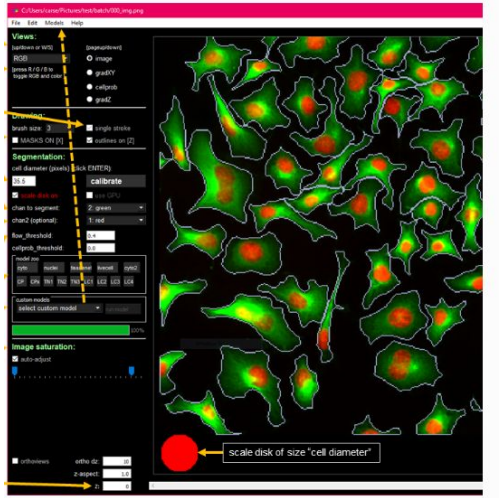
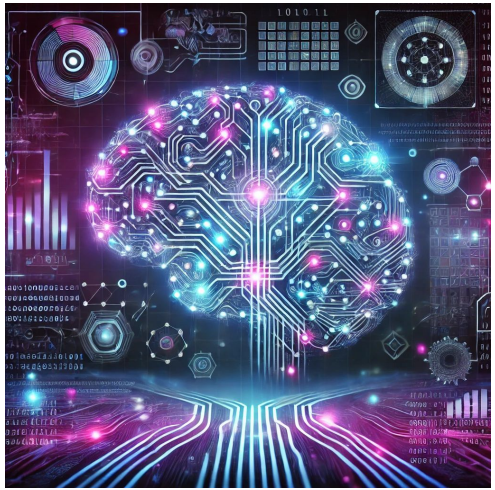
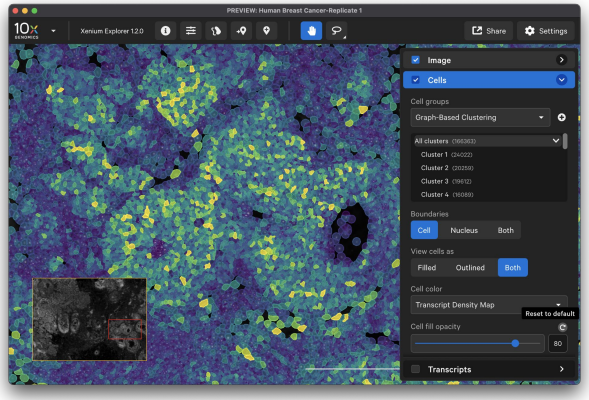
Machine Learning in (Genomics) Image Data with Python



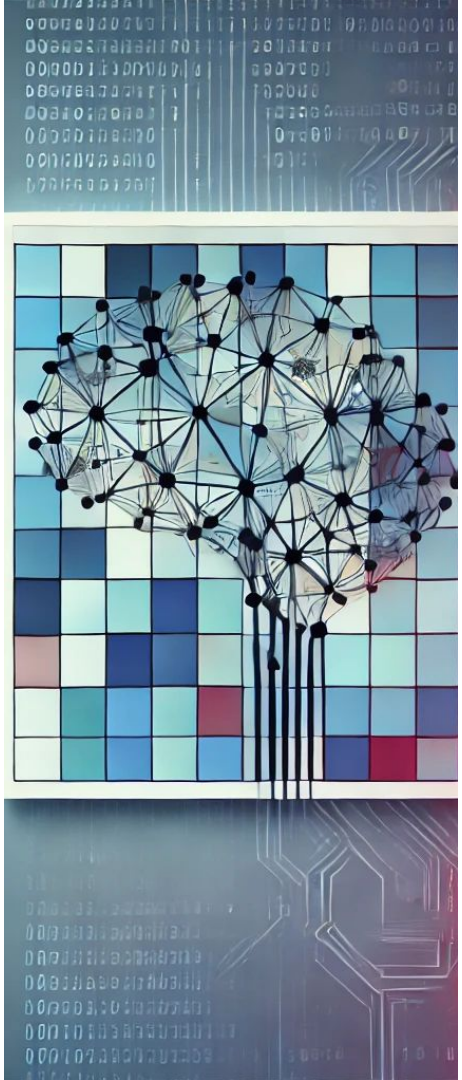




Ivan Chang
March 18th 2025

Outline

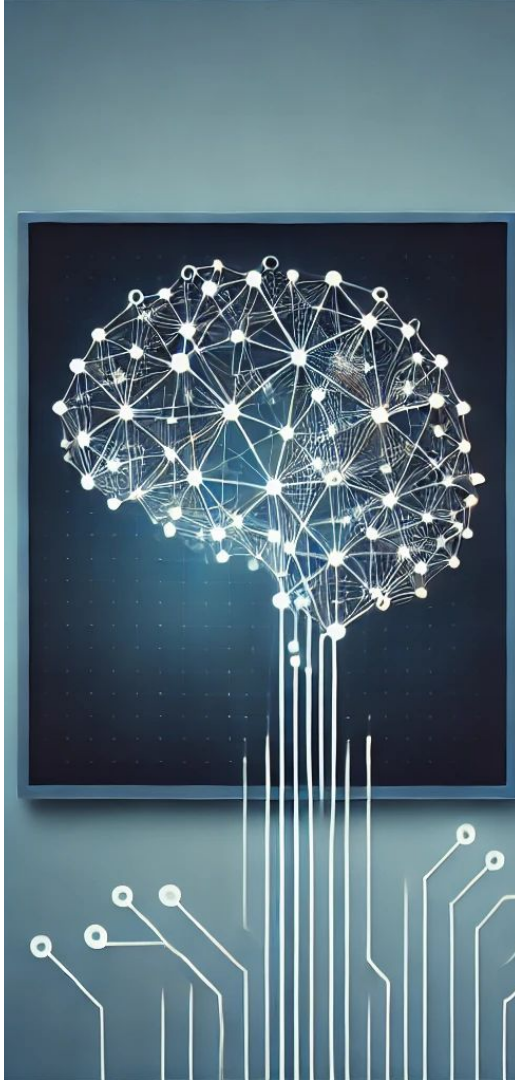
- What is Machine Learning?
- Why ML for Genomics and image analysis?
- Types of Learning
- Ingredients of ML
- Parametric vs non-parametric models
- Why Python for ML?
- Classical ML algorithms
- Image segmentation
- Genomics image analysis software
- Interactive Computing on UCI HPC3
- Hands on material - image segmentation via Cellpose



What is Machine Learning?

