



# Basic Single Cell Lab

Jenny Drnevich, PhD

Assistant Director, HPCBio

Roy J. Carver Biotechnology Center

June 26, 2024



# Learning objectives

1. Examine 10x Genomics's web\_summary.html report from the output of Cell Ranger
2. Walk through the steps of a standard Seurat analysis
  1. Read explanations of why each step done
  2. Run codes to do each step
  3. Explore how to interact with a Seurat object
3. Try both manual and computational cell type calling
4. Get links for examples of more complex single cell and spatial analyses



## Web\_summary.html file

This public one is from [cellranger multi](#) not [cellranger count](#) ([example](#)) but all the same information is in there:

**Cells** tab on left:

1. Any warnings at top?
2. Cell metrics:
  1. **# called** - If >> targeted, high background. If << targeted, poor-quality cells
  2. **mean reads** - If >> targeted, few good cells. If << targeted,
  3. **median genes** - variable between tissue/cell types but ideally > 1000
  4. **total genes** - summed over all genes; 20K+ for mammalian cells.
  5. **median UMIs** - compare with mean reads (PCR dups) and median genes (shows ~UMIs per gene)
  6. **% mapped reads in cells** - if >> 90% can indicate ambient background RNA
3. t-SNE
  1. On left shaded by UMI counts and show strong correlation due to no normalization
  2. right shows first clusters



## Web\_summary.html file

This public one is from [cellranger multi](#) not [cellranger count](#) but all the same information is in there:

**Library** tab on left:

1. Cell Statistics (same as Cells tab)
2. Sequencing and Mapping metrics: click on ? for explanations
3. Metrics Per Physical Library
  1. **Sequencing saturation** - "The fraction of reads originating from an already-observed UMI." Indicates whether more sequencing would result in substantially more reads.
4. Plots
  1. **GEX Barcode Rank Plot** - should see sharp "cliff"
  2. **Sequencing Saturation & Median Gene per Cell** - show downsampled values to observe curve and extrapolate what extra sequencing would gain.



# Log on to Biocluster's [Jupyter Hub](#)

Server options:

Classroom partition

Runtime: 6:00

Other defaults fine

Click "Start"

The screenshot shows a web browser window with the URL `bioapps3.igb.illinois.edu/jupyter/hub/spawn`. The page title is "Server Options". The form contains the following fields:

- Select a partition:** A dropdown menu with "classroom (private)" selected.
- Specify runtime (HHH:MM:SS format, 120hr max):** A text input field containing "006:00:00".
- Specify Number of CPUs/Cores:** A text input field containing "1".
- Specify Memory (GBs):** A text input field containing "15".
- Specify Number of GPUs:** A text input field containing "0".

At the bottom of the form is a large orange button labeled "Start".



