

# 分子动力学软件GROMACS 的使用

# GROMACS

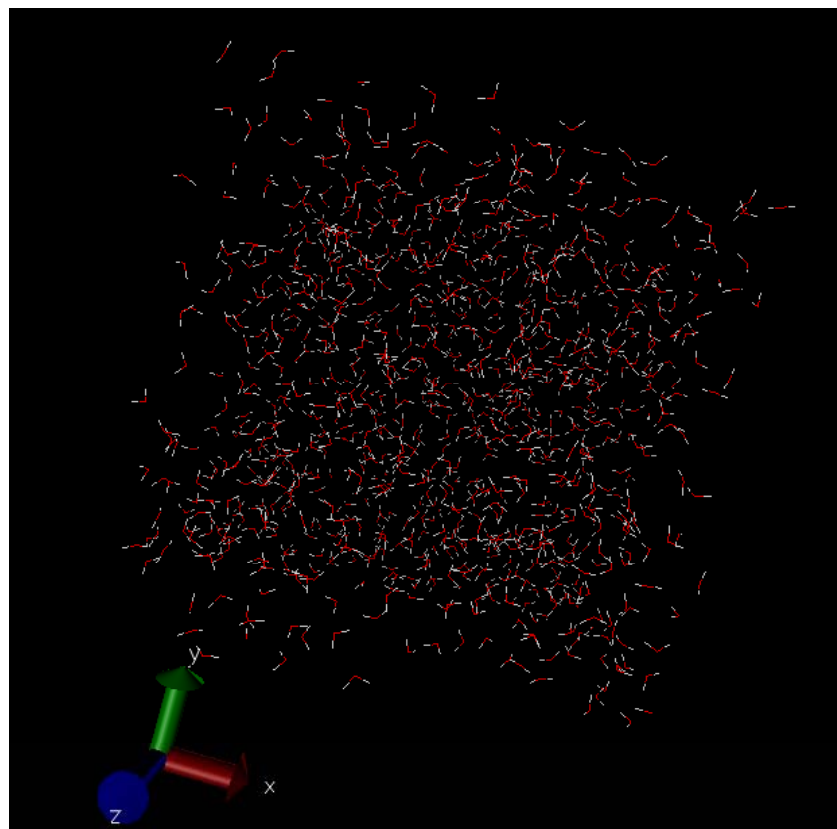
*Groningen Machine for Chemical Simulations*

GROningen MAchine for  
Chemical  
Simulations (GROMACS),  
是一款免费、开源的分子  
模拟软件，由格罗宁根大  
学生物化学系开发。界面  
友好，算法效率高。主要  
用于模拟生物体系，也可  
以模拟一般化学体系。



# 分子动力学软件GROMACS 的使用

## 示例1 液态水的模拟



请同学们先将/scratch/shczhang/ucas\_md文件夹拷贝至自己的家目录下  
用ftp将其中的两个后缀为.sh的文件下载至windows

```
cp -r ~shczhang/ucas-exp-2021-spring/ucas_md/* .
```

# 1. 构建水盒子

首先我们需要得到要模拟的分子结构，并建立盒子（gro文件）

坐标文件 filename.gro 示例

```
MD of 2 waters, t= 0.0 title
6
1WATER OW1 1 0.126 1.624 1.679 0.1227 -0.0580 0.0434
1WATER HW2 2 0.190 1.661 1.747 0.8085 0.3191 -0.7791
1WATER HW3 3 0.177 1.568 1.613 -0.9045 -2.6469 1.3180
2WATER OW1 4 1.275 0.053 0.622 0.2519 0.3140 -0.1734
2WATER HW2 5 1.337 0.002 0.680 -1.0641 -1.1349 0.0257
2WATER HW3 6 1.326 0.120 0.568 1.9427 -0.8216 -0.0244
1.82060 1.82060 1.82060
```

残基编号 残基名称 原子名称 原子编号(各5个字符) 坐标 速度(各8个字符)

盒子大小的选择 根据密度估算

## 1. 构建水盒子

- a) 用gaussview 得到一个水分子的pdb文件
- b) 使用packmol工具产生大量水分子的pdb文件

```
$ packmol < water.inp
```

- c) 使用editconf命令将pdb文件转化为gro文件

```
$ editconf -f water_model.pdb -o water.gro
```

## 2. 构建拓扑文件

结构文件给出了各个原子的位置和速度信息，要确定体系的势能函数，还要知道势能函数的形式和参数（力场）以及不同原子的归属，这些都包含在拓扑文件中。

构建合适的拓扑文件对于模拟的成败非常关键

首先要选择合适的力场——正确描述粒子间相互作用的方程，以及方程的参数

常用力场

amber

charm

opls-aa

gromos

## 2. 构建拓扑文件

top文件示例

```
; The force field files to be included
```

```
#include "rt41c5.itp" 力场文件
```

```
[ moleculetype ]
```

```
; name nrexcl
```

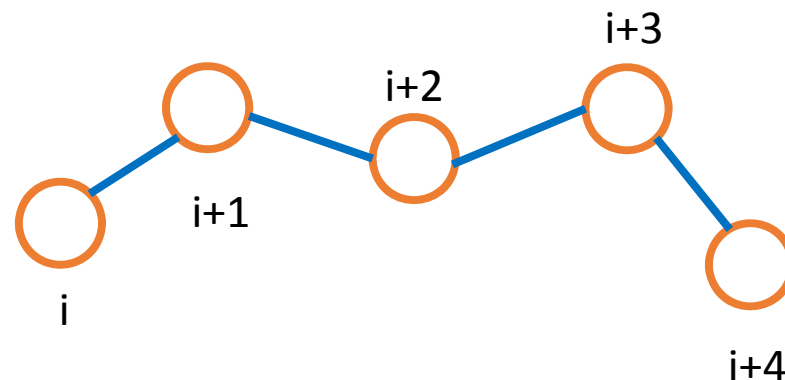
```
Urea          3  计及成键作用的  
              原子间隔个数
```

```
[ atomtypes ]
```

```
; name  at.num  mass      charge  ptype
```

