wannier90-utils

Release v0.1.0

Jamal I. Mustafa

Mar 30, 2021

CONTENTS

This package provides a library of functions for reading/writing and manipulating the data associated with the wannier90 code [1].

CHAPTER

ONE

FEATURES

- Routines for reading/writing/manipulating a variety of files
  - manipulating WIN files (see *here*)
  - parsing WOUT files (see *here*)
  - the `nnkp` file (see *here*)
  - the eigenvalues, overlap matrices, and projection matrices (see *here*)
  - output of `postw90` program, such as bandstructures (see *here*)
- Utilities for computing the centers and spreads of Wannier functions (see *here*)

CHAPTER

TWO

INSTALLATION

To install the latest version of the wannier90-utils package, simply clone the repository and install using `pip`.

```
git clone https://github.com/jimustafa/wannier90-utils.git
cd wannier90-utils && pip install .
```

CHAPTER

THREE

DOCUMENTATION

3.1 Manipulating WIN files

Wannier90 I/O routines pertaining to WIN files

w90utils.io.win.**print_atoms**(*atoms*, *units='crystal'*, *file=<_io.TextIOWrapper name='<stdout>' mode='w' encoding='UTF-8'>*)

w90utils.io.win.**print_kgrid**(*kgrid*, *file=<_io.TextIOWrapper name='<stdout>' mode='w' encoding='UTF-8'>*)

w90utils.io.win.**print_kpoints**(*kpoints*, *mp_grid=None*, *file=<_io.TextIOWrapper name='<stdout>' mode='w' encoding='UTF-8'>*)

w90utils.io.win.**print_unit_cell**(*dlv*, *units='bohr'*, *file=<_io.TextIOWrapper name='<stdout>' mode='w' encoding='UTF-8'>*)

w90utils.io.win.**read_atoms**(*fname*, *units='crystal'*)

w90utils.io.win.**read_dlv**(*fname*, *units='bohr'*)

Read direct lattice vectors from WIN file.

Parameters

- **fname** (*str*) – Wannier90 WIN file
- **units** (*str, {'bohr', 'angstrom'}*) – units of returned lattice vectors

Returns **dlv** (*ndarray, shape (3, 3)*) – direct lattice vectors

w90utils.io.win.**read_kgrid**(*fname*)

w90utils.io.win.**read_kpoints**(*fname*)

w90utils.io.win.**read_proj_line**(*line*, *dlv*, *basis*, *spinors*)

w90utils.io.win.**read_projections**(*fname*)

w90utils.io.win.**remove_comments**(*s*)

3.2 Parsing WOUT files

Wannier90 I/O routines pertaining to WOUT files

w90utils.io.wout.**read_centers_xyz**(*fname*)

w90utils.io.wout.**read_conv**(*fname*)

w90utils.io.wout.**read_sprd**(*fname*)

3.3 Parsing NNKP files

Wannier90 I/O routines pertaining to NNKP files

w90utils.io.nnkp.**read_bvectors**(*fname*, *units='angstrom'*)

w90utils.io.nnkp.**read_dlv**(*fname*, *units='bohr'*)

w90utils.io.nnkp.**read_excluded_bands**(*fname*)

w90utils.io.nnkp.**read_kpoints**(*fname*, *units='crystal'*)

w90utils.io.nnkp.**read_nnkpts**(*fname*)

w90utils.io.nnkp.**read_projections**(*fname*)

w90utils.io.nnkp.**read_rlv**(*fname*, *units='bohr'*)

3.4 Fundamental I/O routines

w90utils.io.read_eig(fname)	Read EIG file.
w90utils.io.write_eig(fname, eig)	Write $E_{n\mathbf{k}}$ to EIG file.
w90utils.io.read_hamiltonian(fname)	Read EIG file and return k-dependent Hamiltonian matrix.
w90utils.io.read_amn(fname)	Read AMN file.
w90utils.io.write_amn(fname, amn[, header])	Write $A_{mn}^{(\mathbf{k})}$ to AMN file.
w90utils.io.read_mmn(fname)	Read MMN file
w90utils.io.write_mmn(fname, mmn, kpb_kidx, ...)	Write $M_{mn}^{(\mathbf{k},\mathbf{b})}$ to MMN file

w90utils.io.**read_eig**(*fname*)

Read EIG file.

Parameters **fname** (*str*) – path to EIG file

Returns *ndarray, shape (nkpts, nbnds, nproj)*

w90utils.io.**write_eig**(*fname*, *eig*)

Write $E_{n\mathbf{k}}$ to EIG file.

Parameters

- **fname** (*str*) – path to EIG file
- **eig** (*ndarray, shape (nkpts, nbnds)*) –

w90utils.io.**read_hamiltonian**(*fname*)
Read EIG file and return k-dependent Hamiltonian matrix.