- The process by which computer systems can be directed to improve their performance over time — that is, to modify its execution on the basis of newly acquired information.

- Subspecialty of artificial intelligence concerned with developing methods for software to learn from experience or extract knowledge from patterns in data



Why ML for Genomics data/image analysis?

High-throughput microscopy: ML can rapidly analyze large numbers of microscopy images, identifying cellular structures, quantifying gene expression, or detecting abnormalities.

Cell classification: ML can categorize different cell types or states based on morphological features in microscopy images, which is useful in developmental biology and cancer research.

Gene expression mapping: ML can analyze in situ hybridization images to map gene expression patterns across tissues or organisms.

Nuclei segmentation: In histology images, ML can accurately identify and segment individual cell nuclei, which is crucial for studying cellular organization and gene expression at the single-cell level.

Phenotype analysis: ML can quantify phenotypic traits from images of model organisms, linking genotypes to observable characteristics.

Time-lapse analysis: For dynamic processes like cell division or tissue vascularization, ML can track and analyze changes over time in image sequences.

Multi-modal data integration: ML can combine image data with other types of genomic data (e.g., sequencing data) for more comprehensive analyses.

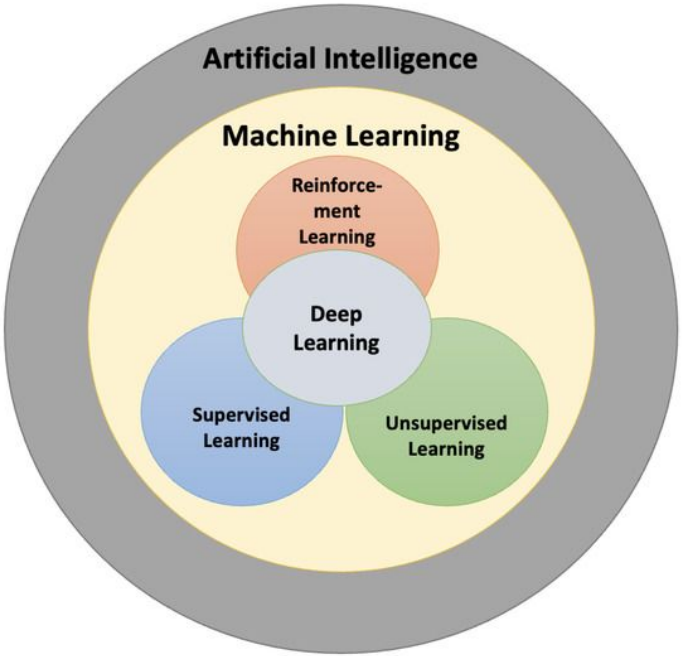
Types of Learning

- Supervised Learning
  - Labels are provided, there is a strong learning signal.
  - e.g. classification, regression.

- Unsupervised learning
  - There is no direct learning signal. Simply trying to find structure in data.
  - e.g. clustering, dimensionality reduction.

- Semi-supervised Learning.
  - Only a portion of the data have labels. Hybrid approach bridging both supervised and unsupervised learning.
  - e.g. a image recognition.

- Reinforcement learning.
  - The learning signal is a reward or punishment and may come with a delay from interacting with environment
  - e.g. trying to learn to play chess, a mouse in a maze.

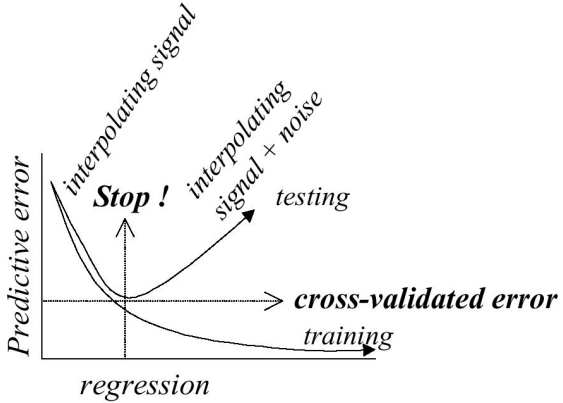


Ingredients

- Data:
  - what kind of data do we have?
- Prior assumptions:
  - what do we know a priori about the problem?
- Representation:
  - How do we represent the data?
- Model / Hypothesis space:
  - What hypotheses are we willing to entertain to explain the data?
- Feedback / learning signal:
  - what kind of learning signal do we have (delayed, labels)?
- Learning algorithm:
  - How do we update the model (or set of hypothesis) from feedback?
  - Minimize/Maximize the objective or “cost” function
- Evaluation:
  - How well did we do, should we change the model?
  - Cross-validation

Cross-validation

How do we ensure good generalization, i.e. avoid "over-fitting" on our particular data sample.

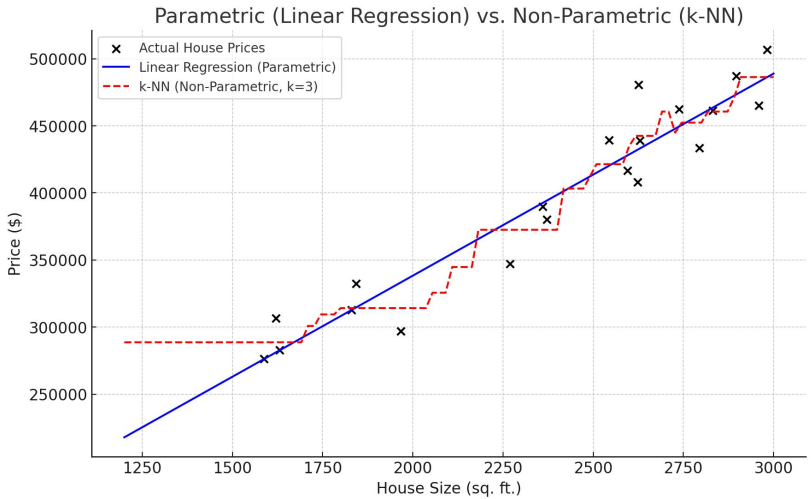


- You are ultimately interested in good performance on new (unseen) test data.
- To estimate that, split off a (smallish) subset of the training data (called validation set).
- Train without validation data and "test" on validation data.
- Repeat this over multiple splits of the data and average results.
- Reasonable split: 90% train, 10% test, average over the 10 splits.

