

# PyMol

как среда визуализации и редактирования 3D структур

Головин А.В.<sup>1</sup>

<sup>1</sup>МГУ им М.В. Ломоносова, Факультет Биоинженерии и Биоинформатики

Москва, 2013

# Содержание:

Введение

Визуализация с PyMol

Selections

Анимация

Моделирование и редактирование в PyMol

Скриптование в PyMol



# Визуализация с PyMol



# Для чего нужен PyMol

- Визуализация pdb и прочих файлов с координатами атомов
- Изготовление высококачественных изображений
- Начальное редактирование структур



# Системные требования

**Компьютер:** чем мощнее процессор и чем больше памяти, тем лучше

**3D монитор** не обязателен, но поддерживается

**Операционная система:** любая, под Linux проще установить, и он лучше работает с памятью.



# Как установить?

- Компиляция из исходников: <http://pymol.svn.sourceforge.net/>
- Установка бинарных пакетов в Ubuntu Linux: `sudo apt-get install pymol`
- Установка бинарных пакетов в Windows:
  - Ресурс для установки с python:  
<http://www.lfd.uci.edu/~gohlke/pythonlibs/#pymol>
  - Компиляция под Windows:  
<http://arcib.dowling.edu/~darakevn/installerpaper.pdf>



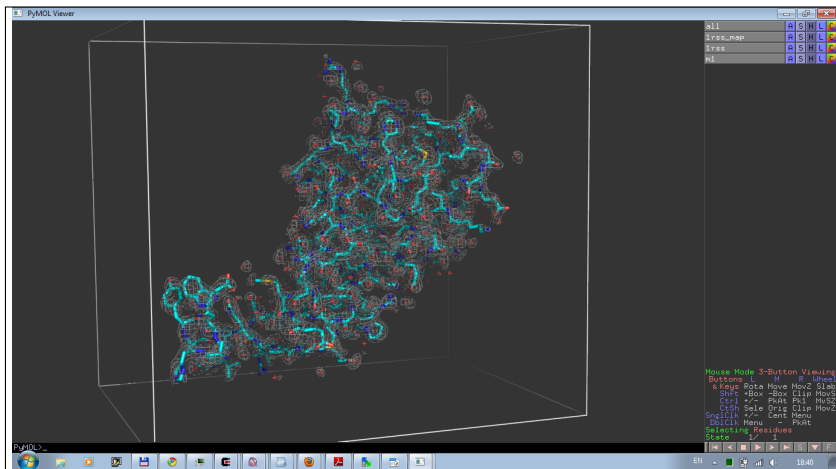
# PyMol - это GPL программа?

Да, PyMol это GPL-программа;

- исходный код доступен на [sourceforge.net](http://sourceforge.net)
- Бинарные пакеты для windows стоят денег и продаются:  
<http://pymol.org/academic.html>
- Бинарные пакеты для Linux собираются майтенерами

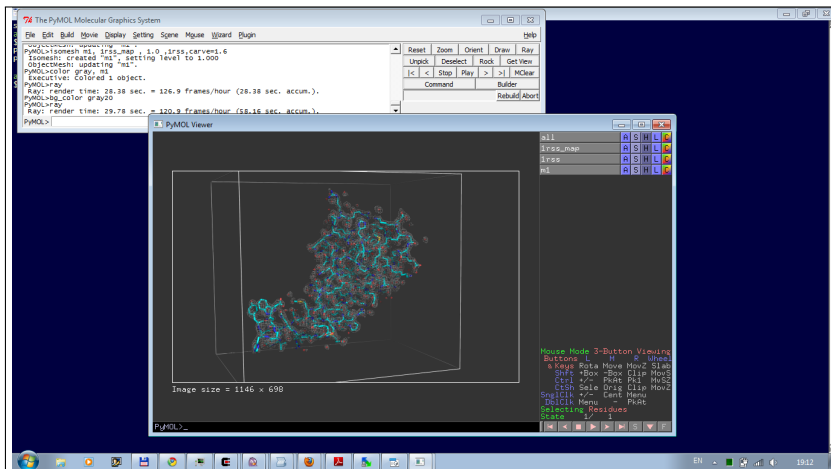


## PyMol



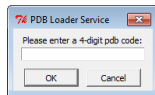


# Основной вид



# Как загрузить структуру?

- Из интернет:
  - в меню выбрать соответствующий plugin
  - или в командной строке: `fetch 1xxx`
- Локальный файл:
  - File->Open



# Использование мыши

- Левый клик + движение = вращение молекулы
- Средний клик + движение = перемещение молекулы
- Правый клик + движение верх/вниз = приближение/удаление молекулы
- Колесо = изменение уровня обрезания молекулы
- **Все манипуляции относятся к камере, а не координатам структуры**



# Меню объекта/выборки

The screenshot shows the PyMOL Viewer interface. A 3D molecular model is displayed in the center. A context menu is open over the model, listing the following objects and their corresponding button sets:

all	A	S	H	L	C
1rss_map	A	S	H	L	C
1rss	A	S	H	L	C
m1	A	S	H	L	C

Below the screenshot, the text **A,S,H,L,C** is displayed in red.

