procedentes de la base de datos de estructuras proteicas *PDB* y archivos de salida procedentes de la ejecución de comandos disponibles en *GROMACS*. [1]

Una vez rescatado el archivo ary.pdb, con la estructura de nuestro tripéptido en el directorio "1-build", procedemos a ojear la estructura y familiarizarnos con las posibles opciones de visualización que nos aporta *Pymol*. A continuación, se muestra el tripéptido en *PyMOL*:

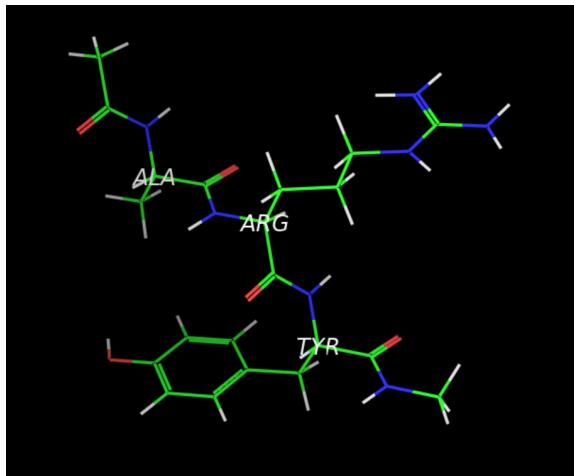


Figura 5: Estructura ARY en *PyMOL*

Esta estructura es el resultado de la unión de estos tres aminoácidos (ARY) mediante un enlace peptídico. Este enlace peptídico une dos aminoácidos mediante la pérdida de una molécula de agua entre el grupo amino de un aminoácido y el grupo carboxilo del otro. El resultado es un enlace covalente CO-NH.

Introduciendo el comando ”less ary.pdb” tendremos acceso a diversos parámetros del archivo creado. Por ejemplo, el número de átomo, que clase de átomo es, el código de proteínas con 3 letras que corresponde, el número de residuo del extremo N terminal al C terminal, las coordenadas en Amstrongs, entre otros.

Volveremos hacer uso de *Pymol* más adelante para visualizar nuestro tripéptido en agua solvatado y neutralizado, detallado en posteriores secciones del informe.

4.3. GROMACS

GROMACS es un software de simulación de dinámica molecular (MD) ampliamente utilizado en el campo de la biofísica, la química computacional y la biología molecular. Su nombre proviene de ”GROningen MAchine for Chemical Simulations”. Esta será la herramienta principal que usaremos en nuestras simulaciones de dinámica molecular. [2]

En nuestro caso, vamos a usar la versión instalada en el cluster ”dayhoff.inf.um.es”, que podemos explorar mediante el comando **gmx -version**, con el que podemos ver que tenemos a nuestra disposición la versión 2016.4. [3]

4.3.1. Topología y solvatación del sistema

El paso inicial de toda simulación será crear la topología de nuestro sistema. En concreto, debemos cambiar el archivo en formato .pdb a un formato legible por GROMACS. Para esto utilizamos el programa **pdb2gmx** que creará los archivos de topología de la molécula.

La topología incluye la información sobre las uniones, campos de fuerza y otros parámetros de los átomos. Es decir, se convierte el archivo *Protein Data Bank* a los formatos que necesita GROMACS para funcionar. Una diferencia importante entre un .pdb y un .gro son las unidades. El .pdb está en Angstroms y el .gro en nm. Dicho esto usamos el siguiente comando:

```
gmx pdb2gmx -f ary.pdb -o ary.gro -p ary.top -ter
```

Una vez introducido este comando nos indicará que elijamos entre varias opciones en 4 ocasiones:

Primeramente que campo de fuerzas vamos a emplear (*Select the Force Field:*). Marcamos el número 8:

```
8: CHARMM27 all-atom force field (CHARM22 plus CMAP for proteins)
```

A continuación, nos indica que escojamos el modelo de agua (*Select the Water Model:*). Marcamos el número 1.

```
1: TIP3P TIP 3-point, recommended
```

Por otro lado, nos indicara que elijamos el término de inicio para ACE-1 (*Select start terminus type for ACE-1:*), En este caso escogemos el número 2 (**2: None**).

Por ultimo, nos indica que seleccionemos también el termino de fin de CT3-7 (*Select end terminus type for CT3-7:*), Escogemos el número 4 (**4: None**).

En este momento, obtenemos un fichero con extensión .gro (contiene las coordenadas iniciales) y un archivo con extensión .top (topología de las moléculas). Para ver más sobre ellos podríamos usar de nuevo el comando less. El siguiente paso es introducir:

```
gmx editconf -f ary.gro -o ary-box.gro -bt cubic -box 3.0 3.0 3.0
```

El resultado es una celda unitaria cubica en la que se encuentra nuestro tripéptido para hacer las simulaciones. El 3 hace referencia a los nanómetros en el eje x, y, z. Este tripéptido estaría simulado en vacío, para hacer la simulación más realista introduciremos a esa caja de cómputo moléculas de agua para que el sistema este hidratado. Para ello usamos:

```
gmx solvate -cp ary-box.gro -cs -o ary-box-solv.gro -p ary.top
```

Con esto conseguimos la solvatación de la molécula, es decir, la formación de interacciones entre las moléculas de un disolvente (agua) con el tripéptido. En otras palabras, este comando rellena el cubo creado anteriormente de agua, a excepción de aquellos espacios ocupados por el tripéptido. A continuación introducimos:

```
gmx editconf -f ary-box-solv.gro -o ary-box-solv.pdb
```

Este comando convierte el formato de estructura a .pdb. Para visualizar este .pdb volvemos a utilizar el visualizador de proteínas PyMOL, con el siguiente comando:

```
pymol ary-box-solv.pdb
```

Para visualizar la imagen de esta manera, introducimos en la línea de comandos de PyMOL "as lines" para verlo en formato línea tanto el tripéptido como la caja.

Sin embargo, al tratarse de una imagen estática en lugar de una vista dinámica en 3D, la forma del cubo puede resultar más difícil de percibir. Aún así, podemos notar cómo las moléculas de agua forman una especie de envoltura cúbica alrededor del tripéptido. Las caras y aristas de este cubo no son claramente definidas debido a que las moléculas de agua ocupan el espacio de manera irregular.

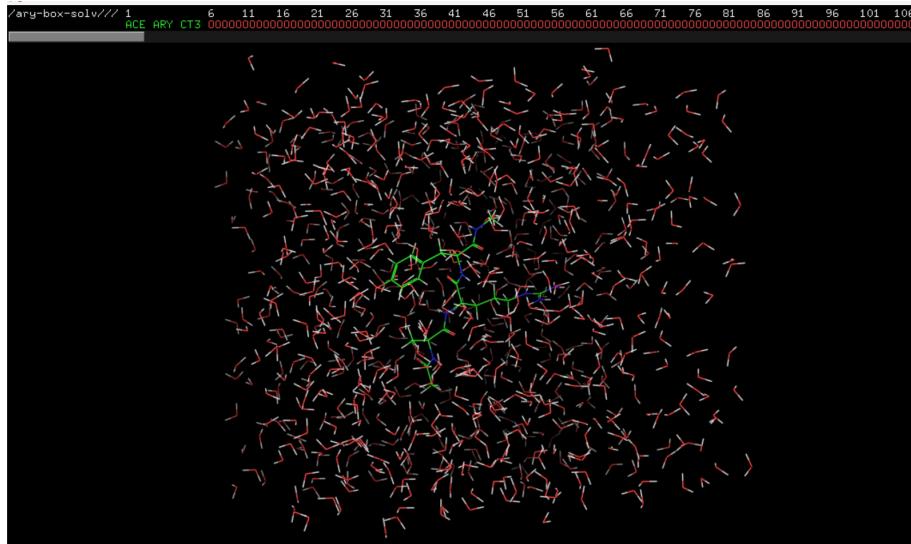


Figura 6: Solvatación ARY en *PyMOL*

4.3.2. Equilibrado

Llegados a este punto, procedemos a equilibrar nuestro sistema, con el objetivo de que se encuentre en una situación lo más realista posible. Se desarrollará para dos temperaturas diferentes, una simulación a 298 K y otra a 400 K.

Para ello, crearemos dos nuevas carpetas dentro de nuestro directorio de trabajo, llamadas 2.1-equilibration (298K) y 2.2-equilibration (400K). En estas carpetas se encuentran todos los ficheros generados por GROMACS durante la fase de equilibrado.

Para llevar a cabo esta fase, necesitaremos los archivos **ary-box-solv.gro**, **ary.top** que copiaremos de la carpeta 1-build, y **equiNVT.md** (procedente de la carpeta for-students). Este último archivo, contiene los parámetros para que GROMACS realice los cálculos del equilibrado como puede ser el paso del tiempo, las velocidades o temperaturas iniciales, entre otras. Además, este archivo deberá ser modificado, cambiando el número de steps (nsteps) de 40000 a 800000 de la siguiente manera:

Resulta importante destacar que, el fichero **equiNVT.md** para la temperatura 400K, será el mismo solo que cambiando los campos `ref_tz` "gen_temp" del fichero cambiando el valor 298 por 400, como se ve en la figura 8.

```
; equiNVTmdp

; RUN CONTROL PARAMETERS
integrator      = md          ; leap-frog integrator
dt              = 0.0005    ; 0.5 fs
nsteps         = 800000    ; 400 ps
define          = -DFLEXIBLE -DCHARMM_TIP3P
constraints     = none
comm-mode       = Linear     ; mode for center of mass motion removal
nstcomm        = 4000      ; number of steps for center of mass motion removal

; OUTPUT CONTROL OPTIONS
nstxout        = 4000      ; Save coordinates every 2.0 ps
nstvout        = 4000      ; Save velocities every 2.0 ps
nstfout        = 0          ; Save forces
nstlog         = 4000      ; Update log every 2.0 ps
nstenergy      = 4000      ; Save energies every 2.0 ps
nstcalcenergy  = 4000

; NEIGHBORSEARCHING PARAMETERS
nstlist         = 10
ns-type         = Grid
pbc             = xyz
rlist           = 1.1
cutoff-scheme  = Verlet
```

Figura 7: Composición fichero equiNVT.mdp

```
; Temperature coupling
Tcoupl        = v-rescale
;ld_seed       = 87654321
tc-grps       = System
tau_t         = 0.1
ref_t         = 400           ; Above fluid phase transition

; GENERATE VELOCITIES FOR STARTUP RUN
gen_vel        = yes        ; Should be the first equilibration
gen_temp       = 400.0      ; Temperature to generate corresponding Maxwell distribution
gen_seed       = 12345678   ; Random seed
```

