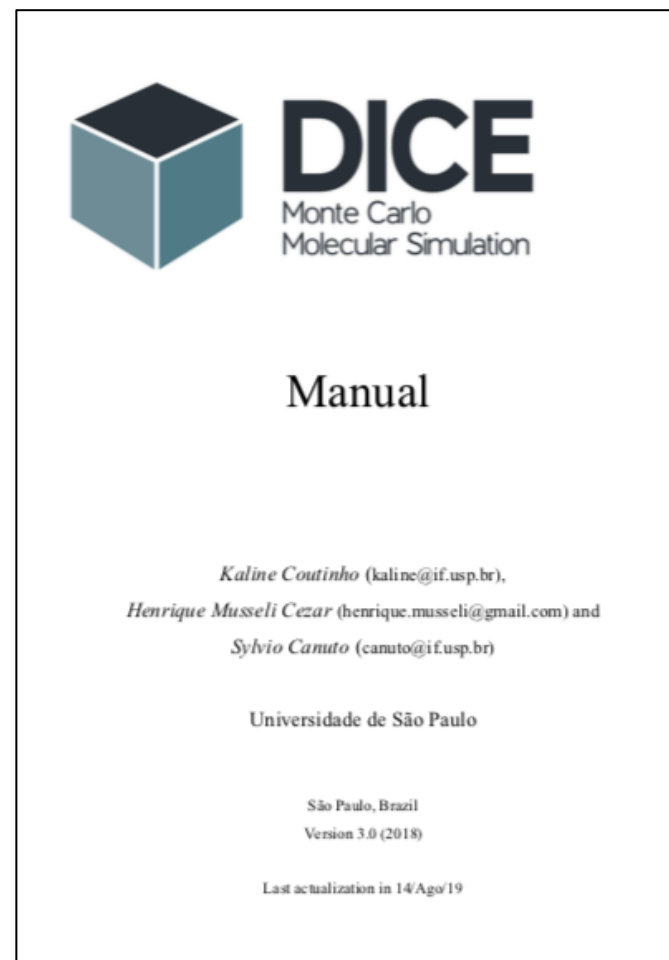


Análise dos resultados de simulações com o DICE



Estrutura do DICE:

1) Aplicabilidade

Realiza simulações de sistemas moleculares com método Monte Carlo utilizando modelo de moléculas rígidas e/ou flexíveis.

2) Arquivos de entrada: 2 (se moléculas rígidas e +1 se flexível)

(i) [input.txt](#) (com informações da molécula: ngr, na, x, y, z, q, ϵ , σ)

```
*
2
13 ala (gromacs2dice) OPT(B3LYP/aug-cc-pVDZ) OPLS(1984JACS,106,6638)
  1  6 -1.2200   1.2800   0.4400   0.000  0.0660  3.5000
  2  1 -2.2600   1.4900   0.1400   0.000  0.0300  2.5000
  2  1 -1.2200   1.0200   1.5100   0.000  0.0300  2.5000
  2  1 -0.6100   2.1900   0.3100   0.000  0.0300  2.5000
  1  6 -0.6700   0.1200  -0.4000   0.200  0.0660  3.5000
  2  1 -0.6700   0.4300  -1.4500   0.000  0.0300  2.5000
  3  7 -1.5400  -1.0500  -0.2700  -0.570  0.1700  3.3000
  2  1 -1.1800  -1.8000  -0.8700   0.185  0.0000  0.0000
  2  1 -1.4200  -1.4200   0.6800   0.185  0.0000  0.0000
  1  6  0.7800  -0.1700  -0.0100   0.520  0.0700  3.5500
  4  8  1.6100   0.8500  -0.3600  -0.530  0.1700  3.1200
  2  1  2.5000   0.5800  -0.0800   0.450  0.0000  0.0000
  4  8  1.1800  -1.1700   0.5300  -0.440  0.2100  2.9600
3 TIP3P (1983JCP,79,926)
  1  8   0.00000   0.117176  0.000  -0.8340  0.1521  3.15061
  2  1   0.75695  -0.468706  0.000   0.4170  0.0000  0.00000
  2  1  -0.75695  -0.468706  0.000   0.4170  0.0000  0.00000
$end
```

Soluto (alanina)

Solvente (água)

(ii) `input.dfr` (necessário para molécula flexibilidade: com informações dos fragmentos e suas conexões, e topologia com parâmetros dos termos do potencial ligado (bonded)).

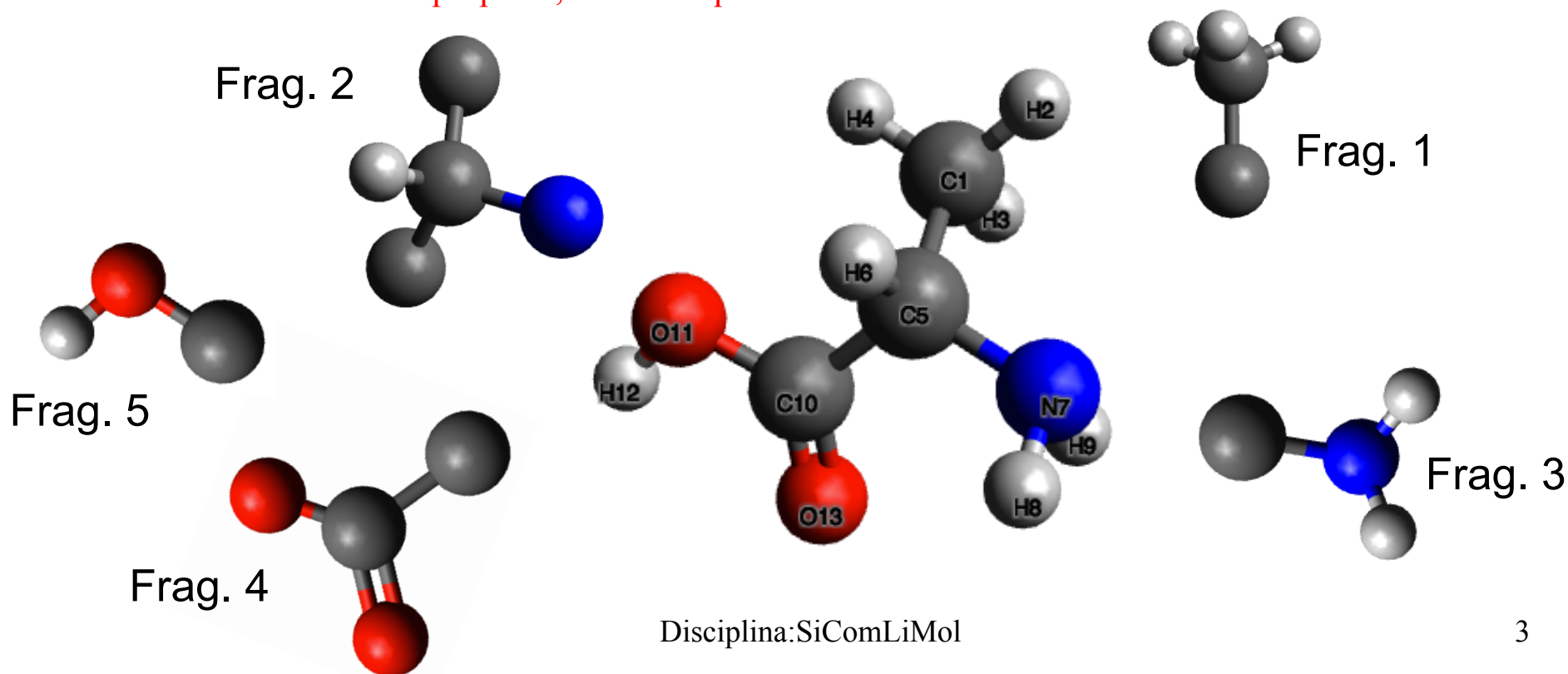
Após otimizar a geometria da molécula converter o xyz para pdb usando o programa babel:

```
$ babel -ixyz alanina.xyz -opdb alanina.pdb
```

Executar o script fragGen.py para gerar os arquivos com uma proposta de fragmentação da molécula:

```
$ python ~/Softwares/dicetools/fragGen.py alanina.pdb
```

Caso não concorde com a proposta, edite o arquivo alanina.dfr



ala.dfr

Este arquivo foi
interamente
gerado pelo
script
gromacs2dice.py
após ter
submetido o
arquivo
alanina.pdb ao
site LigParGen
para uma
proposta inicial
do campo de
força OPLS.
Estes passos
serão
apresentados a
seguir.

\$atoms fragments

1	[1	2	3	4	5	] M
2	[1	5	6	7	10	] M
3	[5	7	8	9	] M	
4	[5	10	11	13	] M	
5	[10	11	12	] M		

\$end atoms fragments

\$fragment connection

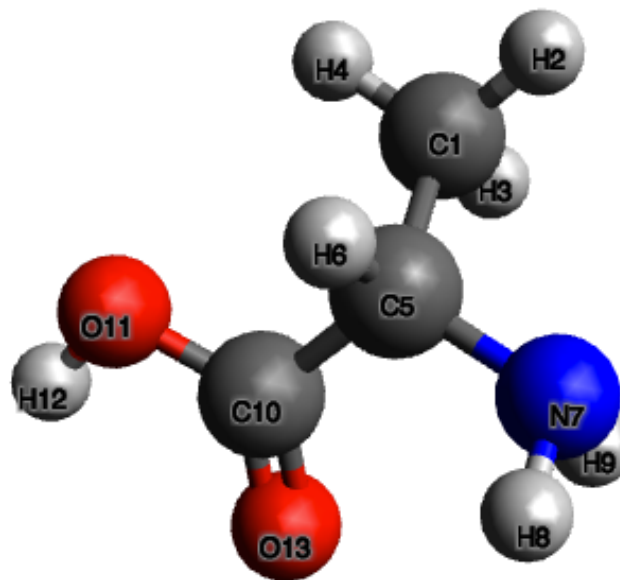
1	2
2	3
2	4
4	5

\$end fragment connection

\$bond

2	1	340.0	1.09
3	1	340.0	1.09
4	1	340.0	1.09
5	1	268.0	1.529
6	5	340.0	1.09
7	5	382.0	1.448
8	7	434.0	1.01
9	7	434.0	1.01
10	5	317.0	1.522
11	10	450.0	1.364
12	11	553.0	0.945
13	10	570.0	1.229

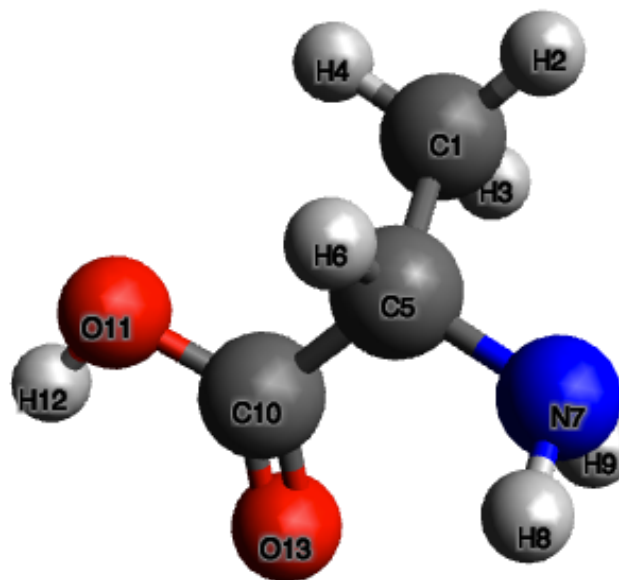
\$end bond



\$angle

2 1 3	harmonic	33.0	107.8
2 1 4	harmonic	33.0	107.8
2 1 5	harmonic	37.5	110.7
1 5 6	harmonic	37.5	110.7
1 5 7	harmonic	56.2	109.47
5 7 8	harmonic	35.0	109.5
5 7 9	harmonic	35.0	109.5
1 5 10	harmonic	63.0	111.1
5 10 11	harmonic	70.0	108.0
10 11 12	harmonic	35.0	113.0
5 10 13	harmonic	80.0	120.4
6 5 10	harmonic	35.0	109.5
3 1 4	harmonic	33.0	107.8
4 1 5	harmonic	37.5	110.7
3 1 5	harmonic	37.5	110.7
6 5 7	harmonic	35.0	109.5
8 7 9	harmonic	43.6	106.4
7 5 10	harmonic	80.0	111.2
11 10 13	harmonic	80.0	121.0

