

# 小工具手册<sup>\*</sup>

作者

2016-08-19

<sup>\*</sup>手册简单地修改于 GitHub 上的 Pro Git Book (progit) 项目模板，从而利用 Pandoc 完成轻量级 Markdown 文稿输出 PDF 手册。相关术语和细节，请参考正文部分。

# 目录

<b>1 引言</b>	<b>1</b>
<b>2 Linux 小工具</b>	<b>2</b>
2.1 编辑	2
2.1.1 编辑器	2
Vim	2
Emacs	2
2.1.2 CLI 编辑	2
正则表达式	2
awk, sed	3
2.2 Shell	3
2.3 脚本	3
2.4 网络	3
2.4.1 SSH	4
2.4.2 FTP, SFTP	4
<b>3 化学小工具</b>	<b>5</b>
3.1 Avogadro	5
3.2 Molden	5
3.3 wxMacMolPlt	5
3.4 Multiwfn	5
3.5 Open Babel	5
<b>4 其他小工具</b>	<b>6</b>
4.1 平面图形图像	6
4.1.1 Imagemagick	6
4.1.2 GIMP	6
4.1.3 Inkscape	6
4.2 3D 模型	7
4.2.1 PovRay	7
4.2.2 FreeWRL, Meshlab	7
4.2.3 Blender	7
4.3 科学软件	7
4.3.1 SciDAVis	7
4.3.2 Gnuplot	7
4.3.3 Sage	7

4.3.4	R	7
4.4	文献管理	8
4.5	TeX 相关	8
4.5.1	Asymptote	8
	思路一	8
	思路二	8
4.5.2	TeXmacs	8
4.6	其他	9
4.6.1	版本控制与 Git	9
4.6.2	脚本思维与 Python	9
4.6.3	轻量编辑与 Markdown	9
<b>5</b>	<b>个人小工具</b>	<b>10</b>
5.1	几何结构变换	10
5.1.1	单分子 XYZ 坐标变换	10
	实现方式和原理	10
	用法示例	11
5.1.2	分子间的几何结构线性插值	11
	实现方式和原理	11
	用法示例	12
5.2	IRD	12
	实现方式和原理	12
	用法示例	12
5.3	能量优化	13
	实现方式和原理	13
	用法示例	13
5.4	ci vector 系数统计	13
	实现方式和原理	13
	用法示例	13



# 第1章

## 引言

这篇文档主要是基于个人经验对自己了解的一些小工具作一下简单介绍，所涉及内容原则上均属于自由的。

文档版本维护于 GitHub 上的 [UM \(Utitlies Manual\)](#) 这个项目中。方便大家参考和学习交流。

## 第2章

# Linux 小工具

本章简单介绍一下常用的 Linux 小工具。主要目的在于对工具的介绍，不涉及过多 Linux 的知识和命令。关于 Linux 知识和命令，可以参考网络上的介绍（比如[鸟哥的 Linux 私房菜](#)）或者任何 Linux 参考书籍。

## 2.1 编辑

Linux 是基于文本文件的系统，这里编辑主要是对文本的操作。

### 2.1.1 编辑器

文本编辑器是最常用的用来进行文本编辑的工具。最有名的是 Vim 和 Emacs。黑客界两大编辑器阵营因此发生过[圣战](#)。

#### Vim

[Vim](#) 即 Vi IMproved 的缩写。一般 vi 作为 Linux 的缺省编辑器使用。Vim 作为一个强大的编辑器，学习曲线很陡，有大量的快捷键和三种编辑模式。

Vim 的技巧我推荐参考相关书籍，比方说 [Learning the vi and Vim Editors](#) 这本（O’ Reilly 的“动物书”之一，我自己读的是影印版），或者直接在互联网上搜集相关技术信息（这是学习技术的最简单有效的方式）。

#### Emacs

[Emacs](#) 我自己用得很少，但同样是一款非常强大的通用编辑器。它是 [GNU](#) 的精神领袖 Richard Stallman 的作品，后面要介绍的 TeXmacs 深受 Emacs 的启发。

