Visium HD – running Space Ranger

Running it with default settings

Running on one sample, one slide: spaceranger.single.sh. We'll use this today.

FYI running on multiple samples and slides: spaceranger.array.sh

For this tutorial, we'll use inputs from the mouse embryo example on 10X: Visium HD Spatial Gene Expression Library, Mouse Embryo (FFPE) | 10x Genomics. The probe set is already provided: `Visium_HD_Mouse_Embryo_probe_set.csv`. Take note of:

- Slide serial number: H1-CQWQN2G
- Area: D-1

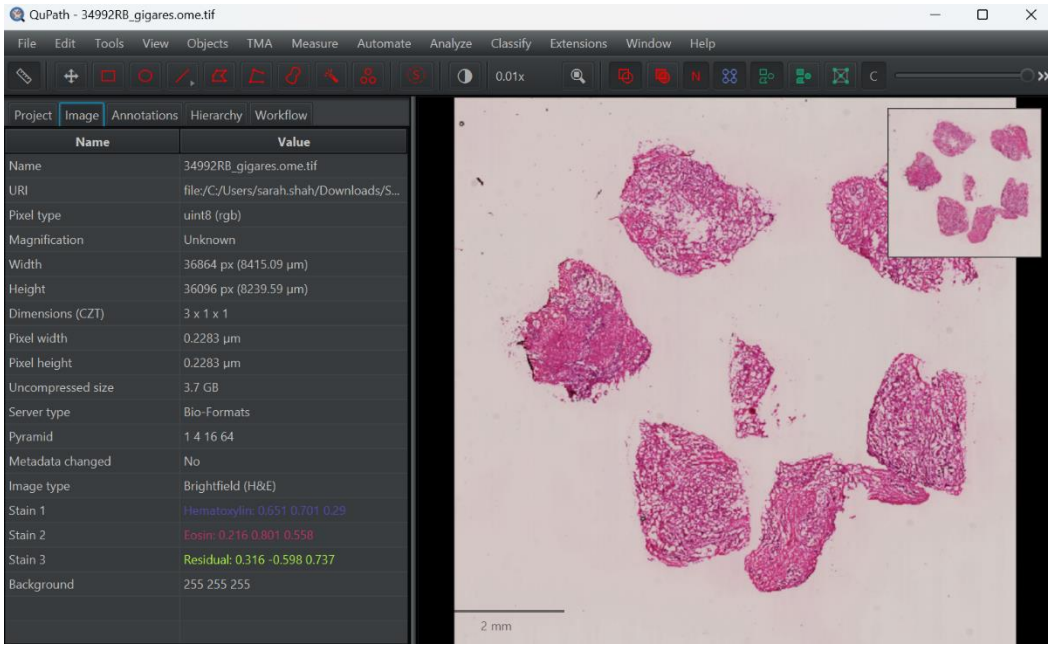
The reference transcriptome folder can be downloaded from here: Download Space Ranger | Official 10x Genomics Support. Choose Mouse reference GRCm39-2024-A.

General Inputs

File	**Source**
Microscopy image	User
CytAssist	User
FASTQ files	User
Reference transcriptome folder	10X
Reference probe csv file	10X

Your microscopy images should be greater than 2000 pixels in width and height, with pixel size equal or lower than 0.25 micrometers. The example datasets on 10X have microscope image format .BTF, but some microscopes save as .NDPI. You can export them as OME.TIF:

1. Open the NDPI file using QuPath
2. Go to File -> Export images -> OME.TIF
3. There would be a pop up box asking for compression format. Choose ZLIB (lossless), tile size 1024 px, and keep pyramidal to 4.0.



Space Ranger command and SLURM script setup

```
spaceranger count \
    --fastqs $fastqdir \
    --transcriptome $transcriptome \
    --id $id \
    --sample $sample \
    --image $micimage \
    --cytaimage $cytaimage \
    --slide $slideid \
    --area $area \
    --probe-set $probes \
    --jobmode local \
    --create-bam false \
    --output-dir $outDir \
    --localcores ${SLURM_CPUS_PER_TASK} \
    --localvmem ${SLURM_MEM_PER_NODE}
```

- We're not keeping the BAM because they are big, and we don't expect to troubleshoot alignments.

- `$sample` is the common start of the name of your FASTQ files.
- To keep SLURM usage efficient, run a small dataset and record the usage (e.g., `seff $jobid`) and scale up for your bigger datasets.
- An intermediate file will be created in your working directory `__tutorial_Visium_HD_Mouse_Embryo_D1_H1-CQWQN2G.mro`. A good way to tell if your array runs aren't overwriting each other is that there should be unique `.mro` files formed for each of your sample-slide-area combo.

On Pawsey, it took me 12 hours to complete running Space Ranger for this example, using `#SBATCH --mem=40G` and `#SBATCH --cpus-per-task=12`.

Training accounts on Pawsey

Sarah Beecroft from Pawsey has generously provided you with temporary accounts.

It would have $HOME, $MYSOFTWARE, $MYSCRATCH attached.

Your account: #SBATCH --account=courses01

Your reservation: #SBATCH --reservation=EmblSagc

Important outputs

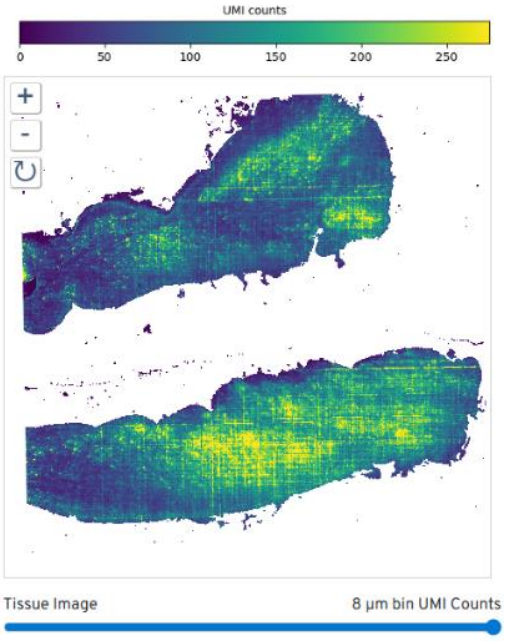
For viewing in Loupe Browser:

- `outs/segmented_outputs/cloupe.cloupe`
- `outs/binned_outputs/square_008um/cloupe.cloupe`

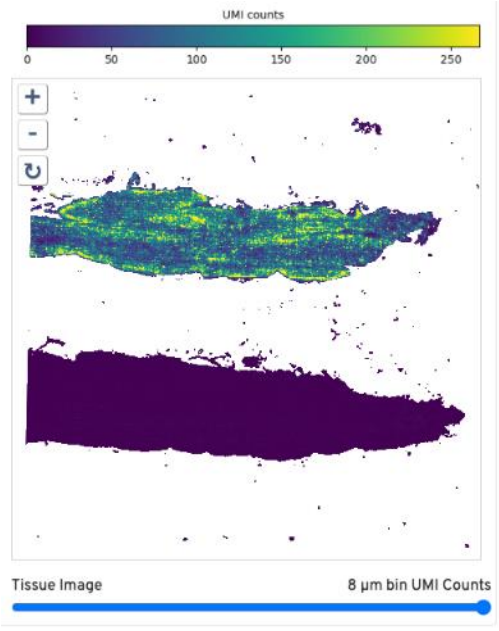
For quick viewing of summary:

- `outs/web_summary.html`

  Slide to 8um bin UMI Counts and you should see UMI counts.
  A good example:



A bad example:



Beware of Errors and Warnings. The following is an example of a bad run, where I used microscopy images barely above 2000 pixels:

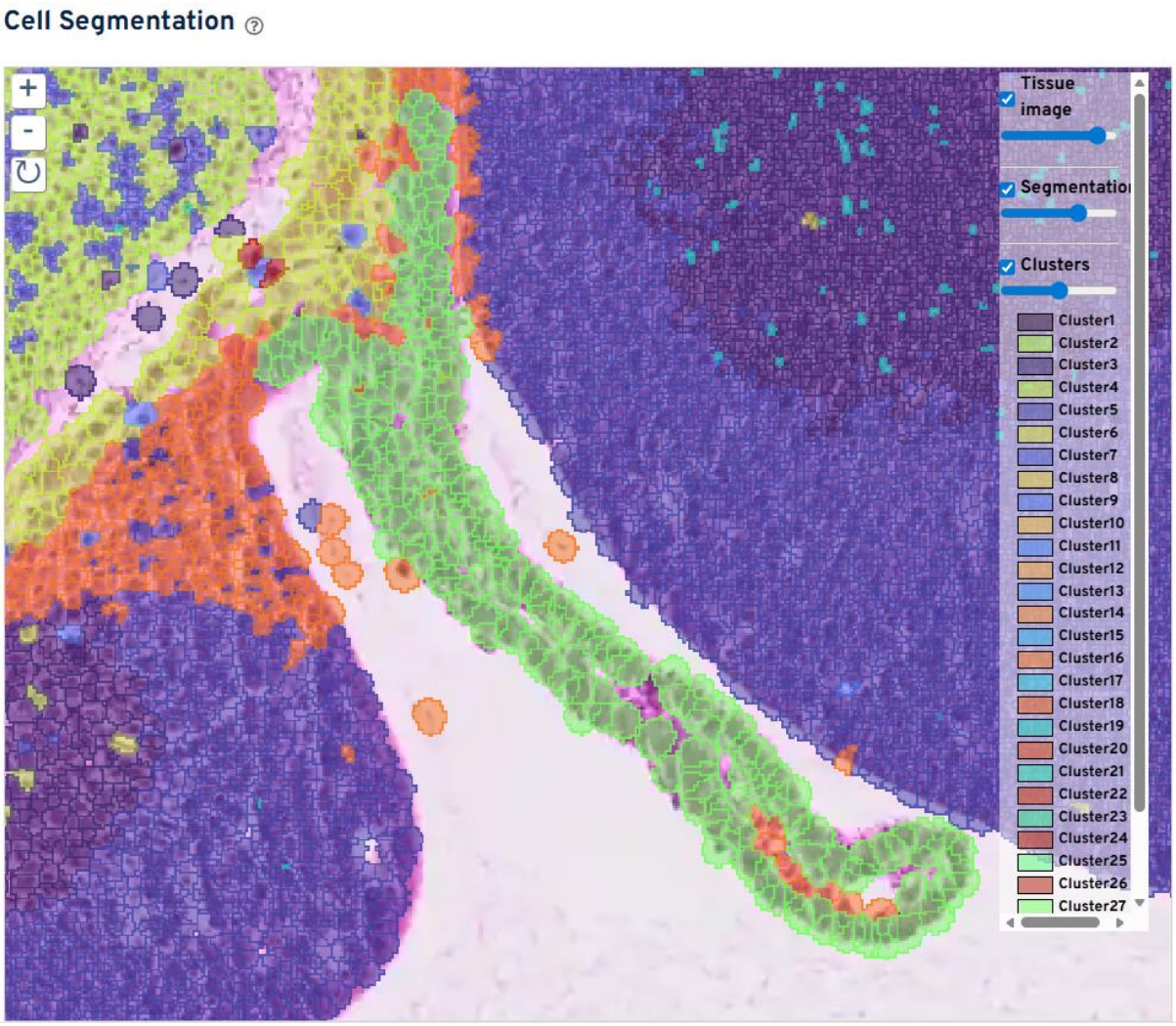
Alerts

The analysis detected 3 errors and 2 warnings.

