**Ora** apriamo la cartella 03-analysis

Da importare:

import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline

def convert\_image(f="plot.eps",o="plot.png"):

    '''

    Function that allows you to convert a file from esp format to png format

    '''

    from PIL import Image

    eps\_image = Image.open(f)

    eps\_image.load(scale=10)

    eps\_image.save(o)

**TUTORIAL 3 (Simulated annealing, accoppiamento termico controllato)**

In primis, facciamo un po’ di analisi a partire dalla dinamica che abbiamo fatto precedentemente con il riscaldamento senza position restrains e un riscaldamento a 500K.

Possiamo diagrammare la distribuzione con gmx analyze sull’ultima parte della curva di T.

Guardiamo la pressione, selezionandola da gmx energy e poi diagrammandola con xmgrace. Siccome non abbiamo accoppiato in pressione i valori che saltano fuori sono privi di senso sul microstato ma genera molto variabilità.

L’energia però non ci racconta nulla sulla struttura, dobbiamo guardare rmsd e rmsf

**RMSD** (è una radice quadrata non sarà mai negativa) si calcola l’RMSD della proteina rispetto a un istante (uno qualsiasi della traiettoria). Tra gli atomi del riferimento e gli atomi della mia proteina, deve esserci una corrispondenza, devono avere numero di atomi simile.

Gmx rms -f md\_nojump.trr -s md.tpr -o rmsd\_Ca.trr

E’ una media pesata di tutti gli spostamenti delle particelle rispetto alla posizione iniziale (info sulla deformazione), tolti gli effetti della rototraslazione rigida (li potremmo considerare se guardassi due pezzi separati). Facciamo un fitting sui c alpha e l’rmsd lo faccio o solo sul backbone (ignoro le distorsioni dei gruppi laterali) o su tutta la proteina. Ci aspetteremmo che fosse un po’ più piatto alla fine se andasse all’equilibrio. Vediamo che sono rmsd un po’ più alti della simu con posre, cresce rispetto a 0 e poi si stabilizza un po’. Possiamo ipotizzare che si sia raggiunto un equilibrio strutturale, ma non è detto perché magari una struttura fluttua tanto ma magari fluttua tra due minimi, bisogna fare sempre un’analisi comparitiva tra più grandezze.

Consider a 500 ns long simulation and the RMSD with respect to the initial conformation. The RMSD stabilized around 1 nm starting from 100 ns. Can you say that the conformations at 450 ns and 200 ns are for sure similar? quindi che arrivano a convergenza

The answer is no! That information suggests you that the conformation are similar, but the actual information is that they are "equally distant" from the initial configuration of the system (reference frame). Therefore, it can happen that:

RMSD(200ns,450ns)>>RMSD(0ns,450ns)=RMSD(0ns,200ns)

Posso calcolare l’RMSD all’inizio e alla fine della simulazione.

**Comando gmx rms**

%%bash

cd 03-analysis/

echo -e "Backbone\nBackbone" | gmx rms -s penetratin.pdb -f penetratin.xtc -o rmsd.xvg -tu ns

echo -e "System" | gmx trjconv -s topol.tpr -f penetratin.xtc -dump 9 -o struct9ns.pdb -tu ns

echo -e "Backbone\nBackbone" | gmx rms -s struct9ns.pdb -f penetratin.xtc -o rmsd\_9ns.xvg -tu ns

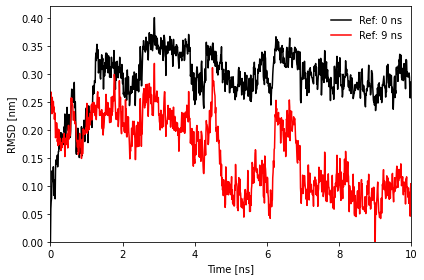
il primo gruppo che ti chiede è quello su cui allineiamo la traiettoria, il secondo backbone è quello su cui si calcola l’RMSD.

Con la prima riga, prendo come riferimento 0 ns (di default).

