Using Python code and Dataframes

UC San Diego

BROAD INSTITUTE

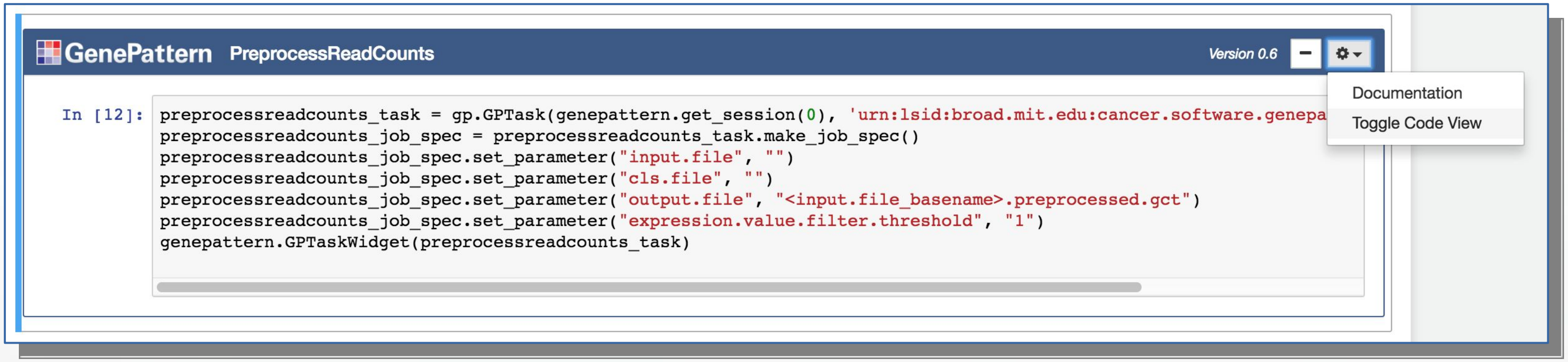
GenePattern Python Library

- Complete programmatic access
- Automatic integration with GenePattern cell data

```
import gp

# Create a GenePattern server proxy instance
gpserver = gp.GPServer('http://localhost:8080/gp','myusername', 'mypassword')

# Obtain GPTask by module name
module = gp.GPTask(gpserver, "PreprocessDataset")

# Load module parameter data
module.param_load()

# Create a job specification
job_spec = module.make_job_spec()

# Upload a file to the server
uploaded_file = gpserver.upload_file("file_name", "/path/to/the/file/file_name")
job_spec.set_parameter("input.filename", uploaded_file.get_url())

# Submit the job to the GenePattern server
job = gpserver.run_job(job_spec)
```

Toggle Code View

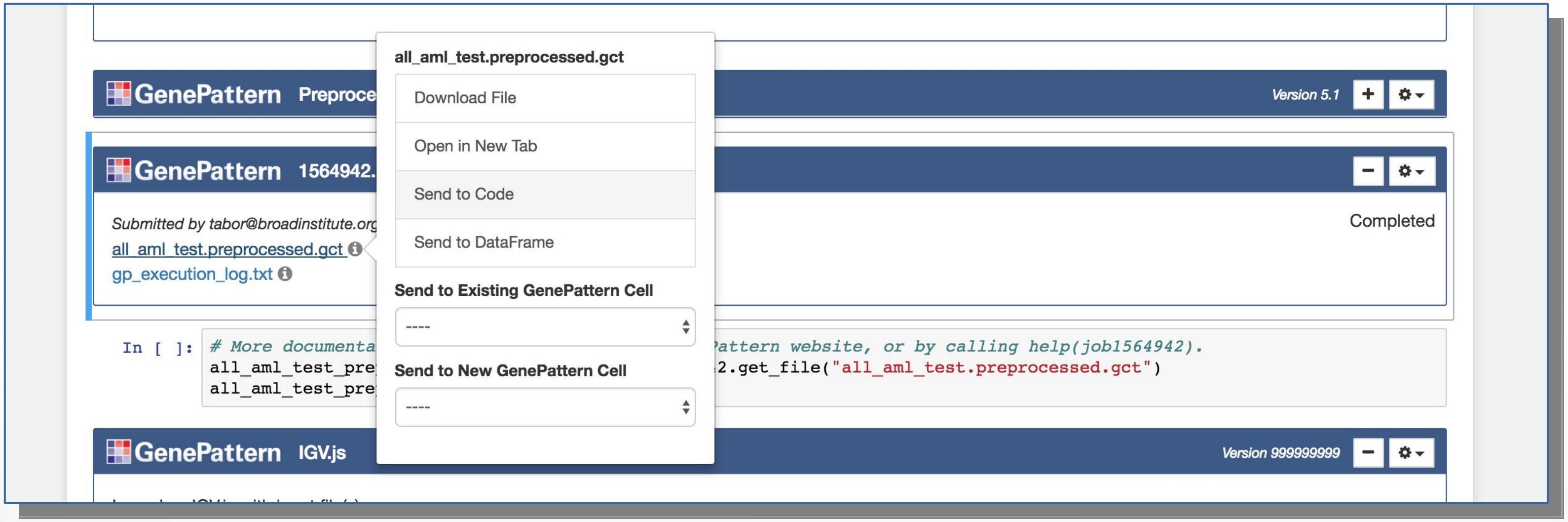
- Under the cover, all GenePattern cells use the Python library.
- Can view the code by selecting “Toggle Code View” in the gear menu in the upper right.



```
In [12]: preprocessreadcounts_task = gp.GPTask(genepattern.get_session(0), 'urn:lsid:broad.mit.edu:cancer.software.genepa
         preprocessreadcounts_job_spec = preprocessreadcounts_task.make_job_spec()
         preprocessreadcounts_job_spec.set_parameter("input.file", "")
         preprocessreadcounts_job_spec.set_parameter("cls.file", "")
         preprocessreadcounts_job_spec.set_parameter("output.file", "<input.file_basename>.preprocessed.gct")
         preprocessreadcounts_job_spec.set_parameter("expression.value.filter.threshold", "1")
         genepattern.GPTaskWidget(preprocessreadcounts_task)
```

