

# 使用 MolAICal 基于单个 PDB 文件计算小分子和蛋白 MM/GBSA 和 MM/PBSA 的教程

作者: Qifeng Bai

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.gitlab.io>

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

## 1. 简介

在本教程中介绍了基于单个 PDB 分子文件, 使用 MolAICal 计算小分子和 Mpro 蛋白受体 MM/GBSA 和 MM/PBSA 的方法。本教程不仅可以用于计算蛋白质-配体的 MM/GBSA 和 MM/PBSA, 还可以用于计算基于 MD 模拟的蛋白质-多肽、蛋白质-蛋白质、DNA-配体、DNA-蛋白质、蛋白质-DNA、蛋白质-RNA 和其它任意复合物的 MM/GBSA 和 MM/PBSA; 只需要用指定的对象替换本教程中的蛋白质和配体即可。例如, 基于本教程中相同的运行命令参数, 本教程中的蛋白质被替换为 DNA, 配体被替换为多肽。

## 2. 工具

### 2.1. 所需软件下载地址

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

2) NAMD (CPU 版本, 如: "Linux-x86\_64-multicore"): <https://www.ks.uiuc.edu/Research/namd/>

**注意:** 本教程可以使用 **NAMD2.x**、**3.x** 或更高版本。例如, 如果使用 **NAMD 3.x** 版本, 则使用命令“**namd3**”替代本教程中的命令“**namd2**”。对于更高版本的 NAMD, 用户可以使用与前面示例类似的替换方式。

3) Carma: <https://github.com/glykos/carma> or <https://utopia.duth.gr/~glykos/Carma.html>

4) APBS (<https://apbs.readthedocs.io>) or DelPhi (<http://compbio.clemson.edu/lab/delphisw>)

### 2.2. 操作示例文件

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/021-single\\_mmpbgsa](https://gitee.com/molaical/tutorials/tree/master/021-single_mmpbgsa)

### 3. 操作流程

为了解决 Linux 中的库依赖问题，Linux 版本的 MolAIcal (Windows 版本的 MolAIcal 没有采用容器)采用了基于容器的方法。如果需要调用外部程序，建议在 MolAIcal 容器内安装这些程序。如果不需要外部程序，可以忽略此步骤。本教程需要外部程序 NAMD 和 VMD，具体步骤如下：

#### 1. 首先，将文件复制到 MolAIcal 容器中

```
# 进入容器文件系统（将进入 "/root" 目录）
#> molaical.exe -eset shell in

# 将 VMD 和 NAMD 安装包从本地机器复制到容器中，'cp' 命令的第一部分（源路径）
# 位于本地主机，第二部分（目标路径）在容器内：VMD 和 NAMD 软件包可通过以下命令移入容器。
#> cp /home/user/<本地文件> /root/soft

# 退出容器文件系统
#> exit
```

#### 2. 其次，进入 MolAIcal 容器的虚拟环境

```
#> molaical.exe -eset sys run molaical
```

注：molaical 是容器名称。

#### 3. 在 MolAIcal 容器虚拟环境中安装软件的方式与在本地主机上安装相同。以下以安装 VMD 和 NAMD 为例：

##### 1) 安装 NAMD：

解压 NAMD 文件（假设解压后的文件夹名为 namdcpu），然后使用以下命令将其路径告知 MolAIcal：

```
#> molaical.exe -call set -n NAMD -p "/root/soft/namdcpu/namd3"
```

注：请将上述 VMD 和 NAMD 的路径替换为您系统中的实际路径。-n 后面的 "VMD" 和 "NAMD"（大小写不敏感）是固定的标识符。为确保 MM/GBSA 结果的可重复性，建议使用 NAMD 的 CPU 版本，因为 CUDA 版本中的 seed 参数似乎对结果可重复性无效。

##### 2) 安装 VMD：

按以下步骤操作：

###### ◆ 解压 VMD 文件：

```
#> tar -xvzf vmd-xxx.tar.gz
```

注：请将上述路径替换为您系统中的实际路径。

◆ 修改 VMD 解压目录中名为 `configure` 的文件中的安装路径：

```
# 默认值:  
$install_bin_dir="/usr/local/bin";  
$install_library_dir="/usr/local/lib/$install_name";  
  
# 修改为:  
$install_bin_dir="/root/soft/vmd193/bin";  
$install_library_dir="/root/soft/vmd193/lib/$install_name";
```

