easyvasp 简介：

easyvasp是pymatgen的重新封装，开发原因是pymatgen自动化的建模的方法十分复杂。在easyvasp中，你可以简单的从一个cif文件或者POSCAR文件开始批量建立你的模型，或者从计算完成的文件夹来批量得到分析数据，enjoy it！

快速安装

首先安装c++14 否则pymatgen安装容易报错

安装Anaconda3 作为python环境

安装pymatgen及其依赖包，建议使用国内源

pip install pymatgen==2018.8.13 -i <https://pypi.mirrors.ustc.edu.cn/simple/>

安装scikit-learn，pandas 方法同上

安装easyvasp

1. 打开cmd
2. cd /d “easyvasp安装目录”
3. Python setup.py install 安装easyvasp
4. 任意安装一个python IDE 现在你就可以编写程序来建模了！（推荐pycharmm）

如何找到你的初始结构？

推荐使用材料基因组计划：<https://materialsproject.org/>

使用你的github或者google账号登陆就可以查询你的账号，就可以使用Formula来查找你的结构了。

结构按照，形成能来排序，你可以按照空间群或者形成能来选择结构，然后下载vasp计算好的模型，你也可以使用python的api来获取，这需要你申请一个id

easyvasp主要结构

easyvasp分为easyjob和dealjob和Toolkt三部分

easyjob设计目的为高通量建模，并负责分析所有跟结构有关的量

dealjob负责高通量后处理

Toolkit 中主要包含一些小工具，包括作图或者计算脚本

导入方式：

from easyvasp.easyjob import easyjob

from easyvasp.dealjob import dealjob

from easyvasp.tool import Toolkit

easyjob简介

该程序目标在于实现vasp工作的自动化和高通量计算的建模部分，目前可以实现此功能的程序主要是pymatgen，但是pymatgen设计复杂，封装过度，各种类的依赖关系复杂，对于普通材料学背景的研究人员过于复杂，所以我对pymatgen做了进一步封装，追求简单实用，便于非计算机背景的研究人员使用。

主要结构

easyjob 类

——poscar类

——structure类

——kpoints类

——potcar 类

——incar 类

创建一个easyjob实例

easyjob(structure,foldname)

easyjob初始化接受两个参数，一个是pymatgen的结构类structure，另外一个是该任务的工作目录foldname

任务的所有输出都会在foldname所指定的目录下进行

easyjob会执行初始化操作，总体如下:

对于poscar，判断是否是表面模型(z轴最大空隙大于6埃米)，如果是表面模型，则平移模型使得模型下表面距离盒子的底面有2埃米的距离。easyjob认为真空层下方是上表面，上方是下表面

对于incar，easyjob初始化一个optim的INCAR，可以在file文件夹下的optimINCAR中自定义，对于Mn,Fe,Co,Ni,Cr元素的体系，将自动设置ISPIN=2，对于表面模型，将自动设置ISIF=2

对于kpoints，easyjob初始化一个k点密度为0.2的Gamma点KPOINTS文件

对于potcar， easyjob初始化一个由vasp推荐赝势组成的POTCAR文件

主要api介绍

**读取结构部分**

easyjob.read\_file(filepath)

该方法是easyjob中唯一一个全局方法，可以自动读取POSCAR，CONTCAR，\*.vasp,\*.cif,\*.xyz格式的文件，除xyz文件外都会返回structure对象，而xyz文件会返回molecule对象

一般这个api用于建立easyjob对象

例子:

s=easyjob(easyjob.read\_file(filename),foldname) #该语句新建一个easyjob对象，读取filename指定的文件中的结构，并指定输出目录为foldname，建议使用绝对路径进行解析，避免出现意料之外的错误。

下面脚本读取当前脚本目录下的aaa.cif文件，并输出工作目录到test文件夹

rootpath= os.path.split(os.path.realpath(\_\_file\_\_))[0] #定位脚本根目录

filename=os.path.join(rootpath,’aaa.cif’) #定位要读取得cif文件

foldname=os.path.join(rootpath,’test’) #定位工作目录

s=easyjob(easyjob.read\_file(filename),foldname) #转化为easyjob对象

s.write\_job() #输出

也可以使用Toolkit的find方法找到目录下指定文件

假设有100个cif在你的目录下，现在把他们输出为可以提交的文件夹,文件夹的名字是cif文件的前缀

from easyvasp.easyjob import easyjob

from easyvasp.tool import Toolkit

for top,name,path in Toolkit.find(‘cif’):

s=easyjob(easyjob.read\_file(path),name)

s.write\_job()

