**1.使用之前make sure你下载了这些库和他们对应的版本**：

**1.1加载必要的module:**

module load apps/python/3.7.1

module load apps/gnuplot/5.4.6

**1.2可以pip之前先创建虚拟环境:（清华源：**

**-i https://pypi.tuna.tsinghua.edu.cn/simple）**

>>>python3 -m venv 目录名

创建虚拟环境

然后：

>>>source 目录名/bin/activate

激活虚拟环境

退出：

>>>deactivate

**1.3安装依赖库：**

pip install <library>==<version>

美观版本：spst可用，但是dm不可用

PyQt5 5.15.9

PyQt5-sip 12.12.1

Pymatgen 2022.0.17

**兼容版本：两平台都可以用（pdos有个bug，切换一下下拉菜单就好了）**

PyQt5 5.15.6 （5.15.2）

PyQt5-sip 12.10.1 （12.8.1）

Pymatgen 2022.0.17

（注意Pymatgen版本和ruamel.yaml是不是一致的，建议是：

ruamel.yaml 0.17.40

ruamel.yaml.clib 0.2.8

同时如果出现bug，在环境变量中配置export QT\_DEBUG\_PLUGINS=1然后bash一下，再次运行看看哪里出问题了

）

Qt5.15.0 :需要安装（查看“预安装库”目录）

然后qmake --version 查看是不是5.15.0版本的Qt库，如果不是需要把PATH\_TO\_YOUR/Qt5.15.0/5.15.0/gcc\_64/bin/添加到环境变量的PATH，把PATH\_TO\_YOUR/Qt5.15.0/5.15.0/gcc\_64/lib/添加到环境变量的LIB：

当然which qmake可以查看qmake的目录

如下所示

#Qt5.15 path:

export LD\_LIBRARY\_PATH=/hpc/data/home/spst/xiety/bin/Qt5.15.0/5.15.0/gcc\_64/lib:$LD\_LIBRARY\_PATH

export PATH=/hpc/data/home/spst/xiety/bin/Qt5.15.0/5.15.0/gcc\_64/bin:$PATH

export LD\_LIBRARY\_PATH=/usr/lib64/qt-3.3:$LD\_LIBRARY\_PATH

export XDG\_RUNTIME\_DIR=/hpc/data/home/spst/xiety/bin/QE-batch

**2.使用方法：**

建议使用pw.x --verson :6.8

2.1 在和QE-batch相同目录下（以下统称“总目录”）上传你的结构文件（POSCAR格式，fractional)

2.2 在总目录建立in\_{mode}文件，文件中你只需要设置和结构信息无关的部分（mode指的是你需要计算的mode，例如relax，scf等，其中pdos和Bader计算**不需要**建in\_pdos和in\_BaderCharge），注意把prefix=“BTO”

2.3\* 如果需要加spin，则在总目录下创建名为SPIN的文件，在里面输入你需要加SPIN的元素以及方向

example：>>>cat SPIN:

Ni1 0.5

Ni2 -0.5

Ni 0.5

Mn -0.5

2.4\* 如果需要加DFT+U，则在总目录下创建名为DFT-U的文件，在里面输入你需要加U的元素以及大小

example：>>>cat DFT-U:

