

# 使用 MolAICal 两步实现对 Mpro 蛋白活性口袋的虚拟筛选

作者: Qifeng Bai (update 2021-12-14)

更多教程 (含英文教程) 请见如下:

MolAICal 官方主页: <https://molaical.github.io>

MolAICal 官方主页中国镜像: <https://molaical.gitee.io>

MolAICal 中文博客: <https://molaical.gitee.io/cntutorial.html>

## 1. 简介

在本教程中, 介绍了使用已知数据库 (如 ZINC 数据库等) 对 Mpro 活性口袋进行快速药物虚拟筛选的方法。本教程学习的前提是你会用 UCSF Chimera 和 MolAICal 处理蛋白质结构。如果你对蛋白质结构处理不熟悉, 可以参考如下网站所提供的教程:

<https://gitee.com/molaical/documents/tree/master/tutorials/002-AIVS>.

## 2. 工具

### 2.1. 所需软件下载地址

1) MolAICal: <https://molaical.github.io> 或 <https://molaical.gitee.io>

国内镜像 MolAICal: <https://molaical.gitee.io>

### 2.2. 操作示例文件

1) 所有用到的操作教程文件均可在下面的网站下载:

<https://gitee.com/molaical/tutorials/tree/master/003-VS>

2) “ligandSet.mol2”文件包含 16 个从 ZINC 数据库中选择的配体作为示例。你也可以选择其它数据库进行练习。

3) 使用 PDBQT 格式的 Mpro 蛋白质文件“pro.pdbqt”用来做分子对接。

## 3. 操作流程

转至教程目录:

```
#> cd 003-VS
```

**3.1.** 将配体分子集“ligandSet.mol2”分割成单个分子文件。如果你所选的药物数据库已经被分割成了单个分子文件, 这一步可以省略。

```
#> molaical.exe -tool mol2 -w split -n 1 -v true -m number -i ligandSet.mol2 -o splitdir
```

**另外选项:** 当 mol2 文件的第二行是有效字符的时候, 你也可以使用第二行的字符作为文件名, 命令如下:

```
#> molaical.exe -tool mol2 -w split -n 1 -v true -i ligandSet.mol2 -o splitdir
```

**注意:** 在 ligandSet.mol2 中的每个配体应该是完整的分子, 如果配体缺氢原子, MolAICal 将可能不会生成对应配体的 PDBQT 文件, 用户可以使用 UCSF Chimera 进行加氢, 或使用 MolAICal 中的命令, 如下 (用户需要把 1.mol2 文件替换成自己的文件名):

```
#> molaical.exe -tool format -i E:/1.mol2 -o E:/1.pdbqt
```

**3.2. 配置对接文件“conf.txt” (内容如下)。**参数“receptor”、“center”和“size”应根据实际受体正确设置。

```
receptor = pro.pdbqt
out = all.pdbqt

cpu = 1
center_x = -10.733
center_y = 12.416
center_z = 68.829

size_x = 25
size_y = 30
size_z = 25

num_modes = 1
```

**3.3. 运行虚拟筛选命令。**

```
#> molaical.exe -dock vs -i splitdir -nc 3
```

**注意:** 如果用户直接使用 PDBQT 格式的配体进行虚拟筛选, 用户可以使用下面的命令:

```
#> molaical.exe -dock vs -i splitdir -b off -k pdbqt -nc 3
```

**-nc:** 代表 CPU 核心数。

**-b:** 是否进行分子格式转化, “on” 代表进行转化, “off” 代表不进行分子格式转化。默认值是 “on”。

**-k:** 指定虚拟筛选的分子格式, 默认值是 mol2 的分子格式。

## 4. 结果

### 4.1 查看结果

用户可以使用 Pymol (<https://github.com/cgohlke/pymol-open-source-wheels> 或 <https://www.cgohlke.com>) 直接载入 PDBQT 格式的分子. 这里使用 UCSF Chimera 来查看虚拟筛选的结果。

