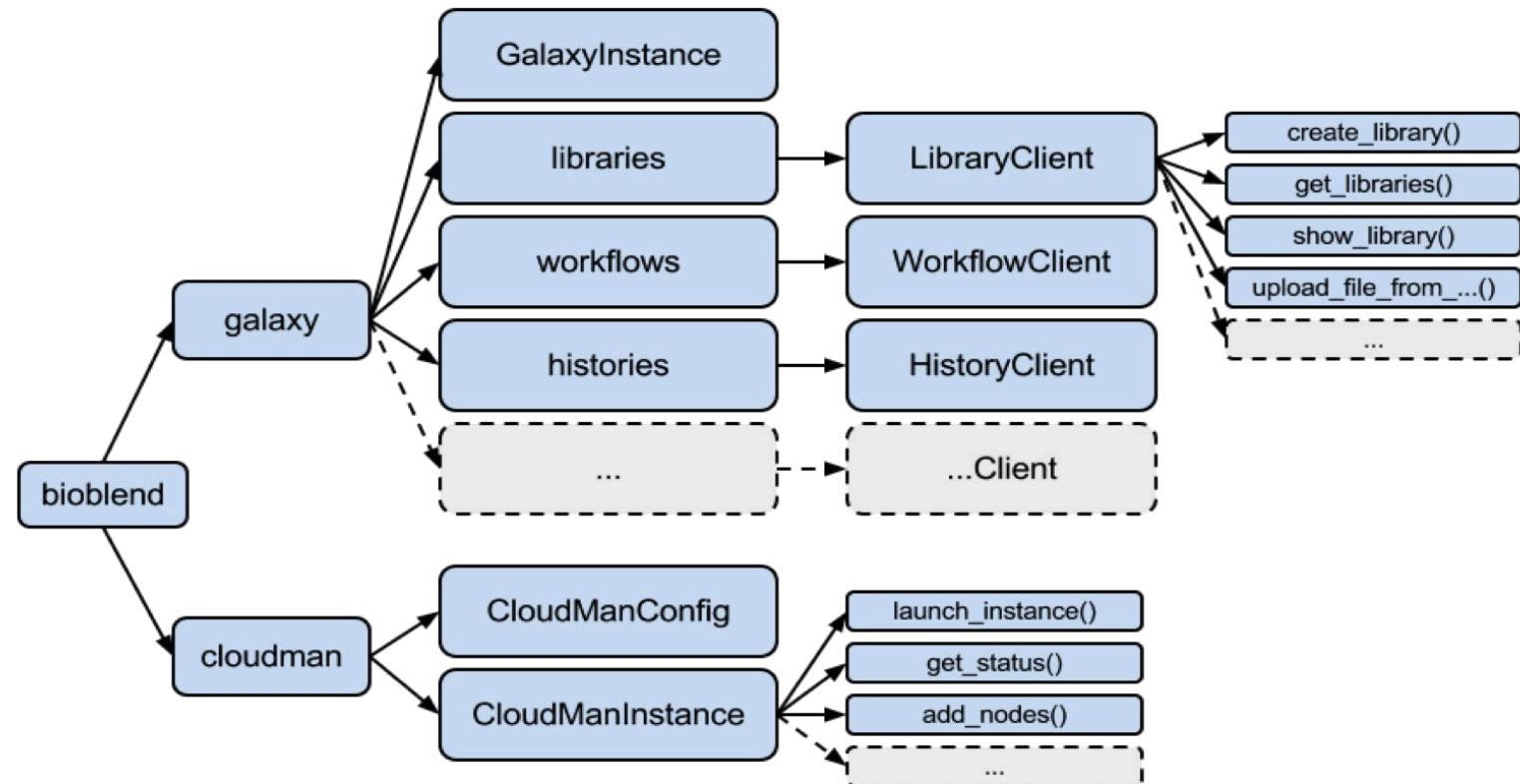


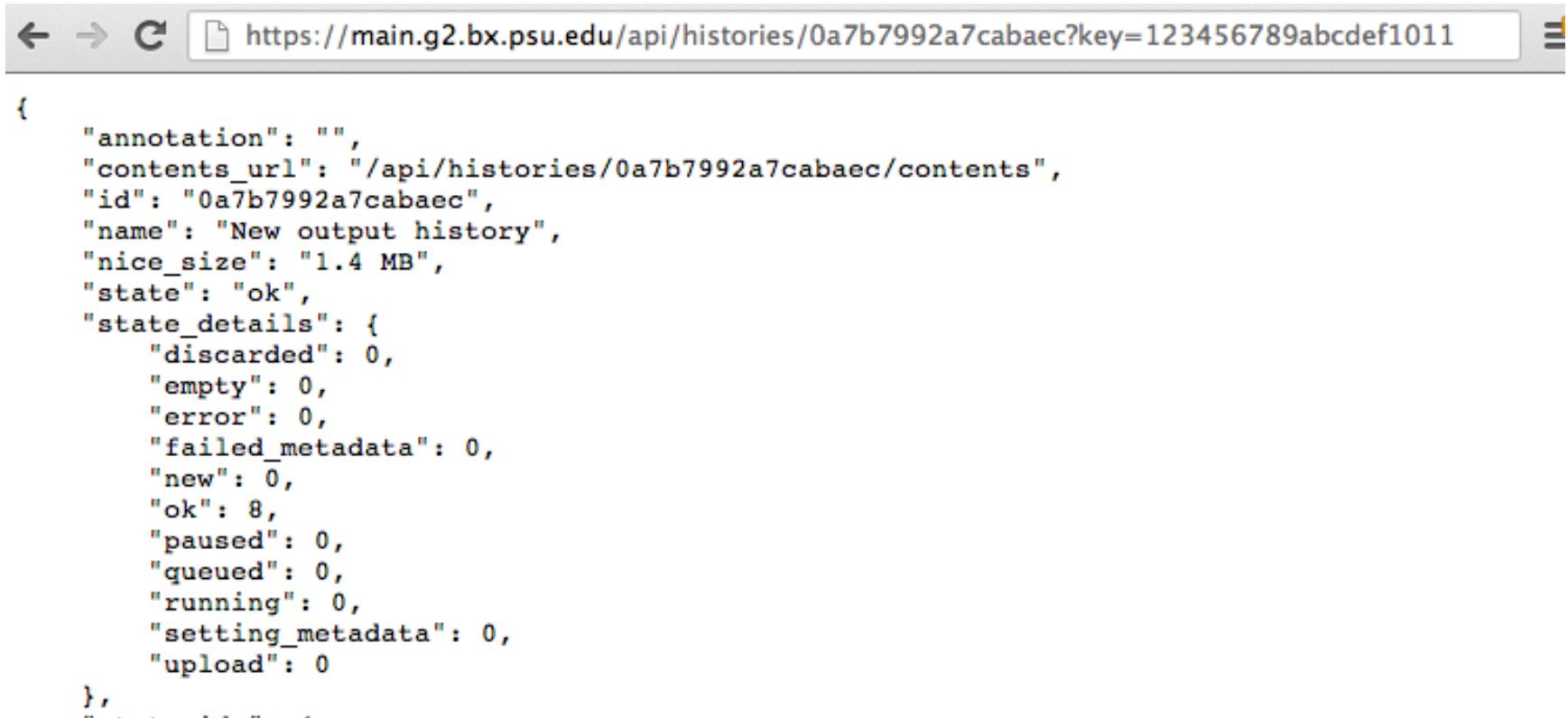
BioBlend: automating analyses with Galaxy and CloudMan

Clare Sloggett, Nuwan Goonasekera,
Enis Afgan

BioBlend is a Python library which wraps the Galaxy API and the CloudMan API (both REST)



Galaxy REST API



A screenshot of a web browser window displaying a JSON response from the Galaxy REST API. The URL in the address bar is `https://main.g2.bx.psu.edu/api/histories/0a7b7992a7cabaec?key=123456789abcdef1011`. The JSON data shows details about a history named "New output history".

```
{  
    "annotation": "",  
    "contents_url": "/api/histories/0a7b7992a7cabaec/contents",  
    "id": "0a7b7992a7cabaec",  
    "name": "New output history",  
    "nice_size": "1.4 MB",  
    "state": "ok",  
    "state_details": {  
        "discarded": 0,  
        "empty": 0,  
        "error": 0,  
        "failed_metadata": 0,  
        "new": 0,  
        "ok": 8,  
        "paused": 0,  
        "queued": 0,  
        "running": 0,  
        "setting_metadata": 0,  
        "upload": 0  
    },  
    "url": "/api/histories/0a7b7992a7cabaec"  
}
```

Galaxy REST API

<http://galaxy-dist.readthedocs.org/>

<http://bitbucket.org/galaxy/galaxy-dist/src>

Admin and end-user functionality, e.g.