	Alert	Value	Detail
!	Low Fraction of Reads Mapped Confidently to the Probe Set	0.0%	Ideal > 50%. This can indicate use of the wrong probe set, the use of FASTQs from a poly-A based assay, or low aggregate expression. Performance may be affected.
!	Low Fraction Reads Confidently Mapped to the Filtered Probe Set	0.0%	Ideal > 50%. This can indicate use of the wrong probe set, the use of FASTQs from a poly-A based assay, or low aggregate expression. Performance may be affected.
!	Low Fraction Gene Expression Reads in Squares	None	Ideal > 75%. Application performance may be affected. Many of the reads were not assigned to tissue covered squares. This could be caused by high levels of ambient RNA resulting from inefficient permeabilization, because the incorrect image was used, or because of poor tissue detection. The latter case can be addressed by using the manual tissue selection option through Loupe. *Ignore this alert if you manually selected a subset of the tissue in the capture area, such as a single core punch in a TMA.*
⚠	High Fraction Reads Half-Mapped to Probe Set	98.7%	Ideal < 20%. This can indicate low RNA content in the sample, poor washing after probe hybridization, deviation from recommended protocol during probe hybridization, or suboptimal sample preparation.
⚠	UMI Registration Warning	True	The automatic UMI-based registration could not determine a confident offset, so the position of the UMIs relative to the image has not been adjusted. If you do in fact observe misalignment between the high-resolution H&E and UMI images, please consider manually adjusting the alignment using the --umi-to-image-offset parameter.

Our mouse embryo web summary had:

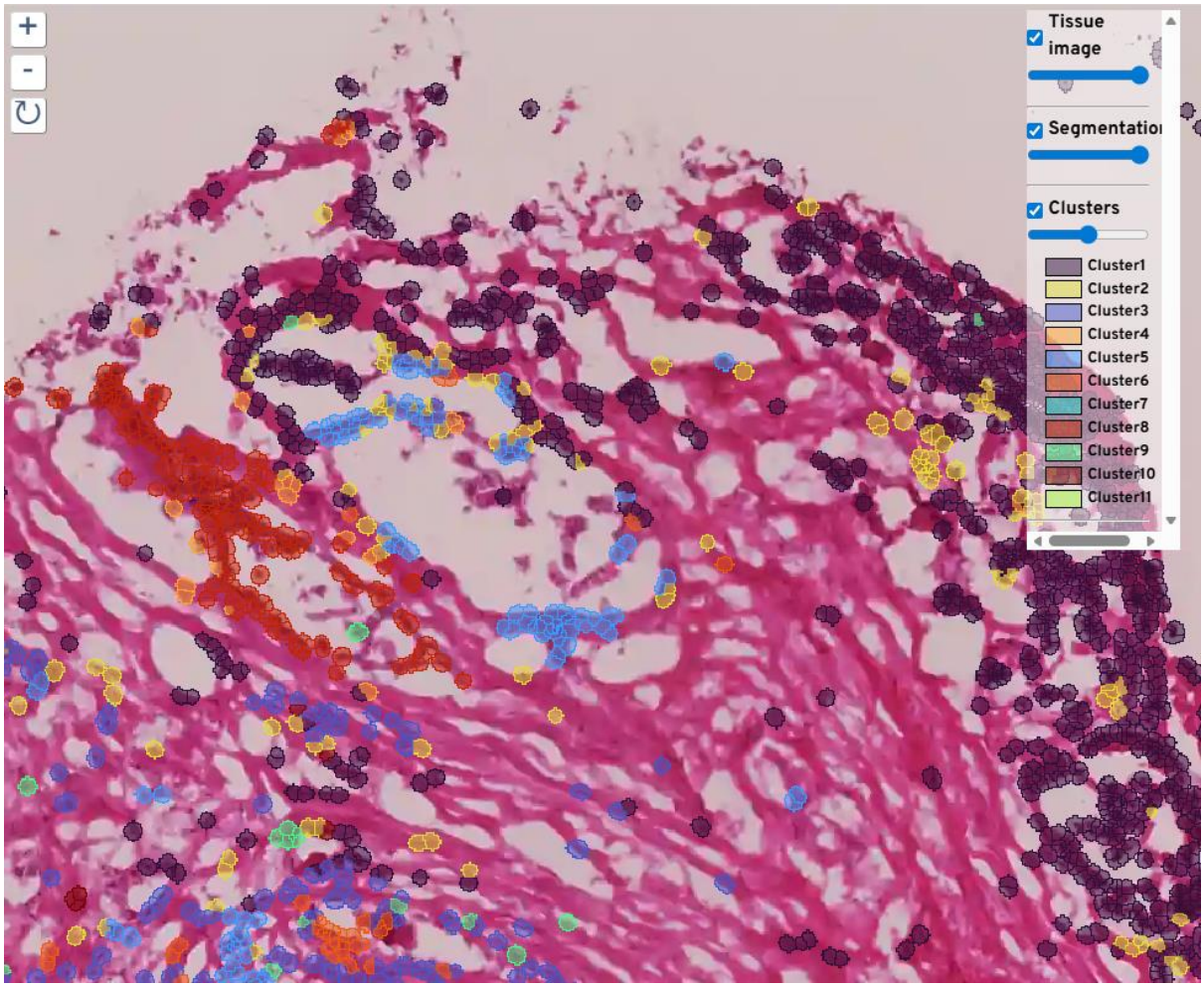
- No alerts
- 716.9 mean UMIs per bin: proxy for genes per bin, i.e., sensitivity
- Sequencing saturation plot shows saturation is below 0.4, i.e., if sequencing depth was increased there'd be more UMI's detected (see CG000686_VisiumHDSpatialApplicationsPerformance_TechnicalNote_RevB.pdf)
- Cell-segmentation matches cells – although double-checking with a pathologist or someone who knows anatomy is recommended.



- UMAP projection can look slightly different as it uses a stochastic algorithm and thus not deterministic (see UMAP: Uniform Manifold Approximation and Projection - GeeksforGeeks, Are umap transformations non-deterministic? · Issue #158 · lmcinnes/umap). Thus, the Clusters may be assigned different colours and be slightly different too.

Re-running Space Ranger if cell-segmentation is bad

Our mouse embryo example outputs looked good. However, look at the following snippet from the web summary of a prostate sample. What's bad about it?