◆ 安装 VMD：

```
#> cd vmd-xxx  
#> ./configure LINUXAMD64  
#> cd src  
#> make install
```

注：请运行 `./configure` 并根据所用计算机选择正确的类型，此处为 "LINUXAMD64"。

◆ 然后使用以下命令将 VMD 路径告知 MolAIcal：

```
#> molaical.exe -call set -n VMD -p "/root/soft/vmd193/bin/vmd"
```

至此，NAMD 和 VMD 在 MolAIcal 容器内的安装与配置已完成。为防止 MolAIcal 出现问题时丢失已安装的程序（例如 VMD 和 NAMD），请参阅附录 1 中的第 3 步。

◆ 记得使用 `exit` 命令退出 MolAIcal 虚拟环境，返回本地计算机进行计算（主要是为了省去文件拷贝步骤；在容器内运行也可行，但需手动将数据从本地计算机复制到 MolAIcal 容器中）。

```
#> exit
```

假设用户已在 MolAIcal 中配置了 VMD 和 NAMD 路径。上述说明适用于 Linux 版 MolAIcal 容器的外部程序安装。对于 Windows 版 MolAIcal，用户需将以下命令中的路径替换为实际系统路径后直接运行：

```
#> molaical.exe -call set -n VMD -p "e:/soft/vmd193/bin/vmd"  
#> molaical.exe -call set -n namd -p "e:/soft/namdcpu/namd3"  
#> molaical.exe -call set -n delphi -p "e:/soft/delphicpp_v8.4.3_serial"
```

注意：请将上述 VMD 和 NAMD 的路径替换为用户系统中的实际路径。当运行 MMPB/GB/SA 功能并通过 MolAIcal 设置环境变量时（例如：#> molaical.exe -call set -n VMD -p "path"），MMPB/GB/SA 中与"-n"参数对应的变量名是固定的，包括：'vmd'、'namd'和'delphi'（不区分大小写）。

◆ 其中 DelPhi 是二进制文件，在 linux 系统上需要赋予可执行权限："chmod +x delphicpp\_v8.4.3\_serial"，所以，可以参考附录 1 或上述 NAMD 安装步骤。  
◆ 请注意，Windows 版与 Linux 版的 MolAIcal 配置存在差异：Linux 版本采用基于 udocker 容器的技术方案，需在容器内部完成设置，而 Windows 版本则无需此步骤。

为确保 MM/PBSA 和 MM/GBSA 计算结果的可靠复现性，必须严格遵循下列要求：

- ✧ 使用 **NAMD 的 CPU 版本**，例如“Linux-x86\_64-multicore”；
- ✧ NAMD 运算时仅调用 **1 个 CPU 核心**；
- ✧ 若使用自定义 NAMD 输入文件（如示例中的“md\_configure.conf”），必须在文件中**设定固定随机数种子**，例如在“md\_configure.conf”内配置“seed 12345”。

### 3.1 发现问题

进入教程目录：

```
#> cd 021-single_mmpbgsa
```

“complex.pdb”是包含蛋白质和配体的文件。为计算 MM/GBSA 或 MM/PBSA，需通过 VMD 的 Autopsf 模块生成其 PSF 格式文件。我们发现 VMD 的 Autopsf 存在拓扑加载漏洞，需按以下方式修复：

```
#> molaical.exe -call run -c fixpsf
```

随后调用 MolAIcal 驱动 VMD 生成 “complex.pdb”的 PSF 文件：

```
#> molaical.exe -call run -c vmdargs -i -pdb complex.pdb -dispdev text -args autopsf autopsf -mol 0 -top ligand.str -prefix complex
```