Parametric vs Non-parametric models

Parametric models make strong assumptions about the data (e.g., assuming it follows a normal distribution), making them efficient but less adaptable.

Non-parametric models make fewer assumptions, allowing them to capture complex relationships but at the cost of higher computation and more data requirements.



Why Python for ML?

Rich Ecosystem of Libraries and Frameworks:
Python boasts a vast collection of libraries and frameworks specifically designed for machine learning, data analysis, and scientific computing, such as ***scikit-learn*** for general machine learning, ***TensorFlow*** and ***PyTorch*** for deep learning, and Pandas and NumPy for data manipulation and analysis.

Strong Community Support:
Python has a large and active community of developers and data scientists, providing ample resources, tutorials, and support for those learning and using the language for machine learning.

Scalability:
Python allows developers to build machine learning models on a small scale and thenV scale them up as needed.

Section Navigation

- Release Highlights
- Biclustering
- Calibration
- Classification
- Clustering
- Covariance estimation
- Cross decomposition
- Dataset examples
- Decision Trees
- Decomposition
- Developing Estimators
- Ensemble methods
- Examples based on real world datasets
- Feature Selection
- Frozen Estimators
- Gaussian Mixture Models
- Gaussian Process for Machine Learning
- Generalized Linear Models
- Inspection
- Kernel Approximation
- Manifold learning
- Miscellaneous
- Missing Value Imputation

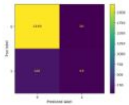
Examples

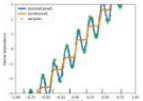
This is the gallery of examples that showcase how scikit-learn can be used. Some examples demonstrate the use of the API in general and some demonstrate specific applications in tutorial form. Also check out our user guide for more detailed illustrations.

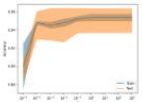
Release Highlights

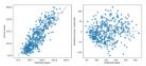
These examples illustrate the main features of the releases of scikit-learn.

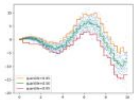
Release Highlights for scikit-learn 1.6


Release Highlights for scikit-learn 1.5

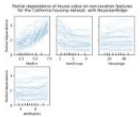

Release Highlights for scikit-learn 1.4

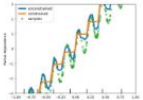

Release Highlights for scikit-learn 1.3

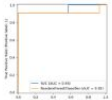

Release Highlights for scikit-learn 1.2


Release Highlights for scikit-learn 1.1

Release Highlights for scikit-learn 1.0


Release Highlights for scikit-learn 0.24


Release Highlights for scikit-learn 0.23


Release Highlights for scikit-learn 0.22

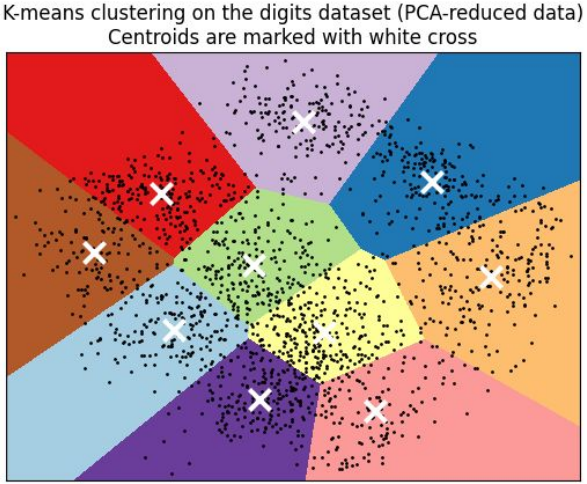
Classical ML algorithms

Main categories of ML

- **Classification**
  - Image or feature segmentation for cell profiling
- **Clustering**
  - Variant calling
  - Admixture analysis
  - Flow cytometry gating
- Dimensionality reduction
  - PCA, tsne & UMAP
- Optimization
  - Multi-omics model tuning and/or structure learning

K-means Clustering

- Unsupervised learning
- Non-parametric modeling



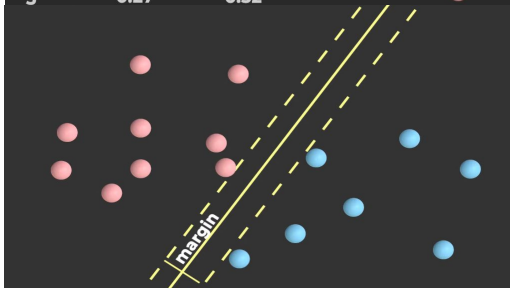
ELI5 K-means clustering algorithm

1. Choose a number, k, which is how many clusters you expect the data to have.
2. Make k guesses as to where those clusters could be. Your guesses will be wrong, but that doesn't matter.
3. For each element in your data, assign it to the cluster it's closest to. ***(Objective or cost function)***
4. Move the center of each cluster to be in the middle of the elements that are assigned to that cluster.
5. Repeat steps 3 and 4 until no more data moves from one cluster to another.