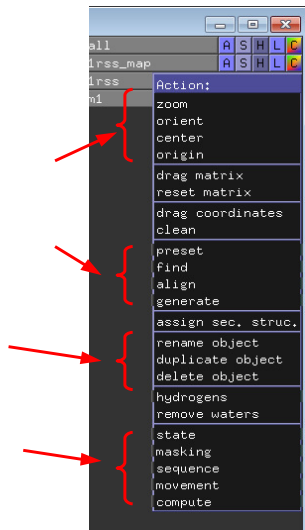
## A=Action

Манипуляции с ориентацией

Предустановки изображения и т.д.

Манипуляция с объектом

Прочее



## S=Show, H=Hide

The image displays a PyMol interface with eight different visualization styles for a protein structure, arranged in a 2x4 grid. The top row shows: 'lines' (a pink skeletal structure), 'sticks' (a yellow and blue stick model), 'spheres' (a cyan and red space-filling model), and 'surface' (a red and blue surface representation). The bottom row shows: 'mesh' (a blue and red wireframe mesh), 'dots' (a grey and red dot-based model), 'ribbon' (a pink ribbon structure), and 'cartoon' (a yellow and white cartoon representation). To the right of the grid is a command menu with a scrollable list of options and a keyboard shortcut column. The options include 'all', 'irss\_map', 'irss', 'm1', 'irss\_e\_chg', 'irss\_e\_map', 'irss\_e\_pot', '(bk)', '(sele)', '(bet)', 'nonbonded', 'dots', 'spheres', 'nb\_spheres', 'mesh', 'surface', 'organic', 'main chain', 'side chain', and 'disulfides'. The keyboard shortcuts are 'A S H L P' for 'all', 'A S H L P' for 'irss\_map', 'Show:' for 'irss', 'as' for 'm1', 'lines' for 'irss\_e\_chg', 'sticks' for 'irss\_e\_map', 'ribbon' for 'irss\_e\_pot', 'cartoon' for '(bk)', 'label' for '(sele)', and 'cell' for '(bet)'. Below the menu, mouse and keyboard controls are listed, including 'Mouse Mode 3-Button Viewing', 'Buttons L M R Wheel', and various key combinations for rotation, translation, and selection. At the bottom right of the interface, there are navigation buttons: left arrow, right arrow, up arrow, down arrow, and a 'S' button.

all	A S H L P
irss_map	A S H L P
irss	Show:
m1	as
irss_e_chg	lines
irss_e_map	sticks
irss_e_pot	ribbon
(bk)	cartoon
(sele)	label
(bet)	cell
nonbonded	
dots	
spheres	
nb_spheres	
mesh	
surface	
organic	
main chain	
side chain	
disulfides	

Mouse Mode 3-Button Viewing  
 Buttons L M R Wheel  
 & Keys Rota Move MovZ Slab  
 Shft +Box -Box Clip MovS  
 Ctrl +/- PkAt Pk1 MvSZ  
 CtrlSh Sele Orig Clip MovZ  
 SnglClk +/- Cent Menu  
 DblClk Menu - PkAt  
 Selecting Residues  
 State 1/ 1

## L=Label

PyMOL Viewer

all ASHL  
 lrss\_map ASHL  
 lrss Label:  
 ml clear  
 lrss\_e\_c residues  
 lrss\_e\_m chains  
 lrss\_e\_p segments  
 (bk)  
 atom name  
 (<sele) element symbol  
 (<bet) residue name  
 residue identifier  
 chain identifier  
 segment identifier  
 b-factor  
 occupancy  
 vdw radius