**注：如无特殊说明，下面的例子中s将代表已经读取结构的easyjob实例。**

**INCAR 部分**

easyjob自动设置INCAR为结构优化（optim）的内容，各种任务脚本的参数可以通过修改file文件夹里的文件修改。自动设置时计算Fe，Co，Ni，Mn，Cr时设置ISPIN=2,表面计算会自动设置ISIF=2

easyjob.get\_incar(template)

设置incar的模板为template，easyjob会自动设置这个模板为结构优化的模板，指定incar模板并非必要

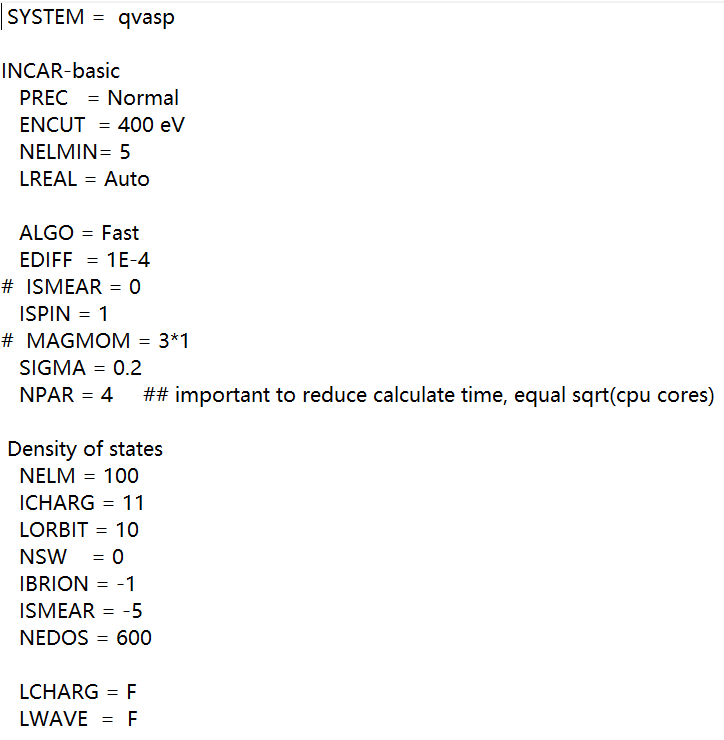
easyjob 支持自定义模板,你只需在easyjob里的file文件夹修改或添加以INCAR结尾的文件

例子:

s.get\_incar(‘dos’)

该语句使用file文件夹下的INCAR\_dos文件作为模板

下面的图片贴出了INCAR\_dos中的内容：



easyjob.incar.set(keys,values)

设置或修改incar中的keys的值为values，注意values统一使用字符串形式以避免问题

例子:

s.incar.set(‘EDIFF’,’1E-5’) #设置EDIFF为1E-5

s.incar.set(‘IVDW’,’11’) #设置IVDW为11,使用DFT-D3矫正

easyjob.incar.pop(keys)

删除incar中的某个关键字

例子:

s.incar.pop(‘NPAR’)

该语句删除incar中的NPAR关键字

**KPOINTS 部分**

easyjob. automatic\_set\_kpoint(kppa)

根据经验公式自动划分k点，kppa是k点密度，越小k点将越密集，默认设置为0.2

该经验公式参考qvasp -k命令生成

以a方向k点数为例子，具体公式为：

ka=40/(a\*2π\*kppa)

easyjob会先行判断体系是表面体系还是固态体系，方式为体系原子的zmax-zmin>4.5埃米为表面体系，否则为体相体系，如果为表面体系将设置该方向k点为1

easyjob. set\_kpoints (a,b,c)

手动指定K点密度为a，b，c

例子:

s.set\_kpoints(1,1,1) 设置K点密度为111

注：目前easyjob只支持Gamma点的K点生成，暂时不支持Line形式的KPOINTS生成，如果准备计算能带，推荐vaspkit来生成k点。

**POTCAR 部分**

easyjob. get\_potcar(style)

easyjob将自动根据poscar来生成potcar，使用的方式是vasp推荐的赝势表，大部分情况下无需手动设置，但是一定要手动设置可以调用以下命令：

例子:

s.potcartype=’ Ca\_sv C O Li’ #注意开头要加一个空格

s.get\_potcar(‘pbe’) #更新potcar，其中pbe指定的是PP文件夹内的pbe文件夹，你也可以添加其他赝势，如LDA等,只需在PP文件夹内添加对应的赝势文件夹即可

easyjob.refresh\_potcar(style,recommend)

easyjob可以使用简单的命令来刷新potcar，style指的是赝势类型，recommend是是否使用vasp官网推荐赝势，easyjob自动设置赝势为pbe，默认使用推荐赝势。

