

PSI

Center for Scientific Computing,
Theory and Data

AiiDAlab Quantum ESPRESSO App

Streamlining Core-Level Spectroscopy Calculations

Xing Wang

Postdoc, Materials Software and Data Group,

Paul Scherrer Institute, Switzerland

RACXS workshop, 18-06-2024

Outline of one-hour code tutorial



Part 1: Streamlining scientific workflows with AiiDA

Target Audience: Computational Scientists

Goals:

- Understand the idea of creating workflows using AiiDA
- Learn about checkpoints, provenance tracking (database), and remote execution

Part 2: AiiDALab Quantum ESPRESSO App

Target Audience:

- Experimental Scientists needing DFT calculations on material properties.
- Computational Scientists want to providing DFT services

Goals:

- Easy-to-use graphical interface
- Accessible, turn-key solution for non-specialists
- Facilitates collaboration between experimentalists and theorists

I will conduct live calculation during the session to showcase practical applications.

Your Role: **Part 1: Only a basic understanding of Python is needed.**

- You are not required to perform the calculations during the session.
- Full instructions will be provided so you can independently run all demonstrated calculations at your convenience.

Interactive Sessions:

- Each part will have ~ 5 minutes for questions and discussions.
- Please feel free to ask questions or share your thoughts during these intervals, or anytime during the conference.

Small request: speak slowly and maybe repeat your question



AiiDA: Automated Interactive Infrastructure and Database for Computational Science

An open-source Python package to help researchers with automating, managing, persisting, sharing and reproducing the complex workflows associated with modern computational science and all associated data.

Key features:

- Automated workflows
- High-throughput
- Data provenance
- Database and advanced queries
- Plugin interface
- HPC interface
- Open science
- Open source



- **Workflow automation**
 - Need tools to define complex workflows with advanced error handling
 - An automated, robust and scalable engine to run the workflows
- **Data management**
 - Data should be stored reliably and efficiently
 - Stored data should be interoperable and queryable, and easy sharing
- **Reproducibility**
 - All produced data should be reproducible by storing the full provenance

Data provenance

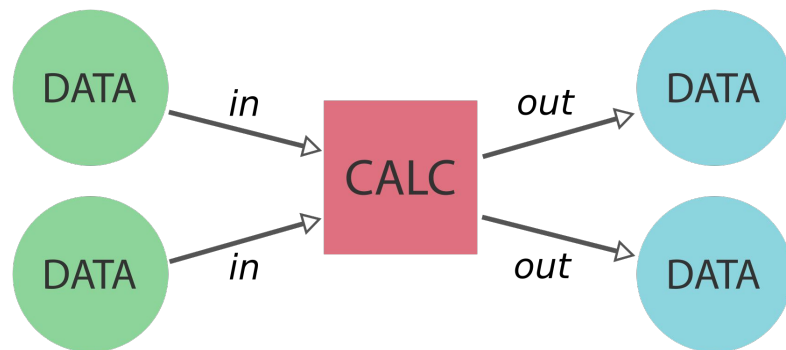


Simple recipe

- Store the **'calculations'**
- Store its **inputs** and their metadata
- Store its **outputs** and their metadata
- Most **crucially** store the **inter-connections**

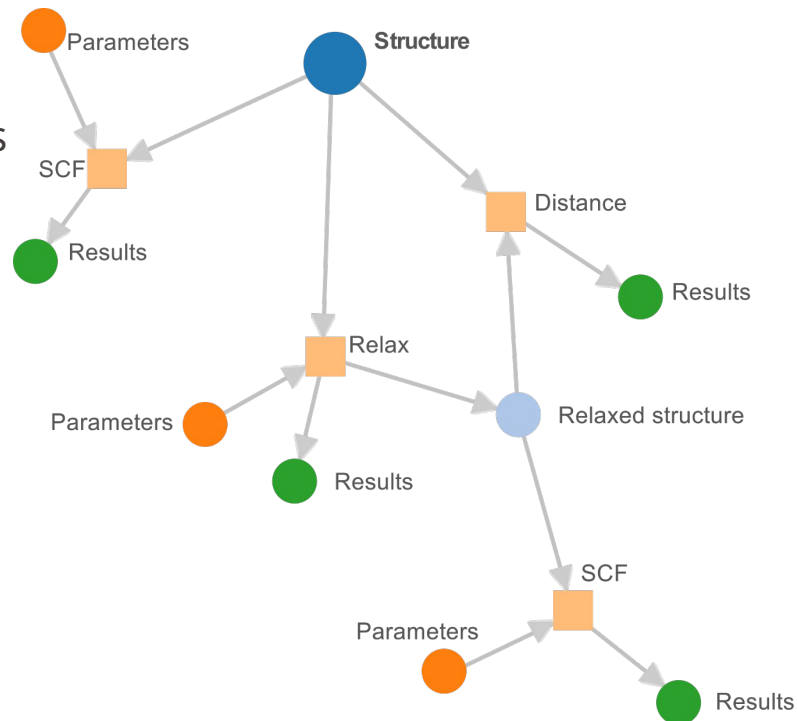
Calculation: $x+y$

```
def add(x, y):  
    return x + y  
  
add(1, 2)
```

