Brief introduction to ASE

Ask Hjorth Larsen
asklarsen@gmail.com

Nano-bio Spectroscopy Group
and ETSF Scientific Development Centre
Universidad del País Vasco UPV/EHU

May 14, 2018
Overview of ASE

ASE is a free (LGPLv2.1+) toolkit to set up and control atomistic calculations in a fully scripted environment using Python.

Main features

▶ The **Atoms** object: A collection of atoms
▶ Calculators: Capable of calculating energies and forces of atoms, often using an external code as backend
▶ Algorithms working with atoms/calculators: Structure optimization, molecular dynamics, basin hopping, minima hopping, nudged elastic band, . . .
▶ Many utilities: Build crystals, surfaces, . . .
▶ Read/write structures in many formats
▶ Also: GUI, command-line utilities
Example: Structure optimization

```python
from ase import Atoms
from ase.optimize import BFGS
from gpaw import GPAW

system = Atoms('H2O', positions=[[-1, 0, 0], [1, 0, 0], [0, 0, 1]])

system.center(vacuum=3.0)

system.calc = GPAW(mode='lcao', basis='dzp')

opt = BFGS(system,
            trajectory='opt.traj',
            logfile='opt.log')

opt.run(fmax=0.05)
```
Example 2: Structure optimization

```python
from ase import Atoms
from ase.optimize import BFGS
from ase.calculators.espresso import Espresso

system = Atoms('H2O', positions=[[1, 0, 0], [-1, 0, 0], [0, 0, 1]])

system.center(vacuum=3.0)

system.calc = Espresso(
    ecutwfc=40., pseudo_dir='.', tprnfor=True,
    pseudopotentials={'H': 'H_ONCV_PBE-1.0.upf', 'O': 'O_ONCV_PBE-1.0.upf'}
)

opt = BFGS(system, trajectory='opt.traj',
           logfile='opt.log')

opt.run(fmax=0.05)
```
Interface through file I/O

- ASE creates input file, runs programme (see figure)

Calculator daemon

- Calculator runs in background
- Read/write using sockets

Direct linking

- Everything within one process → efficient and flexible
- Also rather complicated
Build and view structures

```python
from ase import Atoms
from ase.visualize import view

a = 2.04

gold = Atoms('Au', pbc=True,
              cell=[[0, a, a],
                    [a, 0, a],
                    [a, a, 0]])

print(gold)
view(gold.repeat((2, 2, 2)))

from ase.build import molecule
view(molecule('C6H6'))
```
Try the ASE GUI

- Run ase gui
  (previously: ase-gui)
- Build nanoparticle or something else
- Select, move atoms
  (Ctrl+M)
- Save to your favourite format
More on ASE

- Started as an object-oriented Python interface to the old ultrasoft pseudopotential planewave code Dacapo
- BDFL: Jens Jørgen Mortensen, DTU Physics
- Very large number of contributors
Concluding remarks

- Web page: https://wiki.fysik.dtu.dk/ase/
- Gitlab: https://gitlab.com/ase/ase
- Mailing lists, IRC: https://wiki.fysik.dtu.dk/ase/contact.html