Xplor-NIH basic principle, and usage in protein structure refinement

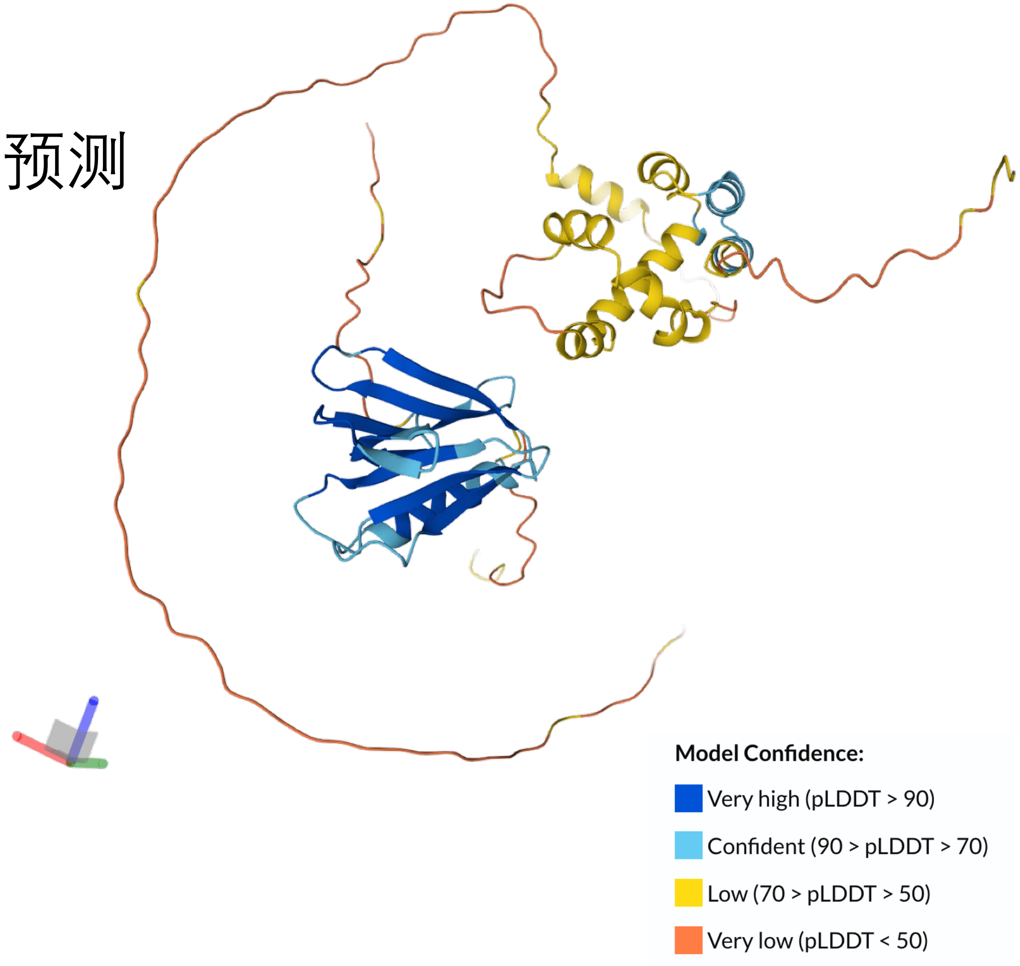
唐 淳

北京大学化学与分子工程学院

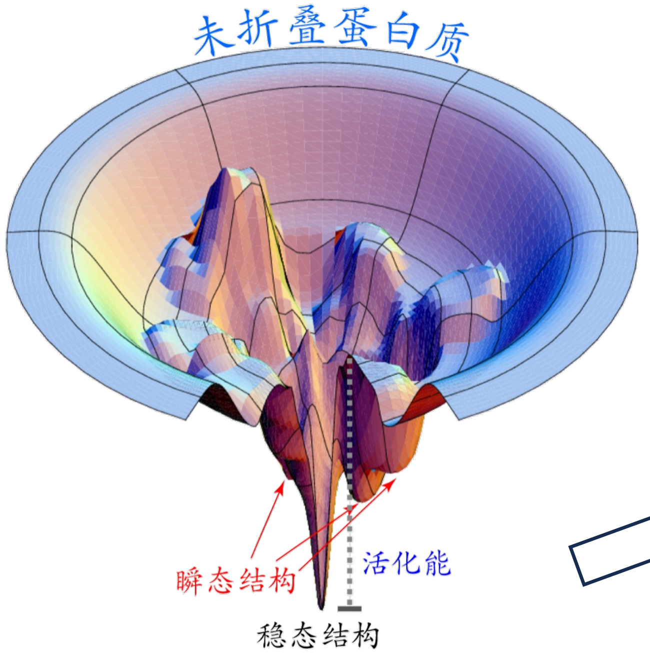
北京大学生命科学联合中心

结构计算的初心 （motivation）

- 基于实验数据的计算优化 ← → 模型预测
- 多种结构状态 ← → MSA
- 柔性和动态区域，以及 IDP/IDR



结构计算的原理 （principle）

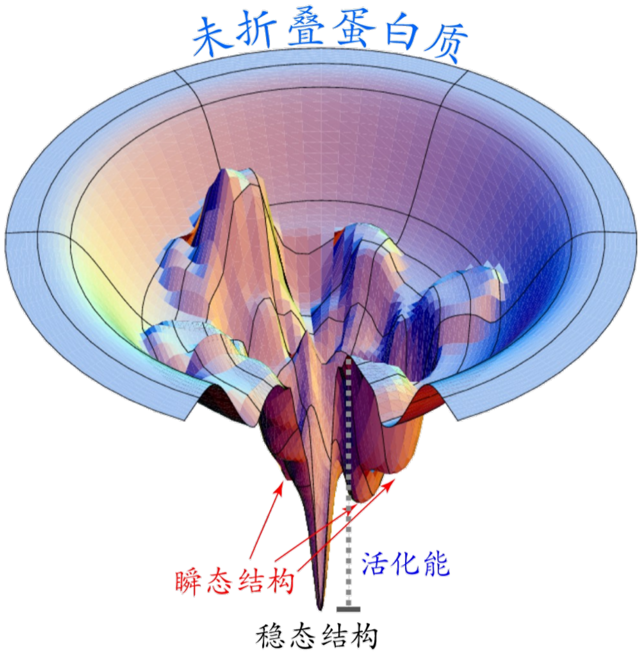


柔性区域

满足约束的一套结构

$$V_{\mathrm{tot}} = V_{\mathrm{physics}} + \boxed{V_{\mathrm{expt}} + V_{\mathrm{knowledge}} + \ldots}$$

结构计算的原理 （principle）



模拟退火（simulated annealing）

- Surface smoothed at high temperature
- Slowly decrease the temperature to find the global minimum
- Gradualy incrase the weight (系数) of the experimental restraints

Xplor/CNS → Xplor-NIH

3.8.5.1 3.7.0.1

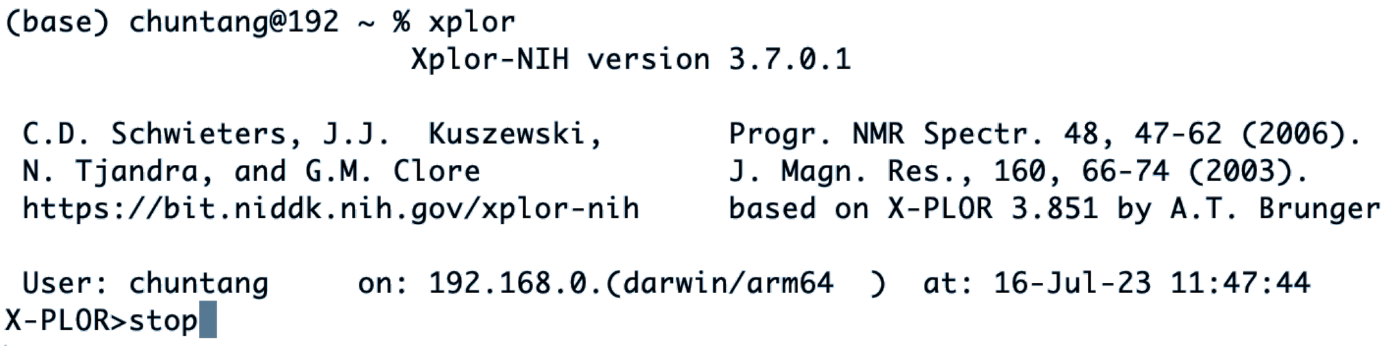
Axel Brunger Charles Schwieters, Marius Clore, 等

1984-1998 ~2000-present （2015）

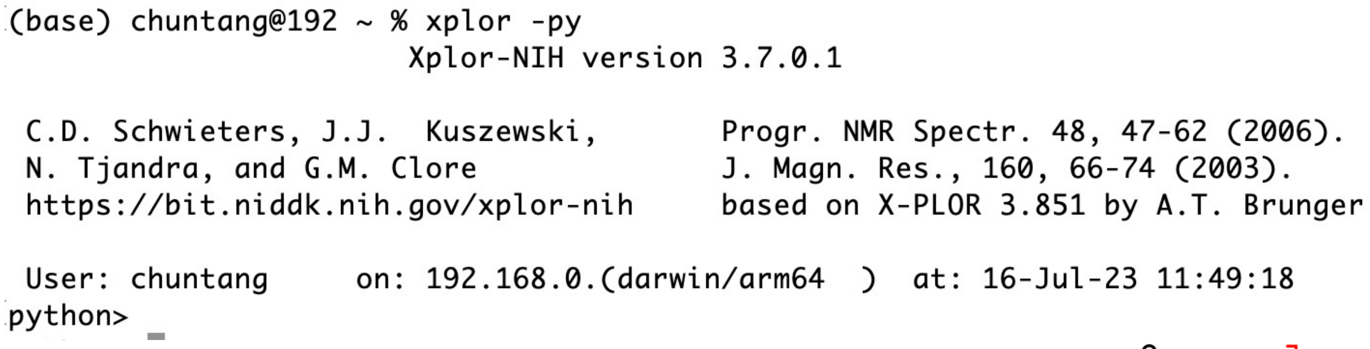
与溶液结构计算相关的势能函数

- noePot - NOE distance restraints
- rdcPot - dipolar coupling
- csaPot - Chemical Shift Anisotropy
- jCoupPot - 3 J-coupling
- prePot - Paramagnetic relaxation enhancement
- diffPot - refine against rotational diffusion tensor
- gyrPot - pseudopotential enforcing correct protein density
- posSymmPot - restrain atomic positions relative to those in a similar structure
- …

启动Xplor-NIH

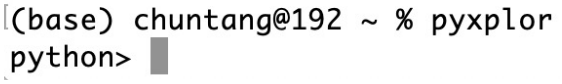


```
(base) chuntang@192 ~ % xplor
                           Xplor-NIH version 3.7.0.1

 C.D. Schwieters, J.J.  Kuszewski,       Progr. NMR Spectr. 48, 47-62 (2006).
 N. Tjandra, and G.M. Clore              J. Magn. Res., 160, 66-74 (2003).
 https://bit.niddk.nih.gov/xplor-nih     based on X-PLOR 3.851 by A.T. Brunger

 User: chuntang     on: 192.168.0.(darwin/arm64  )  at: 16-Jul-23 11:47:44
X-PLOR>stop
```



