Reproducible Computational Scientific Workflows with **signac**

Bradley D. Dice[1], Carl S. Adorf[2], Vyas Ramasubramani[2], Paul M. Dodd[2], Sharon C. Glotzer[234]

1. Department of Physics 2. Department of Chemical Engineering 3. Department of Materials Science and Engineering
4. Biointerfaces Institute, University of Michigan, Ann Arbor, MI 48109

Summary

Large-scale computational studies in physics, chemistry, and materials science are not only scientifically challenging, but also require the management of complex data spaces.

The **signac** framework **[1]**:

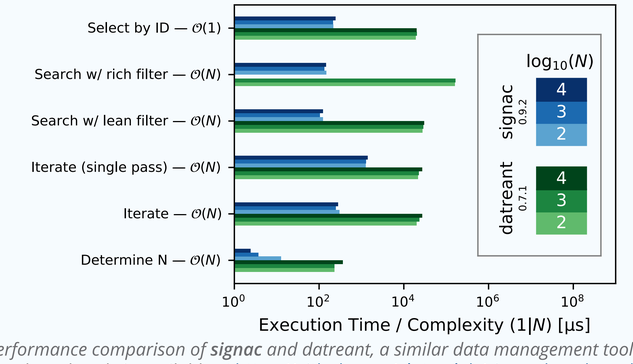
- provides the infrastructure for the rapid development and execution of computational investigations
- integrates well with high-performance computing cluster environments
- simplifies collaboration on shared data spaces
- is available for Python 2.7 and 3.4+ through pip and conda
- is free and open-source (BSD 3-Clause License)

*The pointillistic painting style, pioneered by G. Seurat and P. Signac serves as a metaphor for **signac**'s data model. Illustration based on* Cassis, Cap Lombard *1889, Gemeentemuseum Den Haag.*

The **signac** framework (composed of the core **signac** package, **signac-flow**, and **signac-dashboard**) provides tools to develop complex workflows operating on research data spaces and rapidly visualize the results, enabling the simple, efficient, and reproducible execution of computational studies.

Performant Data Management

At its core, **signac** is a database built directly on top of the file system, leveraging the advantages of direct file system access while also providing functions to efficiently index and search the data space. The user provides "state point" parameters and associated data, while **signac** is responsible for managing the storage of both parameters and data. A JSON document stores metadata, while HDF5 files store numerical arrays. Data files are stored directly in the workspace.



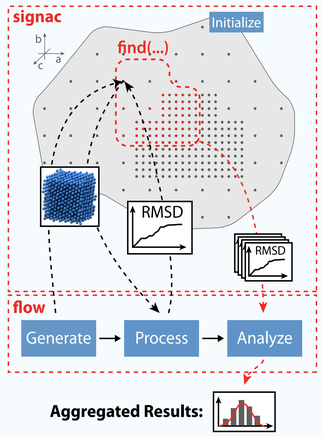
*Performance comparison of **signac** and datreant, a similar data management tool. Benchmark code is available at* https://github.com/glotzerlab/signac-benchmarks

The high performance of **signac** on network file systems is integral to its usability on HPC architectures. The core **signac** application scales well for data spaces exceeding 10^4 jobs **[1]**.

Efficient HPC Workflows

The **signac-flow** package streamlines the execution of user-defined operations on the data space. It automates HPC job submission on PBS, Torque, SLURM, and LSF managed clusters, and provides progress reports to the user.

There are three elements of **signac-flow**: *jobs*, each of which represents the data associated with a single parameter combination; *operations*, which are procedures acting on jobs; and *FlowProjects*, which are collections of operations encapsulating a complete workflow associated with a **signac** data space. The code below shows a sample of job initialization:

