JOHNS HOPKINS
APPLIED PHYSICS LABORATORY

Neuroscience Workflows
Joint Management of Data and Computation

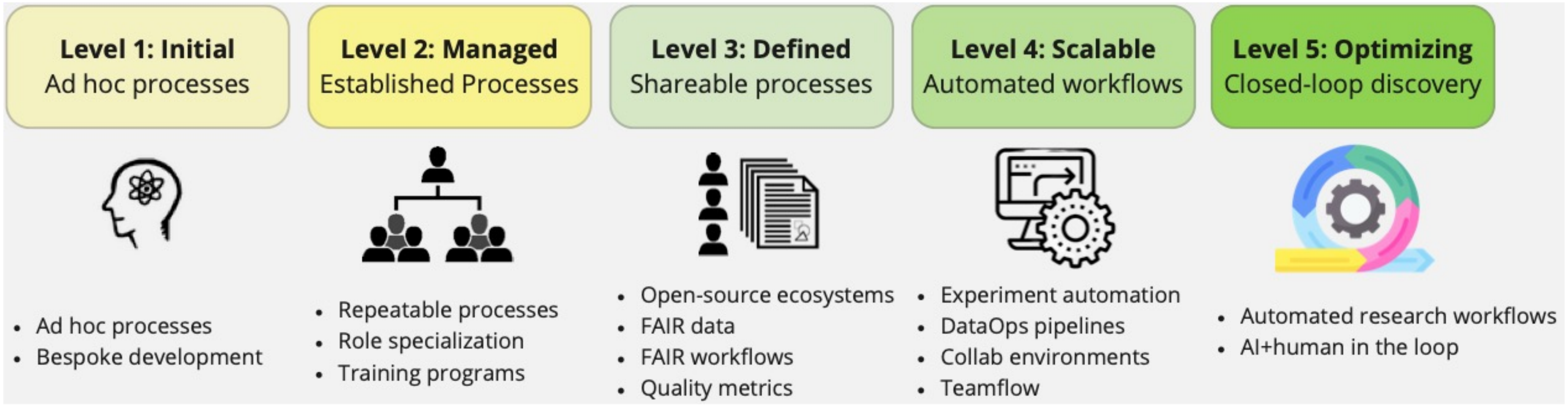
HEP Workflow Management Workshop

Erik C. Johnson, Christa Cooke, Nikoo Dalili, Victoria Rose, Daniel Xenes

erik.c.johnson@jhuapl.edu

Automated research workflows for neuroscience

Neuroscience Operations Maturity Roadmap

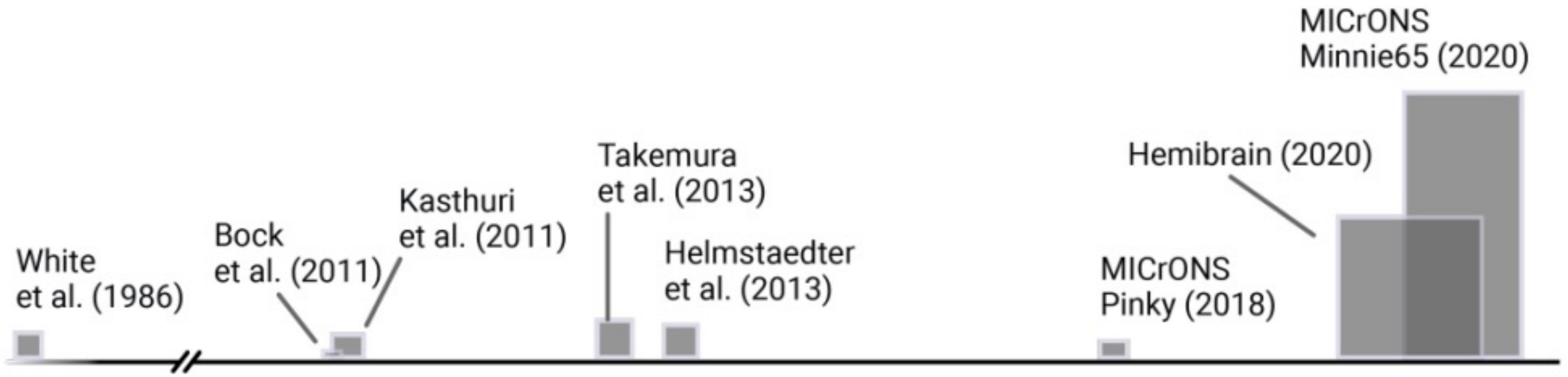


Level 1: Initial Ad hoc processes	**Level 2: Managed** Established Processes	**Level 3: Defined** Shareable processes	**Level 4: Scalable** Automated workflows	**Level 5: Optimizing** Closed-loop discovery
• Ad hoc processes • Bespoke development	• Repeatable processes • Role specialization • Training programs	• Open-source ecosystems • FAIR data • FAIR workflows • Quality metrics	• Experiment automation • DataOps pipelines • Collab environments • Teamflow	• Automated research workflows • AI+human in the loop

Data Archives of the US BRAIN Initiative

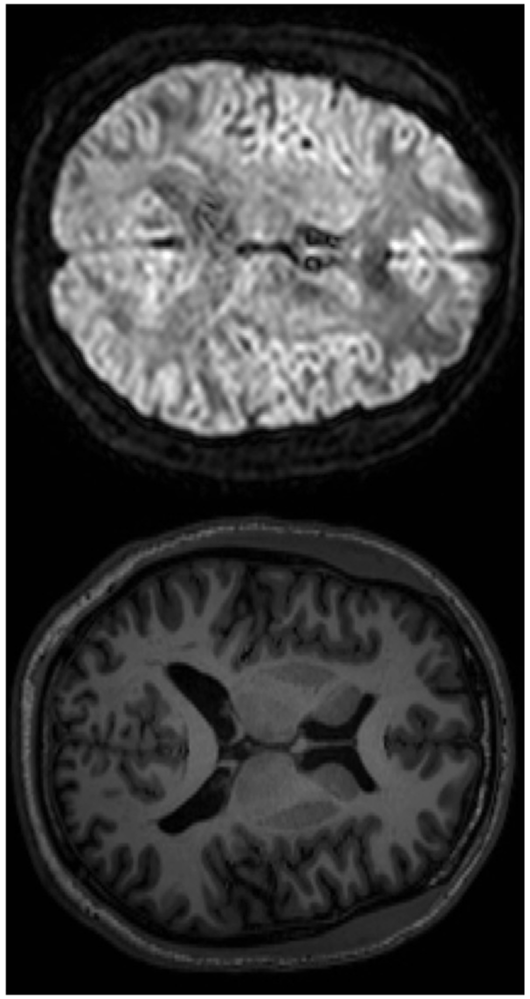
- Increasing set of public data repositories
- For example, US BRAIN Initiative Archives

Nanoscale Structure and Connectomics	Neurophysiology, Behavior	Genomics, Transcriptomics	Macroscale Brain Structure and Function	Human Electrical and MEG Recordings	Human Invasive Physiology
BossDB	DANDI	NeMO The Neuroscience Multi-omic Data Archive	BIL	NEMAR	DABI Data Archive Brain Initiative
900+ TBs Data	500+ TBs Data	390+ TBs Data	5700+ Datasets	12+ TBs Data	110+ Datasets



