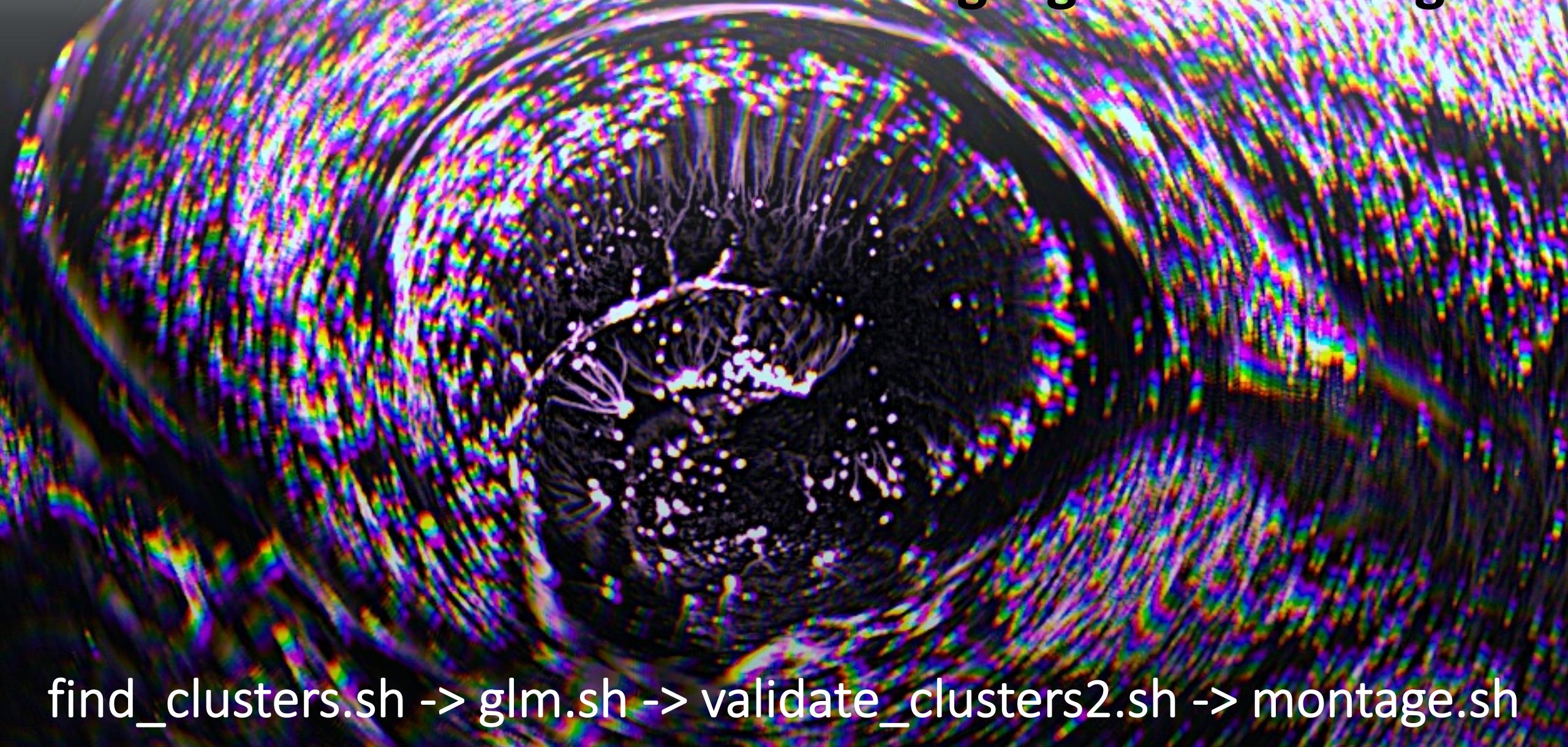


Guide to UNRAVEL: UN-biased high-Resolution Analysis and Validation of Ensembles using Light sheet images



find_clusters.sh -> glm.sh -> validate_clusters2.sh -> montage.sh

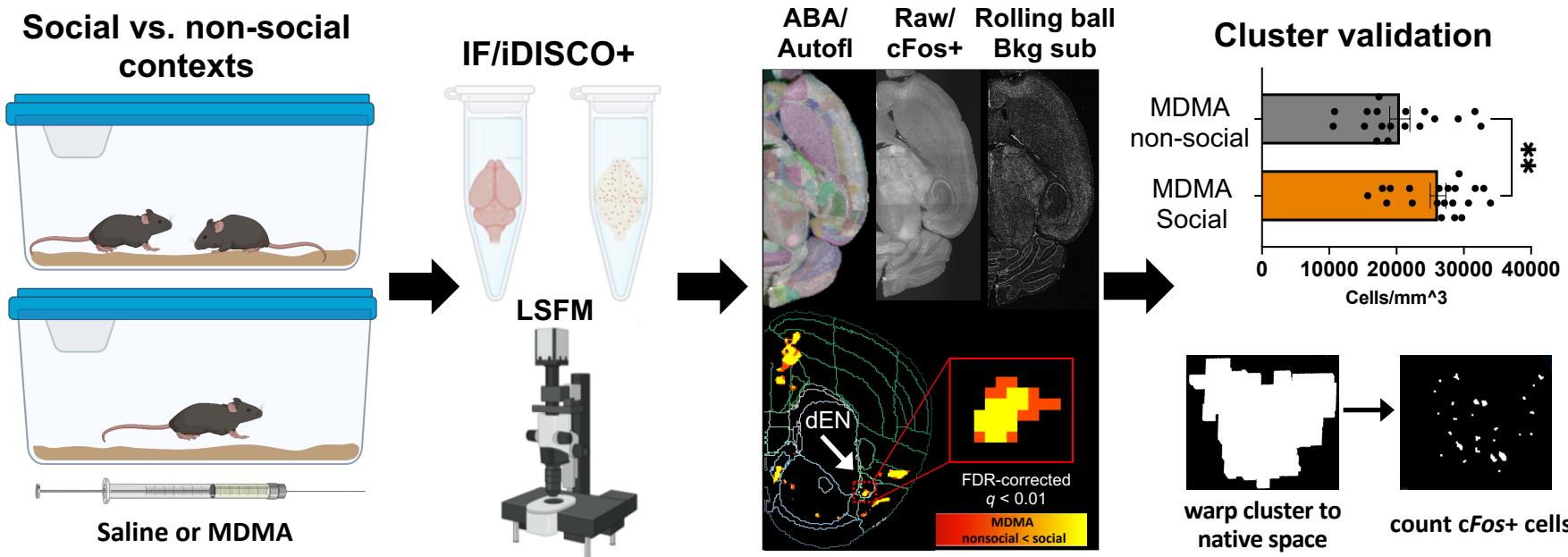
UNRAVEL guide outline

- Pipeline overview figure
- Dependencies
- Software utilities
- Updating scripts for a new install
- Experiment summary folder contents
- Experiment folder contents
- Allen brain atlases (ABA) modified by Gubra, average templates, and masks
- LSFM analysis guide (1 slide)
- `find_clusters.sh`
 - Script functions/outputs
 - Working dir, positional args, subscripts, & FIJI macros
- Voxel-wise stats (`glm.sh` for t-test)
- Voxel-wise stats (`glm.sh` for 2x2 ANOVA)
- `validate_clusters.sh` (quantitative validation)
 - Script functions/outputs
 - Working dir, positional args, subscripts, & FIJI macros
- `montage.sh` (qualitative validation)
- Ideas for future improvements
- Visual summary of steps

Heifets lab



LSFM analysis pipeline overview





github.com/b-heifets/UNRAVEL



Search or jump to...