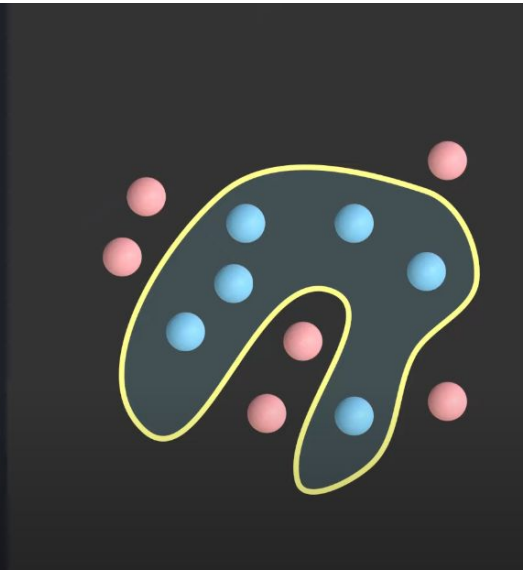
Support Vector Machine

- Supervised learning
- Both parametric (linear) and non-parametric (kernel) models

Point	Feature 1	Feature 2	...	Category
1	0.97	0.27		
2	0.77	0.19		
3	0.33	0.85		
4	0.41	0.93		
5	0.27	0.32		

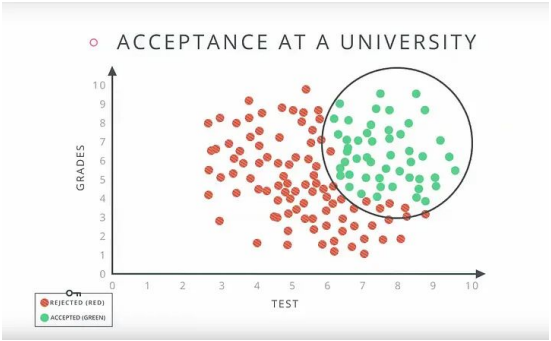


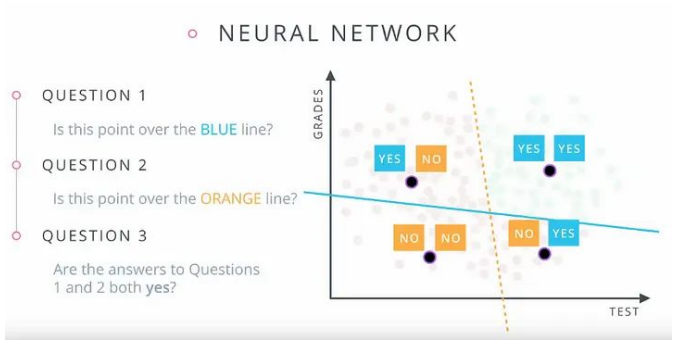

```
from sklearn import svm

# data points
X = [
[ 0.1,  2.3],
[-1.5,  2.5],
[ 2.0, -4.3],
#...
]

# labels
y = [0, 1, 0,  ] # ...

# nonlinear transformation
fX = [ (x[0], x[1], x[0]**2+x[1]**2)
       for x in X]

# fit
svm.SVC().fit(fX, y)
```

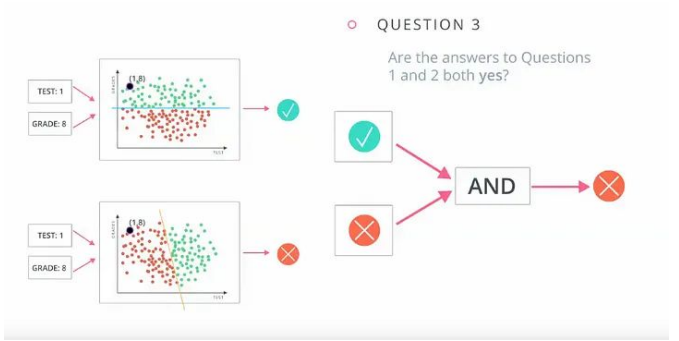


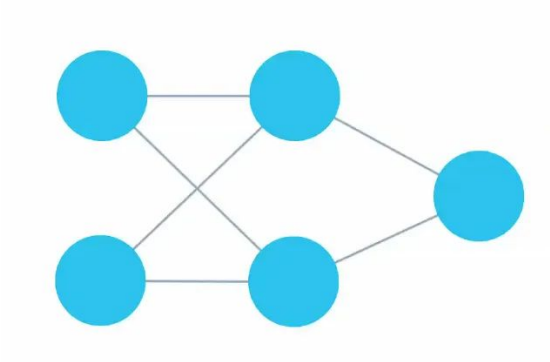
https://www.youtube.com/watch?v=Q7vT0--5VII

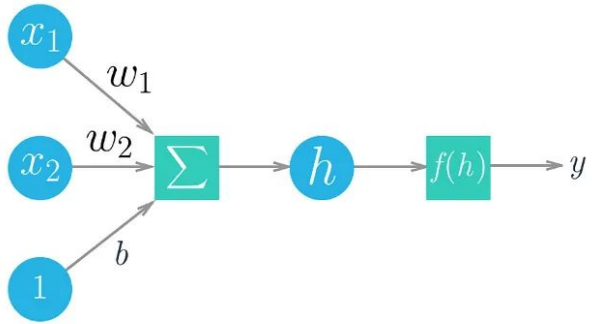
Neural Network











https://becominghuman.ai/deep-learning-an-eli5-intro-to-neural-networks-baf7b02c1ae5

