Tutorial on Spin-Atom Cluster Expansion Using spin_atom_CE3 Package

Eunseok Lee

0. Introduction

This tutorial is to guide the use of spin_atom_CE3 package for the application of the spin-atom cluster expansion for 5 degrees of freedom alloy systems.

- ‘ ’ is used to indicate the name of directories.
- “ ” is used to indicate the name of files and flags.
- **Courier** font is used to indicate executable files and demonstration
- >> indicates the prompt at terminal.

1. Preparation

1.1 Download and extract spin_atom_CE3.tar, which will create ‘spin_atom_CE3’ directory. Or simply clone the spin_atom_CE3 repository from https://github.com/eunseok-lee/spin_atom_CE3.

1.2 There are five directories in ‘spin_atom_CE3’. More specifically, ‘src_clusterlist’, ‘src_data_to_corr_mat3’, ‘src_findcluster3’, and ‘src_predictstructure_ce3’ directories contain four programs to perform the spin-atom cluster expansion, while ‘utils4vasp3’ directory contains utility programs for the data-processing of VASP result. All these programs need to be compiled. Refer to **Sec. 5** for further information on the installation process.

1.3 The spin-atom configuration (atomic species and magnetic moment species on each lattice site) and the corresponding electronic energy (result from DFT or other first-principles calculations) should be provided as the inputs of the cluster expansion. In case you use VASP as your computational tool for electronic energy calculation, utility programs in ‘utils4vasp3’ directory will be useful for data-processing. This tutorial is based on the use of these utility programs and VASP, however you can use any other DFT or first-principles calculation softwares as long as the inputs are provided in appropriate file formats. For the input file formats, refer to README in each sub-directory in ‘spin_atom_CE3’ directory.

2. Data-Processing of VASP Result for the Inputs

2.1 Note that VASP calculations need to perform with “ISPIN = 2” in “INCAR” file to produce magnetic moment on each atom (for further information, refer to VASP website). Each VASP calculation will create several result files. Among these result files, “OUTCAR”, “OSZICAR”, “POSCAR”, and “CONTCAR” are needed to prepare the input files of cluster expansion.

2.2 Create ‘data1’ through ‘dataN’ directories, where N is the number of VASP calculations. Then, copy OUTCAR, OSZICAR, POSCAR, and CONTCAR files from each VASP calculation to each ‘data*’ directory. Collect all ‘data1’-‘dataN’ directories and place them in one directory (let’s say ‘VASP’ directory).

2.3 Copy the executable **extracttotalinfo** from the ‘utils4vasp3’ directory into the ‘VASP’ directory and run **extracttotalinfo** as follows.

```
VASP>> ./extracttotalinfo n1 n2
```

Then the data from OUTCAR, OSZICAR, POSCAR, and CONTCAR files in ‘data$n1’-‘data$n2’ directories will be processed and create *.dat files in each ‘data*’ directory.

2.4 Copy the executable **vasp2datamag** from the ‘utils4vasp3’ directory into the ‘VASP’ directory and run **vasp2datamag** as follows.

```
>> ./vasp2datamag mL mN mM mC
```

Where mL, mN, mM, and mC are the criteria to determine the spin state of Li, Ni, Mn, and Co, respectively. Then the spin and atom configurations and the corresponding energy are read from all *.dat files in all ‘data*’ directories and written into five files, “data_orig.dat”, “magmom_orig.dat”, “nCat_orig.dat”, “E_orig.dat”, and “Ef_orig.dat”. These files are used as input files of cluster expansion, except for “nCat_orig.dat” which is just for information.

2.5 In case the cell needs to be extended to a larger one, copy the executable **extend2largercell** from the ‘utils4vasp3’ directory into the ‘VASP’ directory and run **extend2largercell** as follows.

```
>> ./extend2largercell n1o n2o n3o n1n n2n n3n
```

where n1o, n2o, and n3o indicate the periodicity in a, b, and c axis in present cell, respectively, while n1n, n2n, and n3n indicate the periodicity in a, b, and c axis in the extended cell, respectively. “rp_$n1o_$n2o_$n3o.dat” and “rp_$n1n_$n2n_$n3n.dat” should be provided to specify the fractional coordinate of each cationic lattice site in n1o-by-n2o-by-n3o and n1n-by-n2n-by-n3n cells.

3. Steps of Cluster Expansion (Refer to Sec. 4 for demonstration)

3.1 Generation of the cluster information

Adjust the parameters in param.dat (refer to README in ‘src_clusterlist’ directory for further explanation on parameters) and then run the executable, **run_clusterlist**. Note that param.dat should be placed at the same location as **run_clusterlist**. The result files will be stored in ‘dir_result’ directory, which will be created if not exists.

