

GROMACS
FAST. FLEXIBLE. FREE.



GROMACS与AMBER软件安装以及初 始结构搭建的常用方法

常用力场简介，常用体系的模型搭建

主讲人：许楠
浙江大学博士研究生
VASPKIT合作开发者

四大生物模拟软件简介



生物体系的四大模拟神器。

#生物模拟四大神器

Charmm, Amber, Gromacs, NAMD



Amber

GROMACS
FAST. FLEXIBLE. FREE.



NAMD
Scalable Molecular Dynamics

#Charmm

Charmm是分子模拟软件的鼻祖，Amber, Gromacs, NAMD的开发都离不开Charmm软件。缺点是文档欠缺，不易入门。CHARMM-GUI的出现稍微弥补了以上不足。另外CHARMM软件需要收费。

#Gromacs

完全免费，速度更快，灵活性更高，自带或者第三方分析工具更多。支持charmm, amber, oplis-AA, Gromacs等力场且容易扩展。用户数量最多，文档，教程齐全。是现阶段生物模拟领域的宠儿。

#AMBER

AMBER是老牌的蛋白模拟软件。功能强大，但是要收费。免费的AmberTools具有优秀的建模，后处理功能，其中的sander模块也能做分子动力学模拟。Amber力场对蛋白的模拟非常成熟。

#NAMD

由可视化软件VMD开发组开发，与VMD兼容性更好。力场可扩展性更强，支持功能更强的QM/MM模拟。并行效率奇高，把Charmm的优势发挥到极致。但是用户规模相比Gromacs较小。

Gromacs和Amber 安装

生物体系的四大模拟神器之二。

#安装Gromacs

Ubuntu预编译版在线安装:

```
sudo apt-get install gromacs
```

Centos/RedHat预编译版在线安装:

```
sudo yum install gromacs
```

Linux并行版安装/Windows编译安装:

<http://jerkwin.github.io/9999/11/01/GROMACS程序编译/>

或者 <http://sobereva.com/457> 和 <http://sobereva.com/458>

#安装AmberTools

使用Anaconda环境的conda工具在线安装:

```
https://blog.csdn.net/u013069552/article/details/78349716
```

```
conda install ambertools=19 -c ambermd
```

Linux单机版安装/Windows编译安装:

<http://jerkwin.github.io/2017/12/28/Windows下AmberTools17的编译/>

或者 http://blog.sina.com.cn/s/blog_61fc08490102x2t0.html

力场简介

力场的精度决定了模拟的准确性。

分子动力学的实质

根据牛顿第二定律 $F = ma$ 求解粒子的运动方程，得到粒子随时间变化的运动轨迹。

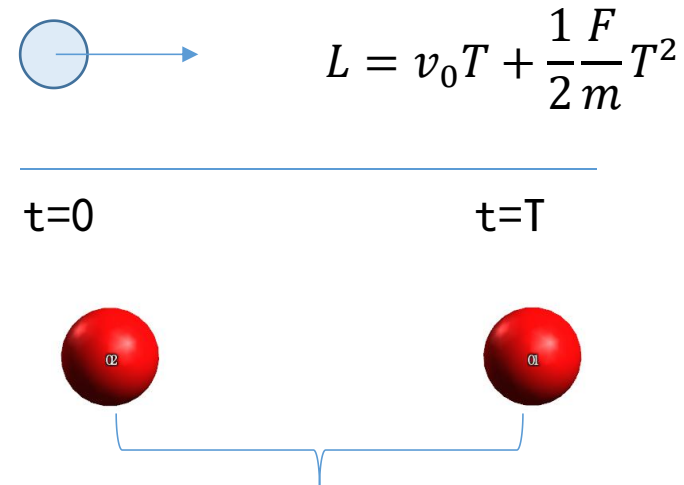
如何获得粒子的受力？

每个时间步，粒子所处的环境是不一样的，因此需要重新计算受力。力可以视为能量的一阶导数，就像重力是重力势能的梯度一样。因此需要在每一步计算体系的能量，并对该粒子的位置做偏导就可以得到粒子的受力： $F_i = \frac{\partial E}{\partial x_i}$ 。能量与系统所有自由度的函数关系叫做势能面。

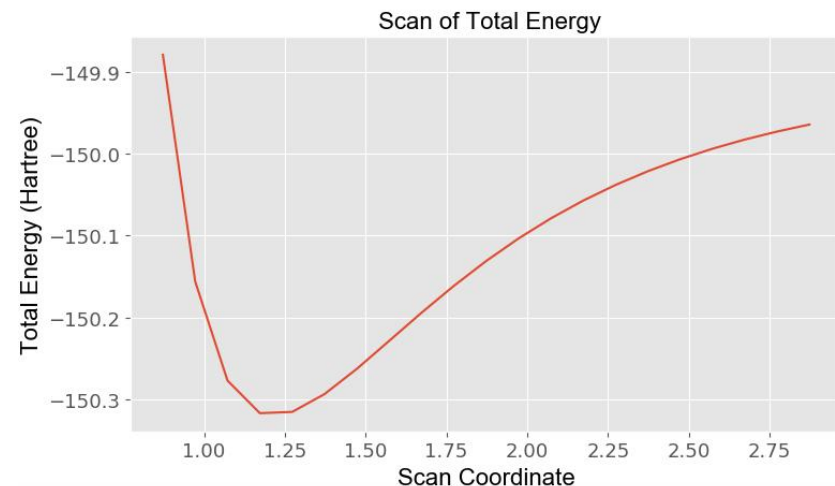
力场的作用

使用数学表达式的组合拟合并重现实验或量化计算得到的势能面。因此，力场的精度决定了模拟的准确性。

<http://jerkwin.github.io/2015/12/09/力场与拓扑之一-力场的概念分类及发展方法>



一维势能面：能量与氧原子距离的关系



力场简介

力场的精度决定了模拟的准确性。

AMBER力场函数形式

$$E(r^N) = E_{\text{bonds}} + E_{\text{angles}} + E_{\text{dihedrals}} + E_{\text{nonbonded}}$$

$$= \sum_{\text{bonds}} k_b (l - l_0)^2 + \sum_{\text{angles}} k_a (\theta - \theta_0)^2$$

$$+ \sum_{\text{torsions}} \sum_n \frac{1}{2} V_n [1 + \cos(n\omega - \gamma)]$$

$$+ \sum_{j=1}^{N-1} \sum_{i=j+1}^N f_{ij} \left\{ \epsilon_{ij} \left[\left(\frac{r_{0ij}}{r_{ij}} \right)^{12} - 2 \left(\frac{r_{0ij}}{r_{ij}} \right)^6 \right] + \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \right\}$$

范德华尔斯作用

静电作用

累加所有的键，角：谐振弹簧势用于计算键和角度的能量

累加所有的扭转二面角：傅里叶级数用于计算扭转二面角的能量

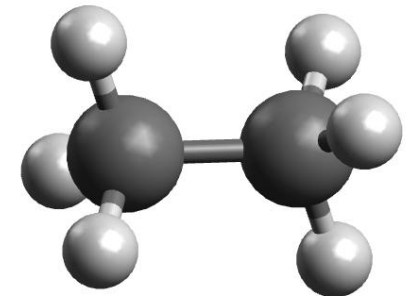
非键相互作用包含范德华尔斯作用和静电作用，只计算相隔3个以上键的原子对之间的非键作用

f_{ij} 为fudge factor迷糊因子，计算1-4原子对时， $f_{ij}=0.5$ （与扭转作用统一数量级），其他 $f_{ij}=1.0$

<https://blog.csdn.net/iSimuLy/article/details/82973050>

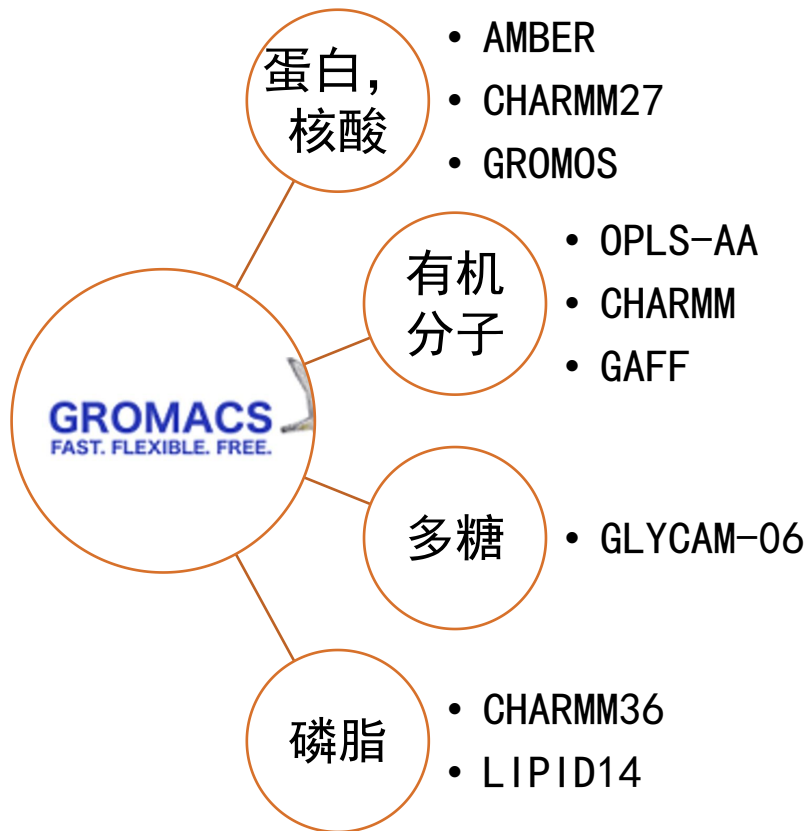
<https://en.wikipedia.org/wiki/AMBER>

<https://en.wikipedia.org/wiki/OPLS>



力场的选用

Gromacs在各个领域都有一席之地。



软件的选择

若Gromacs支持自己体系适合的力场，首选Gromacs，而其他软件如AmberTools等可用于建模。

如何选择合适的力场？

不同的力场都有自己适合的体系。

- 计算有机小分子热力学性质：Charmm >= OPLS-AA >= GAFF
- 模拟蛋白质：Amber (ff99SB+ildn+nmr) > CHARMM27 > GROMOS
- 模拟多糖：GLYCAM-06
- 模拟磷脂：CHARMM36 >= LIPID14

有趣的是，GAFF，GLYCAM-06，LIPID14与AMBER相互兼容，可以混用。

水模型

TIP3P是AMBER，GAFF专用水模型，也可用于CHARMM和OPLS-AA。

常用模型的搭建



材料
Cailiaoren.com



建模和
参数化

Credit to
Jicun Li
and [Gromacs
developers](#)

常用体系的模型搭建

获得体系的坐标文件是模拟的第一步。

有机分子

1. 使用Gaussview, Avogadro软件搭建分子, 另存为PDB结构
2. ChemOffice可以将分子式转为3D结构
3. 在线数据库 <https://pubchem.ncbi.nlm.nih.gov/>

多肽、蛋白

1. 从 <http://www.rcsb.org/> 下载多肽, 蛋白的结构
2. 使用PyMOL软件手动搭建短肽
3. 使用AmberTools的xLeap程序手动搭建

碳纳米管

1. 在线工具 <http://jerkwin.github.io/2015/04/13/碳纳米管的构建方法/>
2. Material Studio 的 Build - Build Nanostructure模块

多糖体系

1. 在线工具 <http://www.glycam.org/>
2. 使用AmberTools的xLeap程序手动搭建
<http://jerkwin.github.io/2017/04/03/Amber构建寡糖的方法/>

磷脂体系

1. 在线工具 CHARMM-GUI 搭建磷脂双分子层以及膜结合蛋白结构
2. 生成混合组分的磷脂双层膜结构的工具 genmixmem <http://sobereva.com/245>



常用体系的模型搭建

复杂体系的搭建。

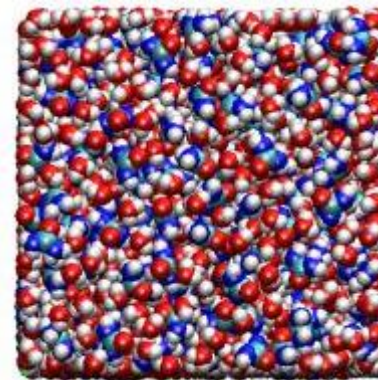
PACKMOL软件-堆积优化方法搭建分子模拟的初始构型

1. 搭建凝聚相体系
2. 搭建两相体系
3. 搭建混合磷脂体系
4. 搭建囊泡结构

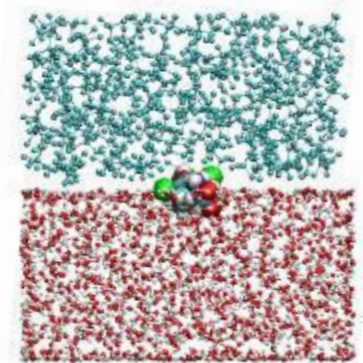
缺点：只考虑几何上可行，
不考虑化学因素

Gromacs子模块

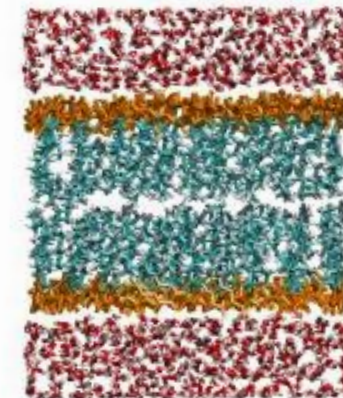
1. insert-molecules用于插入分子
2. editconf模块合并不同的结构实现复杂结构的搭建。



