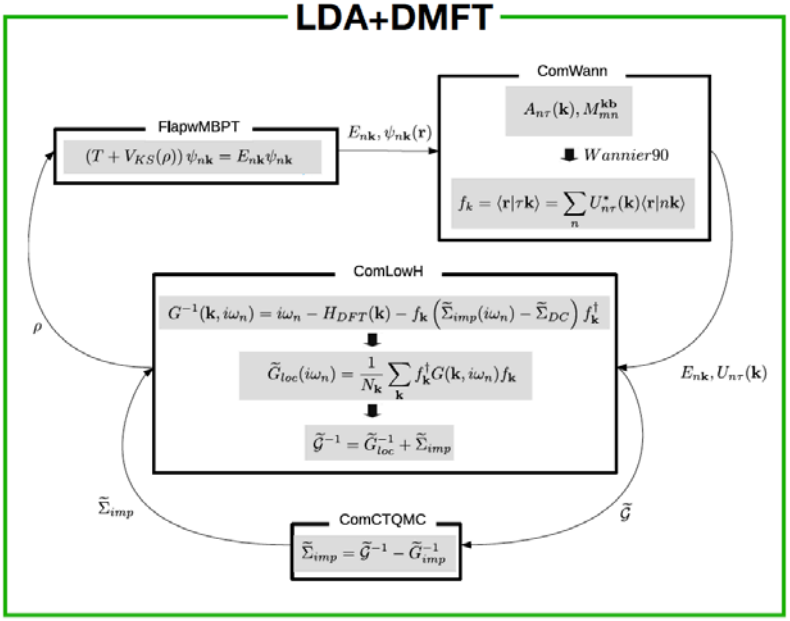
Example: NiO

LDA+DMFT in COMSUITE

We will calculate the electronic structure of a charge transfer insulator NiO within LDA+DMFT. COMSUITE package for LDA+DMFT is composed of four components (softwares). Its work flow is described below.



1. Construction of a Kohn-Sham Hamiltonian within LDA by **FlapwMBPT** (see https://doi.org/10.1016/j.cpc.2017.06.012)
2. Construction of the atom-centered local basis set spanning the low energy Hilbert space by **ComWann utilizing Wannier90 package**
3. Wannier interpolation of the mean-field Hamiltonian and solving the DMFT self-consistent equation by **ComLowH** and **ComCTQMC**
4. Updating the electron density

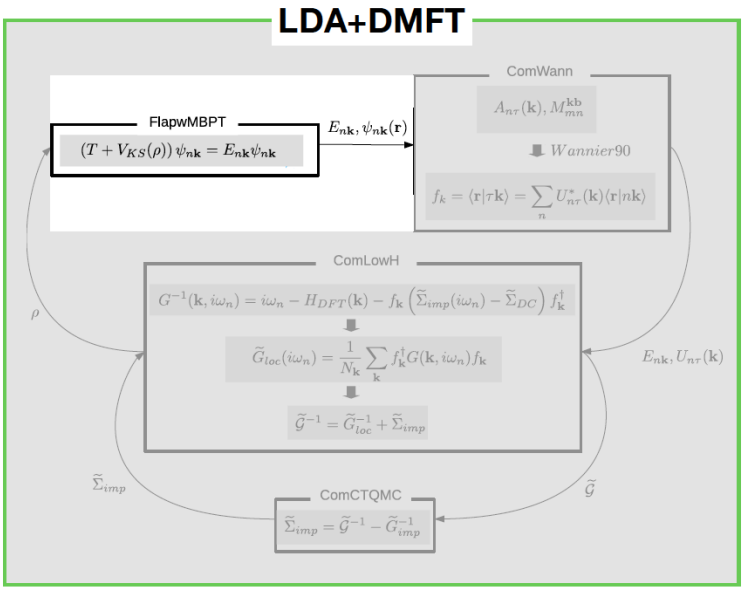
These components (software) along with Wannier90 library are located at install_directory ('install_directory' is described in 'Build and Install' section in Installation of COMSUITE page). To access COMSUITE executables, you should export bin path in your startup shell script.

```
export COMSUITE_BIN=install_directory/bin
```

This tutorial consists of three parts: LDA prerun, LDA+DMFT run, and analysis.

NiO LDA prerun

To run LDA+DMFT, we start with the LDA prerun (unshaded part of following figure).



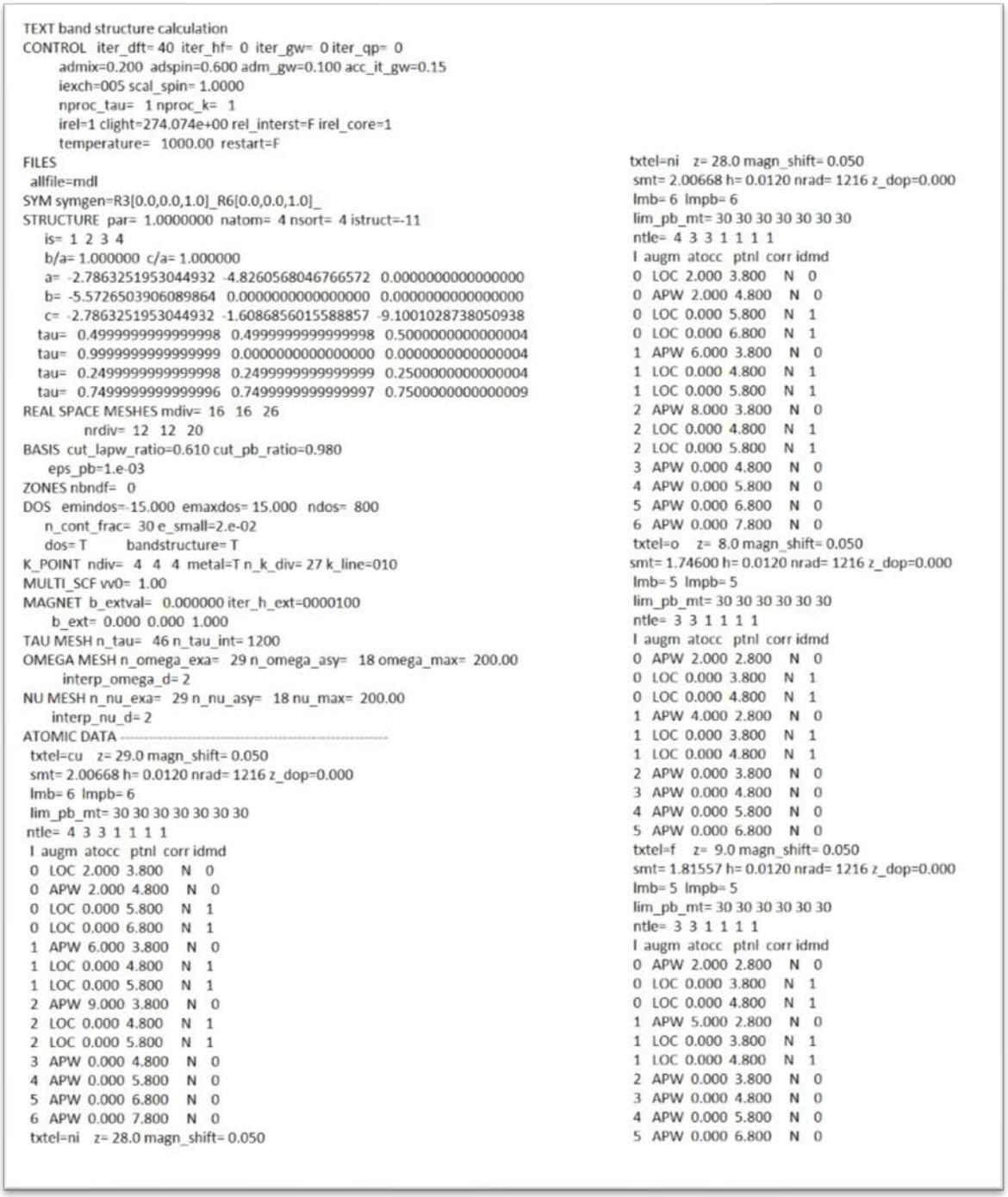
To start the LDA calculation, you need to create a directory named "lda" (or a name of your own choice). Note that this name should be specified in 'comdmft.ini' as will be explained in the next section. Having done this, move to the created directory:

```
$ mkdir lda
$ cd lda
```

