



Universidad de Murcia

FACULTAD DE BIOLOGÍA

DINÁMICA MOLECULAR DE PÉPTIDOS

Modelado molecular

Autor:
Guillermo Sánchez-Cid Bueno

Marzo 2022

Índice

1. Introducción	2
2. Modelado	2
2.1. Estructurando el directorio de trabajo	2
2.2. Construcción de la molécula	2
3. Equilibrado	5
4. Neutralización	6
5. Propagación	7
6. Análisis	8
6.1. Radio de giro	8
6.2. Distancias de enlace	9
6.3. Ángulos de enlace	10
6.4. Ángulos dihedros	13
6.5. Temperatura	14
6.6. Velocidades	15
6.7. Energía total	17
6.8. Energía cinética	18
7. Informe avanzado	19

1. Introducción

En el transcurso de este informe se procederá al modelado y análisis de la dinámica molecular del tripéptido ARF, compuesto por residuos de alanina (A), arginina (R) y fenilalanina (F). Para ello, se ha tomado el archivo de la molécula en formato .pdb (Protein Data Bank) de la [web](#) del profesor Adolfo Bastida, que imparte la asignatura de modelado molecular en el Máster de Bioinformática en la Universidad de Murcia (Curso 2021/2022). Los ficheros y scripts tanto del [informe básico](#) como del [informe avanzado](#) están disponibles para ser consultados.

2. Modelado

En el modelado del tripéptido ARF se han tenido en cuenta las siguientes consideraciones:

- El software utilizado para la simulación es GROMACS.
- Para la realización de cálculos el formato utilizado es coma flotante de simple precisión.
- Para las visualizaciones se utiliza Pymol.
- Las gráficas de Ramachandran de los residuos del tripéptidos están hechas con R.
- El resto de gráficas están hechas con gnuplot.

2.1. Estructurando el directorio de trabajo

```
#Creamos directorio donde realizar la práctica
```

```
> mkdir informe_personal
```

```
> cd informe_personal
```

```
#Obtención del tripéptido
```

```
> cp ../for-students/pymol/arf.pdb .
```

2.2. Construcción de la molécula

```
#Fichero de modelado y visualización
```

```
> mkdir 1-build
```

```
> cd 1-build
```

```
#Obtenemos el campo de fuerza que vamos a utilizar: CHARMM27
```

```
> cp ../for-students/pdb-file-charmm27/arf.pdb .
```

```
#Visualización del tripéptido
```

```
> pymol arf.pdb &
```

Mediante el visualizador de Pymol, se puede observar cuál sería el aspecto de nuestro tripéptido en 3D:

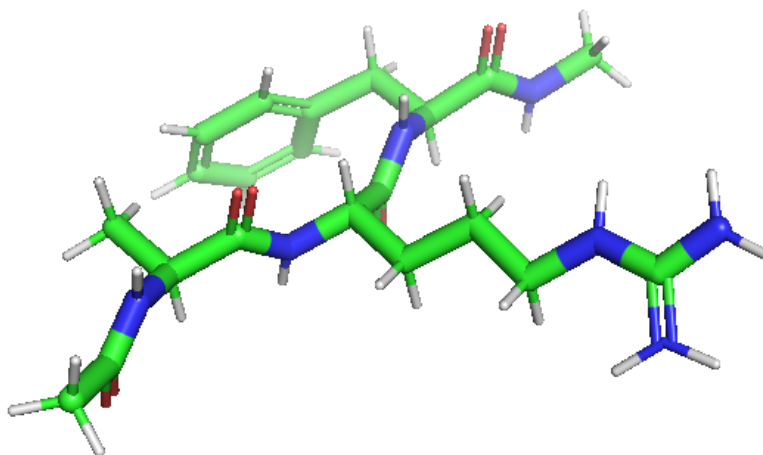


Figura 1: Estructura 3D del tripéptido ARF.

En el fichero arf.pdb podemos encontrar mucha información sobre el tripéptido:

- Átomo
- Número del átomo
- Clase del átomo con respecto a su grupo funcional y posición en la molécula.
- Nombre de residuo al que pertenece (desde el extremo N terminal al extremo C terminal).
- Coordenadas en Amstrongs.
- Ocupancia (rayos-X).
- Factor de temperatura (rayos-X).
- nombre del elemento.

```
#Leemos el fichero  
less arf.pdb
```

A continuación, utilizaremos GROMACS

```
#Es necesario incluir en el fichero .bashrc la siguiente línea para que funcione GROMACS  
> source /usr/local/gromacs/bin/GMXRC.bash  
  
#Creamos el archivo de topología del tripéptido con el módulo pdb2gmx  
> gmx pdb2gmx -f arf.pdb -o arf.gro -p arf.top -ter
```

El primer módulo que vamos a usar transforma un fichero PDB a un formato específico de Gromacs (pdb2gmx). Con `-f` especificamos el fichero de entrada mientras que con `-o` especificamos el fichero de salida. Tras ello aparece la operación `-p`, que lo que hace es asociar un campo de fuerza, lo que se denomina asociar un fichero de topología (de ahí el .top). `-ter` es para que pregunte qué hacer con los grupos terminales.

Una vez ejecutado el comando, deben seleccionarse los siguientes parámetros:

- Campo de fuerza: CHARMM27
- Modelo del agua: TIP3P
- El tipo de término de comienzo para ACE-1: None
- El tipo de término de comienzo para CT3-7: None

Nos genera dos archivos:

- .gro: en este archivo encontramos información sobre el número de átomos, el número de residuo de cada átomo, el nombre del residuo, la clase de átomo, el número del átomo, las coordenadas (x, y, z) expresadas en nanómetros.
- .top: en este archivo encontramos información necesaria para evaluar la función de energía potencial: cargas, los enlaces atómicos y sus ángulos diedros.

#Creamos la unidad celda: un cubo de proporciones 3.0x3.0x3.0 nm mediante el módulo #editconf, en el cual se indica el fichero de entrada para crear la caja, el fichero #de salida y el tamaño de la caja

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

#Solvatamos la unidad celda con moléculas de agua e introducidos el tripéptido

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

Condiciones dentro del cubo de agua:

- Volumen: 27 (nm^3)
- Número de moléculas de disolvente: 850
- A 298K la densidad del agua es 0.997 Kg/l

#Visualizamos el fichero de la unidad celda solvatada con el tripéptido

```
> less arf-box-solv.gro
```

#Generamos el archivo .pdb para visualizarlo en pymol

```
> gmx editconf -f arf-box-solv.gro -o arf-box-solv.pdb
```

#Vemos el renderizado en pymol de la molécula en el cubo de agua

```
> pymol arf-box-solv.pdb
```

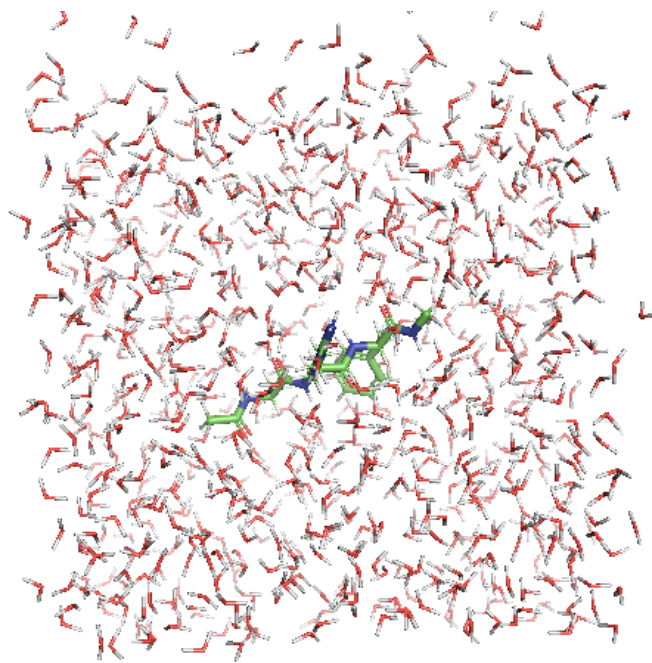


Figura 2: Estructura 3D del tritéptido ARF en la celda unidad solvatada.

