

PyMOL简介及相关操作

报告人：丁洁女

组员：廖杨洁、杨承峰、段博、谭菲

组号：G13

2013-1-18

主要内容

一、PyMOL简介

- PyMOL概述
- PyMOL的特点
- PyMOL的界面介绍

二、PyMOL功能

三、PyMOL的基本操作

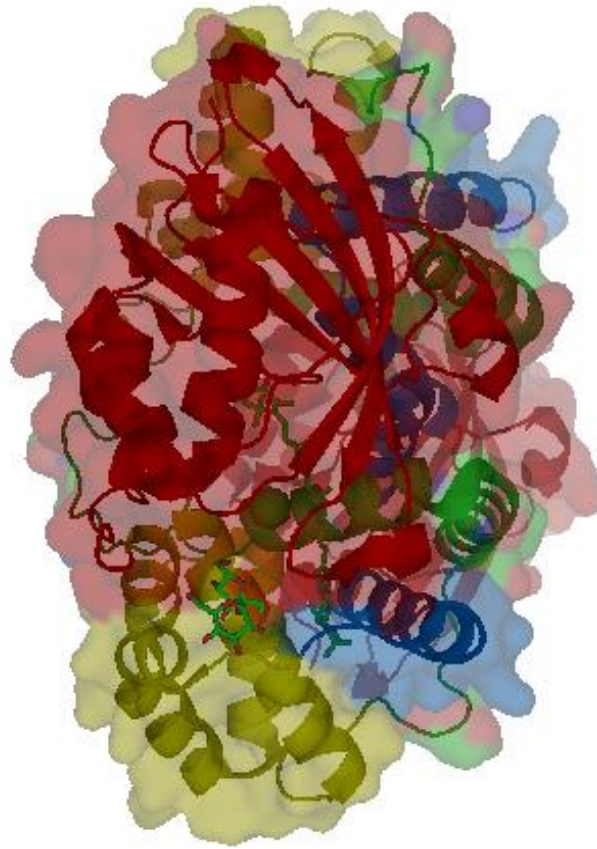
- 鼠标操作
- 命令操作

四、PyMOL应用实例



A PyMOL-generated image on the cover of Nature

Example



PyMOL概述

- PyMOL是一个开放源码，由使用者赞助的分子三维结构显示软件。由Warren Lyford DeLano编写，并且由DeLano Scientific LLC将它商业化。
- Pymol名字的来源：“Py”表示该软件基于python这个计算机语言，“Mol”则是英文分子（molecule）的缩写，表示该软件用来显示分子结构。
- PyMOL适用于创作**高品质**的小分子或是生物大分子（特别是蛋白质）的三维结构图像。软件的作者宣称，在所有正式发表的科学文献中的蛋白质结构图像中，有四分之一是使用PyMOL来制作。
- 网站：<http://www.pymol.org/>

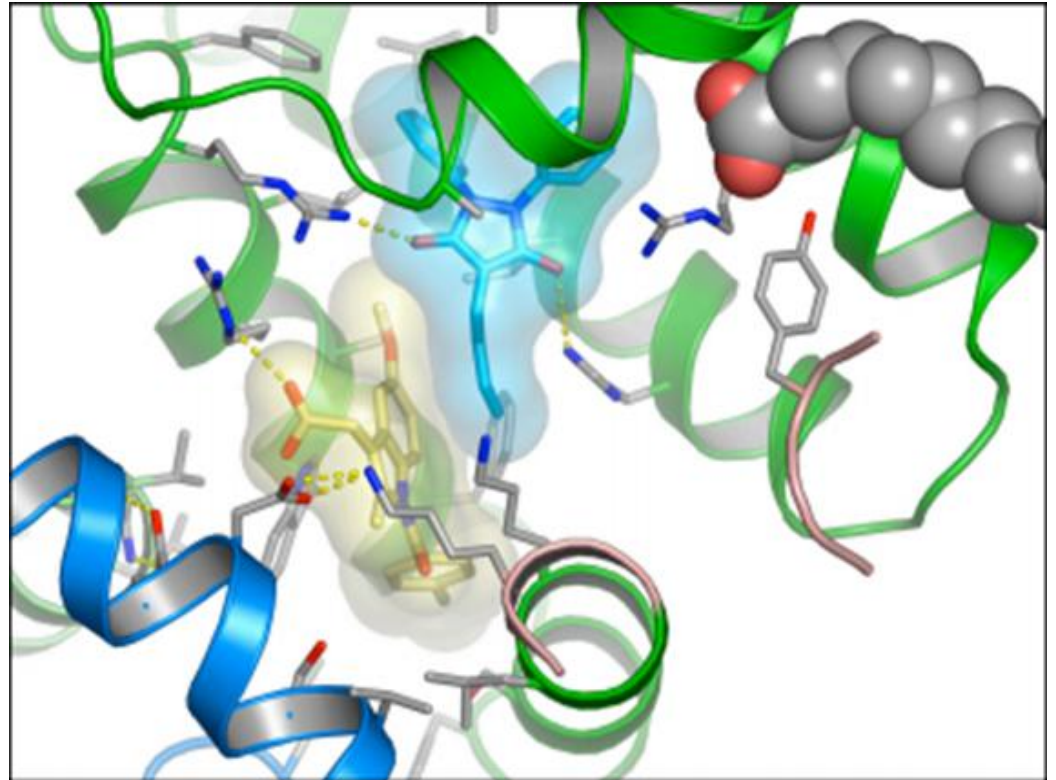
PyMOL的特点

➤ 优点

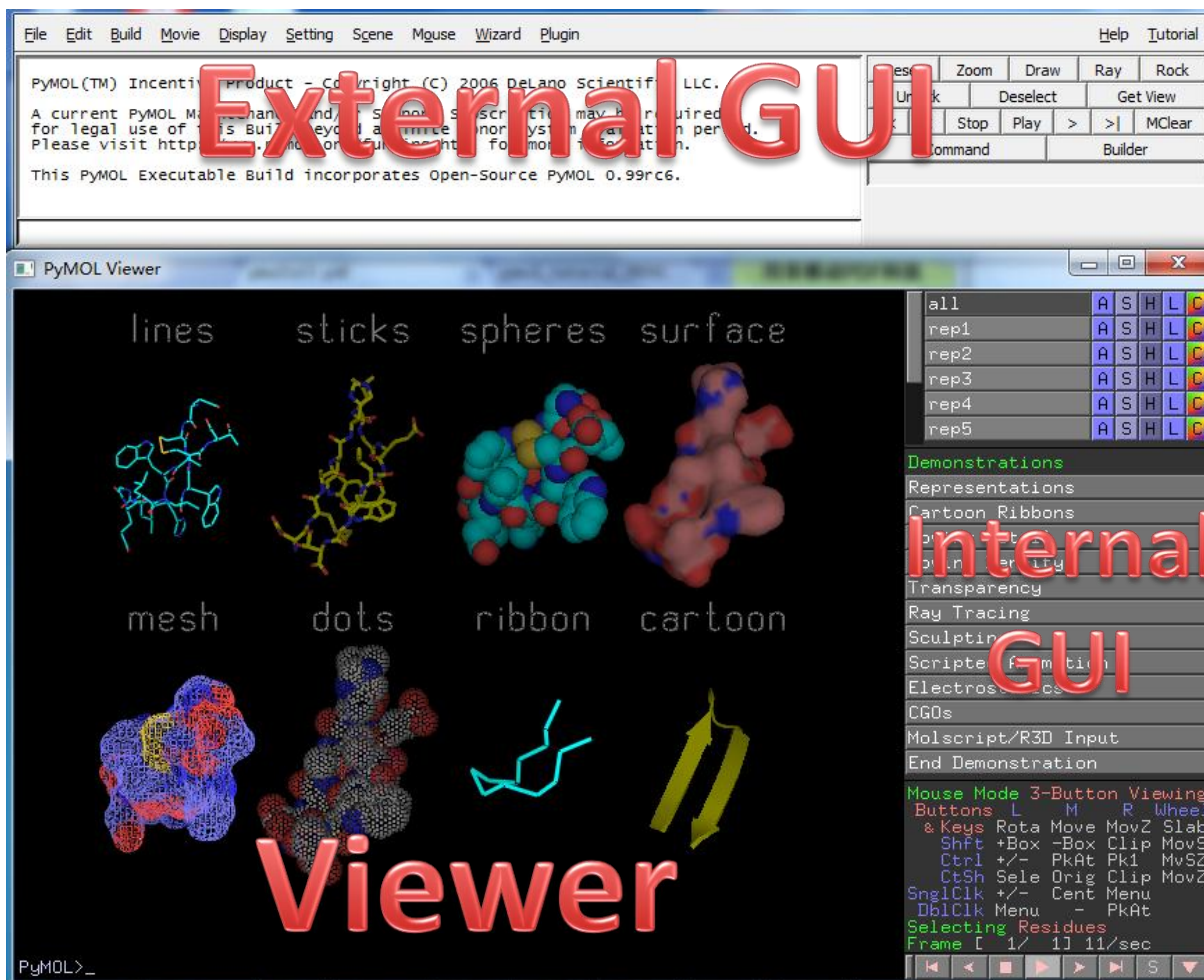
- 强大的分子可视化软件
- 高质量科学论文发表图形
- 动画制作
- 文档文件和会话文件并存
- 鼠标操作与命令行操作
- 免费的开放源码

