

使用 MolAICal 计算纳米管和蛋白质孔道半径的教程

作者：Qifeng Bai (update 2024-07-20)

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

本教程介绍使用 MolAICal 计算纳米管和蛋白质半径的方法。共分为三个部分：纳米管半径计算，蛋白质孔道半径计算和肽通道半径的计算。最后一个教程是肽通道半径的测量，如果你熟悉 VMD 和 NAMD，可以使用由 CHARMM 力场产生的 PDB 和 PSF 文件来测量肽通道的半径，当然也可以只使用肽段的 PDB 文件进行肽通道半径的测量。

2. 工具

2.1. 所需软件下载地址

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

2) VMD: <https://www.ks.uiuc.edu/Research/vmd>

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/005-radiiCal>

3. 操作流程

3.1. 纳米管半径计算

1) 在 VMD 软件中构建纳米管：Extensions→Modeling→Nanotube Builder (如图 1)。

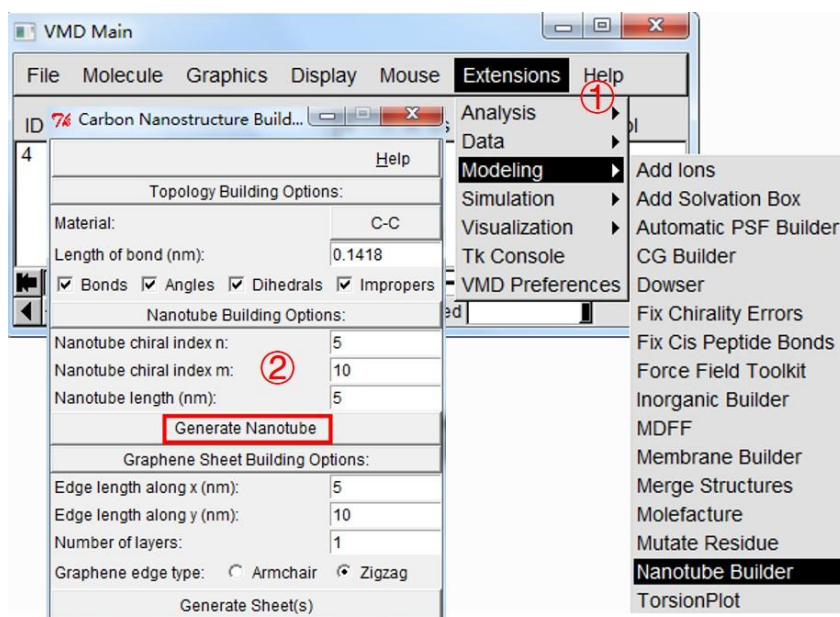


图 1. 构建纳米管

2) 使用类似图 1 的方法打开 VMD Tk Console: Extensions→Tk Console。在 Tk Console 中使用 cd 命令切换到含有文件“nanotube.pdb”和“parameter.dat”的目录，例如：

```
#> cd d:/005-radiiCal/nanotube
```

3) 使用如下命令在 Tk Console 中保存生成的纳米管文件，并命名为“nanotube.pdb”

```
#> set all [atomselect top all]
#> $all writpdb nanotube.pdb
```

4) 选择已构建纳米管内的任意点。你可以在纳米管孔道内表面选择 2 个不同的原子，然后将这两个原子的连线中心作为“cpoint”（见本文件附录 1 中的示例）。在本教程中，选择的点坐标为-0.2015 0.4185 30.147。打开“005-radiiCal\nanotube”文件夹中的“parameter.dat”，将以上所选点坐标添加到“cpoint”。然后按下文所示修改“vector”参数：

```
-----
cpoint      -0.2015 0.4185 30.147
vector      0.00 0.00 1.00
-----
```

“0.00 0.00 1.00”表示沿着 Z 轴方向的半径测量。“0.00 1.00 0.00”表示沿着 Y 轴方向的半径测量。“1.00 0.00 0.00”表示沿着 X 轴方向的半径测量。通道可大致沿任意轴方向放置，即和 vector 的方向大致一致。