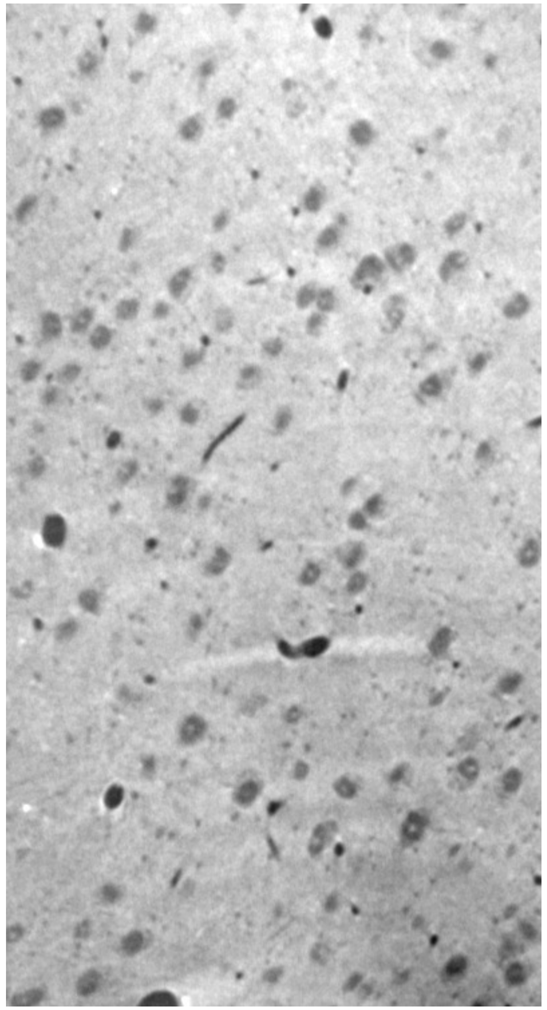
Neural Structure at Different Scales

MRI. 1 mm^3: 1 byte



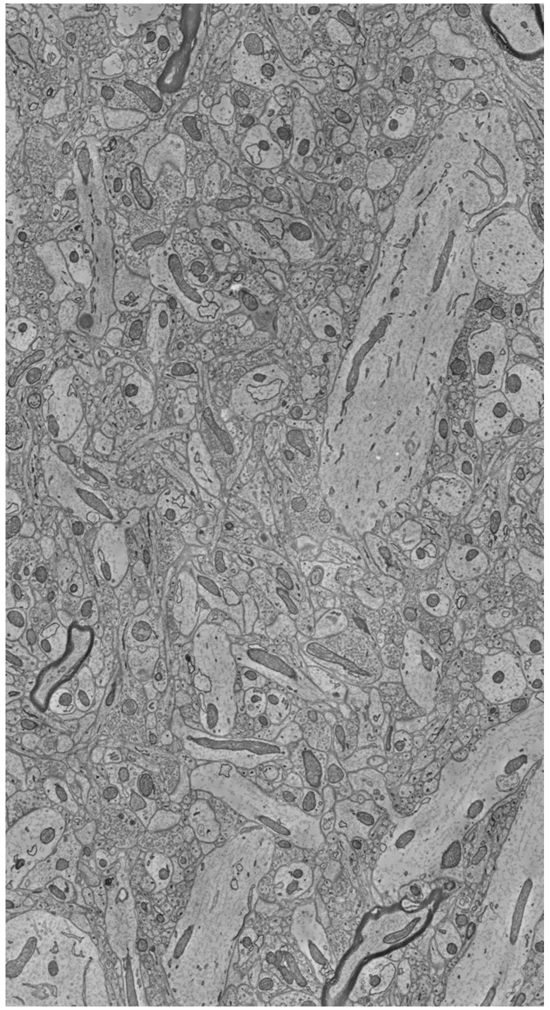
Voxel size: ~1 mm^3

X-ray. 1 mm^3: ~1 gigabyte



Voxel size: ~1 um^3

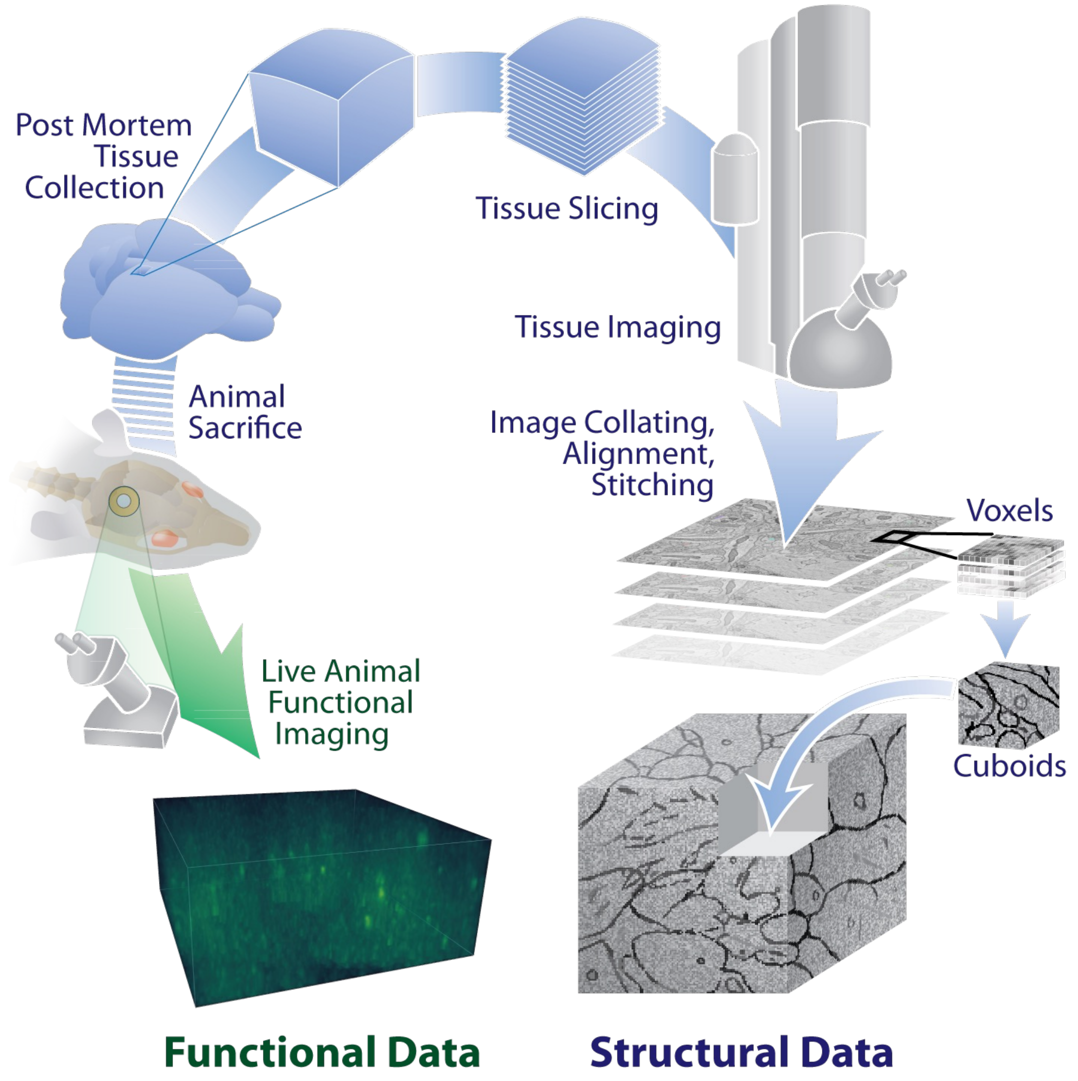
EM. 1 mm^3: ~2 petabytes



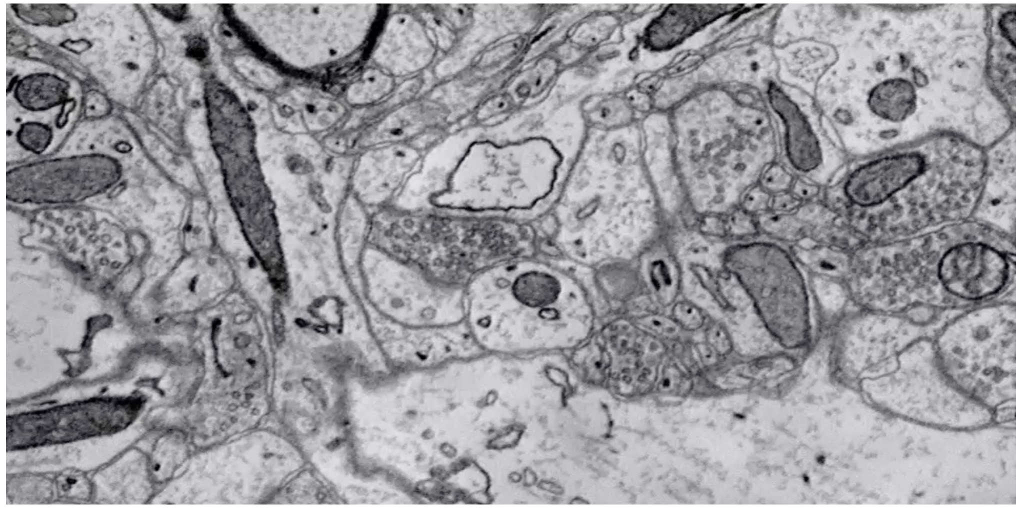
Voxel size: ~270 nm^3

A quick introduction to the field

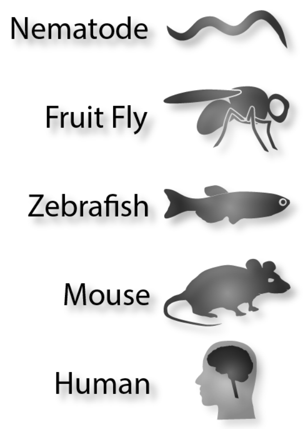
MICrONS (Machine Intelligence from Cortical Networks)


Post Mortem Tissue Collection
Tissue Slicing
Tissue Imaging
Animal Sacrifice
Image Collating, Alignment, Stitching
Voxels
Live Animal Functional Imaging
Cuboids
Functional Data
Structural Data

Structural Data:



BossDB Ecosystem

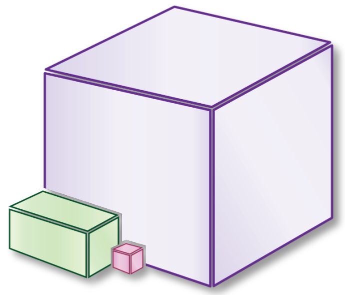


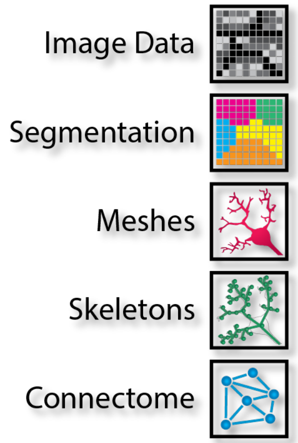
Multiple model organisms:

- Invertebrates
- Vertebrates
- Mammals

Varying imaging modalities and scales:

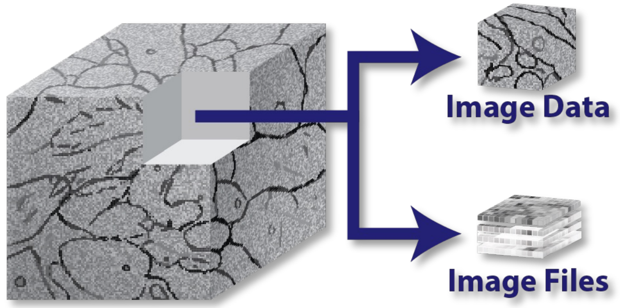
- EM, XRM, XNH Light Microscopy
- Dataset size (0.001 – 400 TB)
- Image extent (10 μm^3 – 1 mm^3)
- Image resolution (1 nm – 1 μm)





Image, segmentation, annotation, connectomics data:

- Image data (e.g., numpy array)
- Image tiles / image files (e.g., *.PNG)
- Object data (e.g., numpy array)
- Connectome (e.g., *.CSV)
- Meshes (e.g., precompute data)

