

Mopac & KrakenX Simple Manual

1. Requirements

1.1 Operating System

全流程操作：建议使用Windows和Linux.

MacOS在大部分场景下能够运行，但是Multiwfn的作者卢天（sobereva）并没有提供MacOS的二进制文件，需要自己编译。

1.2 Environments

需要环境：Java

检测方法：在命令行中输入

```
java -version
```

然后应该得到类似：

```
openjdk version "21.0.2" 2024-01-16
OpenJDK Runtime Environment Homebrew (build 21.0.2)
OpenJDK 64-Bit Server VM Homebrew (build 21.0.2, mixed mode, sharing)
```

的输出。我这里是通过homebrew在MacOS上安装的java，其他操作系统和安装方式安装的java可能不尽相同，但是肯定需要openjdk，如果输出报错，则需要检查java是否安装成功，是否已经添加到环境变量中。

1.3 Softwares

[git](#)

[Multiwfn](#)

任意一个文本编辑器，记事本可以，但是推荐[Visual Studio Code](#)

[mopac](#)

[PyMol](#)

[KRAKENX](#)

1.3.* Software Installation

git的安装教程很多，直接百度搜索git 安装教程 即可。

安装完成后，在命令行输入：

```
git --version
```

应该能得到当前Git的版本号。我现在的版本是Git version 2.39.3（好像挺久没更新了）

Multiwfn提供了二进制文件下载，适用于Windows和Linux，点进网站下就行。这个软件不是双击运行的，先在终端中进入你下载它的路径，然后./Multiwfn.exe 来运行它。如果你想在任意地方运行，请在系统的环境变量中加入Multiwfn的路径。详情请自行查询计算化学论坛和sobereva的博客，他写得挺详细的。

VSCode在安装完成之后，进入“扩展”面板，输入Chinese，先安装简体中文语言包，然后安装Protein Viewer和md-highlighter（如果你用的是浅色主题）或者gmxhelper（如果用的深色主题），这几个插件可以帮助你方便地查看pdb文件。

mopac也提供了二进制文件，在GitHub页面右边的Release页面里面下载。走完安装流程，mopac应该会自动加入系统的环境变量，无需手动配置。如果

```
mopac --version
```

报错，那和Multiwfn一样手动加一下环境变量。

PyMol需要自己申请。上面链接给了教育用途的申请入口。

Registration For Educational-Use-Only PyMOL Builds

Schrödinger offers **Educational-use-only** PyMOL builds available at no cost to **teachers and high school and college students** (including online courses, homeschooling, etc.) for classroom instruction, homework assignments, and to provide a means for creating high quality figures. Please note that it is not provided for the purposes of academic research or publication.

-> [FAQ \(Frequently Asked Questions\)](#)

The Educational-use-only PyMOL builds are provided "AS IS" with no obligation to grant download access, fix bugs, furnish updates, provide documentation, or meet any other need related to the educational-use PyMOL builds.

If you intend to use PyMOL products for academic research or publication, please purchase an Academic PyMOL subscription, which includes access to technical support, screencasts, and additional resources. See <http://pymol.org/academic>.

I am a:	<input type="text"/>
Your First Name:	<input type="text"/>
Your Last Name:	<input type="text"/>
Your Email Address:	<input type="text"/>
Your Telephone Number:	<input type="text"/>
Institution:	<input type="text"/>
Comments (optional):	<input type="text"/>
<input type="button" value="Continue"/>	

把这个表填完，Schrodinger会给你发个邮件，里面有你的License和下载链接。之后就按照安装程序给的提示走就行。

KRAKENX需要自己编译。考虑到今后看到这个教程的可能是没有太多计算机经验的同学，在这里简单进行一点说明。

二进制文件：这是机器唯一看得懂的东西，是一堆二进制（0和1）组成的，在Windows上是.exe文件，在Linux和MacOS上可能没有后缀名，但是应该有一个长得像终端的图标。如果你使用纯命令行，你应该不需要看这一段话。

编译(Compile)：将高级语言（比如C语言、C++、Fortran等）翻译成机器语言的过程。

构建(Build)：将编译后的代码和其他可能用到的资源链接、打包起来，生成最后的包或者可执行文件，也就是.jar包或者.exe程序。

编译和构建通常一起完成。

如果你使用的是Python，这个语言不需要经过编译，因为它是一种解释型语言，直接在你的IDE（比如PyCharm，VSCode）或者交互面板（比如Jupyter Notebook）里运行就行。