例子:

s. refresh\_potcar(style=‘pbe’, recommend=False) #不使用推荐后缀的赝势，统一使用无后缀

**POSCAR 部分**

结构通用api部分

easyjob.fix(splitpoint,reverse=False,abs\_coords=True,axis='z',v='xyz')

按照splitpoint点的位置固定原子，默认设置下固定z方向splitpoint下的所有原子

例子:

s.fix(0.35) 完全固定z方向分数坐标小于0.35的所有原子

reverse是是否反转方向

例子:

s.fix(0.35，reverse=True) 完全固定z方向分数坐标大于0.35的所有原子

abs\_coords代表是否使用绝对分数坐标，如果是False的话程序会以最大坐标与最小坐标作为起始值和终止值

例子:

s.fix(0.5，abs\_coords=False) 完全固定下半层分子

s.fix(0.75，abs\_coords=False) 完全固定下方四分之三的原子

axis和v 指定固定的方向和自由度的限制,v可选x,y,z,xy,xyz五种

例子:

s.fix(0.5,axis=’z’,v=’x’)固定z方向分数坐标小于0.5的所有原子的x方向自由度

easyjob.fix\_element(eledata，v=’xyz’)

方便的固定元素，eledata是存储元素的列表，v是自由度设置，允许值’xyz’,’xy’,’x’,’y’,’z’

例子:

s.fix\_element([‘Cu’,’O’])

固定所有的铜和氧元素的所有自由度

easyjob.move(v)

方便的平移所有原子一个向量

例子:

s.move([0,0,1]) #所有原子向上移动1埃米

easyjob.move\_to(i,v,frac=True)

将原子i向坐标v移动,默认v是分数坐标，注意原子从零开始编号

例子:

s.move\_to(1,[0,0,0]) #1号原子向上移动到盒子左下角

easyjob.move\_by()

将原子i向坐标v移动,默认v是分数坐标，注意原子从零开始编号

例子:

s.move\_to(1,[0,0,1],frac=False) #1号原子向上移动1埃米

easyjob.supercell(ii=1, jj=1, kk=1, amin=None, bmin=None, cmin=None)

方便的构建超晶胞的部分,ii,jj,kk分别是abc方向的扩胞数，amax，bmax，cmax则是最小允许长度,easyjob至少会把晶胞的abc方向扩展到允许长度以上，指定amax等值会使得ii，jj，kk失去效果。

例子:

s. supercell (2,2,1) #进行2\*2\*1扩胞

s.supercell (2,2,1,amin=8,bmin=8) #扩胞使得a向量大于8埃米，b向量大于8埃米，c方向不扩胞

easyjob.add(element,loc，frac=True)

添加一个原子进入指定位置，loc是xyz的分数坐标,frac如果是False,则使用笛卡尔坐标

例子:

s.add(‘Fe’,[0,0,0])

在（0,0,0）处添加一个铁原子

easyjob.remove(i)

删除一个指定序号的原子，注意这里从1开始编号，以求与VESTA编号对齐

例子:

s.remove(1)

删除1号原子

s.remove(‘Cu’)

删除铜原子

easyjob.replace(atom1，atom2)

用原子2替换原子1，注意atom1这里可以用编号和元素名，用编号只替换一个，用元素名替换所有元素，另外，为与VESTA等软件对齐，编号从1开始

例子:

s.replace(1,’Fe’)

替换1号原子为铁原子

s.replace(‘Mn’,’Co’)

替换所有Mn原子为Co原子

easyjob.get\_primitive()

获得原胞，直接调用即可

例子:

s. get\_primitive ()

获得晶胞的原胞，表面结构也可以使用。

easyjob.apply\_strain（strain）

应用应变，strain是三个方向上的应变，[0,0,0]代表不应变，正数代表拉伸，负数代表压缩

例子:

s. apply\_strain([0.1,0,0]) #x轴方向拉伸百分之10

s. apply\_strain([-0.1,0,0]) #x轴方向压缩百分之10

easyjob.copy（）

复制一个新的easyjob对象

例子:

s1=s.copy()