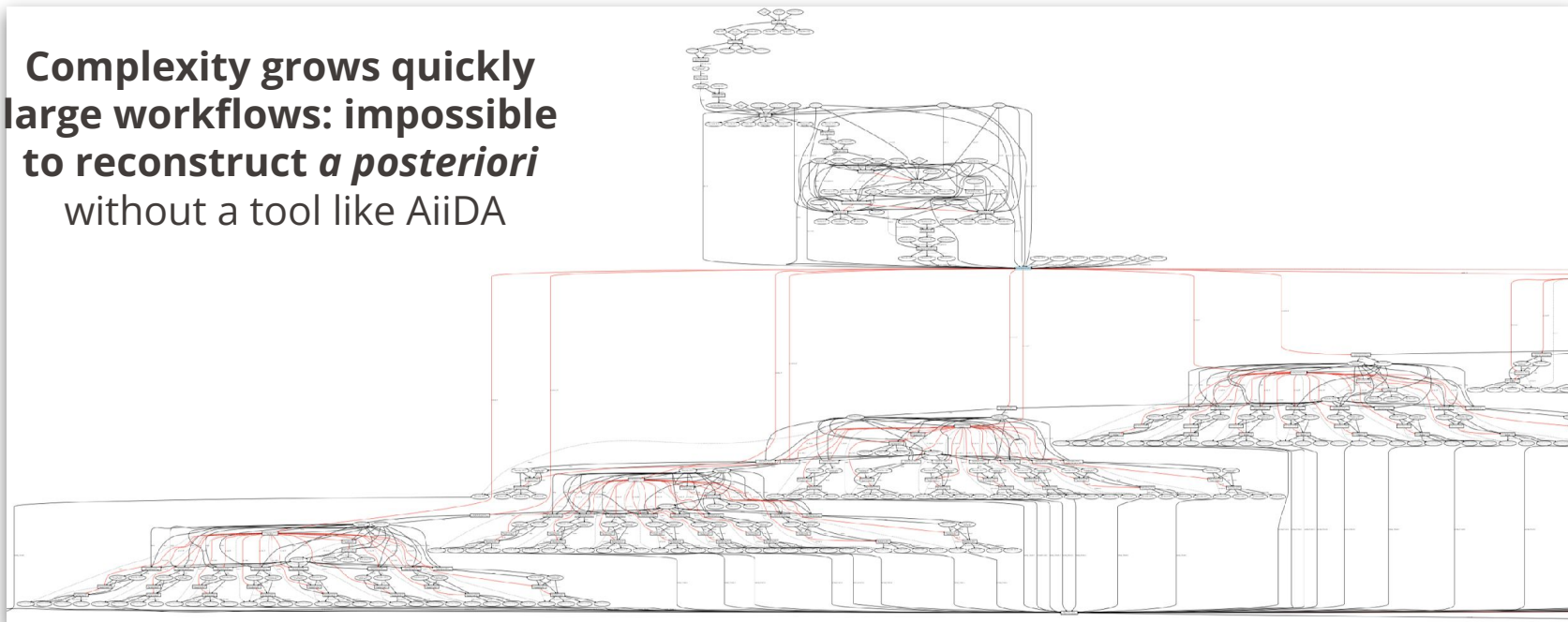


Provenance graphs

- When data gets reused, a directed graph is created
- That quickly grow in complexity for workflows

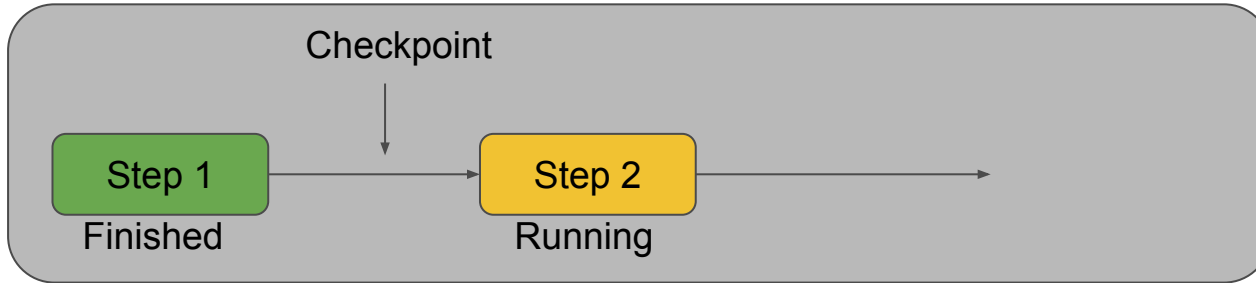


**Complexity grows quickly
large workflows: impossible
to reconstruct *a posteriori*
without a tool like AiiDA**



Molecular dynamics study of Lithium in a solid electrolyte

Automatic checkpointing, which guarantees that work between steps is saved, thus can be restored after it had been interrupted. Important for long-running calculation, like DFT.



- **WorkChain**
- **WorkGraph**: flexible and easy to use

AiiDA-WorkGraph



Workflow example: $(x + y) * z$

step 1 step 2

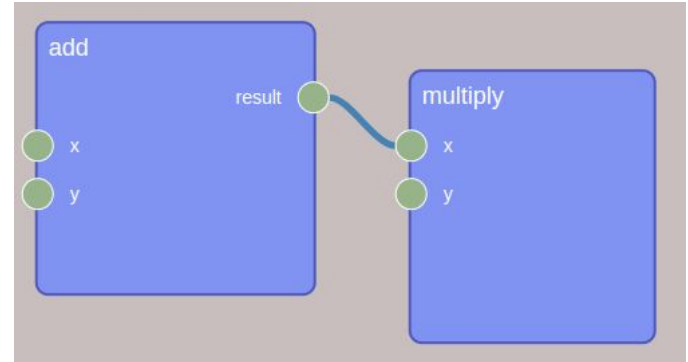
- 1) Create a empty workgraph
- 2) Add the tasks (the add and multiple functions)
- 3) Link the task

```
from aiida_workgraph import WorkGraph

@calfunction()
def add(x, y):
    return x + y

@calfunction()
def multiply(x, y):
    return x*y

wg = WorkGraph("add_multiply")
wg.tasks.new(add, name="add")
wg.tasks.new(multiply, name="multiply",
             x=wg.tasks["add"].outputs["result"])
```



- **CalcJob**: run code (e.g. DFT code) on remote computer.
- **ShellTask**: run shell command on a remote computer
- **PythonTask**: run Python function on a remote computer (Ongoing)

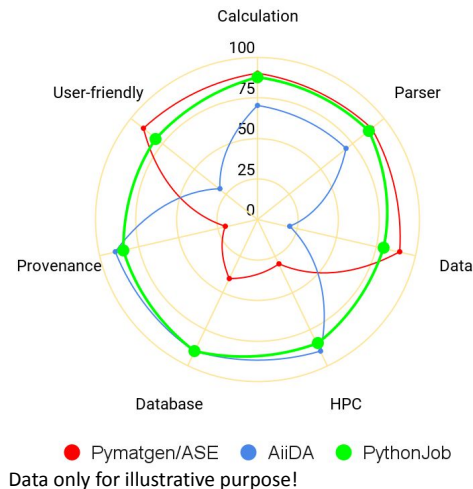
```
@task
def add(x, y):
    return Int(x + y)
```

```
wg = WorkGraph("second_workflow")
wg.tasks.new(add, name="add", run_remotely=True)
```

