Gromacs 中文教程（II）：结果分析

淮海一粟

**MD结果分析**

模拟结束后，就可以分析数据了。分析包括三个阶段。首先，有必要对模拟的质量进行检查，如果检查结果表明模拟良好，那么就可利用该模拟对所研究的问题作出回答；最后，不同的模拟结果可以合并。

注：文件名应该反映文件内容，这根据你的模拟系统不同而不同。这里我们假定使用默认文件名，那么就会产生以下文件：

* **topol.tpr**: 模拟开始时包含完整系统描述的输入文件；
* **confout.gro**: 结构稳健，包含最后一步的坐标和速度；
* **traj.trr**: 全部精确轨迹，包括位置、速度和随时间变化的力
* **traj.xtc**: 压缩的轨迹文件，只包含低精度(0.001 nm)的坐标信息；
* **ener.edr**: 随时间变化的有关能量数据
* **md.log**: 模拟过程的日志

附注：许多分析工具都能生成.xvg文件。这些文件能用xmgr或xmgrace查看，也可用python脚本程序[xvg2ascii.py](http://md.chem.rug.nl/~mdcourse/xvg2ascii.py)在终端显示出来。

Each group writes one report. For general questions a single answer should be given in the report. Questions specific to each simulation should be given in table, indicated with ( T ). Answers to questions from one section should be combined in a single table if possible.

## 先检查一下结果

在开始分析前，要确定模拟是否正确地完成。有很多原因会导致模拟中断，尤其是与力场和系统平衡不充分引起的问题。为了检查模拟是否正确完成，运行gmxcheck程序：

gmxcheck -f traj.xtc

看看是否执行了10纳秒的模拟。

==Q== How many frames are in the trajectory file and what is the time resolution? ( T )

另一个重要的信息源是日志文件。在文件md.log的末尾有模拟过程的统计数据；包括内存和CPU的使用情况和模拟时间。看看日志文件的末尾，使用'less'命令时，你可以用'G' (shift-g) 命令，跳到文件末尾。

==Q== How long did the simulation run in real time (hours), what was the simulation speed (ns/day) and how many years would the simulation take to reach a second? ( T )

==Q== Which contribution to the potential energy accounts for most of the calculations?

Don't be afraid to use the [Gromacs online manual](http://manual.gromacs.org/current/), to search in the [Gromacs mailing list](http://www.gromacs.org/Support/Mailing_Lists/Search) or even to use [Google](http://www.google.com) to get information about the terminology used by Gromacs.

## 结果可视化

好玩的环节开始了。虽然很多分析都能归结为从轨迹文件中提取图像，MD当然首先要关注系统的移动。来看看轨迹文件。

首先用gromacs提供的查看器ngmx来看看。虽然该软件的完善程度和视觉效果不及其他查看器，但它能够在拓扑文件信息的基础上画出键。其他查看器可能隐含远程键，这可能导致这些键被认为太长而不画出，或者会在非常接近的原子之间画出键。这是对模拟结果分析的一个常见错误源。使用ngmx载入拓扑和轨迹文件：

ngmx -s topol.tpr -f traj.xtc

看看程序菜单，试试不同的选项。播放动画。观看过程通过右边的选项控制。右击或左击选择选项来改变查看。

==Q== What happens if the protein diffuses over the boundary of the box?

为了视觉美观，我们将从轨迹文件中提取1000帧 (-dt 10)并忽略水分子（当软件询问时，选择 Protein）。而且，我们还将忽略边界的跨越，作出连续轨迹（-pbc nojump）。为了做这些工作，我们使用瑞士军刀般的gromacs工具trjconv，该工具有1001个选项组合。我们用它写出一个多模型的pdb文件，从而能在Pymol中观看。

trjconv -s topol.tpr -f traj.xtc -o protein.pdb -pbc nojump -dt 10

在Pymol中提取轨迹文件：

pymol protein.pdb

当所有的帧装入完毕，播放动画。

Mplay

动画播放时，其他控制键仍在运行，可以用鼠标旋转、放大或缩小图像，也可以改变分子外观。

Spectrum

Show cell

如果没错的话，你现在能看到蛋白质扩散、翻转跳跃。但我们对内部运动更感兴趣，而不是总体行为。在Pymol中，你能使用命令intra\_fit将其他所有帧与第一帧对齐。随后，你可以用定向工具设定蛋白质中心：

intra\_fit protein and (name c,n,ca)

Orient

现在，所有的帧应该都对齐了，你可以看到蛋白质的哪一部分移动得更厉害。这些差异将在以后定量分析。

当然，在cartoon模式下，蛋白质看起来更舒服，试试这个命令：

show cartoon

因为.pdb文件里面没有二级结构信息，你可能会看到碳骨架是个粗管状，而看不到正确的二级结构。Pymol可以自动计算蛋白质的二级结构，但只计算一帧，并将其映射到其它的帧。例如，下述命令可以计算第一帧的二级结构：

dss

通过设定状态，用于计算的帧可以改变：

dss state=1000

最后，同时查看所有帧并检查蛋白质的柔性和刚性部分。

set all\_states=1

请随便练习Pymol的使用。试试放大柔性或刚性区域，并检查侧链构象。使用['ray'](http://md.chem.rug.nl/~mdcourse/nerd.html" \l "povray)和 'png'制作一份图像，即使浪费点CPU时间也不要紧。但记住，如果图像太复杂，可能会导致pymol的插件ray-tracer崩溃，这种情况下，你可以直接用'png' 得到屏幕上的图像。

The following part, up to 'quality assurance', is optional and it may be best to first finish the other sections.