复制一份easyjob对象，去除两个对象相关性

过渡态插值

easyjob.find\_path(end\_structure,images,sites=[],charge=None)

以当前模型为初态，接受一个末态结构，而images是图像数量，sites是可以放松的位点列表（这里指可以非线性插值的原子列表），charge如果指定一个Chgcar，则可以使用电荷密度作为插值参考，具体信息了解pymatgen的NEBPathfinder类，返回一个列表，foldname默认为00-xx。

easyjob.write\_job（）

输出任务到任务目录，保存时用，输出INCAR，POSCAR，KPOINTS，POTCAR

例子:

s. write\_job()

输出任务，下面的函数用于单独输出各个文件。

easyjob.write\_incar（）

easyjob.write\_kpoints（）

easyjob.write\_poscar（）

easyjob.write\_potcar（）

体相结构部分

easyjob.make\_surface(miller\_index=[0, 0, 1],bonds=None, min\_slab\_size=8, min\_vacuum\_size=20, max\_normal\_search=None,primitive=False)

切表面命令

miller\_index 给出晶面的米勒指数，如[0,0,1]表示切001晶面

bonds接受一个字典，指定不能切断的键的长度

例如：bonds={(**"P"**, **"O"**): 3,(**"C"**, **"O"**): 2.8,(**"Si"**, **"O"**): 2.8} ，指定P-O键小于3埃米的键不能切断，C-O键小于2.8埃米的键不能切断，Si-O键小于2.8埃米的不能切断。

min\_slab\_size 指定最小晶面厚度，默认为至少8埃米

min\_vacuum\_size 指定真空层厚度，默认真空层厚度是20埃米

max\_normal\_search 最大扩晶胞数，使得法向向量尽可能垂直与表面，越大越垂直

primitive 指定切面之前是否先寻找原胞

例子:

s.make\_surface(miller\_index=[0, 0, 1],bonds={(“P”,”O”):3})

切001面但是不切断磷酸根

这个方法返回一个列表，列表里存储了所有可能的表面，并且已经被转化为了easyjob对象,foldname默认为{晶面-断面编号},你也可以修改foldname参数以写到不同文件夹。

例子:

for i in s.make\_surface(miller\_index=[0,1,0]):

i.write\_job()

表面结构api部分

easyjob.set\_vacuum(thick=20)

方便的重新设置真空层厚度，该命令将真空层厚度重设为thick，其单位是埃米，注意，该命令对非表面模型无效。

例子:

s.set\_vacuum(20) 重设真空层厚度为20埃米

easyjob.reverse(frac=True)

方便的调换z轴的位置，原理是沿着x轴旋转180°,再移动旋转后的模型，使其z轴分数坐标中心不变，主要用于获取下表面。

例子:

s.reverse() 翻转模型以获得下表面

s.reverse(frac=False) 将模型沿着x轴旋转180°,但只颠倒实际坐标。

easyjob.reset\_loc(distance=2)

方便的调整表面模型的位置，distance代表模型的下表面距离盒子的底面的距离

例子:

s.reset\_loc(distance=5)

将底面到底的距离设置为5埃米

easyjob.try\_orthogonal(maxnum=2)

试图通过扩胞使得模型正交，maxnum是最大可能扩胞大小，这里使用了穷举算法来列举可能的倍数，所以maxnum不能太大。

例子:

s. try\_orthogonal(2)

尝试得到一个正交的底面，最大扩大两倍

easyjob.find\_site(how=”all”，distance=1.5,frac=False)

返回表面位点的坐标，how指方式，四种选择是’ontop’,’ ‘bridge’,’hollow’,’all’

frac 为是否返回分数坐标，True返回分数坐标，False返回笛卡尔坐标,distance为位点在z方向上的位置。

返回值是列表，列表里包含了长度为3的数组，为xyz的笛卡尔坐标或者分数坐标

例子:

for i in s.find\_site(‘ontop’):

do something

easyjob.adsorb(mol,loc=[0,0,0],spec\_z=False, theta=[0, 0, 0], v=[0, 0, 0], distance=1.5 ifpca=True,frac=True)