Nella seconda riga ho cambiato l’istante di riferimento a 9 ns (o cmq il frame più vicino a 9ns). Nella terza riga ho ricalcalcolato ma rispetto ad un istante diverso (9 ns).

# Now, save the plot (if you want)

fig.savefig('03-analysis/rmsd\_comparison.jpg',dpi=300,facecolor='white')



dalla figura vedo che sia per il riferimento a 0 ns sia per quello a 9 ns, il valore di RMSD rimane circa costante dopo i 7ns, questo vuol dire che dai 7ns in poi abbiamo una simulazione stabile.

Facciamo anche un rmsd rispetto all’istante finale à estraiamo con gmx trjconv -dump 100 l’ultimo istante della simu che salviamo in md\_end.pdb

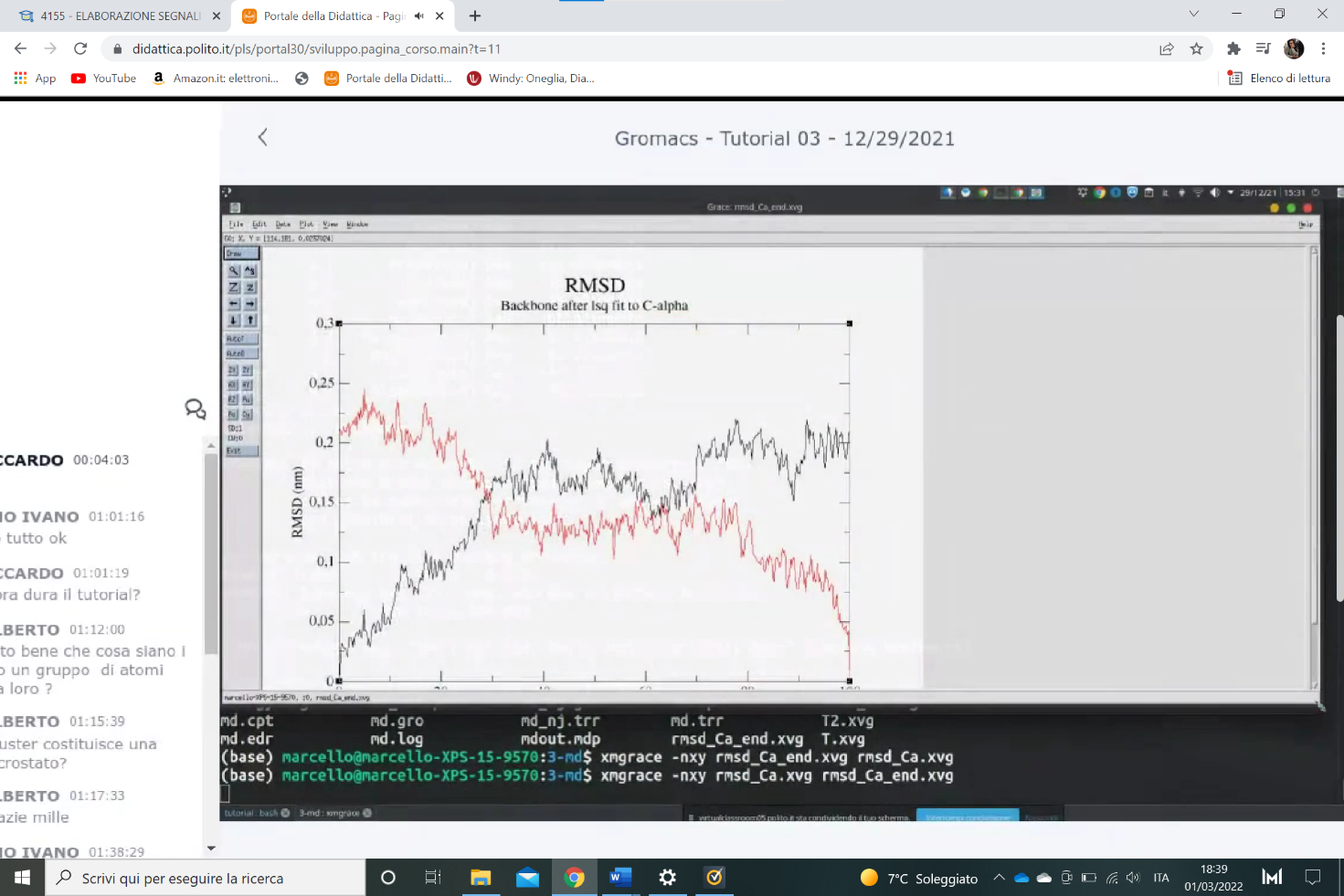
Non è necessario sempre ricrearsi il tpr (che contiene le masse) perché gromacs è in grado di calcolarselo.