\$end angle

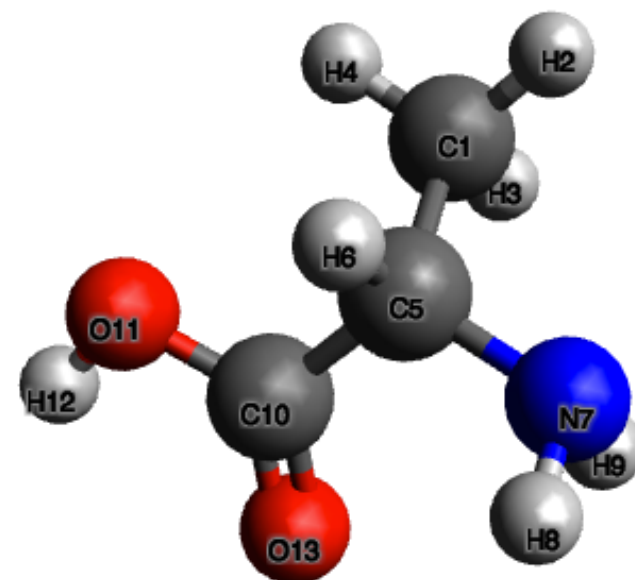


ala.dfr

```

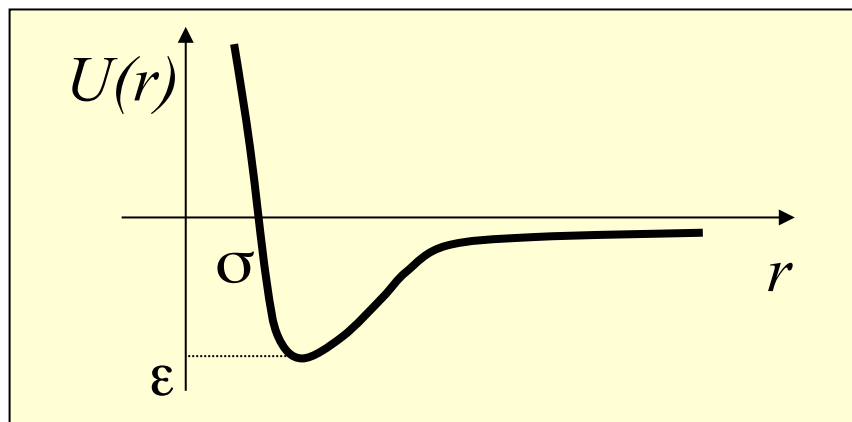
$dihedral
10 5 1 3 OPLS -0.0 -0.0 -0.1 0.0 0.0 0.0
10 5 1 4 OPLS -0.0 -0.0 -0.1 0.0 0.0 0.0
10 5 1 2 OPLS -0.0 -0.0 -0.1 0.0 0.0 0.0
10 5 7 9 OPLS -0.19 -0.417 0.418 0.0 0.0 0.0
10 5 7 8 OPLS -0.19 -0.417 0.418 0.0 0.0 0.0
8 7 5 1 OPLS -0.19 -0.417 0.418 0.0 0.0 0.0
9 7 5 1 OPLS -0.19 -0.417 0.418 0.0 0.0 0.0
8 7 5 6 OPLS 0.0 -0.0 0.4 0.0 0.0 0.0
9 7 5 6 OPLS 0.0 -0.0 0.4 0.0 0.0 0.0
6 5 1 2 OPLS -0.0 -0.0 0.3 0.0 0.0 0.0
6 5 1 4 OPLS -0.0 -0.0 0.3 0.0 0.0 0.0
6 5 1 3 OPLS -0.0 -0.0 0.3 0.0 0.0 0.0
12 11 10 5 OPLS 1.5 5.5 0.0 0.0 0.0 0.0
7 5 1 2 OPLS -1.013 -0.709 0.473 0.0 0.0 0.0
7 5 1 3 OPLS -1.013 -0.709 0.473 0.0 0.0 0.0
7 5 1 4 OPLS -1.013 -0.709 0.473 0.0 0.0 0.0
13 10 5 1 OPLS 0.0 -0.0 0.0 0.0 0.0 0.0
13 10 5 6 OPLS 0.0 -0.0 0.0 0.0 0.0 0.0
13 10 5 7 OPLS 0.0 -0.0 0.0 0.0 0.0 0.0
13 10 11 12 OPLS 0.0 5.5 0.0 0.0 0.0 0.0
11 10 5 1 OPLS 1.0 0.546 0.45 0.0 0.0 0.0
11 10 5 6 OPLS 0.0 -0.0 0.0 0.0 0.0 0.0
11 10 5 7 OPLS 5.26 0.82 0.0 0.0 0.0 0.0
13 10 5 11 AMBER 0.0 21.0 0.0 0.0 180.0 0.00
$end dihedral

```



Potencial implementado no DICE

$$u_{nb} = \sum_i \sum_j f_{ij}^Q \frac{q_i q_j e^2}{r_{ij}} + f_{ij}^{LJ} 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right]$$



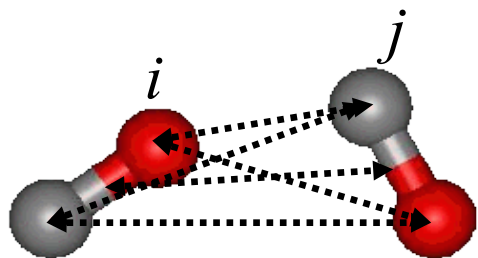
$f_{ij}^Q = f_{ij}^{LJ} = 0$ para as interações: 1-2 e 1-3

para as interações: 1-4

$f_{ij}^{LJ} = 0.5$ and $f_{ij}^Q = 1/1.2$ for the AMBER

$f_{ij}^Q = f_{ij}^{LJ} = 0.5$ for the OPLS

para as interações: além 1-4 $f_{ij}^Q = f_{ij}^{LJ} = 1$



$$\epsilon_{ij} = \sqrt{\epsilon_i \epsilon_j}$$

$$\sigma_{ij} = \sqrt{\sigma_i \sigma_j} \quad \text{ou} \quad \sigma_{ij} = \frac{\sigma_i + \sigma_j}{2}$$

Arquivo: *.txt

Potencial implementado no DICE

$$u_{\text{bond}} = \sum_i^{\text{bonds}} K_r^i (r_i - r_{\text{eq}})^2,$$

$$u_{\text{angle}} = \sum_i^{\text{angles}} K_{\theta}^i (\theta_i - \theta_{\text{eq}})^2,$$

$$u_{\text{improper}} = \sum_i^{\text{improper}} K_{\phi}^i (\phi_i - \phi_{\text{eq}})^2,$$

$$u_{\text{improper}} = \sum_i^{\text{improper}} \frac{V_2^i}{2} [1 + \cos(2\phi_i - f_2^i)].$$

$$\begin{aligned} u_{\text{torsion}}^{\text{OPLS}} = \sum_i^{\text{dihedral}} & \frac{V_1^i}{2} [1 + \cos(\phi_i + f_1^i)] \\ & + \frac{V_2^i}{2} [1 - \cos(2\phi_i + f_2^i)] \\ & + \frac{V_3^i}{2} [1 + \cos(3\phi_i + f_3^i)] \\ & + \frac{V_4^i}{2} [1 - \cos(4\phi_i + f_4^i)], \end{aligned}$$

Arquivo: *.dfr

(iii) `input.in` (arquivos padrão, com informações da simulação: termalização/equilíbrio). Exemplos de arquivos para simulação de moléculas **rígidas em solução aquosa**. `ala-w.ter`

```
ncores = 8
title = Simulation of alanine + 1000 H2O - NPT - thermalization
ljname = ala-w.txt
outname = out-ala-w
init = yes
coolstep = 150
nmol = 1 1000
dens = 1.0
temp = 298.15
press = 1.0
accum = no
vstep = 20000
nstep = 5
iprint = 1
isave = 100
irdf = 0
iratio = 10
seed = 65
upbuf = 50
$end
```

`ala-w.in`

```
ncores = 8
title = Simulation of alanine + 1000 H2O - NPT - equilibrium
ljname = ala-w.txt
outname = out-ala-w
init = no
accum = no
vstep = 40000
nstep = 5
iprint = 1
isave = 100
irdf = 5
iratio = 50
seed = 1234
upbuf = 50
$end
```

(iii) [input.in](#) (arquivos padrão, com informações da simulação: termalização/equilíbrio). Exemplos de arquivos para simulação de moléculas **flexíveis em vácuo**.

```
ncores = 8
title = Simulation of isolated alanine - NVT - thermalization
ljname = ala-ok.txt
outname = out-ala
init = yes
coolstep = 0
nmol = 1
dens = 0.001
temp = 298.15
press = 1.0
accum = no
vstep = 0
nstep = 100000
iprint = 1
isave = 100
irdf = 0
iratio = 10
seed = 65
upbuf = 50
flex = ala
equiphi = yes
ntrialphi = 12
pcbmc = 1.0
fudge1j = 0.5
fudgeclb = 0.5
nsf = 10000
sfint = 1000
savefralib = yes
$end
```

[nvt-ala.ter](#)

(iii) `input.in` (arquivos padrão, com informações da simulação: termalização/equilíbrio). Exemplos de arquivos para simulação de moléculas **flexíveis em solução aquosa**. `ala-w.ter`

```
ncores = 8
title = Simulation of alanine + 1000 H2O - NPT - thermalization
ljname = ala-w.txt
outname = out-ala
init = yes
coolstep = 150
nmol = 1 1000
dens = 1.0000
temp = 298.15
press = 1.0
accum = no
vstep = 40000
nstep = 5
iprint = 1
isave = 100
irdf = 0
iratio = 10
seed = 65
upbuf = 50
$end
```

`ala-w.in`

```
ncores = 8
title = Simulation of alanine +n 1000 H2O - NPT - equilibrium
ljname = ala-w.txt
outname = out-ala-w
init = no
accum = no
vstep = 400000
nstep = 5
iprint = 1
isave = 100
irdf = 5
iratio = 50
seed = 1234
upbuf = 50
sampling = 2
flex = ala
equiphi = yes
ntrialphi = 12
pcbmcmc = 0.8
fudge1j = 0.5
fudgeclb = 0.5
nsf = 10000
sfint = 1000
$end
```

3) Execução:

```
$ dice < ala-w.ter > ala-w.ter.out  
$ dice < ala-w.in > ala-w.in.out
```

4) Arquivos de saída:

- *.avr (médias das propriedades acumuladas),
- *.dat (última configuração),
- *.e12, *.e22 (evolução das componentes da energia de interação),
- *.gr.* (função de distribuição radial de pares),
- *.prb (percentagem média de visitação de cada molécula),
- *.res (arquivo para re-start),
- *.xyz.* (evolução das configurações, ou seja “trajetórias”)

5) Para gerar os arquivos *.txt e *.dfr a partir do LigParGen:

Ir ao site <http://zarbi.chem.yale.edu/ligpargen/> fazer o upload do arquivo alanina.pdb, selecionar passos de otimização, métodos de cargas (ver sugestão na figura), submeter e fazer o download dos arquivos TOP e GRO do GROMACS, colocando os seguintes nomes: ala-tmp.itp e ala-tmp.gro

Step 1: Input structure

SMILES

Enter SMILES Code

OR upload MOL / PDB file **(Structures MUST include all hydrogens)**

Escolher Arquivo alanina.pdb

Step 2: Options

Molecule Optimization Iterations 0

Select charge model:

☒ 1.14*CM1A-LBCC (Neutral molecules)

☐ 1.14*CM1A¹ (Neutral or Charged molecules)

Molecule charge 0

Submit Molecule

Sample Benzene

¹ For charged molecules, CM1A charges are NOT scaled by a factor 1.14

Executar o script `reorder_ligpargen.py` para garantir que a ordem dos átomos no arquivo de saída `ala-tmp.itp` fique igual a ordem dos átomos do arquivo de entrada `alanina.pdb`

```
$ python ~/Softwares/dicetools/reorder_ligpargen.py  
alanina.pdb ala-tmp.gro ala-tmp.itp  
$ mv reordered_ala.itp ala.itp  
$ mv reordered_ala.gro ala.gro
```

Executar o script `gromacs2dice.py` para gerar os arquivos de entrada do Dice: `ala.txt` e `ala.dfr` (este arquivo usará o modelo semi-flexível, ou seja, fragmentos rígidos e ângulos diedrais entre os fragmentos flexíveis). Usando a opção `--flexible-fragments` o arquivo `ala.dfr` será gerado para o modelo flexível amostrando todos os graus de liberdade intramolecular.

```
$ python ~/Softwares/dicetools/gromacs2dice.py -g  
--flexible-fragments -p /usr/share/gromacs/top ala.itp  
ala.gro
```

Editar `ala.txt` para mudar as cargas pelas calculadas com QM e conferir a numeração da 1ª. coluna que será usada para os cálculos das RDFs.

Conferir os fragmentos gerados e o campo de força.