Ni 6.7

Ni1 6.7

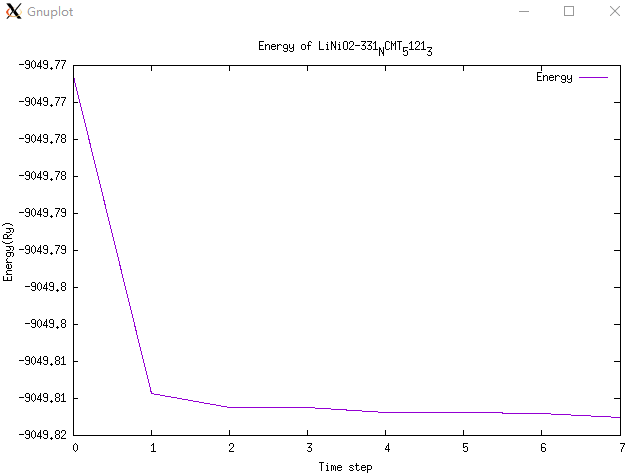
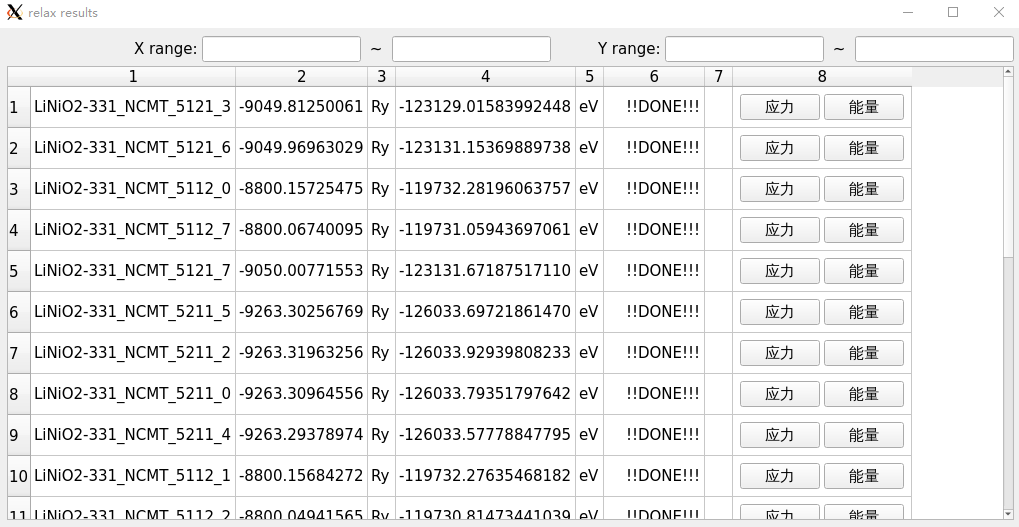
Ni2 6.7