Gmx rms -f md\_nj.trr -s md\_end.pdb -o rmsd\_Ca\_end.xvg

Così mi sono calcolata il rmsd rispetto all’istante finale, sempre fittando su C\_alpha e calcolando il rmsd del backbone.

Posso rappresentare entrambi con xmgrace -nxy rmsd\_Ca.xvg rmsd\_Ca\_end.xvg

La curva nera è rispetto al valore iniziale, rossa rispetto al finale

Si vede che il primo rmsd sarebbe potuto essere ingannevole, perché non raggiungo effettivamente l’equilibrio come vedo dal secondo grafico.

Posso farlo anche rispetto ad un valore medio degli ultimi 10 frame, faccio una media della struttura dei carboni alpha che in realtà non esiste ma ci serve solo per calcolare la deformazione media di tale struttura negli ultimi 10 ps.

STRUTTURA MEDIA IN UN INTERVALLO **RMSF**

È sempre una media quadratica. Qua perdiamo l’info temporale!!

È una media nel tempo: per ogni istante temporale calcola la distanza dell’atomo di interesse rispetto alla sua posizione media nella traiettoria. Ci definisce quanto in media l’atomo ha oscillato durante la simulazione.

Nel caso delle proteine consideriamo un gruppo di atomi, cioè un residuo dell’aa. Per farlo, calcoliamo la media delle singole fluttuazioni di ogni atomo; oppure (metodo che usiamo noi) calcolo la fluttuazione del C-alpha che si considera rappresentativo dell’intero residuo.

Alto RMSF indica zone più mobili, zone stabili hanno basso RMSF.

L’output -ox ti da’ le coordinate medie, e da queste calcolo l’RMSF

Per fare la media uso gmx rmsf -s md.tpr -f md\_nojump.trr -b 90 -ox à crea una struttura media fittizia, scelgo la proteina sennò non funge, me la salva in xaver.pdb e le vedo su vdm come punti vicinisimi.

-res serve per calcolare le fluttuazioni medie di ogni residuo.