- manage users, quotas, roles
- manage Data Libraries
- import and run Workflows
- upload and download data; automate analysis as soon as data is available

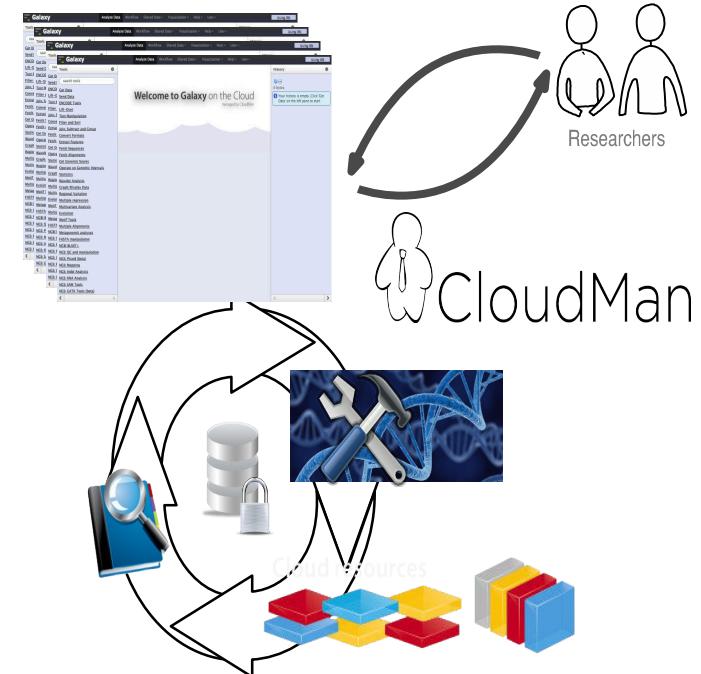
CloudMan

CloudMan is a platform for running a "virtual cluster" in the cloud

- SGE cluster, attached storage
- CloudBioLinux
- Galaxy installed and configured
- Tools and genomes pre-installed

Runs on AWS (Amazon), OpenNebula, Eucalyptus, OpenStack (Australian Research Cloud)

<http://usecloudman.org/>



CloudMan Console

Welcome to [CloudMan](#). This application allows you to manage this instance cloud cluster and the services provided within. Your previous data store has been reconnected. Once the cluster has initialized, use the controls below to manage services provided by the application.

[Terminate cluster](#)[Add nodes ▾](#)[Remove nodes](#)[Access Galaxy](#)

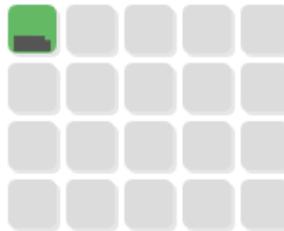
Status

Cluster name: galaxy-dev-vic 

Disk status: 0 / 0 (0%) 

Worker status: Idle: 0 Available: 0 Requested: 0

Service status: Applications  Data 



Autoscaling is **off**.
Turn [on](#)?

Cluster status log

```
05:54:31 - Master starting
05:54:33 - Completed the initial cluster startup process. Configuring a previously existing cluster of type SGE
05:54:37 - SGE service prerequisites OK; starting the service
05:54:44 - Setting up SGE...
05:56:58 - post_start_script found and saved to '/mnt/cm/post_start_script'; running it now (note that this may
take a while)
05:57:27 - Done running post_start_script
05:57:27 - All cluster services started; the cluster is ready for use
```

CloudMan Admin Console

This admin panel is a convenient way to gain insight into the status of individual CloudMan services as well as to control those services.

Services should not be manipulated unless absolutely necessary. Please keep in mind that the actions performed by these service-control 'buttons' are basic in that they assume things will operate as expected. In other words, minimal special case handling for recovering services exists. Also note that clicking on a service action button will initiate the action; there is no additional confirmation required.

Galaxy controls

Use these controls to administer functionality of Galaxy.

- [Access Galaxy](#)
- Current Galaxy admins: system@genome.edu.au
- Add Galaxy admin users What will this do?
- Running Galaxy at revision: [10003:b4a373d86c51](#)
- Update Galaxy from a provided repository What will this do?