打开 003-VS/splitdir

1) 加氢 (可选)

```
#> molaical.exe -dock addh -i 1_out.pdbqt
```

2) 更改格式“pdbqt”成“pdb”

```
#> molaical.exe -dock pdbqt2pdb -i 1_out.pdbqt
```

上述命令将产生一个名为“1\_out.pdb”的文件。然后使用 UCSF Chimera 打开配体分子“1\_out.pdb”和受体分子“protein.pdb”。图 1 显示了对接筛选的结果，表明了 MolAICal 可以虚拟筛选到合适的配体分子。

**注意：**图 1 的显示方面有些步骤省略了，假如用户想得到图 1 这样 surface 的图像，可以使用 UCSF Chimera 工具栏的选项：“Actions→Surface→show”

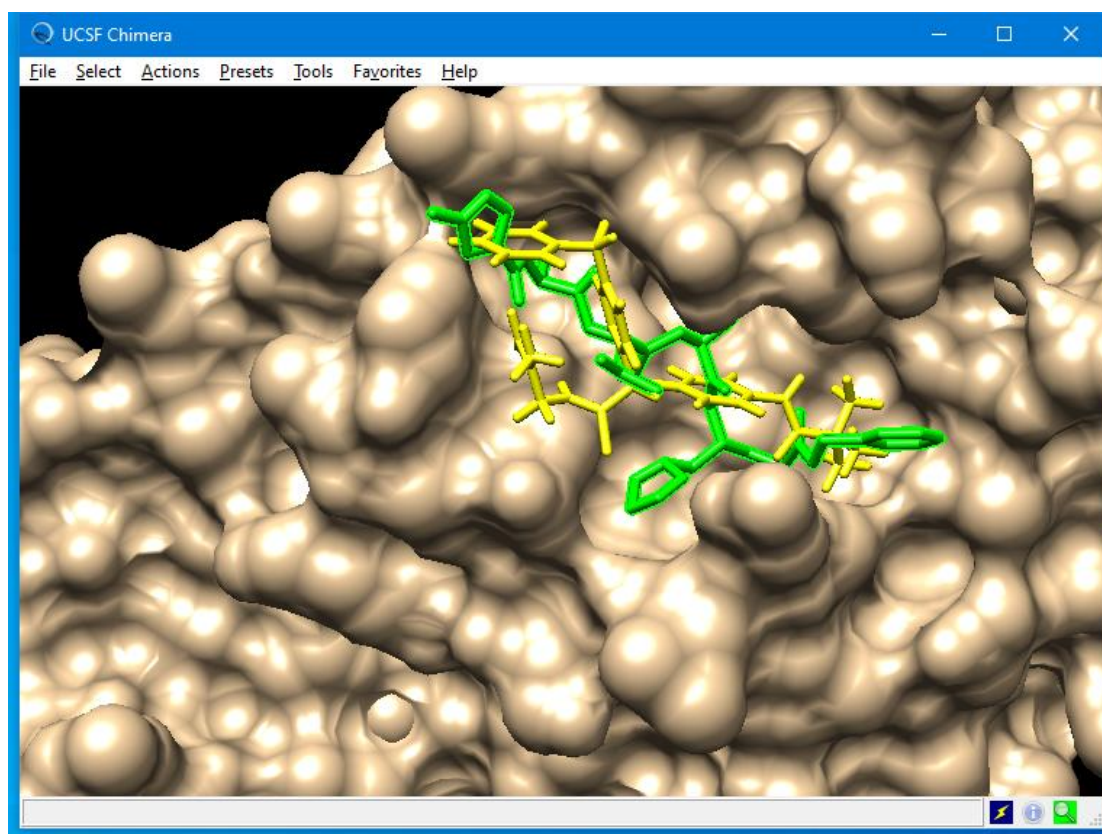


图 1. 绿色分子代表 SARS-CoV-2 Mpro 的抑制剂 N3，黄色分子代表筛选出的分子

#### 4.2 对虚拟筛选的结果进行排序

切换到 003-VS，然后运行如下命令：

```
#> molaical.exe -call run -c sfile -i printscore.py "splitdir/*out.pdbqt"
```