Prepare NiO antiferromagnetic NiO cif file which contain distinct AFM ordered atoms in a supercell (in this case Ni1:Cu, Ni2:Ni, O1:O, O2:f in four atom supercell)
Create an input file "ini" using NiO cif file by executing cif2matdelab.py:

```
$ $COMSUITE_BIN/cif2matdelab.py –m dft –k 3 NiO.cif
```

The generated ini file of NiO looks as follows:



```
TEXT band structure calculation
CONTROL  iter_dft= 40  iter_hf= 0  iter_gw= 0 iter_qp= 0
      admix=0.200  adspin=0.600 adm_gw=0.100 acc_it_gw=0.15
      iexch=005 scal_spin= 1.0000
      nproc_tau=  1 nproc_k=  1
      irel=1 clight=274.074e+00 rel_interst=F irel_core=1
      temperature=  1000.00  restart=F
FILES
 allfile=mdl
SYM symgen=R3[0.0,0.0,1.0]_R6[0.0,0.0,1.0]_
STRUCTURE  par=  1.0000000  natom=  4 nsort=  4 istruct=-11
   is= 1 2 3 4
   b/a= 1.000000  c/a= 1.000000
   a=  -2.7863251953044932  -4.8260568046766572  0.0000000000000000
   b=  -5.5726503906089864  0.0000000000000000  0.0000000000000000
   c=  -2.7863251953044932  -1.6086856015588857  -9.1001028738050938
  tau=  0.4999999999999998  0.4999999999999998  0.5000000000000004
  tau=  0.9999999999999999  0.0000000000000000  0.0000000000000004
  tau=  0.2499999999999998  0.2499999999999999  0.2500000000000004
  tau=  0.7499999999999996  0.7499999999999997  0.7500000000000009
REAL SPACE MESHES mdiv=  16  16  26
         nrdiv=  12  12  20
BASIS  cut_lapw_ratio=0.610 cut_pb_ratio=0.980
    eps_pb=1.e-03
ZONES nbndf=  0
DOS  emindos=-15.000  emaxdos= 15.000  ndos=  800
   n_cont_frac=  30 e_small=2.e-02
   dos= T       bandstructure= T
K_POINT  ndiv=  4  4  4  metal=T n_k_div= 27 k_line=010
MULTI_SCF vv0=  1.00
MAGNET  b_extval=  0.000000 iter_h_ext=0000100
     b_ext=  0.000  0.000  1.000
TAU MESH n_tau=  46 n_tau_int= 1200
OMEGA MESH n_omega_exa=  29 n_omega_asy=  18 omega_max=  200.00
      interp_omega_d= 2
NU MESH n_nu_exa=  29 n_nu_asy=  18 nu_max=  200.00
     interp_nu_d= 2
ATOMIC DATA ------------------------------------------------------------
 txtel=cu   z= 29.0 magn_shift= 0.050
 smt= 2.00668 h= 0.0120 nrad= 1216 z_dop=0.000
 lmb= 6  lmpb= 6
 lim_pb_mt= 30 30 30 30 30 30 30
 ntle= 4  3  3  1  1  1  1
 l  augm  atocc  ptnl  corr idmd
 0  LOC  2.000  3.800    N   0
 0  APW  2.000  4.800    N   0
 0  LOC  0.000  5.800    N   1
 0  LOC  0.000  6.800    N   1
 1  APW  6.000  3.800    N   0
 1  LOC  0.000  4.800    N   1
 1  LOC  0.000  5.800    N   1
 2  APW  9.000  3.800    N   0
 2  LOC  0.000  4.800    N   1
 2  LOC  0.000  5.800    N   1
 3  APW  0.000  4.800    N   0
 4  APW  0.000  5.800    N   0
 5  APW  0.000  6.800    N   0
 6  APW  0.000  7.800    N   0
 txtel=ni   z= 28.0 magn_shift= 0.050
 smt= 2.00668 h= 0.0120 nrad= 1216 z_dop=0.000
 lmb= 6  lmpb= 6
 lim_pb_mt= 30 30 30 30 30 30 30
 ntle= 4  3  3  1  1  1  1
 l  augm  atocc  ptnl  corr idmd
 0  LOC  2.000  3.800    N   0
 0  APW  2.000  4.800    N   0
 0  LOC  0.000  5.800    N   1
 0  LOC  0.000  6.800    N   1
 1  APW  6.000  3.800    N   0
 1  LOC  0.000  4.800    N   1
 1  LOC  0.000  5.800    N   1
 2  APW  8.000  3.800    N   0
 2  LOC  0.000  4.800    N   1
 2  LOC  0.000  5.800    N   1
 3  APW  0.000  4.800    N   0
 4  APW  0.000  5.800    N   0
 5  APW  0.000  6.800    N   0
 6  APW  0.000  7.800    N   0
 txtel=o    z=  8.0 magn_shift= 0.050
smt= 1.74600 h= 0.0120 nrad= 1216 z_dop=0.000
 lmb= 5  lmpb= 5
 lim_pb_mt= 30 30 30 30 30 30
 ntle= 3  3  1  1  1  1
 l  augm  atocc  ptnl  corr idmd
 0  APW  2.000  2.800    N   0
 0  LOC  0.000  3.800    N   1
 0  LOC  0.000  4.800    N   1
 1  APW  4.000  2.800    N   0
 1  LOC  0.000  3.800    N   1
 1  LOC  0.000  4.800    N   1
 2  APW  0.000  3.800    N   0
 3  APW  0.000  4.800    N   0
 4  APW  0.000  5.800    N   0
 5  APW  0.000  6.800    N   0
 txtel=f    z=  9.0 magn_shift= 0.050
smt= 1.81557 h= 0.0120 nrad= 1216 z_dop=0.000
 lmb= 5  lmpb= 5
 lim_pb_mt= 30 30 30 30 30 30
 ntle= 3  3  1  1  1  1
 l  augm  atocc  ptnl  corr idmd
 0  APW  2.000  2.800    N   0
 0  LOC  0.000  3.800    N   1
 0  LOC  0.000  4.800    N   1
 1  APW  5.000  2.800    N   0
 1  LOC  0.000  3.800    N   1
 1  LOC  0.000  4.800    N   1
 2  APW  0.000  3.800    N   0
 3  APW  0.000  4.800    N   0
 4  APW  0.000  5.800    N   0
 5  APW  0.000  6.800    N   0
```