➤ 缺点

- 缺乏足够的文件资料
- 没有UNDO功能
- 功能不完善

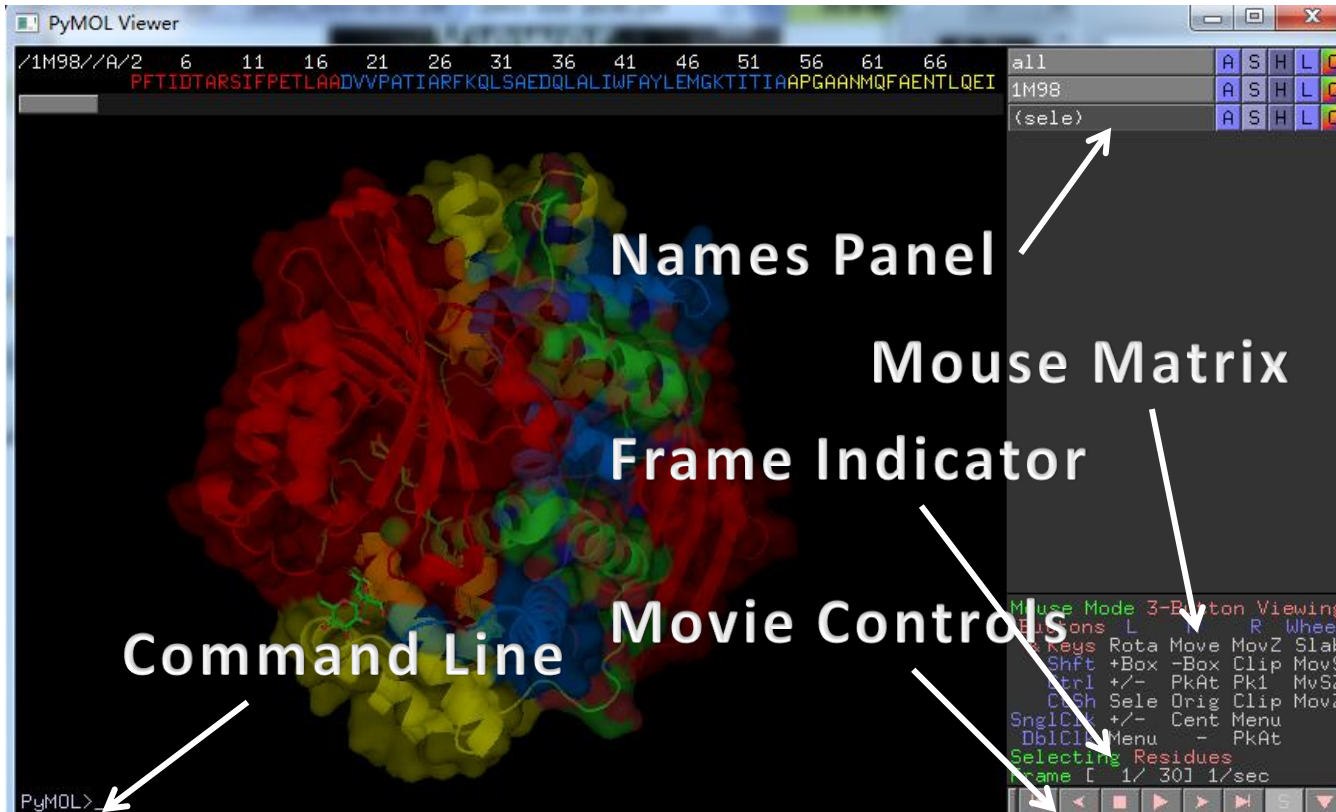


PyMOL的界面介绍



- PyMOL的使用界面：包括一个图形显示窗口和一个GUI窗口。
- GUI是图形用户界面（Graphical User Interface）的缩写，由菜单、按钮、正文框和其他小工具构成。
- Viewer是PyMOL系统的核心。这是一个开放式图形语言（OpenGL）窗口，所有的3D图形在此展示，并且用户可直接操纵这些图形。

The Viewer Window



- A: Action
- S: Show
- H: Hide
- L: Label
- C: Color

- PyMol可以同时打开多个PDB文件，或将某个PDB文件拆分成多个独立单元。每个PDB或独立单元可以通过“A”中的“rename selection”重新命名后显示在Names Panel上。

ASHLC menu

- Action

| Actions: |
|--------------------|
| zoom |
| orient |
| center |
| origin |
| drag |
| preset |
| find |
| align |
| generate |
| assign sec. struc. |
| rename object |
| duplicate object |
| delete object |
| hydrogens |
| remove waters |
| state |
| masking |
| sequence |
| movement |
| compute |

- Show

| As: | Show: |
|------------|------------|
| lines | as |
| sticks | lines |
| ribbon | sticks |
| cartoon | ribbon |
| label | cartoon |
| cell | label |
| nonbonded | cell |
| dots | nonbonded |
| spheres | dots |
| nb_spheres | spheres |
| mesh | nb_spheres |
| surface | mesh |
| | surface |
| | organic |
| | main chain |
| | side chain |
| | disulfides |

- Hide

| Hide: |
|------------|
| everything |
| lines |
| sticks |
| ribbon |
| cartoon |
| label |
| cell |
| nonbonded |
| dots |
| spheres |
| nb_spheres |
| mesh |
| surface |
| main chain |
| side chain |
| waters |
| hydrogens |
| unselected |

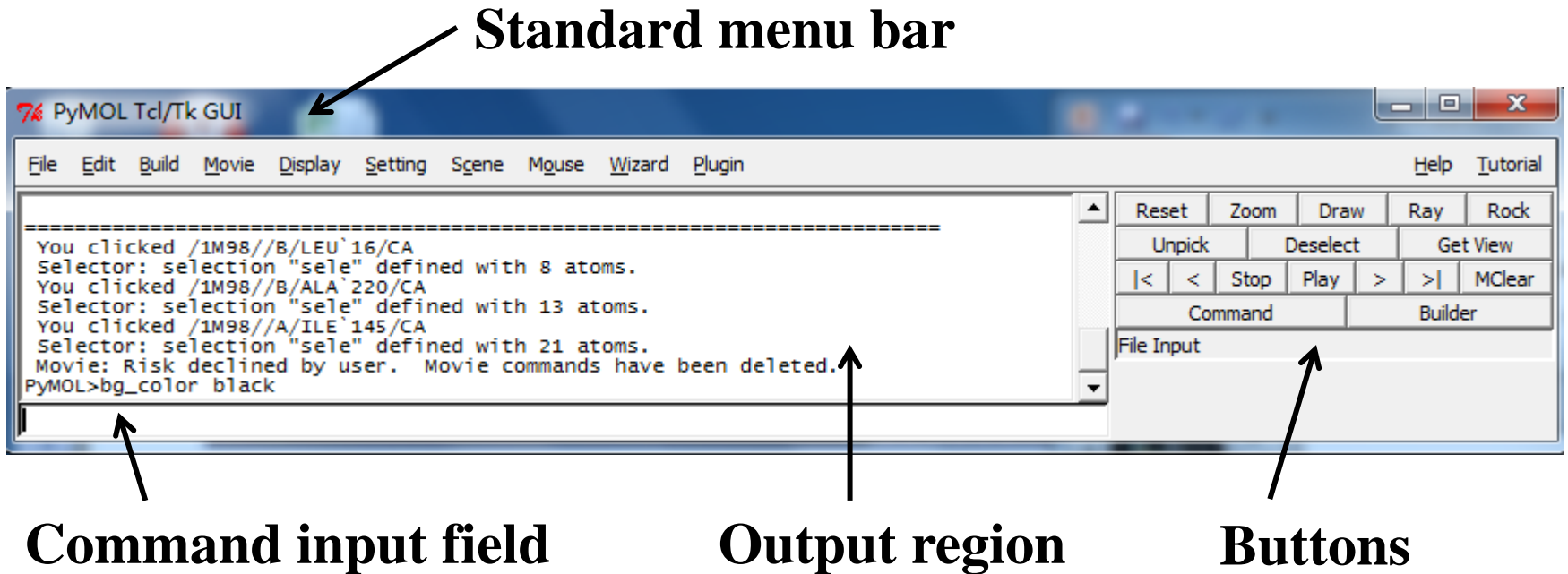
- Label

| Label: |
|--------------------|
| clear |
| residues |
| chains |
| segments |
| atom name |
| element symbol |
| residue name |
| residue identifier |
| chain identifier |
| segment identifier |
| b-factor |
| occupancy |
| vdw radius |
| other properties |
| atom identifiers |

- Color

| Atoms | Color: |
|----------|------------|
| HNOS... | by element |
| CHNOS... | by chain |
| CHNOS... | by ss |
| CHNOS... | spectrum |
| CHNOS... | auto |
| CHNOS... | reds |
| CHNOS... | greens |
| CHNOS... | blues |
| CHNOS... | yellows |
| set 2 | magentas |
| set 3 | cyans |
| set 4 | oranges |
| set 5 | tints |
| | grays |

The External GUI Window

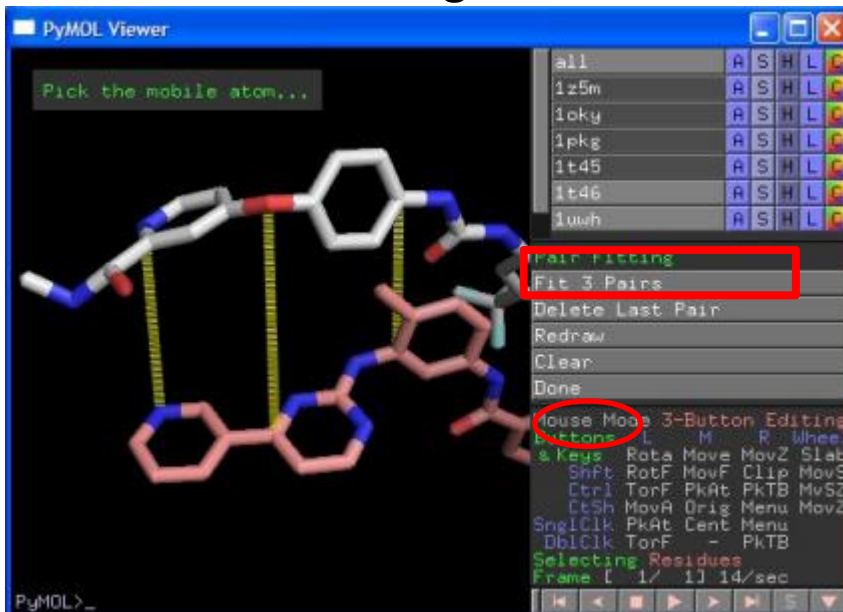


- 与Viewer window相比的优势：能通过Ctrl-X, Ctrl-C, and Ctrl-V使用剪切、复制、黏贴功能

PyMOL功能

👉 比对功能

- ✓ 基于蛋白序列
 - Action -> align
 - Pymol> align (2xuv and n. CA), (hdea and n. CA)
- ✓ 基于原子对
 - Wizard-> Pair Fiting



This screenshot shows the PyMOL command line and a menu. The command line displays the following sequence alignment:

```
76 81
ETDLTQIP MLY VIE
66 71 76
QACTQDKQANFKDKVK
```

The menu shows the 'Align' action for the '2XUV' object. The 'Align' menu is open, showing the following options:

- Align:
- to molecule
- to selection
- enabled to this
- all to this
- states (* /ca)
- states
- matrix from
- matrix to
- matrix reset

The 'Actions' menu is also open, showing the following options:

- zoom
- orient
- center
- origin
- drag
- preset
- find
- align
- generate
- assign sec. struc.
- rename object
- duplicate object
- delete object
- hydrogens
- remove waters
- state
- masking
- sequence
- movement
- compute

👉 测量

✓ 距离

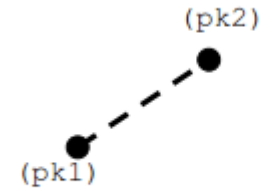
- Pymol> distance (sele1), (sele2)

✓ 角度

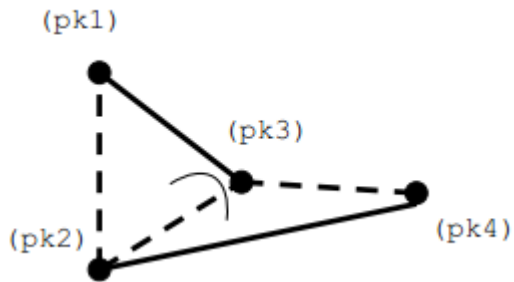
- Pymol> angle (sele1), (sele2), (sele3)