Services controls

Use these controls to administer individual application services managed by CloudMan. Currently running a '[Galaxy](#)' type of cluster.

Service name	Status	Log	Stop	Start	Restart	Update DB
Galaxy	Running	Log	Stop	Start	Restart	Update DB
PostgreSQL	Running	Log	Stop	Start	Restart	
SGE	Running	Log	Stop	Start	Restart	Q conf
Galaxy Reports		Log	Stop	Start	Restart	qstat

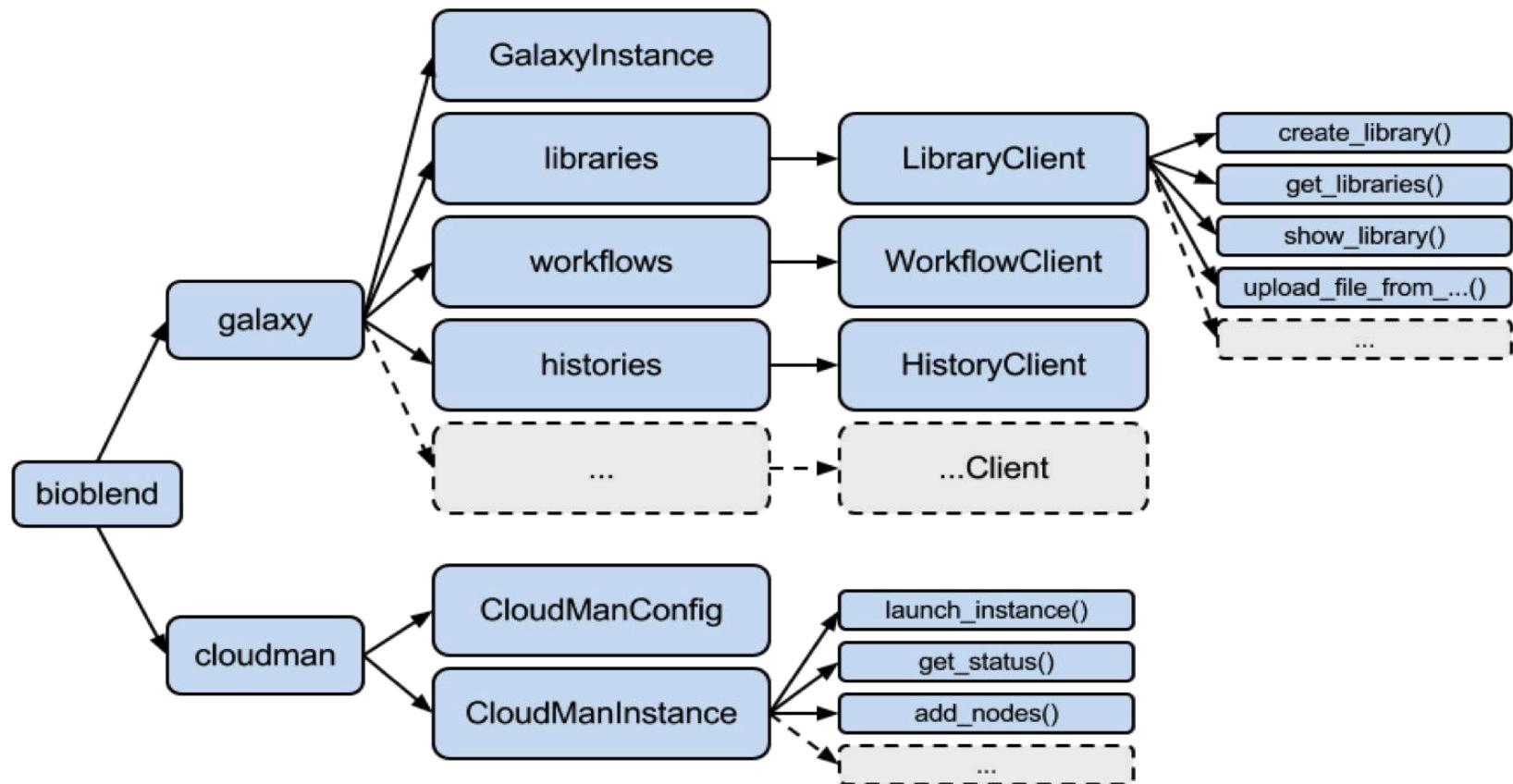
File systems

Name	Status	Usage	Controls
transient_nfs	Running		X Details
galaxy	Running		X  Details
galaxyIndices	Running		X  Details