# Load Mayo module

1. Click on blue hexagon
2. Search for Mayo at top
3. Click on "Load" by Mayo module at bottom.

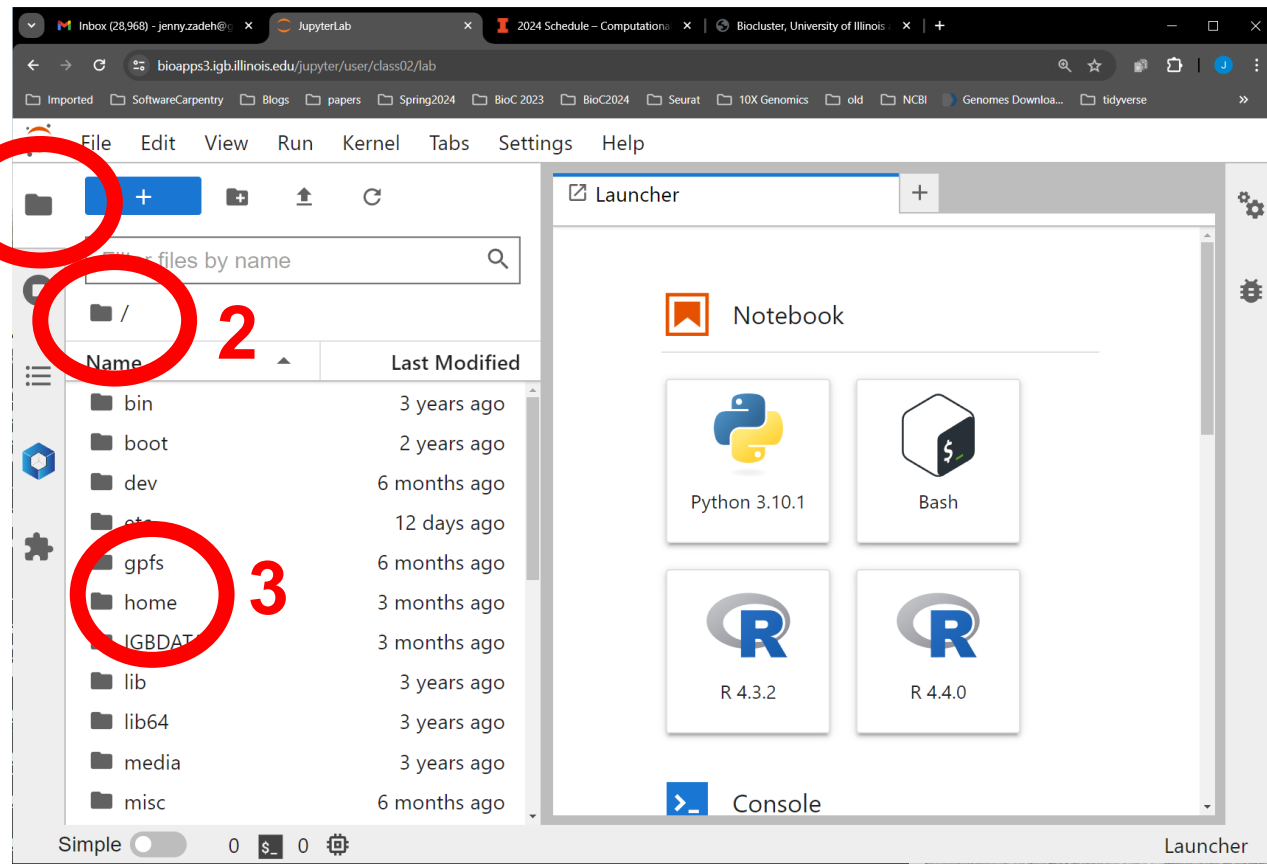
The screenshot shows the JupyterLab interface with the following elements:

- Search Bar:** Contains the text "mayo" (highlighted with a red circle and the number 2).
- Loaded Modules:** A list of currently loaded modules including OpenMPI, tar, bzip2, XZ, MariaDB, OpenSSL, Python, nodejs, pandoc, and jupyterlab.
- Available Modules:** A list of available modules, with "Mayo/2024-IGB-gcc-8.2.0" highlighted (highlighted with a red circle and the number 1).
- Load Button:** A button labeled "Load" next to the Mayo module (highlighted with a red circle and the number 3).
- Launcher:** A panel on the right showing the current directory "home/a-m/class02/JennyTest" and available environments: Notebook, Python 3.10.1, Bash, R 4.3.2, and R 4.4.0.



