This is a quick guide to run the pipeline. The user will interact with the following scripts.

Step1: Stitching

/home/liulab/labdata/scripts/Step_1_loop.sh

Step2: Registration

/home/liulab/labdata/scripts/Step_2_BatchRegistration.sh

Step3: rsFISH+warp

/home/liulab/labdata/scripts/Step_3_rsFISH.sh

Step4: cellpose

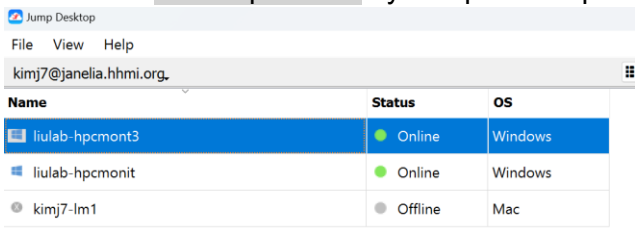
/home/liulab/labdata/scripts/Step_4_cellpose.sh

Step5: spot-to-cell assignment

/home/liulab/labdata/scripts/Step_5_spot_assignment.sh

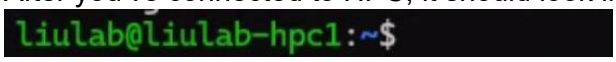
Connecting to the HPC and setting up your working directory

1. Connect to liulab-hpcmont3 by Jump-Desktop

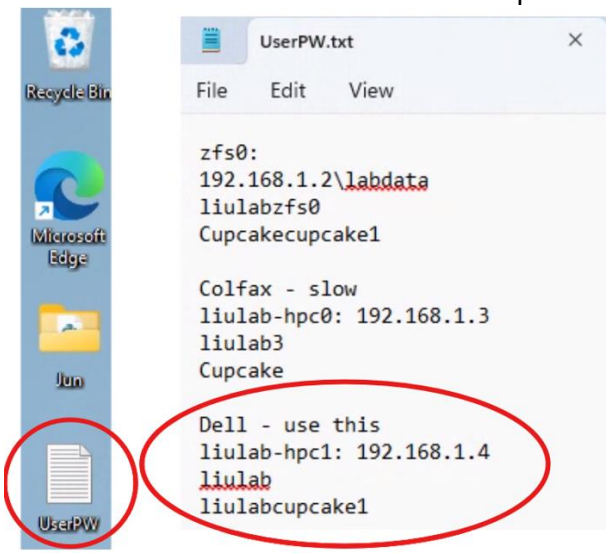


*This is a small PC are NOT the HPC. Use this PC to connect to the HPC, where raw data is stored, and computation happens

2. Connect to HPC using ssh
   a. Open ubuntu – you can find them through Search bar
   b. Type: ssh liulab@192.168.1.4
   c. password: liulabcupcake1
   d. After you've connected to HPC, it should look like this:



This information can be found in desktop 'UserPW.txt'



3. Once in HPC, create your own working directory. For example:
   a. Navigate to lab working directory: cd /home/liulab/labdata
   b. Create your own working directory: mkdir myworkdir

   c. Navigate to your working directory: cd myworkdir

Step1: nd2 to N5 (includes stitching)

The first step is to take each image (nd2 format) and convert to N5 file format. The script will stitch if the image contains multiple tiles. Each N5 is defined by batch number (b0 – Day1) and time number (b0/t0 – Day1, round 1). The script will loop through all batches and all timepoints present in raw data directory. The script can be run repeatedly. If a nd2 file is already processed, it will be skipped.