O	8	15.99940	0.000	A
OM	8	15.99940	0.000	A

```
.....
```



	V(c6)	W(c12)
A	0.22617E-02	0.74158E-06
A	0.22617E-02	0.74158E-06

LJ势参数

## 2. 构建拓扑文件

```
[ bonds ]
: ai aj funct c0 c1
  3 4 1 1.000000e-01 3.744680e+05
  3 5 1 1.000000e-01 3.744680e+05
  6 7 1 1.000000e-01 3.744680e+05
  6 8 1 1.000000e-01 3.744680e+05
  1 2 1 1.230000e-01 5.020800e+05
  1 3 1 1.330000e-01 3.765600e+05
  1 6 1 1.330000e-01 3.765600e+05
```

```
[ angles ]
: ai aj ak funct c0 c1
  1 3 4 1 1.200000e+02 2.928800e+02
  1 3 5 1 1.200000e+02 2.928800e+02
  4 3 5 1 1.200000e+02 3.347200e+02
  1 6 7 1 1.200000e+02 2.928800e+02
  1 6 8 1 1.200000e+02 2.928800e+02
  7 6 8 1 1.200000e+02 3.347200e+02
  2 1 3 1 1.215000e+02 5.020800e+02
  2 1 6 1 1.215000e+02 5.020800e+02
  3 1 6 1 1.170000e+02 5.020800e+02
```

```
[ dihedrals ]
: ai aj ak al funct c0 c1 c2
  2 1 3 4 1 1.800000e+02 3.347200e+01 2.000000e+00
  6 1 3 4 1 1.800000e+02 3.347200e+01 2.000000e+00
  2 1 3 5 1 1.800000e+02 3.347200e+01 2.000000e+00
  6 1 3 5 1 1.800000e+02 3.347200e+01 2.000000e+00
  2 1 6 7 1 1.800000e+02 3.347200e+01 2.000000e+00
  3 1 6 7 1 1.800000e+02 3.347200e+01 2.000000e+00
  2 1 6 8 1 1.800000e+02 3.347200e+01 2.000000e+00
  3 1 6 8 1 1.800000e+02 3.347200e+01 2.000000e+00
```



## 2. 构建拓扑文件

```
[ pairs ]
: ai    aj funct          c0          c1
  2     4     1 0.000000e+00 0.000000e+00
  2     5     1 0.000000e+00 0.000000e+00
  2     7     1 0.000000e+00 0.000000e+00
  2     8     1 0.000000e+00 0.000000e+00
  3     7     1 0.000000e+00 0.000000e+00
  3     8     1 0.000000e+00 0.000000e+00
  4     6     1 0.000000e+00 0.000000e+00
  5     6     1 0.000000e+00 0.000000e+00
```

1,4 相互作用, 特殊的非键相互作用

```
[ system ]
: Name
CYCLOHEXANE

[ molecules ]
: Compound      #mols
MOL              1024
```

分子个数

## 2. 构建拓扑文件

水的拓扑文件，直接加载itp文件

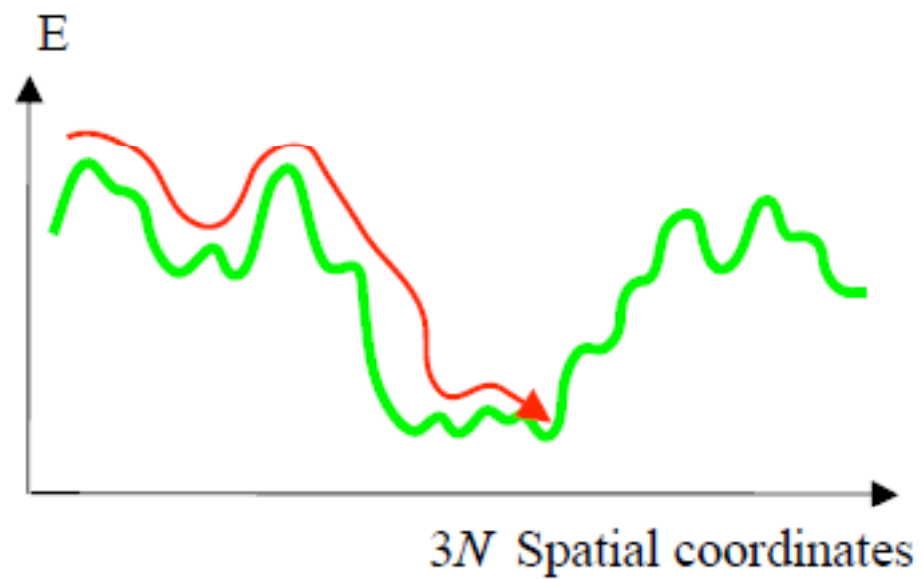
```
1 ;topology for water using spce
2 #include "/share/apps/gromacs/4.6.5/share/gromacs/top/oplsaa.ff/forcefield.itp"
3 #include "/share/apps/gromacs/4.6.5/share/gromacs/top/oplsaa.ff/spce.itp"
4
5 [system]
6 WATER
7
8 [molecules]
9 ; Compound      #mols
10 SOL             1024
```

spce模型

我们已经完成了坐标文件和拓扑文件的构建 **water.gro**    **water.top**

### 3. 能量最小化

目的 产生一个较合理的初始结构，如果体系的初始结构离平衡态太远会导致受力过大使得模拟失败



### 3. 能量最小化

mdp文件

给定模拟参数，grompp的输入文件

```
runcontrol ; minim.mdp - used as input into grompp to generate em.tpr
           ; Parameters describing what to do, when to stop and what to save
integrator = steep ; Algorithm (steep = steepest descent minimization)
emtol = 1000.0 ; Stop minimization when the maximum force < 1000.0 kJ/mol/rm
outputcontrol
emstep = 0.01 ; Energy step size
nsteps = 50000 ; Maximum number of (minimization) steps to perform

neighbor ; Parameters describing how to find the neighbors of each atom and how to calculate the interactions
searching
nstlist = 1 ; Frequency to update the neighbor list and long range forces
ns_type = grid ; Method to determine neighbor list (simple, grid)
rlist = 1.0 ; Cut-off for making neighbor list (short range forces)
coulombtype = PME ; Treatment of long range electrostatic interactions
vdw
rcoulomb = 1.0 ; Short-range electrostatic cut-off
rvdw = 1.0 ; Short-range Van der Waals cut-off
electrostatics
pbc = xyz ; Periodic Boundary Conditions (yes/no)
```

### 3. 能量最小化

grompp 预处理命令 (gromacs preprocessor)

-f .mdp 输入文件  
-c .gro 坐标文件  
-p .top 拓扑文件  
-o .tpr 输出文件, 也是下一步  
mdrun 的输入文件

```
$ grompp -f min.mdp -c water.gro -p water.top -o em.tpr
```

```
$ nohup mdrun -nt 8 -deffnm em &
```

注意mdrun命令必须在分节点上运行

### 3. 能量最小化

最后一步的能量值

writing lowest energy coordinates.

