

Apêndice

Ferramentas para modelagem molecular e parâmetros de simulação

Porcupine

Diego Enry Barreto Gomes¹; Roberto D. Lins^{2,3}; Thereza A. Soares^{2,3}; Pedro G. Pascutti¹

(1) Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ – Brasil. (2) Universidade Federal de Pernambuco, Recife, PE – Brasil (3) Pacific Northwest National Laboratory, Richland, WA – EUA. Distribuído via requisição por e-mail: diego@biof.ufrj.br

Descrição

O programa **porcupine** foi desenvolvido para auxiliar a visualização dos componentes principais de uma simulação de dinâmica molecular (dinâmica essencial) ao projetar no espaço, os autovetores de movimento de acordo com a amplitude calculada por PCA. O programa lê um arquivo no formato PDB contendo os pontos extremos do modo de vibração calculado, opcionalmente filtra vetores de baixa amplitude, e escreve um arquivo de saída no formato VMD contendo setas com direção e tamanho proporcionais à direção e amplitude do modo de vibração.

Exemplo de uso.

1) Realizando o cálculo de dinâmica essencial com o GROMACS

```
g_covar -f trajetória.xtc -s topologia.tpr -o autovalores.svg -v autovetores.trr
```

2) Projeção dos pontos extremos do movimento em uma trajetória no formato PDB, usando o GROMACS

```
g_anaeig -v eigenvec.trr -f trajetória.xtc -first 1 -last 1 -extr extremos.pdb -nframes 2
```

3) Geração das setas para visualização dos autovetores usando o programa **porcupine**

```
porcupine -i extremos.pdb -o setas.vmd -f 1.0 -c
```

4) Visualização no programa VMD

```
vmd -e setas.vmd
```

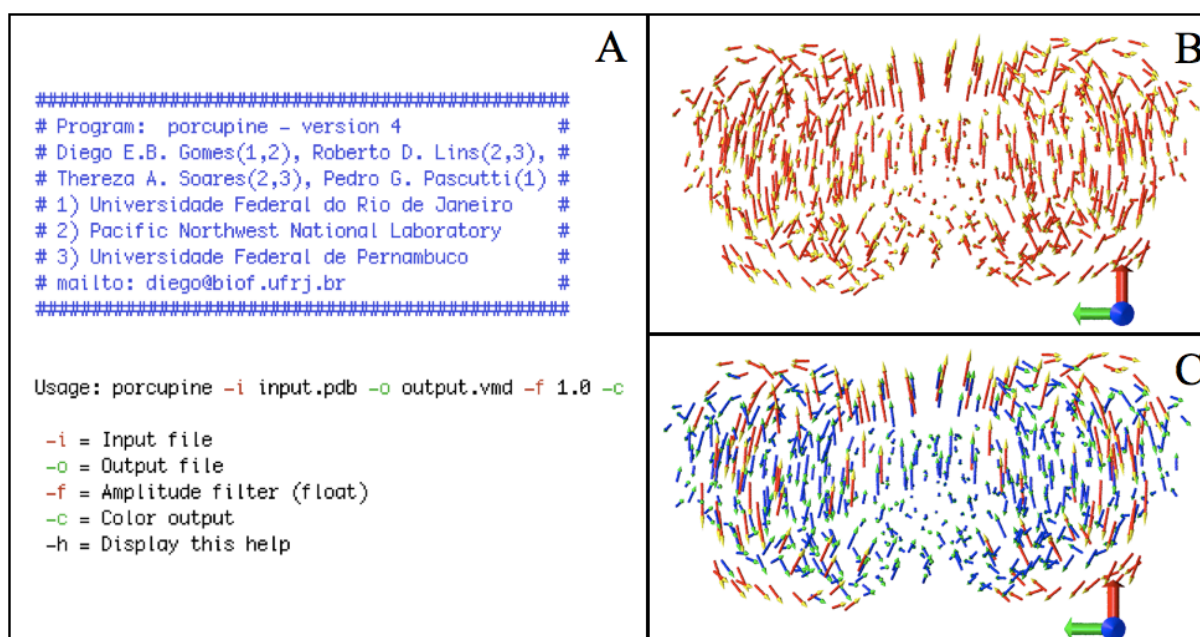


Ilustração de uso do programa **porcupine**. (A) Interface de comando. (B) Exemplo de saída no renderizada no programa VMD, as setas indicam a direção e amplitude do autovetor. (C) O uso avançado do programa permite distinguir vetores de diferentes amplitudes em cores diferentes, ex em vermelho os vetores com grande amplitude e em azul os vetores de menor amplitude.

Trj2pdb

Diego Enry Barreto Gomes¹; Thereza A. Soares^{2,3}; Pedro G. Pascutti¹; Tjerk P. Straatma³

(1) Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ – Brasil. (2) Universidade Federal de Pernambuco, Recife, PE – Brasil (3) Pacific Northwest National Laboratory, Richland, WA – EUA. Distribuído via requisição por e-mail: diego@biof.ufrj.br

Descrição

O **trj2pdb** é um programa desenvolvido para converter trajetórias de simulação de dinâmica molecular no formato TRJ, adotado pelo programa NWChem, para trajetórias no formato PDB, universalmente aceito pelos programas de análise de simulação. Além da função de conversão, o **trj2pdb** permite ajustar a frequência de escrita da trajetória e remover moléculas do solvente.

Exemplo de uso

```
#####  
# Program: NWChem trj2pdb - version 15.2      #  
# Diego E.B. Gomes(1,2), Pedro G. Pascutti(1), #  
# Thereza A. Soares(2,3), Tjerk P. Straatsma(2) #  
# 1) Universidade Federal do Rio de Janeiro    #  
# 2) Pacific Northwest National Laboratory      #  
# 3) Universidade Federal de Pernambuco        #  
# mailto: diego@biof.ufrj.br                  #  
#####
```

```
Usage: trj2pdb -i input.trj -o output.pdb
```

```
-i = Input file  
-o = Output file  
-b = First frame to read  
-skip = Read every 'skip' frames  
-h = Display this help
```

Tela de execução do programa **trj2pdb** apresentando a instrução de uso.

Nanopore

Diego Enry Barreto Gomes¹; Thereza A. Soares^{2,3}; Pedro G. Pascutti¹; Tjerk P. Straatma³

(1) Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ – Brasil. (2) Universidade Federal de Pernambuco, Recife, PE – Brasil (3) Pacific Northwest National Laboratory, Richland, WA – EUA. Distribuído via requisição por e-mail: diego@biof.ufrj.br

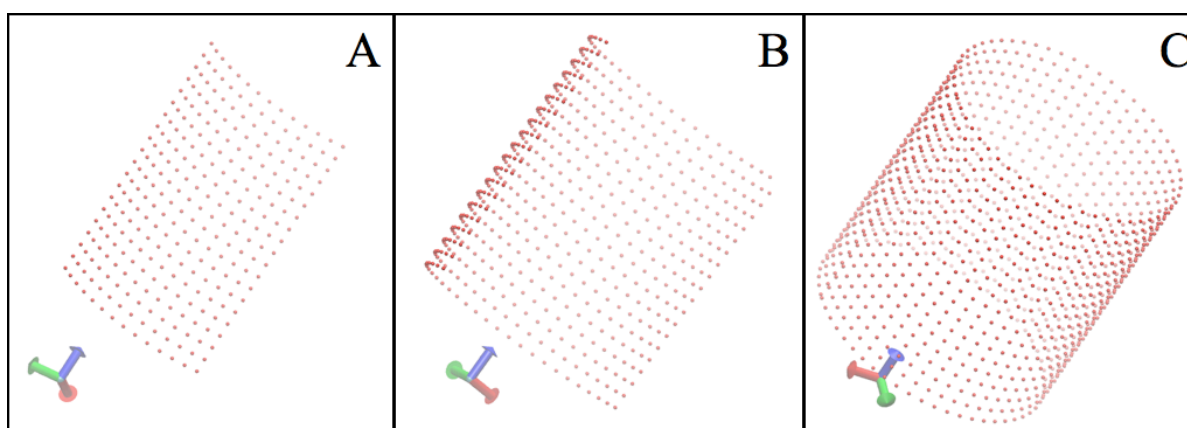
Descrição

O programa **nanopore** escreve um arranjo de átomos no formato PDB, orientado no eixo Z, em diferentes condições e formas. O programa pode desenhar cubos ou cilindros de dimensões e espaço entre os pontos definidos pelo usuário. Além disso podem ser produzidas frações de $\frac{1}{2}$ e $\frac{1}{4}$ de cilindro. Os arranjos produzidos podem ser usados em simulações de mecânica molecular ao incorporar atributos atômicos aos pontos (GOMES, 2008; GOMES, 2010).

