

# 使用 MolAICal 在受体口袋中进行肽虚拟筛选

**Qifeng Bai**

邮箱: molaical@yeah.net

主页: <https://molaical.github.io> 或 <https://molaical.gitlab.io>

## 1. 引言

本教程介绍了多肽的生成, 以及基于受体口袋的多肽虚拟筛选的步骤。**LightDock** 的功能实现依赖于打分函数所支持的氨基酸类型。例如: **DFIRE** 仅支持标准氨基酸以及残基名称为"MMB"的氨基酸, 但若采用 **MolAICal** 的默认筛选方法, 则不受特定氨基酸类型的限制。本教程提供了详细的多肽虚拟筛选的步骤方法; 如果用户想进一步了解 MolAICal 处理受体和配体结构的步骤方法, 请参见

<https://molaical.github.io> 中的"Quick Start 1")。如果用户想了解更详细的分子准备过程, 可以学习教程:

<https://github.com/MolAICal/documents/tree/master/tutorials/002-AIVS>。这里, 我们使用 MolAICal 软件包进行本教程。

## 2. 材料

### 2.1. 软件需求

- 1) MolAICal: <https://molaical.github.io>
- 2) UCSF Chimera: <https://www.cgl.ucsf.edu/chimera/download.html>
- 3) PyMol: <https://github.com/cgohlke/pymol-open-source-wheels>,  
<https://github.com/maabuu/pymol-wheels>,  
<https://pypi.org/project/pymol-open-source-whl>,  
<https://github.com/cnpem/PyMOL4Win>

### 2.2. 示例文件

- 1) 所有必要的教程文件可从以下地址下载:  
[https://gitee.com/molaical/tutorials/tree/master/029-peptide\\_vs\\_molaical](https://gitee.com/molaical/tutorials/tree/master/029-peptide_vs_molaical)

## 3. 步骤

### 3.1 检查材料

进入教程目录:

```
#> cd 029-peptide_vs_molaical
```

将看到文件列表：

- ◆ **7LLL.pdb**: PDB 格式的复合物结构文件，来自 <https://www.rcsb.org>
- ◆ **P\_peptide.pdb**: 来自"7LLL.pdb"的肽配体。准备方法见附录 1 或 MolAIcal 蛋白质-肽对接教程。
- ◆ **conf.txt**: 需要设置的对接参数。
- ◆ **R\_protein.pdb**: 来自"7LLL.pdb"的受体文件。准备方法见附录 1 或 MolAIcal 蛋白质-肽对接教程。

上述分子文件的准备方法见附录 1 或 MolAIcal 蛋白质-肽对接教程 (<https://molaical.gitlab.io/tutorial.html>)。

### 3.2. 准备对接参数

- 1) 获取受体口袋的盒子中心。这里，有一个肽结合到受体上，因此该肽的中心被认为是受体口袋的盒子中心。可以通过以下命令获取：

```
#> molaical.exe -call run -c sfile -i m_center.py P_peptide.pdb
```

将输出：

```
Geometric center - State 1: 96.382238 109.960593 159.069831  
Center of mass - State 1: 96.405446 110.003945 158.961354
```