[+ Add new](#)

System controls

Use these controls to administer CloudMan itself as well as the underlying system.



e.g. REST call:

```
http://main.g2.bx.psu.edu/api/history/0a7b7992a7cabaec?  
key=123456789abcdef10
```

Returns JSON:

```
{ 'id': '0a7b7992a7cabaec',  
  'name': 'Output history',  
  'state_details': { 'discarded': 0,  
                    'empty': 0, .....}
```

in BioBlend becomes Python:

```
GalaxyInstance.histories  
    .show_history('0a7b7992a7cabaec')
```

Returns corresponding Python dictionary

```
{ 'id': '0a7b7992a7cabaec',  
  'name': 'Output history',  
  'state_details': { 'discarded': 0,  
                    'empty': 0, .....}
```

http://bioblend.readthedocs.org/

The screenshot shows a web browser displaying the BioBlend 0.2.3-dev documentation at <http://bioblend.readthedocs.org/en/latest/>. The page has a dark header bar with the text "BioBlend 0.2.3-dev documentation >". On the left, there's a sidebar with "Project Versions" set to "latest" and an "RTD Search" input field. Below that is a "Table Of Contents" section listing various BioBlend modules and their sub-topics. Under "BioBlend", the listed topics are: About, Installation, Usage, Development, API Documentation, CloudMan API, Galaxy API, Configuration, Testing, Getting help, Related documentation, and Indices and tables. There are also "Next topic" and "This Page" sections with links to "API documentation for interact" and "Show Source", "Show on GitHub", and "Edit on GitHub". The main content area is titled "BioBlend" and contains an "About" section. It describes BioBlend as a Python library for interacting with CloudMan and Galaxy's API. It highlights its ability to script and automate cloud infrastructure via Galaxy. Three bullet points follow, each with a code snippet:

- Create a CloudMan compute cluster, via an API and directly from your local machine:

```
from bioblend.cloudman import CloudManConfig
from bioblend.cloudman import CloudManInstance
cfg = CloudManConfig('<your cloud access key>', '<your cloud secret key>')
cmi = CloudManInstance.launch_instance(cfg)
cmi.get_status()
```
- Reconnect to an existing CloudMan instance and manipulate it:

```
from bioblend.cloudman import CloudManInstance
cmi = CloudManInstance("<instance IP>", "<password>")
cmi.add_nodes(3)
cluster_status = cmi.get_status()
cmi.remove_nodes(2)
```
- Interact with Galaxy via a straightforward API:

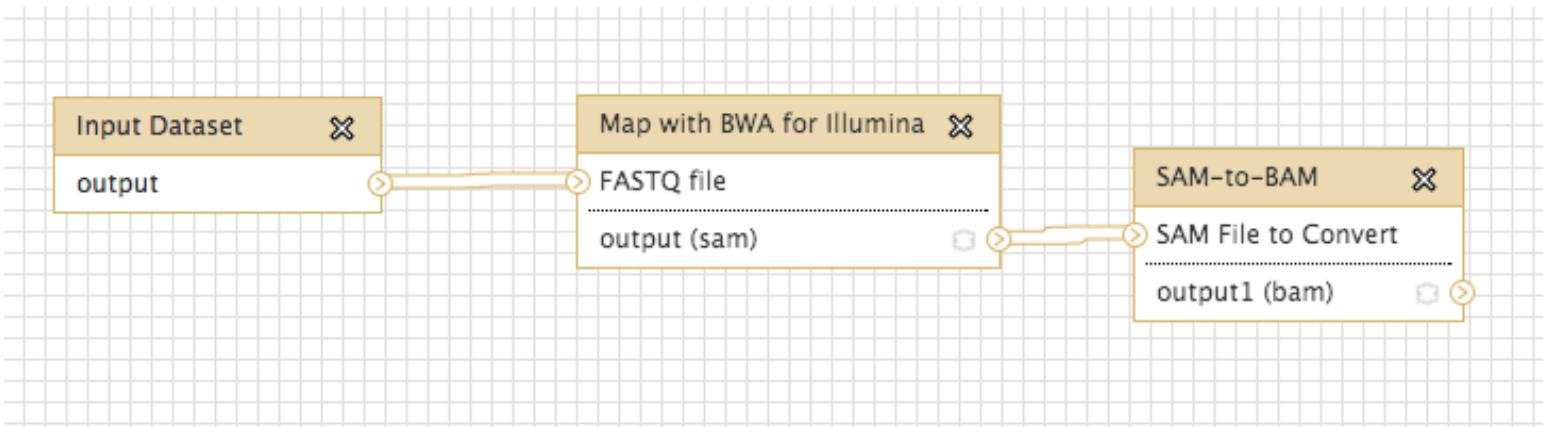
```
from bioblend.galaxy import GalaxyInstance
gi = GalaxyInstance('<Galaxy IP>', key='your API key')
libs = gi.libraries.get_libraries()
gi.workflows.show_workflow('workflow ID')
gi.workflows.run_workflow('workflow ID', input_dataset_map)
```