Exemplo de uso

<pre>#Execute nanopore % ./nanopore #Selecionar entre as formas disponíveis Select the shape “ 1 - cubic box “ 2 - generic XYZ box “ 3 - cylinder” 4 - tube + half tube + quarter tube “ % 4</pre>	<pre>#Indicar espaçamento entre dois pontos. #Esta distância pode determinar o raio do arranjo circular. Please enter the distance between groups (Angstrom) % 10 # Indicar a extensão da rede de pontos. Se “cubic box” for # selecionada, apenas uma dimensão é #requerida. Se “generic # XYZ” é selecionado, as três dimensões serão necessárias. Neste # exemplo X e Y são escalonados para o raio selecionado e suas # dimensões definidas pela distancia entre os #pontos. Enter the Z dimension in Angstroms (integer value) % 200</pre>
---	--

Demonstração da execução do **nanopore**.



Arranjos produzidos pelo programa **nanopore** com espaçamento entre os pontos de 10Å. (A) $\frac{1}{4}$ de cilindro; (B) $\frac{1}{2}$ cilindro; (C) cilindro.

Hbmap2grace

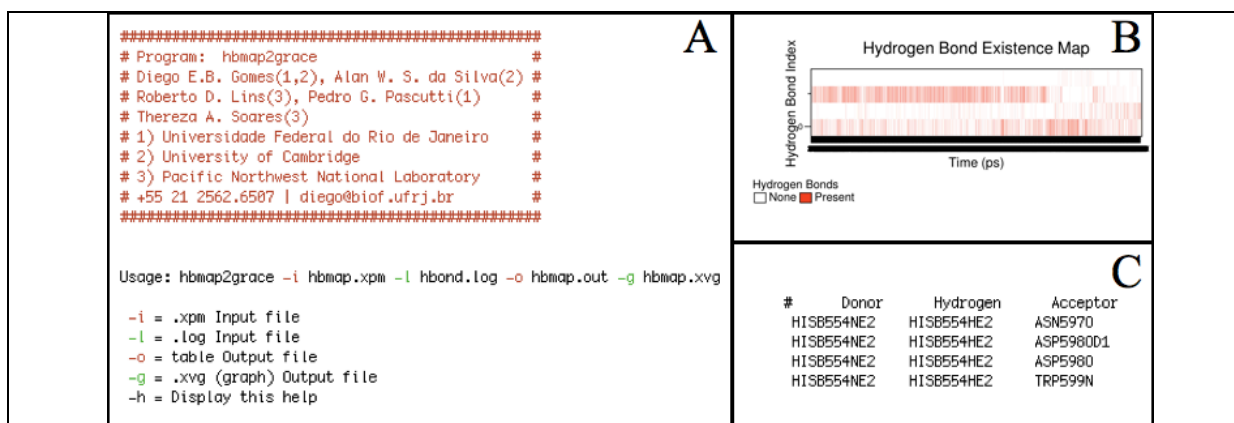
Diego Enry B. Gomes¹, Alan Wilter da Silva², Roberto D. Lins³, Pedro G. Pascutti¹, Thereza A. Soares³

(1) Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ – Brasil. (2) EMBL - EBI, Wellcome Trust Genome Campus, Cambridge - Inglaterra. (3) Pacific Northwest National Laboratory, Richland, WA – EUA. Distribuído via requisição por e-mail: diego@biof.ufrj.br

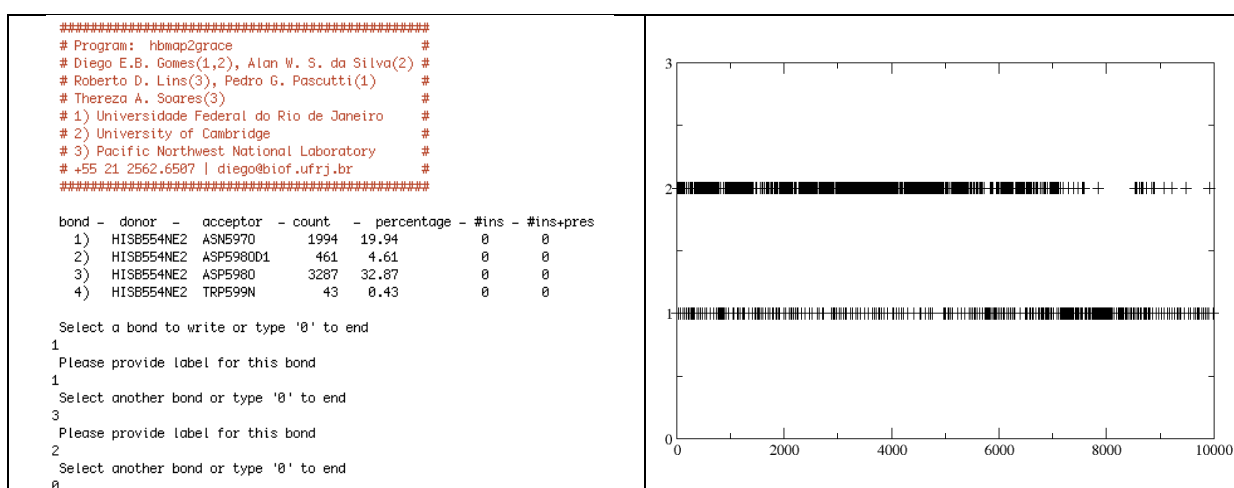
Descrição

O programa **hbmap2grace** calcula a frequência das ligações hidrogênio dos arquivo **hbmap.xpm** e relaciona com a tabela de pares contida no arquivo **hbond.log**, ambos gerados pelo programa **g_hbond** do GROMACS. O programa também permite selecionar conjuntos de ligações interessantes, agregar ligações, uma mesma marcação (*label*) para um aceitador com todos doadores, e reapresentar em escala temporal as ligações selecionadas.

Exemplo de uso



Tela de ajuda do programa **hbmap2grace** (A) e arquivos de teste, distribuídos com o programa. (B) Mapa de ligações hidrogênio gerado a partir do arquivo **hbmap.xpm**, convertida em imagem pelo programa **xpm2ps** (GROMACS). C) Lista de ligações hidrogênio derivada de um arquivo **hbond.log**.



Resultado do programa **hbmap2grace**. À esquerda o programa em execução usando os arquivos de entrada mostrados na figura anterior. Logo que o programa inicia já são apresentadas as frequências das ligações hidrogênio, indexadas para seleção pelo usuário. As ligações podem ser agregadas numa mesma representação se for usada a mesma marcação (*label*). À direita o produto do programa apresentado como gráfico no programa **xmgrace**, os símbolos “+” representam a ocorrência de ligação na escala de tempo do eixo horizontal. No eixo vertical estão os números indicando as ligações selecionadas no exemplo a esquerda.

SurfinMD

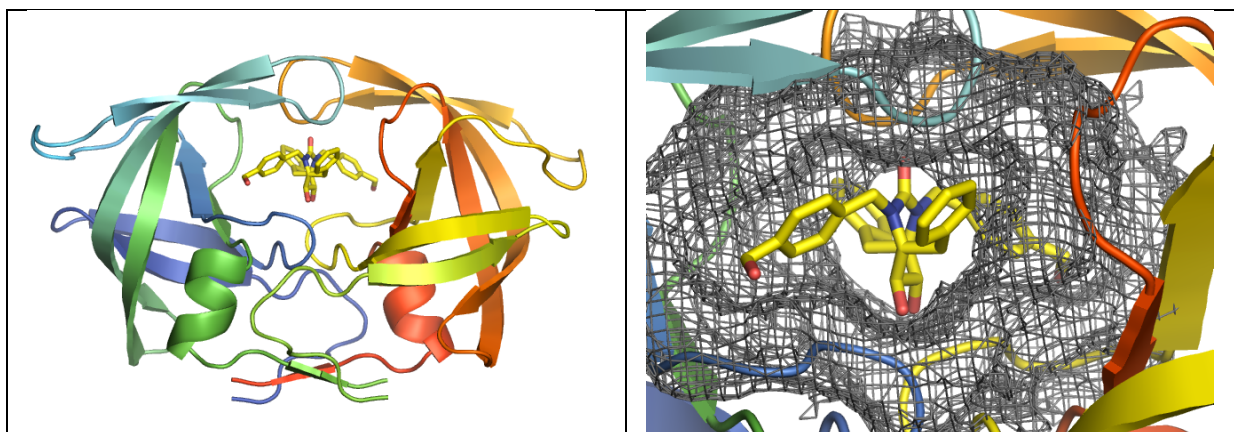
Diego Enry Barreto Gomes¹, Gabriel Limaverde Soares Costa Sousa^{1,3}, Alan Wilter Sousa da Silva², Pedro Geraldo Pascutti¹