这里，使用几何中心(Geometric center)。根据获得的几何中心，可以确定受体口袋的盒子大小，如图 1 所示。

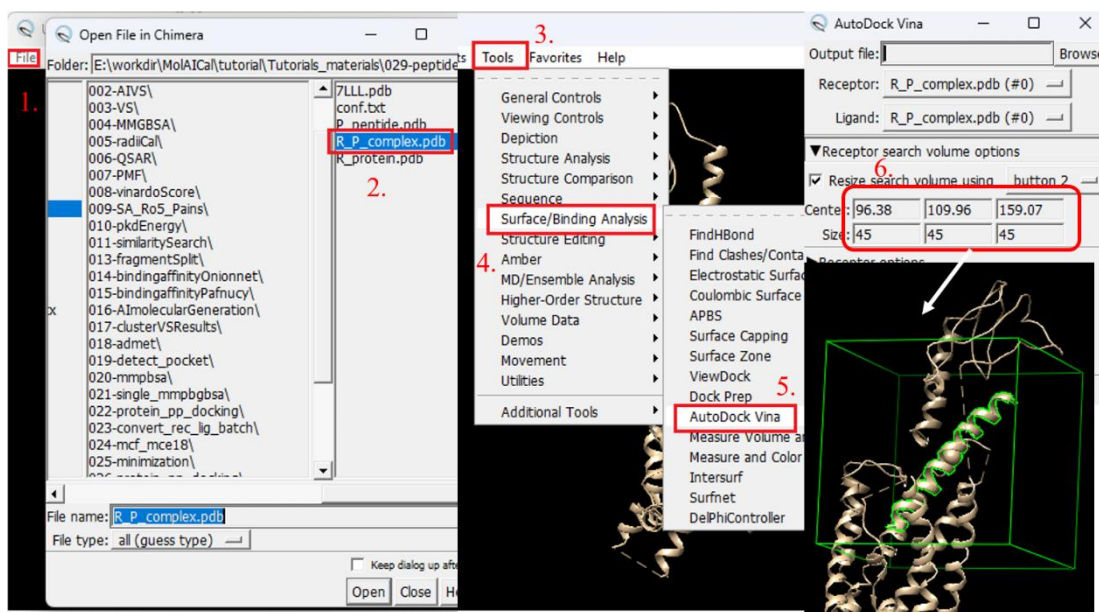


图 1

- 2) 基于以下命令准备 PDBQT 格式的受体文件：

```
#> molaical.exe -dock receptor -i R_protein.pdb
```

将生成文件"R\_protein.pdbqt"。

现在根据上述结果更新对接参数文件"conf.txt"（受体、中心和大小）：

```
receptor = R_protein.pdbqt

cpu = 1
center_x = 96.38
center_y = 109.96
center_z = 159.07

size_x = 45
size_y = 45
size_z = 45

num_modes = 1
```

### 3.3. 肽生成

进入教程目录：

```
#> cd 029-peptide_vs_molaical
```

#### 1) 通过 MolAICal 生成线性肽

```
#> molaical.exe -call run -c pepgen -i -l 8 -n 2 -3d -nc -o linear_seq.
txt -dir linear
```

- ◆ ‘-call’：与外部程序或命令交互。其值可以是‘set’或‘run’。‘set’表示设置外部程序的环境，包括名称和路径。‘run’表示调用外部程序，可以从设置的环境文件中搜索程序的路径。
- ◆ -i：调用外部程序。在‘-i’之后包含外部程序的所有命令行参数。
- ◆ -c（在-i 之前）：如果其值为"pepgen"，则将生成肽。
- ◆ -l：肽中氨基酸残基的数量
- ◆ -n：要生成的唯一序列数量
- ◆ -3d：使用 pyPept 生成 3D 结构
- ◆ -nc：不向序列添加 ac/am 端帽
- ◆ -o：生成序列的输出文件
- ◆ -dir：3D 结构的输出目录

更多详细信息，用户可以参考 MolAICal 手册。

将在"linear"文件夹中看到生成的 3D 肽，可用于进一步的虚拟筛选。

#### 2) 通过 MolAICal 生成具有固定残基的线性肽，可用于肽修饰。

```
#> molaical.exe -call run -c pepgen -i -l 8 -n 3 -rc 3 -3d -dir mpep -s  
EEYVPIST -d none -fix 1 7
```

- ◆ -dir: 3D 结构的输出目录
- ◆ -rc: 要生成的 RDKit 随机构象数量（默认：1），可以给出所需的肽数量。
- ◆ -d: 从种子序列扩展的方向[forward, backward, both, none]。
- ◆ -s: 起始序列（可选）
- ◆ -fix: 在生成过程中保持固定的位置（1-索引）

更多详细信息，用户可以参考 MolAICal 手册。

- 3) 通过 MolAICal 生成环状肽。MolAICal 还支持生成具有不同残基间二硫键等的肽生成（参见 MolAICal 手册）。

```
#> molaical.exe -call run -c pepgen -i -l 8 -n 3 -rc 3 -cc -3d -c 4 -di  
r cyclic
```