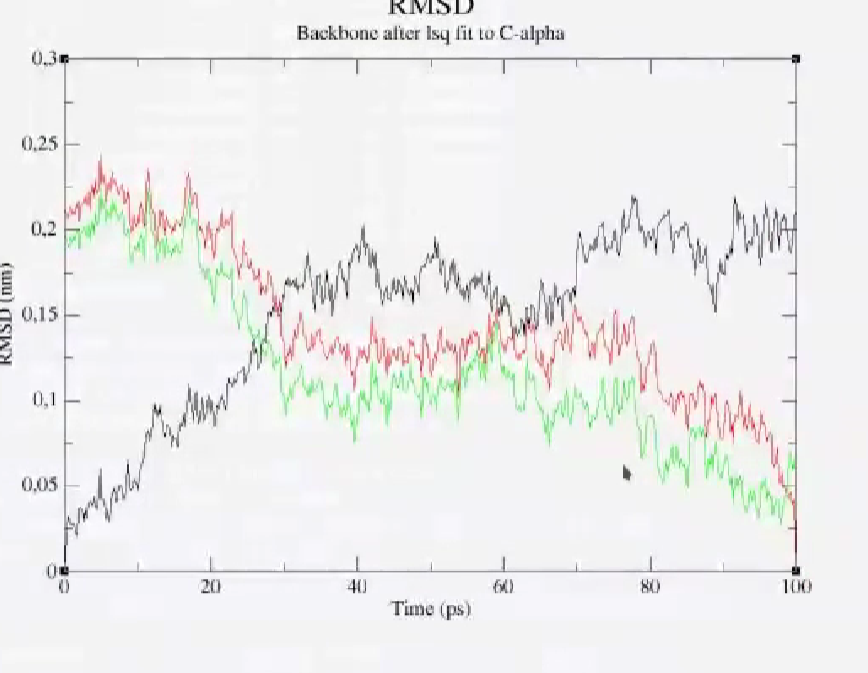
%%bash

cd 03-analysis/

echo "C-alpha" | gmx rmsf -f penetratin.xtc -s topol.tpr -o rmsf.xvg -res

metto -res perché così l’output dell’RMSF, i valori sono il numero del residuo (se non l’avessi messo, sarebbe stato il numero dell’atomo)

nella figura vedo i due rmsd di prima e l’rmsf, due strutture sono simili, prima e dopo i 7 ns, anche se la media è stata calcolata solo da 8 a 9 ns. questo perché la conformazione è stabile.

Gmx rms -f md\_nj.trr -s xaver.pdb -o rmsd\_xaver.trr

Xmgrace -nxy rmsd\_Ca.xvg rmsd\_xaver.xvg rmsd\_Ca\_nd.xvg

Alla fine il rosso crolla a zero, il verde sembra quasi all’equilibrio alla fine. *Certo che va a convergenza tra 90-100 visto che è di quel tratto che ho fatto la struttura media*, guardo allora anche un po’ nell’intorno. In ogni caso, è più conveniente fare il confronto rispetto alla struttura finale perché quella iniziale ci da poche info che potrebbero essere fallaci.

**RADIUS OF GYRATION**

Info sulla dimensione del nostro sistema, sulla distanza radiale a cui la massa del sistema dovrebbe concentrarsi in modo da avere lo stesso momento d’inerzia. È usata per avere info sulle dimensioni del sistema. È sempre una media quadratica.

%%bash

cd 03-analysis/

echo "Protein" | gmx gyrate -f penetratin.xtc -s topol.tpr -o rog.xvg

come fare un istogramma

# histogram

ax2 = fig.add\_subplot(122,)

ax2.hist(rog,color='k',density=True,alpha=0.5) #VUOL dire che l’area sotto l’istogramma è uguale a 1. Devo metterlo per forza perché altrimenti non ho una distribuzione di probabilità.

ax2.set\_ylabel(r"Probability density")

ax2.set\_xlabel(r"$R\_g$ [nm]")

fig.tight\_layout()

fig.savefig('03-analysis/radius\_of\_gyration.png',dpi=300,facecolor='white')

Immagine che contiene grafico

Descrizione generata automaticamente

Guardando la figura, a 6 ns il sistema si è compattato o si è allargato?

Il sistema si è compattato

**RAMACHANDRAN PLOT**

The allowed combinations of torsional angles ψ and φ for a couple of residues are illustrated in the Ramachandran plot.

Gli angoli sono calcolati tra due aa. Ma non tutte le combinazioni degli angoli sono ammesse: le catene laterali degli aa vicini si respingono e quindi non tutte le zone sono accessibili.

Se dopo la minimizzazione gli aa sono tutti in zone non accessibili, c’è qualcosa che non va.

Traiettoria.xtc e .tpr devono avere lo stesso numero di atomi.

!gmx rama -f 03-analysis/penetratin.xtc -s 03-analysis/topol.tpr -o 03-analysis/rama.xvg

Ho un’info temporale e un’info sui residui: ottengo un file con un angolo phi e l’angolo pshi, il nome del residuo per il tempo 1.. così finchè non finisco il tempo 1, poi va avanti al tempo 2, 3 …. Fino all’istante di tempo N.