3. Equilibrado

La forma en la que se ha procedido en este apartado y en los siguientes es realizar el equilibrado, propagación y análisis en paralelo, es decir, todo el proceso es idéntico: los nombres de los ficheros, los comandos y las elecciones de configuración (salvo la temperatura). Lo único que cambia es el nombre del directorio en el que se trabaja. Por ejemplo, el nombre del directorio para este apartado sería 2-equilibration para la simulación de 298K, y 2-equilibration400 para el directorio de la simulación de 400K; y así sucesivamente. Para simplificar el informe, nos limitaremos a indicar los comandos para la simulación de 298K y se indicará en los pasos en los que difiere la simulación de 400K.

#Preparamos el directorio de este apartado con los ficheros necesarios

```
> cd .. > mkdir 2-equilibration
> cd 2-equilibration
> cp ../1-build/arf-box-solv.gro .
> cp ../1-build/arf.top .
```

#Copiamos el fichero del directorio forstudents

```
> cp ../for-students/equilibration-mdp/equiNVT.mdp .
```

#Configuración del fichero equiNVT.mdp

```
> nano equiNVT.mdp
```

Hacemos una simulación con nsteps 400000. Para la simulación de 298K dejamos la temperatura, ref_t, como está. Para la simulación de 400K hay que cambiar ref_t a 400. En el fichero equiNVT.mdp es donde están los parámetros de la dinámica molecular para hacer la simulación de equilibrado:

- integrator = md ; leap-frog integrator
- dt = 0.0005 ; 0.5 fs
- nsteps = 400000

- nstxout = 200 ; Save coordinates every 0.1 ps
- Tcoupl = v-rescale gen_vel = yes ; Should be the first equilibration
- gen.temp = 298.0 ; Temperature to generate corresponding Maxwell distribution
- gen.seed = 12345678 ; Random seed.

4. Neutralización

#Comprobamos el estado del sistema

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

#Output:

Setting the LD random seed to 1579677451

Generated 20503 of the 20503 non-bonded parameter combinations

Generating 1-4 interactions: fudge = 1

Generated 17396 of the 20503 1-4 parameter combinations

Excluding 3 bonded neighbours molecule type 'Protein'

Excluding 2 bonded neighbours molecule type 'SOL'

System has non-zero total charge: 1.000000

Velocities were taken from a Maxwell distribution at 298 K

Removing all charge groups because cutoff-scheme=Verlet

Analysing residue names:

There are: 5 Protein residues

There are: 855 Water residues

Como se puede comprobar con el comando less al archivo arf.pdb, el átomo 26 (hidrógeno), grupo NH1 el cual está cargado positivamente (N1+) y pertenece al residuo de arginina. Para neutralizar esa carga positiva es necesario añadir una carga negativa a la molécula. En este caso, se añade un ión cloruro.

#Neutralizamos la carga con el módulo genion de GROMACS

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

#Seleccionamos grupo SOL para reemplazar un molécula del solvente por el ión cloruro.

#Comprobamos que el sistema ahora incluye el ión cloruro

```
> less 3clpro.gro
```

#Ahora el sistema es neutro. Vamos a comprobarlo

```
> gmx grompp -f equiNVT.mdp -c arf.gro -p arf.top -o arf.tpr
```

#Generamos un fichero pdb para visualizarlo en pymol con el módulo editconf

```
> gmx editconf -f arf.gro -o arf.pdb
```

```
> > pymol arf.pdb
```

Ejecutamos en el cluster el equilibrado

#Copiamos el fichero donde tenemos un script para mandar el trabajo al cluster

```
cp ../for-students/equilibration-mdp/submit_eck.sh .
```

#Veamos el script

```
> cat submit_eck.sh
```

```
#!/bin/bash
#
#SBATCH -p eck-q
#SBATCH --chdir=/home/alumno23/dinamica_molecular/informe_personal/2-equilibration
#SBATCH -J equilibrado
#SBATCH --cpus-per-task=1

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

El output que recibimos es el siguiente:

- slurm-XXXX.out (output de la cola del cluster)
- arf.g96 (posiciones finales y velocidades)
- arf.log (información)

5. Propagación

Destacar que seguimos con las dos simulaciones en paralelo (298K y 400K) pero solo se mostrarán los comandos de la simulación de 298K. Las velocidades ya están equilibradas, por tanto, no queremos que GROMACS las sobrescriba; para ello, se pone `gen velocities=No`.

Preparamos el directorio de este apartado

```
> mkdir 3-run
> cd 3-run
> cp ../2-equilibration/arf.top .
> cp ../for-students/g96-equilibrated/arf.g96 .
> cp ../for-students/run-mdp/runNVT.mdp .
```

Preparamos los archivos para la propagación

```
#Cambiamos ref_t a 400K para la segunda simulación y nsteps a 400000 tanto para la simulación
#de 298K como para la de 400K
> nano runNVT.mdp

#Obtenemos los archivos .tpr (contiene la estructura y la masa del tripeptido) y .trr
 #(información sobre las coordenadas, velocidades, etc)
> gmx grompp -f runNVT.mdp -c arf.g96 -p arf.top -o arf.tpr

#Copiamos el script para lanzar a la cola del cluster la propagación
> cp ../for-students/run-mdp/submit_eck.sh .

#Observamos el contenido del script
> cat submit_eck.sh
```

Output:

```
#!/bin/bash
#
#SBATCH -p eck-q
```



```
#SBATCH --chdir=/home/alumno23/dinamica\_molecular/informe_personal/3-run
#SBATCH -J equilibrado
#SBATCH --cpus-per-task=1

date
gmx mdrun -deffnm arf -c arf.g96 -nt 1
date

#El módulo traj extrae las coordenadas del sistema y las incluye en el fichero .pdb
> gmx traj -f arf.trr -s arf.tpr -oxt cartoon.pdb

#Seleccionamos la opción 1 (Protein)
Output: fichero cartoon.pdb           #Este archivo se puede visualizar en pymol
```

6. Análisis

Preparamos el directorio de este apartado

```
> mkdir 4-analysis

> cd 4-analysis/

> cp ../3-run/arf.tpr .

> cp ../3-run/arf.g96 .

> cp ../3-run/arf.trr .

