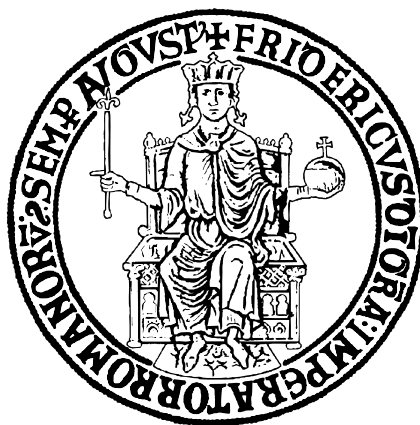


UNIVERSITÀ DEGLI STUDI DI NAPOLI  
FEDERICO II



AREA DIDATTICA DI SCIENZE MATEMATICHE,  
FISICHE E NATURALI

DIPARTIMENTO DI BIOLOGIA

CORSO DI LAUREA MAGISTRALE IN BIOLOGIA

TESI DI LAUREA SPERIMENTALE IN  
BIOCHIMICA

Sviluppo di un Sistema di Dinamica Molecolare per lo Studio dei  
Peptidi in Frazioni Variabili di Trifluoroetano-  
lo: Uno Strumento Versatile per Ricerca e Didattica

**Relatore:**

Dr. Bruno Hay Mele

**Candidato:**

Guerriero Simone

**Matricola:**

N92002267

ANNO ACCADEMICO 2022/2023

# Indice

<b>I</b>	<b>Introduzione</b>	<b>3</b>
<b>1</b>	<b>Peptidi Antimicrobici</b>	<b>6</b>
1.1	Struttura . . . . .	6
1.2	Meccanismo d'Azione . . . . .	8
1.3	Solvente: Trifluoroetanolo . . . . .	10
<b>2</b>	<b>Dinamica Molecolare</b>	<b>13</b>
2.1	Dinamica Molecolare Classica . . . . .	14
2.1.1	Campo di forza: OPLS . . . . .	26
<b>II</b>	<b>Materiali e metodi</b>	<b>28</b>
<b>3</b>	<b>GROMACS</b>	<b>29</b>
3.1	GROMACS_installation.ipynb . . . . .	31
3.1.1	File per la solvatazione . . . . .	31
3.2	Molecular_Dynamics_Simulation_with_Trifluoroethanol.ipynb	39
3.2.1	Parametri di Dinamica Molecolare . . . . .	43
3.3	trajectory_analysis.ipynb . . . . .	44
<b>III</b>	<b>Risultati e conclusione</b>	<b>46</b>
<b>4</b>	<b>Risultati</b>	<b>47</b>
4.1	RMSD . . . . .	49
4.2	Raggio d'Inerzia . . . . .	53

<i>INDICE</i>	2
4.3 Principal Component Analysis . . . . .	58
<b>5 Discussione</b>	<b>62</b>
<b>A Ensemble</b>	<b>64</b>
A.1 Ensemble microcanonico . . . . .	65
A.2 Ensemble canonico . . . . .	66
A.3 Ensemble isotermico-isobarico . . . . .	68
<b>B Jupyter Notebook: Coefficienti Ryckaert-Bellemans</b>	<b>70</b>
<b>C Dinamica: Diagramma di Flusso</b>	<b>77</b>
<b>D Script Bash: Simulazione 100ns in Acqua</b>	<b>79</b>

# Parte I

## Introduzione

I farmaci antimicrobici vengono adoperati per prevenire e curare infezioni batteriche, fungine, virali e parassitarie che possono colpire tanto le piante quanto gli animali, uomo compreso. L'utilizzo inappropriato di questi farmaci in agricoltura, in zootecnica e in medicina ha favorito la nascita di microrganismi resistenti agli agenti antimicrobici.

La resistenza antimicrobica (AMR) degli organismi patogeni rende difficile curare un paziente affetto da un'infezione comune e aumenta, al contempo, il rischio di diffusione di queste malattie.

La diffusione mondiale di batteri multi-resistenti e pan-resistenti, ovvero non trattabili con gli antibiotici, grava fortemente sui sistemi sanitari nazionali perché prolunga la degenza dei pazienti all'interno degli ospedali e rende più rischiosi interventi chirurgici come i parti cesarei e trapianti d'organo a causa delle infezioni nosocomiali.

Il World Economic Forum pubblicò nel 2013 un report sui rischi globali, tra questi compare la resistenza agli antibiotici, la quale fu definita come un rischio che non può essere mitigato dalle sole capacità di una qualsiasi organizzazione o nazione e per questo necessita di attenzione a livello globale ed uno sforzo nell'arginare questa problematica da parte di tutte nazioni.

Al fine di rallentare la diffusione dei microrganismi resistenti, nel 2015 durante l'Assemblea Mondiale della Sanità sono state stilate le linee guida del Global Action Plan on antimicrobial resistance (GAP), grazie alle quali sono stati evidenziati i rischi e le procedure da attuare per contrastare questo fenomeno[105].

Nel GAP viene descritto come l'AMR interessa aspetti che influiscono sulla vita quotidiana delle persone. Questi vengono descritti all'interno del capitolo "The Challenge", in particolare nel paragrafo 12 sono affrontate le conseguenze sulle malattie comuni mentre nel paragrafo 16 sono evidenziati gli effetti negativi dell'AMR sulle spese domestiche:

"12. I microrganismi che causano molte malattie umane comuni e condizioni mediche – tra cui tubercolosi, HIV/AIDS, malaria, malattie sessualmente trasmissibili, infezioni del tratto urinario, pol-

monite, infezioni del flusso sanguigno e intossicazione alimentare — sono diventati resistenti a un’ampia gamma di farmaci antimicrobici. I medici devono utilizzare sempre più farmaci di “ultima istanza”, che sono più costosi, possono avere più effetti collaterali.”

“16. [...] la perdita dell’efficacia degli agenti antimicrobici per trattare gli animali malati danneggia la produzione di cibo e il sostentamento delle famiglie [...] gli agricoltori che lavorano con bovini, suini e pollame infetti da *Staphylococcus aureus* resistenti alla meticillina<sup>1</sup> corrono un rischio molto più elevato di essere colonizzati o infettati da questi batteri.”

Nonostante l’applicazione di questo piano da parte dell’organizzazione mondiale della sanità, è stato stimato che nel 2019, a causa di 33 differenti specie di batteri patogeni, sono morte circa otto milioni di persone ovvero il 13.6% di tutte le morti globali[48].

Per contrastare l’AMR la comunità scientifica ha impiegato i propri sforzi per sviluppare terapie innovative come peptidi antimicrobici, nuovi tipi di antibiotici, vaccini e terapie fagiche specifiche contro le specie multi-resistenti e pan-resistenti.

Le terapie con peptidi antimicrobici sono state approvate dalla Food and Drug Administration per l’utilizzo in ambito clinico contro:

- le polmoniti pediatriche causate da infezioni stafilococciche;
- le infezioni da virus come l’HIV e l’epatite C;
- le infezioni cutanee di diversi patogeni batterici. [34]

L’interesse per questa tematica, di impatto globale, mi ha spinto a realizzare un codice informatico che permette la produzione di simulazioni di dinamica molecolare in ambienti simil-membrana al fine di analizzare e comprendere ulteriormente i peptidi antimicrobici.

---

<sup>1</sup>antibiotico  $\beta$ -lattamico a spettro ristretto, inibisce la sintesi delle pareti cellulari batteriche.

# Capitolo 1

## Peptidi Antimicrobici

I peptidi antimicrobici (AMP) rappresentano una classe diversificata di molecole del sistema immunitario innato. Essi svolgono un ruolo fondamentale come prima linea di difesa contro agenti patogeni invasivi. Negli ultimi anni l'interesse verso gli AMP è aumentato, grazie al loro potenziale impatto clinico e un possibile utilizzo nella filiera alimentare contro patogeni che hanno sviluppato una resistenza agli agenti antimicrobici classici.

L'attività antimicrobica di questi peptidi è selettiva ed efficace contro batteri infettivi (sia Gram positivi che Gram negativi), virus e funghi[3]. Inoltre, in virtù del basso peso molecolare hanno una bassa immunogenicità per l'ospite dell'infezione. Infine è emerso da esperimenti condotti in vivo su ceppi BALB/c di *Mus musculus* che gli AMP hanno una bassa tossicità sia se somministrati oralmente che tramite iniezione peritoneale[23, 45]. Questi e altri studi clinici hanno permesso di sviluppare più di sessanta farmaci peptidici, con diverse centinaia di molecole attualmente in fase di studio pre-clinico o trial clinico.

### 1.1 Struttura

La scoperta degli AMP risale al 1939, quando il microbiologo René Dubos estrasse un agente antimicrobico da un ceppo di *Bacillus* dal suolo[9]. Ad oggi

i peptidi antimicrobici sono stati ritrovati in molti tipi di cellule e tessuti di organismi quali piante, insetti, anfibi e in organismi superiori.

La composizione amminoacidica e le caratteristiche chimico-fisiche correlate alla struttura consentono di interagire in maniera selettiva con il doppio strato lipidico delle membrane dei microrganismi causandone la morte.

A tal proposito è importante che la sequenza primaria di questi peptidi sia composta dal 50% circa di residui idrofobici[101] e da un numero di residui carichi (acidi o basici) tali da generare una carica netta. Oltre alla composizione è fondamentale l'ordine di sequenza affinché il peptide, una volta nella sua conformazione nativa, risulti anfipatico. Queste proprietà hanno un ruolo fondamentale per quanto riguarda l'attività, l'efficacia e la selettività [32, 49].

La lunghezza degli AMP può variare da cinque a oltre cento amminoacidi[3] a seconda della classe considerata. La classificazione in questione viene realizzata in base alla presenza o meno delle strutture secondarie nel fold[28, 38]:

1. Le strutture ad  $\alpha$ -elica lineari sono ricche in alanina, lisina e leucina, che stabilizzano l'elica. Sono tipicamente costituite da dodici a quaranta residui[100];
2. Le strutture formate da foglietti- $\beta$  contengono residui di cisteina che stabilizzano l'oligopeptide in soluzione formando uno o più ponti disolfuro[80];
3. I peptidi che presentano entrambi i tipi di struttura secondarie hanno generalmente da tre a cinque ponti disolfuro[54];
4. I peptidi che non presentano né  $\alpha$ -eliche né foglietti- $\beta$  sono ricchi in glicine, proline, triptofani o istidine. Queste non producono strutture in soluzione e neppure durante l'interazione con le membrane[54];
5. I peptidi ciclici e quelli con strutture topologicamente complesse, stabilizzano la propria struttura con la formazione di ponti disolfuro o ponti tioetere[78][95].



Per migliorare la selettività e l'efficacia degli AMP viene applicato il rational design che permette di modificare la sequenza primaria in modo da influire sulle proprietà chimico-fisiche del peptide[9].

## 1.2 Meccanismo d'Azione

L'interazione tra AMP e membrana coinvolge sia le interazioni ioniche che quelle idrofobiche. Il processo di riconoscimento iniziale tra il peptide e la cellula target è principalmente di natura elettrostatica, mentre l'inserimento effettivo all'interno della membrana avviene grazie ai residui idrofobici che stabiliscono contatti con le code lipidiche, facilitando così l'immersione del peptide nel bilayer[68].

Dato che il riconoscimento avviene per l'affinità di carica tra il peptide e la membrana risulta difficile, da parte dei patogeni, lo sviluppo di una qualsiasi forma di resistenza. Modificare la composizione della membrana plasmatica è un evento che può avvenire solo in tempi lunghi[86, 46].

È possibile discriminare gli AMP in base al loro meccanismo d'azione. Alcuni esplicano la loro funzione direttamente sulla membrana plasmatica mentre altri agiscono all'interno della cellula bersaglio.

Il gruppo che agisce sul bilayer è composto soprattutto da peptidi cationici come le defensine e LL-37. Piuttosto che inibire la sintesi del peptidoglicano — come gli antibiotici beta-*lattamici*[56, 60]— essi causano la rottura dell'involucro esterno inducendo la formazione di un poro transmembrana[37].

Una caratteristica peculiare di questo gruppo è la capacità di provocare la morte cellulare dopo pochi secondi dal riconoscimento[64] della membrana esterna dei procarioti, questa possiede carica negativa a causa della presenza del lipopolisaccaride o per la presenza di acidi tecoici.

Quando gli AMP cationici interagiscono con il layer esterno delle membrane procariotiche[91], si ripiegano producendo una struttura secondaria definita e destabilizzano la membrana aumentandone la permeabilità[65].

I meccanismi d'azione degli AMP attivi sulla membrana vengono divisi per il modello d'interazione:

- Carpet-like (o detergent-like), i peptidi circondano una porzione a doppio strato della membrana formando una micella che produce un vero e proprio buco in essa[88, 5, 8].
- Poro toroidale, i peptidi si allineano perpendicolarmente alla struttura lipidica formando un poro con le regioni idrofobiche del peptide rivolte verso le code lipidiche mentre la parte carica/idrofilica è rivolta verso il centro del poro[12, 106].
- Doga di botte, le doghe (ovvero i singoli AMP) si allineano parallelamente alla membrana cellulare, poi, per le interazioni idrofobiche, si inseriscono perpendicolare alla membrana formando la struttura a botte[109, 27, 97].

Alcuni AMP agiscono diversamente rispetto a quanto scritto sinora, non danneggiano la membrana, ma vengono traslocati all'interno del citoplasma e interferiscono con il normale funzionamento cellulare.

I meccanismi che permettono la traslocazione all'interno del citoplasma sono due: la penetrazione diretta e l'endocitosi[68]. L'assorbimento per endocitosi può avvenire per macropinocitosi oppure per endocitosi mediata da recettori. Per quanto riguarda la macropinocitosi la membrana cellulare produce delle invaginazioni fino a generare delle vescicole — chiamate macropinosomi — grazie all'azione della dinamina, una proteina GTPasi[68, 51]. L'endocitosi mediata da recettori avviene per mezzo delle proteine, clatrina o caveolina, presenti all'interno di leggere inflessioni della membrana da cui dipartono le vescicole[74, 68].

Una volta dentro la cellula target gli AMP possono[20, 55, 69, 65, 92, 1, 81, 44, 93]:

- inibire la sintesi di proteine, acidi nucleici e della parete cellulare;
- bloccare l'attività enzimatica dell'RNA polimerasi e di alcune proteasi;

- attivare l'autolisina, un enzima che agisce rompendo il legame 1-4 $\beta$  glicosidico tra l'N-acetil glucosammina e l'acido N-muraminico del peptidoglicano, provocandone la morte per autolisi[15, 14].

Buona parte degli AMP non presenta una struttura specifica quando sono in soluzione, ma si strutturano all'interazione con la membrana cellulare del microrganismo target. Poiché la loro struttura può essere apprezzata solo quando si trovano in un ambiente simil-membrana, per studiarli sperimentalmente si adottano diverse metodi:

- il peptide viene immesso in una miscela d'acqua e trifluoroetanolo;
- viene aggiunto in soluzione un tensioattivo, ad esempio il sodio dodecil-solfato (SDS);
- si induce l'interazione con micelle o liposomi[7, 12].

### 1.3 Solvente: Trifluoroetanolo

Il trifluoroetanolo (TFE) è un composto organico con formula bruta  $\text{CF}_3\text{CH}_2\text{OH}$ , un derivato dall'etanolo in cui tre degli atomi di idrogeno sono sostituiti con atomi di fluoro.

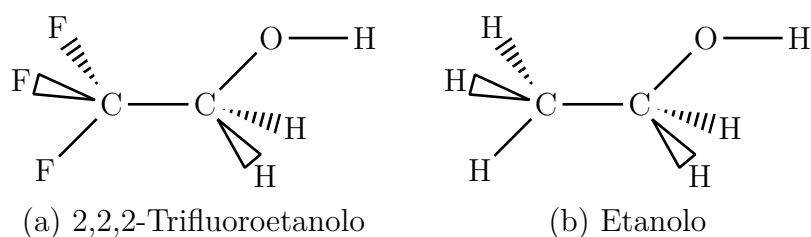


Figura 1.1: Confronto tra TFE ed etanolo

In biochimica il TFE è adoperato principalmente come solvente, come possiamo leggere su DrugBank Online, alla pagina dedicata:

“Un co-solvente non acquoso che funge da strumento per lo studio del ripiegamento delle proteine.”

Il TFE può indurre conformazioni stabili nei peptidi che in acqua non posseggono folding nativo[107, 70, 13, 35, 59]. È un solvente meno polare e più idrofobico rispetto a  $H_2O$ , la sua costante dielettrica, un terzo di quella dell'acqua, risulta molto più vicina al valore interno delle proteine. Agisce perturbando indirettamente le interazioni acqua-peptide concorrendo per la formazione di legami idrogeno con il backbone e le catene laterali delle proteine[89, 33, 26, 103], questo produce a sua volta un rafforzamento delle interazioni tra gruppi carichi.

Dal momento che è stata riscontrata una correlazione tra la struttura indotta da TFE e le frazioni  $\alpha$ -elica transmembrana, la maggior parte degli esperimenti su peptidi e frazioni di proteine — che corrispondono a regioni strutturate elicoidalmente all'interno della struttura terziaria — vengono effettuate in soluzioni acquose in presenza di TFE come co-solvente[66, 62, 50, 25, 84, 77].

In questi casi, il TFE viene spesso utilizzato per studiare l'influenza dell'ambiente sulle strutture proteiche condizionando la struttura e la stabilità delle proteine e dei peptidi durante gli esperimenti di dicroismo circolare<sup>1</sup> (CD). Inoltre, viene adoperato per l'ottenimento di modelli tridimensionali tramite la spettroscopia di risonanza magnetica nucleare (NMR)<sup>2</sup>, utili per studiare la struttura e la dinamica delle proteine.

Esperimenti hanno evidenziato che, in presenza di TFE, il contenuto di  $\alpha$ -elica nel peptide è maggiore rispetto a quando è in acqua pura. Questi prevedono la registrazione dello spettro di assorbimento delle radiazioni ultraviolette (UV-C) di una soluzione di acqua pura contenente un peptide e l'aggiunta graduale, in piccole dosi, di TFE. Con l'aumento della concentrazione di TFE si osserva, nello spettro, la formazione di una buca intorno a 220 nm[85]. Questa banda negativa, nello spettro CD, è una caratteristica specifica delle conformazioni elicoidali[96]. Quindi con l'aggiunta di piccole quantità di TFE è possibile promuovere l'elicità in quei peptidi che, in soluzione acquosa, non

---

<sup>1</sup>tecnica spettroscopica che consente di studiare la struttura secondaria delle proteine, riportando la percentuale di elicità, di struttura a foglietto- $\beta$  e quella disordinata.

<sup>2</sup>la NMR produce dei modelli che rappresentano la media di tutte le conformazioni presenti durante l'acquisizione del dato (il processo dura qualche secondo).

potrebbero adottare nessuna conformazione, come gli AMP cationici.

È importante notare che l'uso del TFE nel CD deve essere accuratamente calibrato e controllato, e i risultati devono essere interpretati considerando attentamente l'effetto dell'additivo sulle proprietà strutturali della proteina, tuttavia concentrazioni eccessive possono alterare la struttura della proteina in modo non fisiologico.

Dato che in diversi articoli viene sostenuto che gli AMP cationici, che presentano una struttura  $\alpha$ -elica lineare, sono più performanti nel provocare la morte dei microrganismi target mediante shock osmotico[9, 67] e che, com'è stato ribadito in questo paragrafo, il TFE risulta capace di stabilizzare le strutture ad  $\alpha$ -eliche imitando gli ambienti simil-membrana, è stata avanzata l'ipotesi che la dinamica molecolare potesse essere vantaggiosa per la progettazione e l'analisi degli AMP.

## Capitolo 2

# Dinamica Molecolare

Con il termine dinamica molecolare (MD) si fa riferimento a una serie di tecniche computazionali che consentono la simulazione del movimento e delle interazioni di atomi e molecole. Questi approcci permettono di comprendere le proprietà dei materiali, le reazioni chimiche e i processi biologici a livello molecolare.

I sistemi studiati in dinamica molecolare sono spesso costituiti da un vasto numero di molecole interagenti, e possono in linea di principio essere analizzati mediante l'integrazione delle equazioni del moto[90]. In pratica, tali sistemi sono troppo complicati per una trattazione analitica, che richiederebbe una capacità computazionale fuori standard. La risoluzione di questa problematica può avvenire in modi differenti in base alla tecnica di dinamica molecolare scelta. Tra i tanti metodi esistenti, riportiamo i due più importanti: la MD classica e la MD quantistica.

La tecnica classica utilizza metodi numerici: le traiettorie di atomi e molecole sono determinate da una integrazione numerica della dinamica newtoniana, in cui le forze tra le particelle e le loro energie potenziali vengono spesso calcolate utilizzando i campi di forza. Il metodo quantistico, invece, associa ad ogni particelle del sistema una funzione d'onda che consente di risolvere l'equazione di Schrödinger a molti corpi. La differenza sostanziale tra i metodi classici e quelli quantistici è che i primi ignorano completamente gli elettroni ed il loro movimento: i calcoli vengono effettuati solo in funzione della posi-

zione nucleare. Nonostante la perdita di informazione, il metodo classico può produrre risultati comparabili a quello quantistico, sebbene sia incapace di descrivere le proprietà che dipendono dalla distribuzione elettronica intorno ad una molecola.

## 2.1 Dinamica Molecolare Classica

La MD classica utilizza i principi della meccanica molecolare (MM) per calcolare l'equazione del moto di un sistema di particelle.

Alcuni degli utilizzi possibili della MD classica sono:

- rilassare le strutture cristallografiche ottenute con la diffrazione a raggi X;
- migliorare le strutture ottenute a partire da un homology modelling;
- dinamizzare le strutture e i modelli tridimensionali statici di macromolecole, ottenuti rispettivamente con la cristallografia a raggi X e con la spettroscopia NMR, per interpretare i risultati di esperimenti biofisici e osservare il comportamento delle proteine in un ambiente specifico;
- analizzare le interazioni tra molecole attraverso il docking, ovvero calcolare la posizione media degli amminoacidi fondamentali che interagiscono con il ligando permettendo di progettare farmacofori per sviluppare nuovi farmaci[87, 18];
- simulare il folding ab initio di una catena polipeptidica partendo dalla struttura primaria.

La dinamica molecolare classica si basa sul concetto di trasferibilità e sfrutta l'approssimazione di Born-Oppenheimer. Questo permette di utilizzare le informazioni derivate sperimentalmente o analiticamente sul comportamento delle molecole unitarie del sistema (solvente, amminoacidi, acidi nucleici, fino a piccoli polimeri) sfruttandone la maggiore accessibilità sperimentale [40], e di

considerare le coordinate dei nuclei come fisse<sup>1</sup>. Di conseguenza, è possibile calcolare l'energia potenziale del sistema in funzione delle coordinate nucleari e dei parametri atomici adoperando campi di forza che descrivono le molecole componenti il sistema.

In fisica, un campo di forza è un campo vettoriale che descrive la presenza di una forza in ogni punto dello spazio. Si tratta di una funzione che associa ad ogni posizione un vettore forza con una sua intensità e direzione.

Nell'ambito della chimica computazionale, con questa locuzione si fa riferimento a una funzione che calcola vari contributi energetici a partire da una serie di parametri[40, 24, 42, 63]:

- massa atomica e raggio di van der Waals;
- lunghezza di legame all'equilibrio;
- angoli di legame all'equilibrio per ogni triade di atomi;
- angoli dei diedri per ogni gruppo di quattro atomi legati tra loro;
- cariche parziali per ogni atomo in base alla molecola considerata.

I parametri in questione differiscono in base all'elemento della tavola periodica al gruppo funzionale a cui appartengono, perchè presentano caratteristiche chimico-fisiche differenti[22].

Esistono due tipologie di campi di forza: all-atom e united-atom (o coarse-grained):

I campi di forza all-atom presentano parametri per ogni tipo di atomo che si trova nel sistema, incluso l'idrogeno, mentre i campi di forza coarse-grained considerano gli atomi d'idrogeno e di carbonio, appartenenti ai gruppi metilici ( $-\text{CH}_3$ ) e ponti metilenici ( $-\text{CH}_2-$ ), come un unico centro di interazione[61, 98].

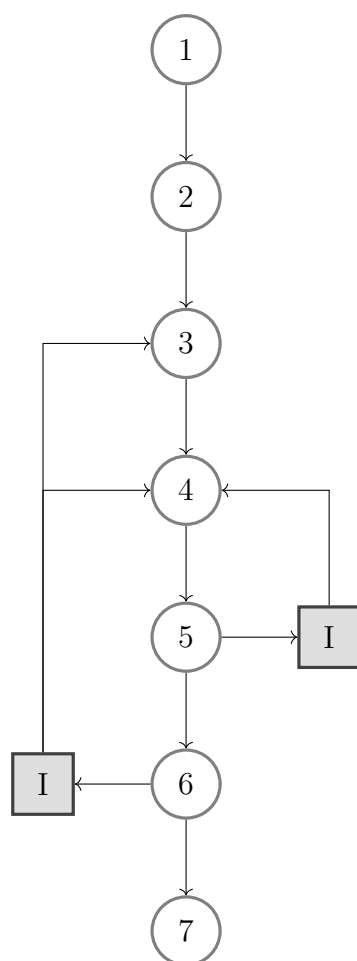
---

<sup>1</sup>Basandosi sull'approssimazione di Born-Oppenheimer la funzione d'onda dei nuclei atomici e quella degli elettroni in una molecola possono essere trattati separatamente dato che la massa dei nuclei risulta relativamente più grande rispetto a quella degli elettroni



Generalmente, per lunghe simulazioni che coinvolgono le macromolecole come proteine, acidi nucleici o complessi proteici, vengono adoperati campi di forza coarse-grained. Questo risulta essere il giusto compromesso tra perdita di dettaglio e la riduzione del tempo di simulazione[71].