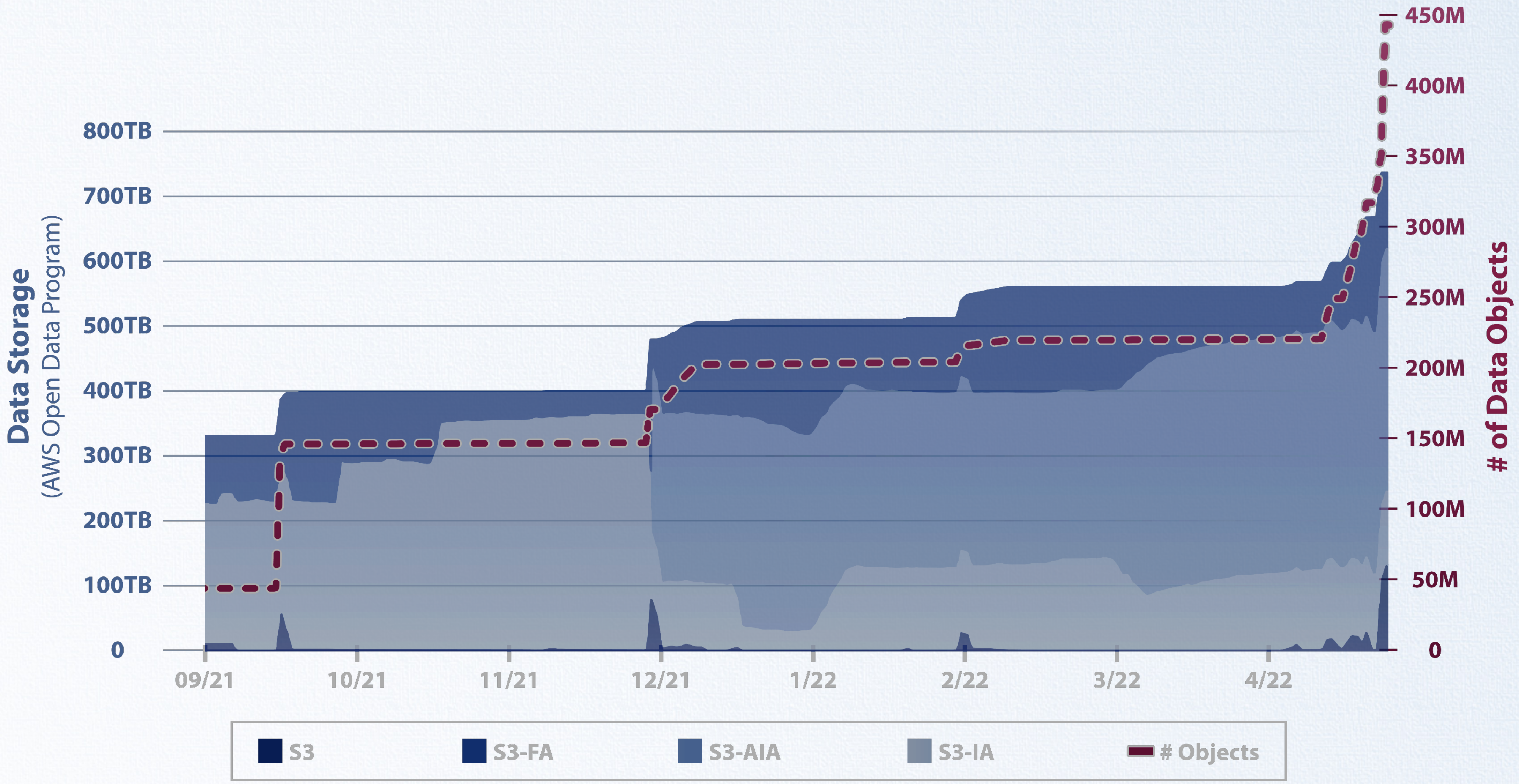


Work easily with sub-volumes, cut-outs

Visualization, Workflows, Standards, and more!

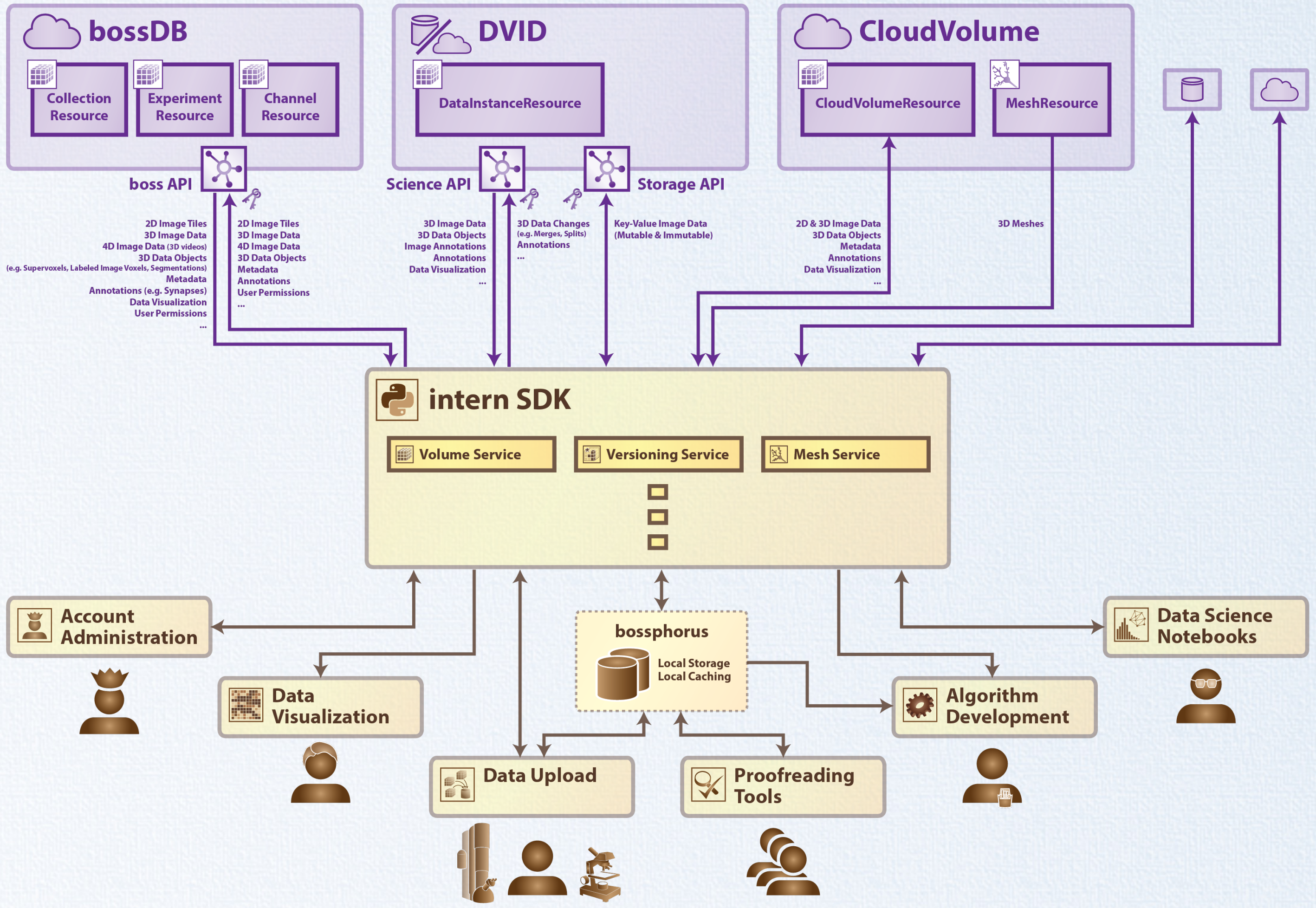
https://bossdb.org/

BossDB – Open Data Storage



intern SDK

- ***intern*** **SDK**
  - https://bossdb.org/tools/intern
    - Python, compatible with Jupyter
    - Node.js
    - Julia (partial)
    - Rust (partial)
  - Connects with multiple community resources (storage, visualization)
    - DVID
    - CloudVolume / precompute
    - Neuroglancer
    - syglass.io
  - Facilitates ingests with a variety of community data formats

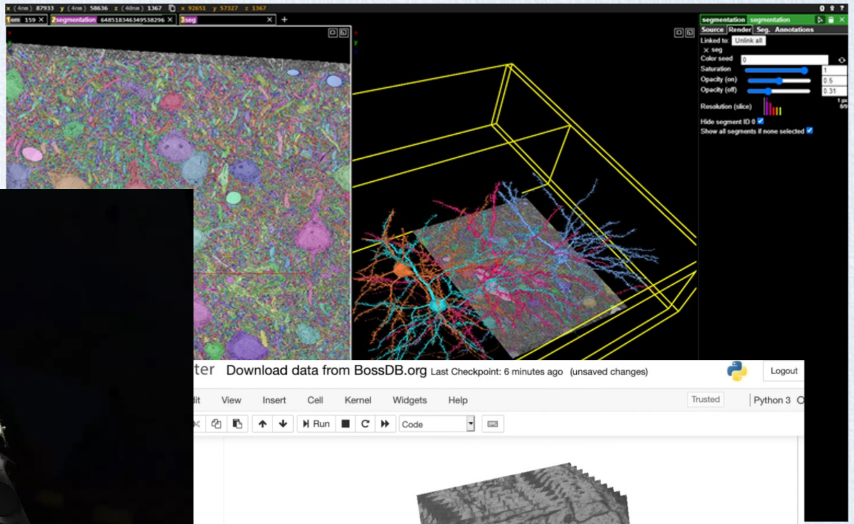


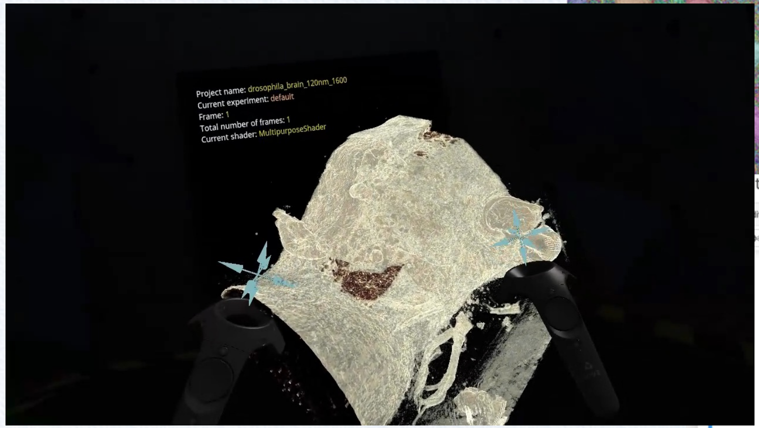
Matelsky 2020 - https://www.biorxiv.org/content/10.1101/2020.05.15.098707v1

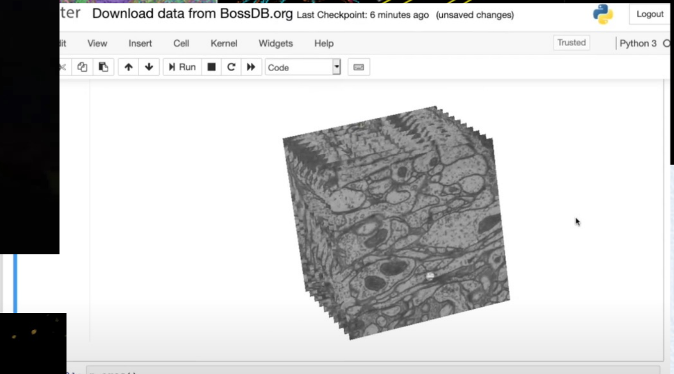
BossDB – Data Access & Visualization

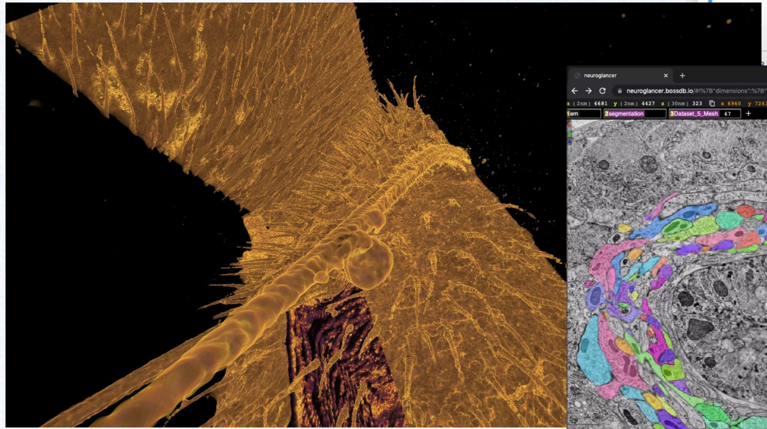
- Multiple Data Visualization Options
  - Neuroglancer
  - syGlass.io
  - Python