#El fichero aarf.edr contiene información sobre las energías.
> cp ../3-run/arf.edr .
```

6.1. Radio de giro

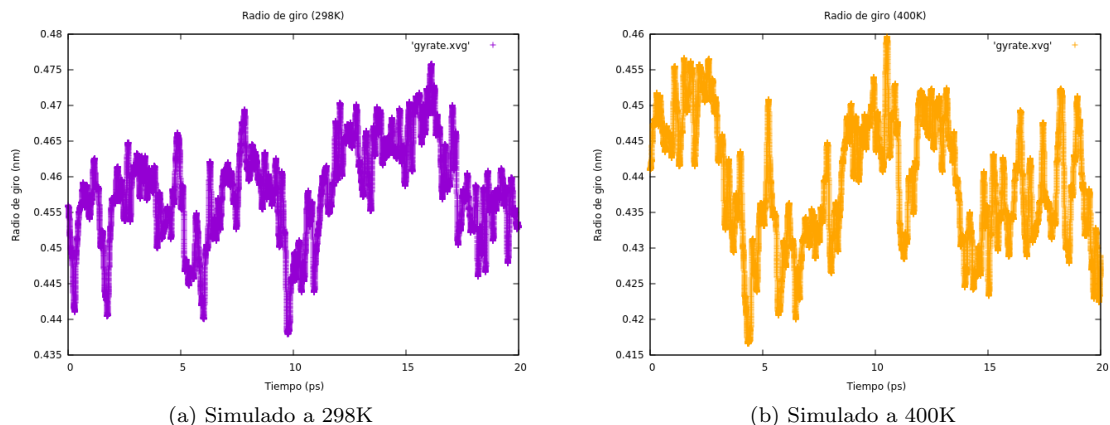
El radio de giro representa el momento de inercia con respecto al centro de masas. Concretamente, es la suma ponderada a la masa de cada átomo y de las distancias al cuadrado de su vector de posición. Nos da una idea del tamaño de nuestras moléculas.

```
#Generamos el archivo para estudiar el radio de giro.
> gmx gyrate -f arf.trr -s arf.tpr -xvg none
#Seleccionamos la opción 1 (Protein)
```

Se genera un fichero dist.xvg mediante el módulo gmx distance. Una vez creado el fichero dist.xvg podemos visualizarlo con gnuplot.

```
#Graficamos el radio de giro con gnuplot

> gnuplot
gnuplot> plot 'gyrate.xvg'
#Para cambiar el color al gráfico pondríamos el comando de la siguiente manera:
#gnuplot> plot 'gyrate.xvg'
gnuplot> set title 'Radio de giro (298K)'
gnuplot> set ylabel 'Distancia (nm)'
gnuplot> set xlabel 'Tiempo (ps)'
gnuplot> replot
```



(a) Simulado a 298K

(b) Simulado a 400K

Figura 3: Radio de giro

Como se puede observar en las gráfica del radio de giro en condiciones de temperatura diferente, cuando se incrementa la temperatura, el radio de giro aumenta lo que puede deberse a que los átomos se separan al desplegarse parte de la proteína debido al incremento de la energía en el sistema.

6.2. Distancias de enlace

Como se verá más abajo en las gráficas, no es un oscilador armónico perfecto: la amplitud varía con el tiempo (la amplitud varía dependiendo de cuánta energía esté almacenada en el enlace) por los flujos de energía que se producen por la interacción de los dos átomos del enlace con el medio en el que está. La distancia de equilibrio no cambia.

#Creamos un fichero con distancias de enlace llamada ndx.file con el enlace que nos interese
less arf.top en el directorio 3-run

```
nano distances.ndx
[CO-ala-2]
15 16
[CAHA-arg-3]
19 20
```

#Calculamos con el módulo distance de GROMACS las distancia entre el par de átomos
> gmx distance -f arf.trr -s arf.tpr -n distances.ndx -oall -xvg none

#Seleccionamos grupo 0
> Ctrl+D

Output (298K):

CO-ala-2:

```
Number of samples: 20001
Average distance: 0.12314 nm
Standard deviation: 0.00208 nm
```

#Seleccionamos grupo 1

> Ctrl+D

Output (298K)

CAHA-arg-3:

```
Number of samples: 20001
Average distance: 0.10824 nm
Standard deviation: 0.00269 nm
```

```

#Seleccionamos grupo 0
> Ctrl+D
Output 400K
CO-ala-2:
  Number of samples: 20001
  Average distance: 0.12316 nm
  Standard deviation: 0.00258 nm

#Seleccionamos grupo 0
> Ctrl+D
Output 400K:
CAHA-arg-3:
  Number of samples: 20001
  Average distance: 0.10840 nm
  Standard deviation: 0.00338 nm

#Graficamos con gnuplot
#Ejemplo con una de las gráficas de 298K. Las otras gráficas se hacen de la misma forma
> gnuplot
gnuplot > set title 'Distancia de enlace [CO-ala-2] (átomos 15 y 16) (298K)'
gnuplot> set ylabel 'Distancia (nm)'
gnuplot> set xlabel 'Tiempo (fs)'
gnuplot> plot 'dist.xvg' with linespoints

```

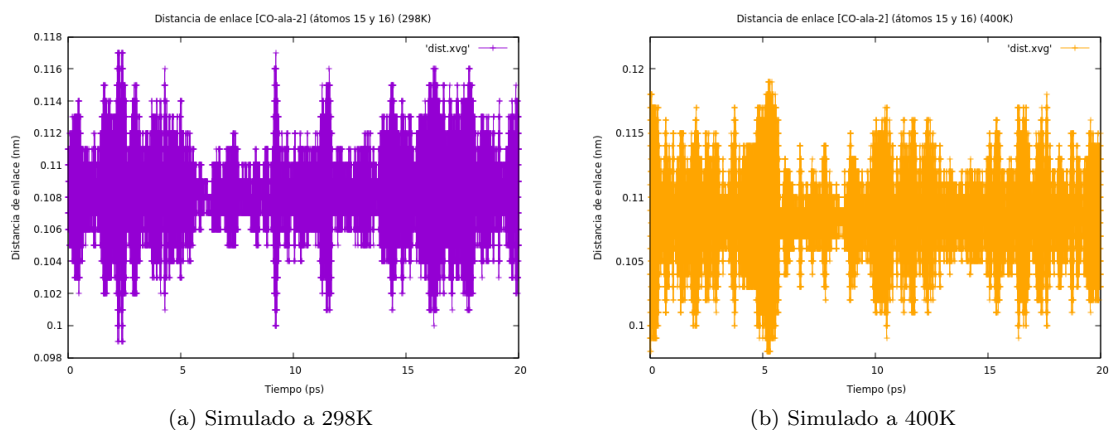


Figura 4: Distancia de enlace [CO-ala-2] (carbono-oxígeno), átomos 15 y 16

La amplitud depende de cuanta energía se introduce a las moléculas (a veces, las moléculas vibran con más energía y por ello la distancia respecto de la distancia de equilibrio del oscilador es mayor).

6.3. Ángulos de enlace

A continuación, calcularemos los ángulos de enlaces. Es necesario mencionar que para que exista un ángulo de enlace debemos tener tres átomos enlazados consecutivamente. Por tanto, para este apartado se ha incluido en el archivo `distances.ndx` del apartado anterior un átomo adyacente a cada ejemplo.

```

#Modificamos el archivo distances.ndx con un editor de texto
> nano distances.ndx
#Output
#Se ha añadido el átomo 14

```

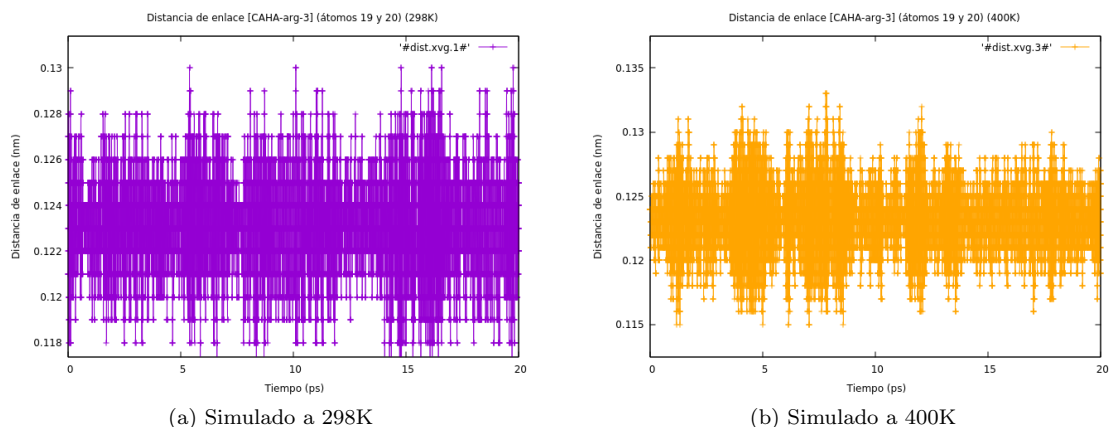


Figura 5: Distancia de enlace [CAHA-arg-3] (carbono aromático-hidrógeno no polar), átomos 19 y 20