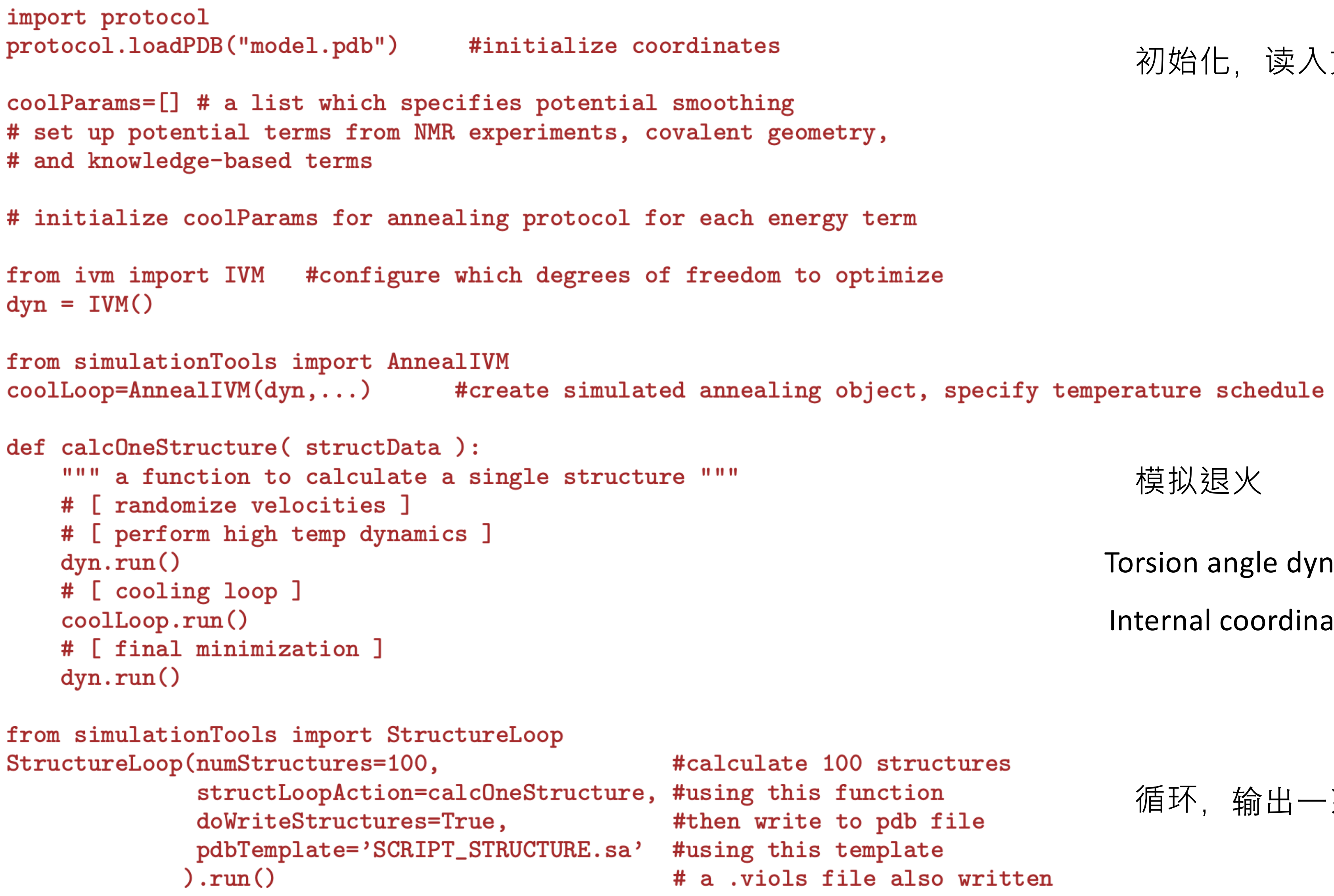
```
(base) chuntang@192 ~ % xplor -py
                           Xplor-NIH version 3.7.0.1

 C.D. Schwieters, J.J.  Kuszewski,       Progr. NMR Spectr. 48, 47-62 (2006).
 N. Tjandra, and G.M. Clore              J. Magn. Res., 160, 66-74 (2003).
 https://bit.niddk.nih.gov/xplor-nih     based on X-PLOR 3.851 by A.T. Brunger

 User: chuntang     on: 192.168.0.(darwin/arm64  )  at: 16-Jul-23 11:49:18
python>
```

```
% xplor script.py
```



```
(base) chuntang@192 ~ % pyxplor
python>
```

% xplor –smp 8 -py …



```
import protocol
protocol.loadPDB("model.pdb")     #initialize coordinates

coolParams=[] # a list which specifies potential smoothing
# set up potential terms from NMR experiments, covalent geometry,
# and knowledge-based terms

# initialize coolParams for annealing protocol for each energy term

from ivm import IVM   #configure which degrees of freedom to optimize
dyn = IVM()

from simulationTools import AnnealIVM
coolLoop=AnnealIVM(dyn,...)      #create simulated annealing object, specify temperature schedule

def calcOneStructure( structData ):
    """ a function to calculate a single structure """
    # [ randomize velocities ]
    # [ perform high temp dynamics ]
    dyn.run()
    # [ cooling loop ]
    coolLoop.run()
    # [ final minimization ]
    dyn.run()

from simulationTools import StructureLoop
StructureLoop(numStructures=100,                 #calculate 100 structures
              structLoopAction=calcOneStructure, #using this function
              doWriteStructures=True,            #then write to pdb file
              pdbTemplate='SCRIPT_STRUCTURE.sa'  #using this template
             ).run()                             # a .viols file also written
```

初始化，读入文件

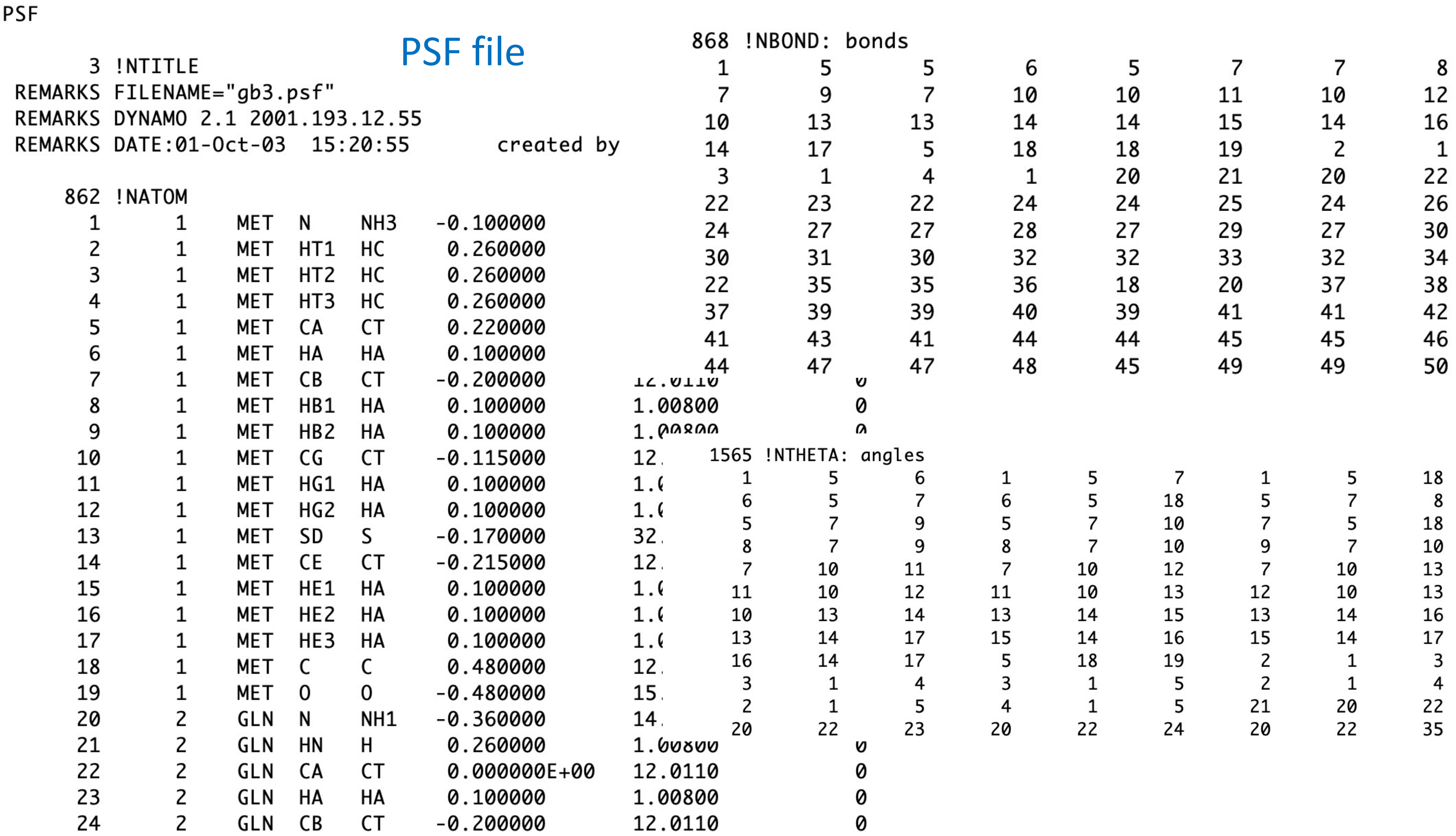
模拟退火

Torsion angle dynamics

Internal coordinates

循环，输出一系列结构

PSF file



```
PSF

       3 !NTITLE
 REMARKS FILENAME="gb3.psf"
 REMARKS DYNAMO 2.1 2001.193.12.55
 REMARKS DATE:01-Oct-03   15:20:55       created by

     862 !NATOM
       1        1      MET   N     NH3    -0.100000
       2        1      MET   HT1   HC      0.260000
       3        1      MET   HT2   HC      0.260000
       4        1      MET   HT3   HC      0.260000
       5        1      MET   CA    CT      0.220000
       6        1      MET   HA    HA      0.100000
       7        1      MET   CB    CT     -0.200000       12.0110           0
       8        1      MET   HB1   HA      0.100000       1.00800           0
       9        1      MET   HB2   HA      0.100000       1.00800           0
      10        1      MET   CG    CT     -0.115000       12.
      11        1      MET   HG1   HA      0.100000       1.0
      12        1      MET   HG2   HA      0.100000       1.0
      13        1      MET   SD    S      -0.170000       32.
      14        1      MET   CE    CT     -0.215000       12.
      15        1      MET   HE1   HA      0.100000       1.0
      16        1      MET   HE2   HA      0.100000       1.0
      17        1      MET   HE3   HA      0.100000       1.0
      18        1      MET   C     C       0.480000       12.
      19        1      MET   O     O      -0.480000       15.
      20        2      GLN   N     NH1    -0.360000       14.
      21        2      GLN   HN    H       0.260000       1.00800           0
      22        2      GLN   CA    CT      0.000000E+00   12.0110           0
      23        2      GLN   HA    HA      0.100000       1.00800           0
      24        2      GLN   CB    CT     -0.200000       12.0110           0
```

```
     868 !NBOND: bonds
       1       5       5       6       5       7       7       8
       7       9       7      10      10      11      10      12
      10      13      13      14      14      15      14      16
      14      17       5      18      18      19       2       1
       3       1       4       1      20      21      20      22
      22      23      22      24      24      25      24      26
      24      27      27      28      27      29      27      30
      30      31      30      32      32      33      32      34
      22      35      35      36      18      20      37      38
      37      39      39      40      39      41      41      42
      41      43      41      44      44      45      45      46
      44      47      47      48      45      49      49      50
```

```
    1565 !NTHETA: angles
       1       5       6       1       5       7       1       5      18
       6       5       7       6       5      18       5       7       8
       5       7       9       5       7      10       7       5      18
       8       7       9       8       7      10       9       7      10
       7      10      11       7      10      12       7      10      13
      11      10      12      11      10      13      12      10      13
      10      13      14      13      14      15      13      14      16
      13      14      17      15      14      16      15      14      17
      16      14      17       5      18      19       2       1       3
       3       1       4       3       1       5       2       1       4
       2       1       5       4       1       5      21      20      22
      20      22      23      20      22      24      20      22      35
```