同样，想深入学习 Emacs 的话，还是可以自己找相关参考书或利用网络。

另外，图 2.1 是一个关于编辑器的学习曲线的幽默图片，vi 和 Emacs 也在其中。

### 2.1.2 CLI 编辑

#### 正则表达式

正则表达式(Regular Expression, RE)有必要在 awk 和 sed 之前介绍，因为这是在 \*nix 系统中一个非常重要的概念（虽然不仅仅如此）。

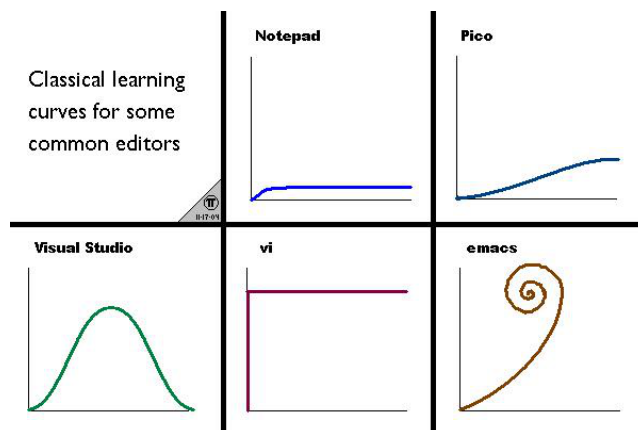


图 2.1: 编辑器学习曲线对比 (图片来源于网络)

正则表达式简单看就是用来匹配查找的一套元字符(metacharacter)集合, 和 \*nix 系统里的通配符类似。总之, 利用好 RE, 可以大大提高文本处理效率。同样, 详细资料可以通过[书籍](#)和[互联网](#)获取, 在此仅抛砖引玉。

### awk, sed

这是两个典型代表。sed 即 **s**tream **e**ditor, awk 取自其开发者姓名 Alfred **A**ho, Peter **W**einberger, 和 Brian **K**ernighan。两者都是通过 RE 进行匹配后, 再进行某些编辑操作, 其中 awk 功能更为强大, 甚至可以当作一门程序语言使用, 语法和 C 语言类似。

除此之外, 常用的 grep 即 **g**lobally search a **r**egular **e**xpression and **p**rint, 也是利用 RE 匹配的工具。

关于这些工具, 最好还是自己读文档(比如 `man awk` 等)、[上网学习](#)和[参考相关书籍](#)。掌握这些工具, 文本编辑必然事半功倍, 因为 CLI 的好处就是自动化、效率化。

## 2.2 Shell

类 Unix 系统(\*nix)分 kernel (内核) 和 shell (外壳) 两部分。我们可以把 shell 理解为用户与系统内核进行交互的一个界面, 界面又分为 GUI(Graphic User Interface, 图形用户界面) 和 CLI(Command Line Interface, 命令行界面)。前者在 \*nix 中常见的有 [X11\(即 X Window 系统\)](#), 后者常见的有 sh, bash, csh, ksh, [zsh](#) 等等。其中 bash 一般是 Linux 缺省默认版本的 shell, 而 zsh 是类似于 bash 却功能更为强大的 shell。

## 2.3 脚本

脚本通常就是指一系列系统命令的组合。由于 \*nix 系统程序之间通信起来非常方便, 利用脚本可以轻松完成各式各样的任务, 特别是处理文本任务, 这是 \*nix 最大的一个优势。我们可以简单地将它理解为批处理, 这通常是在 Windows 系统下的叫法(.bat)。

## 2.4 网络

网络部分最常用的就是 SSH(Secure SHell) 和文件传输的 [FTP](#) 或 SFTP 等等。

### 2.4.1 SSH

关于 SSH，常用的是通过配置 `~/.ssh` 路径下的文件，比如 `config` 文件：

```
Host lab
  HostName 114.212.240.250
  User username
```

这样，就可以简化登录，输入 `ssh lab` 就相当于 `ssh username@114.212.240.250` 了。另外，还可以配置 RSA 密钥，省去每次输入密码的麻烦。具体细节可以参考“[蜗牛书](#)”或者互联网资源。