```
[HB3CO-ala-2]
```

```
14 15 16
```

```
#Se ha añadido el átomo 18
```

```
[HNCAHA-arg-3]
```

```
18 19 20
```

```
#Con el módulo angle de GROMACS podemos calcular los ángulos de interés
```

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

```
Output 298K
```

```
Selected 0: 'HB3CO-ala-2'
```

```
trr version: GMX_trn_file (single precision)
```

```
Last frame      20000 time    20.000
```

```
Found points in the range from 74 to 149 (max 180)
```

```
< angle > = 104.651
```

```
< angle^2 > = 11090.4
```

```
Std. Dev.   = 11.7688
```

```
Select a group: 1
```

```
Selected 1: 'HNCAHA-arg-3'
```

```
trr version: GMX_trn_file (single precision)
```

```
Last frame      20000 time    20.000
```

```
Found points in the range from 109 to 148 (max 180)
```

```
< angle > = 132.33
```

```
< angle^2 > = 17535.1
```

```
Std. Dev.   = 4.87668
```

```
Output 400K
```

```
Selected 0: 'HB3CO-ala-2'
```

```
trr version: GMX_trn_file (single precision)
```

```
Last frame      20000 time    20.000
```

```
Found points in the range from 92 to 148 (max 180)
```

```
< angle > = 115.854
```

```
< angle^2 > = 13495.4
```

```
Std. Dev.   = 8.56007
```

```
Selected 1: 'HNCAHA-arg-3'
```

```
trr version: GMX_trn_file (single precision)
```

```
Last frame      20000 time    20.000
```

Found points in the range from 98 to 152 (max 180)

< angle > = 129.116

< angle² > = 16726.4

Std. Dev. = 7.45228

#Graficamos con gnuplot. Esto es un ejemplo para una de las gráficas

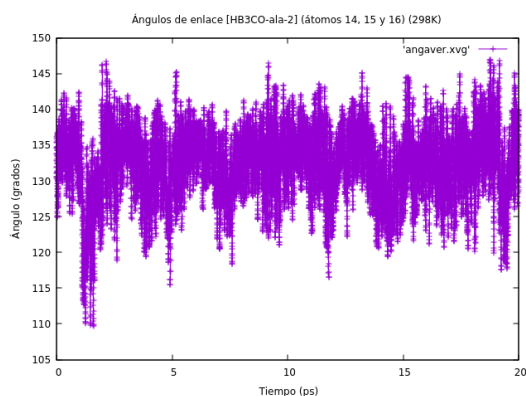
```
gnuplot> plot 'angaver.xvg' with linespoints
```

```
gnuplot> set ylabel 'Ángulo (grados)'
```

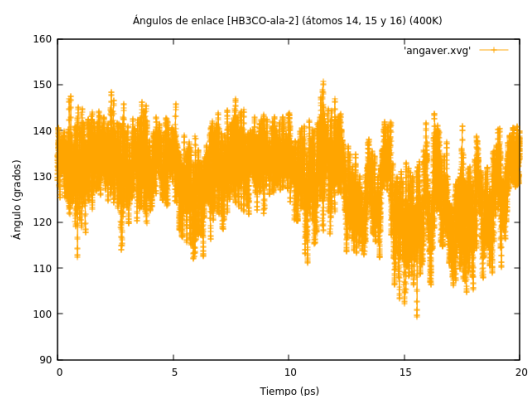
```
gnuplot> set xlabel 'Tiempo (ps)'
```

```
gnuplot> set title 'Ángulo de enlace [HB3CO-ala-2] (átomos 14, 15 y 16) (298K)'
```

```
gnuplot> replot
```

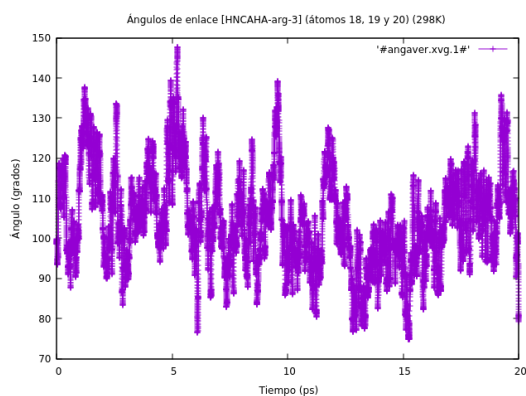


(a) Simulado a 298K

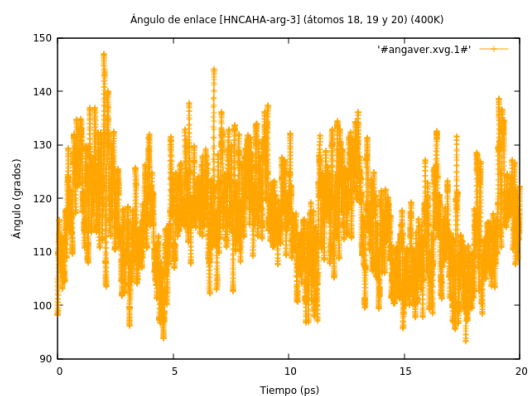


(b) Simulado a 400K

Figura 6: Ángulo de enlace [HB3CO-ala-2] (hidrógeno, de enlace de hidrógeno-carbono-oxígeno), átomos 14, 15 y 16



(a) Simulado a 298K



(b) Simulado a 400K

Figura 7: Ángulo de enlace [HNCAHA-arg-3] (hidrógeno-carbono aromático-hidrógeno no polar), átomos 18, 19 y 20

6.4. Ángulos dihedros

#Residuo del aminoácido Alanina

#Cálculo de los ángulos diedro phi y psi con el módulo rama de GROMACS

```
> gmx rama -f arf.trr -s arf.tpr -xvg none
```

#Este archivo que ha salido como output contiene los ángulos dihedros phi y psi para los tres residuos

```
> less rama.xvg
```

```
> 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
```

#Graficamos los ángulos diedros para el residuo de alanina del tripéptido ARF

```
> gnuplot
```

```
gnuplot> plot 'phi-ala-2.dat' with linespoints
```

Warning: slow font initializationqt_processTermEvent received a GE_fontprops event. This should not h

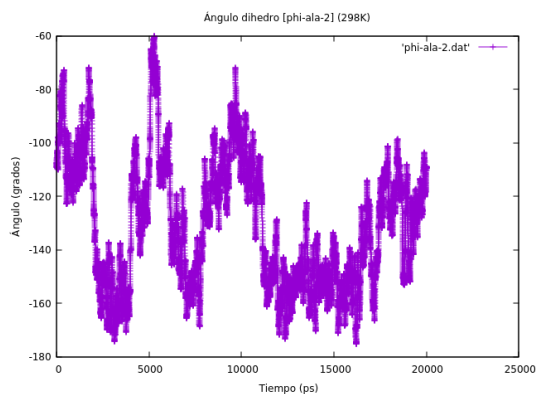
```
gnuplot> set xlabel 'Tiempo (ps)'
```

