

# VMD 教程

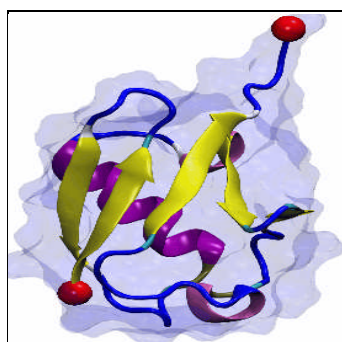
简介:

这个教程为新用户介绍了 VMD 的用法。老用户也可以用本教程进一步熟悉程序的应用，以更好地利用 VMD。本教程是针对 VMD 1.8.3 设计的，需要约 3 个小时来完成。

本教程新增的内容可用三个独立的单元讲解。第一个单元主要内容是分子图形表现方法基础，还会介绍制作形象逼真的图像要了解的知识。另外的两个单元是针对高级用户，介绍了 VMD 的脚本。尽管非技术性用户可以略去脚本的阅读，但是我们鼓励每个人都去试一试着读一下，因为它会提供一些有力而易用的工具，这些工具是简单的图形用户界面所无法提供的。

本教程以一种有趣的小蛋白质泛素的研究为例来说明 VMD 的应用。在本文中，一些资料是在小框中出现的。这些小框中包括教程的补充内容，例如泛素扮演的生物学角色，使用 VMD 的一些提示和捷径等等。

如果你有对本教程的评论和问题，请发邮件至 [tutorial-l@ks.uiuc.edu](mailto:tutorial-l@ks.uiuc.edu)。邮件列表可以在 <http://www.ks.uiuc.edu/Training/Tutorials/mailling list/tutorial-l/> 中找到。



泛素 本教程会用VMD来显示泛素。泛素是一个由76个氨基酸组成的小蛋白质，在所有的真核生物中普遍存在。在所有真核生物蛋白质中，泛素是最为保守的蛋白质之一（在昆虫，鱼，牛和人中，前74个氨基酸是完全一样的）。它已被证明存在于细胞核、细胞质和细胞表面。它首要的功能是介导蛋白质降解，在降解过程中，作为细胞内蛋白水解酶识别的标志。

需要的程序:

以下是本教程中需要的程序

VMD: 可以从 <http://www.ks.uiuc.edu/Research/vmd/> 下载（在所有平台上均可使用）。

绘图程序: 要观看从 VMD 输出的图像，需要专门的程序。VMD 有一个内置的绘图程序，也可以应用外部程序。应用什么程序是由你的操作系统决定的。例如:

- Unix/Linux: xmgrace, <http://plasma-gate.weizmann.ac.il/Grace/>
- Windows: Excel, <http://office.microsoft.com/en-us/FX010858001033.aspx> (需要购买)
- Mac/Multiple Platforms: Mathematica, <http://www.wolfram.com/> (需要购买); gnuplot, <http://www.gnuplot.info/> (免费下载)

现在开始学习 VMD

你可以在 VMD-tutorial-files 目录里找到本教程的文件。如图 1 所示的 VMD-tutorial-files 的文件和目录

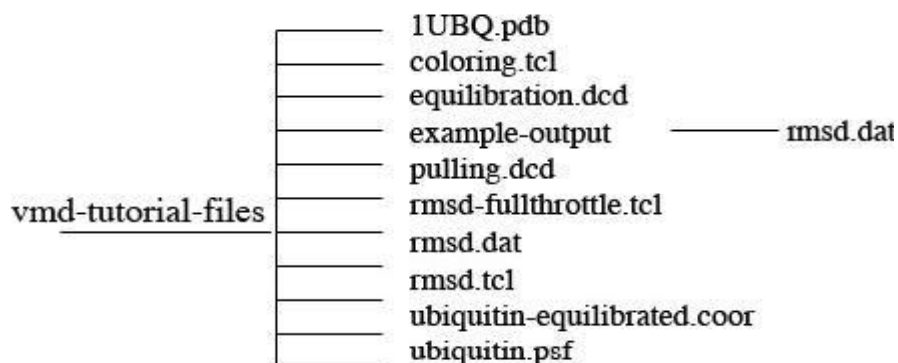


图 1：VMD-tutorial-files 的目录结构

运行VMD，可以在Unix 终端窗口中键入vmd，在Mac OS X的应用文件夹中双击VMD应用程序图标或者在windows中单击开始——程序——VMD。

## 1 VMD基础

在本单元中你会通过构建一个泛素的较美观的图形来熟悉VMD的基本命令。另外，你可以学习怎样用VMD来寻找蛋白质结构上的有趣的特点。

### 1. 1 导入分子

第一个步骤是导入分子。在教程中提供了一个pdb文件1UBQ.pdb，文件中包含了泛素的原子坐标。

1在VMD主窗口的菜单栏中选择File —— New Molecule，如图2（a），屏幕上会显示另外一个窗口，即Molecule File Browser（b）。

2 应用Browse.（c）按钮在 vmd-tutorial-files中找到文件 1UBQ.pdb，注意到当你选择这个文件的时候，就会回到Molecule File Browser窗口。为了精确地导入你要导入的文件一定不要忘了按下Load（d）按钮。

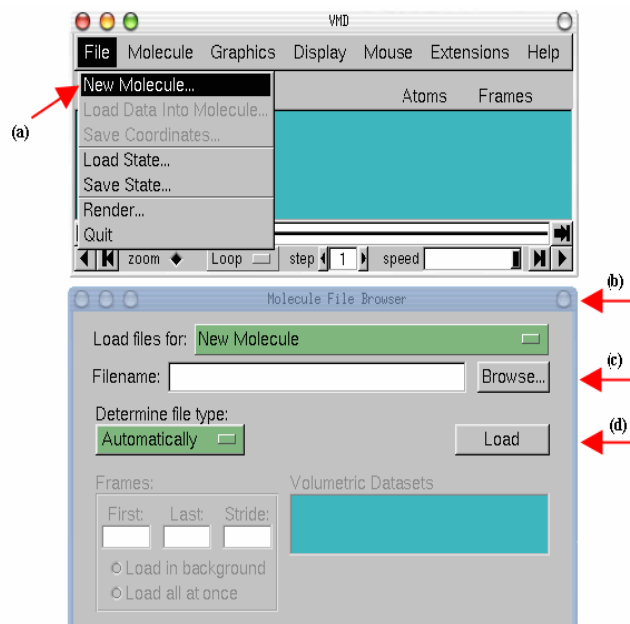
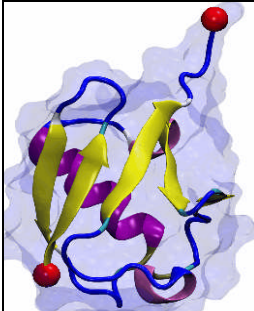


图 2 导入分子

现在，泛素在你的OpenGL Display窗口中显示出来。你可以随时选择Molecule File Browser窗口。



Webpdb. 如果网络连接可用, VMD可以从蛋白质数据库中下载pdb文件。只要在Molecule File Browse窗口的File Name中键入四个字母的蛋白质号, 再点一下load就可以了。VMD会自动下载该pdb文件。



坐标文件。文件1UBQ.pdb与泛素的X射线衍射1.8埃分辨率侧的结果相对应 (Senadhi Vijay-Kumar, Charles E. Bugg and William J. Cook, J. Mol. Biol.

(1987) 194, 531)。注意蛋白质被58个水分子包围, 结果中不包含氢原子。

## 1. 2 显示蛋白质

为了观察蛋白质的三维结构, 我们要用到多种鼠标模式。

1 在OpenGL Display中, 按下鼠标左键同时移动鼠标。进一步观察有什么现象。这是鼠标的旋转模式, 通过这种模式, 你可以让这个分子绕一个与屏幕平行的轴旋转。图3 (a)

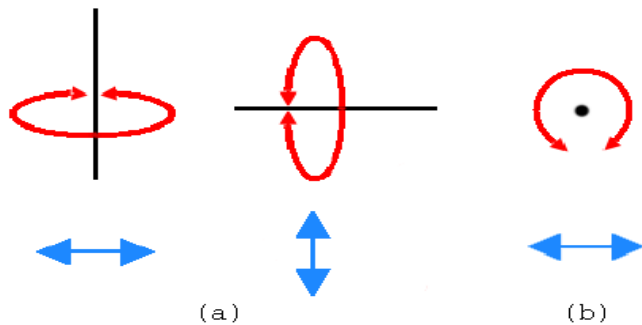


图3 旋转模式

2 如果你按下鼠标右键, 重复上一步骤, 分子会绕一个与屏幕垂直的轴旋转 (b) (对于Mac用户来说, 右键产生的效果与在按下mouse菜单选项后点击命令按钮是一样的)。

3 在VMD主窗口中, 看一下Mouse菜单 (图4), 这里, 你可以把鼠标模式从Rotation 更换到Translation或者Scale modes。

4 Translation模式允许你按住鼠标左键, 在屏幕上移动分子。在Translation模式下, 你可以通过按下鼠标中键来改变剪切板。

5 在Scale模式下, 你可以按住左键水平移动鼠标来缩小或放大分子。

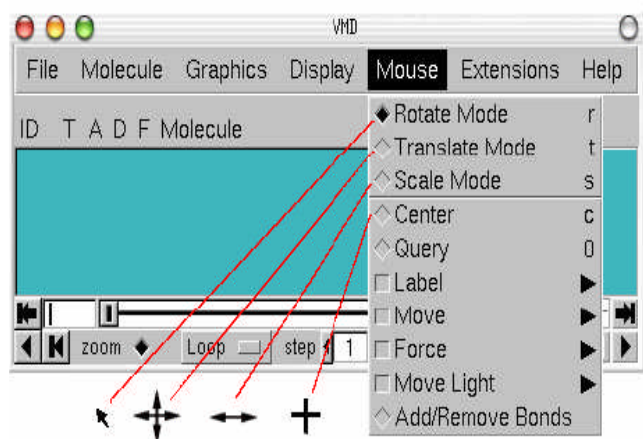


图4 鼠标模式

需要注意的是: 鼠标运动不会改变分子中的原子坐标。



鼠标模式。注意每一种鼠标模式都有独特的指针形状，也有独特的快捷键((r: Rotate, t: Translate, s: Scale)，可以代替菜单使用。(当用快捷键的时候，要保证OpenGL Display窗口是活动的)。在VMD用户手册中可以获得更多的信息。

Mouse —— Center菜单项也很有用，它允许你确定分子绕之旋转的支点。

- 6 选择Center菜单项，在蛋白质一端选择一个原子，这时指针会显示成一个十字。
- 7 现在，按下r，用鼠标旋转分子，看一看你的分子是怎么绕着你选择的支点运动的。
- 8 选择Display —— Reset View菜单项(=快捷键)，回到默认界面。

### 1.3 学习应用不同的绘图模式

VMD可以用很多种绘图模式来显示你的分子。这里，我们要进一步学习那些可以帮助你确定蛋白质中不同结构的绘图模式。

- 1 选择Graphics—— Representations菜单项，一个叫做Graphical Representations的窗口会出现，见图5 (a)中黄色高亮。你可以看到目前显示

- 2 在Draw Style标签中 (b) 我们可以改变所表示的style (d) 和color (c)。在这一部分我们重点来看drawing模式。

- 3 每一种绘图方法都有自己的参数控制。例如，改变线条的稠密度可以用Graphical Representations窗口右侧底部的控制按钮 (e)。

- 4 现在，在Drawing Method中选择VDW(van der Waals)，每一个原子现在都表示为球形。用这种方式你可以更容易地看出蛋白质的体积分布是怎样的。