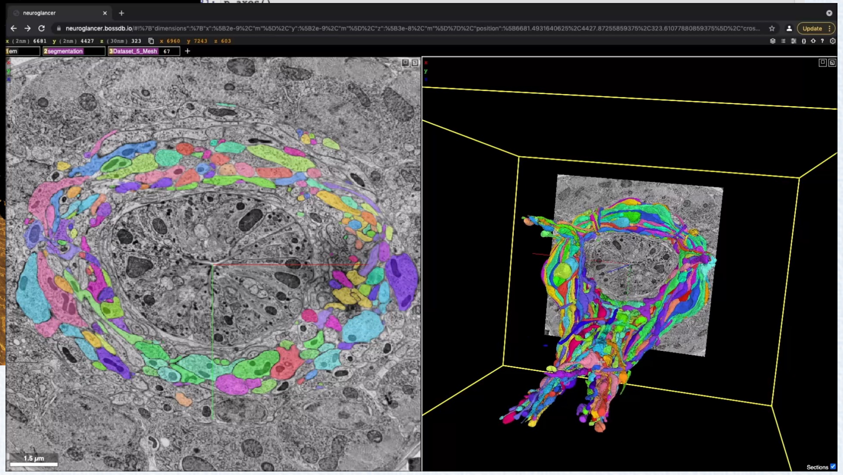
- Account-Free Data Access




Project name: drosophila_brain_120nm_1600
Current experiment: default
Frame: 1
Total number of frames: 1
Current shader: MultipurposeShader

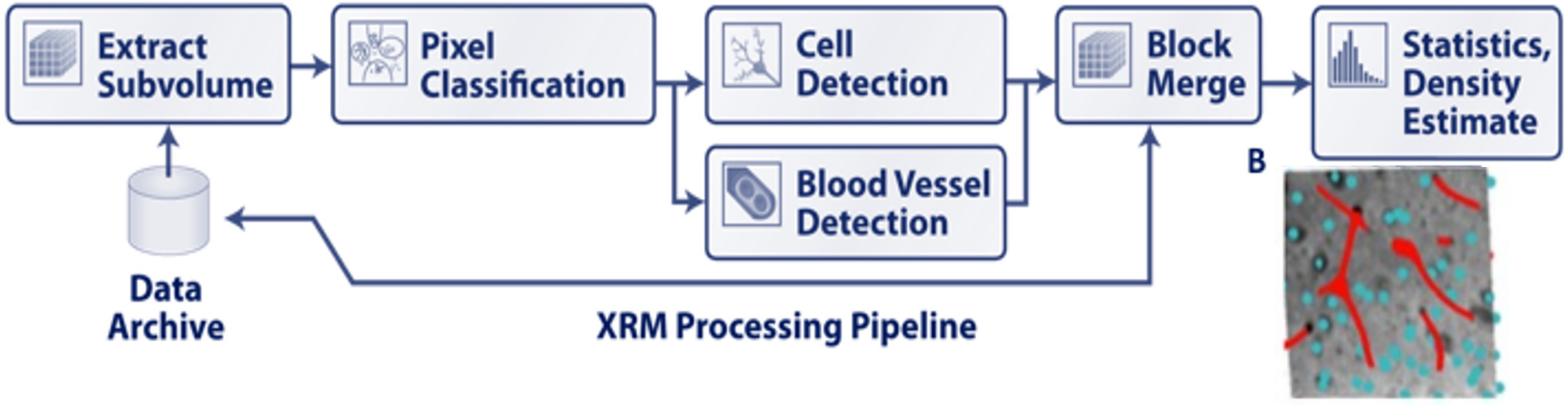

Download data from BossDB.org Last Checkpoint: 6 minutes ago (unsaved changes)
Logout
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

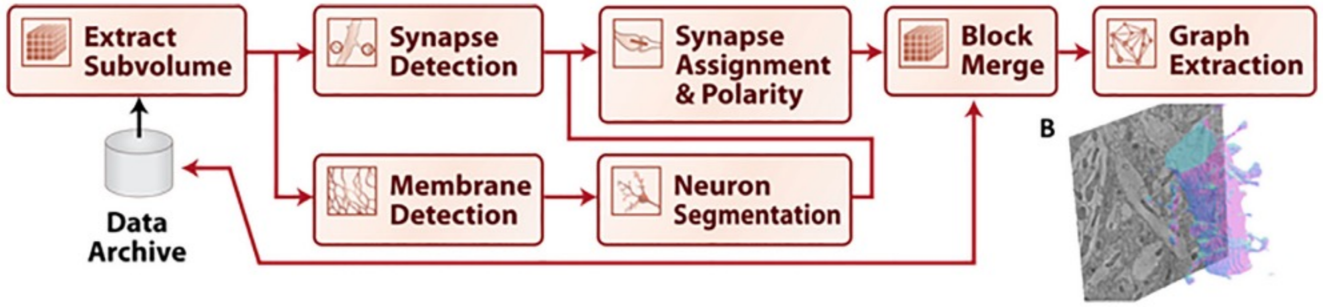


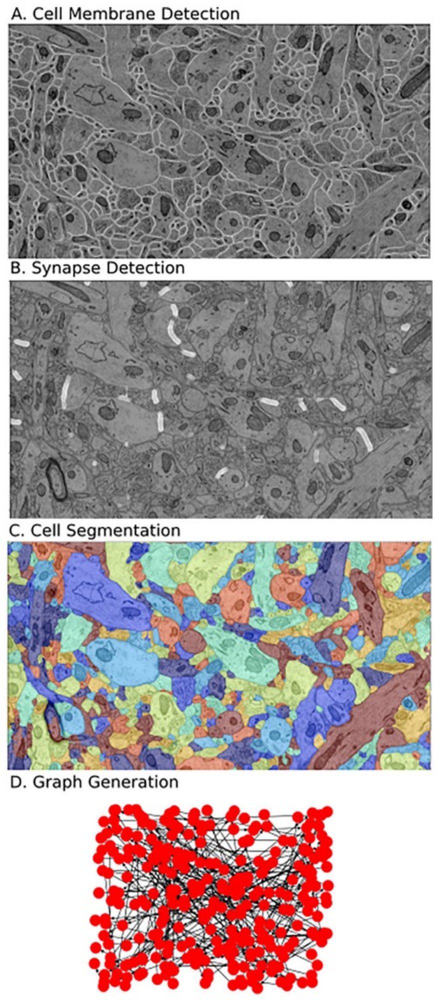


Kuan & Phelps 2020, Witvliet 2021, MICrONS Minnie Dataset

Neuroscience Workflows: Examples from EM







Automated Workflow Management

- Software to execute a defined sequence of processing steps
- Automated execution of computational steps
- Computational dependencies captured as a Directed Acyclic Graph (DAG)
- Simple example use case: automate execution of a series of scripts which are run every time you collect new data
- Hundreds of software solutions for this
  - Academic

    nextflow https://github.com/nextflow-io/nextflow

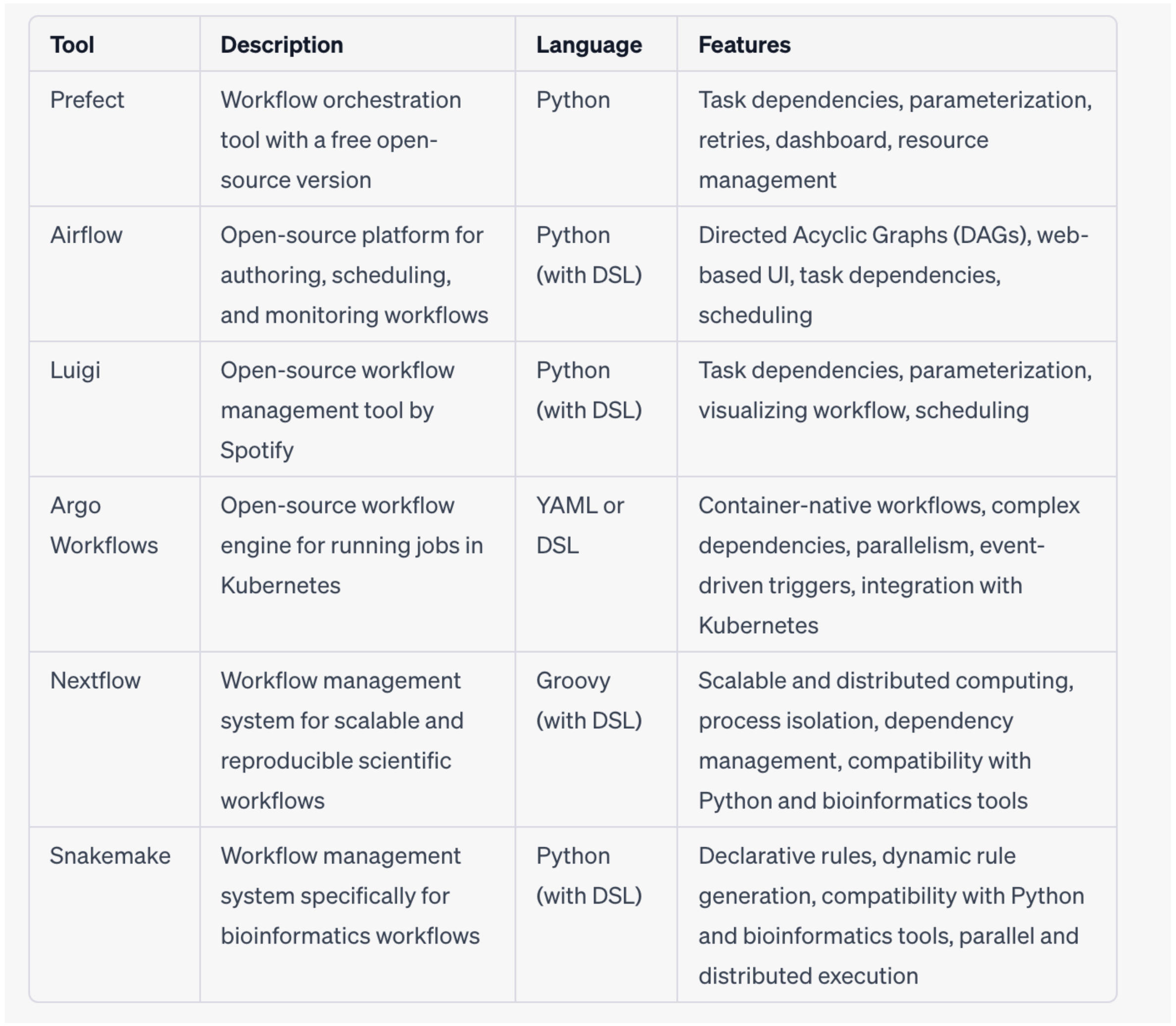
    https://github.com/snakemake
  - Commercial

