

使用 MolAICal 基于 NAMD 模拟结果计算 小分子和蛋白 MM/GBSA 的教程

作者: MolAICal (update 2021-10-16)

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 文章介绍: <https://arxiv.org/abs/2006.09747> 和
<https://doi.org/10.1093/bib/bbaa161>

MolAICal 中文博客: <https://molaical.github.io/entutorial.html>

MolAICal Twitter: <https://twitter.com/molaical>

MolAICal QQ 学术讨论群: 1151656349

1. 简介

在本教程中介绍了基于 NAMD 的分子动力学模拟结果, 使用 MolAICal 计算小分子和 Mpro 蛋白受体 MM/GBSA 的方法。本教程只是一个简单演示。为了节省运行及存储空间, 本教程仅选择了 Mpro 复合物分子动力学模拟的 25 帧用于计算。

2. 工具

2.1. 所需软件下载地址

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

2) NAMD: <https://www.ks.uiuc.edu/Research/namd/>

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://github.com/MolAICal/tutorials/tree/master/004-MMGBSA>

3. 操作流程

第一部分: 问题的解决

问题: 有些用户使用 MM/GBSA 进行计算得到正值, 如下:

https://www.ks.uiuc.edu/Research/namd/mailling_list/namd-l.2020-2021/1295.html

可能的解决方案：必须检查一下配体是否一直跟受体在同一个周期性盒子中，如果不是，请使用 VMD PBC 将配体和受体裹进同一个周期性盒子中。可以参考：

https://www.ks.uiuc.edu/Research/vmd/script_library/scripts/pbcwrap/

<https://www.ks.uiuc.edu/Research/vmd/plugins/pbctools/>

这里我提供 2 个命令在 VMD Tk Console 中运行：

```
%> pbc wrap -centersel protein -center com -compound chain -all
```

```
%> pbc unwrap -sel "not (water or ions)" -all
```

运行上述命令可以将配体和受体包进同一个周期性盒子。使用下面的命令保存轨迹到 DCD 文件中：

```
%> set all [atomselect top all]
```

```
%> animate write dcd pro.dcd sel $all beg 0 end 24 waitfor all
```

这里所有原子都被选中，用户可以根据需求进行选择。为了简便，本教程只提供 25 帧：从 0 到 24 帧。“pro.dcd”是轨迹名字。解决上述问题之后，可以开始下面的教程。

第二部分：基于 NAMD 轨迹进行 MM/GBSA 的计算

3.1 MM/GBSA 的计算

假设已经有跑平衡的轨迹名为“mpro.dcd”，同时，在“mpro.dcd”轨迹中，保证受体和配体在同一个周期性盒子中。用户可以替换成自己的轨迹。转到以下目录：

```
#> cd 004-MMGBSA
```

3.1.1 提取蛋白与配体复合物的轨迹文件：

```
#> vmd -dispdev text -psf "mpro.psf" -e stripDCD.vmd -args protein,or,resname,LIG "mpro.dcd" "complex" mpro.psf mpro.pdb
```

-args: 其用法类似 VMD 软件中的“atomselect”命令，比如“atomselect top protein or resname LIG”，此处逗号，“”代表空格。其中脚本文件 stripDCD.vmd 可以在本教程材料或 MolAICal 软件的“scripts”目录里面找到。

执行上述命令后生成 complex.psf, complex.pdb 和 complex.dcd 文件。将“GBIS”和“sasa”参数设置为 on。打开并按照下文红色标注内容修改“complex.conf”文件：

```
-----  
structure          complex.psf  
coordinates        complex.pdb  
outputName         complex  
  
paraTypeCharmm     on  
parameters         par_all36_prot.prm  
parameters         par_all36_cgenff.prm
```

```
parameters      ligand.str
parameters      toppar_water_ions.str
```

```
coorfile open dcd complex.dcd
```

本教程中命令在 CPU 上运行。你可以选择 GPU 进行运算。在 Linux 系统下运行 NAMD 命令，如下：

```
#> namd2 +p3 complex.conf >& complex.log &
```

其中符号“&”代表程序在 Linux 系统中进行后台运行，如果你使用的是 Windows 操作系统，请不要用“&”，例如，命令换成这样：

```
#> namd2 +p3 complex.conf > complex.log
```