One should modify several input keywords as follows:

- modify atom names & corresponding Z value and sort
  in this case:
  is=1 2 3 4 => is=1 2 3 3
  txtel=cu z=29.0 => txtel=Ni1 z=28.0
  txtel=ni z=28.0 => txtel=Ni2 z=28.0
  txtel=O z=8.0 => txtel=O1 z=8.0
  txtel=f z=9.0 => txtel=O2 z=8.0
- iter_dft: The number of DFT iterations. Set to 200.
- admix: charge mixing in DFT iterations. Set to 0.025.
- nproc_tau: The number of MPI processes associated with imaginary time and frequency. Set to 1. For the details on the MPI parallization in FlapwMBPT, please go to FlapwMBPT homepage (https://www.bnl.gov/cmpmsd/flapwmbpt).
- nproc_k: The number of MPI processes associated with parallelization over Brillouin zone k-points. For the present case, set to 16. For the details on the MPI parallization in FlapwMBPT, please goto FlapwMBPT homepage (https://www.bnl.gov/cmpmsd/flapwmbpt).
- mdiv: The real space mesh in the unit cell used for electronic density, Hartree potential, and LDA exchange-corrleation potential evaluations. Set to [28 28 28].

- nrdiv: The real space mesh associated with the interstitial product basis. Set to [20 20 20]
- cut_lapw_ratio: Defines the maximal value of **k +G** for the LAPW basis in the interstitial region.
- ndiv: The number of k-mesh in Brillouin zone. 5 × 5 × 5 k-mesh for the present case.
- k_line: Three numbers defining the direction in the reciprocal space for certain output purposes. Set to 111.

Note also that the total number of MPI processes is nproc_tau × nproc_k which is 16 for the present case.

We also modified Ni and O muffin-tin basis information for a better convergence. One should modify the muffin-tin basis set information in ATOMIC DATA as figure shown above. For the muffin-tin basis set format, please go to FlapwMBPT homepage (https://www.bnl.gov/cmpmsd/flapwmbpt).

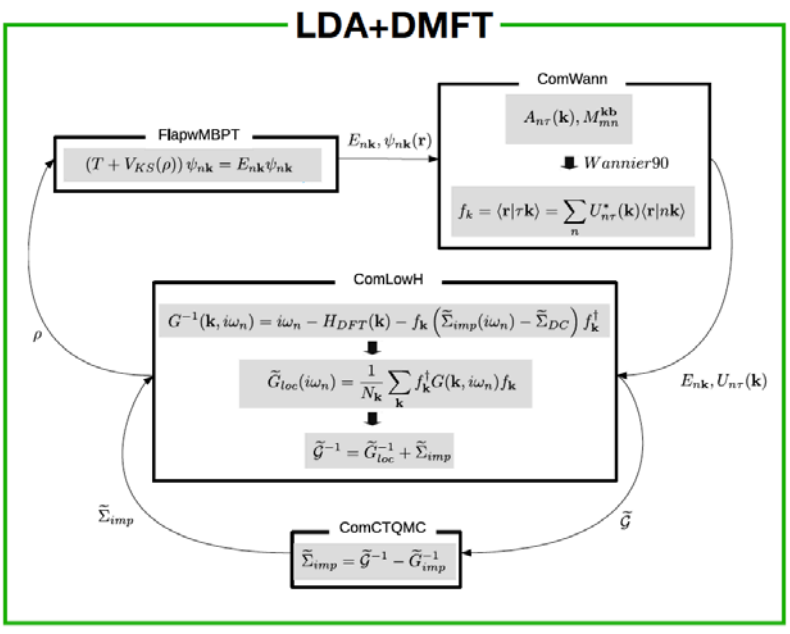
The next step is to run LDA by executing rspflapw.exe. An example of job script to run rspflapw.exe using SLURM is

```
#!/bin/bash -l
#SBATCH -J temp
#SBATCH -p regular
#SBATCH -N 1
#SBATCH -e temp.%j.err
#SBATCH -o temp.%j.out
#SBATCH -L SCRATCH
#SBATCH -C haswell
#SBATCH -t 05:00:00
srun -n 16 $COMSUITE_BIN/rspflapw.exe
```

The total number of processes should be 16 as specified in ini file.

NiO LDA+DMFT run

Once the prerun is finished successfully, the next step is to run the LDA+DMFT calculation (the entire part of the figure below). To run LDA+DMFT, the input file named "comdmft.ini" is needed (see input file section). The calculation of NiO within LDA+DMFT reads output data from the LDA prerun. If you specify the prerun path in comdmft.ini (e.g., '../lda' in this case) correctly, it will read necessary data automatically.



To run LDA+DMFT, move to your work folder, then create lda_dmft folder (you can name what you want) and move to the folder as follows:

```
$ cd ..
$ mkdir lda_dmft
$ cd lda_dmft
```

Then create comdmft.ini (see input file section) file for LDA+DMFT calculation and execute 'comdmft.py' python file in $COMSUITE_BIN. An example of job script using SLURM is

```
#!/bin/bash -l
#SBATCH -J temp
#SBATCH -q regular
#SBATCH -N 12
#SBATCH -e temp.%j.err
#SBATCH -o temp.%j.out
#SBATCH -L SCRATCH
#SBATCH -C haswell
#SBATCH -t 05:00:00
$COMSUITE_BIN/comdmft.py
```

Here 'comdmft.py' is a python script which controls LDA+DMFT calculation. Based on the 'comdmft.ini' input file, 'comdmft.py' generates all necessary input files to run individual programs and execute jobs.

Input file (comdmft.ini)

In order to perform LDA+DMFT calculation, we need only a single input file 'comdmft.ini'. This input file should be written in python dictionary format. All dictionary keys are in small letters. 'comdfmt.ini' is composed of three python dictionaries: 'control', 'wan_hmat', and 'imp':

```
control={'initial_lattice_dir'          : '../lda',
        'method'                : 'lda+dmft',
        'spin_orbit'            : False,
        'mpi_prefix': "srun -n 384",
        'impurity_problem':[[1, 'd'],[2,'d']],
        'impurity_problem_equivalence':[1,-1],
        }

wan_hmat={
       'kgrid': [15, 15, 15],
       'froz_win_min': -10.0,
       'froz_win_max': 10.0,
                  }

imp={'temperature'            : 300, # temperature (in K)
    '1':
    {
   'F0': 10.0,
   'F2': 7.75384615385,
   'F4': 4.84615384615,
   'nominal_n': 8.0,
   'impurity_matrix': [
        [1,0,0,0,0],
        [0,1,0,0,0],
        [0,0,2,0,0],
        [0,0,0,1,0],
        [0,0,0,0,2]
        ],
    'thermalization_time': 1,
    'measurement_time': 60,
    'green_cutoff':  10
    'coulomb': 'full',
    }
```

}

■ **In Control**

These fields contain basic parameters to control LDA+DMFT run.

- 'initial_lattice_dir' : '../lda'
  Enter the path which contains LDA output such as Kohn-Sham eigenvalue and eigenfunctions. It is the LDA prerun folder.

- 'method' : 'lda+dmft'
  Either lda+dmft or lqsgw+dmft. Currently COMSUITE has these two options. Choose 'lda+dmft' for the present work (LDA+DMFT approximation).

