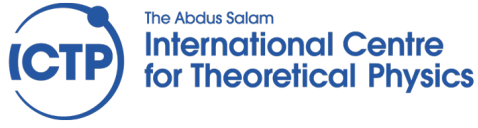
Introduction to YamboPy

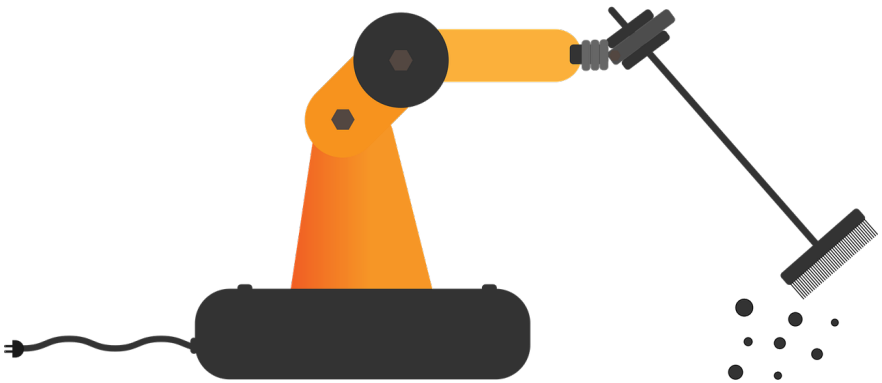
Michael Atambo - Technical University of Kenya RASESMA 2023



CNR
Istituto di Struttura
della Materia

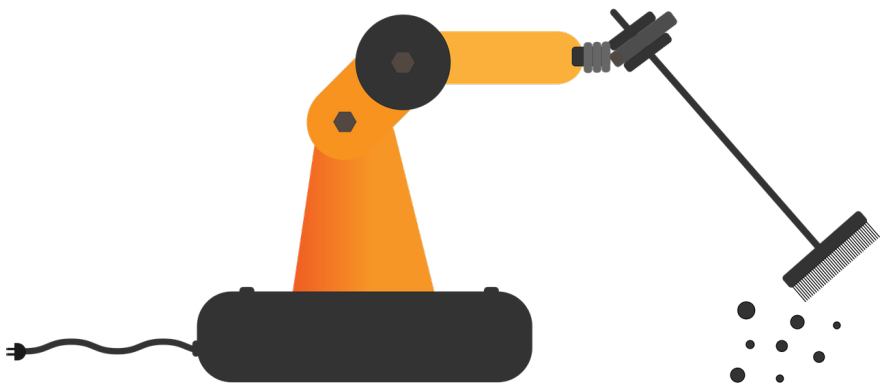
Automation in Condensed Matter: Motivation

- Modern computational science is fast paced
- **Significant** computing resources are available
- The landscape of possible materials is **huge**
  - you want magnetism? >40000 materials on materials project deposited
  - XYZ? probably something exists,



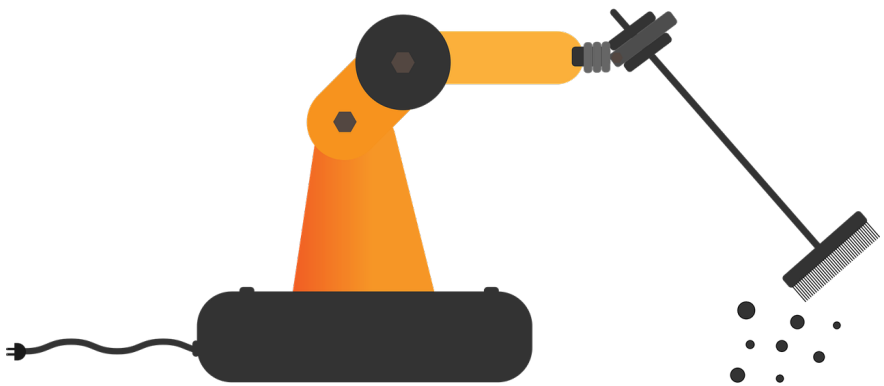
What is Automation In Our Context?