Figura 8: Fichero equiNVT.mdp modificado 400K

Una vez editados ambos archivos, procedemos a unificar toda la información en un archivo .tpr, el cual es necesario para la ejecución del equilibrado en GROMACS en las simulación a ambas temperaturas. Lo haremos con el siguiente comando:

```
gmx grompp -f equiNVT.mdp -c ary-box-solv.gro -p ary.top -o ary-a.tpr
```

Se debe tener en cuenta que, GROMACS también realiza algunos cálculos previos y puede arrojar anotaciones. En este caso nos avisa de que la molécula no es neutra. Esto sucede porque algunos reisiduos en condiciones ambientales están cargados positiva o negativamente, lo cual no es nada positivo para nuestro sistema en dinamica molecular y debemos neutralizar las cargas para evitar la repulsion entre nuestros atomos. La carga que nos sale es positiva (+1), que nos viene especificada de la siguiente manera: *System has non-zero total charge: 1.000000*

Por tanto, para equilibrar la molécula debemos introducir un número de iones negativos igual a 1 para que la carga global sea 0. La naturaleza soluciona el problema de la carga de los residuos usando cloruros (neutraliza carga positiva) y sodios

(neutraliza carga negativa). Esto lo hacemos con el siguiente comando y para ambas temperaturas:

```
gmx genion -s ary-a.tpr -p ary.top -o ary.gro -nn 1
```

Introducimos un ión negativo con `-nn 1`. Una vez introducimos el comando aparece: *Select group:* y seleccionaremos: "SOL"

Lo que sucede es que el sistema, ahora incluye un átomo de Cl- (cloruro) en la parte del disolvente (agua) y por tanto, el sistema ahora sea neutro (sin carga).

El archivo `ary.gro` contiene ahora un ion de Cl y a modo de recordatorio, la creación del archivo `.tpr` debe realizarse dos veces como ya hemos comentado para ambas temperaturas, cada uno en su respectiva carpeta de equilibrado.

```
gmx grompp -f equiNVTmdp -c ary.gro -p ary.top -o ary.tpr
```

El archivo `.gro` puede convertirse a `.pdb` con el objetivo de observar el ión en el disolvente, aunque este ión será pequeño y difícil de observar (bolita verde ubicada a la izquierda del tripéptido en la figura 9). Para ello utilizamos el siguiente comando y posteriormente volvemos a usar *PyMOL*:

```
gmx editconf -f ary.gro -o ary.pdb
```

```
pymol ary.pdb
```

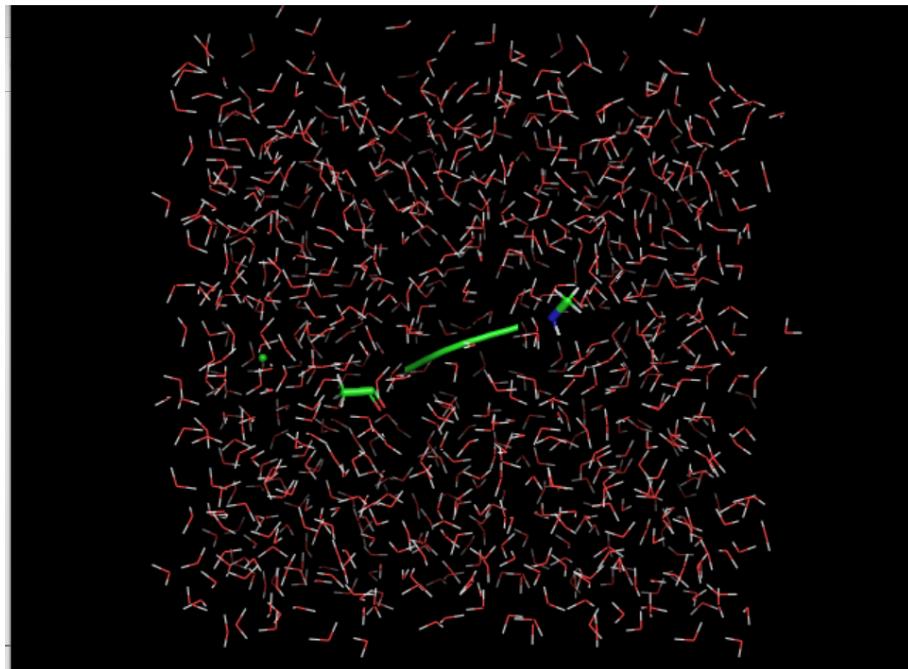


Figura 9: Sistema neutralizado con átomo cloruro

El siguiente paso consiste en la ejecución del equilibrado del sistema, tanto para la versión a 298 K como a 400 K. Este cálculo al ser muy costoso potencialmente se realiza en el cluster "dayhoff.inf.um.es" con el gestor de colas "Slurm". Para lanzar los dos trabajos de equilibrado al cluster es necesario el fichero *submit_eck.sh*. Lo copiamos desde el directorio ".equilibration-mdp" que se encuentra dentro de *for-students* hacia las carpetas 2.1-equilibration y 2.2-equilibration (indicadas por el punto final en diferentes ocasiones). Posteriormente, se modifica en ambas para proporcionar el nombre correcto del archivo a usar en la ejecución.

```
cp ./practica-MM/for-students/equilibration-mdp/submit\_eck.sh .
```

Destacar, que el fichero será el mismo en ambos casos excepto que, se modificará el directorio de salida.

```
GNU nano 2.9.8                                         submit_eck.sh

#!/bin/bash
#
#SBATCH -p eck-q
#SBATCH --chdir=/home/alumno15/personal-report-MM/2.1-equilibration
#SBATCH -J equilibrado
#SBATCH --cpus-per-task=1

date
gmx mdrun -deffnm ary -c ary.g96 -nt 1
date
```

Figura 10: Fichero slurm equilibrado

Para lanzar ambos trabajos debemos poner el siguiente comando situados en ambos directorios:

```
sbatch submit\_eck.sh
```

Cada equilibrado ha tardado aproximadamente unos 30 minutos. Entre los archivos de salida podemos destacar, los .g96 para ambas temperaturas: ary.g96 en 2.1-equilibration y ary.g96. en 2.2-equilibration. Estos incluyen información de las coordenadas finales y las velocidades de los sistemas. Los archivos .log y .out no muestran ningún error en la ejecución, por lo que damos por finalizada la fase de equilibrado y con ella completamos la etapa de preparación de nuestro sistema.

4.3.3. Ejecución

En este apartado se realizarán los pasos enfocados a la ejecución de la simulación del movimiento a nivel atómico del tripéptido estudiado para ambas temperaturas. Al igual que en la etapa anterior, creamos un nuevo directorio dentro del subdirectorio personal-report-MM.

En este caso creamos los directorios 3.1-run y 3.2-run. Al igual que anteriormente, en el 3.1-run trabajaremos con los datos a 298 K y en el 3.2-run con los datos

a 400K. La secuencia de comandos será la siguiente para ambos procedimientos cambiando los directorios correspondientes a 3.1-run y 2.1-equilibration:

```
cd ..
mkdir 3.1-run
cd 3.1.run
cp /home/alumno15/personal-report-MM/2.1-equilibration/ary.top .
cp /home/alumno15/personal-report-MM/2.1-equilibration/ary.g96 .
cp /home/alumno15/personal-report-MM/for-students/run-mdp/runNVTmdp
```

Como se puede apreciar en el último paso, hemos exportado el archivo runNVTmdp. Este archivo procedente de la carpeta for-students contiene los detalles de la ejecución. [4] Este archivo se modifica únicamente en la carpeta 3.2-run para poner ref_t a 400 K de la siguiente manera:

```
; Temperature coupling
;Tcoupl          = Berendsen      ; Modification Berendsen (Bussi): real NVT
Tcoupl          = v-rescale
ld_seed         = 87654321
tc-grps         = System
tau_t           = 0.1
ref_t           = 400            ; Above fluid phase trasition
```

Figura 11: Fichero runNVTmdp modificado

Una vez que tenemos este archivo modificado en esta carpeta, creamos el archivo en extensión .tpr para ambas temperaturas introduciendo el siguiente comando en las carpetas 3.1-run y 3.2-run.

```
gmx grompp -f runNVTmdp -c ary.g96 -p ary.top -o ary.tpr
```

Para realizar la ejecución en el servidor, necesitamos de nuevo el script anterior para enviarlo al gestor de colas. Podemos realizarlo de la misma forma que anteriormente:

```
cp /home/alumno15/personal-report-MM/for-students/run-mdp/submit\_eck.sh .
```

Como hemos realizado anteriormente, este último fichero debemos lanzarlo dos veces correspondiente a las dos temperaturas de estudio y tendrá el mismo contenido a excepción del directorio de salida, que cambiará para ambos casos.

Además, respecto al fichero usado en la fase de equilibrado, el contenido de la carpeta desde la que se lanza es distinto.

A modo de resumen, en este momento se encuentra en cada una de las dos carpetas un archivo de topología (ary.top), un archivo .mdp (298 K en 3.1-run y 400 K en 3.2-run), un archivo .g96 con los que se ha creado un cada caso un archivo .tpr y que han sido llamados con un script submit_eck.sh cada uno. Ahora, ambos trabajos son lanzados de nuevo para su ejecución con el mismo comando utilizado anteriormente:

```
sbatch submit\submit_eck.sh
```

La ejecución de los trabajos en el cluster tarda mucho menos que para la fase de equilibrado, ya que tienen muchos menos pasos (no se ha modificado el número de steps). En cada carpeta se producen un archivo .out, un archivo .log, un archivo .trr (datos de coordenadas), un archivo .edr (datos de energía) y un archivo .g96 que sustituirá al original de cada carpeta.

El paquete traj permite extraer información como velocidades y coordenadas. En este caso se utiliza para crear archivos en formato .pdb que pueden visualizarse en PyMOL de manera dinámica, observando cómo afecta el paso del tiempo.

En este caso creamos cuatro .pdb diferentes, dos a 298 K y otros 2 a 400 K. Se crean dos con cada temperatura ya que queremos visualizar el sistema con y sin moléculas de agua, aunque solo vamos a mostrar una de ellas. En cada una de las dos carpetas run introducimos el siguiente comando:

```
gmx traj -f ary.trr -s ary.tpr -oxt cartoon.pdb
```

A continuación, debemos elegir entre varias opciones (Select:) 0: System, 1: Protein, etc Al elegir *system*, esta opción nos produce el sistema completo con los datos a 298 K en la carpeta 3-run y a 400 K en la carpeta 3.1-run y será la que mostramos en la siguiente imagen.

Destacar que si seleccionáramos la opción *protein*, se producirá un archivo con el tripéptido sin moléculas de agua, es decir, lo que ocurre es que no se incluyen estas moléculas al crear el archivo cartoon.pdb para que se pueda visualizar mejor el tripéptido en PyMOL.