5) 在 Windows DOS 或 Linux console 中运行如下命令计算半径：

```
#> molaical.exe -channel radii -cpp parameter.dat
```

命令运行会生成“channel_radii.dat”，“dot.vmd_plot”和“surf.vmd_plot”文件。“dot.vmd_plot”和“surf.vmd_plot”可通过 VMD 软件展示通道表层。类似图 1 的方式打开 VMD Tk Console: Extensions→Tk Console。然后运行以下命令：

```
#> source dot.vmd_plot
```

本教程省略了纳米管卡通图的做法，你可以根据自己的偏好自行设置。你将看到如图 2 所示的通道点曲面：

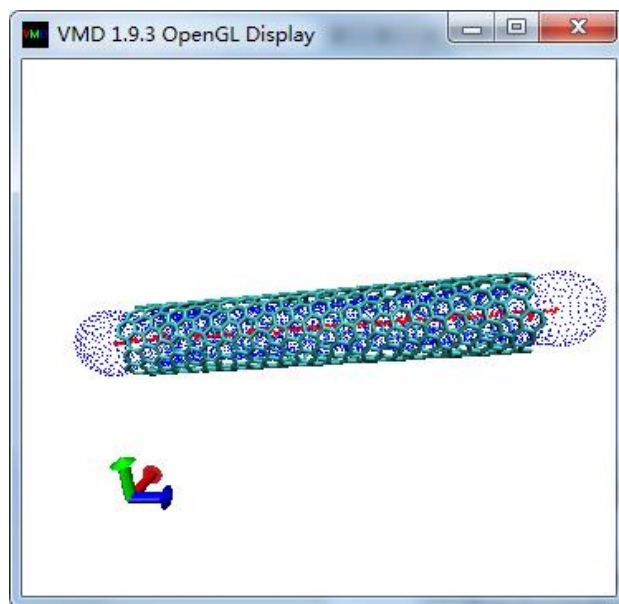


图 2. 纳米管通道表面

文件“channel_radii.dat”包含了反应坐标和半径值。文件“channel_radii.dat”中的第一列是反应坐标，第二列是半径值。可以使用 OriginLab, Microsoft Excel 等工具将其绘制成图。绘制结果如图 3 所示，通过以下命令（如果想自定义图片，请参阅本文档附录 2 中的绘图示例）：

```
#> molaical.exe -call run -c sfile -i m_plot_radii.py -r channel_radii.dat
```