Per estrapolare le info e plottarle usa questo codice:

# Load data

# You have to load separately floating point numbers and string

phi, psi = np.loadtxt('03-analysis/rama.xvg',comments=['@','#'],usecols=(0,1),unpack=True)

residues = np.loadtxt('03-analysis/rama.xvg',comments=['@','#'],usecols=(2),dtype=str)

#i nomi dei residui (str) separati dalle cose numeriche

# This line searches all the points in the array where the first residue of the protein is present

# Therefore, it contains the information to separate information of different frames

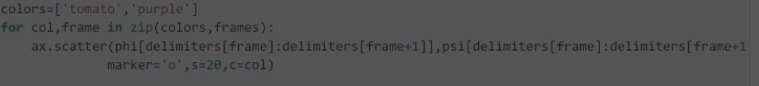
delimiters = np.squeeze(np.argwhere(residues == residues[0])) #cerco tutte le posizioni in residues che sono uguali al primo. Cioè tutti i punti in cui c’è il primo aa.

Immagine che contiene grafico, mappa

Descrizione generata automaticamente

Il grafico che vediamo non racconta l’evoluzione della simulazione, è fatto in due istanti di tempo (frame 0 e frame 800). Racconta la distribuzione degli angoli.

Ogni punto rappresenta la coppia di angoli associati ad un particolare aa, tutti i punti sono tutti gli aa.



Ricordi dei metodi per analizzare la struttura secondaria? Il ramachandran plot, perché ci dice come gli aa appartengono alla proteina, ma non ci da’ troppe info sulla struttura terziaria, anzi quasi nulla.

**DISTANZA TRA ESTREMI**

**Comando gmx distance**: per calcolare la distanza tra gli estremi della proteina.

Definire un indice con i gruppi A e B, diamo un nome e poi forniamo l’indice a gromacs.

%%bash

cd 03-analysis

echo "r 43

name 17 E1

r 56

name 18 E2

q" | gmx make\_ndx -f penetratin.pdb -o index.ndx

gmx distance -f penetratin.xtc -s topol.tpr -select 'com of group "E1" plus com of group "E2"' \

-n index.ndx -oav distance.xvg

**LEGAMI AD IDROGENO**

**Comando gmx hbond**

%%bash

cd 03-analysis

echo -e "Protein \n Water \n " | gmx hbond -f penetratin.xtc -s topol.tpr -num hb\_water\_protein.xvg

echo -e "Protein \n Protein \n " | gmx hbond -f penetratin.xtc -s topol.tpr -num hb\_protein\_protein.xvg

-num: numero di legami ad idrogeno in un istante di tempo

Calcolo il legame tra la prot e l’acqua o tra la proteina e sé stessa.

**SECONDARY STRUCTURES ASSESMENT**

Struttura secondaria, angoli torsionali.

Ci sono diverse metodologie: una di queste è basata sul sotfware DSSP, implementato dal comando **gmx do\_dssp.**

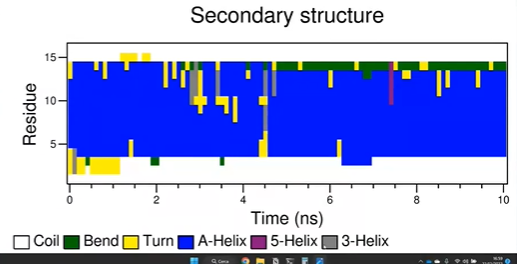
Si determina la struttura più probabile in una regione della proteina.

%%bash

cd 03-analysis/

echo "Protein" | gmx do\_dssp -s penetratin.pdb -f penetratin.xtc -tu ns -dt 0.1

dt 0.1: 1 frame ogni 100 ns (è per una rappresentazione grafica)

 ho usato dei comandi per metterla in formato .png

**CLUSTERIZZARE**

C’è un altro strumento utile per studiare le convergenze: **calcola le distanze di ogni estremo con tutti gli altri, non avrò più una curva ma una matrice** à ci permette poi di fare dei cluster a seconda delle distanze che abbiamo. Una struttura campionata a 20ps magari sarà molto simile a quella campionata a 80ps.