如果你有足够机时做此教程，或者你已经做完了前面的功课，那就来做个不错的电影吧。你可能注意到，这些轨迹噪音非常多，那是热噪音，是正常的；但是对于制作好的电影会有影响。我们可以屏蔽这些高频运动，只保留低速和平滑的总体移动。为了达到这个目的，使用g\_filter程序：

g\_filter -s topol.tpr -f traj.xtc -ol filtered.pdb -fit -nf 5

现在，在Pymol中倒入轨迹文件。计算二级结构 (dss)并显示(show cartoon)，隐藏碳骨架上的侧链 (hide lines, not (name c,n,o)) 并给蛋白质上色，然后orient分子设好视角。现在开始制作电影了：

viewport 640,480

set ray\_trace\_frames,1

mpng frame\_.png

现在退出Pymol (quit) 并显示文件路径 (ls)。你能看到，文件的数量多了好些，包括1000个图像。以每秒30帧的速度，将会产生30秒的电影。下载[mpeg\_encode](http://md.chem.rug.nl/~mdcourse/mpeg_encode)程序和参数文件[movie.param](http://md.chem.rug.nl/~mdcourse/movie.param)，用它来产生单帧图像的电影（你可能需要编辑参数文件来改变文件名）：

mpeg\_encode movie.param

## 质量确认

进行了最初的轨迹图像查看后，该对模拟的质量进行彻底检查了。这个质量检查(QA)包括热动力参数（如温度、压力、势能和动能）的收敛情况。更一般地说， QA试图评价（模拟）是否达到平衡状态。 结构上的收敛也需要检查，这个用起始结构和平均结构的均方差 (RMSD)来表示。随后，还要检查邻近的周期性图像之间没有互相作用，因为这将导致非物理学效应。最后，QA检查包括原子的均方差，这个可以与晶体学数据b-factors进行比较。

### 能量收敛

我们首先从能量文件中提取一些热动力学数据。研究以下性质：温度、压力、势能、动能、单位盒子体积、密度和盒子尺寸。这些大部分性质已在系统准备步骤中检查过了。用工具g\_energy进行能量分析。该程序读出能量文件，也就是模拟过程中产生的扩展名为.edr的文件。g\_energy程序将会问需要提取什么参数并将产生一幅图像。输入下面的命令：

g\_energy -f ener.edr -o temperature.xvg

此命令将产生一列能量及其参量，这些参数存贮在.edr能量文件中。本教程的能量文件可能含有68个参量，每个都可以提取并画出图像。 最开始九个对应于力场中的不同能量。还应注意，从第47个开始同时列出了蛋白和非蛋白的参量，及两者之间的相互作用。为了提取温度，输入"14 0" (Gromacs version 4.0.5)，回车。用xmgrace程序看图，看看在指定温度附近(300 K)的温度如何波动。从波动也可以计算体系热容，热容在g\_energy输出文件的结尾。

xmgrace -nxy temperature.xvg

==Q== What is the average temperature and what is the heat capacity of the system? ( T )

通过调用能量参量名，可以自动运行能量文件。使用'echo'和管道命令 ("|")，实现从一个程序到另一个程序的数据传输，g\_energy可以自动回应。为了提取多个参量，每个参量以"\n"划分。拷贝并粘贴，或者输入以下命令行提取其他参量。不幸的是，能量参量必须用数字指定。

echo 15 0 | g\_energy -f ener.edr -o pressure.xvg

echo -e "11\n12\n13 0" | g\_energy -f ener.edr -o energy.xvg

echo 20 0 | g\_energy -f ener.edr -o volume.xvg

echo 21 0 | g\_energy -f ener.edr -o density.xvg

echo -e "17\n18\n19 0" | g\_energy -f ener.edr -o box.xvg

逐个查看这些文件，看看数值的收敛情况。如果数值没有收敛，就表明模拟尚未达到平衡，需要延时才能进行进一步分析。而且，平衡附近的数据不能用于分析。这里，为了简便起见，我们不管他了，直接用这些数据分析。

==Q== What are the terms plotted in the files energy.xvg and box.xvg

==Q== Estimate the plateau values for the pressure, the volume and the density. ( T )

