AISP v1.0.1

AISP 用户手册

AISP Development Team

Chp1 前言

AISP名称由AISP Is Structure Predictor该句的首写字母大写组成，顾名思义，AISP主要功能是对化学，物理，生物等领域的研究对象进行结构预测，其研究对象主要包括三维体相材料，二维平面材料，范德华晶体，一维原子链，零维分子、团簇（开发完善中）。

AISP主要包含初始结构生成AispGen程序和后续结构优化预测AispOp程序两大部分，一个完整的结构预测筛选通常包含两步: 第一步由Aisp生成需要迭代的初始结构，第二步再由AispOp读取初始结构，并进行后续的结构迭代和预测

用户目前可以使用http://121.196.155.178:8080/index （AispGen云端demo版本）进行结构生成初始结构生成程序AispGen基于空间群约束规则,采用简洁明了的JSON作为输入格式，具体使用方法请参考第2章，关于JSON输入格式的介绍请参考附录，AispOp优化程序目前采用基于第一性原理的遗传算法进行后续的结构优化和结构预测，具体使用方法请参考第3章，关于AispGen的使用例子参考第4章。

Chp2 Aisp.in 输入关键词介绍

2.1 生成结构的核心关键词

- SystemName

生成存放初始结构的文件夹目录名，建议用体系的化学元素构成来命名，如【Li】【TiO】AISP会将生成的初始结构存放在这个文件夹下

> 默认值：None
> 是否必须指定：是

输入实例：

JSON 复制代码

```
"SystemName":"TiO"
```

- Composition

生成结构的化学元素和其对应的原子数目:
对于单组分输入格式如下：
如体系中包含4个Ti和8个O原子【"Ti":4,"O":8】，如体系中包含10个C原子，输入参数为【"C":10】

> 默认值：None
> 是否必须指定：是

输入实例Li_{12}：

```JSON
"Composition":{
   "Li":12
   }
```

输入实例Ti_4O_8:

```JSON
"Composition":{
   "Ti":4,
   "O":8
   }
```

如果需要进行变组分搜索，示例输入格式如下，则体系会生成Ti_2O_4和Ti_3O_4两个组分的结构，需要更多组分，依次按照该格式进行添加，中间用逗号分隔，使用该关键词必须注意，会生成较多的结构

```JSON
"Composition":[ {"Ti": 2, "O": 4}, {"Ti": 3, "O": 4}]
```

- NumofComposition

化学组成的重复次数，是数组形式，比如想搜索Ti_2O_4的1倍，3倍，和4倍分子式的所有结构，可以写成如下形式，中间用英文逗号分隔。使用该关键词必须注意，会生成较多的结构

```JSON
"NumofComposition":[1,3,4]
```

> 默认值：[1]
> 是否必须指定：否

- LatticeType

生成结构的类型，LatticeType=3表示进行三维材料的生成，LatticeType=2表示进行二维材料的生成，LatticeType=1表示进行一维材料的生成，LatticeType=0表示进行团簇，分子的生成，LatticeType=4表示进行范德华二维材料的生成，LatticeType=6表示进行分子晶体的生成

> 默认值：None
> 是否必须指定：是

输入实例LatticeType=3

```JSON
  "LatticeType":3
```

- SpgAndNum

指定在某个群约束条件下生成多少结构，比如【"56":15】表示在56群下生成15个结构，参考输入实例1，如果多个群需要分别指定， 比如需要在5群下生成10个结构，56群下生成15个结构，参考输入实例2，特别需要注意的是
对于三维材料搜索（LatticeType=3），输入群的范围是1-230，对于二维材料（LatticeType=2）， 输入群的范围是1-80，对于一维材料（LatticeType=1）输入群的范围是1-75，对于范德华晶体（LatticeType=6），输入群的范围是1-230

如果希望指定一个连续范围的群，请使用0关键字进行指定，比如像生成1-230空间群范围的结构，请使用【"0":"1-230"】

> 默认值：None
> 是否必须指定：是

输入实例1：

```JSON
"SpgAndNum":{
   "56":15}
```

输入实例2：

```JSON
"SpgAndNum":{
   "5":10,
   "56":15
}
```

多群联合搜索

```JSON
"SpgAndNum":{
   "0":"30-50",
}
```

NumPerGroup

每个空间群产生多少个结构，如果需要进行一个范围区间的空间群搜索，多群联合搜索可以指定该关键词

```JSON
"NumPerGroup":2
```

- MaxAttemps

生成结构的尝试次数，如果无法生成结构，可以增加尝试次数

> 默认值：1000
> 是否必须指定：否

输入实例：

```JSON
   "MaxAttemps":1000
```

2.2 生成结构的联合关键词

- VolumeScale

不指定结构体积时，可以使用该参数，程序预估的体积最后会乘以该系数，默认为1

> 默认值：1
> 是否必须指定：否

当LatticeType=3或者LatticeType=6，可以（非必须）联合下列关键词使用

- Volume3D

用于指定生成三维结构的体积范围，单位Ang^3，例如需要指定生成100–150Ang^3大小的结构，输入【100–150】，当该关键词存在时，aisp会尝试生成该体积范围的结构