- 'spin_orbit': False
  Enter True of False. If False, correlated orbitals correspond to cubic spherical harmonics

$$Y_{lm} = \begin{cases} \frac{i}{\sqrt{2}}\left(Y_l^{-|m|} - (-1)^m Y_l^{|m|}\right), & m < 0 \\ Y_l^0, & m = 0 \\ \frac{1}{\sqrt{2}}\left(Y_l^{-|m|} + (-1)^m Y_l^{|m|}\right), & m > 0 \end{cases}$$

  where Y_l^m is a spherical harmonics.
  if True, correlated orbitals chosen at each correlated atom correspond spin-angular functions |l,i,m>

$$\Omega_{l,i=\pm\frac{1}{2},m} = \sum_{s\pm 1/2} C_{i,s}^{l,m} Y_l^{m-s}(\hat{r}) u_s$$

  where u_s is a spinor, and $C_{i,s}^{l,m} = \langle l, m-s, \frac{1}{2}, s | l+i, m\rangle$.

- 'mpi_prefix': 'srun -n 384'
  MPI prefix used for FlapwMBPT, ComLowH, ComWann, and ComCTQMC. If a different prefix is required for individual program, specifiy the number using 'mpi_prefix_lattice','mpi_prefix_lowh', 'mpi_prefix_wannier', and 'mpi_prefix_impurity'. Note that 384 is the number of total processes we want to use for the present case.

- 'impurity_problem': [[1,'d'],[2,'d']]
  A python list to specify correlated orbitals. The first and second indices indicates the atom index and shell type, respectively. Atom index: in the order listed in the "../lda/coord.xsf".
  Shell index is either "d" or "f". Here, Ni-d shell is treated as an impurity problem.

- 'impurity_problem_equivalence': [1, -1]
  Equivalence of each impurity problem. The value is identified by a positive integer starting from 1. If this value is the same, they are equivalent. For AFM, where two problems are associated with fractional translation and time reversal, you should put opposite sign.

- 'restart': False
  Enter True or False. If True, it will resume the calculation from the previous LDA+DMFT run. The default value is False.

- 'mpi_prefix_lowh':
  MPI prefix for ComLowH. The default value is the one specified in control['mpi_prefix']

- 'mpi_prefix_impurity':
  MPI prefix for the impurity solver. The default value is the one specified in control['mpi_prefix']

- 'mpi_prefix_wannier':
  MPI prefix for ComWann. The default value is the one specified in control['mpi_prefix'].

- 'sigma_mix_ratio':
  Self-energy linear mixing ratio. You can specify any number within 0.0 – 1.0. The default value is 0.5.

- 'max_iter_num_impurity':
  Maximum iteration for the DMFT self-consistent loop. The default value is 50.

- 'proj_win_min':
  Low-energy cutoff to renormalize the projectors. The default value is the one specified in wan_hmat['dis_win_min']

- 'proj_win_max':
  High-energy cutoff to renormalize the projectors. The default value is the one specified in wan_hmat['dis_win_max']

■ **In wan_hmat**:
These fields define frozen window, disentanglement window, and ab initio calculation from which maximally localized Wannier functions (MLFWs) are constructed.

- 'kgrid': [15,15,15]
  Crystal momentum grid for the wannier interpolation of LDA band structure.

- 'froz_win_min': -10.0 eV
  Lower boundary of the inner (frozen) window in eV.

- 'froz_win_max': 10.0 eV
  Upper boundary of the inner (frozen) window in eV.

- 'dis_win_min':
  Lower boundary of the outer (disentanglement) window in eV. The default value is same with wan_hmat['froz_win_min']

- 'dis_win_max':
  Upper boundary of the outer (disentanglement) window in eV. The default value is wan_hmat['froz_win_max'] +40.0

- 'num_iter':
  The number of minimization step for the wannierization process. (gauge dependent part of total spreading). The default value is 0.

- 'dis_num_iter':
  The number of minimization step for the disentanglement process. (gauge independent part of total spreading). The default value is 100.

■ **In imp**:
These fields are related with the Monte-Carlo algorithm and sampling of observables.

- 'temperature': 300
  Simulation temperature in K

- 'F0', 'F2', 'F4': The values of Slater integrals in eV. Note that for "f" shells, 'F0', 'F2', 'F4', and 'F6' should be specified.

- 'nominal_n': Nominal occupancy associated with the impurity shell. This value is required since we adopt the so-called "nominal double counting" for LDA+DMFT which reads:

  $$\tilde{\Sigma}^{DC} = U(N_0 - \frac{1}{2}) - J(\frac{N_0}{2} - \frac{1}{2})$$

  where N_0 is the nominal occupancy specified by 'nominal_n'.

- For each distinct impurity problem indexed by the value in control ["impurity_problem_equivalence"],
  - 'impurity_matrix': [
    [1,0,0,0,0],
    [0,1,0,0,0], 1:t_{2g}
    [0,0,2,0,0], 2:e_g
    [0,0,0,1,0],
    [0,0,0,0,2]
    ],
    Equivalence of the matrix element of the hybridization function and impurity self-energy. Starting from "1", you can set any positive numbers. If these values are the same, hybridization function and impurity self-energy will be identical for those. If an element in the matrix is zero, then it will not be sampled by the impurity solver. Each column and row correspond to the Wannier orbitals in the following order: |xy>,|yz>,|z^2>,|xz>, |x^2-y^2> if control['spin_orbit']==False. If control['spin_orbit']==True, the most rapidly changing index is "m" and the next one is "i". They are sorted in ascending order. For the case of "f" shell, for example, they are ordered as:|3,-0.5, -2.5>,|3,-0.5, -1.5>,|3,-0.5, -0.5>, |3,-0.5, 0.5>,|3,-0.5, 1.5>,|3,-0.5, 2.5>, |3,0.5, -3.5>,|3,0.5, -2.5>,|3,0.5,-1.5>,|3,0.5, -0.5>,|3,0.5, 0.5>,|3,0.5, 1.5>,|3,0.5, 2.5>, |3,0.5, 3.5>,

  - 'thermalization_time': 1,
    Wall time for the thermalization in minutes.

  - 'measurement_time': 60,
    Wall time for the measurement in minutes.

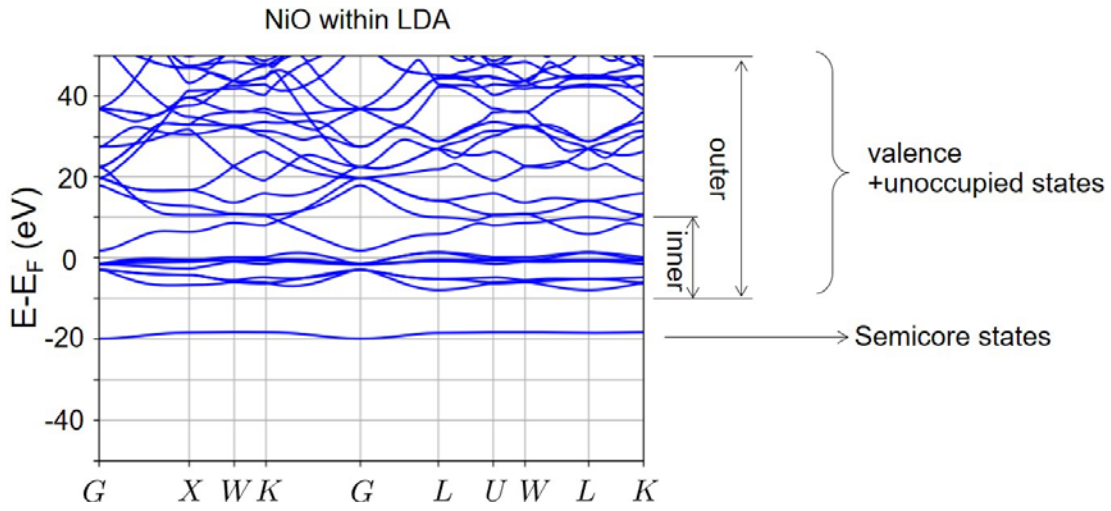
  - 'green_cutoff': 10,
    Cutoff-energy in eV to sample green's function and self-energy. The values beyond this energy will be provided by analytical equations.

