

# Topic 1: Using Materials Project

## Overview

The field of computational thermodynamics—where lattice models of alloys are built in order to compute free energies of different complex structures—rather naturally led to the development of high-throughput density-functional theory computation. Multiple databases are now publicly available, and we will work with Materials Project for this project. It has both a user interface through its website as well as an extensive API that can be accessed using different tools, including pymatgen. For more information on the computational choices made in Materials Project, see [here](#).

Background: Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, and Kristin A. Persson; “Commentary: The Materials Project: A materials genome approach to accelerating materials innovation.” *APL Materials* **1**, 011002 (2013); doi

More information: primary citations and publications

We’ll look at two different uses of Materials Project to get a handle on what we can do with it for querying of materials databases, and doing some of our own analysis. For this, we’ll start by constructing Ashby plots for a series of metal compounds; after that, we’ll look at a recent publication making use of electronic structure data from Materials Project.

## Getting started

You’ll need to do a few things to get going with Materials Project and pymatgen.

1. Get a Materials Project API key. First, create an account with Materials Project. After you have done that, log in, go to the dashboard and “Generate an API key”. You will need to copy this key and either paste it into your scripts, or create / edit the `$HOME/.pmgrc.yaml` file to include the line:

```
PMG_MAPI_KEY: <USER_API_KEY>
```

where you put your actual API key in `<USER_API_KEY>`

2. Install the necessary python modules to use pymatgen. This can be done with `pip install pymatgen`; if you haven’t already installed `numpy` `scipy` `matplotlib` as well, you will want to make sure those are installed as well. If you decide to use a virtual environment for this installation, make sure to register the virtual environment as a new kernel using `ipython3 kernel install --user --name <virtual_env_name>`

As a check, we are running a Jupyter Hub server for the class at: `mse-598dm-jupyter.mse.illinois.edu` You can use this server to run notebooks as well, and use it as a check if you’re not sure that your local installation is working correctly. To get started, after you log in, you’ll need to do a little bit of setup with either our provided virtual environment or make your own. This needs to be done one time: create a new Terminal, and at the terminal execute

```
/srv/shared-kernels/scripts/DM5XX-setup.sh
```

which will put the necessary files in your `.local/share/jupyter/kernels` to access the virtual environment in Jupyter Hub. After doing this, you may need to go to the Control Panel and then return to your server to see the new kernel loaded as `MSE598DM Python 3`. You can then make new notebooks using this kernel; also, if you upload an existing notebook, when you read it in you may need to switch it to use this kernel: when running the notebook, select the menu `Kernel > Change kernel`. You can see the kernel that is running listed in the upper right.

## Ashby plot

We looked earlier at mechanical design, and found that for different applications, one might need to optimize quantities such as  $E/\rho$  (a ratio of Young's modulus to density) or other powers, such as  $E^{1/2}/\rho$ . We want to pull down VASP data from Materials Project for a variety of different metallic and intermetallic systems to make plots of  $E$  and  $\rho$ , as well as search for optimal values.

To get started, you'll need to prepare the `MPRester` object for querying Materials Project:

```
from pymatgen import MPRester
MY_API_KEY = None # replace with string of your API key
m = MPRester(MY_API_KEY)
```

(alternatively, you can use the construction

```
with MPRester(MY_API_KEY) as m:
    ...
```

when running scripts). Once we have our `MPRester` object, we can investigate the different ways of querying data from Materials Project by examining the docstrings of different functions (e.g., `get_data` among others), and examining the output.

We want to select a set of metals and intermetallics to consider: Al, Mg, Ti, Fe, Ni as base elements, along with binary and ternary combinations in that set. When examining the elasticity data, you will want to keep in mind the isotropic elasticity formula:  $E = 2G(1 + \nu)$ .

1. Create Ashby plots with single, binary, and ternary elements including both thermodynamically *stable* and *unstable* materials (define what criteria you use to determine this).
2. Identify maxima in  $E/\rho$  and  $E^{1/2}/\rho$  in your plots.
3. Use the data to identify trends: are there particular alloy classes that always identify as optimal?
4. Consider, with your data, what could happen as you consider *composites* using a simple volume-fraction rule to compute Young's modulus and density of your proposed material. How would you go about constructing optimization in this case?

## Band alignment

Materials selection for band alignment in semiconductor interfaces.

Ethan P. Shapera and André Schleife, "Database-Driven Materials Selection for Semiconductor Heterojunction Design." *Advanced Theory and Simulations* **1**, 1800075 (2018) doi

Dataset: MDF data

Files:

- Abundances.txt
- ChemComps.txt
- CNData.txt
- Descriptors.txt
- Electronegativity.txt
- ExperimentalGaps.csv
- OpticalProperties.txt
- UserData.txt

Scripts:

- CollectMaterialsProjectData.ipynb
  - DesignHeterostructures.ipynb
  - PlotAlignment.ipynb
- 

Discussion: Feb. 11-20, 2020