    Apache Airflow Flyte PREFECT

Key Features of Workflow Management

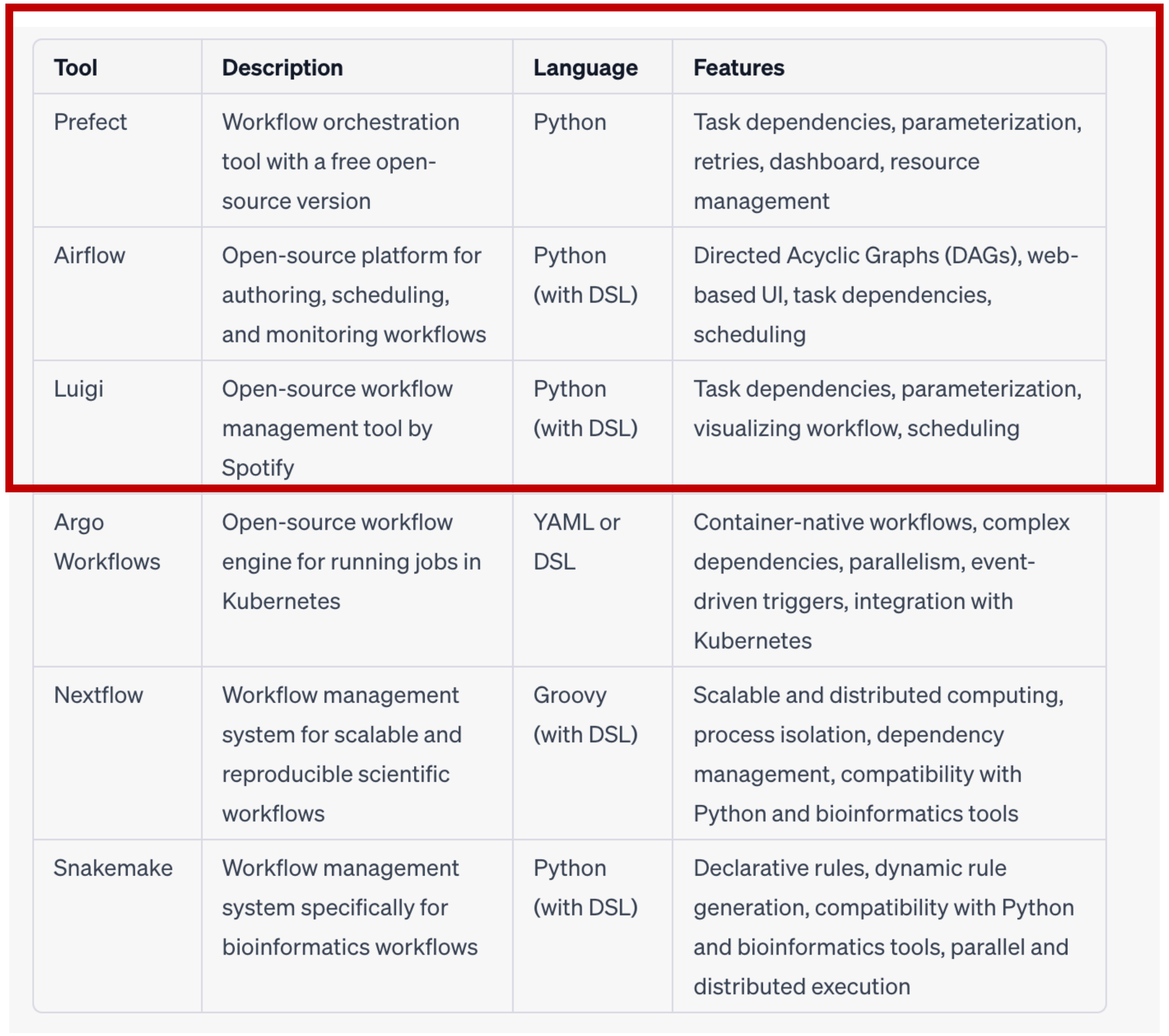
- Workflow execution control- start, stop, restart, schedule
- Monitoring tools- cost, resources, execution success
- Scalable execution from local compute to clusters/cloud
- Some things to consider:
  - Setup complexity
  - Open-source?
  - Can you share and version control workflows?
  - What language is used to specify workflows?
  - How are backend resources managed?
  - What visualization tools are included?

Comparing Low-cost Workflow Orchestration Systems



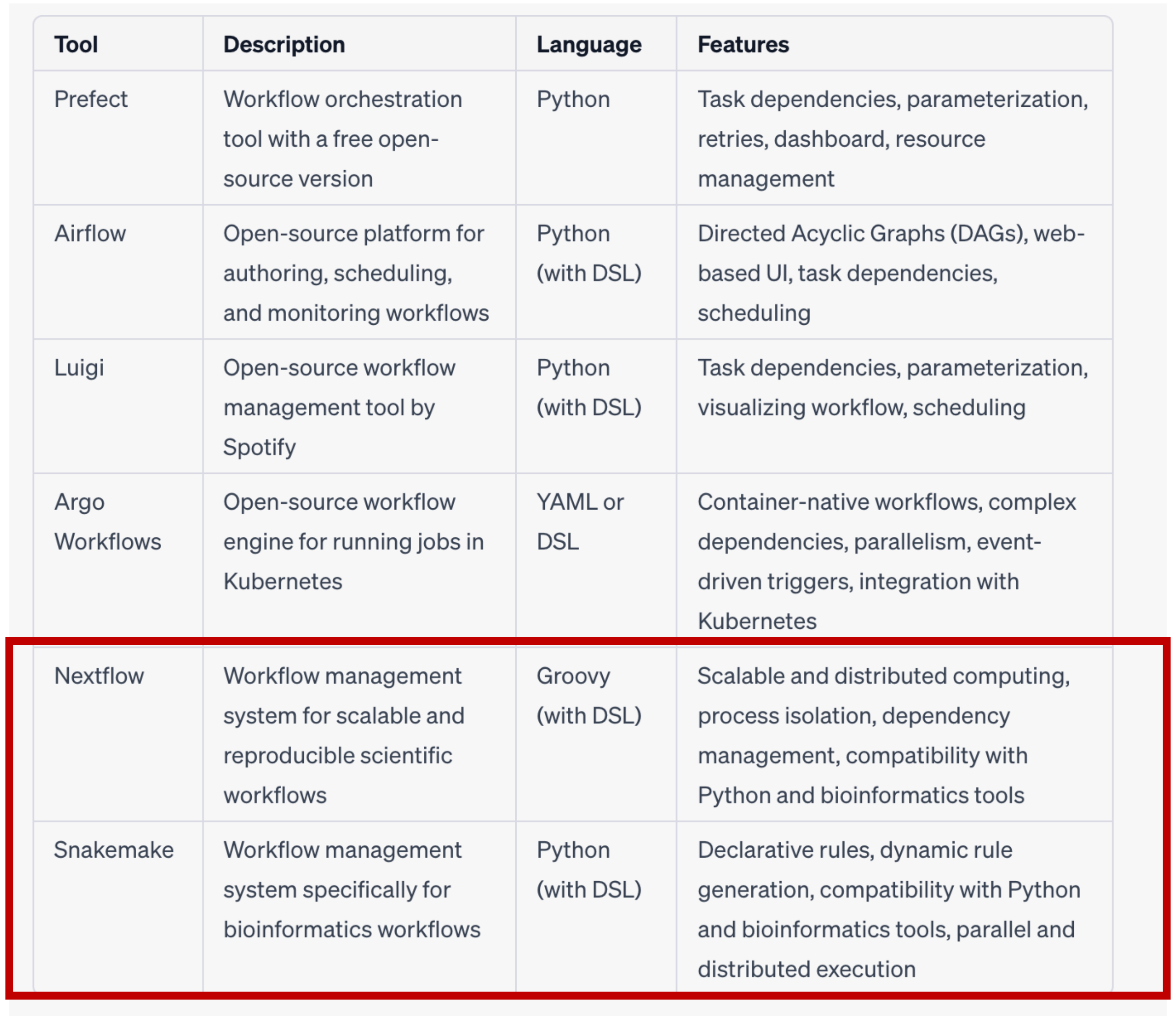
Tool	Description	Language	Features
Prefect	Workflow orchestration tool with a free open-source version	Python	Task dependencies, parameterization, retries, dashboard, resource management
Airflow	Open-source platform for authoring, scheduling, and monitoring workflows	Python (with DSL)	Directed Acyclic Graphs (DAGs), web-based UI, task dependencies, scheduling
Luigi	Open-source workflow management tool by Spotify	Python (with DSL)	Task dependencies, parameterization, visualizing workflow, scheduling
Argo Workflows	Open-source workflow engine for running jobs in Kubernetes	YAML or DSL	Container-native workflows, complex dependencies, parallelism, event-driven triggers, integration with Kubernetes
Nextflow	Workflow management system for scalable and reproducible scientific workflows	Groovy (with DSL)	Scalable and distributed computing, process isolation, dependency management, compatibility with Python and bioinformatics tools
Snakemake	Workflow management system specifically for bioinformatics workflows	Python (with DSL)	Declarative rules, dynamic rule generation, compatibility with Python and bioinformatics tools, parallel and distributed execution

Comparing Low-cost Workflow Orchestration Systems



Tool	Description	Language	Features
Prefect	Workflow orchestration tool with a free open-source version	Python	Task dependencies, parameterization, retries, dashboard, resource management
Airflow	Open-source platform for authoring, scheduling, and monitoring workflows	Python (with DSL)	Directed Acyclic Graphs (DAGs), web-based UI, task dependencies, scheduling
Luigi	Open-source workflow management tool by Spotify	Python (with DSL)	Task dependencies, parameterization, visualizing workflow, scheduling
Argo Workflows	Open-source workflow engine for running jobs in Kubernetes	YAML or DSL	Container-native workflows, complex dependencies, parallelism, event-driven triggers, integration with Kubernetes
Nextflow	Workflow management system for scalable and reproducible scientific workflows	Groovy (with DSL)	Scalable and distributed computing, process isolation, dependency management, compatibility with Python and bioinformatics tools
Snakemake	Workflow management system specifically for bioinformatics workflows	Python (with DSL)	Declarative rules, dynamic rule generation, compatibility with Python and bioinformatics tools, parallel and distributed execution