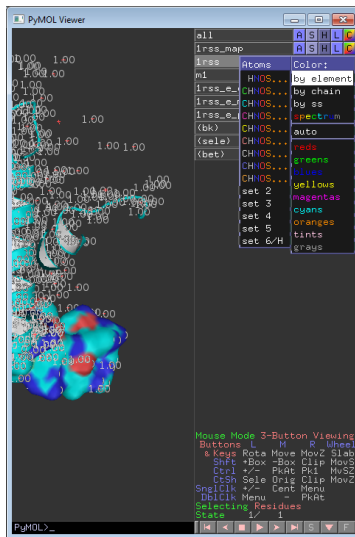
Other Properties:  
 formal charge other properties  
 partial charge (0,00) atom identifiers  
 partial charge (0,0000)  
 elec. radius  
 text type  
 numeric type  
 stereochemistry

Mouse Mode 3-Button Viewing  
 Buttons L M R Wheel  
 & Keys Rotate Move MovZ Slab  
 ShFt +Box -Box Clip MovS  
 Ctrl +/- PKAt PkI MuS2  
 Dsh Sele Driz Clip MovZ  
 SnglClk +/- Cent Menu  
 DblClk Menu - PKAt  
 Selecting Residues  
 State 1/ 1

PyMOL>\_



## C=Color





# Выборки

- Можно задать с помощью кликов мыши, удерживая SHIFT
- Удобнее писать выражения в командной строке

Например: *Select backbone, name sa+c+n*



# Операторы множеств

- Логические операторы AND, OR, NOT  
Операция OR может быть записана как ",",

Упражнение: Документ PDB содержит описание структуры, состоящей из белка, фрагмента ДНК и молекул воды. Что получится, если задать следующие команды ?

*select protein or dna*

*select protein and dna*

*select not water*

- Оператор WITHIN(...)  
*select all within 3.5 of resi 20*  
*select s1, (byres n. ca) within 3.5 of resn LIG*



# Help selections

## Длинное

## Короткое

name <atom names>	n. <atom names>
resn <residue names>	r. <residue names>
resi <residue identifiers>	i. <residue identifiers>
chain <chain ID>	c. <chain identifiers>
id <original-index>	
hydrogen	h.
all	*
visible	v.
hetatm	
byres <selection>	br. <selection>
byobj <selection>	bo. <selection>
around <distance>	a. <distance>
expand <distance>	e. <distance>
in <selection>	
like <selection>	l. <selection>
<selection> within <distance> of <selection>	
<selection> w. <distance> of <selection>	



# Примеры выборов

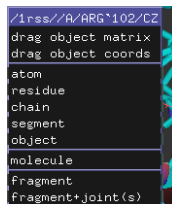
## sel=select

- sel s1, n. ca and c. A : все атомы CA в цепи A
- sel s2, n. ca and (c. A or c. B) : атомы CA цепей A и B
- sel s3, resn GLU and resi 100 : остаток 100 если он GLU
- sel s4, resi 100-120+130 : атомы остатков 100-120 и 130
- sel s5, byres( name CG) : атомы остатков где есть CG



# Иерархическое определение выборки

Легко увидеть иерархию правым  
кликом по атому



*sel s1, a/102/cz* : атом CZ в остатке 102

*sel s2, 100-120/N and c. A* : атомы N в остатках 100-120 цепи а

*sel s3, a/100+120/* : все атомы остатков 100 и 120 в цепи А

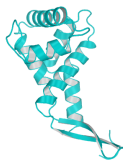


# Трассировка лучей, команда ray

Подробнее: <http://www.pymolwiki.org/index.php/Ray>



No ray



ray\_trace\_mode,0



ray\_trace\_mode,1



ray\_trace\_mode,2



ray\_trace\_mode,3



# Настройки изображения

<http://www.pymolwiki.org/index.php/Category:Settings>

- PyMol содержит порядка 600 настроек
- Не все документированы
- Большинство интуитивно понятны
- Настройки доступны через меню или в командной строке набрать:  
*set первые буквы имени опции и клавиша tab для достроения*



# Примеры

## #initial setup

`viewport 600, 600` --- размер графического окна

`set auto_zoom, off` --- не приближать новые объекты

`set auto_show_lines, off` --- не показывать линии автоматически

`set auto_show_selections, off` --- не показывать выборку автоматически

## #cartoon parameters

`set cartoon_fancy_helices, 1` --- изменение вида спиралей

`set cartoon_highlight_color, grey60` ---цвет внутренней стороны спиралей

`set cartoon_dumbbell_length, 1.0` ---ширина ленты в спирали

`set cartoon_rect_length, 1.40000` --- ширина ленты в бета

`set cartoon_loop_radius, 0.3` --- толщина неструкт. участка

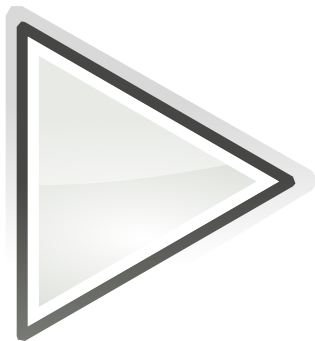
`set cartoon_smooth_loops=0` --- без сглаживания