Deep Learning

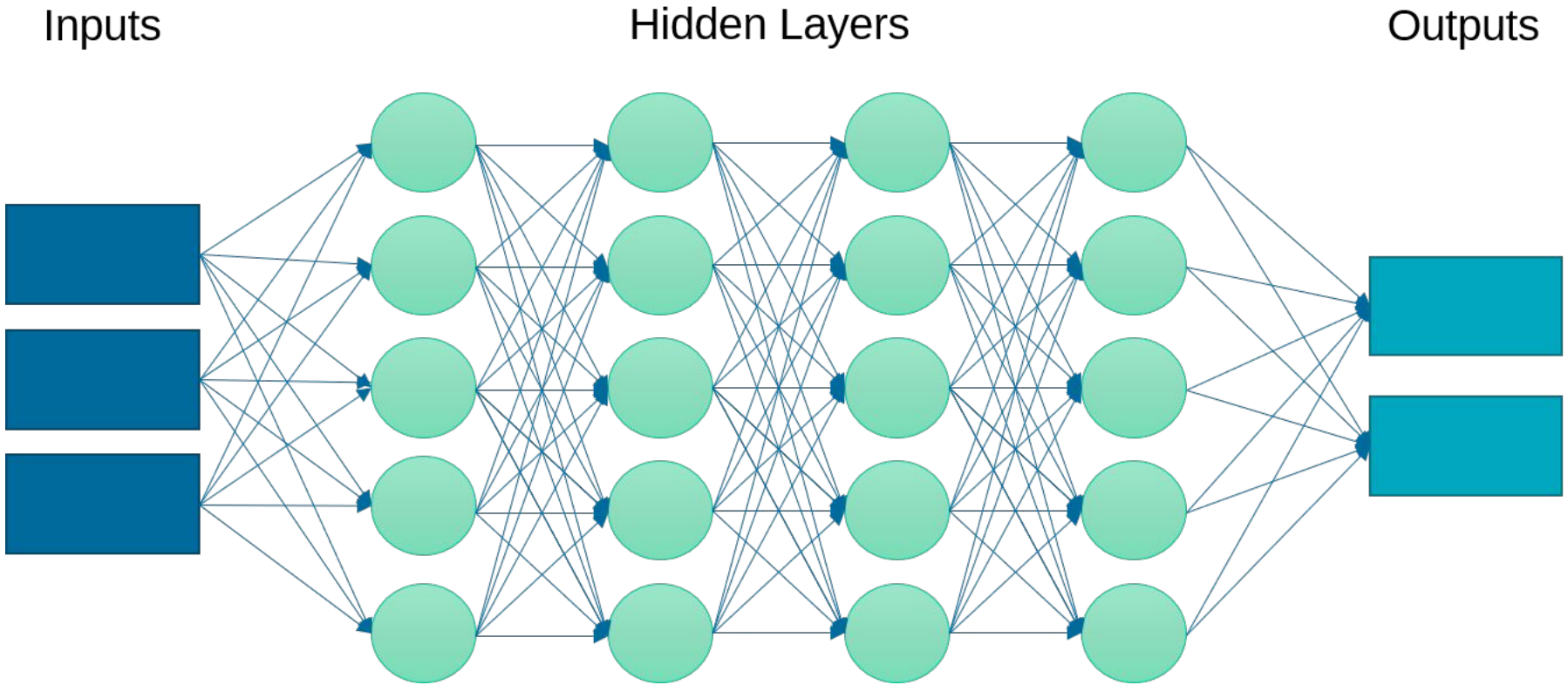
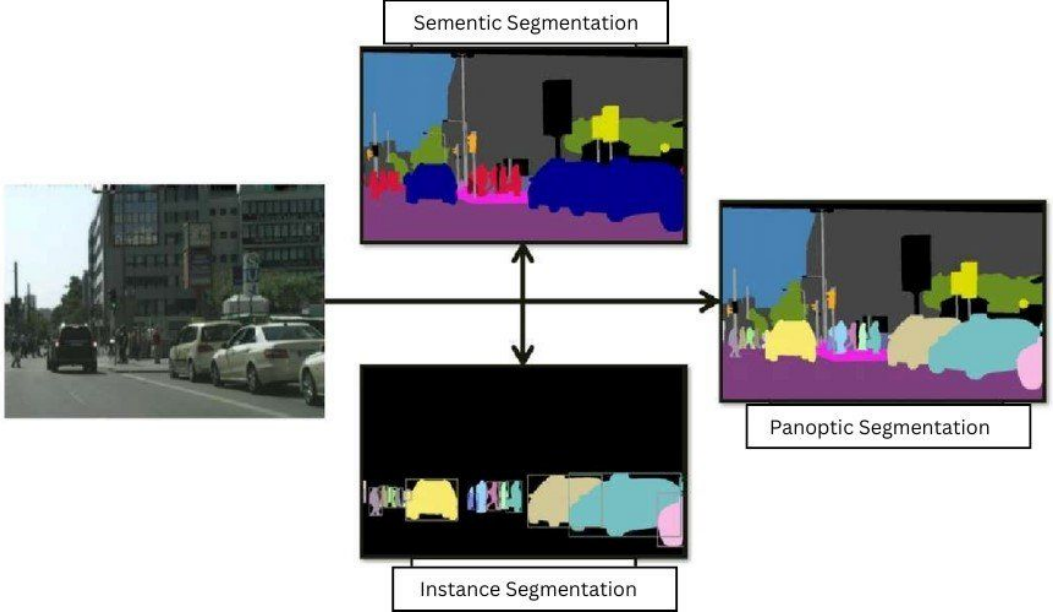


Image segmentation types



Semantic Segmentation: Assigns a class label to each pixel in the image. Does not differentiate between instances of the same class

Example: In a street scene, all pixels belonging to "car" are labeled the same, regardless of how many cars are present.

Instance Segmentation: Detects and delineates each distinct object of interest in the image. Differentiates between separate instances of the same class

Example: In the same street scene, each car would be identified and outlined separately.

Panoptic Segmentation: Combines semantic and instance segmentation. Assigns a class label to every pixel and distinguishes between object instances for specific classes

Example: In a street scene, it would label the road, sky, and buildings (semantic), while also outlining individual cars and pedestrians (instance)

Image segmentation methods

- Traditional approaches
  - Thresholding Segmentation
  - Edge-Based Segmentation
  - Region-Based Segmentation
- Machine Learning approach
  - Clustering-Based Segmentation
    - Applications: Image compression, pattern recognition.
- Deep Learning-Based Segmentation
  - Fully Convolutional Networks (FCNs): Convert fully connected layers to convolutional layers to produce pixel-wise predictions.
  - **U-Net:** A symmetric network with encoder-decoder architecture, widely used in biomedical image segmentation.
  - Mask R-CNN: Extends Faster R-CNN by adding a branch for predicting segmentation masks on each Region of Interest (RoI).
- Graph-Based Segmentation
  - Applications: Image editing, object recognition.