Comparing Low-cost Workflow Orchestration Systems



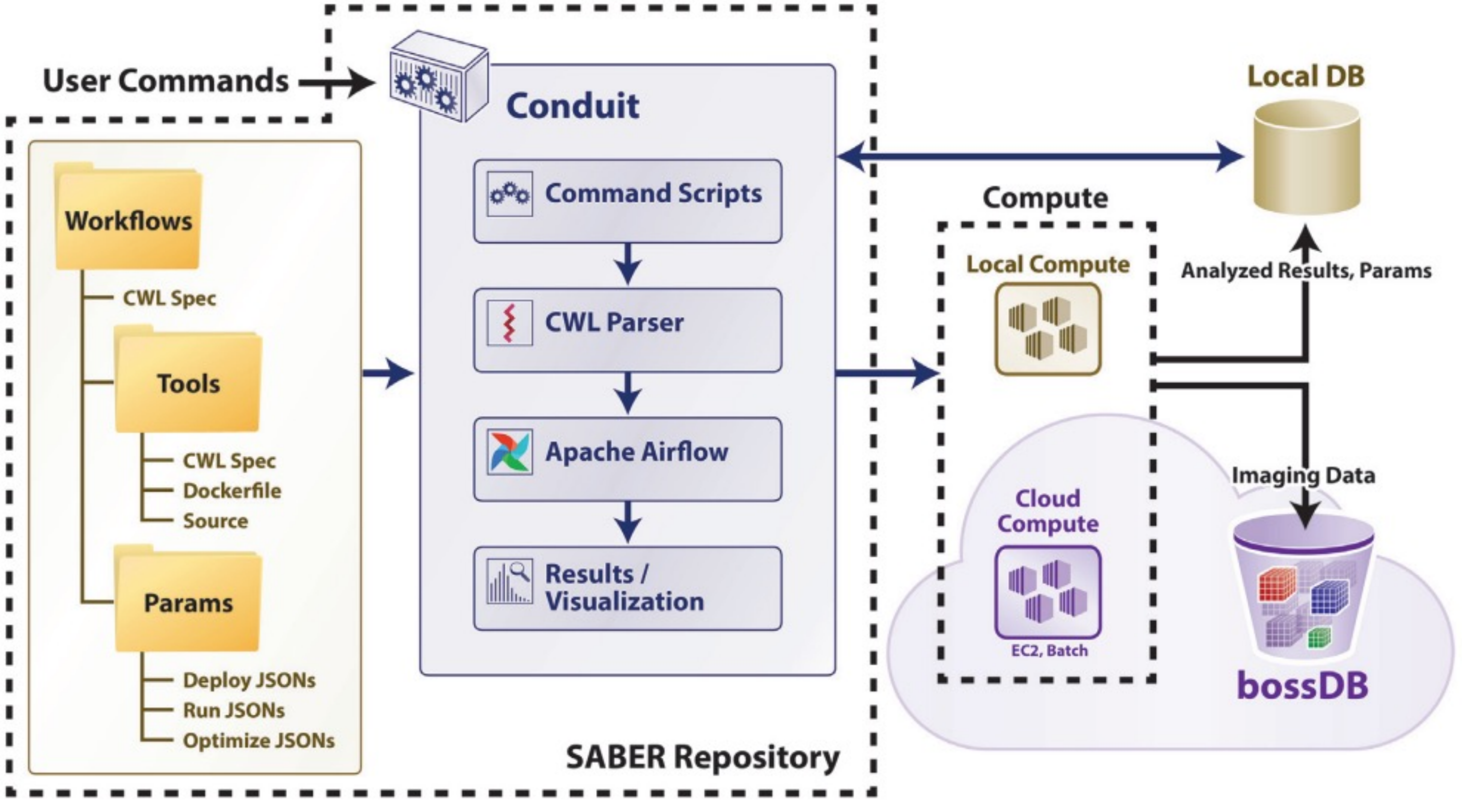
Tool	Description	Language	Features
Prefect	Workflow orchestration tool with a free open-source version	Python	Task dependencies, parameterization, retries, dashboard, resource management
Airflow	Open-source platform for authoring, scheduling, and monitoring workflows	Python (with DSL)	Directed Acyclic Graphs (DAGs), web-based UI, task dependencies, scheduling
Luigi	Open-source workflow management tool by Spotify	Python (with DSL)	Task dependencies, parameterization, visualizing workflow, scheduling
Argo Workflows	Open-source workflow engine for running jobs in Kubernetes	YAML or DSL	Container-native workflows, complex dependencies, parallelism, event-driven triggers, integration with Kubernetes
Nextflow	Workflow management system for scalable and reproducible scientific workflows	Groovy (with DSL)	Scalable and distributed computing, process isolation, dependency management, compatibility with Python and bioinformatics tools
Snakemake	Workflow management system specifically for bioinformatics workflows	Python (with DSL)	Declarative rules, dynamic rule generation, compatibility with Python and bioinformatics tools, parallel and distributed execution

Limitations of Workflow Management for Neuroscience

- Workflow structure and definition domain specific
- Managing heterogenous compute resources is a challenge
- Adoption is not widespread in neuroscience/neuroimaging outside of –omics communities
- Data management is a problem
  - Data sources
  - Data standards
  - Intermediate data products
- Possible approaches to overcome these challenges:
  - Integration with Data Archives
  - Data standards and standard APIs
  - Structured data management (e.g. DataJoint)

SABER: Workflow Management for Neuroimaging

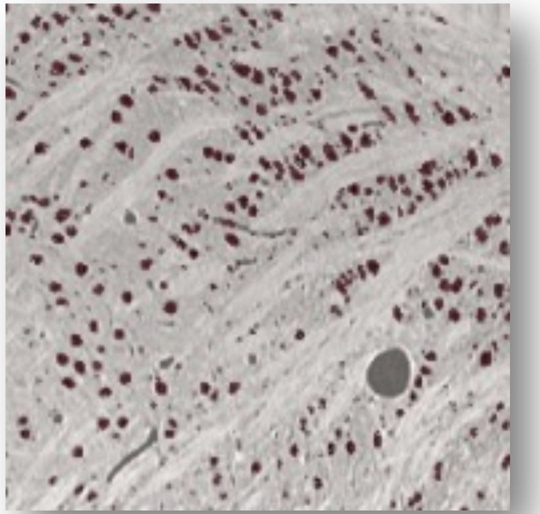
- Workflow management widely used in bioinformatics
  - Galaxy (https://usegalaxy.org/) is a great example
- Some long-established solutions
  - LONI
  - Nipype
- SABER aimed to integrate standard definitions (Common Workflow Language), scalable management, and dockerized tool library

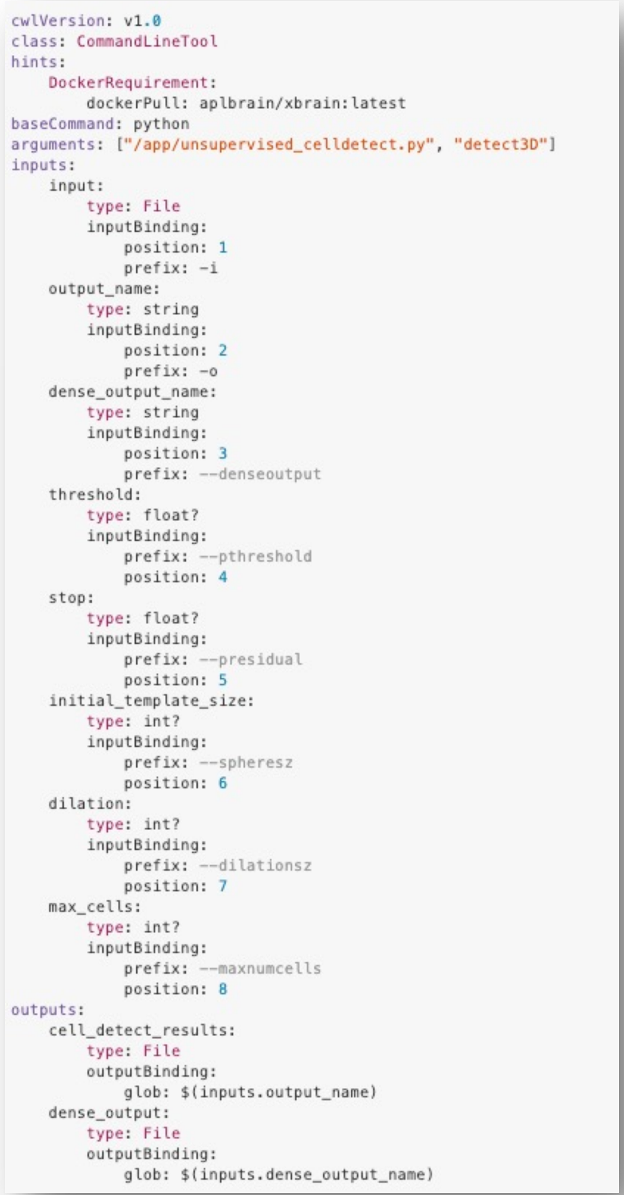


https://academic.oup.com/gigascience/article/9/12/giaa147/6042730

Example Tool – Cell Detection

- Docker-ized tool for detecting cell bodies in X-Ray Microtomography data.
- Leverages CWL’s `DockerRequirement` hint to enforce latest image is used for container creation.
- Defines required and optional hyperparameters.
- Provides a standard interface for future cell detection tools.