I clustering sono l’analisi dei microstati esplorati istante dopo istante di tempo, i **clusters non sono di parti diverse della proteina ma di istanti temporali diversi** à quindi ogni cluster è una configurazione intera non una parte di essa.

Il metodo *gromos* forma dei cluster più grossi senza che ci sia un’ottimizzazione sui più vicini, il *linkage* sottrae a dei grandi cluster per vicinanza per far si che le strutture vicine siano il più omogenee possibili. (manco deriu l’ha capito e sa spiegarlo)

Il cluster più popolato solamente raccoglie le strutture più significative.

**Il centroide è la configurazione media del cluster.**

! Bisogna definire la soglia per cui si discrimina tra un cluster e l’altro, 0.1 nm di default. Se la distanza tra due è maggiore vanno in clusters diversi.

Gmx cluster -f md\_nj.trr -s md.tpr

Scegliamo sempre i C-alpha per clusterizzare

CI restituisce 3 file che sono cluster.log; rmsd-clust.xpm e rmsd-dist.xvg.

Guardando il log vediamo che ha trovato un solo cluster, quindi che le strutture sono tutte sufficientemente vicine, tutte vicine al centroide.

Se cambiamo il cutoff e clusterizziamo rispetto alla proteina

Gmx cluster -f md\_nj.trr -s md.tpr -cutoff 0.07

Troviamo 8 cluster nel log che vengono elencate nella matrice con diverse popolarità. In questo caso, i cluster si muovono e sono raggruppati per vicinanza di frame, difficilmente quello 20 sarà vicino ad 80 ps ma solo perché la simulazione è molto breve.

CALCOLO DEI CENTROIDI

Gmx cluster -f md\_nj.trr -s md.tpr -cutoff 0.07 -cl cluster.pdb

1 1

Mi restituisce 8 pdb (cluster1.pdb ecc) che mi danno il centroide degli 8 clusters creati prima

Per plottare solo tutti si vmd faccio vmd clusters.pdb

**2:25:24**

**ANNEALING (Modifico la temperatura a mio piacimento durante la simu, come se mettessi una forzante che ogni tanto scalda il sistema)**

Annealing=ricottura.

L’aumento della temperatura modifica la possibilità di esplorazione dello spazio degli stati, se io scaldo così tanto mi sollevo dal minimo e mi permette di muovermi ovunque ma non rimane traccia di dove sono andato, perché non incontro mai minimi, non incontro mai barriere. Con l’annealing, ogni tot, forzo la temperatura a cambiare (si abbassa) e io vengo riportato ‘’a terra’’ (300K, gli spazi che mi interessano) e posso esplorare quella zona di minimo, e così via. Scaldando io sono in grado di saltare le barriere ma non posso esplorare veramente lo spazio perché mi appare tutto uguale.

La tecnica dell’annealing è utile quando il sistema si blocca in dei minimi e non ci permette di esplorare lo spazio degli stati, vedendo la dinamica la struttura alpha elica viene persa (soprattutto quando siamo a 1000K) quando siamo a 10K il sistema è congelato e in seguito riprenderà a muoversi.

Vogliamo far seguire alla md, un andamento di temperatura: da freddo, si scalda a 300K, si equilibra, si fa la simulazione, si riscalda di nuovo, si raggiunge un nuovo equilibrio e si rifa la simu.

Come scelgo le temperature? Io sto cambiando la temperatura a tutto il sistema, quindi non riesco facilemente a capire che temp devo raggiungere.

Bisogna dare il tempo al sistema di raggiunger la temp.

Nel flow chart, siamo nella fase di production!!

Quindi inizialmente, devo creare la topologia (si fa come sempre)

Poi creo una box e faccio la solvatazione e lo neutralizzo con gli ioni.

In seguito faccio la minimizzazione, ricordati che c’è sempre il file .tpr (che contiene i 3 file).

Da qua vedo che ho raggiunto un minimo: il potenziale negativo.