# Navigate to proper directory

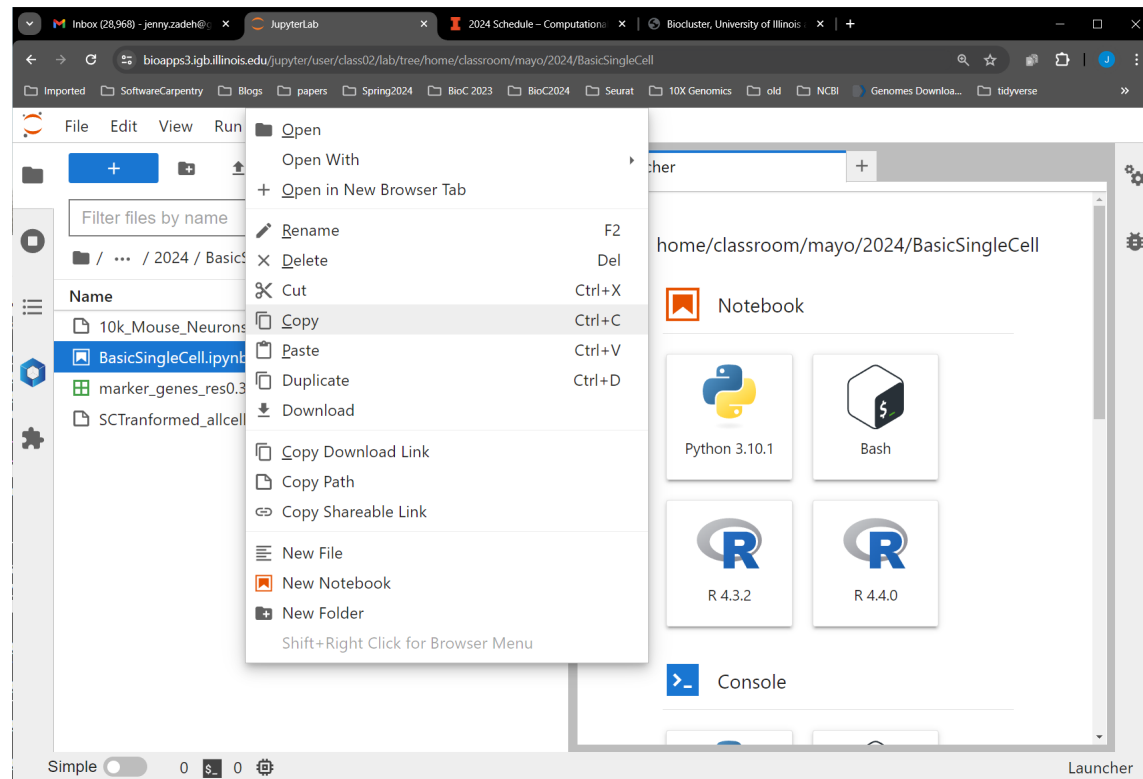
1. Click on gray folder icon on very left
2. Click on second gray folder under search box to get to root directory
3. Double click on "home" directory





## Continue navigating by double clicking:

- Classroom -> mayo -> 2024 -> BasicSingleCell
- Right click on "BasicSingleCell.ipynb" and click "Copy"