Many cells were not recognized because they were:

- Faint nuclei
- Long fibrous cells
- Gaps between cells caused by contraction during sample preparation, or perhaps that's just how the tissue really looks like.

You must play around with manual alignment, search or create your own cell segmentation training model, or adjust the following parameters and re-run Space Ranger.

We won't do the following steps in this class, but I recommend them if you encounter issues.

Inputs for custom nuclei segmentation and binning:

Detailed here: [Nuclei Segmentation and Custom Binning of Visium HD Gene Expression Data | 10x Genomics]

- `outs/binned_outputs/square_008um/filtered_feature_bc_matrix.h5`

- `outs/binned_outputs/square_002um/spatial/tissue_positions.parquet`
- Your microscopy OME.TIF or BTF image

Important cell-segmentation parameters:

Parameter	Description
`--custom-segmentation-file`	Providing this will override 10X's default nuclei-expansion model. For example, open an image in QuPath, select a region, run model InstaSeg, and then when satisfied with cell boundaries, export as GeoJSON.
`--nucleus-segmentation`	Turn it false when you don't want nucleus segmentation.
`--max-nucleus-diameter-px`	Increase this if you expect massive cells.
`--nucleus-expansion-distance-micron`	Like --max-nucleus-diameter-px, set a distance for how Space Ranger will assign a barcode to a nucleus.

Manual Alignment

Additionally, open the CytAssist and microscopy image in Loupe Browser via "Launch Visium Image Alignment". Follow this 10X tutorial: Visium HD Manual Alignment | Official 10x Genomics Support. Feed the exported JSON into parameter `--loupe-alignment` in `spaceranger count`.

Visium HD downstream analysis in Loupe Browser & R

Pre-requisites

- Download Loupe Browser 9.0: Loupe Browser | Official 10x Genomics Support
- Either download Mouse Embryo example dataset output (just Loupe file with cell segmentation) from 10X Visium HD Spatial Gene Expression Library, Mouse Embryo (FFPE) | 10x Genomics, or use yours from Space Ranger.
- Get ready to download files from our GitHub repo: https://github.com/sagc-bioinformatics/EMBL_workshop_2025

Loupe Browser

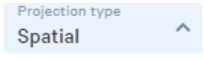
Step 1: Filtering & Reclustering

Following the tutorial: Filtering and Reclustering Workflow -Software -Single Cell Gene Expression -Official 10x Genomics Support

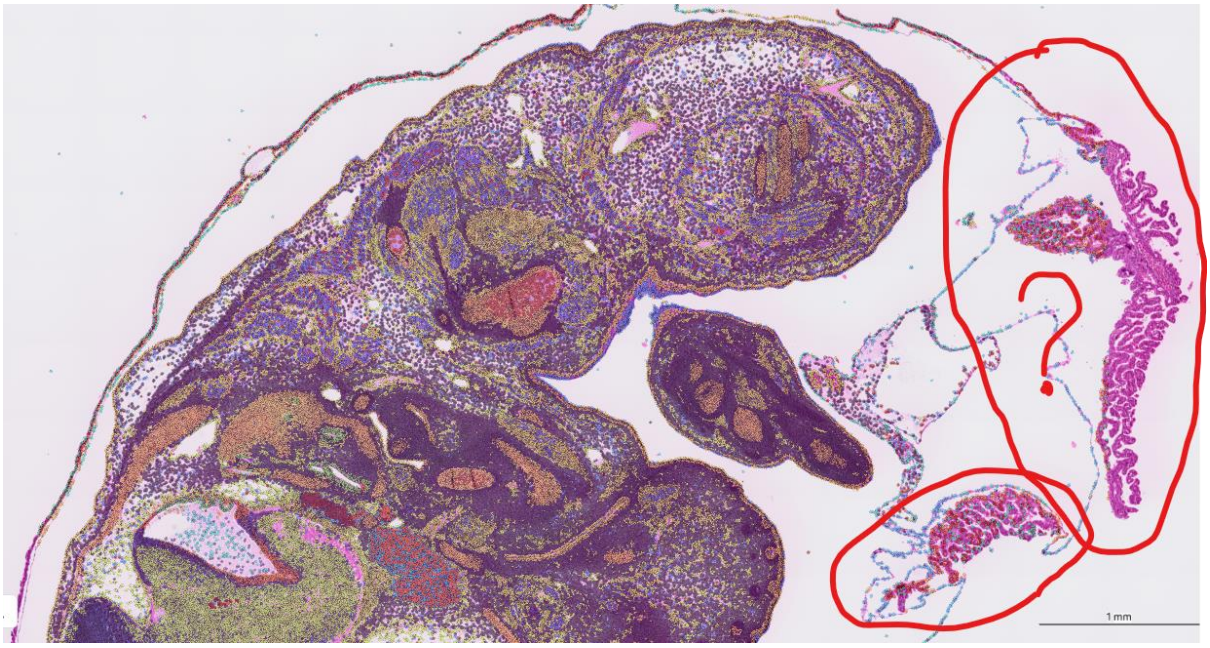
Let's use the cell segmented cloupe file. Looks better delineated than the binned version. Visium's cell segmentation works on recognizing nuclei and expanding outwards.