En la figura 12, se distinguen varias líneas que atraviesan el interior del cubo. Cuando se realizan simulaciones de dinámica molecular en sistemas periódicos, como en un cubo que se repite infinitamente en las tres dimensiones, las moléculas pueden atravesar los límites de la caja periódica y aparecer en el lado opuesto de la caja. Esto se debe a que las moléculas interactúan con sus imágenes reflejadas en las cajas vecinas; lo cual indica que las condiciones periódicas de control funcionan.

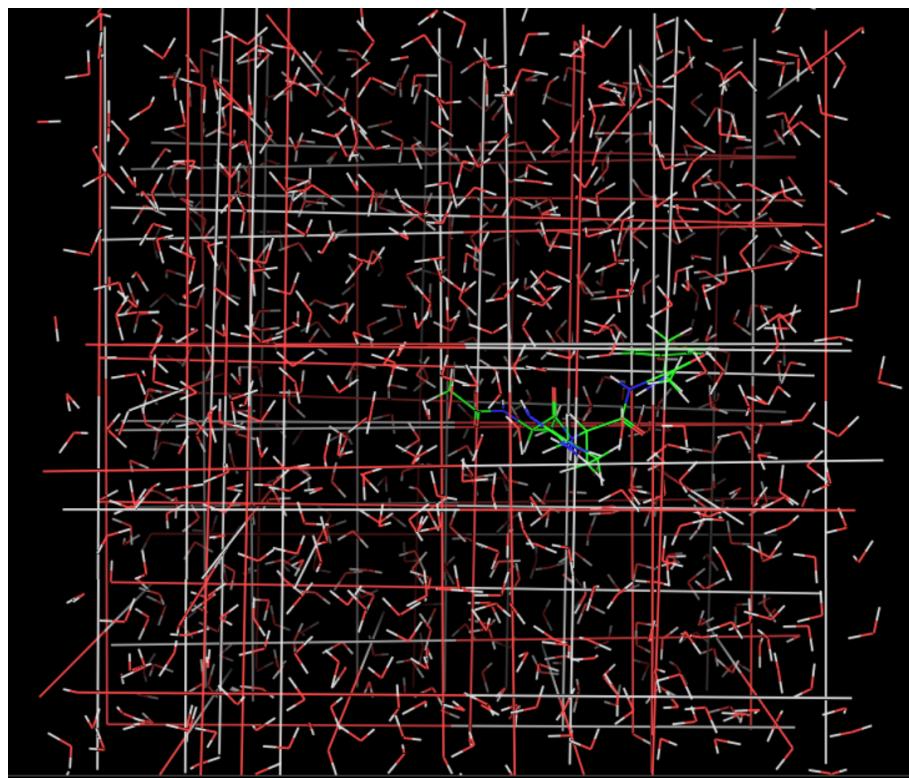


Figura 12: Fichero cartoon.pdb en PyMOL

4.3.4. Análisis

En este último apartado de esta sección, vamos a preparar el directorio para el análisis de los datos que se han ido obteniendo en las distintas simulaciones anteriores y que se llevará a cabo en la próxima sección de resultados y discusión. Ahí, se estudiará la evolución de la energía y temperatura del sistema, así como algunas distancias de enlace, ángulos y ángulos diedros para este tripéptido comparando siempre los resultados para ambas temperaturas (298 K y 400 K).

Para ello, antes se crearán dos directorios nuevos en el directorio de trabajo personal-report-MM, que corresponderán a la simulación de 298 K (llamado 4.1-analysis) y otro directorio que corresponderá a la simulación de 400 K (llamado 4.2-analysis).

A estos dos nuevos directorios creados se les introducen 4 archivos a cada uno procedentes de los directorios anteriores creados. Dependiendo de si es para simular a una temperatura u otra, se obtendrán de 3.1-run (298 K) o de 3.2-run (400 K). Para ello en la carpeta 4.1-analysis y 4.2-analysis, se realizan los siguientes comandos:

```
cp /home/alumno15/personal-report-MM/3.1-run/ary.tpr .
cp /home/alumno15/personal-report-MM/3.1-run/ary.g96 .
cp /home/alumno15/personal-report-MM/3.1-run/ary.trr .
```

```
cp /home/alumno15/personal-report-MM/3.1-run/ary.edr .
```

Será de la misma forma para 4.2-analysis, cambiando 3.1-run por 3.2-run como hemos indicado anteriormente.

4.4. RStudio: uso del paquete *ggplot2*

Los gráficos correspondientes a la simulación básica han sido desarrollados en *RStudio*. Todos ellos han sido editados de la misma manera, por medio del paquete *ggplot2*, representandolo en tipo "lines" o "histogram", ajustando el texto de los dos ejes y el título. Además, hemos representado en dos colores: azul para las representaciones a 298K y rojo a 400K. Además, con este método realizamos la elaboración de las gráficas de tipo Ramachandran de la simulación extendida. En concreto, se han desarrollado dos scripts en *RStudio*, que será aportado al directorio de trabajo.

Las funciones contenidas en dicho script, básicamente generan gráficas mediante la carga de archivos de salida tras la ejecución de los comandos de GROMACS, comentados en la próxima sección del informe. El script podrá ser encontrado en:

```
/home/alumno15/personal-report-MM/basic-plots/script-gráficos-basic-
report.Rmd
```

El script de generación de gráficos ramachandran, consiste en partir de los archivos con los datos de **ángulos phi y psi** en formato **.dat**, seleccionar la edición del gráfico con diversas herramientas de *ggplot2* y finalmente usar la función **create_ramachandran_plot**, para crear el gráfico estilo Ramachandran. En concreto, el script podrá ser encontrado en:

```
/home/alumno15/personal-report-MM/extended-report/rama-plots/extended-
-report-plots.Rmd
```

Los resultados de las gráficas serán discutidos e interpretados en la sección 5.2 del presente informe.

5. Resultados y discusión

5.1. Simulación básica a distinta temperatura

En esta sección del informe, procedemos a comentar e interpretar los resultados del análisis de nuestra simulación. Comentaremos brevemente los comandos utilizados y los resultados representados gráficamente.

5.1.1. Evolución del radio de giro

El radio de giro aporta información sobre la compactación de la molécula. De una forma más técnica, se trata de la media de las distancias de los átomos al centro de masas (masa de cada atomo por su vector de posición más el siguiente y así sucesivamente), ponderado por la masa atomica. En este caso, el comando utilizado es el siguiente:

```
gmx gyrate -f ary.trr -s ary.tpr -xvg none
```

Nos pedirá seleccionar entre varias opciones y seleccionaremos la proteína, ya que no nos interesan el radio de giro de las moléculas de agua incluidas en la opción system.

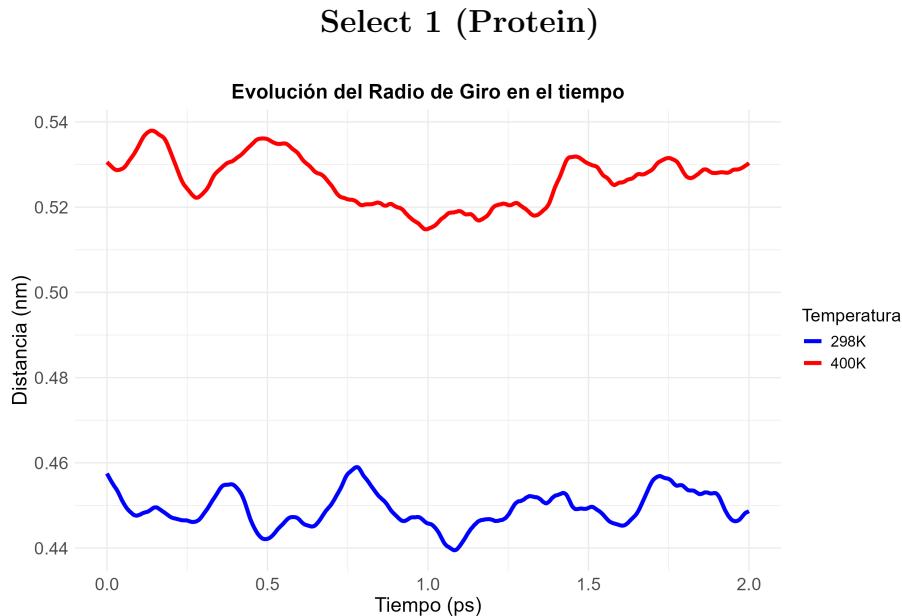


Figura 13: Radio de giro del sistema ARY

En la figura 13, tenemos representado la evolución del radio de giro de nuestro tripéptido a lo largo del tiempo, donde vemos que, va oscilando porque la molécula va interactuando con el disolvente y va cambiando su radio de giro.

Además, vemos un **radio de giro mayor al aumentar la temperatura del sistema**, es decir, a 400K la estructura se encuentra más relajada y extendida linealmente, mientras que a una temperatura menor (298K) nuestra molécula se encuentra más compacta.

Esto es debido a que, a temperaturas más altas, las vibraciones térmicas aumentan, lo que puede hacer que las estructuras proteicas sean más flexibles. Por el contrario, a temperaturas más bajas, las proteínas tienden a ser más estables y tienen menos movimiento térmico, disminuyendo así su radio de giro.

5.1.2. Evolución de distancias de enlace

El próximo paso es analizar los resultados procedentes de dos distancias de enlace seleccionadas. Para seleccionar distancias y ángulos (apartado 5.1.3) haremos uso del archivo *distances.ndx* donde aparecerá una etiqueta que informa del enlace y residuo seleccionados y sus posiciones en el fichero .g96. El comando usado en este caso es:

```
gmx distance -f ary.trr -s ary.tpr -n distances.ndx -oall -xvg none
```

Una de las distancias de enlace seleccionadas, corresponden al enlace producido entre el carbono zeta (CZ, 30) y el nitrógeno épsilon (NE, 32) del aminoácido arginina (posición 0 en el fichero *distances.ndx*). Se trata de un enlace covalente simple de cadena lateral del aa arginina.

