Introduction to AiiDA

Writing reproducible workflows using AiiDA
May 21st-24th, 2019, Lausanne

MARVEL
NATIONAL CENTRE OF COMPETENCE IN RESEARCH

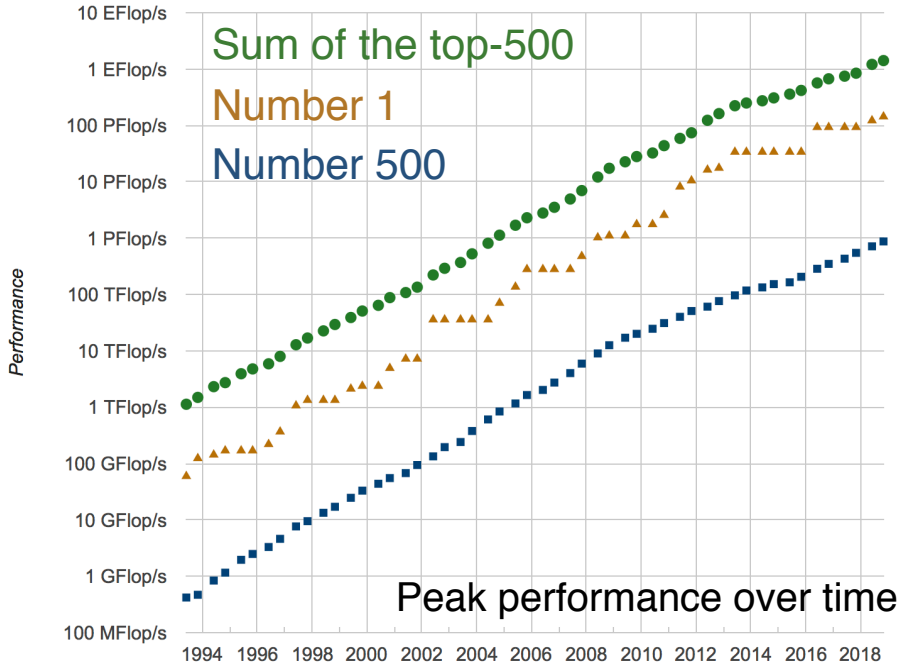
MAX DRIVING THE EXASCALE TRANSITION

INTERSECT

swissuniversities

EPFL

Accuracy and predictive power of quantum engines



150,000x increase in the past 20 years

1 month (1998)

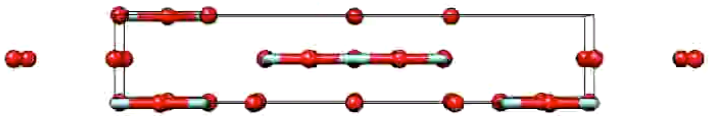
↓

10 seconds (2018)

Result: materials design and discovery via

high-throughput computations

Leverage supercomputers to compute and predict materials' properties



Aim: Compute properties for all of them
(and even new, invented ones)
and **discover novel functional materials**

Open Science Platform: definition

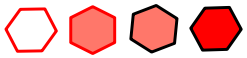
- Our definition of an Open Science Platform [1]:
  - **Open simulation codes**
  - **Open architecture** to manage simulations and **open workflows**
  - Support for **Open Data**, **Data Management Plans** and FAIR-compliant sharing
  - **Straightforward availability** of the tools, with **curated open-data services** enabling turn-key workflows (pseudopotential libraries, …)

[1] Pizzi G. (2018) Open-Science Platform for Computational Materials Science: AiiDA and the Materials Cloud.
In: Andreoni W., Yip S. (eds), Handbook of Materials Modeling (Springer, Cham).

Our goal

Build an open-science infrastructure
with computational services offered
to scientific, industrial community and beyond

Like a synchrotron, but for
open and reproducible simulations



Our two core infrastructures

AiiDA as the "operating system" to manage, automate and store simulations and their results

and

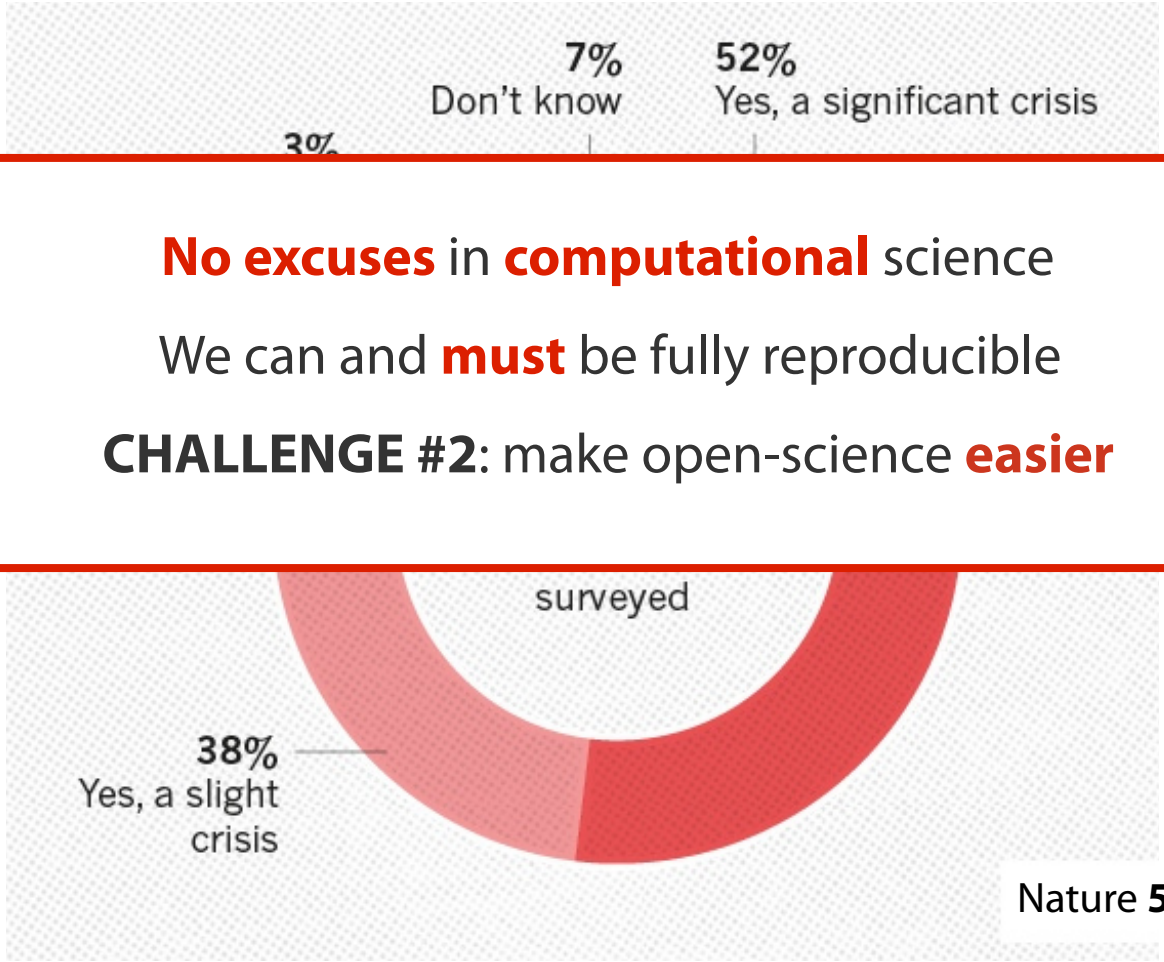
Materials Cloud as the open-science dissemination portal and cloud simulation platform

OPEN SCIENCE PLATFORM:

G. Pizzi et al., Comp. Mat. Sci. 111, 218-230 (2016)

Reproducibility: a cornerstone of the scientific method

IS THERE A REPRODUCIBILITY CRISIS?