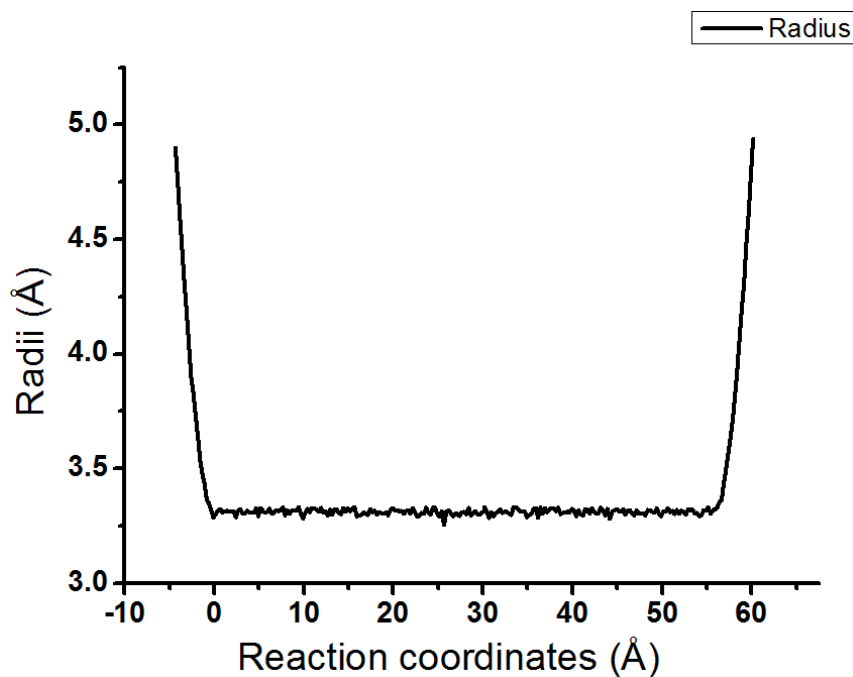


图 3. 半径对反应坐标作图结果

3.2. 蛋白孔道半径计算

转至教程所在目录下：

```
#> cd 005-radiiCal/KcsA
```

在蛋白通道中选择任意点（见本文件附录 1 中的示例），然后在参数文件“parameter.dat”中，将参数“cpoint”设置成为所选点的坐标。按照下文所示设置参数“cpoint”和“vector”：

```
-----  
cpoint    0.001    0.006    1.927  
vector    0.00 0.00 1.00  
-----
```

“0.00 0.00 1.00”表示沿着 Z 轴方向的半径测量。“0.00 1.00 0.00”表示沿着 Y 轴方向的半径测量。“1.00 0.00 0.00”表示沿着 X 轴方向的半径测量。通道可大致沿任意轴方向放置，即和 vector 的方向大致一致。

1) 在 Windows DOS 或 Linux console 中运行如下命令计算半径：

```
#> molaical.exe -channel radii -cpp parameter.dat
```

2) 本运算也会生成“channel_radii.dat”，“dot.vmd_plot”和“surf.vmd_plot”文件。类似图 1 的方式打开 VMD Tk Console：Extensions→Tk Console。然后运行以下命令：

```
#> mol load pdb KcsA.pdb
```

```
#> source dot.vmd_plot
```

本教程省略了蛋白卡通图的做法，你可以根据自己的偏好自行设置。你将看到图 4 所示点曲面：

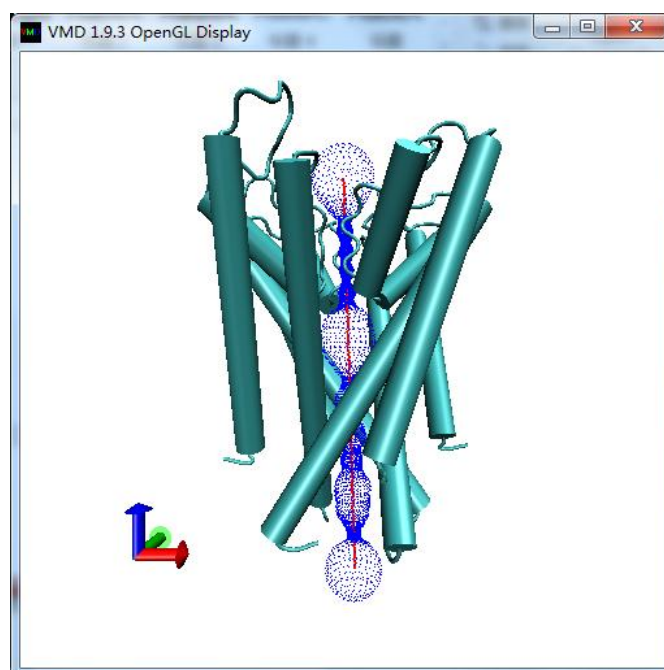


图 4. 蛋白通道的点曲面

文件“channel_radii.dat”包含了反应坐标和半径值。文件“channel_radii.dat”中的第一列是反应坐标，第二列是半径值。可以使用 OriginLab, Microsoft Excel 等工具将其绘制成图。半径绘制如图 5 所示，通过以下命令（如果想自定义图片，请参阅本文件附录 2 中的绘图示例）：

```
#> molaical.exe -call run -c sfile -i m_plot_radii.py -r channel_radii.dat
```