Send to Code

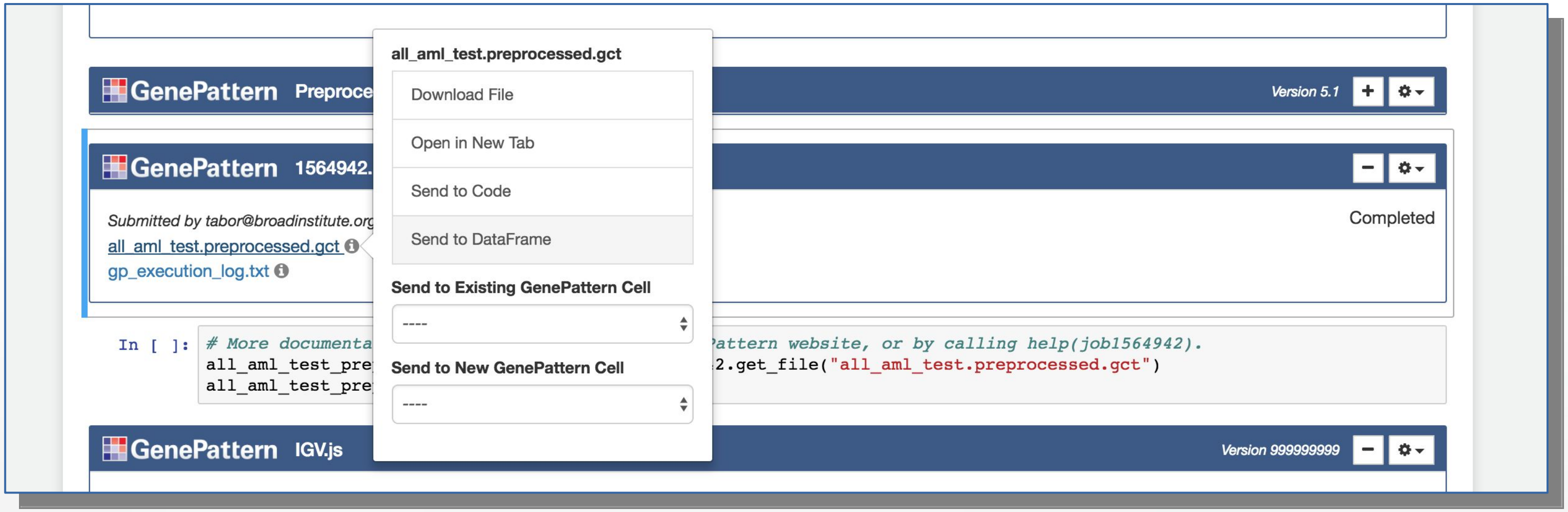
- Example code showing how to programmatically access a GenePattern job result can be generated by clicking on the file and selecting “Send to Code.”



```
In [ ]: # More documenta                                attern website, or by calling help(job1564942).
        all_aml_test_pre                                2.get_file("all_aml_test.preprocessed.gct")
        all_aml_test_pre
```

Send to Dataframe

- Certain GenePattern file formats can be automatically imported into a pandas Dataframe by selecting “Send to Dataframe” in the menu.
- Pandas is a popular Python library for working with data.