Antes de iniciar as simulações é importante conferir/reparametrizar alguns pontos importante do campo de força:

- (i) A distribuição de cargas atômicas do soluto devem ser garadas por cálculos quânticos (QM). A sugestão dos campos de força: HF/6-31G* Pop=ChelpG ou Pop=MK. A sugestão no nosso grupo: MP2 ou B3LYP/aug-cc-pVDZ ou cc-pVDZ Pop=ChelpG ou Pop=MK depois realizar outro cálculo com modelo contínuo do solvente água SCRF=(PCM). Comparar as cargas e verificar se existe ou não grande polarização (>30%). Realizar o último cálculo com o solvente específico e usar estas cargas polarizadas na simulação.
- (ii) Potenciais torsionais entre os fragmentos: usar o script `plot_eff_tors.py` para gerar os arquivos com as rotações e realizar os cálculos QM e o script `fit_torsional.py` que lê as saidas do QM e propoe os parâmetros para os ângulo diedrais clássicos.

Etapas de análise:

1) Conferir dados de entrada no arquivo padrão de saída: [output.out](#)

```
Opening file #####imdz.txt##### ...
Closing file #####imdz.txt##### ...
Generating the initial configuration randomly
Lx= 31.0803967 Ly= 31.0803967 Lz= 31.0803967
20% of the initial config. was generated
40% of the initial config. was generated
60% of the initial config. was generated
80% of the initial config. was generated
100% of the initial config. was generated

Simulation of imidazole + 1000 H2O NPT - thermalization
Geometry and potential file name :imdz.txt
Output file name                :imdz
Initial configuration            : randomly
Cooling steps                   : 150
Cubic box of length             : 31.0804
Cutoff radius, Rc (Angstrom)    : 15.5402
Number of molecules             : 1001
Temperature (Kelvin)            : 298.1500
Pressure (atm)                  : 1.0000
Density (g/cm**3)               : 1.0000
Seed of the random generator    : 65
All molecules                   : MOVING
All boundary conditions         : Molecular Periodic
    NPT ensemble
Number of volume moves (vstep)  : 10000
Number of mol.moves in each vstep: 5
Total number of MC steps        : 50000
Acceptance ratio of new config. : 0.50
Frequency of energy print       : 1
Frequency of configuration print : 100
Frequency of RDF calculation    : 0
Frequency of accept. ratio update: 10
Freq.of vol. accept. ratio update: 50
```

**Especial atenção as
linhas marcadas em
vermelho.**

Ver se a carga total da molécula e o dipole estão adequados.

Molecules used in this simulation

Atomic type	X	Y	Z	Charge	Epson	Sigma	Rvdw
1 molecules of type 1 :							
1 C	-0.0396	0.0548	0.0000	0.2483	0.0800	3.5000	
2 N	-1.1764	0.8061	0.0000	-0.1669	0.1700	3.2500	
1 C	-0.8036	2.1259	0.0000	-0.2622	0.0800	3.5000	
1 C	0.5733	2.1066	0.0000	0.2196	0.0800	3.5000	
3 N	1.0414	0.8161	0.0000	-0.6449	0.1700	3.2500	
4 H	-0.0577	-1.0217	0.0000	0.0729	0.0500	2.5000	
5 H	-2.1192	0.4548	0.0000	0.3104	0.0000	0.0000	
4 H	-1.5210	2.9269	0.0000	0.1773	0.0500	2.5000	
4 H	1.2438	2.9486	0.0000	0.0455	0.0500	2.5000	

Total charge 0.0000

Dipole -4.9182 -0.1068 0.0000 Total Dipole 4.9194 (Debye)

Total Mass 68.1 (a.u.)

1000 molecules of type 2 :

1 O	0.0000	0.0000	0.0000	-0.8476	0.1550	3.1650	
2 H	0.5774	0.8165	0.0000	0.4238	0.0000	0.0000	
2 H	0.5774	-0.8165	0.0000	0.4238	0.0000	0.0000	

Total charge 0.0000

Dipole 2.3512 0.0000 0.0000 Total Dipole 2.3512 (Debye)

Total Mass 18.0 (a.u.)

RDFs that will NOT be calculated

Long-Range Correction (LJ) added in the energy, <U/N>

between molecules type 2 and 1: -0.00007 (kcal/mol)

between molecules type 2 and 2: -0.01158 (kcal/mol)

Total LRC(LJ) per molecule : -0.01165 (kcal/mol)

Energies that will be calculated, see files

type 1 with type 2 : imdz.e12

type 2 with type 2 : imdz.e22

Start cooling process, U/N U1 = 19825.805 65545.250 (kcal/mol)

Step	U/N	U1
1	9424.097	34651.660
2	4161.162	34651.316
3	2104.191	9333.889

```

147      -10.671      -12.496
148      -10.682      -12.502
149      -10.692      -12.399
150      -10.704      -12.407
End    cooling process,      U/N  U1  =      -10.704      -12.407  (kcal/mol)

```

Starts the Markovian Process

NMOVE	RATIO	Hc/N	U_1	Density	
1	0.3716	-10.5902	-12.8523	1.0000	#NPT
2	0.3457	-10.4920	-12.3425	1.0000	#NPT
3	0.3586	-10.4288	-12.3389	1.0000	#NPT
4	0.3337	-10.3681	-12.6824	1.0000	#NPT
5	0.3936	-10.3071	-13.0193	1.0000	#NPT
6	0.4246	-10.2603	-12.2866	0.9979	#NPT
7	0.3836	-10.2261	-13.0863	0.9979	#NPT
8	0.3876	-10.2022	-13.0876	0.9979	#NPT
9	0.4026	-10.1660	-13.3281	0.9979	#NPT
10	0.3956	-10.1394	-12.6357	0.9979	#NPT
11	0.4226	-10.1253	-12.1705	0.9979	#NPT
.					
49999	0.4915	-11.4235	-17.5724	1.0254	#NPT
50000	0.5115	-11.4177	-17.5772	1.0254	#NPT

Averages

```

Potential Energy      <U/N> = -11.3137074
Potential Energy of mol. 1 <U1> = -17.588686
Potential Solute-solvent <Uxs> = -35.177372
Potential Solvent-solvent <Uss> = -11289.8438
Conformational Enthalpy <Hc/N> = -11.3132744
Solvent-solvent Enthalpy <Hc_ss>= -11289.4102

```

Ver se após o processo de resfriamento o sistema tem energia (ou entalpia) por molécula, U/N (ou H/N), e energia (ou entalpia) do soluto, U1 (ou H1), negativos ou próximo.

Verificar a energia de interação soluto-solvente média, <Uxs>.

Standart Deviation

```
sqrt(<(U/N)^2> - <U/N>^2)      = 0.152988344
sqrt(<(U1)^2> - <U1>^2)         = 1.94687712
sqrt(<(Uxs)^2> - <Uxs>^2)       = 3.89375424
sqrt(<(Uss)^2> - <Uss>^2)       = 153.141327
sqrt(<(Hc/N)^2> - <Hc/N>^2)     = 0.15299131
sqrt(<(Hc_ss)^2> - <Hc_ss>^2)   = 153.144302
sqrt(<V^2>-<V>^2)              = 292.46106
sqrt(<Dens^2>-<Dens>^2)         = 0.0098404102
sqrt(<Hc V>-<Hc><V>)           = 176.882141
```

Properties

```
Volume      <V>                (A**3)      = 29790.7031
Density      <Dens>             (g/cm**3)    = 1.00790679
Total Enthalpy <H/N>            (kcal/mol)    = -9.53779125
Thermal expansion ap           (1/K)       = 0.00930593628
Isothermal compressibility bt  (1/atm)      = 7.07256768E-05
Molar specific heat cp         (kcal/mol K)= 0.138736457
See more in file: imdz.avr
```

RANDOM SAMPLING INFORMATION

```
Attempts to move solute : 0.9954
Maximum attempts to move: 1.0131
Minimum attempts to move: 0.9874
See more informations in file: imdz.prb
```

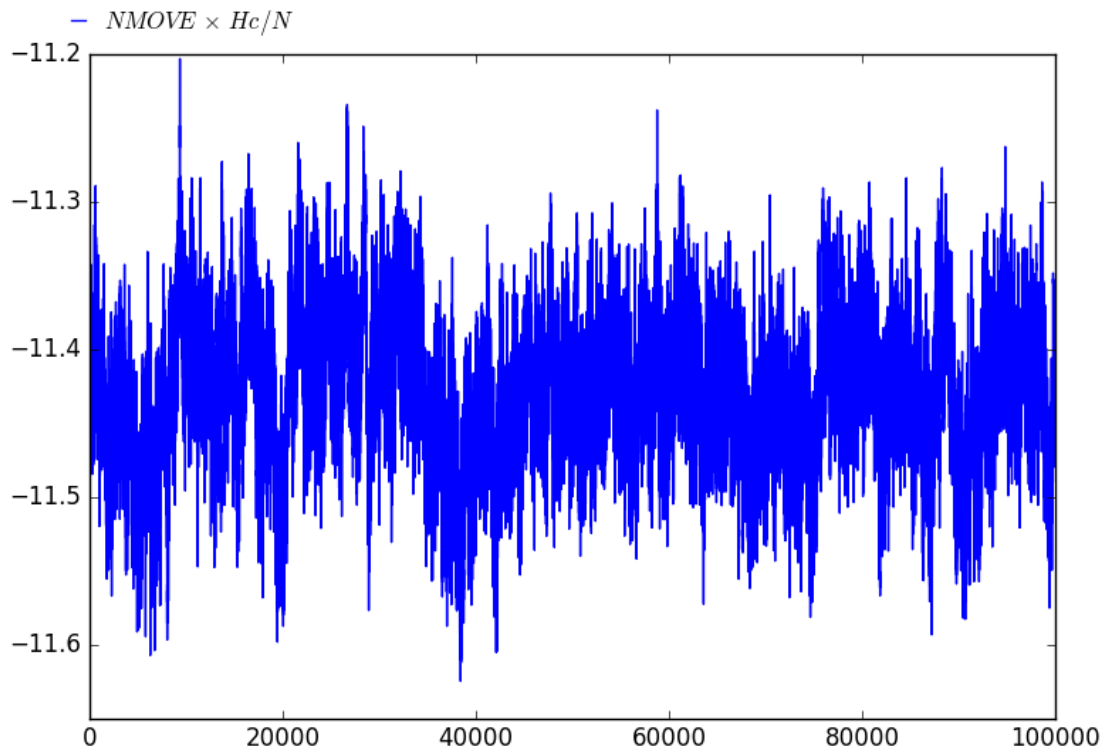
End of the Markovian Process

End of simulation

Verificar a densidade média, <Dens>.

Verificar se a simulação acabou sem mensagem de erro.

2) Graficar grandezas importantes: `imdz.in.out`

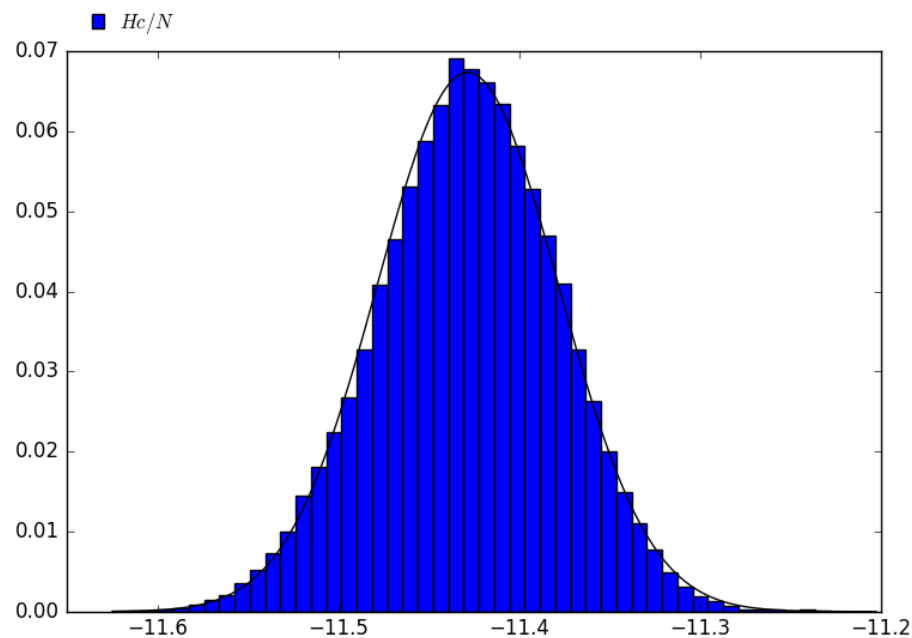


Verificar se a propriedade satisfaz uma distribuição gaussiana.