  - 'susceptibility_cutoff': 300,
    Cutoff-energy to sample susceptibility. The default value is 300 eV.

  - 'Coulomb': 'full',
    'full' or 'ising' are available. We construct Coulomb matrix in the following way.
    $$U_{m_1,m_2,m_3,m_4} = \sum_{k=0}^{2l,even} \frac{4\pi}{2k+1} F_l^k \sum_{q=-k}^{k} \langle Y_l^{m_1} | Y_k^q Y_l^{m_4} \rangle \langle Y_l^{m_2} Y_k^q | Y_l^{m_3} \rangle$$
    If 'full', no additional approximation is considered. If 'ising', only U_{abba} or U_{abab} are non-zero.

Input file (comdmft.ini)-Important concepts for wan_hmat

COMSUITE uses localized orbitals such as Wannier functions to represent the low-energy Hilbert space. To construct the Wannier functions, the inner (frozen) energy window can be set to range from E_F+'froz_win_min' to E_F+'froz_win_max', and the outer (disentanglement) energy window can range from E_F+'dis_win_min' to E_F+'dis_win_max'; see the figure below (Here we take the primitive NiO case as an example).

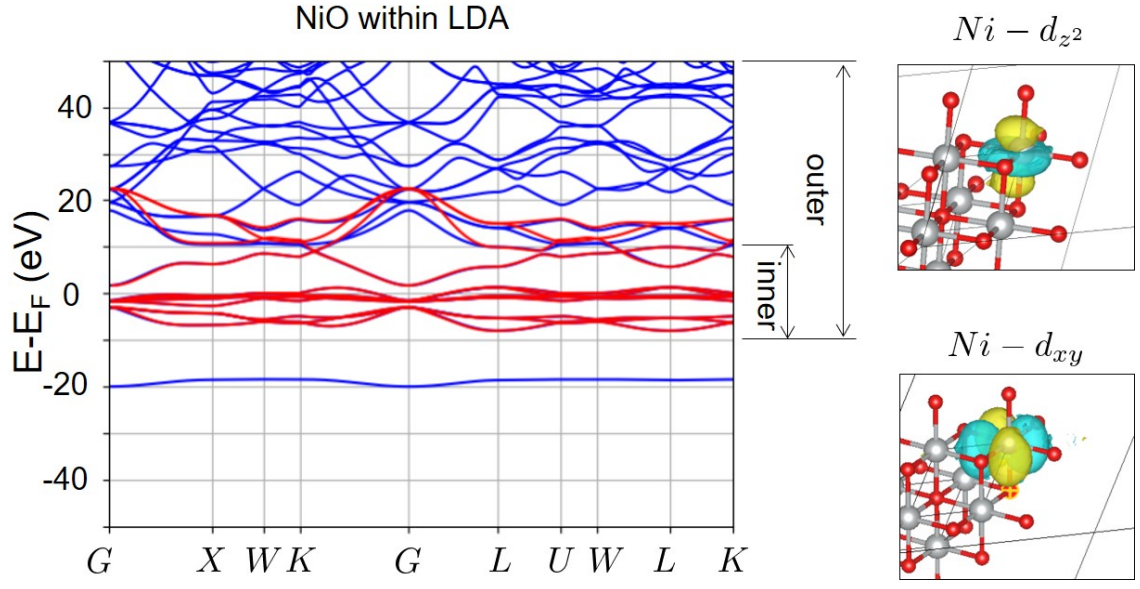


We choose initial trial orbitals $|\tau\mathbf{R}=0\rangle_t$ using MT orbitals with desired angular momentum character. The radial functions of $|\tau\mathbf{R}=0\rangle_t$ are chosen in such a way to maximize

$$\frac{1}{N_{\mathbf{k}}} \sum_{n\mathbf{k}}^{E_{min}^{inner}<E_{n\mathbf{k}}<E_{min}^{inner}} |\langle n\mathbf{k}|\tau\mathbf{k}\rangle_t|^2$$

, where $|\tau\mathbf{k}\rangle_t = \frac{1}{N_{\mathbf{k}}}\sum_{\mathbf{R}} |\tau\mathbf{R}\rangle_t e^{i\mathbf{k}\cdot\mathbf{R}}$. Among the MT orbitals above, we chosse ones which are larger than 0.15. For correlated orbitals, final wannier functions $|\tau\mathbf{R}=0\rangle_f$ usually satisfy a condition of $_f\langle\tau\mathbf{R}=0|\tau\mathbf{R}=0\rangle_t > 0.95$. This means that $|\tau\mathbf{R}=0\rangle_f$ are strongly localized and are regarded as atom-like wavefunctions.

The figure below shows the Wannier functions and interpolated band structure of NiO in comparison with the LDA band structure. The number of bands in the inner window is 10, while the number of bands in the outer window is 25. The number of trial orbitals is 12 (Ni-*s*, Ni-*p*, Ni-*d*, O-*p*).



Output files

COMSUITE places important output files generated from individual programs in the work directory (lda_dmft in this example). The list of files is

cmd.log
convergence.log : convergence log file
sig.dat : impurity self-energy
delta.dat : hybridization function

The format of each file and meaning of fields are discussed below. The corresponding results of the NiO LDA+DMFT calculation are presented with illustrative plots.

■ convergence.log

step	i_outer	i_latt	i_imp	causality	delta_rho	w_sp_min	w_sp_max	mu	std_sig	n_imp	histo_1	histo_2	ctqmc_sign
wannier	1					0.4053267	2.37118547						
delta	1		1	good				0.050843					
impurity_1	1		1	good					2.13330962	8.03442	110.540266	139.386327	0.98089
dft	2	1			0.0004217099								
wannier	2					0.40532646	2.37119739						
delta	2		1	good				0.211430					
impurity_1	2		1	good					1.08586757	8.05260	99.7563187	108.428532	0.98736
dft	3	1			0.03774896								
wannier	3					0.40534312	2.37121046						
delta	3		1	good				0.181122					
impurity_1	3		1	good					0.63852017	8.05826	97.3450919	102.309987	0.98863

- i_outer: The scf step number for a given charge density and impurity self-energy
- i_latt: The iteration number for solving Kohn-Sham equation with a given charge density obtained from ComLowH.
- i_imp: The iteration number for solving impurity problem through ComLowH + ComCTQMC
- causality: causality of hybridization function/self-energy
- delta_rho: The norm difference between the current charge density and the one from the previous scf step.
- w_sp_min: minimum spread of the Wannier functions
- w_sp_max: maximum spread of the Wannier functions
- mu: LDA+DMFT chemical potential w.r.t. LDA chemical potential
- std_sig:

$$\sqrt{\frac{\sum_i (\Sigma_i^j(i\omega_n) - \Sigma_i^{j-1}(i\omega_n))^2}{n_\omega n_{orb}}}$$