Each technique has its strengths and is suited for different applications. Deep learning methods, in particular, have shown remarkable performance in complex segmentation tasks due to their ability to learn hierarchical features from data

Genomics image analysis software

- CellProfiler
  - A powerful and flexible open-source software platform designed for analyzing and measuring biological images, particularly in the field of cell biology and microscopy.
  - It can identify and measure various cellular features such as cell size, shape, intensity of staining, and protein levels.
  - Users can create pipelines of image analysis modules to detect objects (like nuclei, cells, or organisms) and make measurements on these objects.

CellProfiler™
cell image analysis software

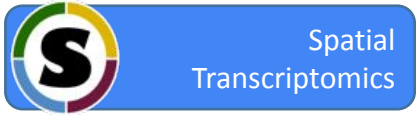
- Cellpose
  - A generalist algorithm for cellular segmentation, designed to work across various cell types and imaging modalities
  - It uses deep learning techniques, specifically U-Net convolutional neural networks, to perform segmentation tasks.
  - Can segment a wide range of cell types without requiring manual annotation or parameter adjustments for each new cell type.
  - The algorithm comes pre-trained on a diverse set of images, but users can also train it on their own data for specific applications.
  - Designed to be fast, especially when run on GPUs.

Jupyter Ecosystem - Running Containers Interactively on HPC3

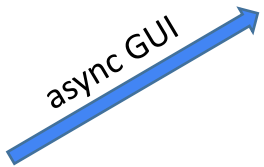
HPC3 User Authentication

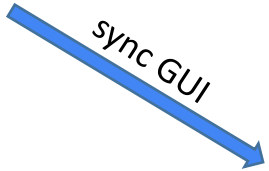
https://hpc3.rcic.uci.edu/biojhub4/

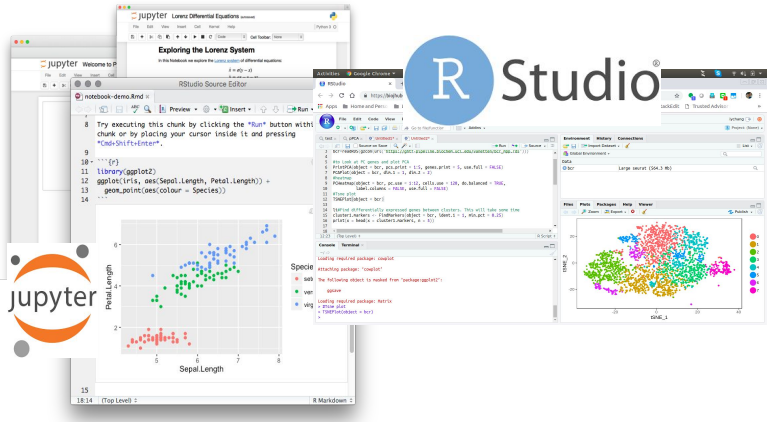
https://hpc3.rcic.uci.edu/bioportal-1/

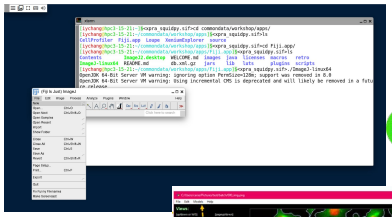


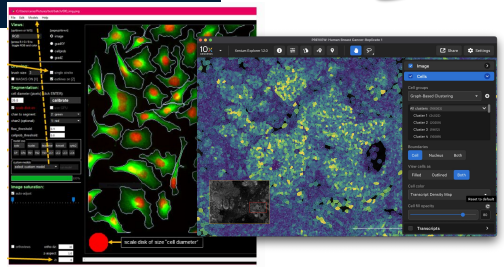
Containerized software environment



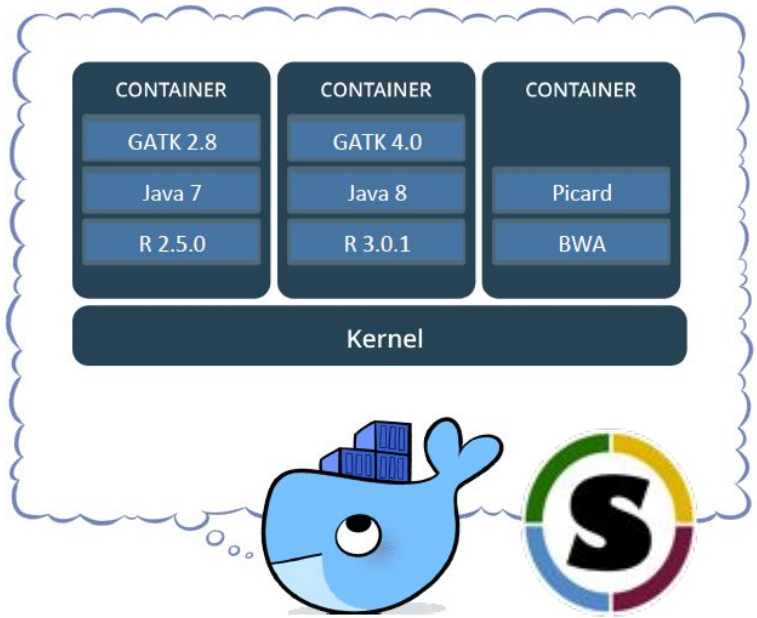








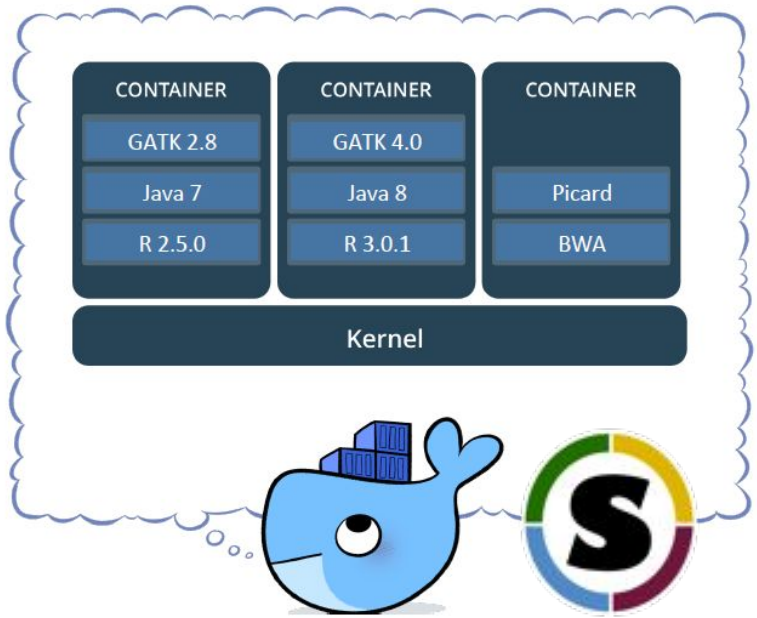
Container Technology Overview



Modified from https://www.docker.com/what-container

- Containers are isolated, but share OS and bins/libraries
- Provide highly customized software environment apart from the host system