(1) Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ – Brasil. (2) EMBL - EBI, Wellcome Trust Genome Campus, Cambridge - Inglaterra (3) Instituto Nacional do Câncer, Rio de Janeiro, RJ – Brasil. Distribuído via requisição por e-mail: diego@biof.ufrj.br

Descrição

O programa **SurfinMD** usa o algoritmo de Connolly¹ para calcular a superfície molecular. Como padrão, o programa calcula a superfície acessível ao solvente, utilizando uma esfera de prova com raio de 1.4Å, compatível com o diâmetro do solvente água. O programa é capaz de identificar a região excluída do solvente na interface entre duas moléculas (ex. enzima-inibidor). A área da superfície pode ser calculada para arquivos no formato PDB, como trajetórias de simulações de dinâmica molecular ou estruturas de Ressonância Magnética Nuclear de proteínas. A área total calculada é dividida em hidrofóbica e hidrofílica, de acordo com a característica em pH 7.0 do resíduo de aminoácido, quando presente (arquivo *surf.dat*). Outros resíduos que não aminoácidos presentes no banco de dados do programa são excluídos desta definição. Além disso o programa discrimina a área média para cada resíduo acompanhada do desvio padrão (arquivo **rsurf.dat**). Os arquivos de saída são simples e compatíveis com o programa **xmgrace**.

Exemplo de uso



Sistema de teste distribuído junto ao programa **SurfinMD**. Protease do vírus HIV-1 complexada com o ligante DMP (PDB 1BVE, estrutura número 1). À esquerda a proteína é apresentada no modelo *Cartoon* e colorida de acordo com a sequência. O ligante DMP, posicionado no sítio ativo da enzima é apresentado pelo modelo de bastões. À direita uma aproximação do sítio ativo onde a superfície molecular dos átomos a 4.0Å do DMP é apresentada como uma rede de pontos conectados (modelo *Mesh*).

¹ Connolly, M.L. "solvent-accessible surfaces of proteins and nucleic-acids" (1983) Science, 221, 709–713

```

#####
# Program: SurfinMD - version 1.05r2
# Diego E.B. Gomes(1), Gabriel Linaverde SCS(1,3)
# Alan Wilter SS (2), Pedro G. Pascutti(1)
# 1) Universidade Federal do Rio de Janeiro
# 2) EMBL - EBI, Wellcome Trust Genome Campus
# 3) Instituto Nacional do Cancer
# mailto: diego@biof.ufrj.br
#####

Usage:
surfinmd -r ref.pdb -i in.pdb -o surf.dat -or rsurf.dat -p 1.4 -d 1.0 -a 6

Please review all usage options [default]

-h = Display this help
-r = Reference Structure [ref.pdb]
-i = Input file [traj.pdb]
-o = Output surface [surf.dat]
-or = Output surface/residue [rsurf.dat]
-p = probe radius [1.4]
-d = probe density [1.0 A^2]
-a = attention number [6]
-ext = (Rvdw + ext) [n]
-scat = (Rvdw * scat) [n]

```

```

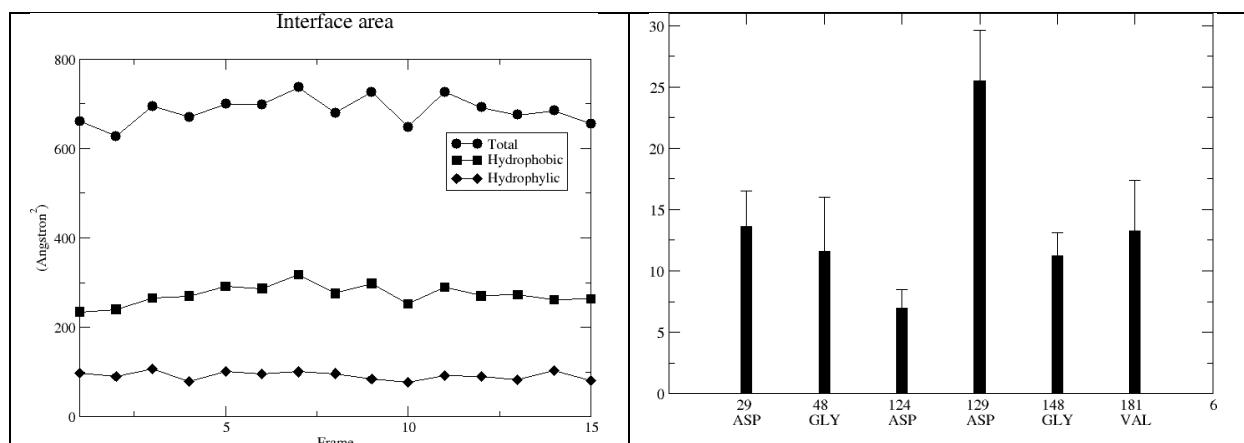
#####
# Program: SurfinMD - version 1.05r2
# Diego E.B. Gomes(1), Gabriel Linaverde SCS(1,3)
# Alan Wilter SS (2), Pedro G. Pascutti(1)
# 1) Universidade Federal do Rio de Janeiro
# 2) EMBL - EBI, Wellcome Trust Genome Campus
# 3) Instituto Nacional do Cancer
# mailto: diego@biof.ufrj.br
#####

model 1 Total = 661.14362 Hydrophobic = 234.44825 Hydrophilic = 96.992592
model 2 Total = 628.13232 Hydrophobic = 240.05519 Hydrophilic = 90.353325
model 3 Total = 694.43530 Hydrophobic = 265.93298 Hydrophilic = 106.72389
model 4 Total = 670.68768 Hydrophobic = 269.95331 Hydrophilic = 79.265175
model 5 Total = 699.53461 Hydrophobic = 291.40411 Hydrophilic = 100.73895
model 6 Total = 698.33350 Hydrophobic = 287.01297 Hydrophilic = 95.645233
model 7 Total = 736.87122 Hydrophobic = 317.93936 Hydrophilic = 100.39503
model 8 Total = 680.43219 Hydrophobic = 276.11438 Hydrophilic = 96.479385
model 9 Total = 727.37067 Hydrophobic = 298.02753 Hydrophilic = 84.102051
model 10 Total = 648.90814 Hydrophobic = 252.24219 Hydrophilic = 76.719124
model 11 Total = 726.03766 Hydrophobic = 289.67581 Hydrophilic = 91.279015
model 12 Total = 692.09900 Hydrophobic = 270.38342 Hydrophilic = 90.170917
model 13 Total = 675.29828 Hydrophobic = 272.61904 Hydrophilic = 83.015640
model 14 Total = 684.70221 Hydrophobic = 261.58954 Hydrophilic = 103.58489
model 15 Total = 656.32068 Hydrophobic = 263.88284 Hydrophilic = 80.889679

Tempo de execucao = 5.6838050 segundos
Tempo por frame = 0.37892035 segundos

```

Telas de execucao do **SurfinMD**. A esquerda o exemplo de uso do programa. A direita uma demonstracao de execucao para o sistema 1BVE.



Resultado do programa **SurfinMD**. A saida do programa consiste em dois arquivos. A esquerda um grafico produzido com o arquivo **surf.dat**, contem a area total, hidrofobica e hidrofila para cada quadro apresentado. A direita, o resultado do arquivo **rsurf.dat**, contendo a area media e o desvio padrao discriminada por resíduo. Pela clareza, apenas algumas areas sao apresentadas no grafico.