### 2.4.2 FTP, SFTP

关于文件传输，可以直接用 `scp` 命令操作。具体可以 `man scp` 查看手册。这里主要推荐使用 GUI 工具，比如全平台的 [FileZilla](#)，OS X 或 Windows 的话还有 [Cyberduck](#)，界面友好。



## 第3章

# 化学小工具

本章简单介绍一些常用的化学工具。

### 3.1 Avogadro

[Avogadro](#) 是一个全平台的自由开源软件。功能类似于 GaussView。可以编辑、查看分子结构等等，还可以输出 PovRay 格式文件渲染成高质量的图片。

### 3.2 Molden

[Molden](#) 可以用来查看输出文件的几何结构、振动频率、轨道图像、电子密度等等，还可以输出 VRML 和 PovRay 格式文件以便作多媒体处理使用。

### 3.3 wxMacMolPlt

[wxMacMolPlt](#) 也是全平台的一个软件，是 GAMESS 的一个 GUI 界面，兼具有 Avogadro 和 Molden 的一些功能。

### 3.4 Multiwfn

[Multiwfn](#) 是国产（北京科技大学陈飞武的博士生卢天开发）的一款波函数分析软件，集成了波函数分析方面的工具。

### 3.5 Open Babel

[Open Babel](#) 可以用来转换计算化学输入文件、进行分子建模等等。类似的工具还有 [Gabedit](#) 等。

## 第4章

# 其他小工具

本章是一些杂项。主要是我常能用到的小工具。

### 4.1 平面图形图像

#### 4.1.1 Imagemagick

[Imagemagick](#) 是一个很方便且功能强大的 CLI 图形处理工具。

比如截图得到批量的 Molden 格式的轨道图像，可以利用它进行方便的批量裁剪、背景透明化等等，比如写成如下脚本：

```
for file in $(ls *.png)
do
    convert -crop 676x689+133+58 -transparent lime $file CT_$file
done
```

这里裁剪出了  $678 \times 689$  大小的图片，起裁点是原图 (133,58) 处，由于将 Molden 背景设为青绿色(lime)，将其透明(transparent)化。效果如图 4.1 所示。



图 4.1: 对 Molden 轨道截图处理示意

#### 4.1.2 GIMP

[GIMP](#) 可以直接理解为 GNU 世界用来替代 Photoshop 的工具。

#### 4.1.3 Inkscape

[Inkscape](#) 是开源免费的矢量图软件，全平台通用。

## 4.2 3D 模型

### 4.2.1 PovRay

[PovRay](#) 用来渲染静态高质量图片，可以得到很漂亮的效果。图 4.2 是我曾用 Diamond 3 软件渲染的一个晶胞的场景。

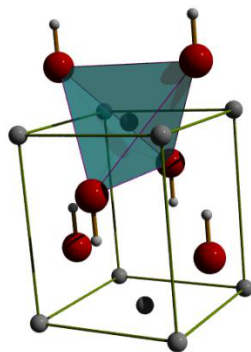


图 4.2: PovRay 渲染效果的一个简单演示

### 4.2.2 FreeWRL, Meshlab

[FreeWRL](#) 和 [Meshlab](#) 主要是我用来查看和转换 [VRML](#) 文件 `.wrl` 的工具。Molden 可以输出为 VRML 格式，然后就可以利用 FreeWRL 来浏览，用 Meshlab 可以将 `.wrl` 文件转换为 `.u3d` 格式，可以嵌入到 PDF 文件里（参考 [Asymptote](#) 一节）。

### 4.2.3 Blender

[Blender](#) 是业界著名的免费开源 3D 软件，从建模、动画、渲染等到甚至用来做 3D 打印模型都应有尽有。

## 4.3 科学软件

### 4.3.1 SciDAVis

[SciDAVis](#) 理解为 Windows 下 Origin 的开源替代品即可。

### 4.3.2 Gnuplot

[Gnuplot](#) 是比较方便好用的一个绘制数据图的软件，优点在于可以 CLI 交互或者脚本批量处理。

### 4.3.3 Sage

[Sage](#) 是开源免费的符号计算软件，可以理解为 Mathematica, Matlab 的替代品。非常有 Python 风格。

### 4.3.4 R

[R](#) 是用来进行统计分析及绘图的强大平台，可类似地看作 SPSS 之类东西的替代品。与 R 相似的还有看起来更优雅的 [Processing](#) 等。