Topology file

```
residue ALA
  group
    atom N   type=NH1 charge=-0.36 end
    atom HN  type=H   charge= 0.26 end
  group
    atom CA  type=CT  charge= 0.00 end
    atom HA  type=HA  charge= 0.10 end
  group
    atom CB  type=CT  charge=-0.30 end
    atom HB1 type=HA  charge= 0.10 end
    atom HB2 type=HA  charge= 0.10 end
    atom HB3 type=HA  charge= 0.10 end
  group
    atom C   type=C   charge= 0.48 end
    atom O   type=O   charge=-0.48 end

  bond N   HN
  bond N   CA     bond CA  HA
  bond CA  CB     bond CB  HB1    bond CB  HB2       bond CB  HB3
  bond CA  C
  bond C   O

  improper HA  N   C   CB  !stereo CA
  improper HB1 HB2 CA HB3  !stereo CB

end
```

原子性质

共价连接

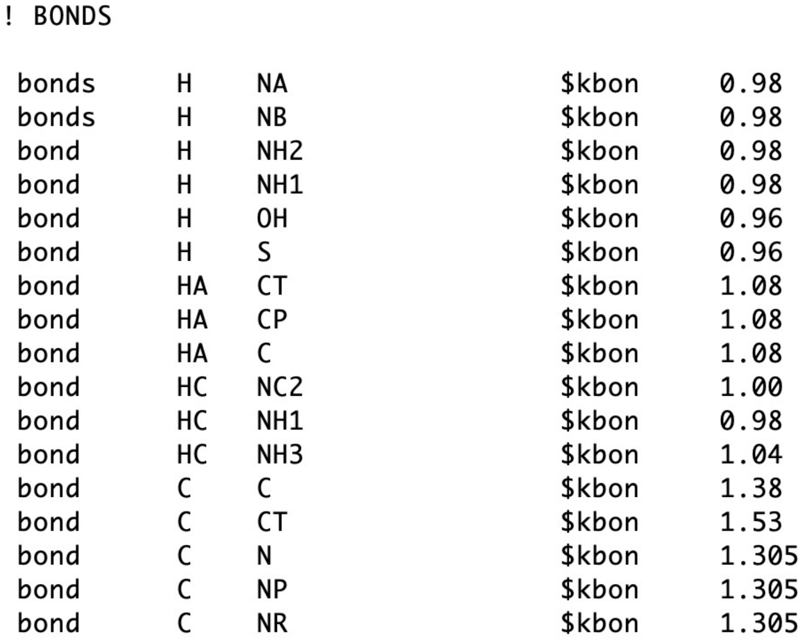
可转动的部分

```
mass NC2 14.007
mass O   15.999
mass OC  15.999
mass OH  15.999
mass S   32.060
```

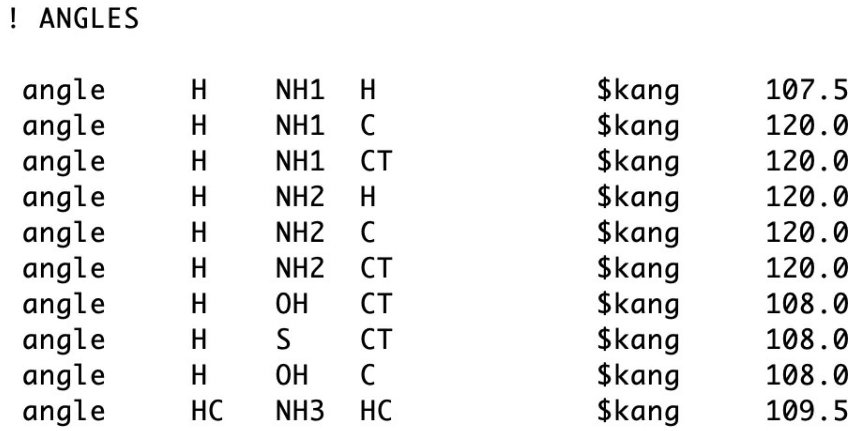
自己画

EDTA-Mn

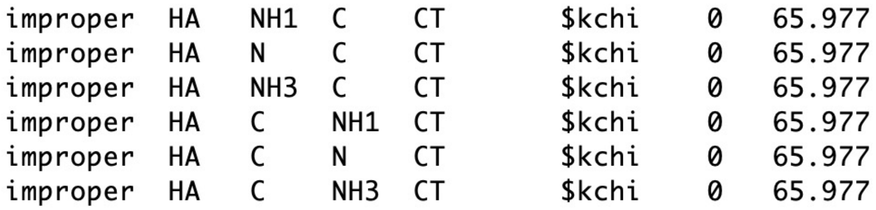
```
residue CHEX  !! ADDED BY MN
  group
    atom N   type=NH1 charge=-0.360 end
    atom HN  type=H   charge= 0.260 end
 group
    atom CA  type=CT  charge= 0.000 end
    atom HA  type=HA  charge= 0.100 end
 group
    atom CB  type=CT  charge=-0.200 end
    atom HB1 type=HA  charge= 0.100 end
    atom HB2 type=HA  charge= 0.100 end
 group
    atom CG  type=CT  charge=-0.200 end
    atom HG  type=HA  charge= 0.100 end
 group
    atom CD1  type=CT  charge=-0.200 end
    atom HD11 type=HA  charge= 0.100 end
    atom HD12 type=HA  charge= 0.100 end
 group
    atom CD2  type=CT  charge=-0.200 end
    atom HD21 type=HA  charge= 0.100 end
    atom HD22 type=HA  charge= 0.100 end
 group
    atom CE1  type=CT  charge=-0.200 end
    atom HE11 type=HA  charge= 0.100 end
    atom HE12 type=HA  charge= 0.100 end
 group
    atom CE2  type=CT  charge=-0.200 end
    atom HE21 type=HA  charge= 0.100 end
    atom HE22 type=HA  charge= 0.100 end
 group
    atom CZ  type=CT  charge=-0.200 end
    atom HZ1 type=HA  charge= 0.100 end
```



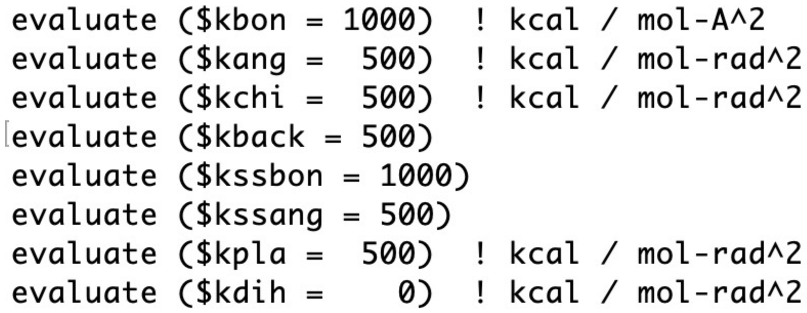
```
! BONDS

 bonds      H    NA                 $kbon     0.98
 bonds      H    NB                 $kbon     0.98
 bond       H    NH2                $kbon     0.98
 bond       H    NH1                $kbon     0.98
 bond       H    OH                 $kbon     0.96
 bond       H    S                  $kbon     0.96
 bond       HA   CT                 $kbon     1.08
 bond       HA   CP                 $kbon     1.08
 bond       HA   C                  $kbon     1.08
 bond       HC   NC2                $kbon     1.00
 bond       HC   NH1                $kbon     0.98
 bond       HC   NH3                $kbon     1.04
 bond       C    C                  $kbon     1.38
 bond       C    CT                 $kbon     1.53
 bond       C    N                  $kbon     1.305
 bond       C    NP                 $kbon     1.305
 bond       C    NR                 $kbon     1.305
```



```
! ANGLES

 angle      H     NH1   H              $kang     107.5
 angle      H     NH1   C              $kang     120.0
 angle      H     NH1   CT             $kang     120.0
 angle      H     NH2   H              $kang     120.0
 angle      H     NH2   C              $kang     120.0
 angle      H     NH2   CT             $kang     120.0
 angle      H     OH    CT             $kang     108.0
 angle      H     S     CT             $kang     108.0
 angle      H     OH    C              $kang     108.0
 angle      HC    NH3   HC             $kang     109.5
```



```
improper   HA   NH1   C     CT       $kchi    0    65.977
improper   HA   N     C     CT       $kchi    0    65.977
improper   HA   NH3   C     CT       $kchi    0    65.977
improper   HA   C     NH1   CT       $kchi    0    65.977
improper   HA   C     N     CT       $kchi    0    65.977
improper   HA   C     NH3   CT       $kchi    0    65.977
```



```
evaluate ($kbon = 1000)  ! kcal / mol-A^2
evaluate ($kang =  500)  ! kcal / mol-rad^2
evaluate ($kchi =  500)  ! kcal / mol-rad^2
evaluate ($kback = 500)
evaluate ($kssbon = 1000)
evaluate ($kssang = 500)
evaluate ($kpla =  500)  ! kcal / mol-rad^2
evaluate ($kdih =    0)  ! kcal / mol-rad^2
```

Parameter file → forcefield

生成PSF文件

```
% seq2psf file.seq
% pdb2psf file.pdb
```

```
protocol.initStruct　（）
```

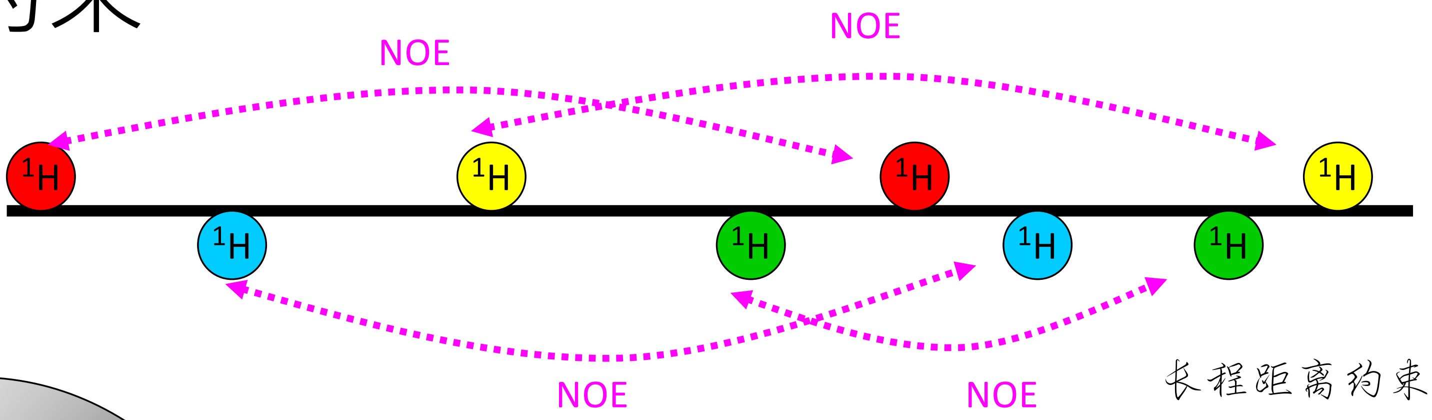
```
protocol.initCoords　（）
  protocol.loadPDB
protocol.genExtendedStructure("gb1_extended_%d.pdb" % seed)
```