Poi faccio il posres: equilibrio il sistema ad una data temp

Nel file .mdp ho due valori per la temp.

Poi procedo con gli step standard, facendo anche la trajconv. Ora ho un sistema “rilassato” cioè si è equilibrato ad una certa temp.

Ora con questo sistema posso iniziare l’annealing.

Immagine che contiene testo

Descrizione generata automaticamenteRicordiamoci che c’è il tau di tempo che serve al sistema per raggiungere la temperatura voluta, quindi quando lo imposto, devo mettere un tau adeguata!

Quindi io devo dare una serie di tempi e di temperature all’.mdp. occhio che nell’mdp io avevo due accoppiamenti in temp (water non-water), quindi quando specifico tutte le temp e i tempi, devo farlo per entrambe le robe.

Tra i due istanti di tempo, gromacs crea una funzione lineare a tratti: la resT del termostato viene cambiata linearmente.

Nella tabella, vedo che c’è la possibilità di fare un annealing periodico: quando finisce, riprende daccapo.

Immagine che contiene testo

Descrizione generata automaticamenteESEMPIO

Ricordati di mettere due volte ogni cosa!

Nei tempi e nelle temperature avrò 6 valori, 3 per ogni gruppo.

REMEMBER: Check the summary printed by gmx grompp if you are unsure!



Questo comando serve per non far comparire l’output sulla cella sotto, ma farlo comparire nel file, così è più comodo.

ORA VEDIAMO L’ANALISI

Vediamo da NGLview che andando avanti con la temp, la proteina perde la sua struttura secondaria, diventando un coil.

Dopo aver fatto la simulazione, non arriviamo allo stesso minimo di prima!

Ora possiamo plottare il grafico della temperatura, così ho un’info qualitativa.

Voglio però avere anche un’info quantitativa e uso il clustering.

3:17

Questa procedura va a sostituire il riscaldamento spiegato sopra.

Apro il template.mdp per cambiarlo in ann.mdp e modificare i parametri della simu:

* Non tocco la prima parte, usiamo md come integratore
* voglio fare una simu di 150 ps (0.0015) quindi 75000 steps
* lascio come temperatura 300
* tolgo gen vel
* Temperature coupling v-rescale
* tau 0.1
* annealing single single (abbiamo due gruppi per accoppiamento termico??)
* annealing points ne mettiamo 6 6, per 6 punti cambio la temperatura durante la simu. Rispetto al tempo, parto da 0 ps a 300 K, voglio il primo cambio a 10 ps e che la annealing-temp sia 1000K; poi a 50 ps voglio 1000K, a 80 sia a 10 K, 100 sto a 10K, 120 sto a 300K. Da 120 ps lui andrà a 300K.

Questi valori devo scriverli intervallati sulla riga di ann-time e ann-temp e poi ripeterli per il secondo gruppo un po’ più a destra. Sono due blocchi da 6.

Faccio gmx grompp -f ann.mdp -c posre.gro -p topol.top -o annealing.tpr -maxwarn 2

Qui ho usato il file con il position restrains ma avrei potuto benissimo usare un altro gro

Gmx mdrun -s annealing.tpr -v -deffnm annealing

RMSD

Se calcoliamo l’rmsd vediamo come si deforma rispetto all’inizio, cresce e poi si appiattisce, varia pochissimo quando è congelata.

STRUTTURE SECONDARIE

Infine, possiamo fare le strutture secondarie con gmx do\_dssp -f ann\_mol.trr -s annealing.tpr -dt 10

Poi xpm2ps e poi lo vedo con okular, vedo come la struttura secondaria a 1000K si perde.

Guardando il grafico colorimetrico in okular: si vede nettamente come oltre una certa temperatura la struttura ad elica si perda e si vada verso un coil, nel tempo potrebbe riuscire a recuperare la struttura ad alpha elica ma non nel tempo che gli diamo noi perché è troppo basso.

Posso fare clusters alle diverse temperature e posso ottenere quantità diversi.