

10 Dec 2024

- Review of proposed syntax changes ([pydra#766](#))
- Roadmap to version 1
- Help with how the `pydra.engine.state.State` class works
- Publishing Pydra?

23 Apr 2024

- Explicit paths being copied and rewritten to workflow cache directory
 - Seems to be a bug
 - Linking/copying one node's cache directory outputs to another's seems fine, but `Path` should be respected
- Pydra support
 - Satra's group is the only one with funding for Pydra explicitly
 - Nipreps aiming to write new tools with Pydra tasks and `nipype1` interfaces, to avoid continuing lock-in
 - Tom has an interface/

16 Apr 2024

- Points for discussion
 - v0.23 released!
 - Ghislain is working through Clinica test suite
 - Nipype2pydra
 - Function task converter ([nipype2pydra#13](#))
 - Package layout (e.g. [nipype/pydra-ants@nipype-auto-conv](#))
 - Testing framework
 - Pytest structure
 - Inputs tested with `TimeoutWorker`
 - Outputs tested from reference working dir
 - Reference data
 - Detailed typing and `Fileformats.sample()` method
 - Where to store it
 - Text-based workflow converter (i.e. regular expression based)
 - Path to v1.0
 - Plans for picking up docs/tutorial work again ([tutorial objectives](#))
 - Proposed syntax changes ([#692](#))
 - Target % of nipype interfaces converted??
 - Improving the “sad path” experience
 - E.g. Crash Jupyter notebooks
 - ...

28 Feb 2024

- Points for discussion
 - v0.23 release
 - Tom's outstanding PRs
 - [Hash change guards #698](#)
 - [Caching of file-set hashes by local path and mtimes #700](#)
 - [Bugfix for optional output template files #709](#)
 - [BYO worker #733](#)
 - Environments feature
 - How does it work?
 - Whether an "environment manager" would work ([#701](#))
 - Anything else?
 - Nipype2pydra
 - Function task converter ([nipype2pydra#13](#))
 - Package layout (e.g. [nipype/pydra-ants@nipype-auto-conv](#))
 - Testing framework
 - Pytest structure
 - Inputs tested with TimeoutWorker
 - Outputs tested from reference working dir
 - Reference data
 - Detailed typing and *Fileformats.sample()* method
 - Where to store it
 - Text-based workflow converter (i.e. regular expression based)
 - Path to v1.0
 - Plans for picking up docs/tutorial work again ([tutorial objectives](#))
 - Proposed syntax changes ([#692](#))
 - Target % of nipype interfaces converted??
 - Improving the "sad path" experience
 - E.g. Crash Jupyter notebooks
 - ...

19 Sep 2023

- Points for discussion (link to related issues/PRs/etc)
 - Tom: Syntax changes proposal [pydra#692](#)
 - Using October "hack week" to focus on documentation/tutorials, preparing notes [here](#)
 - Environments [pydra#516](#)
 - What still needs to be done?
 - Tom: Would concept of "Environment Manager" work? [pydra#701](#)
 - Tom: PRs to finish off typing/hashing work
 - Caching file hashes by path+mtime: [pydra#700](#)
 - Hash change guards: [pydra#698](#)
 - Superclass auto cast: [pydra#696](#)
 - SLURM/dask-related test/coverage failures [pydra#667](#)

- Tom: Making Pydra a proper “namespace package” [pydra#679](#)
- Time of next meeting with daylight savings change (in Oz)
- [Updates to the tutorial](#)
- Notes
 - October hack week to focus on reworking documentation/tutorials
 - Syntax changes (such as [pydra#692](#)) could be proposed as part of this process
 - Look to what is done well (and not) in Nipype docs/tutorials
 - Use Diataxis approach for organising docs (<https://diataxis.fr/>)
 - Structure of docs to be revisited
 - Discussion on whether to combine docs and tutorial into one site/repo or keep separate
 - Use combination of JupyterBook for tutorials and potentially furo sphinx theme for conceptual/api reference
 - Online doc containing separate headers for 4 themes outlined in Diataxis to be circulated prior to hack week for brainstorming
 - Next release
 - Aim to complete within a few weeks
 - To include typing/hashing and environment PRs listed above ([pydra#700](#), [pydra#698](#), [pydra#696](#), [pydra#516](#))
 - Keep the same time for future meetings for October, but revisit when NH clocks go back in Nov