尿素-水混合溶液



水-四氯化碳界面



双磷脂层



双层囊泡

<http://jerkwin.github.io/2016/08/05/Packmol>程序资料整理/

<http://sobereva.com/473>

<http://jerkwin.github.io/2017/08/27/>拼接分子的简单脚本/

<http://jerkwin.github.io/GMX/GMXtut-4/>

搭建KCl水溶液的构型

搭建0.15M的KCl溶液。

传统搭建思路

1. 确定模拟盒子大小
2. 确定盒子内K⁺和Cl⁻离子和水的数目
3. 准备水分子, K⁺和Cl⁻离子的拓扑文件
4. 使用Packmol或者其他软件生成混合溶液。

繁琐, 需要手动
计算数目

巧妙搭建思路

1. 确定模拟盒子大小, 搭建一个空的盒子。
2. 用gromacs自带的水分子填充空盒子。
3. 使用pdb2gmx生成TIP3P的力场参数。
4. 生成一个tpr文件, 利用genion模块生成0.15M的KCL坐标文件和拓扑文件。

巧妙, 简便

```
echo -e "empty\n0\n3 3 3">
```

```
vac. gro
```

创建一个3nm*3nm*3nm的空盒子

```
gmx solvate -cp vac. gro -cs
```

```
spc216. gro -o solvate. gro
```

向空的盒子里填充平衡好的水分子

```
gmx pdb2gmx -f solvate. gro -o
```

```
water. gro -p water. top
```

给水盒子内的水分子赋予TIP3P或

其他水模型

```
gmx grompp -f em. mdp -c
```

```
water. gro -p water. top -o
```

```
water. tpr
```

生成预编译文件

```
gmx genion -s water. tpr -o
```

```
water-ion. gro -neutral -conc
```

```
0.15 -p water. top -pname K
```

向水盒子内自动添加0.15M的KCL离子,

并且自动更新拓扑文件。

<https://jerkwin.coding.me/GMX/GMXtutorial/#第五步-向盒子中填充溶剂及离子并进行能量最小化>

GROMACS
FAST. FLEXIBLE. FREE.



有机体系、膜蛋白体系的模型和 拓扑结构搭建

使用在线工具创建OPLS-AA, charmm27力场文件

主讲人：许楠
浙江大学博士研究生
VASPKIT合作开发者

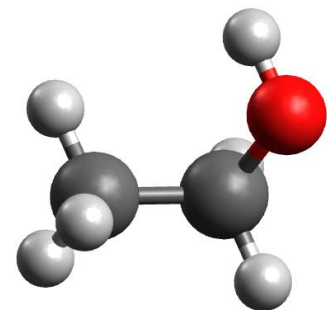
搭建70%乙醇水溶液的构型

使用在线工具生成OPLS-AA力场参数。

名称	力场	易用程度	判断准确性	形式
GMXTOP	OPLS-AA	容易	中	在线，可以自己部署
TPPmktop	OPLS-AA	容易	高	在线，可以安装离线版
LigParGen	OPLS-AA	难	中	在线

模型构建思路

1. 确定模拟盒子大小
2. 确定盒子内乙醇和水分子的数目
3. 使用建模软件搭建乙醇的坐标文件
4. 使用在线工具生成乙醇的拓扑文件
5. 使用Packmol或者GMX模块生成混合溶液的构型gro文件，并整合为总拓扑top文件。



<http://sobereva.com/266>

<http://erg.biophys.msu.ru/tpp/>

<https://jerkwin.github.io/2017/06/29/GMXTOP-OPLSAA力场的GROMACS拓扑文件生成器/>

搭建70%乙醇水溶液的构型

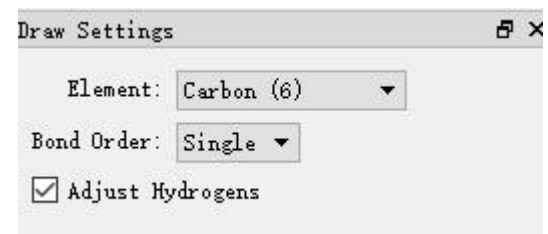
确定水和乙醇的数目。

1. 查化工化学物性手册可知，25°C下70%质量分数的乙醇水溶液密度为 0.86340g/cm^3 。
2. 盒子大小使用30Å，也就是3nm的立方体。
3. 计算盒子内乙醇和水的分子数比例。
4. 确定在3nm的立方体盒子中添加213个乙醇分子，再添加233个水分子（或者用solvate模块自动溶剂化）
5. 使用Avogadro或其他软件生成乙醇的PDB坐标文件。

画笔模式在显示框点击出现一个甲烷分子，再在甲烷的氢上点击就会取代氢生成乙烷分子，Element换成O，取代另外一个氢，就会得到乙醇分子，此时结构不合理，点击菜单Extensions—Optimize Geometry，再另存为PDB。



30Å³的立方盒子，密度为 0.86340g/cm^3 ，因此盒子内分子的总质量为 $30 \times 30 \times 30 \times 10^{-24}\text{cm}^3 \times 0.86340\text{g/cm}^3 = 2.33118 \times 10^{-20}\text{g}$ 。已知乙醇的分子量46.07，水为18.01，该盒子模拟的是质量分数为70%的乙醇水溶液，因此乙醇分子所占的质量为 $2.33118 \times 10^{-20} \times 0.7 = 1.63182 \times 10^{-20}\text{g}$ ，盒子中乙醇的数目为 $1.63182 \times 10^{-20} / 46.07 \times 6.022 \times 10^{23} \approx 213$ ，水的数目约为233个。



搭建70%乙醇水溶液的构型

只需生成乙醇的拓扑，水模型有现成的。

1. 使用 [TPPrenum](#) 在线工具将建模工具生成的非标准PDB转成适用于模拟软件的PDB。
2. 使用 [TPPmktop](#) 在线工具读取处理后的PDB生成拓扑文件itp和残基拓扑文件rtp以及**缺失的参数集合lack.itp**。将他们都下载到本地。生成的残基名为LIG，我们可以命名为ETH（3个字母）以显示残基的名字。
3. 因为TPPmktop使用的扩展的OPLS-AA，需要使用它提供的OPLS-AA力场参数代替原来的OPLS-AA参数。（例子中已提供扩展的oplsaa.ff和new_opls）



材料
Cailiaoren.com

Project directory: ethanolpdb_T4dw5H

Executed command:

Returned:

Input PDB file: [input000.pdb](#) [1.6 kB]

Output PDB file: [output000.pdb](#) [0.8 kB]

tpprenum verbose LOG: [tpprenum.log](#) [3.2 kB]

tpprenum console OUTPUT: [pytp tprenum](#) [4.0 kB]

[main page](#)

TITLE	Written by	TPP:	(topo
ATOM	1 C1	LIG A	1
ATOM	2 1H1	LIG A	1
ATOM	3 2H1	LIG A	1
ATOM	4 3H1	LIG A	1
ATOM	5 C2	LIG A	1
ATOM	6 1H2	LIG A	1
ATOM	7 2H2	LIG A	1
ATOM	8 O3	LIG A	1
ATOM	9 1H3	LIG A	1
END			

将itp和atp中残基名由LIG手动改为ETH(3个字母)以区分

<https://jerkwin.github.io/2017/09/14/GROMACS非标准残基教程1-修改力场与增加残基/>

<https://jerkwin.github.io/2015/12/13/TPPMKTOP-OPLS-AA全原子力场的GROMACS拓扑文件生成器/>

搭建70%乙醇水溶液的构型



创建系统的坐标和拓扑文件。

将new_opls/residuetypes.dat里添加ETH MOL，也就是将ETH视为MOL残基，用于pdb2gmx模块，并将TPPmktop生成所有的rtp文件组合成MOL.rtp，置于oplsaa.ff中

```
export GMXLIB=`pwd`/new_opls
```

将new_opls文件夹设为GMX的力场参数库

```
gmx_mpi pdb2gmx -f ethanol.pdb -ff oplsaa
```

使用TPPmktop提供的扩展的OPLS-AA力场参数文件将PDB转成GMX的坐标文件，对于水模型选择TIP3P水模型//亦可以省略这一步，直接用PDB，grompp以top文件为准

```
gmx_mpi insert-molecules -ci conf.gro -nmol 213 -box 3 3 3 -o box.gro
```

在3nm的立方盒子里插入213个乙醇分子

需要处理系统的topol.top，引入TPPmktop生成的itp文件替换自动生成的乙醇分子拓扑，并去除宏语句#ifdef TPPREV_jun16-#endif

```
gmx_mpi solvate -cp box.gro -cs spc216.gro -o solvate.gro -p topol.top
```

对乙醇盒子进行溶剂化，实际插入236个水分子

```
gmx_mpi grompp -f EM.mdp -c solvate.gro -p topol.top -o EM.tpr
```

```
gmx_mpi mdrun -v -deffnm EM
```

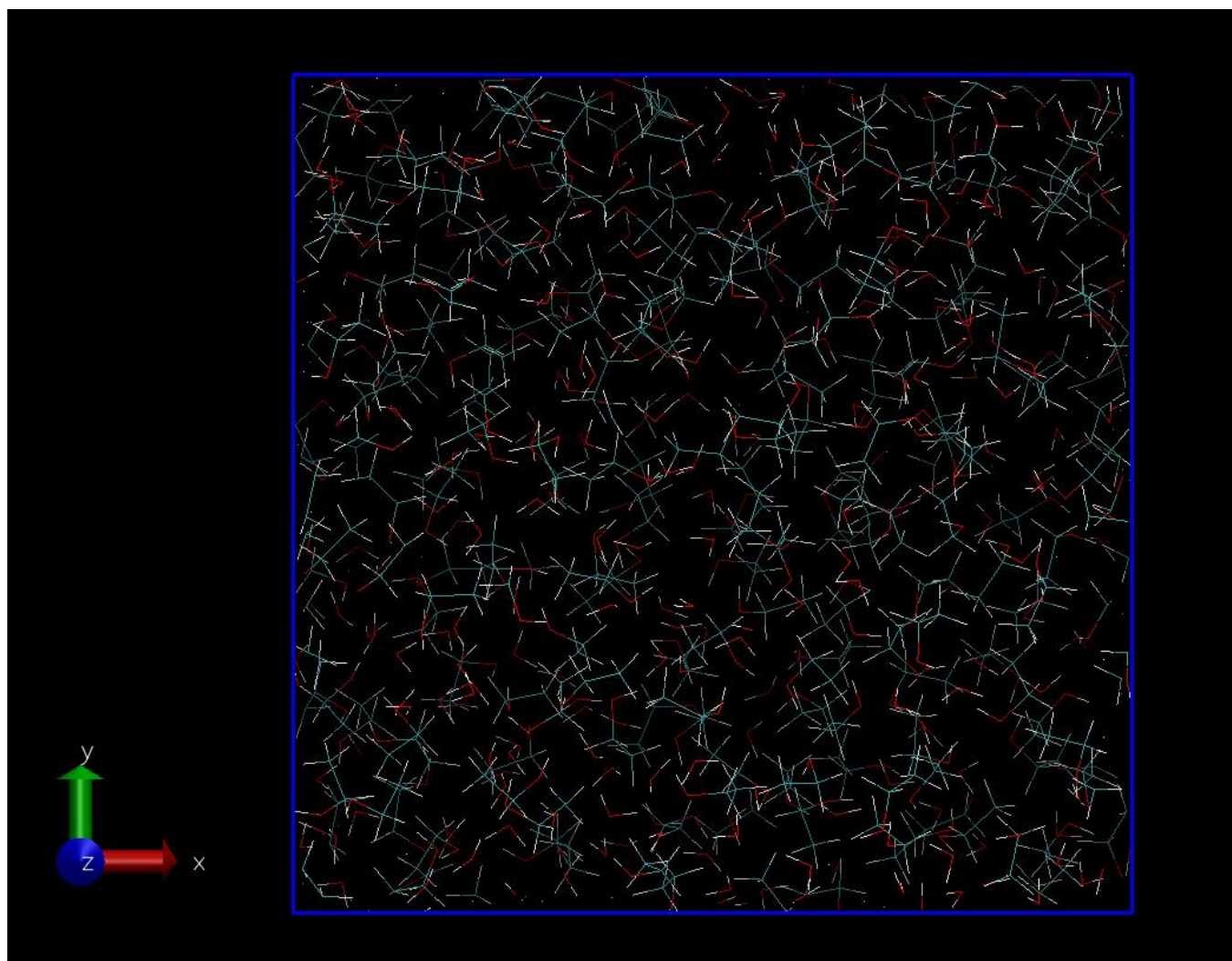
运行能量最小化任务

搭建70%乙醇水溶液的构型



材料
Cailiaoren.com

使用VMD查看gro文件。



Gromacs拓扑文件的组成

拓扑文件中如何体现不同的力场？

1. 查看乙醇水溶液体系的topol.top，;开头的行为注释行，grompp预编译时会忽略，#include “./oplsaa.ff/forcefield.itp” 为引入oplsaa的力场库
2. 程序会自动定义defaults（从ffoplsaa.itp引入）和atomtypes块（从ffnonbonded.itp引入）块。

```
[ defaults ] 定义了非键函数类型，非键参数混合方法以及模糊因子等
; nbfunc      comb-rule    gen-pairs    fudgeLJ fudgeQQ
1             3            yes           0.5 0.5
```

```
[ atomtypes ] 定义了原子类型，质量
; type      mass      charge ptype sigma(nm) epsilon(kJ/mol)
N      14.007      0.794    A     0.315     0.71128      ;
O      15.999     -0.598    A     0.286     0.87864      ;
```

```
#include "eth.itp"
```