```
gnuplot> set title 'Ángulo dihedro [phi-ala-2] (298K)'
```

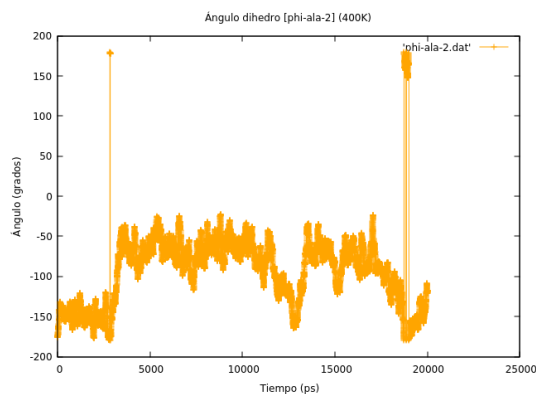
```
gnuplot> set ylabel 'Ángulo (grados)'
```

```
> join phi-ala-2.dat psi-ala-2.dat > ala-2.dat
```

```
> less ala-2.dat
```



(a) Simulado a 298K



(b) Simulado a 400K

Figura 8: Ángulos dihedro ϕ para el residuo de alanina

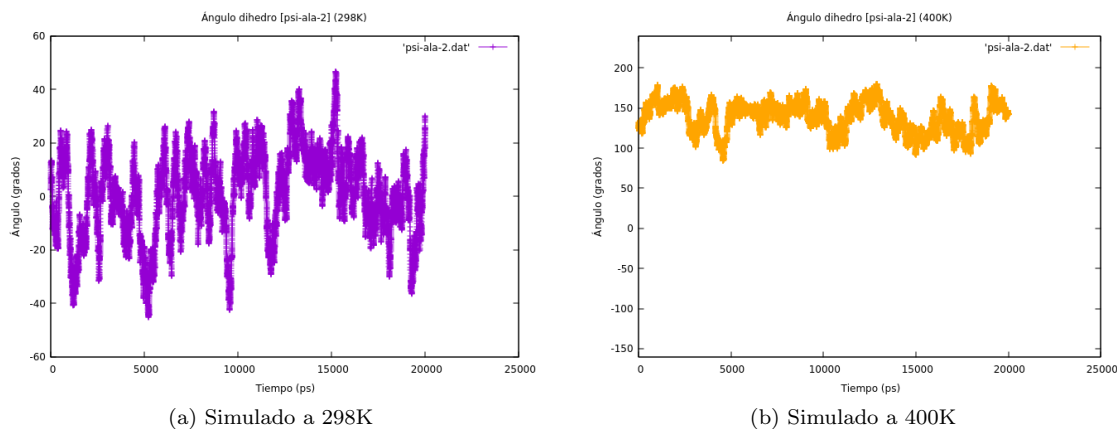


Figura 9: Ángulos dihedro ψ para el residuo de alanina

6.5. Temperatura

*#El módulo traj de GROMACS nos permite con el flag -ot permite generar gráficas
#de temperatura si le proporcionamos las velocidades.*

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

#Seleccionamos el grupo 0

#Obtenemos como output un fichero temp.xvg

#Graficamos con gnuplot

#Serie temporal de la temperatura

```
>gnuplot
```

```
gnuplot> plot 'temp.xvg' with linespoints
```

#Barplot

```
gnuplot> plot 'temp.xvg' with linespoints
```

```
gnuplot> width = 2
```

```
gnuplot> hist(x,width)=width*floor(x/width)+width/2.0
```

```
gnuplot> set boxwidth width*0.9 gnuplot> set style fill solid 0.5
```

```
gnuplot> plot 'temp.xvg' u (hist($2,width)):(1.0) smooth freq w boxes lc rgb'green' notitle
```

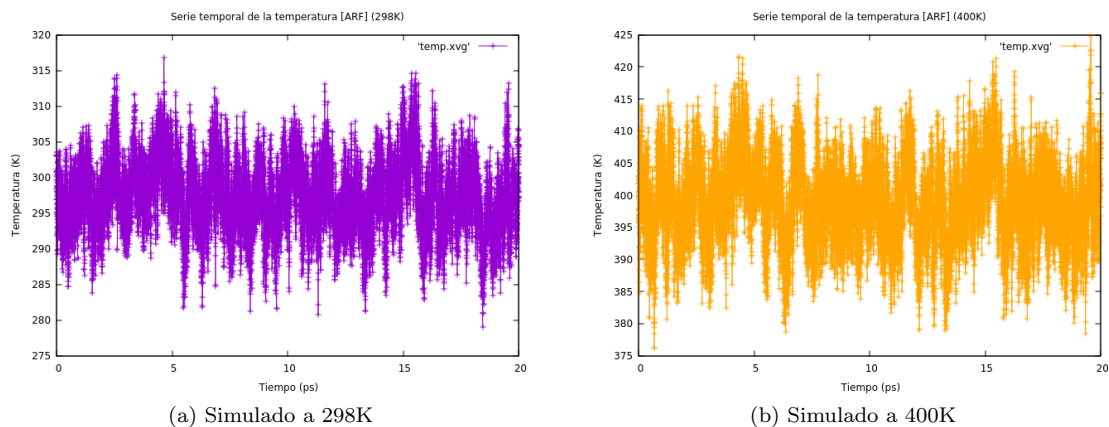


Figura 10: Serie temporal de la temperatura del tripéptido ARF

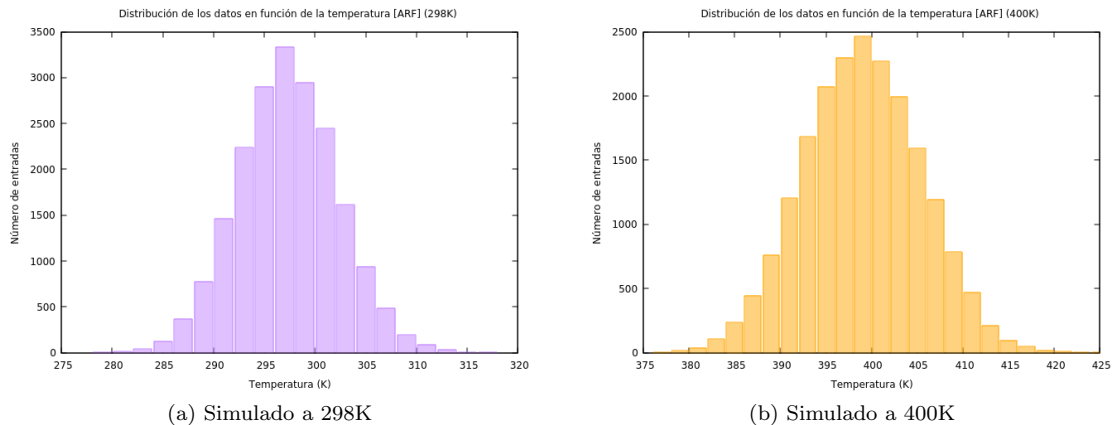


Figura 11: Distribución de los datos de temperatura simulados para el tripéptido ARF

Histogramas de la temperatura. Se asemejan a una distribución gaussiana en la que la moda es 298 en el primero, y 400 en el segundo (valor esperado y amplitud razonable).

6.6. Velocidades

```
> gmx traj -f arf.trr -s arf.tpr -xvg none -ov -len
#Graficamos velocidades
>gnuplot
gnuplot> plot 'veloc.xvg' using 1:5