读入PDB文件

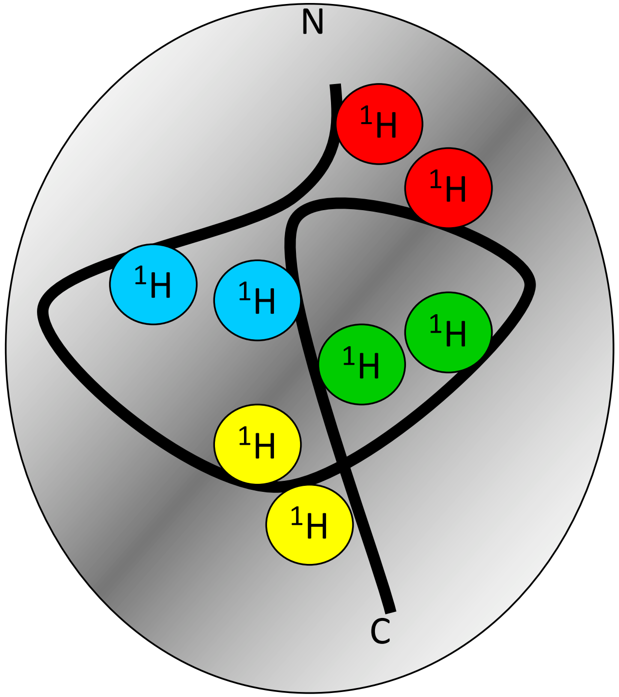
```
protocol.writePDB("file.pdb")
protocol.writeCIF("file.cif")
```

输出计算结果

NOE距离约束

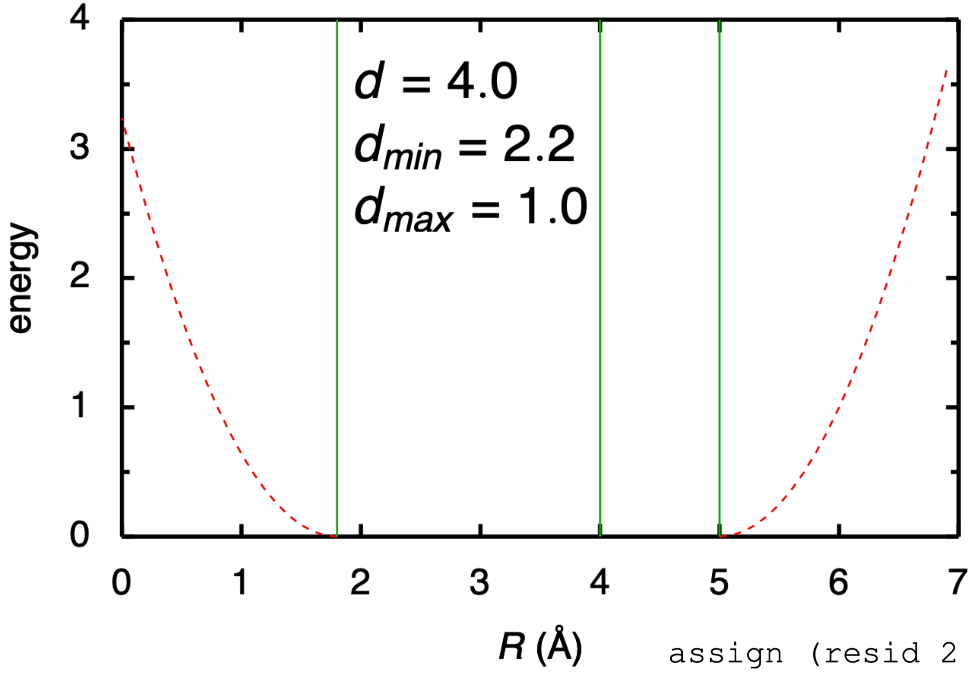


长程距离约束



NOE距离约束

$$R = (\sum_{ij} |q_i - q_j|^{-6})^{-1/6}$$



半定量NOE ⟵ Exact NOE

- ✓ Strong NOE: 1.8~2.7 Å
- ✓ Medium NOE: 1.8~3.5 Å
- ✓ Weak NOE: 1.8~5.0Å

基于蛋白质一级序列

- ✓ Intra-residue: i=j
- ✓ Sequential: |i-j|=1
- ✓ Medium range: 1<|i-j|<=4
- ✓ Long range: |i-j|>4

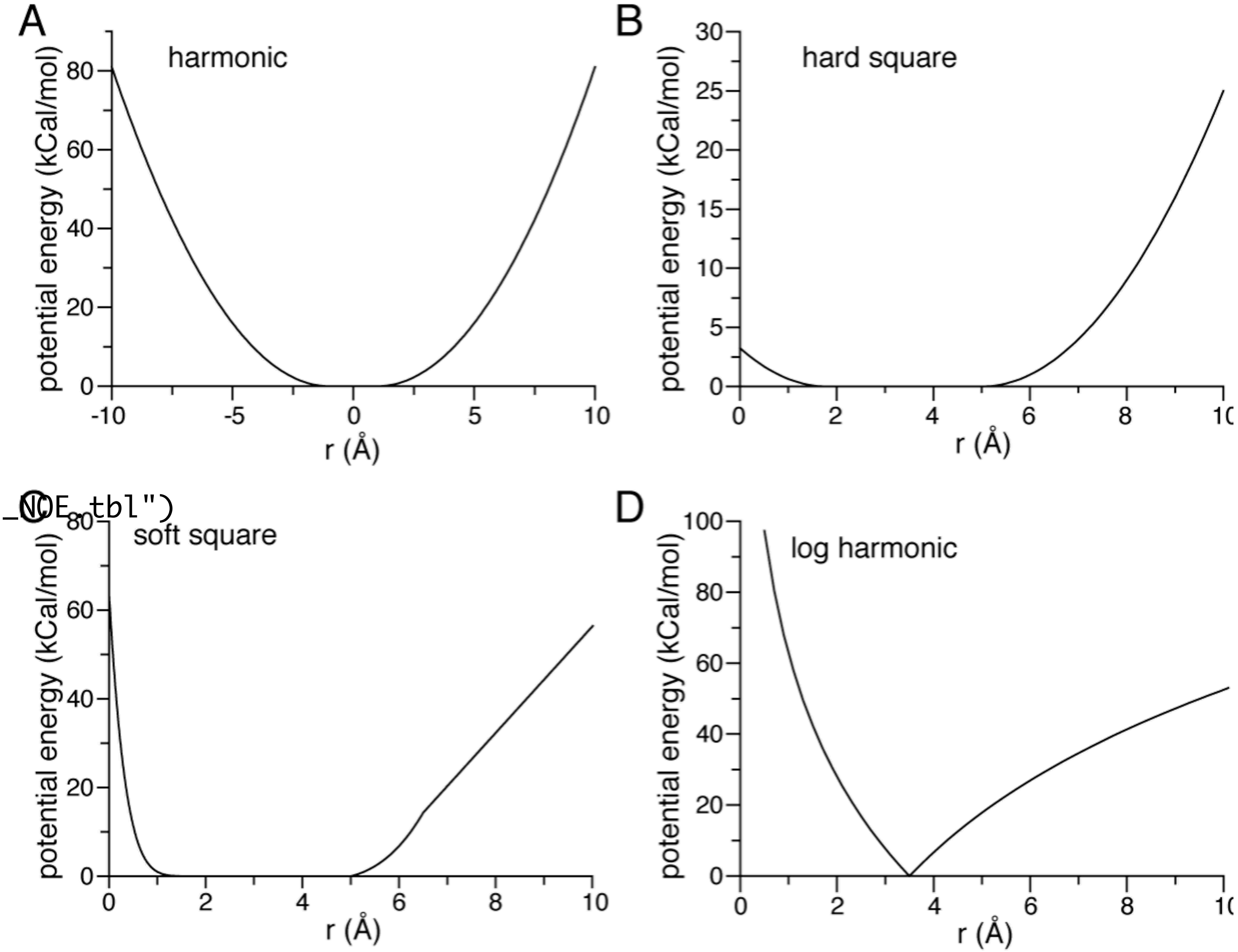
$$V(R) = \begin{cases} (R - d - d_{max})^2 & \text{for} R > d + d_{max} \\ (R - d + d_{min})^2 & \text{for} R < d - d_{min} \\ 0 & \text{in between} \end{cases}$$

a line from NOE restraint file

```
assign (resid 2 and  name HA )(resid  19 and  name HB# ) 4.0 2.2 1.0 !
```

距离约束函数

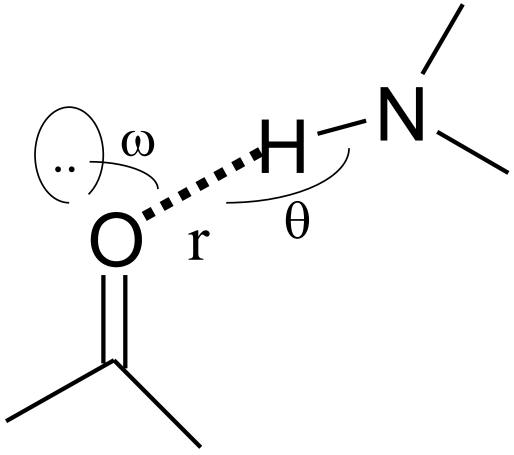
```
# set up NOE potential
from noePotTools import create_NOEPot
noe = create_NOEPot("noe",file="protG_NOE.tbl")
noe.setPotType( "soft" )
noe.setRSwitch( 0.5 )
noe.setAsympSlope( 1. )
noe.setSoftExp(1.)
noe.setThreshold(0.5)
print(noe.info())
potList.append(noe)
```



J. Mol Bol. 2020

Hydrogen-bond restraints

```
assign (resid   20 and  name N    )(resid    1 and  name O    ) 3.3 0.8 0.2
assign (resid    3 and  name N    )(resid   18 and  name O    ) 3.3 0.8 0.2
assign (resid   18 and  name N    )(resid    3 and  name O    ) 3.3 0.8 0.2
assign (resid    5 and  name N    )(resid   16 and  name O    ) 3.3 0.8 0.2
assign (resid   16 and  name N    )(resid    5 and  name O    ) 3.3 0.8 0.2
assign (resid    7 and  name N    )(resid   14 and  name O    ) 3.3 0.8 0.2

assign (resid   55 and  name HN   )(resid   42 and  name O    ) 2.3 0.8 0.2
assign (resid   44 and  name HN   )(resid   53 and  name O    ) 2.3 0.8 0.2
assign (resid   53 and  name HN   )(resid   44 and  name O    ) 2.3 0.8 0.2
assign (resid   46 and  name HN   )(resid   51 and  name O    ) 2.3 0.8 0.2
assign (resid   50 and  name HN   )(resid   46 and  name O    ) 2.3 0.8 0.2
assign (resid   51 and  name HN   )(resid   46 and  name O    ) 2.3 0.8 0.2
assign (resid   22 and  name O    )(resid   26 and  name HN   ) 2.3 0.8 0.2
```



```
from hbPotTools import create_HBPot hb = create_HBPot('hb')
hb. setScale (2.5)
potList .append( hb )
```

Dihedral restraints

```
                                                                            Scale           Exponent
assign (resid   10 and name c )   (resid   11 and name n )
       (resid   11 and name ca)   (resid   11 and name c )  1.0 −120.0 30.0 2
```