Steepest Descents converged to Fmax < 1000 in 26 steps

Potential Energy	=	-2.9616607e+04
Maximum force	=	4.7964157e+02 on atom 19
Norm of force	=	7.2915077e+01

### 3. 能量最小化

mdrun 的输出文件

em.log

em.edr 二进制能量文件

em.trr 二进制轨迹文件

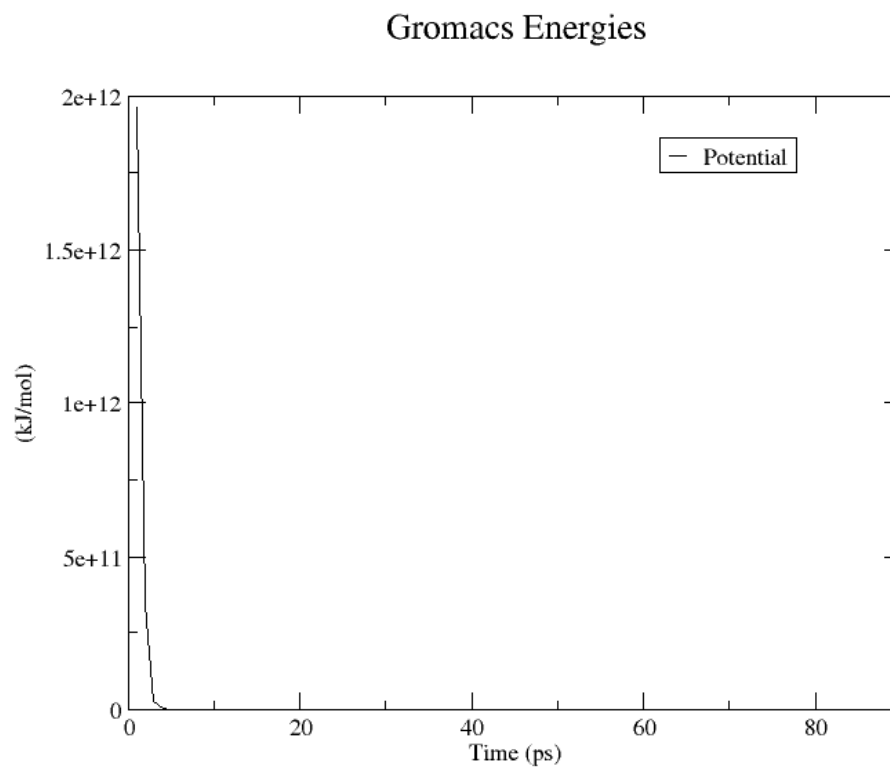
em.gro 最后一步的坐标文件

g\_energy 命令分析二进制能量文件

```
$ g_energy -f em.edr -o potential .xvg
```

### 3. 能量最小化

使用xmgrace 查看输出结果 可以看到每一步能量的变化过程



产生的结果用xmgrace查看并保存图片



## 4. NVT平衡

能量最小化完成后，进行NVT系综模拟

mdp 文件

temperature coupling

```
title = Yo
cpp = /lib/cpp
include = -I../top
define =
integrator = md
dt = 0.002
nsteps = 500000
nstxout = 5000
nstvout = 5000
nstlog = 5000
nstenergy = 250
nstxtcout = 250
xtc-grps = Protein
energygrps = Protein SOL
nstlist = 10
ns-type = grid
rlist = 0.8
coulombtype = cut-off
rcoulomb = 1.4
rvdw = 0.8
tcoupl = Berendsen
tc-grps = Protein SOL
tau-t = 0.1 0.1
ref-t = 300 300
Pcoupl = Berendsen
tau-p = 1.0
compressibility = 4.5e-5
ref-p = 1.0
gen-vel = yes
gen-temp = 300
gen-seed = 173529
constraints = all-bonds
```

## 4. NVT平衡

同上步，运行grompp 和 mdrun

```
$ grompp -f nvt.mdp -c em.gro -p water.top -o nvt.tpr
```

```
$ nohup mdrun -nt 8 -deffnm nvt &
```

用g\_energy命令处理二进制能量文件

```
$ g_energy -f nvt.edr -o temp.xvg
```

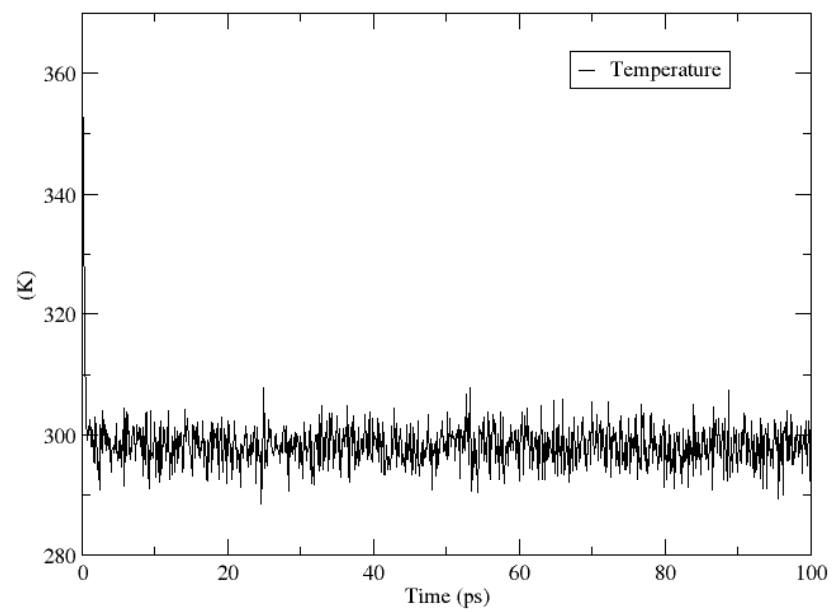
选择temperature

```
$ g_energy -f nvt.edr -o potential.xvg
```

