

Advantages and challenges of using the Galaxy API within an integrated data analysis and visualization platform

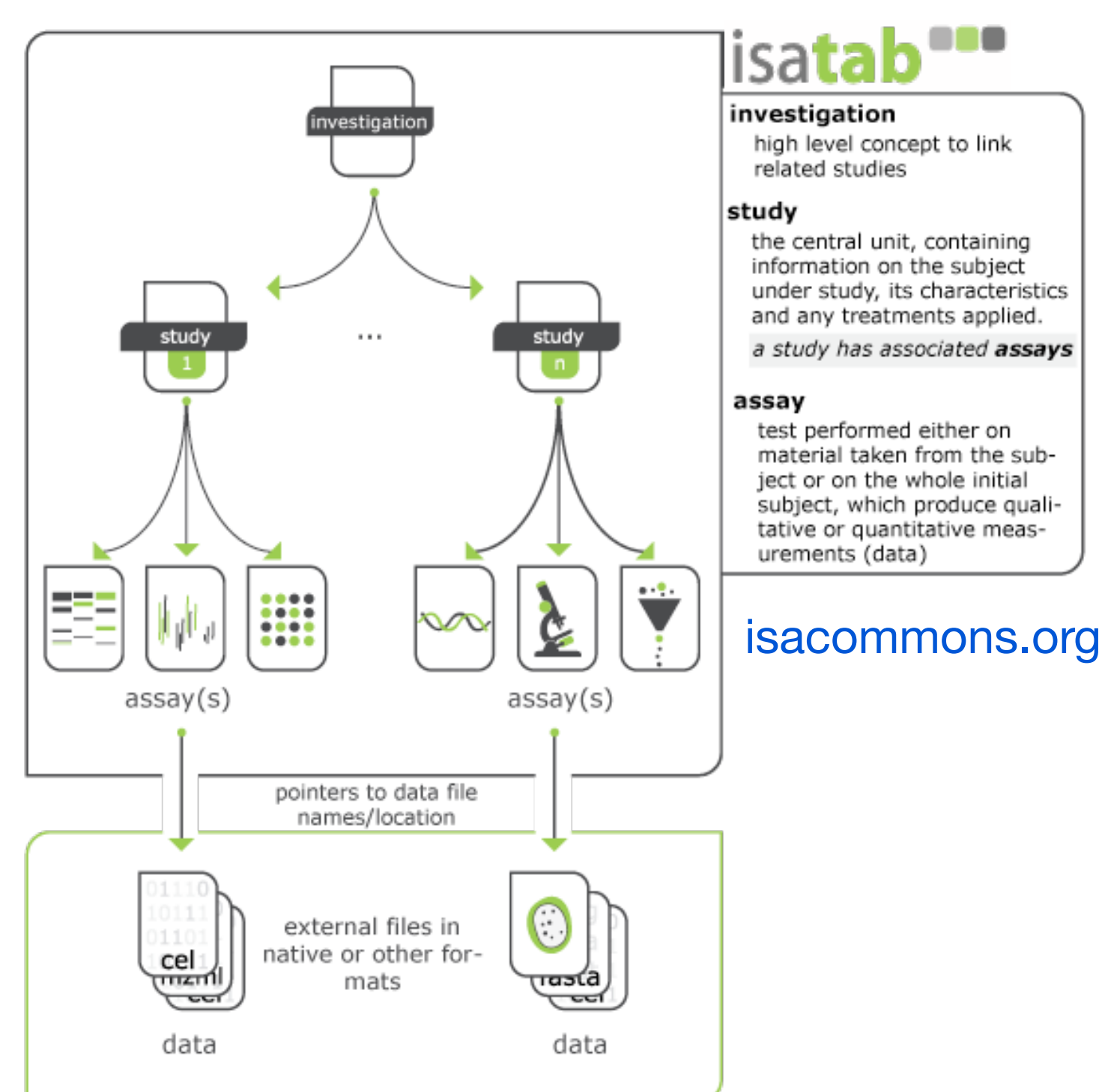
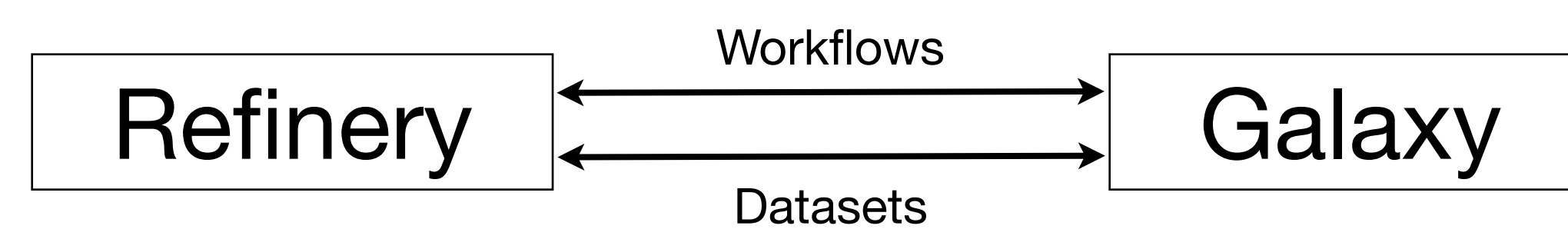
Ilya Sytchev¹, Nils Gehlenborg², Shannan Ho Sui¹, Richard Park^{2,3}, Psalm Haseley², Winston Hide¹, Peter Park²

¹Center for Stem Cell Bioinformatics, Harvard Stem Cell Institute, ²Center for Biomedical Informatics of Harvard Medical School, ³Boston University Bioinformatics Program

Introduction

The **Refinery Platform** (refinery-platform.org) is a web-based data visualization and analysis system powered by an ISA-Tab-compatible data repository for public and private data sets. Analyses are implemented as **Galaxy workflows** and executed through the **Galaxy API** using internally developed code. An instance of Refinery is used by the Stem Cell Commons project (stemcellcommons.org).