引入小分子模板

```
[ system ]
; Name      系统名称
ethanol in water
```

```
[ molecules ]
; Compound      #mols
ETH             213
```

系统包含的小分子的数目
ETH是eth.ito里定义的乙醇分子的残基名称

Gromacs拓扑文件的组成

拓扑文件中如何体现不同的力场？

小分子力场N03. itp参数文件

选用的OPLS力场就决定了非键，bonds，angles，dihedrals的函数类型

```
[ moleculetype ]
; name nrexcl
N03 3
```

[moleculetype]: 定义小分子的名称，nrexcl=3代表排除不远于三条键的原子之间的非键相互作用。

```
[ atoms ]
; nr type resnr residu atom cgnr charge mass
1 N 1 N03 N 1 0.794 14.007
2 O 1 N03 O 1 -0.598 15.999
3 O 1 N03 O 1 -0.598 15.999
4 O 1 N03 O 1 -0.598 15.999
```

原子类型；残基编号；残基名称；
原子名称；电荷组编号；电荷q(e)；
质量m(u)。

```
[ bonds ]
; ai aj funct c0(nm) c1(kJ mol-1 nm-2)
1 2 1 0.1256 530698.56
```

成键原子1序号，成键原子2序号，
键的函数形式，1代表简谐势

```
[ angles ]
; ai aj ak funct c0(deg) c1(kJ mol-1 rad-2)
2 1 3 1 120 1011.022
```

构成角的原子1序号，构成角的原子2序号，构成角的原子3序号，角的函数形式，1代表简谐势

```
[ dihedrals ]
; ai aj ak al funct c0 c1 c2 c3(kJ/mol)
2 1 3 4 5 0.000 8.368 0.000 0.000
```

5代表傅立叶二面角

<http://jerkwin.github.io/GMX/GMXman-5/#553-键参数>

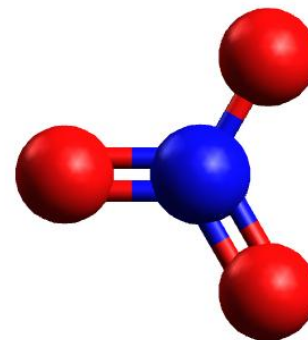
<http://manual.gromacs.org/documentation/>

搭建Cu(NO₃)₂水溶液的构型



如何处理缺失的参数？

1. 用Avogadro软件绘制NO₃⁻的模型并进行优化。
2. 使用TPPmktop生成NO₃的力场参数文件。
3. 发现现有的OPLS-AA库里有部分参数残缺
lack.itp, 尽管数据库里有原子类型, 但是
没有N-O, O-N-O的键参数, 需要手动补充。



```
[ atoms ]
1  op1s_788    1  N03    01      1  -0.598    15.999400 ;
2  op1s_787    1  N03    N2      2   +0.794    14.006700 ;
3  op1s_788    1  N03    03      3  -0.598    15.999400 ;
4  op1s_788    1  N03    04      4  -0.598    15.999400 ;

[ bonds ]
1  2  dfTPP_bon_0_N_-1
2  3  dfTPP_bon_0_N_-1
2  4  dfTPP_bon_0_N_-1

[ angles ]
1  2  3  dfTPP_ang_0_N_0_-1
1  2  4  dfTPP_ang_0_N_0_-1
3  2  4  dfTPP_ang_0_N_0_-1
```

Project directory: NO3pdb_aBVPpr

Force field: OPLS-AA

Executed command:

Returned:

Input PDB file: [input000.pdb](#) [0.4 kB]

Output ITP file: [output000.itp](#) [1.4 kB]

Output RTP file: [output000.rtp](#) [1.0 kB]

Parameter lack-file: [lack.itp](#) [0.1 kB]

tppmktop verbose LOG: [nop](#) [kB]

tppmktop console OUTPUT: [pytpp_tppmktop](#) [32.6 kB]

[main page](#)

```
; TPP topology lack
#define dfTPP_bon_0_N_-1 1 0.00 0.00 ;1
#define dfTPP_ang_0_N_0_-1 1 0.00 0.00 0.00 ;2
```

搭建Cu (NO₃)₂水溶液的构型



如何处理缺失的参数？

1. 通过查询文献，找到了NO₃的力场参数。同时，NO₃中存在N=O双键，存在非正常二面角 (The gromacs documentation (link below) says that the improper dihedral "keeps planar groups (e.g. aromatics rings) planar, or prevents them from flipping over to their mirror images").) 因此还需要手动补充非正常二面角。
2. 文献里给出的单位与GROMACS力场的单位不同。

项目	文献		GMX	
vDw	sigma (Å)	epsilon (kcal/mol)	sigma (nm)	epsilon (kJ/mol)
bond	ro (Å)	kr (kcal mol ⁻¹ Å ⁻²)	ro (nm)	kr (kJ mol ⁻¹ nm ⁻²)
angle	c0 (deg)	ka (kcal mol ⁻¹ rad ⁻²)	c0 (deg)	ka (kJ mol ⁻¹ rad ⁻²)
dihedral	Vx (kcal/mol)		Vx (kJ/mol)	

[hDoherty B , Zhong X , Gathiaka S , et al. Journal of Chemical Theory and Computation, 2017, 13\(12\):6131-6145.](#)

搭建Cu(NO₃)₂水溶液的构型



材料
Cailiaoren.com

如何处理缺失的参数？

```
;#ifndef TPPREV_jun16  
; #error CORRECT VERSION OF FORCE FIELD [TPPREV_jun  
;#endif
```

```
#define dfTPP_bon_O_N -1 1 0.1256      530698.56 ;1  
#define dfTPP_ang_O_N_O -1 1 120      1011.022 ;2
```

去除NO₃. itp中无用的宏语句，将lack.itp补全后粘贴到前面（宏要在有效语句之前）

```
,  
[ dihedrals ]  
; ai aj ak al funct c0 c1 c2 c3(kJ/mol) ;  
1 2 3 4 5 0.000 8.368 0.000 0.000 ;  
3 2 1 4 5 0.000 8.368 0.000 0.000 ;  
4 2 1 3 5 0.000 8.368 0.000 0.000 ;
```

增加三个非正常二面角（O-N-O-O）

[hDoherty B , Zhong X , Gathiaka S , et al. Journal of Chemical Theory and Computation, 2017, 13\(12\):6131-6145.](#)

搭建Cu(NO₃)₂水溶液的构型



如何处理缺失的参数？

对于Cu²⁺，oplsaa.ff中ions.itp有Cu离子的参数，因此只要在系统的拓扑文件中使用CU这个残基名就行。

```
[ moleculetype ]  
; molname      nrexcl  
CU              1
```

```
gmx_mpi insert-molecules -ci N03.pdb -nmol 20 -box 3 3 3 -o box.gro
```

在3nm的立方盒子里插入20个N03-离子

```
gmx_mpi insert-molecules -ci Cu.pdb -nmol 10 -f box.gro -o new_box.gro
```

再往盒子里插入10个Cu原子

手动创建一个系统的拓扑文件，包含我们小分子的itp

```
gmx_mpi grompp -f EM.mdp -c new_box.gro -p topol.top -o EM.tpr -maxwarn 1
```

```
gmx_mpi mdrun -v -deffnm EM
```

运行能量最小化任务，-maxwarn 1 预防出现top文件中的残基名与gro中不一样的警告。

```
; Include forcefield parameters 引入OPLS库
```

```
#include "./oplsaa.ff/forcefield.itp"
```

```
#include "N03.itp" 引入N03的力场
```

```
; Include water topology
```

```
#include "./oplsaa.ff/tip3p.itp" 引入TIP3P水力场
```

```
; Include topology for ions
```

```
#include "./oplsaa.ff/ions.itp" 引入离子力场
```

```
[ molecules ]  
; Compound      #mols  
N03              20  
CU               10
```

在系统的拓扑文件中包含20 个N03
和10个CU，得与itp中的残基名对应

搭建Sn (NO₃)₂水溶液的构型



材料
Cailiaoren.com

如何处理缺失的参数？

1. 如何添加OPLS-AA库中不存在的原子类型？比如Sn²⁺离子？类似于处理NO₃离子一样，我们需要手动建立itp文件，但是NO₃的每个原子类型都能在OPLS-AA数据库中找到，因此不需要指定原子的vDw参数（sigma, epsilon）。而Sn²⁺需要自己定义非键参数。
2. 创建一个新文件ffnonbond-buchong.itp，里面加入新添加的Sn²⁺离子的非键参数，并且需要在定义atomtypes块时（也就是在紧接着#include “./oplsaa.ff/forcefield.itp” 的语句后面）添加Sn²⁺的vDw参数。此例中Sn的力场用Cu的力场代替。

```
[ atomtypes ]  
Sn2+      Sn2+    29    63.54600    2.000    A    2.08470e-01    4.76976e+00
```

3. 创建一个Sn.itp，里面加入Sn²⁺离子的残基定义，此例中Sn的参数复制于Cu（ions.itp）。

```
[ moleculetype ]  
; molname      nrexcl  
SN              1
```

Moleculetype中定义了残基名

```
[ atoms ]  
; id  at type res nr  residu name at name  cg nr  charge  mass  
1     Sn2+    1      SN      Sn      1      2      63.54600
```

原子类型，在GRO中会体现，重要

搭建Sn (NO₃)₂水溶液的构型

如何处理缺失的参数？

在系统的拓扑文件中引入Sn的力场。

```
; Include forcefield parameters 引入NO3的力场
#include "./oplsaa.ff/forcefield.itp"
#include "ffnonbond-buchong.itp"
#include "N03.itp"
#include "Sn.itp" 引入Sn离子的力场
; Include water topology
#include "./oplsaa.ff/tip3p.itp"
; Include topology for ions
#include "./oplsaa.ff/ions.itp"
```

需要在引入具体分子itp之前引入补充的非键参数

```
[ molecules ]
; Compound          #mo
N03                  20
SN                    10
```

系统的拓扑文件中20个SN，
需要与Sn.itp中
MoleculeType中的残基名
一致

```
gmx_mpi insert-molecules -ci N03.pdb -nmol 20 -box 3 3 3 -o box.gro
```

在3nm的立方盒子里插入20个NO₃⁻离子

```
gmx_mpi insert-molecules -ci Cu.pdb -nmol 10 -f box.gro -o new_box.gro
```

再往盒子里插入10个Cu原子，在创立tpr文件时会使用topol中定义的原子类型Sn替换
手动创建一个系统的拓扑文件，包含我们小分子的itp

```
gmx_mpi grompp -f EM.mdp -c new_box.gro -p topol.top -o EM.tpr -maxwarn 1
```

```
gmx_mpi mdrun -v -deffnm EM
```