- Hey siri, *where is my phone?*
- We want to be able to compute something related to condensed matter
- Prefarably with physics codes we already use,



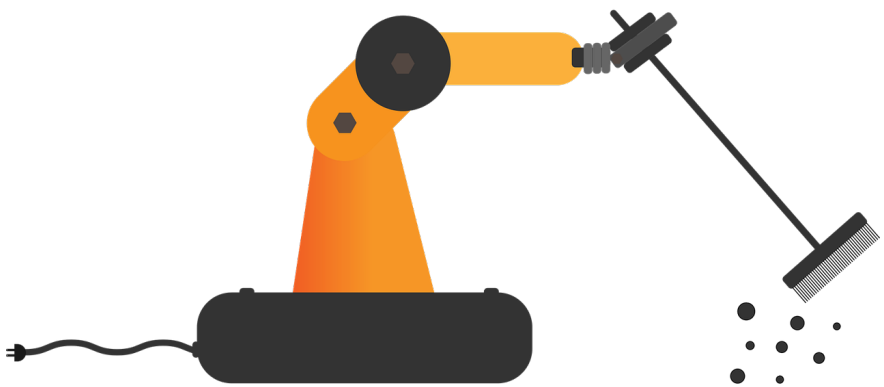
Automation in Condensed Matter

- Tools exists to provide some solutions
  - https://workflows.community/systems
  - Awesome List
- **Varied** in what they can do
- but,



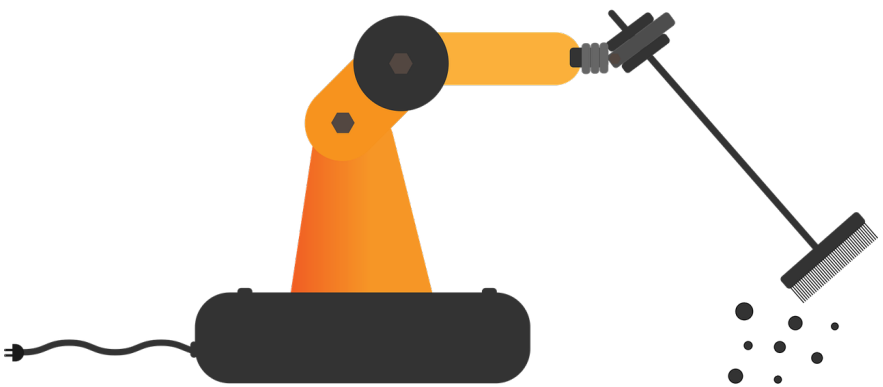
Automation in Condensed Matter

- What would be the user **needs**?
- What has to be **stored**?
- What can be **automated**?



Automation in Condensed Matter

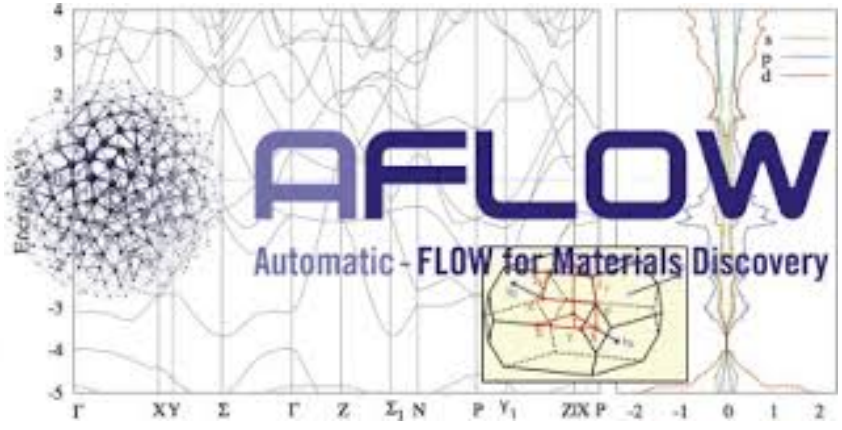
- Some Select (focused) existing tools,
- **AiiDA** (aiida.net)
- **AFLOW** (https://www.aflowlib.org/)
- **Atomate**, Comput. Mater. Sci. 139, 140–152 (2017).
- **Express** (https://github.com/MineralsCloud/Express.jl)