3.2 VASP calculations

Perform VASP calculations (on the same sized cells) for a few different configurations. The set of the result of these calculations will be called the training set.

3.3 Data-processing of VASP result

Data-process the result of VASP calculations as explained in **Sec. 2**.

3.4 Construction of the correlation matrix

Run the executable **run_data_to_corr_mat3** to convert the spin-atom configurations to the correlation matrix. The spin-atom configurations and the electronic energy must be provided as the inputs for **run_data_to_corr_mat3**. Create ‘dir_inputs’ directory at the same location as **run_data_to_corr_mat3** and then copy “map_to_cluster1.dat”, “map_to_cluster2.dat”, “map_to_cluster3.dat”, and “nlist.dat”, resulted from **Sec. 3.1**, and “data_orig.dat”, “magmom_orig.dat”, “E_orig.dat”, and “Ef_orig.dat”, resulted from **Sec. 2.4**, into the ‘dir_inputs’ directory. You also need to create “param.dat” in the ‘dir_inputs’ directory (refer to README in ‘src_data_to_corr_mat3’ directory for further explanation on the input files and parameters).

The result files of **run_data_to_corr_mat3** will be stored in ‘dir_result’ directory, which will be created if not exists.

3.5 Selection of the representative cluster functions and ECIs

Run the executable **run_findcluster3** to select the most representative cluster functions and the corresponding ECIs. The correlation matrix and the formation energy must be provided as the inputs. Create ‘dir_inputs’ directory at the same location as **run_findcluster3**, copy “corr_mat_ugs.dat”, “usefulcorr_col.dat”, and “Ef_ug.dat”, resulted from **Sec. 3.4**, into the ‘dir_inputs’ directory, and create “param.dat” in the ‘dir_inputs’ directory (refer to README in ‘src_findcluster3’ directory for further explanation on the input files and parameters).

The result files of **run_findcluster3** will be stored in ‘dir_result’ directory, which will be created if not exists.

3.6 Prediction of the lowest energy structures

Run the executable **run_predictstructure_ce3** to predict the lowest energy structures. The selected cluster functions and the corresponding ECIs as well as the cluster informations must be provided as the inputs. Create 'dir_inputs' directory at the same location as **run_predictstructure_ce3** and copy "map_to_cluster1.dat", "map_to_cluster2.dat", "map_to_cluster3.dat", and "nlist.dat", resulted from **Sec. 3.1**, and "cluster_set_min.dat" and "x.dat", resulted from **Sec. 3.5**, into the 'dir_inputs' directory (refer to README in 'src_predictstructure_ce3' directory for further explanation on the input files and parameters).

The result files of **run_predictstructure_ce3** will be stored in 'dir_result' directory, which will be created if not exists.

3.7 check if the predicted structures are new.

3.7.a) If yes, add them to the training data set and repeat 3.2-3.7.

3.7.b) If no, which means the predicted structures already exist in the training data sets, then check the convergence of the cluster expansion in terms of cross validation score. If the cross validation score is beyond the criteria, it implies the cluster expansion on this size of cells cannot provide the sufficient cross validation as well as the reliability of the prediction. In this case, cluster expansion needs to perform on larger size cells. To keep the data from the present training set, run the executable **extend2largercell** (refer **Sec 2.5**).

3.8 Level up to larger size cells

In case of **3.7.b**), cluster expansion needs to perform on larger size cells to achieve convergence. Concatenate "data_new.dat", "magmom_new.dat", "E_new.dat", and "Ef_new.dat", generated from the training set of small cells, to "data_orig.dat", "magmom_orig.dat", "nCat_orig.dat", "E_orig.dat", and "Ef_orig.dat on extended cells", respectively. In this way, the data from the training set of small cells can be transferred into the training set of extended cells.

4. Demonstration

Let's say we are interested in cluster expansion on layered NMC cathodes, which have 5 degrees of freedom (Li/Ni/Mn/Co/Va) on each cationic lattice and the R-3m space group as the base structure –the space group will change depending on the atomic arrangement. We will start from the cell that has 4 cationic sites and 4 oxygen sites, which is corresponding to 2 formula units. This cell will play a unit cell in our modeling.

At first, let's create two directories, one for VASP results, and the other for cluster expansion, as follows.

```
~>> mkdir VASP CEM
```

And then, copy the spin_atom_CE3 programs into the 'spin_atom_ce' directory, as follows.

```
~>> cp -r <location of spin_atom_CE3> CEM
```

Note that the executables can be located at any place as long as the inputs are provided, however, for your convenience, in this demonstration we will keep the all existing structures of 'spin_atom_CE3' directory and run the executables in their corresponding 'src_*' directories.