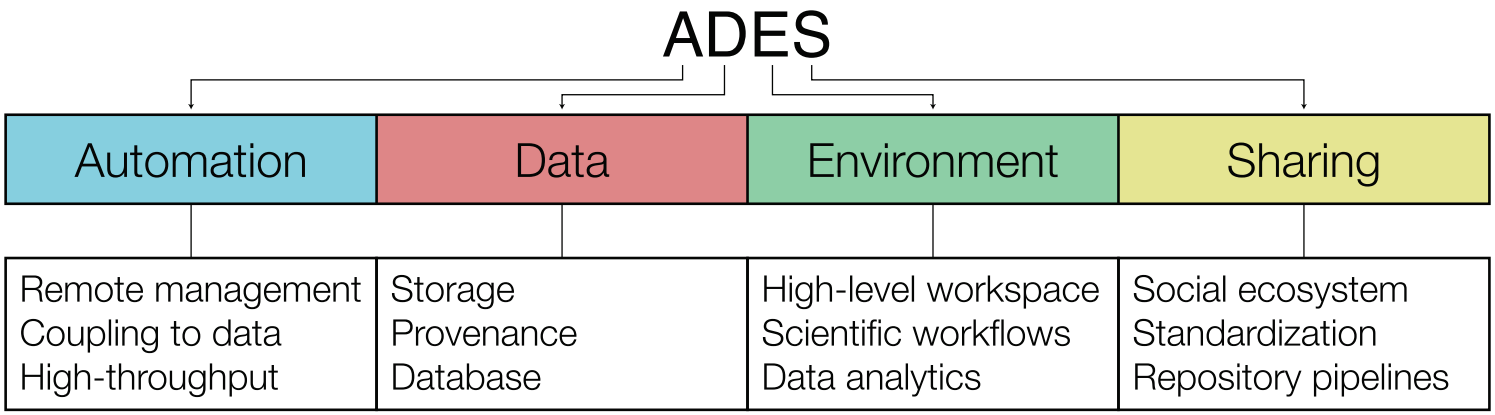
No excuses in **computational** science

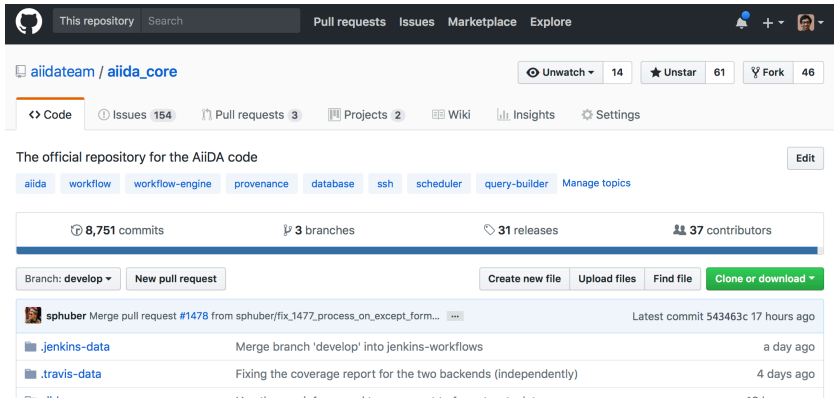
We can and **must** be fully reproducible

CHALLENGE #2: make open-science **easier**

Nature **533**, 452–454 (2016)

AiiDA development timeline

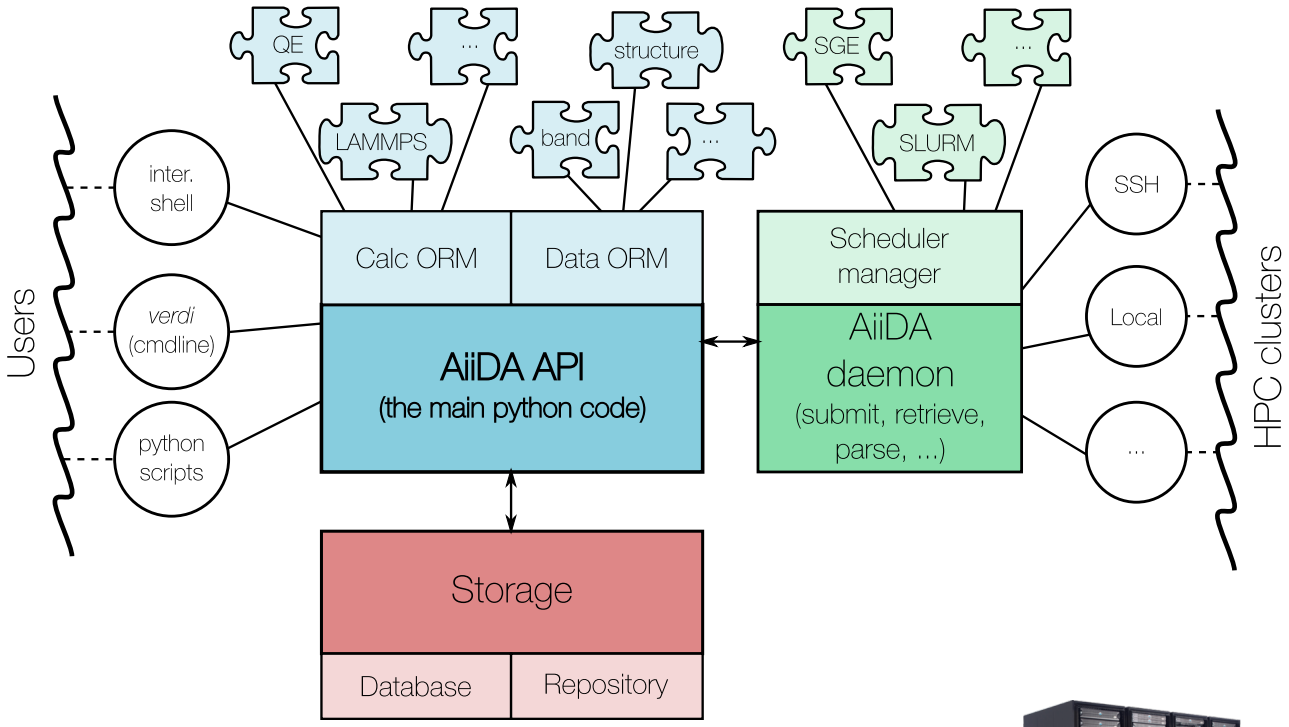




G. Pizzi et al., Comp. Mat. Sci. 111, 218-230 (2016)

http://www.aiida.net

AiiDA



AiiDA

Main features

- Python (2.7 & 3.6) infrastructure
- SQL database backend, access via a Python ORM
- Local connection to clusters, or via ssh using a python API
- Interface to various job schedulers (SGE, Torque, LSF, PBS Pro, SLURM, ...)
- Event-based daemon with remote management and workflow execution manager
- REST APIs using Flask to expose one own's data
- Plugin management system and extended code support
- Easy sharing of the results with other users in the community

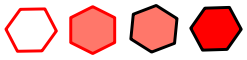
AiiDA submission

```
code = load_code('pw-6.3@daint-mr25')
builder = code.get_builder()

builder.metadata.options = {
    'max_wallclock_seconds': 600,
    'resources': {"num_machines": 2}}

Structure = DataFactory('structure')
structure = Structure(ase = read('TiO2.cif'))

Dict = DataFactory('dict')
parameters = Dict(dict={
    'CONTROL': {
        'calculation': 'scf',
        'restart_mode': 'from_scratch'},
    'SYSTEM': {'ecutwfc': 40.}})

Kpoints = DataFactory('array.kpoints')
kpoints = Kpoints(kpoints_mesh = [4,4,4])

builder.structure = structure
builder.parameters = parameters
builder.kpoints = kpoints
builder.pseudos = get_pseudos_from_family(structure,
'SSSP_efficiency_v1.0')

aiida.engine.submit(builder)
```