1. Copy Step1 script to your working directory:
   cp /home/liulab/labdata/scripts/Step_1_loop.sh .

```
liulab@liulab-hpc1:~/labdata$ cd myworkdir/
liulab@liulab-hpc1:~/labdata/myworkdir$ cp /home/liulab/labdata/scripts/Step_1_loop.sh .
liulab@liulab-hpc1:~/labdata/myworkdir$ ls
Step_1_loop.sh
liulab@liulab-hpc1:~/labdata/myworkdir$
```

2. Follow the below steps to specify the required parameters:
   a. Open the script using text editor:
      nano Step_1_loop.sh

```
 GNU nano 6.2                                  Step_1_loop.sh
inputpath="/home/liulab/labdata/TestingDataSets/test_set/"
outputpath="/home/liulab/labdata/myworkdir/step1_output/"
```

      Must specify inputpath and outputpath
   b. inputpath is the directory containing batches as subdirectories and nd2 files as timepoints

```
liulab@liulab-hpc1:~/labdata/myworkdir$ ls -lR /home/liulab/labdata/TestingDataSets/test_set/
/home/liulab/labdata/TestingDataSets/test_set/:
total 0
drwxrwxrwx 2 root root 0 Dec 17 07:30 b0
drwxrwxrwx 2 root root 0 Jul 24 11:55 b1

/home/liulab/labdata/TestingDataSets/test_set/b0:
total 36219446
-r-xr-xr-x 1 root root 12359753728 Jul 24 11:52 Time00000_Point0000_Channel640,561,488,405_Seq0000.nd2
-r-xr-xr-x 1 root root 12359753728 Jul 24 11:53 Time00001_Point0000_Channel640,561,488,405_Seq0002.nd2
-r-xr-xr-x 1 root root 12359753728 Jul 24 11:53 Time00002_Point0000_Channel640,561,488,405_Seq0004.nd2

/home/liulab/labdata/TestingDataSets/test_set/b1:
total 24146297
-r-xr-xr-x 1 root root 12359753728 Jul 24 11:55 Time00000_Point0000_Channel640,561,488,405_Seq0000.nd2
-r-xr-xr-x 1 root root 12359753728 Jul 24 11:56 Time00001_Point0000_Channel640,561,488,405_Seq0002.nd2
liulab@liulab-hpc1:~/labdata/myworkdir$
```

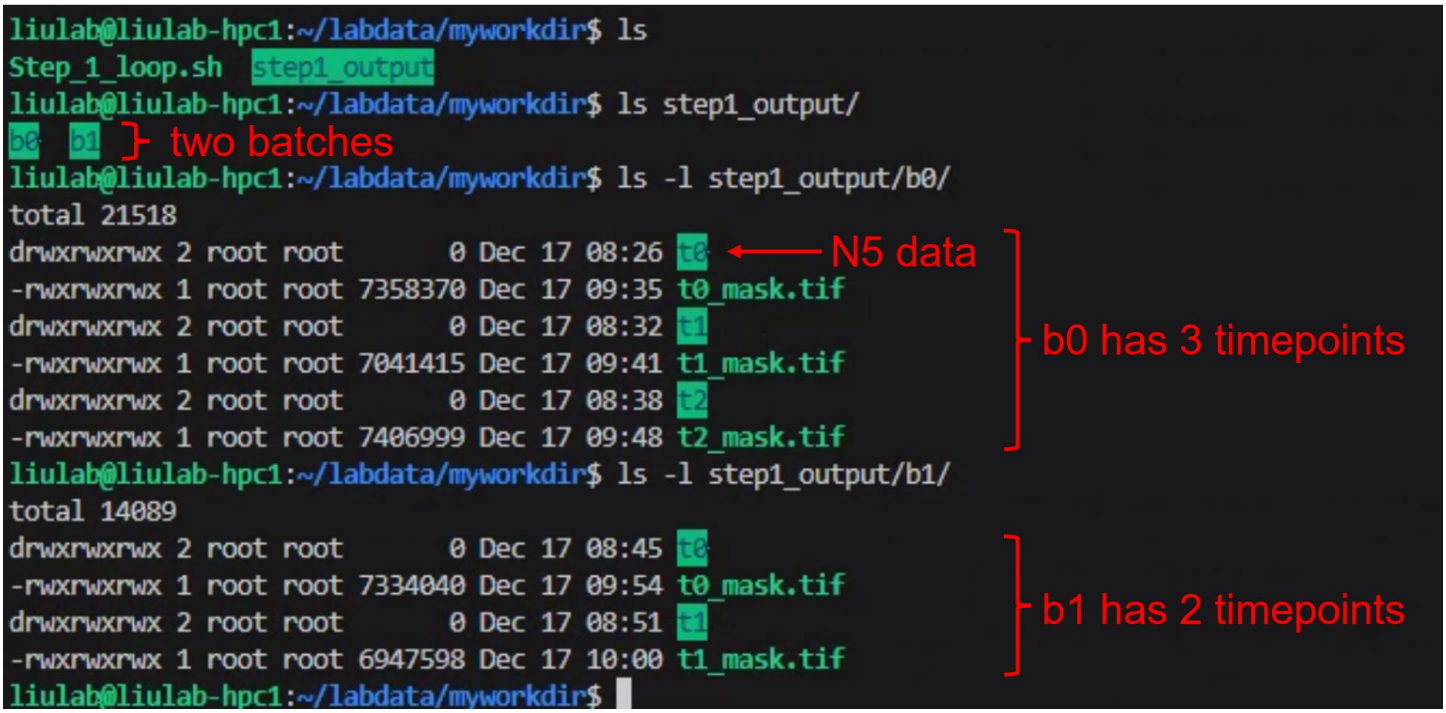
      inputpath has two batches as subdirectories b0, b1

