Pauling's rules guided Monte Carlo search (PAMCARS)
Version 1.0

Yang Zhong

Contents

Chapter 1 What is PAMCARS?

1.1 Introduction to PAMCARS

The full name of PAMCARS is “Pauling's rules guided Monte Carlo search”. PAMCARS uses Pauling's rules and Monte Carlo iteration to predict the crystal structures. Crystal structures generally obey Pauling's rules. PAMCARS can generate a large number of random crystal structures conforming to Pauling's rules by Monte Carlo iteration. We can compare the quality of these random crystal structures based on the number of chemical bonds and some structure constraints, and then just choose the best crystals among them for optimization. This algorithm can predict the ground-state crystal structures and a large number of metastable crystal structures at a lower computational cost. PAMCARS is suitable for predicting the large crystal structures containing multi elements.

1.2 How to install PAMCARS

PAMCARS is written in C ++ codes that follow the C ++11 standard. Before installing PAMCARS, make sure the mpi parallel environment was installed in Linux, such as openmpi and mpich; and make sure the GCC compiler of version 4.8 or above was installed in Linux. Then in the directory where PAMCARS.tar.gz is located, use the following commands to complete the compilation and installation:

tar –zxvf PAMCARS.tar.gz

cd PAMCARS

make

1.3 How to run PAMCARS

Prepare two files, Rcrystal.in and Iterations.in, and then enter mpirun –np N PAMCARS. You can also use a job script to submit a parallel task.

1.4 Related tools

There is no need to optimize all random crystals generated by PAMCARS. The package of crystalpicker released with PAMCARS can compare the quality of these random crystal structures and select the best crystals for optimization. The selected random

crystal structures can be optimized by VASP. Chapter 4 gives a sample shell script that calls VASP to optimize the selected random crystals in batches. The package of VASPAnalyser released with PAMCARS can be used to analyze the results generated by VASP in batches. The use of crystalpicker and VASPAnalyser is discussed in chapter 3 and chapter 5 respectively.

Chapter 2 Input parameters of PAMCARS

The input parameters of PAMCARS are included in two files: Rcrystal.in and Iterations.in. The inputs related to lattice parameters and space group symmetry are included in Rcrystal.in. The inputs related to crystal structure constraints and MC iteration are included in Iterations.in. These input parameters can be given in any order or omitted when default values are used. Lines starting with '#' are treated as comments.

2.1 Input parameters in Rcrystal.in

2.1.1 composition

Example: composition = Ba2Al2F4B2O6
composition = Li6Nb6O18

composition tag provides the number of atoms of each element in a unit cell. Note that for elements with only one atom in a unit cell, "1" cannot be omitted.

2.1.2 spacegroups

Example: spacegroups = 10
spacegroups = 30-32
spacegroups = 10, 30-32, 50-70

spacegroups tag provides a range of space group numbers which the generated random crystal structures belong to. spacegroups tag can be specified as a set of any numbers between 1 and 230.

2.1.3 latticeMins latticeMaxes

Example: latticeMins = 3.0, 3.0, 3.0, 40.0, 40.0, 40.0
latticeMaxes = 13.0, 13.0, 13.0, 150.0, 150.0, 150.0

latticeMins tag and latticeMaxes tag give the minimum and maximum values of the lattice parameters a, b, c, α, β, γ of the generated random crystal structures. The unit of the lengths is Å, and the unit of the angles is º.

2.1.4 minVolume maxVolume

Example: minVolume = 200
maxVolume = 300

minVolume tag and maxVolume tag give the minimum and maximum volume of a unit cell of a random crystal structure in $Å^3$.

2.1.5 numOfEachSpgToGenerate

Example: numOfEachSpgToGenerate = 10

numOfEachSpgToGenerate tag gives the maximum number of random crystals generated in each space group.

2.1.6 forceMostGeneralWyckPos

Example: forceMostGeneralWyckPos = true (default)
forceMostGeneralWyckPos = false

If forceMostGeneralWyckPos = true, the most general Wyckoff position will be used at least once when generating a space group for a random crystal. If the most general Wyckoff position is not used at least once, the space group of the random crystals cannot be guaranteed to be consistent with the space group requested.