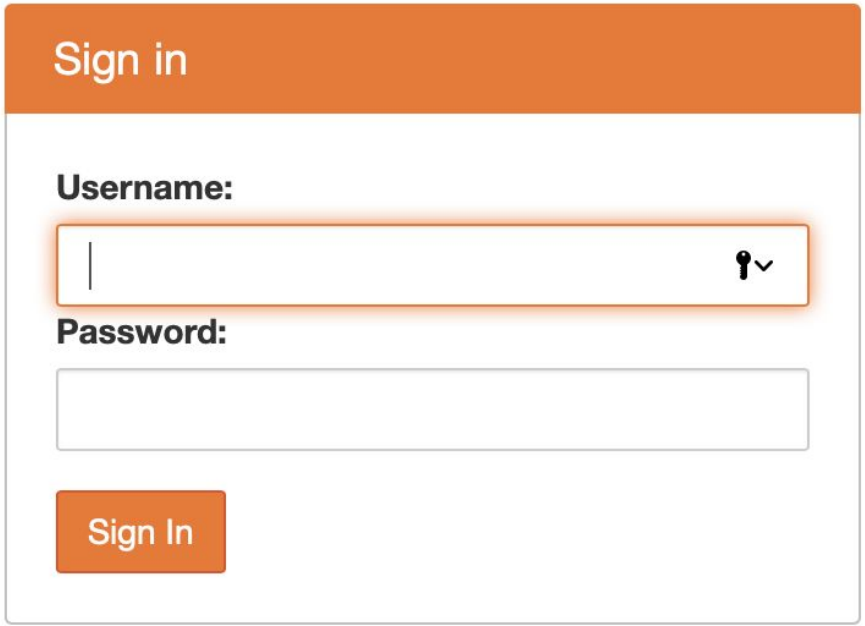
Container Benefits



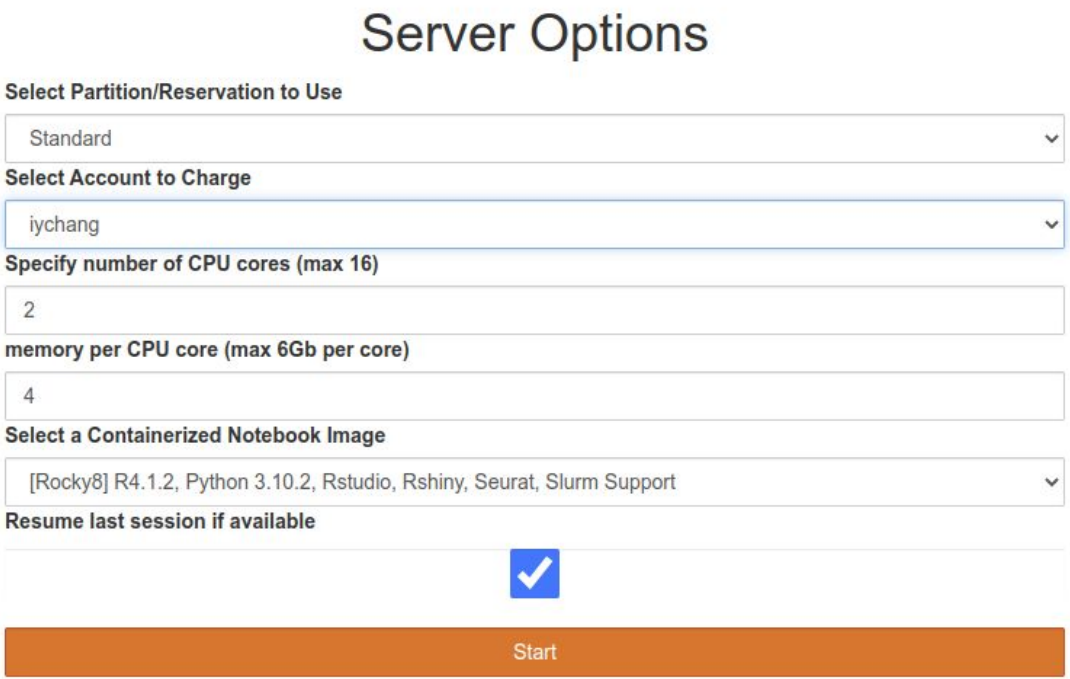
Modified from https://www.docker.com/what-container

- **Portability**: containers can be published and shared via cloud-based container hubs (https://hub.docker.com/, https://dockstore.org/, https://www.singularity-hub.org/) or transferred directly as image files
- **Versioning**: container build files can be stored in git/github repositories
- **Reproducibility**: published containers are immutable, and provides a snapshot of the computing environment used to run analysis

Using your favorite browser go to: https://hpc3.rcic.uci.edu/biojhub4/hub/login You will see the following screen where you will Use your usual HPC3 credentials to sign in:

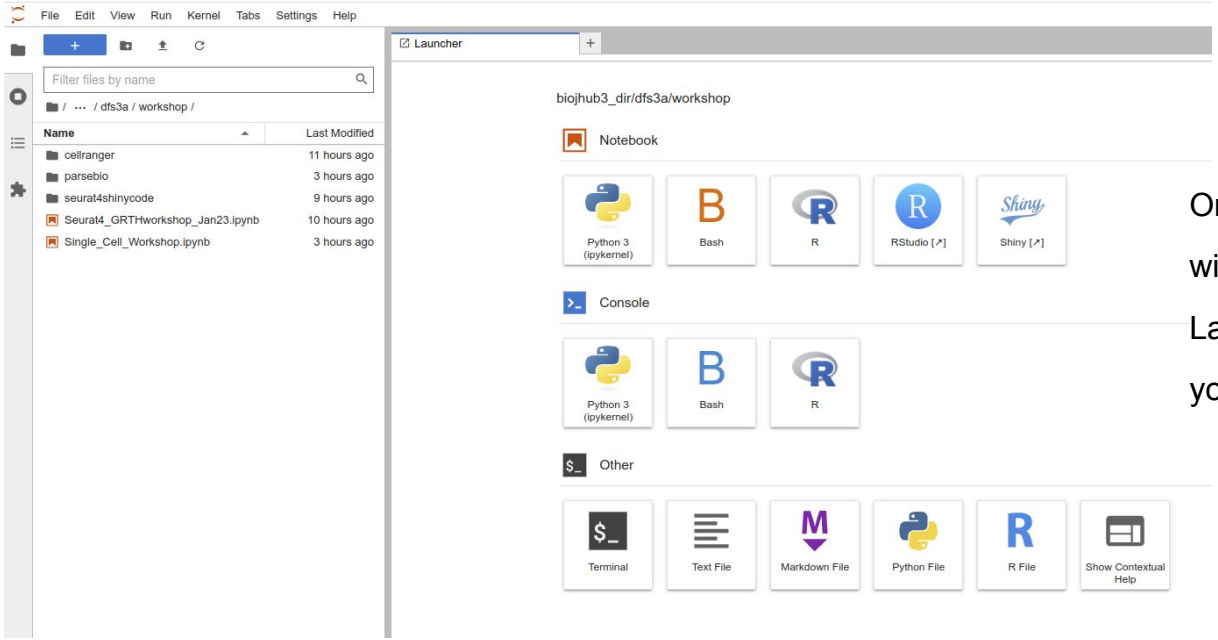


After authentication you will see a screen with server options as in the figure below:



For this workshop, modify the *Select Account to Charge* to be one of your Slurm accounts, change number of CPUs to **6** and the memory per CPU core to **6**, select the "[Rocky8][Beta]R4.3.3/Python 3.10.2 **ML Workshop)**" container and press `Start`.

Main Jupyter Interface

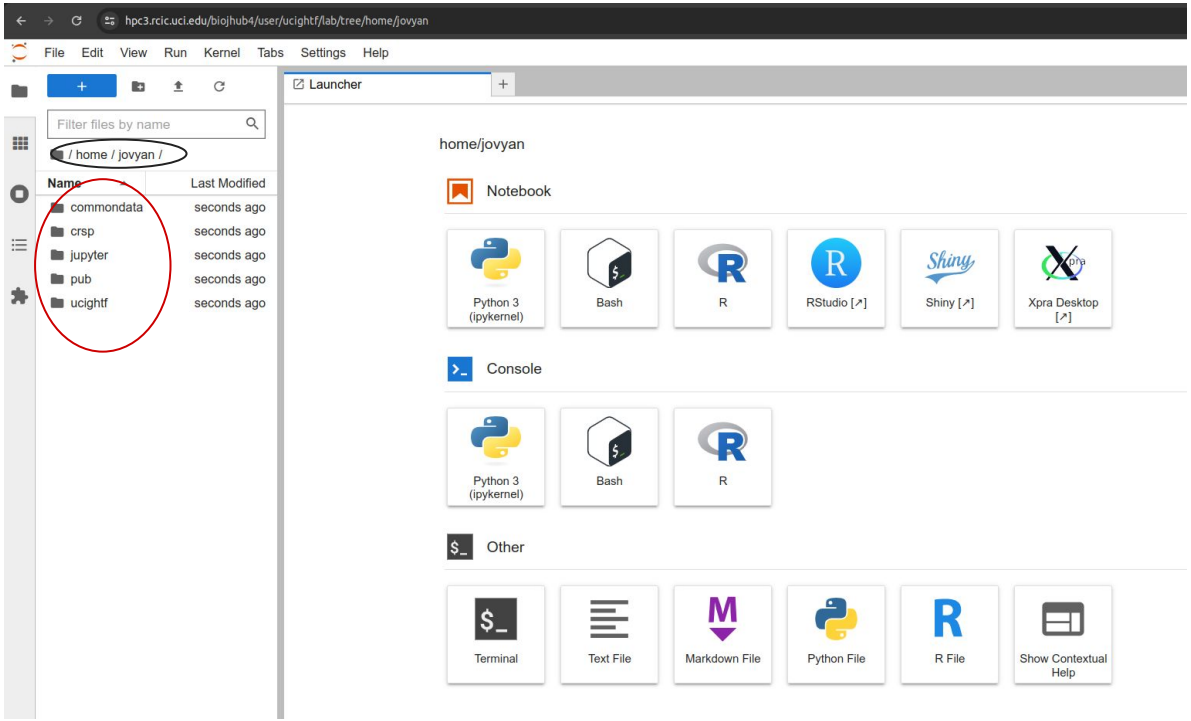


Once the notebook is done spawning, you will get a file browser on the left, and a Launcher screen with a number GUI apps you can use on the right.

Fictitious home directory and short cuts

Default Jupyter container user

Shortcuts to oft used paths



https://hpc3.rcic.uci.edu/biojhub4/

/dfs8/commondata/workshop/ML

Location of XPRA enabled Spatial-omics Apps

/dfs8/commondata/workshop/apps/CellProfiler (Windows)

/dfs8/commondata/workshop/apps/Fiji.app (Linux - X11)

/dfs8/commondata/workshop/apps/Loupe (Windows)

/dfs8/commondata/workshop/apps/XeniumExplorer (Windows)

/dfs8/commondata/workshop/apps/ImageStudio (Windows)

XPRA - CellPose Container Specific Instruction

From the xterm in Xpra Desktop, enter:

python -m cellpose

or

cellpose