29 Aug 2023

- Points for discussion (link to related issues/PRs/etc)
 - Tom: Static type-checking: [pydra#670](#), TLDR:
 - [task definition syntax changes](#)
 - [wf syntax changes](#)
 - [task/wf execution changes](#)
 - [Satra's concerns and my response](#)
 - [Summary of bw-compatible-breaking syntax changes](#)
 - Tom: Update on auto-gen of task packages for nipype2pydra conv: [nipype2pydra:#10](#)
 - “Backlog” (to be carried over but probably won’t have time)
 - Tom: Intermittent failures gathering results with CF: [pydra#671](#)
 - Aditya: Slurm discussion: [pydra#667](#)
- Additional notes
 - Chris: Refactoring Environment PR
 - Dorota:
 - Updating tutorials to use type checking
 - Fixed dask!
 - Newer version of slurm container
 - Tom:

- State of the issues/PRs:
 - #653 has a couple helper functions
 - #655 Changes tasks to dataclasses
 - #670 (issue) has broader discussion
-
-

3 Aug 2023

- Points for discussion (link to related issues/PRs/etc)
 - Tom: Type checking PR: [pydra#662](#)
 - Relaxing type-checking for super classes: [pydra:#682](#)
 - Guards on task hashes changing during execution: [pydra:#681](#)
 - Upper-bounding fileformats dependency
 - Docs failing
 - Splitting of nodes in inner workflows no longer supported
 - Caching of file-hashes no longer performed [pydra:#683](#)
 - Tom: Auto-gen of task packages for nipype2pydra conv: [nipype2pydra:#10](#)
 - Tom: Intermittent failures gathering results with CF: [pydra#671](#)
 - Tom: Static type-checking: [pydra#670](#), TLDR:
 - [task definition syntax changes](#)
 - [wf syntax changes](#)
 - [task/wf execution changes](#)
 - [Satra's concerns and my response](#)
 - Aditya: Slurm discussion: [pydra#667](#)
- Additional notes
 - Typing PR ([pydra#662](#)) merge accepted, to be released as an alpha for now
 - [pydra:#682](#), [pydra:#681](#), [pydra:#683](#) accepted in principle, will look to implement before full release
 - Upper-bounding fileformats dependency rejected due to bad experience doing this with Nipype
 - Docs failing due to sphinx update
 - Splitting of nodes in inner workflows functionality dropped (and test deleted) because it doesn't have a strong use case
 - Auto-gen of task packages ([nipype2pydra:#10](#)) for nipype2pydra accepted in principle:
 - Test out how it works for a few toolkits (I was thinking ANTs, AFNI and FSL) before moving on to generate repositories for all nipype interface subpackages
 - Static type-checking ([pydra#670](#)) to be discussed in the next dev meeting

29 June 2023

- Points for discussion (link to related issues/PRs/etc)

- Tom: Type checking PR: [pydra#662](#)
- Ghis: Lazy field in template PR: [pydra#657](#)
- Yibei: Converter nipy2pydra: [nipy2pydra#6](#)
- Aditya: Slurm discussion: [pydra#667](#)
- Tom: Intermittent failures gathering results: [pydra#671](#)
- Tom: Static type-checking: [pydra#670](#)
- Tom: min/max field values (are they supported?)
- Additional notes
 -

