

Lysozyme in Water

Naf Guo
2025

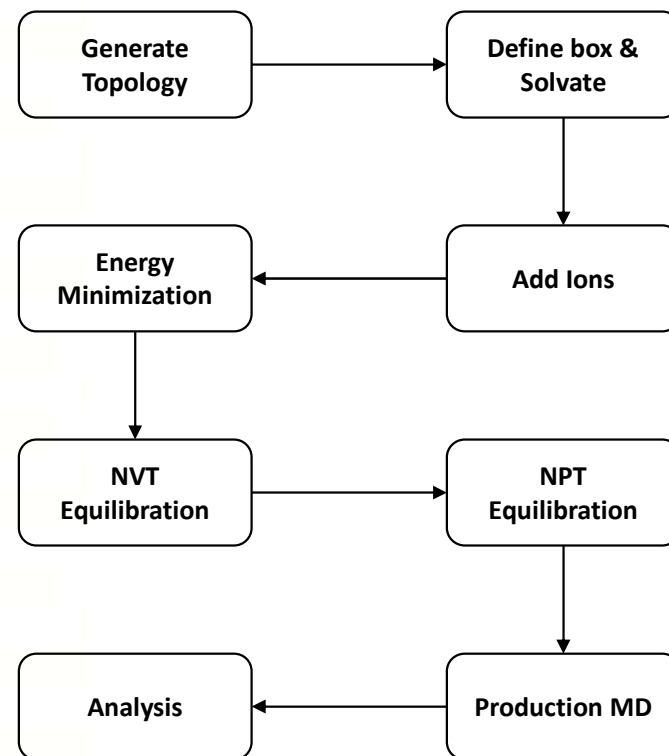
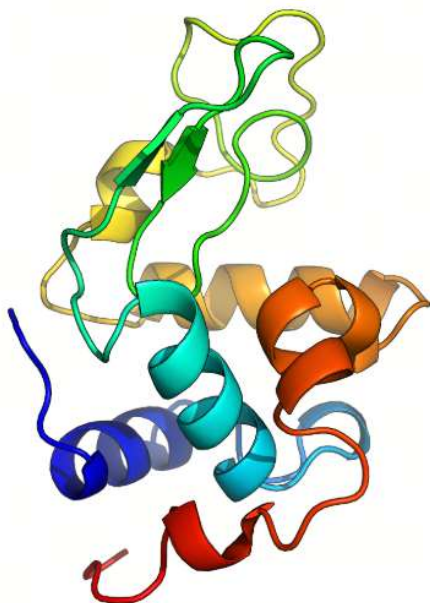
Lysozyme in Water

<http://www.mdtutorials.com/gmx/lysozyme/index.html>

GROMACS Tutorial

Lysozyme in Water

Justin A. Lemkul, Ph.D.
Virginia Tech Department of Biochemistry

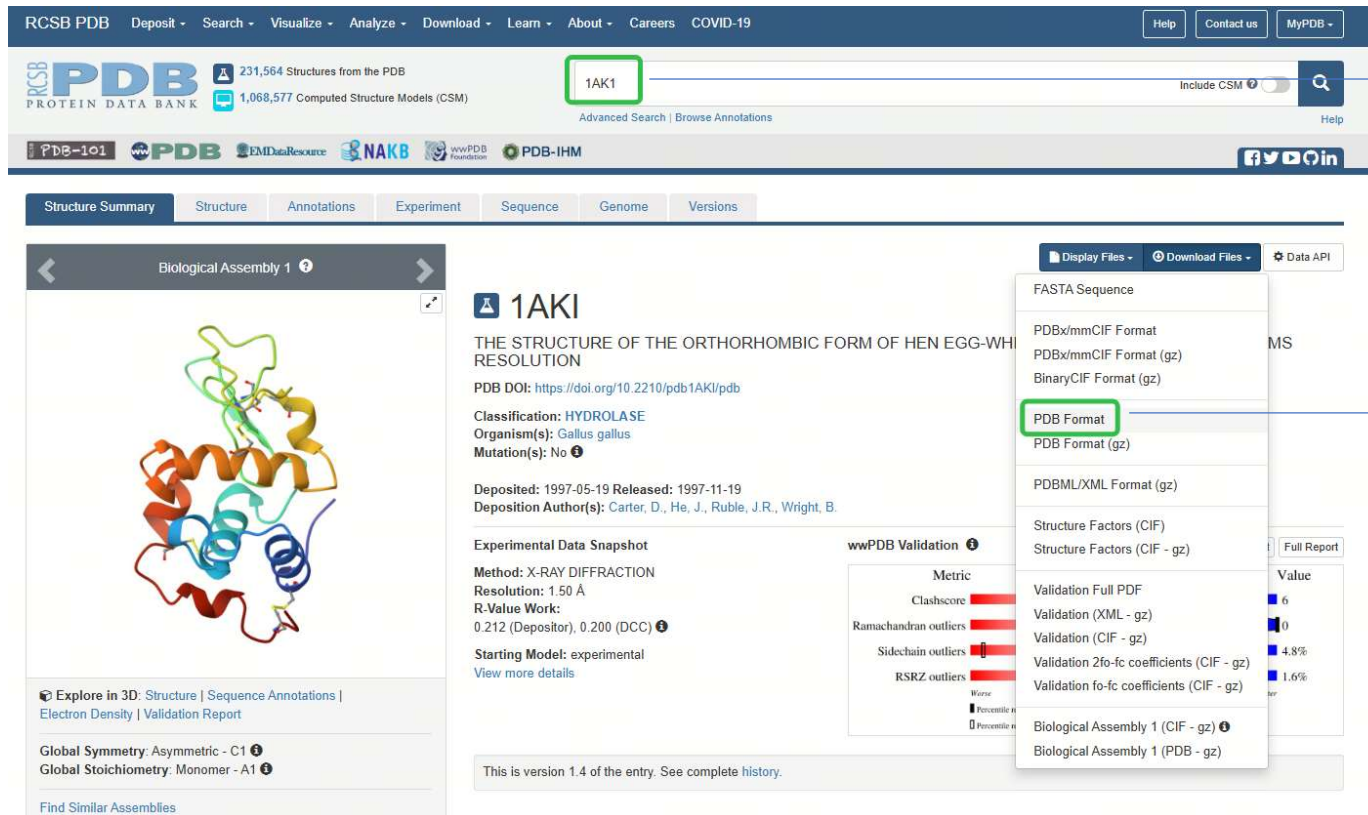


This example will guide a new user through the process of setting up a simulation system containing a protein (lysozyme) in a box of water, with ions. Each step will contain an explanation of input and output, using typical settings for general use.

Download PDB file

Lysozyme Tutorial

We must download the protein structure file with which we will be working. For this tutorial, we will utilize hen egg white lysozyme (PDB code 1AKI) Go to the **RCSB** website and download the PDB text for the crystal structure.



RCSB PDB Deposit Search Visualize Analyze Download Learn About Careers COVID-19

231,564 Structures from the PDB
1,068,577 Computed Structure Models (CSM)

1AKI

Advanced Search | Browse Annotations

Structure Summary Structure Annotations Experiment Sequence Genome Versions

Biological Assembly 1

1AKI

THE STRUCTURE OF THE ORTHORHOMBIC FORM OF HEN EGG-WHITE LYSOZYME
RESOLUTION

PDB DOI: <https://doi.org/10.2210/pdb1AKI/pdb>

Classification: **HYDROLASE**
Organism(s): **Gallus gallus**
Mutation(s): No

Deposited: 1997-05-19 Released: 1997-11-19
Deposition Author(s): Carter, D., He, J., Ruble, J.R., Wright, B.