**注：**运行 VMD “autopsf”时，需通过 MolAIcal 修复 VMD 的拓扑漏洞（详见前文说明）。

'-i' 后的参数为 VMD 参数：

**'-pdb'**：加载 PDB 格式分子文件

**'-dispdev text'**：以文本模式启动 VMD

- ✧ **'-args'**：向 Tk 控制台传递后续参数以执行 VMD 命令；
- ✧ **'-args'**之后的首参数：需加载的包名（如 package require autopsf）
- ✧ **'-args'**之后的次参数：加载包后执行的命令及其参数（如 autopsf autopsf -mol 0 -top ligand.str）
- ✧ Autopsf 默认加载拓扑文件：  
top\_all36\_prot.rtf, top\_all36\_lipid.rtf, top\_all36\_na.rtf, top\_all36\_carb.rtf, top\_all36\_cgenff.rtf, toppar\_all36\_carb\_glycopeptide.str, toppar\_water\_ions\_namd.str, 若无特殊拓扑需求，可省略 -top 参数。

选项：用户可通过以下命令查看 VMD 中 “autopsf”的帮助信息：

```
#> molaical.exe -call run -c vmdargs -i -dispdev text -args autopsf autopsf
```

最终生成 complex\_formatted\_autopsf.pdb 和 complex\_formatted\_autopsf.psf。执行 MM/GBSA 计算：

```
#> molaical.exe -call run -c mmpbgsa1 -i -dispdev text -args -s complex_formatted_autopsf.psf -c
```

```
complex_formatted_autopsf.pdb -cs "segname,AP1,CO1" -rs "segname,AP1" -ls "segname,CO1" -pb 0 -gb 1 -ff  
ligand.str
```

**注意:** 此处 MolAICal 支持将字符串 (如 "segname,AP1,CO1") 中的逗号 "," 视为等同于空格 " ", 即 "segname,AP1,CO1" 会被当作 "segname AP1 CO1" 处理。类似地, 该规则也适用于其他类似字符。这种输入方式有助于避免因命令行中包含空格而导致的解析错误。

**结果显示: 结合能 G binding: 60422.8572 kcal/mol.**

很明显, 未经能量最小化或分子动力学模拟, 非共价相互作用预测值严重失真。因此 MolAICal 提供了基于能量最小化与分子模拟的 MM/GBSA/PBSA 计算方法。

### 3.2 在 Windows 系统中基于复合物文件计算 MM/GBSA

运行以下命令可对 "complex.pdb" 文件进行 MM/GBSA 计算:

```
#> molaical.exe -call run -c script -n runScript_win.dat -i complex.pdb ligand.str complex
```

#### 注意事项:

- ◇ MolAICal 默认使用广义 Born 隐式溶剂 (Generalized Born Implicit Solvent) 模型进行分子动力学模拟, 相关参数可通过命令行参数直接修改, 例如 "-system\_topology" 和 "-system\_coords" 等 (详见下文参数说明)。
- ◇ '-i' 后面的字符串为参数列表, 将以空格分隔。分割后的各字符串将依次替换脚本文件中的 \$molargs1、\$molargs2、\$molargs3 等占位符。占位符的数量与 '-i' 后参数列表经空格分割后的字符串数量一致。请参考名为 "runScript\_win.dat" 的脚本文件:

```
# 1. fix autopsf  
molaical.exe -call run -c fixpsf  
  
# 2. generate psf  
molaical.exe -call run -c vmdargs -i -pdb $molargs1 -dispdev text -args autopsf autopsf -mol 0 -top $molargs2 -  
prefix $molargs3  
  
# 3. default to run 20 fs minimization and 40 fs MD simulation  
molaical.exe -call run -c runnamd -i -dispdev text -args -s $molargs3_formatted_autopsf.psf -c  
$molargs3_formatted_autopsf.pdb -ff $molargs2  
  
# 4. Cut last 10 steps for MM/GBSA calculation  
molaical.exe -call run -c catdcd -i -dispdev text -args -out result.dcd -first 51 -last 60 -stride 1  
$molargs3_formatted_autopsf_md.dcd  
  
# 5. Calculate MM/GBSA  
molaical.exe -call run -c mmpbgsa -i -dispdev text -args -s $molargs3_formatted_autopsf.psf -c  
$molargs3_formatted_autopsf.pdb -t result.dcd -cs "segname,AP1,CO1" -rs "segname,AP1" -ls "segname,CO1"  
-pb 0 -gb 1 -ff $molargs2
```

```
# 6. delete file with command (in windows: del xxx, in linux: rm xxx)
del *md.* *run.* *formatted* result.dcd
```

