# **SOP for ASF/BABS**

# **Background info**

<https://github.com/bahnk/SlideSeq>

The slideseq preprocessing pipeline matches sample reads to their locations of the pucks and enables the researcher to perform a downstream single cell analysis on the data.

## Puck info

For each puck there’s microscope and sequencing data.

The pucks have barcode and coordinates (X and Y) info.

Crick pucks:

* info is found in a csv file under the spatial column (Airtable -> pucks)

Curio pucks:

* find corresponding puck in <https://curiobioscience.com/support/barcode/> by inputting the barcode info (eg. in “Curio\_Tile\_A0017\_006” the barcode is “A0017\_006”);
* download tsv file
* convert it to csv and add a header with columns “Barcode,x,y”

## Folder structure

PI/SCIENTIST/LIMSID/

DOCS/

PROCESSING/

ANALYSIS/

This suggested directory structure allows for the preprocessing to be stored separate from the downstream analysis, while still being kept together under one overall project directory.

This is not essential for the pipeline to work.

# **Set up and Running the pipeline**

1. Store puck info (csv file with barcode and coordinates) in *processing* – this can be found using the pucks ID. Pucks ID is usually either in the email body or in the sample spreadsheets from the ASF
2. Softlink the fastq files in *processing/pucks* – not essential, but suggested for a smoother/simpler run and troubleshooting
3. Create samplesheet (instructions in the github slideseq pipeline docs):
   * Header: “name,fastq\_1,fastq\_2,puck,read\_structure,genome,gtf”
   * Insert information as requested
   * For the read structure:
     + Curio samples have only have one structure: “8C18U6C9M1X”
     + Crick samples might have different structures. It’s good to check but recently they seem to have used consistently “8C18U6C9M1X”. they can be checked this way:
       - Go in airtable -> pucks
       - Within the acquisition column, click on the desired puck
       - Scroll and find read structure (A Structure)
   * Genomes used now are the ones in reference/ … /star/100bp etc.

If it’s a standard run, go ahead and run pipeline as described on github. The whole pipeline should run on tmux without any memory or time issues.

If not standard, things to look for:

* + YALM file doesn’t normally need any changes. If the scientists want a different value for the umi, this would have to be manually edited
  + Normal config is usually fine. Runs with large read numbers (Eg. >100M reads) will require to be run on the *highmem* config instead