- ◆ -cc: 生成环状肽（仅 BILN/HELM）
- ◆ -c（在-i 之后）：用于并行处理的 CPU 核心数

更多详细信息，用户可以参考 MolAICal 手册。

在"linear"、"mpep"和"cyclic"目录中，包含了三种类型的肽，用于不同的目的和应用。这些肽将用于后续的虚拟筛选。由于肽虚拟筛选耗时较长，本教程仅提取有限数量的肽进行演示。用户可以根据其计算资源生成足够的肽用于研究目的。

### 3.4. 肽虚拟筛选

用户可以选择刚性对接或柔性对接进行虚拟筛选。

#### 3.4.1. 基于刚性对接的虚拟筛选

由于 MolAICal 可以生成肽配体的多种构象，可以使用 MolAICal 执行刚性对接以降低计算成本。

```
#> molaical.exe -dock vs -i linear -k pdb -nc 3 -p "-Z"
```

- nc: 表示使用的 CPU 核心数。
- b: 切换分子格式转换。"on"表示转换。"off"表示不转换。默认值为"on"。
- k: 文件包含用于虚拟筛选的关键词。默认值为".mol2"。

**-p:** 可以接受用于格式转换的附加参数。跟随它的值必须用双引号""括起来。如果值包含多个参数，必须用逗号","分隔。默认值为空字符串""。参数'-p'后的值"-Z"将使所有活动扭转失活，以实现刚性对接。

### 3.4.2. 基于柔性对接的虚拟筛选

对于蛋白质和肽对接，计算量非常大且耗时。命令如下（不添加'-p'参数）：

```
#> molaical.exe -dock vs -i linear -k pdb -nc 3
```