在表面吸附分子，只能在表面模型上使用，mol是pymatgen里的Molecular类,可以通过easyjob.read\_file(‘xxx.xyz’)读取xyz文件得到，

spec\_z 是是否手动指定z轴位置，如果为True，则使用loc指定的z位置，如果为False，则easyjob寻找一个表面位置，根据distance指定的距离表面的位置（单位埃米，笛卡尔坐标）确定z坐标，

frca 是 是否使用分数坐标（推荐分数坐标），这个选项影响loc和v的值

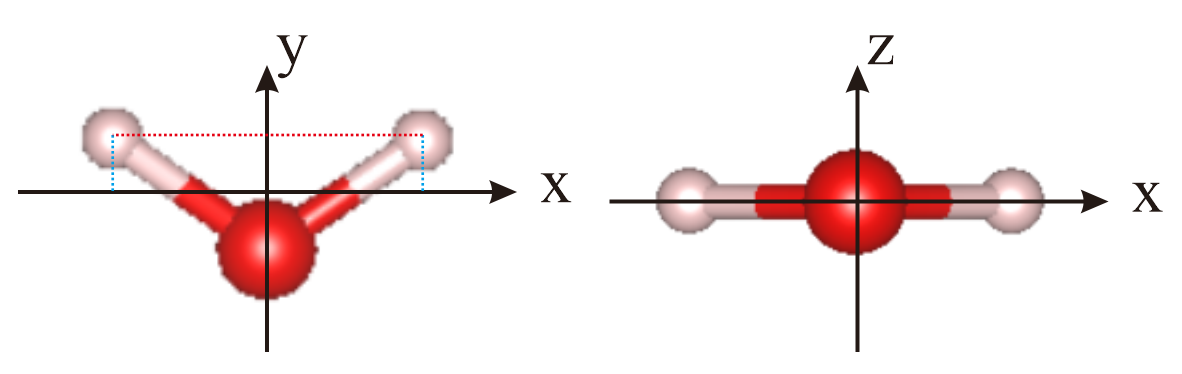
v是平移向量，theta是旋转向量，代表三个轴的旋转角度

ifpca 是 是否对分子实行pca变换（推荐实行），若实行pca变换则：分子的xyz中心为原点，z方向上坐标值方差投影最小，即分子尽可能平躺。

例子:

mol=easyjob.read\_file('H2O.xyz') #读当前目录下水分子xyz文件  
s.adsorb(mol,distance=2) #把水分子平躺放在上表面中心上方2埃米的位置

s.adsorb(mol,distance=3，theta=[0,90,0]) #把水分子竖直放在上表面中心上方3埃米的位置（注意是中心距离表面的距离所以调大），由于第一个轴是方差最大的轴，所以我们旋转第二个轴。如下图：



如图，pca变换之后z轴一定是投影方差最小的轴，x轴是投影方差最大的轴，沿着y旋转90度就可以把分子直立起来。

Index=0

rootpath = os.path.split(os.path.realpath(\_\_file\_\_))[0]

s=easyjob(easyjob.read\_file(‘POSCAR’),os.path.join(rootpath,’test’))

mol=easyjob.read\_file(os.path.join(top,'H2O.xyz'))

for i in s.find\_site(‘ontop’):

index+=1

s1=s.copy()

s1.adsorb(mol,loc=I,spec\_z=True)

foldname=os.path.join(s.foldname,str(index))

s1.foldname=foldname

s1.write\_job()

该例子中读取脚本目录下的POSCAR文件，并在./test/文件夹下生成该结构表面所有的ontop吸附位点上吸附水分子的模型，并使用编号命名

easyjob.split\_from\_element(self,atomselect=['Na'])

按照元素来把结构分为两部分，atomselect是元素列表，在这个列表元素下方的元素被分配给第一个返回结构，上方的被分配给第二个返回结构，incar默认设置为scf，主要用于差分密度建模。

easyjob.split\_from\_z(self,z,frac=True)

按照z坐标来分割，frac是是否使用分数坐标，返回值同上

easyjob.split\_from\_index(sites):

按照sites集合来划分两个easyjob对象，sites中的原子被放在第二个返回值上

信息获取部分

easyjob.c\_to\_f(loc)

通过笛卡尔坐标返回分数坐标

例子:

s.c\_to\_f([3,5,7])

easyjob.f\_to\_c(loc)

通过分数坐标返回笛卡尔坐标

例子:

s.f\_to\_c([0.2,0.5,0.7])

easyjob.get\_cn(i)

获得第i个原子的配位数，为了和VESTA等软件对应，这里从1开始编号

例子:

s. get\_cn (1)

easyjob.get\_ave\_cn()

