

## **HAL Cluster Commands (NCSA):**

- Accessing the cluster :  
`ssh <userid>@hal-login2.ncsa.illinois`

---

*If it is your first time using the cluster you will need to set up a custom environment using the following steps.*








- Type: `conda create --name=wellplate --clone=opence-v1.6.1` and press Enter (this will create a custom environment with all the libraries in opence-v1.6.1)
- Type: `conda activate wellplate` and press Enter (this activate the wellplate environment)
- Type: `pip install xlswriter` and press Enter (this add a library to the wellplate environment and enable it to write excel files)

---

*After successfully using the ssh command to connect to the cluster you will be sent to the login node. Do not run any computations here. Login nodes are only meant to handle files or to gain access to computer nodes. All computations should be performed on compute node(s).*

- Accessing a compute node:
  - Once you're on the login node, type the following command for a gpu node:  
`swrun -p gpux1`
  - Alternatively type the following command for a gpu node with the maximum amount of cpus:  
`srunk --partition=gpu --time=4:00:00 --nodes=1 --ntasks-per-node=96 --sockets-per-node=2 --cores-per-socket=12 --threads-per-core=4 --mem-per-cpu=1200 --wait=0 --export=ALL --gres=gpu:v100:1 --pty /bin/bash`
- Loading the appropriate module for running the pose prediction workflow:  
`conda activate wellplate`
- After obtaining access to a compute node and activating the wellplate environment you are ready to run the scripts with the following commands:
  - To run the runme script, type the following:  
`python runme.py`
  - To run the visualization script, type the following:  
`python visualize.py`

### HAL Cluster Well Plate Program Structure:

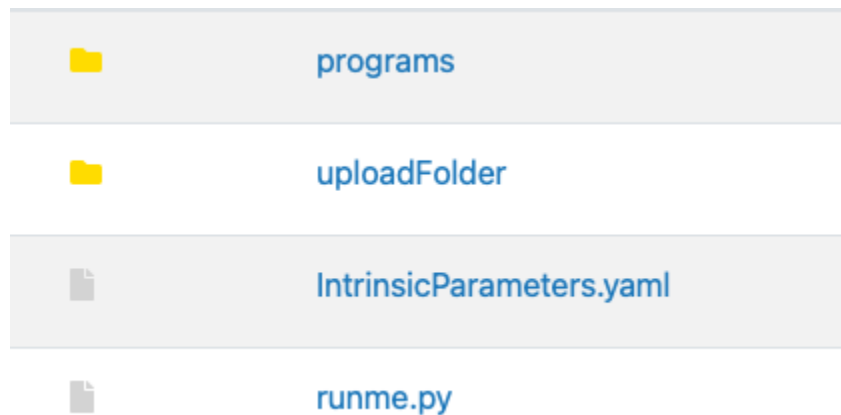
<input type="checkbox"/>		inputs
<input type="checkbox"/>		outputs
<input type="checkbox"/>		programs
<input type="checkbox"/>		uploadYourGridHere
<input type="checkbox"/>		uploadYourVideosHere
<input type="checkbox"/>		runme.py
<input type="checkbox"/>		visualize.py

The program has two main directories used to pass in files to the program. The first of these directories is *uploadYourGridHere*. As the name implies *uploadYourGridHere* is the directory in which you will need to upload your grid file. It is important to note that you should have only one grid file here. The second directory is *uploadYourVideosHere*, in this directory you can upload the videos/folders which you want to be analyzed.

In order to upload files to the cluster you can use the method of your choice, some options are using winscp, using the terminal with the sftp command, or through their [online portal](#). Using the online portal is arguably the easiest. You simply log in, click on the files tab to navigate to the home directory, and from there simply upload or remove files just like in a regular computer.

The program also has two main scripts, *runme.py* and *visualize.py*. *runme.py* is used to predict the pose of the fish and afterwards create an excel file of their velocities. *visualize.py* is used to visualize the outputs by creating videos with the results superimposed on them. It is important to note that *visualize.py* should not be ran before *runme.py*. Both of these programs will save their results to folders, named the same as their video file, located in the *outputs* directory. Information on running these scripts can be found in the following section

### HAL Cluster Generate Intrinsic Parameters Structure:



This program's purpose is to create a yaml file which stores parameters used for rendering the fish. One interacts with it by uploading a folder containing fish images along with their annotations. This folder, the one with fish images and annotations, can be obtained from GUI's Calculate Intrinsic Parameters page. The GUI page allows the user to iterate through frames in a video in order to collect cropped images of fish with a straight backbone. After collecting the images a window appears allowing the user to click on the start of their head and end of their tail. After annotating all of the cropped images one can click the Save Data to create the folder which has to be uploaded to uploadFolder.