Rsurf2grace

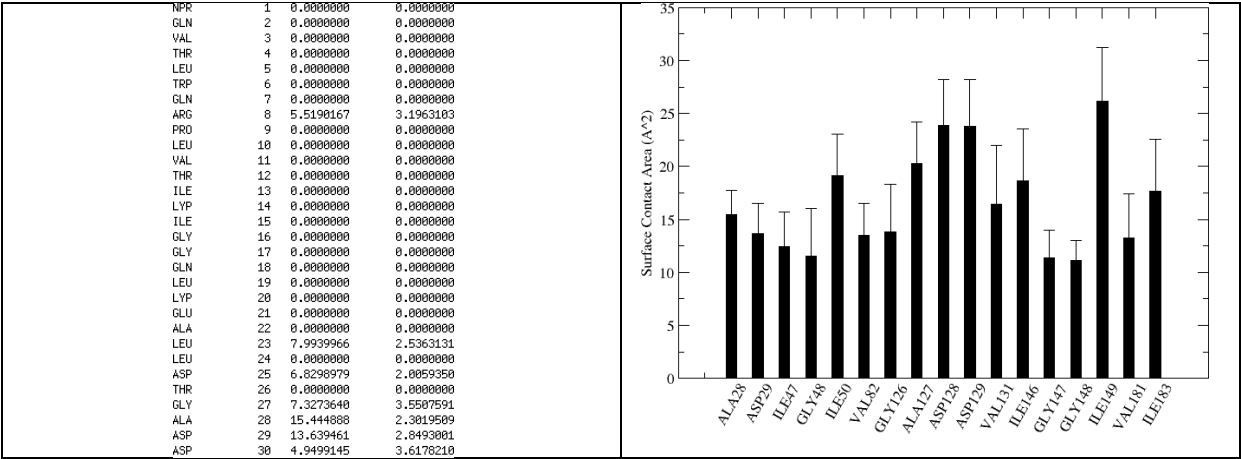
Diego Enry Barreto Gomes¹, Priscila da Silva Figueiredo Celestino¹, Pedro Geraldo Pascutti¹
(1) Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ – Brasil. Distribuído via requisição por e-mail: diego@biof.ufrj.br

Descrição

A informação de área de interface molecular produzida pelo programa **SurfinMD** é de grande valia para interpretação de interações moleculares e planejamento de fármacos, sendo usada amplamente em Teses e trabalhos do nosso grupo e colaboradores. O processamento dos dados da área de interface molecular, entretanto, tem sido um processo lento e tedioso. Para eliminar estes fatores limitantes o programa **rsurf2grace** foi desenvolvido como complemento ao **SurfinMD**, automatizando o pós-processamento do arquivo de interfaces intermoleculares (**rsurf.dat**) e produzindo gráficos apenas com as informações de área consideradas relevantes pelo usuário.

```
#####  
# Program: rsurf2grace #  
# Diego E.B. Gomes, Priscila S.F. Celestino #  
# & Pedro G. Pascutti #  
# Universidade Federal do Rio de Janeiro #  
# +55 21 2562.6507 | diego@biof.ufrj.br #  
#####  
  
Usage: rsurf2grace -i rsurf.dat -o rsurf.agr -c 10.0  
  
-i = .dat Input file (output from SurfinMD)  
-o = .agr Output file (Xmgrace compatible)  
-c = Cut-off in Å² to consider Surface Area  
-h = Display this help
```

Tela de execução do programa **rsurf2grace** apresentando a instrução de uso.



À esquerda o conteúdo parcial do arquivo **rsurf.dat** produzido pelo programa **SurfinMD**, contendo a média e o desvio padrão da área de contato proteína-ligante, discriminada por resíduo. O programa **rsurf2grace** processa os dados apresentando graficamente (por arquivo compatível com o **Xmgrace**) apenas a informação comumente procurada pelos usuários: resíduos com área de interação acima de um critério de corte. À direita, o gráfico automaticamente gerado, contendo apenas as áreas de interação maiores que 10.0Å^2 , entre a protease do HIV e o ligante DMP (PDB 1BVE, estruturas 1 à 20).

Protocolo para construção de sílica amorfa funcionalizada.

Diego E.B. Gomes¹, Roberto D. Lins², Thereza A. Soares², Pedro G. Pascutti¹

(1) Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ – Brasil. (2) Pacific Northwest National Laboratory, Richland, WA – EUA.
Distribuído via requisição por e-mail: diego@biof.ufrj.br – Brasil. Distribuído via requisição por e-mail: diego@biof.ufrj.br

Requisitos para completar o protocolo com sucesso

1) Programas externos

- a) VMD 1.8.7 (Visual Molecular Dynamics)
- b) NAMD-Lite
- c) NAMD

2) Run input file for NAMD-Lite

- a) opt.namdlite
- b) md.namdlite
- c) quick.opt.namdlite
- d) sio.buckingham.inp

2) Nossos programas

- a) pdb2psf

3) Nossos scripts para o VMD

- a) step1.tcl
- b) count_and_label_2breakbonds.tcl

Tipos de átomos

SI = Silício

Ob = Oxigênio "bulk"

Os = Oxigênio de superfície

Hs = Hidrogênio de superfície

Cargas. Note a carga do SI.

SI = 0.900 "bulk"; superfície ligado a silanol isolado ou vicinal

SI = 0.910 superfície ligado a silanol geminal (dois OH)

Ob = -0.450 "bulk"

Os = -0.660 superfície, ligado a hidrogênio

Hs = 0.430 superfície, ligado ao oxigênio

Step 1: building de first structure

Open VMD : inorganic builder: Amorphous SIO

Add exclusion: VMD Selection: z > 10.0 or z < -10.0 : Build device

close VMD

Set unit cell dimensions on "script1.tcl" then run the script

vmd -dispdev text ASiO2.psf ASiO2.pdb -e script1.tcl >script1.log

NOTA: O espaço entre as superfícies pode ser facilmente modificado aumentando o vetor Z.

export PATH=\$PATH:/var/lmdm/dgomes/silica/programas/bin/

export vmdscripts=/var/lmdm/dgomes/silica/programas/vmdscripts/

Podemos usar as cargas de Wensink2000.pdf

#Properties of Adsorbed Water Layers and the Effect of Adsorbed Layers on Interparticle Forces by Liquid Bridging

#E. J. W. Wensink, A. C. Hoffmann, M. E. F. Apol, and H. J. C. Berendsen

#Langmuir, 2000, 16 (19), 7392-7400

#DOI: 10.1021/la000009e

Os = -0.71

```

# Hs = 0.40
# SIs = 0.31

# So que esse paper eh para silica cristalina e nao conta os efeitos de
# usar grupos geminais nem vicinais
# Eu proponho fazer assim:
# 1) Isolada
# Os = -0.71
# Hs = 0.40
# SIs = 0.31
# 2) Geminal
# Os = -0.71 duas vezes
# Hs = 0.40 duas vezes
# SIs = 0.62

Outro metodo eh fazer como Cruz-Chu, mas vai dar muito trabalho e eu nao
tenho o menor tempo agora.
qnet = qsio + qneighbour

*****
Enquanto eu nao faco isso, na minimizacao vamos considerar
SI = 0.9
Ob = 0.45
Os = -0.24
Hs = 0.24
*****
e a carga do grupo funcional, atomo = 1 vai ser 0.24

# As cargas para o 4C_COOH.pdb foram tiradas do GLU

# Step 1: building de first structure
# Open VMD : inorganic builder: Amorphous SIO
# Add exclusion: VMD Selection: z > 10 or z < -10 : Build device
#execute the step1.tcl script then close vmd.
#NOTE
# BEFORE executing the script, remember to correctly set the unit cell
dimensions
#/NOTE
source script1_v5.tcl

# Functions of step1.tcl
# 1) Fix atom properties: names,type,charge,mass.
# 2) Find shell/interior atoms
# 3) Assign "beta 1" to fix interior atoms
# 4) Neutralize the system by randomly removing Dangling atoms ( Silicon <
numbonds 3 or Oxygen numbonds < 2)
# 5) Write output files

# quick and dirty:
# vmd -dispdev text ASiO2.psf ASiO2.pdb < script1_v6beta.tcl >
script1_v6.log

#Step 2: Optimization the structure using NAMD-Lite
# TIP: open the previous log to get the cellBasisVectors :)
mdsim opt.namdlite
mdsim md.namdlite
mdsim quick.opt.namdlite

# Open in VMD and check for periodic disasters ! Manually fix then.
vmd -dispdev text step1.psf step3.coor < fixpbc.tcl > fixpbc.log