Experimental Data Snapshot

Method: X-RAY DIFFRACTION
Resolution: 1.50 Å
R-Value Work: 0.212 (Depositor), 0.200 (DCC)

Starting Model: experimental
[View more details](#)

wwPDB Validation

Metric

Clashscore

Ramachandran outliers

Sidechain outliers

RSRZ outliers

Worse

Percentage

Display Files Download Files Data API

FASTA Sequence

PDBx/mmCIF Format

PDBx/mmCIF Format (gz)

BinaryCIF Format (gz)

PDB Format

PDB Format (gz)

PDBML/XML Format (gz)

Structure Factors (CIF)

Structure Factors (CIF - gz)

Validation Full PDF

Validation (XML - gz)

Validation (CIF - gz)

Validation 2fo-*fc* coefficients (CIF - gz)

Validation fo-*fc* coefficients (CIF - gz)

Biological Assembly 1 (CIF - gz)

Biological Assembly 1 (PDB - gz)

Global Symmetry: Asymmetric - C1
Global Stoichiometry: Monomer - A1

Find Similar Assemblies

登录PDB结构数据库
<https://www.rcsb.org/>
搜索PDB号 1ak1

下载PDB格式文件

See the PDB file

<https://www.cgl.ucsf.edu/chimerax/>

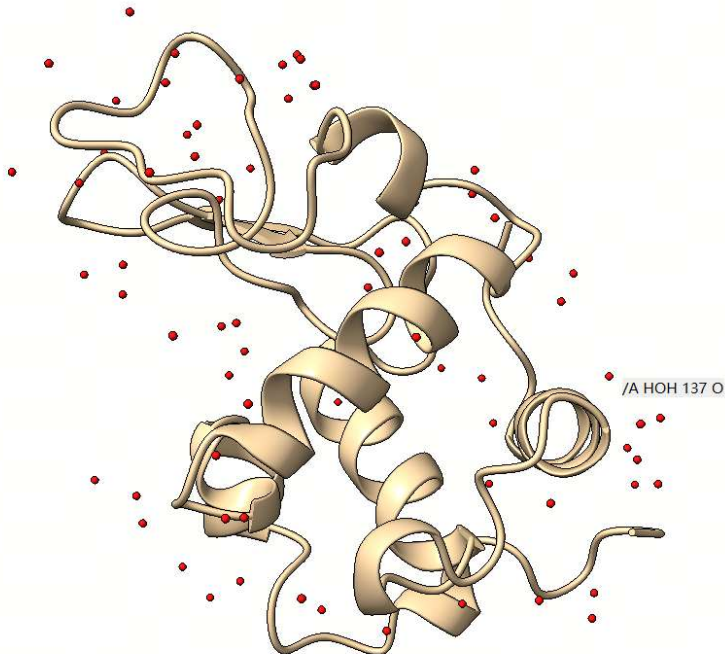
ChimeraX

File Edit Select Actions Tools Favorites Presets Help

Home Molecule Display Nucleotides Graphics Map Medical Image Markers Right Mouse

Open Recent Save Snapshot Spin movie Show Hide Show Hide Stick Sphere Ball stick White Black Simple Soft Full Inspect

File Images Atoms Cartoons Styles Background Lighting Selection



Log

[\[more info...\]](#)

Chain information for 1aki.pdb #1

Chain	Description	UniProt
A	lysozyme	LYSC_CHICK

[preset "overall look" "publication 1 \(silhouettes\)"](#)

Using preset: Overall Look / Publication 1 (Silhouettes)

Preset expands to these ChimeraX commands:

```
set bg white
graphics silhouettes t
lighting depthCue f
```

[display](#)

[select /A:81](#)

6 atoms, 5 bonds, 1 residue, 1 model selected

[select up](#)

33 atoms, 32 bonds, 5 residues, 1 model selected

[select up](#)

1001 atoms, 1025 bonds, 129 residues, 1 model selected

[hide sel atoms](#)

[select clear](#)

Models

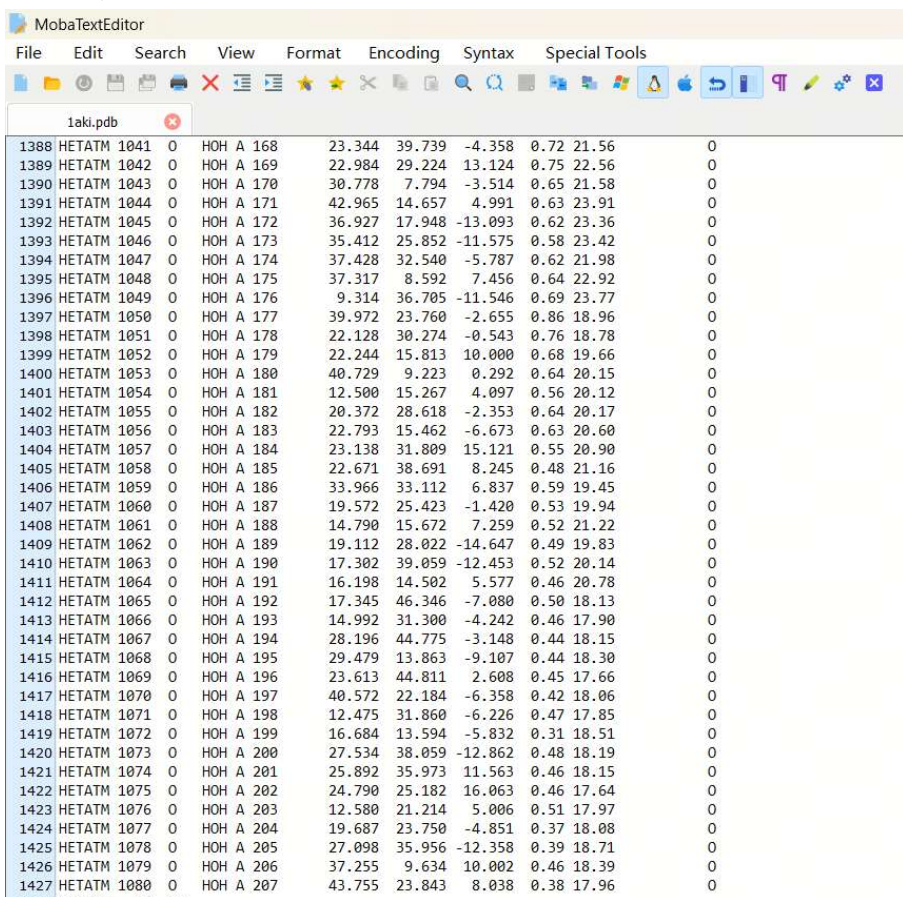
Name	ID	Shown	Select
1aki.pdb	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Close Hide Show View

Command: [disp](#)

Delete Unwanted Atoms

使用一个记事本打开PDB文件，发现最后写了很多水，这里先删除（手动删除并保存即可）



Atom	Label	Residue	Chain	X	Y	Z	Occupancy
1388	HETATM	1041	O	HOH	A	168	23.344 39.739 -4.358 0.72 21.56 0
1389	HETATM	1042	O	HOH	A	169	22.984 29.224 13.124 0.75 22.56 0
1390	HETATM	1043	O	HOH	A	170	30.778 7.794 -3.514 0.65 21.58 0
1391	HETATM	1044	O	HOH	A	171	42.965 14.657 4.991 0.63 23.91 0
1392	HETATM	1045	O	HOH	A	172	36.927 17.948 -13.093 0.62 23.36 0
1393	HETATM	1046	O	HOH	A	173	35.412 25.852 -11.575 0.58 23.42 0
1394	HETATM	1047	O	HOH	A	174	37.428 32.540 -5.787 0.62 21.98 0
1395	HETATM	1048	O	HOH	A	175	37.317 8.592 7.456 0.64 22.92 0
1396	HETATM	1049	O	HOH	A	176	9.314 36.705 -11.546 0.69 23.77 0
1397	HETATM	1050	O	HOH	A	177	39.972 23.760 -2.655 0.86 18.96 0
1398	HETATM	1051	O	HOH	A	178	22.128 30.274 -0.543 0.76 18.78 0
1399	HETATM	1052	O	HOH	A	179	22.244 15.813 10.000 0.68 19.66 0
1400	HETATM	1053	O	HOH	A	180	40.729 9.223 0.292 0.64 20.15 0
1401	HETATM	1054	O	HOH	A	181	12.500 15.267 4.097 0.56 20.12 0
1402	HETATM	1055	O	HOH	A	182	20.372 28.618 -2.353 0.64 20.17 0
1403	HETATM	1056	O	HOH	A	183	22.793 15.462 -6.673 0.63 20.60 0
1404	HETATM	1057	O	HOH	A	184	23.138 31.809 15.121 0.55 20.90 0
1405	HETATM	1058	O	HOH	A	185	22.671 38.691 8.245 0.48 21.16 0
1406	HETATM	1059	O	HOH	A	186	33.966 33.112 6.837 0.59 19.45 0
1407	HETATM	1060	O	HOH	A	187	19.572 25.423 -1.420 0.53 19.94 0
1408	HETATM	1061	O	HOH	A	188	14.790 15.672 7.259 0.52 21.22 0
1409	HETATM	1062	O	HOH	A	189	19.112 28.022 -14.647 0.49 19.83 0
1410	HETATM	1063	O	HOH	A	190	17.302 39.059 -12.453 0.52 20.14 0
1411	HETATM	1064	O	HOH	A	191	16.198 14.502 5.577 0.46 20.78 0
1412	HETATM	1065	O	HOH	A	192	17.345 46.346 -7.080 0.50 18.13 0
1413	HETATM	1066	O	HOH	A	193	14.992 31.300 -4.242 0.46 17.90 0
1414	HETATM	1067	O	HOH	A	194	28.196 44.775 -3.148 0.44 18.15 0
1415	HETATM	1068	O	HOH	A	195	29.479 13.863 -9.107 0.44 18.30 0
1416	HETATM	1069	O	HOH	A	196	23.613 44.811 2.608 0.45 17.66 0
1417	HETATM	1070	O	HOH	A	197	40.572 22.184 -6.358 0.42 18.06 0
1418	HETATM	1071	O	HOH	A	198	12.475 31.860 -6.226 0.47 17.85 0
1419	HETATM	1072	O	HOH	A	199	16.684 13.594 -5.832 0.31 18.51 0
1420	HETATM	1073	O	HOH	A	200	27.534 38.059 -12.862 0.48 18.19 0
1421	HETATM	1074	O	HOH	A	201	25.892 35.973 11.563 0.46 18.15 0
1422	HETATM	1075	O	HOH	A	202	24.790 25.182 16.063 0.46 17.64 0
1423	HETATM	1076	O	HOH	A	203	12.580 21.214 5.006 0.51 17.97 0
1424	HETATM	1077	O	HOH	A	204	19.687 23.750 -4.851 0.37 18.08 0
1425	HETATM	1078	O	HOH	A	205	27.098 35.956 -12.358 0.39 18.71 0
1426	HETATM	1079	O	HOH	A	206	37.255 9.634 10.002 0.46 18.39 0
1427	HETATM	1080	O	HOH	A	207	43.755 23.843 8.038 0.38 17.96 0