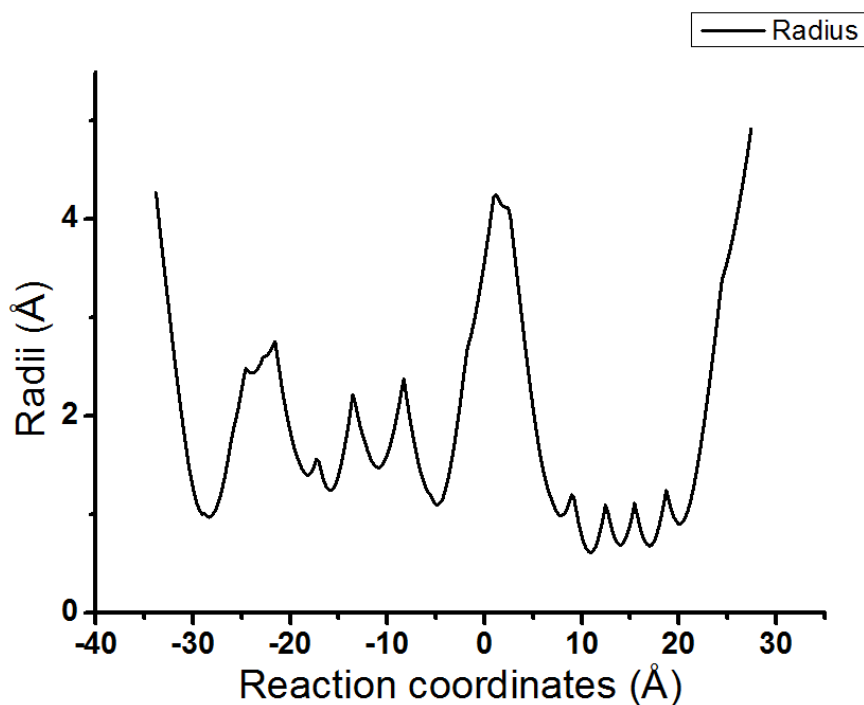


图 5. 半径对反应坐标作图结果

注意事项：文件“parameter.dat”中的参数“conpar”是一个控制参数，参数“conpar”的默认值是 0.15，其数值的增加可以提升随机测量的概率。在这种情况下，有可能出现一些奇怪的测量路线。如果你的蛋白孔道比较规则且大致沿着 X, Y, Z 轴的某一个方向，你可以减少参数“conpar”的数值，比如设置成 0.04，但是参数“conpar”不能设置成 0，这样你就能得到较为规整的测量通道。

3.3. 半径计算的高级教程

本部分示例利用由 CHARMM 力场产生的 PDB 和 PSF 文件计算多肽孔道半径。转至教程所在目录：

```
#> cd 005-radiiCal/GramicidinA
```

选择多肽孔道中的任意点（见本文件附录 1 中的示例）。将参数“cpoint”设置为所选任意点的坐标。按照下文所示设置参数“pdbpath”，“psfpath”，“cpoint”和“vector”：

```
-----
pdbpath    1JNO.pdb
psfpath    1JNO.psf
cpoint     0.1625 -0.629 -1.838
vector     0.00 0.00 1.00
-----
```

“0.00 0.00 1.00”表示沿着 Z 轴方向的半径测量。“0.00 1.00 0.00”表示沿着 Y 轴方向的半径测量。“1.00 0.00 0.00”表示沿着 X 轴方向的半径测量。通道应大致沿任意轴方向放置，即和 vector 的方向大致一致。

1) 在 Windows DOS 或 Linux console 中运行如下命令计算半径：

```
#> molaical.exe -channel radii -cpp parameter.dat -fc charmm
```

2) 本次运算也会生成 “channel_radii.dat”, “dot.vmd_plot” 和 “surf.vmd_plot” 文件。类似图 1 的方式打开 VMD Tk Console: Extensions→Tk Console。运行以下命令:

```
#> mol load pdb 1JNO.pdb
```

```
#> source surf.vmd_plot
```

本教程省略了多肽卡通图的做法, 你可以根据自己的偏好自行设置。你将看到图 6 所示多肽通道 (如图 6 所示):