3.1.2 仅提取蛋白的轨迹文件：

```
#> vmd -dispdev text -psf "mpro.psf" -e stripDCD.vmd -args protein "mpro.dcd" "protein" mpro.psf
mpro.pdb
```

上述命令会生成 protein.psf, protein.pdb 和 protein.dcd。打开“protein.conf”，参考“complex.conf”修改相关参数。

本教程中命令在 CPU 上运行。你可以选择 GPU 进行运算。在 Linux 系统下运行 NAMD 命令，如下：

```
#> namd2 +p3 protein.conf >& protein.log &
```

3.1.3 仅提取配体的轨迹文件：

```
#> vmd -dispdev text -psf "mpro.psf" -e stripDCD.vmd -args resname,LIG "mpro.dcd" "ligand"
mpro.psf mpro.pdb
```

上述命令会生成 ligand.psf, ligand.pdb 和 ligand.dcd。打开“ligand.conf”，参考“complex.conf”修改相关参数。

本教程中命令在 CPU 上运行。你可以选择 GPU 进行运算。在 Linux 系统下运行 NAMD 命令，如下：

```
#> namd2 +p3 ligand.conf >& ligand.log &
```

3.1.4 用 MolAICal 计算 MM/GBSA

```
#> molaical.exe -mmgbsa -c complex.log -r protein.log -l ligand.log
```

输出结果中给出结合自由能 ΔG ：

```
-----
delta E(internal): -0
delta E(electrostatic) + deltaG(sol): 7.7029
delta E(VDW): -44.4361
delta G binding: -36.7332 +/- 3.4202 (kcal/mol)
-----
```

第三部分：残基能量分解

3.2. 残基能量分解

可以参考上面的方法进行残基能量分解，比如：根据报道，SARS-CoV-2 Mpro 的残基 M165 和配体 N3 有相互作用，本教程就以残基 M165 为例，计算残基 M165 的自由能贡献值。首先，切换到教程目录：

```
#> cd 004-MMGBSA/Decompose
```

3.2.1. 提取指定氨基酸和配体的轨迹

```
#> vmd -dispdev text -psf "../mpro.psf" -e ../stripDCD.vmd -args protein,and,resid,165,or,resname,LIG "../mpro.dcd" "res-lig" ../mpro.psf ../mpro.pdb
```

-args: 其用法类似 VMD 软件中的“atomselect”命令，比如“atomselect top protein and resid 165 or resname LIG ”，此处逗号，“代表空格。其中脚本文件 stripDCD.vmd 可以在本教程材料或 MolAICal 软件的“scripts”目录里面找到。

上述命令会生成 res-lig.psf, res-lig.pdb 和 res-lig.dcd。打开“res-lig.conf”，参考“complex.conf”修改相关参数。

本教程中命令在 CPU 上运行。你可以选择 GPU 进行运算。在 Linux 系统下运行 NAMD 命令，如下：

```
#> namd2 +p3 res-lig.conf >& res-lig.log &
```

3.2.2. 仅提取指定氨基酸的轨迹

```
#> vmd -dispdev text -psf "../mpro.psf" -e ../stripDCD.vmd -args protein,and,resid,165 "../mpro.dcd" "res" ../mpro.psf ../mpro.pdb
```

上述命令会生成 res.psf, res.pdb 和 res.dcd。打开“res.conf”，参考“complex.conf”修改相关参数。

本教程中命令在 CPU 上运行。你可以选择 GPU 进行运算。在 Linux 系统下运行 NAMD 命令，如下：

```
#> namd2 +p3 res.conf >& res.log &
```

3.2.3. 仅提取配体的轨迹

```
#> vmd -dispdev text -psf "../mpro.psf" -e ../stripDCD.vmd -args resname,LIG "../mpro.dcd" "lig" ../mpro.psf ../mpro.pdb
```

上述命令会生成 lig.psf, lig.pdb 和 lig.dcd。打开“lig.conf”，参考“complex.conf”修改相关参数。

本教程中命令在 CPU 上运行。你可以选择 GPU 进行运算。在 Linux 系统下运行 NAMD 命令，如下：

```
#> namd2 +p3 lig.conf >& lig.log &
```