注意这里删除水是一个很粗糙的处理，有的时候有些水是想留下来的，比如口袋里的结晶水，比如某些已知具有功能的水

在linux下可以使用命令行解决

```
grep -v HOH 1aki.pdb > 1AKI_clean.pdb
```

```
(gcc) root@bohrium-11312-1251688:~/test# grep --help
Usage: grep [OPTION]... PATTERNS [FILE]...
Search for PATTERNS in each FILE.
Example: grep -i 'hello world' menu.h main.c
PATTERNS can contain multiple patterns separated by newlines.

Pattern selection and interpretation:
-E, --extended-regexp  PATTERNS are extended regular expressions
-F, --fixed-strings    PATTERNS are strings
-G, --basic-regexp     PATTERNS are basic regular expressions
-P, --perl-regexp      PATTERNS are Perl regular expressions
-e, --regexp=PATTERNS  use PATTERNS for matching
-f, --file=FILE        take PATTERNS from FILE
-i, --ignore-case       ignore case distinctions in patterns and data
                        do not ignore case distinctions (default)
--no-ignore-case        do not ignore case distinctions (default)
-w, --word-regexp      match only whole words
-x, --line-regexp       match only whole lines
-z, --null-data         a data line ends in 0 byte, not newline

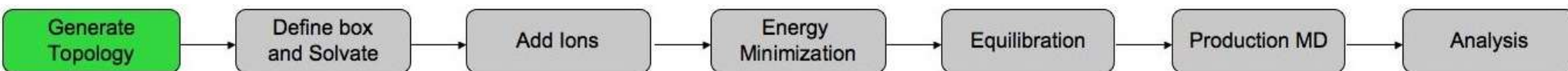
Miscellaneous:
-s, --no-messages      suppress error messages
-v, --invert-match      select non-matching lines
-V, --version           display version information and exit
--help                 display this help text and exit
```

grep命令会按顺序读入1aki.pdb文件的每一行，-v标签代表反选，-v HOH代表只有那些不含HOH的行会被选择，>是重定向符号，不含HOH的行会被导入一个1AKI_clean.pdb的文件中。

All Command Needed

1. `gmx pdb2gmx -f 1AKI_clean.pdb -o 1AKI.gro -p 1AKI.top -ignh`
2. `gmx editconf -f 1AKI.gro -o 1AKI_box.gro -c -d 1.2 -bt cubic`
3. `gmx solvate -cp 1AKI_box.gro -cs spc216.gro -o 1AKI_solv.gro -p 1AKI.top`
4. `gmx grompp -f ions.mdp -c 1AKI_solv.gro -p 1AKI.top -o ions.tpr`
5. `gmx genion -s ions.tpr -o ions.gro -p 1AKI.top -pname NA -nname CL -conc 0.15 -neutral`
6. `gmx grompp -f em.mdp -c ions.gro -p 1AKI.top -o em.tpr`
7. `gmx mdrun -v -deffnm em`
8. `gmx grompp -f nvt.mdp -c em.gro -p 1AKI.top -r em.gro -o nvt.tpr`
9. `gmx mdrun -v -deffnm nvt`
10. `gmx grompp -f npt.mdp -c nvt.gro -p 1AKI.top -r nvt.gro -o npt.tpr`
11. `gmx mdrun -v -deffnm npt`
12. `gmx grompp -f md.mdp -c npt.gro -p 1AKI.top -o md.tpr`
13. `gmx mdrun -v -deffnm md -nt 16`

Generate Topology



```
gmx pdb2gmx -f 1AKI_clean.pdb -o 1AKI.gro -p 1AKI.top -ignh
```

gmx软件的pdb2gmx命令会将pdb文件转化成gmx内部的格式gro文件，并产生一个拓扑文件（topol），通过-p标签可以给此拓扑文件命名，否则会叫topol.top

-ignh标签代表gmx会忽略输入文件中所有的氢原子，并自行重新加氢。这是防止PDB中氢的名字gmx软件不认识

这之后，需要选择使用的力场参数，如果是默认的gromacs，我们一般选择这里的AMBER99SB-ILDN protein, nucleic AMBER94

水的力场就选TIP3P好了

```
(base) root@bohrium-11312-1251688:~/test# gmx pdb2gmx -f 1AKI_clean.pdb -o 1AKI.gro -p 1AKI.top
:-) GROMACS - gmx pdb2gmx, 2024.3 (-:)
```

```
Executable: /root/software/gromacs-2024.3/bin/gmx
Data prefix: /root/software/gromacs-2024.3
Working dir: /root/test
Command line:
  gmx pdb2gmx -f 1AKI_clean.pdb -o 1AKI.gro -p 1AKI.top
```

Select the Force Field:

From '/root/software/gromacs-2024.3/share/gromacs/top':

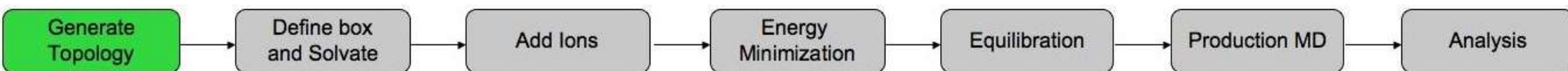
- 1: AMBER03 protein, nucleic AMBER94 (Duan et al., J. Comp. Chem. 24, 1999-2012, 2003)
- 2: AMBER94 force field (Cornell et al., JACS 117, 5179-5197, 1995)
- 3: AMBER96 protein, nucleic AMBER94 (Kollman et al., Acc. Chem. Res. 29, 461-469, 1996)
- 4: AMBER99 protein, nucleic AMBER94 (Wang et al., J. Comp. Chem. 21, 1049-1074, 2000)
- 5: AMBER99SB protein, nucleic AMBER94 (Hornak et al., Proteins 65, 712-725, 2006)
- 6: AMBER99SB-ILDN protein, nucleic AMBER94 (Lindorff-Larsen et al., Proteins 78, 1950-58, 2010)

```
Using the Amber99sb-ildn force field in directory amber99sb-ildn.ff
Opening force field file /root/software/gromacs-2024.3/share/gromacs/top/amber99sb-ildn.ff/watermodels.dat
```

Select the Water Model:

- 1: TIP3P TIP 3-point, recommended

Generate Topology



如果上面的命令成功执行，会看到类似的提示，successfully generate ...
(不管成功与否，gmx都会附赠一句名人名言)

```
----- PLEASE NOTE -----
You have successfully generated a topology from: 1AKI_clean.pdb.
The Amber99sb-ildn force field and the tip3p water model are used.

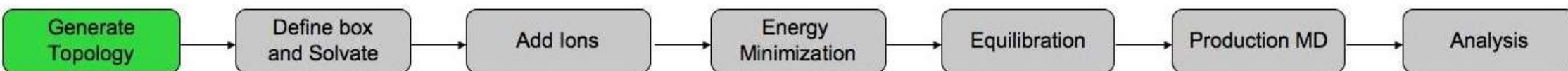
----- ETON ESAELP -----

GROMACS reminds you: "First off, I'd suggest printing out a copy of the GNU coding standards, and NOT read it. Burn them, it's a great symbolic gesture.
" (Linus Torvalds)
```