✓ 二面角

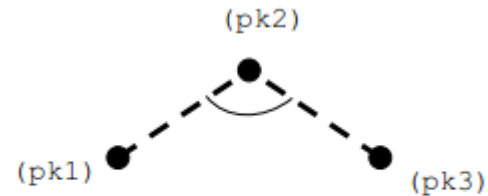
- Pymol> Dihedral (sele1), (sele2), (sele3), (sele4)



```
distance [(sele1), (sele2)]
```



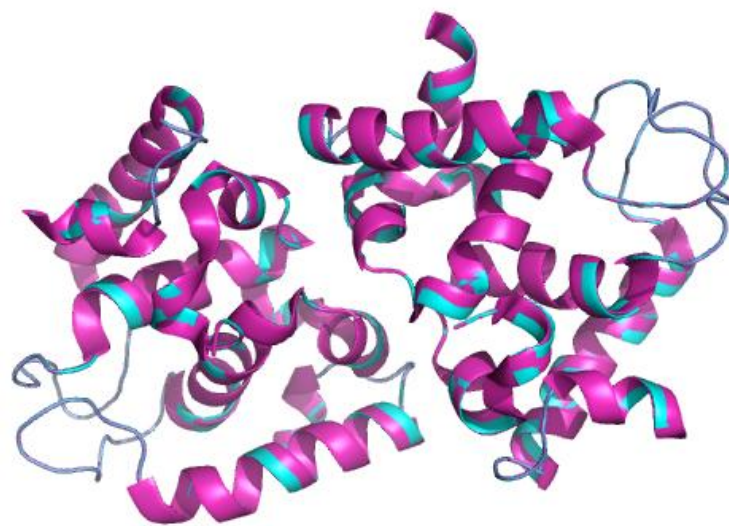
```
dihedral [(sele1), (sele2), (sele3), (sele4)]
```



```
angle [(sele1), (sele2), (sele3)]
```

👉 二级结构归属

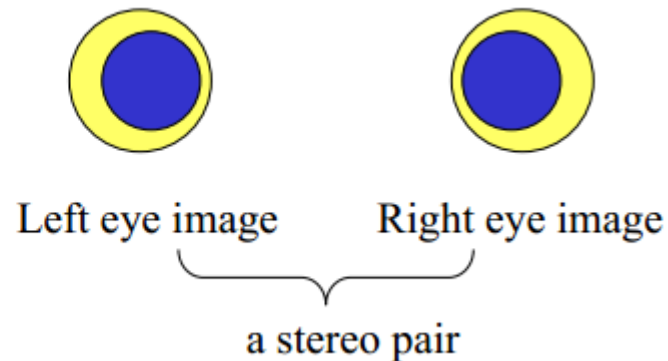
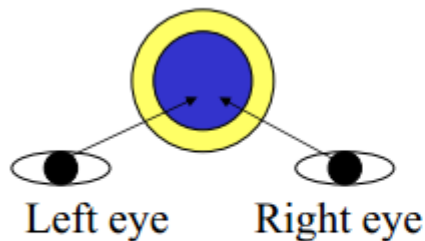
- PyMOL具有快速合理的二级结构归属算法，即“**dss**”，但由于二级结构归属的主观因素，**dss**的结果可能会不同于PDB文件中DSSP程序的结果。
- `Pymol> dss selection`
- 以2xuv.pdb为例
- `Pymol> fetch 2xuv` # 载入对象2xuv
- `Pymol> as cartoon` # 显示cartoon
- `Pymol> color cyans, 2xuv`
- `Pymol> dss 2xuv` # 对2xuv计算二级结构
- `Pymol> fetch 2xuv, hdeb`
- # 载入对象，命名为hdeb
- `Pymol> as cartoon` # 显示cartoon
- `Pymol> color magentas, hdeb`



👉 立体效果

➤ PyMOL能够支持的立体图形模式:

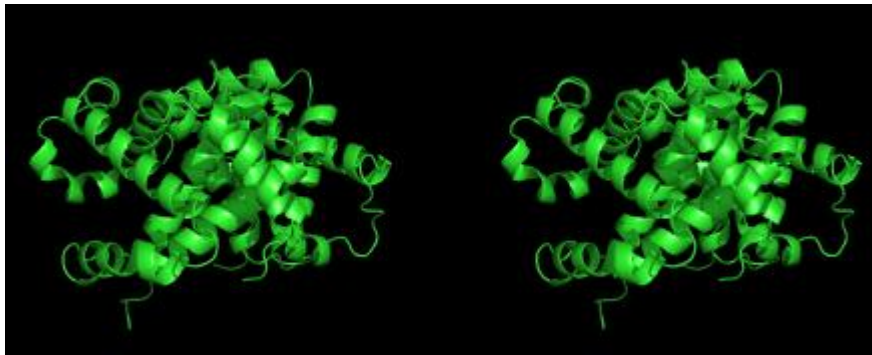
- Crosseye stereo
- **Walleye stereo**
- **Hardware stereo**
- Geowall stereo
- Sidebyside stereo
- Quadbuffer stereo



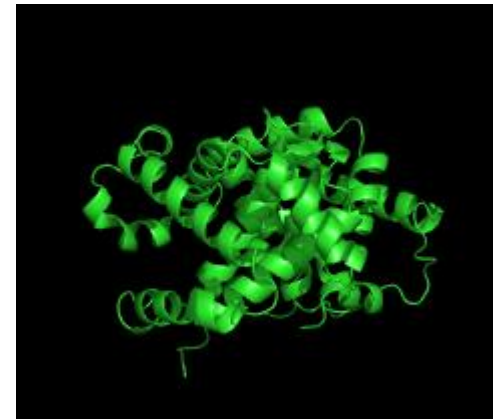
➤ 相关命令

- Pymol> stereo on #开启立体效果
- Pymol> stereo off
- Pymol> stereo crosseye #开启crosseye立体模式
- 注: 如果hardware stereo 可用, 那么quadbuffer stereo 是默认的立体模式, 否则crosseye stereo是默认的。

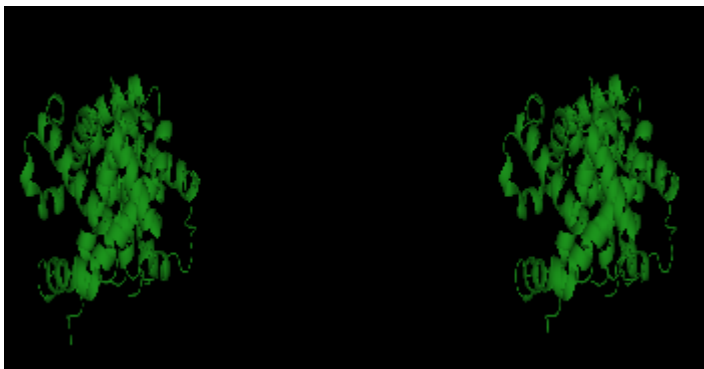
Crosseye stereo



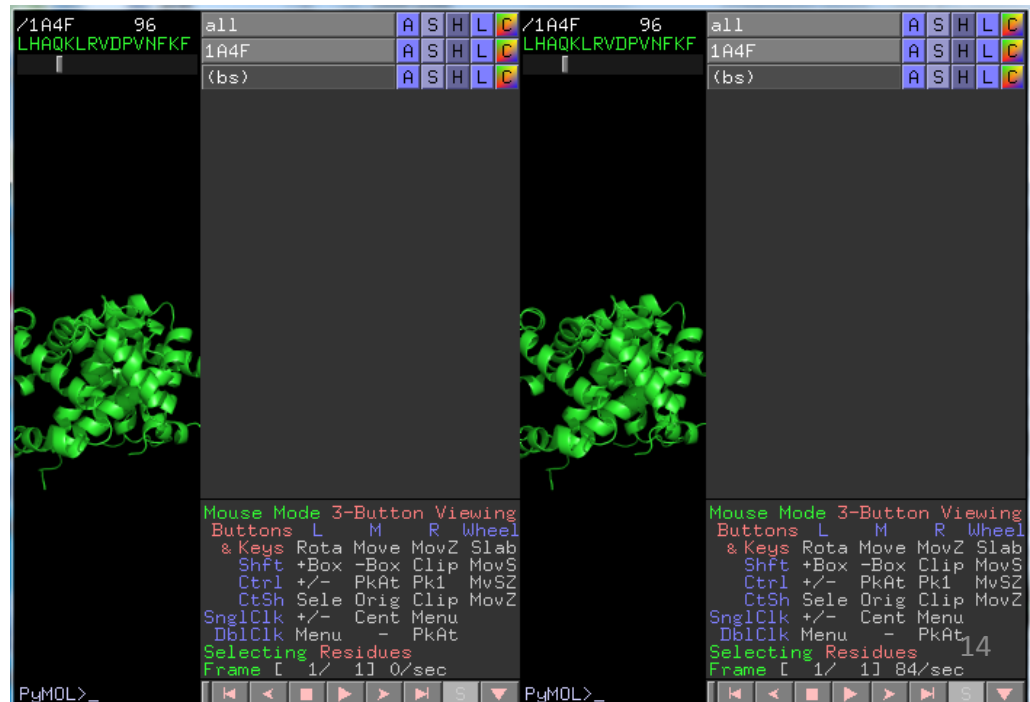
Geowall stereo



Sidebyside stereo

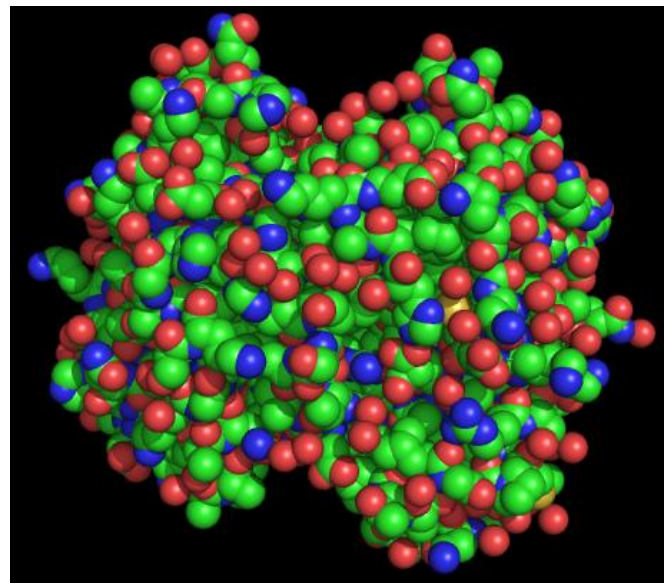
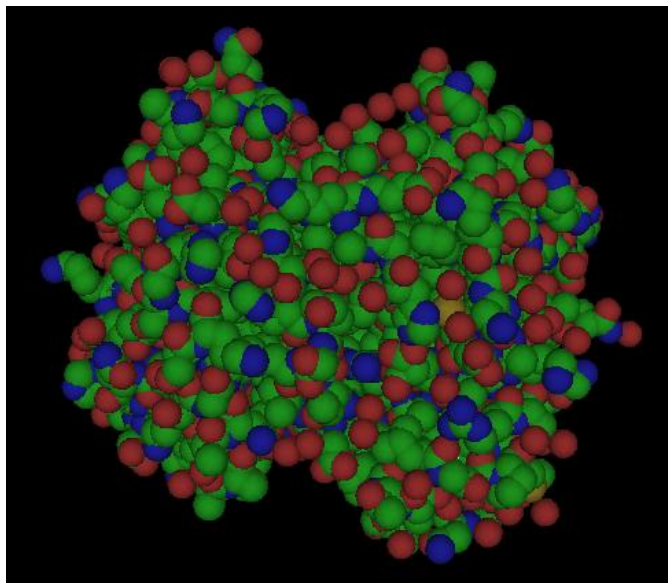