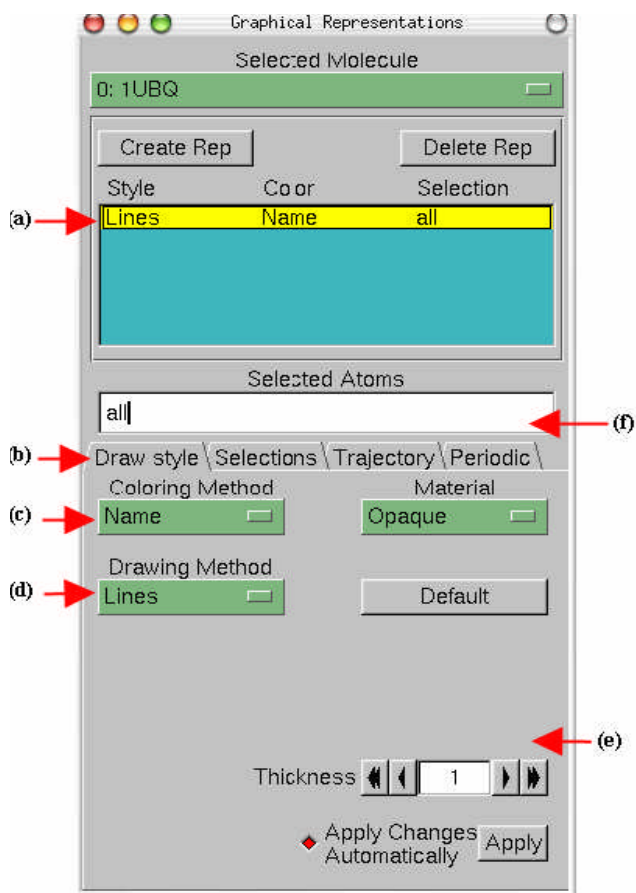


图5 Graphical Representations窗口

- 5 要观察蛋白质内部的原子排布，用窗口右侧底部的控制按钮改变Sphere Scale到 0.5，

Sphere Resolution 到13。注意分辨率越高，分子的显示速度越慢。

6 注意Coloring Method —— Name菜单项，每一个原子都有其自己的颜色，比如，O是红色的，N是蓝色的，C是青色的，S是黄色的。

7 按下Default键，这个操作允许你回到默认的绘图方式中。



更多显示方法。还有有趣的显示方法是 CPK 和 Licorice。在 CPK 中，就像以前化学中的球棒模型，每个原子都用球形表示，每个键都用圆柱棒表示（球和圆柱形棒的半径和分辨率都可以独立地调节）。Licorice 绘图方法（广泛使用）也用球形表示原子，用圆柱形棒表示键，但是球的半径不能被独立调节。

前面的显示方式可以让你看到蛋白质大分子的细节。但是，更多的普遍结构属性可以用抽象的绘图方式来观看。

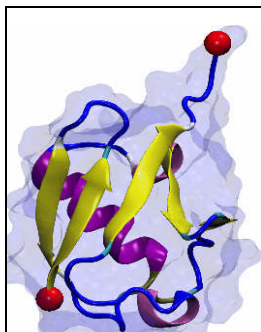
8 在Drawing Method下选择Tube style，观察蛋白骨架。Radius设为0.8。

9 在tube模式下观察你的蛋白质，你可以分辨出它有多少 $\alpha$ 螺旋、 $\beta$ 折叠和无规则卷曲吗？

我们要了解的最后一个绘图模式是NewCartoon。它可以给出一个以二级结构为基础的简化的蛋白质图像。 $\alpha$ 螺旋以卷曲的条带状表示， $\beta$ 折叠以固形箭头表示，所有其它的结构以管状表示。这可能是观察蛋白质分子总体构造的最普遍的方法。

10 选择Drawing Method ——NewCartoon.

11 现在确定蛋白质分子中有多少 $\alpha$ 螺旋、 $\beta$ 折叠和无规则卷曲。



泛素的结构。泛素有一个三圈半的 $\alpha$ 螺旋（残基23到34，其中三个是疏水的），一个 $3_{10}$ -螺旋（残基56到59），还有5个 $\beta$ 折叠（残基1到7，10到17，40到45，48到50，64到72），还有7个反向转角。VMD用STRIDE程序，以一种探索性的法则计算出二级结构

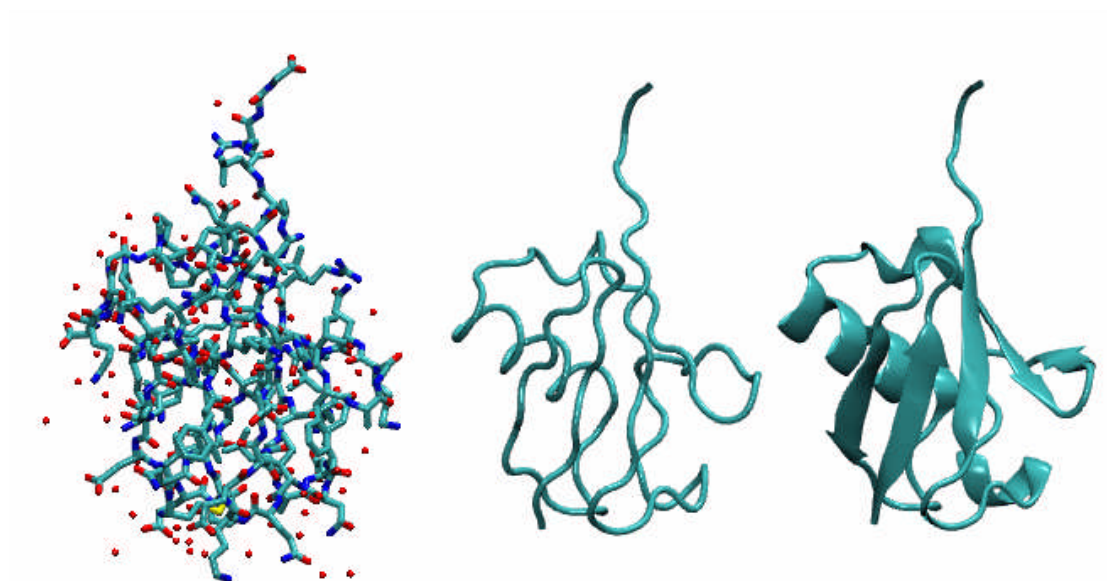


图6 泛素Licorice, Tube and NewCartoon显示方法

#### 1. 4 学习不同的着色方法

1 现在，让我们来改变所显示图像的颜色。选择Coloring Method——ResType图5 (c)，这可以区别非极性基团（白色），碱性基团（蓝色）、酸性基团（红色）和极性基团（绿色）。

2 选择Coloring Method —— Structure (c)，确定NewCartoon表示的图形与二级结构相一致。

#### 1. 5 学习不同选择

让我们来看看分子中不同的独立的部分。

1 如图5 (f)，在Graphical Representations窗口的Selected Atoms文本输入框中删去“all”，输入helix，然后按下apply按钮或者按下键盘上的Enter或者Return键（每当在文本框中输入后都可执行同样的操作），VMD会显示出分子中的 $\alpha$ 螺旋结构。

2 在Graphical Representations窗口中选择Selections标签，如图7 (a)。在Singlewords (b)这一部分中你可以发现可以输入的选项表列。例如，要显示 $\beta$ 折叠而不是 $\alpha$ 螺旋，就可以在Selected Atoms的文本输入框中输入合适的词。



布尔操作组合也可以用于选择时的文本输入。

3 为了看分子除了 $\alpha$ 螺旋和 $\beta$ 折叠的部分，可以在Selected Atoms中输入：(not helix)and(not betasheet)

:

4 Selections 标签 (b) 的Keyword (c)栏可根据蛋白质某些部分的特定值来进行选择。看一看Keyword resname (d)中的可能值。输入(resname LYS)或者(resname GLY)可以显示蛋白质中所有的赖氨酸或者甘氨酸。赖氨酸在泛素的构型中扮演重要的角色。

.

5 现在，把当前显示的Drawing Method改变到CPK模式，把Draw Style中的Coloring Method改到ResID。在屏幕中可以看到不同的赖氨酸和甘氨酸。每一种有多少个你能数得清吗？

6 在Selected Atoms的文本输入框中输入water。选择Coloring Method —— Name。你可以看到在整个系统中的58个水分子（实际上只有氧被显示出来）。

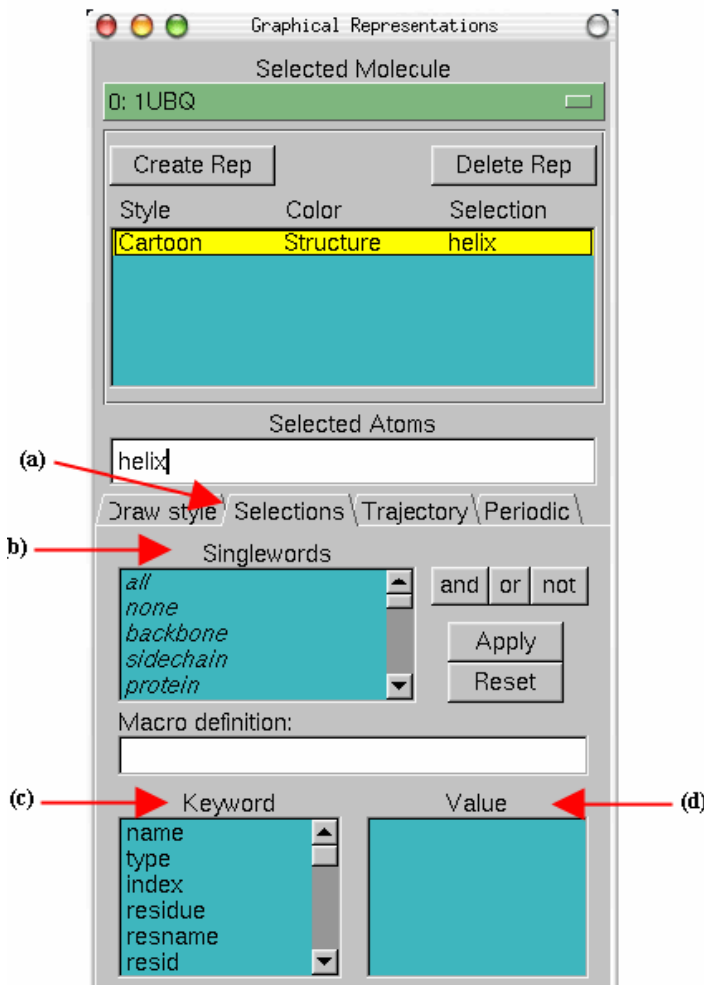


图7 Graphical Representations窗口和Selection标签

7 为了看一看哪些水分子离蛋白质分子更近一些，可以用within命令。输入waterwithin 3 of protein，这就选择了距离蛋白质3埃之内的所有的水分子。

8 最后，在Selected Atoms中键入下列内容：

:

Selection	Action
protein	Shows the Protein
resid 1	The first residues
(resid 1 76) and (not water)	The first and last residues
(resid 23 to 34) and (protein)	The _ helix

前述的选项提供了研究蛋白质或其他分子的有效工具。

### 1. 6 多重显示

如图8 (a) ，在Graphical Representations窗口中,用Create Rep按钮可以创建多重显示图像。因此，你可以让分子的不同部分显示不同的样式和颜色。

1 对当前显示,把Drawing Method 设为 NewCartoon, 把Coloring Method 设为 Structure.

2 在 Selected Atoms 中键入 protein.