Turn it into a PythonTask, and run this function on a remote computer

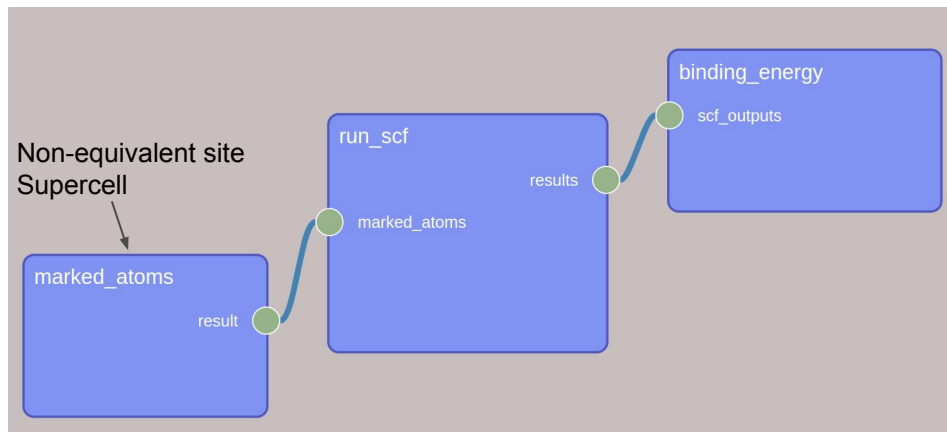
```
# run on CSCS eiger cluster
metadata = {
    "options": {
        'prepend_text' : """
module load cray/22.05 cpeIntel/22.05 QuantumESPRESSO/7.0
eval "$(/users/xingwang/miniconda3/bin/conda shell.posix hook)"
conda activate py3.11
export OMP_NUM_THREADS=1
        """,
        'resources': {
            'num_machines' : 1,
            'num_mpiprocs_per_machine' : 128,
        }
    }
}
```

Integrate Your Favorite Tools with AiiDA



- Espresso calculator

- [Calculator](#)
- [Atomization energy](#)
- [Equation of state \(EOS\)](#)
- [Elastic constants](#)
- [Bands structure](#)
- [Projected density of states \(PDOS\)](#)
- [Bader Charge](#)
- [X-ray photoelectron spectroscopy \(XPS\)](#)



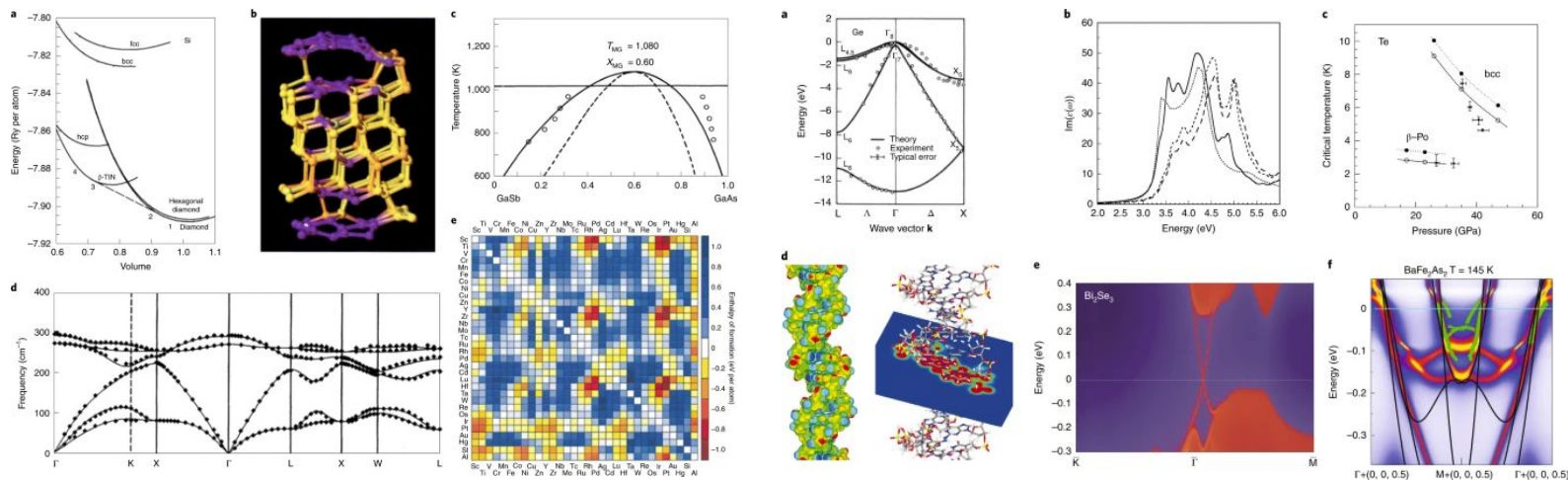
AiiDA: Automated Interactive Infrastructure and Database for Computational Science

Key Features:

- **Provenance:** Track the provenance of the workflow.
- **Easy to build workflow:** Create workflows by linking the input and output socket of different tasks.
- **Flexible:** Extend (modify) the workflow by adding (editing) tasks and links, or combine multiple workflows together.
- **Interactive GUI:** Visualize and interact with the workflow using the GUI. (Ongoing)
- **Checkpoints:** Save the workflow state, and restore the workflow from the a interrupted state.
- **Remote execution:** Execute the task (Python function, Shell command) on a remote machine.

Part 2 AiiDAlab Quantum ESPRESSO App

- Remarkable predictions in the wide range of properties and spectroscopies



Marzari, N., Ferretti, A. & Wolverton, C. Nat. Mater. 20, 736–749 (2021).

Bring the first-principle tools to other communities

Broadly available simulation codes and workflows.

Barrier



Big research and industry communities

Software
Installation and
Configuration

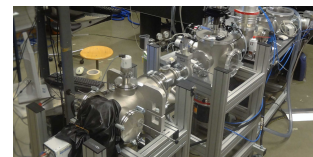
Input File Preparation

Parameter Tuning

Interpreting Results

Computational
Resources

Validation and Benchmarking



Bring the first-principle tools to big communities

Broadly available simulation codes and workflows.