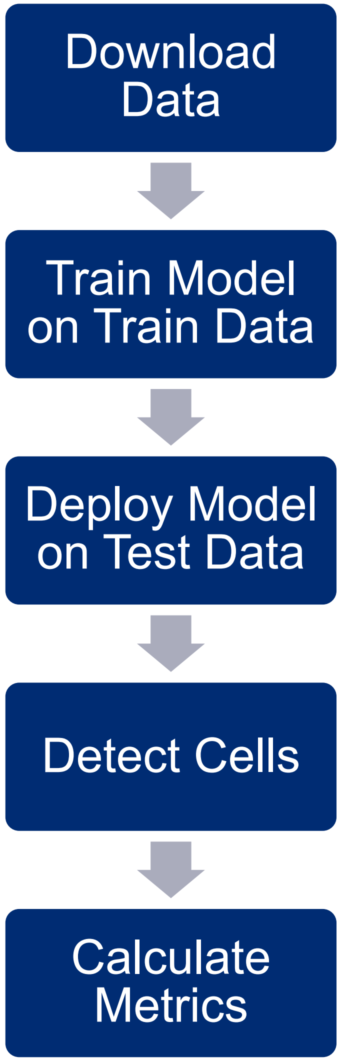


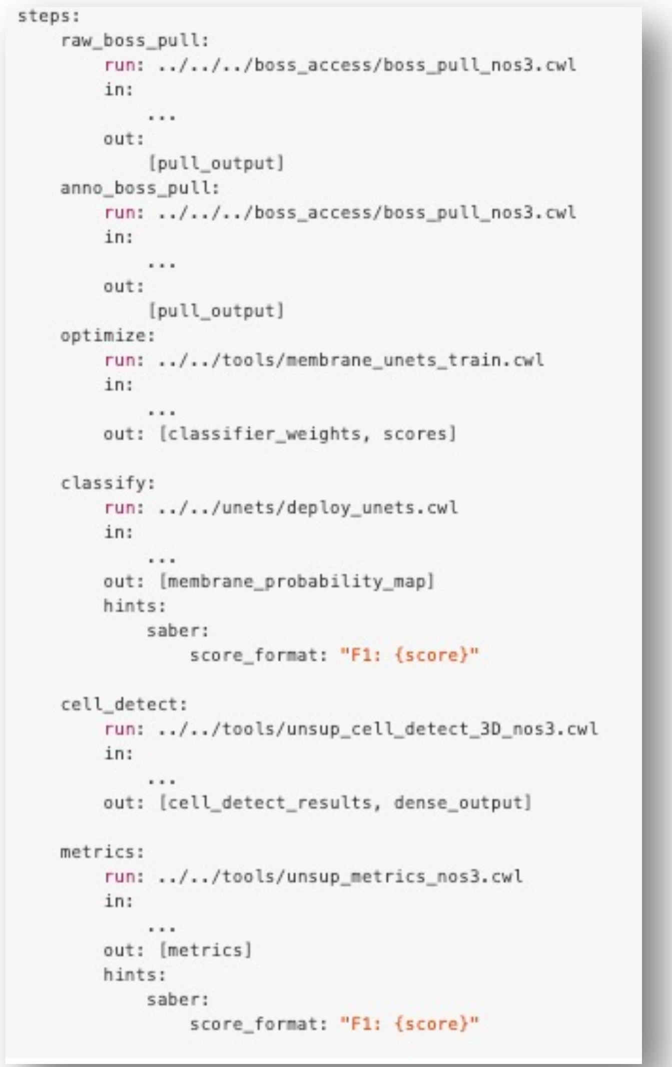


```yaml
cwlVersion: v1.0
class: CommandLineTool
hints:
    DockerRequirement:
        dockerPull: aplbrain/xbrain:latest
baseCommand: python
arguments: ["/app/unsupervised_celldetect.py", "detect3D"]
inputs:
    input:
        type: File
        inputBinding:
            position: 1
            prefix: -i
    output_name:
        type: string
        inputBinding:
            position: 2
            prefix: -o
    dense_output_name:
        type: string
        inputBinding:
            position: 3
            prefix: --denseoutput
    threshold:
        type: float?
        inputBinding:
            prefix: --pthreshold
            position: 4
    stop:
        type: float?
        inputBinding:
            prefix: --presidual
            position: 5
    initial_template_size:
        type: int?
        inputBinding:
            prefix: --spheresz
            position: 6
    dilation:
        type: int?
        inputBinding:
            prefix: --dilationsz
            position: 7
    max_cells:
        type: int?
        inputBinding:
            prefix: --maxnumcells
            position: 8
outputs:
    cell_detect_results:
        type: File
        outputBinding:
            glob: $(inputs.output_name)
    dense_output:
        type: File
        outputBinding:
            glob: $(inputs.dense_output_name)
```

Example Workflow – X-Ray U-Net Cell Detection

- A machine-learning workflow for unsupervised cell detection:
- DAG defined by output dependencies specified in workflow CWL.
- `score_format` hints allow for easy tracking of important performance metrics in logs.





```
steps:
    raw_boss_pull:
        run: ../../../boss_access/boss_pull_nos3.cwl
        in:
            ...
        out:
            [pull_output]
    anno_boss_pull:
        run: ../../../boss_access/boss_pull_nos3.cwl
        in:
            ...
        out:
            [pull_output]
    optimize:
        run: ../../tools/membrane_unets_train.cwl
        in:
            ...
        out: [classifier_weights, scores]

    classify:
        run: ../../unets/deploy_unets.cwl
        in:
            ...
        out: [membrane_probability_map]
        hints:
            saber:
                score_format: "F1: {score}"

    cell_detect:
        run: ../../tools/unsup_cell_detect_3D_nos3.cwl
        in:
            ...
        out: [cell_detect_results, dense_output]

    metrics:
        run: ../../tools/unsup_metrics_nos3.cwl
        in:
            ...
        out: [metrics]
        hints:
            saber:
                score_format: "F1: {score}"
```

CWL Parser

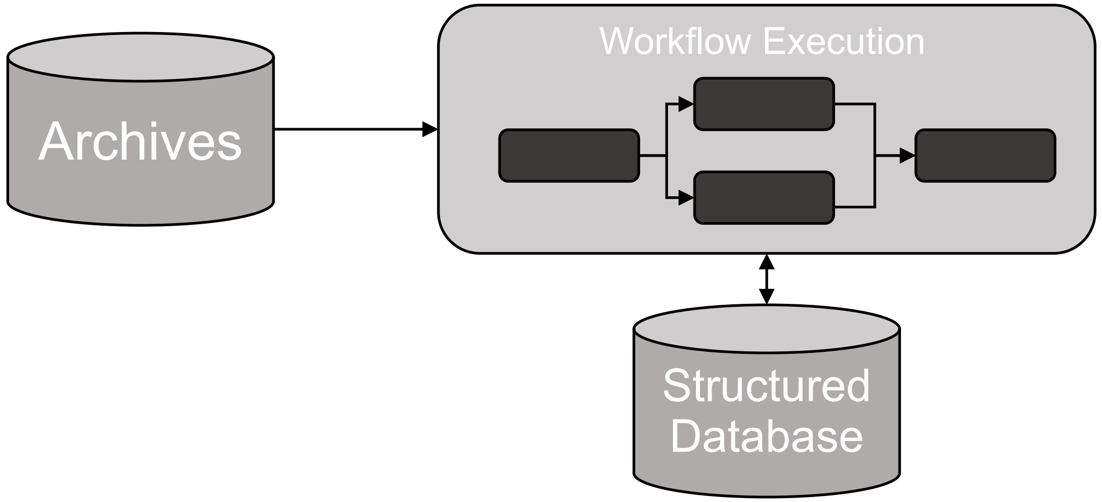
- Python based processor parser
- Part of core SABER docker
- Executed using SABER command line tools
- Similar to CWL-Airflow project for genomics, but with key added features
  - Resource aware (CPU/GPU)
  - Can specify sweeps (in separate yaml parameter file from CWL)

```
▪ x:
range:
    start: 0
    stop: 10000
    step: 10
parameters:
    - xmin
    - xmax
```

- Can specify stdout flags with metrics values, for optimization of pipelines

Next Steps: Integrated Workflow and Data Management

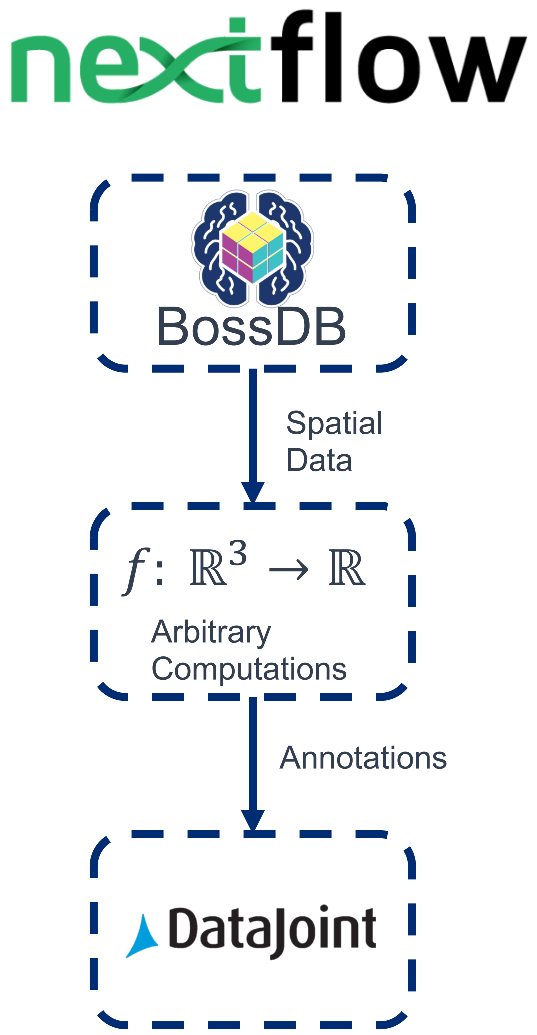
- Benefits of Workflow Managers
  - Scalable execution
  - Capture computation dependencies
  - Manage large numbers of jobs efficiently
- Benefits of Data Management
  - Access to archive
  - Standardized types/formats
  - Structured data storage and dependencies
- How can we combine these benefits?



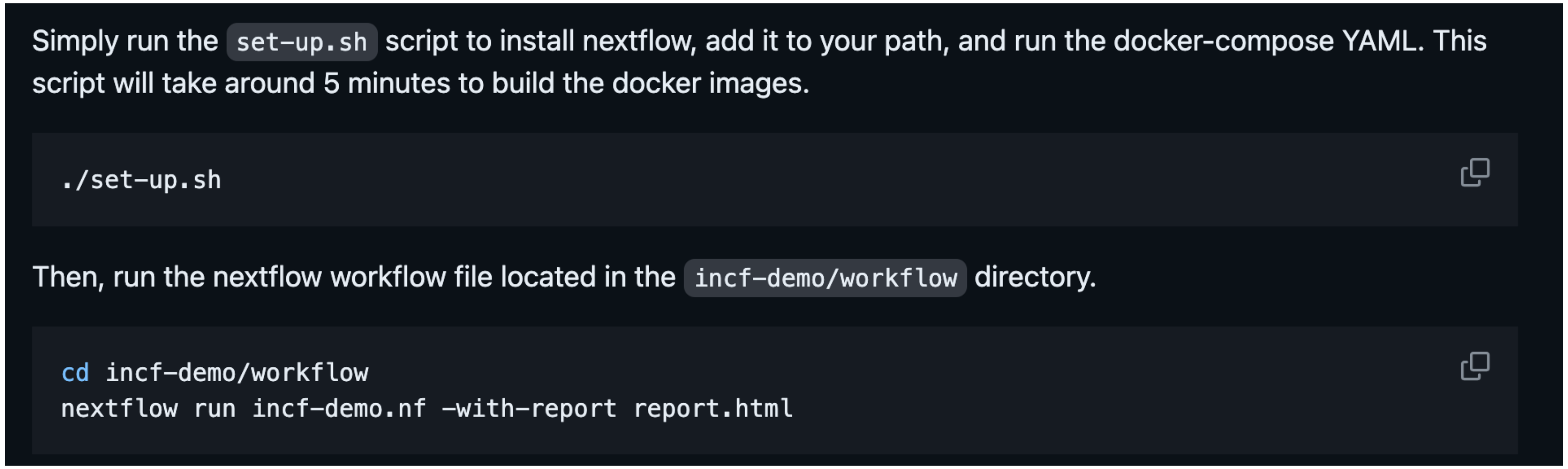
Example: DataJoint + Nextflow

- Low-effort, easy set-up integration of
  - Workflow management
  - Data Management
- **nextflow** provides
  - Task definitions and execution
  - Workflow definition and execution