Pull requests Issues Codespaces Marketplace Explore

b-heifets / UNRAVEL

Private

generated from [github/welcome-to-github](#)

master

1 branch

0 tags

Go to file

Add file

Code

| | | | |
|-----------------------------|---|-----------------------------|-----------|
| nsg-remote-lab | Improved validate_clusters.sh, fixed bugs, added excel templ... | ... f2616bf on Oct 21, 2022 | 4 commits |
| Heifets_lab_FIJI_macros | Improved validate_clusters.sh, fixed bugs, added excel templates, et... | 4 months ago | |
| Heifets_lab_excel_templates | Improved validate_clusters.sh, fixed bugs, added excel templates, et... | 4 months ago | |
| Heifets_lab_scripts | Improved validate_clusters.sh, fixed bugs, added excel templates, et... | 4 months ago | |
| scripts_from_MIRACL | Improved validate_clusters.sh, fixed bugs, added excel templates, et... | 4 months ago | |
| README | Revised LSFM analysis pipeline, adding several new scripts (see guid... | 6 months ago | |

README



#####
Heifets lab brain activity mapping guides:

Guide to immunofluorescence staining, iDISCO+, & lightsheet fluorescence microscopy:

https://docs.google.com/document/d/16yowBhiBQWz8_VX2t9Rf6Xo3Ub4YPYD6qeJP6vJo6P4/edit?usp=sharing

LSFM analysis guide:

https://office365stanford-my.sharepoint.com/:p/g/personal/danrijs_stanford_edu/EbQN54e7SwRHgkmw3yn8fgcBz1xG22AIctZx8nsPr0LFtg?e=S159PM

#####

Dependencies

- Shell scripting
 - GNU BASH 4.4.20, ImageMagick 7.1.0
- MIRACL
 - <https://miracl.readthedocs.io/en/latest/>
- ANTS & c3d
- Fiji
 - CLIJ plugin (custom edits needed for fractional 3D counting of cells at region boundaries)
 - .ijm files from <https://github.com/b-heifets> added to macros folder
- FSL (including FSLEYES)
 - <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FsIInstallation>
- ITK-snap
 - <http://www.itksnap.org/pmwiki/pmwiki.php>
- Ilastik
 - <https://www.ilastik.org/index.html>
- 3D Slicer
 - <https://www.slicer.org/>

Software Utilities

- MIRACL
 - Registration of image volumes to a standard atlas
 - Warping atlas to tissue space (registration) and tissue to atlas space (reverse warp)
 - Image downsampling
- ANTS & c3d
 - Programs called upon by MIRACL
- Fiji
 - File conversion, image thresholding, background subtraction, 3D cell counting, image cropping, visualization
 - CLIJ plugin needed for fractional 3D counting of cells at region boundaries
- FSL (including FSLEYES)
 - Voxel-wise mathematical operations and statistics for 3D image volumes
 - Generalized linear modeling for group comparisons
 - Visualization
- ITK-snap
 - Visualization tool used to inspect registration accuracy by overlaying 2x downsampled image volume with atlas in native space
- 3D Slicer
 - Improving reigstration: manually cropping excess tissue, adding missing tissue, or adding intensity to low intensity pixels
- Illastik
 - Unbiased cell segmentation

Updating scripts for a new install

- `cd <./folder_w_sh_scripts>`
- `sed -i -E "s#/usr/local/miracl/depends/Fiji.app/ImageJ-linux64#<new_FIJI_path/...>#g" *.sh`
- `sed -i -E "s#/usr/local/miracl/depends/Fiji.app/jars/ij-1.53c.jar#<new_FIJI_path/jars/...>#g" *.sh`
- `sed -i -E "s#/usr/local/miracl/depends/Fiji.app#<path/Fiji.app>#g" *.sh`
- #Similarly, update paths to gubra atlas files
- `sed -i -E "s#/usr/local/miracl/atlasses/ara/gubra/#<new_path>#g" *.sh`
- Add folder(s) with scripts to the system path
(<https://askubuntu.com/questions/97897/add-bash-script-folder-to-path>)

Experiment summary folder contents

- **sample_overview.csv**
 - Columns detail: Sample #, hemisphere side (l, r; or w for whole), Marker (cFos), Experimental condition, orientation code (i.e.,Ort_code; e.g. PLS, ALS, PRI, etc), 488min (scaling parameters for decreases autofluorescent noise), xy_res, z_res, olf_bulb (y/n), Exp_Dir (sample folder path)
 - Used to automate cluster validation (validate_clusters.sh)
- **rolling_ball_radius** (text file w/ rb radius in pixels used for subtracting background from raw ochann data via rb.sh. If rb radius is 0 in find_clusters.sh, then raw data is not background subtracted [i.e., ochann_to_nii.sh and ochann_to_gubra.sh are run instead of rb.sh])
- **./ochann_rb*_z_gubra_space** (z-scored, rolling ball bkg sub data in atlas space used for voxel-wise stats; or [ochann_z_gubra_space])
- **Voxel-wise stats folder(s)** (same name as GLM the voxel-wise stats are derived from; e.g., the glm_ttest_<EXP>_rb*_z or glm_ANOVA_<EXP>_rb*_z)
 - <group_name>_sample??_ochann_rb4_z_gubra_space.nii.gz
 - stats
 - vox_p images are uncorrected p-value maps
 - t-tests: group1>group2 for tstat1 and group1<group2 for tstat2
 - 2x2 ANOVA: tstat1 is 1st main effect, fstat2 is 2nd main effect, fstat3 is interactions (these are directional to the contrast set up; fstat is the nondirectional f statistic used for cluster validation)
- **Ilastik projects** (e.g., if consensus approach is used, then: <EXP>_rater1.ilp, <EXP>_rater2.ilp, <EXP>_rater3.ilp, <EXP>_rater4.ilp, <EXP>_rater5.ilp)
- **ilastik_training_slices** (3 slices from 3 samples per condition in a folder)
- **Cluster_validation_summary**
 - <glm_name>_<stats_map>_<stats_thresh>_[min_cluster_size]
 - inputs (text file to rerun validate_clusters.sh) Output folders: cluster_index, native_cluster_index, cluster_masks, bounding_boxes, clusters_cropped, cluster_volumes, ABACluster_masks, ABACluster_cropped, ABAConsensus_cropped, native_stats, stats_cropped, ochann_cropped, ochann_rb*_cropped
- **Text files for rerunning find_clusters.sh & validate_clusters.sh**

Experiment folder(s) contents

- **Directory containing contains sample folders for a given experiment.**
- **E.g.**
 - **sample01**
 - [.czi raw data from Zeiss LS7]
 - [488_original/tifs] (if 488 min display range adjusted [i.e., non-zero])
 - 488/tifs (autofluorescence channel; original if 488 min is 0)
 - ochann/tifs (ochann stands for other raw data channel)
 - **parameters** (488_min, metadata, ort2std.sh)
 - **niftis** (sample01_02x_down_autofl_chan.nii.gz and sample01_02x_down_ochann_rb*_chan.nii.gz or sample01_02x_down_ochann Chan.nii.gz)
 - **reg_final** (main registration outputs: clar_downsample_res10um.nii.gz and gubra_ano_split_10um_clar_downsample.nii.gz)
 - **clar_allen_reg** (transform and intermediate files from reg.sh and to_native.sh)
 - **ochann_rb*/tifs**
 - **seg_ilastik_1/ilastikSegmentation/tifs** (output from rater1)
 - **seg_ilastik_2/ilastikSegmentation/tifs**
 - **seg_ilastik_3/ilastikSegmentation/tifs**
 - **seg_ilastik_4/ilastikSegmentation/tifs**
 - **seg_ilastik_5/ilastikSegmentation/tifs**
 - **consensus** (sample01_consensus.nii.gz & sample01_[ABA]consensus.nii.gz)
 - **sample01_ochann_rb*_gubra_space.nii.gz**
 - **sample01_ochann_rb*_z_gubra_space.nii.gz** (input for glm.sh)
 - **sample02**
 - **sample?? ... n**

Allen brain atlases (ABA) modified by Gubra, average templates, and masks

- ABA template was made using histological sectioning and 2-photon microscopy, which leads to suboptimal registration for brains acquired in 3D by LSFM owing to differences in contrast and morphology. Therefore, we use a template generated by Gubra (Perens et al 2020)
- Templates
 - **gubra_template_10um.nii.gz** or **gubra_template_wo_OB_10um.nii.gz** (used for registration w/ the tissue)
- Atlas
 - **gubra_ano_split_10um.nii.gz** (warped to tissue during registration; intensities on left side are increased by 20000 w/ “split” atlases)
 - **gubra_ano_combined_25um.nii.gz** (used when viewing data in atlas space)
- Tissue mask
 - **gubra_template_25um_thr30_bin.nii.gz** (mask for whole brains)
 - **gubra_template_wo_OB_25um_full_bin_left.nii.gz** (mask for left hemisphere or when left/right hemispheres are pooled for whole brains)
 - **gubra_template_wo_OB_25um_full_bin_right.nii.gz** (RH mask)

Finding clusters (script functions/outputs)