4.1 Generate cluster information

Although our unit cell has 8 basis (4 cationic sites and 4 oxygen sites), we are interested only in 4 cationic sites. Hence only the atomic species on cationic sites will be considered. However, for the param.dat, we need to specify the number of basis as 8.

Let's start from the smallest cell – 1x1x1 cell, which is corresponding to n1=n2=n3=1. For efficient use of memory, the technique of neighbor list is adapted. If you want to consider all cationic sites as the neighbors of a cationic site, just use large enough values of "neighbor_num_max" and "neighbor_dist_max". "neighbor_num_max 100" means you will consider up to 100th neighbor sites, which is pretty sufficient. "neighbor_dist_max 6.5" means you will consider the cationic sites within the distance of 6.5 Angstrom to a cationic site as neighbor sites. After neighbor list is generated according to the value of "neighbor_dist_max", the value of "neighbor_num_max" will be automatically adjusted to match the size of neighbor list.

The suggested param.dat is as follows.

```
~/CEM/src_clusterlist>> more param.dat
```

```
# lattice vectors
pa      2.915150 0.0      0.0
pb      0.0      5.118490 0.0
pc      0.0      1.708168 4.776830
#
# number of basis
nbasis  8
basisfilename   basis.dat
#
# repetition
n1      1
n2      1
```

```
n3      1
neighbor_num_max        100
neighbor_dist_max       6.5
```

Run the executable **run_clusterlist** as follows.

```
~/CEM/src_clusterlist>> ./ run_clusterlist
```

You will see that ~/CEM/src_clusterlist/dir_result directory was created and contains the result files.

4.2 Construct the correlation matrix

Let’s suppose that we performed 8 VASP calculations on 1x1x1 cells and copied “OUTCAR”, “OSZICAR”, “POSCAR”, and “CONTCAR” files from each VASP calculation to ‘data1’, ‘data2’, …, and ‘data8’ directories.

```
~/VASP>> ls
data1    data2    data3    data4    data5    data6    data7    data8
~/VASP>> ls data1
CONTCAR   OSZICAR   OUTCAR   POSCAR
```

Run two executables, **extracttotalinfo** and **vasp2datamag** for data process, as follows.

```
~/VASP>> ./extracttotalinfo 1 8
~/VASP>> ./vasp2datamag 0.5 0.5 0.5 0.5
```

For the values of mL, mN, mM, and mC, we gave initial guess of 0.5. As the modeling proceeds, we will learn more reasonable values of them.

You will see that “data_orig.dat”, “magmom_orig.dat”, “nCat_orig.dat”, “E_orig.dat”, and “Ef_orig.dat” are created as results.

Move to the location where the executable **run_data_to_corr_mat3** exists and create ‘dir_inputs’ directory.

```
~/CEM/src_data_to_corr_mat3>> mkdir dir_inputs
```

Copy “map_to_cluster1.dat”, “map_to_cluster2.dat”, “map_to_cluster3.dat”, and “nlist.dat” into the ‘dir_inputs’ directory.

```
~/CEM/src_clusterlist/dir_result>> cp map_to_cluster*.dat nlist.dat
~/CEM/src_data_to_corr_mat3/dir_inputs/
```

Copy “data_orig.dat”, “magmom_orig.dat”, “E_orig.dat”, and “Ef_orig.dat” into the ‘dir_inputs’, too.

```
~/VASP>> cp data_orig.dat magmom_orig.dat E_orig.dat Ef_orig.dat
~/CEM/src_data_to_corr_mat3/dir_inputs/
```

Create “param.dat” in ‘~/CEM/src_data_to_corr_mat3/dir_inputs’ directory. The suggested “param.dat” is as follows.

```
~/CEM/src_data_to_corr_mat3/dir_inputs >> more param.dat
```

```
np                       4
neighbor_num             3
ncluster1                1
ncluster2                2
ncluster3                1
ndata                    8
# convex hull parameters
Ndim                     4
near_convh_cutoff        1.0
qhullflags               Q0
# anomaly detection parameters
use_anomaly_detection    0
anomaly_score_crit       1.0
knn                      5
```

“np”, “neighbor_num”, “ncluster1”, “ncluster2”, and “ncluster3” can be read from “~/CEM/src_clusterlist/dir_result/result_param.dat”. Note that the value of “neighbor_num” was adjusted to 3 (we initially gave 100). At this beginning stage, it is not so useful to use anomaly detection, so we will deactivate “use_anomaly_detection” – it will be useful later as the number of data set (“ndata”) becomes greater than 100 – and hence, the given values of “anomaly_score_crit” and “knn” will be meaningless. For detailed information on each parameter, refer to “~/CEM/src_data_to_corr_mat3/dir_inputs/README”.