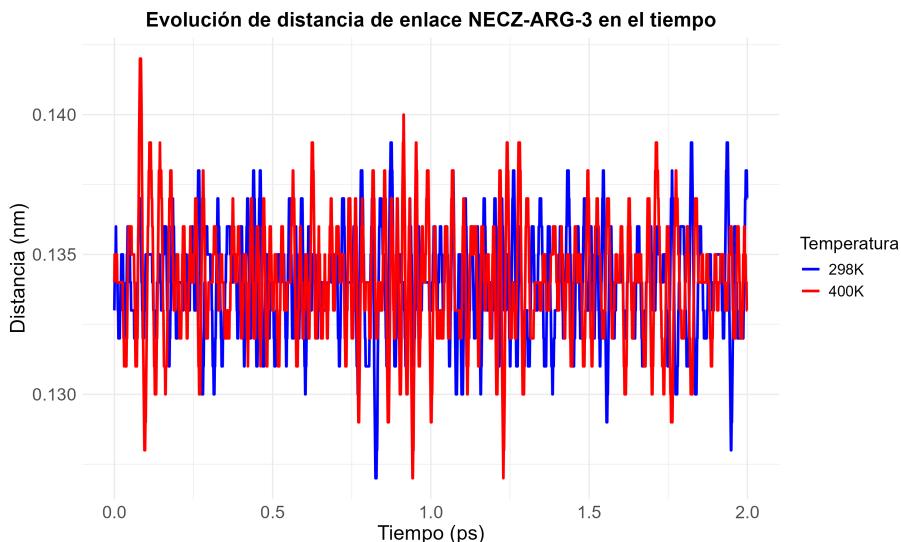


Figura 14: Distancia de enlace NE-CZ-ARG-3

Por otro lado, escogemos el enlace producido entre los carbonos beta (CB, 45) y gamma (CG, 48) del aminoácido tirosina (posición 1 en el fichero *distances.ndx*). Enlace producido entre el primer carbono de la cadena lateral y el primer carbono del grupo fenólico de la cadena lateral.

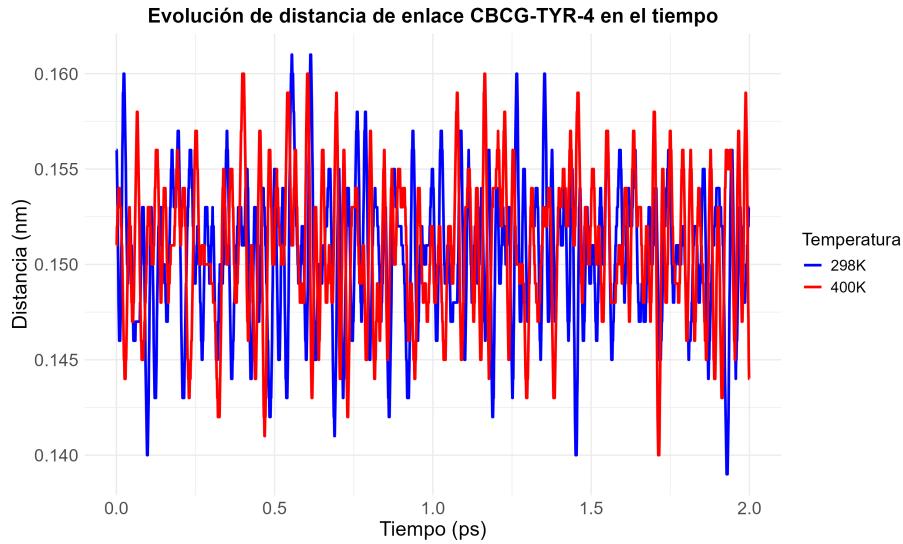


Figura 15: Distancia de enlace CB-CZ-TYR-4

En las figuras 14 y 15, aparecen representadas las distancias de enlace a dos temperaturas (298K y 400K), donde vemos que, prácticamente no hay influencia de temperatura mostrando una evolución muy parecida (distancias medias de los enlaces en torno a 0.135 y 0.150, respectivamente). En ambos casos, la unión es representada mediante un oscilador armónico, pero no es perfecto (la periodicidad parece adulterada en ciertos instantes), debio a que cuando oscilan, puede interactuar con una molécula de agua y rompe la representación armónica.

5.1.3. Evolución de ángulos de enlace

A continuación, analizamos los resultados procedentes de dos ángulos de enlace seleccionados. Estos ángulos están especificados en el archivo *distances.ndx* de manera muy similar al ejemplo de distancias. El comando usado en este caso es:

```
gmx angle -f ary.trr -n distances.ndx -ov -xvg none
```

Los ángulos de enlace seleccionados, corresponden al enlace producido entre el carbono zeta (CZ, 30), el nitrógeno epsilon (NE, 32) y el primer nitrógeno zeta (NH1, 33) del aminoácido arginina (posición 2 en el fichero *distances.ndx*). Por otro lado, escogemos el enlace producido entre los carbonos alfa (CA, 43), beta (CB, 45) y gamma (CG, 48) del aminoácido tirosina (posición 3 en el fichero *distances.ndx*).

En las figuras 16 y 17, podemos ver de nuevo representación correspondiente a oscilador armónico, aunque los angulos se pueden flexionar con mas facilidad que las distancias entre átomos, es decir, muelle menos fuerte que los que unen atomos. Además, vemos en ambos casos que, no hay prácticamente diferencias entre las dos temperaturas.

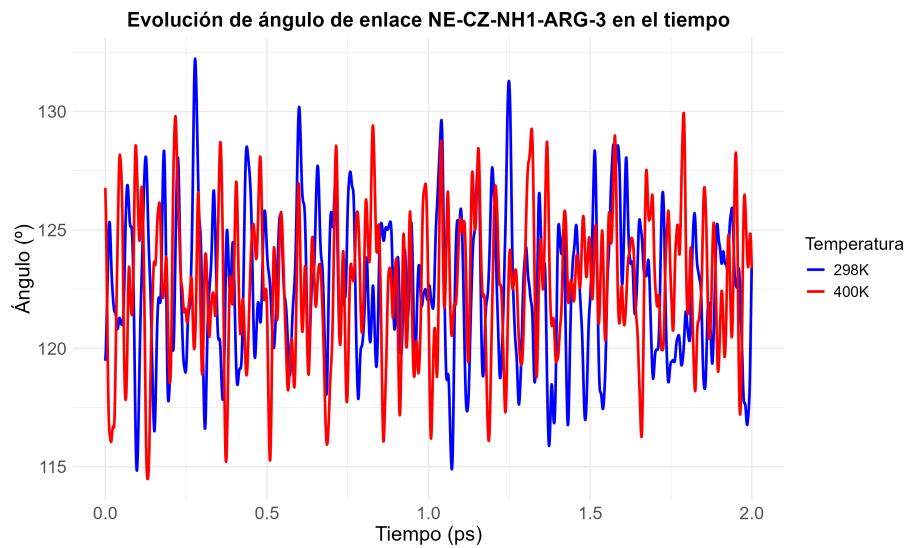


Figura 16: Ángulo de enlace NE-CZ-NH1-ARG-3

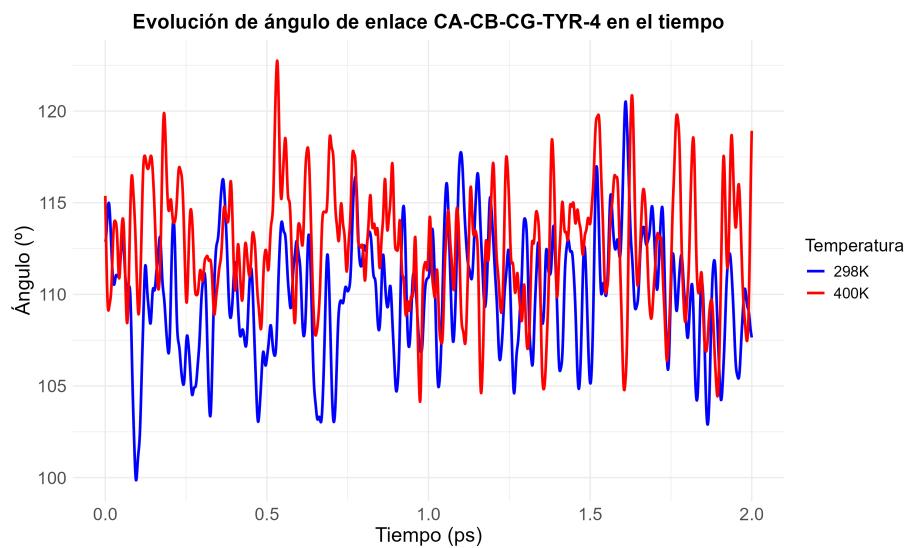


Figura 17: Ángulo de enlace CA-CB-CG-TYR-4

5.1.4. Evolución de ángulos dihedros

Un ángulo diedro se refiere a la separación entre dos planos que se extienden desde una línea común. En este contexto, los ángulos phi y psi corresponden a los ángulos formados por los planos a ambos lados del carbono alfa de un residuo específico en una proteína. Estos ángulos son cruciales para determinar la estructura secundaria de una proteína, ya que representan la rotación alrededor de los enlaces dentro de la cadena polipeptídica. Por lo tanto, juegan un papel fundamental en la descripción y comprensión de la conformación tridimensional de las proteínas.

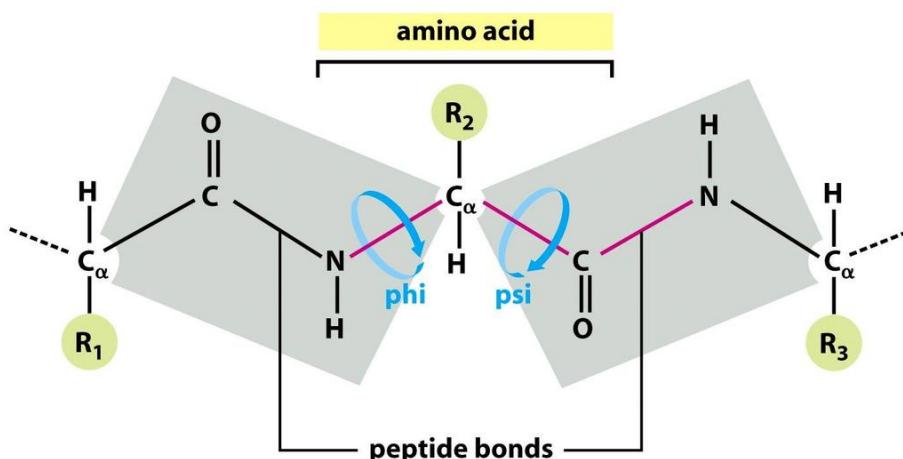


Figura 18: Ángulos de torsión phi y psi

Como vemos en la imagen, el ángulo phi es la rotación en torno al enlace Ca-N y el ángulo psi es la rotación en torno al enlace C-Ca. Como estos son los únicos grados de libertad que presenta la estructura, la conformación de la cadena polipéptídica queda completamente definida cuando se conocen los valores de estos ángulos para cada aminoácido.

