

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()

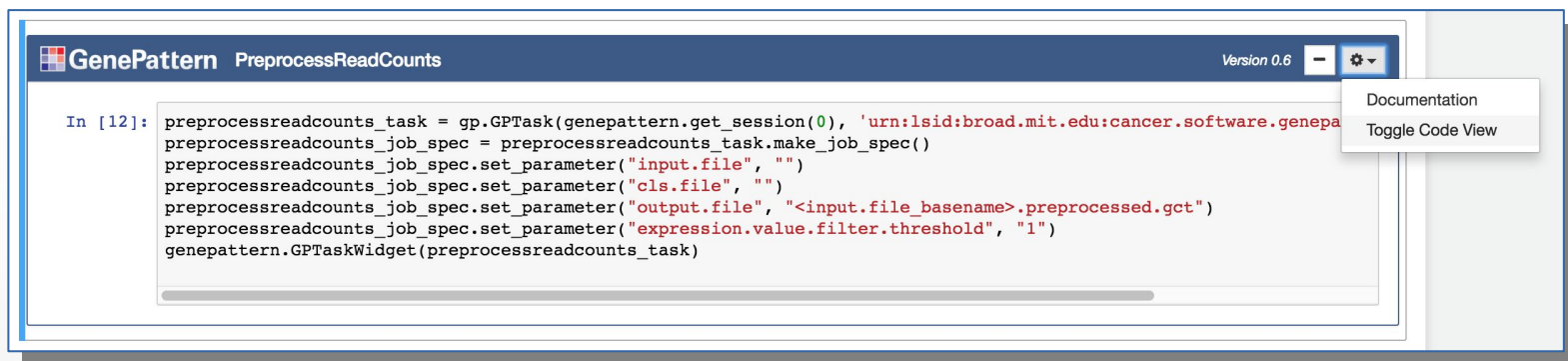
# Upload a file to the server
uploaded_file = gpserver.upload_file("file_name", "/path/to/the/file/file_name")
job_spec.set_parameter("input.filename", uploaded_file.get_url())

# 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.





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.”

The screenshot displays the GenePattern web interface. A context menu is open over a file named `all_aml_test.preprocessed.gct`. The menu options are:

- Download File
- Open in New Tab
- Send to Code (highlighted)
- Send to DataFrame
- Send to Existing GenePattern Cell
- Send to New GenePattern Cell

The background interface shows a job execution log with the following entries:

```
Submitted by tabor@broadinstitute.org  
all_aml_test.preprocessed.gct  
gp_execution_log.txt
```

Below the log, there is a code editor with the following code snippet:

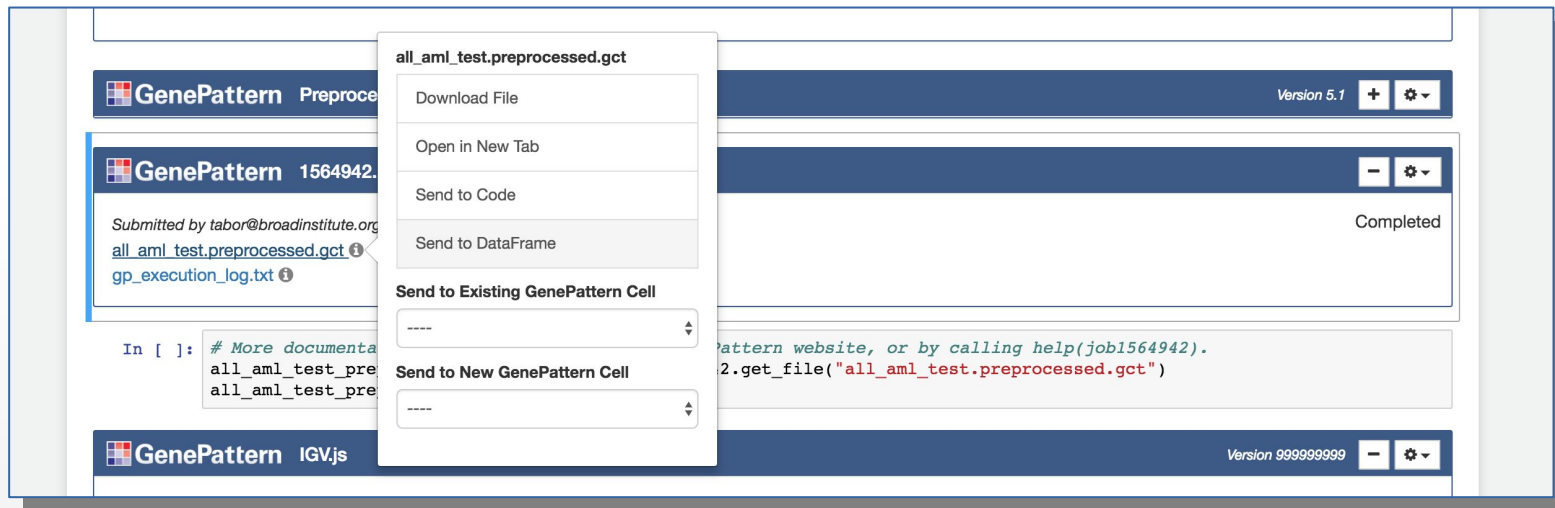
```
In [ ]: # More documenta  
all_aml_test_pre  
all_aml_test_pre
```