      b0 has 3 timepoints (t0, t1, t2)

      b1 has 2 timepoints (t0,t1)

      *Naming of the batch directories (b0, b1, b2…) is important
      *Naming of the timepoints (nd2 files) is not important.
      The script assumes the sorted order to be t0, t1, t2 …
   c. outputpath is the step1 output directory to be created.
   d. Save (ctrl+s) and then Exit (ctrl+x) the text editor
3. Run the script:
   bash Step_1_loop.sh

Step1: output explanation



N5 data

- All channels (c0, c1, …) in the order of imaging. We image high wavelength first, so c0 is usually 640.
- Multi-resolution (s0, s1, …), if multi-tile large image was used as input. If single-tile image was used, only s0 (no down-sampling) will be present.
- Contains meta data (json file): voxel-spacing (micrometer / voxel) in xyz. Down-sampling factor in xyz at each resolution.
- Checkpoint file: empty file generated when nd2 file is successfully processed.

mask.tif

- Binary file indicating foreground and background of the image
- Can be used by Step2 to perform cross-round registration only using the foreground image

Step2: cross-round registration

The second step is to align different imaging rounds. Despite the fluidics system, the images of different rounds are not perfectly aligned. We use information from DAPI channel to 'move' the images such that different rounds are aligned in the image (xyz) coordinate. The script requires the user to select 'fix' image, a reference map to which other rounds will move. The script assumes all other rounds (batch and time) to be moving images. The script can be run repeatedly. If a round is already registered, it will be skipped.