La parametrizzazione del campo di forza presenta un flusso di lavoro ben specifico:



1. recupero di una struttura cristallografica o di una formula chimica;
2. definizione dei tipi atomici;
3. ottenimento di cariche atomiche;
4. assegnazione iniziale dei parametri di legame e quelli di Lennard-Jones;
5. test computazionali di densità e di geometria relativi a dati sperimentali di riferimento;
6. test computazionali di proprietà energetiche relative ai dati sperimentali di riferimento, come l'energia superficiale[43] e quelli d'idratazione[29];
7. validazione secondaria e perfezionamento delle proprietà termiche, meccaniche e di diffusione. [41]

Nel momento in cui i test computazionali non mostrano i risultati sperati, al fine di ottenere un campo di forza valido ed efficace, è necessario attuare dei cicli iterativi, ovvero ripetere il passaggio precedente al test e il test stesso

fino al raggiungimento di un risultato soddisfacente<sup>2</sup>.

La forma generale della funzione dell'energia potenziale adoperata dai campi di forza include il termine di legame e il termine di non-legame. Il termine di legame viene riferito alle interazioni degli atomi che sono legati l'uno all'altro tramite legami covalenti mentre il termine di non-legame, o non-covalente, descrive le forze di van der Waals e le interazioni elettrostatiche a lungo raggio.

$$E_{totale} = E_{legame} + E_{non-legame} \quad (2.1)$$

dove:

$$E_{legame} = E_{stretching} + E_{angolo} + E_{diedro} \quad (2.2)$$

$$E_{non-legame} = E_{elettrostatica} + E_{van\ der\ Waals} \quad (2.3)$$

In linea di principio se i valori calcolati deviano rispetto ai parametri, vengono applicate delle penalità energetiche, che provocano l'aumento dell'energia potenziale del sistema[61].

Negli approcci più semplici, i legami covalenti sono descritti come oscillatori armonici. Quando la distanza tra i due atomi devia dalla lunghezza all'equilibrio viene generata una forza di segno contrario rispetto al verso della deformazione, descritta con la legge di Hooke:

$$F = -k(x - x_0) \quad (2.4)$$

Di conseguenza, più la lunghezza di legame diverge rispetto a quella di equilibrio più aumenta l'energia, secondo l'equazione dell'energia potenziale elastica:

$$U = \frac{k}{2}(x - x_0)^2 \quad (2.5)$$

Il contributo energetico dato dall'alterazione della lunghezza di legame, come quello dato dalla deformazione degli angoli di legame, è plasmato sull'equazione 2.5:

$$E_{stretching} = \frac{k_{ij}}{2}(l_{ij} - l_{0,ij})^2 \quad (2.6)$$

$$E_{angolo} = \frac{k_{ijl}}{2}(\theta_{ijl} - \theta_{0,ijl})^2 \quad (2.7)$$

---

<sup>2</sup>per "soddisfacente" tipicamente si intende "al di sotto di un certo valore soglia".

dove:

- $k_{ij}$  e  $k_{ijl}$  sono le costanti di forza;
- $l_{ij}$  e  $\theta_{ijl}$  sono rispettivamente la lunghezza di legame e l'angolo di legame calcolati;
- $l_{0,ij}$  è il parametro della lunghezza del legame tra gli atomi  $i$  e  $j$  quando tutti gli altri termini del campo di forza sono pari a zero<sup>3</sup>;
- $\theta_{0,ijl}$  è il parametro relativo all'angolo di legame all'equilibrio.

La costante di allungamento  $k_{ij}$  è definita anche frequenza vibrazionale, e viene determinata a partire dallo spettro infrarosso (IR), dallo spettro di Raman<sup>4</sup> o tramite calcoli di meccanica quantistica.

La legge di Hooke risulta accurata solo se le distanze di legame tra due atomi deviano di poco dalla posizione di equilibrio. Viene adoperata nonostante i suoi limiti perché raramente queste distanze si allontanano significativamente dai parametri di riferimento. Nel caso in cui fosse necessaria un'accuratezza maggiore, è possibile impiegare il potenziale Morse al posto della legge di Hooke, che tiene conto anche dell'anarmonicità dei legami covalenti ma che riduce l'efficienza di calcolo[61, 99].

Dopo aver analizzato i contributi riguardanti le interazioni di legame a due e a tre corpi, tratteremo l'ultimo elemento, quello con interazioni a quattro corpi. In alcuni campi di forza questo termine descrive sia i diedri propri che quelli impropri (fig. 2.1).

Il contributo del diedro proprio è dato dalla funzione sinusoidale:

$$E_{diedro} = k_{\phi,ijlh} [1 + \cos(n\phi_{ijlh} - \delta_n)] \quad (2.8)$$

dove:

---

<sup>3</sup>Molto spesso il termine  $l_{0,ij}$  viene confuso con la lunghezza di legame all'equilibrio a 298 K, in realtà questi due valori possono essere leggermente differenti[40].

<sup>4</sup>tecnica spettroscopica per l'analisi dei materiali che sfrutta la diffusione di una radiazione monocromatica attraverso il campione.



Figura 2.1: Confronto tra diedro proprio e improprio

- $k_{\phi,ijlh}$  è la costante di forza;
- $n$  è la molteplicità dei minimi;
- $\delta_n$  è la fase.

Il diedro improprio viene introdotto nel campo di forza per correggere le deformazioni delle geometrie di legame degli atomi ibridati  $sp^2$ ,  $sp^3$  e dei gruppi planari. Il termine improprio è definito come l'angolo tra il piano che passa per gli atomi 1, 2 e 3 e quello passante per gli atomi 2, 3 e 4 della figura 2.1b, ed è modellizzato a sua volta sull'oscillatore armonico.

Per calcolare il contributo di non-legame, che comprende le interazioni di van der Waals e quelle elettrostatiche, bisogna tener conto di tutte le coppie di atomi separate da almeno quattro legami covalenti[47, 104, 76]. Il contributo dato dalle interazioni di van der Waals, in chimica computazionale, è modellizzato come potenziale di Lennard-Jones<sup>5</sup>

$$E = \frac{1}{2} \sum_{i=1}^N \sum_{j=i+k}^N v(|\vec{R}_i - \vec{R}_j|) \quad (2.9)$$

dove:

- $N$  è il numero di oggetti del sistema;
- $\vec{R}$  è il vettore posizione;
- $v$  è il potenziale d'interazione tra le particelle;
- $i$  e  $j$  sono gli oggetti che interagiscono nella coppia;

---

<sup>5</sup>che è un potenziale di coppia, ovvero descrive l'energia potenziale di due oggetti interagenti in funzione della sola distanza che intercorre tra loro[83]

- $k$  viene definito in base al tipo di interazione di coppia<sup>6</sup>.

Il potenziale di Lennard-Jones 6-12 è un modello semplificato che descrive le caratteristiche essenziali delle interazioni di van der Waals, quando tra due atomi c'è una distanza piccolissima, questi si respingono, quando invece si ritrovano ad una distanza moderata si attraggono.

$$V_{LJ}(r) = 4\varepsilon \left[ \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6 \right] \quad (2.10)$$

Dove:

- $r$  è la distanza tra due particelle;
- $\varepsilon$  è la profondità della buca di potenziale;
- $\sigma$  è la distanza alla quale l'energia di potenziale è pari a zero.

Oltre una data distanza, definita cut-off, l'energia non venga calcolata. In questo modo viene ridotto lo sforzo computazionale e aumenta l'efficienza della simulazione.

L'interazione di repulsione è rappresentata con il termine  $(\frac{1}{r})^{12}$ , descrive la repulsione le tra particelle che sovrappongono gli orbitali elettronici quando si ritrovano a brevi distanze tra loro. Quella di attrazione, invece, espressa con il termine  $(\frac{1}{r})^6$ , modula l'attrazione a lungo raggio delle forze di dispersione di London che svaniscono quando la distanza tra gli interagenti tende all'infinito.

L'efficacia di calcolo del potenziale 12-6 è dovuta alla relazione che c'è tra gli esponenti del termine repulsivo e quello attrattivo, il primo, infatti, è il quadrato del secondo. Questo permette di ottimizzare il calcolo dei numerosissimi potenziali da valutare.

Il potenziale 12-6 contiene imprecisioni, che diventano significative a brevi distanze, provocando una bassa compressibilità nelle fasi solide e liquide, per via dell'esponente del termine repulsivo[108].

---

<sup>6</sup>per il potenziale di Lennard-Jones e il potenziale di Coulomb  $k \geq 4$ , per il potenziale di Morse  $k = 1$

Il contributo elettrostatico generalmente è calcolato attraverso il potenziale di Coulomb:

$$E_{elettrostatica} = \frac{1}{4\pi\epsilon} \frac{q_i q_j}{r_{ij}} \quad (2.11)$$

dove:

- $q_i$  e  $q_j$  sono i valori di cariche parziali;
- $r_{ij}$  è la distanza tra gli atomi  $i$  e  $j$ ;
- $\epsilon$  è la costante dielettrica.

Per calcolare questo contributo non è possibile utilizzare un sistema rudimentale on/off come per il contributo di van der Waals, perché tale discretizzazione provocherebbe una discontinuità tra le coppie di atomi posti ad una distanza minore del cut-off e quelle a una distanza maggiore. La questione si risolve in maniera approssimativa, affiancando un fattore di ridimensionamento<sup>7</sup> al cut-off, per far sì che il contributo calcolato non vada direttamente a zero.

Nel caso sia necessaria una maggiore accuratezza, è possibile applicare tecniche più sofisticate ed accurate (e più impegnative in termini computazionali) come il metodo della sommatoria di Ewald e quello del Particle Mesh Ewald (PME).

La sommatoria di Ewald è un metodo computazionale sviluppato per il calcolo delle energie elettrostatiche nei cristalli ionici, adoperato principalmente per il calcolo delle interazioni elettrostatiche a lunga distanza nei sistemi periodici<sup>8</sup>.

L'energia calcolata con il metodo di Ewald può essere suddivisa in tre componenti:

$$E_{Ewald} = E_{reale} + E_{reciproco} + E_{self} \quad (2.12)$$

dove:

---

<sup>7</sup>fattore che varia da uno a zero.

<sup>8</sup>sistema che presenta delle condizioni al contorno periodiche.

- $E_{reale}$  è l'energia dovuta alla somma diretta delle interazioni a corto raggio;
- $E_{reciproco}$  è l'energia dovuta alla somma reciproca delle interazioni a lungo raggio;
- $E_{self}$  è l'energia di auto-interazione che considera l'interazione di ciascuna carica con se stessa.

Il contributo a corto raggio è calcolato nello spazio reale, mentre il contributo a lungo raggio è calcolato nello spazio reciproco, ovvero le immagini del sistema, utilizzando la trasformata di Fourier. Il calcolo dell'ultimo termine,  $E_{self}$ , richiede che il sistema abbia carica netta neutra altrimenti può generare valori infiniti con effetti non fisicamente significativi.

La PME, invece, discretizza lo spazio dividendolo con una griglia tridimensionale e utilizza metodi di interpolazione per stimare i potenziali elettrici in punti intermedi che non coincidono con i nodi della griglia. Per calcolare il contributo energetico, con questa tecnica, vengono utilizzate le trasformate di Fourier multidimensionali.

Con i metodi della sommatoria di Ewald e del PME la convergenza dell'energia avviene in maniera più rapida, il che significa che il troncamento della somma non comporta una perdita di accuratezza.

Per comprendere al meglio quanto possa essere impegnativo calcolare il contributo elettrostatico bisogna introdurre la nozione matematica di O-grande. Tale notazione è un modo per descrivere quanto tempo o spazio richiede un algoritmo in informatica in base alla dimensione del problema.

Applicando la O-grande alle simulazioni di MD che utilizzano il potenziale di Coulomb per calcolare il contributo elettrostatico, vediamo che questa scala in maniera quadratica ( $O(n^2)$ ) in funzione del numero di cariche puntiformi ( $n$ ), laddove per lo stesso numero di cariche, la sommatoria di Ewald scala come  $O(n^{3/2})$  e il PME scala come  $O(n \log(n))$ .



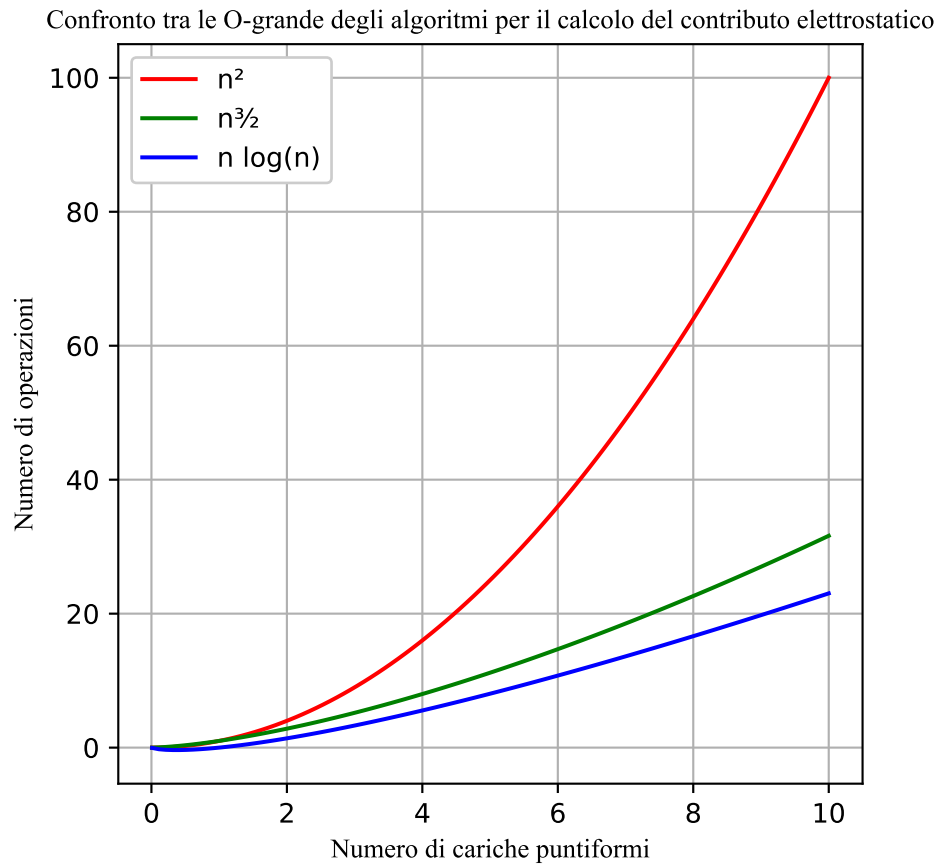


Figura 2.2: O-grande

La differenza tra il numero di operazioni, in funzione del numero di cariche puntiformi, tra i vari metodi mostra che (fig. 2.2) quello PME risulta il più performante, perché scala in maniera log-lineare[21, 10].

Altre caratteristiche da tener conto per progettare una simulazione di MD, e ottenere risultati in tempi ragionevoli, sono:

- la dimensione del sistema;
- la durata totale della simulazione;
- il timestep di integrazione.

Una decisione che ha un forte impatto sulla dimensione del sistema è la scelta del tipo di solvente (implicito ed esplicito). Il primo utilizza un modello teorico che rappresenta l'effetto medio di un solvente sulla proteina mentre il

secondo introduce un gran numero di molecole (generalmente acqua) nel sistema. Come già notato numerose volte in questa tesi, la scelta è tra accuratezza (solvente esplicito) e velocità (solvente implicito). Negli ambienti altamente eterogenei, che contengono proteine, membrane biologiche e elettroliti, è preferibile l'utilizzo di un solvente esplicito poiché quello implicito risulta approssimare grossolanamente l'effetto dell'acqua[94]. Definire un solvente esplicito permette inoltre di introdurre all'interno del sistema altri solventi oltre alle molecole di  $H_2O$ .

Per rendere valido dal punto di vista statistico il prodotto della simulazione, il tempo di simulazione deve essere paragonabile alla durata degli eventi molecolari presi in esame:

1. Stiramento di un legame covalente, circa un femtosecondo ( $10^{-15}s$ );
2. Riorientamento di una molecola d'acqua, circa quattro picosecondi ( $10^{-12}s$ );
3. Chiusura/apertura inter-dominio proteico, da cento nanosecondi ( $10^{-9}s$ ) a dieci microsecondi ( $10^{-6}s$ );
4. Cambiamenti conformazionali proteici, da un microsecondo a diversi secondi;
5. Denaturazione locale di una proteina, da un millisecondo ( $10^{-3}s$ ) a diversi secondi.

Il timestep di integrazione può essere considerato come il tempo che intercorre tra una stima dell'energia potenziale ed un'altra. Questo valore viene impostato in base al più piccolo periodo vibrazionale degli atomi del sistema — nella maggior parte dei casi l'idrogeno detiene il primato con un femtosecondo. È possibile agire sulla grandezza del timestep adoperando gli algoritmi di vincolo, come LINCS e SHAKE, che impongono una restrizione alle distanze tra gli atomi di idrogeno e gli atomi con cui intraprendono un legame covalente al fine di evitare errori di discretizzazione. Questo permette di aumentare il va-

lore del timestep fino a due femtosecondi, dimezzando così il tempo necessario per il completamento della simulazione.

L'operatore deve sempre rammentare che l'obiettivo e la scelta dei parametri sono subordinati alle capacità di calcolo delle infrastrutture che si hanno a disposizione.

Al fine di raggiungere un dato obiettivo la scelta del campo di forza può fare la differenza tra il fallimento e il successo. Pertanto, prima di decidere il campo di forza da utilizzare, è fondamentale comprendere per quali scopi è stato ideato, ricercando in letteratura articoli recenti, che hanno già simulato sistemi analoghi, oppure esaminando gli articoli che hanno caratterizzato l'architettura dei vari campi di forza.

### 2.1.1 Campo di forza: OPLS

Il campo di forza Optimized Potentials for Liquid Simulations (OPLS) è stato sviluppato dal professore William L. Jorgensen, pioniere nel campo della chimica computazionale.

Gli attributi distintivi dei parametri OPLS si distinguono per la loro ottimizzazione mirata ad adattarsi alle proprietà sperimentali dei fluidi.

OPLS presenta due differenti set di parametri:

- OPLS-aa (all-atom), dove vengono esplicitati tutti gli atomi del sistema.
- OPLS-ua (united-atom), dove gli atomi di idrogeno vengono inclusi in maniera implicita all'interno dei parametri del carbonio creando un'unica entità;

Generalmente il modello dell'acqua utilizzato per le simulazioni con il campo di forza OPLS viene scelto tra TIP3P e TIP4P.

L'equazione che descrive il campo di forza è la seguente:

$$R(r^N) = E_{stretching} + E_{angoli} + E_{diedri} + E_{non-legame} \quad (2.13)$$

dove:

$$E_{stretching} = \sum_{legami} k_r (r - r_0)^2 \quad (2.14)$$

$$E_{angoli} = \sum_{angoli} k_\theta (\theta - \theta_0)^2 \quad (2.15)$$

$$E_{diedri} = \sum_{diedri} \left( \frac{V_1}{2} [1 + \cos(\phi - \phi_1)] + \frac{V_2}{2} [1 - \cos(2\phi - \phi_2)] + \right. \\ \left. \frac{V_3}{2} [1 + \cos(3\phi - \phi_3)] + \frac{V_4}{2} [1 - \cos(4\phi - \phi_4)] \right) \quad (2.16)$$

$$E_{non-legame} = \sum_{i>j} f_{ij} \left( \frac{A_{ij}}{r_{ij}^{12}} - \frac{C_{ij}}{r_{ij}^6} + \frac{q_i q_j e^2}{4\pi\epsilon_0 r_{ij}} \right) \quad (2.17)$$

dove  $A_{ij} = \sqrt{A_{ii}A_{jj}}$  e  $C_{ij} = \sqrt{C_{ii}C_{jj}}$  sono le interazioni intramolecolari di non-legame, queste vengono contate solo tra gli atomi che distano tre o più legami di distanza, ad eccezione delle interazioni 1-4 nelle quali viene dimezzata grazie all'utilizzo del fudge factor<sup>9</sup> ( $f_{ij}$ ) [53, 52].

---

<sup>9</sup>per le interazioni 1-4 è pari a 0.5, per tutte le altre assume valore unitario.

## Parte II

### Materiali e metodi

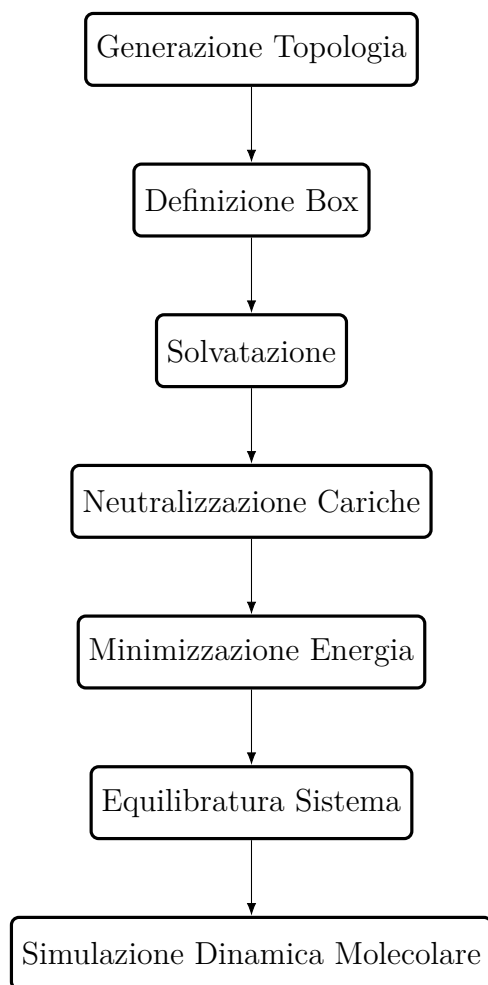
## Capitolo 3

# GROMACS

GROMACS è un software open-source progettato per la bioinformatica, specializzato nelle simulazioni di MD di proteine, lipidi e acidi nucleici ad alte prestazioni.



Il progetto nasce nel 1991 dal dipartimento di chimica fisica biologica dell'università di Groningen in Olanda da cui originariamente prese il nome di GROMACS, acronimo di GROningen MACHine for Chemical Simulations. Dal 2001 lo sviluppo di questo software è portato avanti principalmente in Svezia, dall'Istituto reale di tecnologia KTH e dall'università di Uppsala.



La mappa dei processi in figura 3 mostra le fasi che compongono una simulazione MD eseguita in GROMACS.

Al completamento dello step “Simulazione Dinamica molecolare” si ottiene un file contenente le informazioni sulla traiettoria, il quale sarà l’oggetto di successive analisi.

È fondamentale scegliere accuratamente il campo di forza da utilizzare e i parametri della dinamica: la scelta va fatta in funzione delle informazioni che si vogliono ricavare dall’analisi della traiettoria. Se la simulazione non viene subordinata all’analisi è improbabile ottenere informazioni rilevanti sulla proteina.

Questo progetto di tesi culmina con lo sviluppo di tre documenti che contengono codici informatici ed elementi di testo chiamati notebook:

1. GROMACS\_installation.ipynb
2. Molecular\_Dynamics\_Simulation\_with\_Trifluoroethanol.ipynb
3. trajectory\_analysis.ipynb

Ogni notebook ha una diversa finalità: il primo permette l’installazione di GROMACS su Google Drive in modo tale che possa essere adoperato tramite Google Colaboratory, il secondo genera un sistema molecolare e produce una simulazione MD, l’ultimo permette l’analisi della traiettoria ottenuta dal codice precedente.

## 3.1 GROMACS\_installation.ipynb

Il [codice per l'installazione](#) è stato ottenuto a partire dal codice [tutorial Lab.00](#) [31] accessibile da una repository di GitHub dell'account dell'università cilena [PB<sup>3</sup> Lab](#).

Tra il notebook uno e quello [PB<sup>3</sup> Lab](#) sono presenti delle differenze fondamentali che rendono possibile l'esecuzione di GROMACS anche da qualsiasi account google, implementare il campo di forza OPLS e aggiungere e sostituire i file<sup>1</sup> per la solvatazione.

### 3.1.1 File per la solvatazione

Per produrre delle simulazioni in presenza di TFE è necessario creare nuovi file di input da inserire all'interno di GROMACS e di integrarne altri con informazioni su questo solvente.

I file in questione possono essere distinti per estensione:

- file con estensione .atp (atom type library) sono scritte in ASCII<sup>2</sup> e contengono informazioni generali relative agli atomi, come il loro numero identificativo e le loro masse atomiche;
- file con estensione .dat sono file generici che vengono utilizzati principalmente per i formati di input e occasionalmente per i formati di output;
- file con estensione .itp (*include topology*), anche questi scritti in ASCII, contengono informazioni relative alla topologia delle molecole;
- file con estensione .pdb (*protein data bank*) contengono informazioni sulle proteine come la sequenza amminoacidica primaria e le coordinate degli atomi.