### 3.4.3. 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 linear -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:** 用于搜索的关键词。

```
#> molaical.exe -call run -c sfile -i molaical_batch.py -cvs -il tmplis  
t -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 -cvs -i  
l 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。

## 4. 结果

### 4.1 检查结果

用户可以使用 PyMol 软件 (<https://github.com/cgohlke/pymol-open-source-wheels> 或 <https://www.cgohlke.com>) 直接加载 PDBQT 格式的配体。这里，介绍使用 PyMol 显示对接结果。

打开 029-peptide\_vs\_molaical/linear

使用 PyMol 加载案例"**LKGIWWPL\_alpha-helix\_out.pdbqt**"和"**R\_protein.pdb**"。对接结果如图 2 所示，表明 MolaICal 获得了合适的肽配体。如果用户想分析对接肽和蛋白质之间的氢键相互作用，应使用 PyMol 给多肽和蛋白质添加氢原子。

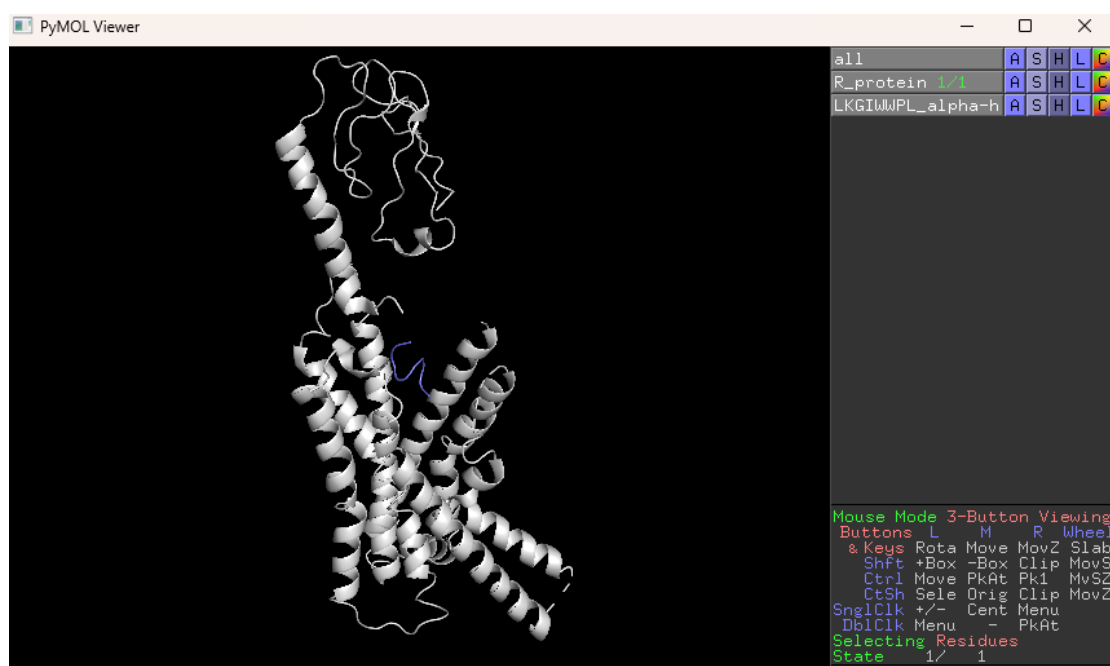


图 2

### 4.2 虚拟筛选结果排名

进入文件夹"029-peptide\_vs\_molaical", 运行以下命令:

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

此处，"linear" 为存放虚拟筛选结果的目录。若已执行续跑任务，请切换至续跑任务完成后生成的目录（例如本任务中的 "l\_complete\_folder" 目录）。将显示配体名称和相应的结合分数（按分数升序排序）。如果用户想将结果保存到文件中，可以使用以下命令：

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