- **find_clusters.sh**
 - **overview.sh** (makes ./sample??/parameters/metadata, ./exp_dir/parameters.csv, ./exp_summary/sample_overview.sh, or **czi_to_tif.sh** (../sample??/*.czi → makes xy dim even → ./sample??/ochann/tifs → adjusts 488 min → ./sample??/488/tifs)
 - **prep_tifs.sh** (makes xy dim even → adjusts 488 min → ./sample??/488/tifs)
 - **488_to_nii.sh** (../sample??/488_tifs → ./sample??/niftis/sample??_02x_down_autofl_chan.nii.gz for **reg.sh**)
 - **reg.sh** (makes ./reg_final/clar_downsample_res10um.nii.gz & ./reg_final/gubra_ano_split_10um_clar_downsample.nii.gz)
 - **rb.sh** (rolling ball subtracts background from ochann, making ./ochann_rb*/tifs, niftis/sample??_02x_down_rb*_chan.nii.gz, & ./sample??/sample??_ochann_rb*_gubra_space.nii.gz)
 - If processing raw data:
 - ochann_to_nii.sh (makes/niftis/sample??_02x_down_ochann_chan.nii.gz)
 - ochann_to_gubra.sh (makes ./sample??/sample??_ochann_gubra_space.nii.gz)
 - **z_brain_template_mask.sh** (z-scores * _gubra_space.nii.gz files)
 - **fsleyes.sh** (view * _gubra_space.nii.gz and the gubra_ano_combined_25um.nii.gz atlas)
- **glm.sh** (outputs voxel-wise t-test or 2x2 ANOVA 1-p value maps to ./<glm_folder>/stats)
- Optional scripts:
 - **mv_samples.sh** (move sample?? folders and auto update ./exp_dir/parameters.csv and ./exp_summary/overview.sh)
 - **mirror.sh** (flips samples in atlas space)

Finding clusters (working dir, positional args, & subscripts)

Working directory:

exp_summary

- **find_clusters.sh** (user input: rolling ball radius)
 - **overview.sh** <exp_dir(s)> (user inputs: side, marker, condition, ort_code, 488min, xy_res, z_res, olf_bulb, exp_dir) (metadata.ijm)
 - **czi_to_tif.sh and prep_tifs.sh** <0 or 488 dr min> [*] (metadata.ijm, czi_to_tif.ijm/prep_tifs)
 - **488_to_nii.sh** <x/y voxel size in microns or metadata> <z voxel res in microns or n> [*] (metadata.ijm & miracl_conv_convertTIFFtoNII.py)
 - **reg.sh** <orientation code> <0/1 for absence/presence of olfactory bulb> <w for wholebrains or l or r for left/right hemisphere> [*] (miracl_reg_clar-allen_whole_brain_iDISCO.sh)
 - **rb.sh** <orientation code> <rolling ball radius in pixels> <x/y voxel res in microns or m to use metadata> <z voxel res in microns or n> [*] (metadata.ijm , rolling_ball_bkg_subtraction.ijm, miracl_conv_convertTIFFtoNII.py, miracl_reg_warp_clar_data_to_gubra.sh)
 - Or for raw data:
 - ochann_to_nii.sh <x/y voxel res in microns or metadata> <z voxel res in microns or n> [*] (metadata.ijm & miracl_conv_convertTIFFtoNII.py)
 - ochann_to_gubra.sh <three letter orientation code> [*]

sample?? | • **z_brain_template_mask.sh** <side of the brain (l, r, or both)> [default: *_gubra_space.nii.gz or list of images separated by spaces]
exp_sum | • **fsleyes.sh** <display range min> <display range max> [default: *.nii.gz or list of specific images]

glm_dir | • **glm.sh** [binary mask in 25 um Gubra atlas space]

- User inputs: side of brain, extra randomise options, # of permutations, kernel radius for smoothing in mm

exp_sum | • Optional scripts:

- **mv_samples.sh** <destination path> <list of path/exp_folders OR path/sample?? >
- **mirror.sh** (or flip.sh and/or nudge.sh)

*default: process all samples (if blank) OR specific samples (list path/sample??)

Voxel-wise stats

- Two-sample unpaired t-test with permutation testing
 - **glm.sh**
 - Time to process on lab computer ~ 1 day
 - 1) make glm folder named succinctly (e.g., `glm_ttest_<EX>_rb<4>z_<contrast>`)
 - 2) Add `*_sample??_gubra_space.nii.gz` files and rename with `<condition1/2>` prefix for each file
 - 3) open inputs in fsleyes & the 25 um atlas to check alignment and that sides are correct
 - `fsleyes.sh <display range min> <display range max> [leave blank to process all .nii.gz files or enter specific files separated by spaces]`
 - 4) run `glm.sh` from glm folder and follow prompts (outputs to `./stats`)
- `tstat1` = group 1 intensity > group 2 intensity (group order is alphabetical [use `ls` to check order])
- `tstat2` = group 1 intensity < group 2 intensity
- `vox_p` images are intensity uncorrected p value maps
- Help
 - `glm.sh help`
 - <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/GLM>
 - <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/Randomise/UserGuide>

Voxel-wise stats

- Two-factor, two level ANOVA with permutation testing
 - **glm.sh**
- Make design matrix, contrast matrix, and F-contrast files in `./glm_ANOVA_<EX>_rb*_z/stats/`
 - Open terminal from `./stats` and run: `fsl`
 - GUI
 - Misc → GLM Setup
 - Higher-level / non-timeseries design → # inputs = total # of samples
 - EVs tab in GLM window: → # of main EVs: 4 → Name EVs (e.g., EV1 = group 1) →
- Make design matrix:
 - Under each EV, enter 1 for group assignment, leave 0 if sample is not part of group
- Contrasts & F-tests tab in GLM window: → Contrasts: 3
 - C1: Main_effect_<e.g., drug> 1 1 -1 -1 (e.g., if EV1/2 are drug groups and EV3/4 are saline groups)
 - C2: Main_effect_<e.g., context> 1 -1 1 -1 (e.g., if EV1/3 were in context1 and EV2/4 were in context2)
 - C3: Interaction 1 -1 -1 1
 - Check boxes for each contrast an f-statistic map is desired for
- GLM Setup window:
 - Save -> click design -> OK
- run: **glm.sh** from ANOVA folder
- Outputs for cell density validation in clusters (1-p value maps):
 - `vox_p_fstat1` = Main effect 1
 - `vox_p_fstat2` = Main effect 2
 - `vox_p_fstat3` = interaction

Cluster validation

- `cluster_validation2.sh` has been optimized for fast measurement of cell densities in each cluster
 - `ABA_volumes.sh` measures regional volumes in atlas space
 - `sunburst.sh` preps regional volume data for Flourish sunburst plots
- `cluster_validation.sh` is an older version that can be used for region-specific cell density measurements in each cluster
 - Regional volumes determined in native space

validate_clusters2.sh (script functions/outputs)

- **validate_clusters2.sh** (check if hot/cold spots identified by voxel-wise stats are valid based on cell density measurements)
 - **ilastik.sh** (after training ilastik (5 raters/projects; see ilastik.sh help), segment cells in ./ochann/tifs & outputs seg_ilastik_?/IlastikSegmentation/tifs, where ? is rater #)
 - **consensus.sh** (If a pixel was classified as a cell by at least 3/5 raters, then preserve it as a cell; outputs: ./sample??/consensus/sample??_consensus.nii.gz)
 - **For voxel-wise correction**
 - **fdr.sh** (makes "clusters" folder for all outputs: <stats_map>_FDR<q_value>_MinCluster<#_of_voxels> in ./stats/, ./sample??/clusters/, & ./cluster_validation_summary/)
 - Determines corrected 1-p value threshold for uncorrected data
 - Makes cluster index with clusters of sig voxels above arbitrary min cluster size (e.g., 100 voxels) and reverses order of cluster IDs (so largest is cluster 1 and so on)
 - Makes adjusted p-value map and zeros out non-sig voxels (<pvalimage>_thresh.nii.gz for montage)
 - **For cluster-wise correction**
 - **ez_thr.sh** (makes "clusters_folder" for all outputs: <stats_map>_ezThr<z_threshold> in ./stats/, ./sample??/clusters/, & ./cluster_validation_summary/)
 - Makes cluster index using only significant clusters surviving and reverses order of cluster IDs
- **ABA_volumes.sh** outputs <input_image>_region_volumes.csv (region volumes in atlas space in cubic mm)
- **sunburst.sh** outputs <path/[validated]_cluster_index>_sunburst.csv and sunburst_RGBs.csv
- **to_native2.sh** (warps and/or scales rev_cluster_index, <pvalimage>_thresh.nii.gz, and warped atlas to full res native space, outputting ./clusters/clusters_folder/output_folder/native_<image>.nii.gz)
- **native_clusters.sh** outputs bounding_boxes/outer_bounds.txt (bbox of cluster index), bounding_boxes/bounding_box_cluster_*.txt (bboxes for each cluster), clusters_cropped/crop_sample??_native_cluster_*.nii.gz (binary cropped cluster)
- **3d_count_cluster2.sh** (3D count cells in cluster on GPU, outputting counts with corresponding region intensities to sample_??_cluster_*_[ABA]consensus_3Dcounts.csv and total count crop_[ABA]consensus"_"sample??_native_cluster_*_3D_cell_count.txt)
- **cluster_densities2.sh** (outputs ./cluster_validation/cluster_outputs/densities_<"cluster_folder">.csv, which has cell densities [cells/mm³] for each sample and cluster)
- **crop_cluster.sh** (makes cluster-specific montage tiles for thresholded p value map, ochann, and ochann_rb*)
- **get_most_sig_slice.sh** (find slice in thresholded stats map with highest integrated density and saves slice # in .csv in ./stats_cropped/)
- **extract_most_sig_slice.sh** (saves most sig slice from cropped: ochann, ochann_rb*, consensus, stats_thr in folders w/ cropped image volumes)
- **rsync_clusters.sh** (copies all validate_clusters.sh data to cluster_validation_summary/"cluster_folder")
- **montage.sh** (makes montage (raw, rb, consensus, stats_thr tiles) for all defined samples and clusters, output to cluster_validation_summary folder)