Nel resto del capitolo verranno analizzati i file inediti aggiunti e le modifiche apportate a GROMACS ottenute dai riferimenti bibliografici [102, 16].

---

<sup>1</sup>I file in questione sono pubblici ed ottenibili nella seguente [github repository](#).

<sup>2</sup>acronimo di *American Standard Code for Information Interchange*.



**ffbonded.itp**

Nel file `ffbonded.itp` sono organizzate in formato tabulare le informazioni relative alle lunghezze di legame, agli angoli di legame, ai diedri e alle energie potenziali associate a uno specifico campo di forza. È possibile ritrovare questo file nelle directory associate a OPLS:

- `../gromacs-2023/share/gromacs/top/oplsaa.ff/;`
- `../gromacs-2023/share/top/oplsaa.ff/.`

Le tabelle da me modificate all'interno del file secondo [16] sono `bondtypes`, `angletypes` e `dihedraltypes`, che contengono rispettivamente informazioni su legami, angoli e diedri.

La tabella 3.1, riporta le lunghezze di legame inserite nella tabella `bondtypes`.

Legami	$r_0$
C-F	0.136
C-C	0.151
C-O	0.141
O-H	0.098
C-H	0.109

Tabella 3.1: Lunghezza dei legami ( $r_0$ ) espressa in nm.

Le costanti di forza associate sono state ottenute manualmente ricercando all'interno dello stesso file le coppie di atomi con una lunghezza di legame uguale, o simile, eccezion fatta per il legame F-C. Quest'ultima è stata ottenuta dalla tabella II di [102], convertendo il valore da  $kJmol^{-1}\text{\AA}^{-2}$  a  $kJmol^{-1}nm^{-2}$ .

I valori presenti nella tabella 3.3 sono valori che fanno riferimento all'equazione 3.1, necessaria a calcolare l'energia potenziale associata agli angoli di legame.

Da qui è stato possibile ottenere le informazioni per le modifiche apportate alla tabella `angletypes`, riportate nella tabella 3.4.

i	j	func	$r_0$	$k_r$	
CF*	F*	1	0.13600	418400.0	$k_r = \text{F-C}[102]$
CF*	CH*	1	0.15100	265265.6	$k_r = \text{C-C3}$
CH*	O*	1	0.14100	323004.8	$k_r = \text{C3-OH}$
O*	HO*	1	0.09800	462750.4	$k_r = \text{HO-OH}$
CH*	H*	1	0.10900	307105.6	$k_r = \text{CS-HA}$

Tabella 3.2: Modifiche apportate nella tabella bondtypes del file ffbonded.itp, dove  $k_r$  sta per la costante di forza ed è espressa in  $\text{kJmol}^{-1}\text{nm}^{-2}$

Angoli	$k_\theta$	$\theta_0$
O-C-C	418.4	111.5
C-C-F	418.4	111.3
F-C-F	644.3	109.1
H-O-C	460.2	108.8
H-C-C	418.4	109.5
H-C-O	418.4	109.5
H-C-H	292.9	110.1

$$V(\theta_{ijk}) = \frac{1}{2}k_\theta(\theta - \theta_0)^2 \quad (3.1)$$

Tabella 3.3: Valori relativi all'equazione 3.1, dove:

$k_\theta$  è la costante di forza espressa in  $\text{kJ/mol}$ ;

$\theta_0$  è l'angolo di legame all'equilibrio espresso in gradi.

Per ottenere le energie potenziali associate ai diedri del TFE bisogna sostituire le costanti dell'equazione 3.2 con i corrispettivi valori presenti nella tabella 3.5.

I valori presentati nella tabella 3.5 non sono nel formato Ryckaert-Bellemans utilizzato da GROMACS (cfr. righe di commento nel file) e devono quindi essere convertiti affinché siano compatibili con la struttura della tabella dhedraltypes.

A tale proposito ci viene in soccorso il manuale di GROMACS, alla voce [Proper dihedrals](#) vengono riportate le equazioni necessarie alla conversione dei valori in coefficienti di Ryckaert-Bellemans.

i	j	k	func	$\theta_0$	$k_\theta$
O*	CH*	CF*	1	111.500	418.400
CH*	CF*	F*	1	111.300	418.400
F*	CF*	F*	1	109.100	644.300
HO*	O*	CH*	1	108.800	460.200
H*	CH*	CF*	1	109.500	418.400
H*	CH*	O*	1	109.500	418.400
H*	CH*	H*	1	110.100	292.900

Tabella 3.4: Modifiche apportate nella tabella angletypes del file ffbonded.itp.

Diedri	$k_n$	$\delta_n$	$n$	
H-O-C-C	9.4	0	1	
	-4.6	180	2	
O-C-C-F	1.1	0	3	$V(\phi_{ijkl}) = k_n[1 + \cos(n\phi - \delta_n)]$ (3.2)
H-O-C-H	1.0	0	3	
H-C-C-F	0.7	0	3	

Tabella 3.5: Valori relativi alle costanti presenti nell'equazione 3.2, dove:

$k_n$  è la costante di forza espressa in kJ/mol;

$\delta_n$  è la fase dell'angolo  $\phi$  espressa in gradi;

$n$  è la molteplicità.

Le conversioni sono state effettuate attraverso metodi visivi con l'ausilio dei pacchetti NumPy e Matplotlib in Python<sup>3</sup>.

---

<sup>3</sup>Il codice Python e gli output associati sono stati inseriti in Appendice B Jupyter Notebook: Coefficienti Ryckaert Bellemans

i	j	k	l	func	Coefficienti Ryckaert-Bellemans					
					$C_0$	$C_1$	$C_2$	$C_3$	$C_4$	$C_5$
HO*	O*	CH*	CF*	3	0.20	-9.40	9.20	0.00	0.00	0.00
O*	CH*	CF*	F*	3	1.10	3.30	0.00	-4.40	0.00	0.00
HO*	O*	CH*	H*	3	1.00	3.00	0.00	-4.00	0.00	0.00
H*	CH*	CF*	F*	3	0.70	2.09	0.00	-2.80	0.00	0.00

Tabella 3.6: Modifiche apportate nella tabella dihedraltypes del file ffbonded.itp.

### atomtypes.atp

In atomtypes.atp vengono associati i codici identificativi degli atomi con le masse atomiche corrispondenti.

Nella prima parte di di questo file sono presenti righe di commento che riportano informazioni sul contenuto del file:

1. ; OPLS atom types and masses.
2. ; Atom types are named opls\_X, where X is the OPLS number.

Il file è strutturato come una tabella composta da due colonne (l'identificativo degli atomi e le loro masse atomiche) ed è presente in duplice copia in due directory differenti. Entrambe le copie sono state sovrascritte come da tabella 3.7.

All'interno di GROMACS sono presenti due copie di questo file, in due directory differenti:

- ../gromacs-2023/share/gromacs/top/oplsaa.ff/ ;
- ../gromacs-2023/share/top/oplsaa.ff/ .

Entrambe sono state sovrascritte con un file omonimo contenente le informazioni relative agli atomi che compongono la molecola di TFE, osservabili nella tabella 3.7.

ID	ma
opls_966	12.01100
opls_967	18.99840
opls_968	12.01100
opls_969	15.99940
opls_970	1.00800
opls_971	1.00800

Tabella 3.7: Modifiche apportate al file `atomtypes.atp`, inserimento degli atomi relativi al trifluoroetano, dove:

ID è l'identificativo dell'atomo della molecola;

ma è la massa atomica ed è espressa in u.m.a.

### **ffnonbonded.itp**

Il file `ffnonbonded.itp` contiene informazioni fondamentali per le interazioni di non-legame. La tabella 3.8 mostra le proprietà chimico-fisiche degli atomi che compongono la molecola di TFE[16].

Atomo	q	$\sigma$	$\varepsilon$
C(F)	0.5323	0.325	0.26
F	-0.2057	0.294	0.26
C(H)	0.1263	0.350	0.28
O	-0.6351	0.307	0.71
H	0.4286	0.000	0.00
H(C)	0.0825	0.250	0.13

Tabella 3.8: Valori relativi agli atomi del TFE, dove:

q per carica in unità di elettroni;

$\sigma$  sta per il diametri di Lennard-Jones espresso in nm;

$\varepsilon$  sta per la buca di potenziale di Lennard-Jones espressa in kJ/mol.

In `ffnonbonded.itp` vengono associati gli identificativi contenuti in `atomtypes.atp` con le proprietà atomiche della tabella 3.8.

Le modifiche riportate di seguito sono state apportate alle due copie presenti nelle directory:

- `../gromacs-2023/share/gromacs/top/oplsaa.ff/` ;
- `../gromacs-2023/share/top/oplsaa.ff/` .

ID	Atomo	TL	ma	q	ptype	$\sigma$	$\varepsilon$
opls_966	CF*	6	12.01100	0.5323	A	3.25e-01	2.60e-01
opls_967	F*	9	18.99840	-0.2057	A	2.94e-01	2.60e-01
opls_968	CH*	6	12.01100	0.1263	A	3.50e-01	2.80e-01
opls_969	O*	8	15.99940	-0.6351	A	3.07e-01	7.10e-01
opls_970	HO*	1	1.00800	0.4286	A	0.00e+00	0.00e+00
opls_971	H*	1	1.00800	0.0825	A	2.50e-01	1.30e-01

Tabella 3.9: Modifiche apportate al file `ffnonbonded.itp`, dove:

TL è un parametro di GROMACS che indica il tipo di legame;

ma è la massa atomica espressa in u.m.a.;

ptype è il tipo di particella (A ovvero atomo).

Con le informazioni mostrate in questo file e in `ffbonded.itp` si ha una descrizione completa delle proprietà delle molecole adoperate in GROMACS.

### **residuetypes.dat**

`residuetypes.dat` è un file di input che presenta una lista di residui identificati con un codice di lunghezza variabile da uno a cinque caratteri. Ogni identificativo è affiancato da una dicitura che specifica la natura del composto che può essere biologica o chimica. Nel file esaminato ritroviamo: Protein, DNA, RNA, Water, Ion.

Questo file viene utilizzato per tutti i campi di forza presenti all'interno di GROMACS. Si trova nelle directory:

- `../gromacs-2023/share/gromacs/top/` ;
- `../gromacs-2023/share/top/` .

Al file è stata aggiunta una stringa di informazione relativa al TFE. La modifica in basso, permette il funzionamento della cella di codice del Colab Notebook “Solvation”<sup>4</sup>.

TFE      Trifluoroethanol

### **trifluoroethanol.itp & trifluoroethanol.pdb**

I file `trifluoroethanol.itp` e `trifluoroethanol.pdb` descrivono rispettivamente le proprietà chimico-fisiche e la disposizione tridimensionale degli atomi della molecola di TFE.

Il file con estensione `.itp` è stato generato da zero con i valori presenti nelle tabelle 3.2, 3.4, 3.6 e 3.9, contiene, quindi, le stesse informazioni aggiunte in `ffbonded.itp` e `ffnonbonded.itp` organizzate in maniera differente.

Il file `trifluoroethanol.itp` è stato aggiunto a GROMACS nelle directory associate a OPLS:

- `../gromacs-2023/share/gromacs/top/oplsaa.ff/;`
- `../gromacs-2023/share/top/oplsaa.ff/.`

Il file con estensione `.pdb`, invece, è stato ottenuto a partire da un file di formato differente.

Dal sito PubChem è possibile scaricare la struttura del TFE in formato `.sdf` cercando [2,2,2-Trifluoroethanol](#). Dopo il download, il file è stato convertito in formato `pdb` con l’ausilio del software di visualizzazione molecolare ChimeraX. Una volta generato `trifluoroethanol.pdb`, per renderlo compatibile con `trifluoroethanol.itp`, sono state riordinate le righe e sono stati modificati i codici identificativi degli atomi e del residuo, locati rispettivamente nella terza colonna e nella quarta colonna <sup>5</sup>.

---

<sup>4</sup>I dettagli relativi a questo codice si trovano nel paragrafo successivo.

<sup>5</sup>È possibile visualizzare entrambi i file nella cartella file della repository GitHub [Molecular\\_Dynamics\\_Simulation\\_with\\_Trifluoroethanol](#).

**tip3p\_added.pdb**

Il file `tip3p_added.pdb` presenta le coordinate di una singola molecola d'acqua.

I file di struttura relativi alla molecola di  $\text{H}_2\text{O}$  sono file che non fanno riferimento ad una singola molecola, bensì a interi box d'acqua che ne contiene un numero elevato — ad esempio, il file `spc216.gro` presenta le coordinate di duecentosedici molecole.

Per generare `tip3p_added.pdb` è stato adoperato il software ChimeraX. Caricato al suo interno `spc216.gro`, per ottenere le coordinate di una sola molecola di  $\text{H}_2\text{O}$  è bastato eliminare tutte le molecole meno una e successivamente salvare il file in formato `.pdb`<sup>6</sup>.

## 3.2 Molecular\_Dynamics\_Simulation\_with\_- Trifluoroethanol.ipynb

Il secondo notebook prodotto per questo lavoro di tesi permette l'avviamento di una simulazione di MD. Esso è formato da una prima parte che predispone i vari elementi del sistema molecolare all'interno del box di simulazione e una seconda che, dopo aver stabilizzato il sistema, procede con la simulazione producendo il file di traiettoria.

Durante la stesura del sorgente si è tenuto in conto, con particolare premura, che il target di utilizzo dello strumento sviluppato in questo elaborato potrebbe trovare impegnativo orientarsi nel codice. Per questo motivo è stata forte la volontà di rendere lo script il più possibile intuitivo e semplice d'adoperare<sup>7</sup>.

Questo fine è stato raggiunto rendendo, quando possibile, intuitiva la modifica dei parametri di interesse attraverso l'utilizzo dello strumento “Campi interattivi” all'interno delle celle: “Get the structure”, “Solvation” e “Exten-

---

<sup>6</sup>ibidem.

<sup>7</sup>il diagramma di flusso del codice è riportato nell'Appendice C.



ded Simulation”. Questo strumento permettono di inserire dei costrutti grafici interattivi che agiscono direttamente sui parametri del codice<sup>8</sup>.

### Parte 1: preparazione del sistema

Per produrre il sistema molecolare è necessario, come prima cosa, l’importazione dal drive dei file di input che comprendono `tip3p_added.pdb` e `trifluoroethanol.pdb`, già citati in precedenza, e dei file parametrici.

Per ottenere la struttura del peptide, invece, ci occorre in aiuto il pacchetto `Bio.PDB` di `Biopython`[19], che permette di scaricare un file con estensione `.pdb` direttamente da RCSB Protein Data Bank. Inserendo il codice alfanumerico a quattro cifre come argomento, `Bio.PDB` scaricherà il `pdb` corrispondente, ovvero la struttura cristallografica di una proteina oppure a un file contenente diversi modelli NMR.

Per quanto riguarda i file che presentano modelli NMR è necessario un rimaneggiamento al fine di prelevare da questi un solo modello.

Nel codice è preimpostata l’estrazione del primo dei modelli presenti nel file. Nel caso in cui l’utente volesse cambiare il modello preimpostato, previa indagine con un programma di visualizzazione molecolare, è possibile scegliere il modello che risulta più interessante semplicemente andando a modificare nella cella “Get the Structure” la stringa di codice<sup>9</sup>:

```
1 input_f=models[0]
```

Dopo aver rimosso tutte le molecole d’acqua e verificato la presenza o meno dei ponti disolfuro, viene generato il file `topol.top` contenente le informazioni relative al campo di forza selezionato, la topologia del peptide e quella delle molecole d’acqua che andranno aggiunte in seguito.

Prima di solvare il peptide devono essere settate le grandezze della box di simulazione cercando di evitare eccessi in ambo i sensi: un sistema troppo piccolo risulta poco verosimile mentre un sistema con dimensioni sproporzionate

---

<sup>8</sup>Per ulteriori dettagli consultare i commenti riportati all’interno del [codice](#).

<sup>9</sup>Nel linguaggio Python il numero 0 equivale al primo della lista.

— rispetto a quelle di un peptide — aumenta a dismisura il tempo necessario ad ottenere una simulazione della giusta durata.

Per questo è imperativo calcolare in maniera parsimoniosa le dimensioni del box di simulazione a partire dalle dimensioni del peptide. Questo calcolo viene effettuato nella cella “Calculation of Box Dimensions” a valle di un comando GROMACS che permette di orientare il peptide in modo tale che l’asse maggiore sia sovrapposto all’asse X.

Nella cella “Solvation” è necessario compiere due scelte, una subordinata all’altra, la prima se si vuole o meno aggiungere TFE al sistema, la seconda scelta riguarda la frazione di TFE, in termini di percentuale volume-volume (%v/v ).

Nel caso in cui si voglia effettivamente inserire TFE nel sistema andrà spuntata l’apposita casella al lato della cella; sotto la casella sarà presente uno slider per l’impostazione di %v/v .

Per calcolare il numero di molecole di TFE da inserire nel sistema ed ottenere la %v/v desiderata, è stato necessario ricavare dalla tabella I del riferimento bibliografico [17] i valori di volume molare per entrambi i solventi<sup>10</sup>.

L’ultimo passo da compiere per ultimare il sistema è neutralizzare le cariche introducendo all’interno del box i controioni. Questo risulta essenziale per produrre simulazioni realistiche utilizzando il metodo PME per il calcolo delle interazioni a lungo raggio.

Arrivati a questo punto risulta necessario controllare che il sistema sia coerente con i passaggi effettuati. A tal proposito è stata aggiunta la cella “Box System Visualization” che raffigura il complesso con il visualizzatore molecolare py3Dmol, in modo da osservare e valutare l’elaborato.

## Parte 2: simulazione del sistema

La seconda parte del notebook serve a rifinire il lavoro di costruzione fatto finora prima di procedere con la simulazione MD.

---

<sup>10</sup>I valori sono stati convertiti da cm<sup>3</sup>/mol a L/mol all’interno del codice.

Il primo passaggio, che viene effettuato dopo aver composto il box di simulazione, è la minimizzazione dell'energia del sistema, poiché durante la costruzione del sistema alcune molecole potrebbero aver dato vita a situazioni di ingombro sterico mentre altre potrebbero riportare distanze incompatibili tra elementi carichi dello stesso segno.

Gli algoritmi di minimizzazione hanno la funzione di dirigere il sistema verso un minimo locale, la maniera in cui questo avviene dipende dai parametri scelti.

Dopo la minimizzazione è necessario che il sistema venga veicolato verso una condizione quanto più simile alla realtà. Lo scopo è quello di portare il sistema all'equilibrio impedendo che ci siano fluttuazioni inverosimili dei valori termodinamici emulando le condizioni all'interno di un sistema biologico. Questa fase, definita equilibratura, procede finché tutti i parametri non raggiungono un valore costante.

Il processo appena descritto, avviene attraverso due fasi, la prima prevede l'accoppiamento del sistema con un bagno termico riproducendo un ensemble di tipo canonico<sup>11</sup>, dove la temperatura aumenta progressivamente fino a raggiungere quella di interesse. Nell'altra, invece, oltre alla temperatura viene avviene l'accoppiamento del valore desiderato di pressione.

La sezione dedicata alla simulazione MD è preceduta da diverse celle che hanno il compito di salvaguardare i progressi della dinamica. Questo obiettivo viene raggiunto attraverso lo spostamento della directory di lavoro all'interno del drive, in questo modo, i file generati nelle fasi successive, saranno prodotti in un ambiente protetto di proprietà dell'utente. I progressi, in questo modo, non verranno mai persi completamente, anche se la connessione con i server Google dovesse essere interrotta.

Sia a valle che a monte del comando che avvia la simulazione, sono presenti due stringhe di codice che trascrivono l'orario su un file di testo chiamato timer, in modo da poter ricavare in seguito il tempo effettivo di questo passaggio.

La dinamica viene spezzata in più parti, ognuna delle quali produce una

---

<sup>11</sup>approfondimento in Appendice A

simulazione di pari durata. L'ultima cella, "Extended Simulation", permette all'utente di acconsentire o meno alla elongazione della traiettoria tramite il click sull'apposita casella<sup>12</sup>.

Con GROMACS è quindi possibile estendere la durata della simulazione producendo traiettorie su due file differenti, anche in tempi diversi, e in seguito concatenarle per ottenere un unico file di traiettoria.

### 3.2.1 Parametri di Dinamica Molecolare

Nella scelta dai parametri per le varie fasi della dinamica bisogna ragionare sul parallelismo tra accuratezza dei risultati ed efficienza computazionale.

I file che hanno un'estensione .mdp (molecular dynamics parameters) sono file fondamentali per l'esecuzione di tutti gli step della dinamica, dalla preparazione del sistema alla simulazione finale. Per l'utilizzo del notebook sono stati generati cinque differenti file parametrici che vengono riportati nello stesso ordine in cui appaiono all'interno del codice:

1. [ions.mdp](#)
2. [minim.mdp](#)
3. [nvt.mdp](#)
4. [npt.mdp](#)
5. [md.mdp](#)

I processi che dipendono fortemente da questi file sono la neutralizzazione delle cariche del sistema, la minimizzazione dell'energia, le fasi di equilibratura e la fase di simulazione MD.

I valori associati ai parametri sono stati scelti in seguito allo studio approfondito della pagina [Molecular dynamics parameters \(.mdp options\)](#) della documentazione GROMACS, tenendo in considerazione che il codice è stato

---

<sup>12</sup>Questa opzione viene fortemente sconsigliata agli account che non possiedono un abbonamento a Colab pro o hanno consumato tutte le unità di calcolo.

ideato per essere lanciato usufruendo delle infrastrutture di calcolo messe a disposizione gratuitamente da Google tramite Colaboratory.

### 3.3 trajectory\_analysis.ipynb

Una volta terminata la simulazione si può procedere con l'analisi della traiettoria attraverso il [codice d'analisi](#).

Per questa analisi vengono considerate diverse caratteristiche e diverse proprietà come la compattezza, la flessibilità e la struttura secondaria del peptide e la stabilità del sistema.

Al fine di valutare la buona riuscita della simulazione si procede con l'analisi dell'energia potenziale, durante tutta la traiettoria, ottenendo in questo modo una stima della stabilità del sistema.

Per estrarre informazioni sulla compattezza del peptide vengono considerati il raggio d'inerzia e la superficie accessibile al solvente (SASA), per la flessibilità, invece, si analizza la root-mean-square fluctuation (RMSF), la root-mean-square deviation (RMSD) e la pairwise RMSD osservando rispettivamente la deviazione media della posizione dei residui rispetto alla posizione di riferimento, la misura della distanza tra le coordinate del peptide e quella tra i carboni alfa dei suoi residui amminoacidici.

La struttura secondaria del peptide viene descritta dagli angoli di Ramachandran (RAMA) per residuo amminacidico e rappresentando graficamente la struttura secondaria della proteina (via DSSP) durante la traiettoria.

Le prime celle del notebook sono dedicate all'importazione dei pacchetti e all'ottenimento dei file di input, prodotti in precedenza, da una simulazione e salvati all'interno del Google drive dell'utente.

Per rendere più semplice l'utilizzo di questo codice è stato aggiunta una cella che ha il compito di rinominare i file spostati dal drive alla directory di lavoro.

Il codice sfrutta uno schema ricorrente per l'analisi delle diverse caratteristiche sopra elencate:

1. La sovrascrittura del file di testo “options” con gli elementi del sistema da analizzare;
2. La produzione di una tabella di dati, un file con estensione .xvg (xm-grace<sup>13</sup>), generata da un comando GROMACS che prende in input “options”;
3. La rappresentazione grafica del file.xvg, attraverso l’impiego di apposite library.

Le ultime celle del notebook contengono, invece, le stringhe di codice che permettono l’esportazione dei file e dei grafici<sup>14</sup> dalla directory di lavoro al drive.

---

<sup>13</sup>utilizzato dal software Grace per la rappresentazione grafica 2D

<sup>14</sup>file di grafica vettoriale in formato eps.

## Parte III

### Risultati e conclusione

# Capitolo 4

## Risultati

In questo capitolo verrà validato il codice andando a eseguire delle simulazioni MD in condizioni e tempi differenti, a partire dai seguenti file pdb ottenuti con la tecnica di spettroscopia NMR:

- (1JRJ) L'exadrina-4 in una soluzione 30%TFE[79]. Il peptide[30] è composto da trentanove residui amminoacidici e mostra un comportamento simile a *Glucagon-like peptide 1* (GLP-1), un ormone prodotto dalle cellule L dell'intestino tenue in risposta all'assunzione di cibo che svolge un ruolo cruciale nella regolazione della glicemia. Questi peptidi hanno un'identità di sequenza pari al 50% ed entrambi legano il recettore di membrana RINm5f[36], di conseguenza l'exadrina-4 risulta essere un agonista di GLP-1[79].
- (1ZRV) L'AMP spingerina, in una soluzione contenente una concentrazione di TFE pari al 50%[58]. Questo è un peptide antimicrobico di 25 residui privo di cisteina, che viene espresso negli emociti<sup>1</sup> di *Pseudocanthotermes spiniger*, una termite che vive in simbiosi mutualistica con funghi del genere *Termitomyces*[57].
- (6FS5) La casocidina I[75], in un solvente composto al 60% di TFE. Il peptide in questione è formato da trentanove amminoacidi ed è stato isolato nel latte bovino[110, 4] in seguito alla proteolisi della caseina. La

---

<sup>1</sup>frazione cellulare dell'emolinfa degli artropodi.



proteolisi della caseina- $\alpha$ s2 da parte della renina produce due AMP cationici, la casocidina I e la casocidina II[110], la prima presenta otto residui amminoacidici in più, uno al C-terminale e sette al N-terminale[75].