Mn 4.2

Co 4.9

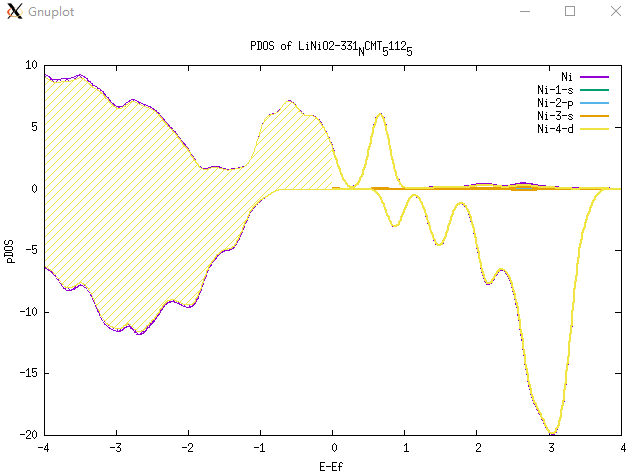
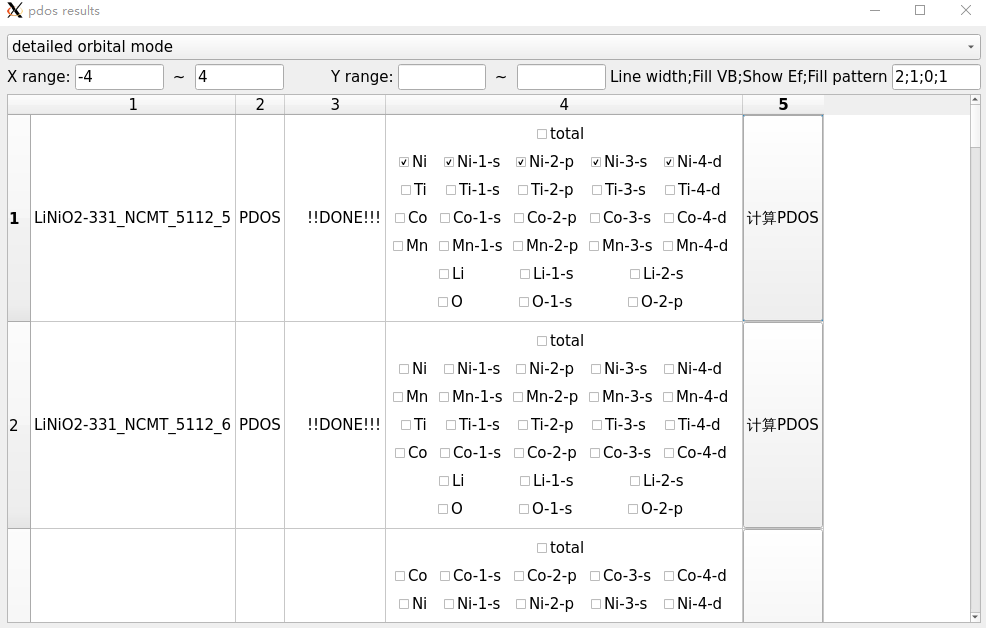
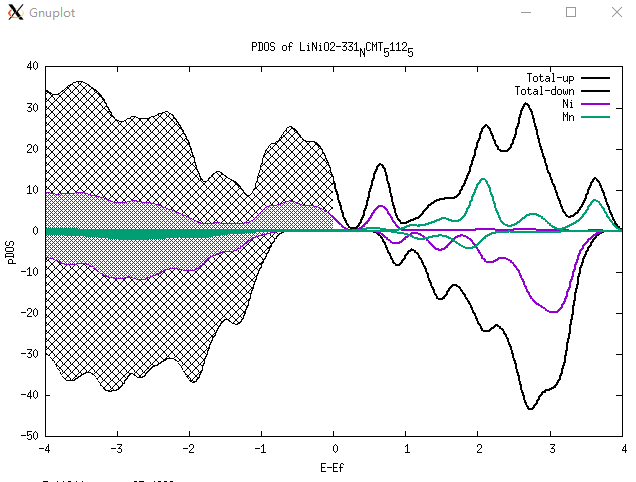
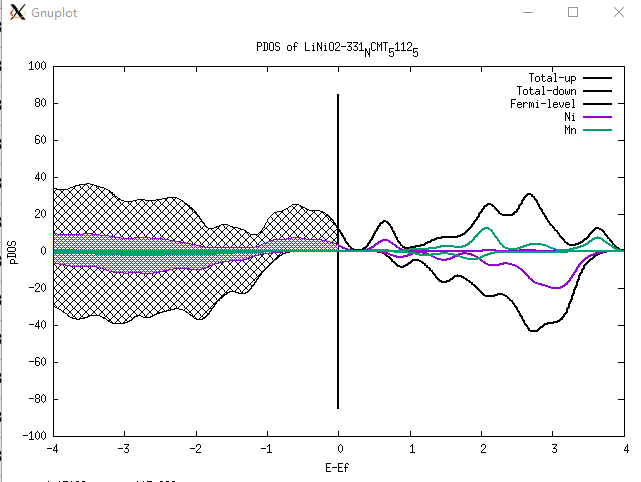
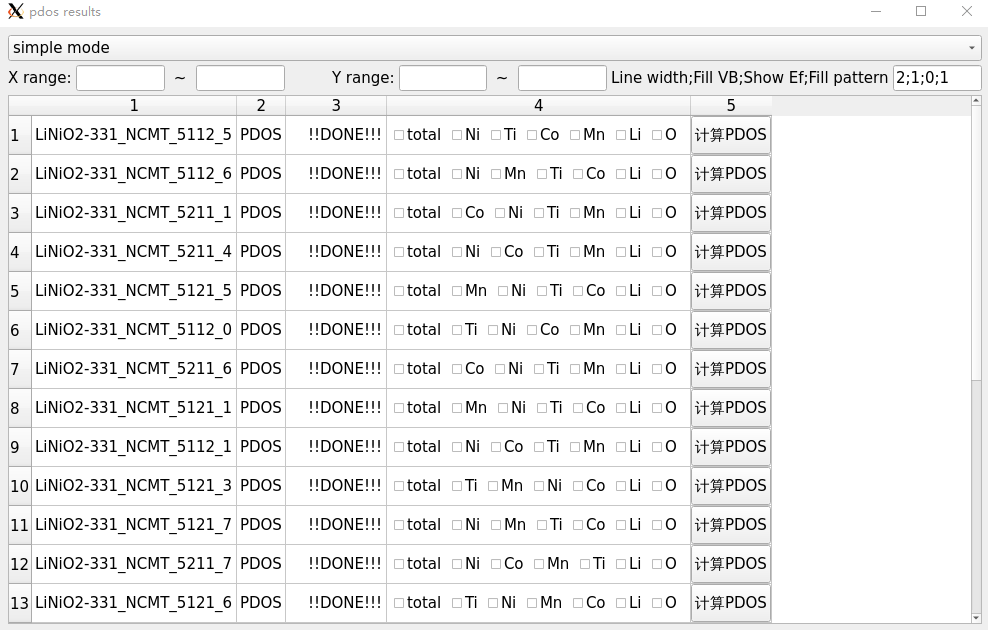
Ti 1.9