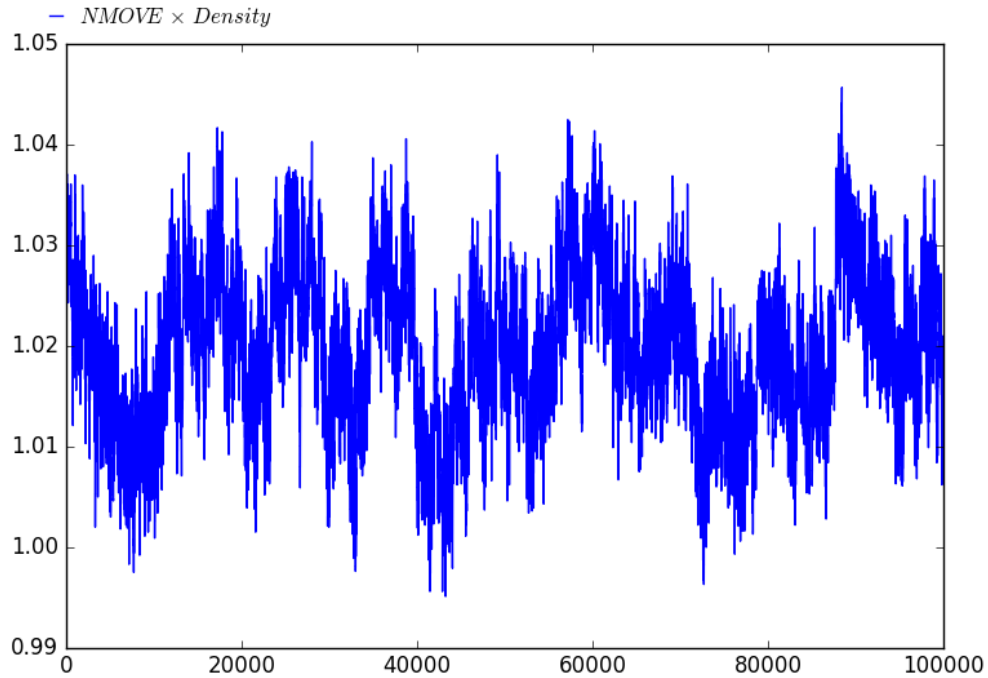
Evolução da entalpia conformacional

Estes gráficos podem ser visualizados com a interface gráfica:

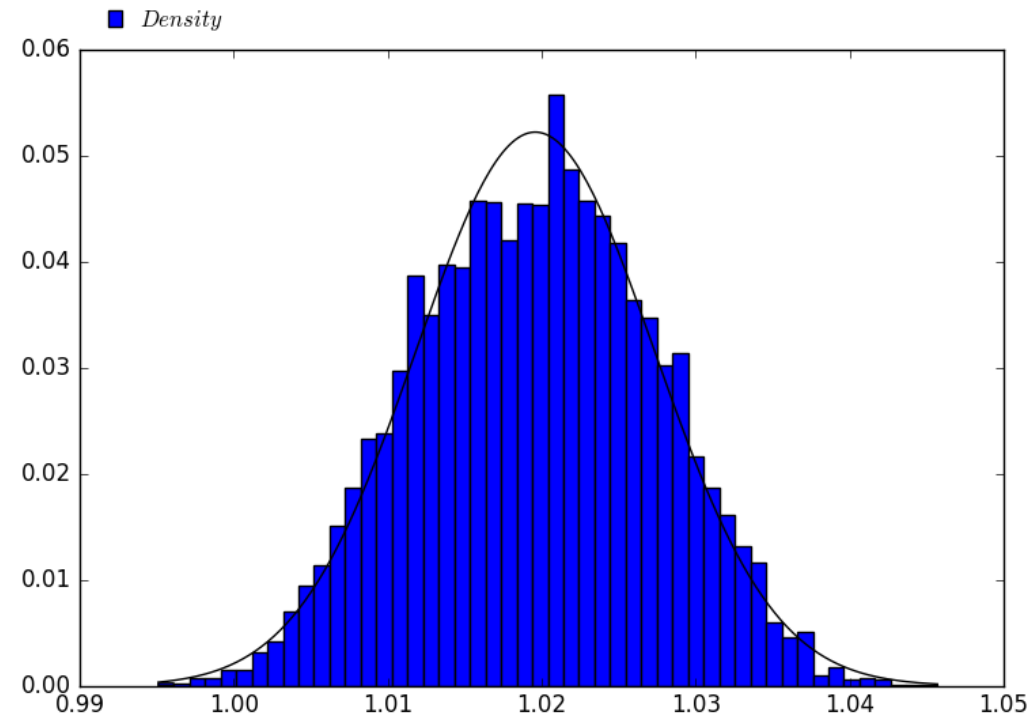
`python path/dicewin.py`



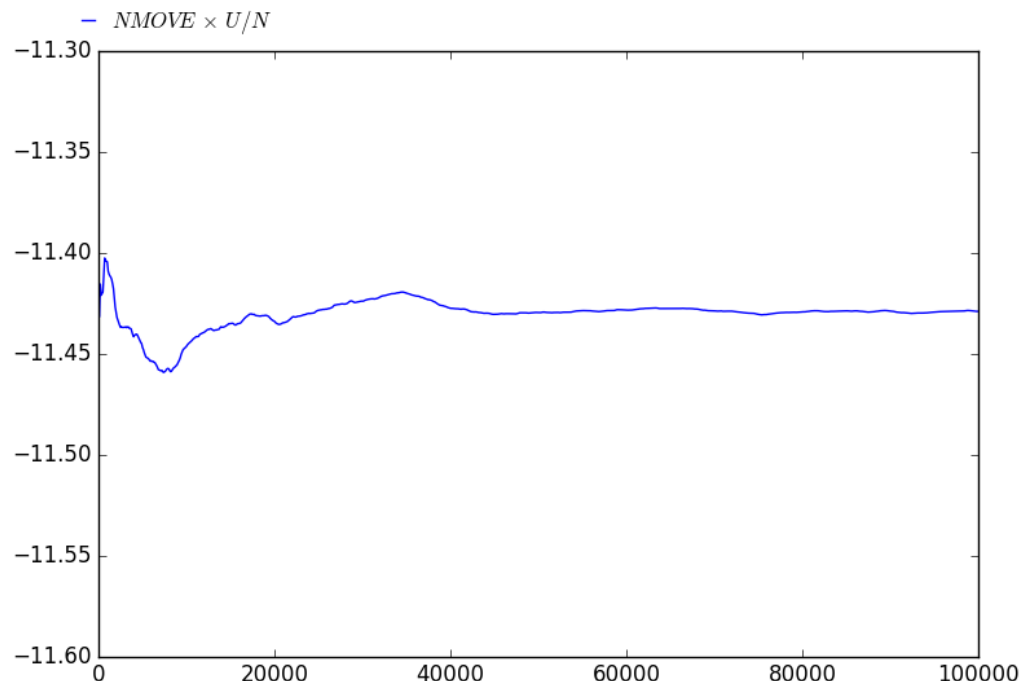
imdz.in.out



Evolução da densidade



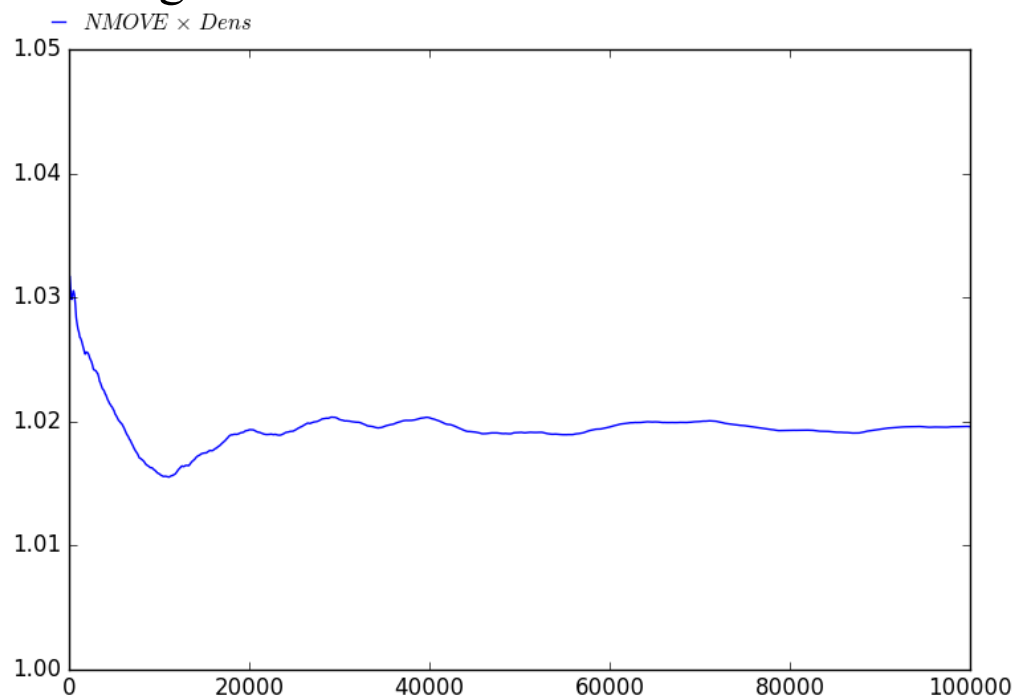
3) Graficar grandezas importantes: `imdz.avr`



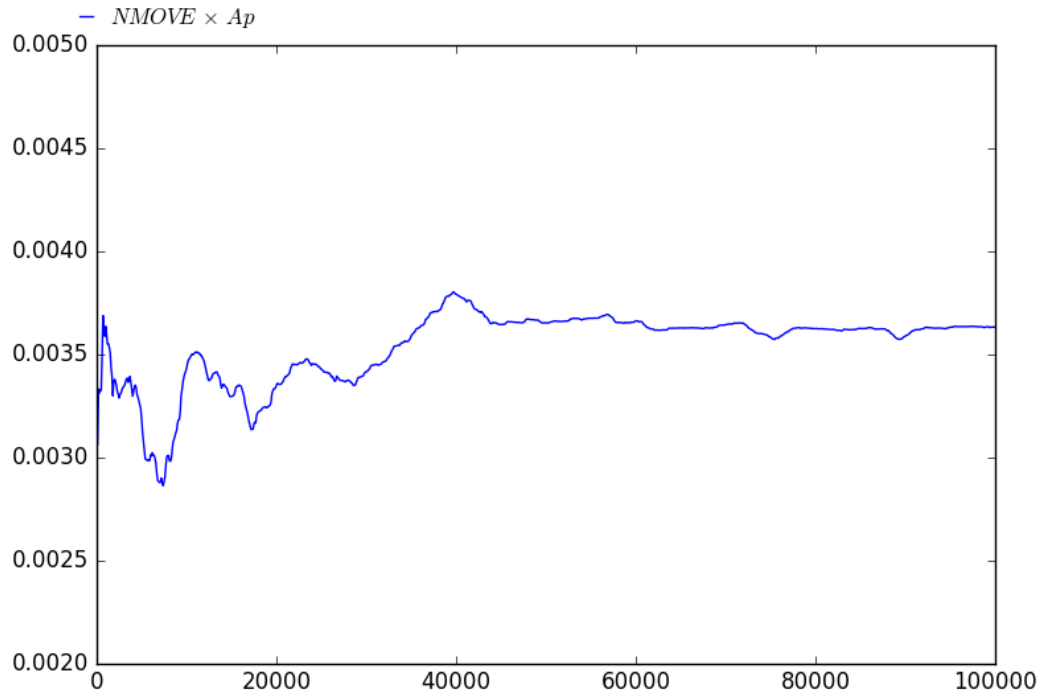
Cuidado com a escala. Pois
nenhum propriedade apresenta
convergência numa escala muito
muito pequena.

Evolução da média acumulada da energia
potencial e da densidade, ou seja, em cada
ciclo MC, NMOVE, o valor apresentado é
a média até ponto.

40 mil ciclos parecem ser suficientes para
ter as médias destas propriedades
convergadas.