一些参量比其他的收敛慢。特别地，温度很容易收敛而系统各部分的相互作用收敛可能较慢。看看蛋白质和溶剂之间的相互关系：

echo -e "51\n53 0" | g\_energy -f ener.edr -o coulomb-inter.xvg

echo -e "52\n54 0" | g\_energy -f ener.edr -o vanderwaals-inter.xvg

### 周期性图像建的最小距离

对于QA，一个最重要的需要检查的事项是，周期性图像之间不应该有相互作用；因为周期性图像是有独立身份的，这些相互作用在物理学上不应该发生。设想双极性蛋白质图像存在直接的相互作用，那么同一蛋白质在盒子边界的两个末端就会产生作用力，这将影响蛋白质的本身行为并使模拟结果失效。 为了确认这些相互作用没有发生，我们用g\_mindist命令计算周期性图像每次的最小距离：

g\_mindist -f traj.xtc -s topol.tpr -od minimal-periodic-distance.xvg -pi

==Q== What was the minimal distance between periodic images and at what time did that occur? ( T )

==Q== What happens if the minimal distance becomes shorter than the cut-off distance used for electrostatic interactions? Is it the case in your simulations?

小距离的事件发生是偶然性的还是持续性的，也会有影响。如果是持续性的，就可能影响蛋白质动力学；但是如果只是偶尔发生，就一般不会有影响。

==Q== Run now g\_mindist on the C-alpha group, does it change the results? What does is mean for your system?

不只是直接相互作用需要关注，那些由水分子介导的间接效应，也会产生问题。例如，蛋白质可能影响最初四层水分子的排列，这相当于1 nm的距离；理想的情况是，最小距离不应该小于2nm。

### 波动的均方根

除了能量本身的检查外，还应该检查模拟过程中，松弛后的蛋白质向平衡状态收敛的情况。通常，松弛只是考虑了与参考结构（例如，晶体结构）的Euclidean距离。这个距离以 均方根偏差 (RMSD)表示。然而，我们推荐再检查一下与平均结构的松弛情况，也就是说，与平均结构的RMSD。个中原因，将在下一段说明。但为了计算与平结构的RMSD，需要首先得到平均结构。这个结构可在计算波动均方根(RMSF)时附加得到。

RMSF的捕获每个原子相对平均位置的波动。这能深入观察蛋白质柔性部位的性质，相应于晶体结构测定中的b-factors (温度因子)。通常，我们希望RMSF与b-factors相近，这能表明模拟结果与晶体结构相适应。RMSF (和平均结构)用g\_rmsf计算。-oq选项运行计算b-factors，并把它们添加到参考结构中。我们最关心每个残基的波动，可用选项-res设定。

g\_rmsf -f traj.xtc -s topol.tpr -o rmsf-per-residue.xvg -ox average.pdb -oq bfactors.pdb -res

用xmgrace看看RMSF，找出柔性和刚性区域。

==Q== Indicate the start and end residue for the most flexible regions and the maximum amplitudes. ( T )

==Q== Compare the results from the different proteins. Are there differences? If yes, which is the most flexible and which least?

为了获得这些结果关联性的印象，这里有个人类prion蛋白质1qlz 的RMSF，图中会导致CJD疾病的突变残基被标示出来。

Many 
mutations that are known to cause
CJD fall in the regions around the most flexible loop of the human prion
protein

现在来看看两个pdb文件，把它们读入Pymol。然后根据b-factors给结构文件bfactors.pdb上色，检查柔性区域。平均结构是非物理结构。看看其中的一些侧链，并注意平均结构对其构想的影响。

pymol average.pdb bfactors.pdb

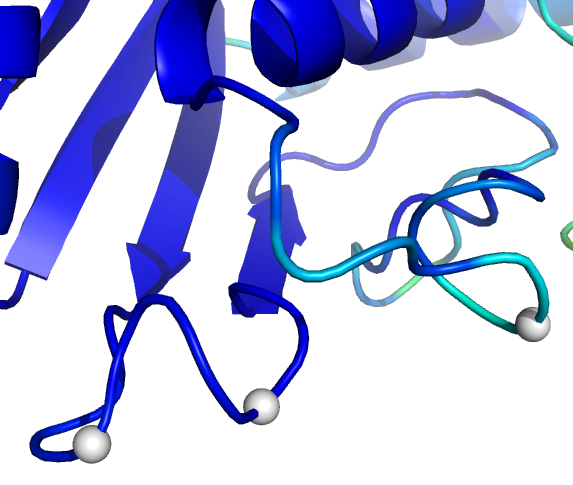
spectrum b, selection=bfactors

颜色分布在b-factor范围内，蓝色代表最低（最稳定），红色代表最高（最易波动）。你可以减少最大值来调整颜色范围，如设为500:

Q = 500; cmd.alter("all", "q = b > Q and Q or b"); spectrum q

以下图像显示的是人类野生型UbcH8蛋白根据模拟计算得到的b-factors上的颜色。蓝色是低，红色是高。白色区域表示目前已知的能够反转蛋白质相互作用特征的残基。在图像右侧，你可以看到那些在Helix 2前后的loops有较高的b-factors。第二个图像是helix 2前loop的放大显示。

Most of the structure has low b-factors, except for some loops and the 
third helix