BioBlend (bioblend.readthedocs.org) is a **Python** library for interacting with the **Galaxy** and **CloudMan APIs**. Its recent development motivated us to replace our existing custom Galaxy API client code with BioBlend library components.



Refinery: analysis setup user interface

Organism	Age	Stage	Imp Antibody	Strain	Type	Name
Danio rerio	3.5	Embryo	none	TU (Tubingen Zebrafish)	Raw Data File	http://stemcellcommons.org
Danio rerio	3.5	Embryo	c-Myc (binds Myc-tagged-N)	TU (Tubingen Zebrafish)	Raw Data File	http://stemcellcommons.org
Danio rerio	4.5	Embryo	none	c-Myc (binds Myc-tagged-Nanog-like)	Raw Data File	http://stemcellcommons.org
Danio rerio	4.5	Embryo	c-Myc (binds Myc-tagged-M)	TU (Tubingen Zebrafish)	Raw Data File	http://stemcellcommons.org
Danio rerio	4.5	Embryo	none	TU (Tubingen Zebrafish)	Raw Data File	http://stemcellcommons.org

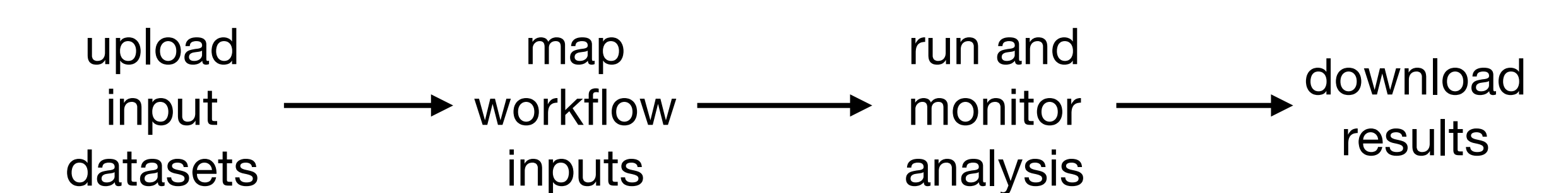
Refinery functionality	BioBlend API methods
import and store workflows as a graph	<code>workflows.get_workflows()</code> <code>workflows.show_workflow()</code>
create a library to store input files	<code>libraries.create_library()</code>
download workflow	<code>workflows.export_workflow_json()</code>
upload expanded workflow	<code>workflows.import_workflow_json()</code>
create a history for workflow outputs	<code>histories.create_history()</code>
add input files to the library	<code>libraries.upload_file_from_local_path()</code>
get expanded workflow information	<code>workflows.show_workflow()</code>
run workflow	<code>workflows.run_workflow()</code>
download results	<i>does not use BioBlend yet</i>
clean up	<code>libraries.delete_library()</code> <code>workflows.delete_workflow()</code> <code>histories.delete_history()</code>
monitor analysis execution	<code>histories.get_status()</code>

Implementation

Refinery is based on **Django** (djangoproject.com) - a free and open source web application framework written in Python.

Galaxy connection information (URL and API key) is stored in a Django model. Its `galaxy_connection()` method returns a `GalaxyInstance` object which is used to interact with the Galaxy API via methods on its class fields (`libraries`, `workflows` and `histories`).

This is a first pass at integration using >10 BioBlend methods to:



Advantages

- Programmatic access to most functionality via the Galaxy API
- BioBlend makes the Galaxy API easy to use from Python applications
- BioBlend is free to use and modify
- Developer community keeps BioBlend up-to-date with Galaxy API changes
- BioBlend has good documentation for basic features

Challenges

Galaxy API

- Fairly low level (REST-style HTTP requests)
- Minimal documentation
- Does not report errors consistently (e.g., HTTP status codes)
- Does not stop a running workflow when its history is deleted

BioBlend

- Only a few types of exceptions available: `ConnectionError`, `DatasetStateException` and `ValueError`
- Documentation incomplete: e.g., unclear which methods raise what exceptions
- Under active development

Acknowledgements

This project is supported by funding from the Harvard Stem Cell Institute and an NIH/NHGRI Pathway to Independence Award award to NG. Special thanks to Yi Zhou and Song Yang (Zon Lab) for pipeline testing, to HSCI researchers for user interface testing, and FAS Research Computing for help running Galaxy instances.



Future work

- Use `BioBlend.objects` (object-oriented interface)
- Allow users to assign parameters to workflow tools
- Use `datasets.download_dataset()` to download analysis results

Contact us

Email: isytchev@hsph.harvard.edu

Documentation: refinery-platform.org

Source code: github.com/parklab/refinery-platform

