



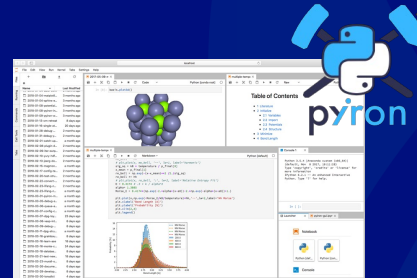
pyiron_base – pyiron without atomistics

Jan Janssen

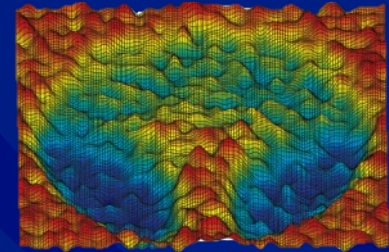
07/27/2022



Simulation Code



pyiron_base



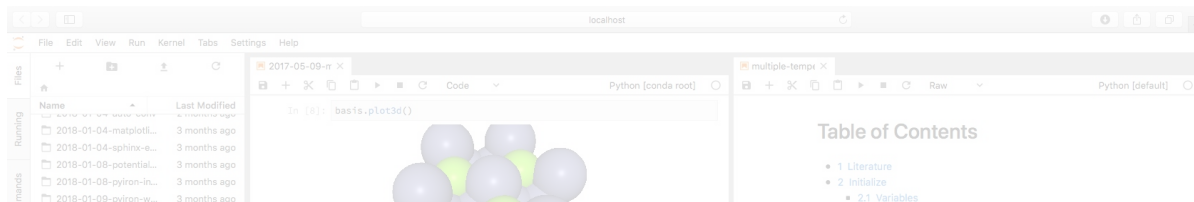
Parameter Study

pyiron: from users to contributors

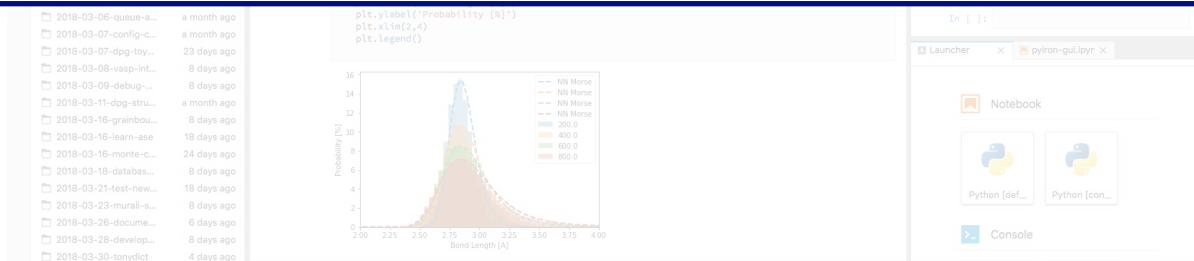
The screenshot displays the Pyiron web interface, which is a JupyterLab-like environment. On the left, a sidebar shows a file browser with a list of files and folders, including '2017-05-09-n', '2018-01-04-matplot...', '2018-01-04-sphinx-e...', '2018-01-08-potential...', '2018-01-08-pyiron-in...', '2018-01-08-pyiron-w...', '2018-01-13-cm-retreat', '2018-01-16-single-at...', '2018-01-28-debug-y...', '2018-01-31-debug-y...', '2018-02-01-catch-up...', '2018-02-08-plugin-d...', '2018-02-09-iter-outp...', '2018-02-10-yury-hdf...', '2018-02-13-joerg-do...', '2018-02-15-magmon...', '2018-02-17-config-te...', '2018-02-20-test-stru...', '2018-02-22-convert-', '2018-02-23-lifang-n...', '2018-02-23-lifang-n...', '2018-03-01-pyiron-m...', '2018-03-05-debug-s...', '2018-03-06-queue-a...', '2018-03-07-config-c...', '2018-03-07-dpg-toy...', '2018-03-08-vasp-int...', '2018-03-09-debug-', '2018-03-11-dpg-stru...', '2018-03-16-grainbou...', '2018-03-16-learn-ase', '2018-03-16-monte-c...', '2018-03-18-databas...', '2018-03-21-test-new...', '2018-03-23-murali-s...', '2018-03-26-docume...', '2018-03-28-develop...', and '2018-03-30-tonydidt'.