Quadbuffer stereo



☞ 光线追踪

- 光线追踪能制作出最高质量的分子图像。PyMOL是第一个拥有高速光线追踪器的全功能分子图像程序。
- 通过`ray`命令或点击“Ray”按钮，可以光线追踪PyMOL内的任意图像。内置的光线追踪器也使组配高质量的动画成为可能。



👉 探测静电力学

- PyMOL能够利用泊松波尔兹曼方程计算水溶液状态下的静电力学。

The image displays the PyMOL software interface. On the left, a menu lists various actions, with 'vacuum electrostatics' highlighted under the 'Generate:' section. On the right, a text box provides a warning about the quality of the 'Vacuum Electrostatics' calculation. The main window shows a 3D protein model with a color scale ranging from -79.978 (red) to 79.978 (blue). The command line on the right shows the selection of the protein and the calculation of the electrostatic potential map.

```
Actions:
zoom
orient
center
origin
drag
preset
find
align
generate
assign sec. struc.
rename object
duplicate object
delete object
hydrogens
remove waters
state
masking
sequence
movement
compute

Generate:
selection
symmetry mates
vacuum electrostatics

Vacuum Electrostatics:
protein contact potential (local)
NOTE: Due to short cutoffs, truncations, and
lack of solvent "screening", these computed
potentials are only qualitatively useful.
Please view with skepticism!

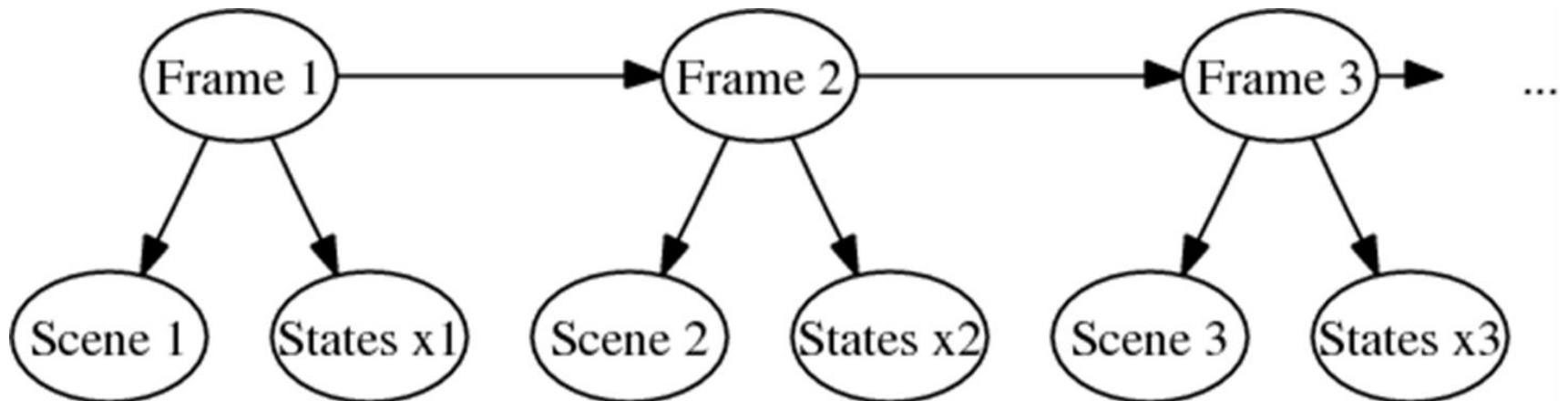
PyMOL Viewer
all A S H L C
2XUV A S H L C
HdeA A S H L C
(sele) A S H L C
HdeA_e_chg A S H L C
HdeA_e_map A S H L C
HdeA_e_pot A S H L C

Mouse Mode 3-Button Editing
Buttons L M R Wheel
& Keys Rota Move MovZ Slab
Shft Rot0 Mov0 MvDZ MovS
Ctrl MovA PkAt PkTB MvSZ
CtSh MvAZ Orig Clip MovZ
SngIClk PkAt Cent Menu
DbIClk MovA DrgM PkTB
Selecting Residues
Frame [ 1 / 1 ] 0/sec

PyMOL>
```

👉 动画制作

- PyMOL有强大的分子动画制作功能。
- 几个重要概念
 - **States (状态)**：状态指对象 (object) 某一个时间点特定的原子坐标。
 - **Scenes (场景)**：场景存储镜头 (camera) 的位置和定向、对象的活
动信息、原子的可见性 (visibility)、着色、表示形式和全局帧索引
(global frame index)
 - **Frames (帧)**：帧就像电影胶片中一个个单独的图片，在PyMOL中，
帧是由状态 (states) 而不是图片构成的，而且对帧可以进行相关操
作 (如camera的选转)。帧存储状态信息和场景信息。



➤ 重要命令

✓ Mset命令

- Mset命令用来指定那些状态作为动画的帧而被包括。
- Mset命令后紧跟定义整个动画的状态列表。每个状态采用以下形式之一：
 - 1 # 一个数字:指定下一个放映的状态
 - X # 一个数字紧随小写“x”（无空格）:指定状态总共该重复的次数
 - - # 一个数字紧随连字号（无空格）:指定状态按载入的顺序的放映。

□ 举例

- `mset 1 x30` # 创建一个由状态1放映30遍组成的30帧的动画
- `mset 1 -30` # 创建一个30帧的动画:从状态1到30,“-”是“到或至”的意思,但其前必须有空格.
- `mset 1 6 5 2 3` # 5帧:状态1, 6, 5, 2, 3放映
- 注: 当只有一个状态时, 状态1到状态x ($x \geq 1$) 只能显示状态1; 当n ($n \geq 2$) 个状态时,若设定的 $x > n$,那么不存在的状态不显示任何对象,而不是一直显示状态n

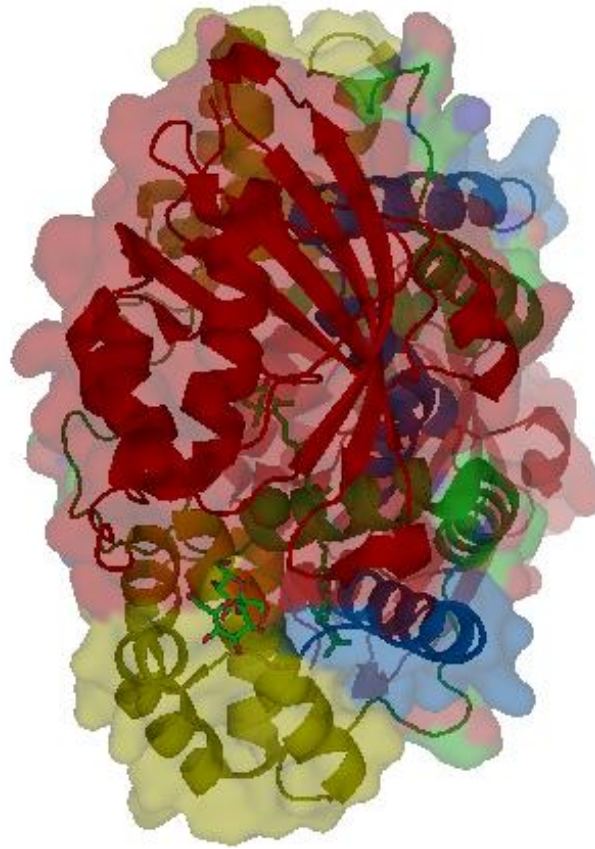
✓ Mdo命令

- Mdo命令可以把一系列的PyMOL命令捆绑到帧上。
- “util” 组件为产生mdo命令有两个脚本命令，“util.mrock” 和“util.mroll”。这些功能还没入档，但源程序可在modules/PyMOL/util.py找到。
- util.mrock start, finish, angle, phase, loop-flag
- util.mroll start, finish, loop-flag

□ 举例

- 下面命令创建了一个30帧的动画，此动画180度摇摆蛋白。
- Pymol> load test/dat/pept.pdb # 载入结构
- Pymol> mset 1 x30 # 定义动画
- Pymol> util.mrock 1,30,180,1,1 # mdo命令创建摇摆+/-180度的30帧动画
- Pymol> mplay

Example



➤ Ray-traced动画

- PyMOL能够在RAM中缓存一系列图片，然后以比它们渲染时高很多的速度放映。

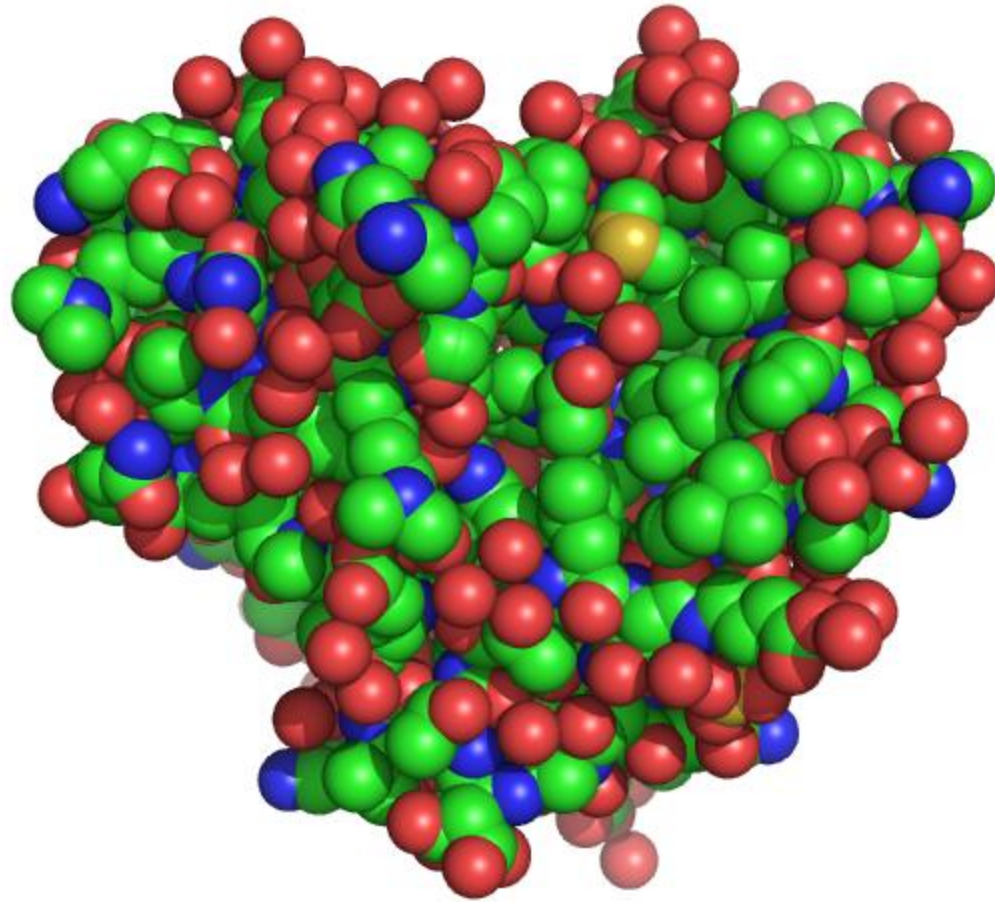
✓ Cache_frames

- Cache_frames控制PyMOL是否把帧保存到内存。注：缓存的图片占很大的内存空间，所以在使用此功能前先用“viewport”命令缩小窗口。

□ 举例

- Pymol> viewport 320,240
- Pymol> orient
- Pymol> show sph
- Pymol> mset 1 x30
- Pymol> util.mrock 1,30,180,1,1
- Pymol> set ray_trace_frames,1
- Pymol> set cache_frames,1
- Pymol> mplay