2.1.7 forceWyckPos

Example: forceWyckPos Ba = g
forceWyckPos Mg = a

forceWyckPos tag allows users to force an element to a specific Wyckoff position. If the users want to assign an element to the same Wyckoff position multiple times, just repeat the tag multiple times (i.e. add "forceWyckPos Mg = a" multiple times on the new lines).

2.1.8 maxAttempts

Example: maxAttempts = 1000

maxAttempts is the maximum number of attempts to generate a random crystal for each space group.

2.1.9 outputDir

Example: outputDir = Al2O3_rstruct

outputDir sets the directory to store the output random crystals. The file name of the

POSCAR file stored in OutputDir is ‘poscarFilename’_’spg’-’i’.vasp, where poscarFilename = composition, spg is the space group of this POSCAR file, and i is the index of this POSCAR file.

2.1.10 verbosity

Example: verbosity = n
verbosity = r (default)
verbosity = v

verbosity indicates how much output to print in the log file PAMCARS.log. ' 'n' is no output, 'r' is regular output, and 'v' is verbose output. The information printed in the log file is often not important. Writing data to the log file frequently will affect the speed of the program, so you can set verbosity to 'n' to avoid the problem.

2.2 Input parameters in Iterations.in

2.2.1 delta_a delta_b delta_c delta_Alpha delta_Beta delta_Gamma

Example: delta_a = 0.2
delta_Alpha = 0.05

delta_a, delta_b, delta_c, delta_Alpha, delta_Beta and delta_Gamma indicate the magnitude of changes in the lattice parameters a, b, c, α, β, γ in every MC iteration step. For example, the new lattice parameter a_n generated by the nth MC iteration satisfy $\left|a_n - a_{n-1}\right| / a_{n-1} < \text{delta\_a}$.

2.2.2 maxDisplacementOfAtoms

Example: maxDisplacementOfAtoms = B:0.5, Ba:0.5, O:0.5, Al:0.5, F:0.5

maxDisplacementOfAtoms tag indicates the maximum displacement (in unit of Å) of each atom in MC iterations.

2.2.3 defaultMaxDisplacementOfAtoms

Example: defaultMaxDisplacementOfAtoms = 0.5

defaultMaxDisplacementOfAtoms tag indicates the default maximum displacement (in unit of Å) of atoms in MC iterations. If there is an element with no maximum

displacement set in the maxDisplacementOfAtoms tag, the maximum displacement of the element will use the value set in defaultMaxDisplacementOfAtoms tag.

2.2.4 maxNumOfSharedVertexes

Example: maxNumOfSharedVertexes= Ba-Ba:3,Ba-Al:2,Al-Al:1

The maximum number of anion vertices that can be shared between the coordinated polyhedrons.

2.2.5 maxCoordinationNum

Example: maxCoordinationNum = Si:4,Al:6

Maximum coordination number of the cations.

2.2.6 bondingPairs

Example: bondingPairs= Al-F:2.0,Al-O:2.0; B-O:1.6,B-F:1.5; Ba-F:3.0, Ba-O:3.2

The maximum bond lengths of the bonding atom pairs on different coordinated polyhedrons which are separated by ';'. If the distance between two atoms in a bonding atom pair is less than the maximum bond length, the two atoms can be considered to form a chemical bond. The MC iteration will be performed in the order of the coordinated polyhedrons in the bondingPairs tag. It is suggested to perform MC iterations on the coordinated polyhedrons with shorter chemical bonds first. This is because the coordination number of the coordinated polyhedrons with shorter chemical bonds grows more slowly in the iteration than that of coordinated polyhedrons with longer chemical bonds.

2.2.7 numOfIterations

Example: numOfIterations= `30;20;20`

The number of iterations for each coordinated polyhedron in bondingPairs tag. Note that the number of parameters in numOfIterations tag is the same as the number of coordinated polyhedrons separated by ';' in bondingPairs tag.

2.2.8 minDistanceOfAtoms

Example: minDistanceOfAtoms = F-F:2.0,F-B:1.2,F-O:2.3
minDistanceOfAtoms = B-B:2.3,B-O:1.2
minDistanceOfAtoms = O-O:2.0

Minimum distances between atoms in Å. This tag can be used multiple times and the program will automatically merge these parameters together.

2.2.9 defaultMinDistanceOfAtoms

Example: defaultMinDistanceOfAtoms = 0.8

Default minimum distances between atoms in Å. If the minimum distance between two atoms is not explicitly set in minDistanceOfAtoms tag, the default minimum distance between the two atoms will be used.