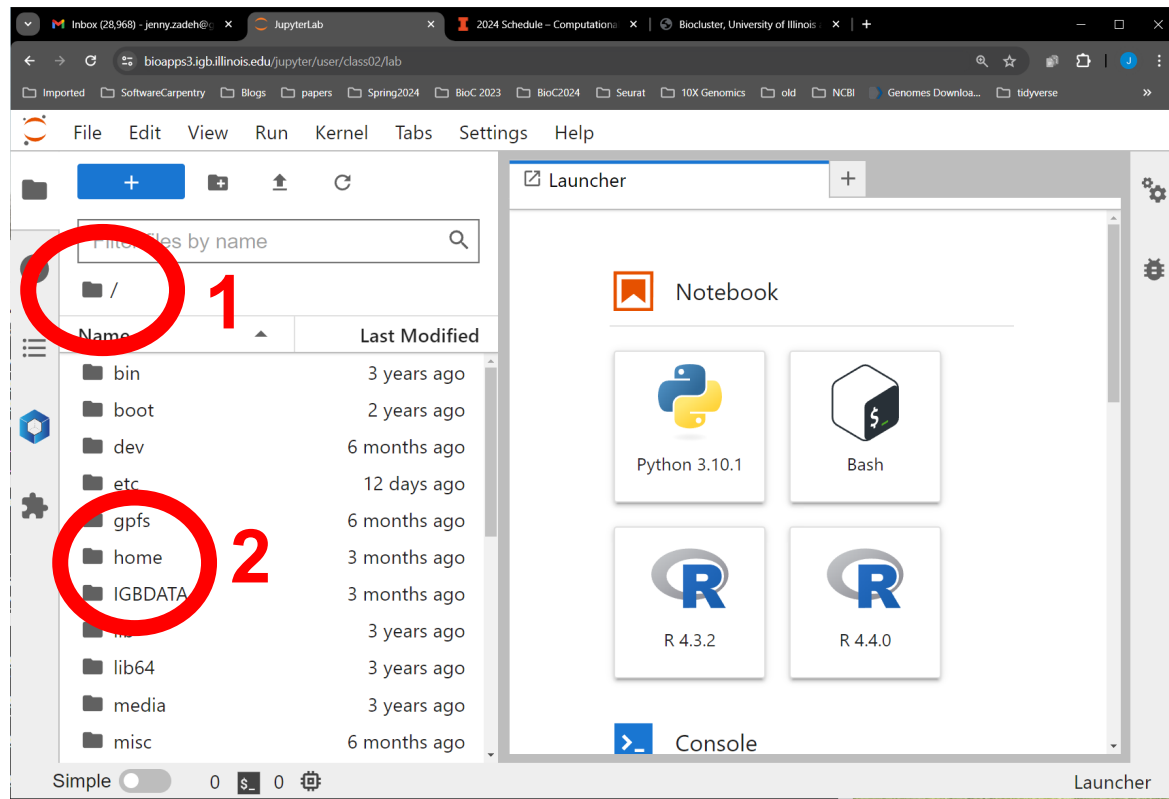






# Navigate back to your home directory

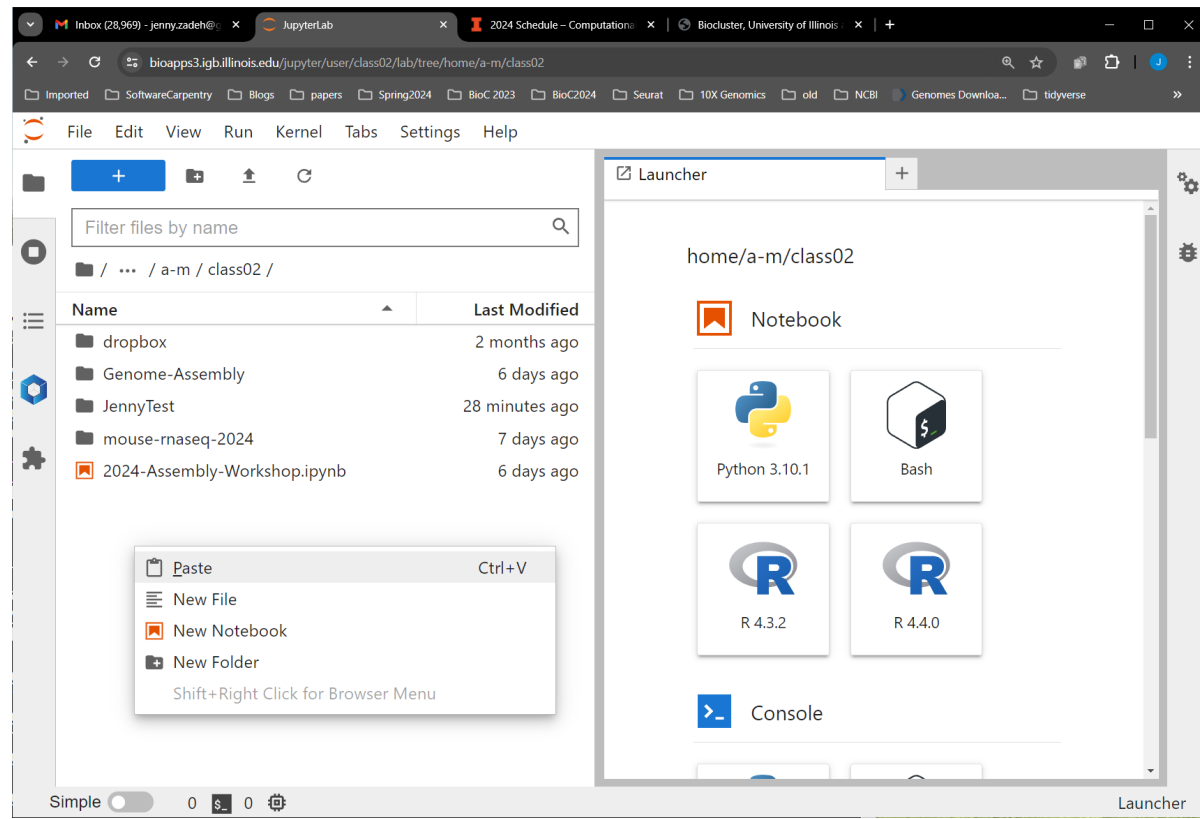
1. Click on second gray folder under search box to get to root directory
2. Double click on "home" directory