Aiida and Aflow:

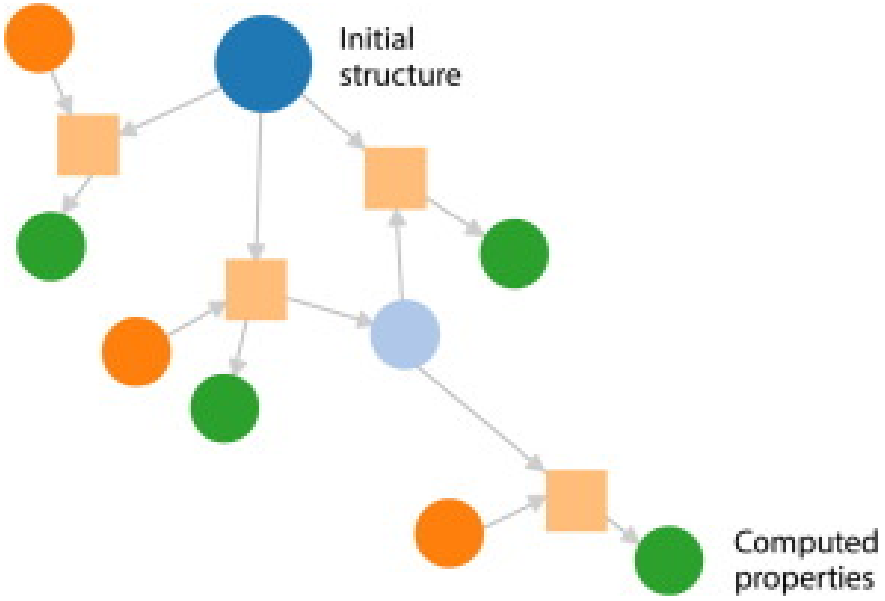
- Large and complex,
- Aims
  - track data as its generated
  - track provenance
  - provide querying interface
  - scripting (automation),
  - **workflows**
  - reproducibility...

AiiDA
Automated Interactive Infrastructure and Database for Computational Science



Workflows

- These provide some means to encode logic,
- This is a big idea:
  - We can encode the users scientific knowledge!
- Engines:
  - https://workflows.community/systems

