Corso di Laurea Magistrale in Scienze Chimiche, Curriculum "Struttura, dinamica e reattività chimica"

Struttura e Dinamica Molecolare di Sistemi Biologici - 2010/2011

Laboratorio 1 - (15 novembre 2010)

---

1 Preliminari

1.1 assegnazione account

- Assegnazione nome utente e password

1.2 ripasso UNIX[1] e configurazione utente

1. Aprire una sessione facendo login con il nome del vostro utente e password.
2. Aprire un terminale usando il menù principale del Desktop.
3. Ripassare la "sintassi generale di un comando" [tab. 4.1]
4. Ripassare il "percorso assoluto e relativo" per indicare file e directory [fig. 4.1]
5. Ripassare i comandi per la "gestione essenziale di file e directory" [tab. 4.2]
6. Copiare il file `~signorini/etc/bashrc.global` su `~/.bashrc`
7. Copiare il file `~signorini/etc/init.el` su `~/.emacs`
8. uscire dal terminale e rientrare

1.3 Editor di testo: emacs[2]

1. differenza tra editor di testo e word processor [3]
2. Lanciare emacs, col comando
   `emacs`
3. provare a uscire
4. rientrare lanciando il comando in background
   `emacs &`
5. osservare le varie aree dello schermo
   (a) barra menu
   (b) area di lavoro: il buffer
   (c) riga di stato (Mode Line)
   (d) minibuffer
6. menu, scorciatoie e comandi "complessi"
7. help: Apropos, Key, Whereis (locate); Tutorial
8. differenza tra buffer e file
9. editare il file `~/.plan`
   (a) inserire n.o matricola e codice corso di laurea
10. "editare" una directory: dired

---

[1] fare riferimento alle dispense sul "sistema operativo" in `http://www.chim.unifi.it/u/signo/did/unix`
[2] sez. 5.1
[3] sez. 4.5.1

2 Piano di lavoro

Nelle nostre esercitazioni studieremo le proprietà di una serie di piccoli peptidi che hanno una certa propensione ad assumere la strutture di "forcina β" (β-*hairpin*). Essi sono derivati da un frammento del dominio B1 della proteina G (PDB code `2gb1`); il modello è costituito da un 10-peptide denominato GPM12, la cui struttura secondaria in soluzione però non è stata risolta

GPM12	Gly1-Tyr2-Asp3-**Asp4-Ala5-Thr6-Lys7**-Thr8-Phe9-Gly10	`GYDDATKTFG`

Possiamo analizzare i seguenti mutanti di GPM12:

D4P	Gly1-Tyr2-Asp3-**Pro4-Ala5-Thr6-Lys7**-Thr8-Phe9-Gly10	`GYDPATKTFG`
K7G	Gly1-Tyr2-Asp3-**Asp4-Ala5-Thr6-Gly7**-Thr8-Phe9-Gly10	`GYDDATGTFG`
D4P/K7G	Gly1-Tyr2-Asp3-**Pro4-Ala5-Thr6-Gly7**-Thr8-Phe9-Gly10	`GYDDATGTFG`

Di quest'ultimo doppio-mutante è nota la struttura in soluzione (PDB code `2e4e`).

Costruiamo le proteine in una conformazione (il più possibile) lineare, in modo da verificare se evolvono spontaneamente verso la struttura a forcina, e quali sono i fattori che condizionano questa evoluzione (sequenza primaria, presenza del solvente, condizioni termodinamiche (temperatura), ...).

Il piano di lavoro è il seguente:

1. costruzione della molecola
2. minimizzazione dell'energia
3. prima simulazione con controllo della conservazione di E
4. simulazione nel vuoto
5. simulazione nel solvente
6. analisi dei risultati della simulazione classica

---

3 programma VMD

Una versione funzionante del programma VMD dovrebbe essere già attiva (con la configurazione creata al punto precedente, il comando `'vmd'` dovrebbe lanciare `~signorini/bin/vmd`)

3.1 Costruzione di proteine: `molefacture`

Usare la versione 1.8.7 di VMD perché il `molefacture` della 1.8.6 non funzione (prolina sbagliata).

Menu `Extensions / Modeling / Molefacture` : partire da molecola vuota

Menu `Build / Protein Builder` : aggiungere i residui configurando gli angoli di Ramachandran a 180° e −180° (straight).

Alla fine modificare i residui terminali, aggiungendo un `H` all'`N` terminale ("*Add hydrogen to selected atom*") e un ossidrile al C terminale (Menu `Build / Replace hydrogen with fragment`)

Salvare come PDB e ricaricare la molecola in VMD. È possibile che alcuni atomi di catene laterali siano venuti troppo vicini e che ce li presenti come legati. Invece di cercare di muovere pezzi della molecola con Molefacture, proviamo a minimizzare l'energia potenziale in ORAC. In ogni caso si possono cancellare gli atomi di `H` perché `ORAC` li riassegna automaticamente.