Our Goal
Automated, reliable,
"turn-key"
simulations tool

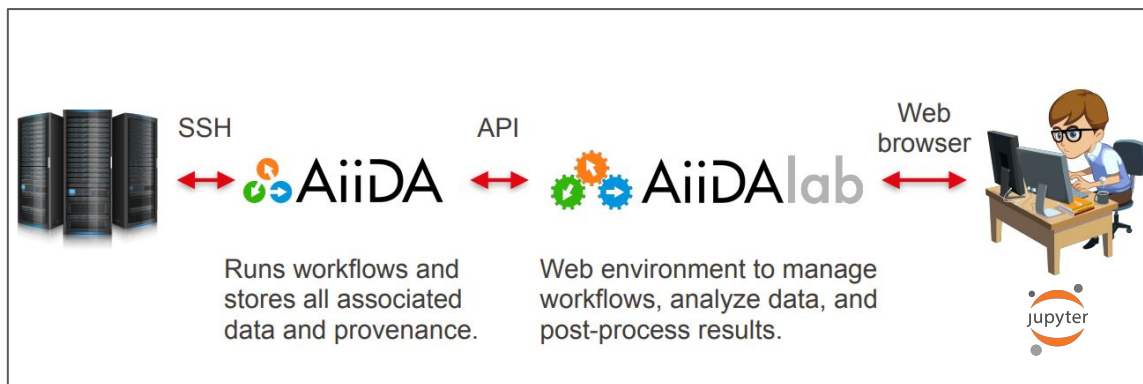
Big research and industry communities



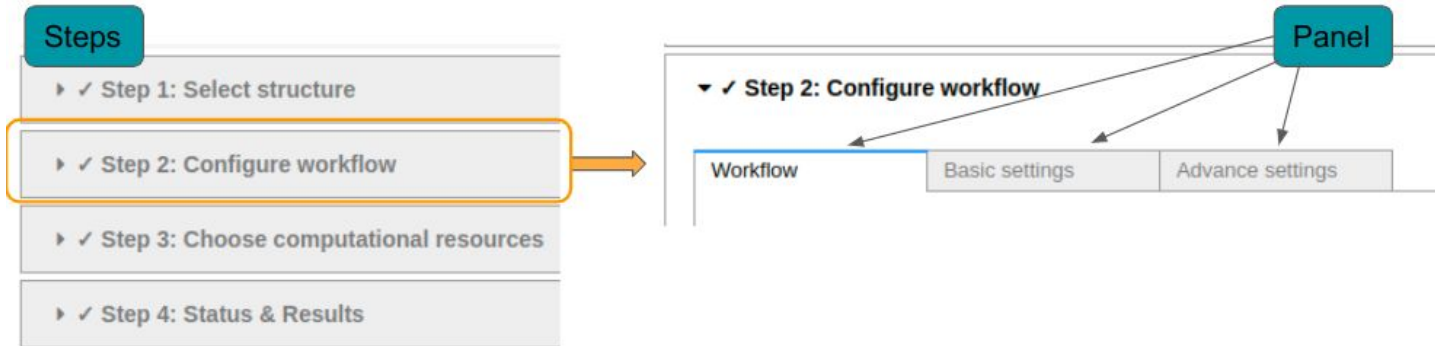
Barrier



- Quantum ESPRESSO (QE) app is an AiiDALab web-based graphical user interface allowing users to conduct first-principle calculations on the web browser directly without writing any code.
 - Automated
 - Reliable
 - Intuitive user interface
 - Easy to share and maintain

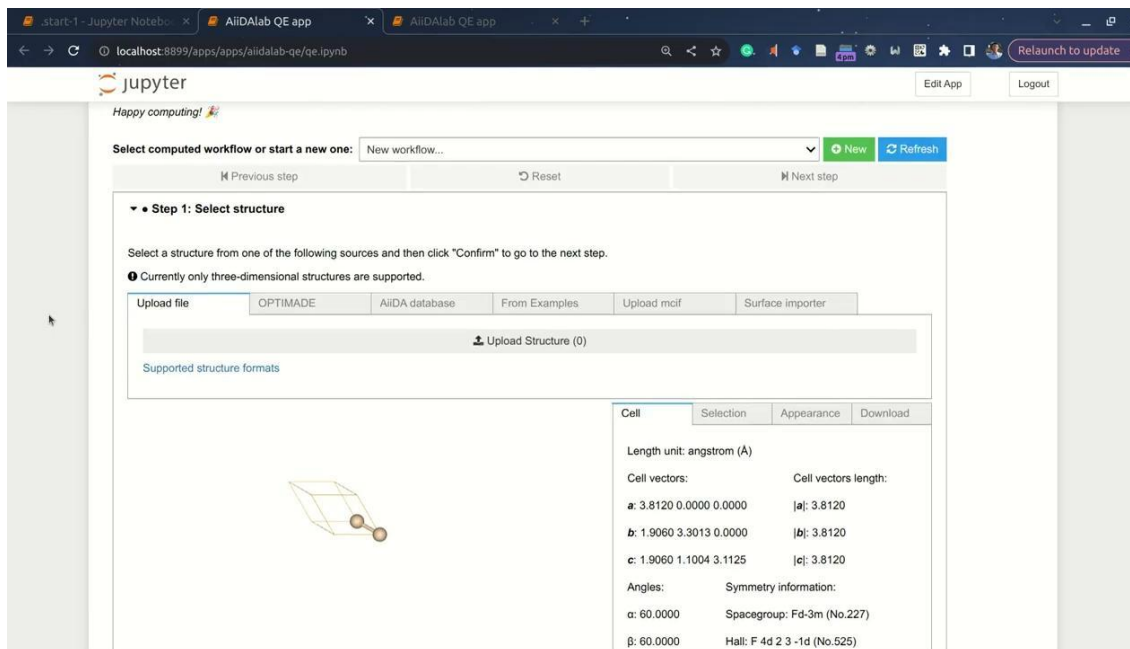


- Quantum ESPRESSO app uses the Wizards UI, which divides one calculation job into four steps.
- Each step contains several panels.



Step 1 Select a structure

- Structures can be uploaded as a file in any standard format (XYZ, CIF, etc.) or be downloaded from online databases (e.g., Materials Cloud, Materials Project) using the [OPTIMADE API](#).



Happy computing!

Select computed workflow or start a new one: New Refresh

Previous step Reset Next step

▼ Step 1: Select structure

Select a structure from one of the following sources and then click "Confirm" to go to the next step.

Currently only three-dimensional structures are supported.