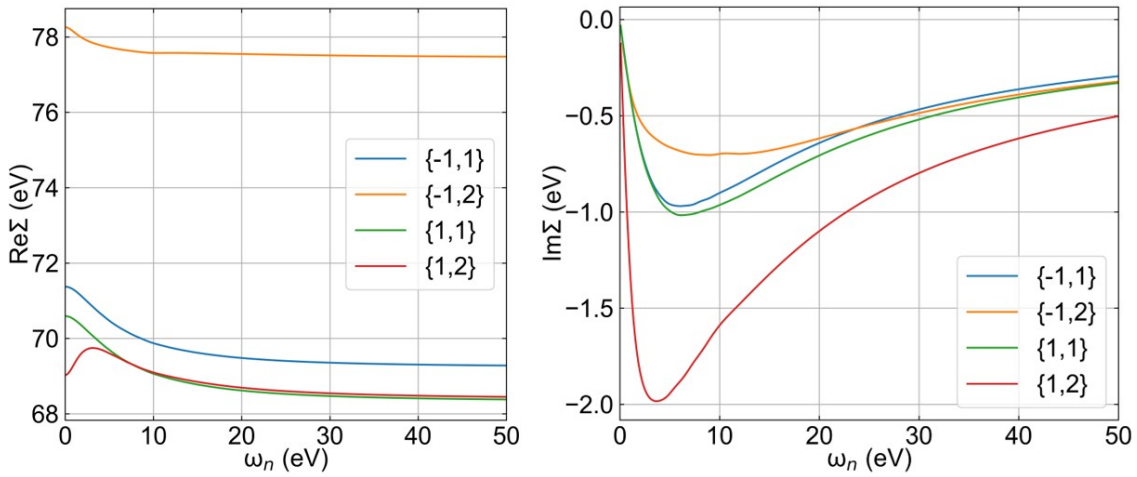
- n_imp: occupation of the impurity orbitals
- histo_1: the first moment of the perturbation order histogram
- histo_2: the second moment of the perturbation order histogram
- ctqmc_sign: the sign of Monte-Carlo simulation

■ sig.dat

'sig.dat' contains impurity self-energies. The first column lists matsubara frequencies and the next columns are real and imaginary parts of self-energies of "t_{2g}" (Sig_{1,1}) and "e_g" (Sig_{1,2}) of Ni d-orbital for corresponding impurity problem 1 or -1.

# omega(eV)	Re Sig_{-1,1}(eV)	Im Sig_{-1,1}(eV)	Re Sig_{-1,2}(eV)	Im Sig_{-1,2}(eV)	Re Sig_{1,1}(eV)	Im Sig_{1,1}(eV)	Re Sig_{1,2}(eV)	Im Sig_{1,2}(eV)
0.081216424692	71.371565796994	-0.032369954998	78.255520026207	-0.036100584092	70.592099407217	-0.032943203712	69.029328584119	-0.125471302314
0.243649274076	71.366784643850	-0.096798944663	78.246560336990	-0.106900756246	70.587062309521	-0.098467104603	69.049266958242	-0.372437608138
0.406082123461	71.356678099732	-0.160097651668	78.228747421470	-0.172942833988	70.576507716407	-0.162668437917	69.089081170390	-0.605493096491
0.568514972845	71.342711318533	-0.221985811503	78.205721278195	-0.233654980759	70.561966146271	-0.225232950545	69.140645429075	-0.822229506932
0.730947822229	71.324621050262	-0.281882144231	78.178595478531	-0.287700385643	70.543313343770	-0.285499119421	69.201507433481	-1.017474279473
0.893380671613	71.302805621055	-0.339382559363	78.149181441040	-0.334892390641	70.521050045926	-0.343058021741	69.267289207605	-1.189242761215
1.055813520997	71.277714272729	-0.394183550500	78.118958376906	-0.375591576936	70.495794156585	-0.397656889417	69.334208759074	-1.337331794516
1.218246370382	71.249728556364	-0.446051495615	78.089026896420	-0.410466809804	70.468097312005	-0.449191236001	69.399305117463	-1.462902922686

Plots of the self-energy:

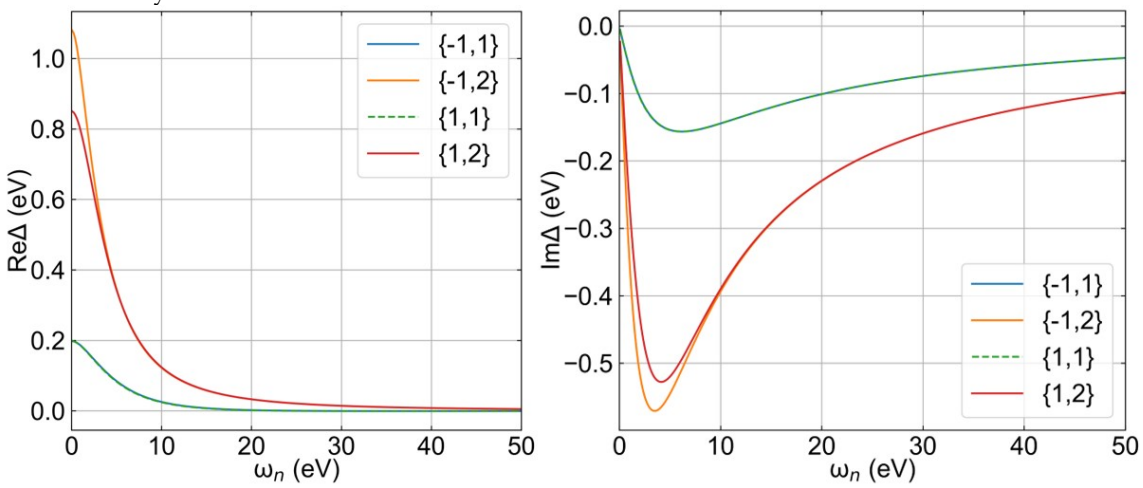


Impurity self-energy associated with the Ni-d orbitals on the imaginary frequency axis. The red and blue lines represent Ni-t2g and Ni-eg characters, respectively.

■ **delta.dat**

'delta.dat' is in the same format of 'sig.dat'

Plots of the hybridization function:



Hybridization function associated with the Ni-d orbitals on imaginary frequency axis. The red and blue lines represent the Ni-t2g and Ni-eg characters, respectively. Ni-t2g orbitals experience less hybridization with the rest of the electrons than the Ni-eg orbitals.

Analytical Continuation of Self-energy

To obtain the density of states (DOS), we need to perform an analytic continuation of 'sig.dat' to produce the impurity self-energy on the real frequency axis. To do this, we will adopt the maximum entropy (maxent) method. Any publicly available maxent code can be employed. For the purposes of this tutorial, however, we will use K. Haule's maxent code (freely available at http://www.physics.rutgers.edu/~haule).

