SPINNER Documentation

Release 21.1

Sungwoo Kang
Wonseok Jeong
Changho Hong
Weungwoo Hwang
Youngchae Yoon
Seungwu Han

Nov 25, 2021

CONTENTS

CHAPTER

ONE

1. INSTALLATION

1.1 1.1 Download SPINNER

Please download the file from https://github.com/MDIL-SNU/SPINNER

1.2 1.2 Python requirements

SPINNER supports Python `3.6` or higher version. SPINNER utilizes Python modules in the following:

- mpi4py (https://bitbucket.org/mpi4py/mpi4py/)
- PyYAML (https://pypi.org/project/PyYAML/)
- numpy (https://numpy.org/)
- pybind11 (https://github.com/pybind/pybind11/)

You can download these modules by

```
pip3 install mpi4py PyYAML numpy pybind11
```

or directly downloading the source codes from the webpages.

1.3 1.3 C++ compilers

SPINNER supports CMAKE 3.20 or higher version and gcc version 7.3.1 or higer version.

1.4 1.4 Python binding of randSpg and LAMMPS (important)

SPINNER utilizes python binding of randSpg and LAMMPS code. However, those codes are slightly modified to be incorporated with SPINNER, so you need to comiple with source codes provided with SPINNER, not with the original ones.

1.4.1 1.4.1 Install randSpg

To install randSpg, do follows

```
cd /SPINNER-directory/randSpg-vspinner/
mkdir build
cd build
cmake ..
make -j3
```

to bind randSpg with python, do follows

```
cd /SPINNER-directory/randSpg-vspinner/python
mkdir build
cd build
cmake ..
make -j3
cp pyrandspg.cpython* /directory-where-your-python-is/lib/python3/site-packages/
```

1.4.2 1.4.2 Install LAMMPS

This is the most tricky part of the installation steps. You can install python-lammps by following steps below but it may not work depending on your environment. Please look into LAMMPS forum (https://www.lammps.org/forum.html) or LAMMPS manual Ch. 2 (https://docs.lammps.org/Python_run.html) for detailed discussion.

```
cd /SPINNER-directory/lammps-vspinner
cd src
make yes-python
make XXX mode=shlib
make install-python
```

Here, XXX is the name of the make file (Makefile.XXX in src/MAKE directory). Note that LAMMPS have to be installed with serial version (not mpi version). The optimization is recommended for your system but default serial option is also possible.

CHAPTER

TWO

2. EXECUTION OF SPINNER

2.1 2.1 Required input files

Running SPINNER requires one `input file` (yaml format), one `input directory` (potential file should be in it), and `src` directory. All of these files and directories should be in the running directory.

2.1.1 2.1.1 Input file

SPINNNER uses YAML style configuration file. All setting parameters used in SPINNER can be controlled in "XXX.yaml" while XXX is the specific name that user defined. These are the mandatory setting:

```
input_dir:        input_directory_name
output_dir:       output_directory_name
initial_volume:   326.0
structure:
    generation: 400
material:
    Li:   4
    Hf:   2
    N:    4
```

See other tags for *3. Input file details* section.

2.1.2 2.1.2 Input directory

In input directory (`input_dir` in input file), LAMMPS potential file `potential` should be located. (Potential file have to be named `potential`.) Potential should be SIMPLE-NN format (https://github.com/MDIL-SNU/SIMPLE-NN).

2.1.3 2.1.3 src directory

`src` directory should be in the running directory. You can copy and paste the src directory to each running directory or you can run multiple calculations in one running folder.

2.2 2.2 Running SPINNER

You can execute SPINNER by follows

```
cd src
mpirun -np core_number python3 main.py XXX.yaml
```

where XXX.yaml is the input file.

CHAPTER

THREE

3. INPUT FILE DETAILS

3.1 3.1 Directories (required)

```
input_dir:  [input directory name]
output_dir: [output directory name]
```

input_dir is the name of the input directory containing the potential file (see *2. Execution of SPINNER*). output_dir is the output directory which is generated when the code executed.

3.2 3.2 Structure of the evolutionary search (required)

```
structure:
    generation: 1000
    i_population: 80
    population:  80
    num_of_best: 0
    re-relax_best: yes
    population_max: 10000
```

generation is the number of generations (required). population is the number of the structure in the pool (except for survival) in one generation. i_population is the population number in generation 1. Default values are the twice the number of atoms in the simulation cell. num_of_best is the number of structure that survive into the next generation. If it is set to 0, then the calculation does not fix the number, but decide it in every generation by the energy window (see energy_criteria in Ch.3.8 below). re-relax_best decides whether survived structures are relaxed again in each generation or not. population_max is the maximum number of population. If the population number exceeds this value, then the calculation stops.