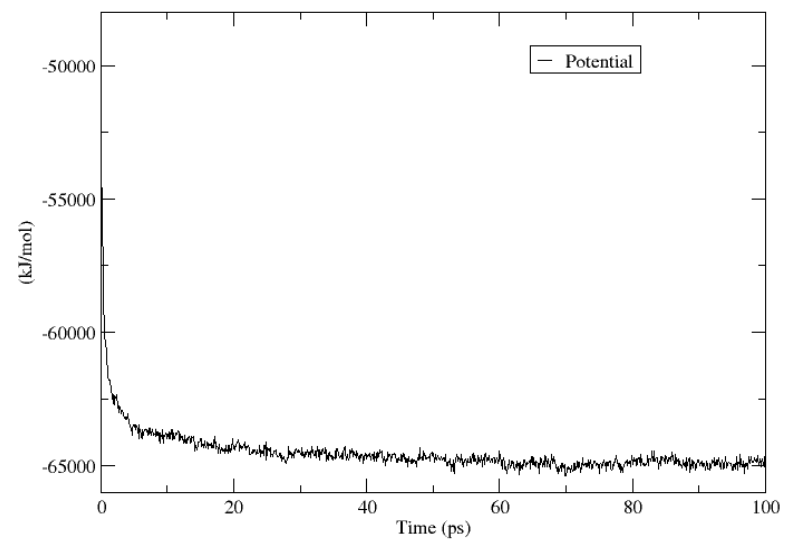
选择potential

## 4. NVT平衡

Gromacs Energies



Gromacs Energies



## 4. NPT平衡

pressure coupling

```
title = Yo
cpp = /lib/cpp
include = -I../top
define =
integrator = md
dt = 0.002
nsteps = 500000
nstxout = 5000
nstvout = 5000
nstlog = 5000
nstenergy = 250
nstxtcout = 250
xtc-grps = Protein
energygrps = Protein SOL
nstlist = 10
ns-type = grid
rlist = 0.8
coulombtype = cut-off
rcoulomb = 1.4
rvdw = 0.8
tcoupl = Berendsen
tc-grps = Protein SOL
tau-t = 0.1 0.1
ref-t = 300 300
Pcoupl = Berendsen
tau-p = 1.0
compressibility = 4.5e-5
ref-p = 1.0
gen-vel = yes
gen-temp = 300
gen-seed = 173529
constraints = all-bonds
```

## 4. NPT平衡

同上步，运行grompp 和 mdrun

```
$ grompp -f npt.mdp -c nvt.gro -p water.top -o npt.tpr
```

```
$ mdrun -nt 8 -deffnm npt
```

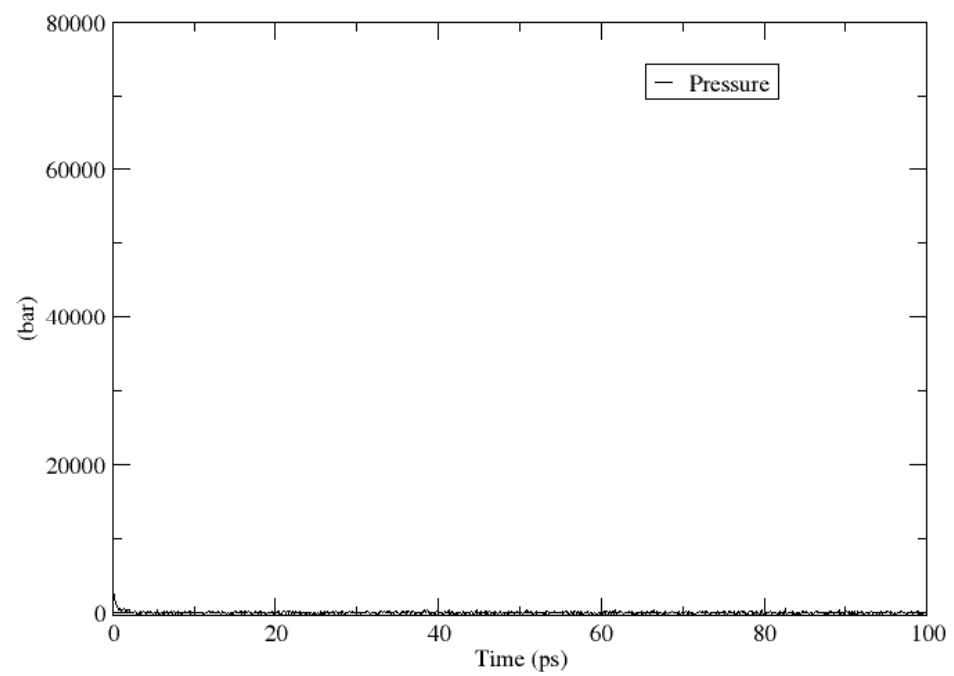
用g\_energy命令处理二进制能量文件

```
$ g_energy -f nvt.edr -o pressure.svg
```

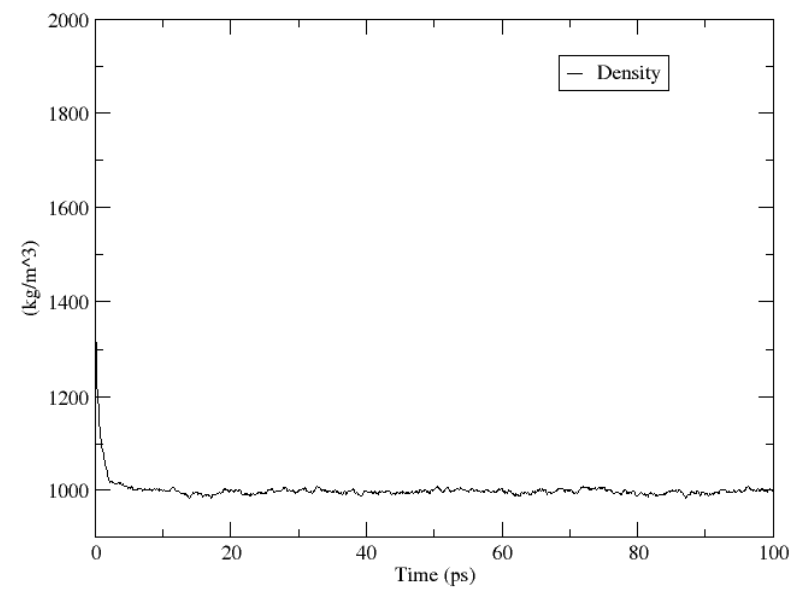
```
$ g_energy -f nvt.edr -o density.svg
```

## 4. NPT平衡

Gromacs Energies



Gromacs Energies



## 4. NPT平衡

用刚才得到的平衡npt构型继续跑2ns

```
$ grompp -f mdrun.mdp -c npt.gro -p water.top -o mdrun.tpr
```