Para calcular los ángulos se utiliza el siguiente comando en ambas carpetas (de nuevo como todos los pasos anteriores de esta sección de resultados, se realiza en 4.1-analysis y 4.2-analysis):

```
gmx rama -f ary.trr -s ary.tpr -xvg none
```

Este comando nos da un archivo rama.xvg. El formato de este archivo sigue una pauta donde la primera columna corresponde al ángulo phi, la segunda al ángulo psi, y la tercera al residuo dueño de esos valores. Además, se puede extraer la información para un residuo y un ángulo seleccionado. Con el comando grep podemos extraer solo los ángulos diedros del residuo que me interese. Lo haremos de la siguiente manera para dos residuos seleccionados (en este caso alanina y tirosina):

```
grep ALA-2 rama.xvg | awk '{print $1}' | cat -n > phi-ala-2.dat
grep ALA-2 rama.xvg | awk '{print $2}' | cat -n > psi-ala-2.dat
grep TYR-4 rama.xvg | awk '{print $1}' | cat -n > phi-tyr-4.dat
grep TYR-4 rama.xvg | awk '{print $2}' | cat -n > psi-tyr-4.dat
```

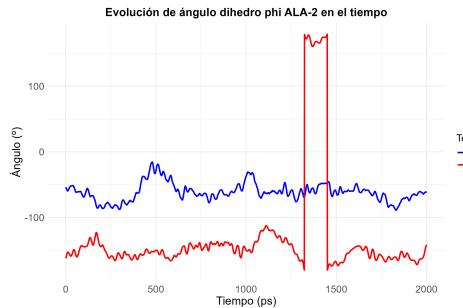


Figura 19: Ángulo dihedral phi ALA-2



Figura 20: Ángulo dihedral psi ALA-2

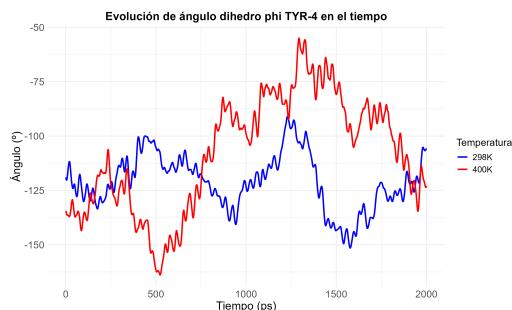


Figura 21: Ángulo dihedral phi TYR-4

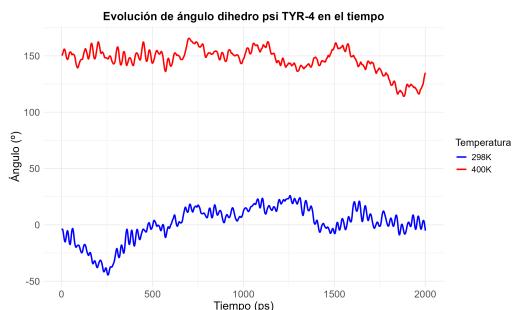


Figura 22: Ángulo dihedral psi TYR-4

Las cuatro representaciones gráficas correspondientes a los 4 ángulos dihedros de los residuos Alanina-2 y tirosina-4, **no muestran ningún patrón de comportamiento regular**. Esto se debe a que los ángulos diédros tienen un potencial de energía mucho más débil y la energía que necesita para rotar y moverse es mucho menor. Esto provoca que tenga mayor libertad de movimiento y por tanto haya mucha mayor variabilidad entre estos ángulos.

En base a esto, se puede observar que **aumenta dicha energía con la temperatura**, observándose mayor nivel de grado de los 4 ángulos a 400K e incluso viendo cambios muy bruscos de oscilación en ciertos instantes para los ángulos de la alanina.

5.1.5. Evolución de temperatura

Para obtener la temperatura del sistema usaremos el siguiente comando en cada una de las dos carpetas creadas para trabajar con ambas temperaturas:

```
gmx traj -f ary.trr -s ary.tpr -xvg none -ot
```

Al introducir este comando en ambas carpetas debemos seleccionar un grupo como ha ocurrido anteriormente y en este caso es: *0 (System)*. Este comando gmx traj, ya usado anteriormente, se utiliza para extraer parámetros del sistema. En este caso el término -ot devuelve los datos de temperatura del sistema en base a las velocidades en dos arhivos de salida llamados temp.xvg.

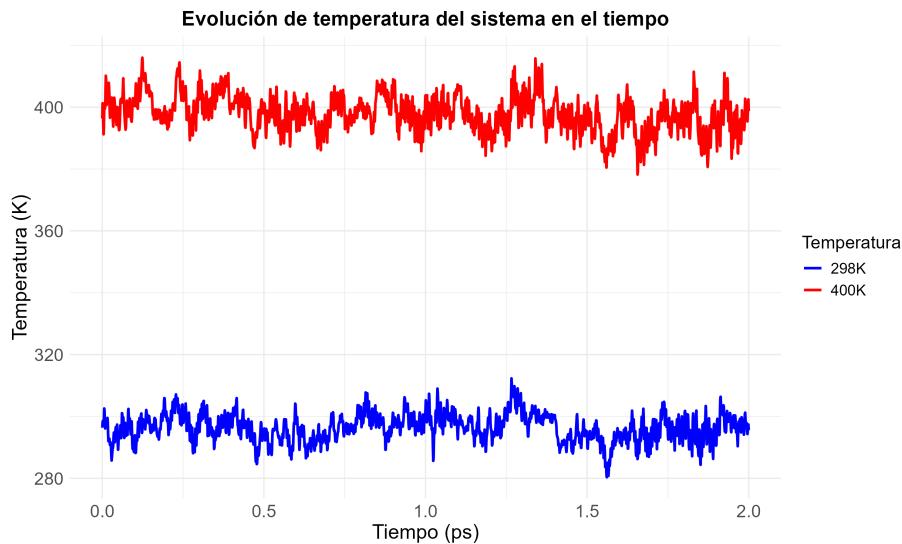


Figura 23: Temperatura del sistema ARY

En ambas representaciones, se observan fluctuaciones de la temperatura en la molécula, sin embargo, estas son muy débiles y no parecen describir una tendencia a aumentar o disminuir globalmente. En concreto, podemos ver que hay cambios hacia arriba o hacia abajo, pero estos no suelen ser superiores a los 15-20 K durante la ejecución del proceso.

5.1.6. Evolución de velocidades

Como hemos comentado en el apartado anterior, volveremos a usar el comando gmx traj para extraer parámetros del sistema. En este caso, el módulo de la velocidad respecto al tiempo de los 5 primeros átomos de nuestra simulación, para ambas temperaturas. En concreto, el comando sería:

```
gmx traj -f ary.trr -s ary.tpr -xvg none -ov -len
```

Las representaciones gráficas proceden del archivo veloc.xvg, donde viene recogido el tiempo en la primera columna, las 3 componentes de la velocidad (x,y,z) en las columnas 2,3 y 4 y la quinta columna corresponde al módulo de la velocidad y será la que representaremos en función del tiempo y a nivel de frecuencia, para ver que valores de velocidad son más frecuentes vía histograma.

Debemos considerar que, la velocidad es un movimiento parecido a la distancia y por tanto, para cada átomo seleccionado veremos que la velocidad formara parte de un oscilador armónico.

En nuestro caso, al elevar la temperatura, aumentan las medias de los módulos de la velocidad de los átomos 1, 3 y 4. Esto se debe a que las partículas se ven aceleradas al aumentar la temperatura. Sin embargo, dependiendo del átomo y los enlaces en los que está implicado, puede no aumentar esta velocidad, como ocurre en los átomos 2 y 5.

Por otro lado, como se puede apreciar en los histogramas, las velocidades siguen una distribución de tipo Maxwell-Boltzmann. Además, valores elevados de velocidad se vuelven más frecuentes en ciertos átomos al aumentar la temperatura.