3.3 3.3 Defining material system (required)

```
material:
    Li: 4
    Hf: 2
    N:  4
```

This tags correpond to [atom type]: [number of atoms in the simulation cell]. The type of atoms must be consistent with that of the potential file.

3.4 3.4 Initial volume (required)

```
initial_volume:  326.0
```

This is the cell volume used in the first generation.

3.5 3.5 Distance constraint

```
distance_constraint:
    Li-Li: 2.1
    Li-Hf: 1.5
    Li-N:  1.2
    Hf-Hf: 2.3
    Hf-N:  1.4
    N-N:   0.7
```

The pairwise distance constraint for atomic pairs. This is applied to both restrain option in structural optimization by LAMMPS and in random structure generation. See scale_factor in Ch.3.7 below for additional setting. The default value is 0.7.

3.6 3.6 Fraction of operators

```
operator:
    random:          0.3
    crossover:       0.5
    all_permutation: 0.1
    latticemutation: 0.1
    con_permutation: 0.0
```

Fraction of operators that are used in one generation. (The sum of them doesn't have to be 1. They are readjusted.) all_permutation is the permutation without restriction, and con_permutation is the restricted permutation for certain atom groups. Restrictions are set with permutation_allow tag in Ch.3.7 below.

3.7 3.7 Specific setting for mutation operators

3.7.1 3.7.1 Random generation

```
random_condition:
  force_general_Wyckoff_site:                          no
  maximum_attempts_for_one_space_group_and_volume:   100
  scale_factor:                                      1.0
  sublattice_genaration:                             0.0
  max_sub_lattice:                                     2
```

force_general_Wyckoff_site determines whether atoms are put into the general Wyckoff sites. If you choose no, then specific Wyckoff sites are also used for structure generation. maximum_attempts_for_one_space_group_and_volume is the number of iteration for generating random structure for fixed space group and lattice volume. If the iteration exceeds this value, then new space group and volumes are chosen randomly. `scale_factor` is the ratio between distance constraints that are used to generate random structure and distance constraints for restrain option in LAMMPS. If `scale_factor` is 1, then the same distance constraints are used for both, and if `scale_factor` is 0, then no distance constraints are used for random structure generation. `sublattice_generation` is the ratio of random structures that is generated by sublattice generation method (not recommended). The `max_sub_lattice` is the maximum number of sublattice (only 2 and 4 are avalialbe).

3.7.2 3.7.2 Crossover

```
crossover_condition:
    num_of_grid_for_cut:              10
    energy_range_for_cut_select:    0.01
    grid_for_shift:                    3
    iteration_for_add_atoms:          50
```

SPINNER rationally choose selection planes and merging those using atomic energies and one-shot energy evaluation. `num_of_grid_for_cut` is the number of grid for each axis direction to generate slab. When choosing the slab, SPINNER chooses low-energy slab with probability proportional to exp(-E*<sub>*ave*</sub>/*) where E*<sub>*ave*</sub> is the average atomic energy of the slab and * is `energy_range_for_cut_select`. When merging slabs, SPINNER considers translational degree of freedom and mirror degree of freedom. `grid_for_shift` is the number of translating vector. For instance, if `grid_for_shift` is 3, then, translation is considered for 3x3 grids. In addition, translation vector is found in more fine grid again by 3x3. For the merged structure, composition is not usually same as the initial composition, so subtracting and adding atoms are required. When subtracting atoms, the highest-atomic-energy atoms are removed first. When adding atoms, we randomly put atoms for `iteration_for_add_atoms` times and select the lowest-energy configuration.

Increasing `num_of_grid_for_cut`, `grid_for_shift`, and `iteration_for_add_atoms` enhances the chance to create the lower energy structure but also increases the computation time.

3.7.3 3.7.3 Conditional permutation

```
permutation_allow:
    group1: [atom1, atom2, atom3 ...]
    group2: [atom4, atom5, ...]
    ...
```

In conditional permutation, only permutation within designated groups are allowed. The number of atoms in one group and the number of group can be freely set.

3.8 3.8 Energy criteria for survival and inheritance

```
energy_criteria:
    energy_cut_for_inheriting_structures: 0.10
    energy_cut_for_best_structures:       0.05
    energy_cut_for_further_relax:         0.50
```

`energy_cut_for_inheriting_structures` is the energy window to choose structures that are inherited to the next generation by mutation operators. `energy_cut_for_best_structures` is the energy window to choose structures that survive to the next generation. `energy_cut_for_inheriting_structures` and energy_cut_for_best_structures are used when structure: re-relax_best is yes (see Ch. 3.2). energy_cut_for_further_relax is the energy criteria to decide whether relaxation proceeds unrestricted relaxation after first relaxation (cell-fix relaxation). All of these paramters have eV/atom as unit.