Once the “param.dat” is prepared, run the executable **run_data_to_corr_mat3** as follows.

```
~/CEM/src_findcluster3>> ./run_data_to_corr_mat3
```

You will see that ‘~/CEM/src_data_to_corr_mat3/dir_result’ directory was created and contains the result files.

* The executable **run_data_to_corr_mat3** can also be run in parallel mode using mpirun or srun depending on your choice of MPI compiler. It will be useful when you need to process many data sets.

4.3 Find cluster functions and ECIs

Create ‘dir_inputs’ directory at the same location as **run_findcluster3** and copy “corr_mat_ugs.dat”, “usefulcorr_col.dat”, and “Ef_ug.dat” from the result of the executable **run_data_to_corr_mat3**.

```
~/CEM/src_findcluster3>> mkdir dir_inputs
```

```
~/CEM/src_dat_to_corr_mat/dir_result>> cp corr_mat_ugs.dat usefulcorr_col.dat
Ef_ug.dat ~/CEM/src_findcluster3/dir_inputs
```

We also need to provide “param.dat” in ‘~/CEM/src_findcluster3/dir_inputs’ directory. The suggested “param.dat” is as follows.

```
~/CEM/src_findcluster3/dir_inputs>> more param.dat
```

```
corr_mat_ugs_filename   corr_mat_ugs.dat
n_corr_mat_ug_row       6
```

```
n_non_singular_col      146
howmanycluster          5
max_iter                10000
kT                      0.15
cvs_tol                 0.01
dispfreq                10
nfu                     2
```

The value of “n_corr_mat_ug_row” and “n_non_singular_col” can be read from “~/CEM/src_data_to_corr_mat3/dir_result/result_param.dat”. Sometimes we may need to examine several different correlation matrixes depending on the definition of clusters, range of neighbors, with or without threebody clusters, etc., that’s why “corr_mat_ugs_filename” is employed as an input parameter. For detailed information on each parameter, refer to “~/CEM/src_findcluster3/README”.

Once the “param.dat” is prepared, run the executable **run_findcluster3** as follows.

```
~/CEM/src_findcluster3>> ./run_findcluster3
```

You will see that ‘~/CEM/src_findcluster3/dir_result’ directory was created and contains the result files.

* The executable **run_findcluster3** can also be run in parallel mode using mpirun or srun depending on your choice of MPI compiler. It will be useful when you find the most representative cluster functions from the training set of large size cells and/or many data sets.

4.4 Predict the lowest energy structures

Now, we will predict the lowest energy structures for all possible combinations of (nLi, nNi, nMn, nCo, nVa) on the condition of nLi+nNi+nMn+nCo+nVa = 4, because we are examining 1x1x1 cell that has 4 cationic lattice sites.

Create ‘dir_inputs’ at the same location as **run_predictstructure_ce3** and prepare input files there.

```
~/CEM/src_predictstructure_ce3>> mkdir dir_inputs
```

```
~/CEM/src_clusterlist/dir_result>> cp map_to_cluster*.dat nlist.dat
~/CEM/src_predictstructure_ce3/dir_inputs/
```

```
~/CEM/src_findcluster3/dir_result>> cp cluster_set_min.dat x.dat
~/CEM/src_predictstructure_ce3/dir_inputs/
```

The suggested param.dat for (nLi, nNi, nMn, nCo, nVa)=(1, 0, 2, 1, 0) is as follows.

```
~/CEM/src_predictstructure_ce3/dir_inputs>> more param.dat
```

```
# control parameters
howmanyLi       1
howmanyNi       0
```

```
howmanyMn       2
howmanyCo       1
newstart        1
max_iter        2000
kT              0.0256
dispfreq        100
# cluster parameters
np              4
neighbor_num    3
howmanyclustercol       5
ncluster1       1
ncluster2       2
ncluster3       1
# initial configuration in case of newstart 0
data_ini_filename       data_initial0.dat
magmom_ini_filename     magmom_initial0.dat
# check with the existing data
check_db        1
data_db_filename        data_orig.dat
corr_mat_db_filename    tmp_corr_mat.dat
ndata           16
```

For detailed information on each parameter, refer to "~/CEM/src_predictstructure_ce3/README".