Submitted by tabor@broadinstitute.org

all_aml_test.preprocessed.gct

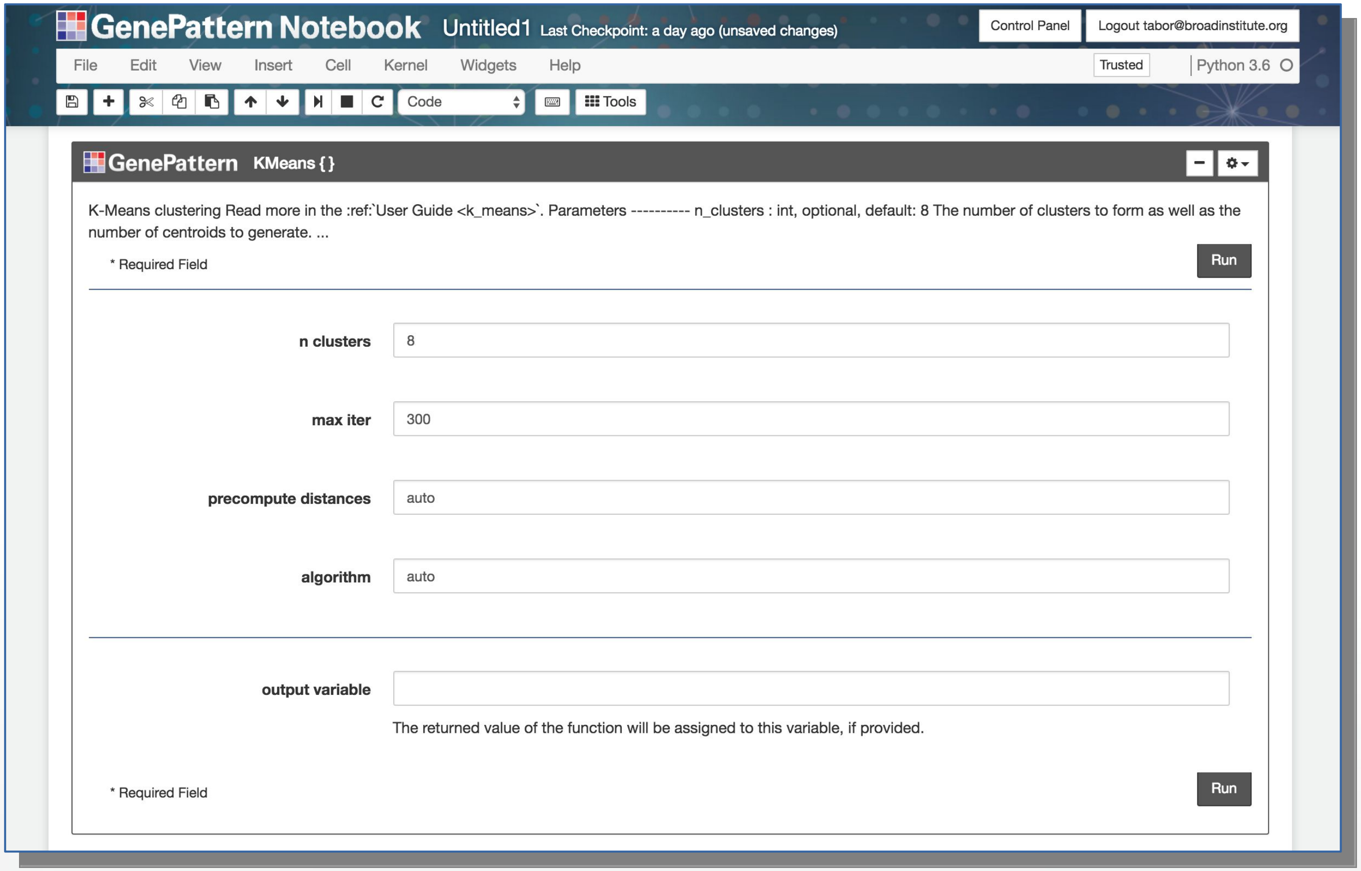
gp_execution_log.txt

Completed

```
In [ ]: # More documenta                                  Pattern website, or by calling help(job1564942).
        all_aml_test_pre                                  2.get_file("all_aml_test.preprocessed.gct")
        all_aml_test_pre
```

UI Builder

- Turns any Python function into an interactive GenePattern widget.



K-Means clustering Read more in the :ref:`User Guide <k_means>`. Parameters ---------- n_clusters : int, optional, default: 8 The number of clusters to form as well as the number of centroids to generate. ...

* Required Field

n clusters	8
max iter	300
precompute distances	auto
algorithm	auto
output variable	

The returned value of the function will be assigned to this variable, if provided.