Segmentation

Quantitative validation

Qualitative validation

validate_clusters2.sh (working dir, positional args, & subscripts)

Working directory:

- exp_sum
 - **validate_clusters2.sh** <path_array> <segment_or_validate> <path/stats_map> <q_value (for fdr.sh)> <ez_thr> <mask> <min_cluster_size> <xy_res> <z_res> <clusters_to_process> <regional_volumes?> <regional_counts?> <make_montage?> <raw_folders>
- exp_dir
 - **ilastik.sh** <path/<EXP>_rater1.ilp> <'{1..5}' (range for raters) or '1 2 4' (for specific rater(s))> [default: process all samples OR list sample?? separated by spaces to process specific samples]
 - **consensus.sh** [default: all samples OR list sample??] (tif_to_nii.ijm, consensus_part1.ijm, consensus_part2.ijm, consensus_part3.ijm, consensus_part4.ijm)
- ./glm/stats
 - **fdr.sh** <path/stat_image.nii.gz> <q_value (e.g., 0.05)> <min cluster size in voxels> <side of the brain (l, r, or both) or enter path/custom_mask>
 - **ez_thr.sh** <path/pvalimage> <side of the brain (l, r, or both)> <z_thresh (*e.g., 3.290527 for 2-tail p<0.001)> <cluster_prob_thresh (e.g., 0.05)>
 - **ABA_volumes.sh** <stats/output_folder/rev_cluster_index.nii.gz> (**ABA_volumes.py**)
 - **sunburst.sh** <stats/output_folder/rev_cluster_index.nii.gz> (**sunburst.py**)
 - **to_native2.sh** <path/image.nii.gz to warp/scale> <xy voxel size (um) or m (metadata)> <z voxel size or m> <clusters/ "\${stats_map::-7}"_FDR"\${q_value}"_MinCluster"\${min_cluster_size}" or clusters/"\${stats_map::-7}"_ezThr"\${ez_thr}"> <output_folder: native_cluster_index, native_atlas, or native_stats> (**to_native.ijm**)
 - **native_clusters.sh** <./sample??/clusters/\$output_folder/native_cluster_index/native_"\$output_folder"_rev_cluster_index.nii.gz> <xy_res> <z_res> <clusters> (**native_clusters.py**)
 - **3d_count_cluster2.sh** <./clusters_folder/cluster_masks/sample??_native_cluster_X.nii.gz> <cluster #> <enter y for region specific counts in clusters or n for just counts> (**ClusterConsensus.ijm**, **3D_count_IncludeEdges.ijm**, **100sliceSubstacks_ClusterConsensus.ijm**, **3D_count_ExcludeEdges.ijm**, **3D_count_CPU.ijm**)
 - **cluster_densities2.sh** [string of all sample numbers (e.g., from sample_overview.csv)] [all conditions (e.g., from sample_overview.csv)]
 - **get_most_sig_slice.sh** <./path/crop_stats_thr_sample??_native_cluster_*.nii.gz> (**integrated_densities_for_stack.ijm**)
 - **extract_most_sig_slice.sh** <./path/image_to_extract_slice_from.nii.gz> <./path/crop_stats_thr_sample??_native_cluster_*.nii.gz> **IntDen-Max_most-sig-slice.csv** (**extract_most_sig_slice.ijm**)
- exp_sum
 - **rsync_clusters.sh** <\${path_array[@]}> \$path_and_stats_map \$q_value \$ez_thr \$min_cluster_size>
- clust_folder
 - **montage.sh** (**dr_adjustment.ijm**)

validate_clusters.sh (script functions/outputs)

- **validate_clusters.sh** (check if hot/cold spots identified by voxel-wise stats are valid based on cell density measurements)
 - **ilastik.sh** (after training ilastik (5 raters/projects; see ilastik.sh help), segment cells in ./ochann/tifs & outputs seg_ilastik_?/IlstikSegmentation/tifs, where ? is rater #)
 - **consensus.sh** (If a pixel was classified as a cell by at least 3/5 raters, then preserve it as a cell; outputs: ./sample??/consensus/sample??_consensus.nii.gz)
 - **For voxel-wise correction**
 - **fdr.sh** (makes “clusters” folder for all outputs: <stats_map>_FDR<q_value>_MinCluster<#_of_voxels> in ./stats/, ./sample??/clusters/, & ./cluster_validation_summary/)
 - Determines corrected 1-p value threshold for uncorrected data
 - Makes cluster index with clusters of sig voxels above arbitrary min cluster size (e.g., 100 voxels) and reverses order of cluster IDs (so largest is cluster 1 and so on)
 - Makes adjusted p-value map and zeros out non-sig voxels (<pvalimage>_thresh.nii.gz for montage)
 - **For cluster-wise correction**
 - **ez_thr.sh** (makes “clusters_folder” for all outputs: <stats_map>_ezThr<z_threshold> in ./stats/, ./sample??/clusters/, & ./cluster_validation_summary/)
 - Makes cluster index using with only significant clusters surviving and reverses order of cluster IDs
 - **to_native.sh** (warps and/or scales rev_cluster_index, <pvalimage>_thresh.nii.gz, and warped atlas to full res native space, outputting to ./clusters/(native_cluster_index, native_stats, or native_atlas folders)
 - **cp_prior_clusters.sh** (if using fdr.sh and changing the min cluster size, copy relevant prior data to avoid redundant processing)
 - **cluster_masks.sh** (creates binary masks for each cluster from rev_cluster_index and outputs to ./cluster_masks/)
 - **bounding_boxes.sh** (outputs bounding box location and size to ./bounding_boxes/<pvalimage>_fslstats_w.txt)
 - **crop_cluster.sh** (cropped, full res cluster masks in native space input will output to ./clusters_cropped/)
 - **ABAconsensus.sh** (Multiplies full res native atlas and sample??_consensus.nii.gz to convert intensities of cells into regional intensities for 3D counting → cropped w/ crop_cluster.sh and output to ./ABAconsensus_cropped/ for 3D counts & montage. [**consensus.sh** can be used instead if just getting total counts w/ no regional info])
 - **3d_count_cluster.sh** (3D count cells in cluster on GPU, outputting counts with corresponding region intensities to sample_??_cluster_*_[ABA]consensus_3Dcounts.csv and total count crop_[ABA]consensus"_sample??_native_cluster_*_3D_cell_count.txt)
 - **cluster_densities.sh** (outputs ./cluster_validation/cluster_outputs/densities_<“cluster_folder”>.csv, which has cell densities [cells/mm³] for each sample and cluster)
 - **ABAcluster.sh** (Multiplies full res native atlas and cluster mask to convert intensities of clusters into regional intensities for regional volumes in clusters → cropped w/ crop_cluster.sh and output to ./ABAcluster_cropped/)
 - **ABAcluster_volumes.sh** (Output : ./cluster_outputs/cluster_*_ABA_histogram.csv (16-bit histogram reporting regional volumes (voxel counts for each intensity))
 - **ABAcluster_counts.sh** (Output: ./cluster_validation_summary/cluster_outputs/counts/cluster_*_ABA_counts_<glm_name>.csv gives cell counts in each ABA region for a cluster)
 - **effect_size_vox.sh** (group1 cluster means – group2 cluster means)/pooled SD → ./cluster_outputs/cluster_*_effect_size.csv
 - **crop_cluster.sh** (makes cluster-specific montage tiles for thresholded p value map, ochann, and ochann_rb*)
 - **get_most_sig_slice.sh** (find slice in thresholded stats map with highest integrated density and saves slice # in .csv in ./stats_cropped/)
 - **extract_most_sig_slice.sh** (saves most sig slice from cropped: ochann, ochann_rb*, consensus, stats_thr in folders w/ cropped image volumes)
 - **rsync_clusters.sh** (copies all validate_clusters.sh data to cluster_validation_summary/“cluster_folder”)
- **montage.sh** (makes montage (raw, rb, consensus, stats_thr tiles) for all defined samples and clusters, output to cluster_validation_summary folder)