Once the "param.dat" is prepared, run the executable **run_predictstructure_ce3** as follows.

```
~/CEM/src_findcluster3>> ./run_predictstructure_ce3
```

You will see that '~/CEM/src_predictstructure_ce3/dir_result' directory was created and contains the result files.

* The executable **run_predictstructure_ce3** can also be run in parallel mode using mpirun or srun depending on your choice of MPI compiler. It will be useful when you predict the structures on large size cells.

5. Installation

5.1 The GNU Scientific Library (GSL)

GSL is to calculate the cross validation score and required for the executables **run_data_to_corr_mat3** and **run_findcluster3**. Most high performance computers may already have it because GSL is one of essential package for scientific C(C++)-programs. If your computer doesn't have the installed GSL library, download and install GSL package from https://www.gnu.org/software/gsl. As this package is almost essential, the installation is well guided in the included README and INSTALL files for most GNU/Linux distributions.

5.2 QHull

QHull (www.qhull.org) library is employed to calculate the distance to the convex hull of each data point and thus is required for the executable **run_data_to_corr_mat3**. QHull is implemented in many scientific softwares such as Octave, MATLAB, etc., however it is not common to have the QHull installed on high performance computing systems.

If you have the superuser authority, it will be very easy to install QHull (follow the instruction in README in Qhull package). The compiled Qhull will be added to the default '/usr/local/' directory. Add the '/usr/local/' directory to LD_LIBRARY_PATH in order to link the installed QHull library during the compilation of **run_data_to_corr_mat3**. Refer to **Sec. 5.5.a**).

However, if you don't have the superuser authority, the installation may be challenging. Copy and extract the Qhull package at any location that you can freely access to. Move into the Qhull directory (probably, 'qhull-2015.2') and type "make". Then the static Qhull library, "libqhullstatic_r.a", will be created in the 'qhull-2015.2/lib/' directory. You need to copy and provide this static library when you compile **run_data_to_corr_mat3**. Refer to **Sec. 5.5.b**).

5.3 Designation of Compiler in "Makefile"s

The following steps are based on gcc compiler and openmpi, so that "CC" flag in "Makefile" was set to "CC = gcc" for serial program or "CC = mpicc" for parallel programs in Sec. 5.4 – 5.7. Adjust this flag according to your own choice of compiler. For example, if you have to use Intel Compiler, use "CC = icc" for serial program and "CC = mpiicc" for parallel programs.

5.4 Compilation of **run_clusterlist**

Move into 'src_clusterlist' directory and type "make" at the prompt.

5.5 Compilation of **run_data_to_corr_mat3**

5.5.a) If you have the superuser authority and hence could install the Qhull library at designated location ('/usr/local'). Move into 'src_data_to_corr_mat3' and type "make" at the prompt, then the compiled Qhull library will be linked though the flag "-lqhull_r". Note that you also need to have GSL library path on your LD_LIBRARY_PATH.

5.5.b) Otherwise, you need to provide the compiled static library, "libqhullstatic_r.a", for the compilation of **run_data_to_corr_mat3**. Copy the "libqhullstatic_r.a" from the 'qhull-2015.2/lib/' directory into the 'src_data_to_corr_mat3' directory. Then, move into 'src_data_to_corr_mat3' directory, copy "Makefile_static" to "Makefile", and type "make". Note that you also need to have GSL library path on your LD_LIBRARY_PATH.

5.6 Compilation of **run_findcluster3**

Move into ‘src_findcluster3’ directory and type “make”. Note that you need to have GSL library path on your LD_LIBRARY_PATH.

5.7 Compilation of **run_predictstructure_ce3**

Move into ‘src_predictstructure_ce3’ directory and type “make”.

5.8 Notes for NERSC machines

You need to load modules “impi” and “gsl” as follows.

```
>> module load impi gsl
```

5.8.a) In ‘src_clusterlist’, ‘src_findcluster3’, and ‘src_predictstructure_ce3’ directories, copy “Makefile_nersc” to “Makefile” and type “make”.

5.8.b) In ‘src_data_to_corr_mat3’ directory, copy the libqhullstatic_r.a (the static qhull library that should’ve been compiled separately) into this directory, “Makefile_nersc” to “Makefile”, and type “make”.

5.9 Compilation of utility programs

Three utility programs exist in the ‘utils4vasp3’ directory.

5.9.a) **extracttotalinfo**: this is shell script program. You don’t need to compile it.

5.9.b) **vasp2datamag**: this is written in C. Compile this program as follows.

```
>> gcc -o vasp2datamag vasp2datamag.c -lm
```

5.9.c) **extend2largercell**: this is written in C. Compile this program as follows.

```
>> gcc -o extend2largercell extend2largercell.c -lm
```