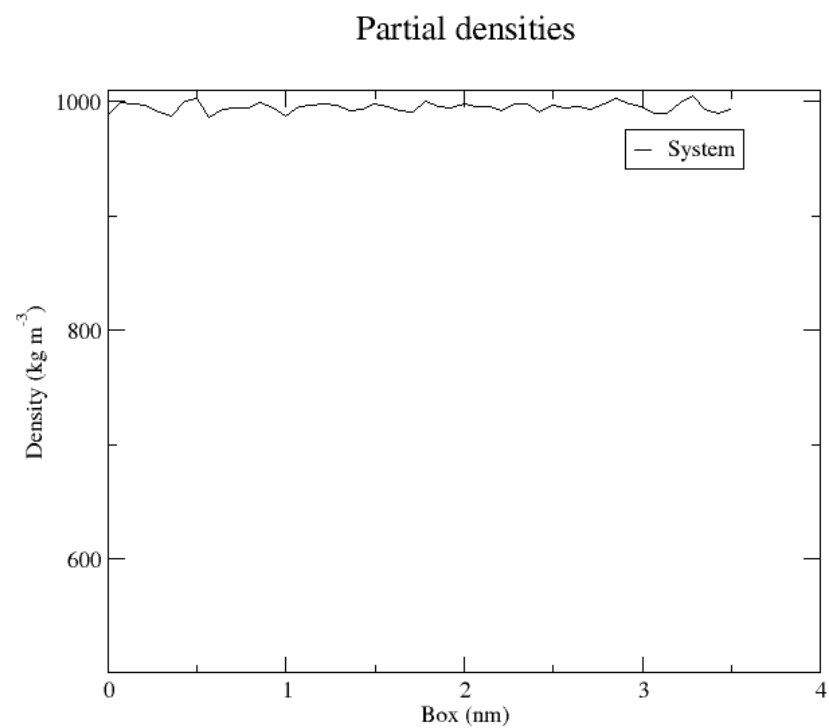
```
$ nohup mdrun -nt 8 -deffnm mdrun &
```

## 4. NPT平衡

结果分析

密度

```
$ g_density -f mdrun.trr -s mdrun.tpr -o density.xvg
```



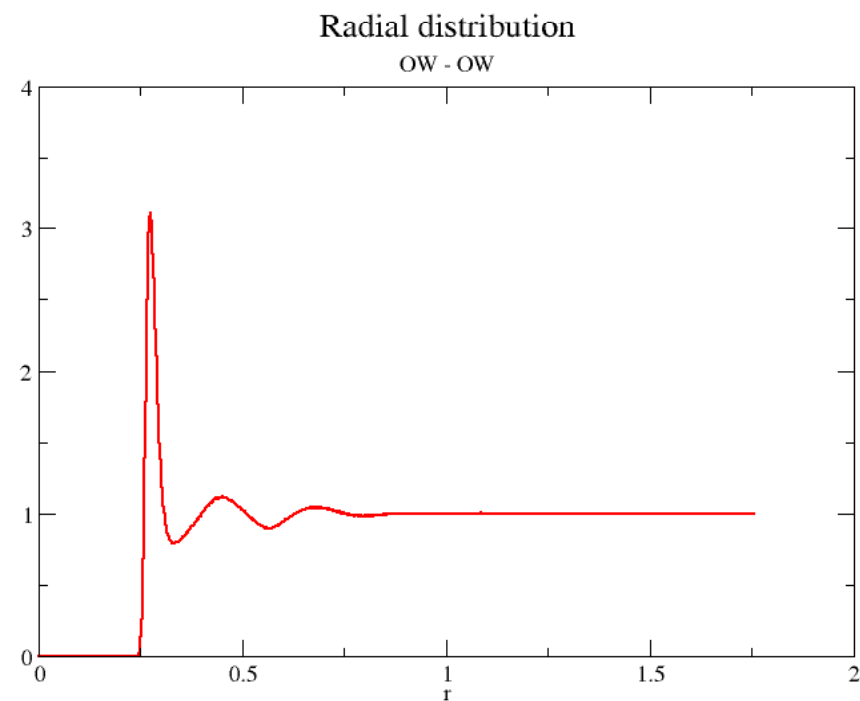


## 4. NPT平衡

结果分析

Radial distribution function

```
$ g_rdf -f mdrun.trr -s mdrun.tpr -n index.ndx -o rdf.xvg
```