> 默认值:系统自动生成
> 是否必须指定：否

输入实例

JSON 复制代码

```
"LatticeType":3,
"Volume3D":"100-150"
```

对于LatticeType=2 时，可以（非必须）联合下列关键词使用

- Thickness2D

用于指定生成二维结构的厚度范围，单位Ang^2，例如需要指定生成2–5Ang^2大小的结构，输入【68–157】，当该关键词存在时，aisp会尝试生成该厚度范围的结构

> 默认值:系统自动生成
> 是否必须指定：否

输入实例

JSON 复制代码

```
"LatticeType":2,
"Thickness2D":"2-5"
```

对于LatticeType=2 时，可以采用下面的关键词指定体系的体积，如果不指定， 程序将自动预估体积

- Volume2D

输入实例

```JSON
"LatticeType":2,
"Thickness2D":"2-5"
"Volume2D:"25-45"
```

对于LatticeType=1时，必须联合下列关键词使用

- **Area1D**

用于指定生成一维结构的面积范围，单位Ang^2，例如需要指定截面积生成5-10Ang^2的结构，输入【5-10】，当该关键词存在时，aisp会尝试生成该面积范围的结构

> 默认值:系统自动生成
> 是否必须指定：是

```JSON
 "LatticeType":1,
 "Area1D":"5-10"
```

对于LatticeType=1，也可以采用下面的关键词指定体系的体积，如果不指定， 程序将自动预估体积

- **Volume1D**

输入实例

```JSON
"LatticeType":2,
"Area1D":"5-10"
"Volume1D:"22-35"
```

对于LatticeType=0时，可以（非必须）联合下列关键词使用

- **Volume0D**

用于指定生成零维结构的体积范围，单位Ang^3，例如需要指定生成38-98Ang^3体积的结构，输入【38-98】，当该关键词存在时，aisp会尝试生成该体积范围的结构

> 默认值:系统自动生成
> 是否必须指定：否

对于LatticeType=6筛选二维范德华层状晶体，必须指定层间距关键词

- **InterLayerSpace,层与层之间的笛卡尔坐标距离范围，单位Ang**

> 默认值:NONE
> 是否必须指定：是

输入实例，aisp将筛选层间距3.0–4.0Ang的结构

JSON 复制代码

```
"LatticeType":6,
"InterLayerSpace":"3.0-4.0"
```

也可以联合下列关键词筛选层内带褶皱起伏的结构，如果不设置，随机筛选平面或者褶皱的材料，如果需要筛选带褶皱起伏的材料，一般需要将该值设为大于一个原子层的厚度

- **LayerThickness 每一层的厚度范围，单位Ang**

> 默认值:0
> 是否必须指定：否

输入实例，aisp将筛选层间距3.0–4.0Ang，层内起伏2.0–3.0 Ang的结构

JSON 复制代码

```
"LatticeType":6,
"InterLayerSpace":"3.0-4",
"LayerThickness":"2-3"
```

- **CheckSpg 如果设置为0，生成的结构将不进行对称性的检查**

> 默认值:1
> 是否必须指定：否

- **CheckStuct 如果设置为0，生成的结构将不进行键长的检查（某些时候可以关闭该选项，会得到更多结果）**

> 默认值:1
> 是否必须指定：否

- **CheckMinB 如果设置为0，生成的结构将不进行最小键长的检查（某些时候可以关闭该选项，会得到更多结果）**

> 默认值:1
> 是否必须指定：否

- **CheckMaxB 如果设置为0，生成的结构将不进行最大键长的检查（某些时候可以关闭该选项，会得到更多结果）**

> 默认值:1
> 是否必须指定：否

Chp3 结构搜索程序AispOp使用方法

- 输入文件的文件名与其他文件必须以以下格式进行：
    - z-vasp.pbs
    - aispop /*主程序*/
    - ./----|structure /*structure 是目录*/
        |------ input /*aisp 输入参数文件*/
        |------ INCAR
        |------ POTCAR
        |---- { 1_POSCAR_Spg1_Se12
        .....
        1_POSCAR_Spg80_Se12
        }