2.2.10 maxNumOfAtomsOnPolyhedron

Example: maxNumOfAtomsOnPolyhedron= Al: F/2, O/4; B: F/1

The maximum numbers of the bonding atoms on the coordinated polyhedrons. Different coordinated polyhedrons are separated by ‘;’. Different coordinated polyhedrons are separated by ‘;’. The atom on the left side of ‘:’ is the cation in the center of the coordinated polyhedron. The atom on the right side of ‘:’ is the anion vertex on the coordinated polyhedron and the number after '/' is its maximum number.

2.1.11 useIterationStartCriteria

Example: useIterationStartCriteria = true
useIterationStartCriteria = false (default)

When useIterationStartCriteria = true, the program will judge whether to start MC iteration based on the number of chemical bonds in the initial random crystals. When useIterationStartCriteria = true, the program will generate a certain amount of trial random crystal structures and cache several maximum numbers of chemical bonds of these crystals as “iteration start criteria”. If the number of chemical bonds of an initial random crystal is lower than all cached numbers of chemical bonds, the initial crystal will not be iterated; otherwise, it can be iterated. This parameter can improve the quality

of the random crystals generated by PAMCARS, but it also brings additional computational overhead.

2.1.12 sizeOfTheSetsOfIterationStartCriteria

Example: sizeOfTheSetsOfIterationStartCriteria = 3

When useIterationStartCriteria = true, this parameter determines how many different values of maximum numbers of chemical bonds in trial crystals to be cached as “iteration start criteria”.

2.1.13 numOfTrialCrystals

Example: numOfTrialCrystals = 100

When useIterationStartCriteria = true, this parameter determines how many trial crystals to generate.

2.1.14 skipIfTheSetOfIterationStartCriteriaIsEmpty

Example: skipIfTheSetOfIterationStartCriteriaIsEmpty = true
skipIfTheSetOfIterationStartCriteriaIsEmpty = false (default)

When useIterationStartCriteria = true and skipIfTheSetOfIterationStartCriteriaIsEmpty = true, if the iteration start criteria generated for a certain space group is empty, then skip this space group. If the iteration start criteria for a space group is empty, it means that no trial crystals were generated successfully. It can be expected that it is also hard to generate an initial random crystal in this space group, so this space group can be skipped.

2.1.15 maxAttemptsToGenerateInitialCrystal

Example: maxAttemptsToGenerateInitialCrystal = 50

The maximum number of attempts to generate an initial random structure. If useIterationStartCriteria = true, this parameter should be increased.

Chapter 3 crystalpicker

3.1 How to install crystalpicker

crystalpicker can select the crystals that satisfy some structure constraints from the random crystals generated by PAMCARS, and can sort them in the order of the number of chemical bonds.

crystalpicker is written in C ++ codes that follow the C++11 standard. Before installing crystalpicker, make sure the GCC compiler of version 4.8 or above was installed in Linux. Then in the directory where crystalpicker.tar.gz is located, use the following commands to complete the compilation and installation:

tar –zxvf crystalpicker.tar.gz

cd crystalpicker

make

3.2 How to run crystalpicker

Prepare Crystalpicker.in file and run the executable directly by typing “./crystalpicker”. The crystals selected by crystalpicker will be output to two folders: outputDir__rank and outputDir_tobeRelaxed. The POSCAR files of the random crystals stored in these two folders are the same, but the file names are different. The format of the file names of the POSCAR files in the outputDir_rank folder is 'poscarFilename'_'spg'rank'm'_'n'. Where spg is the space group number of the POSCAR file, m is the ranking of the number of chemical bonds of the POSCAR file in the space group spg, and n is the number of chemical bonds of the POSCAR file. The format of the file names of the POSCAR files in the outputDir_tobeRelaxed folder is 'poscarFilename'_'i', where i is the index of this POSCAR file. 'i' ranges from 1 to the total number of files. When VASP or other molecular simulation software is used to optimize the random crystals, it is more convenient to use outputDir_tobeRelaxed to write a script to optimize these crystal structures in batches. We will give a sample shell script in Chapter 4.

3.3 The parameters in Crystalpicker.in

3.3.1 poscarFilename

Example: poscarFilename = Ba2Al2F4B2O6

The name of the POSCAR files of the random crystals generated by PAMCARS.