3 按下Create Rep键 (a) ，现在，用 Draw Style菜单项和Selected Atoms文本输入框来更改新的图形显示，可以把 Drawing Method设为VDW, Coloring Method设为ResType, 并键入resname LYS, 使其成为当前选择。

4 重复前述步骤，产生下列两种新的显示方法：

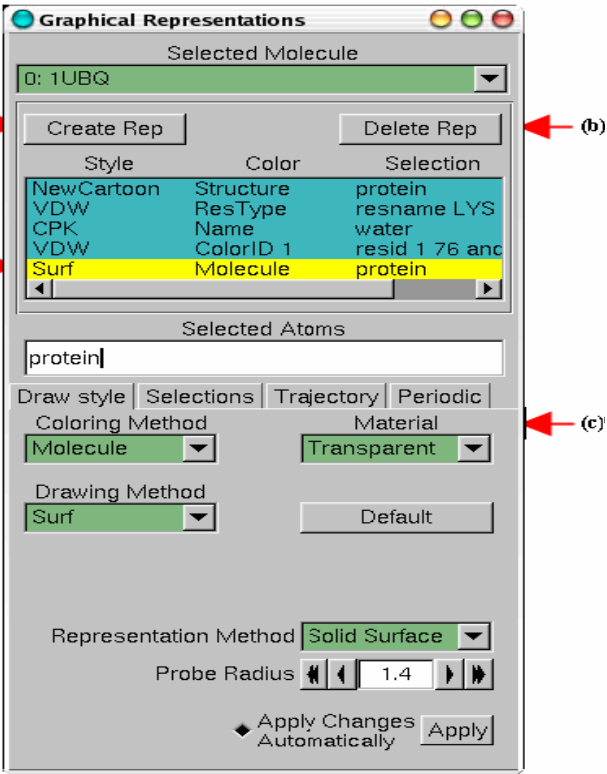


图 8 泛素的多重显示方法

Drawing Style	Coloring Method	Selection
CPK	Name	Water
VDW	ColorID1	Resid 1 76 and name CA

5 再次按下Create Rep按钮，创建最后一个图形显示法。选择Drawing Method —— Surf, Coloring Method —— Molecule, 在Selected Atoms中键入protein。在Material 部分(c)中选择 Transparent菜单项。

6 用鼠标你可以选择已创建的不同显示法，并可以独立地改变其中的任何一种。你也可以用双击鼠标或者Delete Rep按钮打开或关闭它们。关闭第二个和最后一个图形表示法。在这一部分的最后，Graphical Representations窗口的显示如图8



## 1. 7 Sequence Viewer Extension

当第一次处理一个蛋白质分子的时候,快速找出和显示不同的氨基酸是非常有用的。Sequence Viewer Extension可以让你很容易地选出和显示氨基酸残基。

1 选择Extensions ——Analysis—— Sequence Viewer菜单项, 一个包含氨基酸(图9e)和它们属性的列表(b)和(c)的窗口(图9 a)会出现在屏幕上。

2 用鼠标点击列表中不同的氨基酸残基(e), 观察它们是如何被标记为高亮的。另外, 高亮的残基还会在OpenGL Display窗口中以黄色bond drawing方式显示, 因此你可以很容易地观察它们。用鼠标右键可以解除选择。

3 用Zoom滑块控制窗口(f), 使其能够显示所有残基。这在大的蛋白质分子中比较有用。

4 按住shift键时同时按下鼠标, 就可以同时选择多个残基。看一下图中显示的残基48, 63, 11和29(e)。

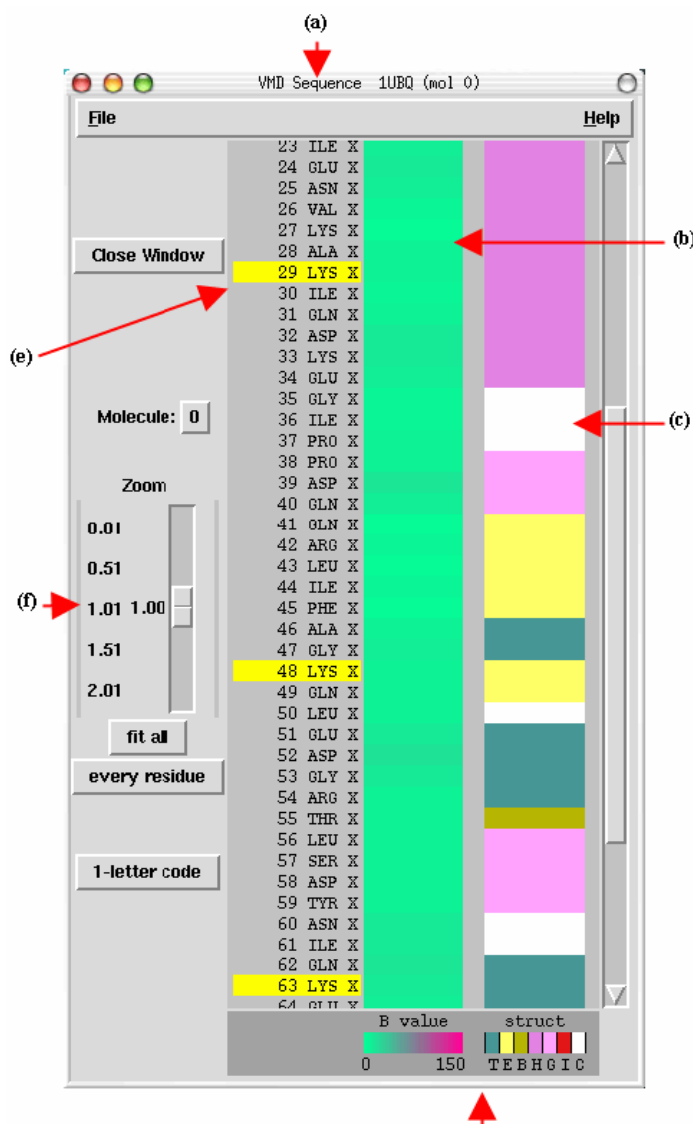
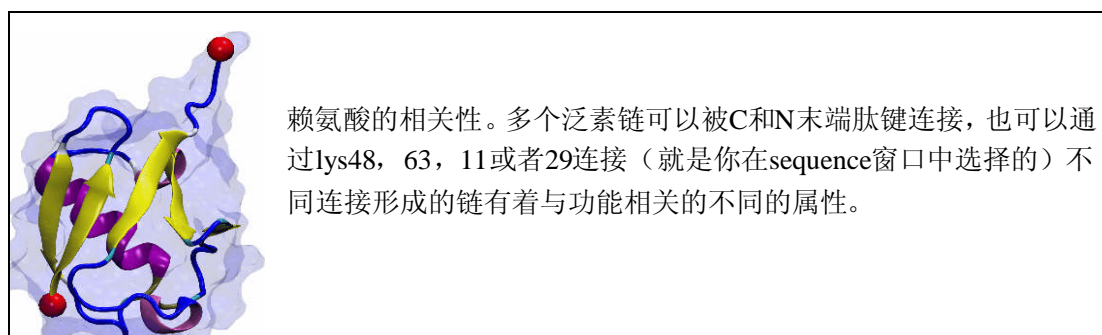


图 9 sequence 窗口

5 观察Graphical Representations窗口, 用SequenceViewer Extension, 你应该能发现一种新的显示所选残基的方法。就像你以前做过的那样, 你可以修改、隐藏、或者删除这种显示方法。



关于残基的信息用柱状彩色标记表示, 它们是从STRIDE中得到的。B-value表示的是温度因

子的变化，`struct`表示二级结构，在图中，各种颜色所代表的意义都用字母作了标注。

T	Turn
E	Extended conformation (_ sheets)
B	Isolated bridge
H	Alpha helix
G	3-10 helix
I	Pi helix
C	Coil

## 1. 8 保存结果

用VMD创建的图形可以与创建的图形显示法，VMD环境设置一起保存。这里提到的VMD环境设置包括你开启一个新的VMD环境所需要的所有信息，这就不会使你以前的工作成果丢失。

1 在VMD主窗口，选择File —— Save State菜单项，写上一个合适的文件名（例如：`myfirststate.vmd`）保存。

File —— Load State 菜单项允许你导入一个保存过VMD环境设置，就像保存时一样，虽然设置好的VMD环境允许你用VMD来处理蛋白质的图像，探索它的性质，但是你通常需要得到能用于论文和其他各种文件的图像。VMD可以润色图像，产生一个可作它用的图像文件。如下所述：

2 用你已经学过的所有关于VMD的知识，用缩放、旋转和改变分子位置的方法找到蛋白质分子的合适的视野。打开和关闭不同的显示方法，提高选择的分辨率，调整其他属性。如果你想得到一个质量很高的分子，注意每一个显示方法的分辨率。

3 注意你用Sequence Viewer extension创建的新显示法。如果需要的话，隐藏或者删除它们。

4 在润色图像之前，选择Graphics —— Colors菜单项，改变背景颜色。选择Display category, Background name和 8 white color.。这样背景就变成白色的了。

5 选择File—— Render菜单项(渲染)，一个叫做File Render Controls的窗口会在屏幕上出现。

6 可以用不同的文件包来润色分子。如果你用的操作系统是Unix或者Mac OS X，就在Render using菜单中选择TachyonInternal，否则选择Tachyon。

7 在Filename文本输入框中键入图像要保存的文件名，例如：`picture.tga` 或者`picture.dat` (默

认文件名是 `plot.tga`或者`plot.dat`，依据不同的操作平台)

8 按下**Start Rendering**按钮，包含有你的图像的文件就会创建。注意到这需要一些时间。你可以以一个图像文件名，如`picture.tga` (MacOS X 或者 Unix) 或者`picture.dat.bmp`(Windows). 结束工作，

9 关闭打开图形文件的程序，这样就可以继续使用VMD（在windows中，这条可以忽略）。

现在学完了本教程的第一单元。我们希望你学会了VMD的基本命令。你也可以做两个文件，第一个是VMD环境设置文件，让你重启一个VMD环境，在其中应用或者修改你在本单元学到的东西。第二个文件是一个关于蛋白质的图像文件，这个文件可以应用于其他文件中。

## 2 多分子处理和脚本

在本单元，你会学到如何同时处理多个分子。同时，你也会学到**Tcl**脚本的基础，用它来编辑原子数据，排列整合两个分子，并用计算出的分子特性给分子上色。

1 从一个新的VMD环境开始。如果你刚刚完成第一单元的学习，你应该退出VMD，然后重新启动。

### 2. 1 导入多个分子

首先，导入你要用到的分子。

1 用**File** —— **New Molecule**.菜单项打开**Molecule File Browser**

你需要载入泛素的X射线三维结构图，可以用第一单元中学到的操作，也可直接用命令导入。

2 确定你正在**vmd-tutorial-files**下。在VMD终端，输入`mol new 1UBQ.pdb`。.