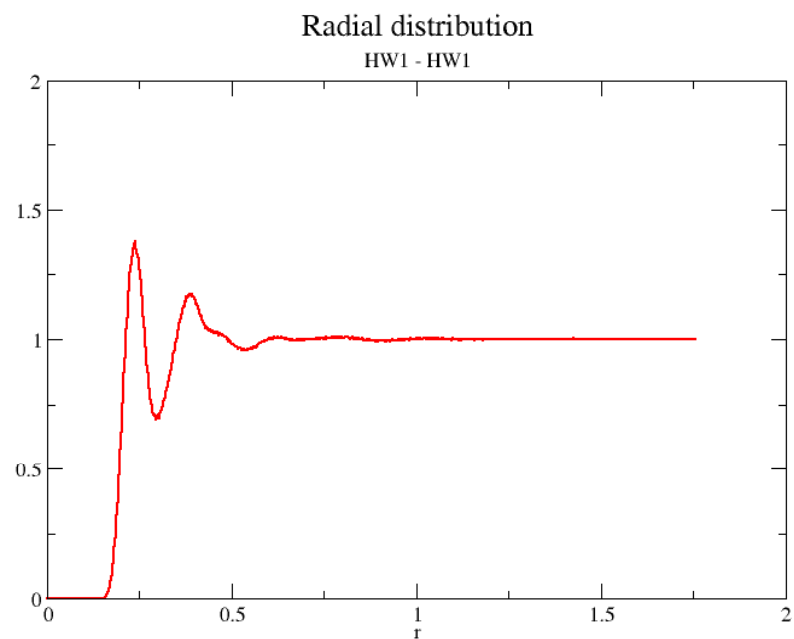


O-O径向分布函数

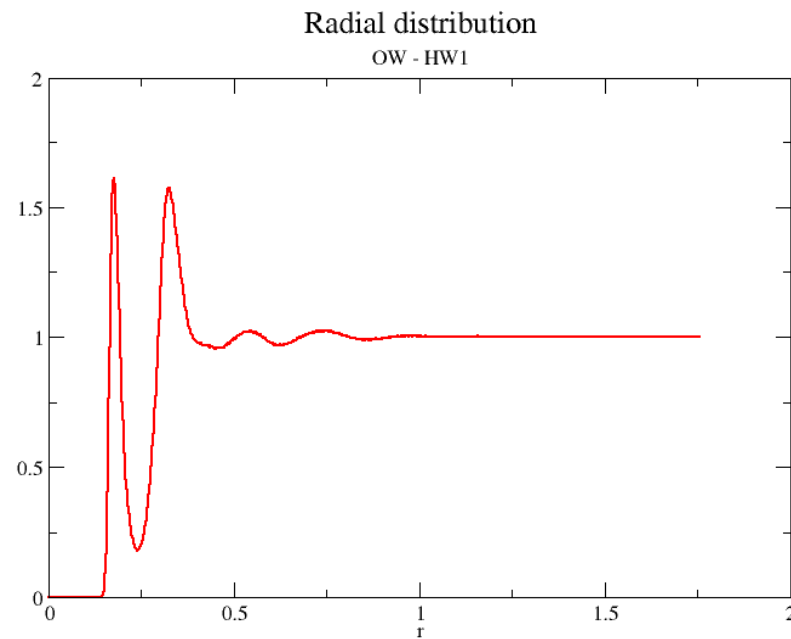
## 4. NPT平衡

结果分析

Radial distribution function



H-H 径向分布函数

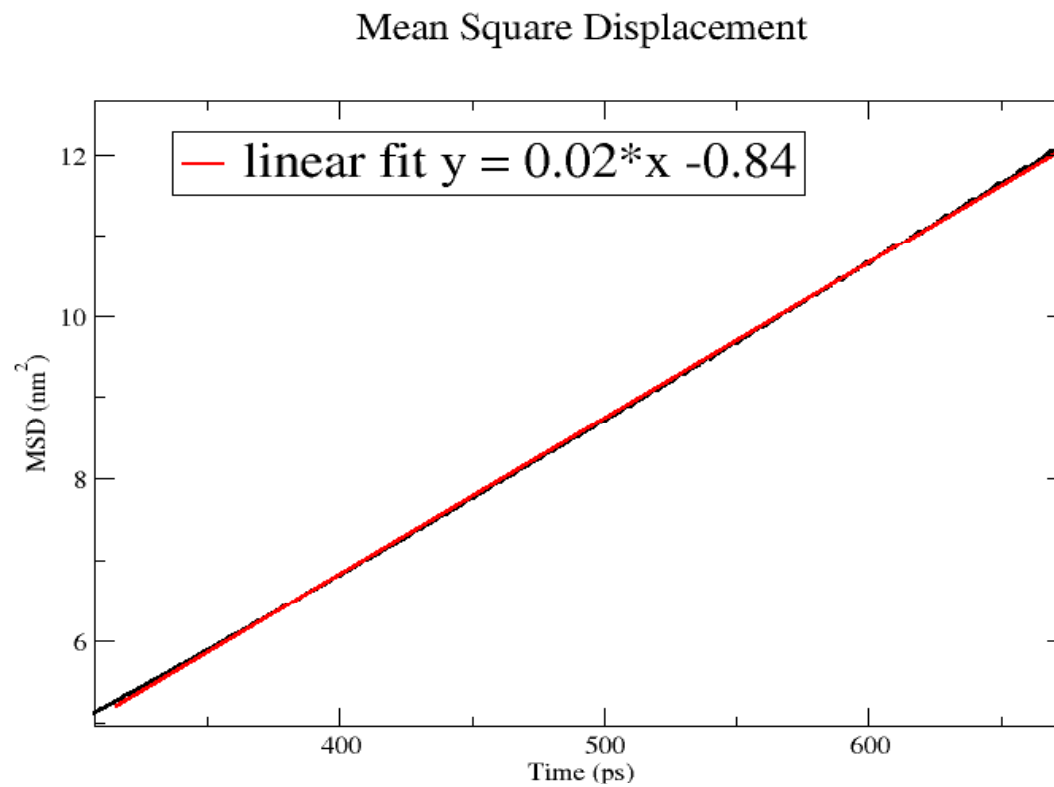


O-H 径向分布函数

## 4. NPT平衡

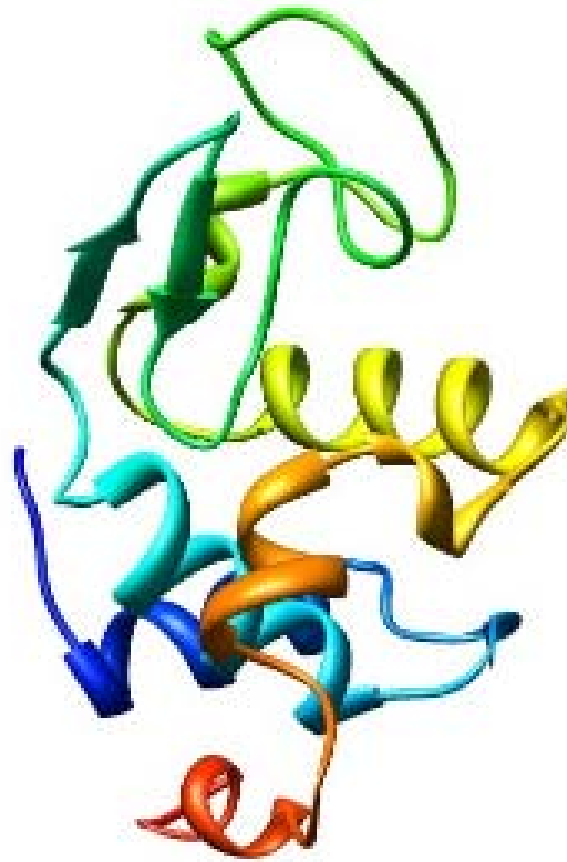
Mean square displacement

```
$ g_msd -f mdrun.trr -s mdrun.tpr -o msd.xvg
```



# 分子动力学软件Gromacs 的使用

示例2 水环境中溶菌酶



## 1. 构建结构文件

下载蛋白质结构文件.PDB

删除水分子

对生物分子可使用pdb2gmx命令生成gro文件和top文件

首先将pdb文件转换为gro文件

```
$ pdb2gmx -f 1AKI.pdb -o 1AKI_processed.gro -water spce
```

选择15 opls\_aa力场

会得到一个gro文件和一个top文件

## 1. 构建结构文件

我们还需要对刚才生成的gro文件定义盒子并添加溶剂水

使用editconf 命令及 genbox 命令

```
$ editconf -f 1AKI_processed.gro -o 1AKI_newbox.gro -c -d 1.0 -bt cubic
```

```
$ genbox -cp 1AKI_newbox.gro -cs spc216.gro -o 1AKI_solv.gro -p topol.top
```

## 1. 构建结构文件

蛋白质带有净电荷，需要添加粒子时期呈中性

可通过genion 命令完成

首先需要使用grompp 生成genion的输入文件

```
$ grompp -f ions.mdp -c 1AKI_solv.gro -p topol.top -o ions.tpr
```