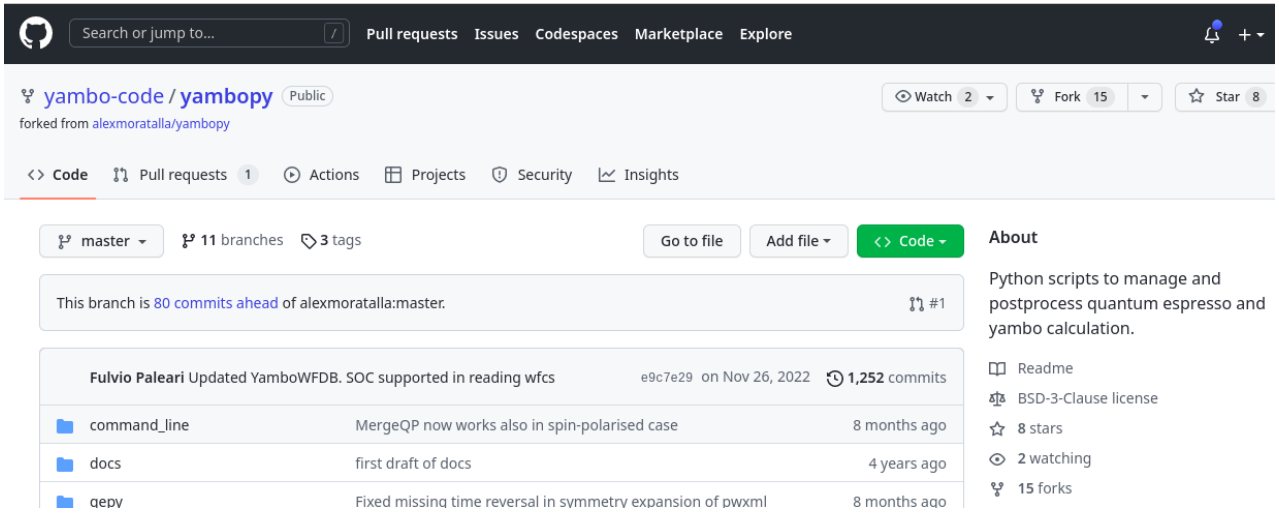


Automation and Scripting

- Reducing our focus:
  - first step is scripting
  - we wont get to workflows today, or AiiDA
- Yambopy is a focused tool that can address what we want to do:
  - automate/script yambo calculation
  - analyze the results of a calculation

Yambopy

- This project provides python scripts to manage and postprocess quantum espresso and yambo calculation.
- Can automate your calculations,
- provide post-processing capabilities
- Visualization
- Input file manipulation, etc.
- Its FOSS! https://github.com/yambo-code/yambopy



- From: Alejandro Molina Sanchez Fulvio Paleari. . .

Yambopy Featues

- Create Yambo **Inputs**
- Run **Yambo**
- Post processing,
  - Yambo output
  - Netcdf examination
  - plotting
- QE **Inputs**
  - Phonons (ph.x)
  - Dynmat (dynmat.x)
  - PW (pw.x)
- Interacting with batch queues (PBS)

Yambopy Installation

- The package is available on pypi.

```
pip install yambopy
```

- can be installed from source also:

```
git clone https://github.com/yambo-code/yambopy.git
cd yambopy
python setup.py install
```

- requirements:
  - Yambo
  - numpy
  - scipy
  - matplotlib
  - **matplotlib***
  - QE...

How Yambopy works:

- Yambopy in written in python,
- The important stuff is all encapsulated in one class each
  - A class for QE input creation
  - ...
  - in general a class for every step

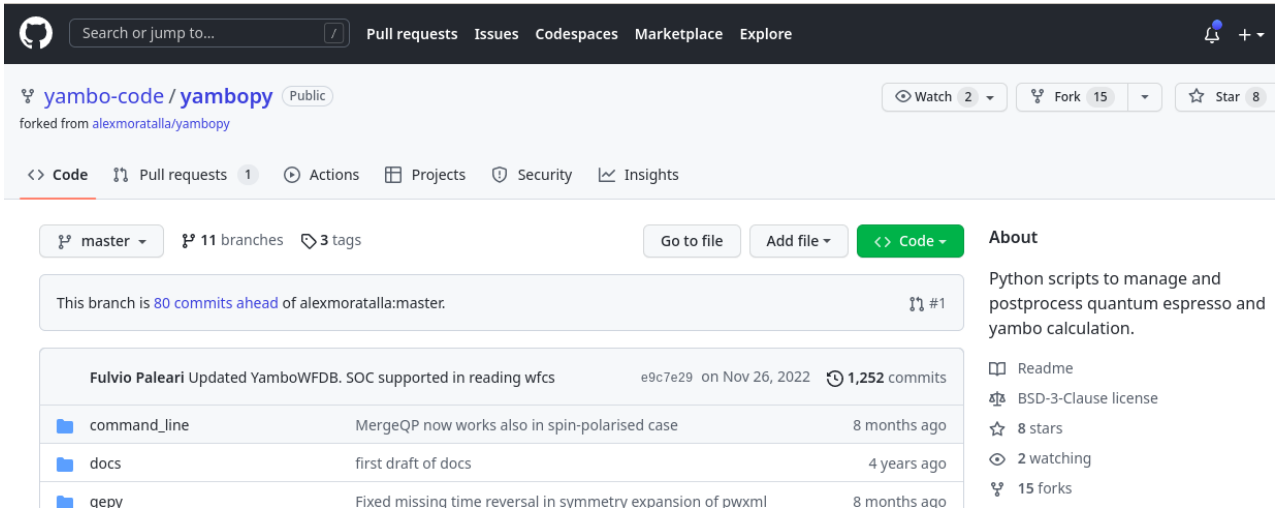
```
from qepy import PwIn
qe = PwIn()
qe.atoms = [['Si',[0.125,0.125,0.125]],
['Si',[-.125,-.125,-.125]]]
qe.atypes = {'Si': [28.086,"Si.pbe-mt_fhi.UPF"]}
...
qe.system['ntyp'] = 1
qe.system['ibrav'] = 2
qe.write('si.scf')
```