2.5 运行python QE-batch/APP\_9-6.py,选择模式后，点击“查看计算进度”即可查看计算进度



2.6 在修改“选择计算平台”中内容的时候，注意修改完点保存

2.7 pdos有三种查看方式，可在下来菜单栏中选择：分别是simple mode，detail orbital mode以及manual mode，分别可以查看元素tot pdos，元素投影轨道的tot pdos以及自定义查看的pdos。



在自定义查看的pdos中，在input\_text中允许的输入格式为：

**分别列出：{需要计算的原子序号/元素符号}-1-s（也可以是-tot）**

**求和：（需要计算的原子序号/元素符号）-1-s（也可以是-tot）**

**example:**

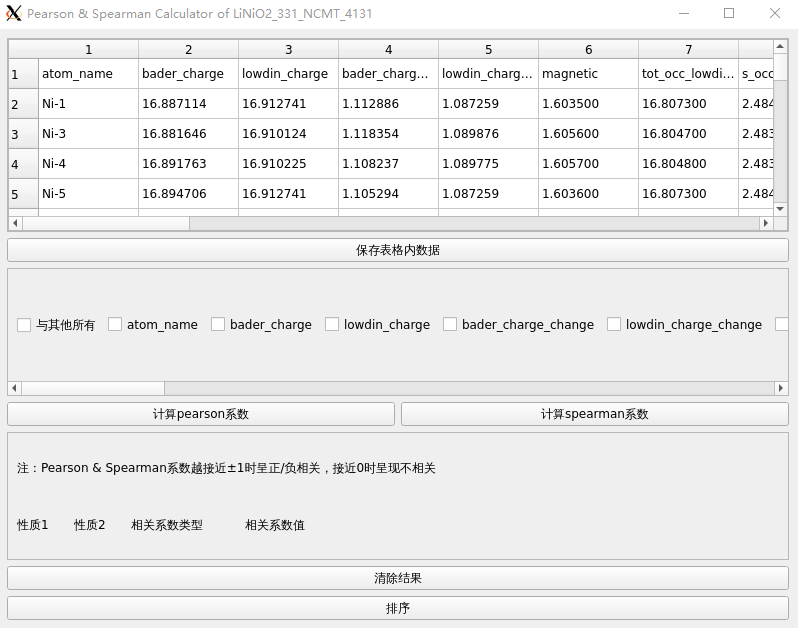
1.分别列出第1-20号原子的1-s轨道的pdos,所有Ni原子的pdos，Co元素4-d轨道pdos之和**:{1-20}-1-s;{Ni}-tot;(Co)-4-d**