此时用ls检查当前目录下的内容，会注意到多出来3个文件，1AKI.gro, 1AKI.top, posre.itp

```
(base) root@bohrium-11312-1251688:~/test# ls
1AKI.gro  1AKI.top  1AKI_clean.pdb  1aki.pdb  posre.itp
```


Generate Topology – Gro File



打开1AKI.gro文件（命令行下可以用vim命令）

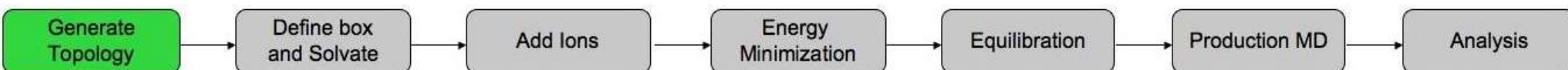
```
LYSOZYME
1960
1LYS      N      1      3.536      2.234      -1.198
1LYS      H1      2      3.472      2.214      -1.272
1LYS      H2      3      3.490      2.285      -1.126
1LYS      H3      4      3.612      2.289      -1.234
1LYS      CA      5      3.589      2.107      -1.143
1LYS      HA      6      3.633      2.055      -1.216
1LYS      CB      7      3.687      2.144      -1.031
1LYS      HB1     8      3.639      2.201      -0.964
1LYS      HB2     9      3.763      2.195      -1.070
1LYS      CG     10     3.745      2.025      -0.956
1LYS      HG1    11     3.770      1.954      -1.023
1LYS      HG2    12     3.676      1.989      -0.894
1LYS      CD     13     3.869      2.065      -0.877
1LYS      HD1    14     3.849      2.147      -0.824
1LYS      HD2    15     3.945      2.083      -0.940
1LYS      CE     16     3.906      1.951      -0.784
1LYS      HE1    17     3.906      1.864      -0.833
1LYS      HE2    18     3.841      1.946      -0.708
1LYS      NZ     19     4.042      1.977      -0.730
```

第一行是蛋白的名字，PDB里没有的话gmx会随便取一个

第二行是整个gro文件里的原子个数，这个数字如果不对，后续会报错，所以如果你对gro文件做了编辑，记得修改原子数到正确的数字

之后的行就是每个原子的信息了

Generate Topology – Posre File



打开posre.itp文件（命令行下可以用vim命令）

```

; In this topology include file, you will find position restraint
; entries for all the heavy atoms in your original pdb file.
; This means that all the protons which were added by pdb2gmx are
; not restrained.

[ position_restraints ]
; atom  type    fx      fy      fz
   1      1    1000    1000    1000
   5      1    1000    1000    1000
   7      1    1000    1000    1000
  10      1    1000    1000    1000
  13      1    1000    1000    1000
  16      1    1000    1000    1000
  
```

这是一个用来给蛋白质加限制的文件，注意到对每一个原子，都在x, y, z方向施加了限制力，单位是kJ mol⁻¹ nm⁻¹

Table 3 Derived units. Note that an additional conversion factor of 10²⁸ a.m.u (≈ 16.6) is applied to get bar instead of internal MD units in the energy and log files

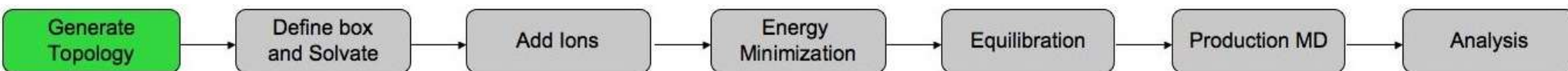
Quantity	Symbol	Unit
energy	E, V	kJ mol ⁻¹
Force	F	kJ mol⁻¹ nm⁻¹
pressure	p	bar
velocity	v	nm ps ⁻¹ = 1000 m s ⁻¹
dipole moment	μ	e nm
electric potential	Φ	kJ mol ⁻¹ e ⁻¹ = 0.010 364 269 19 Volt
electric field	E	kJ mol ⁻¹ nm ⁻¹ e ⁻¹ = 1.036 426 919 × 10 ⁷ Vm ⁻¹

```

1943      1    1000    1000    1000
1945      1    1000    1000    1000
1948      1    1000    1000    1000
1950      1    1000    1000    1000
1954      1    1000    1000    1000
1958      1    1000    1000    1000
1959      1    1000    1000    1000
1960      1    1000    1000    1000
  
```

注意到有1960条记录，对应1960个蛋白质的原子

Generate Topology – Topol File



打开1AKI.top文件（命令行下可以用vim命令），就是所谓的拓扑文件

```
; Created by:
;           :-) GROMACS - gmx pdb2gmx, 2024.3 (-:
;
; Executable: /root/software/gromacs-2024.3/bin/gmx
; Data prefix: /root/software/gromacs-2024.3
; Working dir: /root/test
; Command line:
;   gmx pdb2gmx -f 1AKI_clean.pdb -o 1AKI.gro -p 1AKI.top
; Force field was read from the standard GROMACS share directory.
;
```

拓扑文件，顾名思义，描述了整个蛋白的拓扑，或者说蛋白所有原子的连接方式

所有;开头的行都是被注释的，或者说不会被gmx软件读取，这里1AKI.top文件开头的注释里写了一些基本信息，比如此文件是用哪个位置的gmx，用什么命令生成的

```
; Include forcefield parameters
#include "amber99sb-ildn.ff/forcefield.itp"
```

继续读topol文件，会看到一行导入了整体力场文件，这个forcefield.itp文件非常重要，我们后续再检查这个文件里的内容

Generate Topology – Topol File



打开1AKI.top文件（命令行下可以用vim命令），就是所谓的拓扑文件

```
[ moleculetype ]
; Name          nrexcl
Protein_chain_A 3

[ atoms ]
; nr      type  resnr residue  atom  cgnr      charge      mass  typeB
; residue 1 LYS rtp NLYS q +2.0
  1       N3     1    LYS      N     1       0.0966     14.01
  2       H      1    LYS      H1    2       0.2165     1.008
  3       H      1    LYS      H2    3       0.2165     1.008
  4       H      1    LYS      H3    4       0.2165     1.008
  5       CT     1    LYS      CA    5      -0.0015     12.01
  6       HP     1    LYS      HA    6       0.118      1.008
  7       CT     1    LYS      CB    7       0.0212     12.01
```

```
[ bonds ]
; ai  aj  funct      c0      c1      c2      c3
  1   2   1
  1   3   1
```

```
[ angles ]
; ai  aj  ak funct
  2   1   3   1
  2   1   4   1
```

```
[ dihedrals ]
; ai  aj  ak  al  funct
  2   1   5   6   9
  2   1   5   7   9
  2   1   5  23   9
```

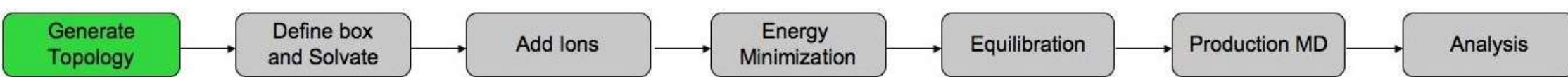
接下来是[moleculetype]字段，一个top文件里可能有多个[moleculetype]字段，会把整个模拟体系里出现的物种一个个都列出来，这里是1AKI里的蛋白，并且被赋予了一个名字Protein_chain_A

之后，对Protein_chain_A的每一个原子，都会给出原子序号，该原子在gmx里的名字（这个名字会被用来匹配力场参数），残基编号，残基类别，该原子在PDB里的名字，电荷与原子质量等

[bonds]字段描述当前[moleculetype]下，所有原子之间是怎么连接的，比如左边的意思就是原子1和原子2之间有键，原子1和原子3之间有键

后续的[angles]字段和[dihedrals]字段依此类推

Generate Topology – Topol File



打开1AKI.top文件（命令行下可以用vim命令），就是所谓的拓扑文件

```
; Include Position restraint file
#ifdef POSRES
#include "posre.itp"
#endif

; Include water topology
#include "amber99sb-ildn.ff/tip3p.itp"

#ifdef POSRES_WATER
; Position restraint for each water oxygen
[ position_restraints ]
; i funct      fcx      fcy      fcz
  1    1      1000      1000      1000
#endif

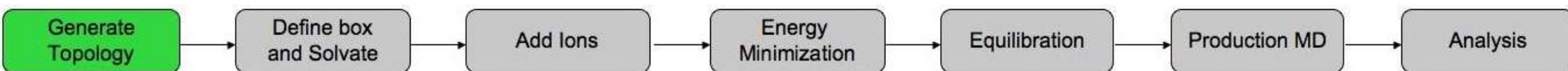
; Include topology for ions
#include "amber99sb-ildn.ff/ions.itp"
```