3.9 3.9 Similarity metric

```
similarity_metric:
    type:           pRDF
    limit:          0.01
    volume_cut:     1000.0
    energy_cut:     1000.0
    gaussian_dist:   0.1
    rdf_grid:        250
```

This is the similarity metric that is used to measure the similarity distance between different structures. For type of the similarity metric, pRDF is only available for now (see J. Chem. Phys. 130, 104504 (2009) for details). limit is the limit that determines whether two structures are same or not. volume_cut and energy_cut is the volume and energy criteria to calculate similarity distance between two structures (units are Å^3 and eV/atom, respectively). gaussian_dist is the Gaussian distribution of the pRDF function (Å). rdf_grid is the bin number of rdf.

3.10 3.10 Relaxation option

```
relax_condition:
    relax_iteration:                              5
    method_of_first_relax:                       cg
    further_calculation_with_accurate_potential: no
    stop_relax_beyond_this_generation:           10
```

Relaxation is performed by relax_iteration × total number of atoms × 3 iterations for both lattice-fix relaxation and lattice-free relaxation. method_of_first_relax is the method for the lattice-fix relaxation (cg/fire/mix is possible while mix is mixing of fire and cg). further_calculation_with_accurate_potential decides whether to further evaluate energy with more accurate potential (when using this tag, potential_accurate file should be provided in the input directory). Relaxation of a structure stops after stop_relax_beyond_this_generation generations.

3.11 3.11 Vacuum constraint

```
Vacuum_constraint:
    apply_vacuum_constraint:   yes
    maximum_vacuum_length:     6.0
    grid:                      1.0
```

SPINNER excludes structures having large vacuum when apply_vacuum_constraint is on. maximum_vacuum_length is the maximum vacuum length allowed. grid is the grid length used in the algorithm (Å).

3.12 3.12 Antiseed option

```
antiseed:
    activation_antiseed:    no
    gaussian_width:        0.2
    selection_gaussian:    0.1
    selection_fraction:    0.5
```

SPINNER switches on antiseed option when `activation_antiseed` is set to `yes`. `gaussian_width` and `selection_gaussian` represents and **<sub>*A*<sub>, which are described equation 1 of our publication [arXiv:2107.02594]. Antiseed option is useful when generating training set for refining NNP but not recommended when proceeding long generation of evolutionary algorithm for finding optimal structure.

3.13 3.13 Continue calculation

```
continue:
    ogirinal_dir:    [original directory]
    continue_num:    100
```

`original_dir` is the directory that one wants to start the calculation. `continue_num` is the starting generation. New calculation results are written in `output_dir`. `original_dir` and `output_dir` can be same.

CHAPTER

FOUR

4. OUTPUT FILES

4.1 4.1 Energy and volume data

- `totalGen`: information of energy, volume, and generation methods of total population.
- `totalbest`: information of energy, volume, and generation methods of the best structure in each generation.
- `BestResults`: information of best results that are survived in the current generation (updated at each generation).
- `best_history`: history of BestResults file.

Energy 10000.0 eV in these files means that the structure violates the geometrical constraints or have the same structure as the other structure in the pool.

4.2 4.2 Structure files

- `POSCARs`: gathered structures before relaxation.
- `CONTCARs`: gathered structures after relaxattion.

4.3 4.3 Time log

- `timelog`: log file containing information of time.
- `specific_time`: log file containing time information of each core used in the calculation.

4.4 4.4 Input information

- `input.yaml`: total input used in the calculation.
- `potential`: potential file used in the calculation.

4.5 4.5 Etc output files

- `ave_atomic_e`: average atomic enegy of each element (used in the crossover algorithm).
- `distance_info`: similarity distance information: format: (generation) (population1) (population2) (distance)
- `random_structure_log`: information of generation time and spacegroup (space group of initial structure not the final structure) of random structure.

4.6 4.6 Directories and files for calculation (does not contain information)

- `popXX` (XX=integer)(directories): lammps calculations are carried in this directory.
- `random_structure` (directory): random structure generations are carried in this directory.
- `potential10`: potential file to generate rdf (generated automatically)

CHAPTER FIVE

5. EXAMPLES

SPINNER includes example files in the directory `examples`. You can run the examples as follows:

```
sh run.sh
```