运行能量最小化任务，-maxwarn 1 预防出现top文件中的残基名与gro中不一样的警告。

搭建Sn (NO₃)₂水溶液的构型



材料
Cailiaoren.com

如何处理缺失的参数？

19NO3	O4	76	0.825	2.012	0.133
20NO3	O1	77	0.441	2.257	2.844
20NO3	N2	78	0.331	2.209	2.877
20NO3	O3	79	0.276	2.286	2.958
20NO3	O4	80	0.347	2.098	2.929
21SN	Sn	81	2.036	1.271	0.696
22SN	Sn	82	1.184	0.183	1.364
23SN	Sn	83	0.293	1.103	1.489
24SN	Sn	84	1.653	1.600	1.778
25SN	Sn	85	2.541	2.312	2.322
26SN	Sn	86	0.584	1.138	2.944
27SN	Sn	87	0.014	1.478	1.132
28SN	Sn	88	0.468	2.232	0.063
29SN	Sn	89	2.411	2.699	0.711
30SN	Sn	90	1.634	1.309	1.112
3.00000	3.00000	3.00000			

残基名称，原子类型已经正确替换，以拓扑文件中定义的为准。

能量最小化后的GRO

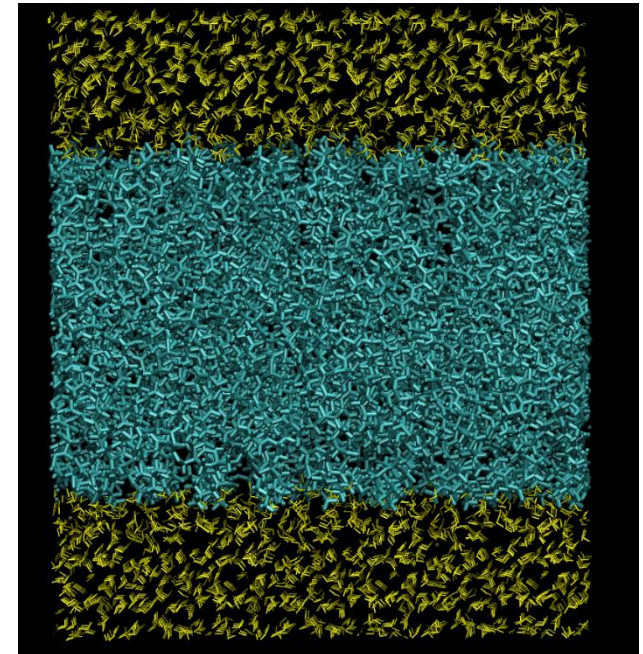
膜蛋白体系的构建

使用CHARMM-GUI生成膜。

1. [教程](#)里主要介绍了使用LIPID14力场模拟DOPC双分子膜。先用CHARMM-GUI在线生成PDB结构,再用charmm lipid2amber.py脚本转成AMBER可以识别的PDB,再用AmberTools工具对其进行参数化。实质上使用的还是LIPID14力场。
2. 在第五步时,选择生成GROMACS的输入文件
3. 下载压缩包,进入gromacs文件夹,进行能量最小化

```
gmx_mpi grompp -f step6.0_minimization.mdp
-c step5_charmm2gmx.pdb -p topol.top -o
mem.tpr -r step5_charmm2gmx.pdb
```

```
gmx_mpi mdrun -v -deffnm mem
```



Input Generation Options:

It may take some time to generate selected inputs.

- ☐ NAMD
- ☒ GROMACS
- ☐ AMBER
- ☐ OpenMM
- ☐ CHARMM/OpenMM
- ☐ GENESIS
- ☐ Desmond
- ☐ LAMMPS

<https://jerkwin.github.io/2017/09/14/GROMACS非标准残基教程>

<https://jerkwin.github.io/2018/02/06/AMBER高级教程A16-Amber脂分子教程-Lipid14力场版本/>

GROMACS
FAST. FLEXIBLE. FREE.



生物蛋白体系模型和拓扑结构 搭建

以AMBER力场为例讲解非标准残基的处理

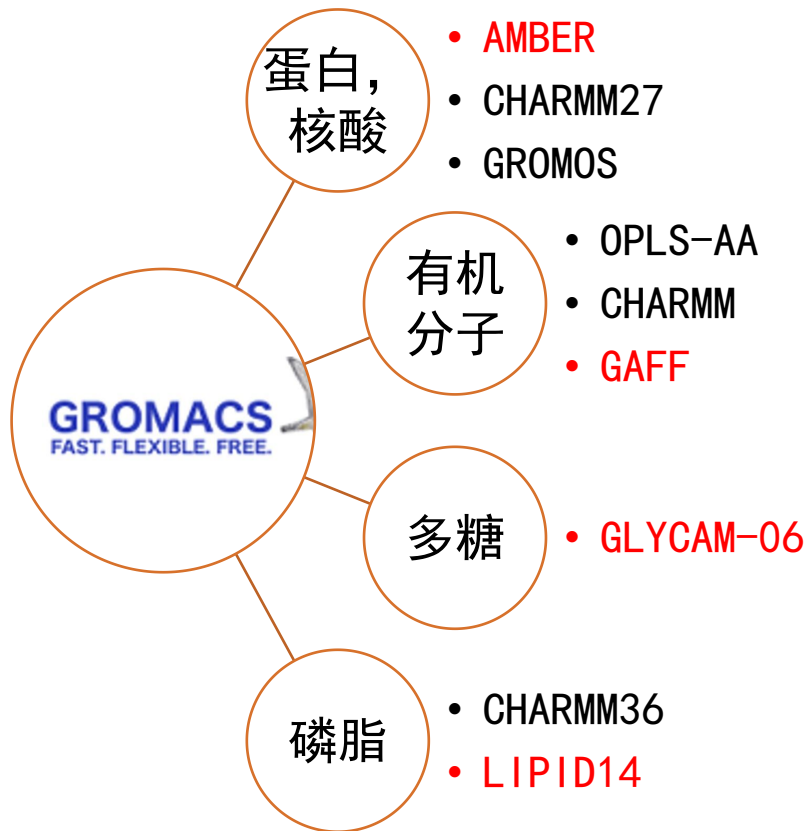
主讲人：许楠

浙江大学博士研究生

VASPKIT合作开发者

AMBER系列力场

AMBER系列力场覆盖了各个领域。



AMBER力场系列相互兼容

- 模拟有机小分子：GAFF足够了
- 模拟蛋白质：Amber (ff99SB+ildn+nmr)
- 模拟多糖：GLYCAM-06
- 模拟磷脂：LIPID14

GAFF, GLYCAM-06, LIPID14与AMBER相互兼容，可以混用。

使用AmberTools建模+GROMACS跑动力学
使用Anaconda环境的conda工具在线安装：
<https://blog.csdn.net/u013069552/article/details/78349716>

```
conda install ambertools=19 -c  
ambermd
```

使用ACPYE.py将AMBER格式的拓扑和坐标转成GMX格式。

<https://jerkwin.coding.me/2015/12/10/力场与拓扑之二-如何选择力场>

<https://jerkwin.coding.me/2015/12/08/使用AmberTools+ACPYE+Gaussian创建小分子GAFF力场的拓扑文件/>

回顾OPLS力场参数化

小分子坐标搭建，拓扑生成，系统构型的生成

乙醇水溶液

1. 搭建小分子模型
2. 使用TPPmktop生成小分子模板
3. 将乙醇分子添加到残基数据库中

1. 使用pdb2gmx将生成含有一个乙醇分子的拓扑和坐标文件
2. 使用gmx insert-molecules 得到乙醇水溶液构型

$\text{Cu}(\text{NO}_3)_2$ 水溶液

1. 搭建小分子模型
2. 使用TPPmktop生成小分子模板
3. 补充缺失的键角，二面角参数

1. 使用gmx insert-molecules或者packmol生成系统构型，
2. 手动编辑拓扑文件，引入小分子的itp文件

$\text{Sn}(\text{NO}_3)_2$ 水溶液

1. 搭建小分子模型
2. 使用TPPmktop生成小分子模板
3. 补充缺失的元素类型，补充非键参数

1. 使用gmx insert-molecules或者packmol生成系统构型，
2. 手动编辑拓扑文件，引入小分子的itp文件

多肽和蛋白的参数化

蛋白的力场是最成熟的。

1. 获取PDB结构，从Protein Data Bank下载小肽的pdb文件10MB. PDB
2. 对PDB文件进行处理。
 - 氢原子（经常命名与力场中的不一致）
 - C端氧原子（经常缺失封端残基）
 - 结晶水（一般去除，但是与活性位点结合的水分子不能去除）
 - 缺失原子/残基/侧链（用软件补充）
 - 二硫键（pdb4amber可以自动识别）
 - 带电残基
3. 用pdb2gmx或者AmberTools进行力场参数化
 - pdb2gmx 可以手动选择忽略H，但是若残基缺少原子将运行失败。
 - tleap（AmberTools工具集之一）可以自动补全缺失的原子，自动添加封端残基。



漏斗网蜘蛛毒素肽10MB. pdb

1. 10MB. pdb 缺少C端的封端氧原子OXT
2. 末端氨基酸残基缺少O原子
3. 氢原子的命名不一致

<http://jerkwin.github.io/GMX/GMXtut-0/#第一步:获取并处理pdb文件>

https://github.com/tamaswells/VASP_script/blob/master/remove_H.py

多肽和蛋白的参数化

标准蛋白残基的处理。

- ✓ 方法1：使用DeepView软件进行处理，去除氢原子，补充缺失的氧原子，添加C端的封端氧原子OXT，得到适合pdb2gmx程序的PDB文件。
(操作麻烦，不够智能)
- ✓ 方法2：利用pdb4amber (AmberTools工具集之一) 去除原有PDB中的H原子和结晶水，并会自动为你检查靠的过近的原子，缺失的参数等。再利用AmberTools强大的tleap工具自动补全缺失的原子，自动添加封端残基。
(功能最完整，推荐)
- ✓ 方法3：利用自写程序[remove H. py](#)去除原有PDB中的H原子和结晶水，利用AmberTools强大的tleap工具自动补全缺失的原子，自动添加封端残基。(无法自动处理双硫键等)

--dry选项移除结构中的结晶水，-y 选项去除PDB得到不含H原子的new.pdb。

`pdb4amber -i 10MB.pdb -o new.pdb --dry -y`

```
----- Histidines (Renumbered Residues!)
No histidine residues found.
#检查是否有组氨酸
----- Cysteines in Disulfide Bonds (Renumbered Residue
CYS_1 - CYS_17: S-S distance = 2.020711 Ang.
CYS_9 - CYS_22: S-S distance = 2.019490 Ang.
CYS_16 - CYS_33: S-S distance = 2.021138 Ang.
CYS_24 - CYS_31: S-S distance = 2.020204 Ang.
The above CYS have been renamed to CYX in the new PDB file.
Disulfide bond CONECT cards have been added to the new PDB
#自动检测二硫键，并将含有二硫键的CYS残基改名为CYX
----- Missing Heavy Atoms (Renumbered Residues!)
PRO_35 misses 1 heavy atom(s)
#检测到末端脯氨酸缺少一个O原子
```

<http://jerkwin.github.io/GMX/GMXtut-0/#第一步:获取并处理pdb文件>

<http://jerkwin.github.io/2018/01/07/Amber基础教程B5-模拟绿色荧光蛋白及构建修饰的氨基酸残基/>

https://github.com/tamaswell/GMX/blob/master/remove_H.py

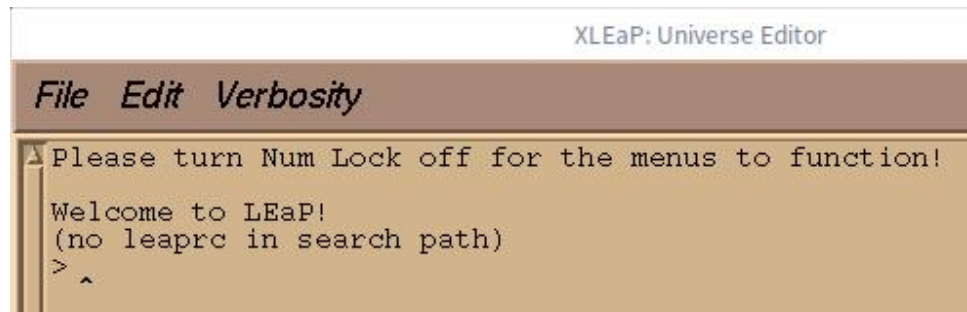
LEaP工具简介

坐标格式转换，力场参数化的强大工具。

```
tamas@tamas-PC:~/Desktop$ tleap
-I: Adding /home/tamas/Downloads/Anconda3/bin/conda
-I: Adding /home/tamas/Downloads/Anconda3/bin/conda
-I: Adding /home/tamas/Downloads/Anconda3/bin/conda
-I: Adding /home/tamas/Downloads/Anconda3/bin/conda

Welcome to LEaP!
(no leaprc in search path)
> source leaprc.protein.ff14SB
```

tleap: 命令行版本



XLEaP: Universe Editor

File Edit Verbosity

Please turn Num Lock off for the menus to function!