3.2.4. 用 MolAICal 计算残基自由能贡献值

```
#> molaical.exe -mmgsa -c res-lig.log -r res.log -l lig.log
```

输出结果中给出残基 M165 的自由能贡献值 ΔG ：

```
-----  
delta E(internal): 0  
delta E(electrostatic) + deltaG(sol): -0.4691  
delta E(VDW): -4.4319  
delta G binding: -4.901 +/- 1.0524 (kcal/mol)  
-----
```

第四部分：熵的计算

3.3. 通过 Carma 和 MolAICal 进行熵的计算

3.3.1. 下载 Carma

Carma 可以在 Windows 或 Linux 操作系统上运行。从 <https://github.com/glykos/carma> or <http://utopia.duth.gr/~glykos/progs> 上下载新版本 Carma V2.01。

推荐使用最新版本的 Carma 进行熵的计算，之前的版本会出现“NaN”的错误等，具体请查看下面的链接：

<https://groups.google.com/g/carma-molecular-dynamics/c/KpyY5sEkj4>

本教程选取 100 帧轨迹进行熵的计算。用户可以根据自己的实际情况选择合适的帧数，越多帧数越耗费计算时间。

3.3.2. 计算复合物的熵

```
#> cd 004-MMGBSA\entropy
```

切换到文件夹“com”。这一步是去掉复合物的旋转和平移：

```
#> carma -v -fit -force -atmid ALLID -segid A -segid C complex.dcd complex.psf
```

这一步是计算熵值：

```
#> carma -v -cov -eigen -mass -force -temp 310 -atmid ALLID -segid A -segid C -last 100
```

```
carma.fitted.dcd complex.psf >& com.log &
```

3.3.3. 计算受体的熵

切换到文件夹“rec”。这一步是去掉受体的旋转和平移：

```
#> carma -force -v -fit -atmid ALLID -segid A protein.dcd protein.psf
```

这一步是计算熵值：

```
#> carma -v -cov -eigen -mass -force -temp 310 -atmid ALLID -segid A -last 100 carma.fitted.dcd protein.psf >& rec.log &
```

3.3.4. 计算配体的熵

切换到文件夹“lig”。这一步是去掉配体的旋转和平移：

```
#> carma -v -fit -force -atmid ALLID -segid C ligand.dcd ligand.psf
```

这一步是计算熵值：

```
#> carma -v -cov -eigen -mass -force -temp 310 -atmid ALLID -segid C -last 100 carma.fitted.dcd ligand.psf >& lig.log &
```

3.3.5. 计算熵值

请检查 log 文件：“com.log”，“rec.log” 和 “lig.log”，并在文件“com.log”，“rec.log” 和 “lig.log” 中找到 Andricioaei 或 Schlitter 的熵值。例如：可以在“com.log”文件中找到 Andricioaei 或 Schlitter 的熵值（见下图）：

```
Writing postscript file carma.fitted.dcd.varcov.ps.
Calculation of eigenvectors and eigenvalues ...
Asking for optimal workspace size : 487662
Starting the calculation ...
Done. Now sorting ...

Entropy calculation will ignore negative eigenvalues !
Entropy (Andricioaei) using only 7220 eigenvalues is 15522.260938 (J/molK)
Entropy (Schlitter) using only 12357 eigenvalues is 10225.524850 (J/molK)
done.
All done in 63.4 minutes.
```

对于 Andricioaei 熵，运行如下命令：

```
#> molaical.exe -entropy -c 15522.260938 -r 15211.880284 -l 1454.409253 -t 310.0
```

这个结果显示：

```
The entropy DELTA S = -84.70646297459386 (kcal/mol)
```

对于 Schlitter 熵，运行如下命令：

```
#> molaical.exe -entropy -c 10225.524850 -r 10028.714754 -l 1465.682932 -t 310.0
```

这个结果显示:

The entropy DELTA S = -93.9502124300494 (kcal/mol)

Where: **-entropy:** 是代表 ΔS 熵值的计算

-c: 复合物的熵值

-r: 受体或第一个分子的熵值

-l: 配体或第二个分子的熵值

-t: 是分子模拟中的开尔文温度