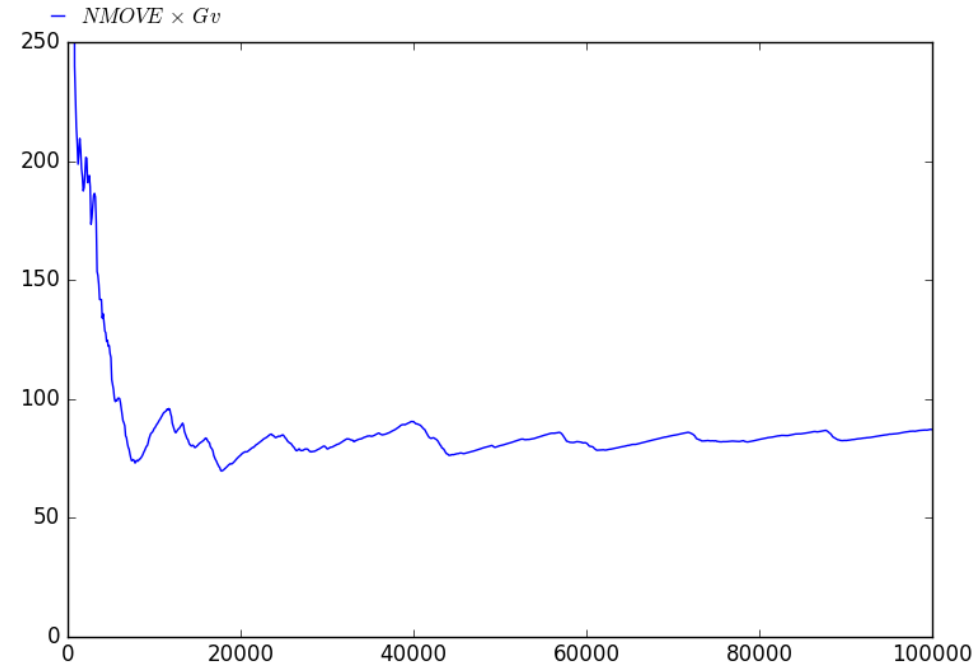


imdz.avr

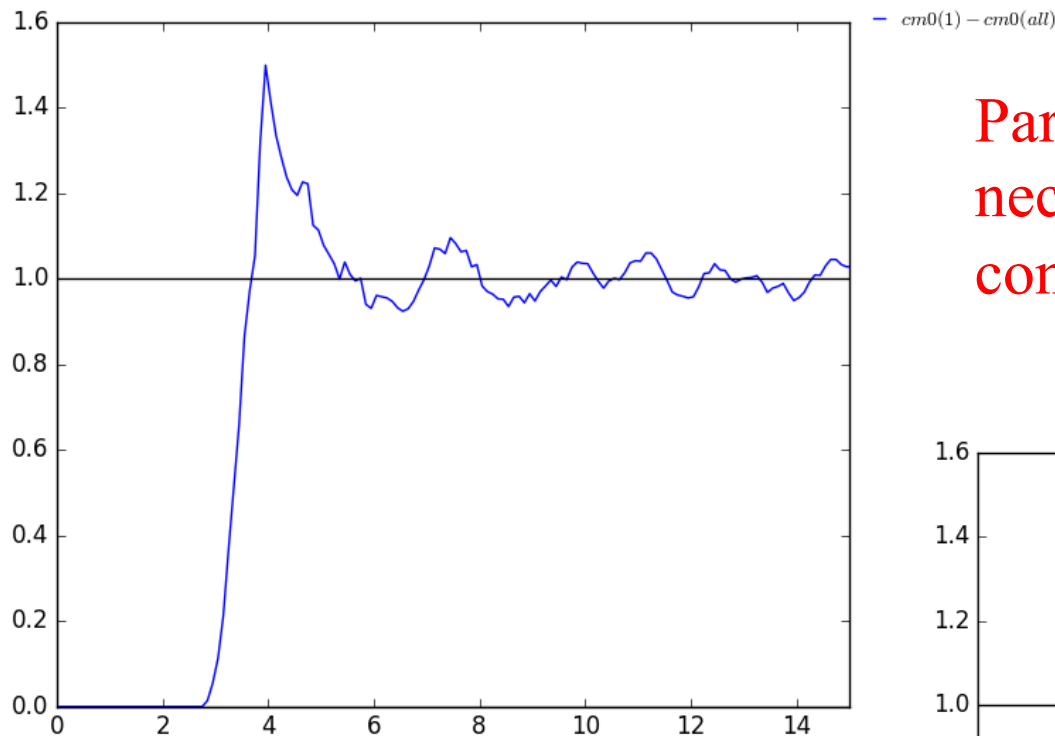


Para estas propriedades a convergência parecer ter sido atingida após 50 mil ciclos. Destes gráficos é possível estimar a variância do valor médio.

Se tiver interesse em propriedades termodinâmicas analisar a evolução da média acumulada. Exemplo: coefs. de expansão térmica (A_p) e pressão térmica (G_v)

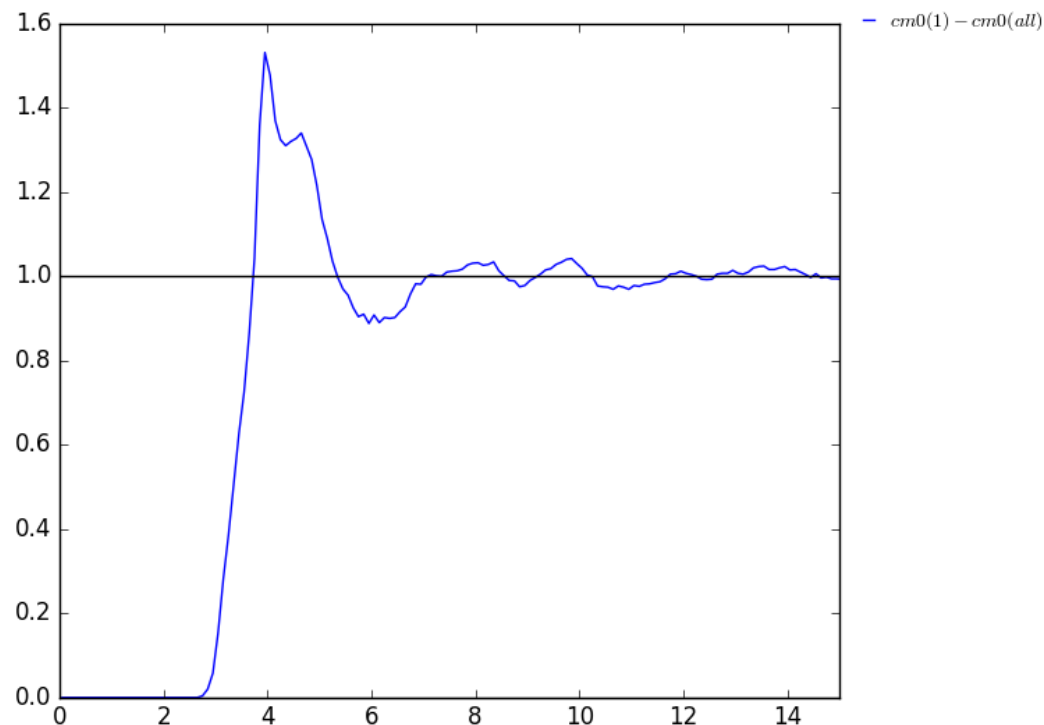


4) Graficar grandezas importantes: imdz.gr

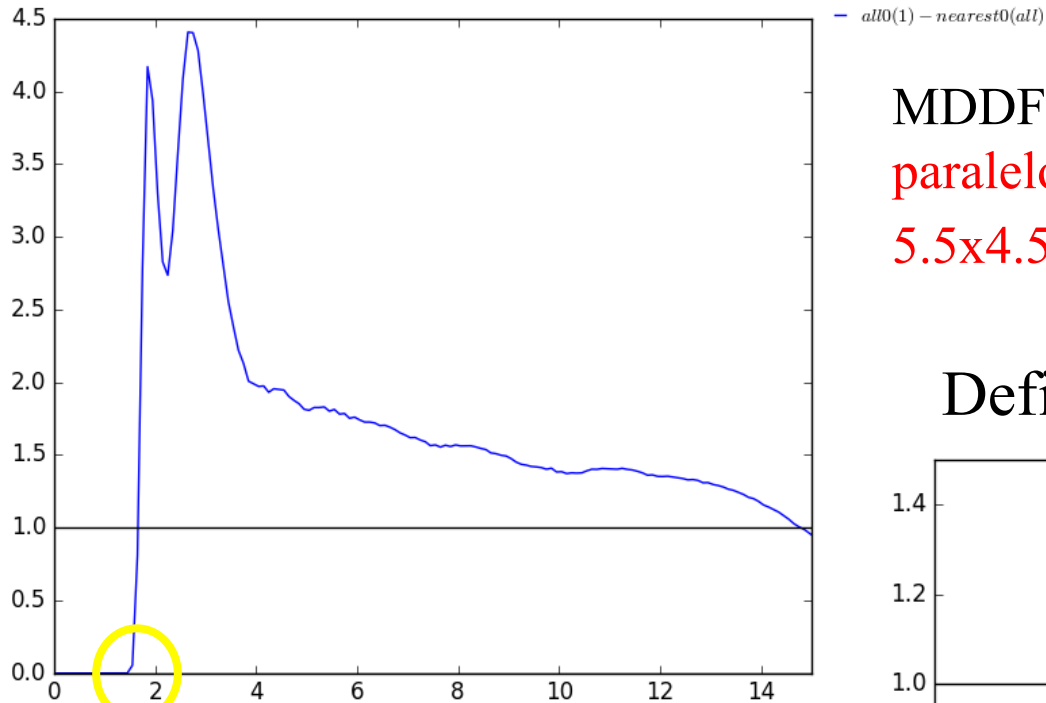


RDF(CM-CM) acumuladas
com 5 mil configurações e 20
mil configurações.

Para obter uma RDF de qualidade são
necessários pelo menos 35 mil
configurações.

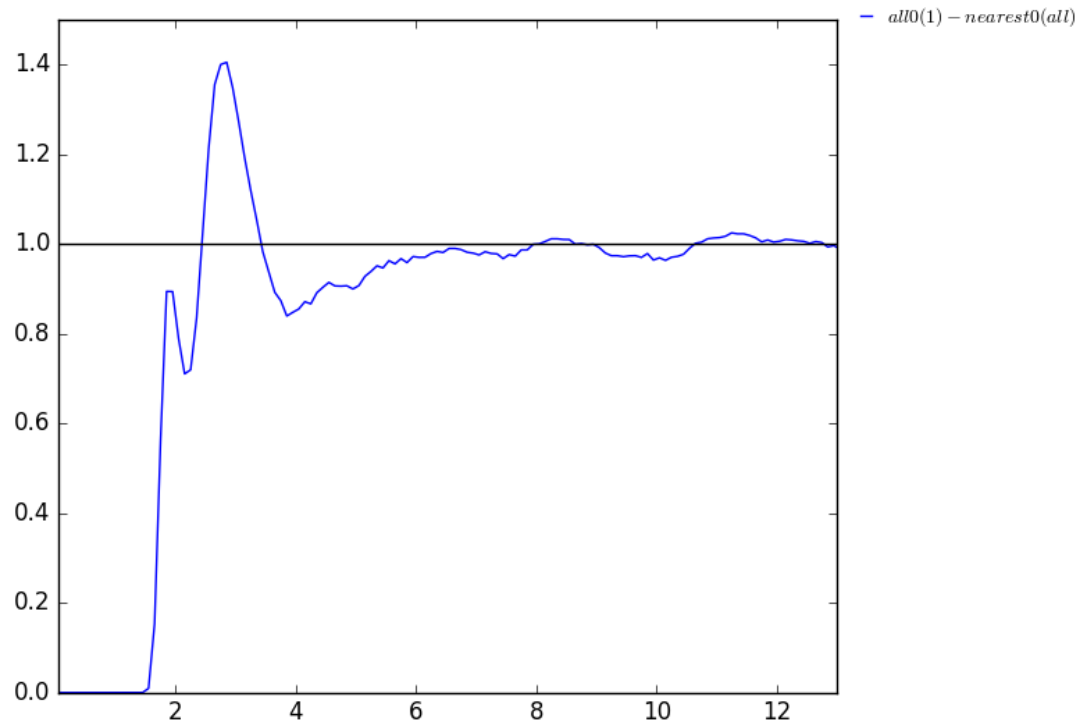


imdz.gr

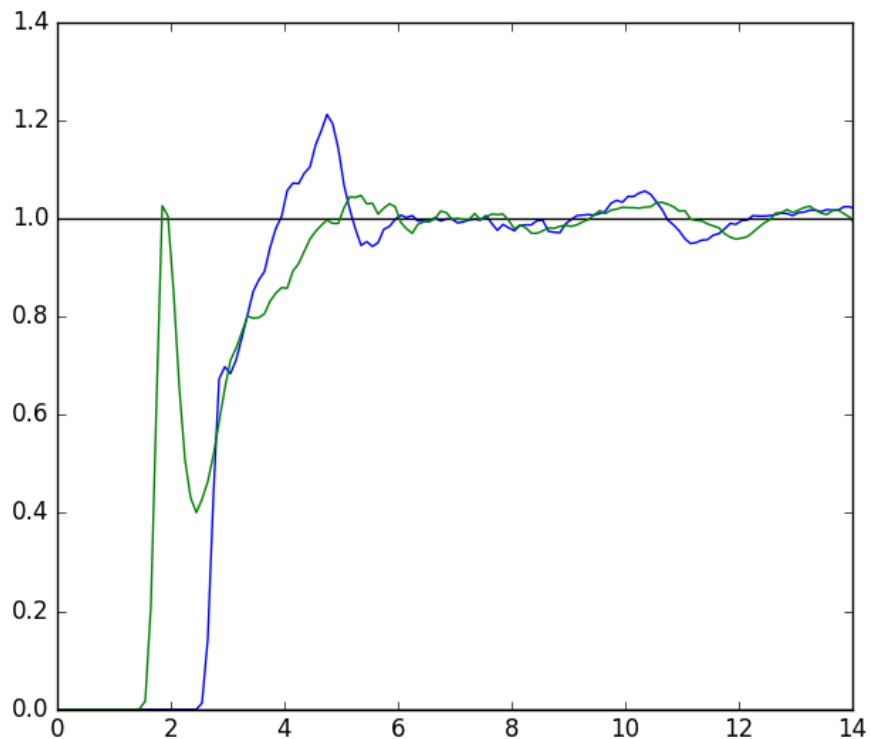


MDDF com normalização esférica e
paralelogrâmica considerando o imidazol com
5.5x4.5x3.5 [J. Chem. Phys. 126 (2007) 34507].