2.分别列出所有Ni原子和1-88号原子的1-s轨道：**{Ni，1-88}-1-s**

3.画出PEA分子的pdos（假设他们原子编号从2-12,23-30）：**（2-12,23-30）-tot**

**！！！注意：这里的原子序号是vesta里面的编号，1-s并不是元素实际的1-s轨道，而是赝势画出来的1-s这里的1代表赝势输出的第1个轨道**

2.8 数据分析板块目前放置在BaderCharge计算中，目前有一个设置不合理的地方，也就是你需要计算pdos之后才可以点击badercharge里面的“**计算电荷**”，点击计算电荷之后重新打开查看badercharge页面，打开“**分析数据**”，就是数据分析界面了，会提炼出一些预设好的原子信息以及电子信息，通过“**查看此类原子**“按钮可以查看对应的信息，并且对里面信息可以计算关联系数，通过关联系数找到这个体系的一些性质之间的关联。



2.9 time\_schedule.py中你需要设置

time\_schedule=[["PEA-C","relax"],["PEA-C","scf"],["PEA-C","bands"],["PEA-A","relax"]]为你需要计算总目录的目录名字以及计算模式，每个总目录下包含QE-batch以及in\_{mode}文件以及.vasp文件。

time\_schedule.py应该和这些总目录在一个目录下。

2.10.1\*关于固定原子：在replace.py中修改：

fixed\_atom\_mode=1,

然后把upper\_layer和down\_layer设置成你需要固定的原子的上下限（fractional）

如果这些原子之外还需要固定原子，则在fixed\_atom\_index输入vesta中的原子序号

2.10.2\*关于修改计算用的赝势：在replace.py中修改：

sp\_format="X.SG15.PBE.UPF"

可供修改的格式是 ：

"X.SG15.LDA.UPF","X.SG15.PBE.UPF","X\_frl\_gga.upf","X\_srl\_gga.upf","ONCV.PWM.X.UPF","ONCV.PWM.X.IN"

**2.10.3\*关于不放心批量替换的结构以及创建的文件夹中的内容，需要检查输入文件，就把replace.py中的sub\_script=0,这样创建完目录后就不会提交任务了。**

**3.关于DM平台的PyQt5安装失败问题（2023/10/31注：稳定版可以使用，请忽略此消息）：**

假设目前依旧使用不了APP\_9-6.py

这些脚本是APP\_9-6.py UI界面的后台程序(以下所有程序都在总目录下运行):

3.1##read\_relax\_E\_for\_UI.py: 查看计算进度

>>> python QE-batch/read\_relax\_E\_for\_UI.py {mode}

**若是要删除DONE的结构的波函数，可以在**read\_relax\_E\_for\_UI.py **设置wavefc\_rm=1，如果你后续有pdos计算则不建议这样做**

3.2##replace.py :计算.vasp文件

>>> python QE-batch/replace.py {structure\_name} {mode} 0 0

3.3##run-all.py:调用replace.py计算当前目录下所有结构的{mode}

3.1.1

>>> python QE-batch/run-all.py {mode}#计算当前目录下所有的.vasp文件

3.1.2

>>> python QE-batch/run-all.py {mode} pv\_result\_out#计算当前目录下没有计算的.vasp文件

3.4##read\_qe\_out\_all.py：提取计算完成的结构：

>>>python QE-batch/read\_qe\_out\_all.py

3.5##auto\_cal.py:自动托管计算/续算当前目录下的结构

>>> python QE-batch/auto\_cal.py {mode}

3.6##sum\_qe\_pdos.py : 加和一种元素/一种元素轨道的pdos：

>>> python QE-batch/ sum\_qe\_pdos.py {元素/元素轨道}

3.7##sum\_qe\_pdos\_diy.py : 自定义查看pdos：

>>> python QE-batch/ sum\_qe\_pdos.py "{1-2,9-10,11}-1-s"

#####关于数据分析的一些脚本，应该你们暂时用不到：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#######readvalence.py {dirname}

在replace.py 计算完badercharge之后，进入dir中，运行bader Bader\_{dirname}.cube

然后python readvalence.py {dirname}即可显示原子上面的charge，第二列是现有charge，第三列是电离失去/得到charge

#######read\_lowdin.py {str\_name} mode

用于读取out\_pdos中的lowdin charge，并且产生Lowdin\_of\_{str\_name}.txt

#######get\_surrounding\_by\_elements\_multi.py: 调用“clustering-xsf.py”

有以下几个参数可以调节：

2.1.修改需要查看的元素周围的最近邻元素种类以及他们的化学环境：[['元素1','元素1最近邻元素']，['元素2','元素2最近邻元素']，['其他元素','其他元素最近邻元素']，]

surround\_specified=[['Li','TM'],['O','tot'],['TM','Li']]

2.2.

