

Progetto di simulazione molecolare per il corso di “Complementi di algoritmi” A.A. 2005-06

13 febbraio 2006

1 Descrizione

Il progetto si compone delle seguenti fasi:

1. caricamento di soluzioni in formato RST
2. creazione e caricamento di una struttura per rappresentare la *box* di simulazione
3. implementazione delle *periodic boundary conditions* con una struttura toroidale
4. implementazione delle *periodic boundary conditions* con un *guscio* che circonda la *box*
5. determinazione dei cluster prodotti dai soli legami fra soluto e soluto
6. determinazione dei cluster prodotti dal soluto, eventualmente con un singolo strato di acqua intermedio

2 Caricamento di soluzioni in formato RST

Dovete costruire un eseguibile che riceve in ingresso come parametri due file

1. un file *statico* (si chiamerà *parm.top*, ma deve essere un parametro), che contiene il tipo di molecole e atomi, l'aggregazione degli atomi in molecole
2. un file *dinamico* (avrà estensione *.rst*, ma deve essere un parametro), che contiene le posizioni degli atomi in un dato istante

Si devono aprire i due file e caricare nelle strutture dati predefinite tutti i dati necessari. Quindi, si lanci la ricerca dei legami e il calcolo dei cluster e si salvi il tutto in formato PDB con le funzioni predefinite (si prenda a modello il file *hb-cluster.c*).

Per verificare Si esegua il tutto sui cinque file *ur4-n.rst* ($n = 1, \dots, 5$) e si confronti il risultato con i cinque file *ur4-n.pdb* ($n = 1, \dots, 5$): i file dovrebbero essere identici, salvo l'ultima cifra decimale significativa di qualche coordinata spaziale.

3 Caricamento di una struttura per la *box*

La struttura deve contenere le lunghezze dei tre lati della *box* e i tre angoli (che non verranno mai usati).

Si scriva una funzione che apre un file (il nome deve essere un parametro, anche se faremo la prova su *parm.top*) e legge i dati richiesti.

Per verificare Si stampino i valori per i soliti cinque file *.rst*.

4 *Periodic boundary conditions a toro*

Si creino versioni modificate (senza cancellare gli originali) delle funzioni che valutano

- la distanza fra due atomi (*Distance()*)
- l'angolo formato da tre atomi (*Angle()*)
- l'appartenenza di un atomo alla shell del soluto (*BelongsToShell()*)

Per verificare Confrontate i risultati con i seguenti, spediti da quattro vostri colleghi.

Configurazione con $\delta_1 = 3.5$ e $\delta_2 = 7.0$ I risultati sono parzialmente discordi (tre a uno). Per le righe che riportano il numero di cluster e di residui in seconda shell ho solo un risultato (che fa parte del gruppo di tre concordi sulla prima riga).

	ur4-1	ur4-2	ur4-3	ur4-4	ur4-5
Bond	91	79	87	93-92	95-94
Cluster	415	426	423	419	416
Residui 2	190	194	186	205	190

Molto verosimilmente, il primo risultato è quello vero, dato che solo nelle tre istanze in cui differiscono si trovano legami (esattamente uno) a cavallo della *box*. Questi legami sono:

- per *ur4-1*: $D = 966, H = 967, A = 1065$
- per *ur4-4*: $D = 1155, H = 1156, A = 411$
- per *ur4-5*: $D = 1155, H = 1156, A = 411$

In rete trovate la lista completa dei legami.

Configurazione con $\delta_1 = 1000$ e $\delta_2 = 1000$ I risultati sono parzialmente discordi (tre a uno) sulla prima riga. Per la seconda, ho solo un risultato (che fa parte del gruppo dei tre concordi sulla prima).

	ur4-1	ur4-2	ur4-3	ur4-4	ur4-5
Bond	666-602	637-580	654-598	663-601	653-593
Cluster	9	18	9	10	12

Ovviamente, tutti i 504 residui sono in seconda shell. In rete trovate la lista completa dei legami.

5 *Periodic boundary conditions a guscio*

Si crei una procedura che arricchisce la soluzione appena caricata di altri residui, che sono l'immagine di quelli originali. In linea di principio, ogni atomo ha 26 immagini, dato che può essere replicato aumentando, diminuendo o mantenendo invariate le sue tre coordinate spaziali sia una per una sia a coppie sia tutte e tre. Ne derivano 27 variazioni, delle quali una consiste nel lasciare l'atomo dov'è. Di queste 26 immagini, però la maggior parte non vanno costruite perché cadono all'esterno del guscio.

Per chiarezza, quando aggiungete le immagini, date loro lo stesso indice dei residui e degli atomi di partenza. Tuttavia, aggiungete alla definizione di residuo una terna di numeri che ci dirà se il residuo è originale $(0, 0, 0)$ o è un'immagine traslata in avanti rispetto all'asse x $(1, 0, 0)$ o indietro $(-1, 0, 0)$, ecc...

A rigore, dovrebbero interessare solo i residui che cadono almeno in parte a distanza $\leq \delta$ (soglia per la formazione dei legami a idrogeno) dalla *box*. Un residuo va duplicato interamente ogni volta che l'immagine di un suo atomo cade nel guscio. Un guscio troppo sottile vi farà perdere legami, un guscio troppo spesso dovrebbe solo rallentare l'algoritmo, ma non cambiare l'insieme dei legami.

Creato il guscio, si scriva una versione modificata della procedura di costruzione dei legami, che crei i legami solo fra residui originali oppure fra residui originali e residui immagine (in entrambi i versi), ma non fra residui immagine.

Per verificare Confrontate i risultati con i seguenti, spediti da quattro vostri colleghi.