- ✧ 请注意“计算 MM/GBSA”部分的以下参数说明：
  - ◆ -cs <复合物选择集：遵循 VMD 中"atomselect"命令的相同语法规则>
  - ◆ -rs <受体选择集：遵循 VMD 中"atomselect"命令的相同语法规则>
  - ◆ -ls <配体选择集：遵循 VMD 中"atomselect"命令的相同语法规则>
  - ◆ 此处需特别注意：MolAICal 支持将字符串中的逗号,"识别为空格" "的等效符号。例如，"segname,A,C"会与"segname A C"被完全等同处理。
- ✧ 必须确保-cs, -rs 和-ls 参数的选择集定义准确。建议用户理解 VMD 中 AutoPSF 模块的默认段名（segname）命名规则，例如：蛋白质的 A 链会自动生成形如 AP1 的段名。

### 3.3 在 Linux 系统中基于单个蛋白质和配体文件计算 MM/PBSA

- ✧ 对于 APBS：运行以下命令，即可基于 APBS 对 "protein.pdb" 和 "ligand.pdb" 进行 MM/PBSA 计算：

```
#> molaical.exe -call run -c script -n runScript_linux_apbs.sh -i protein.pdb ligand.pdb ligand.str
```

注意：APBS 的 Linux 版本高度依赖系统库的版本，因此在某些情况下可能无法正常运行。若出现此问题，可尝试使用不同版本的 APBS（参见：<https://github.com/Electrostatics/apbs>）。一旦找到合适的版本，可在脚本文件（例如用于 MM/PBSA 计算的 "runScript\_linux\_apbs.sh" 文件，文中第 6 部分）中，在命令行末尾添加 -pbe<您的 APBS 可执行文件路径>，以直接指定 APBS 的可执行程序路径。

- ✧ 对于 DelPhi：

DelPhi 在不同平台上均表现出色。即使在 Linux 环境下，它也能直接运行，无需依赖 Linux 系统库文件。然而，DelPhi 存在许可限制——开发者必须通过其官方网站申请授权后方可下载（<http://compbio.clemson.edu/lab/delphisw>）。若授权审批流程耗时较长，作为应急方案，用户可暂时使用官方教程文档中附带的 DelPhi 可执行文件（文件名为 "delphicpp\_release"，下载地址：[http://compbio.clemson.edu/media/download/mmpbsa\\_example.zip](http://compbio.clemson.edu/media/download/mmpbsa_example.zip)）。

然后运行以下命令，即可基于 DelPhi 对 "protein.pdb" 和 "ligand.pdb" 进行 MM/PBSA 计算：

```
#> molaical.exe -call run -c script -n runScript_linux_delphi.sh -i protein.pdb ligand.pdb ligand.str
```

模板脚本文件 "runScript\_linux\_delphi.sh" 如下：

```
# 1. fix autopsf
molaical.exe -call run -c fixpsf

# 2. generate psf
molaical.exe -call run -c vmdargs -i -pdb $molargs2 -dispdev text -args autopsf autopsf -mol 0 -top $molargs3 -
prefix ligand
```

```

# 3. merge file
molaical.exe -call run -c merge -i -dispdev text -args -first ligand_formatted_autopsf.psf
ligand_formatted_autopsf.pdb -second protein_formatted_autopsf.psf protein_formatted_autopsf.pdb -output
complex_final

# 4. default to run 20 fs minimization and 40 fs MD simulation
molaical.exe -call run -c runnamd -i -dispdev text -args -s complex_final.psf -c complex_final.pdb -nf
md_configure.conf -output_prefix com_md

# 5. Cut last 10 steps for MM/PBSA calculation
molaical.exe -call run -c catdcd -i -dispdev text -args -out result.dcd -first 51 -last 60 -stride 1 com_md.dcd

# 6. Calculate MM/PBSA
molaical.exe -call run -c mmmpbgsa -i -dispdev text -args -s complex_final.psf -c complex_final.pdb -t result.dcd
-cs "segname,AP1,CO1" -rs "segname,AP1" -ls "segname,CO1" -pb 1 -ff $molargs3

# 7. delete file with command (in windows: del xxx, in linux: rm xxx)
rm *md.* *run.* *formatted* result.dcd *final* *_m* *_wb*