Switch computers in one line
supports different schedulers, version of codes, …

Define (only) necessary inputs
Interface designed by plugin

Inputs stored in the DB, and handing over to the daemon

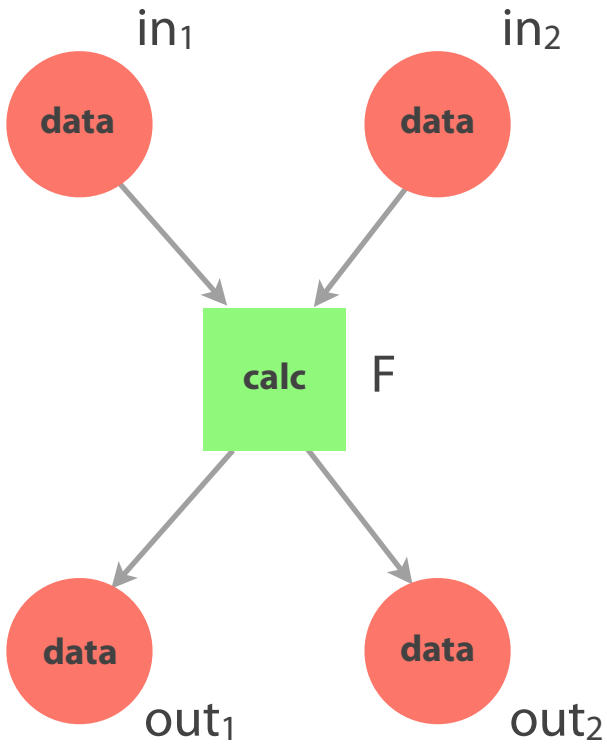


Storage and provenance

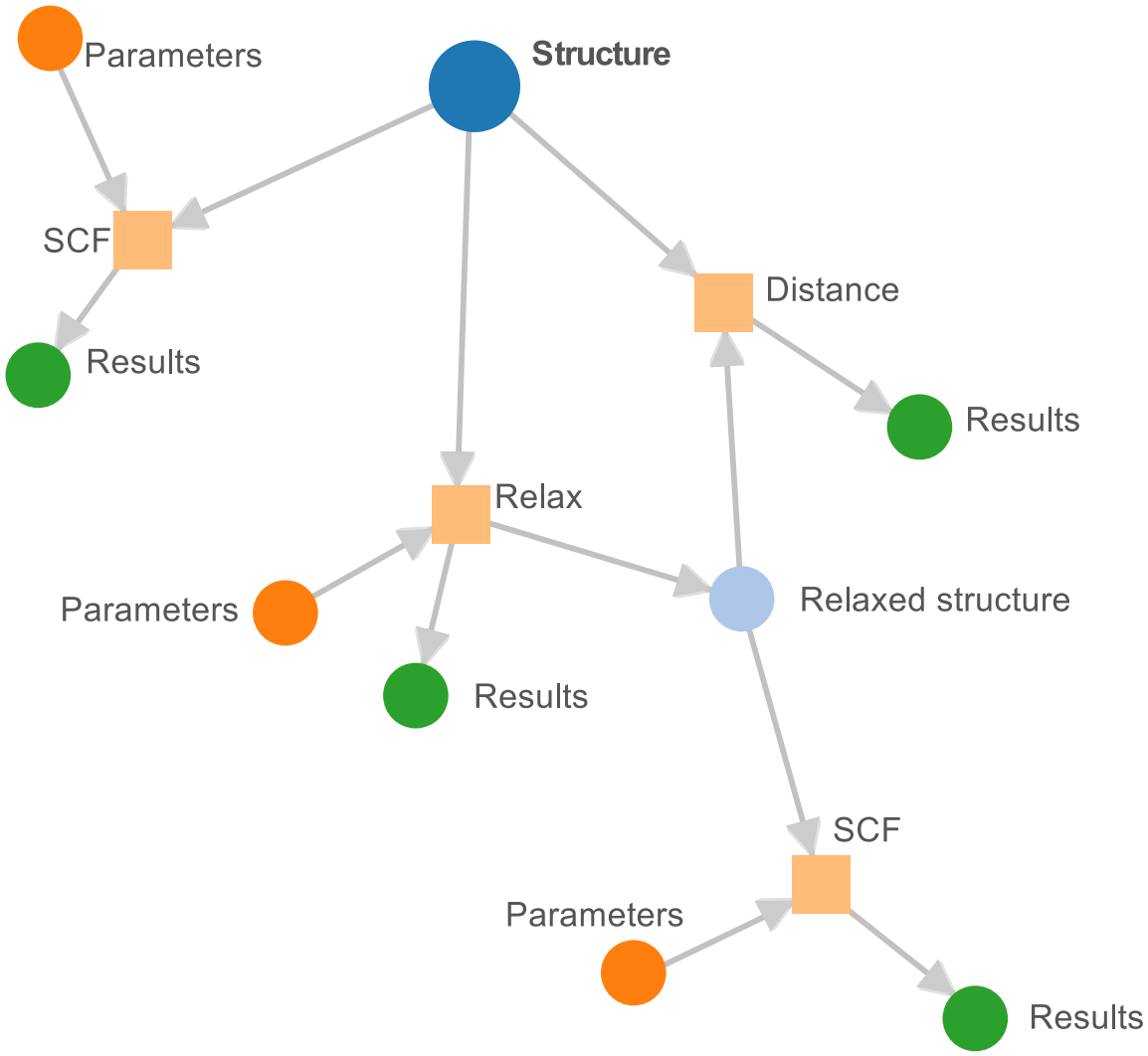
- Calculated properties: result of complex, connected calculations
- How do we store simulations **preserving the connected structure?**
- Inspiration from the open provenance model
- **Any calculation: a function,** converting inputs to outputs:

$$\mathbf{out_1, out_2 = F(in_1, in_2)}$$

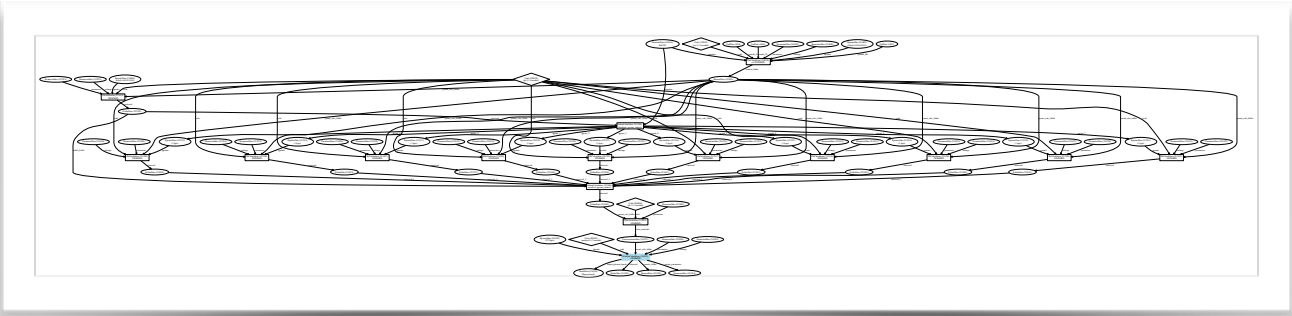
- **Each object is a node in a graph,** connected by directional labeled links
- Output nodes can be used as inputs