3.3.2 spacegroups

Example: spacegroups = 143-194

The range of space group numbers of random crystals generated by PAMCARS.

3.3.3 maxNumOfFilesForEachSpg

Example: maxNumOfFilesForEachSpg = 300

Maximum number of POSCAR files generated by PAMCARS for each space group. Since some POSCAR files may not be generated successfully, the actual number of POSCAR files for a space group may be less than this number.

3.3.4 inputDir

Example: inputDir = BaAlBO3F2_rstruct

inputDir tag sets the directory where the POSCAR files generated by PAMCARS are located.

3.3.5 outputDir

Example: outputDir = BaAlBO3F2_result

The crystals selected by crystalpicker will be stored in outputDir_rank and outputDir_tobeRelaxed.

3.3.6 maxNumOfOutputFilesForEachSpg

Example: maxNumOfOutputFilesForEachSpg = 30 (default)

The maximum number of crystals selected by crystalpicker for each space group.

3.3.7 maxNumOfOutputFilesWithTheSameNumOfBondsForEachSpg

Example: maxNumOfOutputFilesWithTheSameNumOfBondsForEachSpg= 10

The maximum number of output files with the same number of chemical bonds for each space group.

3.3.8 minDistanceOfAtoms

Example: minDistanceOfAtoms = Ba-Ba:3.0, Ba-Nb:3.0, Ba-Na:3.0, Ba-O:2.0

Minimum atomic distances between atoms in each output POSCAR file in Å. If this parameter is the same as the setting in PAMCARS, you don't need to set this parameter again when using crystalpicker.

3.3.9 defaultMinDistanceOfAtoms

Example: defaultMinDistanceOfAtoms = 0.8

The default minimum atomic distance in Å. If the minimum distance between two atoms is not set explicitly in defaultMinDistanceOfAtoms tag, the default minimum distance will be used for them.

3.3.10 localBondingPairs minNumOfLocalBonds

Example: localBondingPair = Al-O:2.0; B-O:1.6, B-F:1.5
minNumOfLocalBonds = 16; 10

The number of chemical bonds specified in localBondingPair in the POSCAR files output by crystalpicker cannot be less than the corresponding value in minNumOfLocalBonds. The values in localBondingPair are the maximum bond lengths of the bonding atom pairs. If the distance between two atoms in a bonding atom pair is less than the maximum bond length, the two atoms can be considered to form a chemical bond.

3.3.11 globalBondingPairs minNumOfLocalBonds

Example: globalBondingPairs = Al-F:2.0,Al-O:2.0,B-O:1.6,B-F:1.5, Ba-F:3.0, Ba-O:3.2
minNumOfGlobalBonds = 30

globalBondingPairs tag sets the maximum bond lengths of all chemical bonds in POSCAR. The total number of chemical bonds in the POSCAR files output by crystalpicker cannot be less than minNumOfGlobalBonds.

3.3.12 maxNumOfSharedVertexes

Example: maxNumOfSharedVertexes= Ba-Ba:3,Ba-Al:2,Al-Al:1

The maximum number of anion vertices that can be shared between the cations of the coordinated polyhedrons.

3.3.13 forbiddenPyramidalPolyhedrons

Example: forbiddenPyramidalPolyhedrons = Si,Ti

The shape of the coordinated polyhedrons of the cations specified by this parameter cannot be pyramidal.

3.3.14 minVolumeOfPolyhedrons

Example: minVolumeOfPolyhedrons = Ti:0.6, Si:0.6

This tag sets the minimum volumes (in unit of $Å^3$) of the coordinated polyhedrons of the cations listed.

3.3.15 minCoordinationNumber

Example: minCoordinationNumber = B:3, Al:6

This tag sets the minimum coordination number of the cations.

3.3.16 maxCoordinationNumber

Example: maxCoordinationNumber = Si:4,Ti:6

This tag sets the maximum coordination number of the cations.

3.3.17 maxNumOfAtomsOnPolyhedron

Example: maxNumOfAtomsOnPolyhedron= Al: F/2, O/4; B: O/3, F/1

This tag can set the maximum number of atoms of particular species on a coordinated polyhedron. Different coordinated polyhedrons are separated by ';'. On the left side of ':' is the cation in the center of the coordinated polyhedron, on the right side of ':' is the anion vertex on the coordinated polyhedron. The number on the right side of '/' is the maximum number of the atoms on the left side of '/'.