## 4.4 文献管理

关于文献管理，商业软件中 EndNote 非常有名。但是，免费软件里面有可以取而代之甚至表现更好的工具。

之前你可能有听说过 Jabref。现在，有了更方便更现代理念的工具 — Zotero、Mendeley 和 Docear 等。它们使用了“云同步”的思想，可以将文献库在浏览器端、个人电脑端以及移动设备端全平台地同步。

Zotero 对浏览器支持非常好，可以方便地一键导入文献。Mendeley 则是 PDF metadata 的识别功能极其强大，而且自带了 PDF 工具和移动客户端（如 iOS 平台）。另外 iOS 平台有一款叫做 Papership 的软件可以兼容管理 Zotero 和 Mendeley。

## 4.5 T<sub>E</sub>X 相关

T<sub>E</sub>X 本身是一个很好的免费平台，可以生成高质量的排版。它本身就是一个很好的排版方面（特别是数理方面）的工具，互联网和图书资料都很多，所以不再赘述，仅简单介绍两个和 T<sub>E</sub>X 相关的工具。

### 4.5.1 Asymptote

Asymptote 是一个矢量绘图语言，可以为 PDF 文件嵌入 .prc 格式的真 3D 图形(PDF 支持 .u3d 和 .prc 的 3D 模型)。这就为在 PDF 文件中嵌入分子轨道或是三维势能面提供了可能。Asymptote 可以与 L<sup>A</sup>T<sub>E</sub>X 结合后绘图与排版。

这里给出两个给 PDF 嵌入 3D 分子模型的思路：

#### 思路一

- 直接使用 Asymptote 绘制，用 L<sup>A</sup>T<sub>E</sub>X 排版时调用 Asymptote。

#### 思路二

- 首先用 Molden 将轨道保存为 VRML 文件；
- 再用 Meshlab 将 .wrl 转为 .u3d；
- 利用 movie15 宏包在 L<sup>A</sup>T<sub>E</sub>X 排版时将 .u3d 文件嵌入。

### 4.5.2 TeXmacs

TeXmacs 虽然是受到了 T<sub>E</sub>X 和 Emacs 的影响，但它是一个完全独立的软件，不依赖于 T<sub>E</sub>X 和 Emacs，却有 T<sub>E</sub>X 的精准版式和类似 Emacs 的强大扩展功能，是一种“所见即所想”的文档软件。写结构化文章、记笔记、甚至用来推导公式和演算都非常方便。类似地，还有 LyX。

TeXmacs 的强大和好用之处只有真正用过了才知道（基本可以完全代替“用户体验方面设计相当蹩脚”的 T<sub>E</sub>X 系统）。不过唯一的毛病可能就是目前还不太稳定。

## 4.6 其他

### 4.6.1 版本控制与 Git

版本控制是一种非常重要的概念，在工程中可以有效地保证效率和流程顺利。对版本控制的方案，常见的有 SVN(Subversion), Mercurial, CVS 以及 [Git](#)。这里面，我对 Git 相对了解，而且随着 [GitHub](#) 的流行，由“Linux 之父”Linus Torvalds 开发的这个分布式版本控制越来越受欢迎。

使用 Git 版本控制的好处就是可以更方便地协作开发，而且版本控制可以随时回溯开发的过程。

### 4.6.2 脚本思维与 Python

脚本思维就是用会利用脚本语言去完成一些合适的任务。脚本语言特点就是好写(编写效率高，适合轻量级编程)，更接近人类自然语言，开发出它们的目的就是简化流程，使人从繁重的机械劳动中解放出来。把 [Python](#) 单独提出来是因为它是一个集大成者，而且在科学领域应用也非常广泛，语法也比较优雅，所以比较适合科研使用。