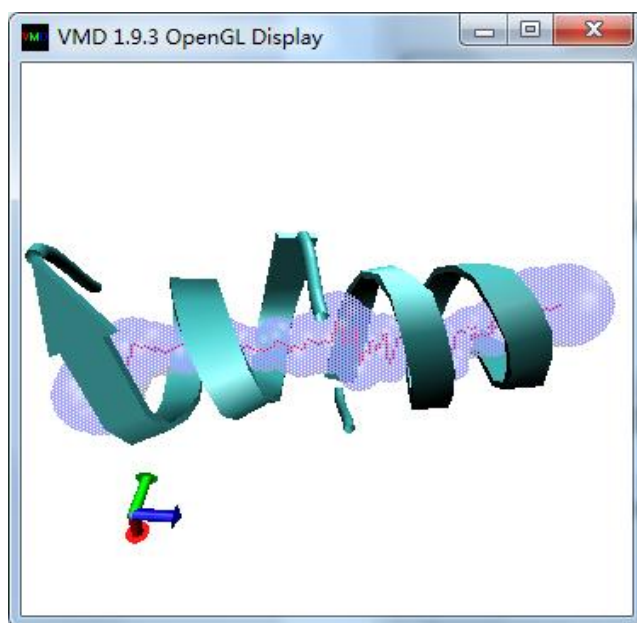


图 6. 多肽通道

文件 “channel_radii.dat” 包含了反应坐标和半径值。文件 “channel_radii.dat” 中的第一列是反应坐标, 第二列是半径值。可以使用 OriginLab, Microsoft Excel 等工具将其绘制成图。半径绘制如图 7 所示, 通过以下命令 (如果想自定义图片, 请参阅本文档附录 2 中的绘图示例):

```
#> molaical.exe -call run -c sfile -i m_plot_radii.py -r channel_radii.dat
```