- https://github.com/aplbrain/neuroworkflows-demo
- https://bossdb.org/



Nextflow Example

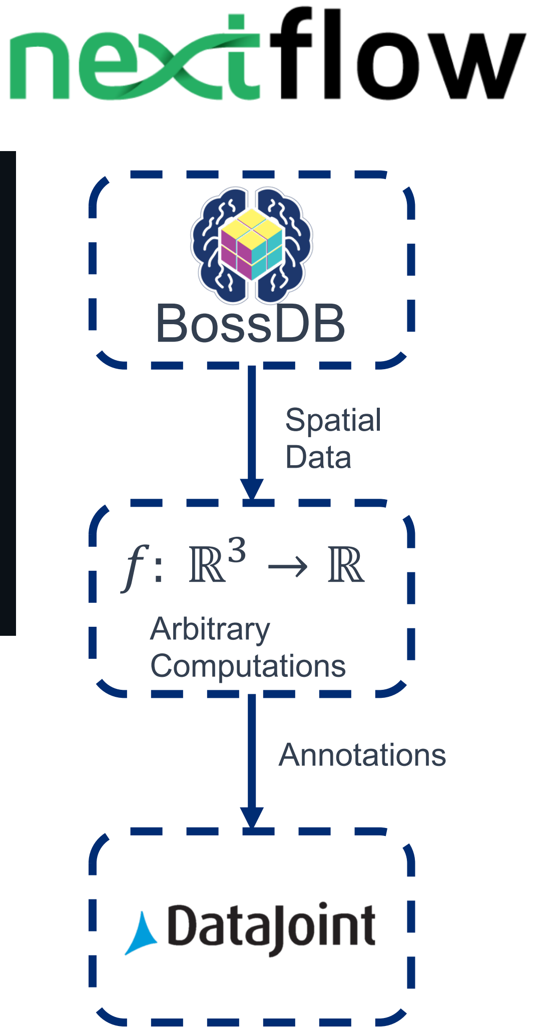


Simply run the `set-up.sh` script to install nextflow, add it to your path, and run the docker-compose YAML. This script will take around 5 minutes to build the docker images.

```
./set-up.sh
```

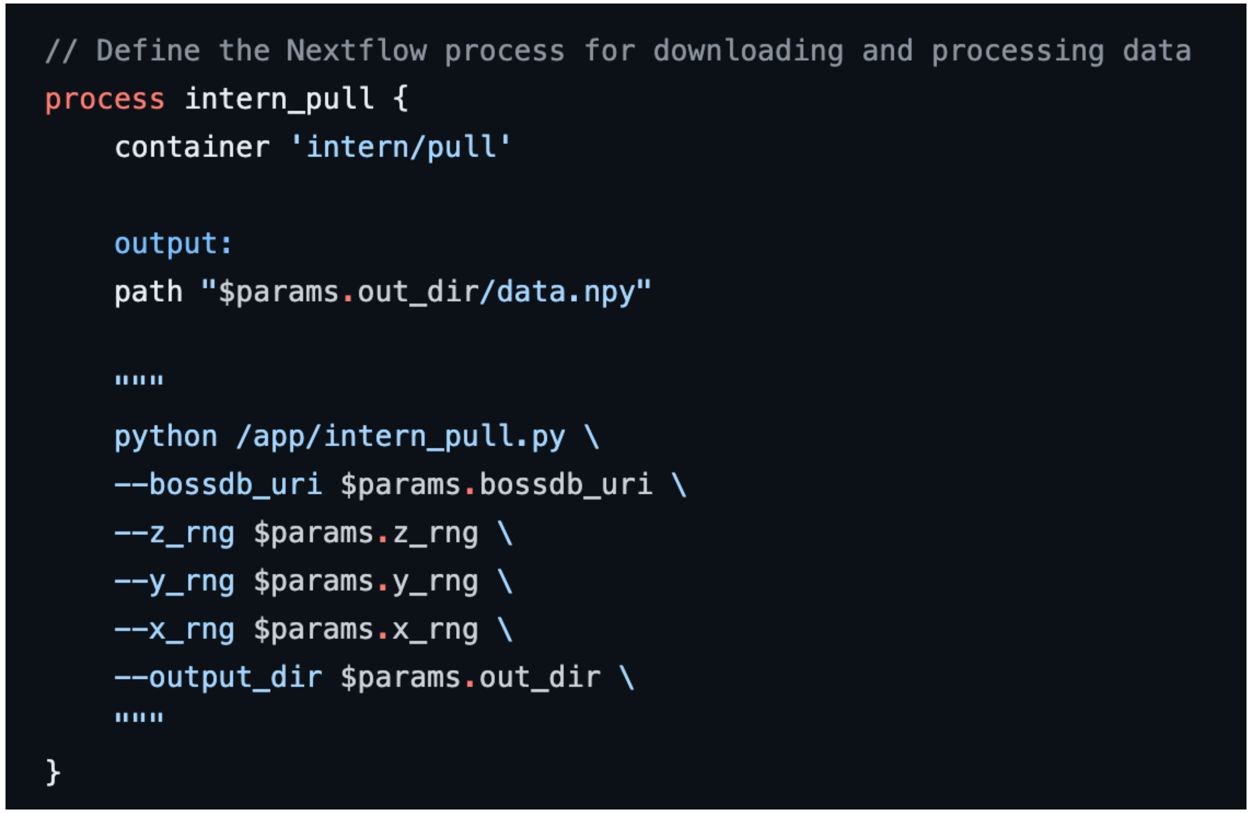
Then, run the nextflow workflow file located in the `incf-demo/workflow` directory.

```
cd incf-demo/workflow
nextflow run incf-demo.nf -with-report report.html
```



- https://github.com/aplbrain/neuroworkflows-demo
- https://bossdb.org/

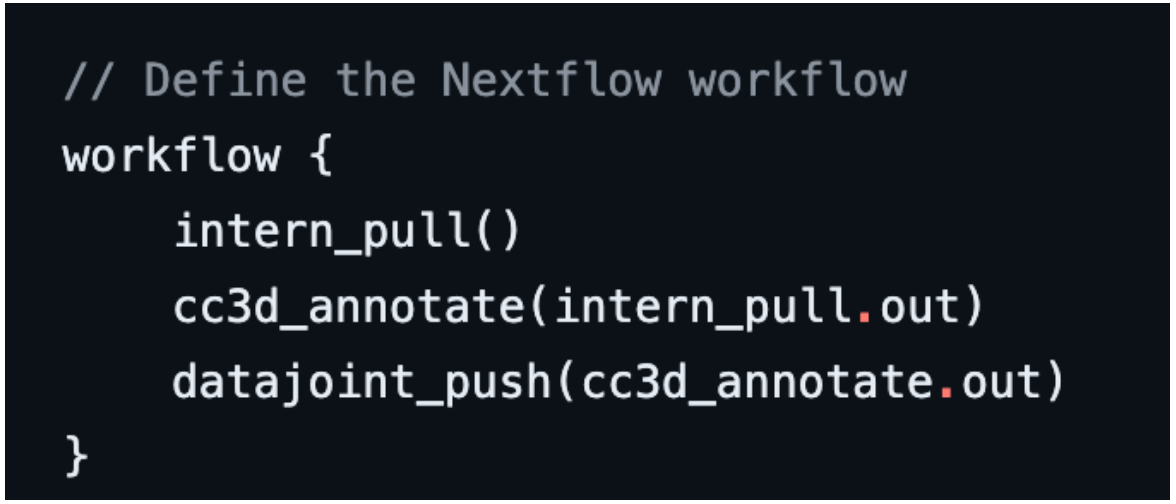
Nextflow Example



```
// Define the Nextflow process for downloading and processing data
process intern_pull {
    container 'intern/pull'

    output:
    path "$params.out_dir/data.npy"

    """
    python /app/intern_pull.py \
    --bossdb_uri $params.bossdb_uri \
    --z_rng $params.z_rng \
    --y_rng $params.y_rng \
    --x_rng $params.x_rng \
    --output_dir $params.out_dir \
    """
}
```



```
// Define the Nextflow workflow
workflow {
    intern_pull()
    cc3d_annotate(intern_pull.out)
    datajoint_push(cc3d_annotate.out)
}
```

Future Directions

- Continue to explore intersection of data and workflow management
- Connect to a range of archives
- Streamline process to create workflows
- Streamline process to create structure data given workflows
- Explore impact for neuroinformatics

Apache Airflow

DataJoint

nextflow

Flyte

Reference Repos:

https://github.com/aplbrain/saber (CWL + Airflow, though a bit out of date!)
https://github.com/aplbrain/neuroworkflows-demo (nextflow)

Team

- Grateful for the support of NIH Grant R24MH114799, R24MH114785, R01MH126684, R44NS129492

Caitlyn Bishop
Christa Cooke
Hannah Cowley
Joseph Downs
Nathan Drenkow
Tim Gion
William Gray-Roncal Ph.D.
Sandy Hider
Mark Hinton
Marisa Hughes, Ph.D.
Justin Joyce
Lindsey Kitchell, Ph.D.
Dean Kleissas
Jordan Matelsky
Sean McDaniel, Ph.D.
Derek Pryor
Devin Ramsden
Elizabeth Reilly, Ph.D.
Corban Rivera, Ph.D.
Jorge Rivera
Patricia Rivlin, Ph.D.
Luis Rodriguez

Kevin Romero
Victoria Rose
Nicole Stock
Marisel Villafañe-Delgado, Ph.D.
Brock Wester, Ph.D.
Perry Wilson
Miller Wilt
Miguel Wimbish
Daniel Xenes

DataJoint
Dimitri Yatsenko
Raphael Guzman
Monty Kosma
Kabilar Gunalan
Drew Yang
Jeroen Verswijver
Georgia Tech
Eva Dyer, Ph.D.
University of Pennsylvania
Konrad Kording, Ph.D.

https://github.com/aplbrain
https://bossdb.org

APL

JOHNS HOPKINS

APPLIED PHYSICS LABORATORY