The interface also shows a job execution status bar with the text "Completed" and a version number "Version 5.1".



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.





UI Builder

- Turns any Python function into an interactive GenePattern widget.

The screenshot displays the GenePattern Notebook interface. At the top, the title bar reads "GenePattern Notebook" followed by "Untitled1" and "Last Checkpoint: a day ago (unsaved changes)". On the right of the title bar are "Control Panel" and "Logout tabor@broadinstitute.org". Below the title bar is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". To the right of the menu bar are "Trusted" and "Python 3.6". Below the menu bar is a toolbar with icons for file operations, navigation, and execution, along with a "Code" dropdown and a "Tools" button.

The main content area shows a widget titled "GenePattern KMeans {}". Below the title bar, there is a description: "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. ...". A "* Required Field" label is present. Below the description is a "Run" button.

The widget has four input fields:

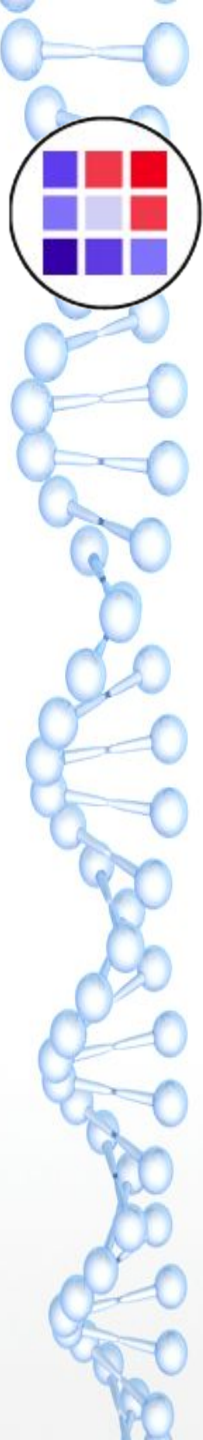
- n_clusters**: A text input field with the value "8".
- max iter**: A text input field with the value "300".
- precompute distances**: A text input field with the value "auto".
- algorithm**: A text input field with the value "auto".

Below these fields is an "output variable" text input field. Below the output variable field is the text: "The returned value of the function will be assigned to this variable, if provided." At the bottom left is a "* Required Field" label, and at the bottom right is a "Run" button.



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

The screenshot displays the GenePattern Notebook web interface. At the top, the header includes the GenePattern logo, the notebook title "2017-12-15_05_CCMI_Python_and_Dataframes (autosaved)", and buttons for "Control Panel" and "Logout ted". Below the header is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. A "Trusted" status indicator and "Python 3.6" version are also shown. A toolbar contains icons for file operations, navigation, and execution. The main content area features the title "Using Python code and Dataframes" and a paragraph: "This notebook will walk you through some of the programmatic features available in the GenePattern Notebook Environment." A light blue box contains the text "First, sign in below." At the bottom, a dark blue footer bar shows the GenePattern logo, the user name "ted", the URL "https://genepattern.broadinstitute.org/gp", and window controls.

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.

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.