```

◇ 请注意"计算 MM/PBSA"部分的以下参数说明:

- ◆ -cs <复合物选择集: 遵循 VMD 中"atomselect"命令的相同语法规则>
- ◆ -rs <受体选择集: 遵循 VMD 中"atomselect"命令的相同语法规则>
- ◆ -ls <配体选择集: 遵循 VMD 中"atomselect"命令的相同语法规则>
- ◆ 此处需特别注意: MolAICal 支持将字符串中的逗号","识别为空格" "的等效符号。  
例如, "segname,A,C"会与"segname A C"被完全等同处理。

◇ 必须确保-cs、-rs 和-ls 参数的选择集定义准确。建议用户理解 VMD 中 AutoPSF 模块的默认段名 (segname) 命名规则, 例如: 蛋白质的 A 链会自动生成形如 AP1 的段名。

### 3.4 在 Linux 系统中基于蛋白质与多肽计算 MM/PBSA 和 MM/GBSA

本教程介绍如何对蛋白质和多肽进行 MM/GBSA 和 MM/PBSA 计算。与小分子化合物不同, 蛋白质和多肽由氨基酸组成, 无需额外的力场参数, 可直接使用默认的 **CHARMM** 力场。这为生物药结合自由能的计算提供了便利。

该方法同样适用于蛋白质-蛋白质复合物、蛋白质-核酸相互作用以及其他生物分子体系的 MM/GBSA 和 MM/PBSA 计算。

进入教程目录:

```
#> cd 021-single_mmmpbgsa/peptide_protein
```

➤ 对于蛋白质和肽的 MM/GBSA 计算, 请运行以下命令:

```
#> molaical.exe -call run -c script -n mmgbgsa_linux.sh -i protein.pdb peptide.pdb PP1
```

"mmgbsa\_linux.sh"文件的内容如下：

```
# 1. fix autopsf
molaical.exe -call run -c fixpsf

# 2. generate psf
molaical.exe -call run -c vmdargs -i -pdb $molargs1 -dispdev text -args autopsf autopsf -mol 0 -prefix protein
molaical.exe -call run -c vmdargs -i -pdb $molargs2 -dispdev text -args autopsf autopsf -mol 0 -prefix ligand

# 3. merge file
molaical.exe -call run -c merge -i -dispdev text -args -first ligand_formatted_autopsf.psf
ligand_formatted_autopsf.pdb -second protein_formatted_autopsf.psf protein_formatted_autopsf.pdb -output
complex_final

# 4. run 2000 fs minimization
molaical.exe -call run -c runnamd -i -dispdev text -args -s complex_final.psf -c complex_final.pdb -minimize
2000 -if_run 1 -output_prefix com

# 5. Cut last 10 steps for MM/GBSA calculation
molaical.exe -call run -c catdcd -i -dispdev text -args -out result.dcd -first 1991 -last 2000 -stride 1 com_md.dcd

# 6. Calculate MM/GBSA
molaical.exe -call run -c mmpbgsa -i -dispdev text -args -s complex_final.psf -c complex_final.pdb -t result.dcd
-cs "protein" -rs "not,segname,$molargs3" -ls "segname,$molargs3" -pb 0 -gb 1

# 7. delete file with command (in windows: del xxx, in linux: rm xxx)
rm *md.* *run.* *formatted* result.dcd
```