1. Load the cell segmented cloupe file into Loupe Browser.

   Play around with the view:

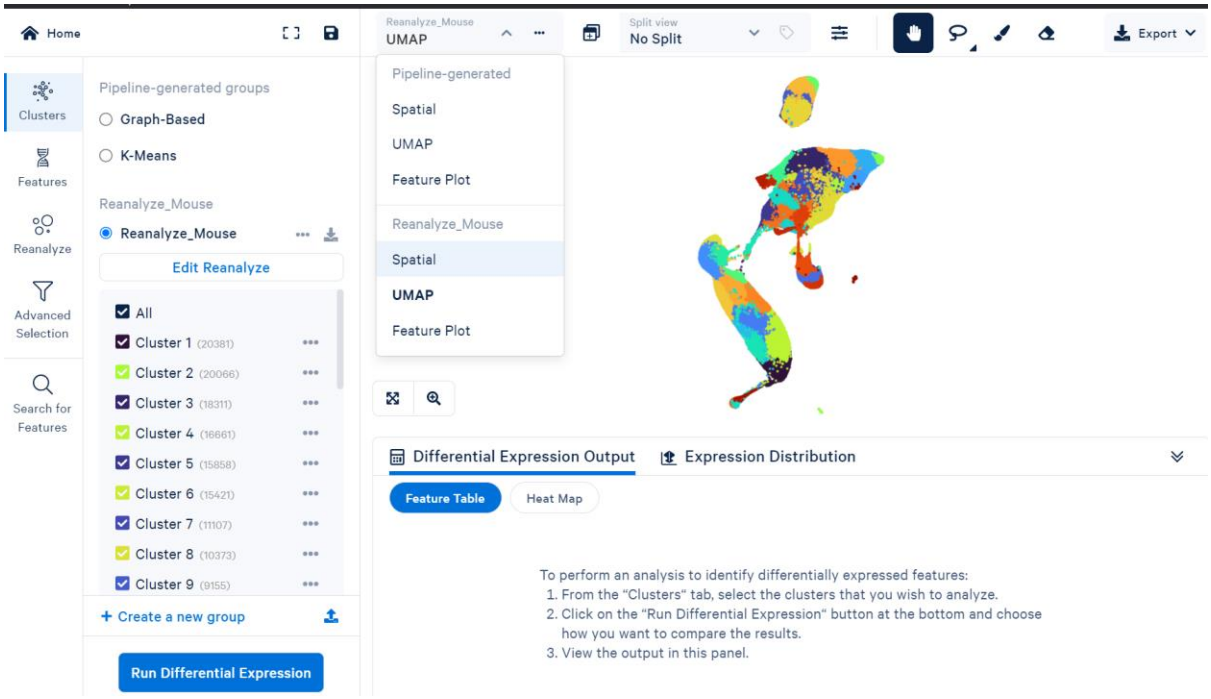
   - Switch between Graph-based and K-Means. How many clusters do you have? I have 30 and 5 respectively.
   - Increase/Decrease Spot Opacity.
   - Projection settings : turn off/on Tissue Display, play with Rotation angles.
   - Look at Projection type 

UMAP for fun.
2. Some clusters are on tissues I have no idea about, so I am going to remove them.



- Un-select the unwanted clusters.
- Click Reanalyze 

, a new window pops up.
- Skip through Review Spot Barcodes.
- Let’s choose UMI threshold of up to 7000. Some researchers choose to be strict, choosing only 5%-95% of the total range of UMI.
- Skip through Threshold by Features.
- We don’t have mitochondrial UMI’s.
- Choose UMAP option, rename the analysis to Reanalyze_Mouse, press Recluster.



In Reanalyze_Mouse, there are now more clusters.

Step 2: Identifying cell types and creating custom clusters

Following the tutorial: [Mapping the Tumor Microenvironment with Visium HD and Loupe Browser | Official 10x Genomics Support]

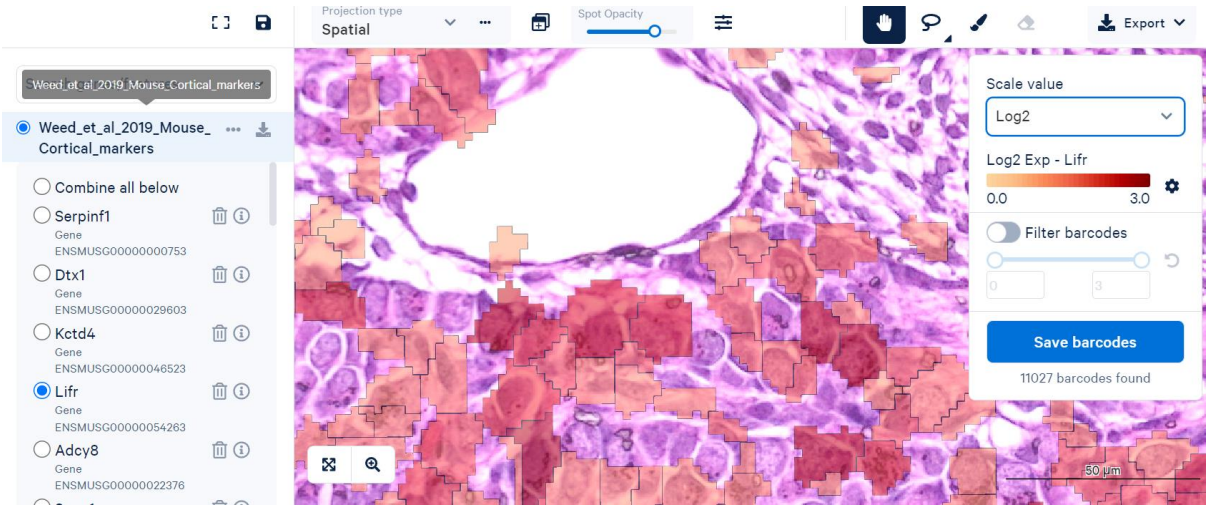
Let's try to locate the temporal lobe and find interesting gene expression there. I looked at Weed et al. 2019 https://doi.org/10.1371/journal.pone.0212898 and got a list of cortical marker genes Weed_et_al_2019_Mouse_Cortical_markers.csv.

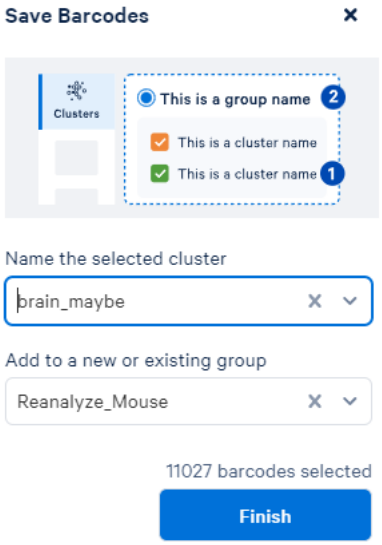
Click on *Lifr*, then *Necab2,* then *Cacnb3*. I assume they belong to the temporal lobe.

