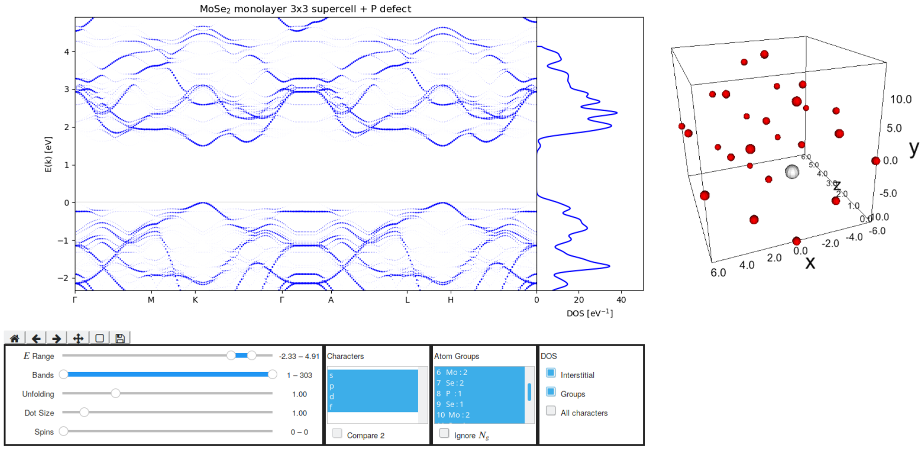
SiScLab 2018 Student Project **Analysis Tool for Materials Design**. Written in Python3.

Authors: Johannes Wasmer, Christian Partmann, and Praneeth Katta.

Overview

This subfolder `studentproject18ws` is currently a largely independent side-project accompanying the main module `masci-tools`. It was created in a student project, and consists of three submodules: * preprocessor: a HDF reader interface, and one implementation for Fleur band structure simulation output * visualization: a plotting interface, and one implementation for Fleur bandstructure+DOS plots * frontends: a Desktop GUI and a Web Dashboard (Tk and Jupyter) for interactive Fleur bandDOS plots.

A more thorough description and example use cases can be found in the project report and presentation.



E Range -2.33 – 4.91
Bands 1 – 303
Unfolding 1.00
Dot Size 1.00
Spins 0 – 0

Characters: s p d f
□ Compare 2

Atom Groups: 6 Mo : 2, 7 Se : 2, 8 P : 1, 9 Se : 1, 10 Mo : 2
□ Ignore N_g

DOS
☑ Interstitial
☑ Groups
□ All characters

Figure 1:

For Frontend Users

The Desktop GUI executable can be received from the developers on request. Otherwise, it can be built using PyInstaller from this repo.

The Web Frontend is a Jupyter Dashboard. It is in experimental phase (no fileupload yet). You can try it out here on Binder. You can run it locally (see developer section). If you have an AiiDaLab account: the dashboard is planned to be published as an app there.

For Developers

Installation

Though `masci-tools` is availabe via PyPI, there is currently no plan to integrate `studentproject18ws`. If you want to use it in your code, clone the repo, use it in an IDE, or append the path to your `sys.path`:

```
import sys
if path_repo not in sys.path:
    sys.path.append(path_repo)

# now import works
from studentproject18w.hdf.reader import Reader
# ...
```

Create project virtual environment

With conda (recommended): - Install Anaconda (3 recommended) - Install the environment `masci-stupro` with the necessary and recommended dependencies:

```
conda create -f environment.yml
source activate masci-stupro
```

With virtualenv (untested):

```
virtualenv masci-stupro
source masci-stupro/bin/activate
pip install -r requirements_pip.txt # install requirements
```

Try out Web Frontend locally

The demo notebook with the Dashboard is `studentproject18w/frontend/jupyter/demo/demo.ipynb`.

If using Jupyter Notebook

If using Windows, omit keyword `source`.

```
source activate masci-stupro
cd mypath/masci-tools/studentproject18ws/
jupyter-notebook .
# if Home is not set to this dir, try this instead:
# /home/you/anaconda3/envs/myenv/bin/python /home/you/anaconda3/envs/myenv/bin/jupyter-note
```

If using Jupyter Lab

Additional installation step needed:

```
source activate masci-stupro
jupyter labextension install @jupyter-widgets/jupyterlab-manager jupyter-matplotlib ipyvolum
cd mypath/masci-tools/studentproject18ws/
jupyter-lab
```

To-do list for publishing the Web Frontend

- (recommended: create `frontend/jupyter/Dashboard.py` widget and put code of demo_back.ipynb notebook inside it. Use aiidalab-widgets-base > StructureUploadWidget as a template. Create `frontend/jupyter/Dashboard.ipynb` notebook. Use StructureUploadWidget Demo Notebook as a template.)
- Add fileupload to widget (again, like in StructureUploadWidget. See binder_fileupload_test.ipynb notebook for a demo that works with binder.)
- Now the Web Frontend should work on Binder.
- For publishing the app on AiiDA Lab, the app has to be registered in the aiidalab-registry.
  - The project code is in Python3, but aiidalab requires Python2. So the code has to first be backported by hand using the `future` package. If this takes too long, maybe try the tool 3to2.
  - Use the simplest app in the registry, aiidalab-units as a template. Adapt code.
  - Try it out first in the Quantum Mobile Virtual Machine, which has aiidalab installed and configured. Else try it in a virtual environment with aiidalab installed from PyPI.
  - Register the app.