泛素在水盒中的溶解平衡模拟已经在1ns的时间范围内实现。你的文件包含了这个平衡的最末帧的坐标。以现在可以比较泛素在平衡末状态的构像和初始态的晶体构象。



坐标文件和结构文件。为了节省空间，模拟输出文件通常只包含原子坐标，而不储存像原子类型、电荷、各部分的名字和键等不变的信息。这一部分信息另存于一个“结构”文件当中（例如一个 **PSF** 文件）。要观察一个模拟结果，你需要把同一个分子的结构和坐标文件融合。

3 现在，导入另一个分子的模拟结果。在**Molecule File Browser**窗口顶端的菜单中选择**New Molecule**。在**vmd-tutorial-files**中找到文件**ubiquitin.psf**，然后按下**Load**按钮，建立了一个有结构而没有坐标的新分子。

4 注意到,窗口最顶端的菜单上显示: 1: ubiquitin.psf. 这保证了载入的下一个文件会增加到那个分子(molecule ID 1)。因为你在用PSF文件合并数据, 所以你不需把菜单切换到New Molecule。现在, 在vmd-tutorial-files中找到文件ubiquitin-equilibrated.coor, 单击Load, 这就可以导入新的坐标, 并把新坐标与先前载入的结构信息合并。

你现在可以看到两个有层次的没有相互重叠的分子, 一个是原始的晶体泛素结构, 另外一个是一个由水盒包围的分子。,

## 2. 2 主窗口的应用

现在需要为你的分子命名, 这样就可以区别它们。

1 在Main窗口的分子列表中双击第一个分子的名字, Rename Molecule对话框弹出。键入crystal。以同样的操作为第二个分子命名为simulation。

这时, Main窗口如图10。在分子的名字之前, 有四个字母, 你可以用它们来实现一些操作。

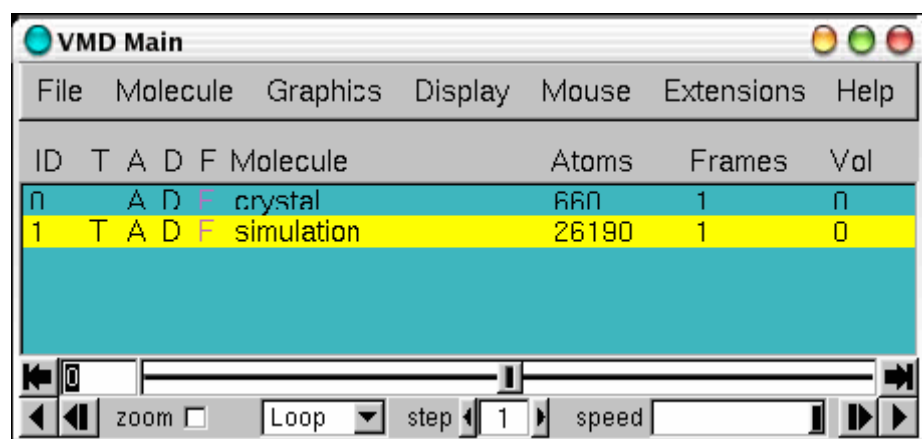


图 10 初始和最后坐标导入两个分子的主窗口显示

F的意义是“Fixed”, 意思是当你移动屏幕时, 分子不会随之移动。当F显示黑色时, 分子是固定的; 当显示灰色时, 分子可以自由移动。

2 在主窗口, 双击分子名左边的F, 当一个分子被固定时, 试着用鼠标调动屏幕。然后试一试两个分子都被固定时的情况。

3 完成操作后, 把两个分子都解固定, 选择Display——Reset View, 让其显示两个分子之间原来的相对位置。

4 然后, 双击一个分子的D, D表示的是“Display”。当D灰化的时候, 分子是隐藏的。你可以通过双击D来显示或者隐藏分子。

5 当操作完时, 确保只有晶体结构分子显示, 两个分子都未固定。

6 最后，双击晶体分子左边的T，T会在前面显示。这使它成为了顶层分子。一个分子置于顶层，它就成为脚本命令操作的目标。

## 2. 3 Tcl脚本基础和Tk控制台

VMD支持Tcl/Tk脚本语言。这一部分会提供运行一些有用功能必需掌握的脚本语言。



Tcl/Tk语言。Tcl是一种丰富的语言，除了典型的条件和循环结构语句之外，还包括许多特征和命令。Tk是Tcl的拓展，允许写程序用户与窗口和按钮接触。关于Tcl/Tk语言更多的信息在<http://www.tcl.tk/doc>上有提供。

执行Tcl命令，需要应用一种很方便的文本控制台，即Tk控制台。

1 选择Extensions——Tk Console。一个控制台窗口出现（图11）。这就可以在其中输入Tcl/Tk命令。

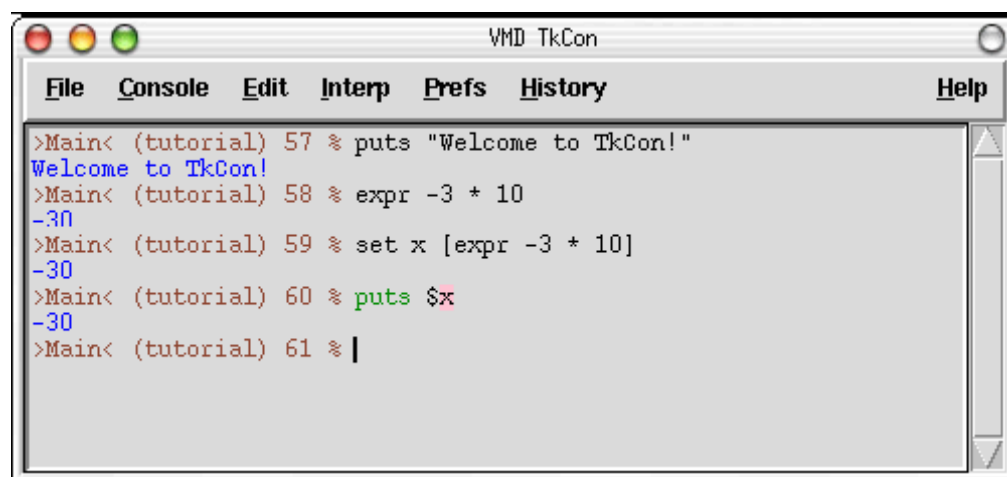


图 11 Tk 控制台

你首先接触到的是Tcl/Tk最基本的部分。这里是Tcl的设置和输出命令。

set variable value – sets the value of variable  
puts \$variable – prints out the value of variable

2 试一试以下命令：

```
set x 10
puts "the value of x is: $x"
set text "some text"
puts "the value of text is: $text."
```

\$variable指的是 variable的值。。

下面是一个数学操作命令：

`expr expression` – evaluates a mathematical expression

### 3 试一试应用expr命令

```
expr 3 - 8  
set x 10  
expr - 3 * $x
```

Tcl最重要的方面之一是可以方括号把Tcl命令嵌入其他命令。方括号括起来的表达式会自动被方括号内表达式的值替换

`[expr.]` – represents the result of the expression inside the brackets

### 4 新建一些命令，然后用方括号测试它们。例如：

```
set result [ expr -3 * $x ]  
puts $result
```

## 2. 4 atomselect命令

你可以用VMD的atomselect命令来编辑原子属性。下面的例子会告诉你怎样操作。

1 当确定晶体分子是顶层的分子后（如果不是，双击T）。用Graphics — Representations. 菜单打开Graphical Representations窗口。



PDB B-因子范围。PDB 文件中“B”的范围通常储存晶体结构的“温度因子”，在 VMD 中读入“Beta”范围。由于我们目前的重点不在此，我们可以利用这个范围来存储我们自己的数字值。VMD 有一个“Beta”上色方法，它可以根据原子的 B 因子给它们上色。当用不同的原子来替换 B 值，你就可以控制不同原子的颜色了。当你想要展示通过计算得出来的系统属性时，这种方法就很有用了。但是这种方法在 VMD 之外是不能应用的。

2 在窗口顶部的Selected Molecule下拉菜单中选择“crystal”的分子，在atom selection中键入protein。把Coloring Method 设为 Beta，把Drawing Method设为VDW。分子大部分会显示出红色和绿色小球的集合。

现在你将学到VMD中一个非常重要的Tcl命令：

`atomselect molid selection` – creates a new atom selection

对于atomselect，首先要讲的是molecule ID（主窗口的最左边），其次是文本原子选择，就



像你在第一单元所学的用来描述图形显示方法的文本原子选择。你要学习的Atomselect，返回的选择是一个命令。

3 在Tk控制台窗口键入`crystal [atomselect top "all"]`，这个选择包括了分子中所有原子，并把所选赋予变量`crystal`。我们用“top”来指代顶层分子，不用molecule ID（是个数字，不好记）。

Atomselect得出的结果是一个功能。因此，`$crystal`是一个针对“all”所指代内容执行的功能。

4 键入`$crystal num`。把num赋给所选一个原子，返回的是原子在这个选择中的序号。检查一下这个序号是不是与分子中原子的序号相同（可以从main窗口中读取）。

5 键入`$crystal set beta 0.`。这样就把所有原子的“beta”域设为0。当你这样做的时候，你会观察到屏幕上原子的颜色会突然变成一样的（因为它们此时具有相同的beta值）。

原子选择中的原子指的是原始分子中的原子。当你通过选择改变一些原子的属性（如beta值），这些变化会反映在所有其它包括这些原子的选择中。

6 现在，输入`set sel [atomselect top "hydrophobic"]`。这新建了一个包括所有疏水残基的选择。

7 让我们把所有的疏水原子的beta值设为1来标记它们。你现在应该明白如何操作了：`$sel set beta 1`。如果OpenGL Display中的颜色没有更新，单击Graphical Representations底部的Apply按钮。



原子属性的例子。你可以通过应用原子选择，包括不同部分、残基、原子名、位置（x，y和z），电荷、质量、体积、半径等等来获得或者设置许多原子的属性

8 现在要通过改变原子的物理性质来进一步说明疏水基团的分布。输入`$crystal set radius 1.0`，以使所有的原子变得小一点，更易观察。输入`$sel set radius 1.5`，让疏水基团大一点。半径的大小范围会影响图形的显示方式（如VDW, CPK）。

现在已经新建了一个图像，可以清楚地区别蛋白质的疏水部分和亲水部分。如果你完全按照教程的要求来做，你得到的蛋白质图像就会如图12所示。

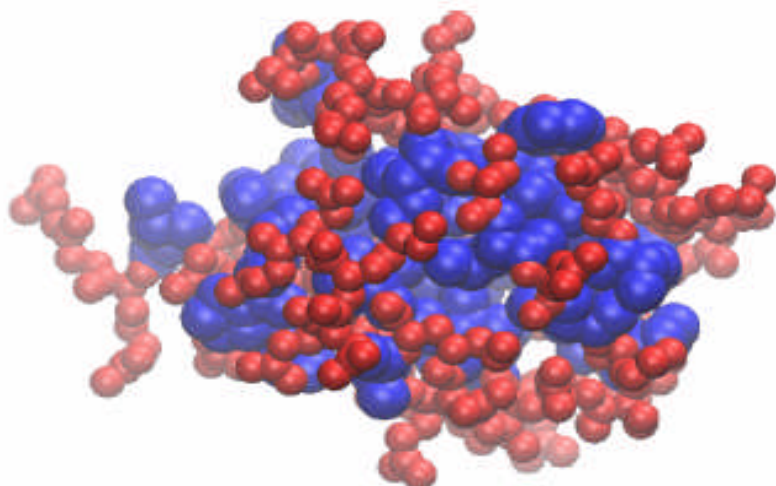
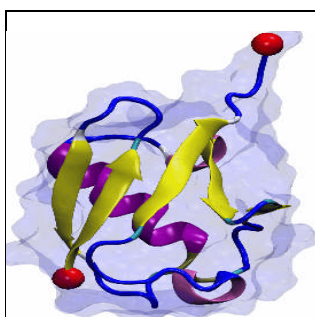


图 12 在 VMD 显示模式下的泛素分子，上色的依据是残基的疏水性



疏水残基。就像你在渲染泛素分子的过程中可能注意到的那样，疏水残基绝大部分都处于蛋白质的核心区。这在小的可溶性蛋白质中是很典型的。当蛋白质折叠的时候，疏水残基就有停留在界面的趋势。这样疏水基团汇集在一起，扮演着结构上的角色。这可以帮助蛋白质正确折叠，并提高稳定性。

