aBuild Sheet

June 28, 2019

aBuild commands start with “`python builder.py **YML**`” and then some tag(s). This prefix is only used for the tags denoted in this table by the “-”.

Table 1: Algorithm steps and their descriptions.

Step	**Description**
-enum	enumerates the crystalline structures up to sizes specified in your yaml –run interactively
-setup_relax	Builds to-relax.cfg and relax.ini; runs calc-grade –run interactively
qsub* jobscript_relax.sh	mlp relax: needs to-relax.cfg, pot.mtp, and relax.ini; generates: relaxed.cfg, unrelaxed.cfg, and candidates.cfg –job submission, parallel, 10-30 cores, 6-30 hrs
-setup_select_add	Concatenates all of the candidate.cfg_#, selection.log_#, relaxed.cfg_# and unrelaxed.cfg_# into one file each. relaxed.cfg file should get bigger and bigger with each iteration. Also builds a submission script. –run interactively
qsub* jobscript_select.sh	mlp select-add: generates: new_training.cfg; needs: train.cfg, candidate.cfg –job submission, single core, 1-4 hrs
-add	builds A folders in training set and creates jobscript –run interactively
qsub* jobscript_vasp.sh	runs vasp calculations for the selected configurations –array job, 6-30 hrs
-setup_train	Pulls data from VASP folders, builds train.cfg and pot.mtp –run interactively
qsub* jobscript_train.sh	mlp train: needs train.cfg, pot.mtp; generates: Potential.mtp –job submission, parallel, 10-20 cores, 6-12 hrs
go back to step -setup_relax	Repeat until model is fully trained, i.e. all structures relax.

*For Marylou use sbatch instead of qsub. **YML** is the yaml file without the .yml extension.

0.1 Other helpful MTP/aBuild commands

These might come in handy at some point

Table 2: aBuild commands and description

Command	**Description**
-status	prints a status report for Vasp calculations
-report	creates data report file from completed Vasp calculations
-report -file path/to/file	creates data report file from the configurations contained in the file specified
-chull -file path/to/file	creates a convex hull from the data report file specified
mlp mindist file.cfg	prints the global minimum distance of the atoms to eachother in a .cfg file. Also adds the mindist attribute to structures in the .cfg file
mlp calc-grade pot.mtp train.cfg train.cfg temp.cfg	creates state.mvs, a file needed for relaxation

VASP files and what they do

Pretty self explanatory name. If you want to learn a bit more about VASP, this is a good section to read.

In order to run any Vasp calculation, you need a POSCAR, POTCAR, INCAR, PRECALC, and KPOINTS. The rest of these files are output files. There are a couple other output files I haven't mentioned here.

POSCAR

This is the "POSition" car. Not really sure who came up with the naming convetion, but whatever `\_(-.-)_/`. In this file there is title. There is also the lattice parameter, which is what you multiply all the lattice vectors by to get the cartesian coordinates for the lattice vectos, which are the three lines that follow. Then there are the atom counts for each species (in reverse alphabetical order), and then the coordinate system. This can be "D" for direct, meaning that you multiply the first number of the basis vector by the first lattice vector, the second by the second, and the third by the third, then add up the x-coordinates, the y-coordinates, and the z-coordinates to find the cartesian coordinate of each atom within the unit cell. The "C" stands for cartesian coordinates, so each number is just an x, y, or z component of the basis atom's position. Next come the basis vectors, in either direct or cartesian coordinates.

If you didn't already know, the lattice vectors are the repeating unit of a crystalline structure. If you slide the origin to the end of one lattice vector, you will end up on the same position (as in, the same position but in a neighboring unit cell). The basis vectors are the positions of all the atoms in the unit cell. Usually there is one basis vector with a value of (0,0,0), meaning that the lattice vectors lie in the middle of one of the atoms.

INCAR

This is is "INput" car. This file has all the settings for the vasp calculations. In refGuide.pdf I talk about some of the settings we've used in past works. They're probably safe to use, but you might want to consult Brother Nelson about this.

PRECALC

This is the input file for k point generation. The only thing you need to touch in here is mindistance, unless you find yourself in some very dire circumstances. See refGuide.pdf for an explanation of such dire circumstances...

POTCAR

This is the "POTential" car. It has the psuedopotentials in it. You can `grep TITEL POTCAR` to make sure that the correct atom types are in here.

KPOINTS

This is generated by the getKPoints script. You probably don't ever really have to worry about it unless you forget to generate it.

OUTCAR

This is the "OUTput" car. It has some of the output in it. You can `grep TOTEN OUTCAR` to see the total energy. If the energy has converged, you will see "`free  energy`" with two spaces.

CONTCAR

This is an output file. If you let the atoms relax, this is the new POSCAR. Kinda. It just specifies the new positions of the atoms. Also I'm not sure what the "CONT" is.

CHGCAR

Not sure, maybe the "CHarGe" car? Ask Brother Nelson

OSZICAR

This has how many iterations Vasp has done in it. I'm not sure what other purpose it serves, and I'm also not sure what the "OSZI" is.