### RMSD的收敛

当心！因为你的蛋白质可能会跳出盒子外， 我们必须重新制作轨迹来把中间周期性图像中的粒子拽回来。为此，运行以下命令：

trjconv -f traj.xtc -o traj\_nojump.xtc -pbc nojump

由于RMSF计算也给出平均结构，我们现在计算均方根偏差（RMSD）。RMSD通常被用作指示到平衡状态的收敛情况。前面说过，RMSD仅仅是个距离单位，值越小越好。RMSD用g\_rms程序计算。首先计算所有原子的RMSD，使用起始结构作为参考：

g\_rms -f traj\_nojump.xtc -s topol.tpr -o rmsd-all-atom-vs-start.xvg

==Q== If observed, at what time and value does the RMSD reach a plateau? ( T )

再次计算RMSD，但只计算骨架原子：

g\_rms -f traj\_nojump.xtc -s topol.tpr -o rmsd-backbone-vs-start.xvg

这次，RMSD达到了一个低值，这是由于排除了侧链原子（侧链原子往往更容易发生运动）。两个RMSD最后都应到达一个平台值。这意味着蛋白质结构达到了与参考结构有一定距离并或多或少保持那个距离。然而，随着距离的增加，可能构象的数目也在增加。这意味着虽然RMSD达到了一个平台值，结构仍然可能在向平衡状态收敛。出于这个原因，建议同时检查一下向平均结构的收敛情况。

echo 1 | trjconv -f traj\_nojump.xtc -s topol.tpr -o protein.xtc

g\_rms -f protein.xtc -s average.pdb -o rmsd-all-atom-vs-average.xvg

g\_rms -f protein.xtc -s average.pdb -o rmsd-backbone-vs-average.xvg

比较与上次图像的差别。注意RMSD值的平衡点。

==Q== Briefly discuss two differences between the graphs against the starting structure and against the average structure. Which is a better measure for convergence?

### 回旋半径的收敛

QA的最后一步，我们将计算回旋半径。这个值表示每次分子形状的变化。将回旋半径与试验得出的回旋半径相比较。可以用g\_gyrate计算回旋半径。该程序也会给出个性因子（individual components），相应于惯性矩阵的本征值。这意味着，第一个individual component对应于原子的最长轴，最后一个individual component相应于最小轴。这三个轴能说明分子的形状。输入以下命令：

g\_gyrate -f traj.xtc -s topol.tpr -o radius-of-gyration.xvg

看看回旋半径和individual components，注意这些值如何达到平衡值。

==Q== At what time and value does the radius of gyration converge? ( T )

这里，我们完成了分析的第一部分，包括图像检查和质量检查。现在该深入一点了，挖掘一下蛋白质内部发生的情况。第二部分的分析包括结构特性，这些特性可用蛋白质形状，例如氢键数量、溶剂可接近表面或特定原子-原子之间的距离等。Let’s go。

## 结构分析：由形状得出的特性

如果提示需要选择，选择 "Protein"；如果没有特别提示或者没有 if no selection is specifically stated or does not follow logically from the text.   
  
To get rid off the *noise*, please use the 'Running Average' method in 'Data->Transformation' to smooth your graphs with xmgrace

如果确认了模拟已经收敛，就可以进行真正的分析了。对模拟数据的分析，可以分为几种类型。第一种包括对单个图形的解释，通过一些函数逐个点进行计算得到一个值或者一些值；例如RMSD和回旋半径的计算。这些特性可称为构型，依赖或瞬间特性。另外一种是可以通过一定时间范围内的平均化来分析的特征，比如相互关系或波动。本部分将进行一些通常的分析，每个都能得到直接来自于轨迹（不同时间下的坐标）的一个时间序列值。这些问题可以参考程序运行时的屏幕输出或者图形。

## 溶剂可及表面面积

一个可能感兴趣的特性是蛋白质表面可被溶剂到达的面积，一般指溶液科技表面 (SAS)或者溶剂可及表面面积 (SASA)。还可细分为亲水性SAS和疏水性SAS。除此之外，SAS可以和一些经验参数一起使用，得到一个溶剂化自由能的估计。 所有这四个参数都可用g\_sas程序完成。本程序也允许计算每个残基或原子一段时间内的平均SAS。输入下面的命令，设置需要计算SAS的基团和输出基团，查看输出文件。

g\_sas -f traj\_nojump.xtc -s topol.tpr -o solvent-accessible-surface.xvg -oa atomic-sas.xvg -or residue-sas.xvg

==Q== Focusing on the loop1 (residues 53-62), which residues are the most accessible to the solvent?

## 氢键

另一可能能提供很多信息的性质是氢键，内部氢键（蛋白质-蛋白质）或蛋白质与其周围的溶剂之间的氢键都是这样。氢键的存在与否可以通过氢键受体和供体对的距离和键角来推断。为了计算氢键，使用如下命令（并用Xmgrace查看输出文件）：

echo 1 1 | g\_hbond -f traj\_nojump.xtc -s topol.tpr -num hydrogen-bonds-intra-protein.xvg

echo 1 12 | g\_hbond -f traj\_nojump.xtc -s topol.tpr -num hydrogen-bonds-protein-water.xvg

==Q== Discuss the relation between the number of hydrogen bonds for both cases and the fluctuations in each.