How Yambopy works:

- Other Classes that encode some functionallity
  - **schedulerpy**
    - job execution (interaction with PBS)
  - **yambopy**
    - read/write input files and read outputs from yambo
  - **PWIn**
    - read/write QE input files, pw.x
  - **PhIn**
    - read/write QE input files, ph.x
  - **DynmatIn**
    - read/write QE input files, dynmat.x
  - **YamboIn**
    - read/write Yambo input files
- Commandline executables
  - **yambopy** scripts that can provide features of yambopy from the command line

What will you get out of the exercise?

- Automation reduces **mistakes**.
- Increase the **pace** of your work.
- **High Througput** studies.
- Key: **reproducibility**, can we achieve this?



Requirements

- These python scripts require **python3**
- We will need:
  1. **yambopy**, Already Installed
  2. **yambo**, Installed
  3. **Quantum Espresso**, installed

Aims of this Exercise:

- We will work on the Convergences and Approximations tutorial: click here
- And the database tutorial: click here
- Finally the BSE exercise: Click here

Retreving the files:

- Copy the files to your home directory and work from there:

```
~$ cp -r /media/ictpuser/smr3694/ictptutor/yambopy/tutorial ~/
    tutorial
~$ cd ~/tutorial
```

Getting started

- You have to use spack to get the environment that provides yambopy:

```
~$ spack load quantum-espresso
~$ spack load yambo
~$ spack load anaconda3
```

Running yambopy:

- First exercise,

```
~$ cd ~tutorial/run_calculations
~$ python gs_bn.py -sn
...
```

- Follow the steps here: click this
- You will run QE followed by GW convergence

Analysis:

- Yambopy provides the capability to read and analyze the yambo NetCDF databases, for this exercise, start from this directory:

```
~$ cd ~tutorial/databases_yambopy
~$ yambopy save -nscf BSE_saves/QE_saves/hBN.save
...
```

- Continue with the tutorial: click here

BSE with Yambopy

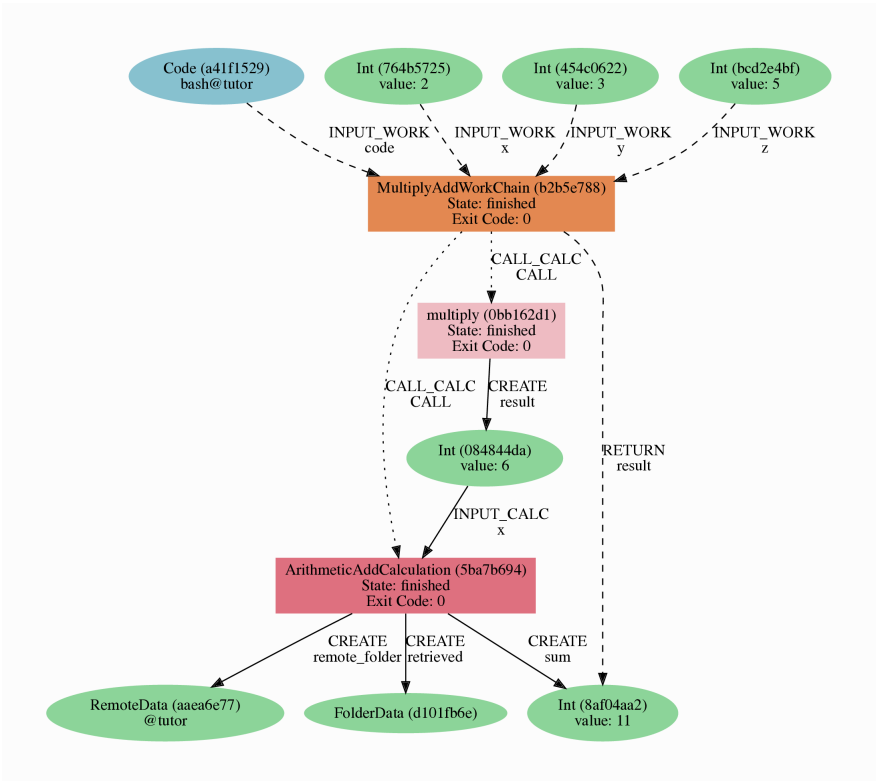
- This requires the first exercise to have been done
- We continue from the first exercise's directory:

```
1 ~$ cd ~tutorial/run_calculations
2 ~$
3 ...
```