Data provenance: Directed Acyclic Graphs

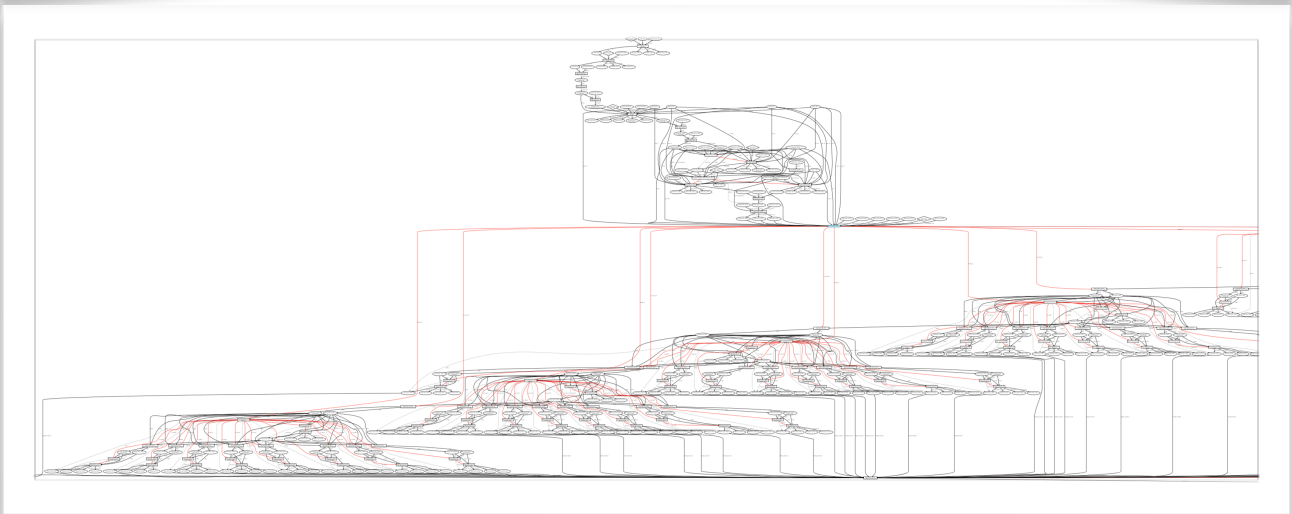


“Simple” graphs of workflows for a single material



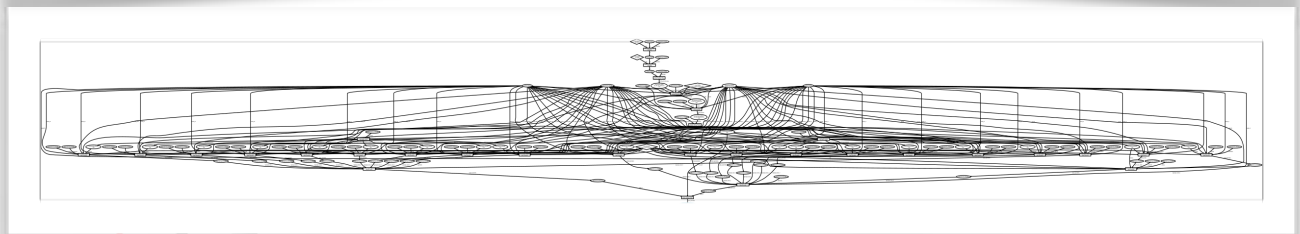
Phonon dispersion

(atom oscillations around equilibrium positions: thermal transport, electronic mobility, …)



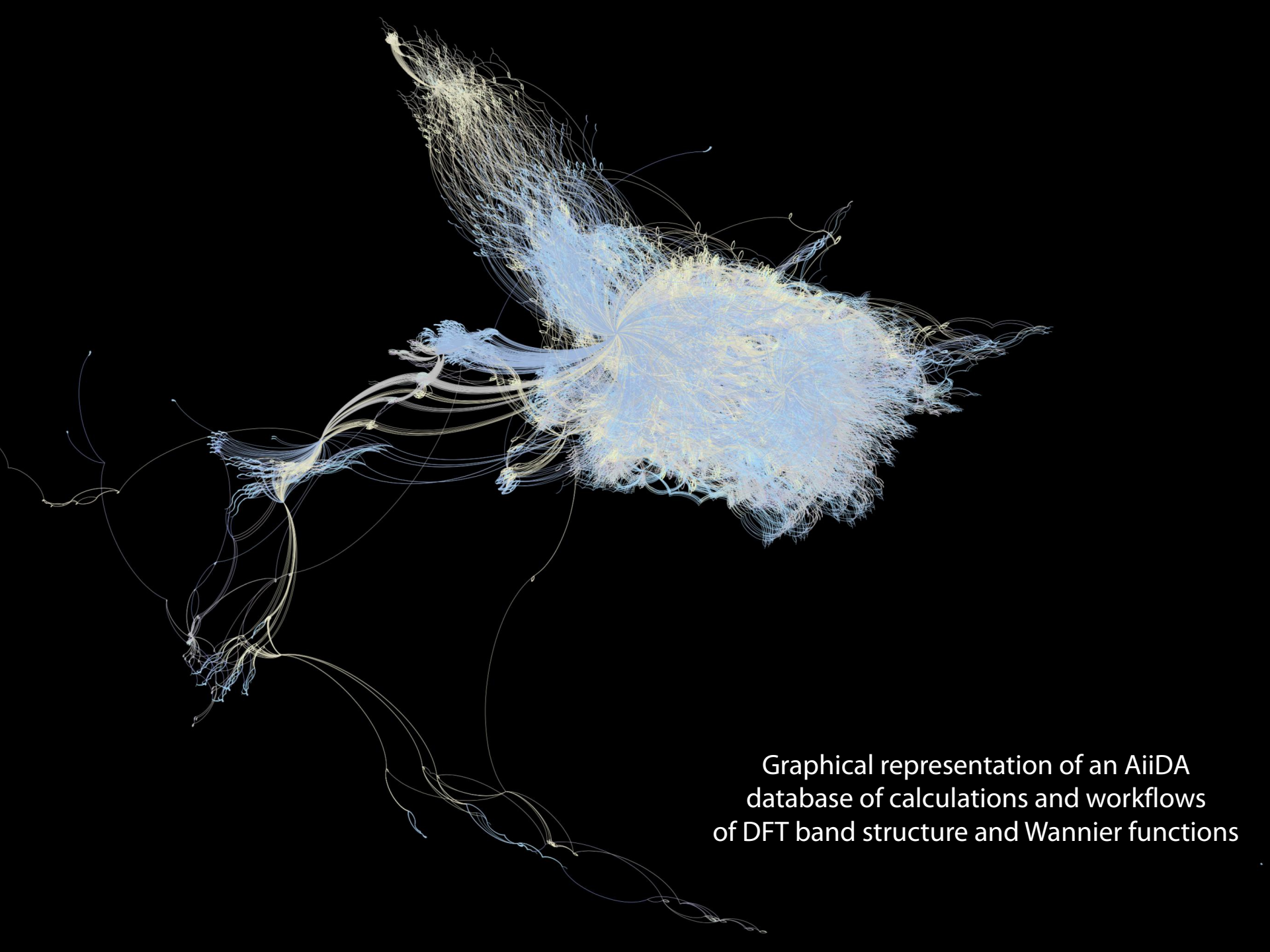
Molecular dynamics of Lithium in a solid electrolyte

(Discover novel, safe and efficient electrolytes for Li-batteries)



Elastic constants

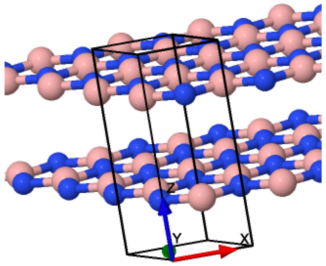
(response of materials to stresses and deformations)