继续看的话，这里引入了之前提到的限制文件 posre.itp，注意这里的ifdef语句，意思是gmx只有在后续的控制文件（mdp格式）中读到了POSRES才会把限制加在对应的原子上

之后引入了水的力场文件和限制

引入了离子的力场文件

Generate Topology – Topol File



打开1AKI.top文件（命令行下可以用vim命令），就是所谓的拓扑文件

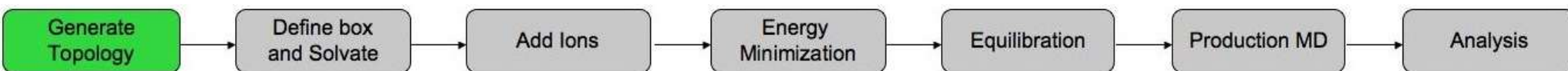
```
[ system ]  
; Name  
LYSOZYME  
  
[ molecules ]  
; Compound      #mols  
Protein_chain_A    1
```

Topol文件的最后，首先是整个体系的名字

[molecules]字段，是模拟体系里所有物种的种类及数目，目前我们还没给体系加水 and 离子，整个体系里只有1个Protein_chain_A，也就是我们的蛋白，这个名字是gmx产生gro文件的时候随意取的，也可以修改，但是记得一定要和上面的[moleculetype]字段里物种的名字保持一致。

后续我们加水加离子以后，[molecules]字段的内容会有变化

Generate Topology – forcefield.itp



```
; Include forcefield parameters
#include "amber99sb-ildn.ff/forcefield.itp"
```

我们在这里回顾一下Topol文件中通过include引入的forcefield.itp文件是什么

```
vim /Path-to-Gromacs/share/gromacs/top/amber99sb-ildn.ff/forcefield.itp
```

```
(base) root@bohrium-11312-1251688:~/test# vim /root/software/gromacs-2024.3/share/gromacs/top/amber99sb-ildn.ff/forcefield.itp
```

进一步打开这个forcefield.itp文件，会发现它引入了非键参数和成键参数

```
#define _FF_AMBER
#define _FF_AMBER99SBILDN

[ defaults ]
; nbfunc      comb-rule      gen-pairs      fudgeLJ      fudgeOQ
1              2              yes             0.5          0.8333

#include "ffnonbonded.itp"
#include "ffbonded.itp"
```



/Path-to-Gromacs/share/gromacs/top/amber99sb-ildn.ff/ffnonbonded.itp

非键项包括原子在gmx里被赋予的名字，对应的质量，电荷，非键作用参数(sigma & epsilon)

atomtypes						
; name	at.num	mass	charge	ptype	sigma	epsilon
Br	35	79.90	0.0000	A	3.95559e-01	1.33888e+00 ; Converted from parm99.dat
C	6	12.01	0.0000	A	3.39967e-01	3.59824e-01
CA	6	12.01	0.0000	A	3.39967e-01	3.59824e-01
CB	6	12.01	0.0000	A	3.39967e-01	3.59824e-01

/Path-to-Gromacs/share/gromacs/top/amber99sb-ildn.ff/ffbonded.itp

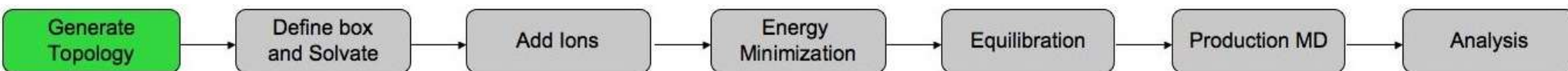
成键项包括键，角，二面角的力常数，平衡位置等信息

bondtypes				
; i	j	func	b0	kb
C	C	1	0.1525	259408.0 ; new99
C	OS	1	0.1323	376560.0 ; new99
C	H4	1	0.1080	307105.6 ; new99
C	H5	1	0.1080	307105.6 ; new99
CA	OH	1	0.1364	376560.0 ; new99

[angletypes]					
; i	j	k	func	th0	cth

[dihedraltypes]							
; i	j	k	l	func	phase	kd	pn

Generate Topology – tip3p.itp



```
; Include water topology  
#include "amber99sb-ildn.ff/tip3p.itp"
```

我们再看看水的力场文件的内容

```
vim /Path-to-Gromacs/share/gromacs/top/amber99sb-ildn.ff/tip3p.itp
```

```
(base) root@bohrium-11312-1251688:~/test# vim /root/software/gromacs-2024.3/share/gromacs/top/amber99sb-ildn.ff/tip3p.itp
```

```
[ moleculetype ]  
; molname      nrexcl  
SOL             2  
  
[ atoms ]  
; id  at type   res nr  res name  at name  cg nr  charge  mass  
1  OW             1      SOL      OW        1     -0.834  16.00000  
2  HW             1      SOL      HW1       1     0.417   1.00800  
3  HW             1      SOL      HW2       1     0.417   1.00800  
  
#ifndef FLEXIBLE  
  
[ settles ]  
; OW      funct  doh    dhh  
1          1    0.09572 0.15139  
  
[ exclusions ]  
1      2      3  
2      1      3  
3      1      2  
  
#else  
  
[ bonds ]  
; i      j      funct  length  force_constant  
1      2      1      0.09572  502416.0  0.09572    502416.0  
1      3      1      0.09572  502416.0  0.09572    502416.0  
  
[ angles ]  
; i      j      k      funct  angle  force_constant  
2      1      3      1      104.52  628.02    104.52  628.02  
  
#endif  
~
```

首先是 [moleculetype] 字段，水在 gro 文件里的名字将会是 SOL，注意如果 gro 里名字和力场文件里不一致的话，gmX 就无法匹配力场，从而报错

同样的，接下来是 [atoms] 字段

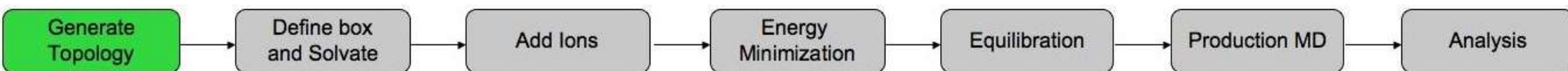
#ifndef，如果没在 mdp 文件里定义 FLEXIBLE 的话，氧与氢原子之间的距离就会被限死在 0.09572 nm，两个氢原子之间的距离会限死在 0.15139 nm。

[exclusions] 里是 3 个非键作用不算，原子 2 和 3，原子 1 和 3，原子 1 和 2

这是为了节省计算资源

#ifndef，如果在 mdp 文件里定义 FLEXIBLE，那水的 3 维构象就是可以动的，此时 [bonds] 字段和 [angles] 字段的力场参数就会发挥作用

Generate Topology



我们稍微小结一下，对处理过的pdb执行pdb2gmx命令后，会产生3个文件

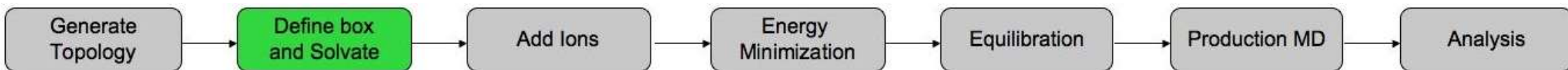
```
gmx pdb2gmx -f 1AKI_clean.pdb -o 1AKI.gro -p 1AKI.top -ignh
```

分别是1AKI.gro 1AKI.top posre.itp

```
(base) root@bohrium-11312-1251688:~/test# ls
1AKI.gro 1AKI.top 1AKI_clean.pdb 1aki.pdb posre.itp
```

- 1AKI.gro里记录了原子数目，每个原子的类型，编号，残基，残基编号，xyz坐标，实际就是类似pdb文件
- 1AKI.top里记录了体系的物种种类，数目，每个物种的原子连接方式，力场参数
- posre.itp是蛋白的限制文件，在后续模拟的时候可以用来给蛋白施加限制

Define box & Solvate



```
gmx editconf -f 1AKI.gro -o 1AKI_box.gro -c -d 1.2 -bt cubic
```

```
129LEU  OC1 1959  5.547  2.785  4.679
129LEU  OC2 1960  5.397  2.745  4.844
8.01008  8.01008  8.01008
```

我们用2个命令加水，首先是editconf命令定义水盒子大小，或者说标定一个区域，这个区域后续会被水填充

-f指定要在哪个文件的基础上定义水盒子大小，这里的输入也可以是PDB文件

-o指定输出文件的名字，是用户自定义的

-c将蛋白放在水盒子的中心，且-d指定蛋白里盒子边界至少1.2nm（建议最少1.2）

-bt指定水盒子的形状是立方体cubic（立方盒子足以处理绝大多数情况）

完成上述命令后，打开1AKI_box.gro，检查最后一行，会看到cubic盒子的长宽高

```
gmx solvate -cp 1AKI_box.gro -cs spc216.gro -o 1AKI_solv.gro -p 1AKI.top
```