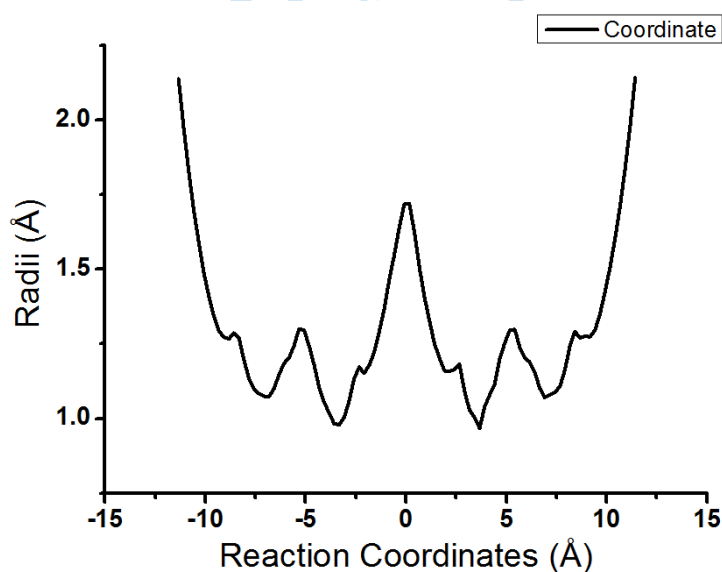


图 7. 半径对反应坐标作图结果

附录 1 选择孔道中的任意点

如何获取通道中的坐标？以“3.2. 蛋白孔道半径计算”为例：

#> cd 005-radiiCal/KcsA

1. 打开 vmd 并加载“KcsA.pdb”，如图 8 所示

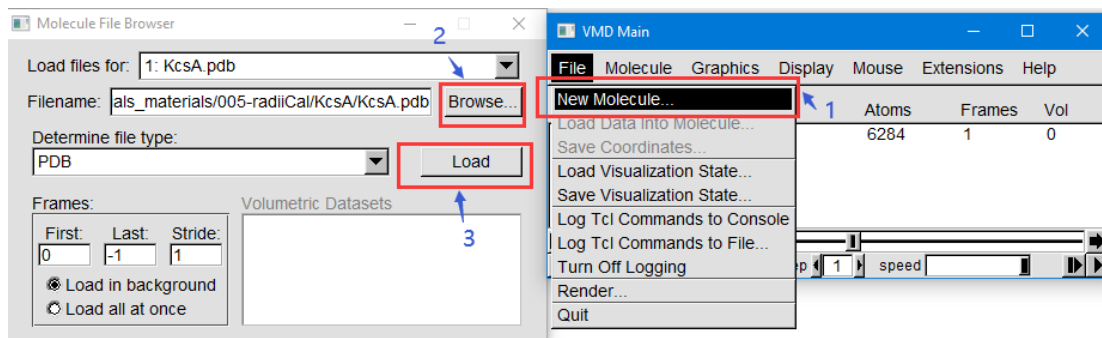


图 8

2. 将蛋白质显示成卡通模式（如图 9 所示），这只是为了更好地观察，如果用户可以轻松操作蛋白质，则可以省略此步骤。

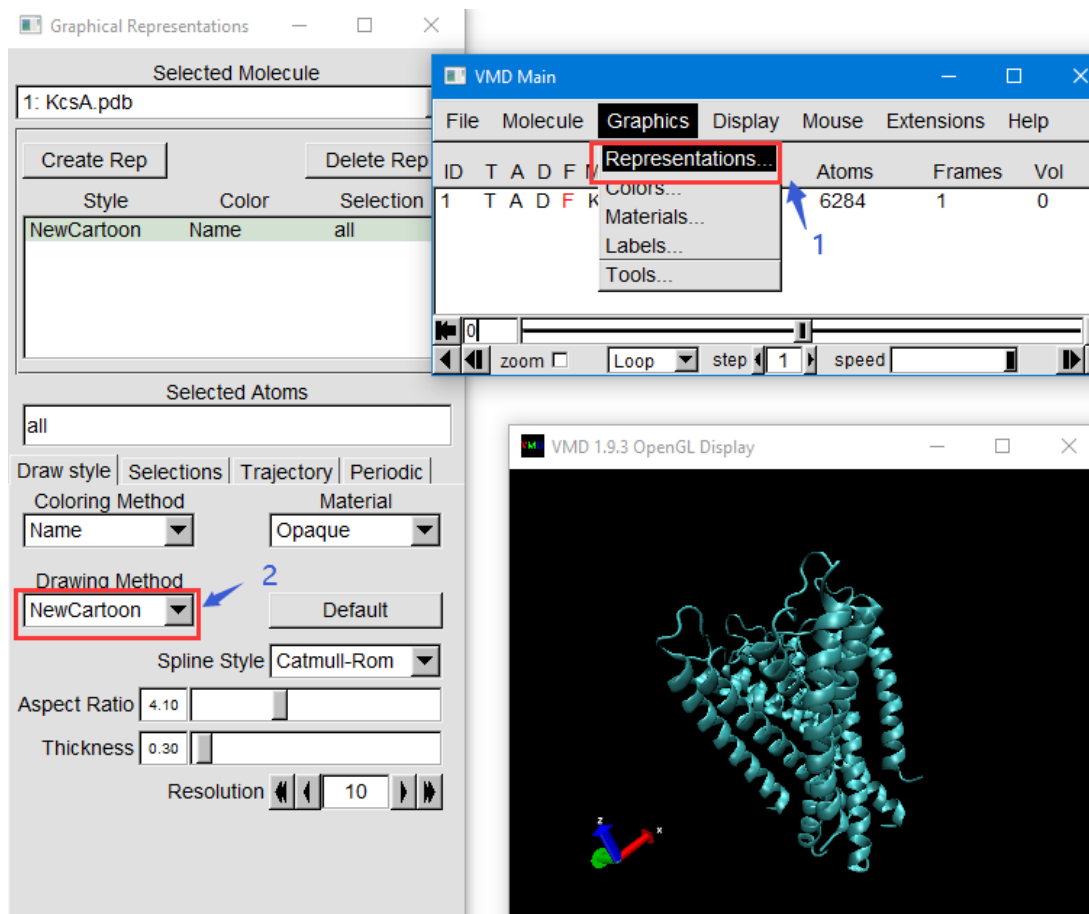


图 9

3. 按下键盘上的字符“r”，它将在 VMD 中旋转对象；按键盘上的数字“1”，在 VMD 中点击一个原子，它将选中这个原子；按键盘上的数字“2”，单击原子 1，然后单击原子 2，它将生