4 programma ORAC

Una versione funzionante del programma ORAC dovrebbe essere già attiva (il comando `'orac'` dovrebbe lanciare `~signorini/bin/orac`). Altrimenti, eseguire installazione e test come di seguito:

4.1 installazione

Scaricare la versione aggiornata del programma ORAC dal sito ufficiale `http://www.chim.unifi.it/orac`, spacchettarla e compilarla sul proprio PC seguendo le istruzioni.

4.2 test

Eseguire i test fondamentali (directory `tests/basic_tests`)

Analisi del file di input

4.3 il programma ORAC

presentazione del programma e dei file di input e output: `http://www.chim.unifi.it/u/signo/did/biomol/orac.pdf`

5 Minimizzazione energia con ORAC

5.1 preparazione ambiente di lavoro

1. Creare una directory di lavoro (ad esempio `~/orac`) con sottodirectory `lib` e `pdb` (Si può copiare tutto il contenuto di `/home/signorini/biomol/orac`)

5.2 preparazione input

5.2.1 input principale

Come già visto, definisce tre file di dati

1. coordinate (PDB)
2. topologia
3. parametri di potenziale

Questi file devono usare le stesse convenzioni sui nomi dei

- residui (es. `ile`, `gly-h`)
- atomi (es. `ca`, `hg21`)
- tipi atomici (es. `hc`)

Nell'istruzione `JOIN SOLUTE` va data la sequenza dei residui, che si può ricavare dal file PDB.

Ricordare che i due residui terminali sono definiti a parte nel file delle topologie.

5.2.2 struttura

La struttura creata con VMD usa in certi casi dei nomi di atomi diversi da quelli di AMBER. Questi vanno "tradotti" nei corrispondenti nomi di AMBER; lo si fa confrontando il file PDB e il file di topologia (nei casi dubbi, usando VMD per identificare gli atomi). Es:

`HN-> H`

5.2.3 topologia

Notare che le due glicine terminali sono, rispettivamente, N-terminale e C-terminale, ovvero la sequenza, in termini di residui definiti nella topologia AMBER, è

`gly-h ... gly-o`

5.2.4 parametri di potenziale

5.3 minimizzazione energia

Si può eseguire la minimizzazione con due metodi: Steepest Descent e Conjugate Gradient

5.3.1 preparazione input ed esecuzione

1. Portarsi sulla directory di lavoro
2. Creare un input (es `minim.in`) che esegua una minimizzazione dell'energia della molecola:

```
&SETUP
    CRYSTAL 200.0 200.0 200.0
    READ_PDB ../pdb/D4PK7G_all.pdb
&END
&PARAMETERS
    READ_TPG_ASCII ../lib/amber03.tpg
    READ_PRM_ASCII ../lib/amber03.prm
    JOIN SOLUTE
       gly-h tyr asp pro ala thr gly thr phe gly-o
    END
&END
&POTENTIAL
    EWALD OFF
    CUTOFF 100.
    STRETCHING
&END
&SIMULATION
    MINIMIZE
       CG 0.00001
       # oppure   SD 0.00001
    END
&END
&INOUT
    ASCII 100.0 OPEN minim.pdb
&END
&RUN
    TIME 3000.0
    PRINT 100.0
    PROPERTY 100.0
&END
```

3. lanciare il programma

5.3.2 analisi risultati

1. Visualizzare i risultati con vmd
2. Misurare con vmd:
   (a) RMSD
   (b) angoli di Ramachandran
   (c) distanza *"head-to-tail"*
3. Calcolare gli RMSD con `rmsd.sh`.

(non fatto)
Ad es:

```
rmsd.sh -r pdb/D4PK7G-backb.pdb pdb/D4PK7G_min.pdb min.pdb > bb.rmsd
plot -0,4 bb.rmsd
```

(a) per confrontare solo il backbone occorre prima costruire un file con i soli atomi di backbone; basta usare comandi tipo

```
awk '$3==''CA''' >> tmp
```

e

```
sort -k3 -nk6 tmp > bb.pdb
```

4. calcolare con strumento `analysis` di ORAC:

   (non fatto)

   (a) *angoli di Ramachandran*

   (b) *distanza "head-to-tail"*

5. Analizzare i diversi contributi all'energia.

   Ad es:

   ```
   orac-post-out P="TotPot Bonded NonBond" min.out > ene
   plot -1,2-4 ene
   ```

6. Osservare

   (non fatto)

   (a) *se vi è correlazione tra variazione nei RMSD e nell'energia*

   (b) *quale modifica conformazionale determina la massima variazione di energia*

   (c) *quale parte del potenziale guida la transizione verso il minimo.*
   N.B. `NonBond=Coulomb+Ener14+`LennardJones (non stampata!)

   (d) *nel caso che ci sia una struttura sperimentale, fare il confronto con quella.*