    注：z-vasp.pbs, aispop, structure 在同一个目录下， structure 中包含所有的输入文件，INCAR,POTCAR,以及产生好的1_POSCAR_Spg1_Se12此类文件；不要自己修改这些文件的名字，否则程序报错！

- input 文件输入参数
    - 调用第一性程序相关参数：

- pbs_nodefile:　节点列表文件名，一般指定为mf，由pbs作业脚本自动生成
- np:　该次作业总核数，一般由作业脚本确定, 例如pbs脚本中“$NP = 节点数*每个节点核数”
- mpirun:　mpirun所在的绝对路径，在设置好环境变量后用“which mpirun”查看
- exe:　第一性程序所在绝对位置，例如：VASP
- optimizer:　明确使用哪种第一性软件, 目前只有 VASP 可选

- 搜索算法部分：
  - algorithm：搜索算法：目前只有GA基因算法可选
  - GA:　搜索算法中分为“std”，“eli”两类，极力推荐使用“std”标准GA。
  - cross_over:　杂交方式
    - single_point
    - mulit_point
    - dummy
    - average
    - hardness
    - 此处 推荐使用hardness
  - check:　false 为默认值 表示是否进行空间群检查 建议为：true
  - energy_choose: 1 使用内聚能
  - generation_number:　GA 迭代步数 （建议先做2-3次迭代）
  - first_num_generation:　初代使用的种子数
  - num_per_generation:　后代产生的种子数：建议为first_num_generation 的0.45为好
  - lowest_choose_rate:　选择多少最低能量的晶体进入下一代：无默认值，建议0.2-0.3
  - crossover_rate:　杂交比率：无默认值，建议0.5-0.7
  - mutation_rate :　突变比率：无默认值，建议0.1-0.2

AispOp输入参数文件input例子

PHP 复制代码

```
pbs_nodefile = mf
np = 24
mpirun = /public/home/users/ustc002/bin/intelmpi/impi/5.0.3.049/intel64/bin/mpirun
exe = /public/home/users/ustc002/bin/vasp.5.4.4_2d/build/std/vasp
algorithm = GA
optimizer = VASP
re_start = 0
GA = std
cross_over = hardness
lattice_type = 2
generation_number = 3
first_num_generation = 64
num_per_generation = 30
check = true
energy_choose = 1
lowest_choose_rate = 0.3
crossover_rate = 0.6
mutation_rate = 0.1
```

PBS输入例子

```bash
#!/bin/sh
#PBS -N vasp
#PBS -l nodes=1:ppn=24
#PBS -q snode
#PBS -e err
#PBS -o out
REAL_NP_PER_NODE=24   #6 MPI on each node
export OMP_NUM_THREADS=1  #call 4 openmp threads
cd $PBS_O_WORKDIR
#generate nodelist
rm -rf $PWD/structure/mf 1>/dev/null 2>&1
for i in `cat $PBS_NODEFILE | sort | uniq`
do
                for j in `seq 1 $REAL_NP_PER_NODE`
                                do
                                echo $i >> $PWD/structure/mf
                                done
done
#nodelist done
cd $PBS_O_WORKDIR
NP=`cat $PBS_NODEFILE | wc -l`
chmod +x ./aispop
./aispop > runlog
```

Chp4 结构生成输入文件aisp.in实例

下面是一些aisp.in输入模板，可以直接使用，也可以修改使用

• 三维体系（LatticeType=3）

生成Ti_2O_4 的三维结构，体系包含2个Ti原子，4个氧原子

JSON 复制代码

```
{
"SystemName":"Ti2O4",

"SpgAndNum":{
   "136":5},


"Composition":{
   "Ti":2,
 "O":4
   },


 "LatticeType":3,

"MaxAttemps":1000


 }
```

金刚石结构 (C8)

JSON 复制代码

```
{
"SystemName":"Dim",

"SpgAndNum":{


    "227":3

},


"Composition":{
    "C":8
   },


"LatticeType":3,

"MaxAttemps":1000,

 "CheckStuct":1

 }
```

生成BN的体相结构

JSON 复制代码

```
{
"SystemName":"BN",

"SpgAndNum":{
   "194":15},


"Composition":{
   "B":2,
   "N":2},


 "LatticeType":3,

"MaxAttemps":1000


 }
```