Some transcripts are mislocalized. Let's clean them.

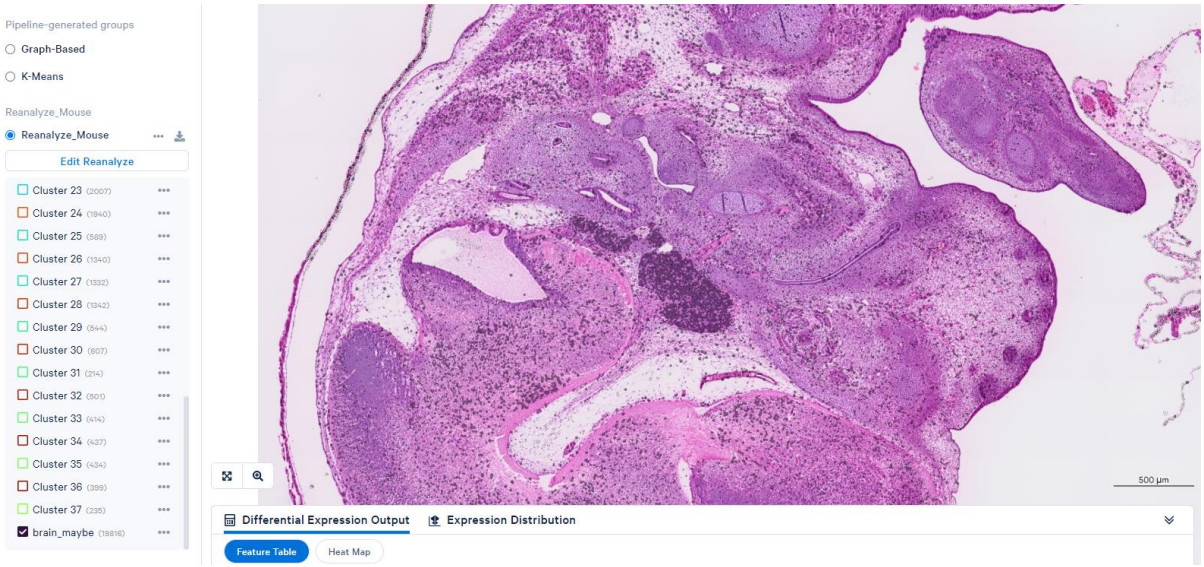
1. Filter barcode minimum for *Lifr* to 1, *Necab2* to 1, and *Cacnb3* to 2. Click on 

to save each of them as new cluster named *brain_maybe.*





The new custom *brain_maybe* cluster looks like this:



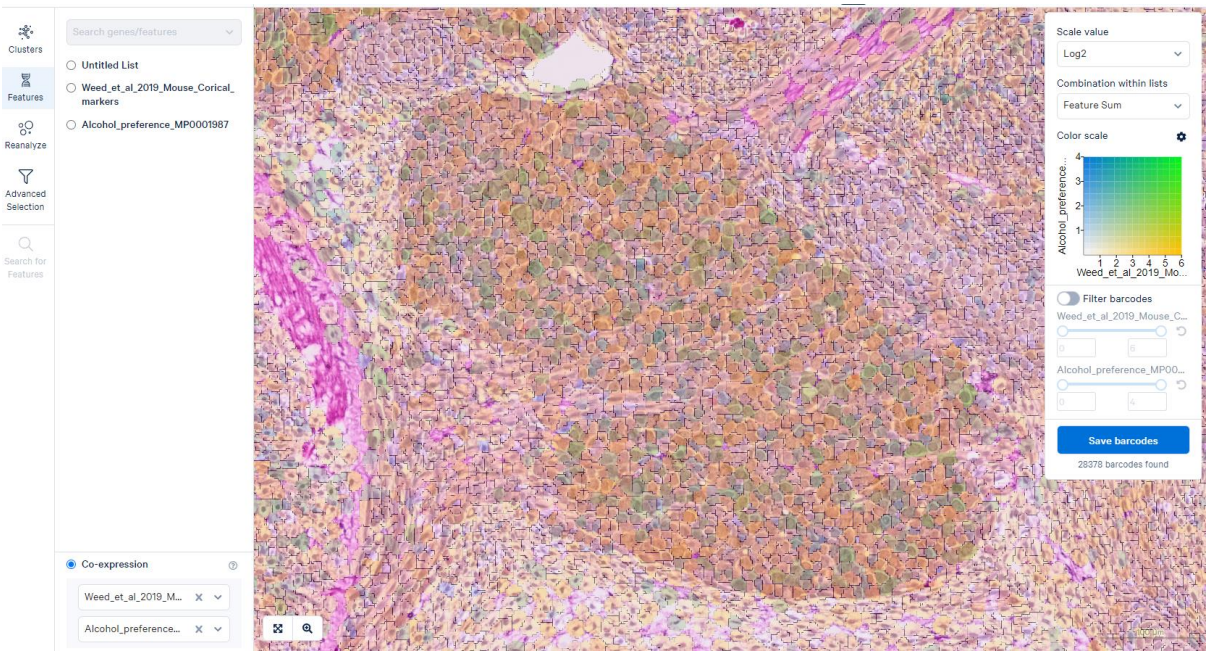
2. To prepare for differential gene expression analysis, import a second set of genes from Alcohol_preference_MP0001987.csv. I chose this gene set because I thought it was funny that the mouse cell line in this example C57BL/6 is described as, “Unlike most mouse strains, it drinks alcoholic beverages voluntarily”. Choose Combine all (the genes in this list), choose filter minimum to 2. Save as new alcohol_preference cluster.

Be careful deleting clusters, as it becomes greyed out and hard to bring back!