之后用solvate命令，它会从一个预平衡过的水文件spc216.gro里把水复制出来，放到1AKI_box.gro里

结果是1AKI_solv.gro文件，通过-p标签提供体系的拓扑文件，因为你加了水，体系里的物种类型和数目有变化

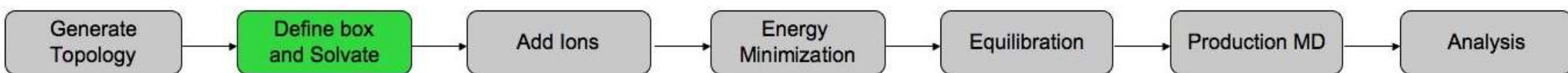
```
vim 1AKI.top
```



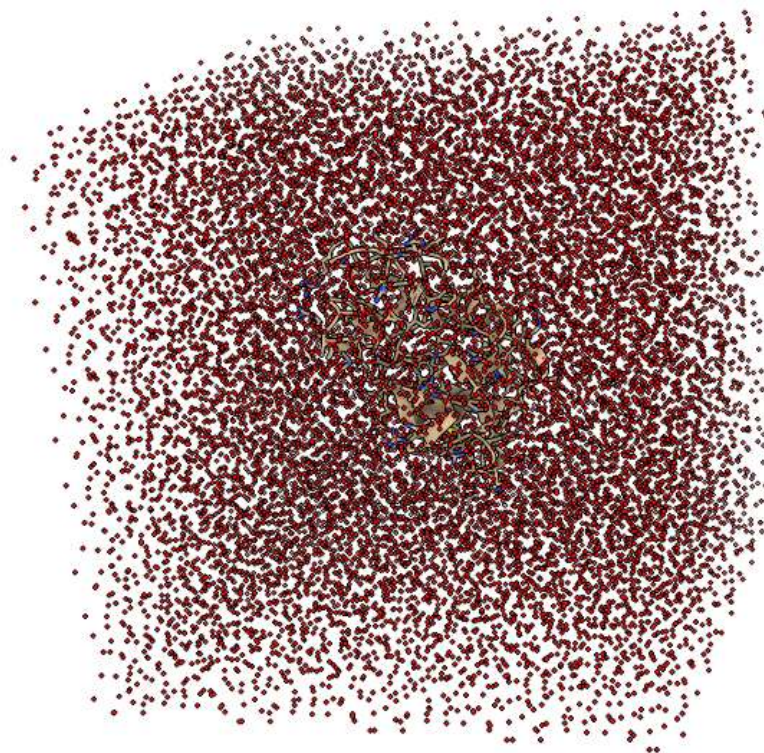
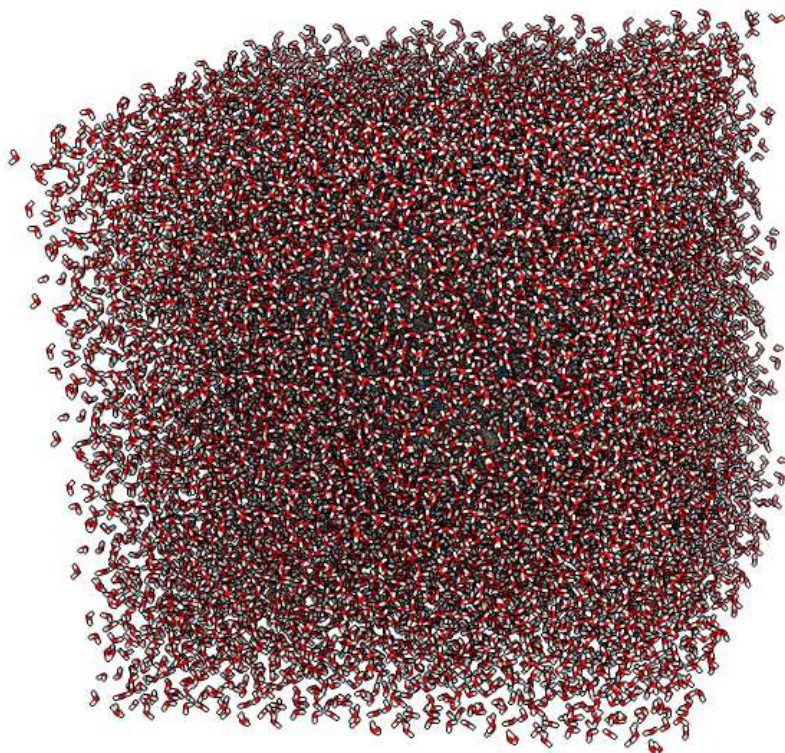
```
[ molecules ]
; Compound      #mols
Protein_chain_A    1
SOL                16336
```

完成上述命令后，打开1AKI.top，检查最后一行，会发现增加了物种SOL，数目为16336，意思是添加了16336个水

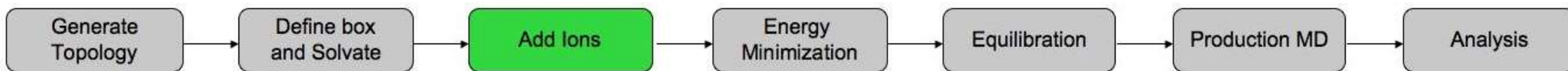
Define box & Solvate



此时用ChimeraX打开，看到的是这种，记得显示所有原子



Add Ions



接下来要向体系中添加离子，添加离子的目的有2个：

- 中和蛋白和配体等可能带有的电荷，使得整个体系处于电中性；
- 添加NaCl，使得模拟体系的盐浓度达到生理标准，也就是0.15M

我们使用2句命令来做这件事，同时需要一个ions.mdp文件


```
gmX grompp -f ions.mdp -c 1AKI_solv.gro -p 1AKI.top -o ions.tpr
```

首先，使用gmX软件的grompp命令，准备一个输入文件 ions.tpr，在gromacs中，凡是涉及到具体模拟的，都需要准备一个tpr格式的输入，注意不同版本gmX制作的tpr文件很可能是不通用的

```
gmX genion -s ions.tpr -o ions.gro -p 1AKI.top -pname NA -nname CL -conc 0.15 -neutral
```

之后，使用genion命令加离子。所谓加离子，其实是选择一些溶剂分子，替换成离子。

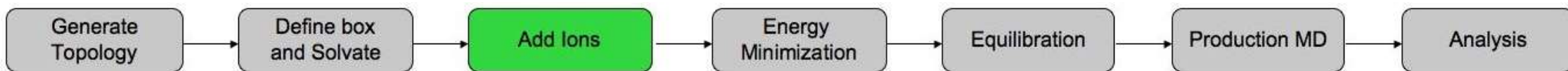
- -pname指定加入什么正离子
- -nname指定负离子
- -conc 0.15 的意思是加入上面确定的正负离子直到盐浓度为0.15M
- -neutral意为加入额外的离子中和体系，比如如果我们的体系带了一个正电荷，这里就会在0.15M之外，再加入一个Cl⁻离子

使用genion命令后需要选择把什么分子替换成离子，记得一定要选水（这里Water和SOL都是水）

```
Command line:
gmX genion -s ions.tpr -o ions.gro -p 1AKI.top -pname NA -nname CL -conc 0.15 -neutral

Reading file ions.tpr, VERSION 2024.3 (single precision)
Reading file ions.tpr, VERSION 2024.3 (single precision)
Will try to add 46 NA ions and 54 CL ions.
Select a continuous group of solvent molecules
Group 0 (      System) has 50968 elements
Group 1 (      Protein) has  1960 elements
Group 2 (    Protein-H) has  1001 elements
Group 3 (      C-alpha) has   129 elements
Group 4 (    Backbone) has   387 elements
Group 5 (    MainChain) has   517 elements
Group 6 ( MainChain+Cb) has   634 elements
Group 7 ( MainChain+H) has   646 elements
Group 8 (    SideChain) has  1314 elements
Group 9 ( SideChain-H) has   484 elements
Group 10 (  Prot-Masses) has   1960 elements
Group 11 ( non-Protein) has 49008 elements
Group 12 (      Water) has 49008 elements
Group 13 (      SOL) has 49008 elements
Group 14 ( non-Water) has   1960 elements
Select a group: 13
```