## Continue navigating by double clicking:

- a-m -> classXX
- Right click below any files/directories and select "Paste"
- You should then see the "BasicSingleCell.ipynb"





# Open BasicSingleCell.ipynb

- Double click on the file to open in right pane
- This is a self-paced lab with explanations and runnable code boxes
- If the second code box gives you an error, see next slide

The screenshot displays a JupyterLab environment. On the left, a file browser shows a directory structure with a search bar and a table of files. The file 'BasicSingleCell.ipynb' is selected and highlighted in blue. The right pane shows the notebook content, which includes a title, author information, course details, and introductory text about the lab's scope and resources.

| Name                      | Last Modified  |
|---------------------------|----------------|
| dropbox                   | 2 months ago   |
| Genome-Assembly           | 6 days ago     |
| JennyTest                 | 31 minutes ago |
| mouse-rnaseq-2024         | 7 days ago     |
| 2024-Assembly-Workshop... | 6 days ago     |
| BasicSingleCell.ipynb     | 2 minutes ago  |

**Single Cell & Spatial Transcriptomics Lab**

Jenny Drnevich, HPCBio

COMPUTATIONAL GENOMICS COURSE, 2024

We will be walking through the analysis of one single cell sample due to time constraints. The scope of the lab does not allow us to get very in-depth about every step of the process. Other excellent resources for more details on each step and how to analyze multiple samples together and/or Spatial data can be found at the following resources:

1. [Seurat's website](#)
2. Tim Stuart's [UCLA-T32 Single-Cell Analysis Workshop](#) (the inspiration for this jupyter notebook)



# stop("Please install hdf5r to read HDF5 files")

If you get this error in the second code box then you did not load the Mayo module before opening the .ipynb.

To fix, follow next slides

6/26/2024

The screenshot shows a JupyterLab interface with a file browser on the left and a code editor on the right. The file browser shows a table of files and folders:

| Name           | Last Modified |
|----------------|---------------|
| dropbox        | 2 months ago  |
| Genome-As...   | 6 days ago    |
| JennyTest      | an hour ago   |
| mouse-rmas...  | 7 days ago    |
| 2024-Asse...   | 6 days ago    |
| BasicSingle... | 2 minutes ago |
| jupyterhub...  | seconds ago   |
| Rplots.pdf     | 4 minutes ago |

The code editor shows the following code cell:

```
[2]: mousecounts <- Read10X_h5("10k_Mouse_Neurons_3p_gemx_10k_Mouse_Neurons_3p_gemx_count_sample_filtered_feature_bc_matrix.h5")
class(mousecounts)
```

The error message is:

```
Error in Read10X_h5("10k_Mouse_Neurons_3p_gemx_10k_Mouse_Neurons_3p_gemx_count_sample_filtered_feature_bc_matrix.h5"): Please install all hdf5r to read HDF5 files
Traceback:
 1. Read10X_h5("10k_Mouse_Neurons_3p_gemx_10k_Mouse_Neurons_3p_gemx_count_sample_filtered_feature_bc_matrix.h5")
 2. stop("Please install hdf5r to read HDF5 files")
```

The note below the error is:

```
NOTE: if you get an error about "Please install hdf5r to read HDF5 files" see beginning slides for a fix.
```

The bottom of the screenshot shows the status bar with the following information:

Simple 0 1 R 4.4.0 | Idle Mode: Command Ln 1, Col 1 BasicSingleCell.ipynb



# Load Mayo module

1. Click on blue hexagon
2. Search for Mayo at top
3. Click on "Load" by Mayo module at bottom.

The screenshot shows the JupyterLab interface. On the left, the 'Environments' panel is open, displaying a search bar with 'mayo' entered. A red circle highlights the search bar (labeled '2'). Below the search bar, a list of modules is shown, with the 'Mayo/2024-IGB-gcc-8.2.0' module highlighted. A red circle highlights the blue hexagonal 'Load' button next to this module (labeled '1'). At the bottom of the interface, a red circle highlights the 'Load' button (labeled '3'). The main area shows a code cell with R code: `mousecounts <- Read10X_h5("10k_Mouse_Neurons_3p_gemx_10k_Mouse_Net... class(mousecounts)`. Below the code, a red error message is displayed: `Error in Read10X_h5("10k_Mouse_Neurons_3p_gemx_10k_Mouse_Neurons_3p_gemx_count_sample_filtered_feature_bc_matrix.h5"): Please install hdf5r to read HDF5 files`. The error message includes a traceback with two steps: `1. Read10X_h5("10k_Mouse_Neurons_3p_gemx_10k_Mouse_Neurons_3p_gemx_count_sample_filtered_feature_bc_matrix.h5")` and `2. stop("Please install hdf5r to read HDF5 files")`. A note below the error message says: `NOTE: if you get an error about "Please install hdf5r to read HDF5 files" see beginning slides for a fix.` The bottom status bar shows 'Simple', '0', '1', 'R 4.4.0 | Idle', 'Mode: Command', 'Ln 1, Col 1', and 'BasicSingleCell.ipynb'.



# Stop Kernel

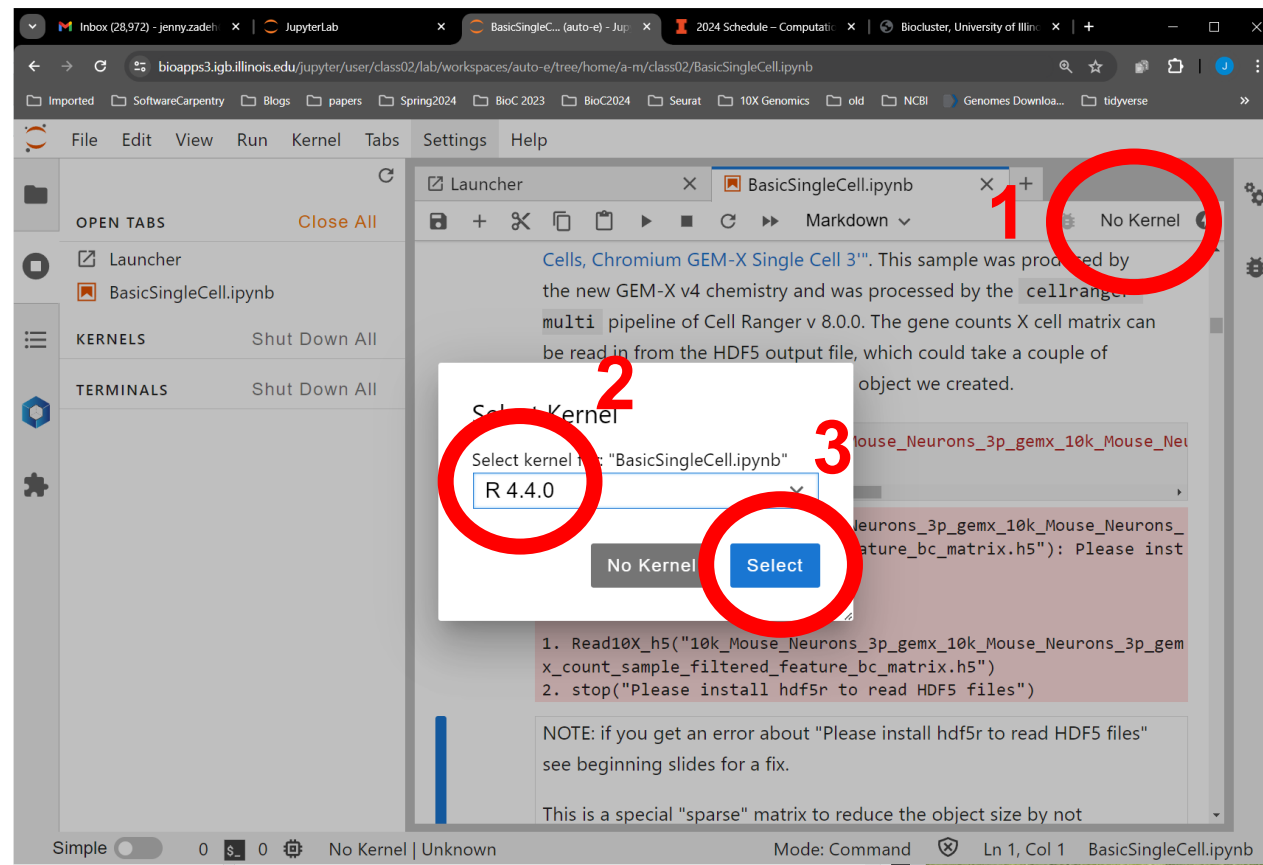
1. Click on gray circle with white square on left
2. Under KERNELS click on the X by BasicSingleCell.ipynb