成一条连接线。在此步骤中，按数字“2”，然后在 VMD 中依次单击通道内的两个原子，如图 10 所示。显然，图 10 中两个原子的连接线大部分都在通道内。

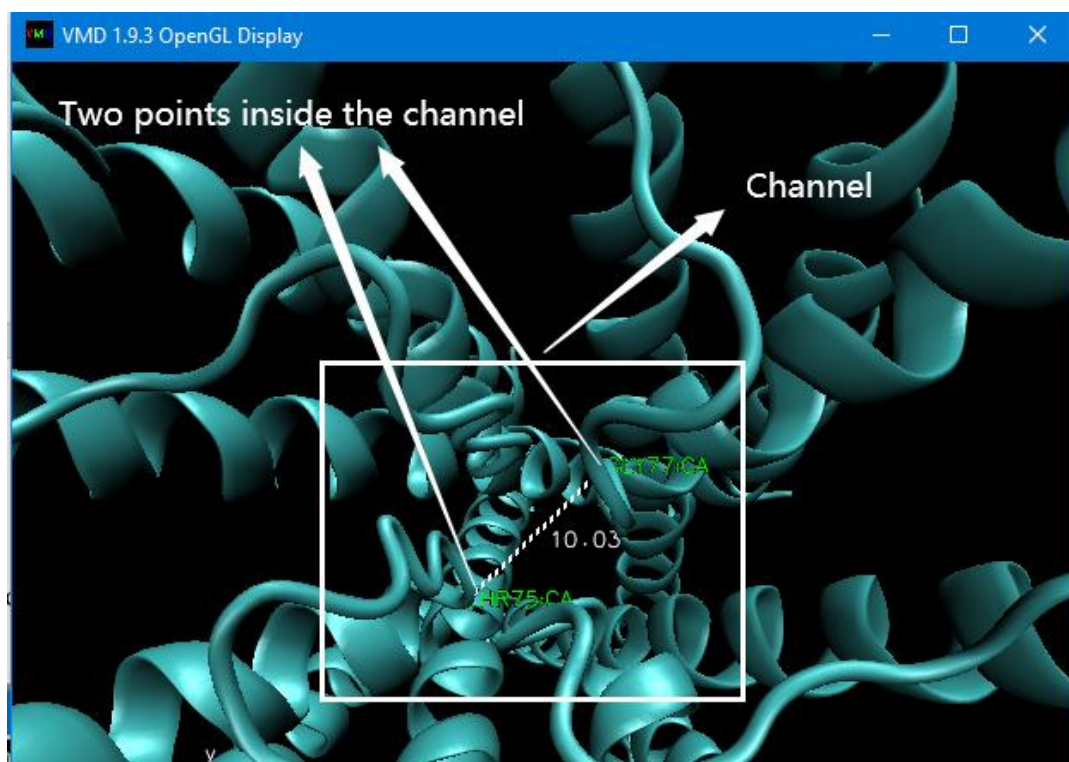


图 10

4. 用户可以通过“Graphics→Labels→Atoms→XYZ”获得所选原子的坐标（见图 11）

图 11

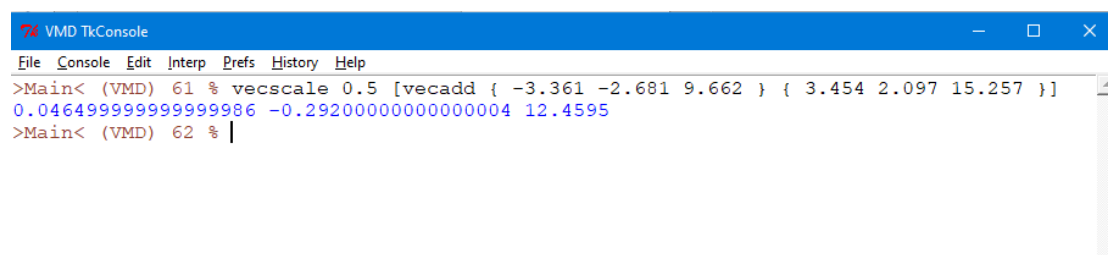
这将获得 2 个孔道内的坐标：

Coordinates of atom 1: -3.361 -2.681 9.662
 Coordinates of atom 2: 3.454 2.097 15.257