这个命令将显示分子名称和对应的打分（按分数升序排列），假如用户想保存输出的结果，可以使用如下命令：

```
#> molaical.exe -call run -c sfile -i printscore.py "splitdir/*out.pdbqt" > results.log
```

**说明：**“molaical.exe” 是实际 MolAICal 的安装目录。所有的脚本文件都保存在 MolAICal 子文件夹“scripts”中。

### 4.3 提取排名靠前的分子到新建的文件夹中

假如用户想将排名靠前的分子移动到新建的文件夹中，以便于分析结果，本教程提供了一种方法，下面的命令是将 2 个排名靠前的分子移到名为“results”的文件夹中：

```
#> molaical.exe -call run -c sfile -i get_top_results.py "splitdir/*out.pdbqt" 2 results
```

运行上述命令，2 个排名靠前的分子将被移动到 “results” 文件夹中

## 5. 续跑虚拟筛选任务（选项）

如果虚拟筛选任务因不可预见的情况而提前终止，恢复过程只需两个步骤：

1. 通过以下命令将所有现有的 VS 结果文件（包含默认“\_out.pdbqt”后缀）移动到另一个目录：

```
#> molaical.exe -call run -c sfile -i molaical_batch.py -sd splitdir  
-ds -ik "_out.pdbqt" > tmplist
```

**-ds:** 是否搜索具有指定字符的文件，如果仅输入“-ds”，表示 True，搜索具有指定字符的文件。

**-sd:** 搜索的目录。

**-ik:** 用于搜索的关键字。

```
#> molaical.exe -call run -c sfile -i molaical_batch.py -cvs -il tmplist  
-ol l_complete_folder
```

**-cvs:** 是否准备继续运行虚拟筛选。如果仅输入"-cvs", 表示 True, 准备继续运行虚拟筛选作业。

**-il:** 配体列表的文件。

**-ol:** 在使用"-cvs"继续虚拟筛选的部分中, 是存储的文件夹。

**注意:** 完整分子将被移动到目录"l\_complete\_folder"。

**选项:** 有时, 一个文件夹中可能有大量分子。为了减少一个文件夹中的分子数量, 可以按如下方式指定一个文件夹中的分子数量:

```
#> molaical.exe -call run -c sfile -i molaical_batch.py -cvs -dp 3 -cv  
s -il tmplist -ol t1_folder
```

**-dp:** 在使用"-cvs"继续虚拟筛选的部分中, 是每个文件夹中存储的筛选分子数量。默认值为"None", 表示将所有筛选分子存储在一个文件夹中。可以是 1、2 或 3..., 表示每个文件夹中存储的分子数量。

**注意:** 将生成文件夹"t1\_folder1"、"t1\_folder2"..., 每个文件夹包含'-dp'后的存储数量。

2. 重复上述虚拟筛选步骤以处理未筛选的分子。

### 6. Linux 版 MolAICal: 架构与加速方案

如图 2 所示, MolAICal 容器不仅能调用容器系统的文件和程序, 还能通过挂载与映射机制访问宿主机库文件, 有效解决了库冲突或依赖缺失等问题。然而, 容器内程序的执行速度确实可能低于本地环境。