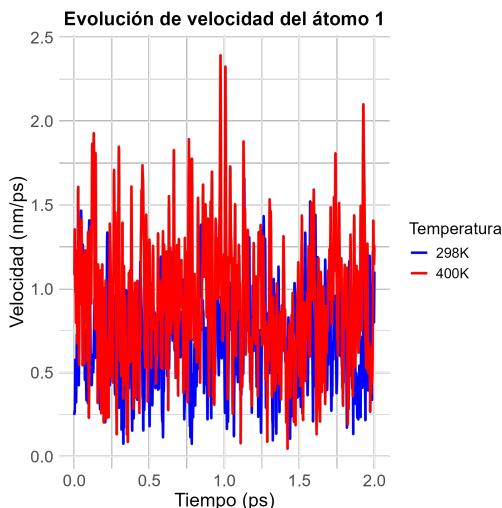


Figura 24: Evolución átomo 1

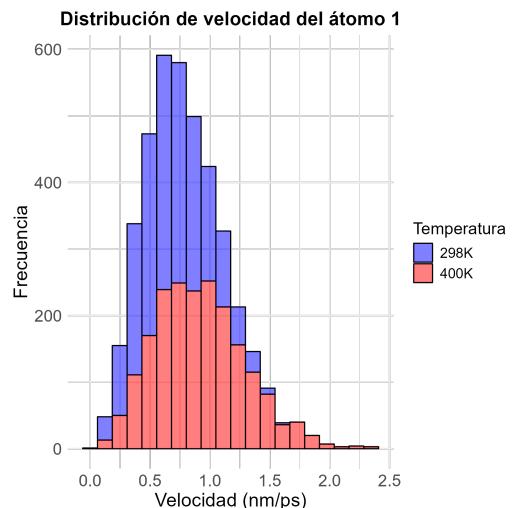


Figura 25: Distribución átomo 1

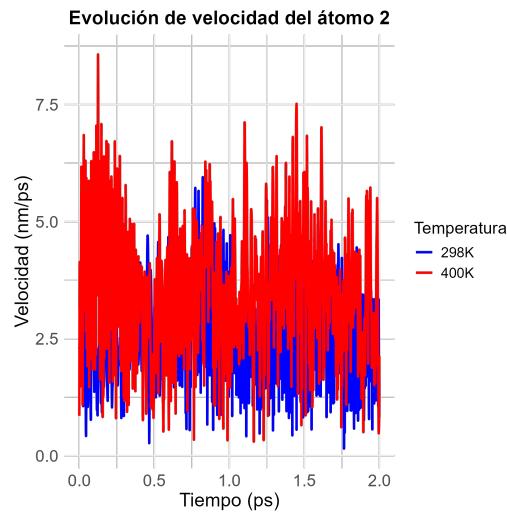


Figura 26: Evolución átomo 2

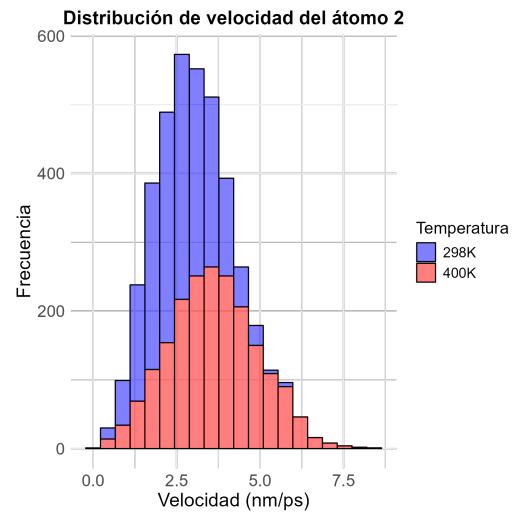


Figura 27: Distribución átomo 2

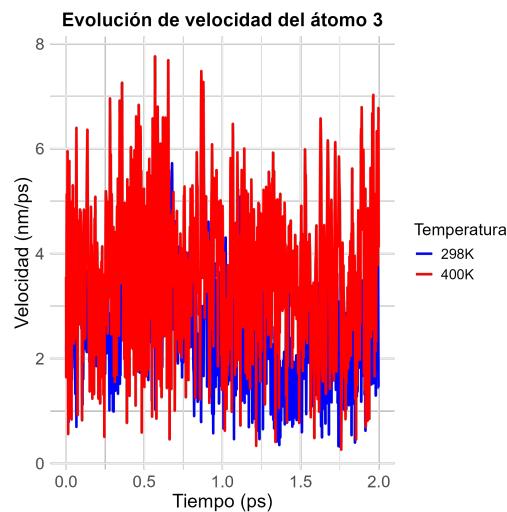


Figura 28: Evolución átomo 3

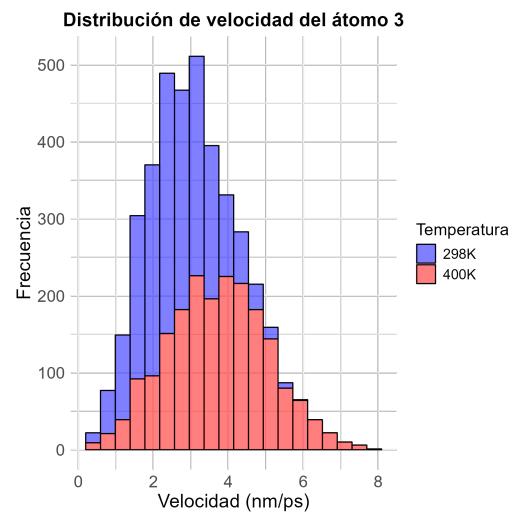


Figura 29: Distribución átomo 3

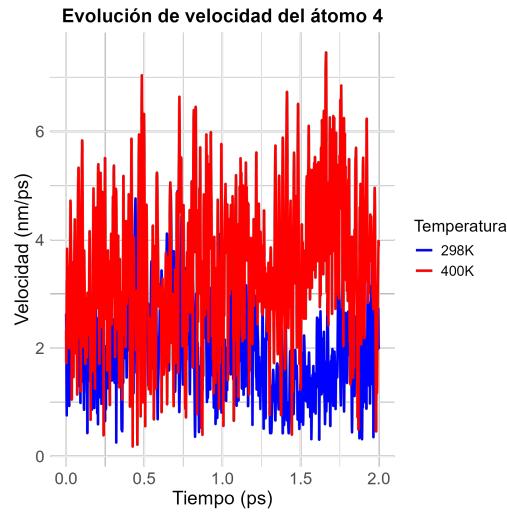


Figura 30: Evolución átomo 4

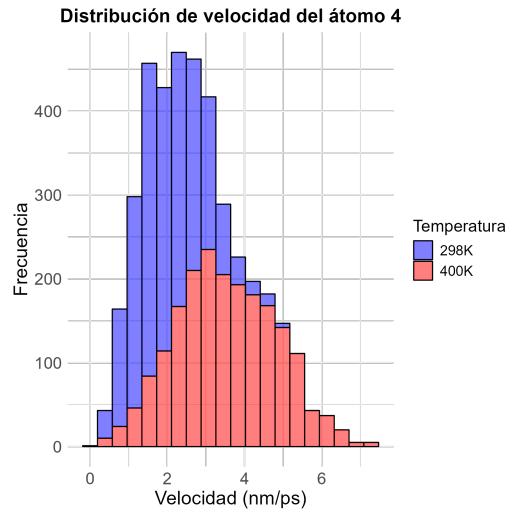


Figura 31: Distribución átomo 4

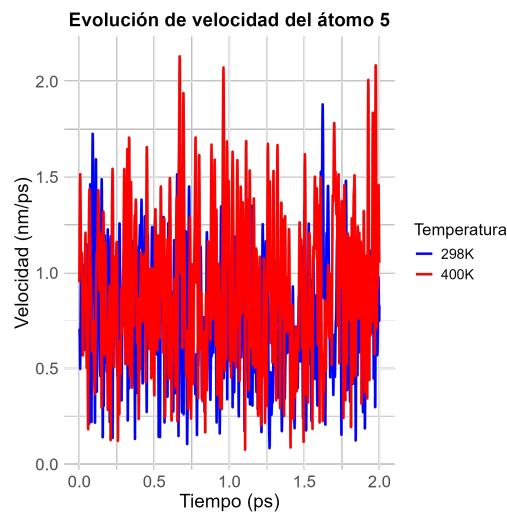


Figura 32: Evolución átomo 5

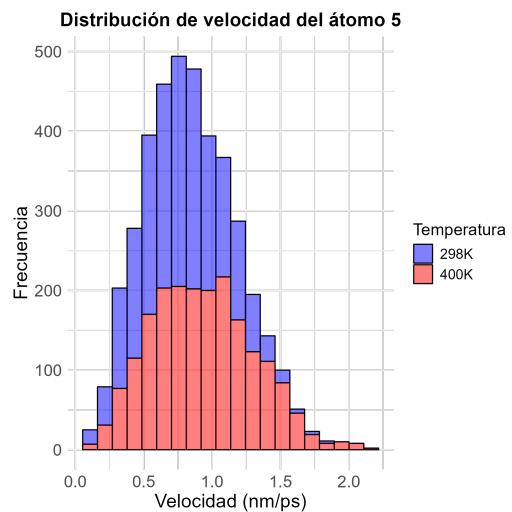


Figura 33: Distribución átomo 5

5.1.7. Energía total del sistema

Otro aspecto a valorar es la energía total del sistema. Para ello se utilizará el siguiente comando (*seleccionando 15: Total-Energy*):

```
gmx energy -f ary.edr -s ary.tpr -xvg none
```

Este comando (gmx energy), permite calcular la energía del sistema. Tras introducir como parámetro este *Total-Energy*, el programa nos da la media, desviación típica, y el "Tot-Drift" del sistema.

Este parámetro denominado "Tot-Drift", indica si hay algún tipo de pendiente en los datos. En teoría, no debería haberla y la energía debería permanecer constante. Con ambas temperaturas este parámetro es pequeño, siendo dos órdenes de magnitud menor al valor de la media.

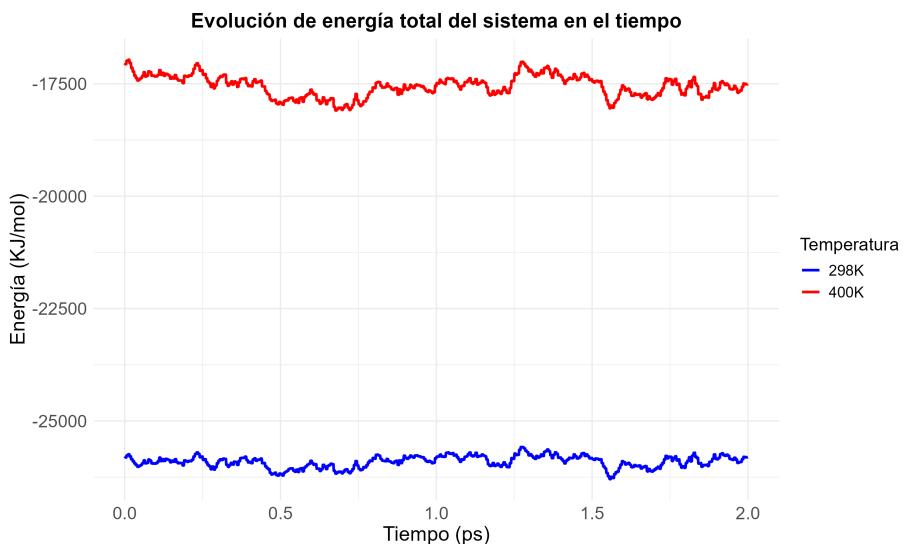


Figura 34: Energía total del sistema ARY

Energy	Average	Err.Est.	RMSD	Tot-Drift
Total Energy	-25907.4	36	130.352	70.0597 (kJ/mol)

Figura 35: Energiá total media a 298K

Energy	Average	Err.Est.	RMSD	Tot-Drift
Total Energy	-17553.7	84	233.887	-118.816 (kJ/mol)

Figura 36: Energiá total media a 400K

Se puede observar en la figura 34 que, la energía sigue un patrón parecido en ambos casos. Se puede observar también que la energía es mayor a 400 K que a 298 K para este tripéptido. Esto se debe a que a más temperatura, más velocidad tendrán las partículas del sistema. Por tanto, La energía interna de un cuerpo y su temperatura son proporcionales, de modo que si la energía interna del cuerpo aumenta la temperatura también aumenta. La constante de proporcionalidad entre la energía interna y la temperatura es el calor específico del cuerpo.

5.1.8. Energía cinética del sistema

Del mismo modo que en el caso anterior, resulta interesante explorar la energía cinética del sistema. Para ello se utilizará de nuevo el comando (*seleccionando 14: Kinetic Energy*):

```
gmx energy -f ary.edr -s ary.tpr -xvg none
```

En este caso, podemos observar que no existen grandes fluctuaciones de la energía cinética a lo largo del tiempo para ambas temperaturas. Además, a mayor temperatura, mayor es la velocidad interna de las partículas como ya hemos comentado. En consecuencia, la energía cinética del sistema aumentará, como ocurre en este caso.