特定氢键可以用包含相应原子数量的索引文件来得出。从分析1得出的RMSF及b-factors显示loops 2 和 3 (helix 2附近)值比较高。实验数据也显示，loop 1可能在UbcH6 and UbcH8的多个行为中起了一定作用。看看这些loop所包含的氢键连接。第一个命令会在菜单中弹出3个新的条目去选择基团，每个loop一个。第二个命令是一个通用命令g\_hbond，需要一个索引文件。你可能想看看每个loop里的氢键、loop1和loop2之间的氢键；例如，某个特定的loop和蛋白质其他部分的氢键（为此可能需要修改索引文件，不明白就找人问问）或者和水形成的氢键，等...

echo "r 53-62\nname 16 loop1\nr 87-94\nname 17 loop2\nr 110-117\nname 18 loop3\nq" | make\_ndx -f confout.gro -o my\_index.ndx

g\_hbond -f traj\_nojump.xtc -s topol.tpr -num hydrogen-bonds-loop.xvg -n my\_index.ndx

==Q== What can you say about the stability of the hydrogen bonds for these loop regions?   
  
==Q== On the hydrogen bonding basis, which loop is the more, respectively less, stable? Did you detect H-bonding between loops?

## 盐桥

除了氢键之外，蛋白质也常在不同的带点残基之间形成盐桥。这对蛋白质的结构有重要的稳定作用，尤其是当它们位于一个憎水环境，例如蛋白质核心。但是盐桥也能在蛋白质暴露表面看到，对于介导蛋白质识别过程往往很重要。盐桥的存在，可以用g\_saltbr来查看。当需要时，通过设置选项-sep，这个程序可以为每对相反的带电残基产生一个输出文件，这些残基位于轨迹中的某点，彼此之间在一定的隔断距离范围内（这里是0.5 nm，通过选项-t设置）。这将产生许多文件，所以分析时最好建立一个单独的文件夹。执行以下命令：

mkdir saltbridge

cd saltbridge

g\_saltbr -f ../traj\_nojump.xtc -s ../topol.tpr -t 0.5 -sep

为了更清晰点，删除有关钠离子和氯离子的文件：

rm -f \*CL-\* \*NA+\* \#\*

看看以下残基之间的相互作用：

* GLU-56 (resp. ASP-56) 和 LYSH-60
* LYSH-60 和 ASP-88

==Q== What can you say about the interactions? The residues K60 and D88 are highly conserved and the single mutation of residue D56 toward E56 in UbcH8 modify its interaction profile. What could be the cause of that? Compare your graphs with the ones generated by your co-group colleagues.

现在回到你的工作目录：

cd ../

## 二级结构

蛋白质结构最常需要判断的是二级结构，如α-螺旋和β-折叠的确定，一个可以用来做出判断的命令是dssp。此程序不在gromacs发行，但是可以在[CMBI（Radboud University](http://swift.cmbi.kun.nl/gv/dssp/)）得到。Gromacs确实为dssp提供了一个交互界面，允许对每帧轨迹计算二级结构：

cd ~/Desktop

wget ftp://ftp.cmbi.ru.nl/pub/molbio/software/dsspcmbi.tar.gz

tar -zxvf dsspcmbi.tar.gz

setenv DSSP "/home/student/desktop/dssp"

do\_dssp -f traj\_nojump.xtc -s topol.tpr -o secondary-structure.xpm -sc secondary-structure.xvg -dt 10

secondary-structure.xvg文件包含一个时间序列，列出了每帧中与各种二级结构有关的残基数量。更详细的信息在.xpm文件中，它对每个残基一段时间的二级结构用颜色表示出来。.xpm文件能用类似Gimp等软件进行观察，但是一些有用的元数据可以用gromacs工具xpm2ps加入，结果可以用gview或类似程序观看。For locals, do note the similarity between the plot and the front of the new building of the Hogeschool Utrecht.

xpm2ps -f secondary-structure.xpm -o secondary-structure.eps

gv secondary-structure.eps

==Q== Discuss some of the changes in the secondary structure, if any.

==Q== Compare the stability of the secondary structures in the different proteins.

## 拉氏图

### Phi, psi and omega angles, and the ramachandran plot

蛋白质骨架的phi-和psi-扭转角是两个对于深入了解蛋白质结构特性非常有用的参数。对phi-相对psi-角度作成的图称为拉氏图；拉氏图的某些特定区域反映了蛋白质二级结构单元或氨基酸的特性，（而这些区域）之外的区域被认为是不可能出现的（不可到达）。phi/psi角随时间的变化可以反映结构的变化。这些叫可以用g\_rama程序计算，虽然方式有点粗略，因为它把所有的东西一股脑儿放在一个文件里。为了观察单个残基，可以用linux程序 'grep'从图中选出来。

g\_rama -f traj\_nojump.xtc -s topol.tpr -o ramachandran.xvg

此文件包含所有残基所有的phi/psi角。为了提取某个残基（例如LYS-60）的角度，输入：

grep LYSH-60 ramachandran.xvg | awk '{print $1 " " $2'} > rama-LYSH-60.xvg

用这个提取后面的拉氏图，并用xmgrace观察。PDB结构中，残基82-86被归为螺旋。看看你的模拟结果，现在还是不是这样？

==Q== What can you say about the conformation of these residues, based on the ramachandran plots (see the plot given above)?