首先在命令行里运行：

```
git clone https://gitlab.com/vishsoft/krakenx
```

之后，这个网页里的文件会出现在你运行git的当前路径下。如果是新开的命令行窗口，路径应该是C:\Users\xxx(Windows)或/home/xxx(Linux)或者/Users/xxx(MacOS和其他Unix-Liked系统)。

之后，

```
cd krakenx/build
```

进入krakenx的路径。你可以随时输入ls来看当前目录下有什么文件，或者输入pwd来看自己在哪个路径里面。

随后，输入

```
./build.sh
```

没有报错的话，build文件夹应该会出现一个叫KrakenX.jar的包。到这里KrakenX构建完成。

2. Usage

2.1 Multiwfn创建MOPAC输入文件

使用方法：先准备好需要分析的分子的.pdb文件，这里我假定为xmolecule.pdb（根据实际的文件名修改）。如果没有pdb，可以自己用GaussView、Material Studio等建模或者到[t3db](#)上下载。

pdb文件的例子如下：

TITLE	etscn-opt-charge						
REMARK	1 File created by GaussView 5.0.9						
HETATM	1	C	0	1.279	-0.204	0.517	C
HETATM	2	S	0	-0.230	-1.027	-0.186	S
HETATM	3	C	0	-1.363	0.222	0.007	C
HETATM	4	N	0	-2.161	1.055	0.124	N

```

HETATM      5  H          0      2.010 -1.015  0.512      H
HETATM      6  H          0      1.059  0.047  1.555      H
HETATM      7  C          0      1.762  0.998 -0.280      C
HETATM      8  H          0      1.013  1.792 -0.293      H
HETATM      9  H          0      1.994  0.725 -1.312      H
HETATM     10  H          0      2.669  1.400  0.181      H
HETATM     11  C          0     -1.842  0.721  0.077      C
END
CONNECT      1      2      5      6      7
CONNECT      2      1      3
CONNECT      3      2      4
CONNECT      4      3
CONNECT      5      1
CONNECT      6      1
CONNECT      7      1      8      9     10
CONNECT      8      7
CONNECT      9      7
CONNECT     10      7

```

随后，进入能运行Multiwfn的地方，在命令行输入：

```
multiwfn /path/to/xmolecule.pdb
```

之后multiwfn会甩出一大串输出，就像：

```

Multiwfn -- A Multifunctional Wavefunction Analyzer
Version 3.8(dev), release date: 2024-Apr-17
Developer: Tian Lu (Beijing Kein Research Center for Natural Sciences)
Below paper ***MUST BE CITED IN MAIN TEXT*** if Multiwfn is used in your work:
      Tian Lu, Feiwu Chen, J. Comput. Chem., 33, 580-592 (2012)
See "How to cite Multiwfn.pdf" in Multiwfn binary package for more information
Multiwfn official website: http://sobereva.com/multiwfn
Multiwfn English forum: http://sobereva.com/wfnbbs
Multiwfn Chinese forum: http://bbs.keinsci.com/wfn

( Number of parallel threads:   4   Current date: 2024-04-22   Time: 14:43:53 )

Please wait...
Totally      12 atoms

Loaded mi.pdb successfully!

Formula: H6 C4 N2      Total atoms:      12
Molecule weight:      82.10399 Da
Point group: Cs

"q": Exit program gracefully      "r": Load a new file
      ***** Main function menu *****
0 Show molecular structure and view orbitals
1 Output all properties at a point      2 Topology analysis
3 Output and plot specific property in a line

```

```

4 Output and plot specific property in a plane
5 Output and plot specific property within a spatial region (calc. grid data)
6 Check & modify wavefunction
7 Population analysis and calculation of atomic charges
8 Orbital composition analysis          9 Bond order analysis
10 Plot total DOS, partial DOS, OPDOS, local DOS and photoelectron spectrum
11 Plot IR/Raman/UV-Vis/ECD/VCD/ROA/NMR spectrum
12 Quantitative analysis of molecular surface
13 Process grid data (No grid data is presented currently)
14 Adaptive natural density partitioning (AdNDP) analysis
15 Fuzzy atomic space analysis
16 Charge decomposition analysis (CDA) and plot orbital interaction diagram
17 Basin analysis                      18 Electron excitation analysis
19 Orbital localization analysis       20 Visual study of weak interaction
21 Energy decomposition analysis       22 Conceptual DFT (CDFT) analysis
23 ETS-NOCV analysis                  24 (Hyper)polarizability analysis
25 Electron delocalization and aromaticity analyses
26 Structure and geometry related analyses
100 Other functions (Part 1)          200 Other functions (Part 2)
300 Other functions (Part 3)