To access maxent code, you should export the path to the executable in your startup shell script.

```
export WIEN_DMFT_ROOT=[path to Haule's code bin directory where maxent_run.py is located]
```

To run the maxent code, move to your working directory, create the maxent directory in the "lda_dmft" directory and then move to it:

```
$ mkdir maxent
```

$ cd maxent

By executing 'maxent_wrapper.py', we can obtain the self-energy on real axis by automatically calling maxent_run.py:

$ $COMSUITE_BIN/maxent_wrapper.py ../sig.dat

You can see its options with -h option:

$COMSUITE_BIN/maxent_wrapper.py –h

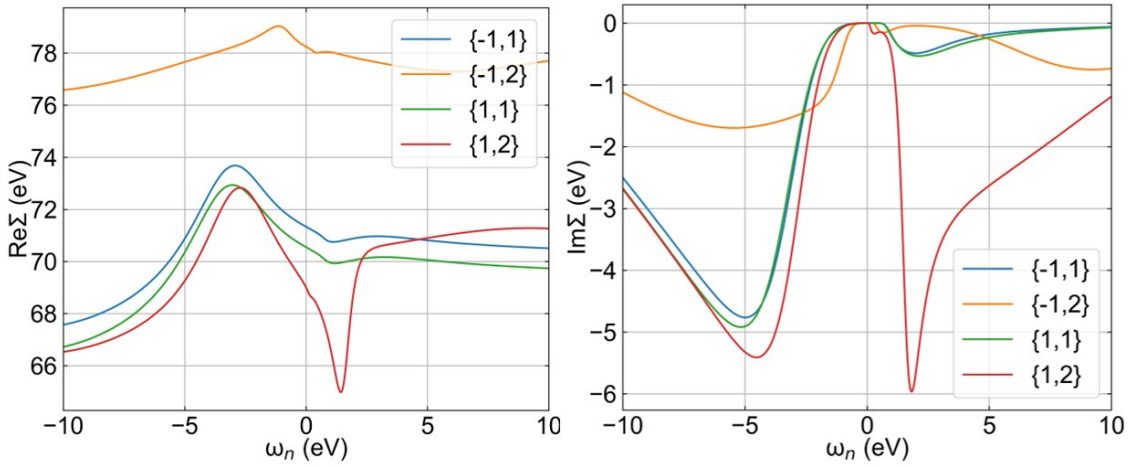
```
usage: maxent_wrapper.py [-h] sig [error]

call maxent_run.py from EDMFTF package and return self-energy on real axis. If
maxent_params.dat is not present in the directory, it generates one. The name
of the output file will be sig_realaxis.dat

positional arguments:
  sig         self-energy file on imaginary axis
  error       Errors for the maxent.Optional. Default value=0.05

optional arguments:
  -h, --help  show this help message and exit
```

Here note that in order to perform maxent, `maxent_params.dat' file is needed. This file should be written in python dictionary format as follows:

```
params={'statistics': 'fermi', # fermi/bose
        'Ntau'      : 3000,     # Number of time points
        'L'         : 70.0,     # cutoff frequency on real axis
        'Nw'        : 501,     # number of frequency points on real axis
        'gwidth'    : 140.0,  # width of gaussian
        'idg'       : 1,       # error scheme: idg=1 -> sigma=deltag ; idg=0 -> sigma=deltag*G(tau)
        'deltag'    : 0.05,   # error
        'Asteps'    : 4000,    # anealing steps
        'alpha0'    : 1000,    # starting alpha
        'x0'        : 0.01,    # low energy cutoff
        'min_ratio' : 0.001,    # condition to finish, what should be the ratio
        'iflat'     : 1,       # iflat=0 : constant model, iflat=1 : gaussian of width gwidth, iflat=2 : input using file model.dat
        'Nitt'      : 1000,    # maximum number of outside iterations
        'Nr'        : 0,       # number of smoothing runs
        'bwdth'     : 0.03,    # smoothing width
        'Nf'        : 5,      # to perform inverse Fourier, high frequency limit is computed from the last Nf points
         }
```

If the 'maxent_params.dat' file is not present in the directory, maxent_wrapper.py will automatically generate the file with default options and execute subsequent jobs. If maxent is finished successfully, you will obtain an analytically continued self-energy file, 'sig_realaxis.dat'. 'sig_realaxis.dat' is in the same format with 'sig.dat' except that the first column is real frequency values.

Plots of the impurity self-energy on the real frequency axis:



The impurity self-energy on the real frequency axis as obtained by using the maximum entropy method. Electronic self-energy for Ni-eg has a pole near Fermi level indicative of the paramagnetic Mott gap in NiO.

LDA+DMFT density of states

To obtain the DOS, we must post-process the data by executing ComLowH again. First, create a directory for the DOS calculation in the "lda_dmft" directory and move to it:

$ mkdir realgrid
$ cd realgrid

Copy necessary files to calculate DOS and partial DOS by using prepare_realaxis.py which is in $COMSUITE_BIN folder.

```
$ $COMSUITE_BIN/prepare_realaxis.py 0.1 ../lowh/ ../wannier/ ../maxent/sig_realaxis.dat 30 30 30 -m 2
```

You can see its options with -h option:

```
$COMSUITE_BIN/prepare_realaxis.py –h
```

```
usage: prepare_realaxis.py [-h] [-m MODE]
                           broadening lowh_directory wan_directory self_energy
                           [kmesh_b1_for_dos] [kmesh_b2_for_dos]
                           [kmesh_b3_for_dos]

prepare inputs of comlowh calculation on real axis

positional arguments:
  broadening            broadening
  lowh_directory        lowh directory
  wan_directory         wannier directory
  self_energy           real-axis self-energy
  kmesh_b1_for_dos      finer kmesh along b1 axis for the DOS. Optional
  kmesh_b2_for_dos      finer kmesh along b2 axis for the DOS. Optional
  kmesh_b3_for_dos      finer kmesh along b3 axis for the DOS. Optional

optional arguments:
  -h, --help            show this help message and exit
  -m MODE, --mode MODE  If 3, code calculates spectral function along the high
                        symmetry line defined in 'kpath.dat'. If it is 2, it
                        calculates projected density of states. Default: 3
```

Run ComLowH with job submission script. An example of job script using SLURM is

```
#!/bin/bash -l
#SBATCH -J temp
#SBATCH -q debug
#SBATCH -N 1
#SBATCH -e temp.%j.err
#SBATCH -o temp.%j.out
#SBATCH -L SCRATCH
#SBATCH -C haswell
#SBATCH -t 00:30:00
srun -n 30 $COMSUITE_BIN/ComLowH
```

Having done so, you will obtain files tdos.dat and pdos.dat.