1. Copy Step2 script to your working directory:
   `cp /home/liulab/labdata/scripts/Step_2_BatchRegistration.sh .`

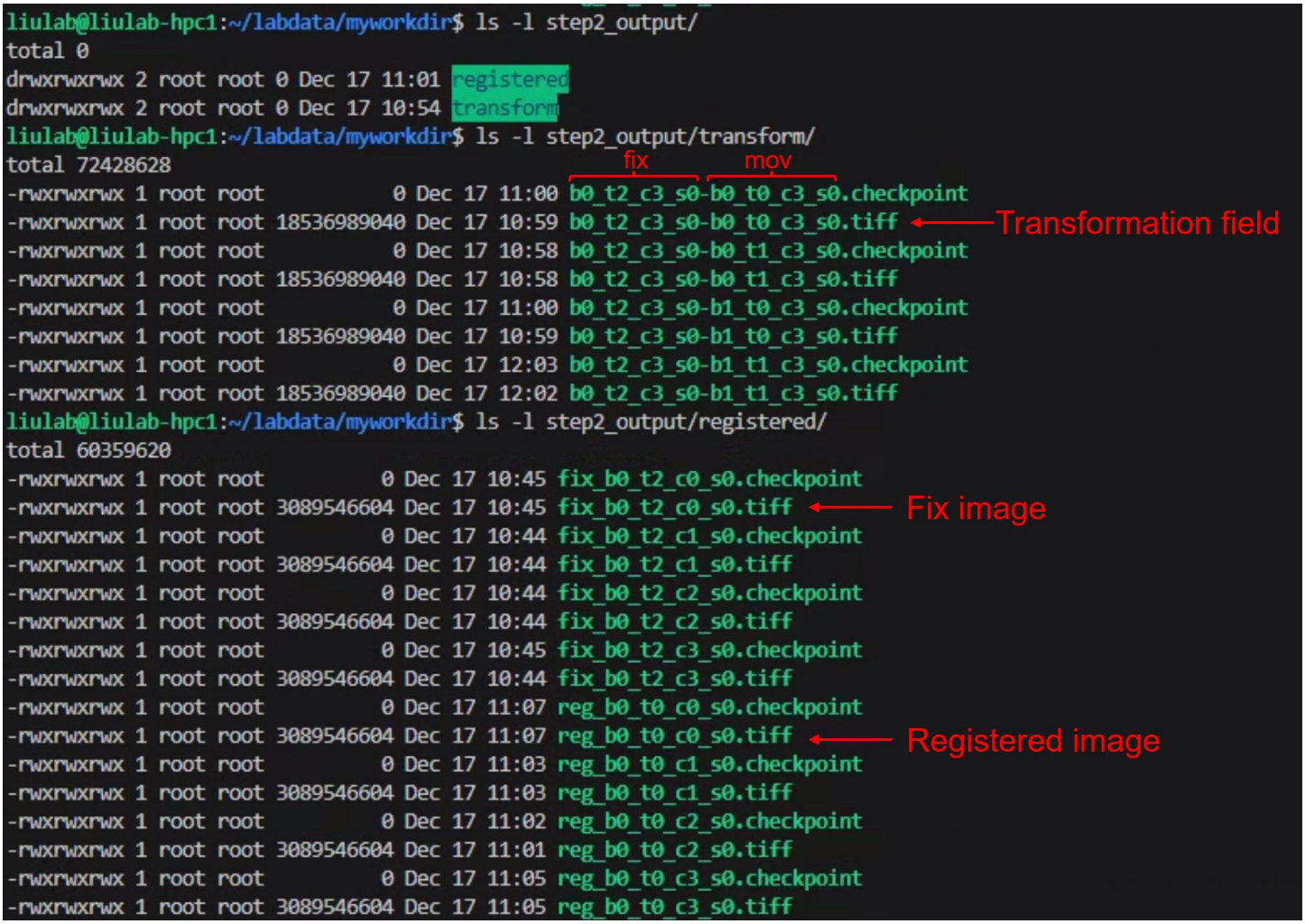
```
liulab@liulab-hpc1:~/labdata/myworkdir$ cp /home/liulab/labdata/scripts/Step_2_BatchRegistration.sh .
liulab@liulab-hpc1:~/labdata/myworkdir$ ls
Step_1_loop.sh  step1_output  Step_2_BatchRegistration.sh
liulab@liulab-hpc1:~/labdata/myworkdir$
```