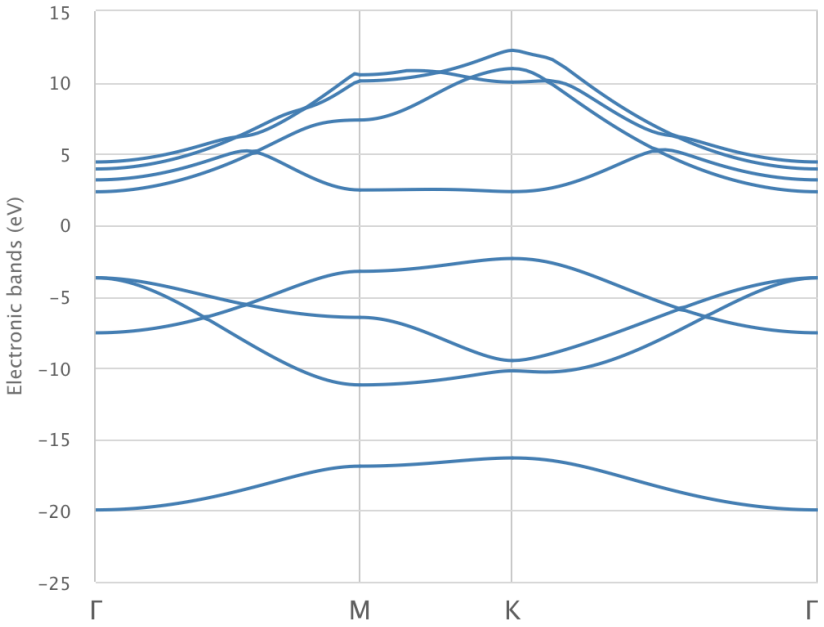


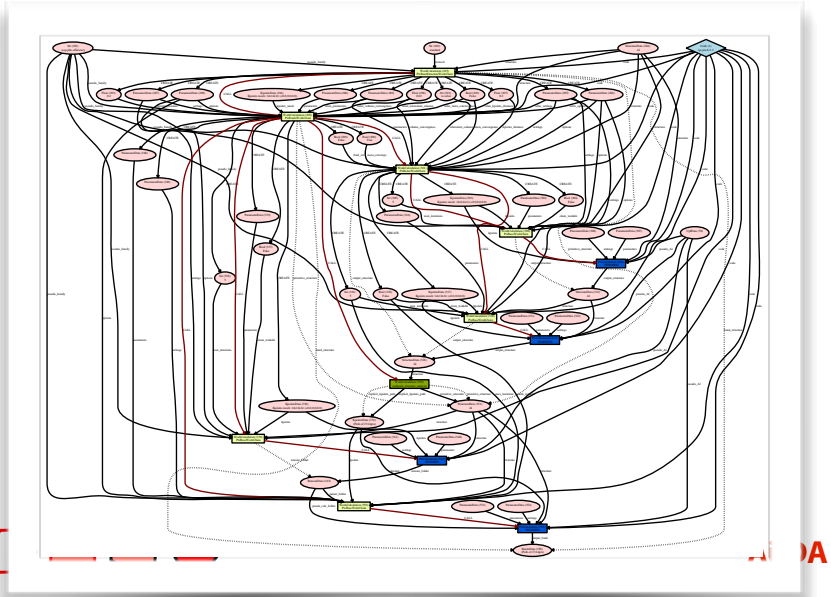
Graphical representation of an AiiDA
database of calculations and workflows
of DFT band structure and Wannier functions

Turn-key workflows in AiiDA

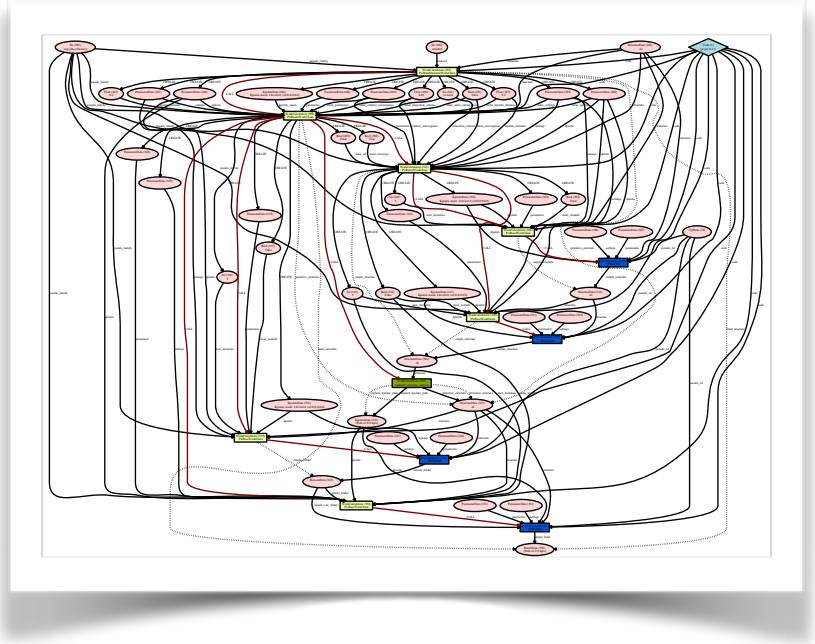
- Given a material, we often need to compute advanced quantities
- These are often non-trivial and result from a complex workflow

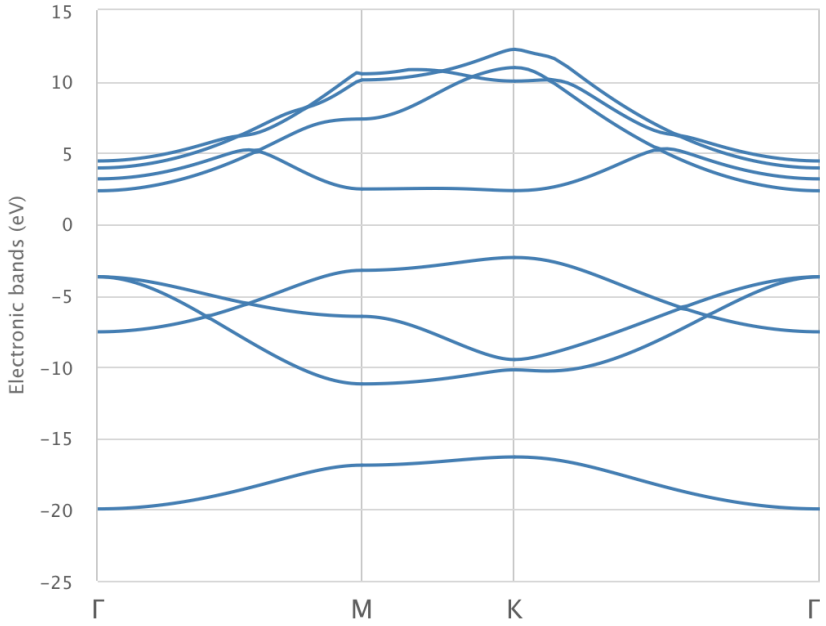






Turn-key workflows in AiiDA





- The AiiDA provenance graph allows to know how the structure was computed and to **reproduce that single specific calculation:** log of "what happened in the past"
- We need also an **easy way to re-run the same calculation again** with different parameters or for a different material: **turn-key workflows**

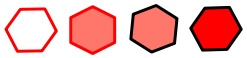
Turn-key workflows in AiiDA

```
class PwBandsWorkChain(WorkChain):
    @classmethod
    def define(cls, spec):

        spec.input('code',
                   valid_type=Code)
        spec.input('structure',
                   valid_type=StructureData)
        spec.input('pseudo_family',
                   valid_type=Str)

        spec.outline(
            cls.setup,
            cls.validate_inputs,
            if_(cls.should_do_relax)(
                cls.run_relax,
            ),
            cls.run_seekpath,
            cls.run_scf,
            cls.run_bands,
            cls.result
        )
```