- Continue with the instructions here: click this

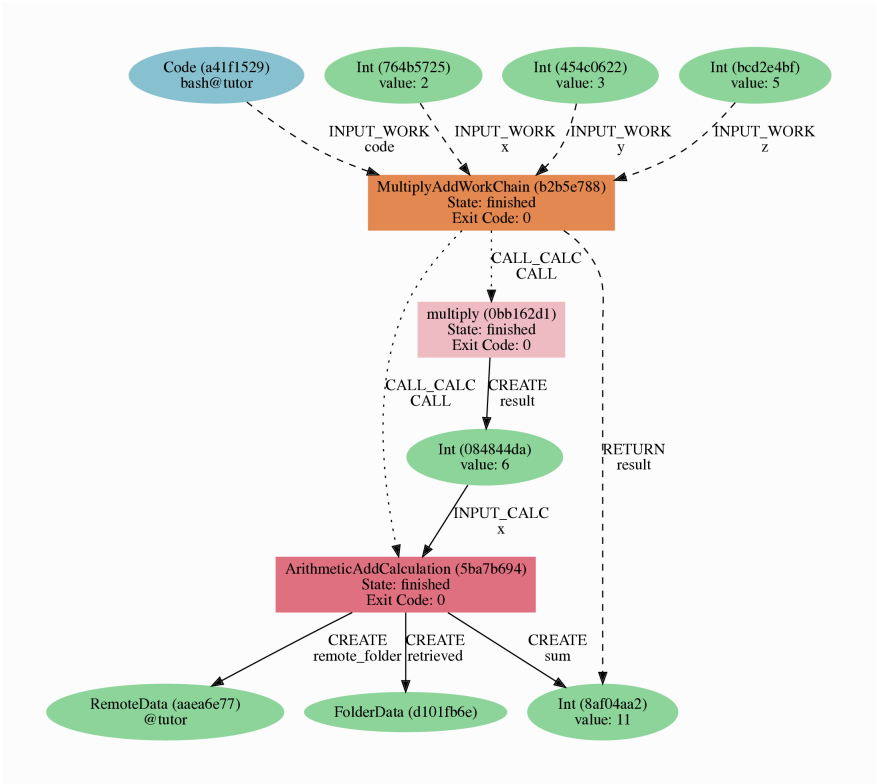
Going beyond Scripting:

- Richer Workflows
- AiiDA
- provenance
- databases
- querying
- HTC



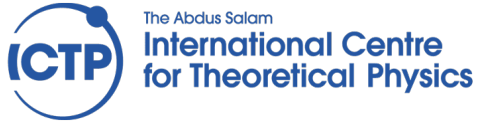
AiiDA-Yambo

- Aiida works through plugins,
- the core (Aiida) provides the management engine, querying, provenance
- plugins interact with the physics codes (QE , VASP, Yambo, Siesta)
- For G0W0 you have **aiida-yambo** (https://github.com/yambo-code/aiida-yambo)
  - I worked on this tool earlier.



Thanks

- Thank you for your attention.
- questions?



CNR
Istituto di Struttura
della Materia