The main area is divided into several panes. The top-left pane shows a 3D molecular model of a cluster of atoms. The top-right pane displays a 'Table of Contents' with a list of links: 1 Literature, 2 Initialize, 2.1 Variables, 2.2 Import, 2.3 Potentials, 2.4 Structure, 3 Minimize, and 4 Bond Length. The bottom-left pane contains a code editor with Python code for plotting a harmonic potential and a relative entropy fit. The bottom-right pane shows a plot of Probability (%) versus Bond Length (Å) for different temperatures (200.0, 400.0, 600.0, 800.0) and Morse potential parameters (NN Morse, NN Morse, NN Morse).

pyiron: from users to contributors



pyiron is designed to simplify the development of simulation protocols.
Disclaimer: An easy user interface can result in a complex backend.



Implement New Classes

```
In [1]: from os.path import join
        from pyiron_base import TemplateJob, Project
```

```
In [2]: class ToyJob(TemplateJob):
        def __init__(self, project, job_name):
            super().__init__(project, job_name)
            self.input['input_energy'] = 100
            self.executable = "cat input > output"

        def write_input(self):
            self.input.write_file(
                file_name="input",
                cwd=self.working_directory
            )

        def collect_output(self):
            file = join(self.working_directory, "output")
            with open(file) as f:
                line = f.readlines()[0]
            energy = float(line.split()[1])
            with self.project_hdf5.open("output/generic") as h5out:
                h5out["energy_tot"] = energy
```

Implement New Classes

```
In [1]: from os.path import join
        from pyiron_base import TemplateJob, Project
```

```
In [2]: class ToyJob(TemplateJob):
        def __init__(self, project, job_name):
            super().__init__(project, job_name)
            self.input['input_energy'] = 100
            self.executable = "cat input > output"

        def write_input(self):
            self.input.write_file(
                file_name="input",
                cwd=self.working_directory
            )

        def collect_output(self):
            file = join(self.working_directory, "output")
            with open(file) as f:
                line = f.readlines()[0]
            energy = float(line.split()[1])
            with self.project_hdf5.open("output/generic") as h5out:
                h5out["energy_tot"] = energy
```

write_input() :
input: python dictionary
output: write Input files

collect_output() :
input: directory of output
output: python dictionary

Implement New Classes

```
In [1]: from os.path import join
        from pyiron_base import TemplateJob, Project
```

```
In [2]: class ToyJob(TemplateJob):
        def __init__(self, project, job name): write_input():
```

In the most basic case only a `write_input()` function and a `collect_output()` function are required to leverage pyiron.

```
def collect_output(self):
    file = join(self.working_directory, "output")
    with open(file) as f:
        line = f.readlines()[0]
    energy = float(line.split()[1])
    with self.project_hdf5.open("output/generic") as h5out:
        h5out["energy_tot"] = energy
```

input: directory of output
output: python dictionary

Opensource Development

The screenshot shows the GitHub repository for **pyiron/pyiron**, which is a public repository. The repository has 13 forks and 215 stars. The main navigation bar includes links for Code, Issues (5), Pull requests, Discussions, Actions, Security (1), and Insights. The repository is currently on the **master** branch. Below the navigation bar, there is a list of recent pull requests, including a merge request by **jan-janssen** titled "Merge pull request #1424 from pyi..." which was merged yesterday and has 7,246 commits. The repository also has a list of files, including **.ci_support**, **.devcontainer**, **.github**, **binder**, and **docs**, each with a description of the file and the date it was last updated. The **About** section describes **pyiron** as an integrated development environment (IDE) for computational materials science. The **pyiron.org** website is also linked, and a list of tags (python, dft, simulation, ide, molecular-dynamics, vasp, hdf5, lammps, development-environment, ab-initio, ase, pyiron) is displayed.