Welcome to LEaP!
(no leaprc in search path)
> ^

xleap: 图形化版本

```
tamas@tamas-PC:~/Downloads/Anconda3/dat/leap/cmd$ ls
leaprc.conste      leaprc.lipid17      leaprc.RNA.ROC
leaprc.constph     leaprc.modrna08     leaprc.RNA.YIL
leaprc.DNA.bsc1    leaprc.music        leaprc.water.fb3
leaprc.DNA.OL15    leaprc.phosaa10     leaprc.water.fb4
leaprc.ff14SB.redq leaprc.protein.fb15  leaprc.water.opc
leaprc.ffAM1       leaprc.protein.ff03.r1 leaprc.water.spce
leaprc.ffPM3       leaprc.protein.ff07 leaprc.water.spceb
leaprc.gaff        leaprc.protein.ff14SB leaprc.water.tip3p
leaprc.gaff2       leaprc.protein.ff14SBonlysc leaprc.water.tip4pew
leaprc.GLYCAM_06EPb leaprc.protein.ff15ipq leaprc.xFPchromophores
leaprc.GLYCAM_06j-1 leaprc.protein.ff15ipq-vac oldff
leaprc.lipid14     leaprc.RNA.OL3
```

GAFF力场

多糖力场

磷脂力场

蛋白质力场

水力场

核酸力场

<http://ambermd.org/tutorials/pengfei/index.php>

使用tleap参数化标准多肽

坐标格式转换，力场参数化的强大工具。

```
tamas@tamas-PC: ~/Desktop$ tleap
> source leaprc.protein.ff14SB 加载蛋白的力场
> 10MB= loadpdb new.pdb 加载我们处理后的没有氢原子，无结晶水的PDB文件，tleap会自动赋予力场
Loading PDB file: ./new.pdb 如果直接加载10MB.pdb，氢原子无法识别会出错，因此需要去氢
Added missing heavy atom: .R<CPRO 35>.A<O 14> 自动添加缺失的氧原子
Added missing heavy atom: .R<CPRO 35>.A<OXT 15> 自动再C端添加封端氧原子OXT
total atoms in file: 255
Leap added 240 missing atoms according to residue templates:
    2 Heavy
    238 H / lone pairs 自动根据力场模板文件在合适的位置添加氢原子
> check 10MB 检查残基是否有力场参数缺失
Checking '10MB'....
Warning: The unperturbed charge of the unit (2.000000) is not zero. 带2单位电荷
Checking parameters for unit '10MB'.
Checking for bond parameters.
Checking for angle parameters.
check: Warnings: 1
Unit is OK. 力场参数齐全10MB
> saveoff 10MB 10MB.lib 创建10MB的残基力场库文件
> savePdb 10MB 10MB.pdb 导出标准的,符合amber力场规则的pdb文件，因为此PDB只包含标准的氨基酸残基，此时可以使用pdb2gmx程序进行参数化: gmx pdb2gmx -ignh -ff amber99sb-ildn -f 10MB.pdb
```

AMBER to Gromacs

将AMBER力场和坐标文件转换为GMX支持的格式。

> check 10MB 检查残基是否有力场参数缺失

Checking '10MB'....

Unit is OK. 力场参数齐全10MB

> setBOX 10MB vdw 5 给10MB创建一个“包裹”整个分子的立方盒子，分子到盒子的边界距离至少5埃。不推荐使用amber进行溶剂化和离子平衡，导出后可能会出现水分子重叠的问题。所以只需创建一个空盒子。

> saveamberparm 10MB 10MB.prmtop 10MB.inpcrd 导出Amber格式的拓扑文件和坐标文件，类似于GMX的top和gro文件。

tamas@tamas-PC:~/Desktop\$ python acpype.py -p 10MB.prmtop -x 10MB.inpcrd -d
acpype.py将Amber格式的拓扑文件和坐标文件转换为GMX格式的top和gro，并生成能量最小化em.mdp文件。此时分子带有两个单位的正电荷，并且没有溶剂化，需要在GMX中使用水分子溶剂化并进行平衡电荷。查看此top文件后发现，所有涉及到的AMBER力场参数（非键和成键）均包含在了top文件中，但是并没有引入水分子和离子的原子类型以及力场参数等，因此不能直接使用GMX溶剂化功能，需要手动编辑top文件。根据上一节课讲的内容，把所有用到原子类型的非键参数需要放到一起，因此在定义完了[atomtypes]后紧接着（也就是在第一个[moleculetype]前的空白位置）引入离子和水的原子类型以及力场参数：

```
#include "sup_atomtypes.itp" ; Additional non-bond parameters for water and ions  
; Include topology for water and ions  
#include "amber99sb-ildn.ff/tip3p.itp" ; TIP3P water model compatible with AMBER  
#include "amber99sb-ildn.ff/ions.itp" ; ions compatible with TIP3P water model
```

<https://github.com/t-/acpype>

<https://jerkwin.coding.me/2015/12/08/使用AmberTools+ACPYPE+Gaussian创建小分子GAFF力场的拓扑文件/>

使用GMX进行溶剂化

不推荐使用AMBER溶剂化，导出到GMX后再溶剂化。

```
tamas@tamas-PC: ~/Desktop$ gmx solvate -cp 10MB_GMX.gro -cs spc216.gro -o
solvate.gro -p 10MB_GMX.top
```

向模拟盒子里填充平衡好的水分子，-p参数能够自动在top里添加水分子的数目

```
tamas@tamas-PC: ~/Desktop$ gmx grompp -f em.mdp -c solvate.gro -p 10MB_GMX.top
-o 10MB_SOL.tpr
```

参考第一节的内容，生成二进制tpr用于添加离子

```
tamas@tamas-PC: ~/Desktop$ gmx genion -s 10MB_SOL.tpr -o 10MB_SOL_ion.gro -
neutral -conc 0.15 -p 10MB_GMX.top
```

替换水分子，添加5个Na和7个Cl平衡2个单位的正电荷

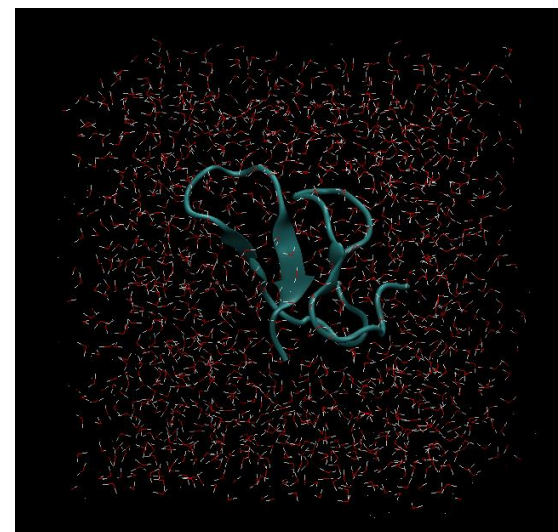
```
tamas@tamas-PC: ~/Desktop$ gmx grompp -f em.mdp -c 10MB_SOL_ion.gro -p
10MB_GMX.top -o em.tpr 对体系进行能量最小化
```

```
tamas@tamas-PC: ~/Desktop$ gmx mdrun -v -deffnm em
```

```
[ atomtypes ]
;name    bond_type    mass    charge    ptype    sigma    epsilon    Amb
N3        N3          0.00000  0.00000  A        3.25000e-01  7.11280e-01 ; 1.82  0.1700
H         H           0.00000  0.00000  A        1.06908e-01  6.56888e-02 ; 0.60  0.0157
CX        CX          0.00000  0.00000  A        3.39967e-01  4.57730e-01 ; 1.91  0.1094

#include "sup_atomtypes.itp" ; Additional non-bond parameters for water and ions
; Include topology for water and ions
#include "amber99sb-ildn.ff/tip3p.itp" ; TIP3P water model compatible with AMBER
#include "amber99sb-ildn.ff/ions.itp" ; ions compatible with TIP3P water model

[ moleculetype ]
;name      nrexcl
10MB       3
```



LEaP参数化非标准残基



材料
Cailiaoren.com

HIV逆转录酶(HIV-RT)抑制剂Sustiva与HIV-RT复合物的建模。

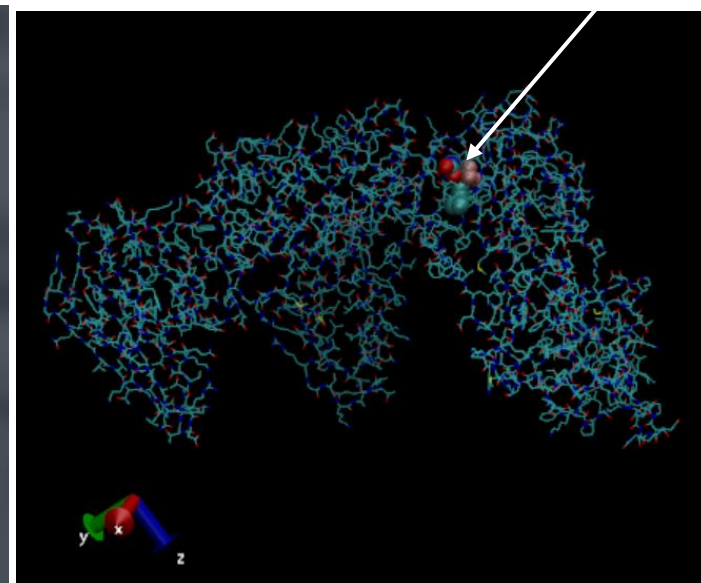
Sustiva

```
----- Non-Standard Residues
The following non-standard residue names in the original PDB file
are not recognized by Amber and have been written to the separate
file 1fko_amber_water_nonprot.pdb
CSD
EFZ

----- Histidines (Renumbered Residues!)
It was not possible to determine the protonation state of the following HIS
residues based on presence of hydrogens. Amber will consider them as HIE
(epsilon-HIS) by default. If other protonation state desired change to HID
(delta-HIS) or HIP (protonated HIS) by hand.
HIS_96
HIS_198
HIS_208
HIS_221
HIS_235
HIS_315
HIS_361
HIS_539

----- Cysteines in Disulfide Bonds (Renumbered Residues!)
No disulfide bonds have been detected.
```

组氨酸的处理需小心，
质子化的位置需要考虑



检查1fko.pdb

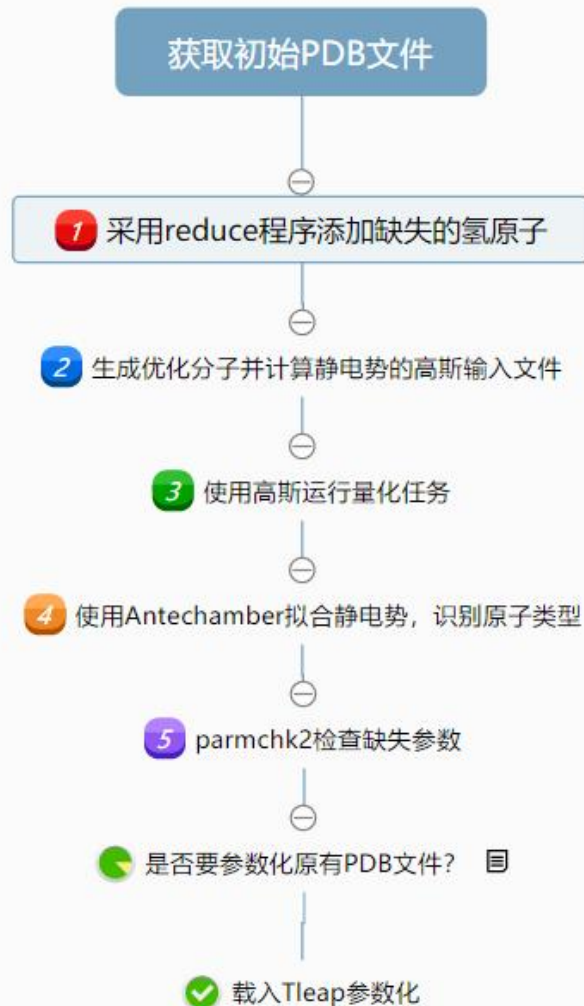
`pdb4amber -i 1fko.pdb -o
1fko_amber.pdb`

HIV-RT (A链) 是标准蛋白质，**ATOM**标志着是标准残基（中间有个残基CSD需要单独处理）。

Sustiva是小分子（**原子标识为HETATM**），非标准残基，需要建立它的残基库文件，与蛋白质之间无键连关系。TER标签分隔两段无连接的残基。

LEaP参数化非标准残基

AMBER不够，GAFF来凑。



<http://sobereva.com/266>

GAFF (general AMBER force field)
力场参数化流程

与OPLS力场参数化过程相似之处

1. 都需从小分子模板搭建
2. 使用软件辅助建模

与OPLS力场参数化过程不同之处

1. GAFF、amber力场御用电荷为RESP电荷（bcc经验电荷也可以）
2. OPLS建议搭配1.2*CM5电荷，用Multiwfn软件可以直接得到（上节课做法使用默认分配的电荷不够严谨）
3. GAFF可以作为AMBER力场的补充，与AMBER系其他力场混用