获得原子的平均配位数,返回一个字典，键是元素名称，值是平均配位数

形如：{'C': 3.30717313701921, 'Ca': 5.525349851524875, 'Mn': 4.9508088628818525, 'O': 3.3629716648189603}

例子:

s. get\_ave\_cn ()

获取平均配位数字典

easyjob.get\_nelect()

获取体系的总电子数，这个命令一般在加电子时使用，如计算离子晶体空位缺陷形成能

例子:

s.incar.set(‘NELECT’,s. get\_nelect()+2)

添加两个电子以计算

easyjob.get\_distance(i,j)

得到两个原子之间的距离，可以得到跨晶胞后的正确距离

easyjob.angle(i,j,k)

得到三个原子的夹角，以原子j为中心

easyjob.frac\_coords()

得到分数坐标的numpy对象

easyjob.cart\_coords()

得到笛卡尔坐标的numpy对象

遍历结构,easyjob对象可以直接遍历

这相当于遍历pymatgen的structure类，你可以使用pymatgen的函数来实现更精确的操作

for i in s:

do something

脚本生成

easyjob.get\_script（）

生成一个vaspjob.pbs 和 一个sub.sh文件，文件的内容可以去file文件夹下修改，以适应自己的体系

Dealjob 简单使用教程

注意！鉴于本人水平有限，请谨慎使用这个类，因为不保证绝对正确。如有bug欢迎指出。另外，请确保你想要得到的属性经过了正确的计算。这个类不检查错误情况。

创建一个dealjob对象，只需要传入文件夹路径即可

d=dealjob(filepath)

任务的值可以直接使用属性来得到，

d.fermi 得到费米能级的值

d.E 得到能量

d.vbm 得到vbm值

d.cbm 得到cbm值

d.gap 得到band gap

d.zpe 得到零点振动能

d.ifreq 得到虚频能量和

d.mag 得到磁矩

d.wf 得到功函数

d.vl 得到真空能级

d.H 得到焓 (298K)

d.G 得到吉布斯自由能 (298K)

d.cvdt 得到热容 (298K)

d.TS 得到TS的值 (298K)

d.time 得到计算耗时

d.nelect 得到体现电子数

d.poscar 得到优化后结构

d.d\_band\_center 得到d带中心

你可以使用函数来计算指定温度热力学性质

dealjob.get\_therm(temp)

返回一个列表E,H,G,cvdt,zpe,TS值，temp是指定温度

dealjob.get\_bader()

得到一个numpy对象，含有每个原子价电荷，注意这需要你在计算文件夹中先运行bader程序

dealjob.get\_diffchg(sufpath,molpath)

输入两个计算了电荷密度的路径，就可以计算差分电荷密度，返回chg对象，并且在当前文件夹写入DIFFCHGCAR文件，这个文件可以用VESTA打开

dealjob. get\_sdos(self, dos\_dict, stack=True, xlim=[-10, 10], ylim=None, sigma=0.15)

在当前目录的上级目录输出dos图片

dos\_dict是要画图的site字典，stack是是否使用堆叠图，sigma是平衡指数，xlim是x轴范围

例如：

d.get\_sdos({‘CO’:[0,5],’Mn’:7})

画0和5号原子的dos并且命名为CO，7号原子命名为Mn

dealjob. get\_edos(self, stack=True, xlim=[-10, 10], sigma=0.15):

直接得到元素dos，用法同上

dealjob. get\_sdos(self, dos\_dict, stack=True, xlim=[-10, 10], ylim=None, sigma=0.15):

直接得到轨道dos 用法同上。

dealjob. get\_cohp(self, xlim=[-4, 4], ylim=[-10, 6]):

得到cohp图，没有深入设计，不建议使用，因为比较丑。使用wxDragon作图更佳

Toolkit类简单介绍

这个类包含一些自用工具，直接使用Toolkit调用

常用的有find，例如

for top,name,path in Toolkit.find(‘cif’):

do something

方便的遍历所有cif，也可以遍历POSCAR等

draw\_reaction\_path(namelist,valuelist,width=1,space=1.5,offsetup=0.1,offsetdown=0.2,color='#1111DD',draw\_value=False):

画电催化火山图，这个可以画很多，然后使用

Toolkit. show\_fig（）

或者

Toolkit. save\_fig（filepath）

来输出或者观察

其他工具可以查看tool文件来了解！欢迎添加新的功能！