# Анимация в PyMol

Если структура содержит более чем одну модель, то в PyMol можно анимировать движение молекулы переходом от одной модели к другой



# Анимация, основы

GUI :

Вращение вокруг объекта на N секунд:

- Movie->Program->Camera->X-Roll->N Seconds
- Movie->Program->Camera->Y-Roll->N Seconds

Покачивание:

- Movie->Program->Camera->X-Rock->X-Degrees over N-Seconds

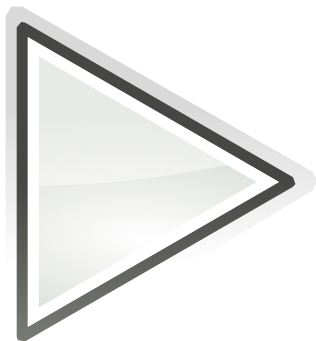


# Пример

- Action->Preset->Technical (viewer gui)
- Scene->Store->F1
- zoom i. 90 # увеличение остатка 90
- Scene->Store->F2
- Movie->Program->Scene Loop->Y-Rock->4 Seconds Each
- File-> Save movie



# Результат



# Анимация, терминология

- Объект и выборка : смотри выше
- states: конформация или набор координат
- scene: позиция камеры и отображение
- frames: это кадры в анимации, содержит state и scene

Movie panel:



# Анимация, команды

`mset 1 -55` : задать анимацию от 1 до 55 state на 55 кадров (frames)

`mset 1 x90` : задать анимацию первого state от 1 до 90 кадров

`mset 1 x30 1 -15 15 x30 15 -1` : первые 30 кадров state 1, следующие 15 кадров это состояния 1-15, следующие 30 кадров состояние 15, следующие 15 кадров состояния от 15 до 1



# Анимация, команды

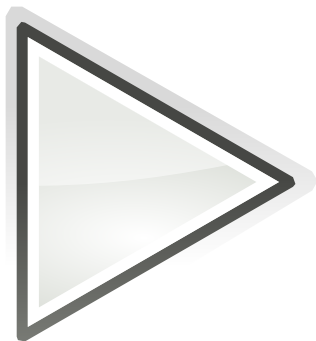
`mview` : команда для создания ключевых точек

Пример :

- `mset 1 x100`
- `frag leu #` создаём LEU
- `orient #` ориентируем его
- `mview store #` запоминаем ключевую точку
- `frame 100 #` переходим в кадр 100
- `zoom ID 10 #` увеличиваем атом №10
- `mview store #` запоминаем ключевую точку
- `mview reinterpolate #` делаем интерполяцию



# Результат mview





# Дополнительные команды

- `mmatrix` : устанавливает вид для первого кадра
- `util.mrock` : покачивание сцены на определённый угол
- `util.mrock(start, finish, angle, phase, loop-flag)`
- `util.mroll` : вращение вокруг оси Y
- `util.mroll(start, finish, loop-flag)`
- `mdo` : (устарело) запуск какой-либо команды в заданном кадре



# Сохранение анимации

## Старый путь:

`set ray_trace_frames,1`

`mpeg movmovie`

Нужны программы avidemux, Virtual Dub, mencoder для того, чтобы собрать ролик с нужным сжатием (кодек)

**Новый путь: File->Save movie** ; есть недостаток, старый офис понимает только avi с определённым кодеком



# Моделирование и редактирование в PyMol

- Можно перемещать объекты и сохранять их новые координаты
- Можно рассчитать вторичную структуру
- Можно менять координаты отдельных атомов
- Можно вносить мутации в белок (но не НК)
- Можно конвертировать L->D аминокислоты
- Можно добавлять протоны
- Можно выравнивать в пространстве молекулы
- Можно добавлять некоторые фрагменты из библиотеки и собственные



# Перемещение объектов

Рекомендуемый порядок действий:

- `set retain_order #` надо сохранить порядок атомов
- `create newobj, sele #` создаём новый объект, страховка
- `translate [0,10,0], newobj #` перемещаем
- `rotate x,90,newobj #` вращаем
- `save newfile.pdb, newobj`