- **"Operating system" for all calculations**
- Automatic provenance tracking in the DB
- Control provenance granularity
  store level of detail relevant to the workflows
- Progress checkpointing
  restart from arbitrary step, retry on failure, allows to shut down daemon and continue later
- Easy debugging, self-documenting

- Turn-key solution:

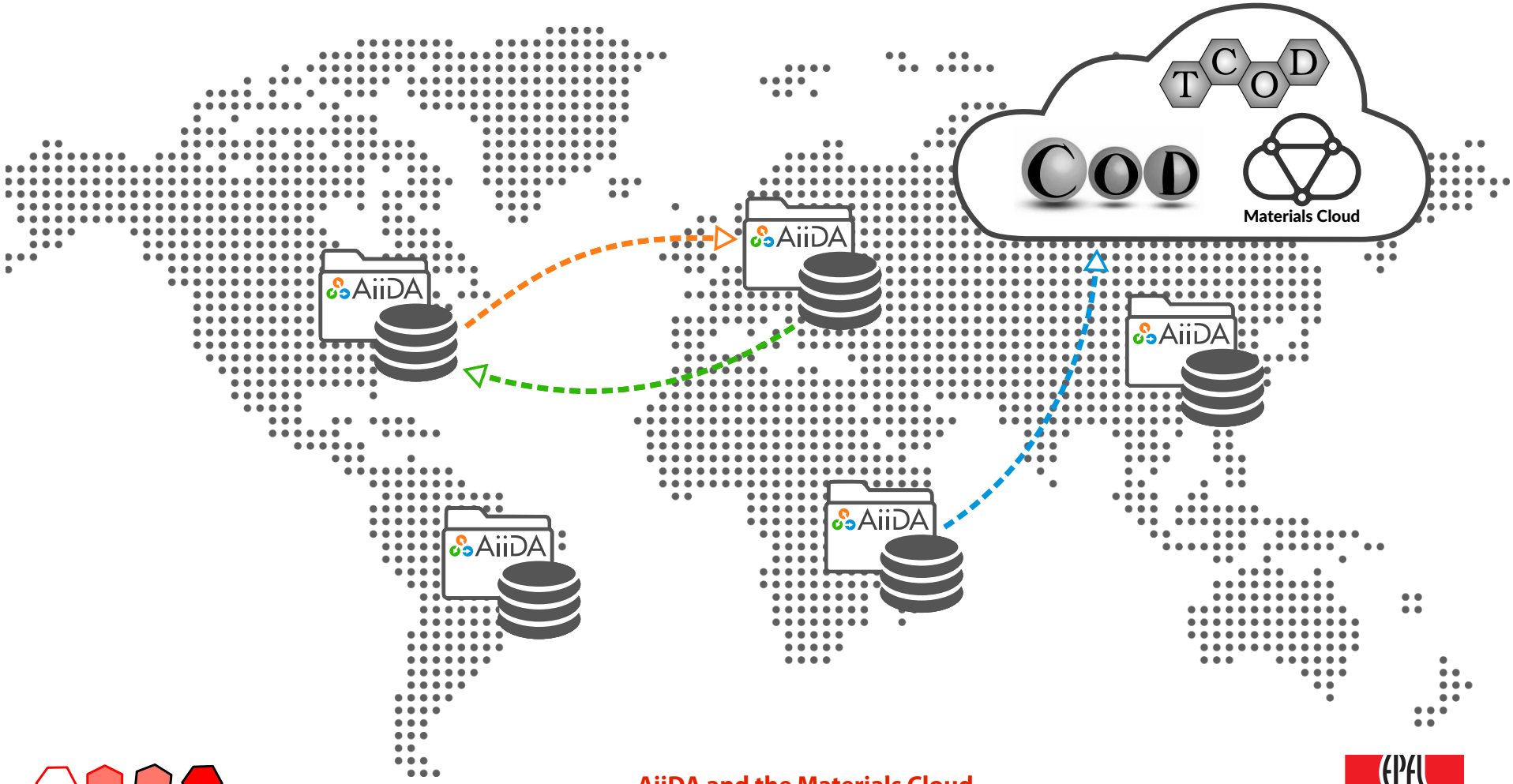
```
PwBandStructureWorkChain.run(
    code=Code.get_from_string(
        'qe-pw-6.2.1@localhost'),
    structure=StructureData(
        ase=ase.build.bulk('Al')),
     pseudo_family=Str('sssp-pbe-efficiency'))
```



Sharing in AiiDA: data and graphs

- Private AiiDA instances
- UUIDs to uniquely identify nodes

- Data can be shared to other AiiDA repositories **or to online repositories**

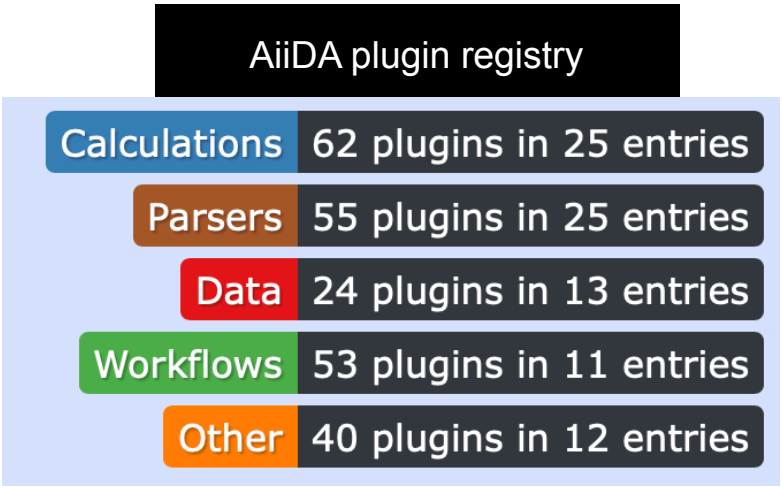


Sharing in AiiDA: codes, plugins and workflows



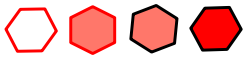



Calculation | **Data** | **Parsers** | **Transport and scheduler** | **Workflows** | **Importers & exporters**



- Plugins are collected in the AiiDA plugin registry
- Over **60 different executables from 25 different code packages currently supported,** with over 50 workflows
- Many are **community-contributed**

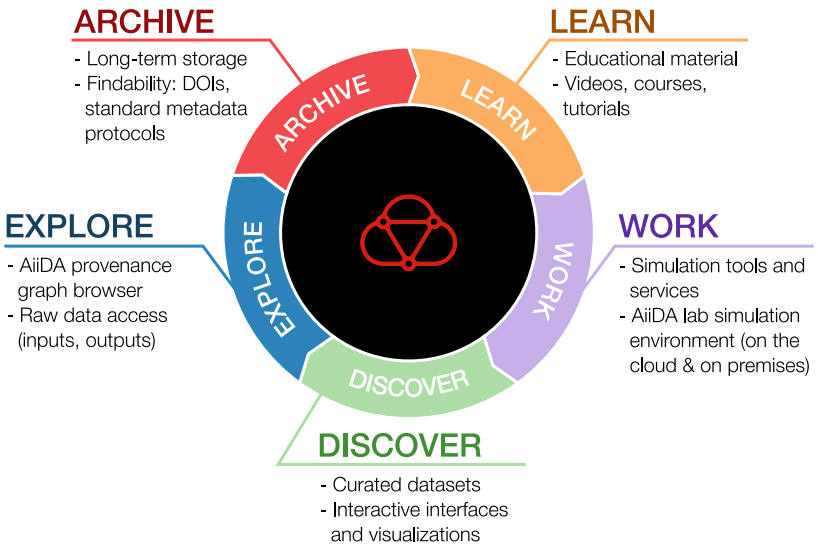
https://aiidateam.github.io/aiida-registry/