Upload file OPTIMADE AiiDA database From Examples Upload mcif Surface importer

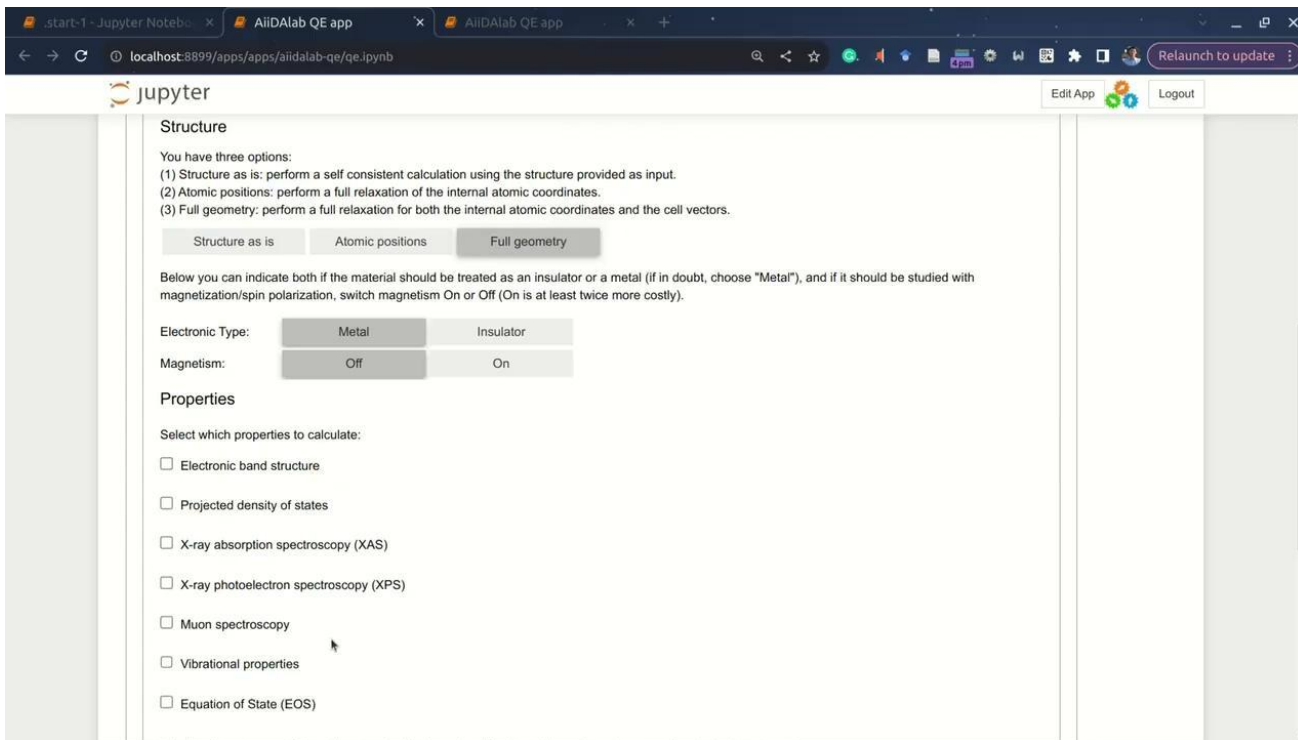
Upload Structure (0)

Supported structure formats

Cell	Selection	Appearance	Download
Length unit: angstrom (Å)			
Cell vectors:		Cell vectors length:	
a:	3.8120 0.0000 0.0000	a :	3.8120
b:	1.9060 3.3013 0.0000	b :	3.8120
c:	1.9060 1.1004 3.1125	c :	3.8120
Angles:		Symmetry information:	
α:	60.0000	Spacegroup:	Fd-3m (No.227)
β:	60.0000	Hall:	F 4d 2 3 -1d (No.525)

Step 2: Set the parameters for DFT calculation

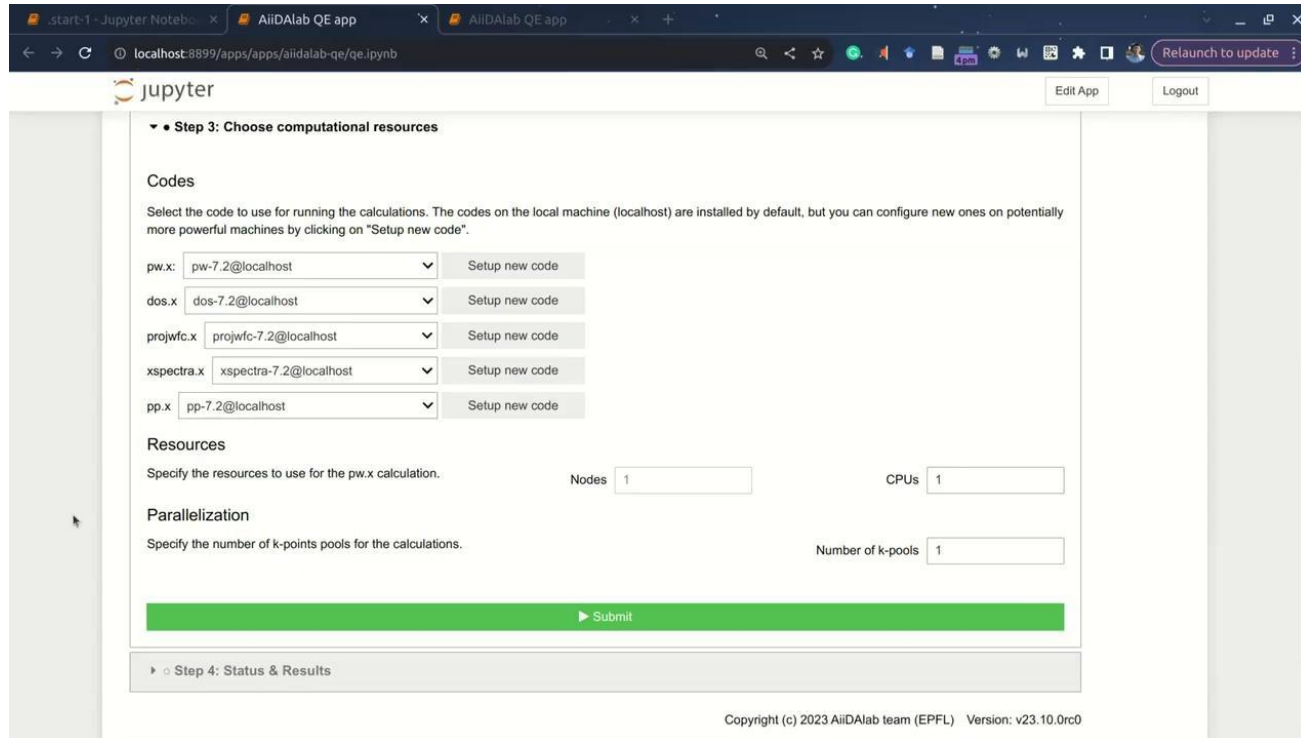
- Easy parameter selection for non-experts by choosing the protocols, which are developed and tested for a large database of structures.



The screenshot shows a web browser window with the URL `localhost:8899/apps/apps/aiidalab-qe/qe.jpynb`. The page is titled "Structure" and provides instructions for three calculation options: (1) Structure as is, (2) Atomic positions, and (3) Full geometry. The "Full geometry" option is selected. Below this, users can choose the material's electronic type (Metal or Insulator) and magnetism (On or Off). The "Metal" and "Off" options are selected. Under the "Properties" section, a list of properties to calculate is shown with checkboxes: Electronic band structure, Projected density of states, X-ray absorption spectroscopy (XAS), X-ray photoelectron spectroscopy (XPS), Muon spectroscopy, Vibrational properties, and Equation of State (EOS). The "Muons spectroscopy" checkbox is currently selected.