```

```

# step3.pdb will be created.

# !! Check if any atom has escaped the surface or whatever weird stuff may
have happened.
# Move step3.pdb to step3.b4.fix.pdb
# Save fixed file as step3.pdb

# PART 2: PREPARE THE SURFACE
# b4_breakbonds
!pdb2psf -i step3.pdb -bonds -d 2.0 -o step4.psf
pdb2psf -i step3.pdb -bonds -d 2.0 -o step4.psf
b4_breakbonds -i step3.pdb -p step4.psf -o step4.pdb

# Execute VMD to measure exposed surface, estimate number of bonds to
delete, and label atoms on the VERY surface.
vmd -dispdev text -pdb step4.pdb -e count_and_label_2breakbonds.tcl >
count_and_label_2breakbonds.log

cp ok2breakbonds.pdb step5.pdb
pdb2psf -i step5.pdb -bonds -d 2.0 -o step5.psf

#check count_and_label_2breakbonds.log for number of bonds to delete.
breakbonds -i step5.pdb -p step5.psf -o step6.psf <<EOF
928
EOF

# POR FAVOR USAR A VERSAO pdb2psf_v9beta2 BETA DOIIS!!!
# Eu coloquei a opcao de ler os ATOMTYPES a partir do .psf e esqueci de
criar uma condicao para o caso do arquivo .psf so conter as BONDS
pdb2psf -i step5.pdb -p step6.psf -o step7.psf -angles

#Attention here, it's really 5 to 7.
cp step5.pdb step7.pdb

../programas/fixcharges.part2.sh step7.psf step8.psf
cp step7.pdb step8.pdb
cp step8.pdb step8_fixed.pdb

# Optimize to remove bad contacts.
/usr/local/bin/charmrun ++local /usr/local/bin/namd2 +p2 step8.em.namd
&>step8.em.log&

*** Adicionar os hidrogenios com o silica protonate
# Olhe o /DATA1/doutorado/programas/2009/silica/add_group_v3/test2

*****
*    necess rio colocar o campo CRYST no arquivo de entrada. *
* Copie de algum .pdb anterior, pode ser at  o step1.pdb *
* os vetores da caixa peri dica n o se alteram. *
*****

# Adicionar H ou OH na silica, e manter n o-fixos os  tomos adicionados.
silica_protonate -i step8.coor -p step8.psf -o step9.pdb -op tmp.step9.psf
## POR FAVOR USAR A VERSAO pdb2psf_v9beta2 BETA DOIIS!!!
## Eu coloquei a opcao de ler os ATOMTYPES a partir do .psf e esqueci de
criar uma condicao para o caso do arquivo .psf so conter as BONDS

# quickly use gromacs to fix residue number.
editconf -f step9.pdb -o step9.pdb
pdb2psf -i step9.pdb -p tmp.step9.psf -o step9.psf

```

```

#corrigir as cargas para o forcefield.
silica_fix_charges -i step9.psf -o step10.psf
cp step9.pdb step10.pdb
cp step10.pdb step10_fixed.pdb

[PARTE 3]
a) Otimizar bonds
/usr/local/bin/charmrun /usr/local/bin/namd2 step10.em.namd ++local +p4 &>
step10.em.log &

b) Otimizar angulos
pdb2psf -i step10.coor -p step10.psf -o step11.psf -angles -psftype
/usr/local/bin/charmrun /usr/local/bin/namd2 step11.em.namd +p4 ++local &>
step11.em.log &

c) Otimizar diedros
# nao precisa mais fazer o fix.charges, o pdb2psf preserva as cargas do psf
anterior com a opcao -psftype
pdb2psf -i step11.coor -p step11.psf -o step12.psf -dihedrals -psftype
/usr/local/bin/charmrun ++local /usr/local/bin/namd2 step12.em.namd +p4 &>
step12.em.log &

d) Pequena simulaco de DM para conferir:
/usr/local/bin/charmrun ++local /usr/local/bin/namd2 step13.em.namd +p4 &>
step13.em.log &

[PARTE 4]
#Adicionar gua. O VMD no consegue lidar com a lista de angulos da silica,
portanto vamos ajuda-lo mantendo apenas as "bonds"
cp step12.psf step13.psf
pdb2psf -i step13.coor -p step13.psf -o step14.psf -psftype

#abra o VMD e salve um .pdb. Isso  necessrio para assinalar o numero do
segmento em todos os atomos. Caso contrario a solvatacao falha.
vmd -dispdev text -psf step14.psf -pdb step13.coor <step14.tcl

#execute o programa silica_charges (NAO  o silica_fix_charges !!!!)
silica_charges (nas simulaces em "oph.silica" eu esqueci de fazer isso
aqui e fiz no "step15"
#ele usa o arquivo step14.psf e produz o arquivo "saida.psf". Copie o
saida.psf de volta para step14.psf

# NOTA
# Mais a frente este numero de segmento vai nos criar um problema.

[Simulaces da superficie SEM GRUPO FUNCIONAL]
Para ficar facil o copy&paste dos comandos:
silica_charges (step14.psf)
ln -s saida.psf step15.psf
ln -s step14.pdb step15.pdb
#Continue do va at [Simulaces FMS] e prossiga a partir de:
pdb2psf -i step15.pdb -p step15.psf -o step16.psf -dihedrals -psftype
# poderia ateh pular o passo step17 mas j que mudamos um pouquinho as
cargas melhor faze-lo for the sake of conciseness
[FIM - Simulaces da superficie SEM GRUPO FUNCIONAL]

[Simulaes FMS]
#Preparamos uma srie de grupos funcionais para serem adicionados
rapidamente a SIO. Processamos os arquivos para que o Carbono 1 seja aquele
que se ligar ao Oxigenio livre na superficie da slica.

```

```

$HOME/silica/grupos_funcionais/
# NOTA por enquanto so o 4CCOO.charmm.pdb, 4CCOOH.charmm.pdb e o
4CNH3.charmm.pdb estao prontinhos.
# As cargas na penultima coluna e os nomes atomos certinhos.
Antes de solvatar precisamos adicionar o grupo funcional passar por uma
nova etapa de otimizaçãõ.
vmd -dispdev text -psf step14.psf -pdb step14.pdb <delete_random_H_v3.tcl
add_function -i h.pdb -p h.psf -o step15.pdb -op step15.psf -fgroup
4CCOO.charmm.pdb

# NOTA
# As vezes os hidrogenios deletados estao em átomos de oxigênio na
superfície porém virados para baixo, em cavidades.
# Nesses casos (raros) use o VMD para mover tanto o grupo funcional quanto
o oxigênio ligado para longe dessa cavidade.
# Depois disso o vai conseguir minimizar e simular a dinamica sem dar erro.
# produce a .psf with angles and dihedrals (ja vai dar para pular o passo
16, as cargas já foram incluídas.
  pdb2psf -i step15.pdb -p step15.psf -o step16.psf -dihedrals -psftype
  ln -s step16.psf step17.psf
  cp step15.pdb step17.pdb
# optimize
/usr/local/bin/charmmrun ++local /usr/local/bin/namd2 +p4 step17.em.namd
&>step17.em.log&

```

[Simulacoes FMS - passo 2]

O VMD (mais especificamente o modulo "psfgen" usado pelo SOLVATE) nao lida direito com os nomes de atomos estranhos que colocamos para a silica, alem disso ele se complica se o "segid" nao esta corretamente escrito no arquivo.

No passo a seguir vamos zerar a configuracao do "numero dos atomos". Eu nao sei usar o VMD para reiniciar a numeracao dos atomos, portanto sem mais demoras, vamos usar o gromacs para nos ajudar aqui e depois o vmd para gerar um novo .psf.