Definição das camadas de solvatação

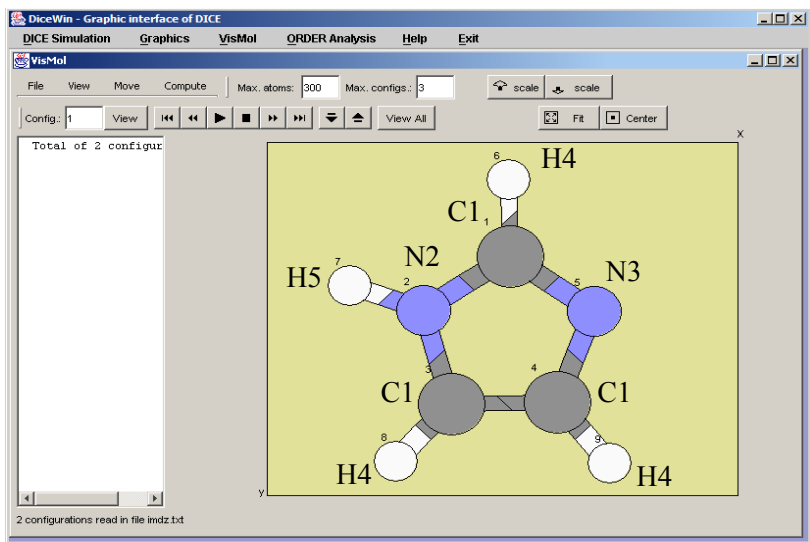
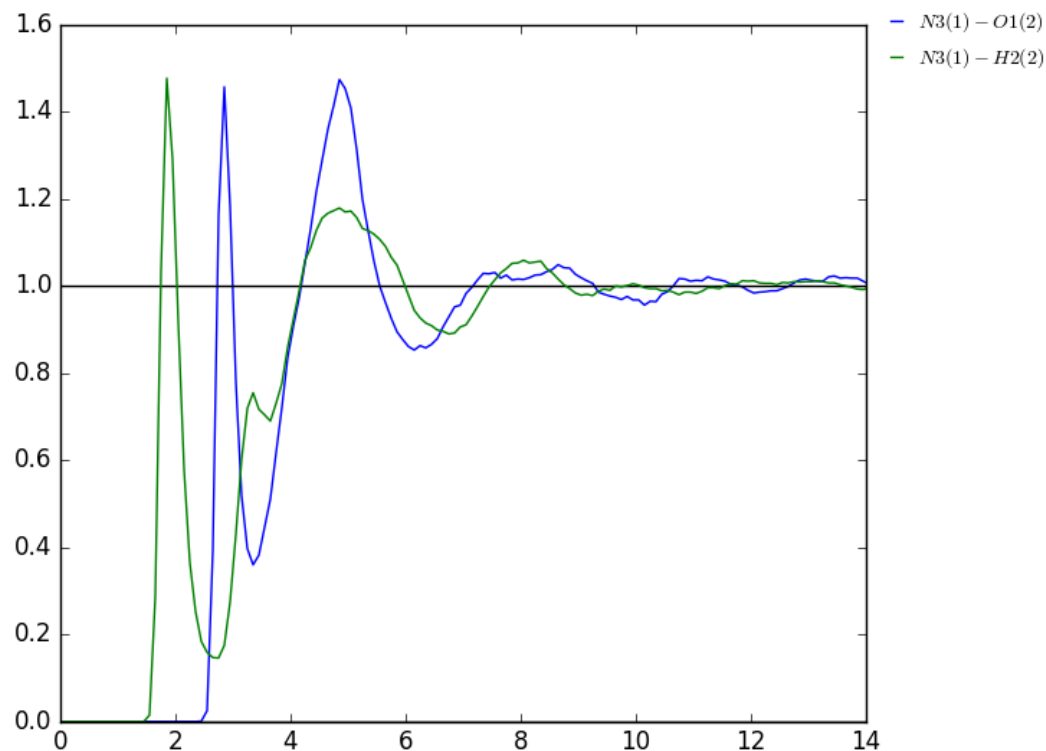


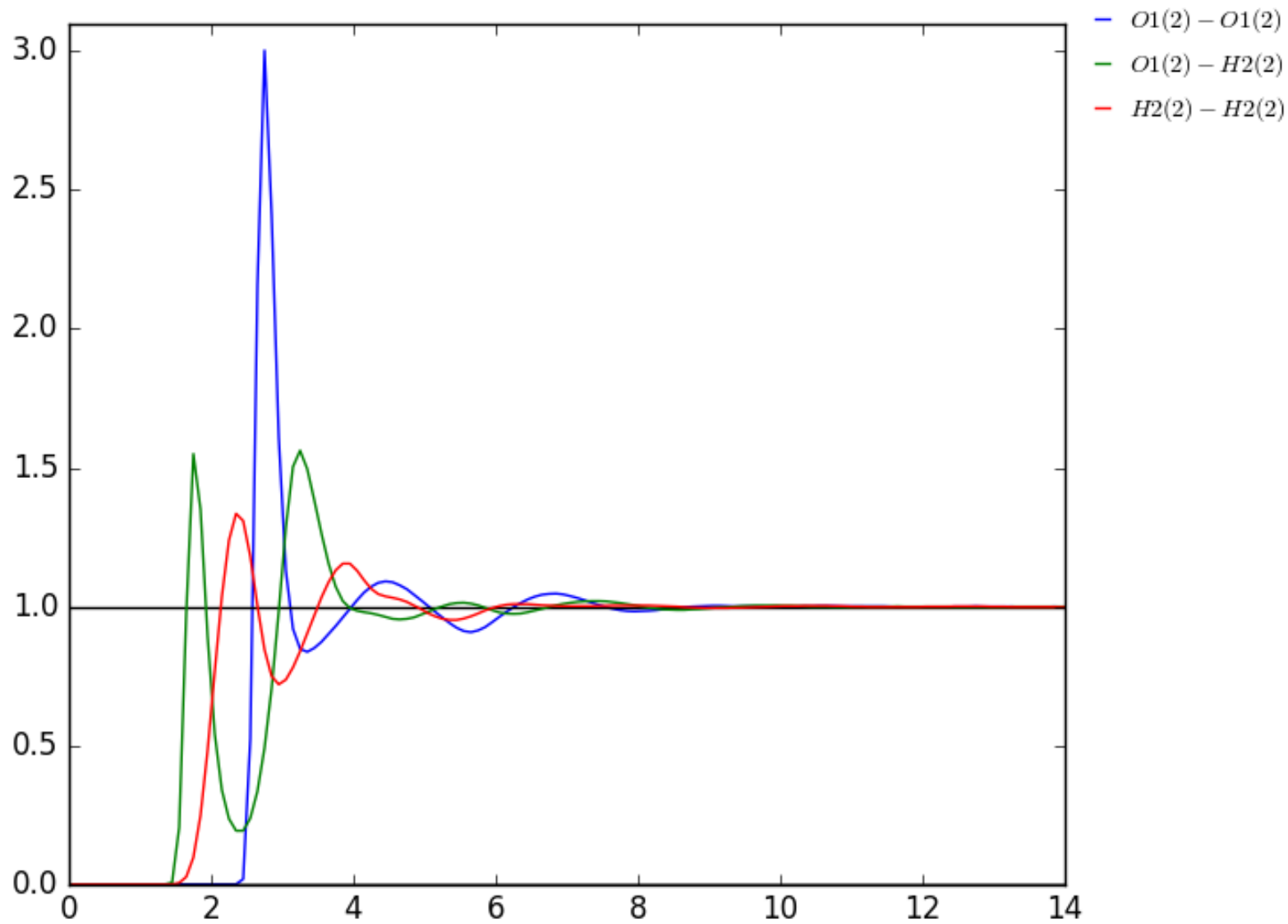
Cuidado com o menor valor!
Deve ser um valor razoável para
distâncias intermoleculares.



Ligações de hidrogênio

Analisar as RDF(N2-O) e RDF(H5-O) para o imidazol como doador e RDF(N3-O) e RDF(N3-H) como aceitador.





RDFs
solvente-
solvente

Programa auxiliar: ORDER

1) Aplicabilidade:

- Calcula RDFs, energias de pares, distâncias, etc.
- Separa configurações num intervalo,
- Separa parte das configurações como: soluto+Hbonds, soluto+camadas, soluto+mais próximo, etc

2) Arquivos de entrada: 3

`input.txt` (com informações da molécula)

`output.xyz.*` (com configurações)

`input.in` (arquivo padrão, com informações da análise)

`top.txt` (informações para QM)

Exemplo para calcular energia de interação entre pares na configuração inicial.

`order.in`

```
ljname = imdz.txt
inname = imdz0.xyz
nmol = 1 1000
dens = 0.00
cm = -1
freeze = 6 1 2
flexible = no
irdf = no
molprint = 501 0
printconfig = no
printformat = 3
topfile = top.txt
printdummy = no
printinterval = 1
angle = no
hbond = no
$end
```

3) Execução:

`order < order.in > order.out`

4) Arquivos de saída: 3 ou mais

*.dst (com informações de energias, distâncias, etc)

all.xyz. (com as configurações selecionadas juntas)

*_g.gjf (arquivos de entrada para QM)

*_g-pc.gjf (arquivos de entrada para QM com ASEC. Só é gerado se “freeze = a1 a2 a3” e “nmol = 1 n” onde a1= número do átomo do soluto que será alinhado no eixo x, a2= número do átomo que será colocado na origem e a3= número do átomo que será colocado no plano xy; n= número de moléculas como cargas pontuais)

*.hbd (com informações das ligações de hidrogênio soluto-solvente. Só é gerado se “hbond = yes”)

*ss.hbd (com informações das ligações de hidrogênio solvente-solvente. Só é gerado se “hbondsolvant = yes”)

*.eij (com informações de energias soluto-solvente)

.or (arquivos de intermediários para QM)

?????.xyz (com cada configuração selecionada separadamente. Só é gerado se “printconfig = yes”)

.gr (com as funções RDFs. Só é gerado se “irdf = yes”)

No	Mol	Rcm	Energy	Rmin(1)	i	j	Rmin(2)	i	j
1	949	2.35545	535.32593	0.91797	7	1	1.381535	1	1
1	993	1.81654	*****	1.16171	7	8	1.812762	7	1
1	504	4.08455	13.76248	1.16725	1	1	2.087215	1	8
1	886	3.60446	29.06610	1.17830	1	1	1.959147	1	8
1	408	3.23503	502.83502	1.37900	1	1	1.810182	1	1
1	33	4.08239	9.42725	1.48231	1	1	2.441798	1	1
1	187	2.92327	*****	1.69579	7	8	2.380640	7	1
1	438	3.26800	-0.13453	1.92220	1	1	2.297140	7	1
1	24	4.39133	4.63077	1.98301	1	1	3.167940	1	1
1	851	3.95793	-3.39751	2.23315	1	1	2.820863	1	8
1	894	4.20383	-0.37269	2.27080	1	1	3.591394	1	1
1	247	2.59681	41.28605	2.28360	1	1	2.369153	6	1
1	194	3.90768	2.19573	2.45705	1	8	2.569758	1	1
1	45	5.18127	-0.46258	2.66156	1	1	4.521725	6	1
1	695	5.61629	0.57343	2.79550	1	1	3.463471	1	1
1	954	4.35676	1.02412	2.84324	1	1	3.251176	1	1
1	211	4.98605	0.24996	2.89439	1	1	3.441900	1	1
1	836	3.53520	2.53516	2.91334	1	1	3.032954	1	1
1	587	5.28588	1.19934	3.04035	1	1	3.608984	1	1
1	546	4.88308	-1.69893	3.04316	1	1	3.104172	1	1
1	450	15.58547	-0.02020	13.17481	1	1	14.117420	1	8
1	361	13.45169	-0.06609	13.17912	7	1	13.209211	6	1
1	556	16.04598	-0.00220	13.18746	1	1	13.693802	1	1
1	502	16.12781	-0.03404	13.19378	1	1	14.138206	1	1
1	197	15.10841	0.00863	13.21771	1	1	13.753456	1	1
1	537	14.42677	0.10786	13.22397	7	8	13.285145	1	8
1	968	15.62071	-0.01337	13.23412	1	1	14.501088	1	1
1	520	15.00999	-0.01478	13.26096	1	8	13.564475	1	1
1	937	14.53488	0.02215	13.26640	1	8	13.494378	1	1
1	985	15.49316	0.00603	13.28657	1	1	14.468156	1	1

Exemplo para calcular energia de interação entre pares na configuração inicial.

