

Mona: **Calculation framework for reproducible science**

Jan Hermann

FHI coffee talk, 28. 11. 2018

Reproducibility and automation

```

xc pbe
relativistic at
sc_accuracy_eev
sc_accuracy_etof
sc_accuracy_rho

...
]
all_atoms.update
tags = {**defar
...

```

Energy and forces in a
 Total energy uncorrected
 Total energy corrected
 Electronic free energy

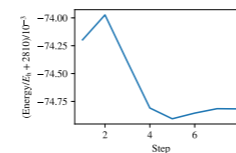
Start decomposition of the
 and C from original XC
 +tree-Fock Energy :

- *Reproducibility*
 = repeat a study and obtain identical results
- *Automation*
 = reduce human involvement in repeating a task
- Reproducible environments
- Cf. build automation

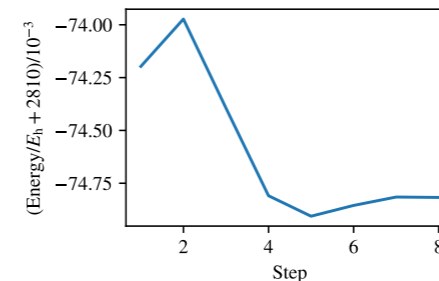
Mona Demo

November 9, 2018

The krypton calculation with PBE functional converged in 8 steps to within $2 \times 10^{-5} E_h$.



Step	(Energy/ E_h + 2810)/ 10^{-3}
1	-74.20
2	-73.97
3	-74.39
4	-74.81
5	-74.91
6	-74.86

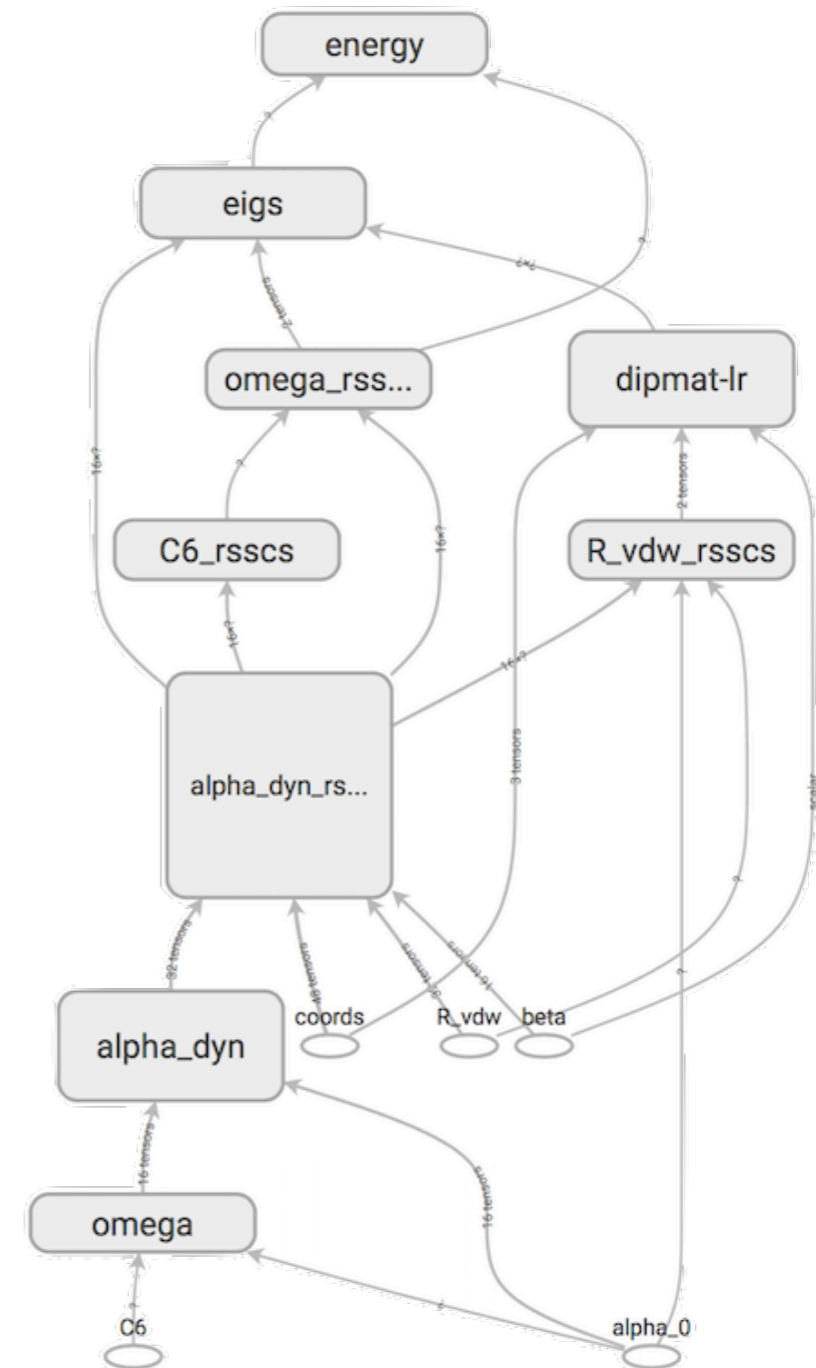


method,scale,N,MRE,I
 PBE,1.0,66.0,0.572,0.1
 PBE,2.0,65.0,0.4002,0
 PBE,inf,527.0,0.6448,(
 PBE+MBD(scs),1.0,66
 PBE+MBD(scs),2.0,65

- *Replicability*
 = arrive independently at the same conclusions

What is a calculation framework?

- Tool for easy creation and execution of calculation workflows (graphs)
- What calculations are supported?
- Should have quick bootstrap
- Caching of intermediate results (memoization)
- Parallel execution & distributed data
- Dynamic workflows
- Cf. Tensorflow
- Cf. Fireworks, AiiDA



Mona

- Calculation framework with Python functions as tasks
- Persistent memoization in a database
- Function call stack turned into a task graph
- Tasks can create new tasks

$$F_n = F_{n-1} + F_{n-2}, \quad F_1 = F_2 = 1$$

```
from mona import Rule
```

```
@Rule
async def fib(n):
    if n <= 2:
        return 1
    return total([fib(n - 1), fib(n - 2)])
```

```
@Rule
async def total(xs):
    return sum(xs)
```