2. Follow the below steps to specify the required parameters:
   a. Open the script using text editor:

```
 GNU nano 6.2          Step_2_BatchRegistration.sh
inputpath="/home/liulab/labdata/myworkdir/step1_output/"
outputpath="/home/liulab/labdata/myworkdir/step2_output/"
fix="b0/t2"
dapi="c3"
res="s0"
```

   b. `inputpath` contains batches (b0, b1, …) and time points (t0, t1, …) for each batch. They are generated from Step1 script.
   c. `outputpath` is the step2 output directory to be created
   d. `fix` is the non-moving N5 (defined by batch and timepoint), to which all other N5 will move to be aligned.
   e. `dapi` will usually be c3. The number corresponds to the order of lambda used in image acquisition. We usually image in the following order: 640 (c0), 561 (c1), 488 (c2), then 405 (c3). If only two channels were used, DAPI will be c1. For example, 561 (c0), then 405 (c3).
   f. `res` is the resolution at which the registration will be performed. We will use s0 (no down-sampling) for single tile images, s2 (down-sampled by 4,4,1) for large images with multiple tiles.
3. Run the script:
   `bash Step_2_BatchRegistration.sh`

Step2: output explanation



Transformation field

- One for each timepoint except for the fix timepoint
- Information on how each timepoint should move such that it aligns with fix timepoint
- 4-dimensional data. First 3-diemsnions have the same xyz as the fix timepoint. The fourth dimension contains xyz vector for each voxel.

Fix images

- This is the user specified reference timepoint
- N5 data simply resaved as tiff

Registered images

- All images have same xyz dimensions. The underlying sample is aligned to the fix timepoint. Use these images to directly visualize your data.

Checkpoint file

- empty file generated when transformation field or registration image is successfully generated.

Step3: calling smFISH spots

The third step is comprised of two steps: first calling the single-molecule spots and then registering the spots. The spots are called in unregistered s0 (N5 output of step1). They are then registered to the reference timepoint using the transformation field solved for cross-round registration in step2. The script can be run repeatedly. If spots are already called for a round, it will be skipped.

1. Copy Step3 script to your working directory:
   `cp /home/liulab/labdata/scripts/Step_3_rsFISH.sh .`

```
liulab@liulab-hpc1:~/labdata/myworkdir$ cp /home/liulab/labdata/scripts/Step_3_rsFISH.sh .
liulab@liulab-hpc1:~/labdata/myworkdir$ ls
Step_1_loop.sh  step1_output  Step_2_BatchRegistration.sh  step2_output  Step_3_rsFISH.sh
liulab@liulab-hpc1:~/labdata/myworkdir$
```

2. Follow the below steps to specify the required parameters:
   a. Open the script using text editor:

```
 GNU nano 6.2                                        Step_3_rsFISH.sh
step1dir=/home/liulab/labdata/myworkdir/step1_output/
step2dir=/home/liulab/labdata/myworkdir/step2_output/
step3dir=/home/liulab/labdata/myworkdir/step3_output/
dapi=c3
rsparam="--rsfish_gb_per_core 8 --rsfish_min 50 --rsfish_max 300 --rsfish_anisotropy 1.1 --rsfish_sigma 1.5 --rsfish_threshold 0.007"
```

   b. step1dir is the output directory of step1. s0 resolution from N5 will be used for calling spots
   c. step2dir is the output directory of step2. Transform field will be used to register called spots.
   d. step3dir is the step3 output directory to be created
   e. dapi will usually be c3. The number corresponds to the order of lambda during the image acquisition. We usually image in the following order: 640 (c0), 561 (c1), 488 (c2), then 405 (c3). If only two channels were used, DAPI will be c1. For example, 561 (c0), then 405 (c3).
   f. rsparam includes parameters that can be passed to rsFISH and bigstream.
3. Run the script:
   `bash Step_3_rsFISH.sh`