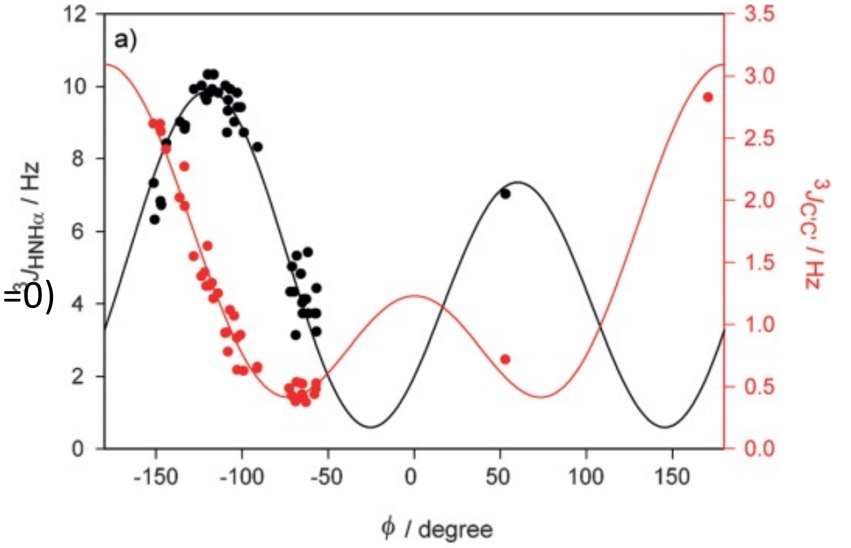
```
from dihedralPotTools import create_DihedralPot dihePot = create_DihedralPot ( ’ dihePot ’ ,
" dihed_g_all . tbl " )
```

J-coupling restraints

$$^3J = A\cos^2(\theta + \theta^*) + B\cos(\theta + \theta^*) + C$$

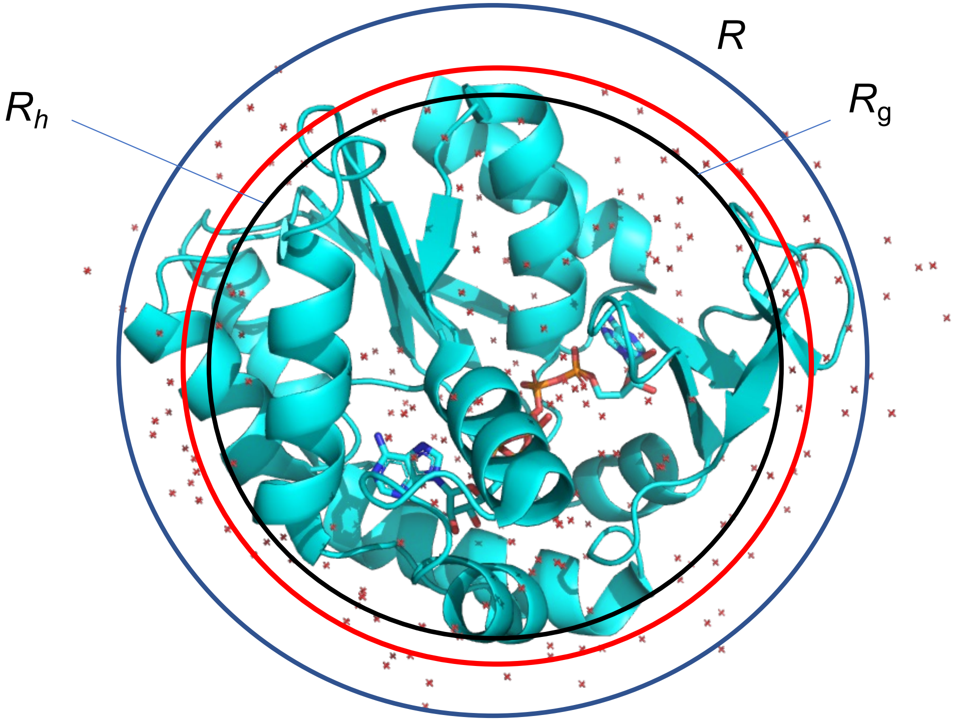
```
from jCoupPotTools import create_JCoupPot
jCoup = create_JCoupPot("hnha","jna_coup.tbl", A=15.3,B=−6.1,C=1.6,phase=0)
```

```
print( Jhnha.rms() )
print ( Jhnha. violations () )
print( Jhnha.showViolations() )
```



Radius of Gyration Restraints

Slightly expanded



Sphere $R_g^2 = 3R^2/5$

Radius of Gyration Restraints

Slightly expanded

$$R_g = 2.2N_{res}^{0.38}$$

```
protocol.initCollapse("resid 4:134", scale=25.0, Rtarget=14.0)

potList.append( XplorPot('COLL') )
```

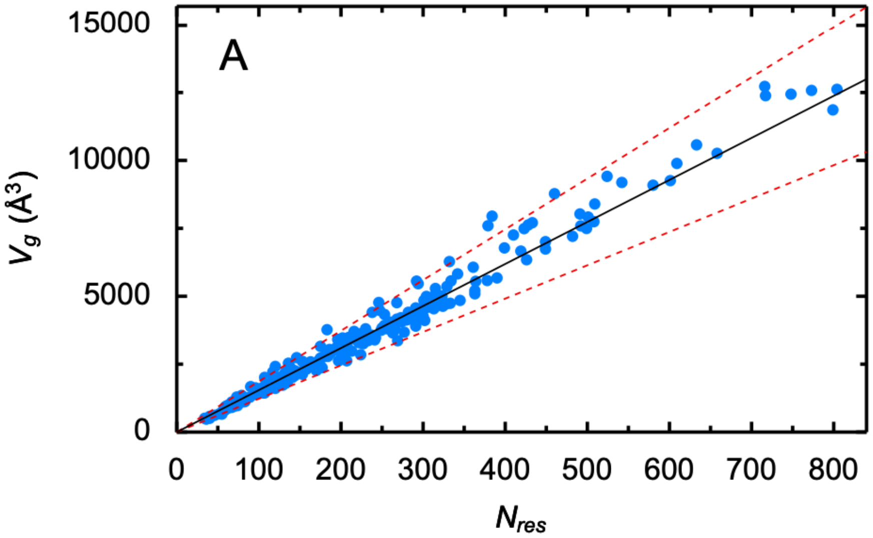
COLLAPSE term

Volume of Gyration Restraints

$$V_g \approx V_g^{res} N_{res}$$

$$\begin{aligned} E_{gyr} \quad &= \quad w_{gyr}\Big(w_{gyr}^{(1)} E_p(V_g - V_g^{res}; 0) \\ &\quad + w_{gyr}^{(2)} E_p(V_g - V_g^{res}; \Delta V_g)\Big) \end{aligned}$$

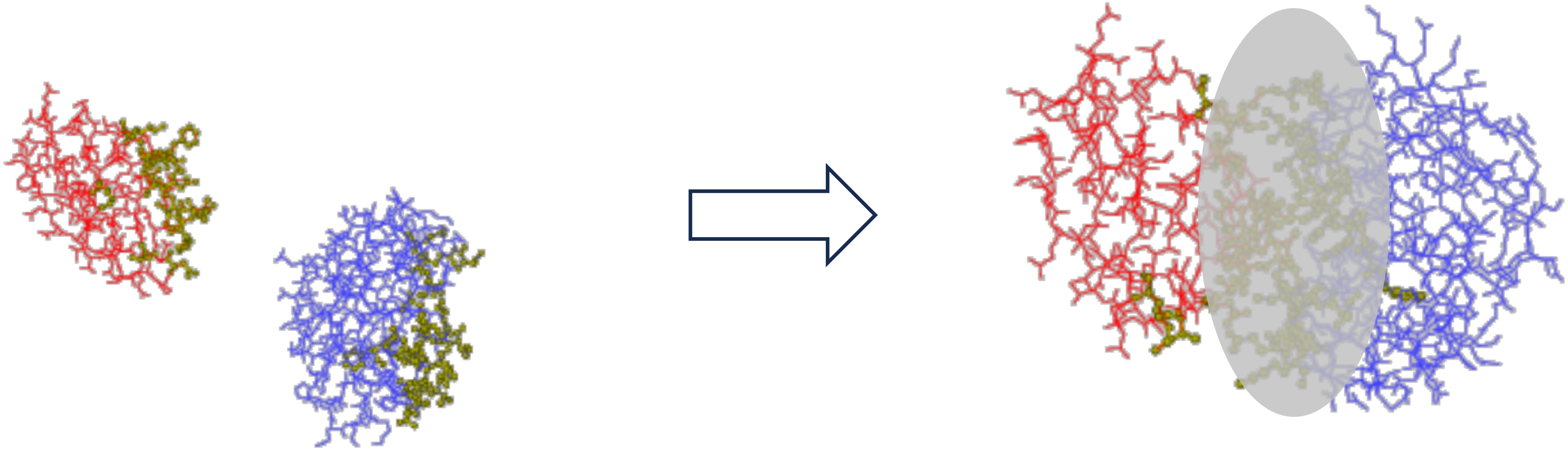
$$E_p(x, \Delta x) = \begin{cases} (x - \Delta x)^2 & \text{for } x > \Delta x \\ (x + \Delta x)^2 & \text{for } x < -\Delta x \\ 0 & \text{otherwise} \end{cases}$$



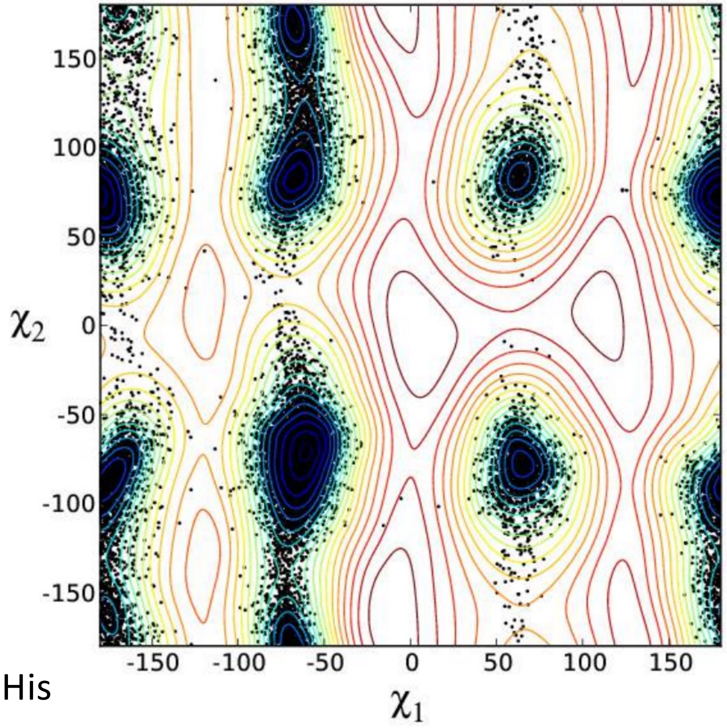
```
from gyrPotTools import create_GyrPot
gyr = create_GyrPot('Vgyr','not resname ANI')
potList.append(gyr)
```