## 动力学和时间平均特性的分析

## 又是RMSD

均方根偏差（RMSD）在模拟的收敛性中已经算出来了，但是它也可用于进一步的分析。RMSD是两个结构之间的比较值。如果我们对轨迹文件中的结构各个合并，计算RMSD值，然后就可以看看是否有属于同一类型或有共同特征的结构组。这些结构中，属于同一组的有较低的RMSD值， 而与其它结构相比则有更高的RMSD值。用矩阵形式表示RMSD值，这也可以用来确定过渡变化。

为了建立RMSD矩阵，调用g\_rms命令和两条轨迹。本教程中，我们采用同一条轨迹。我们在轨迹文件中每隔一帧提取一次，生成了1001x1001矩阵。选择组合"Mainchain+Cb".

==Q== What is interesting by choosing the group "Mainchain+Cb" for this analysis? Think about the different proteins used for this practical.

g\_rms -s topol.tpr -f traj\_nojump.xtc -f2 traj\_nojump.xtc -m rmsd-matrix.xpm -dt 10

得到的矩阵是灰阶图。为了看得更清楚，我们可以采用彩色梯度图：

xpm2ps -f rmsd-matrix.xpm -o rmsd-matrix.eps -rainbow blue

gv rmsd-matrix.eps

==Q== How many transitions do you see?

把你的和你同组同学的轨迹相比较，也是非常有趣的。请重复相同过程，但采用-f和 -f2选项，对不同的轨迹运行g\_rms。你应当和你的组员共享你们的工作结果，那么将会有6种组合可能性：

* UbcH6\_wt vs UbcH8\_wt
* UbcH6\_wt vs UbcH6\_E56D
* UbcH6\_wt vs UbcH8\_D56E
* UbcH8\_wt vs UbcH6\_E56D
* UbcH8\_wt vs UbcH8\_D56E
* UbcH6\_E56D vs UbcH8\_D56E

==Q== What can you conclude from this analysis? Could you expect such a result, justify?

## 簇分析

在结构间距离的基础上，这些RMSD也可以归并为一些反映构象可及性和每个RMSD相对权重的簇。用一个簇算法可以完成这个任务，g\_cluster包含多个簇算法。该程序产生多个输出文件。查看帮助文件了解它们的含义，然后运行这个程序。注意，我们已经算好了RMSD矩阵，可以把它作为g\_cluster的输入。

g\_cluster -h

echo 6 6 | g\_cluster -s topol.tpr -f traj\_nojump.xtc -dm rmsd-matrix.xpm -dist rmsd-distribution.xvg -o clusters.xpm -sz cluster-sizes.xvg -tr cluster-transitions.xpm -ntr cluster-transitions.xvg -clid cluster-id-over-time.xvg -cl clusters.pdb -cutoff 0.1 -method gromos -dt 10

==Q== How many clusters were found and what were the sizes of the largest two? ( T )

改变cutoff值，得到适当数量的簇 (10 < Nclusters < 100)。

用pymol打开文件clusters.pdb，比较前两个簇的结构。

disable all

split\_states clusters

delete clusters

/for i in range(3,100): cmd.delete( "clusters\_%04d" %i )

dss

show cartoon

util.cbam clusters\_0002

align clusters\_0001 and ss h, clusters\_0002 and ss h

==Q== Are there notable differences between the two structures?

## 距离RMSD

采用RMSD 比较结构的一个不足之处是，它包含了最小平方拟合，这会影响结果。一个蛋白质的结构也可以用一套原子间距离来表示。这也可以用来获得一个比较量，这个比较量不依赖于拟合，这就是距 RMSD (dRMSD)。可采用g\_rmsdist程序：

g\_rmsdist -s topol.tpr -f traj\_nojump.xtc -o distance-rmsd.xvg

==Q== At what time and value does the dRMSD converge and how does this graph compare to the standard RMSD?