Aqui acontece o seguinte:

- 1) O editconf renumera os atomos e "ResID"
- 2) O VMD designa o "segid" e escreve novos arquivos .psf e .pdb contendo essa informacao.

```

cp step17.coor step18.pdb
editconf -f step18.pdb -o step19.pdb
vmd -dispdev text -e step19.tcl
vmd -dispdev text -e step20.tcl

```

```

../..programas/silica_readpsfinfo/readpsfinfo <<EOF
step16.psf
step21.psf
EOF

```

```

cp step21.pdb mergedstructure.pdb
ln -s mergedstructure.psf min.nvt.psf
ln -s mergedstructure.pdb min.nvt.pdb
ln -s mergedstructure.pdb fixed.pdb

```

Edite o tamanho da celula no eixo Z e de uma folga para as aguas.

```

cellBasisVector1 57.306999 -0.100018 0.000000
cellBasisVector2 0.000000 57.306912 0.101530
#cellBasisVector3 0.000000 0.000000 58.172912
cellBasisVector3 0.000000 0.000000 93.537

```

agora e partir para o abraço

```
/usr/local/bin/charmrn ++local +p4 /usr/local/bin/namd2 min.nvt.namd
>min.nvt.log &&
/usr/local/bin/charmrn ++local +p4 /usr/local/bin/namd2 heat.nvt.namd >
heat.nvt.log &&
/usr/local/bin/charmrn ++local +p4 /usr/local/bin/namd2 md.npt.namd >
md.npt.log &&
/usr/local/bin/charmrn ++local +p4 /usr/local/bin/namd2
md.npt.part0002.namd > md.npt.part0002.log &&
/usr/local/bin/charmrn ++local +p4 /usr/local/bin/namd2
md.npt.part0003.namd > md.npt.part0003.log &&
exit 0
```

```
*****
*****
*****
[Simulacao com a OPH E grupo funcional]
*****
# Preparando o sistema SEM íons. A idéia é abusar do efeito eletrostático
# Mentirinha tá, sobram oxigênios e a gente arranhou uma desculpa p n ter
que inventar um jeito do annealing não fazer "cagada" ex. SI-ligadoa 5
oxigênios.
# As coisas em .tcl que aparecem aqui são para fazer PARA FAZER NA MAO !!!

#abre os arquivos
mol new step17.psf
mol addfile step17.pdb
mol new system.psf
mol addfile center.pdb
set sel [atomselect top all]
# Orienta a OPH com o sitio virado para cima em Z>0
set M [transaxis x -90]
$sel move $M
set M [transaxis y -90]
$sel move $M
# Agora anote o minmax da OPH. E mova a enzima para que o mínimo fique a
20Å da ORIGEM (0.0 0.0 0.0)
measure minmax $sel
{-26.968000411987305 -48.54399871826172 -22.760000228881836}
{27.618000030517578 48.37799835205078 24.302000045776367}
#aqui é s√> somar o mínimo em Z (-22.76) com + 20.
$sel moveby {0.0 0.0 42.76}

#escreve o arquivo final.
source set_unitcell.tcl
set_unitcell 114.613 114.613 58.173 top 90.00 90.00 90.00
$sel writepdb oph.over.silica.pdb
# fim do .tcl

cp step17.coor step18.pdb
editconf -f step18.pdb -o step19.pdb
vmd -dispdev text -e step19.tcl

# Abra o vmd
vmd -f step20.psf step20.pdb -f system.psf oph.over.silica.pdb

set all [atomselect 0 all]
```

```

set sio [atomselect 0 "resname SIO"]
$sio set segid U0
set fms [atomselect 0 "resname FMS"]
$fms set segid F1
$all writepsf step20.fixed.psf
$all writepdb step20.fixed.pdb

vmd -f step20.fixed.psf step20.fixed.pdb -f system.psf oph.over.silica.pdb
1) use o modulo Merge structures e junte no arquivo "tudo.psf , tudo.pdb"
Ok, o VMD estraga com as coordenadas de tudo que eu adicionei p silica.
ignora o "tudo.pdb" e faz isso:

cat step20.fixed.pdb oph.over.silica.pdb > merged.pdb

2) Abra o merged e junte direito os atomos, removendo END e colocando CRYST
apenas no inicio

3) Solvate
vmd -dispdev text -e step20.FMS.tcl

4) Agora preciso das ligacoes e dos nomes e cargas originais.

#### Abra o "step19.pdb" e copie o numero dos RESIDUOS em cima do
"step17.psf"
#### Salve como "step17.fixed.psf"
readpsfinfo <<EOF
step17.fixed.psf
step21.psf
EOF

# Faltam algumas coisinhas que o VMD estraga.
cat mergedstructure.psf | sed s/"HSD CG CPH "/"HSD CG CPH1"/ | sed
s/"HSD CE1 CPH "/"HSD CE1 CPH2"/ | sed s/"HSD CD2 CPH "/"HSD CD2
CPH1"/ > mergedstructure.fixed.psf

# fixe todo mundo menos o solvente
vmd -dispdev text fixnonwater.tcl

# Otimiza !!!
namd2 +p2 silica.oph.em.namd

# Agora é s√> alegria !
cp step21.pdb mergedstructure.pdb
ln -s mergedstructure.psf min.nvt.psf
ln -s mergedstructure.pdb min.nvt.pdb
#ln -s mergedstructure.pdb fixed.pdb
/usr/local/bin/charmrun ++local +p4 /usr/local/bin/namd2 min.nvt.namd
>min.nvt.log &&
/usr/local/bin/charmrun ++local +p4 /usr/local/bin/namd2 heat.nvt.namd >
heat.nvt.log &&
/usr/local/bin/charmrun ++local +p4 /usr/local/bin/namd2 md.npt.namd >
md.npt.log &&
/usr/local/bin/charmrun ++local +p4 /usr/local/bin/namd2
md.npt.part0002.namd > md.npt.part0002.log &&
/usr/local/bin/charmrun ++local +p4 /usr/local/bin/namd2
md.npt.part0003.namd > md.npt.part0003.log &&

# DONE !

```

Parâmetros de simulação para construção de sílica amorfa funcionalizada.

Diego E.B. Gomes¹, Roberto D. Lins², Thereza A. Soares², Pedro G. Pascutti¹

(1) Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ – Brasil. (2) Pacific Northwest National Laboratory, Richland, WA – EUA.
Distribuído via requisição por e-mail: diego@biof.ufrj.br – Brasil. Distribuído via requisição por e-mail: diego@biof.ufrj.br

```
##### step1.tcl #####
# Arquivo step1.tcl
##### step1.tcl #####

#just in case
source /home/dgomes/vmdscripts/set_unitcell.tcl
#surface
set_unitcell 57.307 57.307 58.173 top 89.90 90.00 90.10
#pore
#set_unitcell 171.920 171.920 58.173 top 89.90 90.00 90.10

#create selections
set all [atomselect top all]
set oxygen [atomselect top "name O or type O or element O"]
set silicon [atomselect top "name Si or name Si or type Si or type Si or element Si"]
set num_oxygen [$oxygen num]
set num_silicon [$silicon num]
#correcting name
$oxygen set type O
$silicon set type Si
#setting charge for the BKS force field
$oxygen set charge -1.2
$silicon set charge 2.4
#correcting mass
$oxygen set mass 15.9994
$silicon set mass 28.0855
#find shell atoms
set gridsz 1 ; # grid spacing
set radius 6 ; # sphere radius
set dist 6 ; # distance from surface
set shellatoms [measure surface $all $gridsz $radius $dist]
set shell [atomselect top [concat "index" $shellatoms]]
set interior [atomselect top [concat "not index" $shellatoms]]
set num_shell [$shell num]
set num_interior [$interior num]
#create the shell and interior selections
puts "Found $num_shell shell atoms"
puts "Found $num_interior interior atoms"
#fixing selection !keep this order !
set fix $interior
set free $shell
$fix set beta 1
$free set beta 0

#create list of the dangling atoms on the surface
set dsi [atomselect top "numbonds < 3 and beta 0 and (name Si or name Si or type Si or type Si or element Si)"]
set dox [atomselect top "numbonds < 2 and beta 0 and (name O or type O or element O)"]

#check electroneutrality
set evenodd [expr $num_oxygen %2 ]
if { $evenodd == 1 } { set num_oxygen [expr $num_oxygen -1 ] }
set difference [ expr $num_silicon - ($num_oxygen/2) ] ; # TCL rounds up the difference

#adds set selections to exclude from output file.
if { $difference > 0 } {
  puts "Delete $difference Si"
  set list_to_delete [$dsi get index]
  for {set i 0} {$i < $difference} {incr i} {
    set random [expr int(rand()*[$dsi num])] ; # generate random number within number of dangling silicon atoms
    set deletethis [atomselect top "index [lindex $list_to_delete $random]" ] ; # get the index number of the atom to be deleted
    puts "Deleting atom index [$deletethis list]"
    $deletethis set beta 9 ;# we assign "9" to bfactor, so latter we won't write this kind of atom to output file.
  }
}
if { $evenodd == 1 } {
  set list_to_delete [$dox get index] ; #list of dox
  set random [expr int(rand()*[$dox num])] ; #random number within number of dangling silicon atoms
  set deletethis [atomselect top "index [lindex $list_to_delete $random]" ] ; # get the index number of the atom to be deleted
  puts "Deleting atom index [$deletethis list] (one extra Oxygen to make it even)"
  $deletethis set beta 9 ;# we assign "9" to bfactor, so latter we won't write this kind of atom to output file.
}
```



```

    }
}

if { $difference < 0 } {
    puts "Delete [expr (-1)*($difference)*2] O"
    set difference [expr ((-1)*($difference)*2 )] ; #remove double the of atoms just for oxygen
    set list_to_delete [$dox get index] ; #list of dox
    for {set i 0} {$i < $difference} {incr i} {
        set random [expr int(rand()*[$dox num])] ; #random number within number of dangling silicon atoms
        set deletethis [atomselect top "index [lindex $list_to_delete $random]"] ; # get the index number of the atom to be deleted
        puts "Deleting atom index [$deletethis list]"
        $deletethis set beta 9 ;# we assign "9" to bfactor, so latter we won't write this kind of atom to output file.
    }
    if { $sevenodd == 1 } {
        set list_to_delete [$dox get index] ; #list of dox
        set random [expr int(rand()*[$dox num])] ; #random number within number of dangling silicon atoms
        set deletethis [atomselect top "index [lindex $list_to_delete $random]"] ; # get the index number of the atom to be deleted
        puts "Deleting atom index [$deletethis list] (one extra Oxygen to make it even)"
        $deletethis set beta 9 ;# we assign "9" to bfactor, so latter we won't write this kind of atom to output file.
    }
}

if { $difference == 0 } {
    puts "Great cut ! Your molecule is neutral"
}

# create output selection, excluding atoms labeled with beta 9
set output [atomselect top "not beta 9"]

#writing output
$output writpsf step1.psf
$output writpdb step1.pdb
$output writpdb step1_fixed.pdb
##### Fim do step1.tcl #####