#Graficamos las velocidades para los 5 primeros átomos del tripéptido ARF

#Gráficas de las velocidades de los 5 primeros átomos 298K

veloc298 <- read.delim("C:/Users/guill/Downloads/veloc.xvg", header=FALSE)

hist(x=veloc298$V5, main="Evolución velocidad de átomo 1",
     xlab = "Velocidad (nm/ps)", ylab = "Frecuencia",
     col="purple")
hist(x=veloc298$V9, main="Evolución velocidad de átomo 2",
     xlab = "Velocidad (nm/ps)", ylab = "Frecuencia",
     col="purple")
hist(x=veloc298$V13, main="Evolución velocidad de átomo 3",
     xlab = "Velocidad (nm/ps)", ylab = "Frecuencia",
     col="purple")
hist(x=veloc298$V17, main="Evolución velocidad de átomo 4",
     xlab = "Velocidad (nm/ps)", ylab = "Frecuencia",
     col="purple")
hist(x=veloc298$V21, main="Evolución velocidad de átomo 5",
     xlab = "Velocidad (nm/ps)", ylab = "Frecuencia",
     col="purple")

#Gráficas de las velocidades de los 5 primeros átomos 400K
veloc400 <- read.delim("C:/Users/guill/Downloads/veloc.xvg", header=FALSE)

hist(x=veloc400$V5, main="Evolución velocidad de átomo 1",
     xlab = "Velocidad (nm/ps)", ylab = "Frecuencia",
     col="orange")
```



```

hist(x=veloc400$V9, main="Evolución velocidad de átomo 2",
     xlab = "Velocidad (nm/ps)", ylab = "Frecuencia",
     col="orange")
hist(x=veloc400$V13, main="Evolución velocidad de átomo 3",
     xlab = "Velocidad (nm/ps)", ylab = "Frecuencia",
     col="orange")
hist(x=veloc400$V17, main="Evolución velocidad de átomo 4",
     xlab = "Velocidad (nm/ps)", ylab = "Frecuencia",
     col="orange")
hist(x=veloc400$V21, main="Evolución velocidad de átomo 5",
     xlab = "Velocidad (nm/ps)", ylab = "Frecuencia",
     col="orange")

```

A continuación, se puede apreciar en las velocidades para los cinco primeros átomos del tripéptido a dos temperaturas (298K y 400K) que los datos siguen una distribución normal, como era de esperar; y que, además, la distribución se desplaza hacia la derecha cuando la temperatura en el sistema es mayor. Esto se debe al incremento de la energía interna del sistema que provoca una mayor agitación atómica.

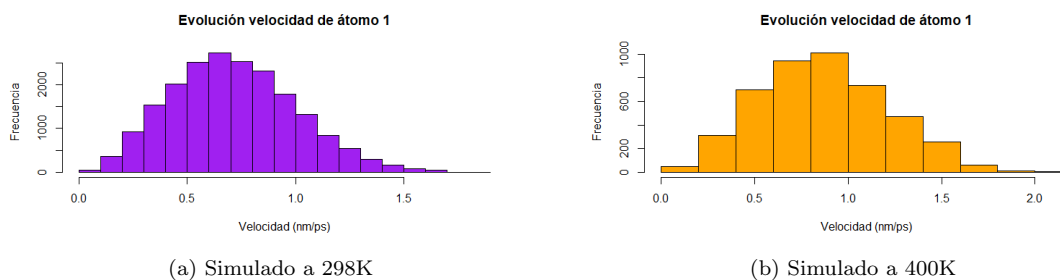


Figura 12: Velocidad del átomo 1 (Carbono alifático sp^3 del grupo CH_3) del tripéptido ARF

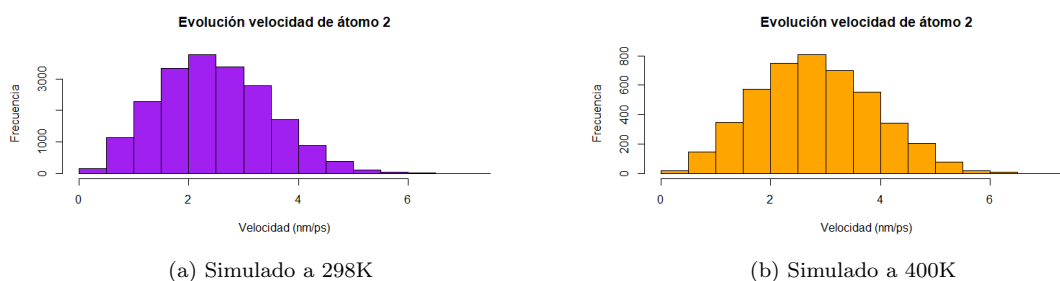
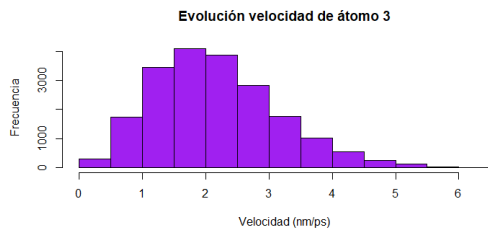
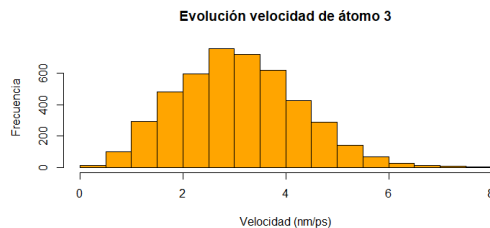


Figura 13: Velocidad del átomo 2 (Hidrógeno no polar) del tripéptido ARF

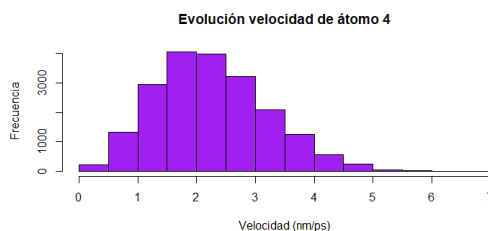


(a) Simulado a 298K

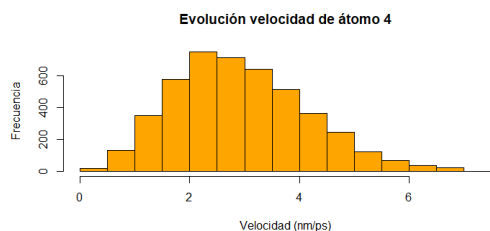


(b) Simulado a 400K

Figura 14: Velocidad del átomo 3 (Hidrógeno no polar) del tripéptido ARF

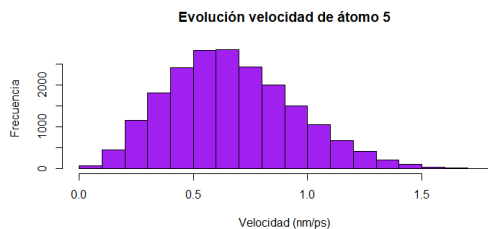


(a) Simulado a 298K

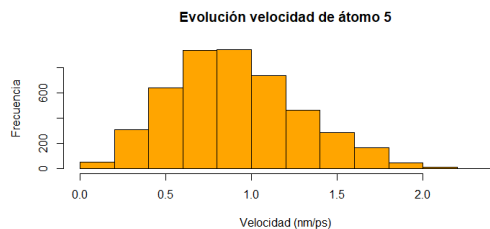


(b) Simulado a 400K

Figura 15: Velocidad del átomo 4 (Hidrógeno no polar) del tripéptido ARF



(a) Simulado a 298K



(b) Simulado a 400K

Figura 16: Velocidad del átomo 1 (carbonilo C') del tripéptido ARF

6.7. Energía total

La temperatura es un indicador de la energía interna de un sistema. Como se puede ver a continuación, cuando se incrementa la temperatura del sistema (hay más energía en el sistema, lo que produce mayor agitación atómica), aumenta la energía total del mismo.