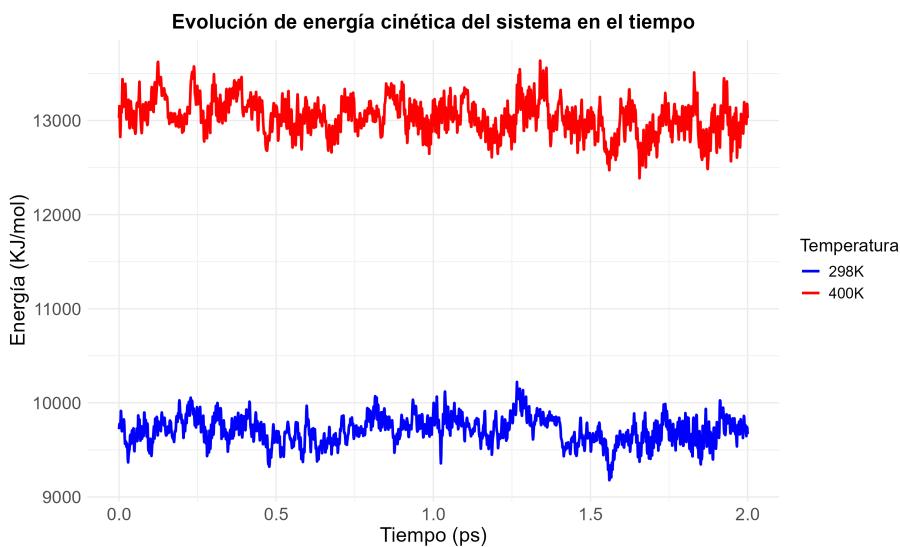


Figura 37: Energía cinética del sistema ARY

Energy	Average	Err.Est.	RMSD	Tot-Drift
Kinetic En.	9710.28	20	143.834	-41.3112 (kJ/mol)

Figura 38: E. cinética media a 298K

Energy	Average	Err.Est.	RMSD	Tot-Drift
Kinetic En.	13038.6	36	189.993	-234.879 (kJ/mol)

Figura 39: E. cinética media a 400K

El análisis de la energía cinética es una manera de comprobar que las simulaciones están siendo correctas, es decir, podemos calcularla por medio del número de átomos de nuestro sistema y la Kbt (constante Boltzmann) y calcular a su vez el error relativo. De esta manera, buscaremos un error lo más pequeño posible que indique la correcta simulación de nuestro sistema.

El número de átomos lo extraemos del archivo .g96 (2627), mientras que la constante de Boltzmann tiene un valor de 2,478kJ/mol

$$2627 \text{ atoms} \times \frac{3}{2} \times kBT \left(2,478 \frac{\text{kJ}}{\text{mol}} \right) = 9764,6 \frac{\text{kJ}}{\text{mol}} \quad (1)$$

La fórmula para calcular el error relativo se expresa como:

$$\text{Error relativo} = \left(\frac{V_v - V_a}{V_v} \right) \times 100 \% \quad (2)$$

Donde:

V_v = Valor esperado

V_a = Valor observado

En este sentido el resultado será:

$$\text{Error relativo} = \left(\frac{9764,60 - 9710,28}{9764,60} \right) \times 100 = 0,56 \% \quad (3)$$

Realizamos los cálculos solamente para la temperatura de 298K, donde vemos que obtenemos un error muy bajo, menor del 1% , por lo que damos por favorable la simulación realizada.

5.2. Simulación extendida de A-R-Y

Además de la simulación básica a 400 ps, obteniendo resultados cada 1 fs, realizaremos una simulación más profesional ejecutando a 500 ps y obteniendo resultados cada 10 fs. Esto se va a realizar a la temperatura de 298K y para ello partiremos de los mismos directorios y comandos de topología y equilibrado en la simulación básica.

Lo que vamos a modificar es el archivo de ejecución runNVT.mdp, que en este caso tendrá 200000 nsteps más:

```
nsteps = 1000000 ; 500 ps
nstxout = 20 ; Save coordinates every 10 fs
nstvout = 20 ; Save velocities every 10 fs
nstlog = 20 ; Update log every 10 fs
nstenergy = 20 ; Save energies every 10 fs
```

Este archivo será ejecutado con el mismo comando que en el apartado de ejecución y será lanzado al gestor de colas con el mismo archivo que en la simulación básica. Los archivos de esta simulación podrán ser encontrados en:

```
/home/alumno15/personal-report-MM/extended-report
```

5.2.1. Distribución de temperatura

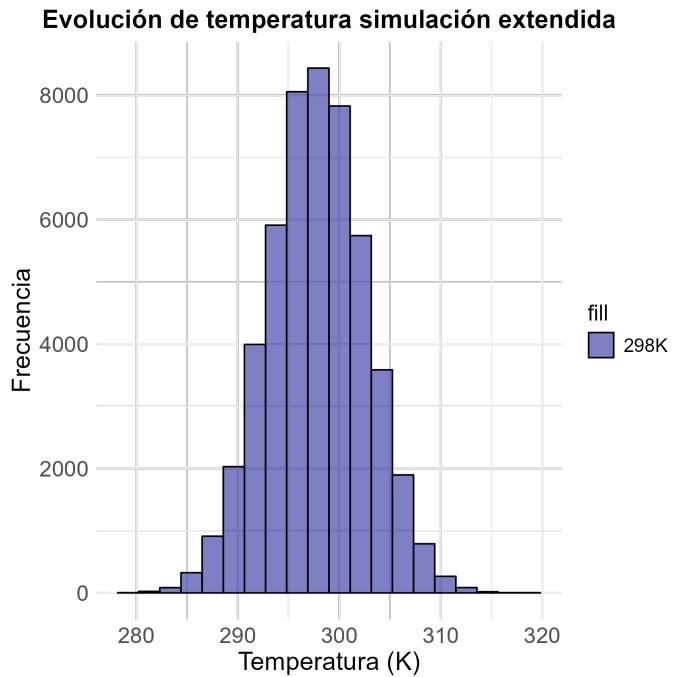


Figura 40: Distribución temperatura simulación extendida

En cuanto a la distribución de velocidad, en este caso al tratarse de una simulación más robusta vemos claramente como el valor más frecuente se sitúa en torno a los 300K, permaneciendo en un rango de temperatura muy estable a lo largo de la simulación (figura 40).

5.2.2. Distribuciones de velocidad

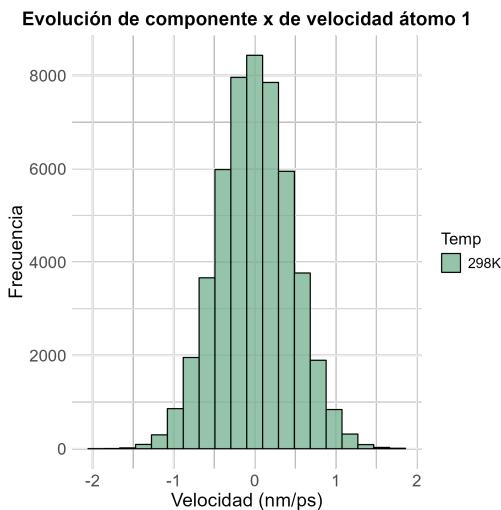


Figura 41: Componente x átomo 1

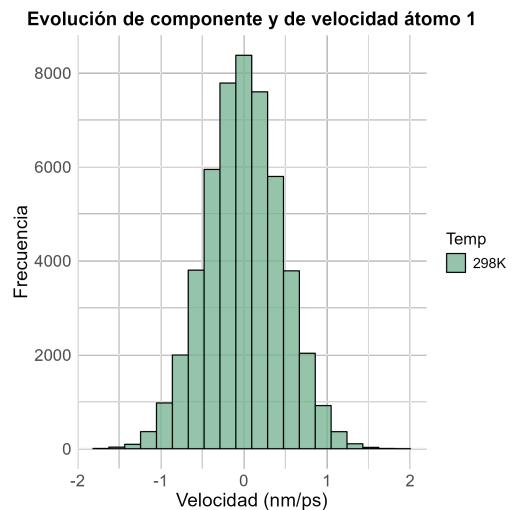


Figura 42: Componente y átomo 1

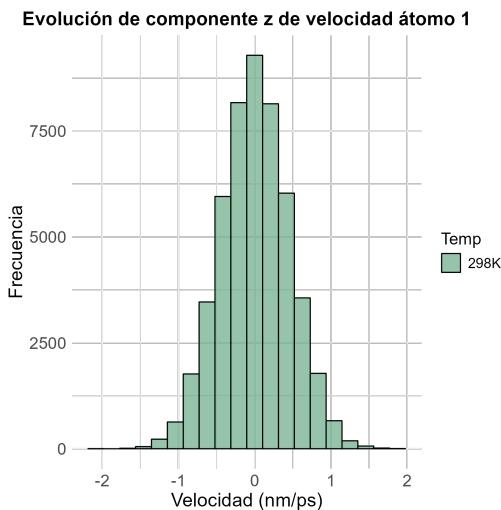


Figura 43: Componente z átomo 1

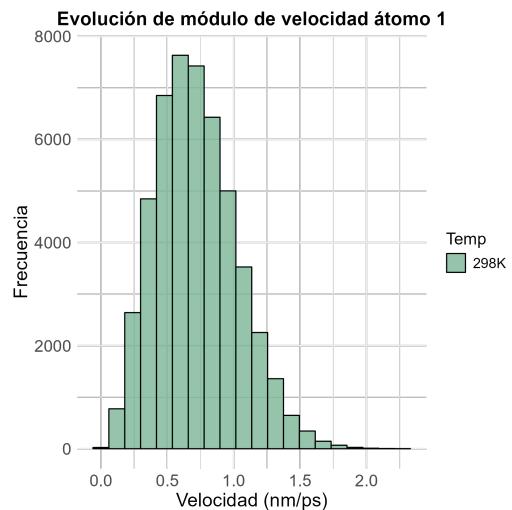


Figura 44: módulo átomo 1

En cuanto a las distribuciones de probabilidad del átomo 1 (átomo seleccionado), vemos que sigue una distribución de Maxwell-Boltzmann para las 3 componentes y el módulo de la velocidad. Esto indica que estamos ante un sistema en equilibrio térmico y proporciona información valiosa sobre la distribución de velocidades de las partículas en el sistema.

5.2.3. Diagrama de Ramachandran

El diagrama de Ramachandran es un tipo de representación gráfica, donde se pueden visualizar todas las combinaciones posibles de ángulos de torsión psi contra phi en los aminoácidos de un polipéptido, y que contribuyen a la conformación de la estructura de las proteínas. [5]

Este diagrama es una forma de visualizar las regiones energéticamente permitidas para los ángulos diedros del esqueleto de una proteína. Así, nos permite determinar cuántos residuos se encuentran formando parte de un elemento de estructura secundaria determinado, y cuántos presentan unos ángulos de torsión forzados de acuerdo con la estructura de la proteína de la que disponemos, de modo que cuantos más residuos presenten ángulos forzados, más probable será que la estructura de la proteína no sea la más adecuada.