```

```

#####
# Arquivo sio.buckingham.inp, , para simulação com NAMD-Lite
#####
set echo=false end
!      eps      sigma      eps(1:4) sigma(1:4)
!      (kcal/mol) (Å)
!      -----
! I got this values from Table 2
! Cruz-Chu et al
! Water-Silica Force Field for Simulating Nanodevices
! J.Phys. Chem. B, Vol 110, No. 43, 2006
NONBonded O      0.15  4.295  0.15  4.295
NONBonded SI     0.30  3.5   0.30  3.5
set echo=true end

#####
# Arquivo sio.h.inp, para simulação com NAMD
#####
! from supplementary material of:
! Lorenz, C. D.; Crozier, P. S.; Anderson, J. A. & Travesset, A.
! Molecular Dynamics of Ionic Transport and Electrokinetic Effects in Realistic Silica Channels
! The Journal of Physical Chemistry C, 2008, 112, 10222-10232
! and inorganic builder:
!silicon section (experimental alek@ks.uiuc.edu)

! ATOMS
! Bulk silicon      SI
! Bulk oxygen       Ob
! Silanol silicon   SI
! Silanol oxygen    Os
! Silanol Hydrogen  Hs
! Water oxygen      Ow
! Water hydrogen    Hw

BONDS
SI Ob  885.10 1.61
SI Os  428.00 1.42

```

Os Hs 545.00 0.96
SI O 885.10 1.61

ANGLES

SI Ob SI 4.66 174.22
Ob SI Ob 159.57 110.93
Ob SI Os 153.26 111.09
Os SI Os 89.62 116.26
SI Os Hs 57.50 106.00
SI O SI 4.66 174.22
O SI O 159.57 110.93
Ob SI SI 0.0 0.0
SI SI SI 0.0 0.0

DIHEDRALS

!H O SI O 1.3300 1 180.00 ! ALLOW ALC
!H O SI O 0.1800 2 180.00 ! ALLOW ALC
!H O SI O 0.3200 3 180.00 ! ALLOW ALC
Hs Os SI Ob 1.3300 1 180.00 ! ALLOW ALC
Hs Os SI Ob 0.1800 2 180.00 ! ALLOW ALC
Hs Os SI Ob 0.3200 3 180.00 ! ALLOW ALC
!SI O SI O 0.0000 1 180.00 ! ALLOW ALC
!SI Ob SI Ob 0.0000 1 180.00 ! ALLOW ALC
!Ob SI Os Hs 0.0000 1 180.00 ! ALLOW ALC
!SI Ob SI Os 0.0000 1 180.00 ! ALLOW ALC
!Os SI Os Hs 0.0000 1 180.00 ! ALLOW ALC

NONBONDED

!atom ignored epsilon Rmin/2 ignored eps,1-4 Rmin/2,1-4
SI 0.0 -0.3000 2.1475 0.0 -0.3000 2.1475
Ob 0.0 -0.1500 1.7500 0.0 -0.1500 1.7500
Os 0.0 -0.1500 1.7700 0.0 -0.1500 1.7700
Hs 0.0 -0.0460 0.2245 0.0 -0.0430 0.2245
O 0.0 -0.1500 1.7500 0.0 -0.1500 1.7500

HBOND CUTHB 0.5 ! If you want to do hbond analysis (only), then use
! READ PARAM APPEND CARD
! to append hbond parameters from the file: par_hbond.inp

END

Arquivo quick.opt.namdlite
#####

protocol params
numsteps 100
cgmin on
initial config
coordinates step1.pdb
bincoordinates step2.coor
temperature 300
seed 12345

output params
outputname step3
binaryoutput yes
dcdfile step3.dcd
dcdfreq 1

integrator params
timestep 1.0

force field params
structure step1.psf
parameters sio.buckingham.inp
exclude scaled1-4
1-4scaling 1.0
switching on
switchdist 8.0
cutoff 12.0

buckingham = on # replaces van der Waals interaction potential with Buckingham
buckparam = bks # uses BKS parameterizations, other possible values are ttam and fb
bucksafe = on # sets up a safety switch eliminating the artificial well in the potential

```

# mobile atom selection:
# z > 5 or z < -5
fixedAtoms      on
fixedAtomsFile  step1_fixed.pdb
fixedAtomsCol   B

cellBasisVector1 57.306999 -0.100018 0.000000
cellBasisVector1 0.000000 57.306912 0.101530
cellBasisVector1 0.000000 0.000000 58.172912
cellOrigin       0.0      0.0      0.0

#####
# Arquivo opt.namdLite
#####
# protocol params
numsteps      1000
cgmin on
# initial config
coordinates    step1.pdb
temperature    300
seed          12345

# output params
outputname     step1
binaryoutput   yes
dcdfile        step1.dcd
dcdfreq        1

# integrator params
timestep       1.0

# force field params
structure      step1.psf
parameters     sio.buckingham.inp
exclude        scaled1-4
l-4scaling     1.0
switching      on
switchdist     8.0
cutoff         12.0

buckingham = on # replaces van der Waals interaction potential with Buckingham
buckparam = bks # uses BKS parameterizations, other possible values are ttam and fb
bucksafe = on   # sets up a safety switch eliminating the artificial well in the potential

# mobile atom selection:
# z > 5 or z < -5
fixedAtoms      on
fixedAtomsFile  step1_fixed.pdb
fixedAtomsCol   B

cellBasisVector1 57.306999 -0.100018 0.000000
cellBasisVector1 0.000000 57.306912 0.101530
cellBasisVector1 0.000000 0.000000 58.172912
cellOrigin       0.0      0.0      0.0

#####
# Arquivo md.namdLite
#####
# protocol params
numsteps      30000
cgmin on
# initial config
coordinates    step1.pdb
bincoordinates  step1.coor
temperature    300
seed          12345

# output params
outputname     step2
binaryoutput   yes
dcdfile        step2.dcd
dcdfreq        10

```

```

# integrator params
timestep      1.0

# force field params
structure      step1.psf
parameters     sio.buckingham.inp
exclude        scaled1-4
1-4scaling     1.0
switching      on
switchdist     8.0
cutoff         12.0

buckingham = on  # replaces van der Waals interaction potential with Buckingham
buckparam = bks  # uses BKS parameterizations, other possible values are ttam and fb
bucksafe = on    # sets up a safety switch eliminating the artificial well in the potential

```

```

# mobile atom selection:
# z > 5 or z < -5
fixedAtoms      on
fixedAtomsFile   step1_fixed.pdb
fixedAtomsCol    B

cellBasisVector1 57.306999 -0.100018 0.000000
cellBasisVector1 0.000000 57.306912 0.101530
cellBasisVector1 0.000000 0.000000 58.172912
cellOrigin       0.0      0.0      0.0

```

```

#####
# Arquivo min.ptn.fixed.namd
#####

```

```

# input
coordinates      ionized.pdb
structure         ionized.psf
parameters        par_all27_prot_lipid_na.inp
parameters        par.LCX.inp
paratypecharmm    on