Search or jump to... **Pulls** **Issues** **Marketplace** **Explore**

pyiron / pyiron Public

Edit Pins Unwatch 13 Fork 34 Star 215

Code Issues 5 Pull requests Discussions Actions Security 1 Insights

master Go to file Add file Code

About

pyiron - an integrated development environment (IDE) for computational materials science.

pyiron.org

python dft simulation ide molecular-dynamics vasp hdf5 lammps development-environment ab-initio ase pyiron

jan-janssen Merge pull request #1424 from pyi... yesterday 7,246

File	Description	Last Updated
.ci_support	[dependabot skip] Update environment	yesterday
.devcontainer	Update devcontainer.json	2 years ago
.github	Update deploy.yml	3 months ago
binder	Update postBuild	2 months ago
docs	Document module purge recommendati...	9 months ago



Opensource Development



Growing number of collaborations and contributors based on publishing pyiron as opensource software.

.ci_support	[dependabot skip] Update environment	yesterday
.devcontainer	Update devcontainer.json	2 years ago
.github	Update deploy.yml	3 months ago
binder	Update postBuild	2 months ago
docs	Document module purge recommendati...	9 months ago

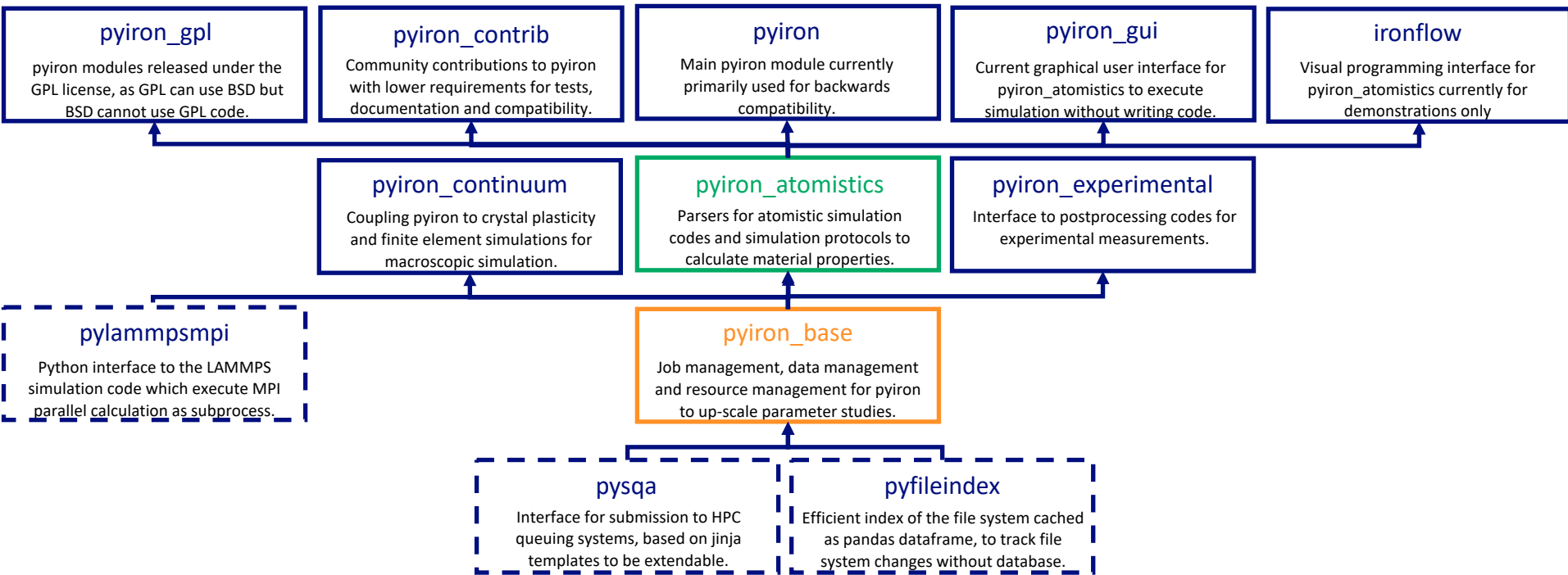
pyiron.org

python dft simulation ide
molecular-dynamics vasp hdf5
lammps development-environment
ab-initio ase pyiron