Step3: output explanation

```
liulab@liulab-hpc1:~/labdata/myworkdir$ ls -l step3_output/
total 0
drwxrwxrwx 2 root root 0 Dec 19 09:07 spots
drwxrwxrwx 2 root root 0 Dec 19 09:34 spots_registered
```

```
liulab@liulab-hpc1:~/labdata/myworkdir$ ls -l step3_output/spots
total 233383
-rwxrwxrwx 1 root root        0 Dec 19 07:49 spots_b0_t0_c0_s0.checkpoint
-rwxrwxrwx 1 root root  1822963 Dec 19 08:56 spots_b0_t0_c0_s0.csv
-rwxrwxrwx 1 root root  4485600 Dec 19 08:58 spots_b0_t0_c0_s0.txt
-rwxrwxrwx 1 root root        0 Dec 19 07:49 spots_b0_t0_c1_s0.checkpoint
-rwxrwxrwx 1 root root  4464063 Dec 19 08:55 spots_b0_t0_c1_s0.csv
-rwxrwxrwx 1 root root 10825300 Dec 19 08:58 spots_b0_t0_c1_s0.txt
-rwxrwxrwx 1 root root        0 Dec 19 07:49 spots_b0_t0_c2_s0.checkpoint
-rwxrwxrwx 1 root root  1730765 Dec 19 08:58 spots_b0_t0_c2_s0.csv
-rwxrwxrwx 1 root root  4250200 Dec 19 08:58 spots_b0_t0_c2_s0.txt
```

spots called on step1 output

```
liulab@liulab-hpc1:~/labdata/myworkdir$ ls -l step3_output/spots_registered/
total 106106
-rwxrwxrwx 1 root root     2253 Dec 19 09:32 b0_t0.tmp
-rwxrwxrwx 1 root root     2253 Dec 19 09:33 b0_t1.tmp
-rwxrwxrwx 1 root root     1919 Dec 19 09:33 b0_t2.tmp
-rwxrwxrwx 1 root root     2253 Dec 19 09:33 b1_t0.tmp
-rwxrwxrwx 1 root root     2253 Dec 19 09:34 b1_t1.tmp
-rwxrwxrwx 1 root root        0 Dec 19 09:45 fix_spots_b0_t2_c0_s0.checkpoint
-rwxrwxrwx 1 root root  3371048 Dec 19 09:45 fix_spots_b0_t2_c0_s0.csv
-rwxrwxrwx 1 root root        0 Dec 19 09:45 fix_spots_b0_t2_c1_s0.checkpoint
-rwxrwxrwx 1 root root  2041209 Dec 19 09:45 fix_spots_b0_t2_c1_s0.csv
-rwxrwxrwx 1 root root        0 Dec 19 09:45 fix_spots_b0_t2_c2_s0.checkpoint
-rwxrwxrwx 1 root root  5841915 Dec 19 09:45 fix_spots_b0_t2_c2_s0.csv
-rwxrwxrwx 1 root root        0 Dec 19 09:14 reg_spots_b0_t0_c0_s0.checkpoint
-rwxrwxrwx 1 root root  3059869 Dec 19 09:14 reg_spots_b0_t0_c0_s0.csv
-rwxrwxrwx 1 root root        0 Dec 19 09:09 reg_spots_b0_t0_c1_s0.checkpoint
-rwxrwxrwx 1 root root  7410654 Dec 19 09:09 reg_spots_b0_t0_c1_s0.csv
```

Fix image spots

Registered image spots

spots

- rsFISH output. The spots are called on N5 data

Fix image spots

- This is simply a resave of spots file, since the spots for fix image do not need to be registered

Registered image spots

- Spots files are registered using the transformation field.
- The registered spots (ex. reg_spots_b0_t0_c0_s0.csv) should align with registered image (ex. reg_b0_t0_c0_s0.tiff)

Checkpoint file

- empty file generated when transformation field or registration image is successfully generated.

Step4: segmentation

Fourth step is to define what a cell is. DAPI channel will be used for this purpose. This step requires the user to download cellpose 2.0 and train a custom model using human-in-the-loop strategy. Because xy and yz/xy image view can differ, the user has the option to train two separate models to evaluating xy or yz/xy.