其他著名的一些还有：比如文本编辑处理类的 sed, awk, [Perl](#) 等；和 Python 类似但是在网络开发中更常用的 [Ruby](#)(名字来源就是为了取代 Perl)；Web 常见的 JavaScript, PHP；编辑器的 Emacs Lisp；以及嵌入式的 [Lua](#) 等等。

### 4.6.3 轻量编辑与 Markdown

之前已不止一次提到「轻量级应用」这个概念。现代设计理念之一有「极简主义」的风潮——希望能从繁复的现象中返璞归真，抽象出最核心的东西。由此诞生了基于「化繁为简」的理念，进行「奥卡姆剃刀」般修剪的做减法的设计哲学，成为轻量化设计。

[Markdown](#) 就是把繁复的 HTML 语言保留最核心的要素，重新利用更加易读易写的语法构成。简体中文语法可参考[这里](#)。

本手册的编写即采用这种思想，因为没有过于繁复的文本结构，所以直接原稿采用 Markdown 格式书写，再利用 [Pandoc](#) 这样的转换工具，加上  $\text{T}_\text{E}\text{X}$  作为排版引擎，最后可以输出结构质量上乘的文本。特别是可以直接利用 GitHub 上的 [Pro Git Book 项目](#)，再加上本身同样可以利用 Git 进行版本维护，就使本文档写作更加高质高效。

类似地，和 Python 的文档维护方式相同，[reStructuredText+Sphinx](#) 也是一种方案。

## 第5章

# 个人小工具

本章介绍一下自己开发的一些小工具。因为经历时期比较长，而且并没有按照开发顺序编写，所以没有一个清晰统一的程序设计风格。

### 5.1 几何结构变换

几何结构变换主要指对 XYZ 坐标的变换操作。目前包括对一个分子几何结构的旋转操作和两个结构之间的线性插值工具。

#### 5.1.1 单分子 XYZ 坐标变换

##### 实现方式和原理

工具采用 Python 编写，一个实例程序代码如下：

文件名为 `Geom_XY_rotate.py`：

```
import math
x1=0.0504214577
y1=1.0819963150
x2=-1.0520729230
y2=0.2576932903
k=(y2-y1)/(x2-x1)
b=y2-k*x2
x6=(x1+x2)/2
y6=(y1+y2)/2
r=math.sqrt(x6*x6+y6*y6)
theta=math.asin(r/b)

atom=[]
x=[]
y=[]
z=[]
sample=open('Geom.xyz','r')
while True:
```

```
line=sample.readline()
if len(line)==0:
    break
word=line.split()
atom=atom+[word[0]]
x=x+[float(word[1])]
y=y+[float(word[2])]
z=z+[float(word[3])]
sample.close()

N=56
f=open('GR_Rotated.xyz','w')
for i in range(N):
    xx=x[i]*math.cos(-theta)+y[i]*math.sin(-theta)
    yy=-x[i]*math.sin(-theta)+y[i]*math.cos(-theta)
    f.write(atom[i]+' '+str(-xx)+' '+str(yy)+' '+str(-z[i]))+'\n')
f.close()
```

代码第一部分是一个实例中先计算出旋转角度  $\theta$ 。第二部分是读取几何结构。第三部分是对坐标操作并输出。

主要原理就是读取几何结构，然后进行空间操作后输出。实例程序为将分子在 XY 平面内旋转角度  $\theta$  的工具。

### 用法示例

仍以上述工具为例，将需要变换的几何结构复制到 `Geom.xyz` 这个文件中，在终端中运行此 Python 脚本，

```
python Geom_XY_rotate.py
```

即可产生文件名为 `GR_Rotated.xyz` 的输出文件。

## 5.1.2 分子间的几何结构线性插值

### 实现方式和原理

采用 Python 编写，源代码如下：[Geom\\_Inter.py](#)

其中，前三段分别是初始化并读取起始和终了几何结构，最后一段进行线性插值计算。为了照顾对称性，保留了 6 位小数。

本工具附带了几个 Shell 脚本配合使用：

这里是一个生成 molpro 输入文件的样例。`GI_molpro_Gen.sh` 代码如下：

```
python Geom_Inter.py
for name in `ls *.com`
do
    cat Head $name Tail > tmp
    mv tmp $name
done
```