The screenshot shows the JupyterLab interface. On the left sidebar, the 'Kernels' section is visible, with a gray circle containing a white square icon circled in red (labeled '1'). Below it, the 'BasicSingleCell.ipynb' kernel is listed with a red 'X' icon circled in red (labeled '2'). The main area shows a code cell with R code and an error message. The error message is highlighted in pink and reads: "Error in Read10X\_h5(...): Please install hdf5r to read HDF5 files". Below the error is a traceback with two steps: 1. Read10X\_h5(...) and 2. stop("Please install hdf5r to read HDF5 files"). A note below the error says: "NOTE: if you get an error about 'Please install hdf5r to read HDF5 files' see beginning slides for a fix." The bottom status bar shows "Simple", "0", "1", "R 4.4.0 | Idle", "Mode: Command", "Ln 1, Col 1", and "BasicSingleCell.ipynb".



# Re-start Kernel

1. Click on the "No Kernel" in the top right of the .ipynb to get the pop-up
2. Pick R 4.4.0 from the drop down menu
3. Click on "Select"





# Re-run the code boxes

You need to re-run the **first code box** to load the packages from the library, then you can successfully run the second code box

The screenshot shows a JupyterLab interface with the following elements:

- Browser Tabs:** Includes 'Inbox (28,973) - jenny.zadel...', 'JupyterLab', 'BasicSingleCell... (auto-e) - Jup...', '2024 Schedule - Computat...', and 'Biocluster, University of Illin...'. The active tab is 'BasicSingleCell.ipynb'.
- File Browser:** Shows a directory structure with folders like 'Imported', 'SoftwareCarpentry', 'Blogs', 'papers', 'Spring2024', 'BioC 2023', 'BioC2024', 'Seurat', '10X Genomics', 'old', 'NCBI', 'Genomes Downloa...', and 'tidyverse'.
- Launcher:** Shows 'BasicSingleCell.ipynb' as the active notebook.
- KERNELS:** Shows 'BasicSingleCell.ipynb' as the active kernel.
- Code Cell [4]:** Contains the code `mousecounts <- Read10X_h5("/home/classroom/mayo/2024/BasicSingleCell...")`. The output is `'dgCMatrix'`, which is circled in red with the word **Success!** overlaid in large red text.
- Text:** Below the code cell, there is a note: "NOTE: if you get an error about 'Please install hdf5r to read HDF5 files' see beginning slides for a fix." This is followed by an explanation: "This is a special 'sparse' matrix to reduce the object size by not recording zeros as numbers. We can ask how many rows/genes and how many columns/cells are in the matrix with the `dim()` function (lists # rows first, then # columns), then take a look at the first 7 rows and first 5 columns:"
- Code Cell [107]:** Contains the code `dim(mousecounts)` with the output `33696 · 12441`.
- Code Cell [108]:** Contains the code `mousecounts[1:7, 1:5]`.
- Bottom Bar:** Shows 'Simple' mode, 'R 4.4.0 | Idle', and 'Mode: Command | Ln 1, Col 9 | BasicSingleCell.ipynb'.