注意: "molaical.exe"是您安装 MolAICal 的目录路径。所有有用的脚本都提供在 MolAICal 的"scripts"文件夹中。

### 4.3 将排名靠前的分子提取到新文件夹

如果用户想将排名靠前的分子移动到新文件夹,例如,在本教程中,如果用户想将 2 个排名靠前的分子移动到名为"results"的新文件夹,可以使用以下命令:

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

两个排名靠前的分子将被移动到文件夹"results"中。

## 5. Linux 版 MolAICal: 架构与加速方案

如图 3 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。

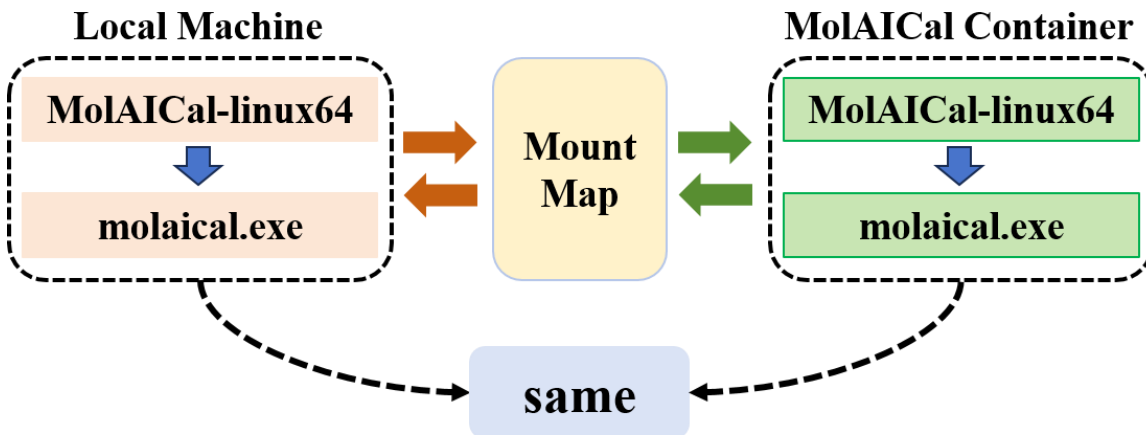


图 3. MolAICal Linux 版本架构图

图 3 表明: 由于映射关系的存在, 容器中"molalcal.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计算机上执行虚拟筛选，可显著加速筛选过程。



## 附录 1

下面的附录 1 描述了为本教程准备材料的方法，这与蛋白质-肽分子对接的步骤相同；如果您已经熟悉蛋白质-肽分子对接工作流程，可以跳过本节。

### 1. 处理分子文件

将 PDB 文件 7LLL.pdb 加载到 PyMol 中，并使用图 a1 的步骤分别保存 "R\_P\_complex.pdb"、"R\_protein.pdb"和"P\_peptide.pdb"。

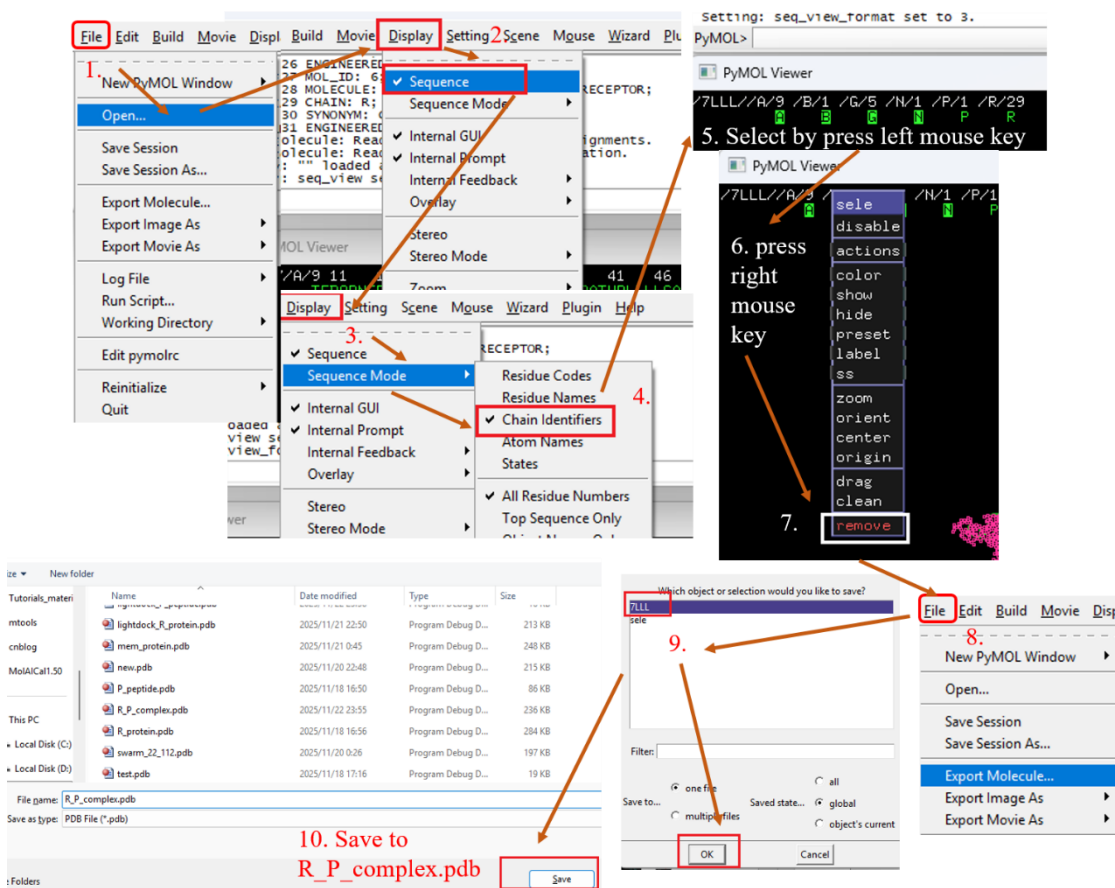


图 a1

文件 "R\_P\_complex.pdb"（链名为 "P" 和 "R"）、"R\_protein.pdb"（链名为 "R"）和 "P\_peptide.pdb"（链名为 "P"）是从胰高血糖素样肽-1 受体 (GLP-1R) 和 exendin-4（首个 FDA 批准的 GLP-1R 激动剂，PDB ID: 7LLL.pdb）的复合物中提取的。激动剂 exendin-4 是一种肽。如果用户想进行蛋白质-蛋白质或蛋白质-核酸虚拟筛选，请用蛋白质或核酸文件替换肽文件，因为肽可以被视为小蛋白质。