- **三维范德华晶体（LatticeType=6）**

生成黑磷的二维范德华结构

JSON 复制代码

```
{
"SystemName":"p8",

"SpgAndNum":{
   "64":6},


"Composition":{
   "P":8
   },


 "LatticeType":6,

 "InterLayerSpace":"3-3.6",


 "LayerThickness":"1.5-2.5",

"MaxAttemps":5000


 }
```

- **二维单层（LatticeType=2）**

类石墨烯结构

JSON 复制代码

```json
{
"SystemName":"C",

"SpgAndNum":{
   "80":30},


"Composition":{
  "C" : 2
 },


 "LatticeType":2,

 "MaxAttemps":1000


 }
```

CdS

JSON 复制代码

```json
{
"SystemName":"CdS",

"SpgAndNum":{
   "164":3},


"Composition":{
  "Cd" : 2,
  "S" : 2

},

 "LatticeType":3,


 "MaxAttemps":1000


 }
```

In2Se 单层

JSON 复制代码

```
{
"SystemName":"InSe",

"SpgAndNum":{


    "78":15

},


"Composition":{
    "In":2,
     "Se":2
   },


"LatticeType":2,



"MaxAttemps":2000,

 "CheckStuct":1

 }
```

- 一维原子链（LatticeType=1）

JSON 复制代码

```json
{
"SystemName":"CdS",

"SpgAndNum":{
   "60":10},


"Composition":{
  "Cd" : 2,
  "S" : 2

},

 "LatticeType":1,



 "MaxAttemps":1000


 }
```

多空间群联合生成结构

TiO2（6个原子）

JSON 复制代码

```
{
"SystemName":"Ti2O4",

"SpgAndNum":{
    "136":5,
    "53":5
},


"Composition":{
   "O":4,
    "Ti":2
   },


 "LatticeType":3,


"MaxAttemps":1000


 }
```

多空间群，多分子式生成结构

JSON 复制代码

```
{
"SystemName":"Ti2O4",

"SpgAndNum":{
   "0":"30-40"},


"Composition":{
   "Ti":2,
 "O":4
   },

"NumofComposition":[1,3,4],

"NumPerGroup":2,


 "LatticeType":3,


"MaxAttemps":1000


 }
```

多空间群，多分子式，同时变组分生成结构

JSON 复制代码

```
{
 "SystemName": "TiO"
 "LatticeType": 3,

 "NumofComposition": [1, 2],


 "MaxAttemps": 1000,
 "SpgAndNum": {"151": 2, "131": 2},

"Composition":[ {"Ti": 2,"O": 4}, {"Ti": 3,"O": 4 }],


}
```

Appendix 附录 JSON输入格式简介

JSON是目前最流行的一种数据交换格式，json是一种轻量级的数据格式，不是一种编程语言。可以简化表示复杂数据结构的工作量。

JSON语法

①json字符串必须使用双引号
②没有声明变量
③没有末尾分号
④json对象中的属性名必须加双引号

数据交换格式方式

- 基本类型
- 数组类型
- 对象嵌套

实例分析

基本类型

{"键"：值, "键"："值",...}，以大括号开始，键的名称加上冒号，然后跟上对应的的值，若有其他键值对则以逗号进行分割。

JSON 复制代码

```json
{
    "name": "张三",
    "age": 18,
    "sex": true
}
```

数组类型

[{"键"：值，"键"："值"},{"键"：值，"键"："值"},...]，以中括号开始，其间的数据通过逗号进行分割。

JSON 复制代码

```json
[
    {
        "name": "张三",
        "age": 18,
        "sex": true
    },
    {
        "name": "李四",
        "age": 19,
        "sex": false
    }
]
```

对象嵌套

由上面两种类型，因为值的不固定性，可以演变出各种各样的嵌套类型。

JSON 复制代码

```json
{
    "name": "teacher",
    "computer": {
    "CPU":"intel7,
    "disk": "512G"
    },
    "students": [
        {
            "name": "张三",
            "age": 18,
            "sex": true
        },
        {
            "name": "李四",
            "age": 19,
            "sex": false
        }
    ]
}

```

JOSN的6种数据类型

1. string：字符串，必须要用双引号引起来。
2. number：数值，与JavaScript的number一致，整数（不使用小数点或指数计数法）最多为 *15* 位。小数的最大位数是 *17*。
3. object：JavaScript的对象形式，{ key:value }表示方式，可嵌套。
4. array：数组，JavaScript的Array表示方式[value]，可嵌套。
5. true/false：布尔类型，JavaScript的boolean类型。
6. null：空值，JavaScript的null。