Ciascuno dei peptidi descritti è stato sottoposto ad una prima simulazione di dieci nanosecondi in una soluzione contenente TFE, adoperando la stessa concentrazione utilizzata per la produzione dei modelli NMR, per verificare che la struttura di partenza, sottoposta al campo di forza OPLS modificato ( $\text{OPLS}_{mod}$ ), non subisca alcun tipo di modifica durante la traiettoria ( $T_1$ ). Successivamente è stata prodotta una seconda dinamica, più lunga, composta da due traiettorie differenti, la prima ( $T_2$ ), con una durata di cento nanosecondi, in acqua, e la seconda ( $T_3$ ) nuovamente in TFE con le stesse condizioni ritrovate per i modelli NMR. Con questo lungo processo si è voluto, durante  $T_2$ , perturbare il sistema generando modifiche drastiche alla conformazione di partenza per poi, con  $T_3$ , procedere con il percorso inverso<sup>2</sup>. Nella figura sottostante viene schematizzato il processo appena descritto.

---

<sup>2</sup>per produrre  $T_1$  e  $T_3$  è stato adoperato il codice `Molecular_Dynamics_Simulation_with_Trifluoroethanol.ipynb`, mentre per generare  $T_2$  si è ricorso allo script riportato in appendice D.

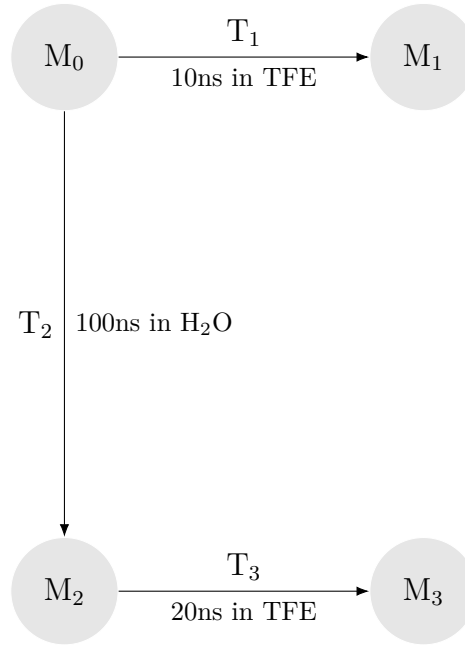


Figura 4.1: Dove:

$M_0$  sono i modelli NMR ottenuti da Protein Data Bank;

$M_1$  sono i modelli prodotti durante  $T_1$ , con  $OPLS_{mod}$ ;

$M_2$  sono i modelli prodotti durante  $T_2$ , con  $OPLS_{aa}$ ;

$M_3$  sono i modelli prodotti durante  $T_3$ , con  $OPLS_{mod}$ .

Verranno utilizzati come oggetto dell'analisi i set di modelli raffigurati nello schema. In particolare, verranno considerati esclusivamente i carboni- $\alpha$  al fine di ottenere risultati più chiari e privi di rumore.

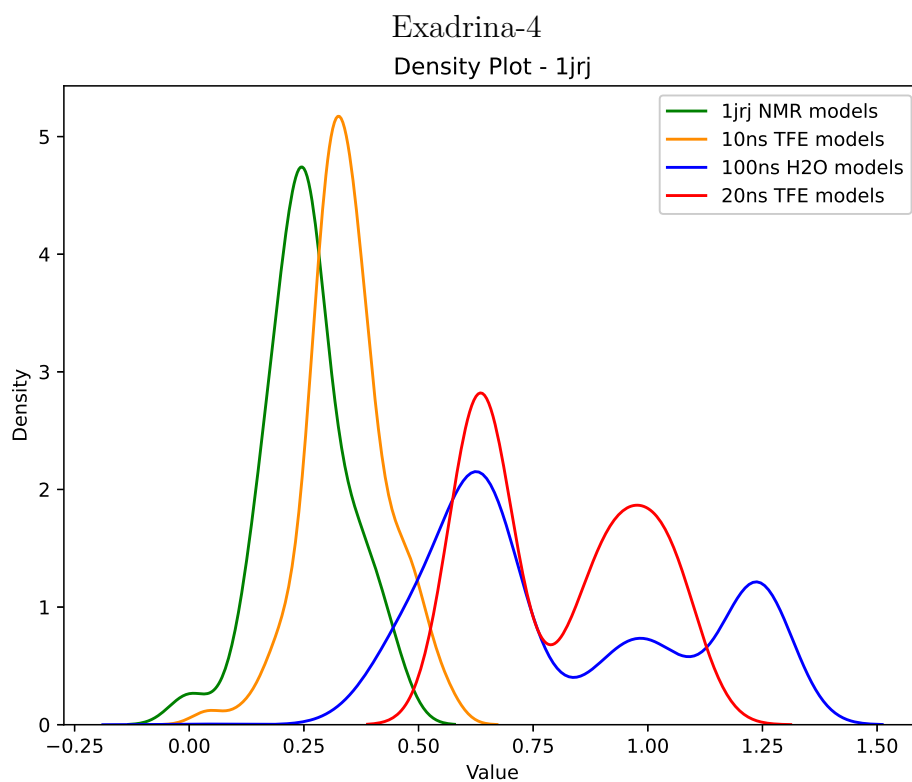
## 4.1 RMSD

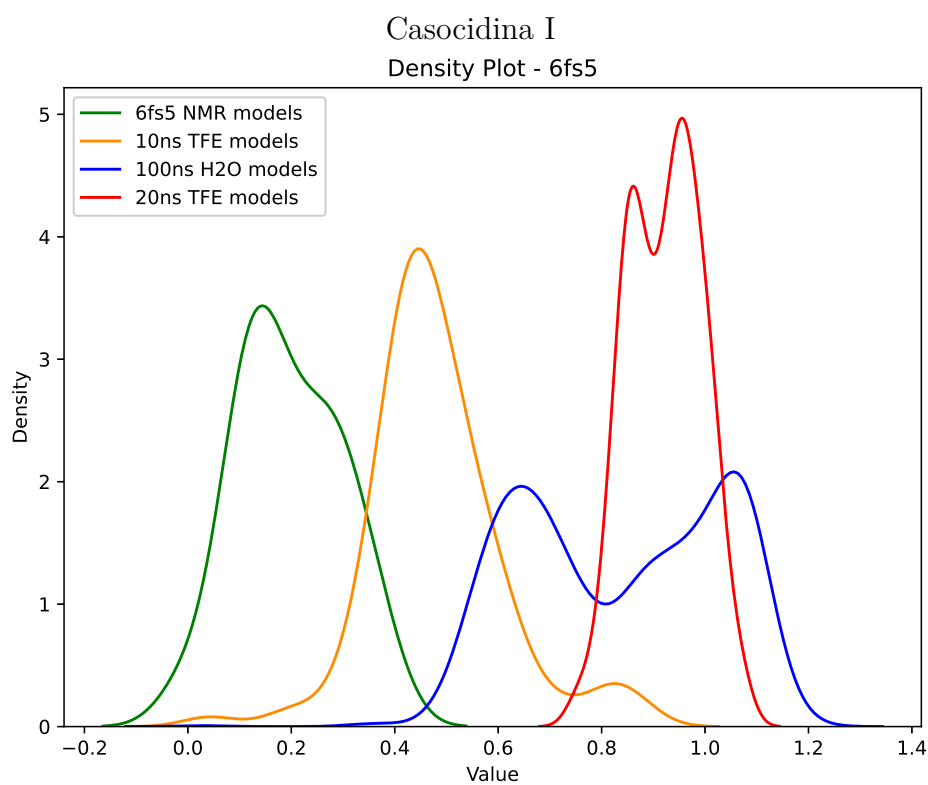
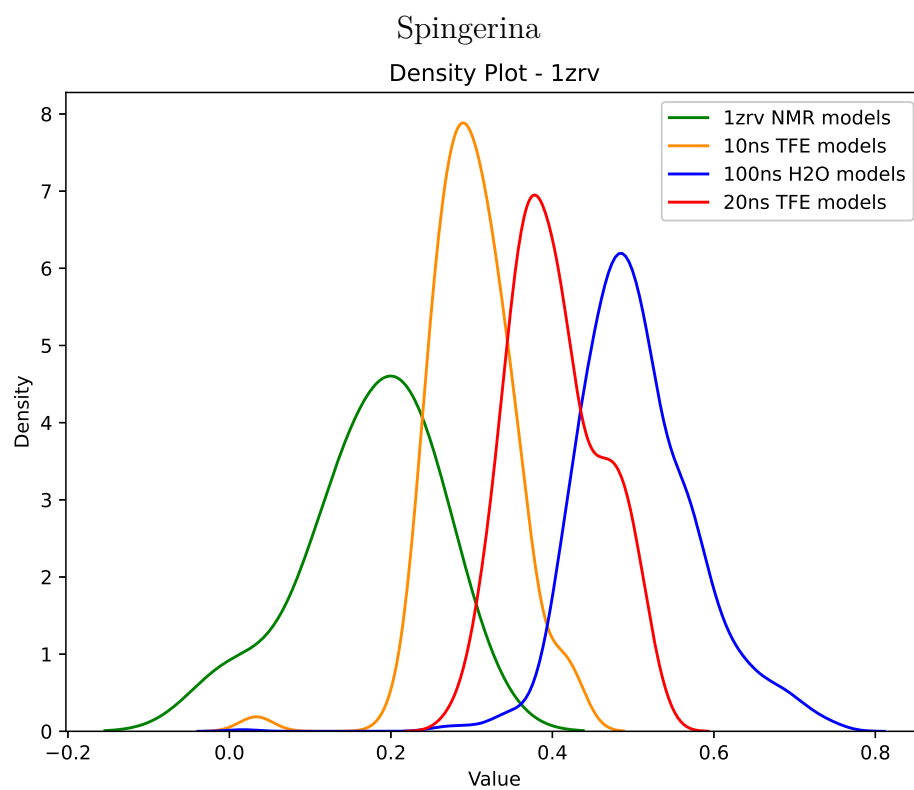
L'RMSD è una misura utilizzata per valutare la similarità tra due modelli. Tutti i modelli estratti dalle traiettorie verranno paragonati al primo modello del file pdb, da adesso definito come riferimento.

$$\text{RMSD} = \sqrt{\frac{1}{N} \sum_{i=1}^N \delta_i^2} \quad (4.1)$$

Dove  $\delta$  è la distanza tra l'atomo  $i$ -esimo del modello analizzato e l'atomo  $i$ -esimo del riferimento.

Per ognuno dei peptidi adoperati per la simulazione è stata raffigurata, mediante un density plot, la distribuzione dell'RMSD per tutti i modelli in nostro possesso ( $M_0$ ,  $M_1$ ,  $M_2$  e  $M_3$ ).





In ognuno dei grafici mostrati, la curva in verde rappresenta il set  $M_0$  (file pdb) , quella in arancione raffigura  $M_1$  (TFE), in blu  $M_2$  (TFE  $\rightarrow$   $H_2O$ ) e

in rosso  $M_3$  ( $H_2O \rightarrow TFE$ ). È possibile notare come la distribuzione degli RMSD per le diverse condizioni siano parzialmente sovrapposte e mostrino una chiara differenza in termini di valore medio. In particolare, si può notare come l'RMSD medio dei modelli tenda ad aumentare muovendosi lungo un gradiente crescente di polarità ( $TFE \rightarrow H_2O$ , ovvero da verde ad arancione a blu). L'arretramento della distribuzione rossa, che parte dal risultato della dinamica in solvente polare, ma esiste in trifluoroetanolo, indica un ritorno della struttura verso il modello NMR. Infine, la presenza di "spalle" (o addirittura di bimodalità) nelle distribuzioni suggerisce il transito della dinamica in conformazioni associate a minimi energetici locali. In sostanza, queste distribuzioni forniscono una prima indicazione della flessibilità del peptide e della sua difformità dal modello NMR, chiarendo che l'utilizzo di  $OPLS_{mod}$  impatta sulla conformazione del peptide, allontanandola come atteso nel caso della dinamica in solvente polare ( $M_2$ , blu). La vicinanza delle distribuzioni arancioni a quelle dei modelli NMR (verde) suggeriscono una modesta deviazione dalla struttura sperimentale (i.e.  $OPLS_{mod}$  è coerente con le aspettative teoriche e sperimentali). La sharpness della distribuzione di  $M_1$  per tutti i peptidi implica il rapido ritrovamento di una conformazione possibilmente stabile. I peptidi mostrano il massimo grado di flessibilità/variabilità conformazionale durante la dinamica in  $H_2O$ , e durante la transizione  $H_2O \rightarrow TFE$  il peptide non riesce a revertire completamente alla sua conformazione originale (quella in TFE, curva arancione). Questo fenomeno è dovuto alla minore durata di queste dinamiche ( $M_3$ , rosso), la quale risulta un quinto rispetto alla precedenti ( $M_2$ , blu), e per possibili barriere cinetiche che impediscono al peptide di tornare rapidamente alla sua conformazione nativa in TFE. In sintesi, le distribuzioni della RMSD sono responsive al cambio di solvente, e sembrano potenzialmente regredire tra solventi. Questo suggerisce che  $OPLS_{mod}$  funzioni.

## 4.2 Raggio d’Inerzia

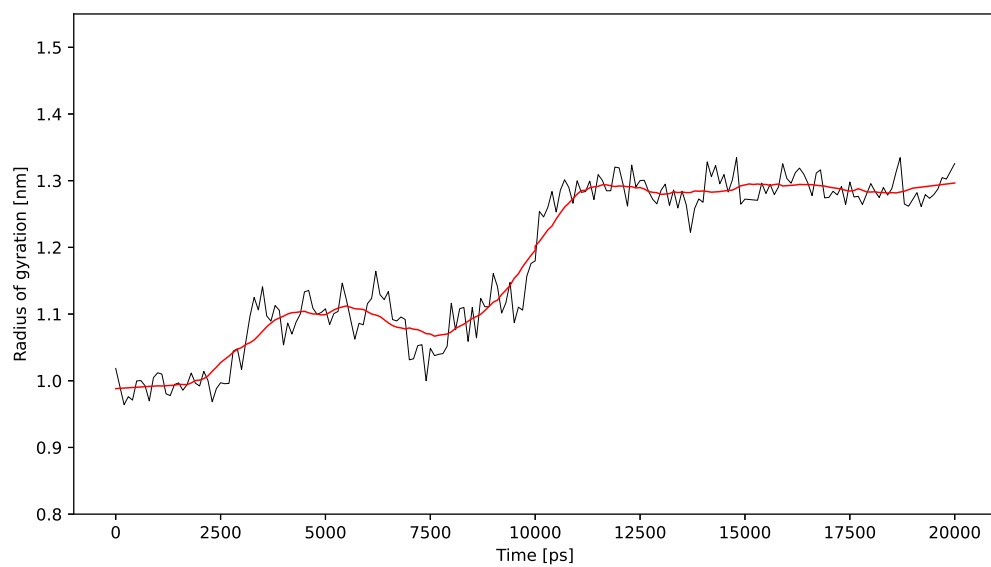
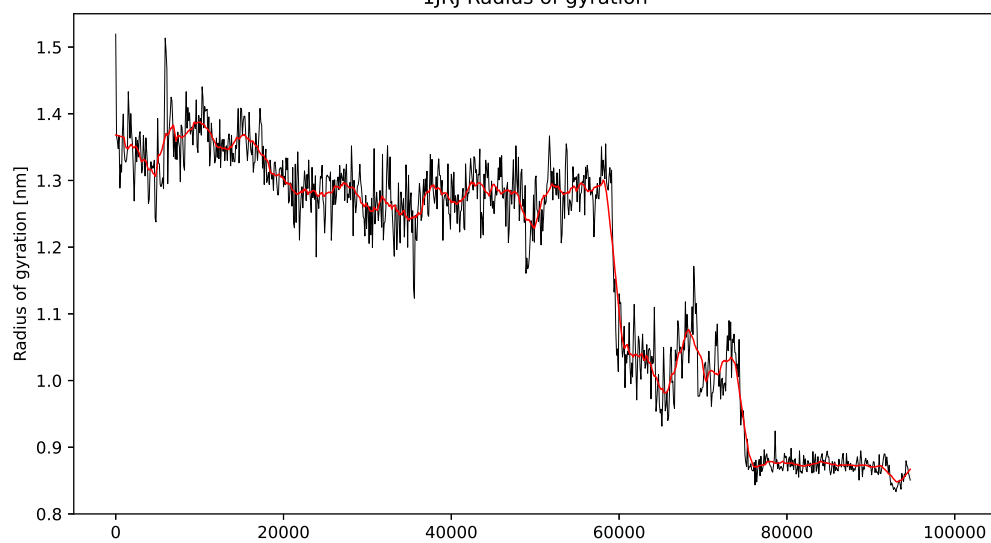
Il raggio d’inerzia ( $R_g$ ) descrive la distribuzione delle masse atomiche di una proteina intorno al proprio asse di rotazione, il quale passa per il centro di massa.

$$R_g^2 = \frac{1}{N} \sum_{i=1}^N |r_i - r_{mean}|^2 \quad (4.2)$$

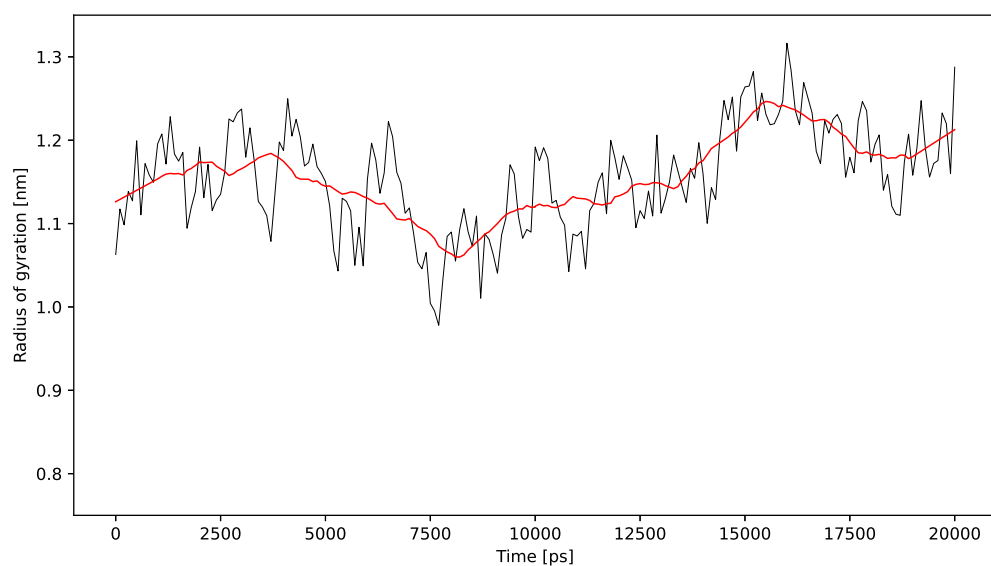
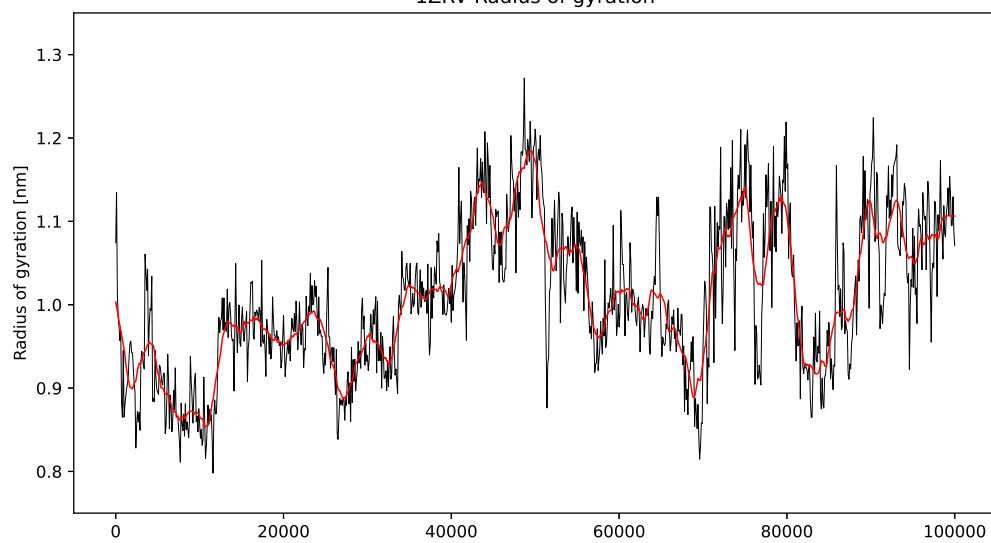
In biochimica è utilizzato per descrivere le dimensioni della catena peptidica.

Per questa analisi sono stati presi in esame soltanto i modelli estratti dalla simulazione in acqua e la sua continuazione in TFE ( $M_2$  e  $M_3$ ).