Step 3: Set up the computational resources

- Set up the code on the local machine or on the remote HPC.



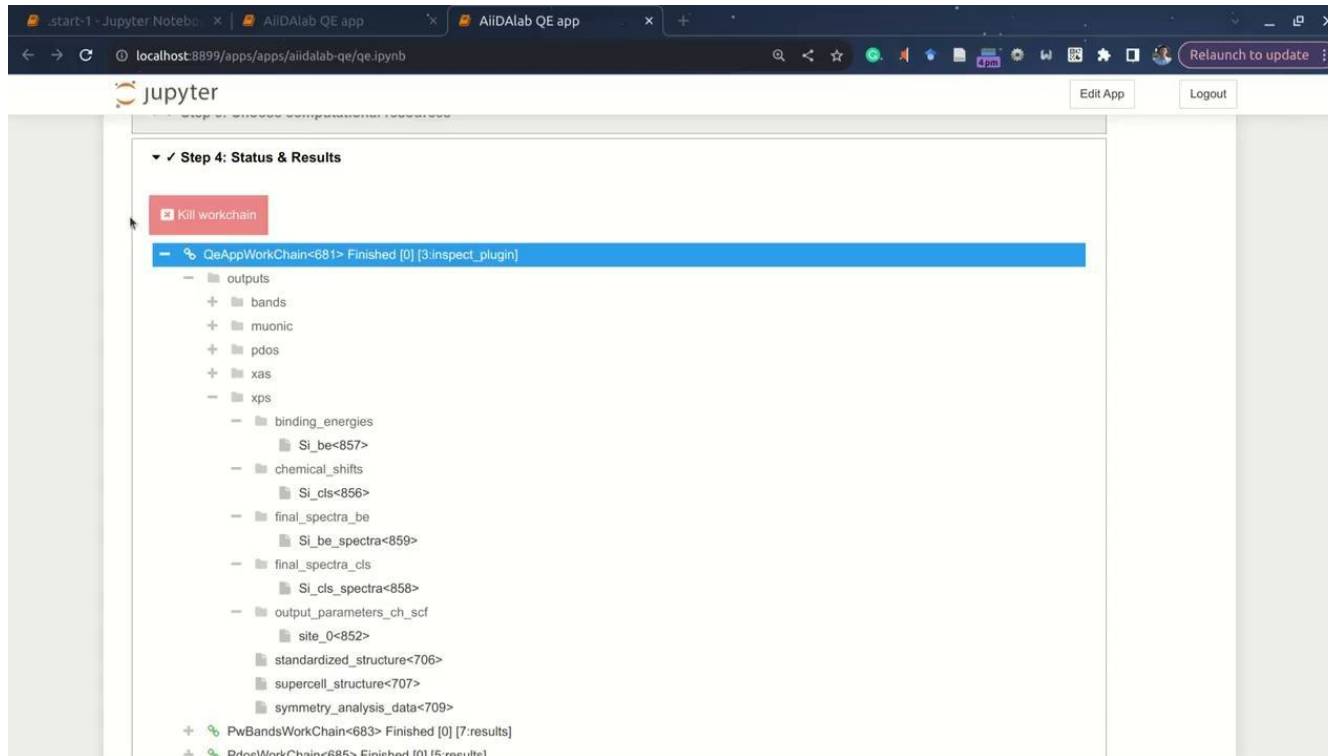
The screenshot shows the JupyterLab interface for setting up computational resources. The browser address bar indicates the URL is localhost:8899/apps/aiidalab-qe/aiidalab-qe.ipynb. The JupyterLab header shows the 'jupyter' logo and 'Edit App' and 'Logout' buttons. The main content area is titled 'Step 3: Choose computational resources' and is divided into three sections:

- Codes:** A section for selecting code environments. It lists five codes: pw.x, dos.x, projwfc.x, xspectra.x, and pp.x. Each code has a dropdown menu showing its current location (all are set to localhost) and a 'Setup new code' button.
- Resources:** A section for specifying resources for the pw.x calculation. It includes input fields for 'Nodes' (set to 1) and 'CPUs' (set to 1).
- Parallelization:** A section for specifying the number of k-points pools. It includes an input field for 'Number of k-pools' (set to 1).

A large green 'Submit' button is located at the bottom of the configuration area. Below the configuration area, there is a link to 'Step 4: Status & Results'. At the bottom right of the page, the copyright information reads: 'Copyright (c) 2023 AiIDALab team (EPFL) Version: v23.10.0rc0'.

Step 4: Visualize the results

- Visualize the result interactively.

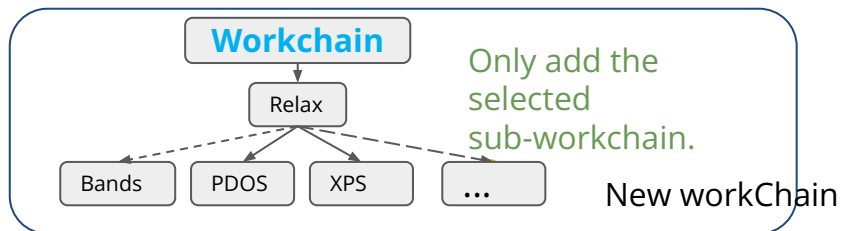
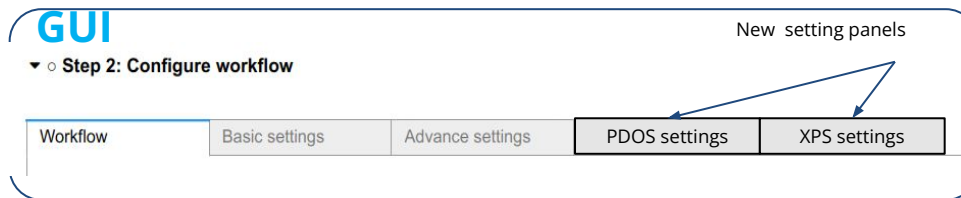
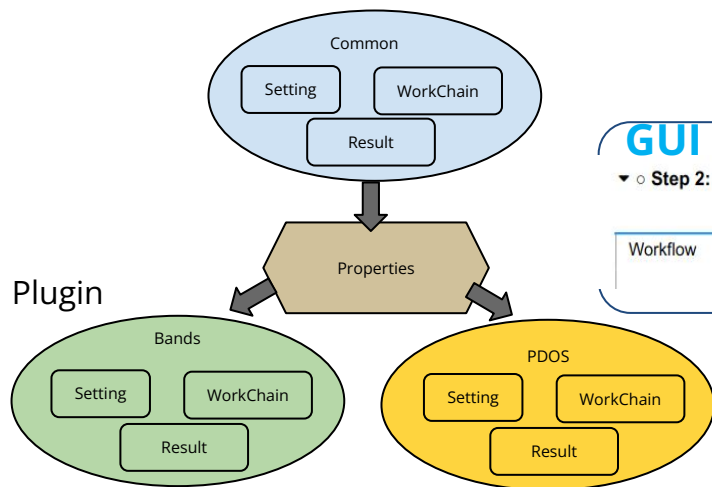
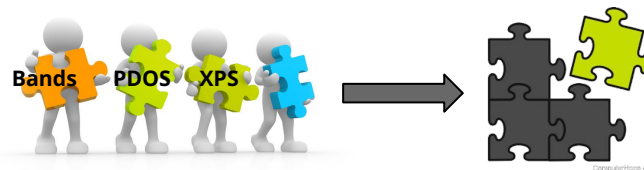