Add Ions



你现在应该有一个问题，就是你没有ions.mdp, 这里给出一个例子

ions.mdp

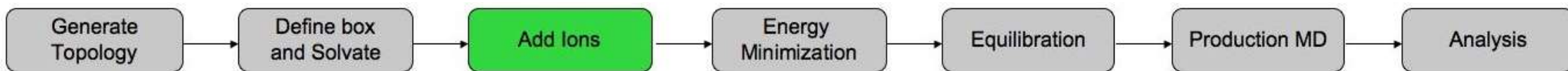
```
; LINES STARTING WITH ';' ARE COMMENTS
title      = Minimization    ; Title of run

; 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 < 10.0 kJ/mol
emstep      = 0.01           ; Energy step size
nsteps      = 50000          ; Maximum number of (minimization) steps to perform

; Parameters describing how to find the neighbors of each atom and how to calculate the interactions
nstlist     = 1              ; Frequency to update the neighbor list and long range forces
cutoff-scheme = Verlet
ns_type     = grid           ; Method to determine neighbor list (simple, grid)
rlist       = 1.0            ; Cut-off for making neighbor list (short range forces)
coulombtype  = cutoff        ; Treatment of long range electrostatic interactions
rcoulomb     = 1.0           ; long range electrostatic cut-off
rvdw         = 1.0           ; long range Van der Waals cut-off
pbc         = xyz            ; Periodic Boundary Conditions
```

加离子使用的mdp文件其实很任意，几乎所有情况都可以用左边这个，在这里我们暂时不去理解它的内容了，可以无脑使用

Add Ions



加完离子以后，会产生ions.gro，当然这个名字是我执行genion命令的时候随意取的，你也可以自己取别的名字，我的建议是不要改变，方便记录这个文件是哪一步的结果

```
gmX genion -s ions.tpr -o ions.gro -p 1AKI.top -pname NA -nname CL -conc 0.15 -neutral
```

同时，我们的拓扑文件，也就是-p后指定的1AKI.top也会发生变化

1AKI.top的最后几行，加离子前

```
[ molecules ]
; Compound      #mols
Protein_chain_A    1
SOL                16336
```



1AKI.top的最后几行，加离子后

```
[ molecules ]
; Compound      #mols
Protein_chain_A    1
SOL                16236
NA                  46
CL                  54
```

你会注意到水的数目变少了，同时出现了NA和CL离子

Energy Minimization



接下来要对加完离子的体系做能量最小化了

- **目的：**EM步骤的主要目标是消除系统中的空间冲突和不合理的几何结构。这一步有助于减少由于粒子间距离过近而产生的过大的力，从而稳定系统的初始构型。
- **结果：**能量最小化后，系统达到较低的势能状态，原子位置和键长更加合理。这一步对于防止不切实际的力在后续动力学模拟中引起不稳定至关重要。

我们也使用2句命令来做这件事，同时需要一个em.mdp文件

```
gmx grompp -f em.mdp -c ions.gro -p 1AKI.top -o em.tpr
```

首先使用grompp命令准备输入文件

- **-f em.mdp** 提供控制文件，下一页PPT会给出此文件内容
- **-c ions.gro** 提供加完离子的坐标文件，是上一步的输出
- **-p 1AKI.top** 提供拓扑文件，里面有体系内所有原子的连接方式，以及力场参数
- **-o em.tpr** 提供一个能进行模拟的tpr文件，.tpr文件是一种非常重要的文件格式，它包含了进行分子动力学模拟所需的所有信息。.tpr文件通常被称为“便携式运行输入文件”（Portable Run Input File），它是一个二进制文件，包含了拓扑信息、初始坐标、速度、力场参数以及模拟参数等。

```
gmx mdrun -v -deffnm em
```

gmx mdrun: 启动分子动力学模拟。

- **-v:** 启用详细输出模式，显示每一步的详细信息。
- **-deffnm em:** 指定输入tpr文件的前缀为em，输出文件的默认文件名前缀为em，也就是说会产生em.gro

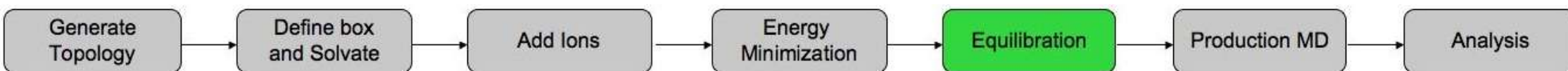
Energy Minimization



em.mdp

```
; 能量最小化核心参数
;-----
integrator      = steep ;最陡下降法（适合初步结构优化，稳定但较慢）
; integrator    = cg   ;可选：共轭梯度法（收敛更快，可在steep后切换使用）
emtol          = 10.0  ;最大力收敛阈值（Amber推荐≤10.0 kJ/mol/nm）
nsteps         = 10000 ;最大优化步数（确保足够步数以达emtol）
;-----
; 非键相互作用参数
;-----
cutoff-scheme   = Verlet ;Verlet截断方案（自动管理邻居列表，推荐）
nstlist        = 20    ;邻居列表更新频率
rlist          = 1.0   ;邻居列表截断半径（必须≥rvdw/rcoulomb）
vdwtype        = Cut-off ;Amber标准范德华截断
rvdw           = 1.0   ;范德华截断半径
coulombtype     = pme   ;PME处理长程静电
rcoulomb       = 1.0   ;直接空间静电截断（与rvdw对齐）
;-----
; 约束算法
;-----
constraints     = h-bonds ;仅约束氢键（Amber力场参数化基于此设置）
constraint_algorithm = LINCS ;LINCS算法
```

NVT Equilibration



接下来要对EM结束的体系进行NVT平衡（恒离子数-恒体积-恒温）

•目的：NVT步骤旨在将系统加热到所需的温度并使其稳定。这一步确保系统的温度达到所需的值并保持稳定，对于生物系统通常为300 K左右。

•结果：NVT平衡结束时，系统的温度应保持稳定，动能分布应与所需温度一致。这一步有助于确保系统在进入下一阶段之前达到热平衡。

```
gmX grompp -f nvt.mdp -c em.gro -p 1AKI.top -r em.gro -o nvt.tpr
```

首先使用grompp命令准备输入文件

- -f nvt.mdp 提供控制文件，下一页PPT会给出此文件内容
- -c em.gro 提供加完离子的坐标文件，是上一步的输出
- -p 1AKI.top 提供拓扑文件，里面有体系内所有原子的连接方式，以及力场参数
- -o nvt.tpr 提供一个能进行模拟的tpr文件
- -r em.gro 此时，在平衡的时候，对体系施加了限制，此选项提供一个平衡位置坐标，偏离此坐标即会受到约束

```
gmX mdrun -v -deffnm nvt
```

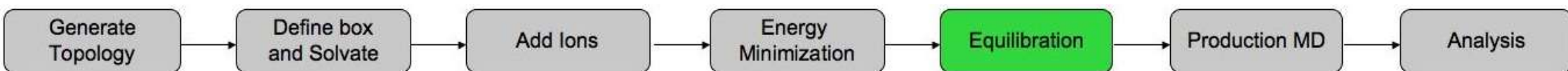
gmX mdrun：启动分子动力学模拟。

- -v：启用详细输出模式，显示每一步的详细信息。
- -deffnm nvt：指定输入tpr文件的前缀为nvt，输出文件的默认文件名前缀为nvt，也就是说会产生nvt.gro

这一步需要站起来倒一杯水，做50个俯卧撑的时间

```
starting mdrun 'LYSOZYME in water'
500000 steps,      500.0 ps.
step 3800: timed with pme grid 72 72 72, coulomb cutoff 1.000: 385.6 M-cycles
step 4000: timed with pme grid 60 60 60, coulomb cutoff 1.113: 352.4 M-cycles
step 4200: timed with pme grid 52 52 52, coulomb cutoff 1.284: 418.7 M-cycles
```

NVT Equilibration



nvt.mdp

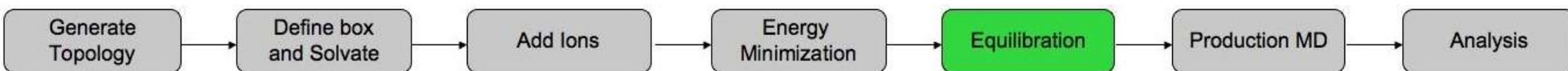
```
define          = -DPOSRES ;按需启用位置限制
integrator       = md      ;分子动力学积分器
dt              = 0.001    ;时间步长1 fs
nsteps          = 500000   ;500 ps模拟时间
nstxtcout       = 5000    ;每5 ps输出坐标
nstvout         = 5000    ;每5 ps输出速度
nstfout         = 5000    ;每5 ps输出力
nstcalcenergy   = 100     ;每100步计算能量
nstenergy       = 1000    ;每1 ps输出能量
nstlog          = 1000    ;每1 ps输出日志
;非键相互作用
cutoff-scheme   = Verlet   ;Verlet截断方案
nstlist         = 20      ;邻居列表更新频率
rlist           = 1.0     ;邻居列表半径
coulombtype     = pme      ;PME处理静电
rcoulomb        = 1.0     ;静电截断1.0 nm
vdwtype         = Cut-off  ;范德华截断
vdw-modifier    = None
rvdw            = 1.0     ;范德华截断1.0 nm
;温度控制
tcoupl          = V-rescale ;V-rescale控温
tc_grps        = Protein Non-Protein ;分组控温
tau_t           = 0.2 0.2 ;V-rescale的时间常数
ref_t           = 310.0 310.0 ;参考温度310 K
```