Scalable workflows

For some workflows the GUI batch mode isn't enough, because we need to know which file is which.

To make the workflow automated AND scalable, we need to base the logic on metadata fields like sample, patient, or experimental run.

Simple: single-end mapping to reference

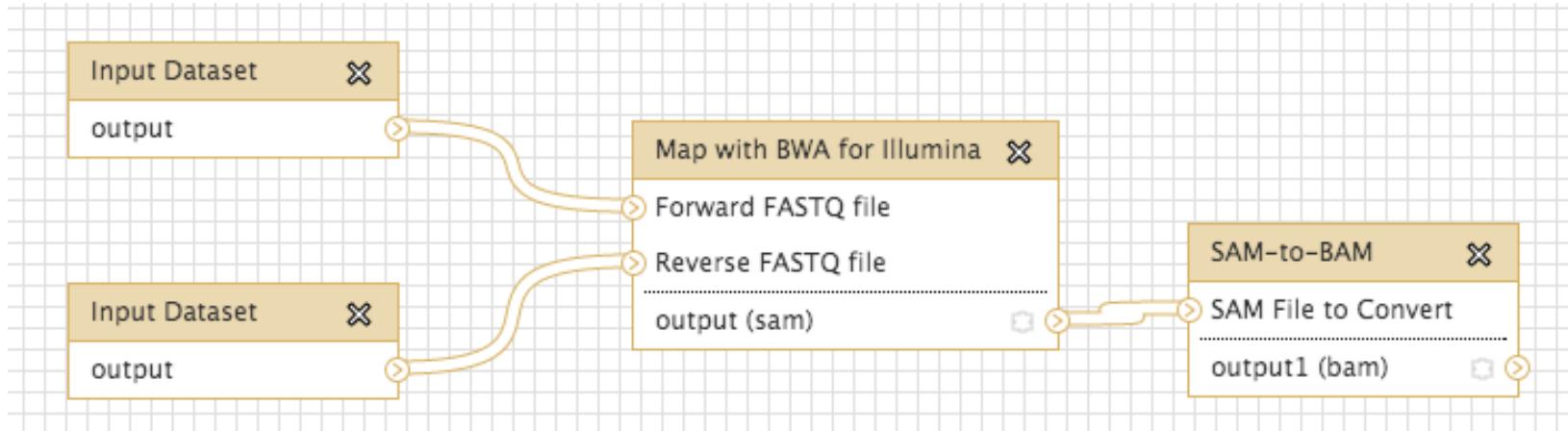


Sample1.fastq → Sample1.bam

Sample2.fastq → Sample2.bam

Sample3.fastq → Sample3.bam

Less simple: paired-end mapping to reference



Sample1_R1.fastq → Sample1.bam

Sample1_R2.fastq → Sample1.bam

Sample2_R1.fastq → Sample2.bam

Sample2_R2.fastq → Sample2.bam

Contribute!

```
class HistoryClient(Client):
    def __init__(self, galaxy_instance):
        self.module = 'histories'
        super(HistoryClient, self).__init__(galaxy_instance)

    def show_history(self, history_id, contents=False):
        """
        Get details of a given history. By default, just get the
        history meta information. If ``contents`` is set to ``True``,
        get the complete list of datasets in the given history.
        """
        return Client._get(self, id=history_id, contents=contents)
```

<http://bioblend.readthedocs.org/>

<https://github.com/afgane/bioblend>

<https://pypi.python.org/pypi/bioblend>

blend4j: <https://github.com/jmchilton/blend4j>

clj-blend: <https://github.com/chapmanb/clj-blend>

Thanks to...