原子选择不止在设置原子数据方面有用，而且可以用来获取信息。如果你想得到疏水基团，你要做的只是新建一个疏水选择，这里应该用 `get`，而不是 `set`。

#### 9 在你选择的疏水原子上应用 `get`

```
$sel get resname
```

但这里有个问题。每个残基都包含了很多原子，这样导致的结果是有很多重复输入。你能想到一个办法来克服这个问题吗？我们知道，每个氨基酸都有同样的骨架原子。如果你在每个残基中只选择一个这样的原子，每个残基就只会在你的选择中出现一次。

#### 10 让我们试试这种解决办法。每一个原子有且只有一个 $\alpha$ -碳(命名 `CA = alpha`):

```
set sel [atomselect top "hydrophobic and alpha"]
```

```
$sel get resname
```

哈，成功了！

#### 11 你也可以同时得到多种属性。试试下面的命令：

```
$sel get resid
```

```
$sel get {resname resid}
```

```
$sel get {x y z}
```

#### 12 一旦你完成了一个选择，可以很方便地用下面的命令来删除它们以节省内存开支：

```
$sel delete
```

## 2.5 Aligning Two Molecules

要正确地观察泛素在模拟中是怎么变化的，你必须首先确保两个分子在空间上尽量重合。VMD提供了很多方便的命令可以完成这个功能。现在要给两个分子上不同的颜色以区别它们。

- 1 确保两个分子都处于显示状态（可以通过双击VMD主窗口中的D来实现）。
- 2 在Graphical Representations窗口，改变晶体分子的图像显示（记得要从窗口顶部的菜单中选择它）。把Coloring Method 设为 ColorID, color 设为 0 blue, Drawing Method 设为Tube.
- 3 可以在Graphical Representations窗口最顶部的Selected Molecule菜单中选择目标分子, 在不同的分子上应用不同的显示方法。对分子simulation选择同样的显示法(ColorID and Tube), 而把颜色设为1 red.
- 4 在Tk控制台窗口中键入下面的命令，为每个分子新建一个包括蛋白质骨架中 $\alpha$ 碳的原子选择:

```
set atomsel1 [atomselect 0 "alpha"]  
set atomsel2 [atomselect 1 "alpha"]
```

这里我们选择 $\alpha$ 碳是因为它们比蛋白质松散的侧链更稳定，而且它们给出了蛋白质空间构象方面有用的信息。

即使这两个分子非常不同（晶体结构中没有氢），你所建立的选择也包括了完全相同的原子。VMD提供了一个命令measure fit可以用于找出两个选择中相同原子的最佳吻合度。

measure fit atomsel1 atomsel2 – calculates the best fit matrix
--

注意atomsel1和 atomsel2必须具有相同数目的原子。可以用\$atomsel1 num命令来检查原子的个数。

- 5 用下面的命令可以找出能够使两个选择以最佳方式配对的转换矩阵M  
set M [measure fit \$atomsel1 \$atomsel2]

- 6 通过输入命令

```
$crystal move $M
```

你可以把刚刚得到的矩阵应用于第一个分子所有部分（即你以前定义的叫做 crystal 的原子选择）

在OpenGL窗口，两个分子以 $\alpha$ 碳的位置为基础排列。注意到分子的一部分吻合得很好，但另一些摆动的尾部(loop区)看起来比较松散（它们在平衡中的运动性更强）。用了 $\alpha$ 螺旋后，再用 $\beta$ 折叠重新排列分子，得到一个如图13的图像

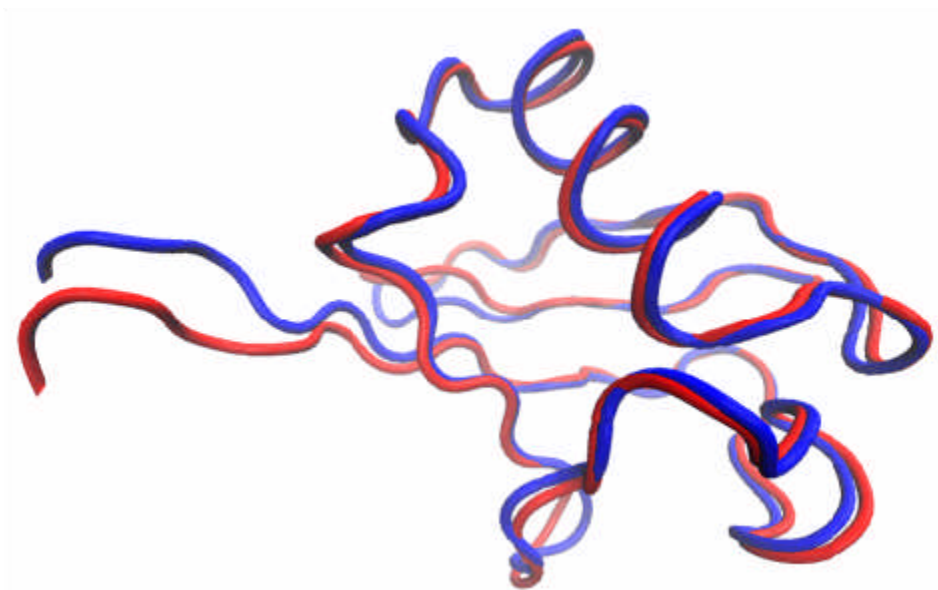


图 13 泛素最初和最后状态三维结构拟合

## 2. 6 用颜色来显示偏离值

一旦两个分子被排列在一起，你可以比较两个原子的位置。你可以用一个预写的脚本（因为直接写比较麻烦）来完成。如果你喜欢刺激的话，那就看一下这个脚本。如果你已经对此很熟练，写这样一个脚本就是小菜一碟了。从文件中运行一个脚本，可以用 Tcl 源命令：

source file – runs a script from a text file

1 脚本在vmd-tutorial-files的文件coloring.tcl中。在VMD Tk Con窗口中应用cd命令找到这个目录。它会改变晶体分子的beta值，反映在平衡后分子中原子的位置改变。我们输入以下命令，运行这个脚本：

```
source coloring.tcl
```

2 要看到新的beta值，需设置分子crystal的Coloring Method为Beta， 隐藏分子simulation（通过双击VMD主窗口中的D实现）。

晶体分子现在依据模拟中的总改变程度上色。为了进一步的工作，我们来调整一下颜色的范围。

3 在Graphical Representations窗口中，选择Trajectory标签。在Color Scale Range标签下，你可以设置beta值颜色范围的最大和最小值。确保晶体分子右边的显示法被选定，然后输入0和5，单击set。这就把颜色的范围设置为0到5埃之间。

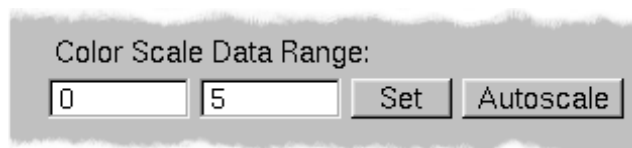


图 14 Representation 窗口中的颜色范围控制

4 现在，从VMD主窗口中选择Graphics—— Colors，打开Color Controls窗口（如图15），选择Color Scale标签。

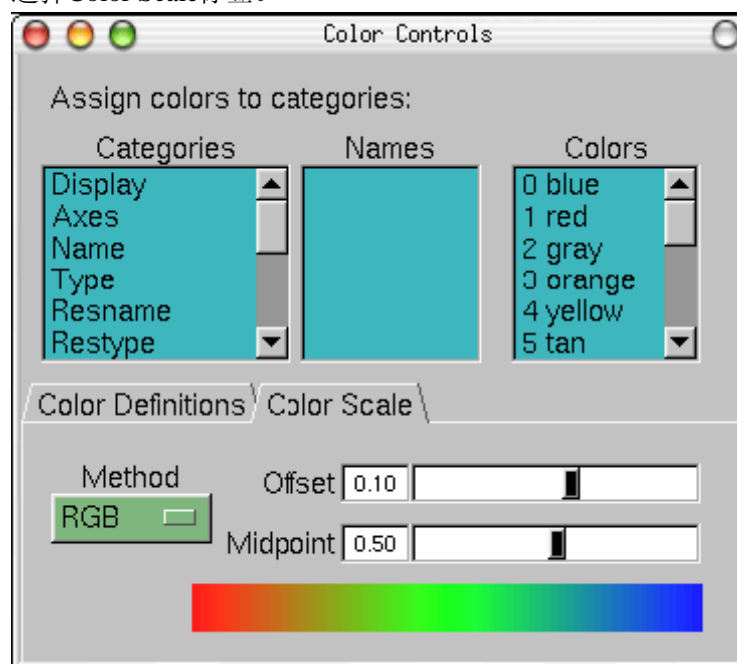


图 15 显示 Color Scale 标签的 Color Controls 窗口

5 在color scale的Method菜单中，选择BWR。这样就可以把位置变动小的原子（0埃）染上蓝色，把位置变动大的原子（5埃）染上红色，在两者之间的染上白色。

6 你现在也可以调整color scale的 Midpoint来改变被染成白色的原子位置变动的水平。试试把中点设为0.1。

你现在应该明白怎么对一般数据进行连续的上色操作。看一看你的分子，它如图 16 所示。你现在应该可以确定泛素的那一部分是稳定的（蓝色），哪一部分是松散的（红色）。总起来说，末端和 loop 是灵活的，容易动的，而  $\alpha$  螺旋和  $\beta$  折叠相比之下是不容易活动的。

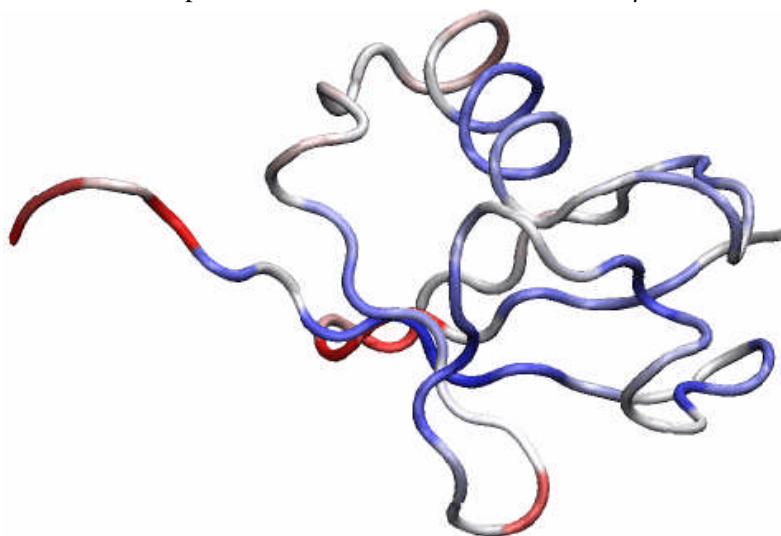


图 16 根据 1ns 平衡后泛素分子着色

### 3 Trajectories, Macros and Labels

在第三单元，你将要学习怎样载入运动轨迹，创建 macros，在原子和键上标注标签，用简单的 tcl 脚本计算一个运动轨迹的 RMSD 值。在最后，你应该通过看 RMSD 点分布来确定泛素系统是否达到平衡。