LEaP参数化非标准残基

Sustiva与HIV-RT复合物中Sustiva的处理。

建模加氢

- 将HETATM的部分复制出来，另存为EFZ.pdb。使用工具补充氢原子。

高斯计算

- 由PDB生成gjf文件，利用高斯计算得到格点密度

电荷拟合

- 使用Antechamber拟合静电势。-c bcc 使用经验静电势亦可

参数检查

- 使用parmchk或者parmchk2检查是否有参数缺失

```
reduce EFZ.pdb > sustiva_h.pdb
```

AmberTools有个reduce工具可以在合适的位置加上氢原子

LEaP参数化非标准残基

GAFF参数化小分子。

```
antechamber -i sustiva_h.pdb -fi pdb -o EFZ.gjf -fo gcrt -pf y -gm "%  
mem=1GB" -gn "%nproc=2" -nc 0 -gk "#B3LYP/6-311G** em=GD3BJ opt  
scrf=solvent=water iop(6/33=2) pop=CHELPG"
```

使用Antechamber工具由PDB生成高斯输入文件（[建议使用B3LYP加隐式溶剂模型计算静电势](#)）-nc 0 表示分子不带电荷，solvent=water 表现隐式水溶液环境。使用高斯计算。

```
antechamber -i EFZ.out -fi gout -o EFZ.mol2 -fo mol2 -s 2 -rn  
EFZ -nc 0 -pf y -c resp
```

使用Antechamber从高斯计算的out文件中拟合静电势。-rn EFZ 定义了残基名字叫EFZ。-at amber 可以定义识别为AMBER原子类型（默认为gaff）。

还需用原来PDB中的原子名称替换新生成的mol2中的原子名称。

加的氢是在mol2文件的最后，因此只需把原来重原子的元素名称替换掉就可以，否则原子名称不一致的话，也无法正确进行参数化。原子名称应该以最后读取的系统PDB中非标准残基的名称为准，因为tleap在处理系统PDB中非标准残基的时候，一一将原子名称与mol2中的进行比较。

LEaP参数化非标准残基

脚本批处理。

检查是否缺少参数，可以使用parmchk2根据经验自动补全参数。在运行模拟前你需要仔细检查这些参数。如果Antechamber不能根据经验计算得到值或者没有类比，它都会添加它认为合理的默认值，要不然就插入一个占位符（所有的值都为0），并且会加上注释” ATTN: needs revision”。

```
parmchk2 -i EFZ.mol2 -f mol2 -o EFZ.frcmod
```

```
ANGLE
c1-c1-cx 56.160 178.510 same as c1-c1-c3, penalty score= 8.0
```

DIHE

Antechamber 自动为EFZ残基补全了一个角，和4个正常二面角。

IMPROPER

```
c -ca-n -hn 1.1 180.0 2.0 Using general improper torsional angle X- X- n-hn,
ca-ca-ca-n 1.1 180.0 2.0 Using the default value
```

写了[两个自用脚本用于批处理](#) pre.py 和 post.py

```
tamas@tamas-PC: ~/Desktop$ python pre.py sustiva_h.pdb 0
```

从 sustiva_h.pdb 开始，净电荷为0，生成gjf, 用高斯进行量化计算

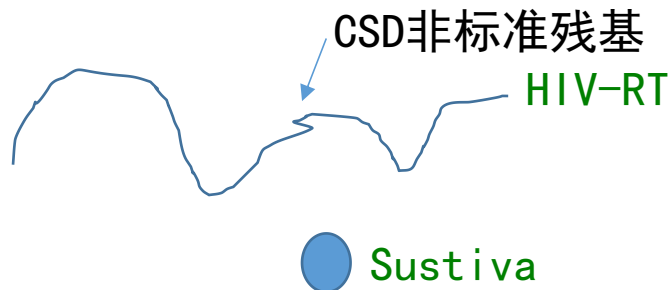
```
tamas@tamas-PC: ~/Desktop$ python post.py
```

拟合静电势，生成_rename.mol2文件，尽管没有缺失的参数，还是需要自己仔细检查。

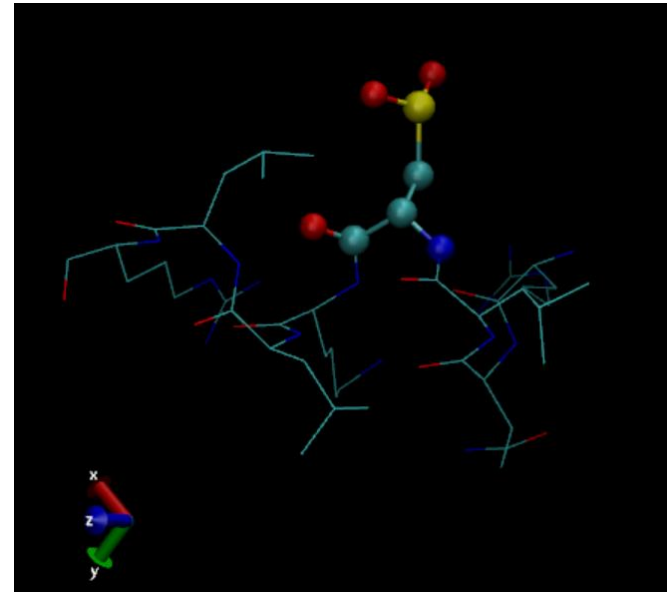
```
Fitting RESP charge...
Checking parameters...
No parameters are missing, but should be checked by yourself!
```

LEaP参数化非标准残基

蛋白内非标准残基的处理。



CSD的处理流程不太相同因为它需要与其他残基成键。



CSD基本保持了蛋白的结构，绝大部分原子适用AMBER力场参数，硫原子除外。推荐首先以amber力场参数化CSD，如果首先以gaff力场参数化，它与其他标准蛋白质残基之间的力场就需要自己手动补充。我们优先考虑利用parmchk2的自动补充参数功能。

通用的做法是连同非标准残基两端的两个残基都取下，做量化计算，并利用antechamber识别原子类型，好处是可以**自动生成标准残基与非标准残基之间连接的力场参数**。但是电荷需要小心处理。处理思路参考[我翻译的教程](#)。下节课将演示如果用GAFF力场参数化嵌段残基。

GROMACS
FAST. FLEXIBLE. FREE.



特殊体系的模型和拓扑结构 搭建

使用GAFF力场为dendrimer聚合物建模

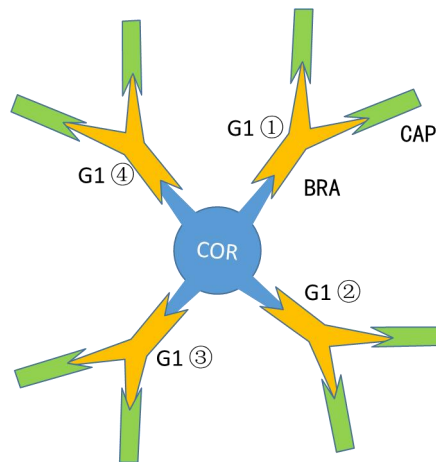
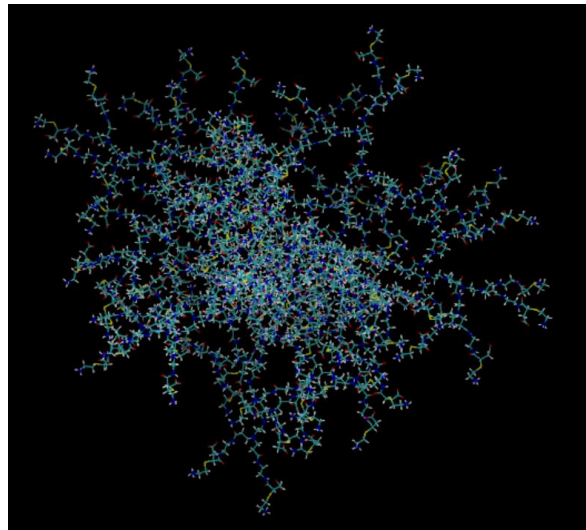
主讲人：许楠

浙江大学博士研究生

VASPKIT合作开发者

Dendrimer 介绍

树状多枝有机大分子。



Dendrimer 树状高分子

- 3维
- 多分支
- 树状高分子聚合物

结构特点

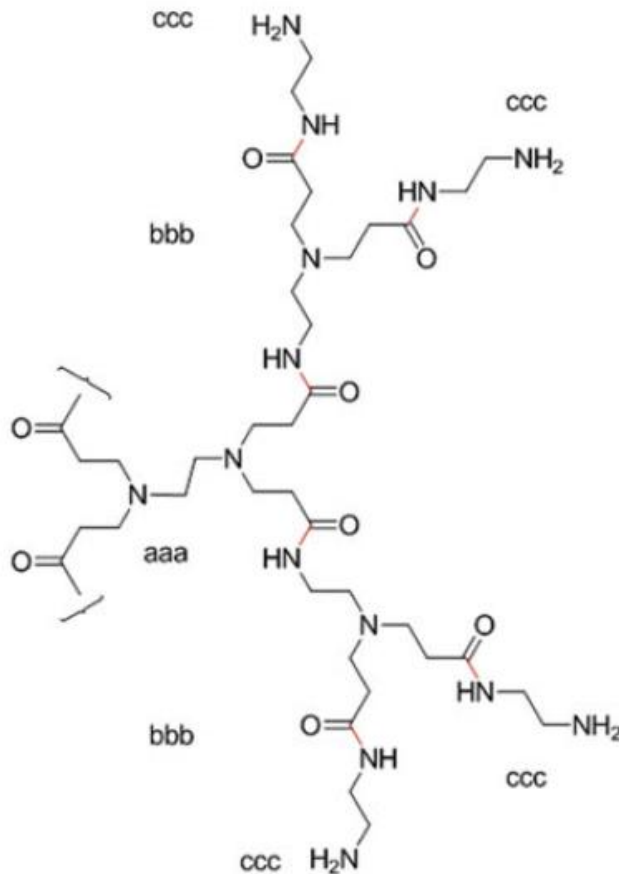
- 三部分构成：中心分子，枝节和外表面功能团，由中心分子向外开叉生长
- 中心分子决定了聚合物的形状，大小
- 枝节决定了dendrimer内部空间，不同的代数(G_x)就决定了枝节的数目。
- 外表面可以作为功能官能团作药物载体。

建模方法

- Material Studio, 但是可调节性不强, 也无法用常用力场参数化
- Dendrimer builder toolkit (DBT), 免费, 可自行定制, 在建模的同时赋予 gaff 力场。