Segmentation

Quantitative validation

Qualitative validation

validate_clusters.sh (script functions/outputs)

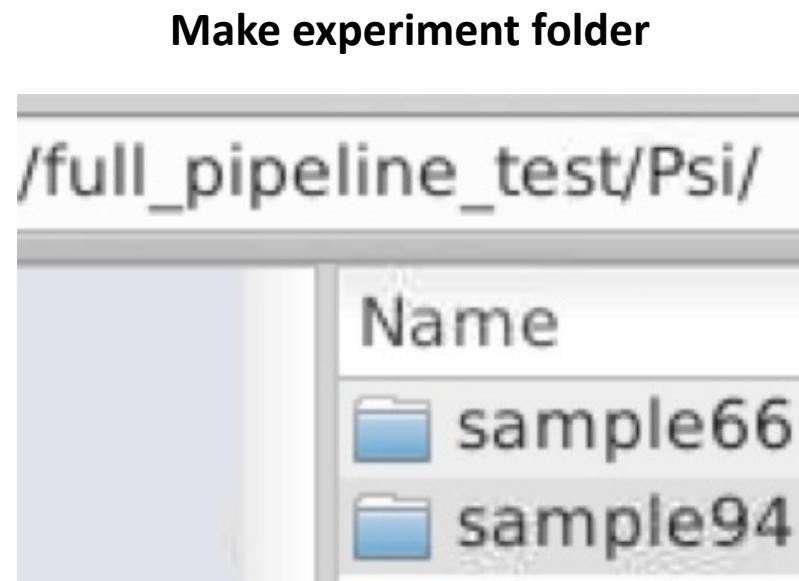
Working directory:

| | |
|--------------|---|
| exp_sum | • validate_clusters.sh (user inputs: path_array, segmentation_or_valalidation, path/stats_map , q_value (for fdr.sh), ez_thr, mask, min_cluster_size, xy_res, z_res, clusters_to_process, regional_data?, montage?, raw_folders) |
| exp_dir | <ul style="list-style-type: none">ilastik.sh <path/<EXP>_rater1.ilp> <'{1..5}' (range for raters) or '1 2 4' (for specific rater(s))> [default: process all samples OR list sample?? separated by spaces to process specific samples]consensus.sh [default: all samples OR list sample??] (tif_to_nii.ijm, consensus_part1.ijm, consensus_part2.ijm, consensus_part3.ijm, consensus_part4.ijm) |
| ./glm/stats | <ul style="list-style-type: none">fdr.sh <path/stat_image.nii.gz> <q_value (e.g., 0.05)> <min cluster size in voxels> <side of the brain (l, r, or both) or enter path/custom_mask>ez_thr.sh <path/pvalimage > <side of the brain (l, r, or both)> <z_thresh (*e.g., 3.290527 for 2-tail p<0.001)> <cluster_prob_thresh (e.g., 0.05)>to_native.sh <path/image.nii.gz to warp/scale> <xy voxel size (um) or m (metadata)> <z voxel size or m> <clusters/ "\${stats_map::7}"_FDR"\${q_value}"_MinCluster"\${min_cluster_size}" or clusters/"\${stats_map::7}"_ezThr"\${ez_thr}"> <output_folder: native_cluster_index, native_atlas, or native_stats> (to_native.ijm)cp_prior_clusters.sh <prefix_for_matching_clusters_folders> <cluster_index_path>cluster_masks.sh <./clusters_folder/native_cluster_index/sample??_native_cluster_index.nii.gz> <./clusters_folder/cluster_masks/sample??_image_cluster_x.nii.gz> <cluster #>bounding_boxes.sh <./clusters_folder/cluster_masks/sample??_image_cluster_*.nii.gz>crop_cluster.sh <sample??/clusters/\$output_folder/bounding_boxes/sample??_native_cluster_*_fslstats_w.txt> <image type: clusters, ABAconsensus, consensus, stats, ochann, or ochann_rb*> <path/image_to_crop.nii.gz or path/first_tif_in_series_to_crop> (crop_cluster.ijm or crop_cluster_image_sequence.ijm)ABAconsensus.sh (convert_to_ABA_intensities.ijm)3d_count_cluster.sh <./clusters_folder/cluster_masks/sample??_native_cluster_X.nii.gz> <cluster #> <enter y for region specific counts in clusters or n for just counts> (ClusterConsensus.ijm, 3D_count_IncludeEdges.ijm, 100sliceSubstacks_ClusterConsensus.ijm, 3D_count_ExcludeEdges.ijm, 3D_count_CPU.ijm)cluster_densities.sh [string of all sample numbers (e.g., from sample_overview.csv)] [all conditions (e.g., from sample_overview.csv)]ABAcluster.sh <./clusters_folder/cluster_masks/sample??_native_cluster_*.nii.gz> (convert_to_ABA_intensities.ijm)ABAcluster_volumes.sh [string of all sample numbers (e.g., from sample_overview.csv) or leave blank to process all samples]effect_size_vox.sh <glm_folder_path> <./stats/clusters_folder/*_rev_cluster_index.nii.gz> <exp_summary_dir/cluster_validation_summary/clusters_folder>get_most_sig_slice.sh <./path/crop_stats_thr_sample??_native_cluster_*.nii.gz> (integrated_densities_for_stack.ijm)extract_most_sig_slice.sh <./path/image_to_extract_slice_from.nii.gz> <./path/crop_stats_thr_sample??_native_cluster_*.nii.gz> IntDen-Max_most-sig-slice.csv> (extract_most_sig_slice.ijm)rsync_clusters.sh <\${path_array[@]}> \$path_and_stats_map \$q_value \$ez_thr \$min_cluster_size> |
| exp_sum | montage.sh (dr_adjustment.ijm) |
| clust_folder | |

Visual summary of LSFM analysis (following slides)

- Prepare and image mouse brains
 - https://docs.google.com/document/d/16yowBhiBQWz8_VX2t9Rf6Xo3Ub4YPYD6qeJP6vJo6P4/edit?usp=sharing
 - Set up folders and data:
 - Make experiment folder, cd to it, and run for i in {01..99}; do mkdir -p sample\$i sample\$i/488 sample\$i/ochann ; done #change 99 to your N
 - There can be multiple experiment folders for parallel processing. Parallel processing should occur on separate RealVNC virtual desktops to avoid FIJI macro crosstalk.
 - Move .czi files to sample folders or tif series to 488 and ochann folders
 - Open .czi or ./488/tifs in FIJI, find min display range value to zero out most external voxels, and note this for each sample (see czi_to_tif.sh or prep_tifs.sh)
 - Make experiment summary folder and cd to it
 - Run: **. activate miracl**
 - The miracl virtual environment must be activated before running scripts
 - Run: **find_clusters.sh**
 - makes sample_overview.csv
 - Preps data for voxel-wise stats
 - Run **mv_samples.sh**
 - Relocates sample?? folders
 - Check atlas → sample registration and sample → atlas space reverse warping
 - Atlas space → native space registration:
 - Use itksnap (for info, run: reg.sh help)
 - Native space → atlas space reverse warping
 - Use fsleyes.sh (for ./<EXP>_summary/ochann_rb*_z_gubra_space_z/sample??_ochann_rb*_z_gubra_space.nii.gz).
 - To improve registration, open ./sample??/niftis/sample??_02x_down_autofl_chan.nii.gz in 3DSlicer and either trim extra tissue ,add missing tissue, or 3D paint accordingly
 - Before rerunning registration with using rerun reg.sh and the modified sample??_02x_down_autofl_chan.nii.gz, delete or rename prior reg_final and clar_allen_reg
- Make directory for voxel-wise stats (e.g., glm_ttest_<EXP>_rb*_z) in the experiment summary folder
 - Add all well registered sample??_ochann_rb*_z_gubra_space.nii.gz images & use mirror.sh as needed to overlay hemispheres (e.g., to pool hemispheres of wholebrains)
 - Add prefixes (e.g, <group1/2>_sample??_ochann_rb4_z_gubra_space.nii.gz). Group order is alphabetical.
- Run: **glm.sh**
 - Performs voxel-wise stats (t-test or 2x2 ANOVA)
- Run: **validate_clusters.sh** or **validate_clusters2.sh**
 - 1) segment cells w/ Ilastik (can start when ./sample/ochann/tifs exists) & refine by consensus (see ilastik.sh and consensus.sh)
 - 2) get cell densities etc., in hot/cold spots (i.e., clusters)
- For more info on each step run: <script_name>.sh help (script set up instructions are in **find_clusters.sh**)
- Direct ?s/suggestions to Daniel Ryskamp Rijsketic , Austen Casey, and/or Boris Heifets