Операции по перемещению и вращению можно делать мышкой в режиме editing



# Изменение координат отдельных атомов и объектов

`alter_state 1,(pdb1cse),x=x-10.0`  
Или `translate [0,10,0], A/100/NZ`



## Удаление связей, но не атомов

- Выберите первый атом, ctrl+middle click, выберите второй атом, ctrl+middle click
- И unbond или ctrl+D

**Внимание! Координаты атомов не меняются, только исчезает изображение связи**



# Мутация аминокислот

- Запустите wizard->mutagenesis
- Выберите аминокислоту для мутации
- Справа выберите, на что мутировать
- Выберите ротаметр с помощью управления movie
- Закончите процедуру с Apply



# Добавление протонов

Работает с молекулами, т.е. объектами

```
create gln, A/101/
```

```
h_add gln
```

Или через меню action объекта.

Есть вероятность, что протоны будут добавлены неверно, если PyMol неправильно угадал валентность тяжёлых атомов.





# Суперпозиция в пространстве

Задача достаточно нетривиальная, и есть разные пути:

Белки:

`align, super, fit`

Другое:

`pair_fit`

Желательно указывать родственные атомы в молекулах

`pair_fit ( trna10 and resid 10:15 and name P ), ( ref4 and resid 10:15 and name P )`



## Добавление органических фрагментов или а.к.

- С помощью `ctrl+middle click` выделите шариком атом, к которому будет присоединяться фрагмент
- В меню `Build` выберите нужный фрагмент
- С помощью `ctrl+left click` выберите торсионный угол  
Или
- Создайте свою молекулу (ChemSketch)
- Сохраните как `pkl` в `<pymol_path>/data/chempy/fragments`
- `editor.attach_fragment('pk1','my_fragment_name',11,0)`  
11 - это номер атома в фрагменте для связи



# Sculpting, что ЭТО?

Это похоже на real-time оптимизатор геометрии, но это алгоритм, который старается сохранить значения длины связей, углов, торсионных углов при изменении координат.



# Как запустить sculpting?

У вас достаточно мощный компьютер? Тогда:

- Переводим мышь в режим редактирования
- Выбираем "auto-sculpting" из меню Sculpting
- Выбираем Sculpting из меню Wizard
- Выбираем центральный атом для модификаций  
Ctrl-middle-click
- Тянем атом в любую сторону ctrl-left-click-and-drag



# Скриптование в PyMol

Возможны как скрипты из команд, так и скрипты на Python

Запуск скриптов из команд:

`@ myfile.pml`

Запуск скриптов на питоне:

`run myfile.py`

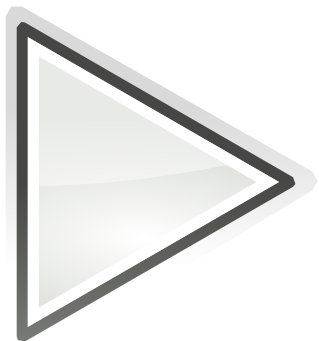


# Пример

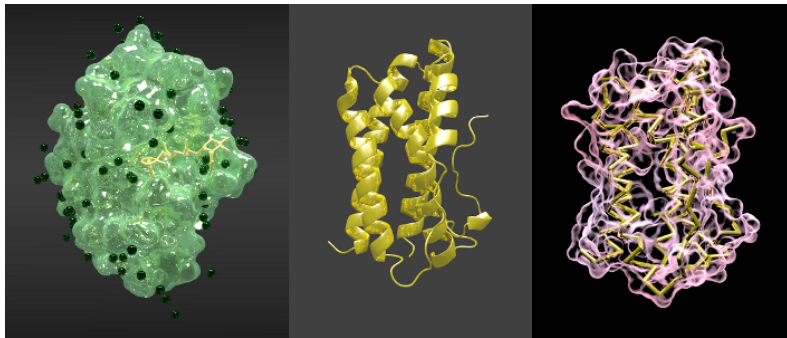
```
fetch 1c1l, async=0
as lines, n. C+O+N+CA
zoom i. 4+5
mset 1 x1440
mview store
python
for x in range(0,144):
    cmd.frame(((10*x)+1))
    cmd.zoom("n. CA and i. " + str(x) + "+" + str(x+1))
    cmd.mview("store")
python end
frame 288
mview store
mview reinterpolate
```



# Результат



# Объекты из PyMol можно использовать в разных 3D программах





# Анимация структуры в Blender