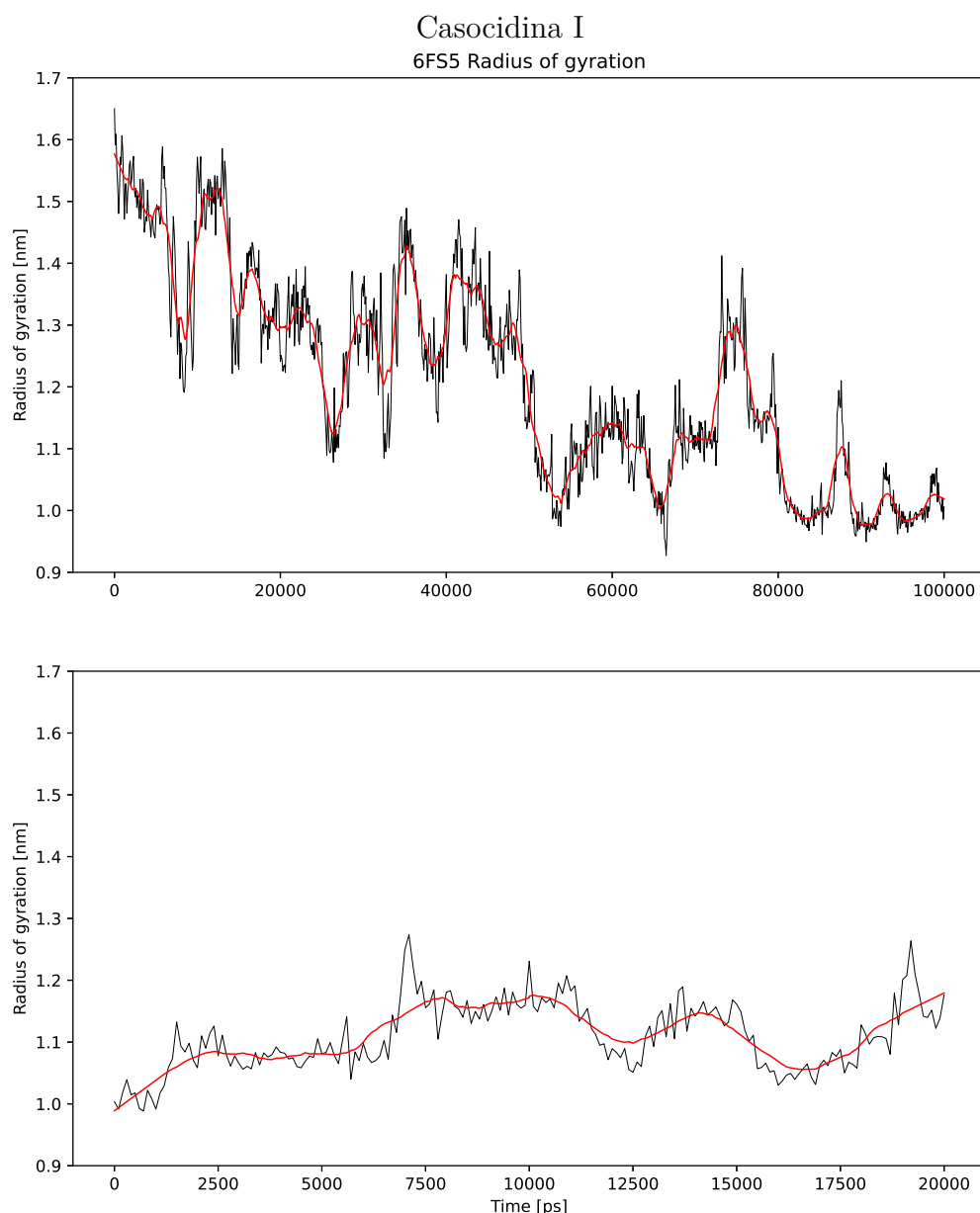
Exadrina-4  
1JRJ Radius of gyration



Spingerina  
1ZRV Radius of gyration





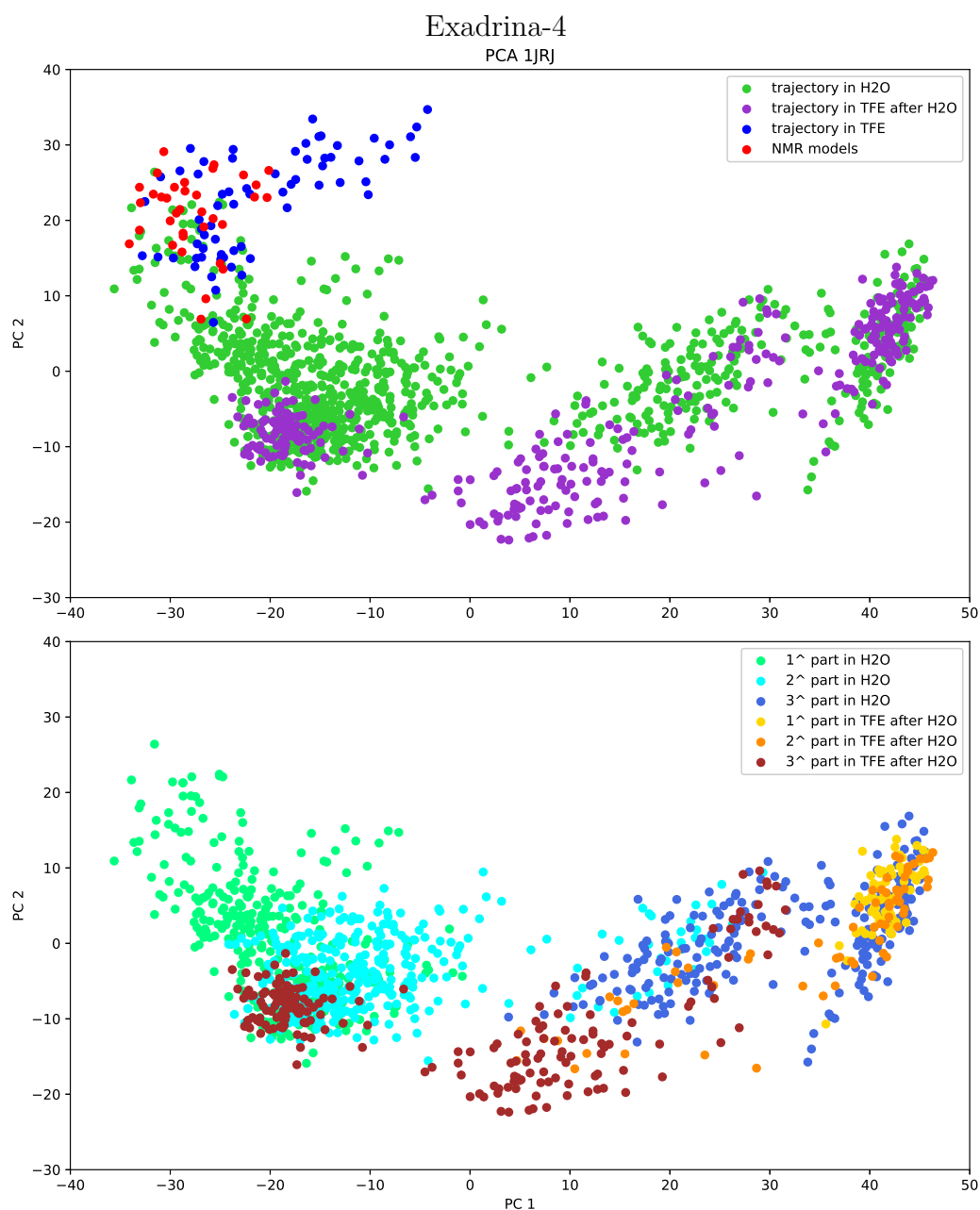


Il rate of gyration dell'Exadrina-4 in acqua varia nel corso della dinamica passando per tre fasi: la struttura raggiunge rapidamente un equilibrio dinamico (primo plateau) intorno ad una conformazione maggiormente espansa e per più della metà della dinamica la struttura oscilla intorno questo equilibrio. Successivamente, intorno al 60-esimo nanosecondo, il gyration rate crolla, suggerendo una compattazione del peptide in due tempi (60 e 80 ns). Nella fase finale della dinamica (>80ns), le fluttuazioni del gyration rate sono estremamente contenute. Nel caso della dinamica in TFE si osserva il trend opposto: la struttura parte da una conformazione compatta (1.0 nm) e tende a rilassarsi

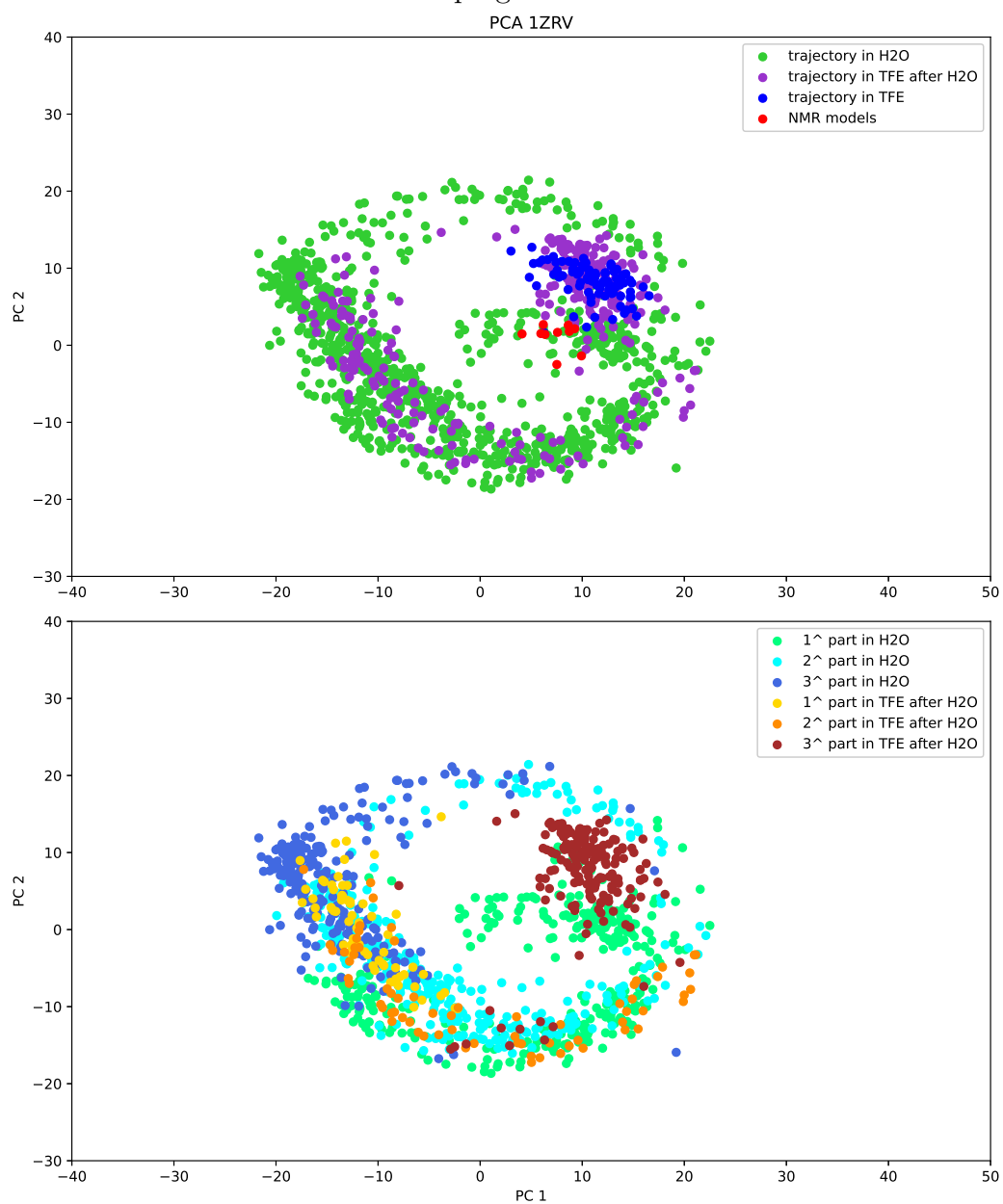
lentamente, plausibilmente a causa dell'ambiente apolare. Le fluttuazioni per il TFE sembrano essere meno violente che durante la prima fase della simulazione in acqua: questo sembra suggerire che il peptide converga gradualmente in una conformazione maggiormente stabile. In sintesi, per l'exadrina-4 il peptide mostra segnali di adattabilità strutturale all'ambiente di solvatazione, con l'acqua che contribuisce a compattare il peptide e il TFE che invece ne causa, invece, il rilassamento. La differenza tra i gyration rates ed il raggiungimento di equilibri dinamici in entrambe le condizioni suggerisce che il campo di forza sia in grado di distinguere e descrivere le interazioni solvente-specifiche. È possibile notare come la simulazione in acqua provoca una riduzione dell' $R_g$  in quanto il peptide passa da una conformazione distesa ( $\alpha$ -elica) ad una compatta. Quella seguente mostra come, riportata la struttura in una soluzione contenente TFE, aumenta l' $R_g$  e, di conseguenza, il peptide torna a distendersi. Nel caso della spingerina, la dinamica del peptide in acqua non mostra trend definiti di rilassamento o compattazione: il grafico mostra un certo grado di variabilità intorno al valore unitario ( $\mu_{R_g} = 1.00$ ). Le fluttuazioni sono mediamente contenute, e questo suggerisce che la conformazione del peptide non varia significativamente nel tempo. La dinamica della spingerina in TFE produce un radius of gyration relativamente stabile intorno all'unità ( $\mu_{R_g} = 1.15$ ). Infine, nel caso della casocidina I, l'analisi del radius of gyration nella traiettoria in acqua mostra una graduale compattazione della struttura (i.e. una progressiva diminuzione del gyration rate). L'entità delle fluttuazioni suggerisce che il peptide campiona con una certa vivacità lo spazio delle conformazioni nel tentativo di convergere verso una struttura che minimizzi le interazioni con il solvente. Il sistema sembra convergere molto lentamente verso una struttura maggiormente rilassata dopo il trasferimento in TFE. Questo trend conferma l'ipotesi anche per possibili barriere cinetiche che impediscono al peptide di tornare rapidamente alla sua conformazione nativa in questo solvente.

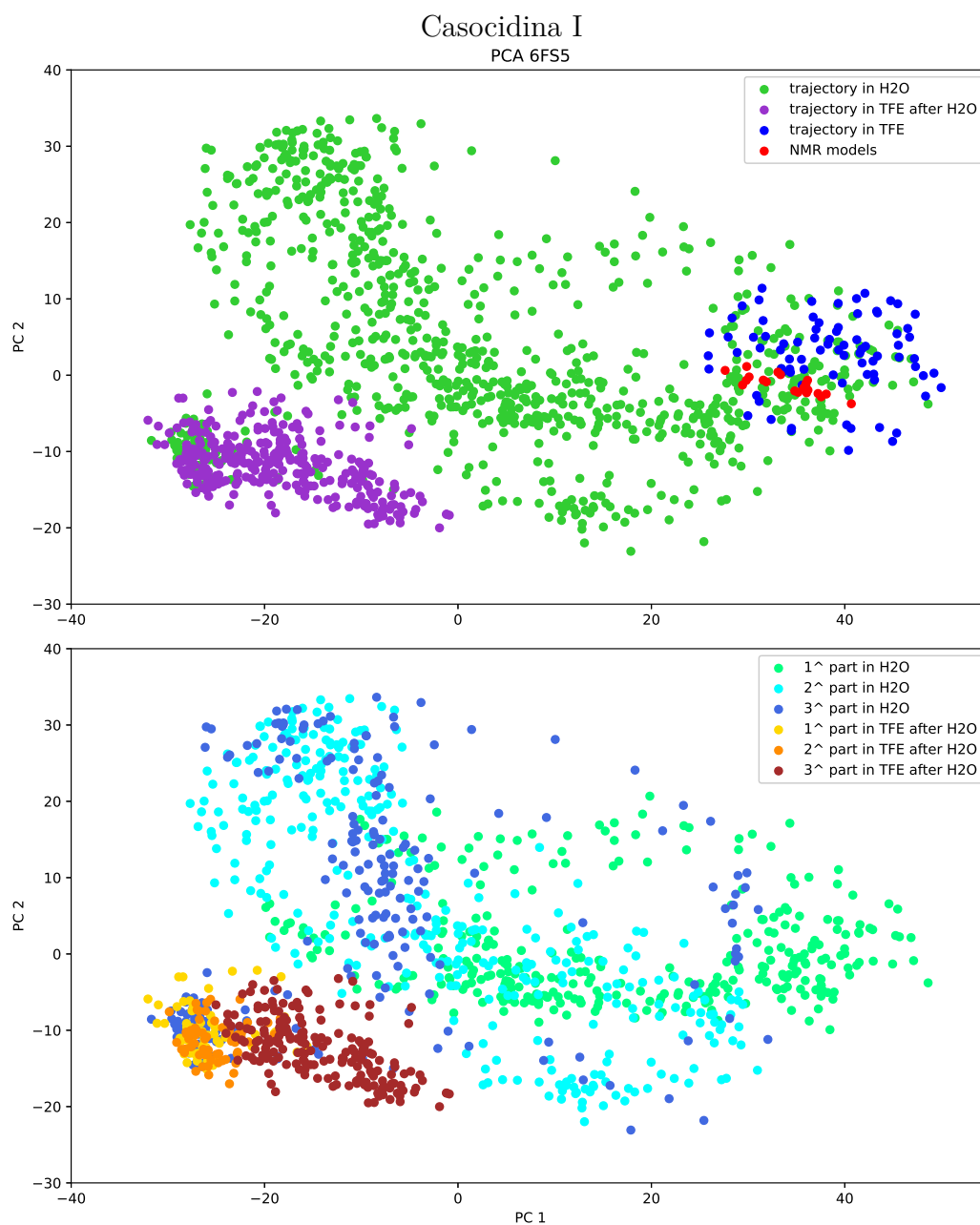
### 4.3 Prinicipal Component Analsys

L'analisi delle componenti principali (PCA) è una tecnica statistica utile per diminuisce la complessità di un set di dati riducendo il rumore e la dimensionalità.



## Spingerina





Per ogni peptide è stata effettuata la PCA, i risultati sono poi stati graficati in due differenti plot, nel primo sono stati raffigurati tutti i modelli presenti in  $M_0$ ,  $M_1$ ,  $M_2$  e  $M_3$ , com'è possibile leggere dalla legenda, mentre, per il secondo, sono stati considerati solo i gruppi  $M_2$  e  $M_3$ . Questi sono stati frazionati in tre parti per poter mostrare, approssimativamente, il decorso della dinamica.

È possibile notare che i modelli NMR e quelli estratti dalla prima dinamica in TFE risultano essere sovrapposti nel grafico, questo avviene anche per i modelli della spingerina estratti dall'ultima porzione della seconda simulazione

in TFE. Nel secondo grafico è possibile osservare il progressivo distacco, dei modelli estratti dalla simulazione in  $\text{H}_2\text{O}$ , dalla porzione occupata dal gruppo  $\text{M}_{0,1}$  e come, la susseguente dinamica in TFE, mostri un comportamento migratorio inverso.

# Capitolo 5

## Discussione

Il sistema informatico generato, descritto in questo elaborato, permette di produrre simulazioni anche senza avere delle macchine all'avanguardia, sfruttando la GPU T4 offerta dai server Google. Nonostante permetta di utilizzare il calcolo a GPU soltanto per 4 ore, la proposta gratuita è sufficiente per produrre una traiettoria di dieci nanosecondi a partire da un piccolo sistema. Nel caso di dubbi, sull'effettiva grandezza, è possibile ridurre la durata della dinamica a cinque nanosecondi per poi estenderla successivamente.

I codici per l'installazione, la simulazione in TFE e quelli per l'analisi sono stati inseriti in una repository su GitHub insieme ai file parametrici adoperati, a differenza dei sistemi individuati in letteratura, che spesso non riportano il codice informatico utilizzato, neanche tra i materiali supplementari.

Dai grafici generati e riportati nel capitolo precedente si evince che le simulazioni prodotte non distorcono la natura fisica del solvente in quanto i modelli estratti dalle simulazioni in TFE risultano simili a quelli ottenuti tramite spettroscopia NMR. È possibile riscontrare che nel caso della dinamica susseguente quella in  $H_2O$ , che ha una durata inferiore, il peptide viene riportato in una conformazione simile a quella di partenza. Nel caso della casodicina I questo non è osservabile, ma risulta evidente dal grafico della PCA che presenta la stessa tendenza delle altre simulazioni.

Nel capitolo dei materiali e metodi sono stati forniti tutti i mezzi spendibili per implementare nuovi campi di forza che presentano altri parametri per il

TFE o altri solventi, reperibili in letteratura, all'interno di GROMACS. Lo step successivo potrebbe essere quello dell'implementazione del sistema e la sua conversione in file Docker così da permetterne l'utilizzo su macchinari più performanti che sfruttano l'High-Performance Computing (HPC), adoperando piattaforme container, come Singularity, che non necessitano di una conoscenza approfondita delle specifiche di sistema al fine di installare un programma.

Questa metodologia potrebbe risultare utile sia nella previsione della struttura proteica che nella comprensione del folding dei peptidi antimicrobici e delle frazioni transmembrana.

La Critical Assessment of protein Structure Prediction (CASP) è un esperimento comunitario globale che consente di valutare le prestazioni di diversi approcci di previsione della struttura proteica. Nonostante i recenti miglioramenti nelle risorse computazionali e nella parametrizzazione dei campi di forza, i metodi di dinamica molecolare combinati con l'homology modelling non producono predizioni di struttura tanto accurate come le tecniche che sfruttano l'applicazione del machine learning (ML).

Nelle ultime edizioni di CASP, l'avvento di AlphaFold, sviluppata da DeepMind, ha attirato l'attenzione della comunità scientifica in quanto capace di predire la struttura tridimensionale di una proteina a partire dalla sequenza amminoacidica con un'accuratezza e una precisione tali da poter competere con le strutture prodotte sperimentalmente. Tuttavia, nonostante i notevoli traguardi raggiunti da AlphaFold, è importante sottolineare che attualmente le capacità dei metodi ML si limitano a una prospettiva statica anziché dinamica. Pertanto l'utilizzo combinato di approcci basati su ML con approcci di dinamica molecolare potrebbe offrire la possibilità di comprendere maggiormente i meccanismi coinvolti nel folding proteico.

Negli ultimi anni un numero crescente di gruppi di ricerca ha intuito la sinergia potenziale di questi due tecniche al fine di ottenere uno strumento preciso, accurato e con un basso costo in termini sia di tempo che computazionali.



# Appendice A

## Ensemble

Qui vengono fornite definizioni fondamentali di fisica che risultano cruciali per una comprensione approfondita della dinamica molecolare. Una delle nozioni centrali introdotte riguarda l'ensemble, noto anche come ensemble statistico. Tale concetto rappresenta un insieme di sistemi di particelle, teorici o ipotetici, che vengono impiegati nella meccanica statistica al fine di descrivere il comportamento collettivo di un singolo sistema fisico.

L'ensemble termodinamico, in particolare, è una specifica variante degli ensemble statistici. Questo, oltre a possedere le proprietà precedentemente descritte, si trova in uno stato di equilibrio statistico.

Attraverso l'uso dell'ensemble termodinamico, è possibile effettuare previsioni sulle grandezze termodinamiche e comprendere il comportamento macroscopico dei sistemi complessi a partire dalle leggi della meccanica classica o quantistica, aprendo la strada a una comprensione più approfondita della fisica molecolare e delle sue applicazioni pratiche.

I paragrafi successivi esamineranno in dettaglio tre tipi specifici di ensemble che giocano un ruolo fondamentale nella comprensione della dinamica molecolare. Questi includono: l'ensemble microcanonico, l'ensemble canonico e l'ensemble isotermico-isobarico. Ciascuno di questi approfondisce ulteriormente la comprensione dei sistemi complessi e delle loro proprietà, offrendo una prospettiva dettagliata sull'interazione tra particelle e il loro comportamento collettivo.

## A.1 Ensemble microcanonico

L'ensemble microcanonico, rappresenta uno specifico ensemble statistico che caratterizza uno stato del sistema meccanico, in cui l'energia totale è accuratamente specificata. Le condizioni che definiscono l'ensemble microcanonico implicano che il sistema è isolato, incapace di scambiare energia o materia con l'ambiente circostante, e che la sua energia rimane costante nel tempo.

L'ensemble microcanonico assegna una probabilità uniforme a ogni microstato la cui energia ricade all'interno di un determinato intervallo centrato sull'energia specificata, mentre tutti gli altri microstati hanno una probabilità assegnata pari a zero. Questo principio è espresso dalla relazione inversa tra probabilità ( $P$ ) e il numero di microstati ( $W$ ) all'interno dell'intervallo di energia, come rappresentato dall'equazione:

$$P = \frac{1}{W} \quad (\text{A.1})$$

Nel contesto dell'ensemble microcanonico, le equazioni del moto di Newton assumono un ruolo centrale, poiché l'energia del sistema rimane costante nel corso del tempo. La forza che agisce su ciascuna particella del sistema può essere calcolata come il gradiente dell'energia potenziale, con segno opposto, come rappresentato dall'equazione:

$$F(X) = -\nabla U(X) = M \frac{dV}{dt} \quad (\text{A.2})$$

$$V(t) = \frac{dX}{dt} \quad (\text{A.3})$$

dove:

- $\nabla U$  è il gradiente dell'energia potenziale;
- $X$  è l'insieme delle coordinate delle particelle del sistema;
- $V$  è la velocità associata a una particella.

Il processo di integrazione delle posizioni e delle velocità di ogni particella avviene attraverso l'uso di un metodo di integrazione simplettico<sup>1</sup> ad ogni timestep. Si definisce “traiettoria” l'evoluzione nel tempo delle coordinate e delle velocità delle particelle.

Partendo dalle posizioni iniziali, scelte dall'operatore, e dalle velocità iniziali, generate in maniera gaussiana randomizzata<sup>2</sup>, è possibile calcolare tutte le posizioni e le velocità passate e future, consentendo un'analisi dettagliata dell'evoluzione del sistema nel tempo.

## A.2 Ensemble canonico

L'ensemble canonico è un ensemble statistico che rappresenta i possibili stati microscopici di un sistema chiuso in equilibrio termico con una sorgente di calore, definita come termostato. Per calcolare la temperatura in una simulazione di MD, viene spesso impiegata la definizione dell'energia cinetica nel teorema di equipartizione dell'energia, come rappresentato dall'equazione:

$$\frac{3}{2}Nk_B T = \left\langle \sum_{i=1}^n \frac{1}{2}m_i v_i^2 \right\rangle^3 \quad (\text{A.4})$$

dove:

- $N$  è il numero dei gradi di libertà del sistema;
- $k_B$  è la costante di Boltzmann;
- $T$  è la temperatura;
- $\frac{1}{2}m_i v_i^2$  è l'energia cinetica associata alla particella  $i$ .

Nei sistemi in equilibrio termico, l'energia si distribuisce uniformemente tra le diverse forme. Gli algoritmi termostato utilizzano proprio questa definizione

---

<sup>1</sup>metodo di analisi che consente di studiare l'evoluzione nel tempo di un sistema attraverso le sue variabili dinamiche.

<sup>2</sup>valori casuali che seguono una distribuzione normale

<sup>3</sup>Le parentesi angolate  $\langle \rangle$  indicano solitamente un valore medio oppure un valore atteso

dell'energia per il calcolo della temperatura istantanea, la quale oscilla intorno alla temperatura target ( $T$ ).

Questi algoritmi alterano le equazioni di moto newtoniane aggiungendo o rimuovendo energia dalla simulazione, in modo da approssimare in maniera più o meno realistica il comportamento del termostato. Essi possono essere classificati come deterministici o stocastici, in base all'utilizzo o meno di un random seed, e come locali o globali, a seconda che operino su tutto il sistema o su un suo sottoinsieme.

Il metodo predominante utilizzato dagli algoritmi termostato per il controllo della temperatura è il ridimensionamento delle velocità. Tra gli algoritmi termostato più comunemente impiegati si annoverano: il termostato di Berendsen (deterministico e globale), il termostato di Andersen (stocastico e locale) e il termostato di Nosé-Hoover (deterministico e globale).

Il termostato di Berendsen incorpora un termine di rilassamento che consente al sistema di raggiungere la temperatura target[6], ma produce spesso artefatti, noti come "flying ice cube", che rendono le traiettorie risultanti incoerenti con l'ensemble canonico. Ciò ne sconsiglia l'uso diretto, sebbene alcune varianti siano state sviluppate per affrontare questo problema[39, 11].

L'algoritmo di Andersen, basato sulla riassegnazione delle velocità di alcune particelle scelte casualmente utilizzando la distribuzione di Maxwell-Boltzmann[2], riproduce correttamente le proprietà strutturali dell'ensemble canonico ma potrebbe non essere adatto per ottenere informazioni sulle proprietà dinamiche del sistema.

Infine, il termostato di Nosé-Hoover adotta una strategia differente rispetto agli altri termostati, poiché non agisce direttamente sulle particelle o sulle loro velocità, ma introduce un grado di libertà fittizio nel sistema, simile a un serbatoio termico, al fine di mantenere la temperatura del sistema desiderata.

### A.3 Ensemble isotermico-isobarico

In ambito dell'ensemble termodinamico, è di fondamentale importanza mantenere costanti non solo la temperatura, ma anche la pressione del sistema. Ciò si realizza mediante l'utilizzo di un algoritmo barostato, che agisce in concomitanza con un algoritmo termostato, per controllare la pressione durante la simulazione<sup>4</sup>.