```
> gmx energy -f arf.edr -s arf.tpr -xvg none
```

```
select 15
```

Output 298K:

Energy	Average	Err.Est.	RMSD	Tot-Drift
<hr/>				
Total Energy	-25903.7	65	267.024	-219.112 (kJ/mol)

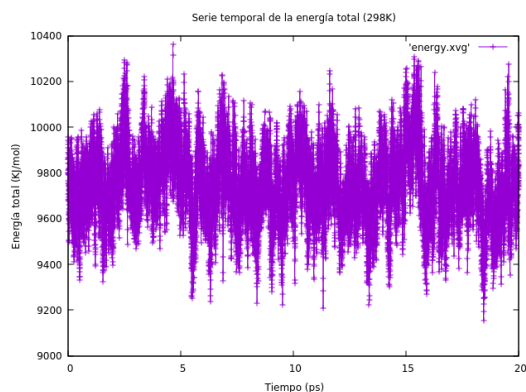
Output 400K:

Energy	Average	Err.Est.	RMSD	Tot-Drift
--------	---------	----------	------	-----------

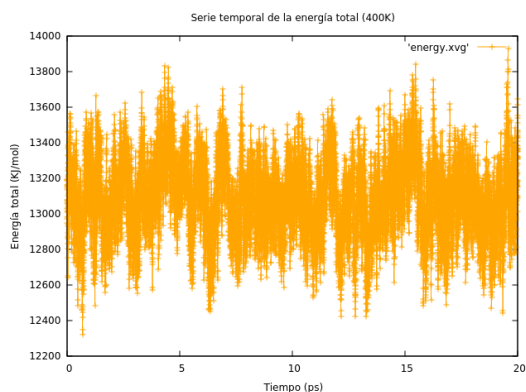
Total Energy	-17356.7	40	314.592	-199.113	(kJ/mol)
--------------	----------	----	---------	----------	----------

#Graficamos

```
> gnuplot
gnuplot> plot 'energy.xvg' with linespoints lt rgb"orange"
gnuplot> set title 'Serie temporal de la energía total (400K)'
gnuplot> set ylabel 'Energía total (KJ/mol)'
gnuplot> set xlabel 'Tiempo (ps)'
gnuplot> replot
```



(a) Simulado a 298K



(b) Simulado a 400K

Figura 17: Serie temporal de la energía total del sistema

6.8. Energía cinética

#El módulo energy de gromacs nos permite entre otras cosas calcular la energía cinética del sistema

```
> gmx energy -f arf.edr -s arf.tpr -xvg none
```

Select Group 14 (Kinetic-En.)

Output 298K:

Energy	Average	Err.Est.	RMSD	Tot-Drift	
Kinetic En.	9747.94	11	159.295	-51.1733	(kJ/mol)

Output 400K:

Energy	Average	Err.Est.	RMSD	Tot-Drift	
Kinetic En.	13081.5	13	208.379	-39.1202	(kJ/mol)

Análisis de la energía cinética según la termodinámica estadística

Según la termodinámica estadística, podemos calcular la energía de un sistema con la siguiente expre-

sión:

$$E_c = N \times \frac{3}{2} \times k_B \times T$$

Energía cinética del sistema a 298K:

$$k_B \times T = 1,38 \times 10^{-23} \frac{J}{moléculas \times K} \times 298K \times 6,022 \times 10^{23} \times \frac{moléculas}{mol} = 2,4765 \times 10^3 \times \frac{J}{mol}$$

$$E_c = 2629 \times \frac{3}{2} \times 2,4765 \times 10^3 \frac{J}{mol} = 9,766 \times 10^6 \frac{J}{mol} = 9,766 \times 10^3 \frac{KJ}{mol}$$

Energía cinética del sistema a 400K:

$$k_B \times T = 1,38 \times 10^{-23} \frac{J}{moléculas \times K} \times 400K \times 6,022 \times 10^{23} \times \frac{moléculas}{mol} = 3,3241 \times 10^3 \times \frac{J}{mol}$$

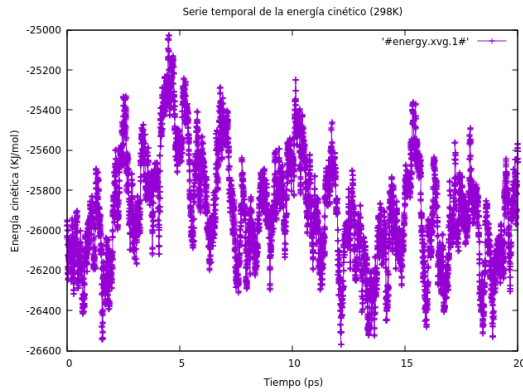
$$E_c = 2629 \times \frac{3}{2} \times 3,3241 \times 10^3 \frac{J}{mol} = 1,31085 \times 10^7 \frac{J}{mol} = 1,31085 \times 10^4 \frac{KJ}{mol}$$

Como se puede comprobar, los resultados predichos por la termodinámica estadística sobre la energía cinética para el sistema en el que se encuentra el tripéptido se asemejan bastante a los resultados obtenidos con las simulaciones de GROMACS para temperaturas de 298K y 400K:

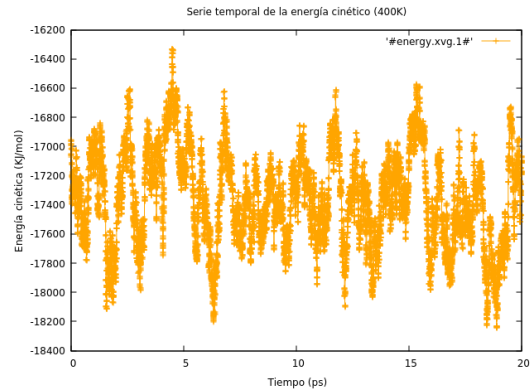
Temperatura (K)	E_c con GROMACS	E_c con Termodinámica Estad.
298	$9,748 \times 10^3 \frac{KJ}{mol}$	$9,766 \times 10^3 \frac{KJ}{mol}$
400	$1,3081 \times 10^4 \frac{KJ}{mol}$	$1,31085 \times 10^4 \frac{KJ}{mol}$

#Graficamos con gnuplot

```
>gnuplot
gnuplot> set title 'Serie temporal de la energía cinético (298K)'
gnuplot> set ylabel 'Energía cinética (KJ/mol)'
gnuplot> set xlabel 'Tiempo (ps)'
gnuplot> plot '#energy.xvg.1#' with linespoints
```



(a) Simulado a 298K



(b) Simulado a 400K

Figura 18: Serie temporal de la energía cinética del sistema

7. Informe avanzado

#Creamos un nuevo directorio para esta tarea

```
> mkdir ../3-run-Ext-Repo
> cp ../3-run/arf.top .
> cp ../3-run/arf.g96 .
```

```
> gmx grompp -f runNVT.mdp -c arf.g96 -p arf.top -o arf.tp
> cp ../3-run/submit_eck.sh .
#Modificamos el fichero submit-eck.sh con nano para poner el nuevo path

> cat submit_eck.sh

#!/bin/bash
#
#SBATCH -p eck-q
#SBATCH --chdir=/home/alumno23/dinamica_molecular/informe_personal/3-run-Ext-Repo
#SBATCH -J equilibrado
#SBATCH --cpus-per-task=1

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

Creamos las gráficas de Ramachandran (ángulos dihedros) para los tres residuos de ARF