Make experiment folder with sample folders



```
#Open terminal  
cd experiment folder  
for i in {01..99}; do mkdir -p sample$i sample$i/488 sample$i/ochann ; done #change 99 to your N
```

Data organization in experiment folder at start

./488/tifs (autofl)

| Name |
|-----------------------|
| sample66_Ch1_0000.tif |
| sample66_Ch1_0001.tif |
| sample66_Ch1_0002.tif |
| sample66_Ch1_0003.tif |
| sample66_Ch1_0004.tif |

./ochann/tifs (e.g., cFos IF)

| Name |
|-----------------------|
| sample66_Ch2_0000.tif |
| sample66_Ch2_0001.tif |
| sample66_Ch2_0002.tif |
| sample66_Ch2_0003.tif |
| sample66_Ch2_0004.tif |

Or

| Name |
|----------|
| sample66 |
| sample94 |

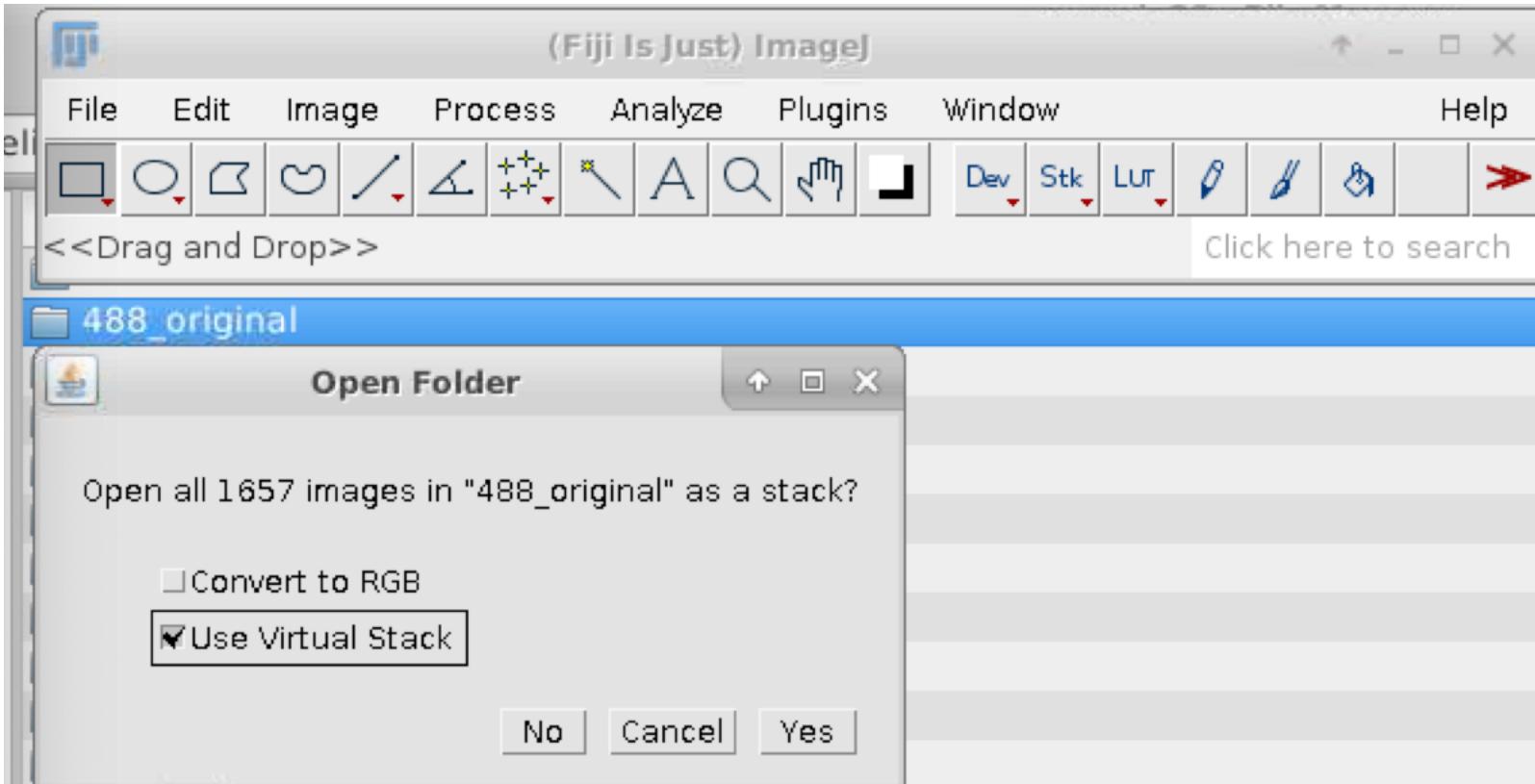
line_test/Psi/sample66/

| Name |
|--|
| Psi_NE_s66_488-8_638_20_50ms_1061thickLS_688x800_Overlap10_glue-Stitching-06.czi |

<EXP_sample??_488-%power_638-%power_ExposureTime_SheetThickness_FOV_%tileOverlap_glueOrAgarose_Stitched>.czi

Naming is flexible

Note display range min for 488 that would zero out most external voxels for each sample

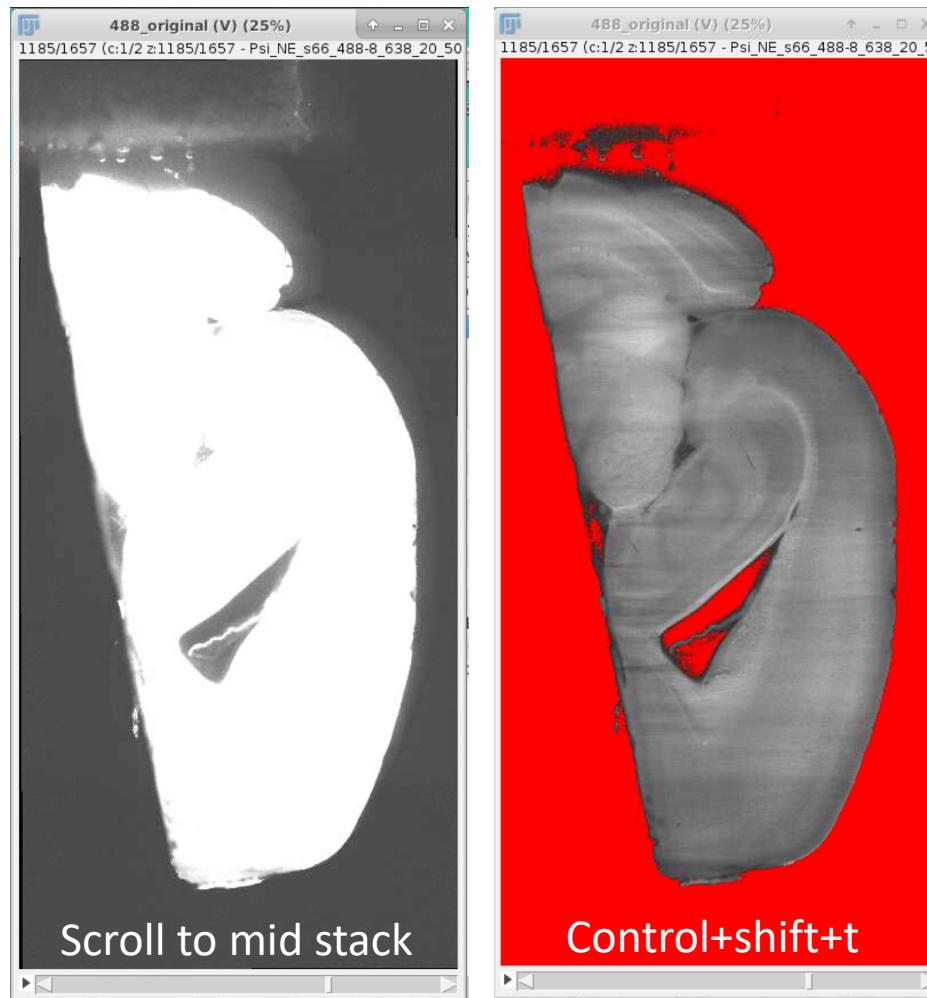


Open FIJI (e.g., assuming alias is set up in .bash_rc, run: Fiji)