使用genion命令

```
$ genion -s ions.tpr -o 1AKI_solv_ions.gro -p topol.top -pname NA -nname CL -nn 8
```

离子名称

负离子  
数目

现在我们已经完成了蛋白质体系的坐标文件和拓扑文件的构建，接下来的步骤和液态水模拟的步骤一致

## 2. 能量优化

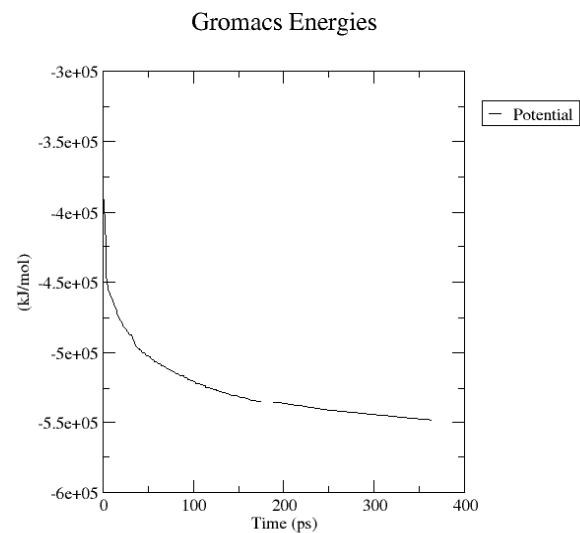
同液态水一样我们首先要进行能量优化

```
$grompp -f minim.mdp -c 1AKI_solv_ions.gro -p topol.top -o em.tpr
```

```
$mdrun -v -deffnm em
```

注意mdrun命令必须在分节点上运行

```
$g_energy -f em.edr -o potential.xvg
```



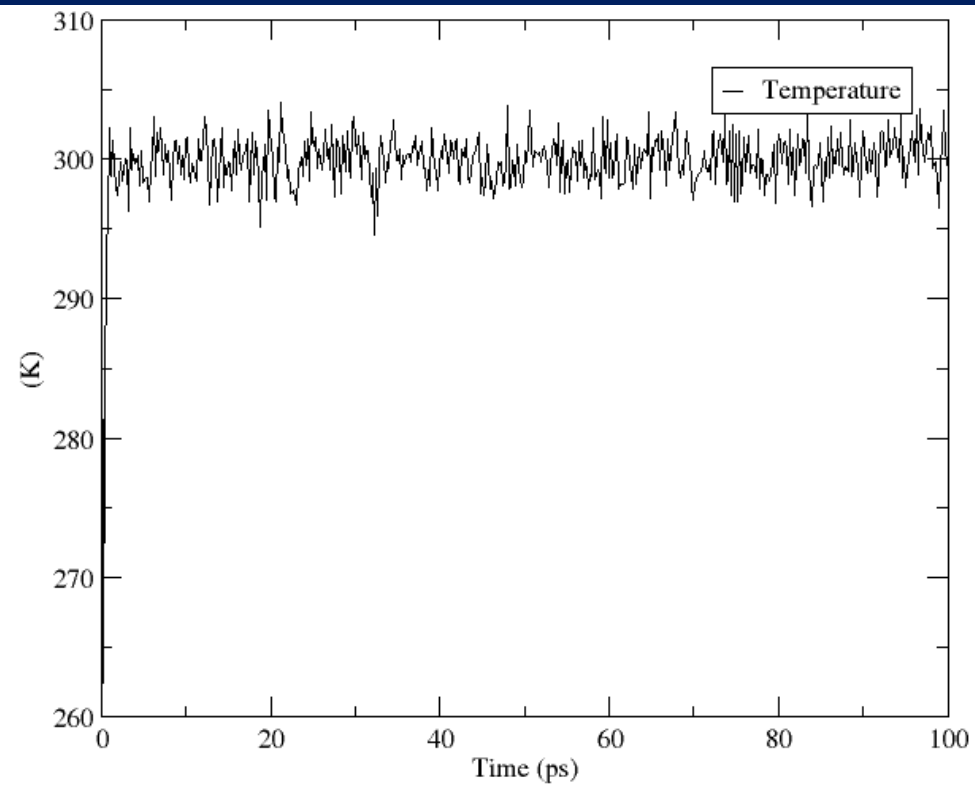


### 3. NVT平衡

```
$ grompp -f nvt.mdp -c em.gro -p topol.top -o nvt.tpr
```

```
$ mdrun -deffnm nvt
```

```
$ g_energy -f nvt.edr
```

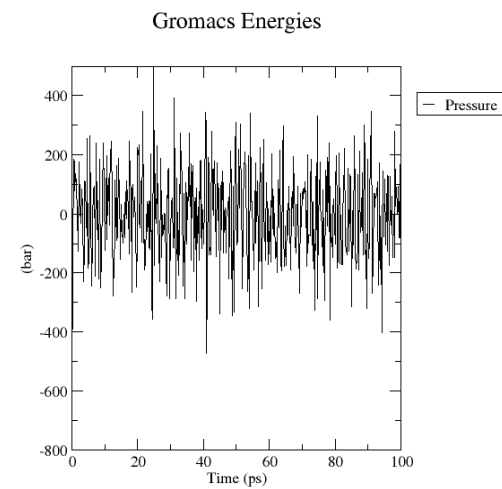


## 4. NPT平衡

```
$ grompp -f npt.mdp -c nvt.gro -t nvt.cpt -p topol.top -o npt.tpr
```

```
$ mdrun -deffnm npt
```

```
$ g_energy -f npt.edr -o pressure.xvg
```



## 5. md Run

在NPT 下跑1ns

```
$ grompp -f md.mdp -c npt.gro -t npt.cpt -p topol.top -o md_0_1.tpr
```

```
$ mdrun -deffnm md_0_1
```