*#Podemos echar un vistazo al fichero rama.xvg para ver como seleccionar el aminoácido
#y sus respectivas filas*

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

> 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

> join phi-ala-2.dat psi-ala-2.dat > ala-2.dat
```

```
#GRÁFICA CON R
#Cargamos el paquete
library(MDplot)
#Leemos el fichero
ala.2 <- read.table("C:/Users/guill/OneDrive/Escritorio/arf/ala-2.dat",
quote="\\"", comment.char="")

#Convertimos la tabla en dataframe
ala.2 <- as.data.frame(ala.2)

#Eliminamos la columna que enumera las filas
ala.2$V1 = NULL

#Gramicamos con la función ramachandran()
ramachandran(ala.2, printLegend = TRUE)
```

Gráfica de Ramachandran

Trata de representar, mediante los ángulos dihedros (Φ y Ψ), las configuraciones más probables que puede adoptar un polipéptido en base a su secuencia de residuos. Cabe destacar que cada residuo puede explorar conformaciones diferentes.

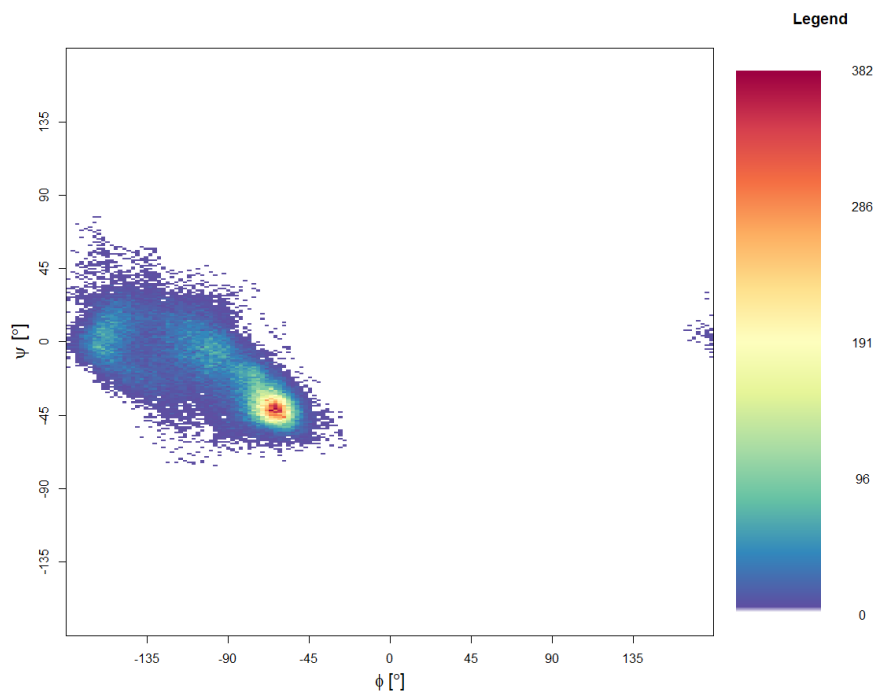


Figura 19: Gráfica de Ramachandran para el residuo de alanina del tripéptido ARF.

En la gráfica para el residuo de alanina se puede observar que el residuo parece adoptar una estructura hélice- α

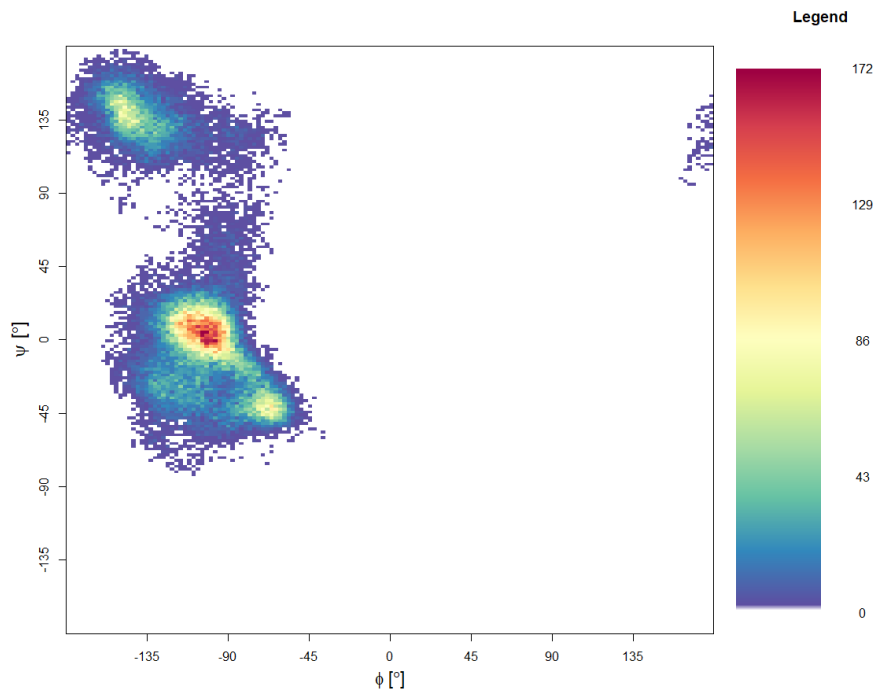


Figura 20: Gráfica de Ramachandran para el residuo de arginina del tripéptido ARF.

En la gráfica para el residuo de arginina se puede observar que el residuo parece tener como conformaciones posibles la hoja- β y la estructura hélice- α , siendo la segunda la más probable de producirse.

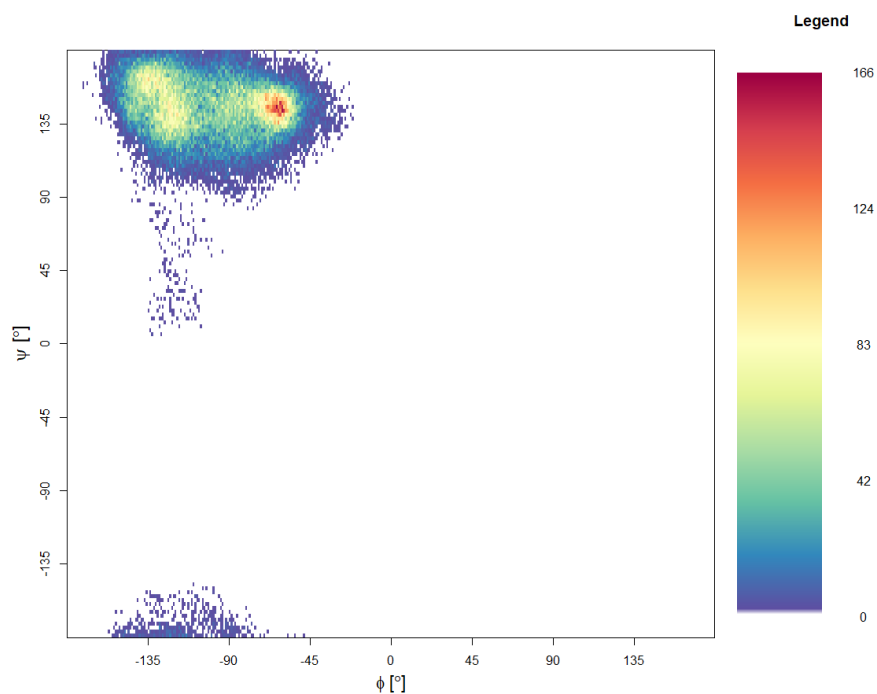


Figura 21: Gráfica de Ramachandran para el residuo de fenilalanina del tripeptido ARF.

En la gráfica para el residuo de fenilalanina se puede observar que el residuo parece adoptar una estructura hoja- β .