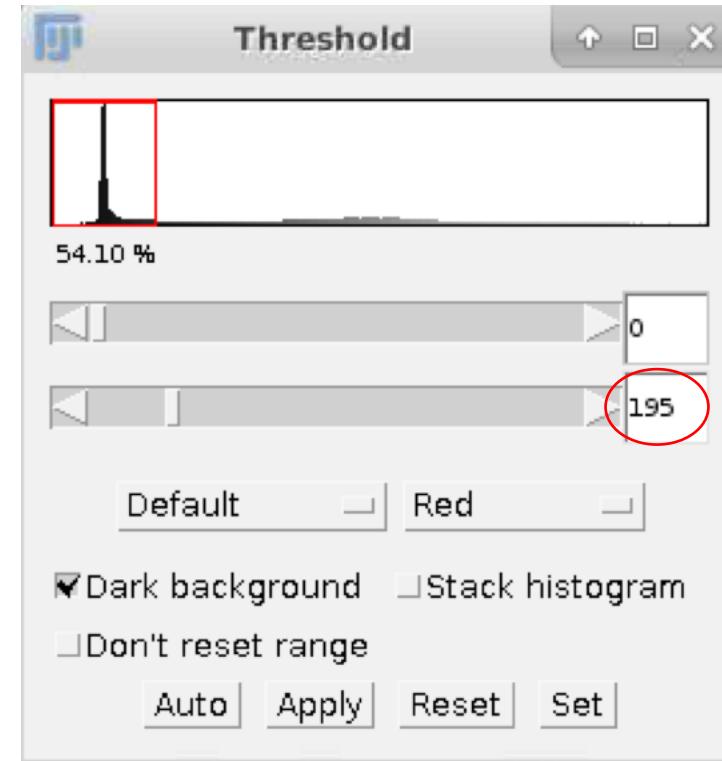
Drag/drop in 488 folder and open as a Virtual Stack (fast and saves RAM)

Note, if opening images not as virtual stack, quit FIJI GUI as soon as you are finished to free up RAM

Note display range min for 488 that would zero out most external voxels for each sample



Adjust display range sliders to find ~ value of external voxels



Note value and repeat for all samples.

If 488 min is similar for all samples, you can use the same 488 min to make processing slightly easier

You can adjust brightness and contrast with control+shift+c

Adjusting 488 min improves registration by zeroing out most voxels outside of the brain (to prevent the template/atlas from being pulled outward if external background is not fully masked during registration)

Make experiment summary folder

- #Open terminal
- cd <experiment folder> #or right click → open terminal from there
- . activate miracl
- find_clusters.sh

find_clusters.sh

```
(miracl) bear@cb-Precision-7920-Tower:/SSD3/full_pipeline_test/Psi_summary$ find_clusters.sh
#####
# Summary #####
This script from Boris Heifets's lab at Stanford automatically preps 3D images of immunostained and cleared mouse brains or hemispheres for voxel-wise analysis in a
tlas space to find clusters of voxels with significantly different intensities between groups (e.g., hot/cold spots in cFos+ cell density from an experimental treat
ment).

For more details and tips, run: find_clusters.sh help
For more info on subscripts, run: <subscript> help
Subscripts: overview.sh, czi_to_tif.sh or prep_tifs.sh, 488_to_nii.sh, reg.sh, rb.sh (or ochann_to_nii.sh), z_brain_template_mask.sh, fsleyes.sh

Additional outputs: ./exp_dirs/parameters.csv (local info w/ no headers) ./exp_summary/sample_overview.csv (global info w/ headers)

#####
# Steps for find_clusters.sh #####
. activate miracl

#Make experiment folder(s) and
cd <./EXP_folder>

#To make sample folders for 99 samples, run:
for i in {01..99}; do mkdir -p sample${i} sample${i}/488 sample${i}/ochann ; done

#If starting with a tif series for 488 and ochann, move it to respective folders. If starting with .czi, move it to sample?? folder

#Determine & note the new min for the 488 display range for all samples. If an approximate value works uniformly, use that.

#Make an experiment summary folder and
cd <EXP_summary>

#Follow prompts in terminal after running:
find_clusters.sh [Enter space seperated list of experiment folder paths (to process all samples) OR sample folder paths (to process specific samples)]
#####

To make list for following input, drag and drop path/exp_dir or path/exp_dir/sample?? folders into the terminal
```

User inputs

Enter space seperated list of experiment folder paths (to process all samples) OR sample folder paths (to process specific samples): '/SSD3/full_pipeline_test/Psi'

Rerun script with:

find_clusters.sh '/SSD3/full_pipeline_test/Psi' starting at Fri Aug 5 16:18:32 PDT 2022

Enter number of pixels for rolling ball radius (~radius of largest object of interest) for background subtraction or 0 for using raw data: 4 rb radius prompt
skipped in future

find_clusters.sh #user inputs gathered once by overview.sh

```
Running overview.sh /SSD3/full_pipeline_test/Psi
```

```
Determine 3 letter orientation codes (A/P=Anterior/Posterior, L/R=Left/Right, S/I=Superior/Interior):
```

```
Open z-stack in FIJI -> 1st letter is side facing up, 2nd is side facing left, 3rd is side at stack start
```

```
Examples:
```

```
Zeiss LS7: ALS in agarose (axial images w/ dorsal z-stack start, dorsal toward LSFM front, & anterior up; in z-stacks A is up, L is left, S is at stack start)
```

```
Zeiss LS7: PLS if glued (axial images w/ dorsal z-stack start, dorsal toward LSFM front & anterior down; in z-stacks P is up, L is left, S is at stack start)
```

```
UltraII: AIL=LH (sagittal images w/ lateral z-stack start, medial side down, & anterior toward LSFM back; in z-stacks A is up, I is left, L is at stack start)
```

```
UltraII: ASR=RH (sagittal images w/ lateral z-stack start, medial side down, & anterior toward LSFM back; in z-stacks A is up, S is left, R is at stack start)
```

```
##### Input parameters for whole experiment #####
```

```
Do all samples in the experiment have the same ...
```

```
antigen? (<antigen> or n): cFos
```

```
brain shape (whole (w) or left (l)/right (r) hemisphere)? (w, l, r, or n): l
```

```
orientation? (<3 letter ORT> or n): n
```

```
488 min display range? (0, <new min>, or n): 200
```

```
xy voxel size? (<xy voxel size in microns> or <m to use metadata from 1 sample for all> or <n to use metadata from each sample>): m
```

```
presence (1) or absence (0) of an olfactory bulb? (or: n): 0
```

```
##### Input parameters for experiment folder #####
```

```
Do all samples in /SSD3/full_pipeline_test/Psi have the same ...
```

```
orientation? (<3 letter ORT> or n): n
```

```
Enter 3 letter orientation code for sample66: PRT
```

```
Enter 3 letter orientation code for sample94: PLS
```

```
Enter condition for sample66: Psilocybin_Homecage
```

```
Enter condition for sample94: Saline_Homecage
```

Note that if not all samples in the experiment have the same property, it will ask regarding samples in an experiment folder and then individual samples

These questions will be skipped if parameters.csv and sample_overview.csv already exist.