Gyration Restraints used in docking

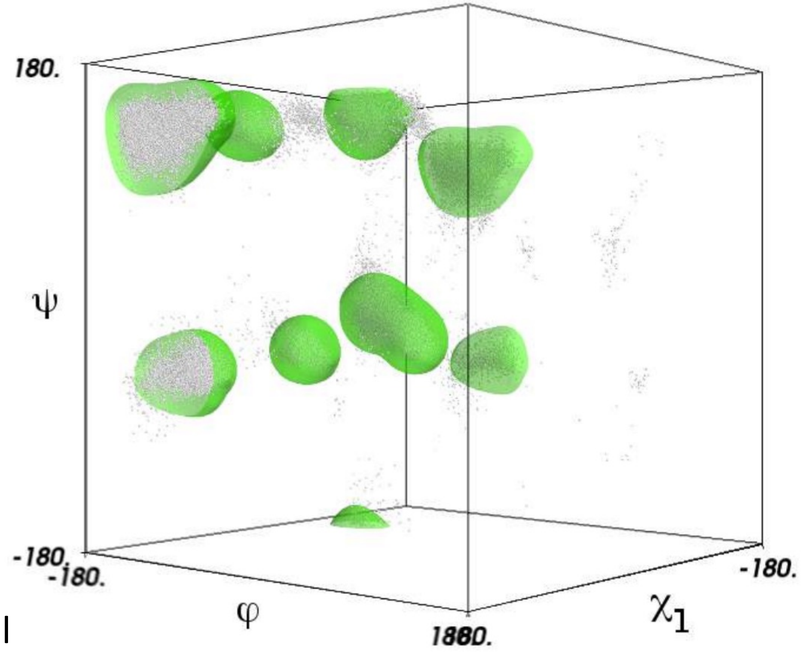
Mapped interfacial residues



Database potential 二面角的相关性



His

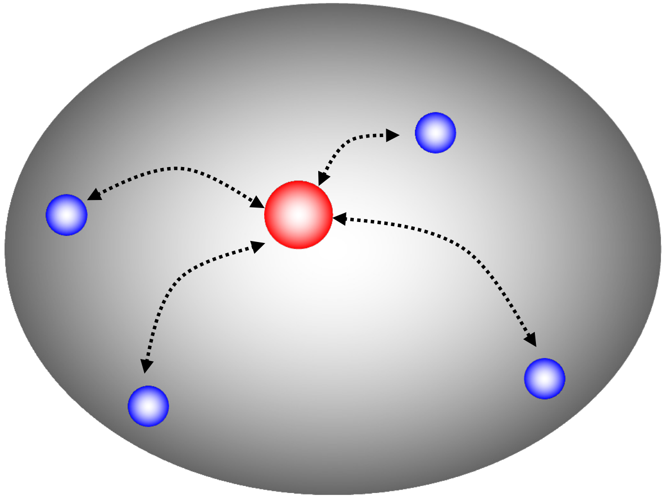


Val

```
from torsionDBPotTools import create_TorsionDBPot
torsionDB=create_TorsionDBPot(’torsionDB’, selection=’not recall nTerminus’)
potList.append( torsionDB ) rampedParams.append(
MultRamp(.002,2,"torsionDB.setScale(20)") )
```

Paramagnetic Relaxation Enhancement (PRE)

$$\Gamma_2 = \frac{1}{15}\left(\frac{\mu_0}{4\pi}\right)^2 \frac{\gamma_I^2 g_e^2 \mu_B^2 S\left(\boxed{S}+1\right)}{r^6}\left[4\tau_c + \frac{13\tau_c}{1+\omega_S^2\tau_c^2} + \frac{3\tau_c}{1+\omega_I^2\tau_c^2}\right]$$

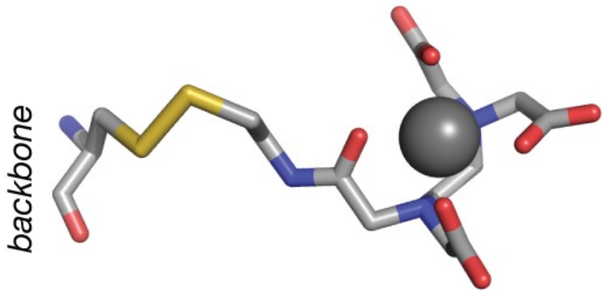


Solomon-Bloembergen-Morgan equation

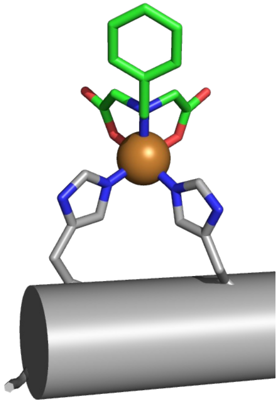
● nuclei

● paramagnetic center

PRE restraints



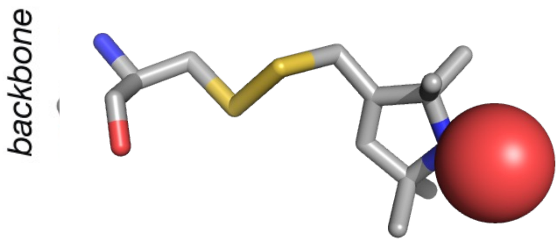
EDTA-Mn^{2+}



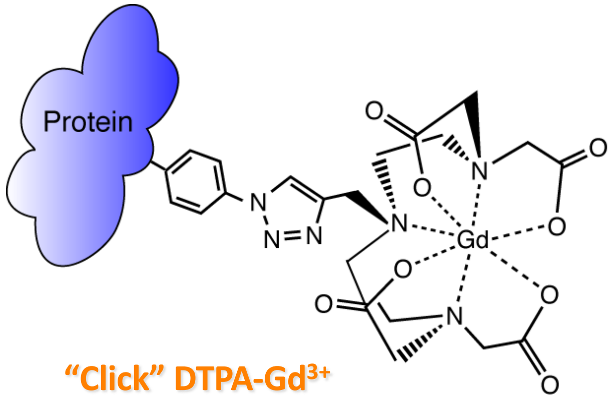
diHis-Cu^{2+}

- Spin quantum number
- Correlation time

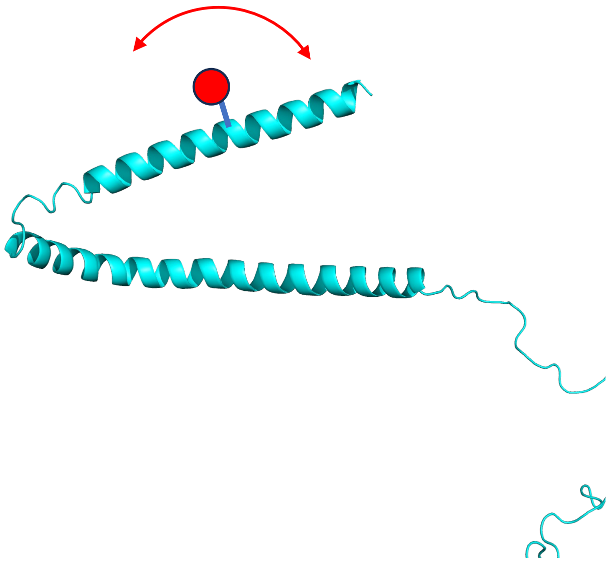
$$\tau_C^{-1} = \tau_e^{-1} + \tau_r^{-1} \boxed{+ \tau_i^{-1}}$$



TEMPO

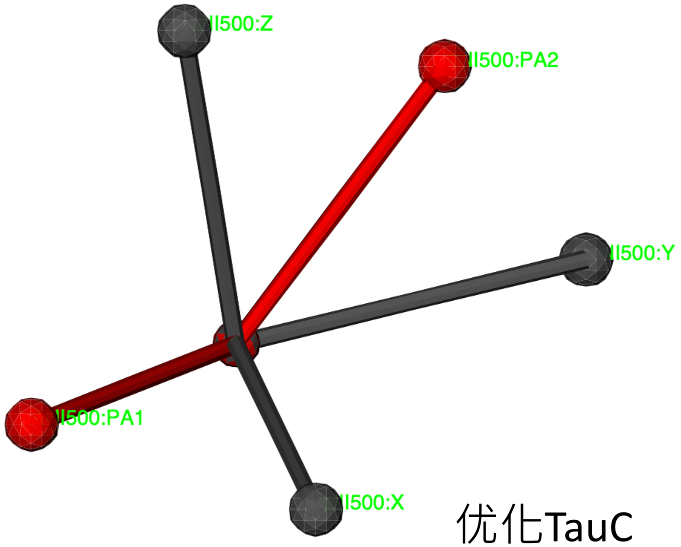


"Click" DTPA-Gd^{3+}



PRE restraints

$$\Gamma = S_{AB}(\tau_c) r_{AB}^{-6}$$



优化TauC

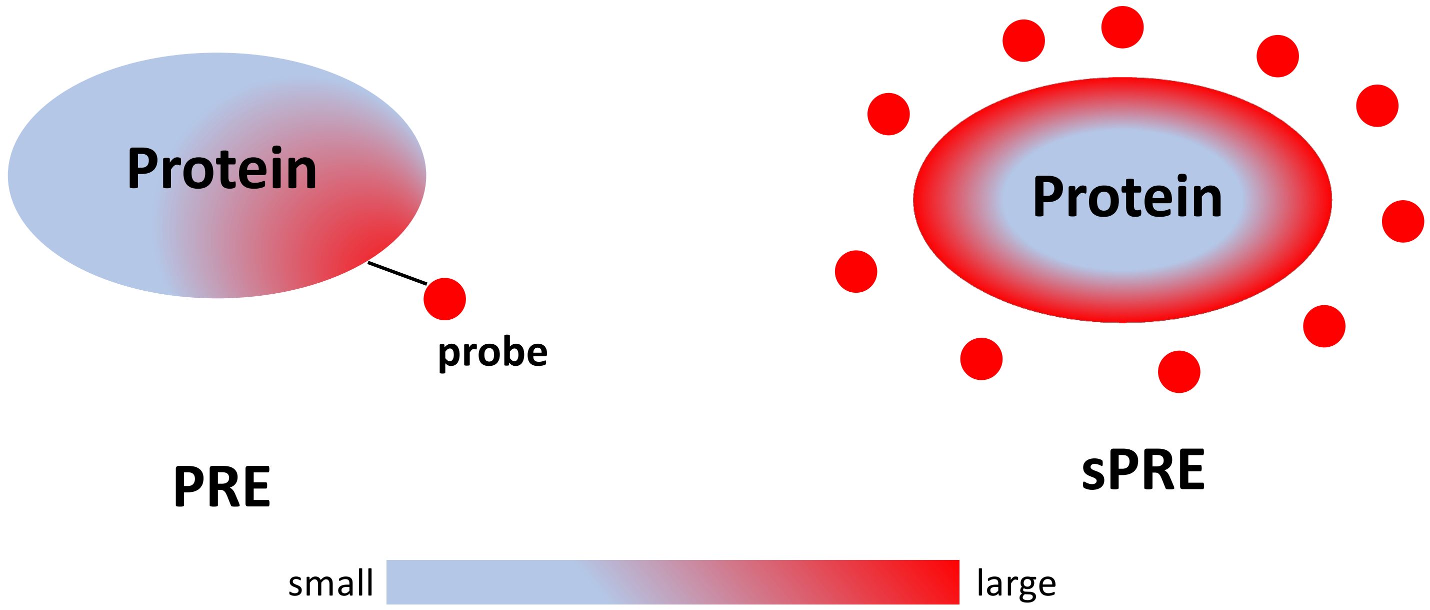
```
from prePotTools import create_PREPot
pre = create_PREPot("pre","file.tbl",
                    eSpinQuantumNumber=2.5,
                    freq=500,            # Larmor frequency in MHz
                    tauc=3.0,            # correlation time in ns
                    fixTau=True)
potList.append(pre)
```

