Crystal Structure Prediction Module

安装

依赖

- Python 2/3
- Numpy
- Scipy
- scikit-learn
- ASE
- spglib
- pandas
- networkx

在集群上加载`anaconda/3-5.0.1`模块，所有依赖库都已安装

目录结构

```
/your/path/csp-test
├── csp
│   ├── atomgp.py
│   ├── bayes.py
│   ├── crystgraph.py
│   ├── dataset.py
│   ├── descriptor.py
│   ├── fingerprint.py
│   ├── fmodules.cpython-36m-x86_64-linux-gnu.so
│   ├── fmodules.f90
│   ├── fmodules.so
│   ├── __init__.py
│   ├── initstruct.py
│   ├── localopt.py
│   ├── mlpot.py
│   ├── parallel.py
│   ├── readparm.py
│   ├── readvasp.py
│   ├── renewstruct.py
│   ├── serial.py
│   ├── setfitness.py
│   ├── utils.py
│   └── writeresults.py
├── docs
│   └── readme.md
│   └── readme.pdf
└── tools
    ├── csp-clean
    ├── csp-prepare
    ├── fpsetup.yaml
    ├── randSpg
    └── summary.py
```

安装过程

- 加载`Anaconda`:

  ```
  module add anaconda/3-5.1.0
  ```

- 在`~/.bashrc`中设置路径:

  ```
  export PYTHONPATH=$PYTHONPATH:/your/path/csp-test
  export PATH=$PATH:/your/path/csp-test/tools
  export CSP_TOOLS=/your/path/csp-test/tools
  ```

- 设置`ASE`的VASP calculator:

  建一个`run_vasp.py`:

  ```
  import subprocess

  exitcode = subprocess.call("mpiexec.hydra /your/path/to/vasp", shell=True)
  ```

建立`mypps`目录存放vasp赝势，可以用软连接：

```
mypps/
├── potpaw
├── potpaw_GGA
└── potpaw_PBE
```

```
ln -s /your/path/PBE-5.4 mypps/potpaw_PBE
```

三个子目录分别对应LDA, PW91, PBE

也可以加入其他赝势库。

设置环境变量：

```
export VASP_SCRIPT=/your/path/run_vasp.py
export VASP_PP_PATH=/your/path/mypps
```

更多信息见https://wiki.fysik.dtu.dk/ase/ase/calculators/vasp.html#module-ase.calculators.vasp

输入文件

- `inputFold/`: 结构优化输入文件的目录，使用VASP时，把INCAR文件保存维`inputFold/INCAR_*`
- `fpFold/fpsetup.yaml`: 计算结构指纹的设置，一般不用改动
- `input.yaml`

input.yaml的设置

- `input.yaml`例子：

```
calcType: fix
calculator: vasp
setAlgo: bayes
xc: PBE
popSize: 20
numGen: 20
minAt: 6
maxAt: 12
symbols: ['Ti', 'O']
ppLabel: ['_sv', '']
formula: [1, 2]
dRatio: 0.7
randFrac: 0.2
saveGood: 5
pressure: 0
addSym: False
calcNum: 5
numParallel: 2
numCore: 12
queueName: e52692v2ib!
waitTime: 200

### Bayesian
kappa: 2
kappaLoop: 2
scale: 0.0005
parent_factor: 0.1

### BBO
grids: [[2, 1, 1], [1, 2, 1], [1, 1, 2]]
migrateFrac: 0.4
mutateFrac: 0.4
```

参数介绍

- `calcType`: 计算类型

  values: `fix`（定组分）, `var`（变组分）
- `calculator`: 结构优化程序
  values: `vasp`, `gulp`
- `setAlgo`: 结构搜索算法
  values: `bayes`(Bayesian Optimization), `bbo`(Biogeography-Based Optimization)
- `xc`: 交换关联类型

  values: `PBE`, `LDA`, `PW-91`
- initSize: 初代种群数量
- popSize: 种群数量

- numGen: 迭代次数

- minAt: 最小原子数

- maxAt: 最大原子数

- symbols: 元素类型

  例：['Ti', 'O'], 外层是方括号，每个元素用引号括起来
- ppLabel: VASP赝势的后缀

  例：['_sv', ''], 与symbols顺序一致，若无后缀则填入''
- formula: 元素比例

  例： [1, 2]
- fullEles: 若值为`True`,则产生的结构含有'symbols'中所有元素，只在变组分搜索时生效
- eleSize: 变组分搜索时，初代每种单质随机产生的结构数
- volRatio: 随机产生结构时的体积参数
- randFrac: 随机结构比例
- dRatio: 判断原子距离是否过近的标准
- saveGood: 保留结构数
- pressure: 压强(GPa)
- addSym: 产生结构之前是否为父代加入对称性
- exeCmd: 运行结构优化程序的命令（只有`calculator`为`gulp`时才需要）
- calcNum: 结构优化次数
- numParallel: 并行优化结构的数目
- numCore: 结构优化使用的核数
- queueName: 结构优化任务的队列
- waitTime: 检查结构优化任务的时间间隔

Bayesian Optimization 参数

- parent_factor: 父代能量的系数，该参数越大，越接近进化算法
- kappa: 2
- kappaLoop: 2
- scale: 0.0005

BBO 参数

- grids: 切割晶胞的网格
  例：[[2, 1, 1], [1, 2, 1], [1, 1, 2]], 两层方括号
- migrateFrac: 迁移操作产生结构的数目

- mutateFrac: 变异算子产生结构的数目

其他参数（开发中）

- jobPrefix
- permNum
- latDisps
- ripRho
- rotNum
- molDetector

参数默认值

- spacegroup: list(range(1, 231))
- eleSize: 1
- fullEles: False
- volRatio: 1.5
- dRatio: 0.7
- exeCmd: ""
- initSize: parameters['popSize']
- jobPrefix:""
- permNum: 4
- latDisps: list(range(1,5))
- ripRho: [0.5, 1, 1.5, 2]
- molDetector: 0
- rotNum: 5

输出文件

输出文件保存在`results`目录下，分为四种：

- `gen*.yaml`: 每代所有结构
- `pareto*.yaml`: 每代最优结构
- `keep*.yaml`: 每代保留到下一代的结构
- `good.yaml`: 当前最好的`popSize`个结构

提取结构信息需要`summary.py`, 运行方式：`summary.py good.yaml`

计算流程

以TiO_2结构搜索为例：

运行`csp-prepare`, 产生`BuildStruct`, `fpFold`, `inputFold`三个目录以及`summary.py`

- 准备输入文件`input.yaml`（如上所示）和`INCAR_*`(1-5), 把`INCAR_*`放入`inputFold`/
- 提交任务脚本：

```
#BSUB -q e52692v2ib!
#BSUB -n 1
#BSUB -J test-tio2

python -m csp.parallel
```

- 运行过程中的输出信息保存在`log.txt`中，所有结构信息都保存在`results`目录下

注意事项

- 目前`input.yaml`中的参数没有默认值
- 在.bashrc中配置路径时，不要直接复制pdf文档中的字符，可能会有无法识别的空白字符
- 用vasp优化时，需要在INCAR文件中设置KSPACING，与USPEX不同