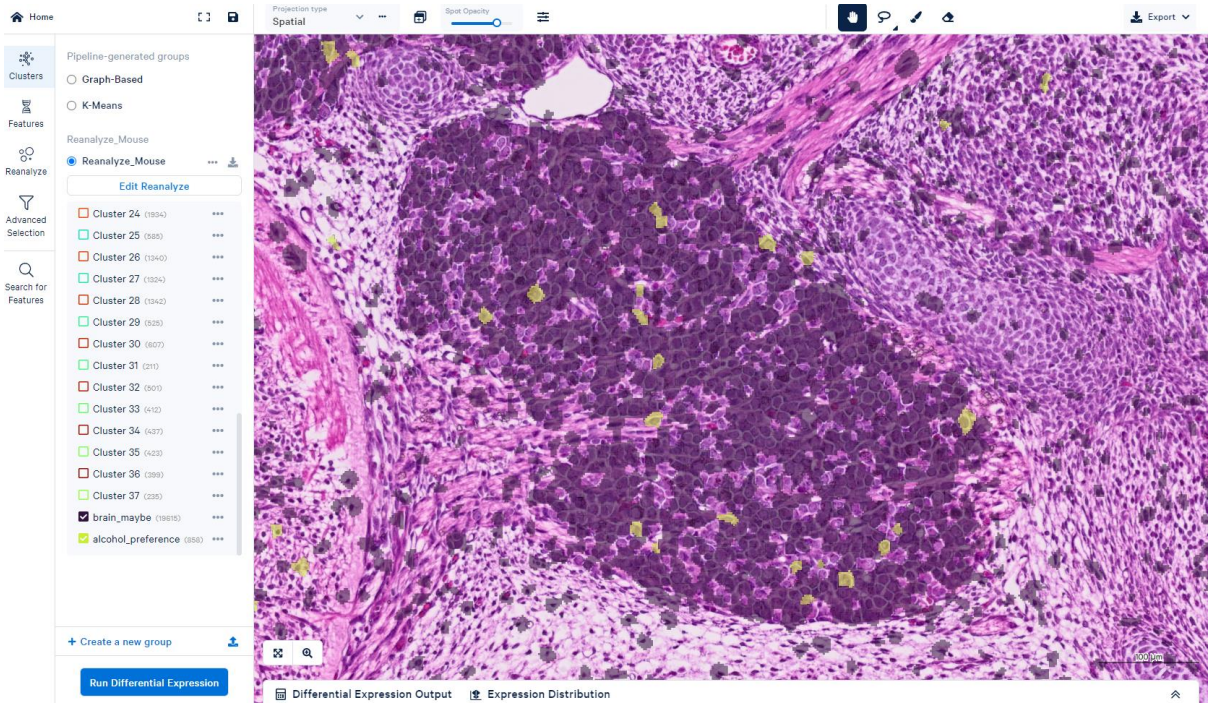
Step 3: Differential Gene Expression analysis

Here, let’s see if alcohol preference genes are differentially expressed in what we think is the temporal lobe.

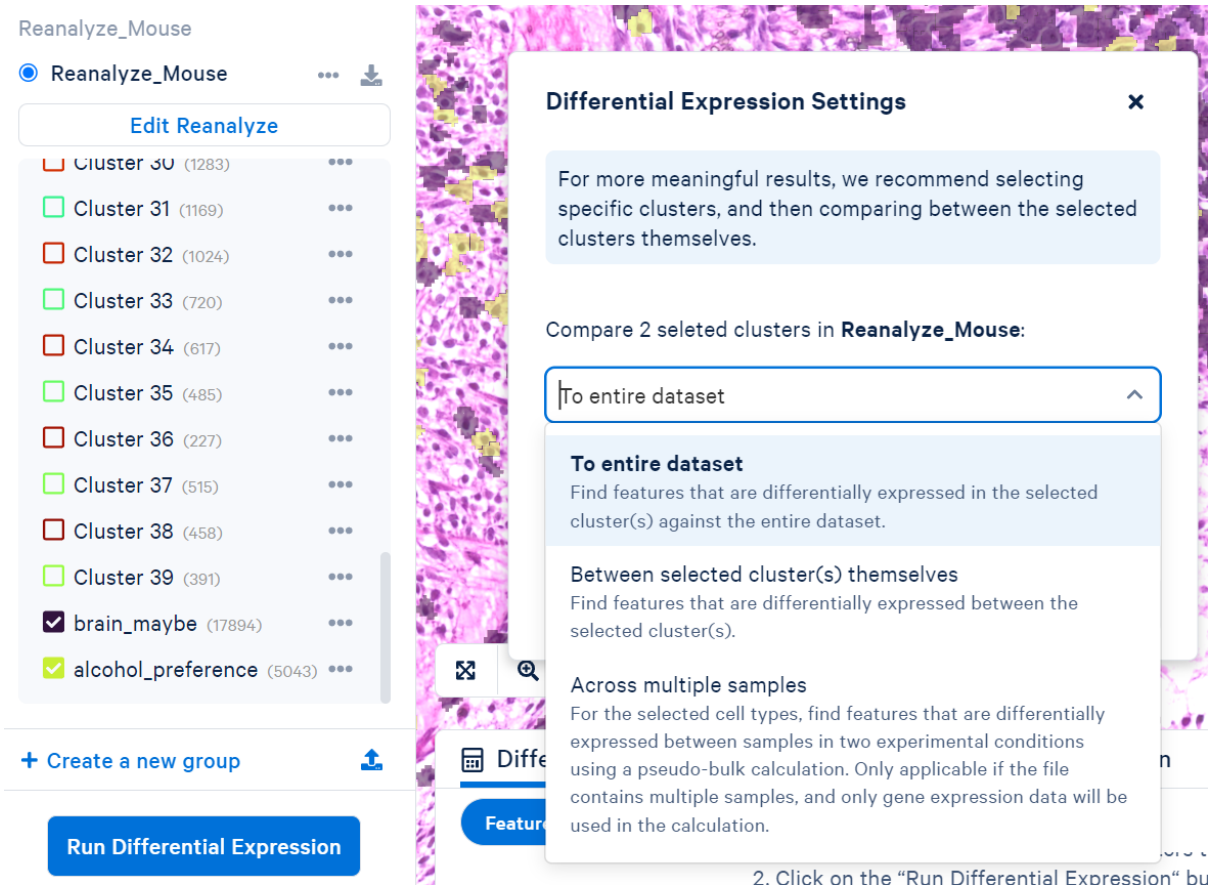
1. In the Features tab: although we won’t use this feature today, click on Co-expression and choose the cortical marker and alcohol preference genes sets. This shows co-expression of both sets in green.