The screenshot displays a JupyterLab interface with the following elements:

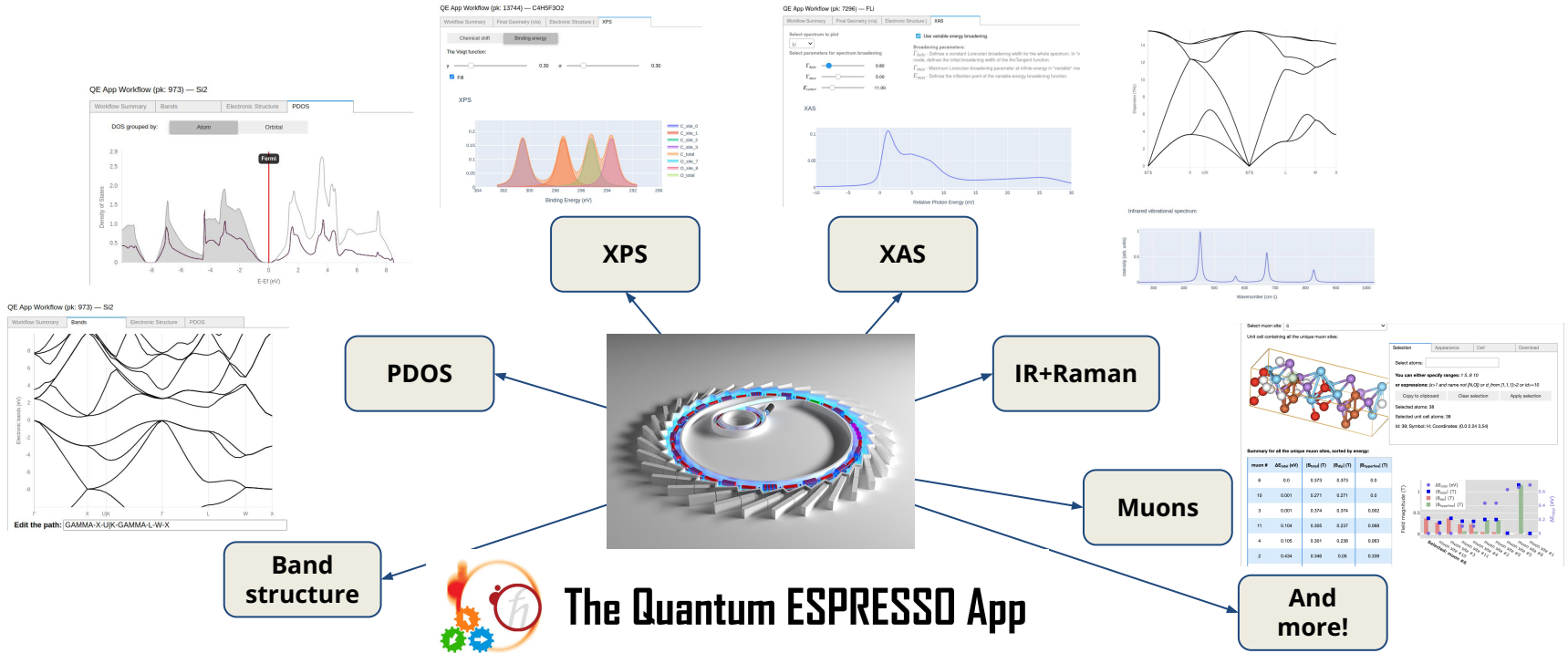
- Browser Tabs:** Three tabs are open: ".start-1 - Jupyter Notebo...", "AiiDALab QE app", and "AiiDALab QE app".
- Address Bar:** Shows the URL "localhost:8899/apps/apps/aiidalab-qe/qe.ipynb".
- JupyterLab Header:** Includes the "jupyter" logo, "Edit App", and "Logout" buttons.
- Main Content Area:**
 - Step 4: Status & Results:** A section header with a dropdown arrow.
 - Kill workchain:** A red button with a trash icon.
 - QeAppWorkChain<681> Finished [0] [3:inspect_plugin]:** A blue bar representing the selected workchain.
 - Tree View:** A hierarchical list of outputs:
 - outputs
 - bands
 - muonic
 - pdos
 - xas
 - xps
 - binding_energies
 - Si_be<857>
 - chemical_shifts
 - Si_cls<856>
 - final_spectra_be
 - Si_be_spectra<859>
 - final_spectra_cls
 - Si_cls_spectra<858>
 - output_parameters_ch_scf
 - site_0<852>
 - standardized_structure<706>
 - supercell_structure<707>
 - symmetry_analysis_data<709>
 - PwBandsWorkChain<683> Finished [0] [7:results]:** A green bar representing another workchain.
 - PdosWorkChain<685> Finished [0] [5:results]:** A green bar representing a third workchain.

Architecture: Plugin

- Running multiple calculations (bands, pdos, XPS, etc) in one app using the same interface.