#### 3.1 导入运动轨迹


你现在要学习导入与时间相关的坐标系统，叫做运动轨迹（Trajectory）。你可以用你的系统来看一场电影了。

Trajectory文件通常是二元文件，包含了系统的多套坐标。每一套坐标对应者时间中的一帧。DCD文件可以作为Trajectory文件的例子。Trajectory文件不包括PSF文件中的内容。因此，我们需要首先导入结构文件，再在结构文件的基础上导入Trajectory文件，导入方法如第二单元中所介绍。

1 启动一个新的 VMD 窗口

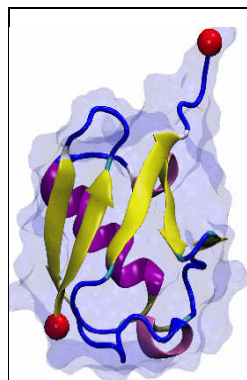
2 导入PSF文件ubiquitin.psf，方法如第二单元所述。

3 在Molecule File Browser窗口，单击Browse按钮，确保ubiquitin.psf在菜单上已被选中。在vmd-tutorial-files中找到文件pulling.dcd，单击OK，然后单击Load按钮。在它们导入分子的时候，你可以看到这些帧。

4 在Trajectory文件被导入之后，你会看到Trajectory的最后一帧。想要回到第一帧，你会用到Animation Tools，它的用法会在后文详细解释。在主菜单的左下侧，单击。

5 选择Graphics——Representations，在Drawing Method中，选择NewCartoon，在Selected Atoms窗口，键入protein

6 在同一个菜单中，双击Create Rep按钮，新建另外一个图形显示方法，在Drawing Method中，选择Lines，在Selected Atoms 窗口，键入water，然后，双击关闭这个显示方法。



泛素的弹性。泛素在细胞中有许多功能。目前的观点证明，这些功能依赖于泛素的弹性。

只有一部分蛋白质表现弹性，比如说肌联蛋白，它是肌肉的弹性源泉。这些分子的弹性依靠的是 $\beta$ 折叠中残基之间的氢键，像泛素中的一样。你刚载入的trajectory是一个泛素分子的AFM（原子力显微镜）实验的模拟。我们要看的是当它被从一端牵拉时是怎么伸展开来的。

在分子动力学模拟中，在开始模拟之前，像在AFM实验中显示的那样，首先需要实现蛋白质的平衡。你将有机会在本教程的后面看到这样的平衡轨迹



在下一部分，你会学到怎样用一个有用的叫做**macros**的特征来创建显示方法。一旦你创建了与**trajectory**有关的显示法，下一部分就会告诉你如何用**Animation Tools**来看**trajectories**。

### 3.2 Macros

你现在可以做出与你在第一单元做出的相似的图形显示方法。当你创建这些输入法时，你会学到**macros**。**Macro**表示一种选择的文本。当你常用一些显示方法的时候，**macro**就很有用了。**Macro**可用你在第二单元学到的**atomselect**命令创建。

**atomselect macro name selection – creates a macro for selection**

泛素由5个 $\beta$ 折叠和1个 $\alpha$ 螺旋构成。 $\beta$ 折叠在蛋白质的伸展态中扮演重要的角色。要为这些结构创建**macro**：需要

- 1 通过选择Main——Extensions ——Tk Console.打开Tk Console窗口。
- 2 在Tk Console窗口中，键入atomselect macro bstrand1 {protein and resid 2 to 6 }

这就为第一个 $\beta$ 折叠创建了一个**macro**，包括残基2——6。

对于其它的结构，你可以用第一单元介绍过的**sequence viewer**找出哪些残基是属于这些结构的，然后创建相似的 **macros**。

- 3 确保你现在处于**trajectory**的第一帧，**STRIDE**(在**VMD**中计算二级结构的程序)会确定这一帧的结构

- 4 选择Extensions——Analysis——Sequence Viewer菜单

就像你在第一单元中学到的，第二个颜色栏表示的是蛋白质的结构特色，呈现黄色的部分指的是 $\beta$ 折叠。

- 5 用鼠标点击并且拖动第二个 $\beta$ 折叠，使它高亮显示。这步操作会在**Graphical Representations**窗口产生一种新的显示方法。

- 6 在**Graphical Representations**窗口中，点击新的显示法，与选择相对应的文本会出现在**Selected Atoms**窗口。你会得到（键U

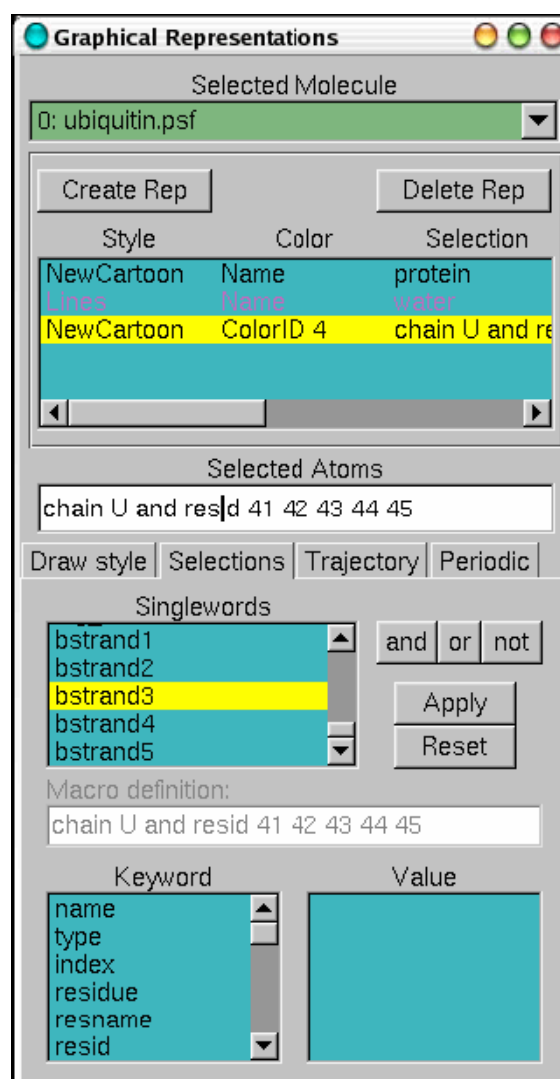


图 17 macros 列在**Graphical Representation** 菜单的 **Selection** 标签中

和残基12 13 14 15 16)

7 在 Tk 控制台中键入

```
atomselect macro bstrand2 { chain U and resid 12 to 16 }
```

用此文本来建立一个新的 macro。这个 macro 叫做 bstrand2，包括蛋白质链 U 的 12 到 16 残基。

8 注意到sequence viewer extension定位了5个 $\beta$ 折叠。你已经为前两个新建了macros；用 bstrand2中用到的sequence extension为剩下的三个建立macros。

一旦macro被建立，你可以在Tk控制台中用它，也可以在Representations selections中用它。

你建立的macros和VMD自带的macros可以在Graphical Representations的Selections标签中看到。在Singlewords窗口中有macros的列表。单击一个macro会在Macro Definition中显示它的定义。双击会选定它，同时会在Selected Atoms中输出它的定义

9 删除用sequence viewer创建的图形显示法。

你现在用第三和第五个 $\beta$ 折叠新建一个显示法。

10 在Graphics——Representations菜单中，单击Create Rep按钮。

11 在Selected Atoms窗口中，删除其中的文本。

12 点击Selections标签，在Singlewords的列表中寻找，直到找到你新创建的macros。

13 双击bstrand3，点击按钮or，然后双击bstrand5（图17），再点击Apply按钮。

14 在Draw Style标签中，选择NewCartoon表示法，染成黄色。你现在应该可以看到 $\beta$ 折叠。

15 现在，用其它3个 $\beta$ 折叠创建一个相似的表示法。单击Create Rep键，现在，在Selected Atoms中输入bstrand1 or bstrand2 or bstrand4.。键入后立刻就可以看到效果。

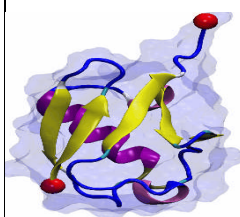
就像你看见的一样，macro 非常有用。当你保存已存的文件时，macros 会被包含在已存文件内。



Macros。你可以创建一些macros，在根目录下把它们存成一个VMD 参数选择文件 .vmdrc（Windows用文件vmd.rc）。在启动VMD的时候，VMD会自动寻找并且读出所有macros。要想了解VMD开始文件的更多内容，参考VMD用户手册。

要完成这一部分的学习，你将学习一个非常有趣的显示法，它展现了我们看到的 trajectory

的关键特征。这种表示法就是氢键显示法。



氢键。氢键可以以多种方式来稳定蛋白质，在  $\beta$  折叠骨架原子之间的氢键是蛋白质弹性的标志；它们使蛋白质稳定，而且，它们在机械力作用下的形成与断裂赋予蛋白质弹性。

16 通过选择betasheet 和 backbone 来创建一个新的显示法。选择Drawing Method —— Hbonds, 把它染成红色, 选择Coloring Method—— Color ID, 在选项内, 把Distance Cutoff 设为3.2, Angle Cutoff 设为 30, Line Thickness设为 5.

你现在可以从这个伸展状态的trajectory中看出泛素最重要的特征，你的蛋白质现在看起来应该与图18相似。

17 保存VMD环境，这样如果你想回来继续研究此部分教程，你就不用再重新设定那些显示法。保存时用到你在第一单元学习的File —— Save State菜单选项。.



重新应用保存状态。当你已经保存了一个系统，你可以用它观察不同的坐标（PDB 或者另外的 trajectory）。如果它们具有相同的 PSF 文件，你可以用以下的步骤实现操作。

导入你保存状态 文件：File——Load State

选择你以前保存的文件（比如 nice-ubiquitin . vmd）

打开 Molecule——Delete Frames 菜单，你需要删除目前导入的所有帧。默认选择会执行这样的操作。点击 Delete 按钮。

导入新文件yournewtrajectory.dcd到PSF，执行菜单为File ——Load Data Into Molecule.

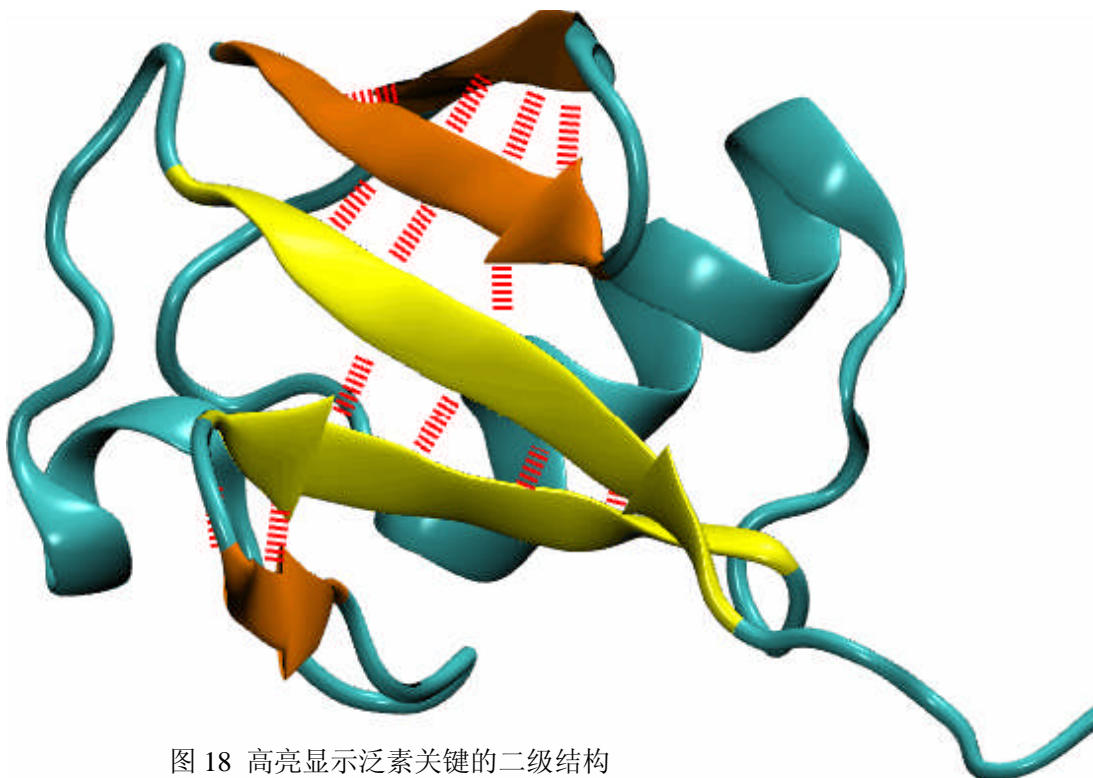


图 18 高亮显示泛素关键的二级结构

### 3.3 主菜单动画工具

既然现在你已经有了非常好的显示法，你就可以观察trajectory的特点。Animation Tools可以帮助你。在主菜单中包含了研究trajectory的所有的动画工具。它们位于菜单的底部（图19）。