Configurazione con $\delta_1 = 3.5$ e $\delta_2 = 7.0$ I risultati sono parzialmente discordi (tre a uno) sulla prima riga. Per le altre due, ho solo due risultati, ottenuti dal gruppo dei tre che sono concordi sulla prima.

	ur4-1	ur4-2	ur4-3	ur4-4	ur4-5
Bond	91	79	87	93-92	95-94
Cluster	416	426	423	420	417
Residui 2	190-192	192-198	186-188	201-212	187-197

Molto verosimilmente, per la prima riga il primo risultato è quello vero, dato che solo nelle tre istanze in cui differiscono si trovano legami (esattamente uno) a cavallo della *box*. Questi legami sono:

- per *ur4-1*: $D = 966$, $H = 967$, $A = 1065$ (x incrementata)
- per *ur4-4*: $D = 1155$ (x incrementata), $H = 1156$ (x incrementata), $A = 411$
- per *ur4-5*: $D = 1155$ (x incrementata), $H = 1156$ (x incrementata), $A = 411$

In rete trovate la lista completa dei legami.

Per quanto riguarda i residui in seconda shell, invece, credo che la differenza derivi dal fatto di includere nel guscio (e quindi di considerare potenzialmente in seconda shell) solo i residui legati in qualche modo a residui della *box* oppure tutti i residui non più lontani di una data soglia dalla *box* stessa. La prima definizione è ovviamente più stretta. Se la differenza deriva veramente da questa causa, non è rilevante.

Configurazione con $\delta_1 = 1000$ e $\delta_2 = 1000$ Qui i risultati sono concordi. Per la seconda e terza riga, ho solo un risultato (che fa parte del gruppo dei tre concordi sulla prima).

	ur4-1	ur4-2	ur4-3	ur4-4	ur4-5
Bond	791	749	758	774	759
Cluster	26	35	26	19	21
Residui 2	711	692	673	689	679

In rete trovate la lista completa dei legami.

6 Cluster con legami soluto-soluto

Si scriva una versione modificata della procedura che costruisce i cluster, in modo che ignori i legami fra soluto e solvente o fra solvente e solvente e che costruisca solo i cluster formati da soluti. Si salvino i cluster attraverso la funzione che salva in formato PDB. Ho qualche dubbio sulla correttezza di questa funzione, nel senso che potrebbe salvare tutti i cluster, e non solo quelli formati dal soluto, oppure fare cose strane.

Per verificare Questi sono i primi risultati che mi sono arrivati. Si riferiscono ai file *ur4-n.rst* ($n = 1, \dots, 5$), che sono quasi banali (4 molecole di urea, quasi sempre divise tra loro) e ai file *ur16*, *ur27*, *ur64* e *ur125*, che trovate in fondo alla pagina del progetto, e che contengono più molecole di urea. Per i file con 4 residui di urea, i risultati derivano da due fonti concordi; per gli altri, ho una sola fonte.

	ur4-1	ur4-2	ur4-3	ur4-4	ur4-5
Cluster	4	4	3	3	3
	ur16.1	ur16.2	ur16.3	ur16.4	ur16.5
Cluster	15	13	12	12	13
	ur27.1	ur27.2	ur27.3	ur27.4	ur27.5
Cluster	22	23	23	23	21
	ur64.1	ur64.2	ur64.3	ur64.4	ur64.5
Cluster	31	37	36	43	41
	ur125.1	ur125.2	ur125.3	ur125.4	ur125.5
Cluster	53	47	52	62	50

I risultati non dovrebbero dipendere dall'ampiezza della shell, dato che il soluto per definizione sta sempre in prima shell, qualunque sia l'ampiezza. In effetti, i risultati che mi sono arrivati sono identici.

7 Cluster di soluti con pellicole monostrato di acqua

Si vuole considerare come appartenenti allo stesso cluster anche soluti collegati non direttamente, ma indirettamente da una molecola di acqua. Dopo consultazione con il Prof. Fornili, siamo arrivati a una definizione sensata di cluster con

pellicola monostrato. In pratica, l'interesse è puntato sui soluti, per cui si tratta di visitare il grafo come sopra, visitando anche le molecole di acqua, purché subito dopo si torni subito in una molecola di soluto. Al termine della visita, si stampino solo le molecole di soluto appartenenti a ciascun cluster.

Per verificare Questi sono i primi risultati che mi sono arrivati. Si riferiscono ai file *ur4-n.rst* ($n = 1, \dots, 5$), che sono quasi banali (4 molecole di urea, quasi sempre divise tra loro) e ai file *ur16*, *ur27*, *ur64* e *ur125*, che trovate in fondo alla pagina del progetto, e che contengono più molecole di urea. Per i file con 4 residui di urea, i risultati derivano da due fonti, parzialmente discordi; per gli altri, ho una sola fonte.

	ur4-1	ur4-2	ur4-3	ur4-4	ur4-5
Cluster	4	4-3	3	3	3-2
	ur16.1	ur16.2	ur16.3	ur16.4	ur16.5
Cluster	14	13	10	12	12
	ur27.1	ur27.2	ur27.3	ur27.4	ur27.5
Cluster	22	23	22	22	18
	ur64.1	ur64.2	ur64.3	ur64.4	ur64.5
Cluster	26	34	30	38	37
	ur125.1	ur125.2	ur125.3	ur125.4	ur125.5
Cluster	42	34	40	50	39

I risultati non dovrebbero dipendere dall'ampiezza della shell, a meno che si assegni alla seconda shell un valore inferiore alla distanza di legame, dato che i cluster che cerchiamo sono formati da molecole di soluto (sempre in prima shell) e molecole di acqua immediatamente legate al soluto (e quindi a distanza di legame). In effetti, i risultati che mi sono arrivati dalla prima fonte sono identici per i due modelli (toro e guscio).