Durante una simulazione, la pressione viene calcolata tramite il teorema del viriale[61], il quale stabilisce che l'energia cinetica totale di un sistema di particelle confinate in uno spazio limitato è correlata alla somma delle forze esercitate sulle singole particelle moltiplicate per le loro rispettive posizioni. Questo principio è espresso dall'equazione:

$$\langle K \rangle = -\frac{1}{2} \sum_{i=1}^n \langle F_i \cdot r_i \rangle \quad (\text{A.5})$$

dove:

- $K$  è l'energia cinetica totale del sistema composto da  $n$  particelle;
- $n$  è il numero di particelle del sistema;
- $F_i$  è la forza della particella  $i$ -esima;
- $r_i$  è la posizione della particella  $i$ -esima.

Utilizzando il teorema del viriale per calcolare la pressione istante per istante, si ottiene una pressione che oscilla intorno al valore di riferimento.

L'algoritmo barostato di Berendsen, mediante l'uso di un fattore di scala, regola periodicamente il volume del sistema, inducendo fluttuazioni realistiche della pressione man mano che questa si avvicina al valore target. La semplicità di questo algoritmo comporta tuttavia il rischio che il sistema non risulti effettivamente un ensemble isoentalpico-isobarico o isoentalpico-isobarico, pertanto

---

<sup>4</sup>L'impiego esclusivo dell'algoritmo barostato, in assenza di un termostato, conduce alla realizzazione dell'ensemble isoentalpico-isobarico, in cui il numero di particelle, la pressione e l'entalpia del sistema sono mantenuti costanti.

è consigliabile utilizzarlo esclusivamente nelle prime fasi di equilibratura del sistema.

L'algoritmo Parrinello-Rahman, introducendo un grado di libertà aggiuntivo all'equazione del moto, genera una pressione che consente un ridimensionamento anisotropo del sistema, garantendo un maggiore controllo sulla configurazione finale[82].

Infine, il barostato Martyna-Tuckerman-Tobias-Klein (MTTK), simile al precedente ma con alcune migliorie, è considerato adatto anche per sistemi di dimensioni ridotte, rendendolo una scelta più versatile rispetto alle sue controparti[72, 73].

## Appendice B

Jupyter Notebook:

Coefficienti Ryckaert-Bellemans

```
[1]: import numpy as np
import matplotlib.pyplot as plt

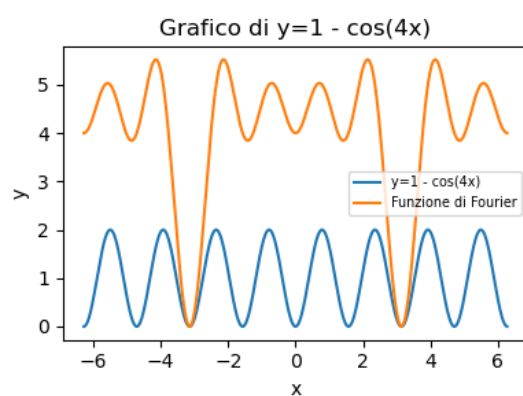
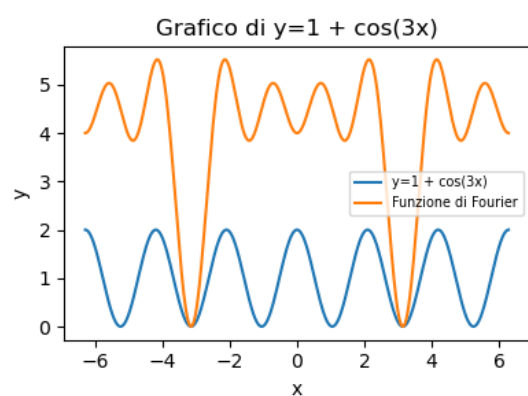
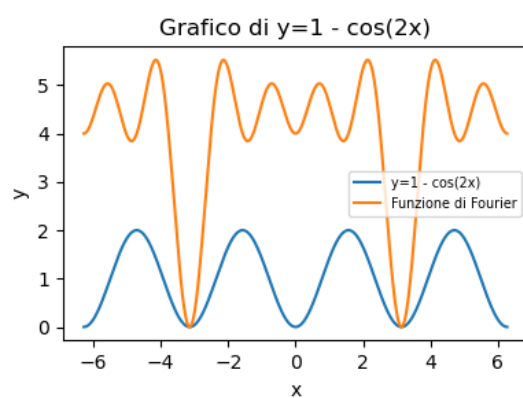
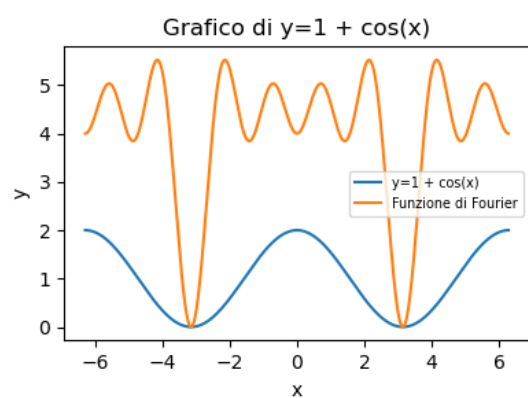
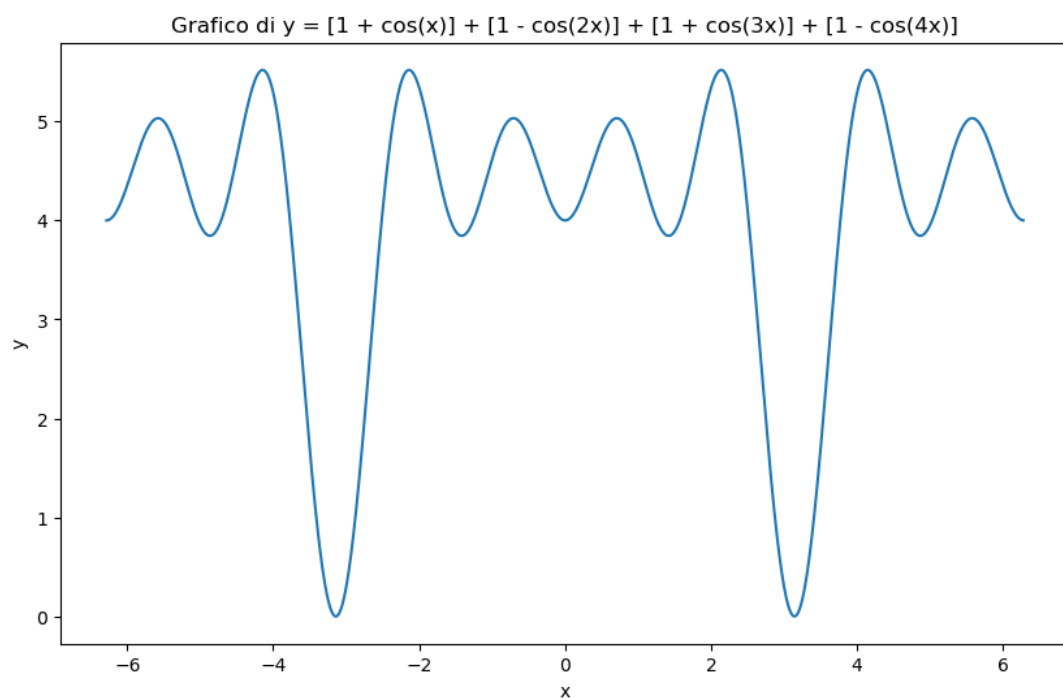
# Genera un array di valori x da -2pi a 2pi con un passo di 0.01
x = np.arange(-2*np.pi, 2*np.pi, 0.01)
# Calcola i valori corrispondenti di y
y_fourier= (1+np.cos(x))+(1-np.cos(2*x))+(1+np.cos(3*x))+(1-np.cos(4*x))

plt.figure( figsize=(10, 6)) # Dimensione dell'immagine
plt.plot(x, y_fourier) # Traccia il grafico di y rispetto a x
# Etichette degli assi
plt.xlabel('x')
plt.ylabel('y')
# Titolo del grafico
plt.title('Grafico di y = [1 + cos(x)] + [1 - cos(2x)] + [1 + cos(3x)] + [1 - cos(4x)]')
plt.show() # Visualizza il grafico

# Lista degli addendi della funzione di Fourier per il calcolo del potenziale dei
# diedri propri
y_comp= [1 + np.cos(x), 1 - np.cos(2*x), 1 + np.cos(3*x), 1 - np.cos(4*x)]
ttl=['1 + cos(x)', '1 - cos(2x)', '1 + cos(3x)', '1 - cos(4x)']

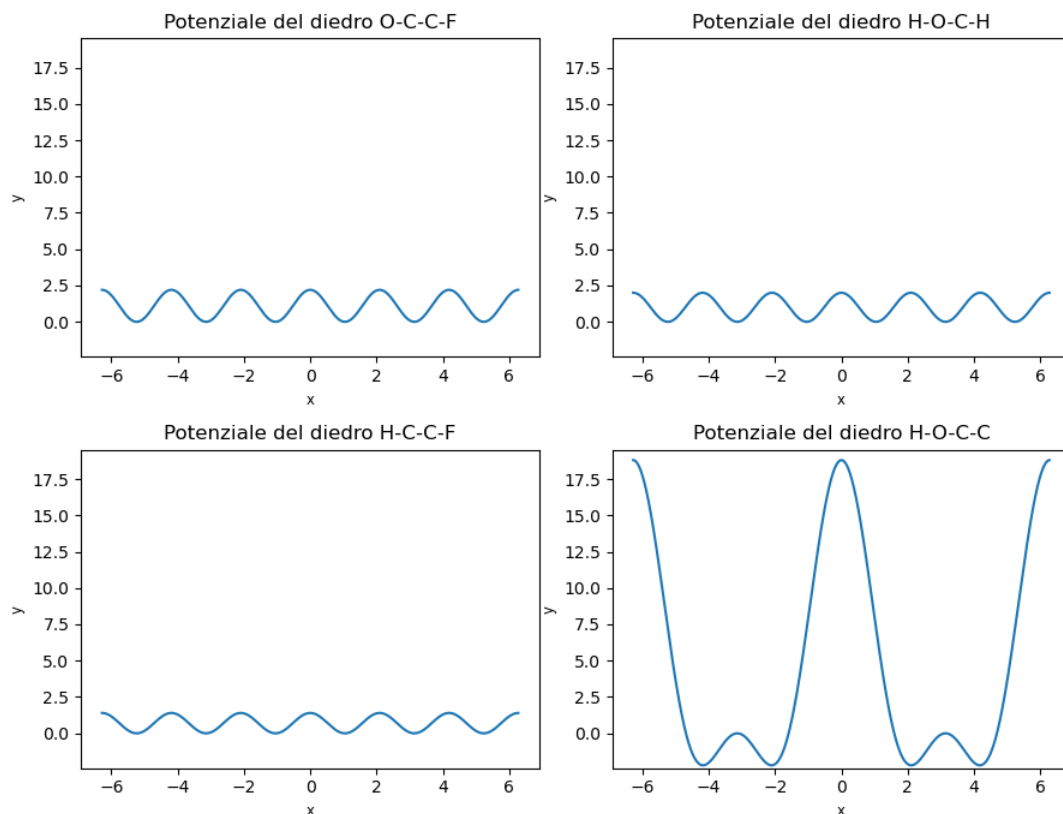
# Dimensione dell'immagine e dell'array
fig, axs = plt.subplots(int(len(y_comp)/2), int(len(y_comp)/2), figsize=(8, 6))
axs = axs.flatten() # Forzare l'array su un solo asse
for i in range(len(y_comp)):
    ax = axs[i]
    # Traccia i grafici
    ax.plot(x,y_comp[i], label=f'y={ttl[i]}')
    ax.plot(x,y_fourier, label=('Funzione di Fourier'))
    ax.legend(fontsize='x-small', loc='right') # Legenda formata dalle voci label
    # Etichette degli assi
    ax.set_xlabel('x')
    ax.set_ylabel('y')
    # Titoli dei grafici
    ax.set_title(f'Grafico di y={str(ttl[i])}')
# Visualizza il panel
plt.tight_layout()
plt.show()
```





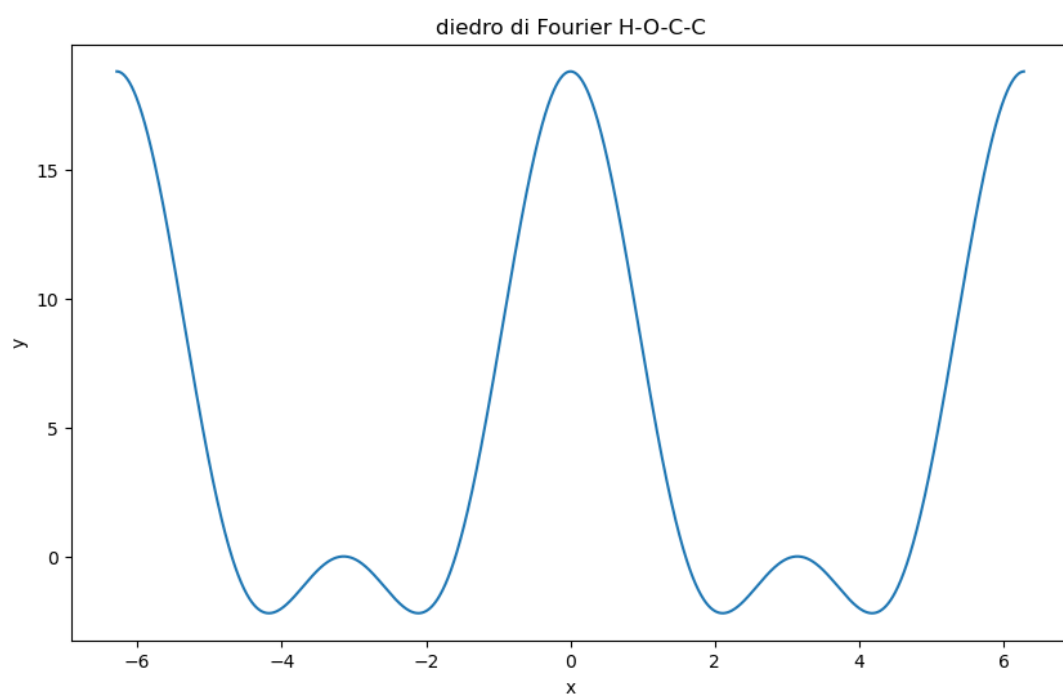
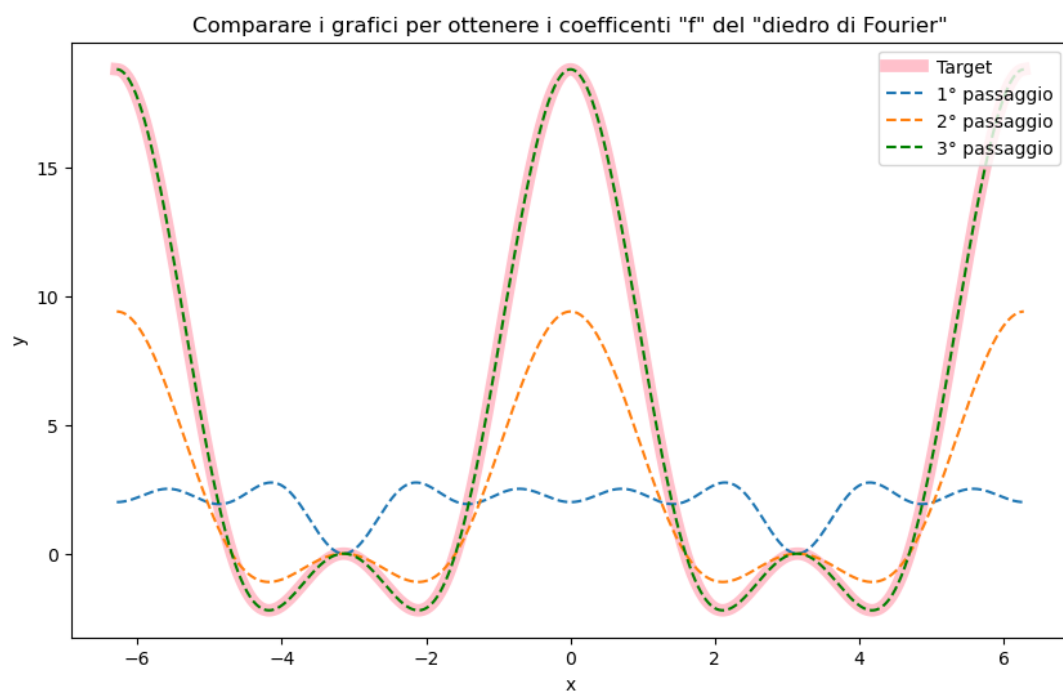
```
[2]: # Potenziale diedri
k, n, x, l= [1.1,1.0,0.7], 3, np.arange(-2*np.pi, 2*np.pi, 0.01),
↳ ['O-C-C-F', 'H-O-C-H', 'H-C-C-F']
# Potenziale diedro H-O-C-C
k1, n1, d= [9.4,-4.6], 2, np.pi
y1 = (k1[0]*(1+np.cos(x)))+(k1[1]*(1+np.cos(n1*x-d))) # Calcola i valori
↳ corrispondenti di y

# Dimensione dell'immagine e dell'array
fig, axs = plt.subplots(2, 2, figsize=(9, 7))
axs = axs.flatten() # Forzare l'array su un solo asse
for p in range(len(k)):
    y=k[p]*(1+np.cos(n*x)) # Calcola i valori corrispondenti di y
    ax = axs[p]
    ax.plot(x,y) # Traccia i grafici
    ax.set_title(f'Potenziale del diedro {l[p]}') # Titoli dei grafici
    ax.set_ylim([-2.4, 19.5]) # Impostazione dell'asse y
    # Etichette degli assi e impostazione della grandezza dei caratteri
    ax.set_xlabel('x',fontsize='small')
    ax.set_ylabel('y',fontsize='small')
axs[3].plot(x, y1) # Traccia il grafico 4
axs[3].set_title(f'Potenziale del diedro H-O-C-C') # Titolo del grafico 4
# Etichette degli assi e impostazione della grandezza dei caratteri
axs[3].set_xlabel('x',fontsize='small')
axs[3].set_ylabel('y',fontsize='small')
axs[3].set_ylim([-2.4, 19.5]) # Impostazione dell'asse y
# Visualizza il panel
plt.tight_layout()
plt.show()
```



```
[3]: plt.figure( figsize=(10, 6)) # Dimensione dell'immagine
# Titolo del grafico 5
plt.title('Comparare i grafici per ottenere i coefficienti "f" del "diedro di
↳Fourier"')
y1 = (k1[0]*(1+np.cos(x)))+(k1[1]*(1+np.cos(n1*x-d)))
plt.plot(x, y1, 'pink', label='Target', linewidth=7)
y2= 0.5*((1+np.cos(x))+(1-np.cos(2*x))+(1+np.cos(3*x))+(1-np.cos(4*x)))
plt.plot(x, y2, label='1° passaggio', linestyle='dashed')
''' confrontando i grafici ho eliminati gli elementi che contenevano cos(3*x) e
↳cos(4*x),
    e moltiplicato l'equazione per "la lista k1"'''
y3= 0.5*(k1[0]*(1+np.cos(x))+k1[1]*(1-np.cos(2*x)))
plt.plot(x,y3, label='2° passaggio',linestyle='dashed')
''' bisogna moltiplicare i coefficienti "della lista k1" per due per ottenere i
↳coefficienti f dei diedri di foudrer'''
y4= 0.5*(2*k1[0]*(1+np.cos(x))+2*k1[1]*(1-np.cos(2*x)))
plt.plot(x, y4, 'green', label='3° passaggio',linestyle='dashed')
# Etichette degli assi del grafico 5
plt.xlabel('x')
plt.ylabel('y')
plt.legend(loc='upper right') # Legenda
# Visualizza il grafico 5
plt.show()

# Fourier dihedrals:
f1, f2, f3, f4= 2*k1[0], 2*k1[1], 0, 0
yf= 0.5*(f1*(1+np.cos(x))+f2*(1-np.cos(2*x))+f3*(1+np.cos(3*x))+f4*(1-np.cos(4*x)))
plt.figure( figsize=(10, 6)) # Dimensione dell'immagine
# Titolo del grafico
plt.title('diedro di Fourier H-O-C-C')
plt.plot(x, yf)
# Etichette degli assi
plt.xlabel('x')
plt.ylabel('y')
# Visualizza il grafico
plt.show()
# Ryckaert-Bellemans coefficients:
c0, c1, c2, c3, c4, c5= f2+0.5*(f1+f3), 0.5*(-f1+3*f3), -f2+4*f4, -2*f3, -4*f4, 0
print('Coefficienti di Ryckaert-Bellemans per il diedro H-O-C-C:', '\n', 'c0=', c0,
↳'c1=', c1, 'c2=', c2, 'c3=', c3, 'c4=', c4, 'c5=', c5)
```



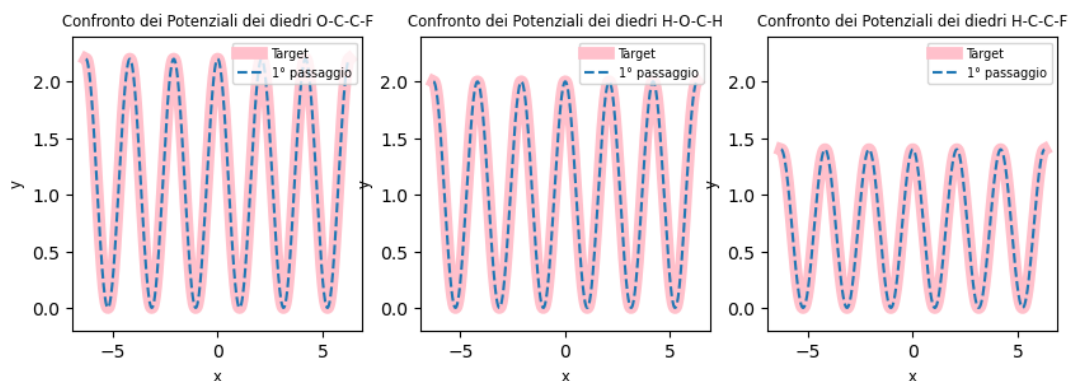
Coefficienti di Ryckaert-Bellemans per il diedro H-O-C-C:  
 $c_0 = 0.200000000000000107$   $c_1 = -9.4$   $c_2 = 9.2$   $c_3 = 0$   $c_4 = 0$   $c_5 = 0$

```
[4]: '''dalla cella precedente rivedo anche in questi uno schema,
      bisogna azzerare tutti gli elementi differenti da cos(3x)
      e raddoppiare i coefficienti della "lista k"'''

k, n, x, l= [1.1,1.0,0.7], 3, np.arange(-2*np.pi, 2*np.pi, 0.01),
↳ ['O-C-C-F', 'H-O-C-H', 'H-C-C-F']
# Fourier dihedrals:
f1, f2, f3, f4= 0, 0, [], 0
for c in k:
    f3.append(c*2)

# Dimensione dell'immagine e dell'array
fig, axs = plt.subplots(1, 3, figsize=(10, 3))
for p in range(len(k)):
    y=k[p]*(1+np.cos(n*x))
    ax=axs[p]
    ax.plot(x, y, 'pink', label='Target', linewidth=7)
    yf= 0.5*(f1*(1+np.cos(x))+f2*(1-np.cos(2*x))+f3[p]*(1+np.cos(3*x))+f4*(1-np.
↳ cos(4*x)))
    ax.plot(x, yf, label='1° passaggio', linestyle='dashed')
    # Etichette degli assi
    ax.set_xlabel('x', fontsize= 'small')
    ax.set_ylabel('y', fontsize= 'small')
    ax.set_ylim([-0.2, 2.4]) # Impostazione dell'asse y
    ax.legend(loc= 'upper right', fontsize= 'x-small')
    # Titolo del grafico
    ax.set_title(f'Confronto dei Potenziali dei diedri {l[p]}', fontsize= 'small')
    # Visualizza il grafico
plt.show()

for p in range(len(k)):
    c0, c1, c2, c3, c4, c5= f2+(0.5*(f1+f3[p])), 0.5*(-f1+(3*f3[p])), -f2+(4*f4),
↳ -2*f3[p], -4*f4, 0
    print(f'Coefficienti di Ryckaert-Bellemans per il diedro {l[p]}:', '\n', 'c0=',
↳ c0, 'c1=', c1, 'c2=', c2, 'c3=', c3, 'c4=', c4, 'c5=', c5)
```



Coefficienti di Ryckaert-Bellemans per il diedro O-C-C-F:  
c0= 1.1 c1= 3.3000000000000003 c2= 0 c3= -4.4 c4= 0 c5= 0

Coefficienti di Ryckaert-Bellemans per il diedro H-O-C-H:  
c0= 1.0 c1= 3.0 c2= 0 c3= -4.0 c4= 0 c5= 0

Coefficienti di Ryckaert-Bellemans per il diedro H-C-C-F:  
c0= 0.7 c1= 2.0999999999999996 c2= 0 c3= -2.8 c4= 0 c5= 0

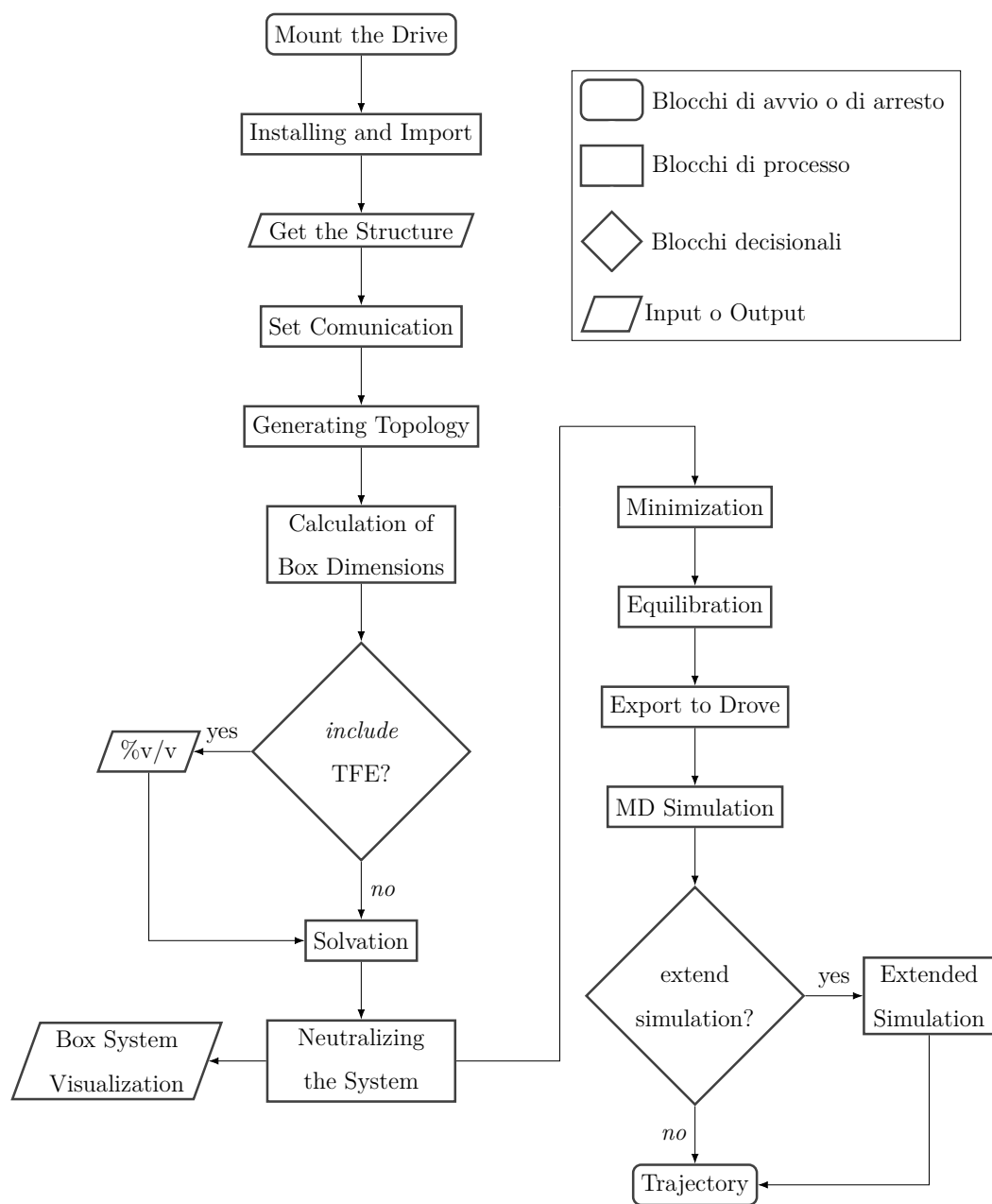
# Appendice C

## Dinamica:

## Diagramma di Flusso

Il diagramma di flusso è una rappresentazione grafica che schematizza le fasi di un processo.