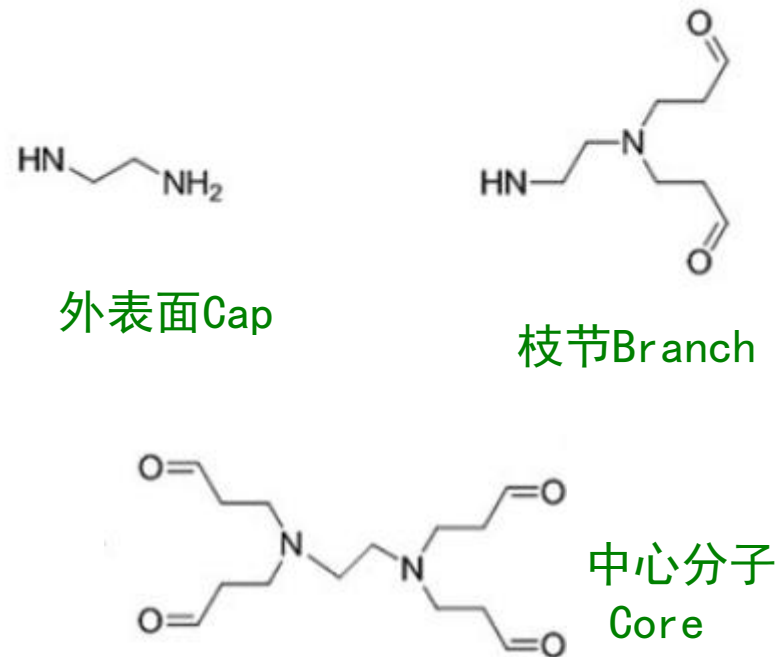
Dendrimer 介绍

聚酰胺PAMAM Dendrimer。



PAMAM Dendrimer 树状高分子

- 聚酰胺Poly(amidoamine) (PAMAM) Dendrimer
- 无毒的药物输运载体



Dendrimer 建模

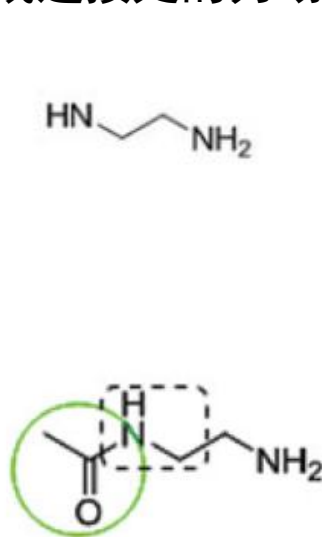
GAFF力场参数化。

Dendrimer 经常与多肽一起出现，amber 家族gaff力场最合适。

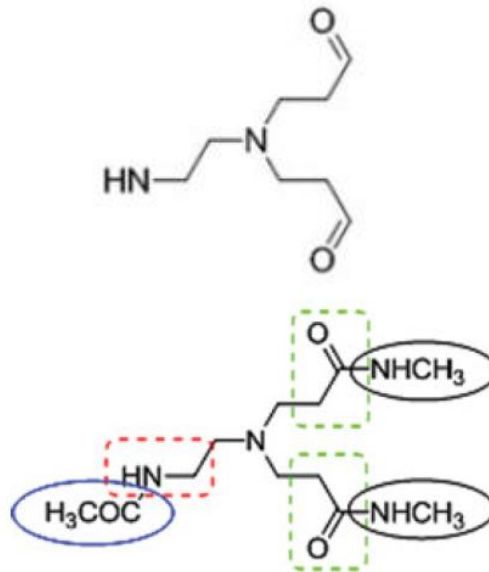
如何建模？

不能直接对子单元参数化，这样会无法准确识别连接处的原子类型。因此需要带着周围的分子一起处理。这样不仅可以正确识别连接处的原子类型，也可以生成连接处的力场参数。

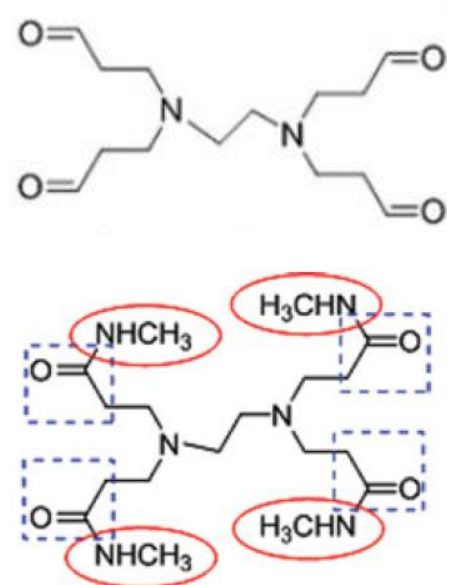
补充连接的成键原子，保证化学环境与原来的一致。



外表面Cap



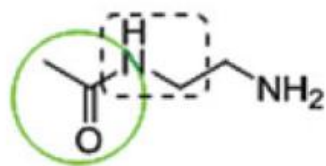
枝节Branch



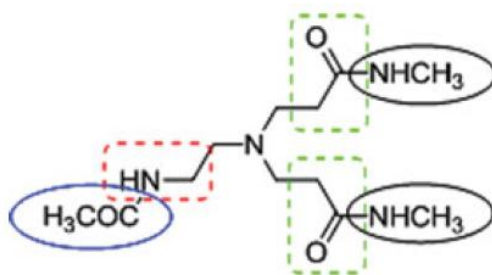
中心分子 Core

Dendrimer 建模

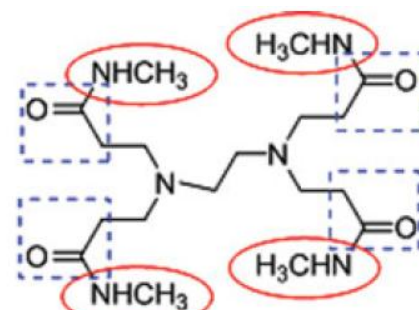
GAFF力场参数化。



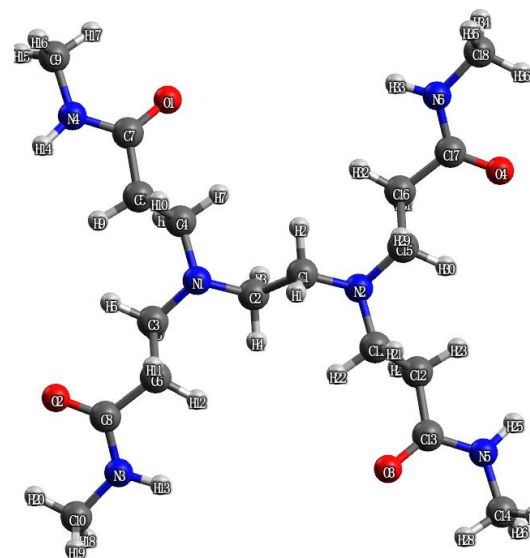
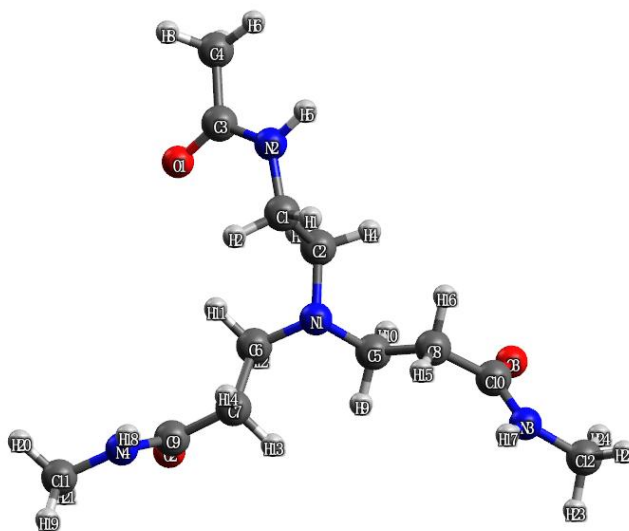
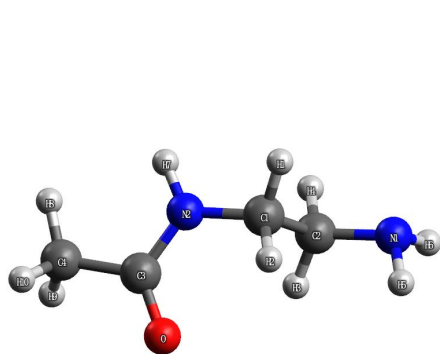
外表面Cap



枝节Branch



中心分子 Core



Maingi V , Jain V , Bharatam P V , et al. 2012, 33(25) :1997---2011.

Dendrimer建模

Auto-Amber工具自动以GAFF力场参数化。

以Branch为例讲解参数化过程

使用自写的Auto-Amber工具（[pre.py](#) 和 [post.py](#)）自动进行参数化

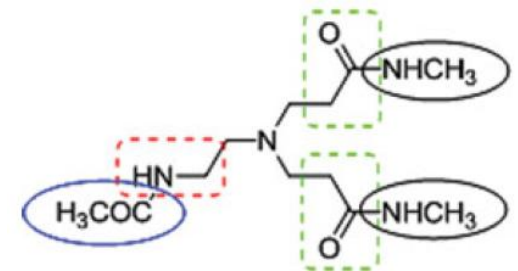
tamas@tamas-PC: ~/Desktop\$ **python** pre.py BRA.pdb 0

从 BRA.pdb 开始，净电荷为0，生成gjf，用高斯进行量化计算

tamas@tamas-PC: ~/Desktop\$ **python** post.py

拟合静电势，程序推荐生成_rename.mol2文件，也就是生成的mol2文件复制的原来PDB中的原子名称，但是起初的PDB中原子没有原子序号，不方便后续处理，所有载入BRA.mol2。同时程序自动检查发现没有缺失的参数，但还是需要自己仔细检查。

```
checking parameters...
Please sentences for your reference.
()
source leaprc.gaff
set default PBRadii mbondi3
BRA= loadmol2 BRA_rename.mol2
loadAmberParams BRA.frcmod
()
No parameters are missing, but should be checked by yourself!
```



还需进一步处理，
去除多余的成键原子

Dendrimer建模

Xleap工具可视化编辑残基。

使用LEaP的可视化版本Xleap进行后续处理。处理思路参考[我翻译的教程](#)。

Xleap需要在有桌面的Linux系统上运行，亦可设置X11转发后，使用远程终端。为了正常使用Xleap，请关闭小键盘上的NUMLOCK键。XLEAP的窗口操作亦可参考教程。

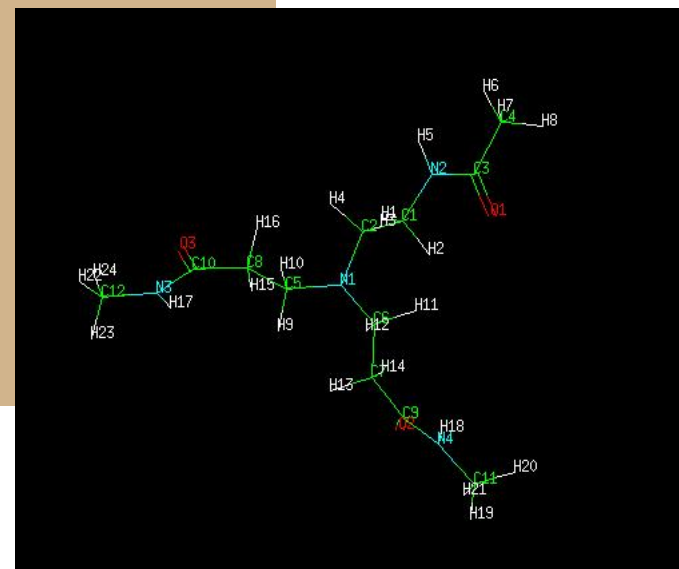
```
(no leaprc in search path)
> source leaprc.gaff
----- Source: /home/goldnan/Documents/conda/dat/leap/cmd/leaprc.gaff
----- Source of /home/goldnan/Documents/conda/dat/leap/cmd/leaprc.gaff done
Log file: ./leap.log
Loading parameters: /home/goldnan/Documents/conda/dat/leap/parm/gaff.dat
Reading title:
AMBER General Force Field for organic molecules (Version 1.8, Mar 2015)
> BRA=loadmol2 BRA.mol2
Loading Mol2 file: ./BRA.mol2
Reading MOLECULE named BRA
> loadAmberParams BRA.frcmod
Loading parameters: ./BRA.frcmod
Reading force field modification type file (frcmod)
Reading title:
Remark line goes here
> check BRA
Checking 'BRA'....
Checking parameters for unit 'BRA'.
Checking for bond parameters.
Checking for angle parameters.
Unit is OK.
> edit BRA
>
```

载入BRA残基

载入补充参数

检查BRA单元

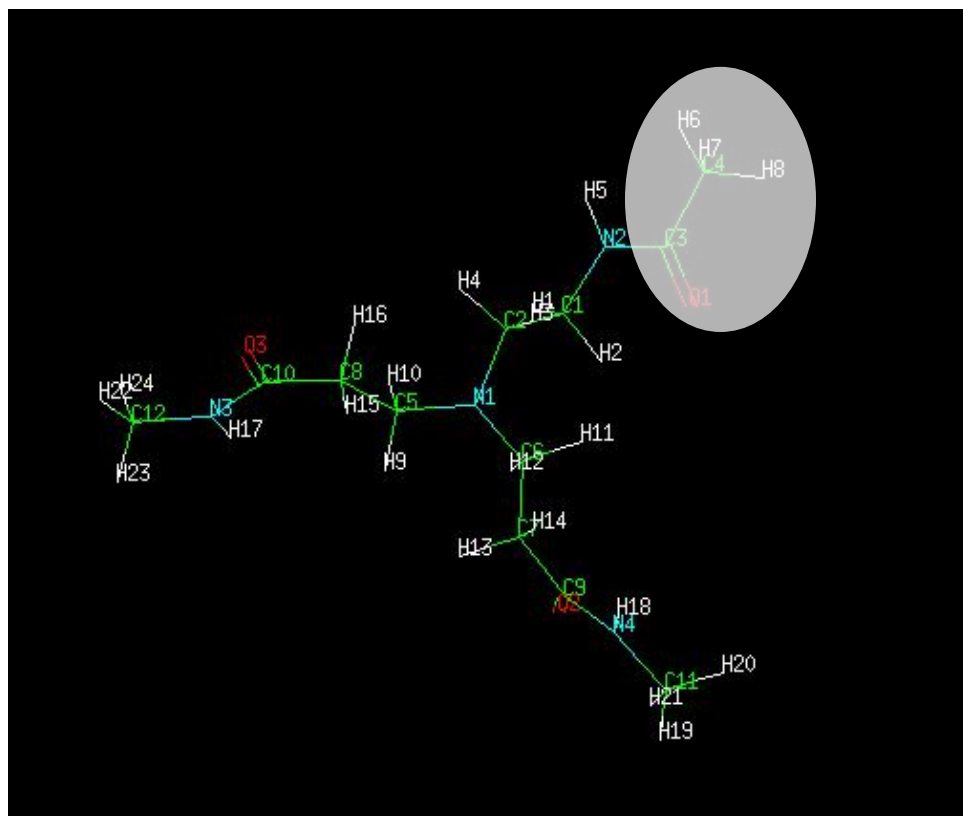
开启窗口编辑BRA单元



Dendrimer建模

Xleap工具可视化编辑残基。

1. 换成Erase模式，擦掉一根枝条上的（N2端）多余原子（这里是C3, O1, C4, H6, H7, H8），橡皮擦在这些原子上单击。



2. 再查看去掉6个原子后残基的电荷，窗口输入命令：charge BRA，意味着体系带有-0.038354的负电荷，我们把它加到N2原子上。

```
> charge BRA
Total unperturbed charge: -0.038354
Total perturbed charge: -0.038354
>
```