原子间距离分析，模拟得到的NOE's

上面用到的g\_rmsdist程序也可用于进一步的举例分析。特别地，为了了解结构及其稳定性的，通过查看原子间的平均距离及其这些距离波动可能会有用处。对蛋白质中的每对原子的平均距离和距离波动进行计算获得矩阵，用上面的步骤给影像重新上色，并显示.png文件的结果 (rmsmean.png和rmsdist.png).

g\_rmsdist -s topol.tpr -f traj.xtc -rms rmsdist.xpm -mean rmsmean.xpm -dt 10

==Q== Briefly explain the images: rmsmean in terms of structure and rmsdist in terms of flexibility/stability. Recall the information from earlier analysis and viewing the structure

从实验角度讲，这些距离同样重要。原子间距离的边界可以由NMR实验推断——利用核的[Overhauser效应 (NOE)](http://en.wikipedia.org/wiki/Nuclear_Overhauser_effect)——这也是NMR结构计算间的主要推动力。如果蛋白质模型正确，这个应该可以预期得到的NOE信号，可以通过MD模拟来计算。这些信号与距离密切相关，特别是r-3 和 r-6 权重的距离。这些信号也可以用g\_rmsdist来计算。

g\_rmsdist -s topol.tpr -f traj.xtc -nmr3 nmr3.xpm -nmr6 nmr6.xpm -noe noe.dat -dt 10

给nmr3.cpm 和 nmr6.xpm重新上色，看看这些矩阵 ，也看看文件noe.dat。

==Q== What are the smallest 1/r^3 and 1/r^6 averaged distances in the simulation? ( T )

## 松弛和序参量

向量的松弛的计算与其自身相关。对蛋白质，通常包括碳骨架N-H 或侧链C-H向量。这种自相关给出了一个向量能保持其方向多长时间的量度，因而为表征可变性和稳定性提供了指示。序参量是自相关的长程限制。如果一个分子能够自由旋转，序参量将不可避免地减小到零；但在分子框架内（内部参考框架），序参量常有一个明确的值，这个值表明总体稳定性。通过固定（fitting mistake for fixing?）蛋白质，这个参考框架可以固定下来，因此序参量就确定了。

N-H序参量可以用g\_chi计算。这个程序可以写出一个.pdb文件，该文件把序参量加入了b-factor场，使观看更容易。该程序也计算[J-couplings](http://en.wikipedia.org/wiki/J-coupling)，这个参数可以和NMR 的结果相比较，或者用于指导NMR实验。

g\_chi -s topol.tpr -f traj.xtc -o order-parameters.xvg -p order-parameters.pdb -jc Jcoupling.xvg

看看.xvg文件中的序参量，并用Pymol 查看.pdb文件，根据在b-factor场中的值，给残基上色。

==Q== Write down the start and end residues, and the average value for the two regions having highest order parameters. ( T )

==Q== How do the order parameters compare to the fluctuations (RMSF)?

## 构象主成分分析

See Leach Chapter 9.14 for more information on the following section.

一个常用的，但常常对之理解不深的方法是轨迹的主成分分析法。这种方法——有时候也指“本征动力学（ED）”——目的在于鉴别大量原子的整体运动并因此揭示隐藏在原子波动后面的结构信息。在MD中，粒子波动确实互相关联，因为粒子间的彼此作用。关联的程度有大有小，但需注意的是，那些通过键直接相连或者彼此位置接近的粒子会产生谐动。粒子之间的运动相关性产生系统总体波动中的结构，对于大分子来说，这种结构常常与其功能和生物物理学特性直接有关。因此，原子波动结构的研究可以对了解这些大分子的行为提供有价值的信息。然而，它确实需要相当程度的线性代数和多变量统计方面的知识，来解释结果并鉴别该方法的缺陷。特别是，PCA的目标是用新的变量来描述原始数据，这些新变量与原始变量线性关联。这也是PCA最主要的问题：它仅仅依据原子运动间的线性关系来解释问题。

PCA的第一步是构建协方差矩阵，它将捕捉每对原子间的原子运动相关度。协方差矩阵从定义上说，是个对称矩阵。这个矩阵随后将用于分析，产生一个特征向量矩阵和特征向量的对角线矩阵。每个特征向量描述了粒子的集体运动，这里向量的值表示相应原子参与了多少运动。给出的相关的特征值等于波动的总和，这些波动以集体运动中每个原子来描述，因此是与特征向量有关的全部运动的一个度量值。通常，系统中的大多数 (>90%) 运动是用少于10个特征向量或主成分来描述的。

协方差分析产生相当多的文件，因此最好在一个新的子文件夹中运行：

mkdir COVAR

cd COVAR

协方差矩阵的构建和对角线化可用g\_covar程序。输入下述命令进行分析。注意，这可能要花费一些时间。

g\_covar -s ../topol.tpr -f ../traj.xtc -o eigenvalues.xvg -v eigenvectors.trr -xpma covar.xpm

==Q== What are the dimensions of the covariance matrix and what is the sum of the eigenvalues? ( T )

特征值之和是系统总运动性的一个度量。它可用于比较不同情况下蛋白质的可变性，但是对不同大小的蛋白质进行比较时，就难以得到有意义的解释。

现在，看看协方差矩阵自身，它在covar.xpm文件里。

xview covar.xpm

矩阵显示了原子间的协方差。红色表示两个原子一起移动，而蓝色表示它们彼此向相反的方向运动。对角线化相应于早前得到的RMSF图，红色的深度表示波动的振幅大小。

==Q== Look at the two most moving parts, excluding the termini. How do they move with respect to each other and to the rest of the protein?