```

请记得引用流程中用到的所有软件的对应论文，没论文给个链接也行。

进入功能100->2->14，直接按回车（或者自定义你想要的名字），再随便选一个方法（比如PM6，就输入1），回车，之后你的路径下就多出了一个xmolcule.mop文件。

2.2 编辑和执行.mop文件

用随便哪个文本编辑器打开.mop文件，会看到如下内容：

```

PM6 PRNT=2 precise charge=0
molecule
All coordinates are Cartesian
N      0.61000000 1      0.01600000 1      0.00000000 1
C     -0.22800000 1      1.11300000 1      0.00000000 1
C     -1.50300000 1      0.60700000 1      0.00000000 1
N     -1.47400000 1     -0.76800000 1      0.00000000 1
C     -0.19900000 1     -1.08600000 1      0.00000000 1
C      2.06400000 1      0.03200000 1      0.00000000 1
H      0.15300000 1      2.12200000 1      0.00000000 1
H     -2.43400000 1      1.15200000 1      0.00000000 1
H      0.20400000 1     -2.08800000 1      0.00000000 1
H      2.44500000 1      0.53700000 1      0.89000000 1
H      2.44500000 1      0.53700000 1     -0.89000000 1
H      2.42700000 1     -0.99600000 1      0.00000000 1

```

把第一行改成想要执行的方法，并且在文件最后继续加入想执行的方法。这里给一个完成的.mop的例子：

```

PM6 PRECISE POLAR MMOK XYZ ENPART STATIC SUPER BONDS LARGE PRTXYZ DISP ALLVEC CYCLES =8000
bmi
All coordinates are Cartesian

```

```

N      0.72700000 1    -0.29000000 1     0.44300000 1
C      1.64600000 1    -0.96100000 1    -0.33900000 1
C      2.62400000 1    -0.04500000 1    -0.63500000 1
N      2.33500000 1     1.16900000 1    -0.05900000 1
C      1.19800000 1     0.98700000 1     0.57500000 1
C     -0.51400000 1    -0.82900000 1     0.99500000 1
H      1.52600000 1    -1.99900000 1    -0.60000000 1
H      3.51700000 1    -0.19300000 1    -1.22200000 1
H      0.66900000 1     1.73300000 1     1.15000000 1
H     -0.27800000 1    -1.73900000 1     1.55400000 1
C     -1.57800000 1    -1.13400000 1    -0.07000000 1
H     -0.88400000 1    -0.10000000 1     1.71800000 1
H     -1.17900000 1    -1.88600000 1    -0.75800000 1
C     -2.06500000 1     0.08100000 1    -0.87300000 1
H     -2.42800000 1    -1.60300000 1     0.44000000 1
H     -2.69500000 1    -0.28300000 1    -1.69200000 1
H     -1.20700000 1     0.57000000 1    -1.34700000 1
C     -2.85700000 1     1.10500000 1    -0.05300000 1
H     -3.73200000 1     0.64300000 1     0.41800000 1
H     -3.21400000 1     1.92000000 1    -0.68700000 1
H     -2.25400000 1     1.55500000 1     0.74000000 1

```

PM6 OLDGEO FORCE THERMO

保存，然后，在命令行输入：

```
mopac xmolecule.mop
```

计算应该在几秒内完成（如果分子小的话），如果分子很大（比如是个很大的体系或者蛋白质），那就打开任务管理器，看看CPU是否被完全吃满。mopac的多线程能力很强，能吃掉基本所有的算力。