23 May 2023

- Discussion notes
 - Introductions (welcome, Aditya!)
 - GSoC proposal: [link](#)
 - Possible topics (late breaking agenda)
 - BIDS-Prov: [BIDS Extension Proposal \(BEP028\): Provenance](#)
 - File hashing
 - Debugging
 - Tom's suggestion for Jupyter notebook reconstructing path to allow interactive inspection
 - Maybe a dashboard based on the audit interface could cover much of this use case? This was a previous idea of Satra's.
 - File hashing / input coercion
 - Tom: Coerce file-likes and basic containers (lists, dicts, sets), otherwise leave inputs alone
 - Need to consider map/combine-related (nested) lists
 - LazyInputs/Outputs should have type tags (need not be proper Python annotations) so we can check connection validity without values

Table for the meetings week 2

| Topics for meeting / hacking in week 2 | People interested in working | Additional people for discussion | Meeting time: week2 In EDT (Boston) |
|-----------------------------------------------|------------------------------|----------------------------------|-----------------------------------------------|
| General updates | DJ, YC, GV, TC, CM | | 8-8:30 Tuesday (DJ, GV, TC, CM) |
| Updates for autoconverter to | DJ, YC, GV, TC | | 12-13 Tue (DJ+YC) |

| | | | |
|----------------------------------------------------|----------------|--------|------------------------------------|
| translate the interfaces | | | 8-9 Wed (DJ, GV, TC) |
| Workflow provenance | DJ, TC | SG (?) | |
| Input/output convention updates | GV, CM, DJ, TC | | 8:30-9:00 Tuesday (DJ, GV, TC, CM) |
| Hash dispatch updates plus file format Integration | TC, DJ, CM | | |
| File manipulation task: datalad (?) | DJ, YC | YH (?) | |
| File manipulation: fsspec types exploration (?) | CM | | |

| Topics | People interested in working | Additional people for discussion | Weeks: week1 (May 2nd, 3d), week 2 (May 9th, 10th) | Meeting time: week1 In EDT (Boston) |
|-------------------------------------------------|------------------------------|----------------------------------|----------------------------------------------------|---------------------------------------------------------------------------|
| Using autoconverter to translate the interfaces | DJ, YC, GV, TC | | week 1 | Tue 8-9am (DJ, GV, TC); Tue 10:30-11:00 am: updates for Yibei (DJ, YC) |
| Workflow provenance | DJ, TC | MD (?) - no answer | Week 2 | |
| SPM matlab | DJ, GV, CM(?), | SG (?) - Satra in DC | Week 1 | Tue 9am (DJ, GV, CM) |
| Input/output | GV, CM, DJ, | | Week 1 | Wed 8-9am (GV, |

| | | | | |
|------------------------------------------------------|---------|--------|--------|---------------------------|
| convention | TC(?) | | | CM, DJ, TC) |
| Hash dispatch | CM, TC | | Week1 | Tue 4:30-5:00 (CM, TC) |
| File format Integration | TC, DJ, | | Week 2 | |
| File manipulation task: datalad | DJ, YC | YH (?) | Week 2 | |
| File manipulation: fsspec types exploration | CM | | Week 2 | |
| IPython notebook "crashfile" | TC | | Week 2 | |

Hack Days: May 2023

Discussion about File/Directory

- Why do we need it:
 - Hashing
 - Containers
 - validation that exists
- Finishing Environment class first [516](#)
- Finishing hashing: [644](#)
- For type checking:
 - `File = ty.NewType("File", Path)`
This probably will not work
 - Alias - this is probably the way to go
- Internally we will be using Path-like
- In the future we should allow for more specific formats (Tom's package) that would provide some extra validation
-

Converter meeting:

- Tom's generic converter repo (not tested yet):
<https://github.com/Australian-Imaging-Service/nipype2pydra>
Tom was pretty convinced that we can build generic converter after working with several packages:
 - Ants converter:
<https://github.com/tclose/pydra-ants/blob/port-converter/tools/converter.py>
 - Freesurfer:
<https://github.com/ablachair/pydra-freesurfer/blob/port-converter/tools/converter.py>
- Issues with converter:
 - Naming convention:
- Example from Ghislain: interfaces converted manually:
 - Harder since I didn't use the nipype as a starting points
<https://github.com/aramis-lab/pydra-freesurfer/blob/main/src/pydra/tasks/freesurfer/gtmseg.py>
 - <https://github.com/aramis-lab/pydra-fsl/blob/main/src/pydra/tasks/fsl/eddy.py>
 - <https://github.com/aramis-lab/pydra-fsl/tree/main/src/pydra/tasks/fsl/utils>