就是将插值好的坐标文件与除几何结构外输入文件的部分和坐标输入拼接起来，即可得到需要的所有输入文件（例子中在始末结构之间插入 15 个点）。最后就是 `molpro_run.gen` 和 `molpro_run.sh` 分别用来生成脚本和批量运行这些输入文件，以及 `Traj_print.sh` 用来将插值的轨迹保存为一个 XYZ 格式轨迹动画，`trajectory.xyz`。

### 用法示例

- 先把需要生成的计算输入文件将除几何结构的部分替换到 `Head` 和 `Tail` 两个文件中；
- 将初始和终了的几何结构 XYZ 坐标（注意前后要一致对应才有效）替换到 `Geom0` 和 `GeomX` 两个文件内，运行 `./GI_molpro_Gen.sh`；
- 运行 `./molpro_run.gen`；
- 运行 `./molpro_run.sh`；
- 计算完成后，运行 `./Traj_print.sh`。可以使用 Avogadro 看一下轨迹 `trajectory.xyz` 是否正确。

## 5.2 IRD

IRD 的方法来源于 Robb 的[一篇文献](#)。这个工具开发得很不成熟。

### 实现方式和原理

将文献中的方法用程序实现了出来。仍使用的 Python 实现，需要用到 `SciPy` 库。

源代码如下：[IRD.py](#)

这个程序注释得较为详尽，基本上就是从输出文件读出需要的数据后，利用文献的方法计算，最值优化方法是库中的 L-BFGS-B 方法。因为程序比较不成熟，所以目前针对性比较强，计算对象是环丁二烯，格式解析 molpro 的 `.out` 文件。

除计算主程序外，和几个 Shell 脚本一起使用。其中，文件名含 `MW` 是使用了质重坐标系，`get_*` 是用来获取 `.out` 中的几何结构、梯度和 Hessian，`format_*` 用来将获得的数据格式化为 IRD.py 读取的模式。

其他辅助性脚本有 `MW2XYZ.sh` 用来进行质重坐标向 Cartesian 坐标转换（此处针对环丁二烯），`trajectory.sh` 用来看 IRD 轨迹。（源码请参考 `src` 路径下）

### 用法示例

- 先用一个 `*.com` 计算频率以得到 Hessian；
- 在 `IRD.py` 里可修改形如 `# <<--` 的箭头指向的的注释的变量，比如初始能量 `E0` 和步长 `rmax` 等；



- 运行 `./IRD.sh` 或者 `./IRD_MW.sh` 即可得到 Cartesian 或质重坐标系下的结果。通过修改 `IRD*.sh` 中的循环次数可以控制 IRD 轨迹点数。

## 5.3 能量优化

### 实现方式和原理

这个是和 IRD 工具类似的思路，利用 Python 和 Shell 脚本调用 GAMESS 完成 MRMP 的能量优化任务。最值优化方法仍然是 SciPy 里的 L-BFGS-B。Shell 进行调用程序或者文本格式处理，Python 进行算法流程的实现。

源代码文件：[OPT.py](#)

### 用法示例

- 替换初始几何结构 `Geom0`;
- 替换除几何结构外的输入文件部分 `Head` 和 `Tail`;
- 运行 `./RUN_OPT.sh`。

## 5.4 ci vector 系数统计

ci vector 的系数统计在将轨道局域化后变得十分有用，因为局域化后局域化的原子轨道的 ci vector 系数变得分散。这里最主要的是统计出离子性、共价性的表现。

### 实现方式和原理

核心思路就是用 Linux 文本编辑工具进行文本处理，写成 Shell 脚本方便使用。

源码文件：[couple.sh](#)

即提取需要的数据后再利用正则表达式匹配并进行统计求和。

### 用法示例

以环丁二烯为例。在仅有局域化轨道结果输出文件的路径下，运行 `./couple.sh` 即可得到对环丁二烯的离子性、共价性的统计分析。（molpro 软件通过 `thresh,thrpri=1e-7` 可以将 ci vector 系数大于  $1 \times 10^{-7}$  的结果全部输出）