

Coarse-Graining de Melitina e POPC

Hubert Stassen

Instituto de Química - UFRGS, gullit@iq.ufrgs.br

04 de agosto de 2014

Resumo: Essa parte do minicurso demonstra a associação de uma molécula melitina com 128 moléculas do fosfolípido palmitoil-oleil-fostadidil-colina (POPC) em solução aquosa utilizando coarse-graining com o campo de força MARTINI.

1 Ponto de Partida

Nessa experiência, transformamos a estrutura cristalina do polipeptídeo melitina (2MLT.pdb [1]) em uma topologia para coarse-graining utilizando o campo de força MARTINI e suas ferramentas [2]. Posteriormente, associa-se a melitina com moléculas POPC aleatoriamente distribuídas em solução aquosa com o objetivo de obter uma bicamada de POPC interagindo com a melitina.

Colocou-se os arquivos necessários para iniciar esse minicurso no diretório `mini_03/start`. Entramos no diretório de trabalho.

`cd mini_03`

Acha-se os arquivos necessários dentro do diretório `start`. Listando com

`ls -l start`

mostra o conteúdo:

```
-rw----- 1 drg users 46170 2014-08-13 16:00 2MLT.pdb
-rw----- 1 drg users 50536 2014-08-13 15:58 martini.itp
-rwx----- 1 drg users 229323 2014-08-13 15:58 martinize.py
-rw-r--r-- 1 drg users 3933 2014-08-13 16:04 md.mdp
-rw----- 1 drg users 3929 2014-08-13 16:04 min.mdp
-rw----- 1 drg users 936 2014-08-13 16:03 popc.gro
-rw-r--r-- 1 drg users 1117 2014-08-13 16:03 popc.itp
-rw-r--r-- 1 drg users 109 2014-08-13 16:04 water.gro
```

O arquivo `2MLT.pdb` contém a estrutura cristalina da melitina. Observa-se que esse arquivo já foi editado na segunda parte desse minicurso eliminando a segunda molécula do polipeptídeo. Os arquivos `popc.gro` e `water.gro` possuem as coordenadas para o coarse-graining de uma molécula POPC e água, respectivamente. O campo de força MARTINI (parâmetros em `martini.itp`) para o POPC é definido em `popc.itp`. O script `martinize.py` é uma ferramenta do MARTINI para obter topologia e coordenadas coarse-graining de uma proteína. Chamamos esses arquivos para a nossa área de trabalho.

```
cp start/* .
```

Pode-se verificar o conteúdo de cada arquivo através de

```
more arquivo
```

2 Topologia MARTINI da Melitina

O script `martinize.py` produz a topologia coarse-graining para uma proteína dentro da filosofia do campo de força MARTINI. Como ponto de partida desse script, necessita-se as coordenadas atomísticas (aqui: `2MLT.pdb`) e a estrutura secundária da melitina. Aproveita-se aqui das informações experimentais relatadas no arquivo `2MLT.pdb`: os primeiros 11, bem como os últimos 13 resíduos definem α -hélices (H) e os resíduos 12 e 13 não são estruturados (assume-se coil C). Precisa-se editar um arquivo de duas linhas, com a primeira linha contendo o número de resíduos (26) e a segunda linha a estrutura secundária de cada resíduo. Gera-se esse arquivo novo.

```
vi ssd ou outroeditor ssd
```

Cria-se as duas linhas.

- 26
- HHHHHHHHHHHHCCHHHHHHHHHHHHHH

Salvando e saindo desse arquivo, aplicamos o script. Com opção `-h`,

```
./martinize.py -h
```

obtemos as opções. Escolhemos o campo de força mais recente (MARTINI22) com o uso de diedrais. Cabe mencionar que a melitina é um polipeptídeo muito pequeno. Para proteínas maiores, deve-se considerar eventualmente o uso da rede de ligações elásticas com o objetivo de manter a sua estrutura melhor em simulações coarse-graining.

```
./martinize.py -f 2MLT.pdb -o melitina.top -x melitina.pdb -v  
-ss ssd -ff martini22 -ed
```

Nesse comando, utilizou-se a estrutura experimental da melitina para criar um arquivo de coordenadas coarse-grained (`melitina.pdb`), topologia da melitina (`Protein_A.itp`), bem como um arquivo descrevendo a composição do sistema (`melitina.top`). Visualização desses arquivos é possível via

more arquivo

e permite a análise do conteúdo da topologia `Protein_A.itp` e das coordenadas `melitina.pdb` para o coarse-graining da melitina. Centraliza-se a melitina numa caixa cúbica com dimensões que correspondem mais ou menos a um sistema contendo uma bicamada hidratada com 128 fosfolípidos. A ferramenta `editconf` assume essa tarefa.

```
editconf -f melitina.pdb -c -bt triclinic -o melitina.gro -box 9.5  
9.5 8.5
```