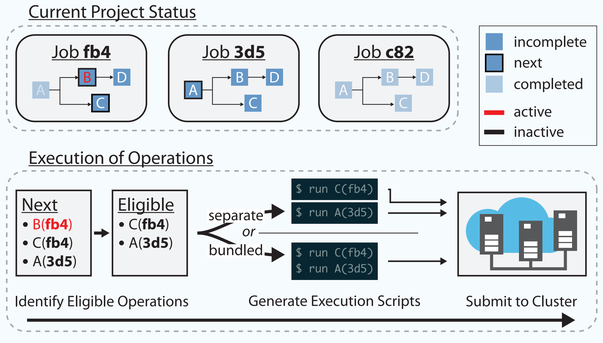


*A typical workflow: jobs are initialized using **signac**, and then operations to generate, process, and analyze data are submitted to an HPC cluster by **signac-flow**.*

```python
import signac
project = signac.init_project('IdealGasProject')

# Iterate over the variable of interest:
for p in 0.1, 1.0, 10.0:
    # Obtain a handle for the full state point:
    job = project.open_job({'p': p, 'kT': 1.0, 'N': 1000})
    # Store the volume in the job's document data:
    job.doc.volume = job.sp.N * job.sp.kT / job.sp.p
    # Or write it to a file within the job's workspace:
    with open(job.fn('volume.txt'), 'w') as file:
        file.write(str(job.sp.N * job.sp.kT / job.sp.p))
```

Cluster Job Submission



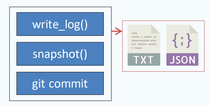
Job submission is managed by the **signac-flow** *FlowProject*, which uses a series of pre-conditions and post-conditions to determine and submit eligible operations to the HPC for processing. Operations are Python functions or shell commands; **signac-flow** is generally agnostic to the applications or scripting languages used to generate and process data. Operations may be bundled and run in parallel.

Reproducibility & Collaboration

Within the **signac** framework, several tools assist users with creating reproducible workflows. By supporting a wide range of systems and keeping workspace definitions and workflow logic alongside operations' source code, **signac** ensures the portability needed to reproduce results and the simplicity needed for collaboration.

(a) Data Import/Export

(b) Execution Hooks*



(c) Containerized Workflow Support

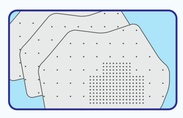
(a) A data space exported from **signac** can be uploaded to data repositories such as Zenodo or Figshare **[2, 3]**. **(b)** Execution hooks can trigger before/after operations are called, providing a traceable log. **(c) signac-flow** supports running operations in containers.

(d) Schema Detection

```
'T': int([298, 300, 302])
'P': float([0.1, 1.0])
```

(e) Dashboard Visualization

(f) Workspace Views



(d) Heterogeneous data spaces can be quickly summarized via their detected schema. **(e)** The **signac-dashboard** package enables rapid data visualization. **(f)** Generate views of the workspace for simplified filesystem access.

* upcoming release

Data Selection and Aggregation

Searching, grouping, and filtering jobs can be done in Python, bash, or the **signac-dashboard** web interface.

```python
# Filter jobs by state point metadata (in Python)
for job in project.find_jobs({'p': 0.1}):
    print(job)
# Filter jobs by document data
for job in project.find_jobs(doc_filter={'volume': 100.0}):
    print(job)
# Group jobs by state point parameter 'p'
for p, group in project.groupby('p'):
    print(p, list(group))
```

```
# Or use the simple filter syntax in Bash:
$ signac find p 0.1
```

Acknowledgements

Development and deployment supported by MICCoM, as part of the Computational Materials Sciences Program funded by the U.S. Department of Energy, Office of Science, Basic Energy Sciences, Materials Sciences and Engineering Division, under Subcontract No. 6F-30844. Project conceptualization and implementation supported by the National Science Foundation, Award # DMR 1409620. Bradley Dice acknowledges support from the National Science Foundation Graduate Research Fellowship under Grant No. 1256260 DGE.

[1] Carl S. Adorf, Paul M. Dodd, Vyas Ramasubramani, Sharon C. Glotzer, Simple data and workflow management with the **signac** framework, Computational Materials Science, Volume 146, 2018, 220-229. https://doi.org/10.1016/j.commatsci.2018.01.035
[2] https://zenodo.org
[3] https://figshare.com

More Information

For more information, including the full documentation, please visit:
https://signac.io

Install **signac** with pip or conda:

```
pip install signac
conda install signac -c conda-forge
```