

# RosettaScripts Tutorial

线上文件: [📖 RosettaScripts Tutorial](#)

## 准备工作

1. 找到系统中加载的第二个硬盘，将下载好的rosetta\_scripts\_tutorial文件夹放在这个盘中
2. 打开Terminal，创建一个link便于快速到达这个硬盘

```
1 ln -s /media/ubuntu/a8850f6f-75fc-49c4-8d7f-8841a0215149 ~/disk2
```

3. 如果遇到问题请助教帮忙

## 1. 启用RosettaScripts

- A. 打开vscode
- B. 在vscode左侧Explorer中打开文件夹rosetta\_scripts\_tutorial: 菜单栏File -> Open Folder...
- C. 在vscode下侧打开Terminal: 菜单栏Terminal
- D. 在vscode的terminal窗口启动rosetta\_scripts应用程序

```
1 ~/disk2/rosetta/source/bin/rosetta_scripts.default.linuxgccrelease
```

⚠注意理解此处的输出内容

- E. 在vscode目录栏双击打开 `empty.xml`



.xml文件格式的基本语法特点

<总条目>

<子条目1>

<子条目1的子条目/>

<子条目1/>

<子条目2/>

<总条目/>

? RosettaScripts xml文件中的条目和条目包含关系是：

```
1 <ROSETTASCRIPTS>
2     <SCOREFXNS>
3     </SCOREFXNS>
4     <FILTERS>
5     </FILTERS>
6     <MOVERS>
7     </MOVERS>
8     <PROTOCOLS>
9     </PROTOCOLS>
10    <OUTPUT />
11 </ROSETTASCRIPTS>
```

## 2. 启动RosettaScripts执行simple.xml

### 任务1：为top7 重新设计sequences (fix backbone protein design)

A. 在vscode目录栏中双击打开 `simple.xml` 文件

B. 与empty.xml内容相比，simple.xml中定义了一个 `scorefunction`，一个 `mover`，一个 `filter`

```
1 <SCOREFXNS>
2 <ScoreFunction name="ref2015" weights="ref2015"/>
3 </SCOREFXNS>
4 <FILTERS>
5 <NetCharge name="net_charge" min="-100" max="100"/>
6 </FILTERS>
7 <MOVERS>
8 <PackRotamersMover name="pack_all" scorefxn="ref2015"/>
9 </MOVERS>
10
11 <PROTOCOLS>
12     <Add mover="pack_all"/>
13     <Add filter="net_charge"/>
14 </PROTOCOLS>
```

C. 在terminal启动rosetta\_scripts应用程序执行simple.xml:

```
1 ~/disk2/rosetta/source/bin/rosetta_scripts.default.linuxgccrelease \  
2 -s ./1qys.pdb \  
3 -parser:protocol ./simple.xml \  
4 -out:prefix simple_ \  
5 -nstruct 3
```

⚠注意理解此处各项代表内容和程序执行后的输出

D. 在terminal打开pymol查看输出pdb结果

```
1 conda deactivate  
2 pymol 1qys.pdb simple_*.pdb
```

E. 用vscode打开 `simple_score.sc` 查看模型得分情况



rosetta\_scripts应用程序通过执行simple.xml对输入的蛋白质结构做了什么？

新设计的三个蛋白质模型分别带电多少？

各自总能量是多少？

### 3. 认识residue selector和task operation

**任务2：将top7表面正电荷增加到 +10 或更高，以便与带负电的mRNA粘在一起。**

A. 在vscode目录栏中双击打开 `complex.xml` 文件

B. 与simple.xml内容相比，complex.xml中定义了更多内容：

在RESIDUE\_SELECTORS条目下：

```
1 <RESIDUE_SELECTORS>  
2   <Layer name="surface" select_core="false" select_boundary="false"  
   select_surface="true"/>  
3   <Layer name="boundary" select_core="false" select_boundary="true"  
   select_surface="false"/>  
4   <Layer name="core" select_core="true" select_boundary="false"  
   select_surface="false"/>  
5 </RESIDUE_SELECTORS>
```

在TASKOPERATIONS条目下：

```
1 <TASKOPERATIONS>
2   <OperateOnResidueSubset name="fix_core" selector="core">
3     <PreventRepackingRLT/>
4   </OperateOnResidueSubset>
5
6   <OperateOnResidueSubset name="design_surface" selector="surface">
7     <RestrictAbsentCanonicalAASRLT aas="KR"/>
8   </OperateOnResidueSubset>
9 </TASKOPERATIONS>
```

FILTERS, MOVERS, PROTOCOLS也有相应的变化：

```
1 <FILTERS>
2   <NetCharge name="net_charge" min="-100" max="100"/>
3   <ResidueCount name="lys_num" min_residue_count="0" residue_types="LYS"
4     residue_selector="surface"/>
5   <ResidueCount name="arg_num" min_residue_count="0" residue_types="ARG"
6     residue_selector="surface"/>
7 </FILTERS>
8
9 <MOVERS>
10   <FastDesign name="design_mover" scorefxn="ref2015"
11     task_operations="fix_core,design_surface"/>
12 </MOVERS>
13
14 <PROTOCOLS>
15   <Add mover="design_mover"/>
16   <Add filter="net_charge"/>
17   <Add filter="lys_num"/>
18   <Add filter="arg_num"/>
19 </PROTOCOLS>
```

C. 在terminal启动rosetta\_scripts应用程序执行complex.xml

```
1 ~/disk2/rosetta/source/bin/rosetta_scripts.default.linuxgccrelease \
2 -s ./1qys.pdb \
3 -parser:protocol ./complex.xml \
4 -out:prefix complex_ \
5 -nstruct 3
```

#### D. 在pymol中查看结果

```
1 pymol 1qys.pdb complex_*.pdb
```

#### E. 在vscode中打开 `complex_score.sc` 查看模型得分情况



rosetta\_scripts应用程序通过执行complex.xml对输入的蛋白质结构做了什么？

新设计的三个蛋白质模型分别带电多少？

这样设计的三个蛋白质模型分别带电多少？

各自总能量是多少？

这是合理的蛋白质模型吗？

## 4. 从fixed backbone sequence design到flexible backbone sequence design

#### A. 在vscode目录栏中双击打开 `complex_fastdesign.xml` 文件

#### B. 仔细查看complex\_fastdesign.xml内容，识别出与complex.xml的不同

```
1 <MOVERS>
2     <FastDesign name="design_mover" scorefxn="ref2015"
3     task_operations="fix_core,design_surface"/>
4 </MOVERS>
```


#### C. 在terminal启动rosetta\_scripts应用程序执行complex\_fastdesign.xml

```
1 ~/disk2/rosetta/source/bin/rosetta_scripts.default.linuxgccrelease \
2 -s ./1qys.pdb \
3 -parser:protocol ./complex_fastdesign.xml \
4 -out:prefix complex_fastdesign_ \
5 -nstruct 1
```


#### D. 在pymol中查看输出pdb结果

```
1 pymol 1qys.pdb complex_fastdesign_*.pdb
```

E. 用vscode双击打开 `complex_fastdesign_score.sc` 查看模型得分情况

 这次的设计的蛋白质模型又发生了什么变化?

## 5. BONUS: 是否还有其它方法实现任务2的设计目标?

 请通过对以上所学到的关于filter, mover, residue selector 以及task\_operation的组合, 编写新的RosettaScripts可以读取的xml文件, 进行蛋白质设计和结果分析。