* Required Field

UI Builder - continued

- Functions created with the UI Builder appear in the tool menu

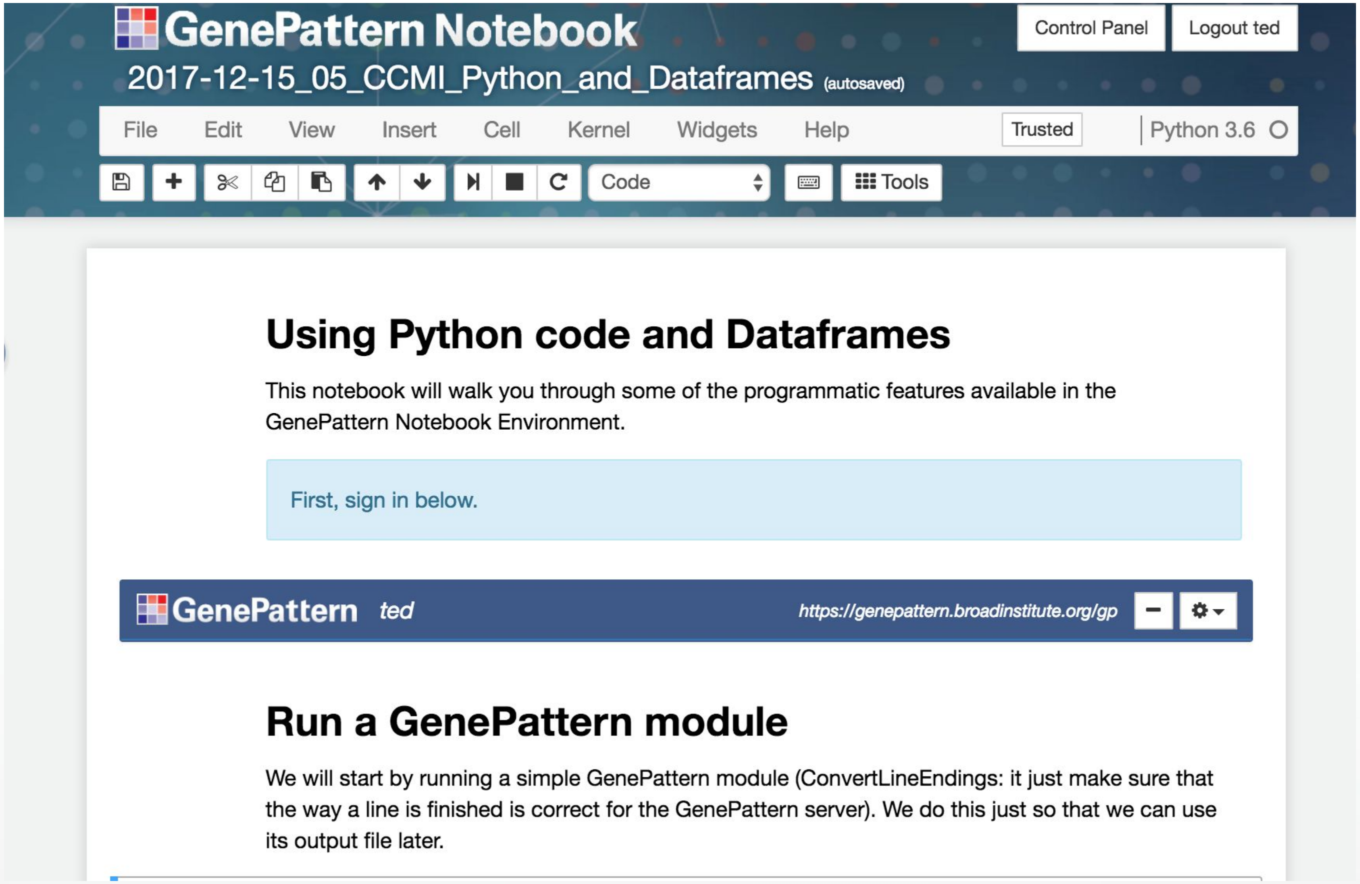
- UI parameters can be given types.
  - text, number, password, choice, boolean and file
  - for a **“file”**, it becomes available via the ‘Send to’ menu from GenePattern cells

- UIBuilder cell inputs (and all GenePattern cell inputs) can accept Python variable names as inputs

Follow along...

Copy and open the public notebook:

2017-12-15_05_CCMI_Python_and_Dataframes


GenePattern Notebook
Control Panel
Logout ted
2017-12-15_05_CCMI_Python_and_Dataframes (autosaved)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3.6
Code
Tools

Using Python code and Dataframes

This notebook will walk you through some of the programmatic features available in the GenePattern Notebook Environment.

First, sign in below.

GenePattern ted https://genepattern.broadinstitute.org/gp

Run a GenePattern module

We will start by running a simple GenePattern module (ConvertLineEndings: it just make sure that the way a line is finished is correct for the GenePattern server). We do this just so that we can use its output file later.