- ◆ “-if\_run 1” 表示不进行分子动力学 (MD) 模拟，因为难以获得一个稳定的体系相态。
- ◆ MolAIcal 默认使用广义 Born 隐式溶剂 (Generalized Born Implicit Solvent) 模型进行 MM/GBSA 计算，并提供相应的分子动力学模拟。
- ◆ 本文中进行了 2000 fs 的最小化计算。该数值需由用户根据实际情况评估。此处由于肽段与蛋白质构成的体系较大，因此需要进行 2000 fs 的最小化。
- ◆ 请特别注意以下“计算 MM/GBSA”部分中的参数说明：
  - ◆ -cs <复合物选择：规则与 VMD 中的“atomselect”命令相同>
  - ◆ -rs <受体选择：规则与 VMD 中的“atomselect”命令相同>
  - ◆ -ls <配体选择：规则与 VMD 中的“atomselect”命令相同>
  - ◆ 此处，MolAIcal 支持将字符串中如 “segname,A,C” 的逗号 “,” 视为等同于空格 “ ”，即 “segname,A,C” 会被当作 “segname A C” 处理。
- ◆ 务必确保 -cs、-rs 和 -ls 所指定的选择区域正确无误。用户可参考 VMD 的 AutoPSF 中默认的 segname 命名规则，例如：若为蛋白质的 A 链，则生成的 segname 通常为 “AP1”。

➤ 对于蛋白质和肽的 MM/PBSA 计算，请运行以下命令：

```
#> molaical.exe -call run -c script -n mmpbsa_linux_apbs.sh -i protein.pdb peptide.pdb PP1
```

“mmpbsa\_linux\_apbs.sh”文件的内容如下：

```
# 1. fix autopsf
molaical.exe -call run -c fixpsf

# 2. generate psf
molaical.exe -call run -c vmdargs -i -pdb $molargs1 -dispdev text -args autopsf autopsf -mol 0 -prefix protein
molaical.exe -call run -c vmdargs -i -pdb $molargs2 -dispdev text -args autopsf autopsf -mol 0 -prefix ligand

# 3. merge file
molaical.exe -call run -c merge -i -dispdev text -args -first ligand_formatted_autopsf.psf
ligand_formatted_autopsf.pdb -second protein_formatted_autopsf.psf protein_formatted_autopsf.pdb -output
complex_final

# 4. default to run 60 fs minimization
molaical.exe -call run -c runnamd -i -dispdev text -args -s complex_final.psf -c complex_final.pdb -nf
md_configure.conf -output_prefix com_md

# 5. Cut last 10 steps for MM/PBSA calculation
molaical.exe -call run -c catdcd -i -dispdev text -args -out result.dcd -first 51 -last 60 -stride 1 com_md.dcd

# 6. Calculate MM/PBSA
molaical.exe -call run -c mmpbgsa -i -dispdev text -args -s complex_final.psf -c complex_final.pdb -t result.dcd
-cs "protein" -rs "not,segname,$molargs3" -ls "segname,$molargs3" -pb 2

# 7. delete file with command (in windows: del xxx, in linux: rm xxx)
rm *md.* *run.* *formatted* result.dcd *final.* *_md* *_wb*
```

- ◆ 文件“md\_configure.conf”用作 NAMD 的输入模板配置文件。MolAICal 将自动修改该输入配置文件中的必要参数。
- ◆ 由于 MM/PBSA 计算中不使用广义 Born (Generalized Born) 隐式溶剂模型，这里进行 60 fs 的最小化已足够。
- ◆ 请特别注意以下“计算 MM/PBSA”部分中的参数说明：
  - ◆ -cs <复合物选择：规则与 VMD 中的“atomselect”命令相同>
  - ◆ -rs <受体选择：规则与 VMD 中的“atomselect”命令相同>
  - ◆ -ls <配体选择：规则与 VMD 中的“atomselect”命令相同>
  - ◆ 此处，MolAICal 支持将字符串中如“segname,A,C”的逗号“,”视为等同于空格“ ”，即“segname,A,C”会被当作“segname A C”处理。
- ◆ 务必确保 -cs、-rs 和 -ls 所指定的选择区域正确无误。用户可参考 VMD 的 AutoPSF 中默认的 segname 命名规则，例如：若为蛋白质的 A 链，则生成的 segname 通常为“AP1”。

## 附录 1

在 MolAICal 容器内安装外部程序（推荐，适用于 Linux 版本的 MolAICal）

请注意，Windows 版与 Linux 版的 MolAICal 配置存在差异：Linux 版本采用基于 udocker 容器的技术方案，需在容器内部完成设置，而 Windows 版本则无需此步骤。