Per chiarire il codice presente nel paragrafo 3.2, in questa appendice viene mostrato il corrispondente diagramma di flusso. Ogni fase è raffigurata da un blocco differente in base alla sua funzione, come mostrato nella legenda. I blocchi sono collegati da frecce, la cui direzione mostra l'ordine di esecuzione. Le frecce che originano dai blocchi decisionali sono affiancate da i termini *yes* e *no*, la scelta di una delle due opzioni è di competenza dell'utente, in base all'obiettivo della dinamica.



# Appendice D

## Script Bash:

## Simulazione in Acqua 100ns

Il seguente script è stato eseguito sull'HPC Ibisco.

Per utilizzare il bash script riportato in questa appendice è necessario digitare nella command line “sbatch -p gpus -gpus 12” seguito dal nome del file “s\_gromacs\_n3” e dal codice identificativo della struttura pdb in maiuscolo.

Nel nome del file è presente la dicitura “n3” per rammentare che verranno richieste al sistema tre nodi per far girare in parallelo lo script, e dato che ogni nodo è composto da 4 GPU si abbiamo inserito come argomento dell'opzione “gpus” il numero 12.

```
1 #!/bin/bash
2 #SBATCH --job-name=gromacs          # create a short name for your
   job
3 #SBATCH --nodes=3                   # node count
4 #SBATCH --cpus-per-task=8           # cpu-cores per task (>1 if
   multi-threaded tasks)
5 #SBATCH --mem-per-cpu=4G            # memory per cpu-core
6
7 # Download the pdb file
8 wget https://files.rcsb.org/view/"$1".pdb
9 # Move the pdb file in the input folder
```



```

10 mv "$1".pdb ./input/"$1".pdb
11 # Make a new directory
12 mkdir -p processing_"$1"
13 # Copy the mdp file in the processing directory
14 cp -R ./input/parameters/ ./processing_"$1"
15 #Removing water
16 grep -v HOH input/"$1".pdb > processing_"$1"/file_clean.pdb
17 # Enter in the working directory
18 cd processing_"$1"
19 # Create a variable
20 SINGULARITY="singularity run --nv -B ${PWD}:/host_pwd --pwd /
    host_pwd docker://nvcr.io/hpc/gromacs:2022.3"
21 set - $SINGULARITY
22 # Prepare the topology with amber99sb ff and spce water model
23 $SINGULARITY gmx pdb2gmx -f file_clean.pdb -o file_processed.
    pdb -water tip3p -ff oplsa -ignh
24 # Define the unit cell as cubic, 1.3 nm min margin
25 $SINGULARITY gmx editconf -f file_processed.pdb -o file_newbox.
    pdb -c -d 1.3 -bt cubic
26 # Add solvent (water) and use the spc216 equilibrated 3-point
    solvent model
27 $SINGULARITY gmx solvate -cp file_newbox.pdb -cs spc216.gro -o
    file_solv.pdb -p topol.top
28 # Generate the run input for genion using ions.mdp as parameter
    file
29 $SINGULARITY gmx grompp -f parameters/ions.mdp -c file_solv.pdb
    -p topol.top -o ions.tpr -maxwarn 1
30 # Creating options file
31 echo -e "SOL \n" > options
32 # Replace some water molecules (SOL) with ions to remove net
    charge
33 $SINGULARITY gmx genion -s ions.tpr -o file_solv_ions.gro -p
    topol.top -pname NA -nname CL -conc 0.15 -neutral < options
34 # Assemble the structure for Energy Minimization (EM) using
    minim.mdp as parameter file
35 $SINGULARITY gmx grompp -f parameters/minim.mdp -c
    file_solv_ions.gro -o em.tpr -maxwarn 1

```

```

36 # Relax structure through EM
37 $SINGULARITY gmx mdrun -v -deffnm em -ntmpi 4 -nb gpu -pin on -
    noconfout -ntomp 8
38 # Assemble the structure for phase I equilibration (T) using
    nvt.mdp as a parameter file
39 $SINGULARITY gmx grompp -f parameters/nvt.mdp -c em.gro -r em.
    gro -p topol.top -o nvt.tpr -maxwarn 1
40 # Equilibration phase I (under NVT/isothermal-isochoric/
    canonical) - stabilizes T
41 $SINGULARITY gmx mdrun -v -deffnm nvt -ntmpi 4 -nb gpu -pin on
    -noconfout -ntomp 8
42 # Assemble the structure for phase II equilibration (P) using
    ../conf/npt.mdp as a parameter file
43 $SINGULARITY gmx grompp -f parameters/npt.mdp -c nvt.tpr -r nvt
    .tpr -p topol.top -o npt.tpr -maxwarn 2
44 # Equilibration phase II (under NPT/isothermal-isobaric) -
    stabilizes P
45 $SINGULARITY gmx mdrun -v -deffnm npt -ntmpi 4 -nb gpu -pin on
    -noconfout -ntomp 8
46 # Assemble the structure for a 1-ns MD simulation using ../conf
    /npt.mdp as a parameter file
47 $SINGULARITY gmx grompp -f parameters/md.mdp -c npt.tpr -r npt.
    tpr -p topol.top -o md_1.tpr -maxwarn 1
48 # Run the 10 ns MD simulation
49 $SINGULARITY gmx mdrun -v -deffnm md_1 -ntmpi 4 -nb gpu -pin on
    -noconfout -ntomp 8
50 # Overwrite file input named "options"
51 echo -e "Protein \n System \n" > options
52 # Puts the center of mass of molecules in the box
53 $SINGULARITY gmx trjconv -f md_1.xtc -s md_1.tpr -o
    md_1_protPBC.xtc md_1.gro -pbc mol -center < options
54 # Overwrite again file input named "options"
55 echo -e "Protein \n" > options
56 # Extract the last frame
57 $SINGULARITY gmx trjconv -f md_1_protPBC.xtc -s md_1.tpr -o
    last_frame.pdb -dump 100000 < options

```

# Bibliografia

- [1] *Amino Acids, Peptides and Proteins: Volume 42*. The Royal Society of Chemistry, Nov. 2017.
- [2] H. C. Andersen. Molecular dynamics simulations at constant pressure and/or temperature. *The Journal of Chemical Physics*, 72(4):2384–2393, July 2008.
- [3] A. A. Bahar and D. Ren. Antimicrobial Peptides. *Pharmaceuticals*, 6(12):1543–1575, Dec. 2013. Number: 12 Publisher: Multidisciplinary Digital Publishing Institute.
- [4] F. Baum, M. Fedorova, J. Ebner, R. Hoffmann, and M. Pischetsrieder. Analysis of the Endogenous Peptide Profile of Milk: Identification of 248 Mainly Casein-Derived Peptides. *Journal of Proteome Research*, 12(12):5447–5462, Dec. 2013. Publisher: American Chemical Society.
- [5] B. Bechinger. Detergent-like properties of magainin antibiotic peptides: A  $^{31}\text{P}$  solid-state NMR spectroscopy study. *Biochimica et Biophysica Acta (BBA) - Biomembranes*, 1712(1):101–108, June 2005.
- [6] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak. Molecular dynamics with coupling to an external bath. *The Journal of Chemical Physics*, 81(8):3684–3690, Oct. 1984.
- [7] S. E. Blondelle, K. Lohner, and M.-I. Aguilar. Lipid-induced conformation and lipid-binding properties of cytolytic and antimicrobial peptides: determination and biological specificity. *Biochimica et Biophysica Acta (BBA) - Biomembranes*, 1462(1):89–108, Dec. 1999.

- [8] D. S. Bolintineanu and Y. N. Kaznessis. Computational studies of pro-tegrin antimicrobial peptides: a review. *Peptides*, 32(1):188–201, Jan. 2011.
- [9] J. K. Boparai and P. K. Sharma. Mini Review on Antimicrobial Peptides, Sources, Mechanism and Recent Applications. *Protein and Peptide Letters*, 27(1):4–16, Jan. 2020.
- [10] E. Braun, J. Gilmer, H. B. Mayes, D. L. Mobley, J. I. Monroe, S. Prasad, and D. M. Zuckerman. Best Practices for Foundations in Molecular Simulations [Article v1.0]. *Living Journal of Computational Molecular Science*, 1(1), 2019.
- [11] E. Braun, S. M. Moosavi, and B. Smit. Anomalous Effects of Velocity Rescaling Algorithms: The Flying Ice Cube Effect Revisited. *Journal of Chemical Theory and Computation*, 14(10):5262–5272, Oct. 2018. Publisher: American Chemical Society.
- [12] K. A. Brogden. Antimicrobial peptides: pore formers or metabolic inhibitors in bacteria? *Nature Reviews Microbiology*, 3(3):238–250, Mar. 2005. Number: 3 Publisher: Nature Publishing Group.
- [13] M. D. Bruch and L. M. Gierasch. Comparison of helix stability in wild-type and mutant LamB signal sequences. *Journal of Biological Chemistry*, 265(7):3851–3858, Mar. 1990.
- [14] S. N. Chitnis, K. S. Prasad, and P. M. Bhargava. Bacteriolytic activity of seminalplasmin. *Journal of General Microbiology*, 133(5):1265–1271, May 1987.
- [15] S. N. Chitnis, K. S. Prasad, and P. M. Bhargava. Isolation and characterization of autolysis-defective mutants of Escherichia coli that are resistant to the lytic activity of seminalplasmin. *Journal of General Microbiology*, 136(3):463–469, Mar. 1990.

- [16] R. Chitra and P. E. Smith. A comparison of the properties of 2,2,2-trifluoroethanol and 2,2,2-trifluoroethanol/water mixtures using different force fields. *The Journal of Chemical Physics*, 115(12):5521–5530, Sept. 2001.
- [17] R. Chitra and P. E. Smith. Properties of 2,2,2-trifluoroethanol and water mixtures. *The Journal of Chemical Physics*, 114(1):426, 2001.
- [18] C. Choudhury, U. D. Priyakumar, and G. N. Sastry. Dynamics Based Pharmacophore Models for Screening Potential Inhibitors of Mycobacterial Cyclopropane Synthase. *Journal of Chemical Information and Modeling*, 55(4):848–860, Apr. 2015. Publisher: American Chemical Society.
- [19] P. J. A. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, and M. J. L. De Hoon. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, June 2009.
- [20] M. Cudic and L. O. Jr. Intracellular Targets of Antibacterial Peptides. *Current Drug Targets*, 3(2):101–106.
- [21] T. Darden, D. York, and L. Pedersen. Particle mesh Ewald: An Nlog(N) method for Ewald sums in large systems. *The Journal of Chemical Physics*, 98(12):10089–10092, June 1993.
- [22] P. Dauber-Osguthorpe, V. A. Roberts, D. J. Osguthorpe, J. Wolff, M. Genest, and A. T. Hagler. Structure and energetics of ligand binding to proteins: Escherichia coli dihydrofolate reductase-trimethoprim, a drug-receptor system. *Proteins: Structure, Function, and Bioinformatics*, 4(1):31–47, 1988. \_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.340040106>.

- [23] R. de Almeida Vaucher, C. de Campos Velho Gewehr, A. P. Folmer Correa, V. SantAnna, J. Ferreira, and A. Brandelli. Evaluation of the immunogenicity and in vivo toxicity of the antimicrobial peptide P34. *International Journal of Pharmaceutics*, 421(1):94–98, Dec. 2011.
- [24] C. C. Dharmawardhana, K. Kanhaiya, T.-J. Lin, A. Garley, M. R. Knecht, J. Zhou, J. Miao, and H. Heinz. Reliable computational design of biological-inorganic materials to the large nanometer scale using Interface-FF. *Molecular Simulation*, 43(13-16):1394–1405, Nov. 2017. Publisher: Taylor & Francis \_\_eprint: <https://doi.org/10.1080/08927022.2017.1332414>.
- [25] H. J. Dyson, M. Rance, R. A. Houghten, P. E. Wright, and R. A. Lerner. Folding of immunogenic peptide fragments of proteins in water solution: II. The nascent helix. *Journal of Molecular Biology*, 201(1):201–217, May 1988.
- [26] M. D. Díaz, M. Fioroni, K. Burger, and S. Berger. Evidence of Complete Hydrophobic Coating of Bombesin by Trifluoroethanol in Aqueous Solution: An NMR Spectroscopic and Molecular Dynamics Study. *Chemistry - A European Journal*, 8(7):1663–1669, Apr. 2002.
- [27] G. Ehrenstein and H. Lecar. Electrically gated ionic channels in lipid bilayers. *Quarterly Reviews of Biophysics*, 10(1):1–34, Feb. 1977.
- [28] P. M. Elias and E. H. Choi. Interactions among stratum corneum defensive functions. *Experimental Dermatology*, 14(10):719–726, 2005. \_\_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1600-0625.2005.00363.x>.
- [29] F. S. Emami, V. Puddu, R. J. Berry, V. Varshney, S. V. Patwardhan, C. C. Perry, and H. Heinz. Correction to Force Field and a Surface Model Database for Silica to Simulate Interfacial Properties in Atomic Resolution. *Chemistry of Materials*, 28(1):406–407, Jan. 2016. Publisher: American Chemical Society.

- [30] J. Eng, W. A. Kleinman, L. Singh, G. Singh, and J. P. Raufman. Isolation and characterization of exendin-4, an exendin-3 analogue, from *Heloderma suspectum* venom. Further evidence for an exendin receptor on dispersed acini from guinea pig pancreas. *Journal of Biological Chemistry*, 267(11):7402–7405, Apr. 1992.
- [31] F. Engelberger, P. Galaz-Davison, G. Bravo, M. Rivera, and C. A. Ramírez-Sarmiento. Developing and Implementing Cloud-Based Tutorials That Combine Bioinformatics Software, Interactive Coding, and Visualization Exercises for Distance Learning on Structural Bioinformatics. *Journal of Chemical Education*, 98(5):1801–1807, May 2021.
- [32] M. Fernández-Vidal, S. Jayasinghe, A. S. Ladokhin, and S. H. White. Folding amphipathic helices into membranes: amphiphilicity trumps hydrophobicity. *Journal of Molecular Biology*, 370(3):459–470, July 2007.
- [33] M. Fioroni, M. D. Diaz, K. Burger, and S. Berger. Solvation Phenomena of a Tetrapeptide in Water/Trifluoroethanol and Water/Ethanol Mixtures: A Diffusion NMR, Intermolecular NOE, and Molecular Dynamics Study. *Journal of the American Chemical Society*, 124(26):7737–7744, July 2002.
- [34] B. Gomes, M. T. Augusto, M. R. Felício, A. Hollmann, O. L. Franco, S. Gonçalves, and N. C. Santos. Designing improved active peptides for therapeutic approaches against infectious diseases. *Biotechnology Advances*, 36(2):415–429, Mar. 2018.
- [35] D. Greff, F. Toma, S. Femandjian, M. Löw, and L. Kisfaludy. Conformational studies of corticotropin<sub>132</sub> and constitutive peptides by circular dichroism. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 439(1):219–231, July 1976.
- [36] R. Göke, H. C. Fehmann, T. Linn, H. Schmidt, M. Krause, J. Eng, and B. Göke. Exendin-4 is a high potency agonist and truncated exendin-(9-39)-amide an antagonist at the glucagon-like peptide 1-(7-36)-amide

- receptor of insulin-secreting beta-cells. *Journal of Biological Chemistry*, 268(26):19650–19655, Sept. 1993. Publisher: Elsevier.
- [37] R. E. W. Hancock and A. Patrzykat. Clinical Development of Cationic Antimicrobial Peptides: From Natural to Novel Antibiotics. *Current Drug Targets - Infectious Disorders*, 2(1):79–83.
- [38] R. E. W. Hancock and H.-G. Sahl. Antimicrobial and host-defense peptides as new anti-infective therapeutic strategies. *Nature Biotechnology*, 24(12):1551–1557, Dec. 2006.
- [39] S. C. Harvey, R. K.-Z. Tan, and T. E. Cheatham III. The flying ice cube: Velocity rescaling in molecular dynamics leads to violation of energy equipartition. *Journal of Computational Chemistry*, 19(7):726–740, 1998. \_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/%28SICI%291096-987X%28199805%2919%3A7%3C726%3A%3AAID-JCC4%3E3.0.CO%3B2-S>.
- [40] H. Heinz, T.-J. Lin, R. Kishore Mishra, and F. S. Emami. Thermodynamically Consistent Force Fields for the Assembly of Inorganic, Organic, and Biological Nanostructures: The INTERFACE Force Field. *Langmuir*, 29(6):1754–1765, Feb. 2013.
- [41] H. Heinz and H. Ramezani-Dakhel. Simulations of inorganic/bioorganic interfaces to discover new materials: insights, comparisons to experiment, challenges, and opportunities. *Chemical Society Reviews*, 45(2):412–448, Jan. 2016. Publisher: The Royal Society of Chemistry.
- [42] H. Heinz and U. W. Suter. Atomic Charges for Classical Simulations of Polar Systems. *The Journal of Physical Chemistry B*, 108(47):18341–18352, Nov. 2004.
- [43] H. Heinz, R. A. Vaia, B. L. Farmer, and R. R. Naik. Accurate Simulation of Surfaces and Interfaces of Face-Centered Cubic Metals Using



- 126 and 96 Lennard-Jones Potentials. *The Journal of Physical Chemistry C*, 112(44):17281–17290, Nov. 2008. Publisher: American Chemical Society.
- [44] K. Hilpert, B. McLeod, J. Yu, M. R. Elliott, M. Rautenbach, S. Ruden, J. Bürck, C. Muhle-Goll, A. S. Ulrich, S. Keller, and R. E. W. Hancock. Short Cationic Antimicrobial Peptides Interact with ATP. *Antimicrobial Agents and Chemotherapy*, 54(10):4480–4483, Oct. 2010.
- [45] L. Holfeld, N. Herth, D. Singer, R. Hoffmann, and D. Knappe. Immunogenicity and pharmacokinetics of short, proline-rich antimicrobial peptides. *Future Medicinal Chemistry*, 7(12):1581–1596, Aug. 2015. Publisher: Future Science.
- [46] A. Hollmann, M. Martinez, P. Maturana, L. C. Semorile, and P. C. Maffia. Antimicrobial Peptides: Interaction With Model and Biological Membranes and Synergism With Chemical Antibiotics. *Frontiers in Chemistry*, 6, 2018.
- [47] J. Huang and A. D. MacKerell Jr. CHARMM36 all-atom additive protein force field: Validation based on comparison to NMR data. *Journal of Computational Chemistry*, 34(25):2135–2145, 2013. \_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.23354>.
- [48] K. S. Ikuta, L. R. Swetschinski, G. Robles Aguilar, F. Sharara, T. Mestrovic, A. P. Gray, N. Davis Weaver, E. E. Wool, C. Han, A. Gershberg Hayoon, A. Aali, S. M. Abate, M. Abbasi-Kangevari, Z. Abbasi-Kangevari, S. Abd-Elsalam, G. Abebe, A. Abedi, A. P. Abhari, H. Abidi, R. G. Aboagye, A. Absalan, H. Abubaker Ali, J. M. Acuna, T. D. Adane, I. Y. Addo, O. A. Adegboye, M. Adnan, Q. E. S. Adnani, M. S. Afzal, S. Afzal, Z. B. Aghdam, B. O. Ahinkorah, A. Ahmad, A. R. Ahmad, R. Ahmad, S. Ahmad, S. Ahmad, S. Ahmadi, A. Ahmed, H. Ahmed, J. Q. Ahmed, T. Ahmed Rashid, M. Ajami, B. Aji, M. Akbarzadeh-Khiavi, C. J. Akunna, H. Al Hamad, F. Alahdab, Z. Al-Aly, M. A.