3. 换成select模式，选中N2原子，菜单栏选择Edit-Edit selected atoms。（请关闭小键盘上的NUMLOCK键）

NAME	TYPE	CHARGE	ELEMENT
N2	n	-0.663000	N

Dendrimer建模

Xleap工具可视化编辑残基。

4. N2现在的电荷是-0.663，加上0.038354，新的电荷是-0.624646，同时记下N2是Branch单元的头原子。

NAME	TYPE	CHARGE	ELEMENT
N2	n	-0.663000	N

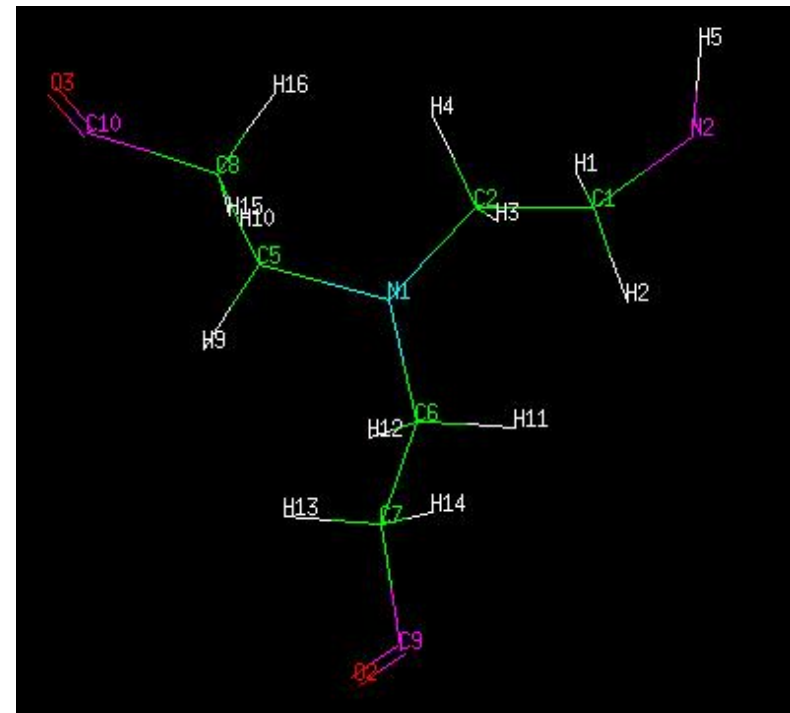
NAME	TYPE	CHARGE	ELEMENT
N2	n	-0.624646	N

5. 再使用同样的方法对两个尾端进行处理。
处理后再查看总电荷为0，C9和C10是尾原子。

6. 最后把处理好的残基保存为新的mol2:

```
savemol2 BRA BRA_new.mol2 1
```

残基名	COR	BRA	CAP
坐标文件	COR_new.mol2	BRA_new.mol2	CAP_new.mol2
补充力场	COR.frcmod	BRA.frcmod	CAP.frcmod
头原子	无	N2	N2
尾原子	C7, C8, C13, C17	C9, C10	无



DBT工具建模



材料
Cailiaoren.com

Dendrimmer building toolkit using amber force field.

使用perl-tk的有界面Dendrimmer建模工具。

Dbtv3

Dendrimmer Builder Toolkit

Number of layers desired, these configuration will be reserved to

Core's tail_num.	Index or name of each tail atom	Fragment's tail_num: 2 only PREFERRED.	Head atom in the tail_num+1 row	Terminal s' head_num.	Head atom
1	C7	1	C9	1	N2
2	C17	2	C10		
3	C13	3	N2		
4	C8	4			
5		5			
6					

force field(gaff or protein.ff14SB)

frmod file for 'core' or leave this option blank

frmod file for 'fragment' or leave this option blank

frmod file for 'cap' or leave this option blank

mol2 for core

mol2 for fragment

mol2 for cap

--> now press sander

DBT工具建模

Dendrimer building toolkit using amber force field.

- 文献中的版本很老，不再与现在版本的Ambertools适配。并且已经不再更新，我已经对其进行了更新，适配了新版本Ambertools。
- 老版本界面提示信息缺乏，并且只能使用aaa, bbb, ccc这样的残基名字命名core, branch和cap残基，灵活性太差。我对交互界面做了大更新，界面更友好，对输入文件的要求也不高。
- 使用了不常见的perl-tk界面库，Linux和mac发行版都不标配，并且Windows用户无法使用。我计划以更友好的python-tk提供交互，请关注我的[GitHub](#)。
- 建模算法不够理想，在代数多时，效率比较低。未来计划推出我自己的Dendrimer建模程序。
- Linux用户若想使用此版本，需要安装perl-tk第三方库：
 Ubuntu: `sudo apt install perl-tk`
 Redhat: `sudo yum install perl-Tk`
- 使用方法：将用到的所有mol2, frcmod文件和dbtv3.pl放在同一目录，在**有界面的**Linux系统上输入（需要安装Ambertools套件）：

```
perl dbtv3.pl
```

自动打开一个GUI界面。

DBT工具建模



材料
Cailiaoren.com

Dendrimmer building toolkit using amber force field.

Dendrimer建模工具界面。

DBT将保留从第3代（G3）到第4代（G4）的Dendrimer坐标。

Dbtv3

Dendrimer Builder Toolkit

Number of layers desired, these configuration will be reserved 3 to 4

Core's tail_num.	Index or name of each tail atom	Fragment's tail_num: 2 only PREFERRED.	Head atom in the tail_num+1 row	Terminal s' head_num.	Head atom
1	C7	1	C9	1	N2
2	C17	2	C10		
3	C13	3	N2		
4	C8				
5					
6					

Core有4个尾原子
分别写上它们的
原子名称。

Branch有两个尾原子，只能有一个头原子N2，写在第3列。

Cap只有一个头原子

force field(gaff or protein.ff14SB) gaff 使用gaff力场

frmod file for 'core' or leave this option blank COR.frcmod

frmod file for 'fragment' or leave this option blank BRA.frcmod

frmod file for 'cap' or leave this option blank CAP.frcmod

mol2 for core COR.mol2

mol2 for fragment BRA.mol2

mol2 for cap CAP.mol2

分别载入坐标和补充力场文件

程序建模开始

① go

② OK → now press sander

③ sander minimization

quit

L3, L4. pdb分别为所需的dendrimer构型

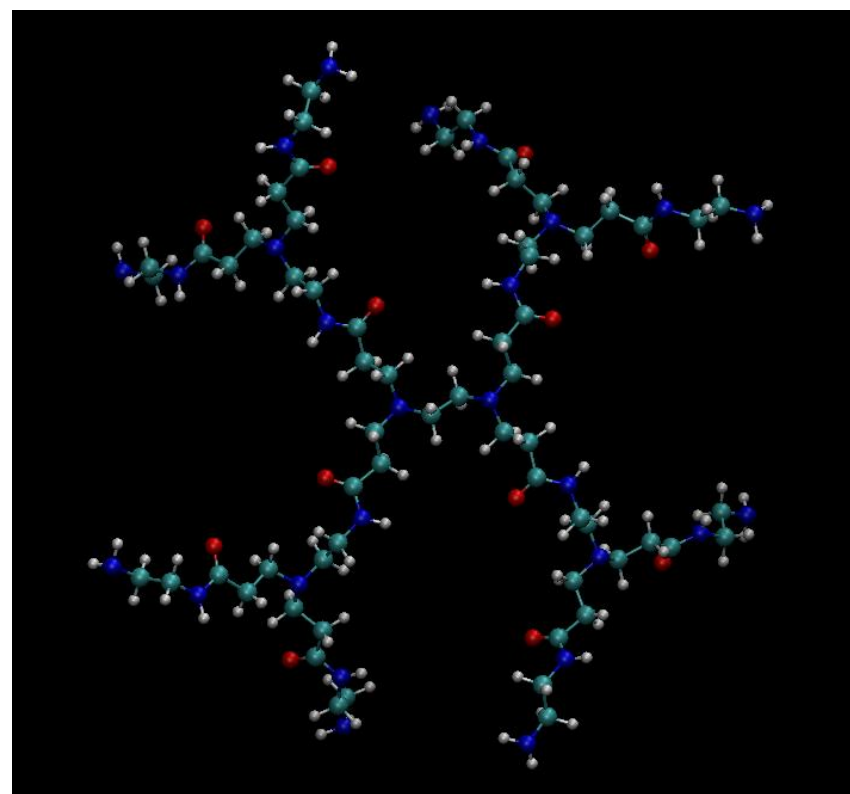
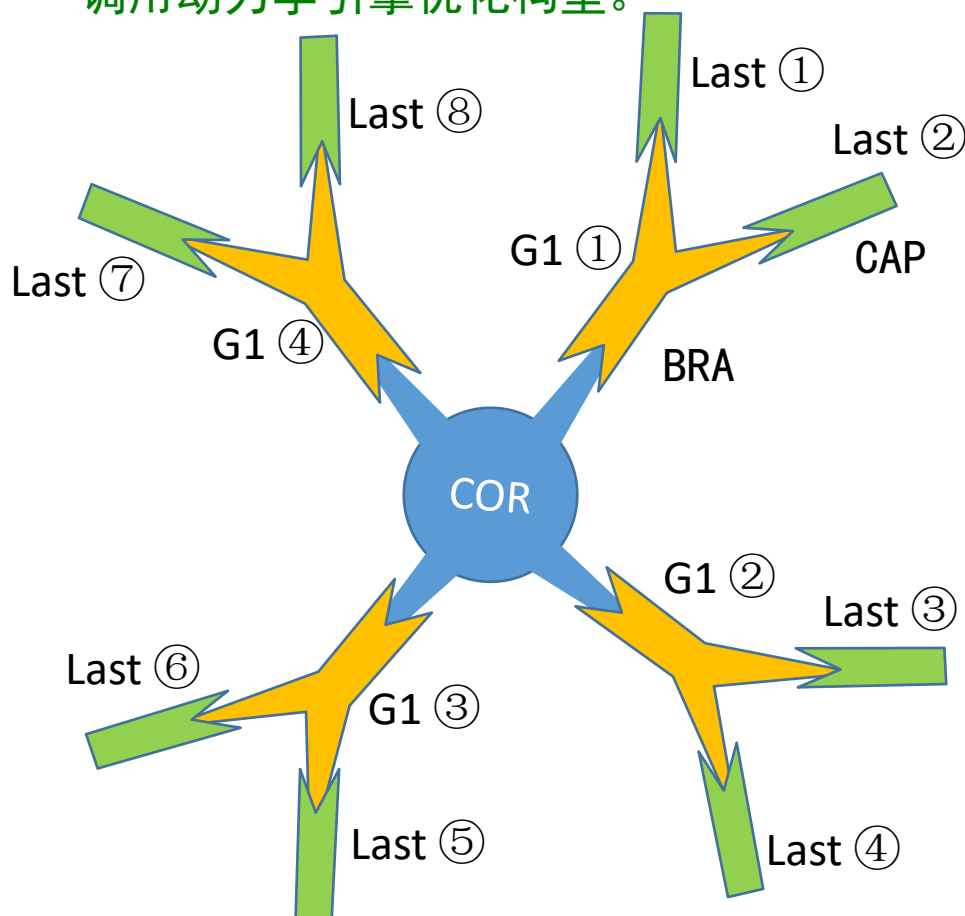
DBT工具建模



材料
Cailiaoren.com

Dendrimer building toolkit using amber force field.

以第一代为例讲解搭建过程。循环利用tleap中的sequence命令拼接单体，再调用动力学引擎优化构型。



Maingi V , Jain V , Bharatam P V , et al. 2012, 33(25):1997---2011.