显然，图 10 中两个原子的连接线大部分都在通道内。因此，这里选择：原子 1 和 2 之间的中心点 = $(\text{Coordinates of atom 1} + \text{Coordinates of atom 2}) / 2 = \{[-3.361, -2.681, 9.662] + [3.454, 2.097, 15.257]\} / 2 = 0.0465, -0.292, 12.4595 \rightarrow$ 这是一个选择通道或孔道中任意点的实例。

此外，用户可以使用 VMD Tkconsole 按以下命令计算原子 1 和 2 之间的中心点，然后按键盘上的“Enter”键：

```
#> vecscale 0.5 [vecadd { -3.361 -2.681 9.662 } { 3.454 2.097 15.257 }]
```



附录 2 在 MolAICal 中的半径绘图

为了在 MolAICal 中作图，转到虚拟环境：

在 Windows 系统中，打开 DOS，输入以下命令并按“回车键”：

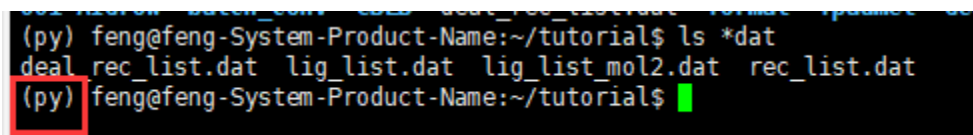
```
#> D:\MolAICal-win64\mtools\py\Scripts\activate.bat
```

在 Linux 系统中，打开 Linux 终端，输入以下命令并按“回车键”：

```
#> source /home/feng/tutorial/MolAICal-linux64/mtools/py/bin/activate
```

注意：请将“D:\MolAICal-win64”或“/home/feng/tutorial/MolAICal-linux64”替换为您的真实 MolAICal 路径。

如果您进入了虚拟环境，您将看到类似于以下的截图：



转入到文件夹“005-radiiCal/KcsA”中：

```
#> cd 005-radiiCal/KcsA
```

然后，输入以下命令绘制并保存孔道半径图（见图 12）：

```
#> python plot_radii.py
```

注意：如果用户熟悉 Matplotlib (<https://matplotlib.org>)，可以根据具体的作图要求修改文件“plot_radii.py”。

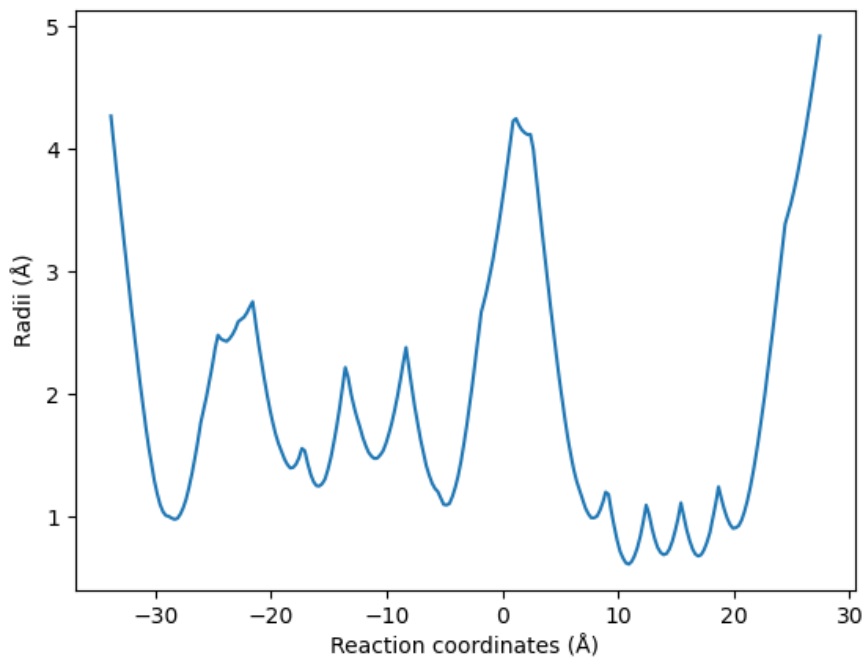


图 12