### 1. 首先，将文件复制到 MolAICal 容器中

# 进入容器文件系统（将进入 "/root" 目录）

#> molaical.exe -eset shell in

# 将 VMD 和 NAMD 安装包从本地机器复制到容器中，'cp' 命令的第一部分（源路径）

# 位于本地主机，第二部分（目标路径）在容器内：VMD 和 NAMD 软件包可通过以下命令移入容器。

#> cp /home/user/<本地文件> /root/soft

# 退出容器文件系统

#> exit

### 2. 其次，进入 MolAICal 容器的虚拟环境

#> molaical.exe -eset sys run molaical

注：molaical 是容器名称。

3. 在 MolAICal 容器虚拟环境中安装软件的方式与在本地主机上安装相同。以下以安装 VMD 和 NAMD 为例：

#### 1) 安装 NAMD:

解压 NAMD 文件（假设解压后的文件夹名为 namdcpu），然后使用以下命令将其路径告知 MolAICal：

#> molaical.exe -call set -n NAMD -p "/root/soft/namdcpu/namd3"

注：请将上述 VMD 和 NAMD 的路径替换为您系统中的实际路径。-n 后面的 "VMD" 和 "NAMD"（大小写不敏感）是固定的标识符。为确保 MM/GBSA 结果的可重复性，建议使用 NAMD 的 CPU 版本，因为 CUDA 版本中的 seed 参数似乎对结果可重复性无效。

#### 2) 安装 VMD:

按以下步骤操作：

##### ◆ 解压 VMD 文件：

#> tar -xzvf vmd-xxx.tar.gz

注：请将上述路径替换为您系统中的实际路径。

##### ◆ 修改 VMD 解压目录中名为 configure 的文件中的安装路径：

```
# 默认值:  
$install_bin_dir="/usr/local/bin";  
$install_library_dir="/usr/local/lib/$install_name";  
  
# 修改为:  
$install_bin_dir="/root/soft/vmd193/bin";  
$install_library_dir="/root/soft/vmd193/lib/$install_name";
```

◆ 安装 VMD:

```
#> cd vmd-xxx  
#> ./configure LINUXAMD64  
#> cd src  
#> make install
```

注: 请运行 `./configure` 并根据所用计算机选择正确的类型, 此处为 "LINUXAMD64"。

◆ 然后使用以下命令将 VMD 路径告知 MolAIcal:

```
#> molaical.exe -call set -n VMD -p "/root/soft/vmd193/bin/vmd"
```

至此, NAMD 和 VMD 在 MolAIcal 容器内的安装与配置已完成。

✧ 记得使用 `exit` 命令退出 MolAIcal 虚拟环境, 返回本地计算机进行计算 (主要是为了省去文件拷贝步骤; 在容器内运行也可行, 但需手动将数据从本地计算机复制到 MolAIcal 容器中)。

```
#> exit
```

### 3) 保存并加载已修改的容器 (可选)

为保留容器内所做的更改, 并避免日后重新安装软件 (因为一旦容器被删除, 所有数据都会丢失), 可执行以下操作:

✧ 获取容器 ID 和名称:

```
#> molaical.exe -eset sys ps
```

✧ 克隆该容器:

```
#> molaical.exe -eset sys export --clone -o molaicalv2.tar molaical
```

注: `molaical` 是容器名称, 也可使用容器 ID。如果报错, 那可能是因为挂载的文件系统的权限问题, 可以忽略。

✧ 假如需要, 导入克隆的容器:

```
#> molaical.exe -eset sys import --clone --name molaicalv2 molaicalv2.t  
r
```

✧ 将新导入的容器名称更改为默认容器名 "`molaical`", 原容器重命名为 "`bakmolaical`":

```
#> molaical.exe -eset sys rename molaical bakmolaical
#> molaical.exe -eset sys rename molaicalv2 molaical
```

**注意：**容器（动态）是从镜像（静态）生成的。软件安装等操作必须在动态容器中进行；如果克隆了该容器，恢复时仍基于原始底层镜像。

更多详情请参阅 MolAICal 用户手册，或运行以下命令获取帮助：

```
#> molaical.exe -eset sys --help
```