Ray-traced动画

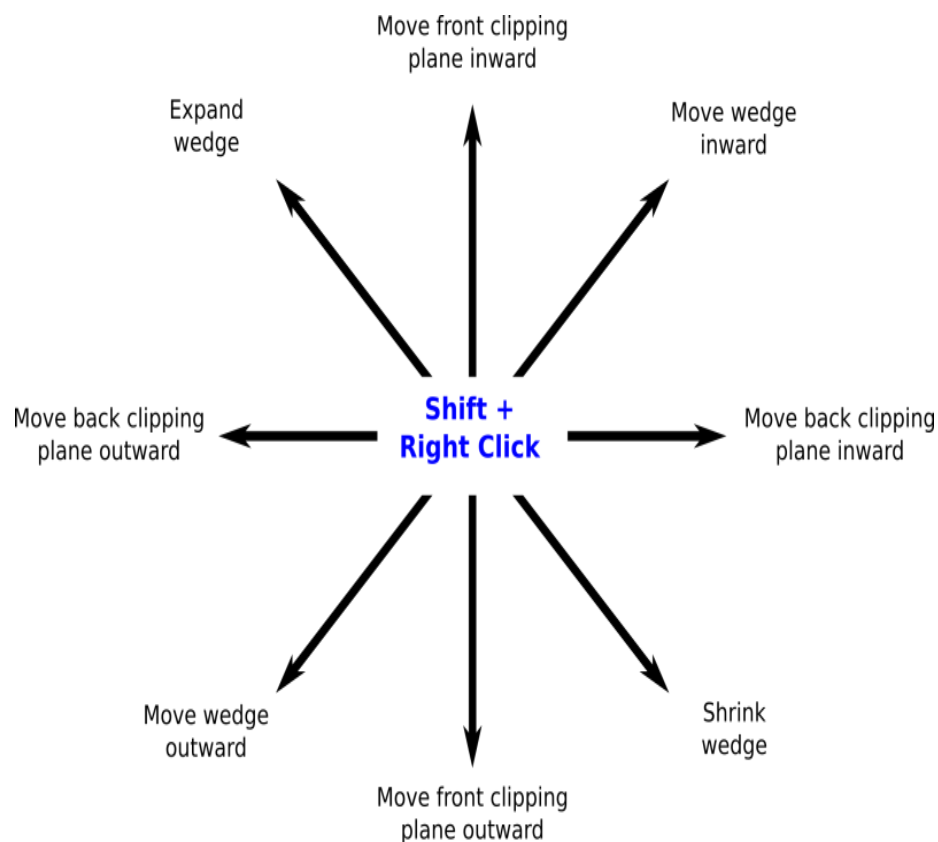
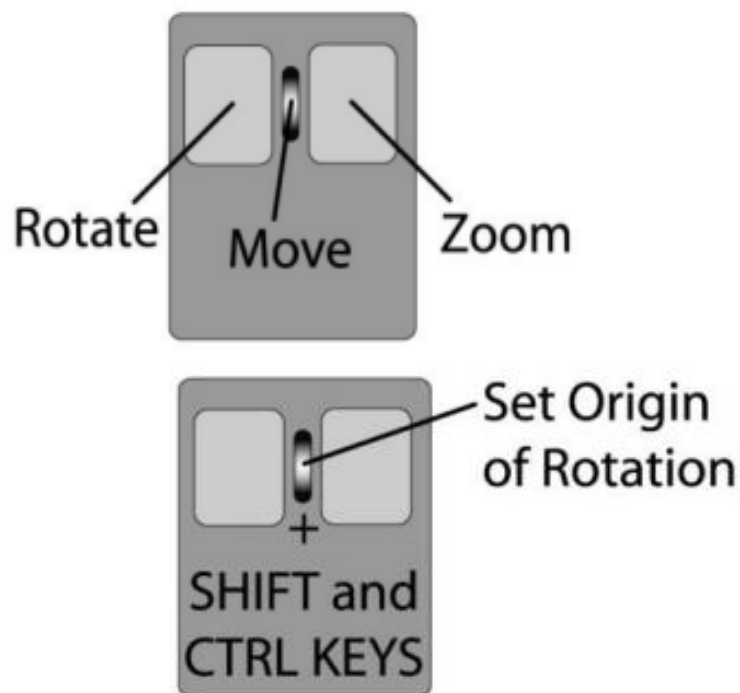


- ✓ 清除缓存
 - 一旦把一系列帧载入RAM，这些帧会一直存在,即使操纵这个模型。通过“mclear”命令或mclear按钮可清除缓存:
 - Mclear #清除帧的缓存
- 保存动画
 - ✓ 通过“mpng”命令或“File”菜单可保存动画
 - Mpng mov #将自动创建mov0001.png mov0002.png.....
 - ✓ 如果想每帧都被光线追踪，应打开光线追踪，关闭并清除缓存:
 - Pymol> set ray_trace_frames,1
 - Pymol> set cache_frames,0
 - Pymol> Mpng mov
 - Pymol> mclear

PyMOL基本操作

- Pymol的基本操作，包括窗口菜单、加载文件、图像处理等等。可以用鼠标操作，也可以用命令操作。

➤ 鼠标操作



➤ 命令操作

- Pymol是区分大小写的，不过目前为止Pymol还是只用小写。
- Pymol的命令都是由关键词（keyword）加上一些变量（argument）组成，格式如下：
- Pymol> keyword *argument*
- 其中关键词（keyword）如load、zoom、color、set等等，是必须的，；而变量则不是必须的，比如退出命令quit就不需要附加变量：
- Pymol> quit
- 通常情况下需要加变量，当不加任何变量时，Pymol会默认一个变量all
- Pymol命令中需要避免的符号：
- ! @ # \$ % ^ & * () ' " [] { } \ | ~ ` < > ? /

☞ 对象选择

- Pymol> load *name.pdb, name* # 载入pdb文件，并命名
- Pymol> **fetch** object # 直接从网上下载，不用加后缀
- ✓ 如果打开了多个PDB文件，想暂时关闭/打开某个对象，可以这样：
 - Pymol> disable *object-name*
 - Pymol> enable *object-name*
- ✓ 删除选定的目标或者整个对象：
 - Pymol> delete *selection-name*
 - Pymol> delete *object-name*

☞ 对象显示

- Pymol> show *representation* #以不同方式显示蛋白质结构
- Pymol> hide *representation*
- Pymol>as *representation* #不论原来有多少种表示形式，只显示一种
- 其中*representation*可以为：cartoon, ribbon, dots, spheres, surface和mesh。

👉 保存对象

1、保存文档文件

- `Pymol> log_open script-file-name .pml` #记录一个文本文档，该文件的后缀名应为.pml
- `Pymol> log_close` # 终止记录
- `Pymol> @script-file-name` # 调用该文档

2、保存会话文件

- 外部GUI窗口里面的File — Save Session，创建一个会话文件（.pse），下次打开Pymol时直接回到当前所在的状态。
- 两者区别：文档文件可以编辑，但会话文件不可以；记录文档文件前必须先运行log_open命令，而会话文件可以随时创建；最后文档文件以文档形式运行（@），而打开会话文件则必须选择外部GUI窗口中的File — Open。

3、保存图片

- `Pymol> ray` # 优化图像，使图像具有三维的反射及阴影特效
- `Pymol> png file-name` # 图片被保存在PyMOL安装默认的文件夹中

👉 关于视点

- Pymol中的视点可以通过`zoom`、`orient`、`view`这三个命令来改变。
 - ❑ **Zoom**（变焦）命令可使对象或选择在视野**中央显示**：如果对象或选择没显示在当前的视野，命令会使它显示；如果当前视野仅显示了一小部分，命令会使它充满视野。
- Pymol> `zoom selection-expression`
 - ❑ **Orient** 命令会调整对象或选择，使其**最大维度水平显示**，**次最大维度垂直显示**，方便重新查看分子：
- Pymol> `orient selection-expression`
 - ❑ **View**可用来**保存或调用**视角
- Pymol> `view key, action`
- 其中“key”是你随便给当前视角定的名字，“action”可以为：`store`或者`recall`。如果不加任何“action”，则默认为`recall`：
- Pymol> `view v1, store` # 当前视野被命名为v1并保存
Pymol> `view v1, recall` # 调用保存的v1定向
Pymol> `view v1` # `recall`是默认的view语句，所以此命令行也是调用

👉 关于选择

- Pymol> select *selection-name*, *selection-expression*
- *selection-name*即给选择表达起个名字，这个名字可以由字母[A/a—Z/z]，数字[0—9]已经下划线[_]组成，但是要避免使用：
!@#\$%^&*()' "[] { } \ | ~ ` < > ? /
- *selection-expression*表示一些被选中的部分，它们可以是一些个原子，一些个Helix，一些个Beta sheet，或者它们的混合物。
- *selection-expression=selector identifier*，其中"selector"定义了某类属性，而"identifier"则在该类属性下需要被选择的部分。
- eg. Pymol> select test, name c+o+n+ca