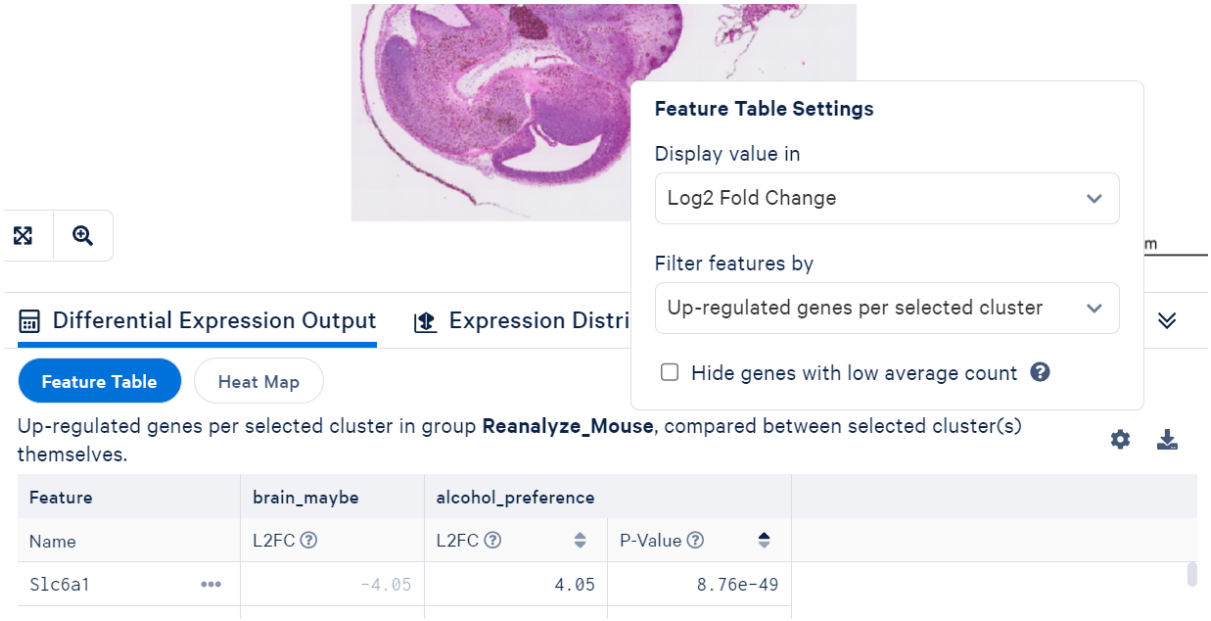
2. Back to the Clusters tab: choose our new clusters brain_maybe and alcohol_preference. Press Run Differential Expression.



3. Choose “Between selected cluster(s) themselves” for Differential Expression Settings.



4. Look at the Differential Expression Output for alcohol preference. The genes can be arranged in L2FC or P-value order, choose the Filter features by “All detected genes in the cluster”. You can also look at the Heatmap.



5. Save the differential gene expression analysis as a table by clicking Export to CSV.

Note that all the analyses above can be done in R or Python – there are many third-party packages and workflows out there. Famous ones include: Seurat, hoodscanR, scider, FlexiDeconv. See this workflow too: 16 Workflow: Visium HD (segmented) – Orchestrating Spatial Transcriptomics Analysis with Bioconductor

R

Let's use the DE gene list exported from Step 3 in Loupe Browser and run Gene Set Enrichment Analysis. Following the tutorial labelled "7: GSEA as an example" in: Hands-on Tour of the Visium HD Spatial Gene Expression Analysis Journey | 10x Genomics

1. Load the Pawsey module of Singularity:
   `module load singularity/4.1.0-nompi`
2. Pull the ready-made Bioconductor Singularity:
   `singularity pull bioc3.22.sif docker://bioconductor/bioconductor_docker:RELEASE_3_22`
3. Download my R script from GitHub: GSEA_analysis_tutorial_03.R
4. Ask me to share the R libraries on Pawsey and untar it.
5. Create a directory named `project` in your working directory and put the Rscript and the Rlibs/ in there. Also copy the DEG result exported from the Loupe Browser analysis in there (you can also download it grom GitHub: brain_vs_alcohol_DEGall.csv).
6. Run the Bioconductor Singularity with the `project` folder and Rlibs bound:
   `singularity shell -B /your/path/to/project:/project -B /your/path/to/project/Rlibs:${HOME}/Rlibs bioc3.22.sif`
7. You're now inside the container and should see: `>Singularity`
8. Export the R library paths:
   `export R_LIBS_USER="~/Rlibs:/usr/local/lib/R/site-library:/usr/local/lib/R/library:/home/${USER}/R/x86_64-pc-linux-gnu-library/4.3"`
9. Run the Rscript:
   `Rscript /project/GSEA_analysis_tutorial_03.R -w /project -d /project/brain_vs_alcohol_DEGall.csv -c alcohol_preference`

The R script is hard-coded to use MSigDBr: Introduction to msigdbr • msigdbr GO:BP for mouse. You can open the R script interactively (via command line or RStudio) and explore ways to customize it.

We'll end up with four plots:

- Volcano plot of the DEG.
- GO:BP enrichment terms vs Normalized Enrichment Score.
- Enrichment score plot with a panel for ranked metric and a panel for running sum enrichment score vs position in gene list for one specific gene set. See GSEA User Guide for detailed interpretation.
- CNET plot showing linkage of one specific gene set to other genes.