1 试着应用  跳到 trajectory 的末尾，用  回到开头。你可以看到，trajectory 的最后和初始状态分别对应蛋白质的伸展和折叠状态。

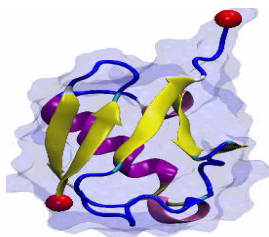


图 19 动画工具

2 再次打开water representation.，回到VMD主窗口，选择Graphics ——Representations菜单，双击标有文本water的显示法，将其打开。

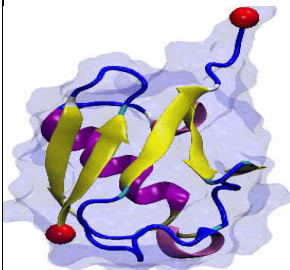
你可以通过点击滑块并来回拖动以实现对你的trajectory的操纵。你可以在任何时候停下，或者以你想要的速度拖动。当你想看一个trajectory，想测量一个有趣现象发生时间时，这个功能是很有用的

3 用滑块观察 trajectory 起始蛋白质周围的水分子的运动。



自由能最小化。注意到水盒的形状从管状变成球形。要实现能量最小化，暴露在真空中的水分子要尽可能少，所以水盒要采取球状构型。检测一下水形成此种构型有多快（每一帧的步长与 10ps 相当）。

4 现在，双击 Water Representation，从 Graphical Representations 窗口删除它。这样做可以更近地观察蛋白质。滑动滑块，观察蛋白质是怎样伸展开的。你观察到什么特点了吗？



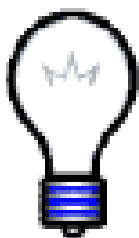
打断氢键。看一看中间黄色的两个  $\beta$  折叠。它们靠氢键连接在一起。注意  $\beta$  折叠是如何先分离的，这意味着氢键首先断裂。我们在后面会进一步研究这种相关性。

在 Animation tools 的下部分，你能找到需要播放一个动画所有的必需工具，这些工具不需要滑块。它是由 Play 按钮来控制的，它可以播放或者倒放。

5 把 trajectory 倒放，你认为这样做，蛋白质又会回到自然折叠态吗？

改变你的动画播放速度的方式有两种。你可以用 Speed Slider 来调节，你也可以用 Step Window 调节步长，如果步长值设为 3，动画会每 3 帧播放，这样它就可以让播放速度快一些。

6 把步长值设为 5，然后播放，可以看到它播放速度快了，但是看起来不如以前平滑了。然而，当你看一个很长的 trajectory 的时候，这种播放方式就比较方便了。



循环方式。当播放动画的时候，你可以在 Style Chooser 中选择 3 种循环方式，它们是 Once, Loop 和 Rock。一个控制窍门：

回到 trajectory 的开始（第 0 帧）

把步长数设为在 trajectory 中的总帧数（100）

把 style 设为 Rock

把速度设为最低

播放

你现在可以比较 trajectory 中的第一和最后一帧。这个诀窍用起来比点击 Start 和 End 按钮方便。

在 Animation Tools 左边的窗口中输入帧的序号，可以跳到 trajectory 中的任意一帧。



重新计算二级结构。你可以让VMD用STRIDE一个新的帧的二级结构。图18是在第28帧，通过在Tk控制台中键入命令`mol ssrecalc top`实现的。

注意： Animation Tools在顶层分子的帧中循环，但是它可以应用到所有活动分子中。

### 3.4 标签

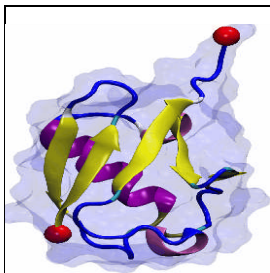
在 VMD 中，你可以在选定的内容上标注标签。我们现在就来感受应用标签的乐趣。标签用鼠标选择。在这个例子当中，我们会涉及到标注原子和键的标签，角和二面角也可以标注标签。

1 选择Mouse——Labels——Atoms菜单项。鼠标设为“Display Label for Atom”模式。现在你可以点击你的分子中的任何一个原子，为它标注标签。再点击一次会删除标签。

我们现在来试一试键。

2 选择Mouse——Label——Bonds菜单，鼠标模式选择为“Display Label for Bond”

你可以为lysine48和C末端的 $\alpha$ 碳建立一个VMD Representation。在pulling simulation中，前者是固定的，后者被500pN的恒力牵拉。



K48 残基。在第一单元中提到，多泛素链可以通过一个泛素分子的碳末端和另一个泛素分子的 K48 残基连接而形成。这个模拟模仿的是在这种连接中 C 末端的牵拉力。

要想找到这些原子的索引，

3 在 Tk 控制台中键入以下命令，建立一个包括这两个原子的选择  
`set sel [atomselect top "resid 48 76 and name CA"]`

4 获得索引\$sel get index

命令返回的索引值为 770 1242。





PDB 和 VMD 原子编号。注意在 `pdb` 文件中这些原子的编号是 771 和 1243。VMD 从 0 开始计数索引，因此 VMD 可以读取 `Representation` 中的文本数字。由于 VMD 并不从 PDB 文件中读取索引，这是唯一的例子。其他的关键字，就像残基，与 PDB 文件中的一致。

##### 5 用选择的索引值770 1242新建一个VDW Representation

6 现在，先后点击两个原子，你应该得到连接两个原子的一条线，在线的附近显示的值是两个原子之间的距离，单位是埃（图 20）。

得到的距离值是本帧中两个原子之间的距离值



标签。标签的快捷键有: 1 Atoms, 2 Bonds。你可以用它们来代替 `Mouse menu`。当用这些快捷键时，确保 `Open GL display` 窗口是活动的。

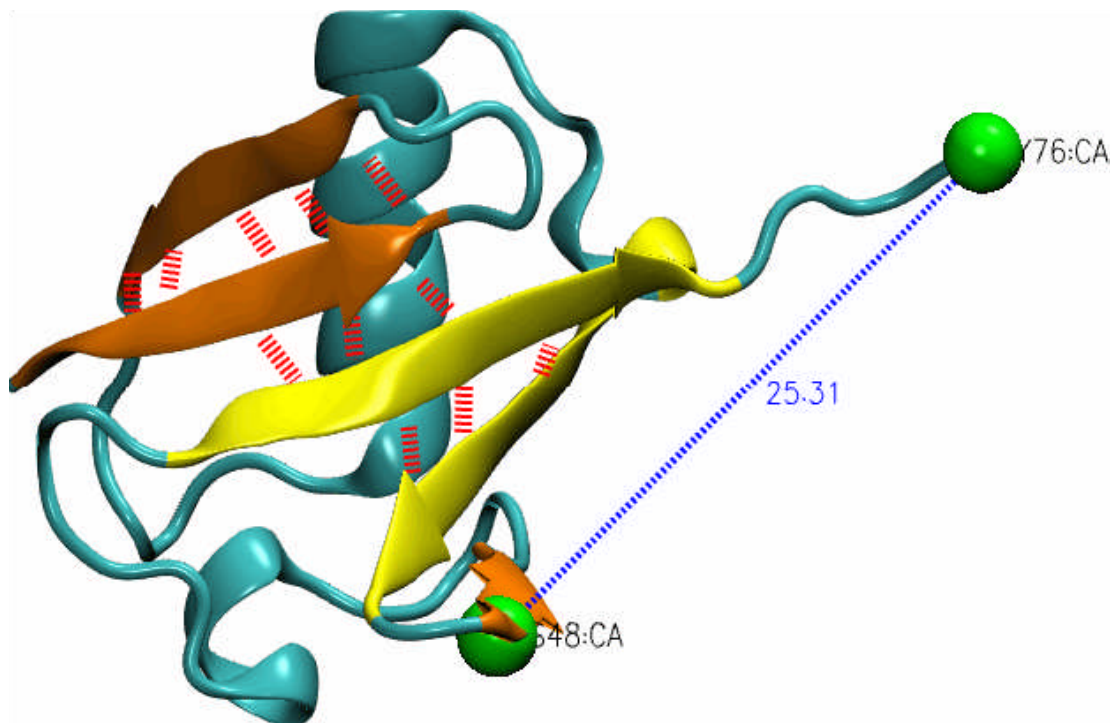


图 20 模拟中固定和被拉动的原子的键的选择。两个被选择原子都已用黑色标签标注。键用蓝色表示，蓝色数字表示的是两个原子之间的距离（埃）

7 通过Graphics —— Labels菜单项，你还可以行使另外一些功能。在窗口的左侧（图21），有一个下拉菜单，你可以在其中选择标签的类型

（Atoms, Bonds, Angles, Dihedrals）。现在，设为Atoms。你可以看到你标注的原子的列表。

8 单击其中的一个原子，这个原子的信息就会显示出来。你可以通过单击按钮来删除，隐藏或显示这些标签。

注意到这些信息对于选择是很有用的。原子信息对应着当前帧，当帧发生改变时，信息也随之发生改变。

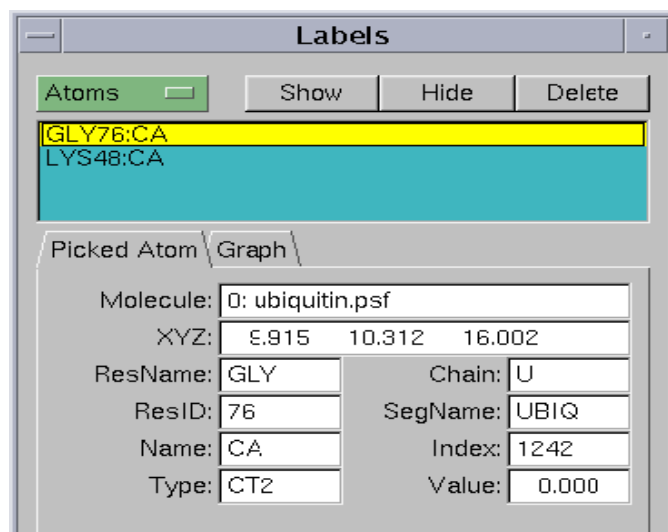


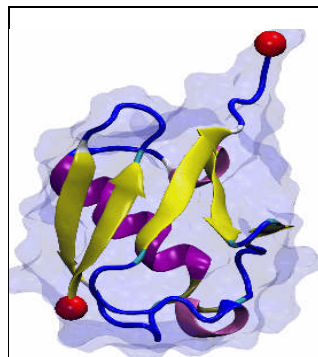
图 21 标签窗口

9 现在，在 Label 窗口，选择 label type 为 Bonds，然后选择你要标记的键。注意到信息只与键中的第一个原子对应，但是 Value field 数与键的长度对应，单位是埃。单击 Graph 标签，选择你标记的原子 770 和 1242 之间的键。单击 Graph 按钮。这会创建两个原子之间的距离图。你也可以把这个数据保存到文件里，操作方法为：单击 Save 按钮，然后用一个标图程序来观察数据。



xmgrace。VMD用的是二维标图程序xmgrace。这个程序可以从<http://plasma-gate.weizmann.ac.il/Grace/>下载。在Mac OSX中，xmgrace在X11环境中运行。运行xmgrace之前要保证X11正在运行。