Aldeyab, A. V. Aleman, F. A. N. Alhalaiqa, R. K. Alhassan, B. A. Ali, L. Ali, S. S. Ali, Y. Alimohamadi, V. Alipour, A. Alizadeh, S. M. Aljunid, K. Allel, S. Almustanyir, E. K. Ameyaw, A. M. L. Amit, N. Anandavelane, R. Ancuceanu, C. L. Andrei, T. Andrei, D. Anggraini, A. Ansar, A. E. Anyasodor, J. Arabloo, A. Y. Aravkin, D. Areda, T. Aripov, A. A. Artamonov, J. Arulappan, R. T. Aruleba, M. Asaduzzaman, T. Ashraf, S. S. Athari, D. Atlaw, S. Attia, M. Ausloos, T. Awoke, B. P. Ayala Quintanilla, T. M. Ayana, S. Azadnajafabad, A. Azari Jafari, D. B. B, M. Badar, A. D. Badiye, N. Baghcheghi, S. Bagherieh, A. A. Baig, I. Banerjee, A. Barac, M. Bardhan, F. Barone-Adesi, H. J. Barqawi, A. Barrow, P. Baskaran, S. Basu, A.-M. M. Batiha, N. Bedi, M. A. Belete, U. I. Belgau, R. G. Bender, B. Bhandari, D. Bhandari, P. Bhardwaj, S. Bhaskar, K. Bhattacharyya, S. Bhattarai, S. Bitaraf, D. Buonsenso, Z. A. Butt, F. L. Caetano Dos Santos, J. Cai, D. Calina, P. Camargos, L. A. Cámara, R. Cárdenas, M. Cevik, J. Chadwick, J. Charan, A. Chaurasia, P. R. Ching, S. G. Choudhari, E. K. Chowdhury, F. R. Chowdhury, D.-T. Chu, I. S. Chukwu, O. Dadras, F. T. Dagnaw, X. Dai, S. Das, A. Dastiridou, S. A. Debela, F. W. Demisse, S. Demissie, D. Dereje, M. Derese, H. D. Desai, F. N. Dessalegn, S. A. A. Dessalegni, B. Desye, K. Dhaduk, M. Dhimal, S. Dhingra, N. Diao, D. Diaz, S. Djalalinia, M. Dodangeh, D. Dongarwar, B. T. Dora, F. Dorostkar, H. L. Dsouza, E. Dubljanin, S. J. Dunachie, O. C. Durojaiye, H. A. Edinur, H. B. Ejiogu, M. Ekholuenetale, T. C. Ekundayo, H. El-Abid, M. Elhadi, M. A. Elmonem, A. Emami, L. Engelbert Bain, D. B. Enyew, R. Erkhembayar, B. Eshrati, F. Etaee, A. F. Fagbamigbe, S. Falahi, A. Fallahzadeh, E. J. A. Faraon, A. Fatehizadeh, G. Fekadu, J. C. Fernandes, A. Ferrari, G. Fetensa, I. Filip, F. Fischer, M. Foroutan, P. A. Gaal, M. A. Gadanya, A. M. Gaidhane, B. Ganesan, M. Gebrehiwot, R. Ghanbari, M. Ghasemi Nour, A. Ghashghaee, A. Gholamrezanezhad, A. Gholizadeh, M. Golechha, P. Goleij, D. Golinelli, A. Goodridge, D. A. Gunawardane, Y. Guo, R. D. Gupta, S. Gupta, V. B. Gupta, V. K. Gupta,

A. Guta, P. Habibzadeh, A. Haddadi Avval, R. Halwani, A. Hanif, M. A. Hannan, H. Harapan, S. Hassan, H. Hassankhani, K. Hayat, B. Heibati, G. Heidari, M. Heidari, R. Heidari-Soureshjani, C. Herteliu, D. Z. Heyi, K. Hezam, P. Hoogar, N. Horita, M. M. Hossain, M. Hosseinzadeh, M. Hostiuc, S. Hostiuc, S. Hoveidamanesh, J. Huang, S. Hussain, N. R. Hussein, S. E. Ibitoye, O. S. Ilesanmi, I. M. Ilic, M. D. Ilic, M. T. Imam, M. Immurana, L. R. Inbaraj, A. Iradukunda, N. E. Ismail, C. C. D. Iwu, C. J. Iwu, L. M. J, M. Jakovljevic, E. Jamshidi, T. Javaheri, F. Javanmardi, J. Javidnia, S. K. Jayapal, U. Jayarajah, R. Jebai, R. P. Jha, T. Joo, N. Joseph, F. Joukar, J. J. Jozwiak, S. E. O. Kacimi, V. Kadashetti, L. R. Kalankesh, R. Kalhor, V. K. Kamal, H. Kandel, N. Kapoor, S. Karkhah, B. G. Kassa, N. J. Kassebaum, P. D. Katoto, M. Keykhaei, H. Khajuria, A. Khan, I. A. Khan, M. Khan, M. N. Khan, M. A. Khan, M. M. Khatatbeh, M. M. Khater, H. R. Khayat Kashani, J. Khubchandani, H. Kim, M. S. Kim, R. W. Kimokoti, N. Kissoon, S. Kochhar, F. Kompani, S. Kosen, P. A. Koul, S. L. Koulmane Laxminarayana, F. Krapp Lopez, K. Krishan, V. Krishnamoorthy, V. Kulkarni, N. Kumar, O. P. Kurmi, A. Kuttikkattu, H. H. Kyu, D. K. Lal, J. Lám, I. Landires, S. Lasrado, S.-w. Lee, J. Lenzi, S. Lewycka, S. Li, S. S. Lim, W. Liu, R. Lodha, M. J. Loftus, A. Lohiya, L. Lorenzovici, M. Lotfi, A. Mahmoodpoor, M. A. Mahmoud, R. Mahmoudi, A. Majeed, J. Majidpoor, A. Makki, G. A. Mamo, Y. Manla, M. Martorell, C. N. Matei, B. McManigal, E. Mehrabi Nasab, R. Mehrotra, A. Melese, O. Mendoza-Cano, R. G. Menezes, A.-F. A. Mentis, G. Micha, I. M. Michalek, A. C. Micheletti Gomide Nogueira De Sá, N. Milevska Kostova, S. A. Mir, M. Mirghafourvand, S. Mirmoeeni, E. M. Mirrakhimov, M. Mirza-Aghazadeh-Attari, A. S. Misganaw, A. Misganaw, S. Misra, E. Mohammadi, M. Mohammadi, A. Mohammadian-Hafshejani, S. Mohammed, S. Mohan, M. Mohseni, A. H. Mokdad, S. Momtazmanesh, L. Monasta, C. E. Moore, M. Moradi, M. Moradi Sarabi, S. D. Morrison, M. Motaghinejad, H. Mousavi Isfahani, A. Mousavi Khaneghah,

S. A. Mousavi-Aghdas, S. Mubarik, F. Mulita, G. B. B. Mulu, S. B. Munro, S. Muthupandian, T. S. Nair, A. A. Naqvi, H. Narang, Z. S. Natto, M. Naveed, B. P. Nayak, S. Naz, I. Negoï, S. A. Nejadghaderi, S. Neupane Kandel, C. H. Ngwa, R. K. Niazi, A. T. Nogueira De Sá, N. Noroozi, H. Nouraei, A. Nowroozi, V. Nuñez-Samudio, J. J. Nutor, C. I. Nzoputam, O. J. Nzoputam, B. Oancea, R. M. Obaidur, V. A. Ojha, A. P. Okekunle, O. C. Okonji, A. T. Olagunju, B. O. Olusanya, A. Omar Bali, E. Omer, N. Otstavnov, B. Oumer, M. P A, J. R. Padubidri, K. Pakshir, T. Palicz, A. Pana, S. Pardhan, J. L. Paredes, U. Parekh, E.-C. Park, S. Park, A. Pathak, R. Paudel, U. Paudel, S. Pawar, H. Pazoki Toroudi, M. Peng, U. Pensato, V. C. F. Pepito, M. Pereira, M. F. P. Peres, N. Perico, I.-R. Petcu, Z. Z. Piracha, I. Podder, N. Pokhrel, R. Poluru, M. J. Postma, N. Pourtaheri, A. Prashant, I. Qattee, M. Rabiee, N. Rabiee, A. Radfar, S. Raeghi, S. Rafiei, P. R. Raghav, L. Rahbarnia, V. Rahimi-Movaghar, M. Rahman, M. A. Rahman, A. M. Rahmani, V. Rahmanian, P. Ram, M. M. A. N. Ranjha, S. J. Rao, M.-M. Rashidi, A. Rasul, Z. A. Ratan, S. Rawaf, R. Rawassizadeh, M. S. Razeghinia, E. M. M. Redwan, M. T. Regasa, G. Remuzzi, M. A. Reta, N. Rezaei, A. Rezapour, A. Riad, R. K. Ripon, K. E. Rudd, B. Saddik, S. Sadeghian, U. Saeed, M. Saefaei, A. Safary, S. Z. Safi, M. Sahebazzamani, A. Sahebkar, H. Sahoo, S. Salahi, S. Salahi, H. Salari, S. Salehi, H. Samadi Kafil, A. M. Samy, N. Sanadgol, S. Sankararaman, F. Sanmarchi, B. Sathian, M. Sawhney, G. K. Saya, S. Senthilkumaran, A. Seylani, P. A. Shah, M. A. Shaikh, E. Shaker, M. Z. Shakhmardanov, M. M. Sharew, A. Sharifi-Razavi, P. Sharma, R. A. Sheikhi, A. Sheikhy, P. H. Shetty, M. Shigematsu, J. I. Shin, H. Shirzad-Aski, K. M. Shivakumar, P. Shobeiri, S. A. Shorofi, S. Shrestha, M. M. Sibhat, N. B. Sidemo, M. K. Sikder, L. M. L. R. Silva, J. A. Singh, P. Singh, S. Singh, M. S. Siraj, S. S. Siwal, V. Y. Skryabin, A. A. Skryabina, B. Socea, D. D. Solomon, Y. Song, C. T. Sreeramareddy, M. Suleman, R. Suliankatchi Abdulkader, S. Sultana, M. Szócska, S.-A. Tabatabaeizadeh, M. Tabish, M. Taheri, E. Taki, K.-

- K. Tan, S. Tandukar, N. Y. Tat, V. Y. Tat, B. N. Tefera, Y. M. Tefera, G. Temesgen, M.-H. Temsah, S. Tharwat, A. Thiyagarajan, I. I. Tleyjeh, C. E. Troeger, K. K. Umapathi, E. Upadhyay, S. Valadan Tahbaz, P. R. Valdez, J. Van Den Eynde, H. R. Van Doorn, S. Vaziri, G.-I. Verras, H. Viswanathan, B. Vo, A. Waris, G. T. Wassie, N. D. Wickramasinghe, S. Yaghoubi, G. A. T. Y. Yahya, S. H. Yahyazadeh Jabbari, A. Yigit, V. Yiit, D. K. Yon, N. Yonemoto, M. Zahir, B. A. Zaman, S. B. Zaman, M. Zangiabadian, I. Zare, M. S. Zastrozhin, Z.-J. Zhang, P. Zheng, C. Zhong, M. Zoladl, A. Zumla, S. I. Hay, C. Dolecek, B. Sartorius, C. J. L. Murray, and M. Naghavi. Global mortality associated with 33 bacterial pathogens in 2019: a systematic analysis for the Global Burden of Disease Study 2019. *The Lancet*, 400(10369):2221–2248, Dec. 2022.
- [49] Z. Jiang, A. I. Vasil, J. D. Hale, R. E. W. Hancock, M. L. Vasil, and R. S. Hodges. Effects of Net Charge and the Number of Positively Charged Residues on the Biological Activity of Amphipathic  $\alpha$ -Helical Cationic Antimicrobial Peptides. *Biopolymers*, 90(3):369–383, 2008.
- [50] M. A. Jiménez, J. L. Nieto, J. Herranz, M. Rico, and J. Santoro.  $^1\text{H}$  NMR and CD evidence of the folding of the isolated ribonuclease 5061 fragment. *FEBS Letters*, 221(2):320–324, Sept. 1987.
- [51] A. T. Jones. Macropinocytosis: searching for an endocytic identity and role in the uptake of cell penetrating peptides. *Journal of Cellular and Molecular Medicine*, 11(4):670–684, 2007.
- [52] W. L. Jorgensen, D. S. Maxwell, and J. Tirado-Rives. Development and Testing of the OPLS All-Atom Force Field on Conformational Energetics and Properties of Organic Liquids. *Journal of the American Chemical Society*, 118(45):11225–11236, Nov. 1996.
- [53] W. L. Jorgensen and J. Tirado-Rives. The OPLS [optimized potentials for liquid simulations] potential functions for proteins, energy minimiza-

- tions for crystals of cyclic peptides and crambin. *Journal of the American Chemical Society*, 110(6):1657–1666, Mar. 1988.
- [54] J. Koehbach and D. J. Craik. The Vast Structural Diversity of Antimicrobial Peptides. *Trends in Pharmacological Sciences*, 40(7):517–528, July 2019.
- [55] A. Krizsan, D. Volke, S. Weinert, N. Sträter, D. Knappe, and R. Hoffmann. Insect-Derived Proline-Rich Antimicrobial Peptides Kill Bacteria by Inhibiting Bacterial Protein Translation at the 70 S Ribosome. *Angewandte Chemie International Edition*, 53(45):12236–12239, 2014. \_\_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/anie.201407145>.
- [56] P. Kumar, J. N. Kizhakkedathu, and S. K. Straus. Antimicrobial Peptides: Diversity, Mechanism of Action and Strategies to Improve the Activity and Biocompatibility In Vivo. *Biomolecules*, 8(1):4, Jan. 2018.
- [57] M. Lamberty, D. Zachary, R. Lanot, C. Bordereau, A. Robert, J. A. Hoffmann, and P. Bulet. Insect Immunity: CONSTITUTIVE EXPRESSION OF A CYSTEINE-RICH ANTIFUNGAL AND A LINEAR ANTIBACTERIAL PEPTIDE IN A TERMITE INSECT \*. *Journal of Biological Chemistry*, 276(6):4085–4092, Feb. 2001. Publisher: Elsevier.
- [58] C. Landon, H. Meudal, N. Boulanger, P. Bulet, and F. Vovelle. Solution structures of stomoxyn and spinigerin, two insect antimicrobial peptides with an  $\alpha$ -helical conformation. *Biopolymers*, 81(2):92–103, 2006. \_\_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/bip.20370>.
- [59] S. Y. Lau, A. K. Taneja, and R. S. Hodges. Synthesis of a model protein of defined secondary and quaternary structure. Effect of chain length on the stabilization and formation of two-stranded  $\alpha$ -helical coiled-coils. *The Journal of Biological Chemistry*, 259(21):13253–13261, Nov. 1984.

- [60] C.-F. Le, C.-M. Fang, and S. D. Sekaran. Intracellular Targeting Mechanisms by Antimicrobial Peptides. *Antimicrobial Agents and Chemotherapy*, 61(4):e02340–16, Mar. 2017.
- [61] A. R. Leach. *Molecular Modelling: Principles and Applications*. Pearson Education, 2001.
- [62] T. Leist and R. M. Thomas. Synthesis and physicochemical characterization of major fragments of human leukocyte interferon .alpha.1. *Biochemistry*, 23(12):2541–2547, June 1984.
- [63] J. Liu, E. Tennessen, J. Miao, Y. Huang, J. M. Rondinelli, and H. Heinz. Understanding Chemical Bonding in Alloys and the Representation in Atomistic Simulations. *The Journal of Physical Chemistry C*, 122(26):14996–15009, July 2018.
- [64] J. M. Loeffler, D. Nelson, and V. A. Fischetti. Rapid killing of *Streptococcus pneumoniae* with a bacteriophage cell wall hydrolase. *Science (New York, N.Y.)*, 294(5549):2170–2172, Dec. 2001.
- [65] K. Lohner. Membrane-active Antimicrobial Peptides as Template Structures for Novel Antibiotic Agents. *Current Topics in Medicinal Chemistry*, 17(5):508–519, Dec. 2016.
- [66] Z. X. Lu, K. F. Fok, B. W. Erickson, and T. E. Hugli. Conformational analysis of COOH-terminal segments of human C3a. Evidence of ordered conformation in an active 21-residue peptide. *Journal of Biological Chemistry*, 259(12):7367–7370, June 1984.
- [67] G. Maccari, M. Di Luca, and R. Nifosì. In silico design of antimicrobial peptides. *Methods in Molecular Biology (Clifton, N.J.)*, 1268:195–219, 2015.
- [68] F. Madani, S. Lindberg, U. Langel, S. Futaki, and A. Gräslund. Mechanisms of cellular uptake of cell-penetrating peptides. *Journal of Biophysics (Hindawi Publishing Corporation: Online)*, 2011:414729, 2011.

- [69] S. C. Mansour, O. M. Pena, and R. E. Hancock. Host defense peptides: front-line immunomodulators. *Trends in Immunology*, 35(9):443–450, Sept. 2014.
- [70] D. Marion, M. Zasloff, and A. Bax. A two-dimensional NMR study of the antimicrobial peptide magainin 2. *FEBS letters*, 227(1):21–26, Jan. 1988.
- [71] S. J. Marrink, H. J. Risselada, S. Yefimov, D. P. Tieleman, and A. H. De Vries. The MARTINI Force Field: Coarse Grained Model for Biomolecular Simulations. *The Journal of Physical Chemistry B*, 111(27):7812–7824, July 2007.
- [72] G. J. Martyna, D. J. Tobias, and M. L. Klein. Constant pressure molecular dynamics algorithms. *The Journal of Chemical Physics*, 101(5):4177–4189, Sept. 1994.
- [73] G. J. Martyna, M. E. Tuckerman, D. J. Tobias, and M. L. Klein. Explicit reversible integrators for extended systems dynamics. *Molecular Physics*, 87(5):1117–1157, Apr. 1996. Publisher: Taylor & Francis \_eprint: <https://doi.org/10.1080/00268979600100761>.
- [74] S. Mayor and R. E. Pagano. Pathways of clathrin-independent endocytosis. *Nature Reviews. Molecular Cell Biology*, 8(8):603–612, Aug. 2007.
- [75] F. A. Mercurio, A. Scaloni, S. Caira, and M. Leone. The antimicrobial peptides casocidins I and II: Solution structural studies in water and different membrane-mimetic environments. *Peptides*, 114:50–58, Apr. 2019.
- [76] R. K. Mishra, L. Fernández-Carrasco, R. J. Flatt, and H. Heinz. A force field for tricalcium aluminate to characterize surface properties, initial hydration, and organically modified interfaces in atomic resolution. *Dalton Trans.*, 43(27):10602–10616, 2014.



- [77] L. Moroder, B. Filippi, G. Borin, and F. Marchiori. Studies on cytochrome c. XI. Circular dichroism studies on synthetic peptides related to the C-terminal region of baker's yeast iso-1-cytochrome c. *Biopolymers*, 14(10):2075–2093, 1975. \_\_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/bip.1975.360141008>.
- [78] M. Mura, J. Wang, Y. Zhou, M. Pinna, A. V. Zvelindovsky, S. R. Denison, and D. A. Phoenix. The effect of amidation on the behaviour of antimicrobial peptides. *European Biophysics Journal*, 45:195–207, 2016.
- [79] J. W. Neidigh, R. M. Fesinmeyer, K. S. Prickett, and N. H. Andersen. Exendin-4 and Glucagon-like-peptide-1: NMR Structural Comparisons in the Solution and Micelle-Associated States. *Biochemistry*, 40(44):13188–13200, Nov. 2001. Publisher: American Chemical Society.
- [80] L. T. Nguyen, E. F. Haney, and H. J. Vogel. The expanding scope of antimicrobial peptide structures and their modes of action. *Trends in Biotechnology*, 29(9):464–472, Sept. 2011.
- [81] P. Nicolas. Multifunctional host defense peptides: intracellular-targeting antimicrobial peptides. *The FEBS journal*, 276(22):6483–6496, Nov. 2009.
- [82] M. Parrinello and A. Rahman. Polymorphic transitions in single crystals: A new molecular dynamics method. *Journal of Applied Physics*, 52(12):7182–7190, Dec. 1981.
- [83] J. Pei, L. F. Song, and K. M. J. Merz. Pair Potentials as Machine Learning Features. *Journal of Chemical Theory and Computation*, 16(8):5385–5400, Aug. 2020. Publisher: American Chemical Society.
- [84] M. C. Pena, M. Rico, M. A. Jiménez, J. Herranz, J. Santoro, and J. Nieto. Conformational properties of the isolated 123 fragment of human hemoglobin -chain. *Biochimica et Biophysica Acta (BBA) - Protein Structure and Molecular Enzymology*, 957(3):380–389, Dec. 1988.

- [85] A. F. Pereira, V. Piccoli, and L. Martínez. Trifluoroethanol direct interactions with protein backbones destabilize  $\alpha$ -helices. *Journal of Molecular Liquids*, 365:120209, Nov. 2022.
- [86] A. Pfalzgraff, K. Brandenburg, and G. Weindl. Antimicrobial Peptides and Their Therapeutic Potential for Bacterial Skin Infections and Wounds. *Frontiers in Pharmacology*, 9, 2018.
- [87] M. Pinto, J. J. Perez, and J. Rubio-Martinez. Molecular dynamics study of peptide segments of the BH3 domain of the proapoptotic proteins Bak, Bax, Bid and Hrk bound to the Bcl-xL and Bcl-2 proteins. *Journal of Computer-Aided Molecular Design*, 18(1):13–22, Jan. 2004.
- [88] Y. Pouny, D. Rapaport, A. Mor, P. Nicolas, and Y. Shai. Interaction of antimicrobial dermaseptin and its fluorescently labeled analogues with phospholipid membranes. *Biochemistry*, 31(49):12416–12423, Dec. 1992.
- [89] D. Roccatano, G. Colombo, M. Fioroni, and A. E. Mark. Mechanism by which 2,2,2-trifluoroethanol/water mixtures stabilize secondary-structure formation in peptides: a molecular dynamics study. *Proceedings of the National Academy of Sciences of the United States of America*, 99(19):12179–12184, Sept. 2002.
- [90] V. Salmaso and S. Moro. LINFLUENZA DEL TEMPO NEL COMPORTAMENTO MOLECOLARE: UNINTRODUZIONE ALLA DINAMICA MOLECOLARE. *La Chimica e l'Industria online*, (2):8, 2018.
- [91] M.-A. Sani and F. Separovic. How Membrane-Active Peptides Get into Lipid Membranes. *Accounts of Chemical Research*, 49(6):1130–1138, June 2016.
- [92] F. Savini, S. Bobone, D. Roversi, M. L. Mangoni, and L. Stella. From liposomes to cells: Filling the gap between physicochemical and microbiological studies of the activity and selectivity of host-

- defense peptides. *Peptide Science*, 110(5):e24041, 2018. \_\_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/pep2.24041>.
- [93] K. H. Scheit, E. S. Reddy, and P. M. Bhargava. Seminaplasmin is a potent inhibitor of *E. coli* RNA polymerase in vivo. *Nature*, 279(5715):728–731, June 1979.
- [94] C. N. Schutz and A. Warshel. What are the dielectric constants of proteins and how to validate electrostatic models? *Proteins: Structure, Function, and Bioinformatics*, 44(4):400–417, 2001. \_\_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.1106>.
- [95] T. M. A. Shafee, F. T. Lay, T. K. Phan, M. A. Anderson, and M. D. Hulet. Convergent evolution of defensin sequence, structure and function. *Cellular and Molecular Life Sciences*, 74(4):663–682, Feb. 2017.
- [96] W. Shalongo, L. Dugad, and E. Stellwagen. Distribution of Helicity within the Model Peptide Acetyl(AAQAA)3amide. *Journal of the American Chemical Society*, 116(18):8288–8293, Sept. 1994.
- [97] K. Shimazaki, T. Tazume, K. Uji, M. Tanaka, H. Kumura, K. Mikawa, and T. Shimo-oka. Properties of a Heparin-binding Peptide Derived from Bovine Lactoferrin. *Journal of Dairy Science*, 81(11):2841–2849, Nov. 1998.
- [98] S. Stephan, M. T. Horsch, J. Vrabec, and H. Hasse. MolMod an open access database of force fields for molecular simulations of fluids. *Molecular Simulation*, 45(10):806–814, July 2019.
- [99] H. Sun, S. J. Mumby, J. R. Maple, and A. T. Hagler. An ab Initio CFF93 All-Atom Force Field for Polycarbonates. *Journal of the American Chemical Society*, 116(7):2978–2987, Apr. 1994.
- [100] D. Takahashi, S. K. Shukla, O. Prakash, and G. Zhang. Structural determinants of host defense peptides for antimicrobial activity and target cell selectivity. *Biochimie*, 92(9):1236–1241, Sept. 2010.

- [101] A. Tossi, L. Sandri, and A. Giangaspero. Amphipathic, alpha-helical antimicrobial peptides. *Biopolymers*, 55(1):4–30, 2000.
- [102] A. R. Van Buuren and H. J. C. Berendsen. Molecular dynamics simulation of the stability of a 22-residue  $\alpha$ -helix in water and 30% trifluoroethanol. *Biopolymers*, 33(8):1159–1166, 1993. \_\_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/bip.360330802>.
- [103] R. Walgers, T. C. Lee, and A. Cammers-Goodwin. An Indirect Chaotropic Mechanism for the Stabilization of Helix Conformation of Peptides in Aqueous Trifluoroethanol and Hexafluoro-2-propanol. *Journal of the American Chemical Society*, 120(20):5073–5079, May 1998.
- [104] J. Wang, R. M. Wolf, J. W. Caldwell, P. A. Kollman, and D. A. Case. Development and testing of a general amber force field. *Journal of Computational Chemistry*, 25(9):1157–1174, 2004. \_\_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.20035>.
- [105] World Health Organization. *Global action plan on antimicrobial resistance*. World Health Organization, Geneva, 2015. Section: The Estonian version published by WHO Office for Europe.
- [106] M. Wu, E. Maier, R. Benz, and R. E. W. Hancock. Mechanism of Interaction of Different Classes of Cationic Antimicrobial Peptides with Planar Bilayers and with the Cytoplasmic Membrane of *Escherichia coli*. *Biochemistry*, 38(22):7235–7242, June 1999. Publisher: American Chemical Society.
- [107] Y. Yamamoto, T. Ohkubo, A. Kohara, T. Tanaka, T. Tanaka, and M. Kikuchi. Conformational requirement of signal sequences functioning in yeast: circular dichroism and  $^1\text{H}$  nuclear magnetic resonance studies of synthetic peptides. *Biochemistry*, 29(38):8998–9006, Sept. 1990.
- [108] M. Zgarbová, M. Otyepka, J. ponar, P. Hobza, and P. Jureka. Large-scale compensation of errors in pairwise-additive empirical force fields:

- comparison of AMBER intermolecular terms with rigorous DFT-SAPT calculations. *Physical Chemistry Chemical Physics*, 12(35):10476–10493, Aug. 2010. Publisher: The Royal Society of Chemistry.
- [109] L. Zhang, A. Rozek, and R. E. Hancock. Interaction of cationic antimicrobial peptides with model membranes. *The Journal of Biological Chemistry*, 276(38):35714–35722, Sept. 2001.
- [110] H.-D. Zucht, M. Raida, K. Adermann, H.-J. Mägert, and W.-G. Forssmann. Casocidin-I: a casein-s2 derived peptide exhibits antibacterial activity. *FEBS Letters*, 372(2):185–188, Sept. 1995.