## 6. 结果分析

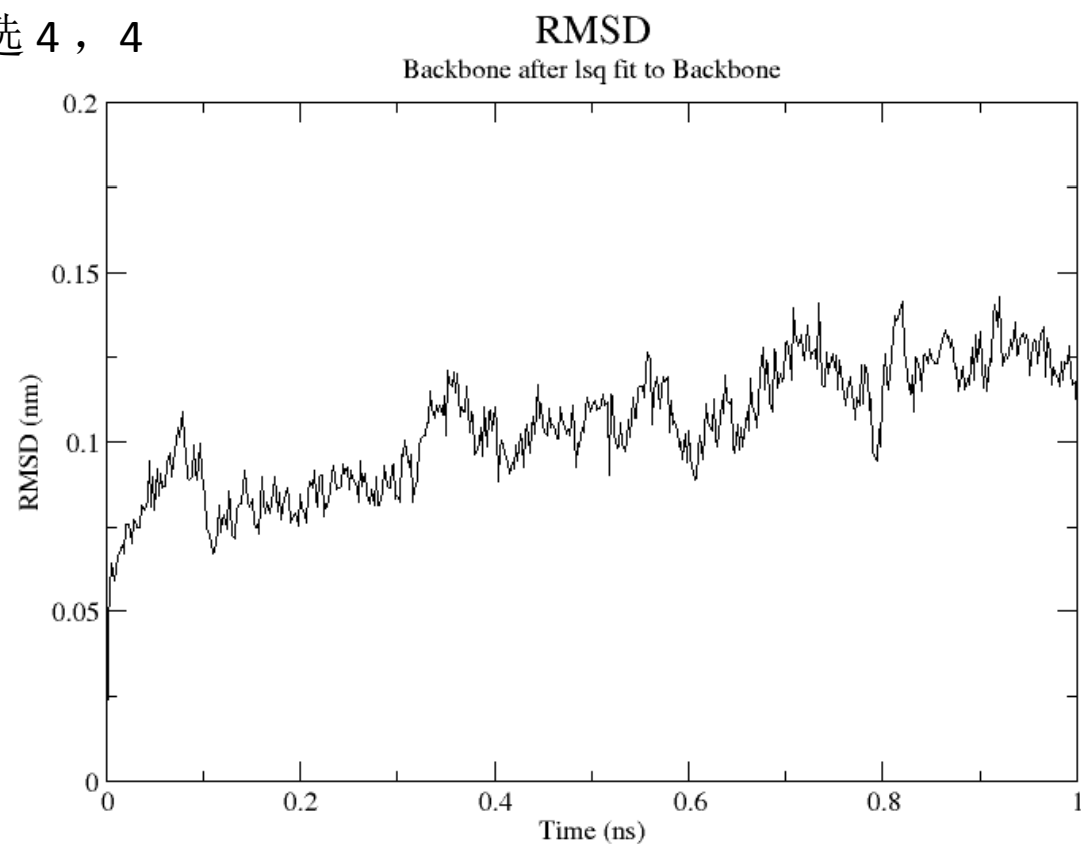
RMSD

```
$ trjconv -s md_0_1.tpr -f md_0_1.xtc -o md_0_1_noPBC.xtc -pbc mol -ur compact
```

去除周期性 选0system

```
$ g_rms -s md_0_1.tpr -f md_0_1_noPBC.xtc -o rmsd.xvg -tu ns
```

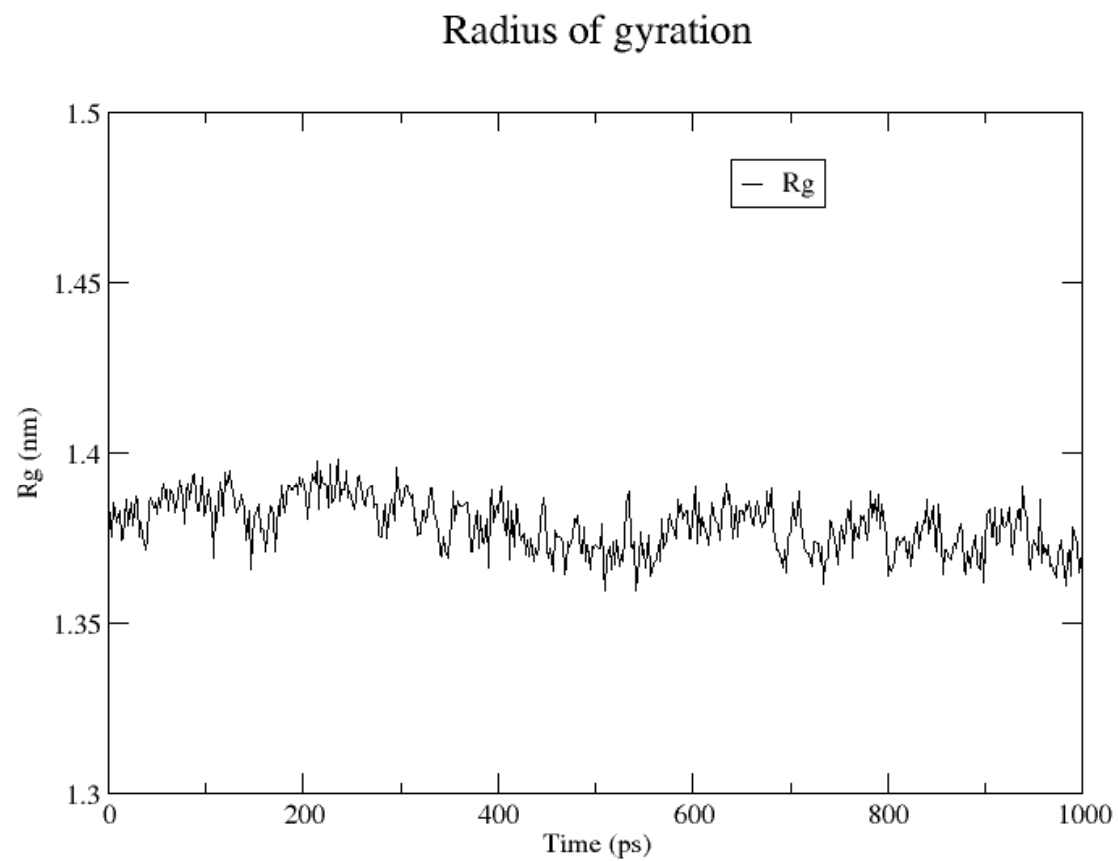
产生RMSD, 选 4, 4



## 6. 结果分析

回转半径

```
$ g_gyrate -s md_0_1.tpr -f md_0_1_noPBC.xtc -o gyrate.svg
```



## 7. 结果分析

利用 vmd 观看蛋白质动态变化

打开vmd，点击file newmolecule  
在browser窗口里加载1aki\_solve\_ion.gro点击load  
在同一个窗口里加载md\_0\_1.trr轨迹文件  
在vmd窗口点击graphics菜单选择representation选项  
在select框里输入protein按enter键  
drawingmethod 选择newcartoon  
coloringmethod选择resname

## 练习

构建一个5000个水分子的盒子，用相同的mdp文件进行md模拟。分析结果（密度，RDF，RMSD），与1500个分子的结果进行比较。