常用selector

| Selector | 简写 | Identifier及例子 |
|----------|-----|--|
| symbol | e. | chemical-symbol-list 周期表中的元素符号 Pymol> select polar, symbol o+n |
| name | n. | atom-name-list pdb文件中的原子名字 Pymol> select carbons, name ca+cb+cg+cd |
| resn | r. | residue-name-list 氨基酸的名字 Pymol> select aas, resn asp+glu+asn+gln |
| resi | i. | residue-identifier-list pdb文件中基团的编号 Pymol> select mults10, resi 1+10+100 residue-identifier-range Pymol> select nterm, resi 1-10 |
| alt | alt | alternate-conformation-identifier-list 一些单字母的列表，选择具有2种构型的氨基酸 Pymol> select altconf, alt a+b |
| chain | c. | chain-identifier-list 一些单字母或数字的列表 Pymol> select firstch, chain a |

| Selector | 简写 | Identifier及例子 |
|--------------|------|--|
| segi | s. | segment-identifier-list 一些字母（最多 4 位）的列表 Pymol> select ligand, segi lig |
| flag | f. | flag-nummer 一个整数（0 — 3 1） Pymol> select f1, flag 0 |
| numeric_type | nt. | type-nummer 一个整数 Pymol> select type1, nt. 5 |
| text_type | tt. | type-string 一些字母（最多 4 位）的列表 Pymol> select subset, tt. HA+HC |
| id | id | external-index-number 一个整数 Pymol> select idno, id 23 |
| index | idx. | internal-index-number 一个整数 Pymol> select intid, index 23 |
| ss | ss | secondary-structure-type 代表该类结构的单字母 Pymol> select allstrs, ss h+s+l+"" |

有关比较的selector

| Selector | 简写 | Identifier及例子 |
|-----------------------|-----|--|
| b | b | comparison-operator b-factor-value 一个实数，用来比较b-factor Pymol> select fuzzy, b > 12 |
| q | q | comparison-operator occupancy-value 一个实数，用来比较occupancy Pymol> select lowcharges, q > 0.5 |
| formal_charge | fc. | comparison-operator formal charge-value 一个整数，用来比较formal charge Pymol> select doubles, fc. = -1 |
| partial_charge | pc. | comparison-operator partial charge-value 一个实数，用来比较partial charge Pymol> select hicharges, pc. > -1 |

不需要加identifier的selector

| Selector | 简写 | 描述 |
|----------|------|-------------------------|
| all | * | 所有当前被Pymol加载的原子 |
| none | none | 什么也不选 |
| hydro | h. | 所有当前被Pymol加载的氢原子 |
| hetatm | het | 所有从蛋白质数据库HETATM记录中加载的原子 |
| visible | v. | 所有在被“可见”的显示的对象中的原子 |
| present | pr. | 所有的具有定义坐标的原子 |

- 在Identifier中用到的原子以及氨基酸的命名规则可以在下面的网址中找到：<http://www.wwpdb.org/docs.html>

配合逻辑操作子的selector

| Operator | 简写 | 效果与例子 |
|------------|----------|---|
| not s1 | ! s1 | 选择原子但不包括s1中的 Pymol> select sidechains, ! bb |
| s1 and s2 | s1 & s2 | 选择既在s1又在s2中的原子 Pymol> select far_bb, bb & farfm_ten |
| s1 or s2 | s1 s2 | 选择s1或者s2中的原子（也就是包含全部的s1和s2原子） Pymol> select all_prot, bb sidechain |
| s1 in s2 | s1 in s2 | 选择s1中的那些原子，其identifiers (name, resi, resn, chain, segi) 全部符合s2中对应的原子 Pymol> select same_atom, pept in prot |
| s1 like s2 | s1 l. s2 | 选择s1中的那些原子，其identifiers (name, resi)符合s2中对应的原子 Pymol> select similar_atom, pept like prot |

| Operator | 简写 | 效果与例子 |
|--------------------------|---------------|--|
| s1 gap X | s1 gap X | 选择那些原子，其van der Waals半径至少和s1的van der Waals半径相差X Pymol> select farfrm_ten, resi 10 gap 5 |
| s1 around X | s1 a. X | 选择以s1中任何原子为中心，X为半径，所包括的所有原子 Pymol> select near_ten, resi 10 around 5 |
| s1 expand X | s1 e. X | 选择以s1中任何原子为中心，X为半径，然后把s1扩展至该新的范围所包含的所有原子 Pymol> select near_ten_x, near10 expand 3 |
| s1 within X of s2 | s1 w. X of s2 | 选择以s2为中心，X为半径，并包含在s1中的原子 Pymol> select bbnearten, bb w. 4 of resi 10 |
| byres s1 | br. s1 | 把选择扩展到全部residue Pymol> select complete_res, br. bbnear10 |
| byobject s1 | bo. s1 | 把选择扩展到全部object Pymol> select near_obj, bo. near_res |
| neighbor s1 | nbr. s1 | 选择直接和s1相连的原子 Pymol> select vicinos, nbr. resi 10 |

另外，利用括号可以多重逻辑选择，如选择chain b，但不选择其中的residue 88:

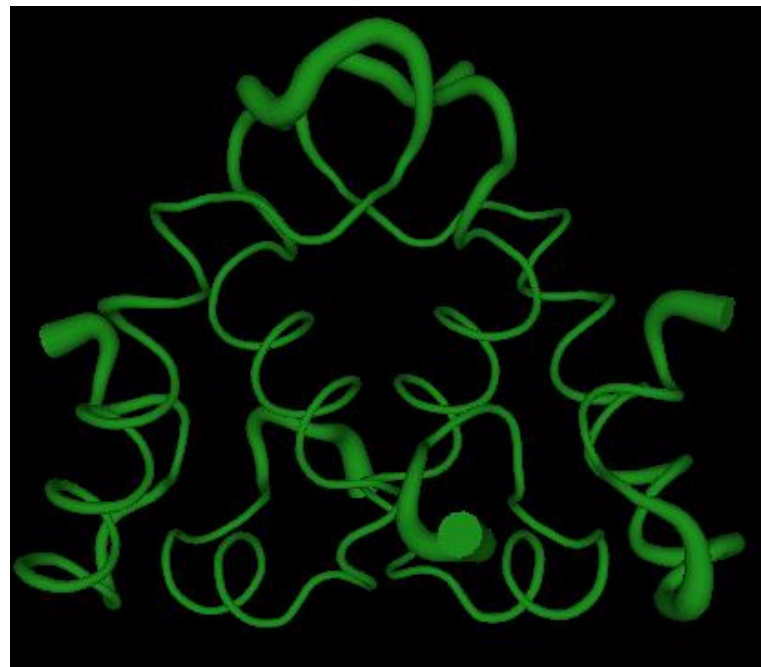
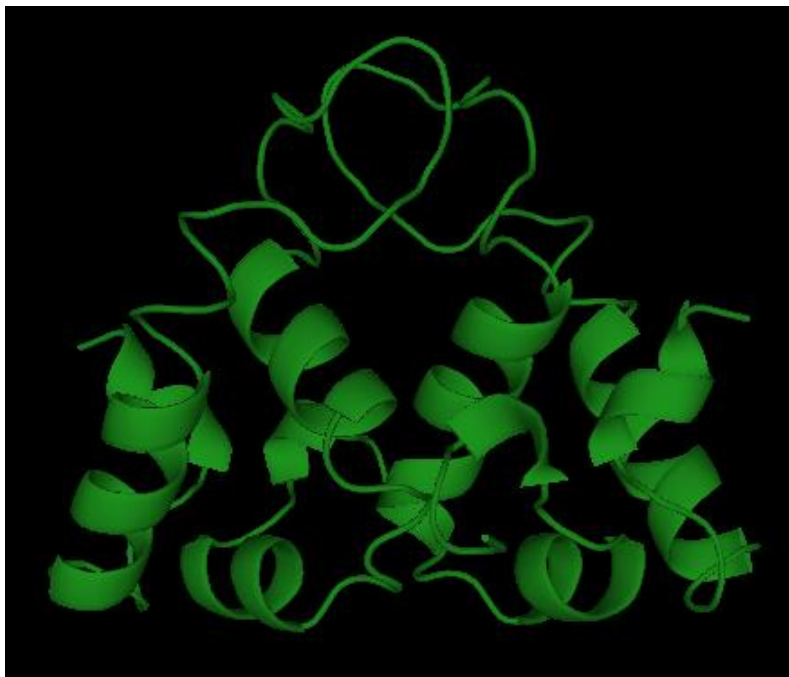
Pymol> select chain b and (not resi 88)

注：当有多个括号时，Pymol优先处理最里层括号里面的内容。

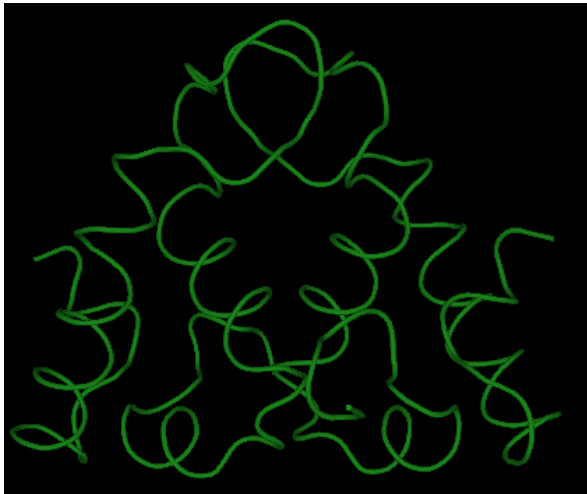
👉 关于cartoon

- cartoon的命令格式如下：
 - Pymol> cartoon *type*, (*selection*)
- cartoon的显示类型：
 - Automatic: 默认的显示方式