完成后，路径中应该会存在：

```

xmolecule.arc
xmolecule.mop
xmolecule.out
xmolecule.pol

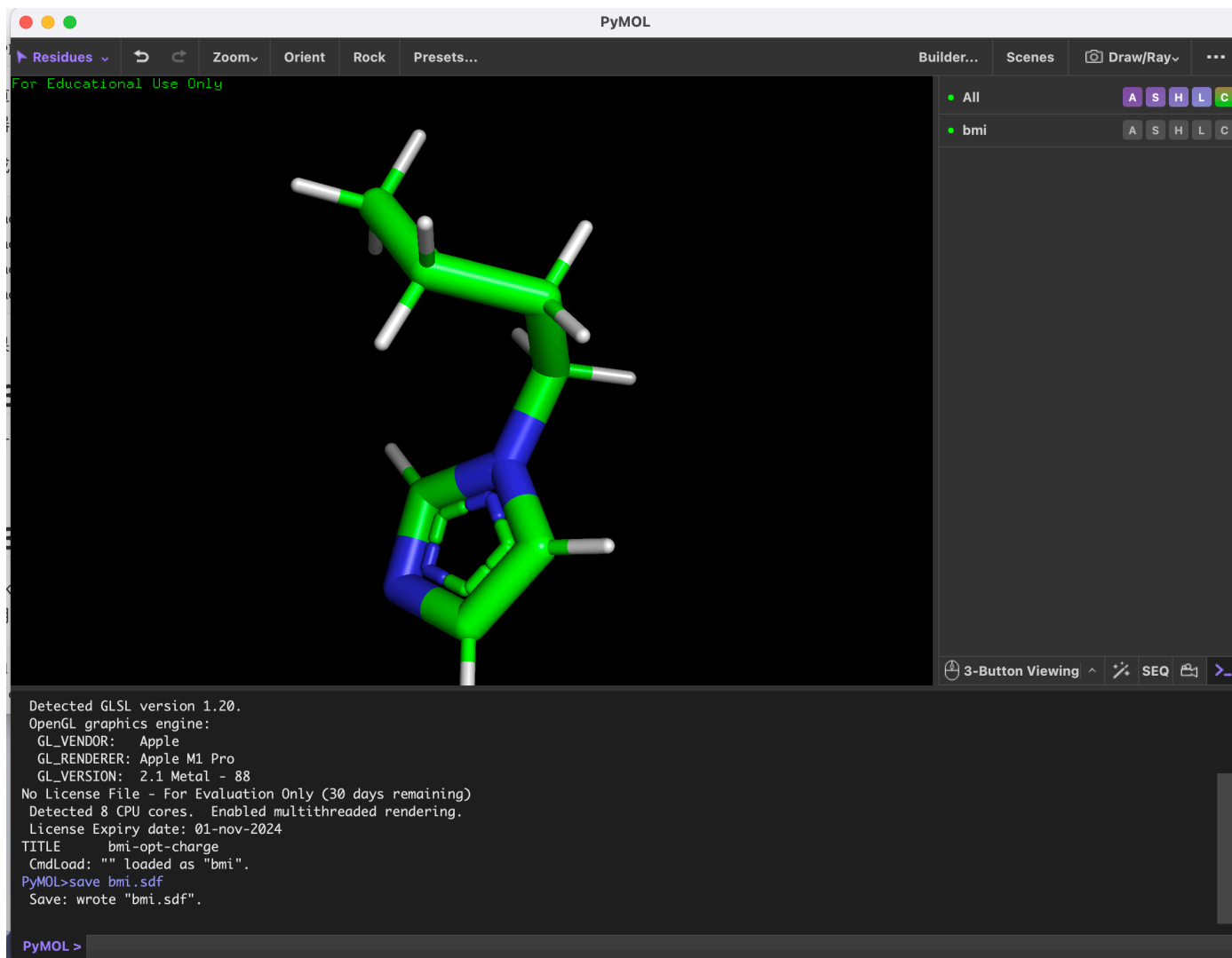
```

如果只有.out文件，那就是报错了，自己看去。

2.3 PyMol把pdb转换成Krakenx能处理的sdf

先打开PyMol，打开xmolecule.pdb文件，在最下面的命令行里输入save xmolecule.sdf，即可完成转换。

如下图：



2.4 Krakenx计算

KrakenX的链接里给出了计算脚本样例，论文里也写了怎么写脚本。如果用的是example文件夹里的caldesc.sh，使用方法如下：

```
cd /somewhere/to/the/script
```

```
./caldesc.sh /somewhere/to/save/sdf /somewhere/to/save/mopdata [prefix]
```

这么写很难理解。给个例子：

```
./caldesc.sh /Users/liz/Documents/Repos/graduate-dissertation/f-file/mop/bmi  
/Users/liz/Documents/Repos/graduate-dissertation/f-file/mop/bmi bmi
```

这里，bmi是前缀，/Users/liz/Documents/Repos/graduate-dissertation/f-file/mop/bmi 是我存放我的x molecule.mop的路径。

之后，应该可以在example的路径里看到一个.txt，里面写着KrakenX计算的描述符。