3.3.18 xtalComp

Example: xtalComp= true
xtalComp= false (default)

If xtalComp = true, crystalpicker will perform a similarity comparison on the random crystals. The optimized structures of similar crystal structures are usually the same, so there is no need to output too many crystals from each similar structure type.

3.3.19 maxNumOfOutputFilesForEachSimilarStructureType

Example: maxNumOfOutputFilesForEachSimilarStructureType = 5

If xtalComp = true, this tag sets the maximum number of POSCAR files that can be output for each similar structure type.

3.3.20 lenTol angleTol posTol

Example: lenTol = 2.5
angleTol = 3.0
posTol = 0.6

lenTol and posTol are in Å and angleTol is in degree(º). If the difference between the cell lengths of two crystal structures is greater than lenTol, or the angle difference is greater than angleTol, the two structures are not similar. If lenTol and angleTol are satisfied, then check whether the positions of the atoms in the two crystals match within a tolerance of posTol under the assumption that they have the same lattice parameters.

Chapter 4 Structure optimization

The random crystals output by crystalpicker also needs to be optimized by molecular simulation software such as VASP. Here, a sample shell script that automatically performs structure optimization is given as follows:

```
#!/bin/bash

#Directory of POSCAR files to be optimized

dirOfStructuresToRelax=" Ba4Tb4O12_toBeRelaxed"

#Directory to store result files of optimized crystals

dirToStoreTheRelaxedStructuresAndResultFiles="Ba4Tb4O12-relaxed"

#The prefix of the name of the POSCAR files to be optimized

prefixOfTheFilenamesOfStructures="Ba4Tb4O12"

#Starting index of the POSCAR files to be optimized

minIndexOfFile=1

#End index of the POSCAR files to be optimized

maxIndexOfFile=100

#Files to be saved after each structure optimization

resultFilesToStore="POSCAR CONTCAR OUTCAR vasp.out"

#Old files that need to be deleted after each structure optimization

resultFilesToDelete="POSCAR CONTCAR OUTCAR vasp.out WAVECAR CHG CHGCAR"

#Path of VASP executable

vasp_path = /home-gk/users/nsgk114_KYF/quantumlab/V5.4/vasp_std

for ((i=$minIndexOfFile;i<=$maxIndexOfFile;i++))

    do

        mkdir -p $dirToStoreTheRelaxedStructuresAndResultFiles/${prefixOfTheFilenamesOfStructures}_$i

        cp $dirOfStructuresToRelax/${prefixOfTheFilenamesOfStructures}_$i vasp ./POSCAR

        mpirun -np 24 $vasp_path &> vasp.out

        cp $resultFilesToStore $dirToStoreTheRelaxedStructuresAndResultFiles/${prefixOfTheFilenamesOfStructures}_$i/

        rm $resultFilesToDelete

done
```

The users need to modify this script according to the specific structure optimization tasks and the operating system environment used. The users can also write a PBS job script based on this shell script. The users need to put the INCAR, KPOITS and POTCAR files used in structure optimization and the XXX_toBeRelaxed folder where the POSCAR files are stored in the same directory first, and then run this script.

Chapter 5 Results analysis

After optimizing the crystals output by crystalpicker, the result files stored for the crystals can be automatically analyzed using the VASPAnalyser program. To build VASPAnalyser, just type:

tar –zxvf VASPAnalyser.tar.gz

cd VASPAnalyser

make

The input parameters of this software are included in VASPAnalyser.in, and the results are printed to VASPAnalyser.out. There are only four parameters in VASPAnalyser.in:

5.1 rootDirectory

Example: rootDirectory = BABF_relaxed

The root directory where the results files are saved.

5.2 prefixOfSubdirectory

Example: prefixOfSubdirectory = Ba2Al2F4B2O6

The prefix of each subdirectory in rootDirectory. In rootDirectory, the POSCAR, CONTCAR, and OUTCAR files for each crystal structure are stored in their respective subdirectories. Such as Ba2Al2F4B2O6_1, Ba2Al2F4B2O6_2 ...

5.3 numOfSubdirectories

Example: numOfSubdirectories = 134

The number of subdirectories.

5.4 symprec

Example: symprec = 0.05 (default)

The precision used in finding the space group symmetry of the crystal.