还记得吗？1AKI.top里，决定限制文件posre.itp是否被启动的行

```
; Include Position restraint file
#ifdef POSRES
#include "posre.itp"
#endif
```

```
;约束参数
constraints      = h-bonds ;约束
constraint_algorithm = LINCS ;LINCS算法
;质心运动移除
nstcomm         = 100     ;每100步移除质心运动
comm_mode       = linear  ;线性模式
comm_grps       = Protein Non-Protein ;全系统质心移除
;初始速度
gen-vel         = yes     ;生成初始速度
gen-temp        = 310.0   ;初始温度310 K
gen-seed        = -1     ;随机种子
;参考坐标
refcoord_scaling = com    ;基于质心缩放
```

NPT Equilibration



接下来要对NVT结束的体系进行NPT平衡（恒离子数-恒压-恒温）

- **目的：**NPT步骤通过施加恒定压力同时保持所需温度来稳定系统的密度。这一步确保系统达到正确的密度，这对于模拟现实条件至关重要。

- **结果：**NPT平衡后，系统的密度应达到所需的稳定值。这一步对于压力和体积波动显著的系统尤为重要，例如涉及水或其他溶剂的模拟。

```
gmx grompp -f npt.mdp -c nvt.gro -p 1AKI.top -r nvt.gro -o npt.tpr
```

首先使用grompp命令准备输入文件

- -f npt.mdp 提供控制文件，下一页PPT会给出此文件内容
- -c nvt.gro 提供加完离子的坐标文件，是上一步的输出
- -p 1AKI.top 提供拓扑文件，里面有体系内所有原子的连接方式，以及力场参数
- -o npt.tpr 提供一个能进行模拟的tpr文件
- -r nvt.gro 此时，在平衡的时候，对体系施加了限制，此选项提供一个平衡位置坐标，偏离此坐标即会受到约束

```
gmx mdrun -v -deffnm npt
```

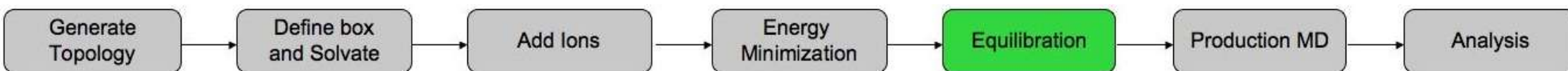
gmx mdrun：启动分子动力学模拟。

- -v：启用详细输出模式，显示每一步的详细信息。
- -deffnm npt：指定输入tpr文件的前缀为npt，输出文件的默认文件名前缀为npt，也就是说会产生npt.gro

这一步需要数十分钟到一小时（有GPU）

```
starting mdrun 'LYSOZYME in water'
1000000 steps, 2000.0 ps.
step 3100: timed with pme grid 72 72 72, coulomb cutoff 1.000: 341.8 M-cycles
step 3300: timed with pme grid 60 60 60, coulomb cutoff 1.113: 370.0 M-cycles
step 3500: timed with pme grid 52 52 52, coulomb cutoff 1.284: 443.8 M-cycles
```

NPT Equilibration

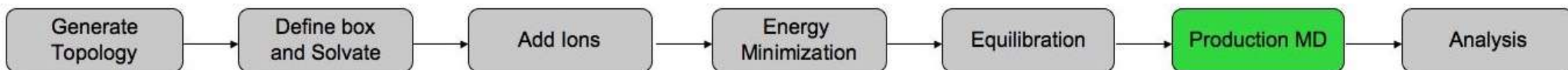


npt.mdp

```
define          = -DPOSRES ;启用位置限制（按需调整）
integrator       = md      ;分子动力学积分器
dt              = 0.002    ;时间步长2 fs（需约束所有键）
nsteps          = 1000000  ;2000 ps模拟时间
nstxtcout       = 5000    ;每10 ps输出坐标
nstvout         = 5000    ;每10 ps输出速度
nstfout        = 5000    ;每10 ps输出力
nstcalcenergy   = 100     ;每100步计算能量
nstenergy       = 1000    ;每2 ps输出能量
nstlog          = 1000    ;每2 ps输出日志
;非键相互作用
cutoff-scheme   = Verlet   ;Verlet截断方案
nstlist         = 20      ;优化邻居列表更新频率
rlist           = 1.0     ;邻居列表半径
coulombtype     = pme      ;PME处理静电
rcoulomb        = 1.0     ;静电截断1.0 nm
vdwtype        = Cut-off   ;范德华截断
vdw-modifier    = None
rvdw            = 1.0     ;范德华截断1.0 nm
;温度控制
tcoupl          = V-rescale ;V-rescale控温
tc_grps        = Protein Non-Protein ;分组控温
tau_t           = 0.2 0.2 ;V-rescale的时间常数
ref_t           = 310.0 310.0 ;参考温度310 K
```

```
;压力控制
pcoupl          = C-rescale ;C-rescale控压
pcoupltype      = isotropic ;各向同性压力
tau_p           = 0.5     ;压力耦合时间常数
compressibility = 4.5e-5   ;压缩率（水体系常用4.5e-5 bar-1）
ref_p           = 1.0     ;参考压力1.0 bar
;约束参数
constraints     = h-bonds ;约束
constraint_algorithm = LINCS ;LINCS算法
;质心运动移除
nstcomm        = 100     ;每100步移除质心运动
comm_mode      = linear  ;线性模式
comm_grps      = Protein Non-Protein ;全系统质心移除
;参考坐标
refcoord_scaling = com    ;基于质心缩放
```


Production MD



历经千难万险，终于可以进行生产相的模拟了

```
gmx grompp -f md.mdp -c npt.gro -p 1AKI.top -o md.tpr
```

首先使用grompp命令准备输入文件

- 生产相的模拟不加约束，所以没有-r了
- md.mdp在下一页

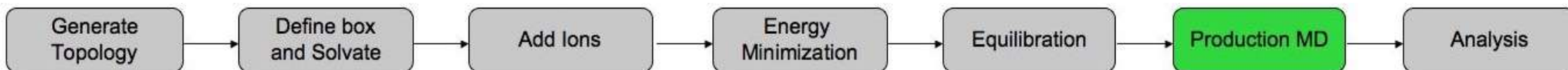
```
gmx mdrun -v -deffnm md -nt 16
```

第三次见面了，不多做解释了

- -nt 16
作用：指定使用16个CPU线程并行计算。适用于多核工作站或计算节点。
(若物理核心不足(如CPU仅有8核)，超线程可能导致性能下降。)
- 不设置-nt时会默认使用所有线程

此步可以按10万原子体系一块4090一天超过100ns估计速度

Production MD



md.mdp

```
integrator = md ; leap-frog积分器
nsteps = 250000000 ; 500 ns模拟时间
dt = 0.002 ; 时间步长2 fs
nstxout = 50000 ; trr输出, 设置为0则不输出trr
nstvout = 50000 ;
nstfout = 50000 ;
nstxout-compressed = 50000 ; 每100 ps输出压缩轨迹
compressed-x-precision = 1000 ; 轨迹精度
compressed-x-grps = Protein ; xtc格式输出组
nstlog = 5000 ; 每10 ps输出日志
nstenergy = 5000 ; 每10 ps输出能量
nstcalcenergy = 100 ;
; 非键相互作用
cutoff-scheme = Verlet
nstlist = 20
rlist = 1.0
coulombtype = pme
rcoulomb = 1.0
vdwtype = Cut-off
vdw-modifier = None
rvdw = 1.0
```

```
; 温度控制
tcoupl = Nose-Hoover ;
tc_grps = Protein Non-Protein ; 分组控温
tau_t = 1.0 1.0 ;
ref_t = 310.0 310.0 ; 参考温度310 K
; 压力控制
pcoupl = Parrinello-Rahman
pcoupltype = isotropic
tau_p = 5.0 ;
compressibility = 4.5e-5
ref_p = 1.0
; 约束参数
constraints = h-bonds ; 约束
constraint_algorithm = LINCS
continuation = yes
; 质心运动移除
nstcomm = 100 ; 每100步移除质心运动
comm_mode = linear
comm_grps = Protein Non-Protein
; 参考坐标
refcoord_scaling = com
```