10 你现在可以关闭xmgrace或者相应的标图程序。



泛素去折叠。图形展示了键长随时间的长度变化。你可以想象为当分子被一个恒力牵拉时，分子“长度”的变化图。注意到曲线的起始部分是平的，就像拉力没有影响分子结构一样。突然，距离有了较大的增长。看一看这发生在什么时候：当两个黄色的 $\beta$ 折叠分离的时候。这意味着把 $\beta$ 折叠结合起来的氢键是蛋白质去折叠的第一个阻力，一旦氢键断裂了，蛋白质以更有规律的方式去折叠。试着确定在第二次突变中发生了什么。你会发现泛素去折叠的关键特点。

### 3.5 关于 Tcl 脚本的一个例子：计算 trajectory 的 RMSD 值

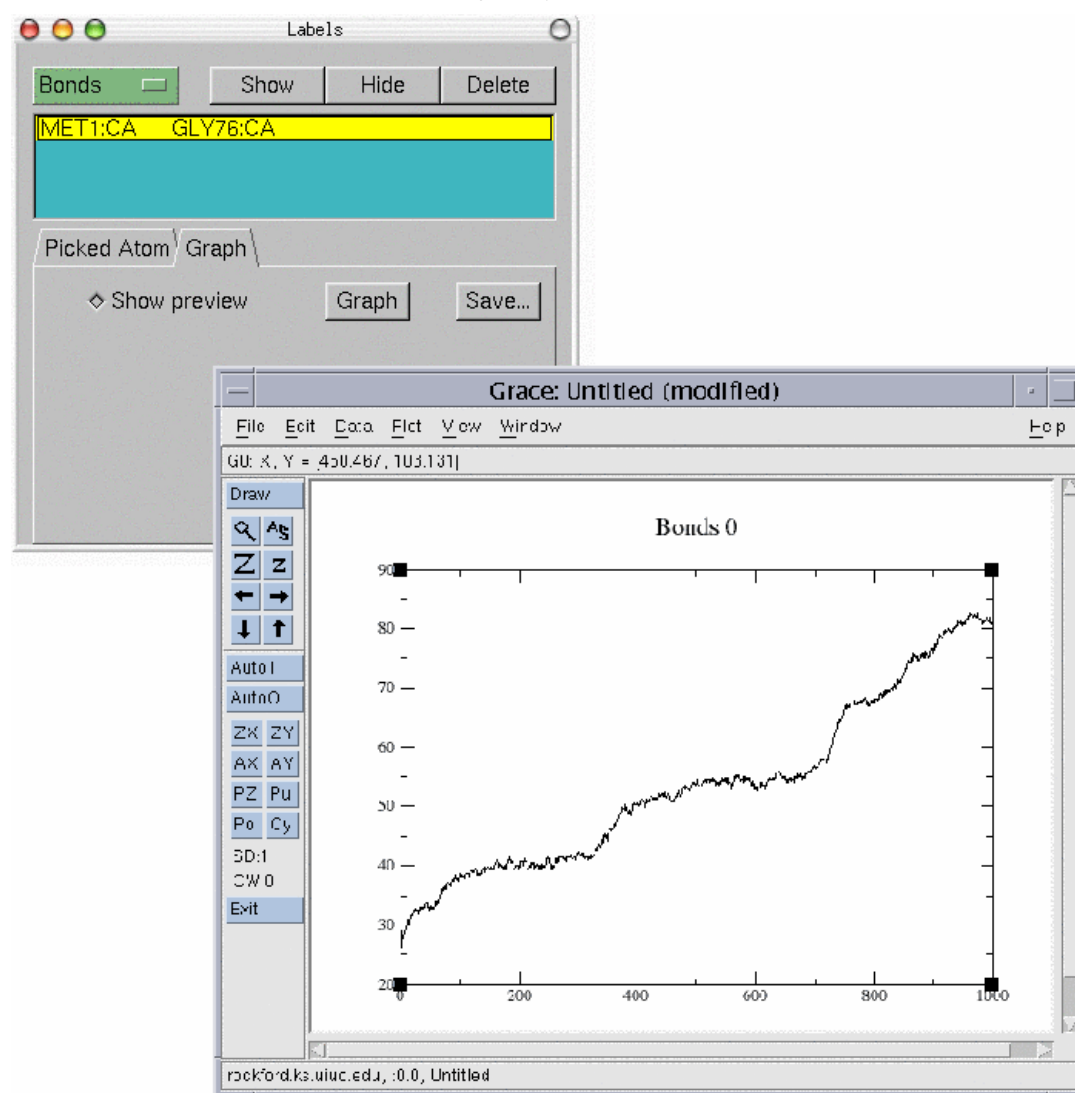
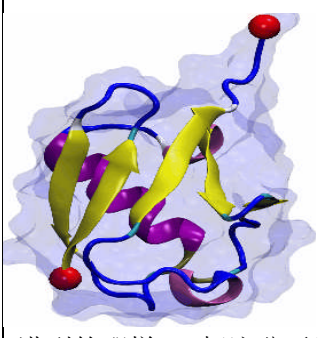


图 22 标签窗口和选择的键对时间作图，用到 Graph 按钮和 xmgrace。

在 MD 分析中，VMD 是一个强大的工具。在这一部分你将要使用 Tcl 脚本来分析 trajectory。你需要导入一个新的关于泛素平衡系统的 trajectory。在 MD 运行时，你将要使用到一个计算蛋白质 RMSD 值的短脚本，来确定这个系统是否已经达到平衡，是否可以继续进行模拟。



偏离值均方根。偏离值均方根（RMSD）是从数字上衡量两个结构的不同的。它的定义是：

$$RMSD = \sqrt{\frac{\sum_{i=1}^{N_{atoms}} (r_i(t_1) - r_i(t_2))^2}{N_{atoms}}} \quad (1)$$

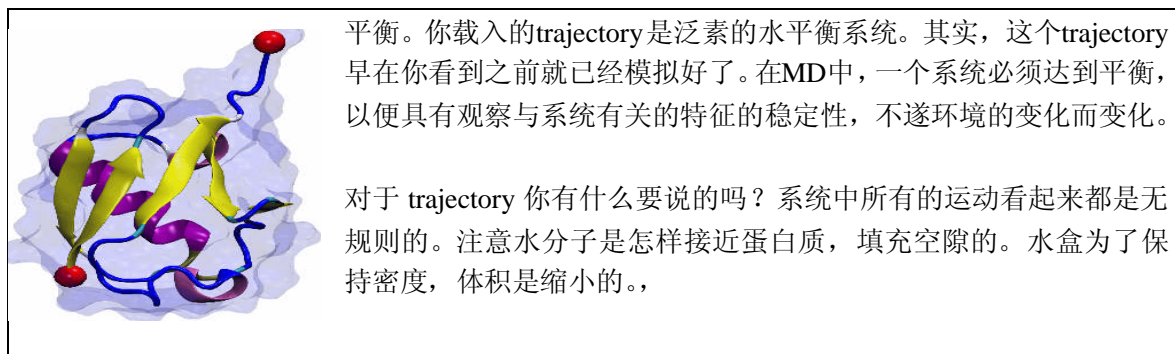
$N_{atoms}$  是指被比较位置的原子数， $r_i(t)$  指的是原子  $i$  在时间  $t$  所处的位置（或者，如果你比较两个分子，像第二单元中讲到的那样， $t$  标注分子）

1 删除目前的 trajectory，在 Main 窗口中单击选择单一的分子。然后用菜单项 Molecule——

Delete Frames。此窗口的默认值会删除整个trajectory。单击Delete按钮。

2 用File——Load Data Into Molecule导入平衡的trajectory到psf文件中。这个文件是equilibration.dcd

3 打开water representation，用你以前学过的Animation tools来看一看trajectory。



现在你会明白怎么判断蛋白质是否已经达到平衡。其中一个重要的方法就是通过判断trajectory中蛋白质的RMSD值。

4 我们将要用到的脚本叫做rmsd.tcl。这是脚本的内容：

```
set outfile [open rmsd.dat w]
set nf [molinfo top get numframes]
set frame0 [atomselect top "protein and backbone and noh" frame 0]
set sel [atomselect top "protein and backbone and noh"]
set all [atomselect top all]
# rmsd calculation loop
for { set i 1 } { $i <= $nf } { incr i } { $sel frame $i
$all frame $i
$all move [measure fit $sel $frame0]
puts $outfile "[measure rmsd $sel $frame0]"
}close $outfile
```

5 这个脚本执行的是以下任务：

打开文件rmsd.dat  
set outfile [open rmsd.dat w]

得到 trajectory 中的帧数，把此值赋予变量 nf  
set nf [molinfo top get numframes]

选择分子的第一帧，作为其他帧比较的标准。选择包括蛋白质的骨架原子，不包括氢。  
set frame0 [atomselect top "protein and backbone and noh" frame 0]

如前，为没有确定的帧建立同样的选择。

```
set sel [atomselect top "protein and backbone and noh"]
```

选择所有的原子

```
set all [atomselect top all]
```

# denotes comment. 之后的文本

# rmsd calculation loop

在 trajectory 中的所有帧中循环。

```
for { set i 1 } { $i <= $nf } { incr i } {
```

把选择\$sel更改为要比较的帧（注意所选的内容在每一帧中必须一致，否则需要增加一个\$sel更新命令）

```
$sel frame $i
```

更改所选的\$all为当前帧

```
$all frame $i
```

计算同时满足这两个选择的矩阵，把这个矩阵应用到第二个选择中，排列两个分子。

```
$all move [measure fit $sel $frame0]
```

计算这两个选择中的 RMSD 值，在文件中写入：

```
puts $outfile "[measure rmsd $sel $frame0]"
```

你可以在系统中用这个脚本来检测平衡。

6 在Tk控制台中键入source rmsd.tcl（确保你所在的目录为vmd-tutorial-files）。这样脚本中所有的命令都会运行。脚本会写一个包括蛋白质骨架RMSD值随时间变化值，为rmsd.dat文件。

在VMD之外，你可以用标图程序来观察得到的数据。这样的程序有gnuplot, xmgrace, excel, Mathematica.

7 用上述程序之一来绘制文件rmsd.dat（例如，在Unix系统，你可以在终端输入xmgrace rmsd.dat）你能看到RMSD曲线变平了吗？这意味着你的系统达到平衡了。



更多RMSD。你可以试试获取另一个叫rmsd-fullthrottle.tcl的脚本。看一看这个脚本，学习一下tcl中简单的步骤，计算一下每一个残基对时间的RMSD值，根据RMSD值为残基上色。

本教程讲到这里也就结束了。我们希望你能从中学习到不少东西，希望VMD的功能能够对你有所帮助。