Volume Predictor

# Dependencies

* MongoDB
  + A local database will hold the set of materials used for training the predictor
* Pymatgen, matminer
  + See handbook
* Python libs
  + conda create –name <env\_name> python=2 numpy matplotlib pandas pymongo scipy scikit-learn plotly
* Jupyter notebooks
  + ipyparallel – parallel processing
  + jupyter\_contrib\_nbextensions – useful extensions (mostly code folding)

# Importing Materials Project data set

* setup
  + open (at least) two terminals, activate the python dev environment
    - source activate <env\_name>
  + open a jupyter notebook
    - jupyter notebook
  + start mongodb instance
    - (sudo) mongod
* follow import\_structures.ipynb
  + add Materials Project API key to api\_key variable

# Running the Volume Predictor

* setup
  + open (at least) three terminals, activate the python dev environment
    - source activate <env\_name>
  + open a jupyter notebook
    - jupyter notebook
  + start mongodb instance
    - (sudo) mongod
  + start the parallel processing engines
    - ipcluster start
* running
  + run all the cells under imports and functions
  + run the MAIN section
    - can either initialize everything or import from own data set
      * see section under functions to import previous data
  + remember to set the following before each run – this allows you to take breaks and analyze the data in between runs
    - init\_iter
    - num\_iter
    - percent\_bonds\_to\_change\_per\_iter
    - weight\_decay\_per\_iter
    - collection = db.materials\_project (or db.mp\_test)
* analysis
  + see analysis functions
    - display\_bond\_data(bond\_name)
    - display\_material\_data(mat\_name)