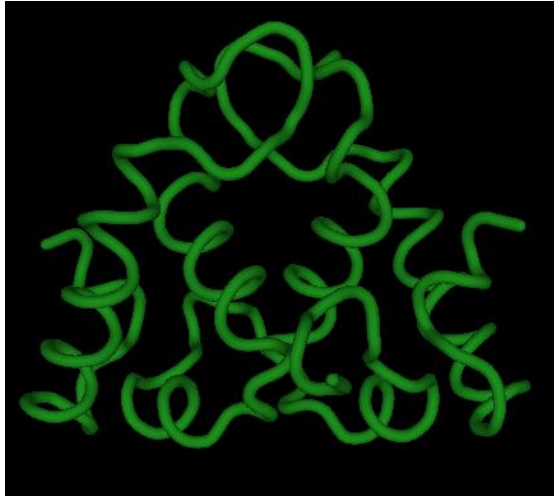
Putty: 按R-factor显示, 值越大越粗



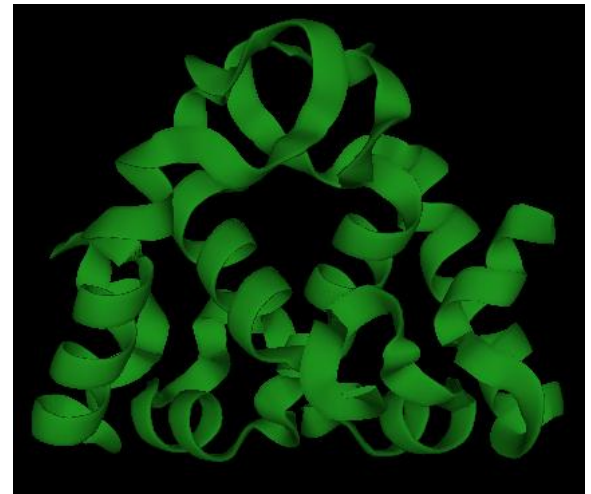
Loop



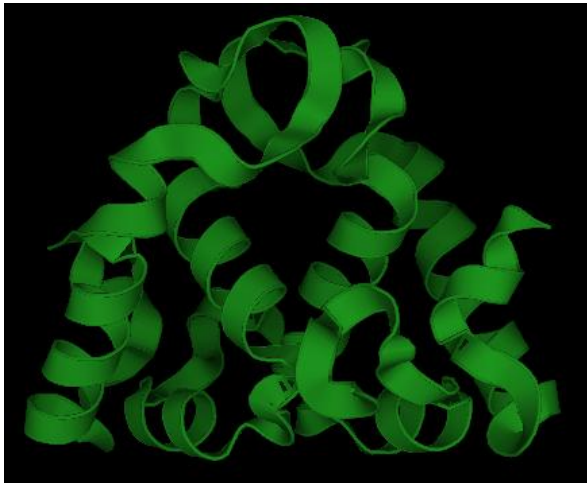
Tube



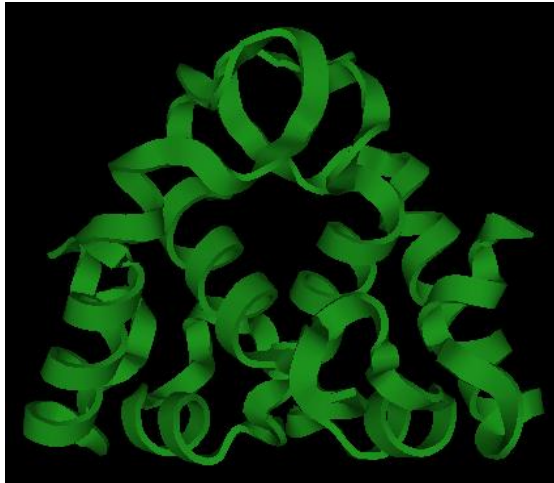
Oval



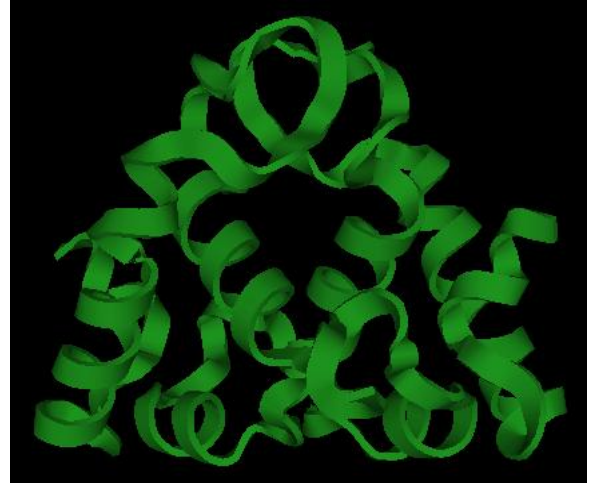
Dumbbell



Rectangle



Arrow



➤ 细节设置

✓ Sheet显示

- Pymol> set cartoon_flat_sheets, 1 # 1开启
- Pymol> set cartoon_flat_sheets, 0 # 0关闭

✓ Loop显示

- Pymol> set cartoon_smooth_loops, 1
- Pymol> set cartoon_smooth_loops, 0

✓ Helix的厚度和宽度

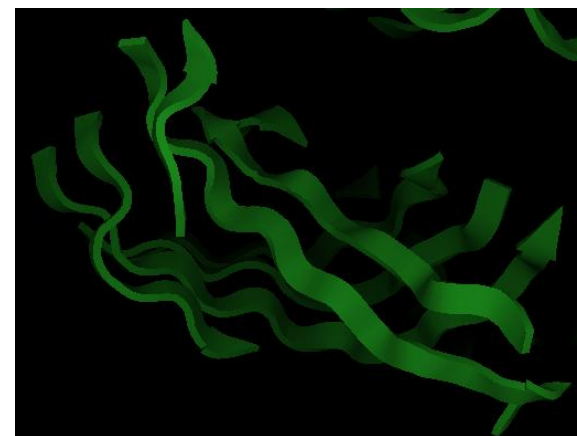
- Pymol> set cartoon_oval_width, 0.2
- Pymol> set cartoon_oval_length, 1.5

✓ sheet的厚度和宽度

- Pymol> set cartoon_rect_width, 0.5
- Pymol> set cartoon_rect_length, 1.5

✓ loop的半径

- Pymol> set cartoon_loop_radius, 0.2



✓ **cartoon**显示风格为**fancy**

- Pymol> set cartoon_fancy_helices, 1 # helix的边上会带有一个很细的cylinder

Pymol> set cartoon_fancy_sheets, 1

- 此时设置helix的厚度，宽度，以及这个cylinder的半径分别是：

- Pymol> set cartoon_dumbbell_width, 0.1

Pymol> set cartoon_dumbbell_length, 2

Pymol> set cartoon_dumbbell_radius, 0.2

✓ **上色**

- Pymol> set cartoon_color, green

✓ **Refine**

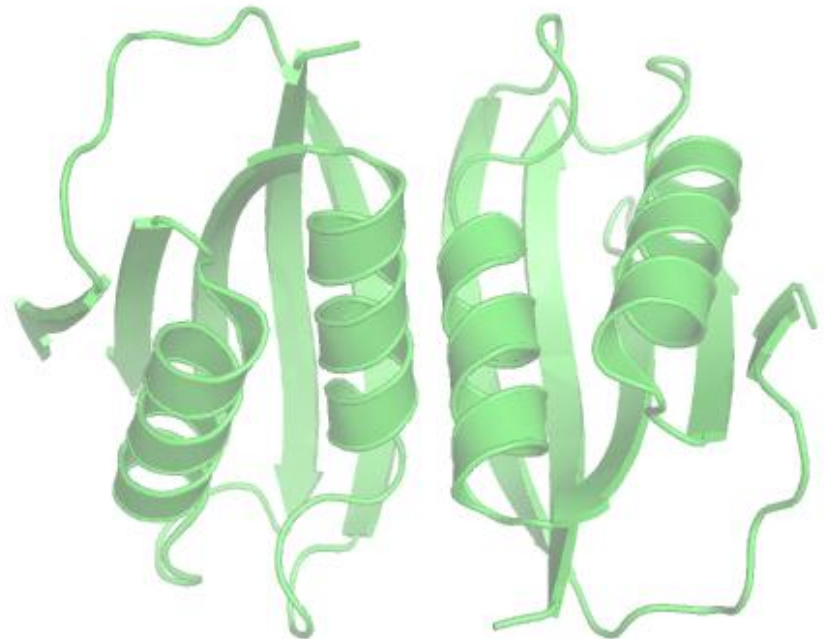
- Pymol> set cartoon_refine, 20

- # 数字越大越漂亮，可选范围为1~20

✓ **设置透明**

- Pymol> set cartoon_transparency, 0.5

- # 可选范围为0~1



👉 关于label

➤ label的命令格式如下：

- Pymol> label selection, expression
- Selection为已命名的对象，expression为标签的内容，如name, resn, resi, chain等等。

➤ Label的一些设置

✓ 投影模式

- Pymol> set label_shadow_mode, 3
- 可选值
- 0: 无投影，
- 1: object有投影到label上，但是label本身无投影，
- 2: object有投影到label上，label也有投影，
- 3: object不投影到label上，label本身有投影

✓ 文字颜色

- Pymol> set label_color, color-name, selection

✓ 字体

- Pymol> set label_font_id, 5 # pymol内置了12种字体, 编号为5—16

✓ 字体大小

- Pymol> set label_size, -0.5 #正值单位为px, 负值单位为Å
Pymol> set label_size, 4

✓ label位置

- Pymol> set label_position, (x,y,z)

PyMOL应用实例

👉 Cartoon及表面显示

1. Load the PDB file

- File -> Open -> 1w2i.pdb

2. Hide everything and then show protein cartoon

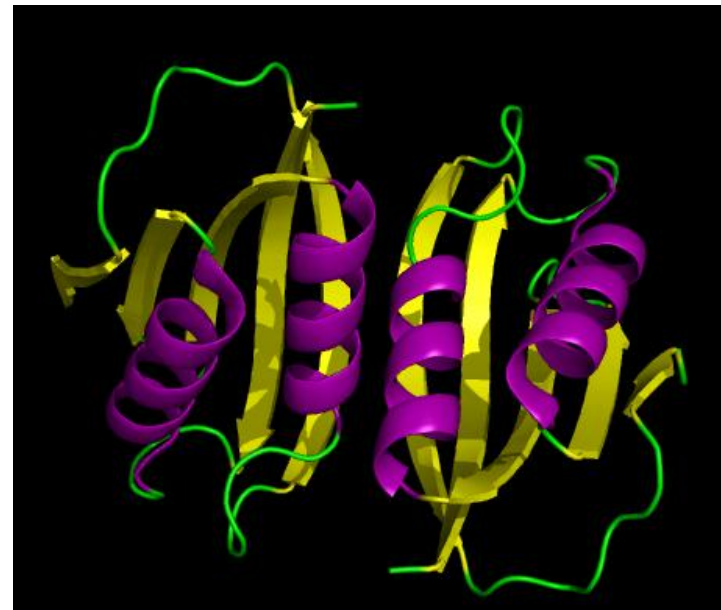
- PyMOL> hide everything, all
- PyMOL> show cartoon, all

3. Color the helix, sheet, and loop

- PyMOL> color purple, ss h
- PyMOL> color yellow, ss s
- PyMOL> color green, ss ""

4. Color chain A and B

- PyMOL> color red, chain A
- PyMOL> color blue, chain B



5. Create a surface display for chain A

- PyMOL> create obj_a, chain A
- PyMOL> show surface, obj_a

6. Color the active site residue

- PyMOL> select active, (resi 14-20,38 and chain A)
- PyMOL> color yellow, active
- PyMOL> turn y, -60; turn x, -20
- PyMOL> zoom active

Note: Rotate the molecule to see a hole around the yellow surface. That's the active site cradle for binding phosphate