The format of tdos.dat file is:

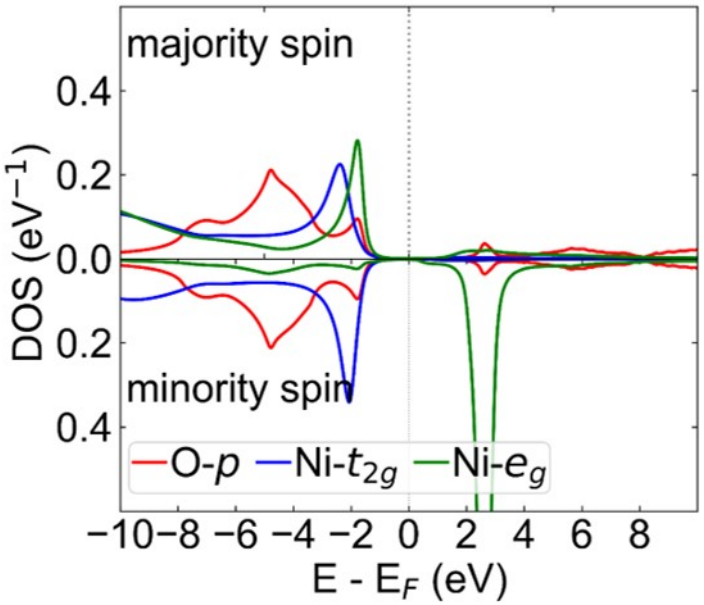
```
#       omega (eV)           TDOS (1/eV)
    -70.000000000000      0.000078774138
    -65.733023713200      0.000089675403
    -61.955278682000      0.000101306654
    -58.587128546700      0.000113731263
    -55.565331497900      0.000127038129
    -52.839027896300      0.000141121324
    -50.366851286200      0.000156212142
    -48.114812211200      0.000171985663
    -46.054723948000      0.000188862783
    -44.163014879300      0.000206717996
```

The format of pdos.dat file is:

```
#     omega (eV)              (1,1,-1)            (1,1,0)             (1,1,1)             (2,0,0)             (2,1,-1)            (2,1,0)             (2,1,1)             (2,2,-2)
(2,2,-1)              (2,2,0)             (2,2,1)             (2,2,2)
   -70.000000000000       0.000006795631      0.000006795631      0.000006795631      0.000005877936      0.000004860717      0.000004860719      0.000004860717      0.000007968630
0.000007968631      0.000007010631      0.000007968631      0.000007010632
   -65.733023713200       0.000007740465      0.000007740464      0.000007740465      0.000006635899      0.000005427666      0.000005427667      0.000005427666      0.000009171980
0.000009171981      0.000008009584      0.000009171981      0.000008009585
   -61.955278682000       0.000008753001      0.000008753001      0.000008753001      0.000007438074      0.000006018208      0.000006018210      0.000006018208      0.000010470404
0.000010470405      0.000009071868      0.000010470405      0.000009071869
   -58.587128546700       0.000009834666      0.000009834665      0.000009834666      0.000008284518      0.000006631635      0.000006631637      0.000006631635      0.000011872835
0.000011872836      0.000010214667      0.000011872836      0.000010214668
   -55.565331497900       0.000010986949      0.000010986948      0.000010986949      0.000009175366      0.000007267330      0.000007267333      0.000007267330      0.000013411824
0.000013411825      0.000011432225      0.000013411825      0.000011432226
```

- (atom index, l, m) if spin_orbit==False and (atom index, I, i,m) if spin_orbit==True.

Plot of pdos.dat:



The projected density of states of O-p, Ni-s, Ni-t_{2g}, and Ni-e_g orbitals within charge self-consistent LDA+DMFT, marked by red, blue, and green colors, respectively.

LDA+DMFT spectral function

To obtain spectral function, we must post-process the data by executing ComLowH again. First create a directory for the spectral function in "lda_dmft" directory and move to it:

```
$ mkdir realaxis
$ cd realaxis
```

Having done so, copy the necessary files to calculate the spectral function by using prepare_realaxis.py which is in $COMSUITE_BIT directory. Note that the option should be '-m 3'.

```
$ $COMSUITE_BIN/prepare_realaxis.py 0.1 ../lowh/ ../wannier/ ../maxent/sig_realaxis.dat -m 3
```

Then you should create a k-path file (kpath.dat) In the first line, the number of k-points is specified. In the following lings, k-points are written in terms of the reciprocal lattice vector.

```
186      #  (kx,  ky,  kz)
  0.00000000   0.00000000   0.00000000
  0.01666667   0.00000000   0.01666667
  0.03333333   0.00000000   0.03333333
  0.05000000   0.00000000   0.05000000
  0.06666667   0.00000000   0.06666667
  0.08333333   0.00000000   0.08333333
  0.10000000   0.00000000   0.10000000
  0.11666667   0.00000000   0.11666667
  0.13333333   0.00000000   0.13333333
  0.15000000   0.00000000   0.15000000
```

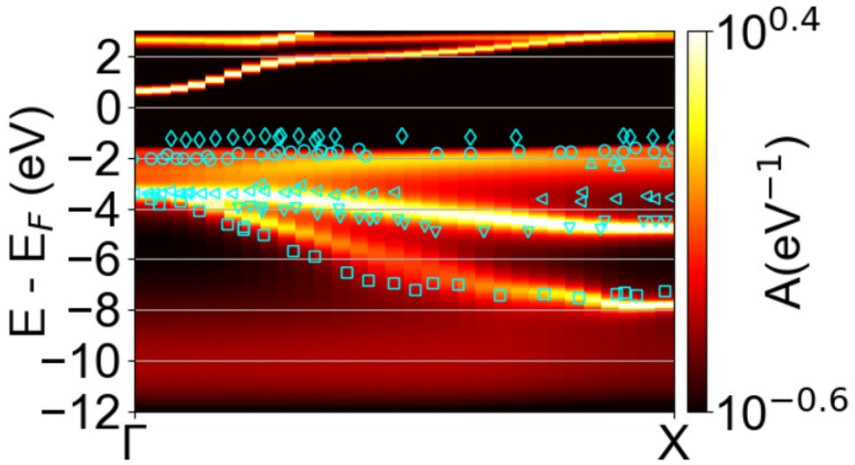
Run ComLowH with job submission script. An example of job script using SLURM is

```
#!/bin/bash -l
#SBATCH -J temp
#SBATCH -q debug
#SBATCH -N 1
#SBATCH -e temp.%j.err
#SBATCH -o temp.%j.out
#SBATCH -L SCRATCH
#SBATCH -C haswell
#SBATCH -t 00:30:00
srun -n 30 $COMSUITE_BIN/ComLowH
```

Having done so, you will have obtained a file named spectral.dat. The format of this file is

# kpoint	E (eV)	A (1/eV)
1	-70.000000000000	0.000076733675
2	-70.000000000000	0.000076740217
3	-70.000000000000	0.000076759794
4	-70.000000000000	0.000076792257
5	-70.000000000000	0.000076837385
6	-70.000000000000	0.000076894890
7	-70.000000000000	0.000076964412
8	-70.000000000000	0.000077045477
9	-70.000000000000	0.000077137421
10	-70.000000000000	0.000077239291

Plot of spectral.dat file along a high symmetry line in the first Brillouin zone.:



Spectral function along $\Gamma - X$ in the first Brillouin zone within charge self-consistent LDA+DMFT. The spectral function is unfolded into the first Brillouin zone of the primitive cell of the paramagnetic phase using Green's function and Wannier function.[1] The experimental data with cyan six symbols has been obtained from Fig. 6 in reference of angle-resolved photoemission spectroscopy data.[2] The experiment data are aligned with calculated valence band edge.

[1] B.Kang and S. Choi, arXiv:1908.05643 [cond-mat.str-el]

[2] Z.-X. Shen, R. S. List, D. S. Dessau, B. O. Wells, O. Jepsen, A. J. Arko, R. Barttlet, C. K. Shih, F. Parmigiani, J. C. Huang, P. A. P. Lindberg, Phys. Rev. B **44**, 3604 (1991)