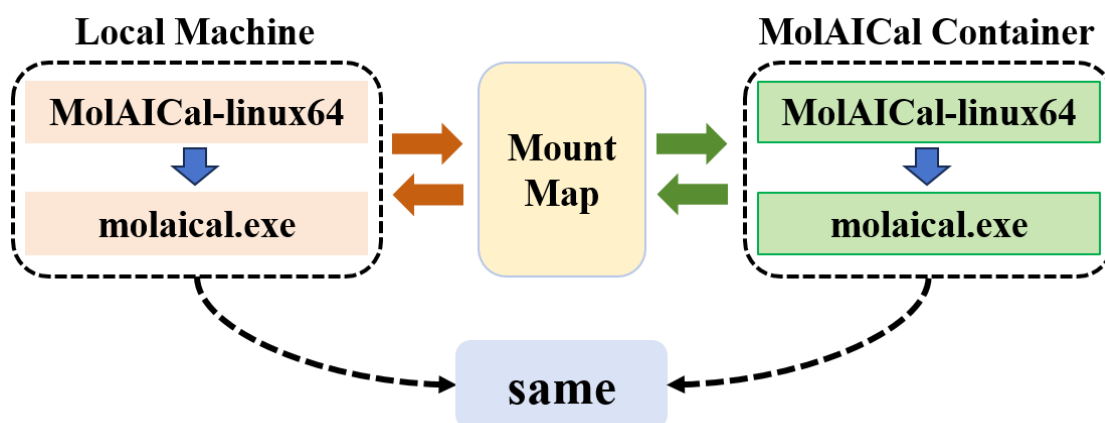


图 2. MolAICal Linux 版本架构图

图 2 表明: 由于映射关系的存在, 容器中"molaical.exe"执行的计算命令与本地"molaical.exe"运行命令完全一致。

- ◆ 因此对于库依赖较少的任务(如使用 MolAICal 进行虚拟筛选), 建议将 MolAICal 从容器复制到本地:

# 1. 进入容器文件系统 (默认进入"/root"目录)

```
#> molaical.exe -eset shell in
```

# 2. 'cp'命令前半部分指向容器路径, 后半部分指向宿主机路径

```
#> cp -r soft/MolAICal-linux64 <local machine directory>/
```

# 3. 退出容器文件系统

```
#> exit
```

- ◆ 最终通过相同方法在本地执行虚拟筛选, 可显著加速筛选过程。

# 附录

参考其它排序方法：

假如你在 Windows 环境下：

使用 Excel 打开“results.log”，其中分隔符“Separator”选择空格。或者直接将“results.log”的内容复制到 Excel 中。在第二列，选择所有数据，并且根据需要选择工具栏中的“Sort Largest to Smallest”（降序）或“Sort Smallest to Largest”（升序）（见图 3）。

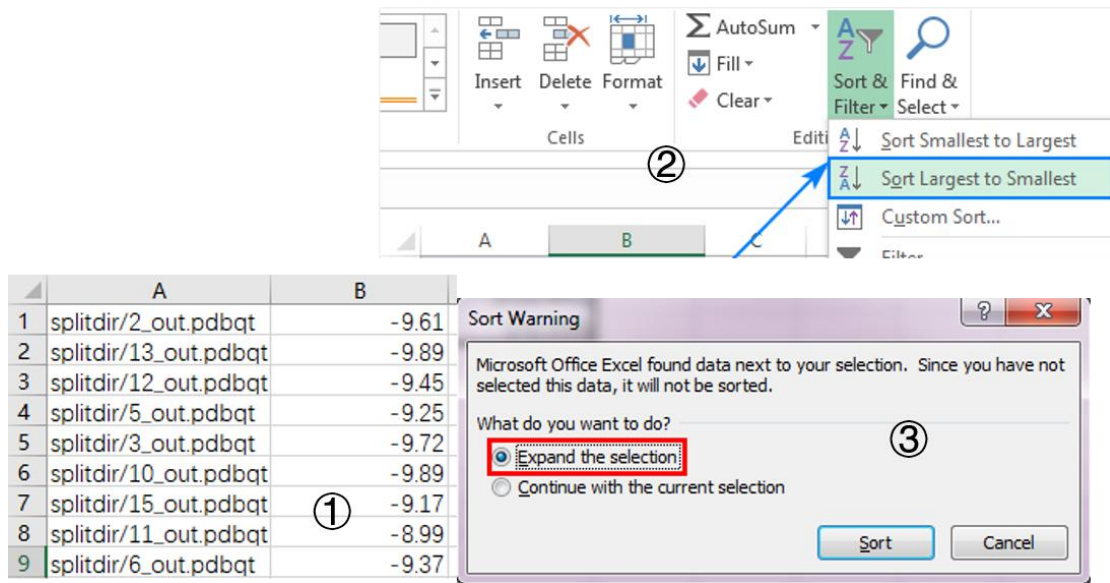


图 3. 排序结果

假如你在 Linux 环境下，用户可以使用如下命令：

```
#> sort -n -t ' ' -k 2r results.log > rank.dat
```

说明：参数“2r”是升序排列，而“1r”是降序排列。