Visualiza-se.

```
grompp -v -o min.tpr -c melitina.gro -f min.mdp -p melitina.top
```

```
ngmx -s min.tpr -f melitina.gro
```

Observa-se no output do `grompp` que a carga da melitina apresenta +5. Cuidamos da eletroneutralidade posteriormente. Relaxamos um pouco a estrutura gerada (500 passos steepest descent).

```
mdrun -nice 0 -v -s min.tpr -c min.gro -e min.edr -o min.trr  
-g min.log
```

3 Preparação da Solução Associativa

A estratégia para gerar um sistema autoassociativo contendo moléculas POPC, melitina, e água consiste na montagem de uma mistura aleatória. Insere-se inicialmente 128 moléculas POPC (coordenadas coarse-graining de uma molécula em `popc.gro`) utilizando a ferramenta `genbox` aplicada ao arquivo `min.gro` com a estrutura relaxada da melitina. Define-se um raio de van der Waals um pouco maior do que o default (0,1 nm), pois partículas coarse-grained possuem raios maiores que átomos comuns.

```
genbox -cp min.gro -ci popc.gro -nmol 128 -o 00.gro -vdwd  
0.15
```

Edita-se a topologia,

```
vi melitina.top ou outroeditor melitina.top
```

inserindo a linha `#include "popc.itp"` após `#include "Protein_A.itp"`. Também precisa-se incluir no fim do arquivo a composição do sistema incluindo a linha `POPC 128`. Salvando e saindo do arquivo, relaxamos a configuração com o `steepest descent`.

```
grompp -v -o min.tpr -c 00.gro -f min.mdp -p melitina.top  
mdrun -nice 0 -v -s min.tpr -c min.gro -e min.edr -o min.trr  
-g min.log
```

Insere-se ainda 1500 moléculas de água seguindo o procedimento apresentado,

```
genbox -cp min.gro -ci water.gro -nmol 1500 -o 01.gro -vdwd  
0.15
```

e edita-se a topologia,

```
vi melitina.top ou outroeditor melitina.top
```

incluindo como última linha `W 1500`. A topologia para a água `W` já existe em `martini.itp`.

Salvando e saindo desse arquivo, tentamos a minimização de energia desse sistema. Compila-se 500 passos `steepest-descent`.

```
grompp -v -o min.tpr -c 01.gro -f min.mdp -p melitina.top  
mdrun -nice 0 -v -s min.tpr -c min.gro -e min.edr -o min.trr  
-g min.log
```

Visualizamos.

```
ngmx -s min.tpr -f min.gro
```

Finalmente, iniciamos a simulação (definiu-se 10 ns).

```
grompp -v -o 01.tpr -c min.gro -f md.mdp -p melitina.top  
mdrun -nice 0 -v -s 01.tpr -c 01.gro -e 01.edr -g 01.log -o 01.trr  
&
```

Fiscaliza-se os logs da simulação.

```
tail -f 01.log
```

(abortar com `Ctrl-C`).

4 Simulação do Sistema

Visualizando a trajetória obtida,

```
ngmx -s 01.tpr -f 01.trr
```

observa-se a tendência de agregação entre moléculas POPC. Porém, ainda não temos a bicamada completa. Continuamos a simulação por mais 10 ns editando o arquivo de parâmetro **md.mdp**.

```
vi md.mdp ou outroeditor md.mdp
```

Aplica-se as modificações:

- **tinit = 10000** (já simulamos por 10 ns)
- **gen_vel = no** (temos velocidades em **01.gro**)

Salvando e saindo do arquivo, tomamos ainda conta da neutralização de carga. Para isso, abrimos o arquivo gerado de coordenadas

```
vi 01.gro ou outroeditor 01.gro
```

e substituímos os últimos cinco resíduos **W** na coluna esquerda do arquivo por **ION**. Também substituímos nessas 5 linhas o outro **W** por **CL-**. Não distorçam as colunas em aplicar essas modificações. Salvando e saindo desse arquivo, corrigimos também o arquivo contendo a composição do sistema,

```
vi melitina.top ou outroedito melitina.top
```

adicionando uma nova linha (no fim do arquivo) contendo **CL-** 5. O número de moléculas água **W** torna-se 1495 em vez de 1500. Salva-se saindo desse arquivo, continua-se a simulação.

```
grompp -v -o 02.tpr -c 01.gro -f md.mdp -p melitina.top
```

```
mdrun -nice 0 -v -s 02.tpr -c 02.gro -e 02.edr -g 02.log -o 02.trr  
&
```

Fiscalizando os logs,

```
tail -f 02.log
```

Depois, visualizamos a trajetória,

```
ngmx -s 02.tpr -f 02.trr
```

Agora conseguimos a bicamada.