SPM meeting:

- Drop MCR vs Matlab distinctions in-task

March 2023

- Hackday : May 2-3rd, May 9-10th (Tuesday, Wednesday)

Todo before/during hack day:

- [Pydra-fsl](#) moved to nipype and update metadata
 - Test autoconverter and prepare the summary of the issues (DJ +YC will try to test before)
- The same for pydra-freesurfer
- Logging

- Workflow Provenance
 - Summary what was done by the Ryan and what is in the BIDS-Prov
- The SPM-task class (Satra for a meeting, discussion before the meeting)
- Naming of the input/output, perhaps some consistency among tools
 - Example for fsl/freesurfer from Ghislain (prepare a doc)
- File manipulation tasks
 - Pydra-datalad
 - DVC: <https://dvc.org/>
 - Basic data writing/reading, similar to DataGrabber; dynamic?
 - fsspec: <https://filesystem-spec.readthedocs.io/en/latest/>
 -
- File format integration from Tom's package
- Reimplementation interface hashing using `functools singledispatch` a la <https://github.com/nipype/pydra/issues/626> (could be included in point above)

February 2023

- discussion from Tom:
 - <https://github.com/nipype/pydra/discussions/620#discussioncomment-5125962>
 - Yes, enable `attrs`
 - When define file in pydra; file=path-like, path-like + hash; intersection()
- Using mypy
- Pydra-fsl in PyPI
 - Plan when & how release (policy)
 - Lack of mechanisms of registering plug-ins...
 - semi-auto-converter - new name space?
 - Longer-descriptive name
 - Annotate xxx correspond to yyy in nipype (either in yml or code)
 - Combine ghislain's fsl package with pydra-fsl (did I miss anything here)
 - Add ghislain's fsl to pydra-fsl to see whether it can auto-converted
 - Keep the history how the task was developed/maintained; dependency problem
 -
- New syntax for Docker tasks (env class), ask for feedback:
 - https://github.com/nipype/pydra/blob/rf/environments/pydra/engine/tests/test_dockertask.py#L35
- GSoC:
 - <https://neurostars.org/t/gsoc-2023-project-idea-16-2-converting-existing-scientific-workflows-to-new-dataflow-engine-written-in-python-pydra-350-h/24593/4>
 - <https://neurostars.org/t/gsoc-2023-project-idea-16-1-adding-new-workers-and-resource-management-to-new-dataflow-engine-written-in-python-pydra-175-h/24592/8>
 -

TO DISCUSS (for the future meeting):

- Dynamical workflow connections
 - Doing type-checking when constructing the workflow
 - Type-checking when people use pydra to construct their workflows
- Decision on name space (pydra hackathon late April & early May, Dorota will start a poll on github)

January 2023

- Fileformat: <https://arcanaframework.github.io/fileformats/index.html>
- Converting types
- New environment class: syntax for Docker:
https://github.com/nipy/pydra/blob/493071d17fa3d069ff74cb7b9a4904ef4eb823c2/pydra/engine/tests/test_dockertask.py#L35

November 2022

- Pydra tasks packaging:
 - <https://github.com/nipy/pydra/issues/582>
- Testing patterns
 - Doctests for cmdline generation (a la nipy 1)
 - `--dry-run` for tools that support it?
- Handling multiple versions of underlying tools
 - Old style: min/max_ver in input spec, state machine code in _list_outputs
 - Possibilities:
 - Separate packages: pydra-fsl5 and pydra-fsl6
 - Submodules: pydra.tasks.fsl.ver5, pydra.tasks.fsl.ver6, pydra.tasks.fsl.latest
 - Some level of min/max_ver still kept?
 - Considerations
 - Workflows written agnostic to underlying tools
 - Some overlaps with provenance...