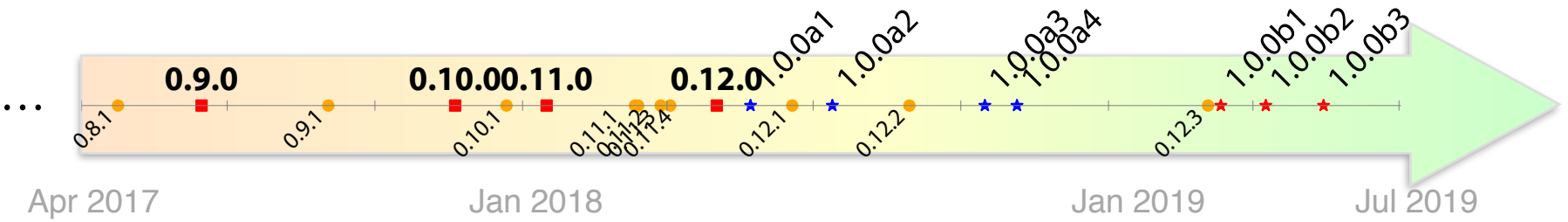
OPEN SCIENCE PLATFORM:

MATERIALSCLOUD

https://www.materialscloud.org



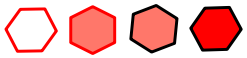
AiiDA (recent) timeline



- **AiiDA** development started from scratch in late 2013, from earlier versions from Boris Kozinsky (BOSCH) and Giovanni Pizzi and Andrea Cepellotti in EPFL
- AiiDA **0.12** released mid 2018, last major release before 1.0
  - Still supported, 0.12.3 released in March 2019
- Since 2018, over 1 year of development into **AiiDA 1.0**, the next-generation AiiDA
- You will be using **1.0.0 beta 3**, already good for production
- **v1.0 official release scheduled in the summer**
  - Roadmap towards v1.0: https://github.com/aiidateam/aiida_core/wiki/AiiDA-release-roadmap
- If you like technical details: follow project #14 on the GitHub page for the remaining v1.0 issues: https://github.com/aiidateam/aiida_core/projects/14

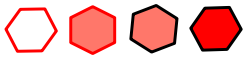
AiiDA v1.0 major improvements

- **Improved design of provenance model**
  - Formalisation of new type of links (create, return, call, input) of the type of nodes (workflows vs. calculations) and which links are allowed
- **Completely redesigned engine for high-throughput and performance**
  - Event-based, super fast, scalable
- **New daemon**
  - Multiple daemons per installation possible
  - Multiple workers per daemon
- **Significantly improved command-line interface**
  - homogenised, with tab-completion, easy to extend



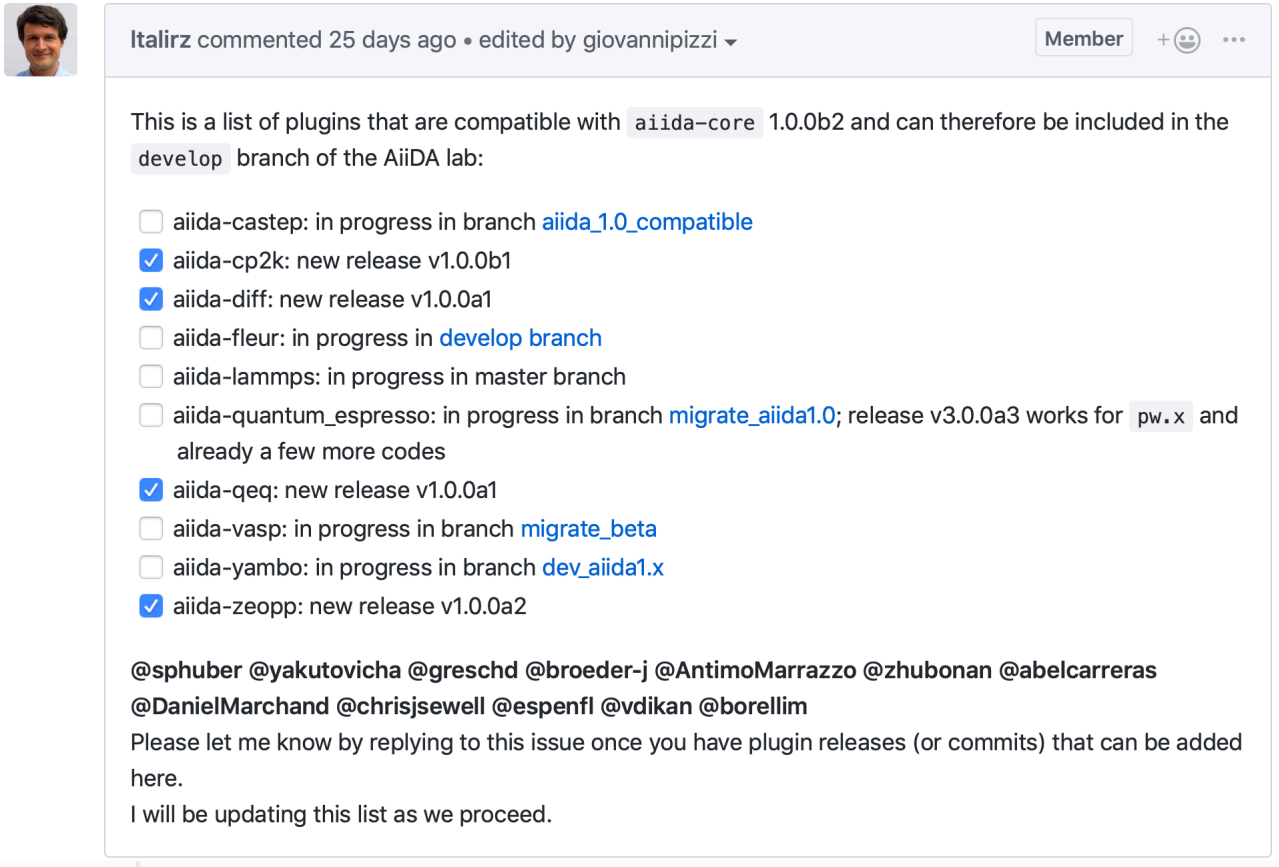
AiiDA v1.0: the time it took

- It took more than 1 year because
  - The engine has been completely redesigned to ensure **high-throughput capabilities and performance** (~50'000 processes/hour possible)
    - New engine being heavily tested, DB backend being optimised
  - We had to **redesign** the provenance model
  - We try to **minimise impact for users**
    - Change the Python API only if needed (and discussions involved)
    - Implement automatic migrations for your databases (with tests)
    - Rewrite documentation (and prepare tutorial!)
  - Full **python2 + python 3** support
  - We want to **fix all critical bugs** before v1.0 (only few remaining)
  - We want most of the **plugins** to be ready to work with AiiDA 1.0
    - 1-week workshop in March 2019 in EPFL, ~25 plugin developers



AiiDA v1.0 plugin readiness

- You can follow the state on this issue of the AiiDA lab metapackage: https://github.com/aiidalab/aiidalab-metapkg/issues/10



Italirz commented 25 days ago • edited by giovannipizzi — Member

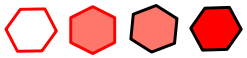
This is a list of plugins that are compatible with `aiida-core` 1.0.0b2 and can therefore be included in the `develop` branch of the AiiDA lab:

- [ ] aiida-castep: in progress in branch aiida_1.0_compatible
- [x] aiida-cp2k: new release v1.0.0b1
- [x] aiida-diff: new release v1.0.0a1
- [ ] aiida-fleur: in progress in develop branch
- [ ] aiida-lammps: in progress in master branch
- [ ] aiida-quantum_espresso: in progress in branch migrate_aiida1.0; release v3.0.0a3 works for `pw.x` and already a few more codes
- [x] aiida-qeq: new release v1.0.0a1
- [ ] aiida-vasp: in progress in branch migrate_beta
- [ ] aiida-yambo: in progress in branch dev_aiida1.x
- [x] aiida-zeopp: new release v1.0.0a2

@sphuber @yakutovicha @greschd @broeder-j @AntimoMarrazzo @zhubonan @abelcarreras @DanielMarchand @chrisjsewell @espenfl @vdikan @borellim
Please let me know by replying to this issue once you have plugin releases (or commits) that can be added here.
I will be updating this list as we proceed.