Por tanto, el diagrama de Ramachandran permite obtener una aproximación de cuál será la estructura secundaria del péptido, ya que determinadas combinaciones de ángulos están asociadas a determinadas estructuras, como la hélice alfa o la lámina beta.

Los diagramas serán representados uno para cada residuo. Primero debemos extraer los ángulos phi y psi de los residuos en base a los datos de la simulación extendida que se encuentran en la carpeta extended-report. Para ello se usa el módulo *rama* con el siguiente comando:

```
gmx rama -f ary.trr -s ary.tpr -xvg none
```

Además del archivo .xvg con todos los ángulos diedros, debemos separar los ángulos de cada residuo en distintos archivos, para lo cual usaremos los siguientes comandos:

```
grep ALA-2 rama.xvg |cat -n > phi-psi-ala-2.dat
grep ARG-3 rama.xvg |cat -n > phi-psi-arg-3.dat
grep TYR-4 rama.xvg |cat -n > phi-psi-tyr-4.dat
```

Una vez que hemos obtenido estos datos procedentes de cada uno de los aminoácidos del tripéptido, representamos los gráficos.

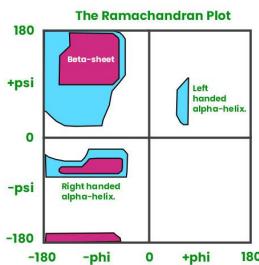


Figura 45: Diagrama de Ramachandran teórico

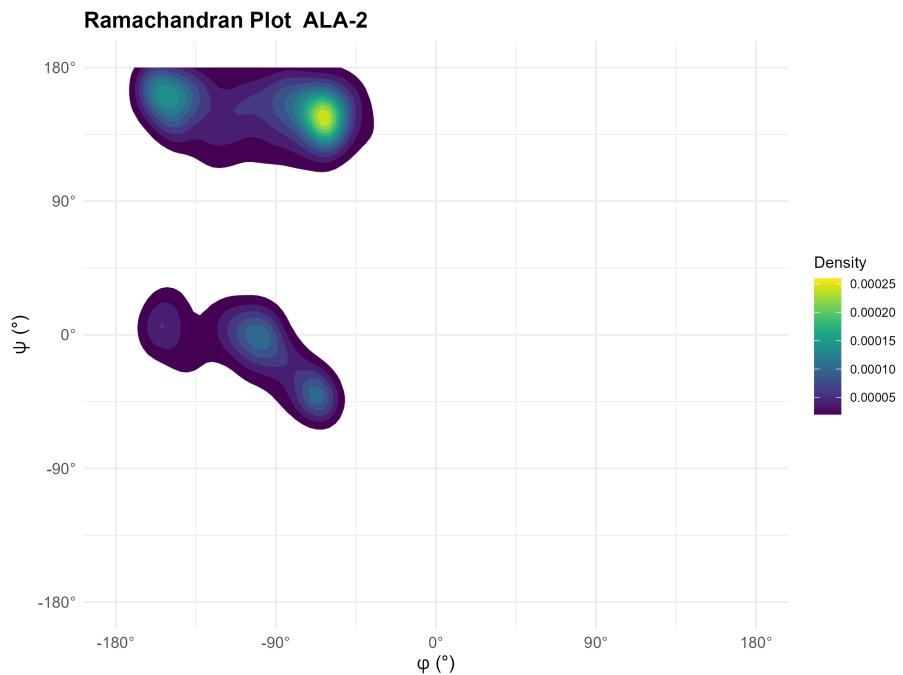


Figura 46: Diagrama de Ramachandran ALA-2

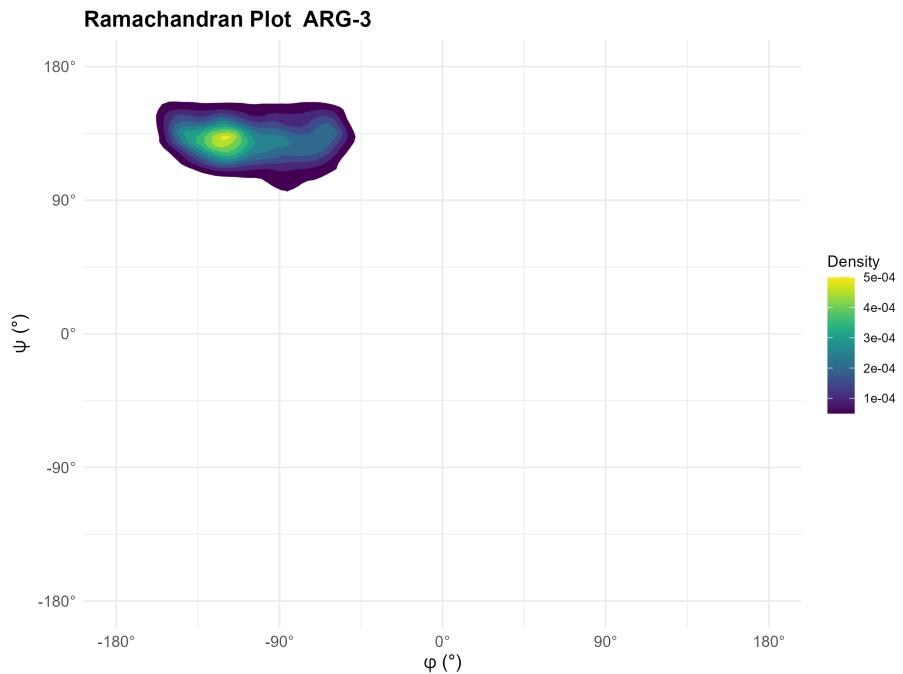


Figura 47: Diagrama de Ramachandran ARG-3

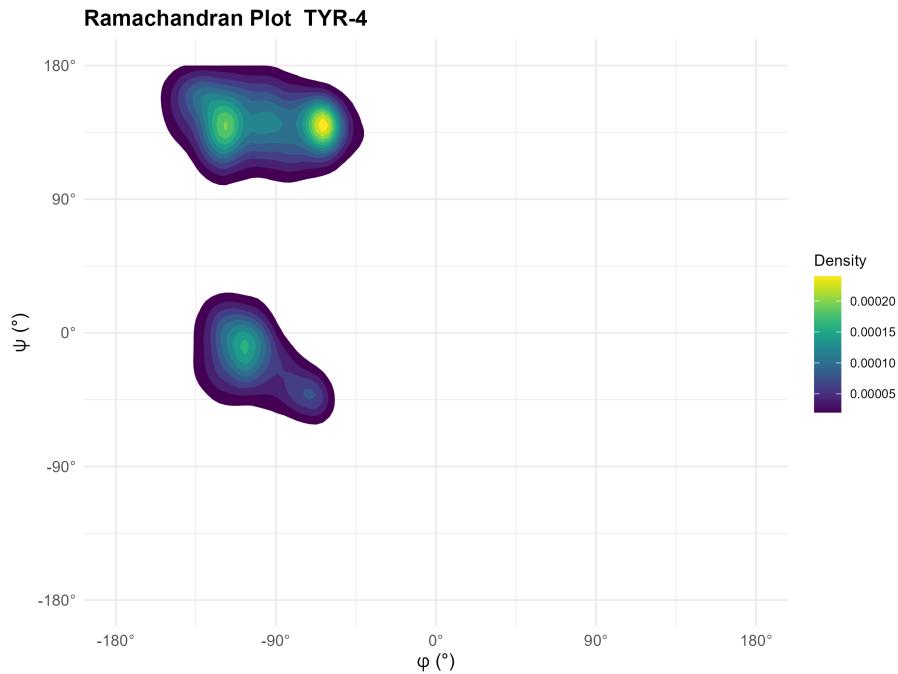


Figura 48: Diagrama de Ramachandran TYR-4

5.2.4. Conformaciones de mayor probabilidad

Para interpretar las representaciones, debemos conocer que en el diagrama (figura 45), las zonas blancas corresponden a conformaciones donde los átomos del polipéptido están más cercanos que sus radios de van der Waals. Estas regiones son estéricamente no permitidas para todos los aminoácidos, excepto para Glicina, el cual es el único aminoácido que no presenta cadena lateral.

Las regiones rojas corresponden a conformaciones donde no hay impedimentos estéricos, es decir estas son zonas permitidas llamadas conformaciones alfa-hélices y láminas-beta.

En base a esto, si observamos las zonas coloreadas en el diagrama correspondiente al residuo Alanina-2 (figura 46), vemos que la zona del primer cuadrante corresponde a zonas permitidas y corresponden a estructuras de láminas-beta y la del tercer cuadrante a helices-alfa, por lo que serán las conformaciones más probables en estructura secundaria.

Esto mismo ocurre para el caso de Tirosina-4, donde vemos conformaciones más probables láminas.beta y hélices alfa.

En cambio, para el caso del residuo Arginina-3, solo vemos coloreadas la zona del primer cuadrante por lo que solo encontraremos como estructura secundaria más probable el tipo de lámina-beta.

6. Conclusiones

Por último, a modo de conclusión, destacar que con la realización de esta práctica se han cumplido todos los objetivos, tanto generales como específicos.

Más concretamente, hemos sido capaces de llevar a cabo dos simulaciones básicas y profesionales, a dos temperaturas diferentes sobre un tripéptido específico en agua, poniendo en práctica nociones de dinámica molecular y simulación mediante el software GROMACS. Además, hemos ido visualizando con PyMOL ciertas estructuras de nuestro sistema tanto de forma independiente (tripéptido) como en agua (tripéptido solvatado).

Por último, hemos representado e interpretado todos los parámetros que han resultado interesantes de nuestras simulaciones y comprobado por medio de la energía cinética que se ha desarrollado dicha simulación adecuadamente.

7. Referencias

- [1] D. Jones, *Student Tutorial for PyMOL: for educational use Visualising proteins using molecular graphics.* Feb. 2021.
- [2] M. J. Abraham, T. Murtola, R. Schulz, S. Páll, J. C. Smith, B. Hess, and E. Lindahl, “GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers,” *SoftwareX*, vol. 1-2, pp. 19–25, Sept. 2015.
- [3] “Welcome to the GROMACS documentation! — GROMACS 2016.4 documentation.”
- [4] A. Astuti, R. Refianti, and A. Mutiara, “Molecular Dynamics Simulation on Protein Using Gromacs,” *International Journal of Computer Science and Information Security*, vol. Vol. 9, pp. 16–20, Feb. 2011.
- [5] S. W. Park, B. H. Lee, S. H. Song, and M. K. Kim, “Revisiting the Ramachandran plot based on statistical analysis of static and dynamic characteristics of protein structures,” *Journal of Structural Biology*, vol. 215, p. 107939, Mar. 2023.