CENTER FOR
MOLECULAR MODELING



pyiron Projects



pyiron Projects

pyiron_gpl

pyiron modules released under the GPL license, as GPL can use BSD but BSD cannot use GPL code.

pyiron_contrib

Community contributions to pyiron with lower requirements for tests, documentation and compatibility.

pyiron

Main pyiron module currently primarily used for backwards compatibility.

pyiron_gui

Current graphical user interface for pyiron_atomistics to execute simulation without writing code.

ironflow

Visual programming interface for pyiron_atomistics currently for demonstrations only

Focus on pyiron_base, the job management, data management and resource management for pyiron to up-scale parameter studies.

Python interface to the LAMMPS simulation code which execute MPI parallel calculation as subprocess.

Job management, data management and resource management for pyiron to up-scale parameter studies.

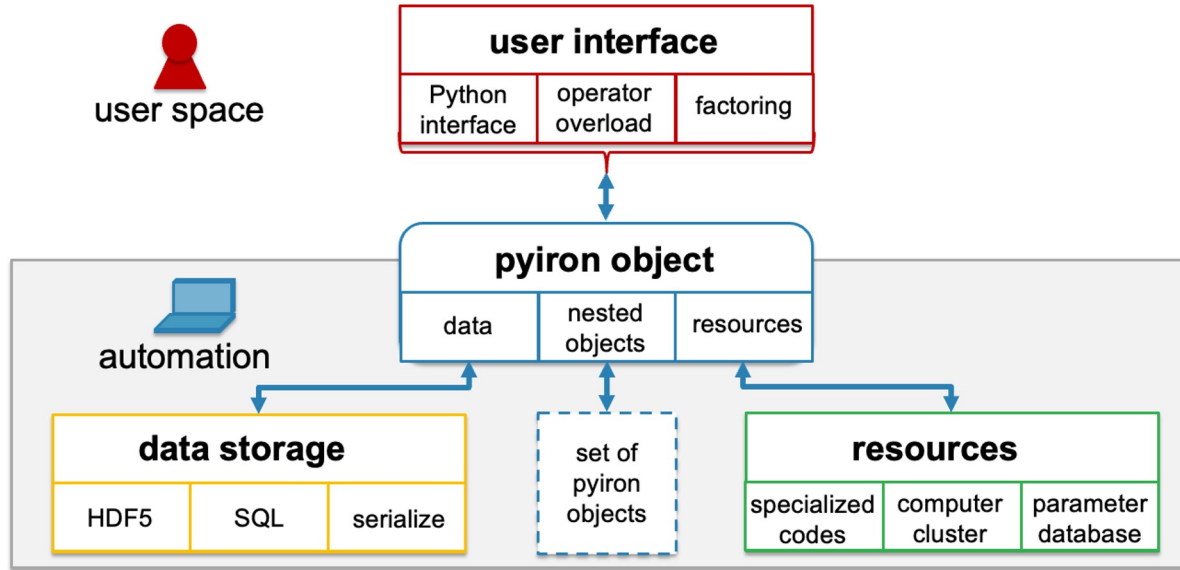
pysqa

Interface for submission to HPC queuing systems, based on jinja templates to be extendable.

pyfileindex

Efficient index of the file system cached as pandas dataframe, to track file system changes without database.

pyiron Objects



pyiron Objects



Abstract the technical complexity
to focus on the implementation of the scientific simulation protocol.