```

```

# output
set output        ionized
outputname         $output
dcdfile            ${output}.dcd
xstFile            ${output}.xst
dcdfreq           500
xstFreq           500

```

```

binaryoutput      no
binaryrestart      yes
outputEnergies     1000
restartfreq        1000

```

```

# mobile atom selection:
# not water and not name CLA
fixedAtoms      on
fixedAtomsFile   ionized_fixed.pdb
fixedAtomsCol    O

```

```

# Basic dynamics
exclude          scaled1-4
1-4scaling        1
COMmotion         no
dielectric        1.0

```

```

# Simulation space partitioning
switching         on
switchdist        9
cutoff            10
pairlistdist      12

```

```

# Multiple timestepping
firsttimestep     0
timestep          1
stepspercycle     20

```

```

nonbondedFreq      2
fullElectFrequency  4

# pme parameters
# ldbUnloadPME      yes
#PME                on
#PMETolerance       10e-6
#PMEInterpOrder     4
#PMEGridspacing     1
#PMEPencils         12

PME                yes
PMEGridSizeX       128
PMEGridSizeY       128
PMEGridSizeZ       128
dcdUnitCell        yes

# Temperature control

set temperature     298
temperature         $temperature; # initial temperature

# Pressure coupling !dgomes
useGroupPressure    yes
useFlexibleCell     yes
#useConstantRatio   yes
#useConstantArea    yes # XY is constant

# Berendsen Pressure Coupling
#BerendsenPressure  on
#BerendsenPressureTarget  1.01325
#BerendsenPressureCompressibility  4.57E-5
#BerendsenPressureRelaxationTime  10.0
#BerendsenPressureFreq  2

#Langevin Temperature
langevin            on
langevinDamping     10
langevinTemp        298.15
langevinHydrogen    no

#Langevin Pressure stuff
langevinPiston      on
langevinPistonTarget  1.01325
langevinPistonPeriod  200
langevinPistonDecay  10
langevinPistonTemp   298.15

#reading from .xst file
cellBasisVector1    128.000000  0.000000  0.000000
cellBasisVector2    0.000000  128.000000  0.000000
cellBasisVector3    0.000000  0.000000  128.000000
cellOrigin          0.0      0.0      0.0

# wrap stuff
wrapAll             on
wrapNearest         on

# shake
rigidbonds          all      # SHAKE
rigidtolerance       0.00001 # SHAKE
rigiditerations      400     # SHAKE

# dgomes
margin 5

# Scripting
# run one step to go into scripting mode
minimize 0

# turn off until later
langevinPiston off
minimize          10000

```

```
#####
# Arquivo heat.npt.namd
#####
# input
coordinates      ionized.pdb
structure         ionized.psf
parameters        par_all27_prot_lipid_na.inp
parameters        par.LCX.inp
paratypecharmm    on

bincoordinates    heat.backbone.fixed.restart.coor
binvelocities     heat.backbone.fixed.restart.vel
extendedsystem    heat.backbone.fixed.restart.xsc

# output
set output        heat.backbone.fixed.npt
outputname        $output
dcdfile           ${output}.dcd
xstFile           ${output}.xst
dcdfreq           500
xstFreq           500

binaryoutput      no
binaryrestart      yes
outputEnergies    1000
restartfreq       1000

# mobile atom selection:
# not water and not name CLA
fixedAtoms        yes
fixedAtomsFile     backbone_fixed.pdb
fixedAtomsCol      O

# Basic dynamics
exclude           scaled1-4
l-4scaling         1
COMmotion         no
dielectric         1.0

# Simulation space partitioning
switching         on
switchdist        9
cutoff            10
pairlistdist      12

# Multiple timestepping
firsttimestep      0
timestep           1
stepspcycle       20
nonbondedFreq     2
fullElectFrequency 4

# pme parameters
# ldbUnloadPME    yes
# PME             on
# PMETolerance     10e-6
# PMEInterpOrder   4
# PMEGridspacing   1
# PMEPencils       12

PME               yes
PMEGridSizeX      128
PMEGridSizeY      128
PMEGridSizeZ      128
dcdUnitCell       yes

# Temperature control

#set temperature   298
#temperature       $temperature; # initial temperature

# Pressure coupling !dgomes
useGroupPressure   yes
#useFlexibleCell   yes
#useConstantRatio   yes
#useConstantArea   yes # XY is constant
```

```

# Berendsen Pressure Coupling
#BerendsenPressure          on
#BerendsenPressureTarget    1.01325
#BerendsenPressureCompressibility 4.57E-5
#BerendsenPressureRelaxationTime 10.0
#BerendsenPressureFreq      2

#Langevin Temperature
langevin          on
langevinDamping    10
langevinTemp       298.15
langevinHydrogen   no

#Langevin Pressure stuff
langevinPiston     on
langevinPistonTarget 1.01325
langevinPistonPeriod 200
langevinPistonDecay 10
langevinPistonTemp 298.15

#reading from .xst file
#cellBasisVector1 128.000000 0.000000 0.000000
#cellBasisVector2 0.000000 128.000000 0.000000
#cellBasisVector3 0.000000 0.000000 128.000000
#cellOrigin        0.0      0.0      0.0

# wrap stuff
wrapAll            on
wrapNearest        on

# shake
rigidbonds         all      # SHAKE
rigidtolerance     0.00001  # SHAKE
rigiditerations    400      # SHAKE

# dgomes
margin 5

# Scripting
# run one step to go into scripting mode
minimize 0

# turn off until later
langevinPiston on
run        10000

#####
# Archivo md.namd
#####
# input
coordinates      ionized.pdb
structure         ionized.psf
parameters        par_all27_prot_lipid_na.inp
parameters        par.LCX.inp
paratypecharmm    on

bincoordinates    heat.backbone.fixed.npt.restart.coor
binvelocities     heat.backbone.fixed.npt.restart.vel
extendedsystem    heat.backbone.fixed.npt.restart.xsc

# output
set output        md.part0001
outputname         $output
dcdfile            ${output}.dcd
xstFile            ${output}.xst
dcdfreq           5000
xstFreq           5000

binaryoutput       no
binaryrestart       yes
outputEnergies     1000
restartfreq        5000

# mobile atom selection:

```

```

# not water and not name CLA
fixedAtoms      no
fixedAtomsFile  backbone_fixed.pdb
fixedAtomsCol   O

# Basic dynamics
exclude         scaled1-4
1-4scaling      1
COMmotion       no
dielectric       1.0

# Simulation space partitioning
switching        on
switchdist       9
cutoff           10
pairlistdist     12

# Multiple timestepping
firsttimestep    0
timestep         1
stepspcycle      20
nonbondedFreq    2
fullElectFrequency 4

# pme parameters
# ldbUnloadPME   yes
#PME             on
#PMETolerance     10e-6
#PMEInterpOrder   4
#PMEGridspacing   1
#PMEPencils       12

PME              yes
PMEGridSizeX     128
PMEGridSizeY     128
PMEGridSizeZ     128
dcdUnitCell      yes

# Temperature control

#set temperature  298
#temperature      $temperature; # initial temperature

# Pressure coupling !dgomes
useGroupPressure  yes
#useFlexibleCell  yes
#useConstantRatio yes
#useConstantArea  yes # XY is constant

# Berendsen Pressure Coupling
#BerendsenPressure on
#BerendsenPressureTarget 1.01325
#BerendsenPressureCompressibility 4.57E-5
#BerendsenPressureRelaxationTime 10.0
#BerendsenPressureFreq 2

#Langevin Temperature
langevin          on
langevinDamping   10
langevinTemp      298.15
langevinHydrogen  no

#Langevin Pressure stuff
langevinPiston    on
langevinPistonTarget 1.01325
langevinPistonPeriod 200
langevinPistonDecay 10
langevinPistonTemp 298.15

#reading from .xst file
#cellBasisVector1 128.000000 0.000000 0.000000
#cellBasisVector2 0.000000 128.000000 0.000000
#cellBasisVector3 0.000000 0.000000 128.000000
#cellOrigin       0.0 0.0 0.0

# wrap stuff

```



```
wrapAll      on
wrapNearest  on

# shake
rigidbonds   all      # SHAKE
rigidtolerance 0.00001 # SHAKE
rigiditerations 400    # SHAKE

# dgomes
#margin 5 #nao usar margin na simulacao, so na minimizacao

# Scripting
# run one step to go into scripting mode
minimize 0

# turn off until later
langevinPiston on
run          10000000
```