Em uma ou outra tentativa de agregação de fosfolipídeos pode acontecer que a formação da bicamada necessita mais tempo. Nesse caso, devemos continuar a simulação. Também pode acontecer que a bicamada não se orienta no plano xy . Nesse caso, precisa-se girar o sistema forçando a bicamada para esse plano. Sugere-se o seguinte procedimento: i) `tail 02.gro` para obter as dimensões da caixa (aqui: 6.46180 6.46180 8.22535), ii) gerar uma caixa cúbica com as dimensões do lado maior (aqui: 8.22535) girando pelo eixo desejado. O `editconf` faz isso: `editconf -f 02.gro -o 03.gro -c -bt cubic -box 8.22535 8.22535 8.22535 -rotate 0 90 0` (girando pelo eixo y) por exemplo. Possivelmente, a rotação por 90° envolve outro eixo. O `ngmx` mostra o resultado: `ngmx -s 02.tpr -f 03.gro`. Experimentem quando necessário.

Na nossa experiência, estamos com sorte. Não se preocupem com a bicamada não ficando no centro da caixa. Temos condições do contorno periódico e convenção da imagem mínima na simulação.

Simulamos o sistema zerando o tempo de simulação,

```
vi md.mdp ou outroedito md.mdp
```

colocando

- `tinit = 0`

Salvando e saindo desse arquivo, simula-se,

```
grompp -v -o 03.tpr -c 02.gro -f md.mdp -p melitina.top
```

```
mdrun -nice 0 -v -s 03.tpr -c 03.gro -e 03.edr -g 03.log -o 03.trr  
&
```

fiscalizando as entradas nos logs.

```
tail -f 03.log
```

No final dessa simulação, pode-se visualizar a trajetória.

```
ngmx -s 03.tpr -f 03.trr
```

Com o objetivo de fiscalizar a equilibração do sistema, determina-se a evolução temporal de propriedades termodinâmicas, energias de interação, entre outras. A ferramenta `g_energy` desenvolve esse cálculo.

```
g_energy -s 03.tpr -f 03.edr -o 03.termo
```

Sugere-se as opções 1-5, 7-11, 13, 15-19, 47-50, 52, 55, 56, e 58. Tendo escolhidas essas opções, digita-se 0 para iniciar o cálculo. Fiscaliza-se com o `gnuplot`.

gnuplot

```
p "03.termo.xvg" u 1:n w l
```

Como ilustrado nos tutoriais dos outros módulos, `n` representa o número da coluna a ser plotado contra o tempo. As colunas estão descritas no cabeçalho do arquivo `03.termo.xvg`.

Mesmo não tendo conseguido equilibrar perfeitamente esse sistema, podemos aplicar umas análises para a trajetória a partir de 6 ns. Calcula-se a área por fosfolipídeo. A membrana fica no plano xy . Assim, multiplica-se a extensão da caixa em x e y (ambos os valores em torno de 6,43 nm) e divide-se o produto por 64 (número de moléculas POPC por camada) produzindo $64,6 \text{ \AA}^2$.

Calculamos o perfil de densidade do sistema a partir de 6 ns.

```
g_density -s 03.tpr -f 03.trr -b 6000. -o total (opção System)
```

```
g_density -s 03.tpr -f 03.trr -b 6000. -o melit (opção Protein)
```

```
g_density -s 03.tpr -f 03.trr -b 6000. -o popc (opção POPC)
```

```
g_density -s 03.tpr -f 03.trr -b 6000. -o agua (opção W)
```

Visualizando,

gnuplot

```
p "total.xvg" u 1:2 w l, "melit.xvg" u 1:2 w l, "popc.xvg" u  
1:2 w l, "agua.xvg" u 1:2 w l
```

indica uma espessura em torno de 3,4 nm dessa membrana (admito: não temos máximas bem definidas) com a melitina localizada dentro da membrana (transmembrana).

Precisa-se prolongar essa simulação.

Bibliografia

- [1] T.C. Terwilliger e D. Eisenberg, *J. Biol. Chem.* **257** (1982), 6010.
- [2] Sugere-se uma consulta profunda no site md.chem.rug.nl/cgmartini e um estudo rigoroso das publicações mencionadas.