PRE restraints

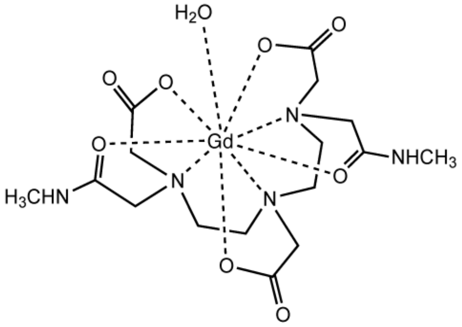
```
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 8 and name HN) 29.8 3.2
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 24 and name HN) 33.3 5.3
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 25 and name HN) 25.4 3.8
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 28 and name HN) 17.8 2.4
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 29 and name HN) 12.9 1.9
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 30 and name HN) 9.9 1.8
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 31 and name HN) 9.9 3.4
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 32 and name HN) 12.4 2.4
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 33 and name HN) 13.8 1.5
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 34 and name HN) 17.0 1.0
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 35 and name HN) 26.2 1.7
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 53 and name HN) 25.3 3.4
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 54 and name HN) 27.5 2.9
```

Solvent PRE (sPRE)

A label-free approach

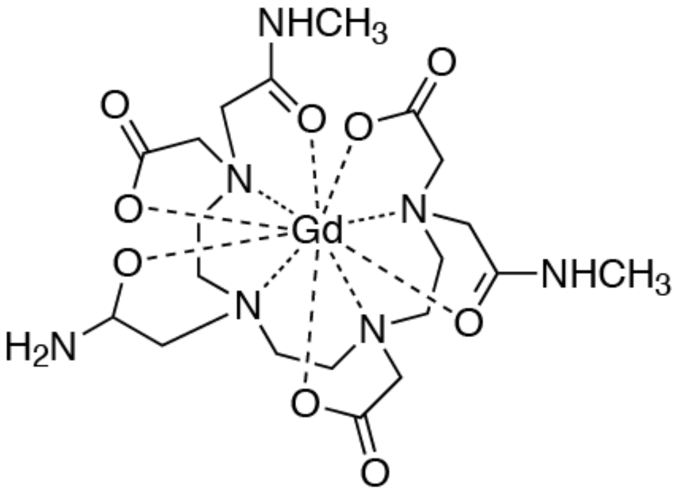


Solvent PRE probes

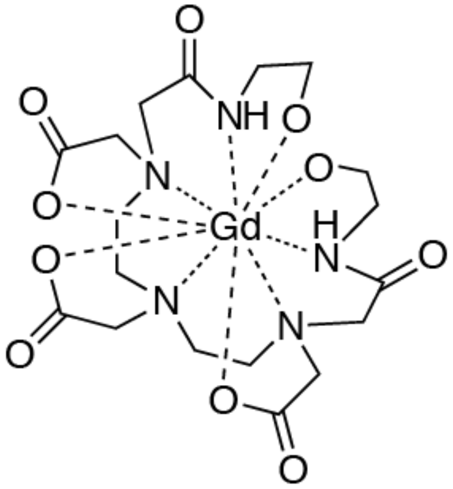


Gd(III)-DTPA-BMA

Omniscan



Gd(III)-TTHA-TMA



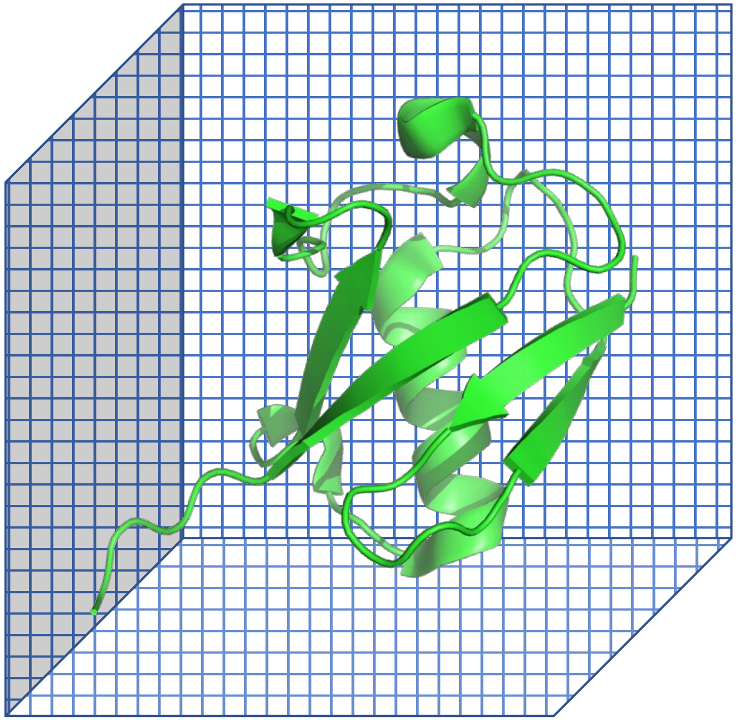
Gd(III)-DTPA-BEA

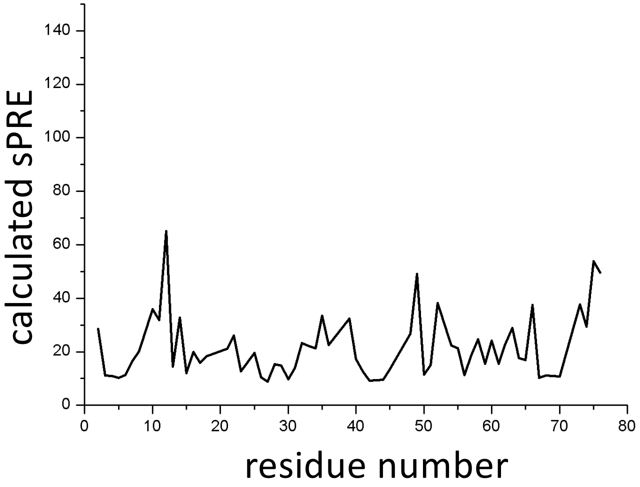
J. Bio NMR 2015

Angew Chem 2016

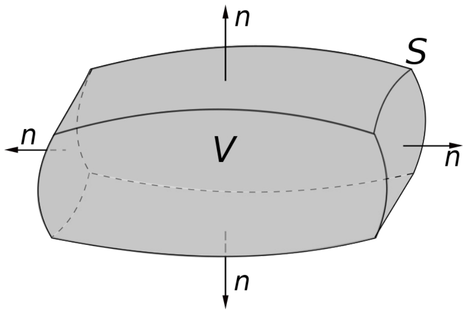
sPRE的直观计算

Grid model





sPRE restraints



Volume integral → surface integral

$$\Gamma_{\mathrm{sPRE}} \sim \int_{V_e} dv\; k' 1/r^6$$

$$\Gamma_{\mathrm{sPRE}} = -k'/3 \int_S ds\; \mathbf{n} \cdot \mathbf{r}/|\mathbf{r}|^6$$

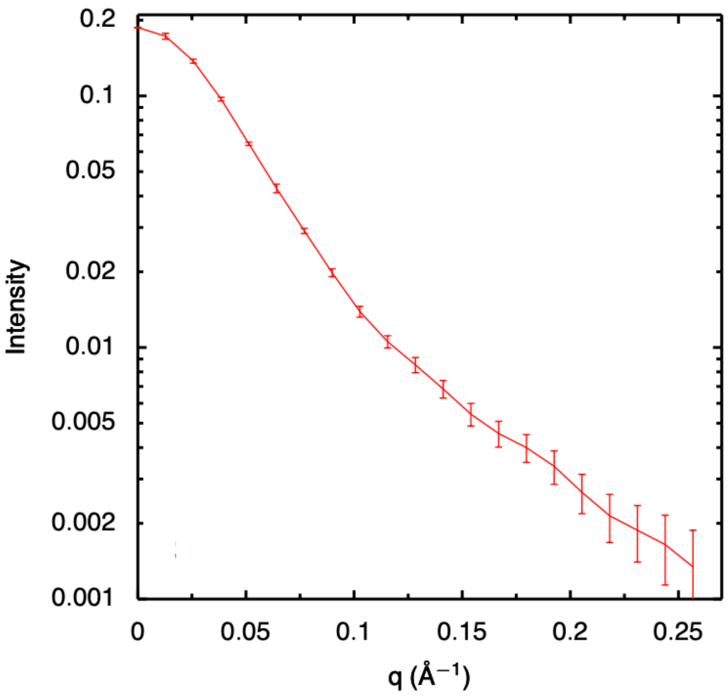
n is the outward pointing surface normal

r is distance from surface to nucleus

```
from psolPotTools import create_PSolPot
psol = create_PSolPot("psol",file='sPRE.tbl')
psol.setRmin(0.1)
psol.setThreshold(0)
psol.setProbeRadius(4.0)
psol.setTargetType("correlation")
potList.append(psol)
```

Methods 2018

整合其他实验数据 -- SAXS



```
from solnXRayPotTools import create_solnXRayPot
import solnXRayPotTools
xray=create_solnXRayPot('xray',
                        experiment='saxs.dat',
                        numPoints=26,
                        normalizeIndex=-3,preweighted=False)

xrayCorrect=create_solnXRayPot('xray-c',
                               experiment=saxs.dat',
                               numPoints=26,
                               normalizeIndex=-3,preweighted=False)

solnXRayPotTools.useGlobs(xray)
xray.setNumAngles(50)
xrayCorrect.setNumAngles(500)
potList.append(xray)
crossTerms.append(xrayCorrect)

#corrects I(q) for globbing, small angular grid and
# includes solvent contribution corrections
from solnScatPotTools import fitParams
rampedParams.append( StaticRamp("fitParams(xrayCorrect)") )
rampedParams.append(StaticRamp("xray.calcGlobCorrect(xrayCorrect.calcd())"))
```

Globbing

系综优化 ensemble refinement

Ambiguous Restraints

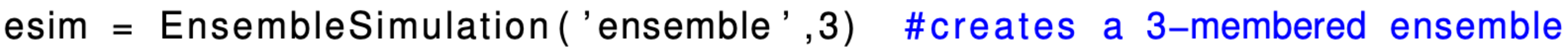
```
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 8 and name HN) 29.8 3.2
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 24 and name HN) 33.3 5.3
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 25 and name HN) 25.4 3.8
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 28 and name HN) 17.8 2.4
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 29 and name HN) 12.9 1.9
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 30 and name HN) 9.9 1.8
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 31 and name HN) 9.9 3.4
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 32 and name HN) 12.4 2.4
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 33 and name HN) 13.8 1.5
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 34 and name HN) 17.0 1.0
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 35 and name HN) 26.2 1.7
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 53 and name HN) 25.3 3.4
assign (segid ALT* and resid 41 and name MN) (segid  " " and resid 54 and name HN) 27.5 2.9
```

Segid CNF1

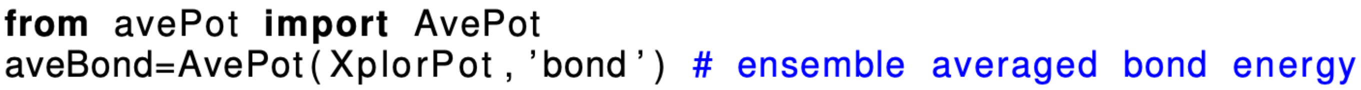
or Segid CNF2

…

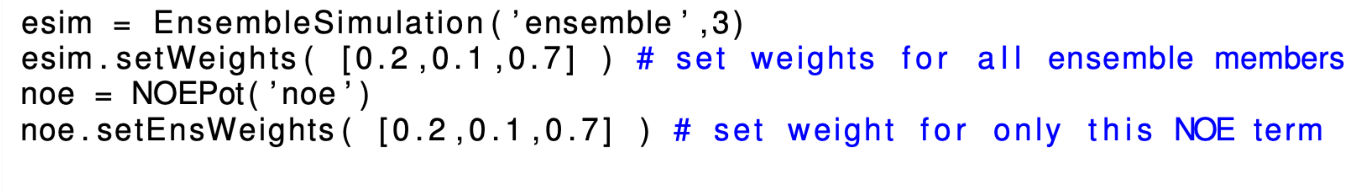
系综优化 ensemble refinement



```
esim = EnsembleSimulation(’ensemble’,3)  #creates a 3−membered ensemble
```