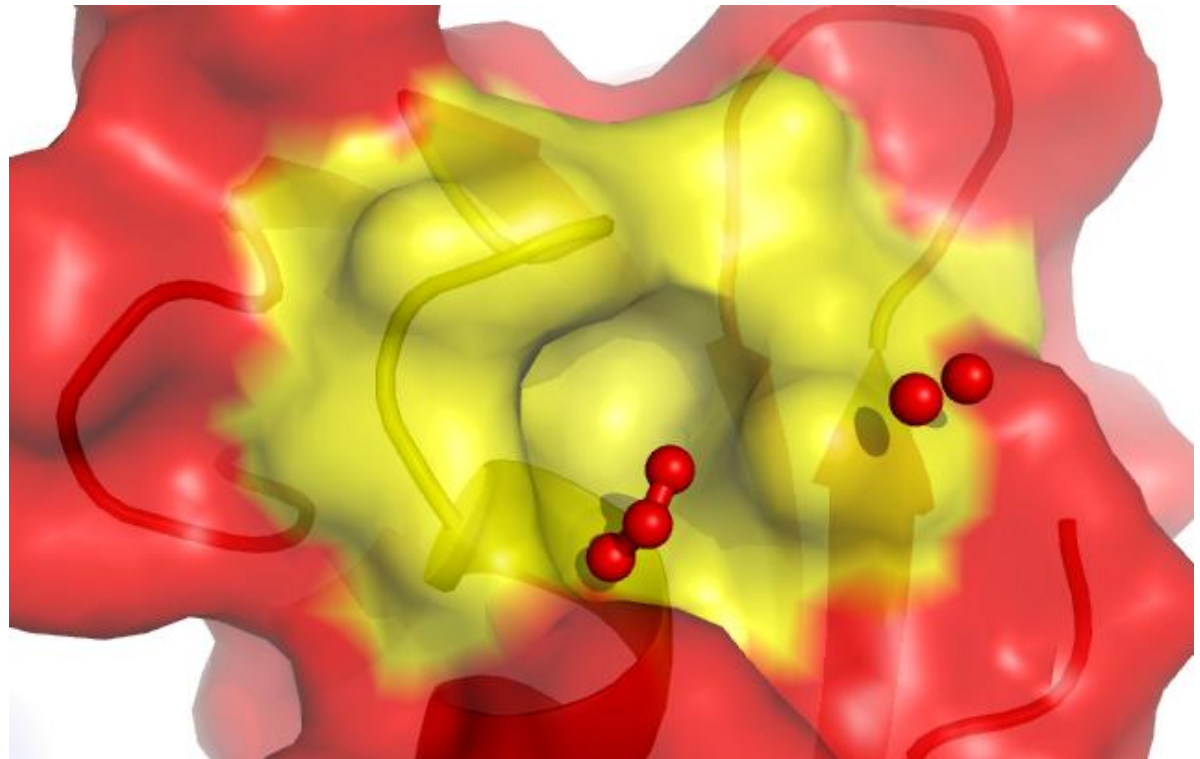
7. Locate and display the bound formate ion in the active site.

- PyMOL> select ligand, active around 3.5 and resn FMT
- PyMOL> show sticks, ligand
- PyMOL> show spheres, ligand

- PyMOL> alter ligand, vdw=0.5
- PyMOL> rebuild
- PyMOL> set transparency=0.25

8. Rendering and output

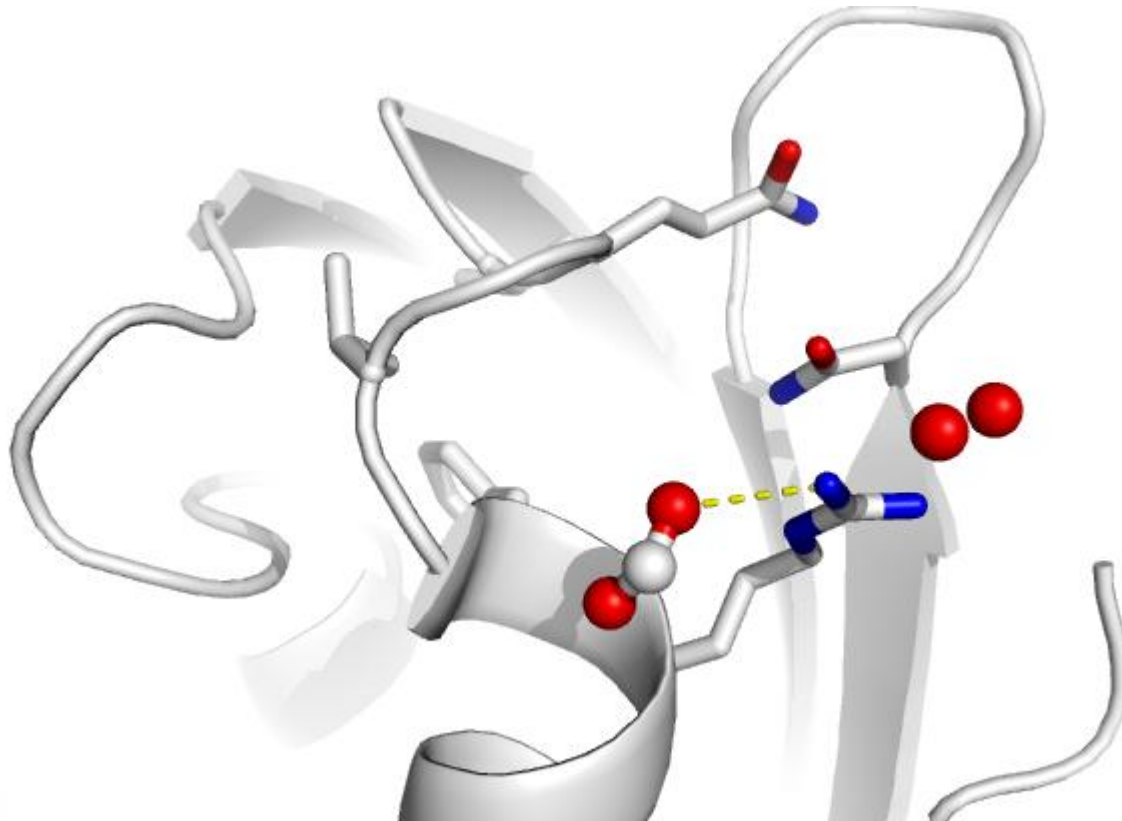
- PyMOL> bg_color white
- PyMOL> ray
- File -> Save Image



☞ 活性位点侧链显示及距离的测量

- Display the side-chain of active site residues on top of the cartoon representation
 - PyMOL> hide surface
 - PyMOL> select sidechain, not (name c+n+o)
 - PyMOL> show sticks, (active and sidechain)
 - PyMOL> color blue, name n*
 - PyMOL> color red, name o*
 - PyMOL> color white, name c*
- Display and measure distances
 - Wizard -> Measurement -> Distance
 - Use this to measure the distance between the arginine N atoms and the oxygen atoms of formate ion.

- PyMOL> distance resi 20 and name NH2 and chain A, resi 1092 and name O2 and chain A
- PyMOL> hide labels
- PyMOL> ray
- File -> Save Image



☞ 电子密度图

1. Loading PDB file

- File -> Open -> 1w2i.pdb

2. Load the map file

- File -> Open -> 1w2i.map.xplor
- It takes a while to load the map file.

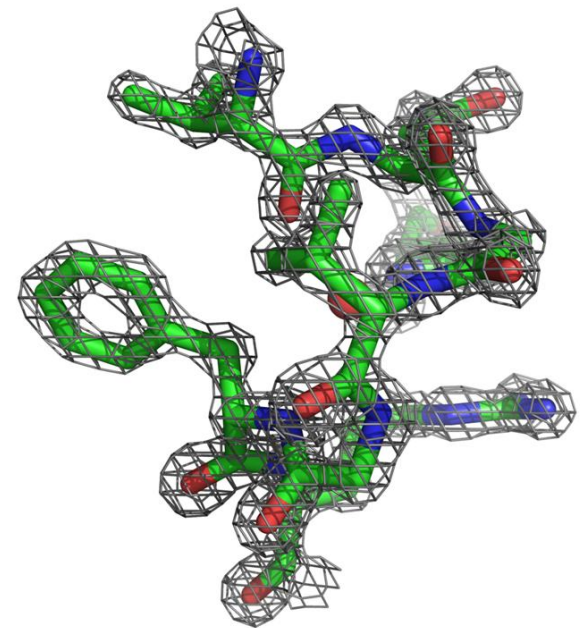
3. Zoom in the active site

- PyMOL> select active, (resi 14-20,38) and chain A
- PyMOL> zoom active
- PyMOL> hide all
- PyMOL> show stick, active

4. Locate and Display the active site water

- PyMOL> select active_water, ((resi 38 and name ND2 and chain A) around 3.5) and (resn HOH)

- PyMOL> show spheres, active_water
 - PyMOL> alter active_water, vdw=0.5
 - PyMOL> rebuild
5. Display the electron density around the active site atoms at sigma level=1.0
- PyMOL> **isomesh** mesh1, 1w2i.map, 1.0, (resi 14-20,38 and chain A), carve=1.6
 - PyMOL> isomesh mesh1, 1w2i.map, 1.0, active, carve=1.6
6. Change the color of the map
- PyMOL> color grey, mesh1
7. Set the background color
- PyMOL> bg_color white
8. save the figures in PNG
- To render a figure with high resolution
 - PyMOL> ray 2400,2400
 - File -> Save Image



一. 含配体蛋白的表面显示

1. 先把蛋白显示成cartoon模型，按链的不同给蛋白着色，A链红色，B链蓝色
as cartoon;color red, chain A;color blue, chain B

2. 将A链的表面显示出来

```
create obj_a, chain A
```

```
show surface, obj_a
```

3. 选择配体（甲酸根离子）并将配体显示成球形

```
select ligand, resn FMT;color cyan, ligand; show sticks, ligand
```

```
show spheres, ligand
```

4. 改变球形直径大小，并将表面透明度设成0.25

```
alter ligand, vdw=0.5
```

```
rebuild
```

```
set transparency=0.25
```

5. 选择活性位点，并将活性位点的颜色显示成黄色，

```
select active, ligand around 6.5 and chain A;color yellow, active
```

```
zoom active
```

```
PyMOL> ray
```

```
File -> Save Image
```

动画制作

定义动画

```
mset 1 x30
```

mdo命令创建摇摆+/-180度的30帧动画

```
util.mrock start, finish, angle, phase, loop-flag
```

```
util.mrock 1,30,180,1,1
```

```
mplay
```

二. 活性位点侧链显示及距离的测量

Display the side-chain of active site residues on top of the cartoon representation

```
PyMOL> hide surface
```

```
PyMOL> select sidechain, not (name c+n+o)
```

```
PyMOL> show sticks, (active and sidechain)
```

```
PyMOL> color blue, name n*
```

```
PyMOL> color red, name o*
```

```
PyMOL> color white, name c*
```

Display and measure distances

Wizard -> Measurement -> Distance

Use this to measure the distance between the arginine N atoms and the oxygen atoms of formate ion.

```
PyMOL> distance resi 20 and name NH2 and chain A, resi 1092 and name O2 and chain A
```

```
PyMOL> hide labels
```

```
PyMOL> ray
```

File -> Save Image

三. 电子密度图

```
PyMOL> show spheres, active_water
```

```
PyMOL> alter active_water, vdw=0.5
```

```
PyMOL> rebuild
```

5. Display the electron density around the active site atoms at sigma level=1.0

```
PyMOL> isomesh mesh1, 1w2i.map, 1.0, (resi 14-20,38 and chain A), carve=1.6
```

```
PyMOL> isomesh mesh1, 1w2i.map, 1.0, active, carve=1.6
```

6. Change the color of the map

```
PyMOL> color grey, mesh1
```

7. Set the background color

```
PyMOL> bg_color white
```

8. save the figures in PNG

To render a figure with high resolution

```
PyMOL> ray 2400,2400
```

File -> Save Image



Thank you!