Parameters **fname** (*str*) – path to EIG file

Returns *ndarray, shape (nkpts, nbnds, nbnds)*

w90utils.io.**read_amn**(*fname*)
Read AMN file.

Parameters **fname** (*str*) –

Returns *ndarray, shape (nkpts, nbnds, nproj)*

w90utils.io.**write_amn**(*fname*, *amn*, *header='HEADER'*)
Write $A_{mn}^{(\mathbf{k})}$ to AMN file.

Parameters

- **fname** (*str*) –
- **amn** (*ndarray, shape (nkpts, nbnds, nproj)*) –
- **header** (*str*) –

w90utils.io.**read_mmn**(*fname*)
Read MMN file

Parameters **fname** (*str*) –

Returns *ndarray, shape (nkpts, nntot, nbnds, nbnds)*

w90utils.io.**write_mmn**(*fname*, *mmn*, *kpb_kidx*, *kpb_g*)
Write $M_{mn}^{(\mathbf{k},\mathbf{b})}$ to MMN file

Parameters

- **fname** (*str*) –
- **mmn** (*ndarray, shape (nkpts, nntot, nbnds, nbnds)*) –

3.5 Wannier function centers and spreads

Functions for computing Wannier centers and components of the spread

w90utils.sprd.**omega**(*Mmn*, *bvectors*, *bweights*)
Compute the spread functional

Parameters

- **Mmn** (*ndarray, shape (nkpts, nntot, nbnds, nbnds)*) – the overlap matrix
- **bvectors** (*ndarray, shape (nkpts, nntot, 3)*) –
- **bweights** (*ndarray, shape (nntot,)*) –

w90utils.sprd.**omega_d**(*m*, *bvectors*, *bweights*, *idx=None*)
Compute the diagonal contribution to the spread functional

Parameters

- **m** (*ndarray, shape (nkpts, nntot, nbnds, nbnds)*) – the overlap matrix
- **bvectors** (*ndarray, shape (nkpts, nntot, 3)*) –

- **bweights** (*ndarray, shape (nntot,)*) –

w90utils.sprd.**omega_dod**(*Mmn*, *bvectors*, *bweights*)
Compute the sum of the diagonal and off-diagonal contribution to the spread functional

Parameters

- **Mmn** (*ndarray, shape (nkpts, nntot, nbnds, nbnds)*) – the overlap matrix
- **bvectors** (*ndarray, shape (nkpts, nntot, 3)*) –
- **bweights** (*ndarray, shape (nntot,)*) –

w90utils.sprd.**omega_i**(*Mmn*, *bweights*)
Compute the invariant contribution to the spread functional

Parameters

- **Mmn** (*ndarray, shape (nkpts, nntot, nbnds, nbnds)*) – the overlap matrix
- **bweights** (*ndarray, shape (nntot,)*) –

w90utils.sprd.**omega_iod**(*m*, *bweights*, *idx=None*)
Compute the sum of the invariant and off-diagonal contribution to the spread functional

Parameters

- **m** (*ndarray, shape (nkpts, nntot, nbnds, nbnds)*) – the overlap matrix
- **bweights** (*ndarray, shape (nntot,)*) –

w90utils.sprd.**omega_od**(*Mmn*, *bweights*)
Compute the off-diagonal contribution to the spread functional

Parameters

- **Mmn** (*ndarray, shape (nkpts, nntot, nbnds, nbnds)*) – the overlap matrix
- **bweights** (*ndarray, shape (nntot,)*) –

w90utils.sprd.**wannier_centers**(*m*, *bvectors*, *bweights*)

3.6 Processing `postw90` output

w90utils.io.postw90.**print_kpoints**(*kpoints*, *header=''*, *units='crystal'*, *file=<_io.TextIOWrapper name='<stdout>' mode='w' encoding='UTF-8'>*)

w90utils.io.postw90.**read_band_velocities**(*fname*)

w90utils.io.postw90.**read_bands**(*fname*)

w90utils.io.postw90.**read_bands_kpoints**(*fname*)
Read k-points from the `geninterp` dat file

Parameters **fname** (*str*) –

Returns *ndarray, shape (nkpts, 3)* – array of kpoints using for `geninterp`, in units of $x212B^{-1}$

w90utils.io.postw90.**read_dos**(*fname*)

w90utils.io.postw90.**read_elcond**(*fname*)

w90utils.io.postw90.**read_kpoints**(*fname*)

w90utils.io.postw90.**read_vnk**(*fname*)

w90utils.io.postw90.**write_kpoints**(*fname*, *kpoints*)

3.7 Examples

3.7.1 Read the Hamiltonian in the Wannier representation

```
from w90utils import io as w90io

HR, Rvectors, Rweights = w90io.read_hr('wannier_hr.dat')
```

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

W

INDEX