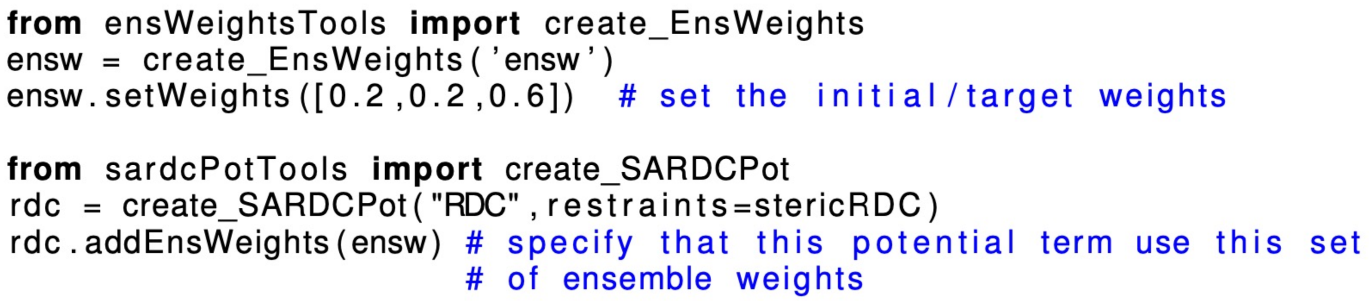


```
from avePot import AvePot
aveBond=AvePot(XplorPot, ’bond’) # ensemble averaged bond energy
```



```
esim = EnsembleSimulation(’ensemble’,3)
esim.setWeights( [0.2,0.1,0.7] ) # set weights for all ensemble members
noe = NOEPot(’noe’)
noe.setEnsWeights( [0.2,0.1,0.7] ) # set weight for only this NOE term
```

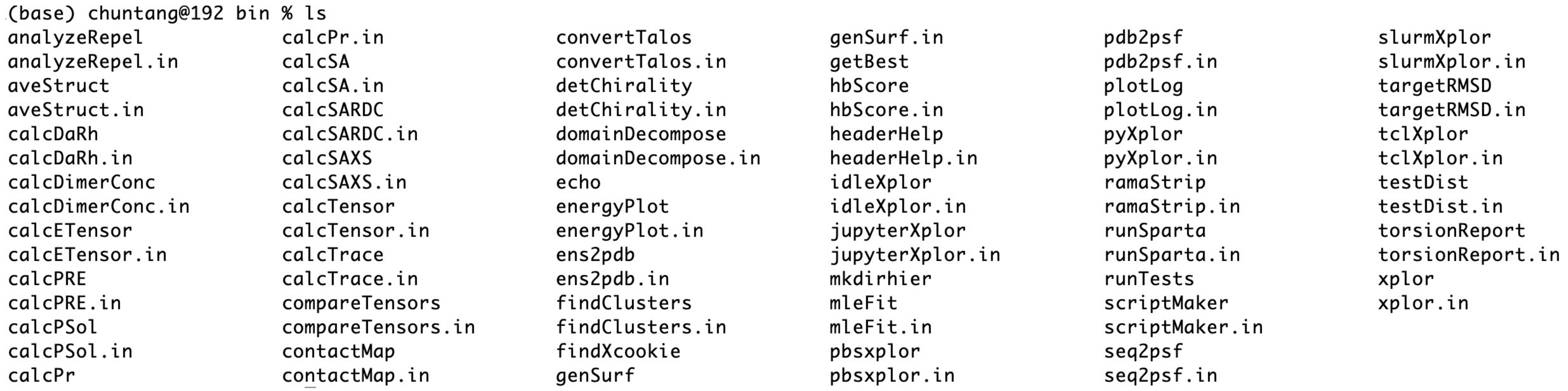
不同比例的贡献



```
from ensWeightsTools import create_EnsWeights
ensw = create_EnsWeights(’ensw’)
ensw.setWeights([0.2,0.2,0.6])  # set the initial/target weights

from sardcPotTools import create_SARDCPot
rdc = create_SARDCPot("RDC",restraints=stericRDC)
rdc.addEnsWeights(ensw) # specify that this potential term use this set
                        # of ensemble weights
```

对权重进行优化

Xplor-NIH自带的小程序



```
(base) chuntang@192 bin % ls
analyzeRepel            calcPr.in               convertTalos            genSurf.in              pdb2psf                 slurmXplor
analyzeRepel.in         calcSA                  convertTalos.in         getBest                 pdb2psf.in              slurmXplor.in
aveStruct               calcSA.in               detChirality            hbScore                 plotLog                 targetRMSD
aveStruct.in            calcSARDC               detChirality.in         hbScore.in              plotLog.in              targetRMSD.in
calcDaRh                calcSARDC.in            domainDecompose         headerHelp              pyXplor                 tclXplor
calcDaRh.in             calcSAXS                domainDecompose.in      headerHelp.in           pyXplor.in              tclXplor.in
calcDimerConc           calcSAXS.in             echo                    idleXplor               ramaStrip               testDist
calcDimerConc.in        calcTensor              energyPlot              idleXplor.in            ramaStrip.in            testDist.in
calcETensor             calcTensor.in           energyPlot.in           jupyterXplor            runSparta               torsionReport
calcETensor.in          calcTrace               ens2pdb                 jupyterXplor.in         runSparta.in            torsionReport.in
calcPRE                 calcTrace.in            ens2pdb.in              mkdirhier               runTests                xplor
calcPRE.in              compareTensors          findClusters            mleFit                  scriptMaker             xplor.in
calcPSol                compareTensors.in       findClusters.in         mleFit.in               scriptMaker.in
calcPSol.in             contactMap              findXcookie             pbsxplor                seq2psf
calcPr                  contactMap.in           genSurf                 pbsxplor.in             seq2psf.in
```

torsionReport - collect and average protein torsion angle values

aveStruct - average structures and report per-atom RMSD to the mean- unregularized.

pairRMSD.py - report pairwise RMSD

calcTensor - calculate an SVD alignment tensor and report back-calculated RDC values given one or more structures.

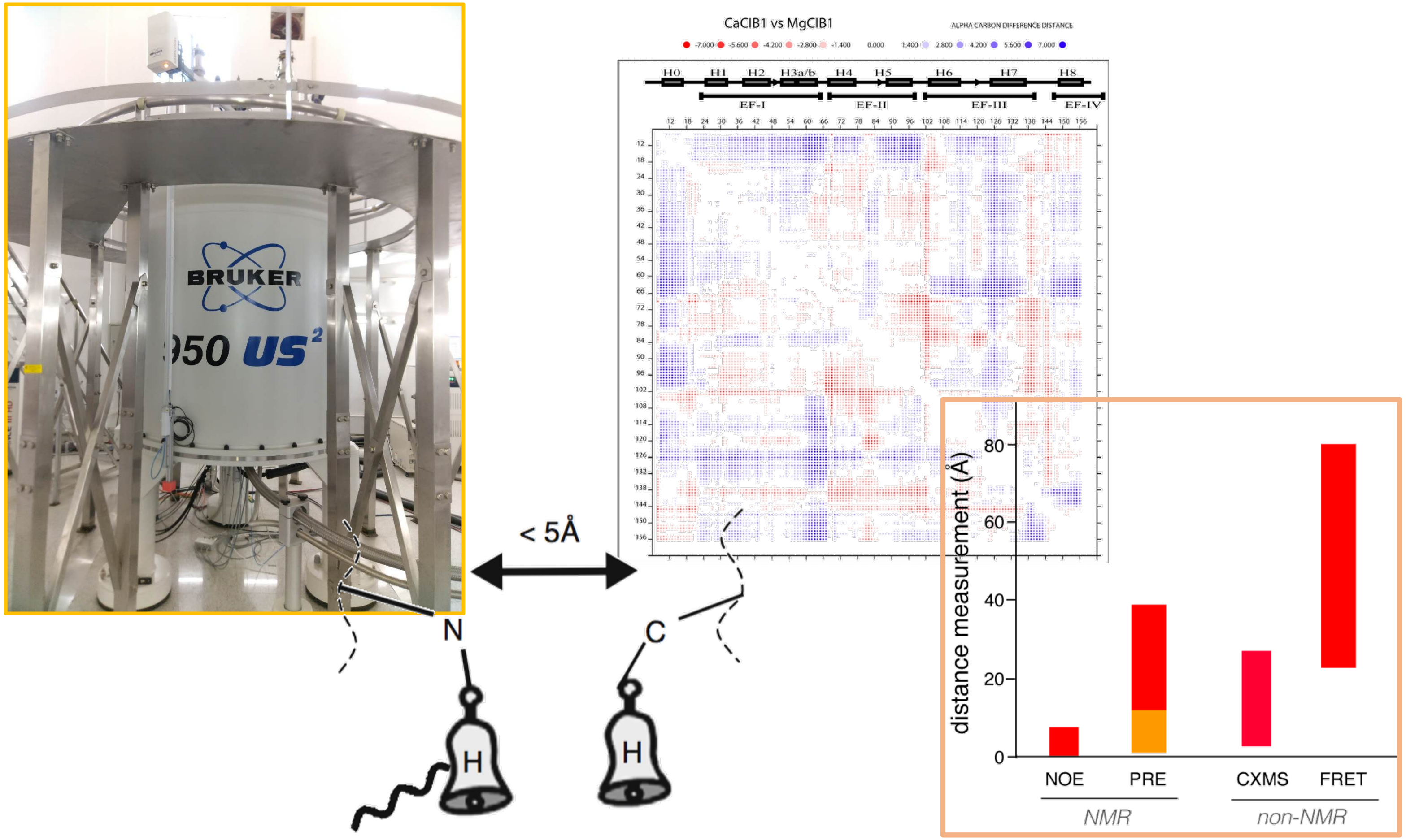
calcDaRh - calculate estimates of *Da* and rhombicity given only RDC values (no structures)

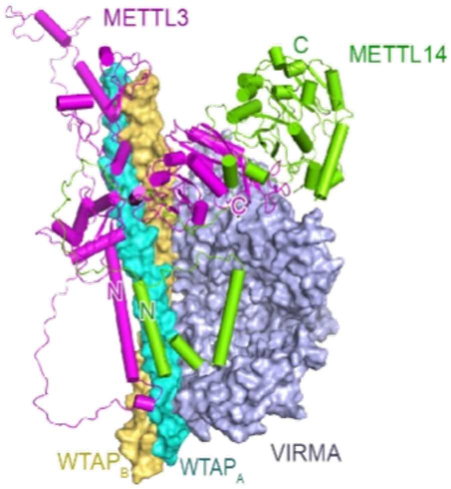
calcSAXS - given a structure, calculate a SAXS or SANS curve, optionally comparing with experiment.

calcPRE - Compute and optionally plot PRE values given a molecular structure and a restraint list.

calcPSol - Compute the solvent PRE given a molecule structure and a restraint list.

谢谢大家





http://www.tanglab.cn

http://tanglab.pku.edu.cn

Cell Res. 2022