从协方差矩阵能看到一组组相关或反相关运动的原子。这允许往原子组的集体运动中重新写入总运动。我们提到过，特征值储存在eigenvalues.xvg文件中，通过相应特征值表示出总波动。

==Q== Look at the file eigenvalues.xvg with a text editor. Calculate the percentage and [cumulative](http://www.merriam-webster.com/dictionary/cumulative) percentage of the motion explained for the first five eigenvectors. ( T )

Eigenvalues and cumulative sum of 
eigenvectors

典型地，最初五个特征值将捕获主要运动，这相当于 >80% 的总运动。如果解释的总波动较低，就提示没有明确的集体运动。

为了了解特征值的实际意义，可用g\_anaeig工具作进一步的分析。为了更近地看看前两个特征值，输入以下命令：

g\_anaeig -s ../topol.tpr -f ../traj.xtc -v eigenvectors.trr -eig eigenvalues.xvg -proj proj-ev1.xvg -extr ev1.pdb -rmsf rmsf-ev1.xvg -first 1 -last 1

g\_anaeig -s ../topol.tpr -f ../traj.xtc -v eigenvectors.trr -eig eigenvalues.xvg -proj proj-ev2.xvg -extr ev2.pdb -rmsf rmsf-ev2.xvg -first 2 -last 2

特征值对应于运动方向。选项-extr沿着选定的特征值从轨迹中提取极端结构。把这些结构导入PyMOL看看：

pymol ev?.pdb

把pdf文件中的模型分开，删除原始结构。

split\_states ev1

split\_states ev2

delete ev1 or ev2

给模型上色。特征值1中的极端结构显示为蓝-绿色而特征值二为黄-红色。

spectrum count

dss

hide everything

show cartoon

用PyMOLs的'align'命令，能画出表示两种构象差异的小条。

align ev1\_0001 and (n. c,n,ca),ev1\_0002 and (n. c,n,ca),object=diff1

align ev2\_0001 and (n. c,n,ca),ev2\_0002 and (n. c,n,ca),object=diff2

==Q== What is the largest difference between the extreme structures for eigenvector 1? And for eigenvector 2?

为了理解特征值的意义，想象一下欧洲城市间卖货人的移动所发生的事情。这种移动可用地球坐标体系解释，这种情况下我们需要对每个位置采用三个坐标。虽然这是对的，但如果你只想解释卖货人的移动，就不是最佳的。因为理论上，任何坐标体系都一样好，我们可以定义一个新的（坐标体系）来解释卖货人的移动。实际上，因为地球表面可以用二维空间代表，我们只需要两个坐标而无须三个。 直观看来，有人会拿南北极和东西轴说事，但也可以从移动中推断出轴。比方说卖货人在伦敦-雅典轴上走的最多。这个轴作为第一个特征值；第二个特征值与第一个正交。用这种方式，卖货人在欧洲的每个位置就可以用在这两个特征值上的投影唯一地确定下来。对它们的投影作图，就可以显示出旅行路线。沿着第一个坐标的极端投影对应于Reykjavik和莫斯科，即使它们实际上不在这个轴上。

蛋白质构象也和上面所述的一模一样。你看到的极端投影并不一定对应于物理结构，但是它们允许看到沿轴的运动和总的运动程度。为了解蛋白质沿构象空间的移动，我们可以画出特征值2对特征值1的投影图。为此，从两个.xvg文件中提取投影数据并且合并到文件ev1-vs-ev2.dat中。注意'>'表示输出由屏幕重定向到一个文件中，所以你看不到任何屏幕输出。

grep -v "^[#@]" proj-ev1.xvg | awk '{print $2}' > proj-ev11.dat

grep -v "^[#@]" proj-ev2.xvg | awk '{print $2}' > proj-ev12.dat

paste proj-ev11.dat proj-ev12.dat > ev1-vs-ev2.dat

为了解释模拟过程，我们也提取了最后7.5 ns（最后1500点）和5.0 ns（最后1000点）的投影。然后把它们导入xmgrace。

tail -1500 ev1-vs-ev2.dat > last-7.5ns.dat

tail -1000 ev1-vs-ev2.dat > last-5.0ns.dat

xmgrace ev1-vs-ev2.dat last-7.5ns.dat last-5.0ns.dat

==Q== What is the shape of the projections? Are these mutually independent (oval distribution)?

==Q== Would the same eigenvectors (axes) be obtained if the analysis were performed on the last 7.5 ns? And on the last 5.0 ns?

## 结语

Write a concluding paragraph, comparing the results obtained for the different proteins. Also reflect on the overall stability and the probability that the structure deposited in the PDB properly reflects the solution structure. In other words, does the structure stay close to the starting structure or does it drift away and how much? Can you suggest a mechanism that would explain the differences in UbcH6 and UbcH8 interaction profiles and why a single conserved mutation can restore the rich interaction profile of UbcH6 when starting from UbcH8? What are the limits of the simulations to explain such a complex biological process?