**** se referem a energias maiores que 999.99999 kcal/mol.

Moléculas muito próximas podem ter energia positiva de interação, ou seja, estão tão próximas que são repulsivas.

Moléculas muito distantes a energia de interação deve ir para zero.

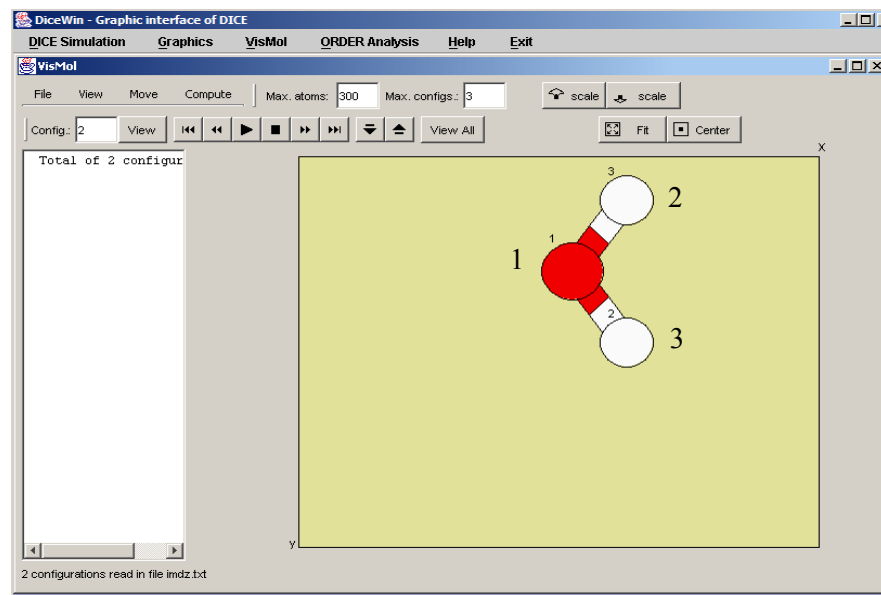
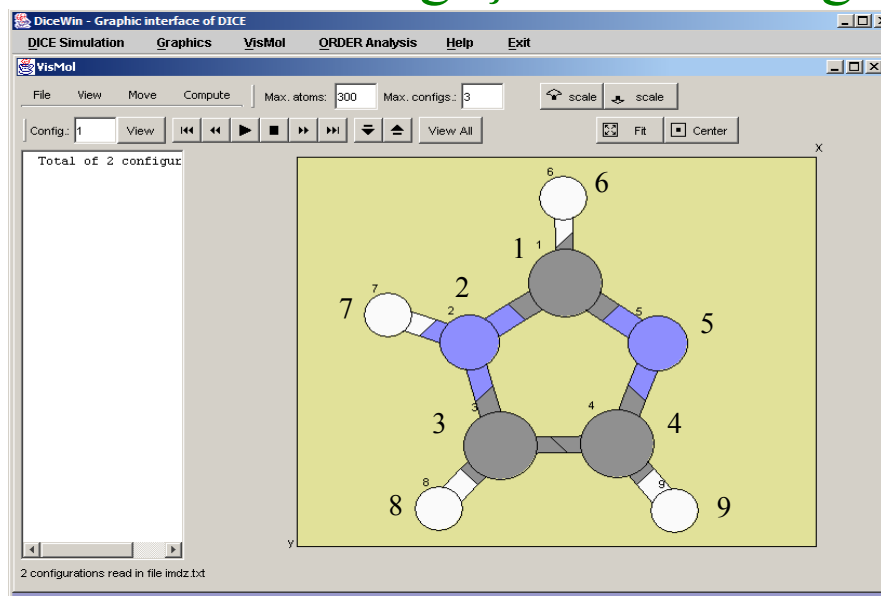
order.in

```

ljname = imdz.txt
iname = imdz.xyz.2
nmol = 1 1000
dens = 0.00
cm = -1
freeze = 6 1 2
irdf = no
molprint = 0 0
printconfig = no
printformat = 3
printdummy = no
printinterval = 1
angle = no
hbond = yes
hbondcriteria = 3.8 40.00 -0.01
soluteacceptor = 2
2 5
solventdonor = 2
2 1
3 1
solventacceptor = 1
1
solutedonor = 1
7 2
$end

```

Exemplo para análise das ligações de hidrogênio



order.out

```
read lname= imdz.txt
Opening file:####imdz.txt####
read inname= imdz.xyz.2
read nmol= 1 1000
read dens= .000000
read cm= -1 (minimum distance)
read irdf= no
read molprint= 0 0
read printconfig= no
read printformat= 3 (GAUSSIAN)
read printdummy = no
read printinterval= 1
irdf = no (Will not calculate the RDFcm-cm
and MDDF)
Opening file:####imdz.txt####
read freeze= 6 1 2
read angle= no
read hbond= yes
read hbondcriteria= 3.80000 40.0000
-1.000000E-02
read soluteacceptor= 2
N 2
N 5
read solventdonor= 2
H 2 O 1
H 3 O 1
read solventacceptor= 1
O 1
read solutedonor= 1
H 7 N 2
```

RDFs that will NOT be calculated

```
Opening file:####imdz.xyz.2####
Analysing configuration 1 L = 30.7640 30.7640 30.7640
Warning 2: The molecule 3 form HB with more than one atom
1( 5 - 1 2 ) 3 3.4861 31.66 -3.6139 3.4115 2.6867
Only 6 molecules were printed in file: imdz00001.or
Analysing configuration 2 L = 30.7989 30.7989 30.7989
Warning 2: The molecule 2 form HB with more than one atom
2( 5 - 1 3 ) 2 3.6566 36.12 -2.2420 3.2903 2.9092
Only 5 molecules were printed in file: imdz00002.or
Analysing configuration 3 L = 30.7378 30.7378 30.7378
Warning 2: The molecule 2 form HB with more than one atom
3( 5 - 1 3 ) 2 3.6993 36.70 -1.2462 3.1221 2.9584
.
Analysing configuration 994 L = 30.8625 30.8625 30.8625
Warning 3: In configuration 994 The molecule 9 does not pass in the
angular and energetic criteria: 3.34315 40.0348 -2.88629
.
Analysing configuration 1000 L = 30.9542 30.9542 30.9542
Only 4 molecules were printed in file: imdz01000.or

1000 configurations were analyzed
```

```
Over 1000 configuration, there are 162 Hbonds in the acceptor atom N
2 of the solvent. ( .16 in average)
Over 1000 configuration, there are 2209 Hbonds in the acceptor atom N
5 of the solvent. ( 2.21 in average)
Over 1000 configuration, there are 1086 Hbonds in the donor group ( N
2, H 7) of the solvent. ( 1.09 in average)
```

imdz.hbd

#	Criteria	3.800	40.000	-.001									
#	No	Hbond	Mol	R_OO	Th_OOH	Energy	Rcm	R_OH	Dip_A	Dip_B	Dip_AB	Th_dip	
	1(2 - 1 3)	2	3.5194	33.07	-.2941	3.3821	2.7363	4.9194	2.3512	3.6729	134.58		
	1(2 - 1 2)	3	3.5290	39.85	-3.6139	3.4115	2.8347	4.9194	2.3512	4.5155	113.81		
	1(5 - 1 2)	4	2.9470	22.86	-3.1075	3.6630	2.0625	4.9194	2.3512	6.9313	37.65		
	1(5 - 1 2)	6	2.7861	2.62	-6.7056	3.7159	1.7877	4.9194	2.3512	6.7997	44.48		
	1(5 - 1 3)	7	3.3255	39.75	-1.3757	3.7883	2.6354	4.9194	2.3512	7.0257	31.93		
	1(1 - 2 7)	9	2.7712	21.56	-5.6534	3.8728	1.8723	4.9194	2.3513	6.9130	38.68		
	2(2 - 1 3)	2	3.4754	26.68	-2.2420	3.2903	2.6206	4.9194	2.3513	4.7558	107.90		
	2(5 - 1 2)	5	2.7927	9.46	-6.1966	3.5264	1.8138	4.9194	2.3512	7.1116	25.69		
	2(5 - 1 2)	8	2.8711	7.43	-6.9137	3.9086	1.8839	4.9194	2.3512	6.4330	59.75		
	2(1 - 2 7)	13	3.3185	33.09	-.6809	4.3390	2.5357	4.9194	2.3512	6.7645	46.14		
	2(1 - 2 7)	14	3.3597	33.80	-3.5284	4.3612	2.5850	4.9194	2.3512	5.3235	93.44		
	3(2 - 1 3)	2	3.1917	29.96	-1.2462	3.1221	2.3783	4.9194	2.3513	4.5778	112.29		
	3(5 - 1 2)	4	2.7287	14.21	-5.2093	3.6050	1.7763	4.9194	2.3512	6.7110	48.57		
	3(5 - 1 2)	6	2.8523	36.13	-5.6350	3.9184	2.1280	4.9194	2.3512	7.2009	16.98		
	3(1 - 2 7)	14	3.4853	26.40	-3.0331	4.5490	2.6225	4.9194	2.3513	6.1530	69.42		
	4(5 - 1 3)	2	3.6121	20.05	-4.0468	3.1175	2.6946	4.9194	2.3513	3.6068	136.28		
	4(5 - 1 2)	5	2.9070	3.49	-6.9747	3.7767	1.9098	4.9194	2.3512	6.5767	54.23		
	4(5 - 1 2)	6	2.7923	20.77	-6.6576	3.8283	1.8909	4.9194	2.3512	7.2677	3.48		
	4(1 - 2 7)	13	3.2555	25.28	-3.8290	4.3348	2.3847	4.9194	2.3512	6.4936	57.47		
	5(2 - 1 3)	2	3.0808	28.78	-3.2435	3.1013	2.2562	4.9194	2.3512	3.8893	129.14		

Executando o dice para alanina flexível em vácuo:

```
$ dice < nvt-ala.in > nvt-ala.in.out
```

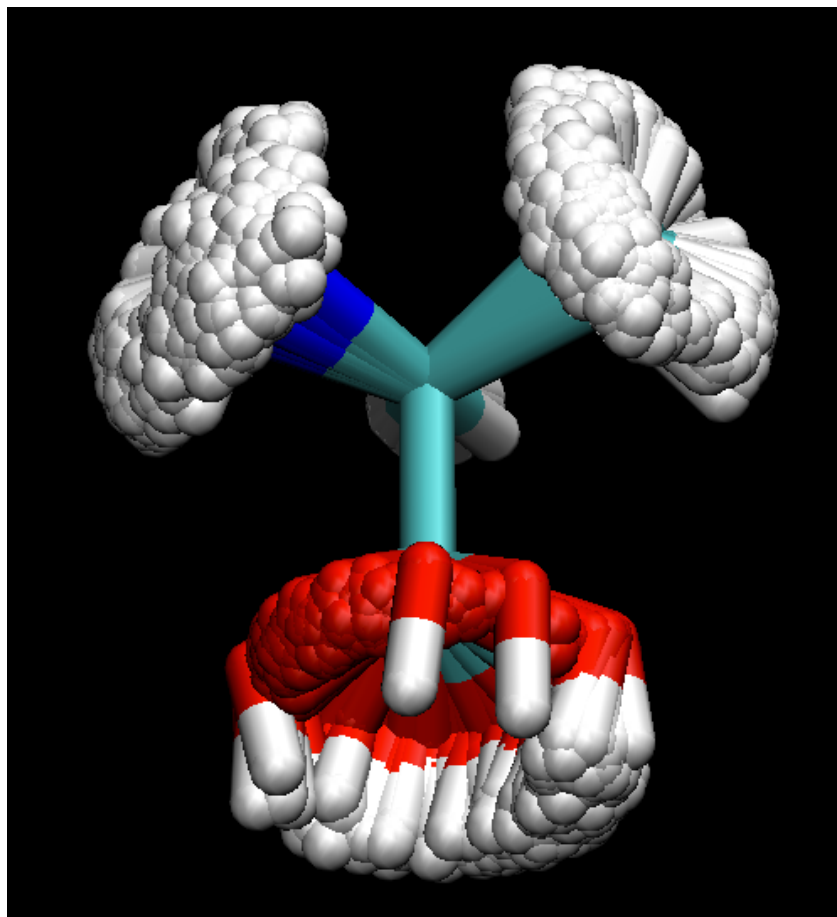
```
$ order < order.in > order.out
```

```
$ anlgeom < anlgeom.in > anlgeom.out
```