1. Prepare custom cellpose models.

```
liulab@liulab-hpc1:~/labdata/myworkdir$ ls cellpose_models/
CP_noxy  CP_xy
```

2. Copy Step4 script to your working directory:
   cp /home/liulab/labdata/scripts/Step_4_cellpose.sh .

```
liulab@liulab-hpc1:~/labdata/myworkdir$ cp /home/liulab/labdata/scripts/Step_4_cellpose.sh .
liulab@liulab-hpc1:~/labdata/myworkdir$ ls
cellpose_models  Step_1_loop.sh  step1_output  Step_2_BatchRegistration.sh  step2_output  step3_output  Step_3_rsFISH.sh  Step_4_cellpose.sh
liulab@liulab-hpc1:~/labdata/myworkdir$
```

3. Follow the below steps to specify the required parameters:
   a. Open the script using text editor:

```
 GNU nano 6.2                                Step_4_cellpose.sh

/home/liulab/labdata/scripts/cellpose.sh \
        -i /home/liulab/labdata/myworkdir/step1_output/b0/t2 \
        -o /home/liulab/labdata/myworkdir/step4_output/seg_b0_t2_c3_s2.tif \
        -c c3 -s s0 -m 400 \
        --model_xy /home/liulab/labdata/myworkdir/cellpose_model/CP_xy \
        --model_yz /home/liulab/labdata/myworkdir/cellpose_model/CP_noxy
```

   b. -i, --input is N5 data used as the reference timepoint during registration
   c. -o, --output is the output file to be generated
   d. -c, --channel is the DAPI channel of the N5 data
   e. -s, --scale is the resolution to be used in the N5 data
   f. -m, --minseg is the minimum size of segment (in voxel units)
   g. --model_xy is a custom model to be evaluated on xy slice over z.
   h. --model_yz is a custom model to be evaluated on yz slice over x.
4. Run the script:
   bash Step_4_cellpose.sh

Step4: output explanation

```
liulab@liulab-hpc1:~/labdata/myworkdir$ ls -l step4_output/
total 3017981
-rwxrwxrwx 1 root root 3089546508 Dec 19 13:34 seg_b0_t2_c3_s2.tif
```

The segmentation tif contains unique identifier for each segment. Background = 0, cellID = 1,2,3…

Step5: spot-to-cell assignment

Fifth step is to assign registered spots to the cell. The script keeps only spots that are located within the segment. The segments can be dilated / eroded.

1. Copy Step4 script to your working directory:
   cp /home/liulab/labdata/scripts/Step_5_spot_assignment.sh .
2. Follow the below steps to specify the required parameters:
   a. Open the script using text editor:

```
 GNU nano 6.2                                  Step_5_spot_assignment.sh
spots_registered="/home/liulab/labdata/myworkdir/step3_output/spots_registered/"
segmentation_tif="/home/liulab/labdata/myworkdir/step4_output/seg_b0_t2_c3_s2.tif"
step5dir="/home/liulab/labdata/myworkdir/step5_output/"
```

   b. spots_registered is the directory of containing registered spots from step3 output
   c. segmentation_tif is the segmentation data from step4 output
   d. step5dir is the output directory to be created

Step5: output explanation

```
liulab@liulab-hpc1:~/labdata/myworkdir$ ls -l step5_output/
total 649
-rwxrwxrwx 1 root root 611637 Dec 19 17:09 cell_by_transcript_matrix.csv
-rwxrwxrwx 1 root root    623 Dec 19 17:09 percent_spots_assigned.csv
liulab@liulab-hpc1:~/labdata/myworkdir$
```

Cell_by_transcript_matrix.csv

The columns are timepoints (different genes that were imaged) and the rows are individual cells (segmented from Step4). The values indicate the number of spots detected per cell.