产生f"SURROUNDING\_AROM\_of\_{str\_name}.txt"文件，格式如下：

8-Ni#5:['16-Li#3', '15-Li#4', '18-Li#5', '27-Li#9', '23-Li#3', '24-Li#1']

9-Ni#4:['17-Li#4', '12-Li#3', '25-Li#8', '26-Li#6', '20-Li#7', '11-Li#2']

10-Li#1:['105-Ti#2', '108-Ti#2', '4-Ni#3', '7-Ni#3', '99-Mn#1', '102-Mn#1']

11-Li#2:['97-Mn#1', '108-Ti#2', '100-Mn#3', '4-Ni#3', '9-Ni#4', '106-Ti#3']

也会产生一个f"{df\_dir}/{str\_in}\_cluster\_data.pkl"文件用于储存分类的数据，分类的数据可以通过以下方式读取到spe\_df这个字典中：

if os.path.isfile(f"{df\_dir}/{str\_in}\_cluster\_data.pkl"):

#print("!!!!!!!!!!!!!!!!!!!!!!!!!!!!")

loaded\_data={}

with open(f"{df\_dir}/{str\_in}\_cluster\_data.pkl", "rb") as file:

loaded\_data = pickle.load(file)

spe\_df=loaded\_data['spe\_df']

WARNING！！！

WARNING！！！

WARNING！！！该脚本目前不能用于读取不带磁矩的体系,除非clustering-xsf.py的数据分类方式改变！！！

#######separate\_sur\_atom\_info.py {str\_name}和separate\_sur\_atom\_info\_lst.py {str\_name} 分别用于统计get\_surrounding\_by\_elements\_multi.py产生的数据，得到所有元素中最近邻的元素个数以及他们的环境类型，储存在Separate\_Surrouding\_of\_{str\_name}.txt和Separate\_Surrouding\_List\_of\_{str\_name}.txt文件中

！！！注意其中的O元素只会被统计一次

#######search\_data.py {str\_name} "{11-20,Ni}"" properties1+properties2+..."

在{str\_name}下查找11-20,Ni的“properties1，properties2，...”信息，并且print出来

#######extractor.sh A：去读取目录A下面的A\_1\_2,A\_1\_3等目录下面的out\_relax\_A\_1\_2,in\_relax\_A\_1\_2,\*.UPF,xty\_test以及QE-batch整个目录到A-extractor里面（还有A\_1\_2.vasp也会被读取)

#######atomic\_info\_extractor.py {str\_name}会生成： 八面体体积；八面体因子OF；平均角度畸变；畸变因子DI 数据到ATOMIC\_INFO\_of\_{str\_in}.txt文件里面

#######search\_multi\_dupli\_data.py：search\_data.py的衍生版本，除了会计算重复的原子信息之外，还会把输入的带\*的性质（如果不在available的props里面）依旧会返回值， 不过值都是0，若是带\*的性质（如果在available的props里面），那就正常返回值

#######exchange\_E\_extractor.py {str\_name} “props1+props2” 在当前目录下寻找{str\_name}-exhange\_file-extractor,并且读取{str\_name}-exhange\_file-extractor/{str\_name}\_Li2\_Ni3这类文件夹（互换了2号Li和3号Ni）的能量，并把对应的原子的props抠出来，注意，这里的props除了调用search\_multi\_dupli\_data.py进行搜索（也就意味着你可以加上\*返回0的default值），还可以使用props:Li来获取独特元素的props，如果不加：则默认两种元素的props都有

#######Delta\_E-app.py，基于exchange\_E\_extractor.py的UI交互界面，也拥有spearman和pearson系数计算功能。