Visualizar o arquivo `out-ala_all.xyz` com o VMD

order.in

```
ljname = ala-ok.txt  
inname = out-ala.xyz  
nmol = 1  
dens = 0.00  
cm = -1  
freeze = 10 5 1  
flexible = yes  
irdf = no  
molprint = 1 0  
printconfig = no  
printformat = 3  
printdummy = no  
printinterval = 1  
topfile = top.txt  
angle = no  
hbond = no  
$end
```



anlgeom.in

```
format = 3  
nametop = ala.dfr  
ljname = ala-ok.txt  
nametraj = out-  
ala_all.xyz  
moltype = 1  
nmol = 1  
natom = 13  
6  
1  
1  
1  
6  
1  
7  
1  
1  
6  
8  
1  
8  
$end
```

Análise de clustering usando o script clustering_traj.py:

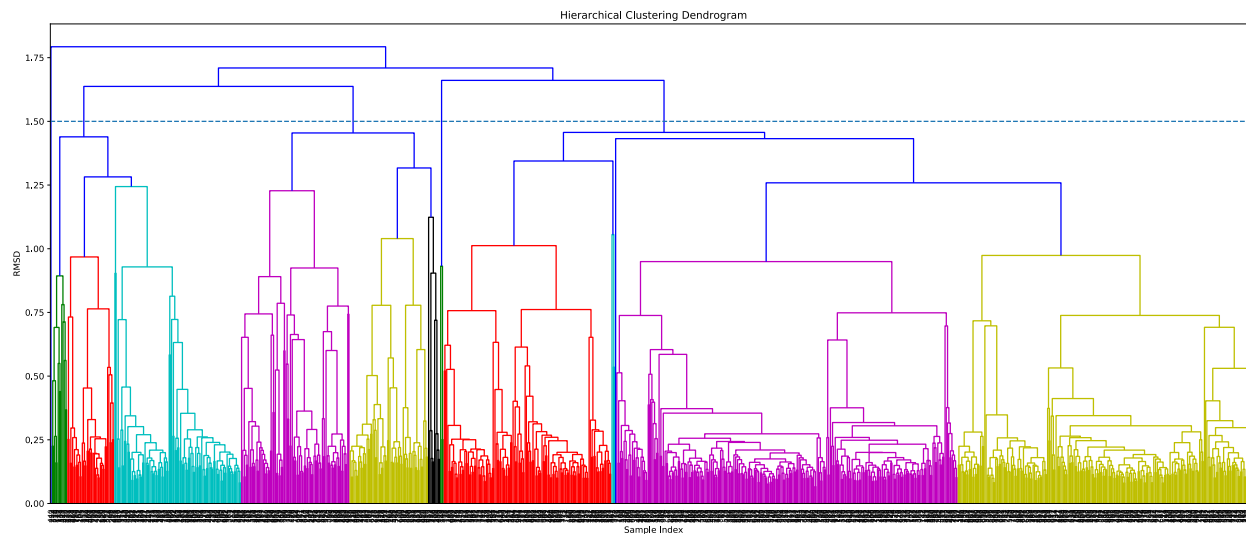
```
$ python ~/Softwares/dicetools/clustering_traj.py -p -np 2 -cc xyz  
out-ala.xyz 1.0
```

Neste caso estamos usando 2 processadores (-np 2) para executar o script, o formato do arquivo de entrada é o xyz (-cc xyz) e o critério de similaridade é de 1.0 Å (valor típico da literatura), porém pode não ser adequado para a total as moléculas. A sugestão para esta análise não ser muito demorada é usar cerca de 1 a 2 mil configurações. Então, use a palavra-chave do ORDER printinterval para reduzir o arquivo com as configurações, caso necessário.

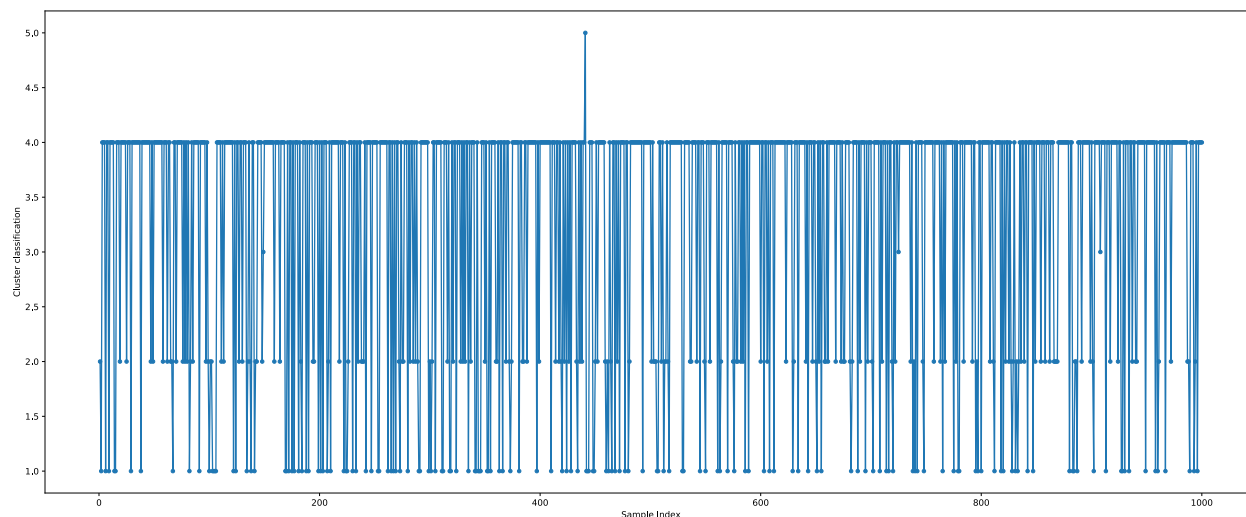
Re-analisar os tipos de clusters considerando um critério de 1.5 Å que parece apresentar uma quantidade razoável de clusters. É necessário inicialmente apagar os arquivos com as configurações dos clusters anteriores, o arquivo com informações dos clusters (cluster.dat) e manter o mapa de distâncias anteriormente calculados (distmat.dat):

```
$ rm clusters_confs_*.xyz  
$ rm clusters.dat  
$ python ~/Softwares/dicetools/clustering_traj.py -p -np 2 -cc xyz  
-i distmat.dat out-ala.xyz 1.5
```

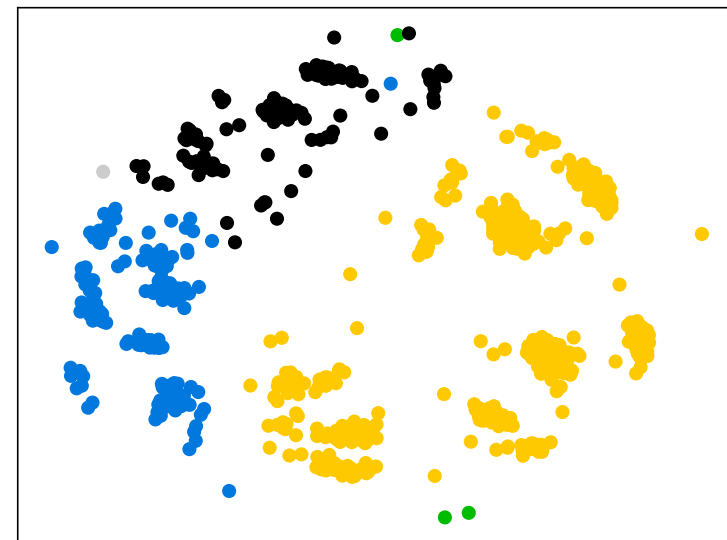
Visualizar o arquivo clusters_dendrogram.pdf, clusters_evo.pdf e cluster.pdf



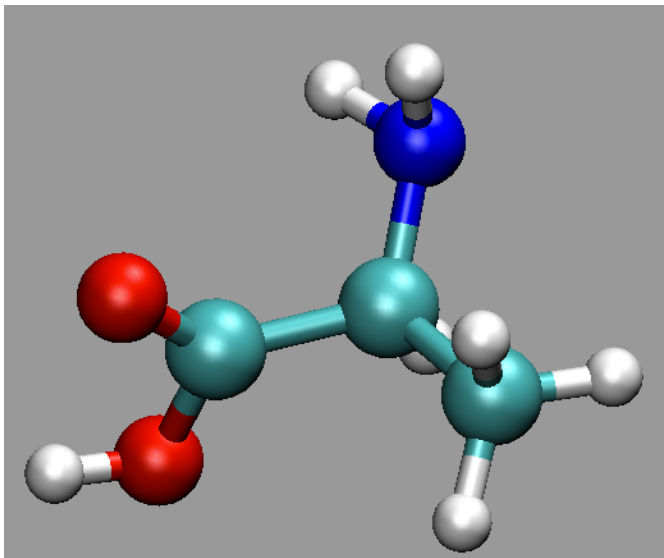
Com o critério de $1.5\text{\AA}$ existem 5 clusters.



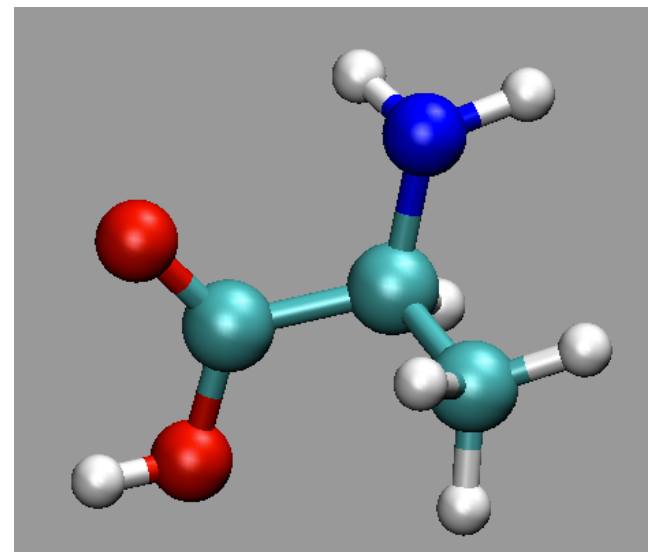
A distribuição de mudança de um cluster para outro é uniforme.



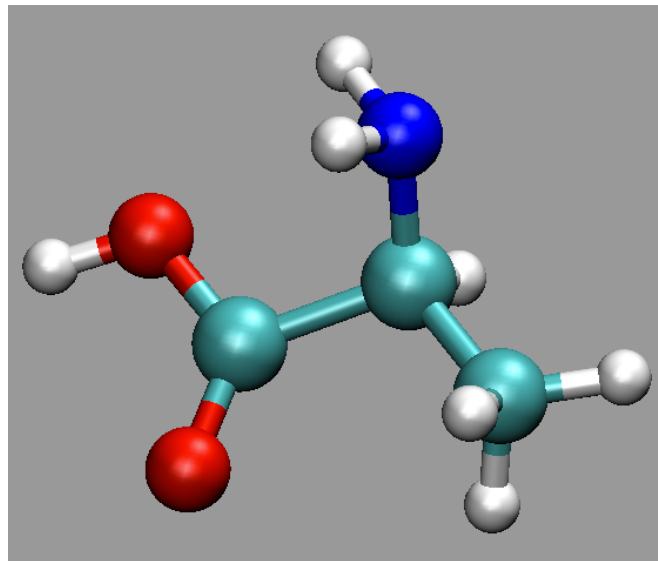
Pelo mapa de cores podemos identificar que apenas 3 clusters contribuem mais.



Cluster 4: 67.4%



Cluster 2: 16.5%



Cluster 1: 15.7%