Power of Quantum ESPRESSO app



Contributors: E. Bainglass (PSI), M. Berx (PSI), M. Bonacci (PSI), D. Dou (EPFL), P. Gillespie (CNR-NANO Modena), M. A. Hernández-Bertrán (CNR-NANO Modena), D. Hollas (Bristol), N. Marzari (EPFL, PSI), A. Ortega-Guerrero (Empa), C. Pignedoli (Empa), D. Prezzi (CNR-NANO Modena), G. Pizzi (PSI), I. Timrov (EPFL, PSI), J. Yu (PSI), A. Yakutovich (Empa)

Theory

Δ KS

$$E_B = E_+ - E_0$$

Pseudo-potential with core-hole

Molecular systems:

$$E_B = E_{\text{ch}}^+ - E_0$$

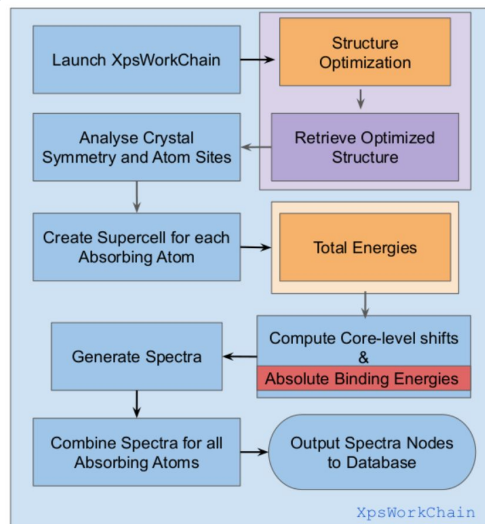
Solid systems:

$$E_B^F = E_{\text{ch}}^+ - E_0 + \epsilon_F$$

$$E_{\text{ch}}^0 = E_{\text{ch}}^+ + \epsilon_L$$

$$E_B^F = E_{\text{ch}}^0 - E_0 + \epsilon_F - \epsilon_L$$

AiiDA workflow



AiiDALab GUI

QE App Workflow (pk: 13744) — C4H5F3O2



Collaborate with Dr. Prezzi's group
(CNR-NANO Modena), Dr. Iurii
Timrov (PSI)

Example: Phenylacetylene (PA) molecule

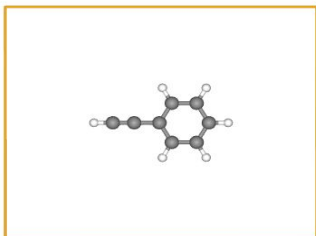
Step 1: Select structure

Select a structure from one of the following sources and then click "Confirm"

Currently only three-dimensional structures are supported.

Upload file | OPTIMADE | AiiDA database

Phenylacetylene molecule



Label C8H6

Description

Step 2: Configure workflow

Basic settings | Advanced settings | XPS Settings

Structure

Below you can indicate if the material should be treated as a molecule or crystal

Molecule | Crystal

Core-Hole pseudopotential group

Please select a pseudopotential group, which provide the ground-state and downloaded from this [repository](#).

Group: pseudo_demo_pbe

Select core-level

The list of core-levels to be considered for analysis.

O_1s

C_1s

H, not supported by the selected pseudo group

Step 3: Choose computational resources

Codes

Select the code to use for running the calculations. The codes on the local machine (localhost) are installed by default, but you can configure new ones on potentially more powerful machines by clicking on "Setup new code".

pw.x: pw-7.0@merlin-cpu Setup new code

Resources

Specify the resources to use for the pw.x calculation.

Nodes 1 CPUs/node 44

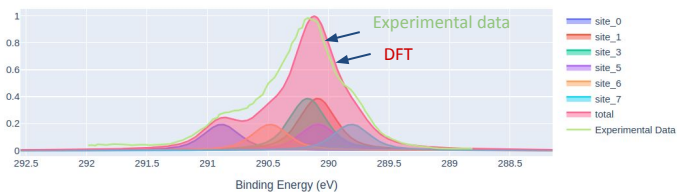
Parallelization

Specify the number of k-points pools for the calculations.

Number of k-pools 4

Submit

Step 4 Result



<https://aiidalab-qe.readthedocs.io/howto/xps.html>

How to access?

- Docker image: [aiidalab/qe:latest](https://aiidalab.org/docker/aiidalab/qe:latest), with pre-built environments, e.g., pseudopotentials, code setup.

Single User



1-50 Users



100-1000+ Users



[Documentation](#)



[GitHub](#)



- We also maintain a few official cloud-backed deployments
<https://www.aiidalab.net/deployments/>
<http://aiidalab.psi.ch/>

Code tutorial (Part 1)



Run the docker image

```
docker run -p 8888:8888 superstar54/qeapp-tutorial
```

Click **File Manager** icon

Go to `/home/jovyan/tutorial/`

Open the **tutorial** notebook, and follow the guides there.

Code tutorial (Part 2)



Run the docker image

```
docker run -p 8888:8888 superstar54/qeapp-tutorial
```

Then follow this tutorial:

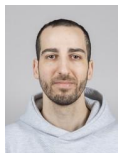
<https://aiidalab-qe.readthedocs.io/howto/xps.html>

- AiiDAlab Quantum ESPRESSO app allows researchers to arrange DFT calculations on the web browser directly without writing any code.
- QEApp is modularized and pluggable, by using plugin to extend its functionality. Developers can maintain and share their plugin easily.
- QEApp serves as an optimal platform for seamlessly incorporating the theoretical advancements while concurrently facilitating accessibility for both experimentalists and theorists within the scientific community.

Acknowledgements - AiiDA, AiiDALab, Materials Cloud



The current AiiDA, AiiDALab
and Materials Cloud teams



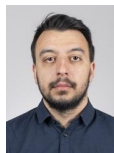
Edan
Bainglass
(PSI)



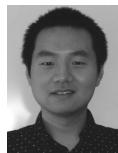
Roberto
Bendinelli
(EPFL)



Marnik
Bercx
(EPFL)



Miki
Bonacci
(PSI)



Dou
Du
(EPFL)



Kristjan
Eimre
(EPFL)



Guillaume
Fraux
(EPFL)



Valeria
Granata
(EPFL)



Daniel
Hollas
(U. Bristol)



Sebastiaan
P. Huber
(Microsoft)



François
Liot
(EPFL)



Nicola
Marzari
(EPFL, PSI)



Carlo
Pignedoli
(Empa)



Giovanni
Pizzi
(PSI)



Louis
Ponet
(EPFL)



Francisco F.
Ramirez
(EPFL)



Thomas
Schulthess
(ETHZ,CSCS)



Chris
Sewell
(EPFL)



Leopold
Talirz
(Microsoft)



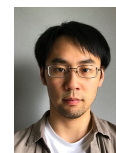
Joost
VandeVondele
(ETHZ,CSCS)



Xing
Wang
(EPFL)

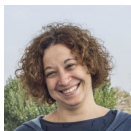


Aliaksandr
Yakutovich
(Empa)



Jusong
Yu
(EPFL)

XPS and XAS app: Collaborate with Dr. Prezzi's group
(CNR-NANO Modena), Dr. Iurii Timrov (PSI)



And all former team members, who actively contributed with
ideas and code to make these platforms what they are today.