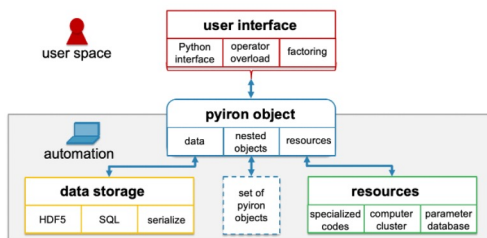


Source Code

Summary

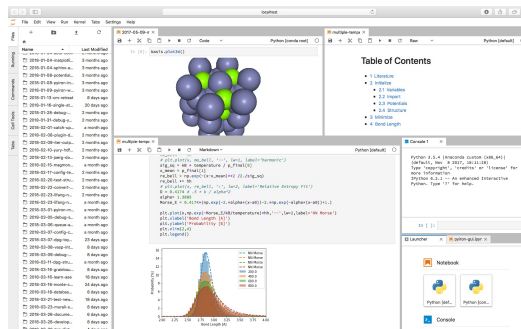
Making the Materials Science Accessible

pyiron objects



Separate the technical and scientific complexity.

pyiron IDE



Develop simulation protocols inside jupyter notebooks.

Application

```
from os.path import join
from pyiron_base import TemplateJob, Project

class ToyJob(TemplateJob):
    def __init__(self, project, job_name):
        super().__init__(project, job_name)
        self.input['input_energy'] = 100
        self.executable = "cat input > output"

    def write_input(self):
        self.input.write_file(
            file_name="input",
            cwd=self.working_directory
        )

    def collect_output(self):
        file = join(self.working_directory, "output")
        with open(file) as f:
            line = f.readlines()[0]
            energy = float(line.split()[1])
            with self.project_hdf5.open("output/generic") as h5out:
                h5out["energy_tot"] = energy
```

Discover previously unknown relations from parameter studies.

Collaboration

Max Planck Institut für Eisenforschung



Jörg Neugebauer



Tilmann Hickel



Sudarsan
Surendralal



Osamu Waseda



Liam Huber



Marvin Poul



Muhammad
Hassani

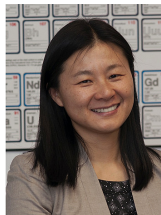


Niklas Siemer

Los Alamos National Laboratory



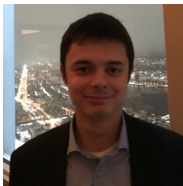
Danny Perez



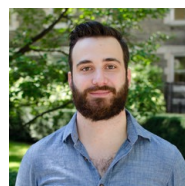
Ping Yang



Michael Taylor



Daniel Burrill



Michael Tynes

Collaborators



Alexander Shapeev

Yury Lysogorskiy

Shoji Ishibashi

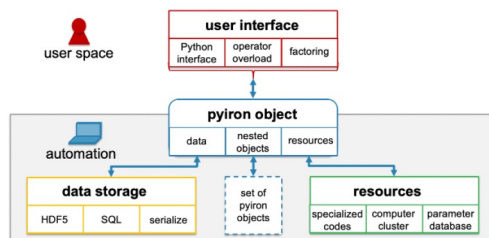
...

Blazej Grabowski

Summary – Thank you

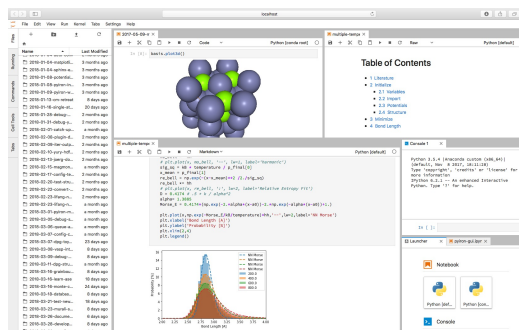
Making the Materials Science Accessible

pyiron objects



Separate the technical and scientific complexity.

pyiron IDE



Develop simulation protocols inside jupyter notebooks.

Application

```
from os.path import join
from pyiron_base import TemplateJob, Project

class ToyJob(TemplateJob):
    def __init__(self, project, job_name):
        super().__init__(project, job_name)
        self.input['input_energy'] = 100
        self.executable = "cat input > output"

    def write_input(self):
        self.input.write_file(
            file_name="input",
            cwd=self.working_directory
        )

    def collect_output(self):
        file = join(self.working_directory, "output")
        with open(file) as f:
            line = f.readlines()[0]
            energy = float(line.split()[1])
            with self.project_hdf5.open("output/generic") as h5out:
                h5out["energy_tot"] = energy
```

Discover previously unknown relations from parameter studies.