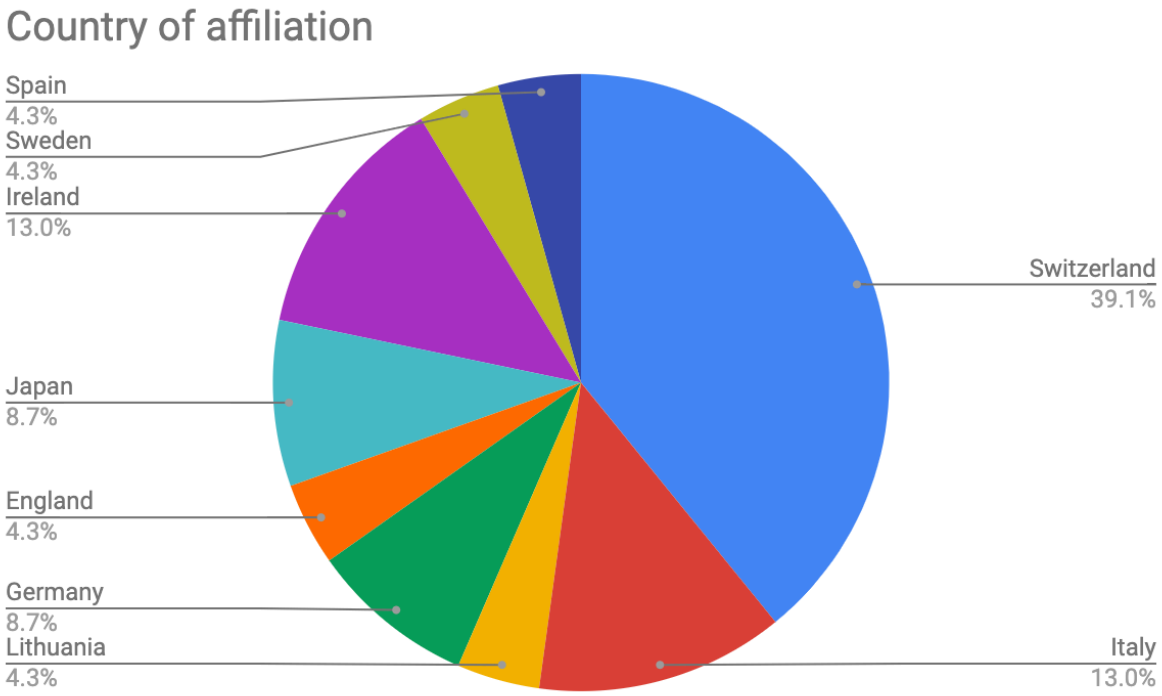
After v1.0 stabilisation

- We are now in a phase (and will stay for a few months after the release of v1.0) where we **focus on stabilising the code** rather than adding new features, unless critical
- Some of the planned features afterwards:
  - **Drop python2 support** (not earlier than end of 2019; probably early 2020)
  - Improved **file repository** management (compression and packing of files on disk, possibility of storing on object store)
  - Improve **import/export interface** (towards a push/pull mechanism rather than only export-file based)
  - …



AiiDA-powered research projects

- Questionnaire on AiiDA mailing list
  - 27 research projects
  - 25 plugins



Project status: published

Project title	DOI
Two-dimensional materials from high-throughput computational exfoliation of experimentally known compounds [1]	10.1038/s41565-017-0035-5
Arsenene MOSFET [2]	10.1038/ncomms12585
Polar discontinuities [3]	10.1038/ncomms6157
Ideal adhesive and shear strengths of solid interfaces: A high throughput ab initio approach [4]	10.1016/j.commatsci.2018.08.006
Storing reproducible workflows in the Theoretical Crystallography Open Database [5]	10.1186/s13321-017-0242-y
Ab-initio studies for battery materials design [6]	10.1103/PhysRevMaterials.2.103805
SSSP Pseudopotential libraries [7]	10.1038/s41524-018-0127-2
Electronic and optical properties of doped TiO_2 by many-body perturbation theory [8]	10.1103/PhysRevMaterials.3.045401

AiiDA plugins

[1] N. Mounet et al., Nat. Nanotech. 13, 246 (2018)
[2] G. Pizzi et al., Nat Comm. 7, 12585 (2016)
[3] M. Gibertini et al., Nat. Comm. 5, 5157 (2014)
[4] P. Restuccia et al., Comp. Mat. Sci. 154, 517 (2018)
[5] A. Merkys et al., J. Cheminf. 9, 56 (2017)
[6] D. Gresch et al., Phys. Rev. Materials 2, 103805 (2018)
[7] G. Prandini et al., npj Comp. Mat. 4, 72 (2018)
[8] M. O. Atambo et al., Phys. Rev. Mat. 3, 045401 (2019)

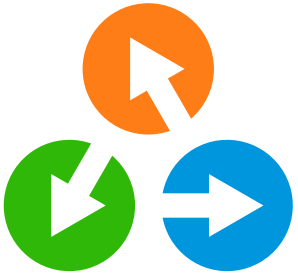
Project status: simulations completed

Project title	Contact
A high-throughput search for new solid-state electrolytes for Li-ion batteries	Leonid Kahle, EPFL
Applicability of tail-corrections in the molecular simulations of porous materials	Kevin Jablonka, EPFL
Determining interface structures between fluorite and pervoskite materials	Bonan Zhu, University of Cambridge
Lattice thermal conductivities of materials	Yue-Wen Fang, Kyoto University
Siesta LDA pseudo test	Emanuele Bosoni, Trinity College Dublin
Gas adsorption in covalent organic frameworks	Daniele Ongari, EPFL
Ab-initio studies for battery materials design	Andreas Stamminger, Bosch

Project status: underway

Project title	Contact
Transferability of Machine Learning Models for Complex Properties of Porous Materials	Kevin Jablonka, EPFL
Systematic study of the Atomic Self Interaction Correction	Emanuele Bosoni, Trinity College Dublin
High-Throughput simulations of Complex Band Structures	Emanuele Bosoni, Trinity College Dublin
3DD	Sebastiaan Huber, EPFL
High-throughput Investigation of the Effect of Impurities on the Transport Properties of Topological Insulators	Philipp Rüssmann FZ Jülich
Graphene adhesion on surfaces: a van der Waals density functional study	Karim Elgammal, KTH
Simulation of Water Isotherms	Kevin Jablonka, EPFL
Flat-land nanoelectronics	Víctor García, University of Oviedo
High throughput phonon calculation	Togo Atsushi, Kyoto University
GW convergence workflows	Miki Bonacci, Unimore

Contacts



Website: http://www.aiida.net

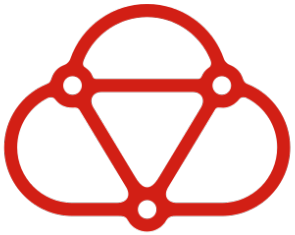
Docs: http://aiida-core.readthedocs.io

Git repo: https://github.com/aiidateam/aiida_core/

Plugin registry: http://aiidateam.github.io/aiida-registry

https://www.facebook.com/aiidateam

@aiidateam



Materials Cloud: http://www.materialscloud.org

- **AiiDA lab**: http://aiidalab.materialscloud.org
- **Archive**: http://archive.materialscloud.org

Quantum Mobile: http://www.materialscloud.org/work/quantum-mobile