find_clusters.sh #auto runs shown scripts (skipping existing files)

```
Running overview.sh /SSD3/full_pipeline_test/Psi
sample_overview.csv exists in /SSD3/full_pipeline_test/Psi_summary, skipping.
For small changes, edit it and corresponding exp_dir/parameters.csv manually before rerunning find_clusters.sh
To remake it and parameters.csv, delete /SSD3/full_pipeline_test/Psi_summary/sample_overview.csv and:
/SSD3/full_pipeline_test/Psi/parameters.csv
Then, rerun find_clusters.sh or overview.sh

Running czi_to_tif.sh 200 sample66 from /SSD3/full_pipeline_test/Psi
488 and ochann tifs for sample66 exist, skipping

Running 488_to_nii.sh 3.5292 3.5 sample66 from /SSD3/full_pipeline_test/Psi

Running reg.sh PRI 0 l sample66 from /SSD3/full_pipeline_test/Psi
Registration already complete for /SSD3/full_pipeline_test/Psi/sample66, skipping

Running rb.sh PRI 4 3.5292 3.5 sample66 from /SSD3/full_pipeline_test/Psi
Rolling ball subtraction already run for sample66, skipping
sample66_02x_down_ochann_rb4_chan.nii.gz exists, skipping
sample66_ochann_rb4_gubra_space.nii.gz exists, skipping

Running z_brain_template_mask.sh l sample66_ochann_rb4_gubra_space.nii.gz from /SSD3/full_pipeline_test/Psi/sample66
sample66_ochann_rb4_z_gubra_space.nii.gz exists, skipping

Running czi_to_tif.sh 200 sample94 from /SSD3/full_pipeline_test/Psi
488 and ochann tifs for sample94 exist, skipping

Running 488_to_nii.sh 3.5292 3.5 sample94 from /SSD3/full_pipeline_test/Psi

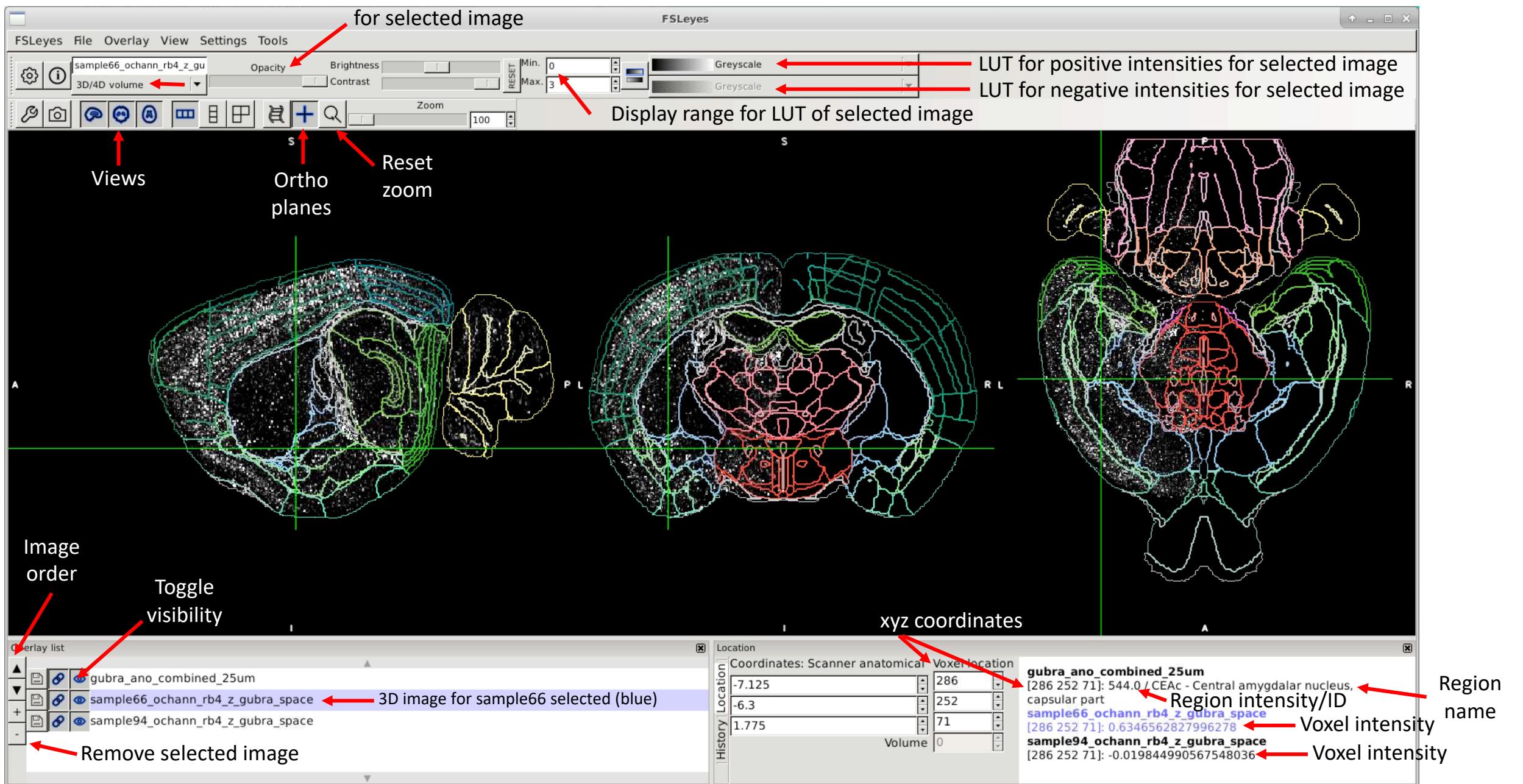
Running reg.sh PLS 0 l sample94 from /SSD3/full_pipeline_test/Psi
Registration already complete for /SSD3/full_pipeline_test/Psi/sample94, skipping

Running rb.sh PLS 4 3.5292 3.5 sample94 from /SSD3/full_pipeline_test/Psi
Rolling ball subtraction already run for sample94, skipping
sample94_02x_down_ochann_rb4_chan.nii.gz exists, skipping
sample94_ochann_rb4_gubra_space.nii.gz exists, skipping

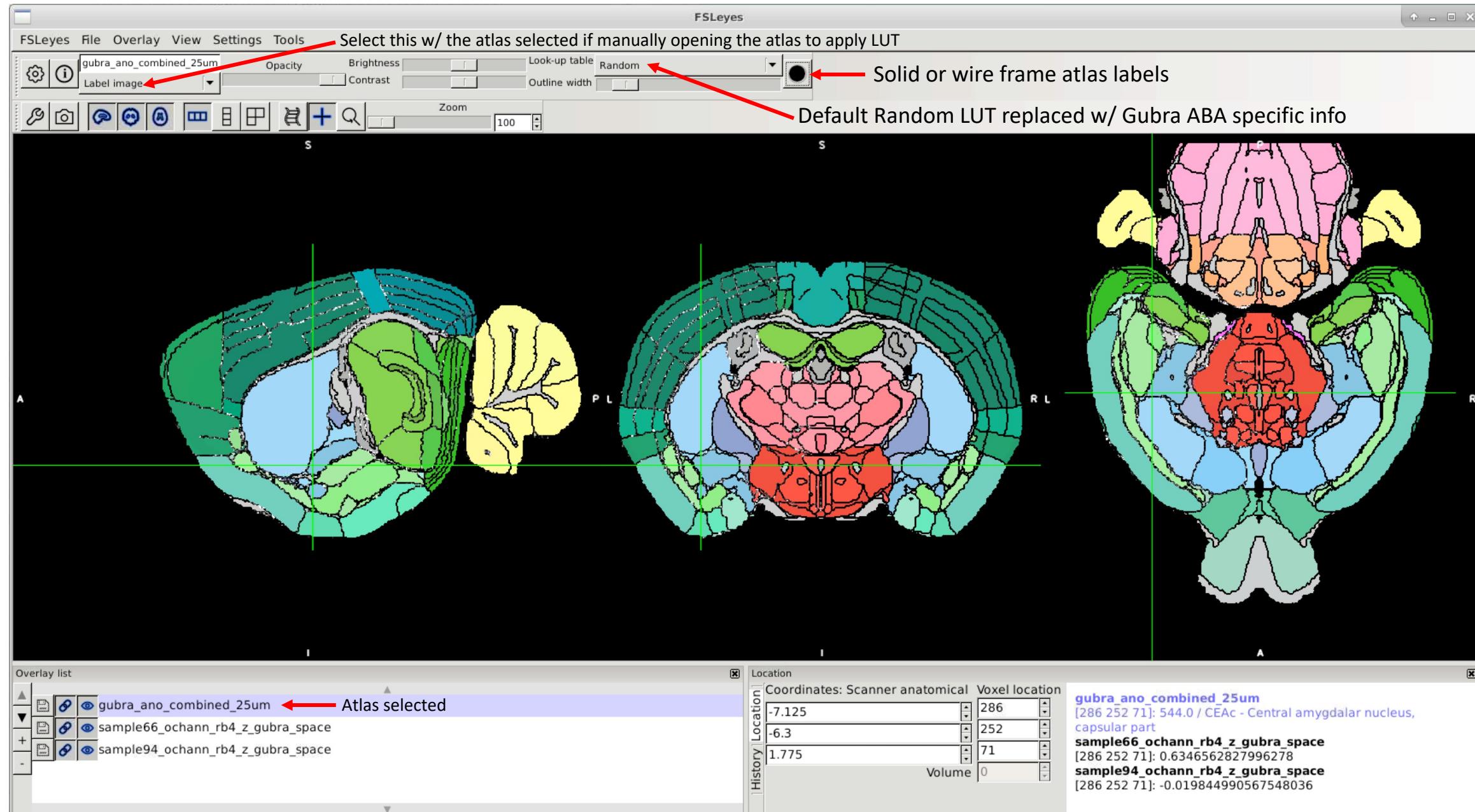
Running z_brain_template_mask.sh l sample94_ochann_rb4_gubra_space.nii.gz from /SSD3/full_pipeline_test/Psi/sample94
sample94_ochann_rb4_z_gubra_space.nii.gz exists, skipping

Running fsleyes.sh 0 3 from /SSD3/full_pipeline_test/Psi_summary/ochann_rb4_z_gubra_space
```

find_clusters.sh #generates z-scored rb bkg subtracted volumes in atlas space



find_clusters.sh #fsleyes.sh 0 3 will set display range of inputs and load 25 um atlas



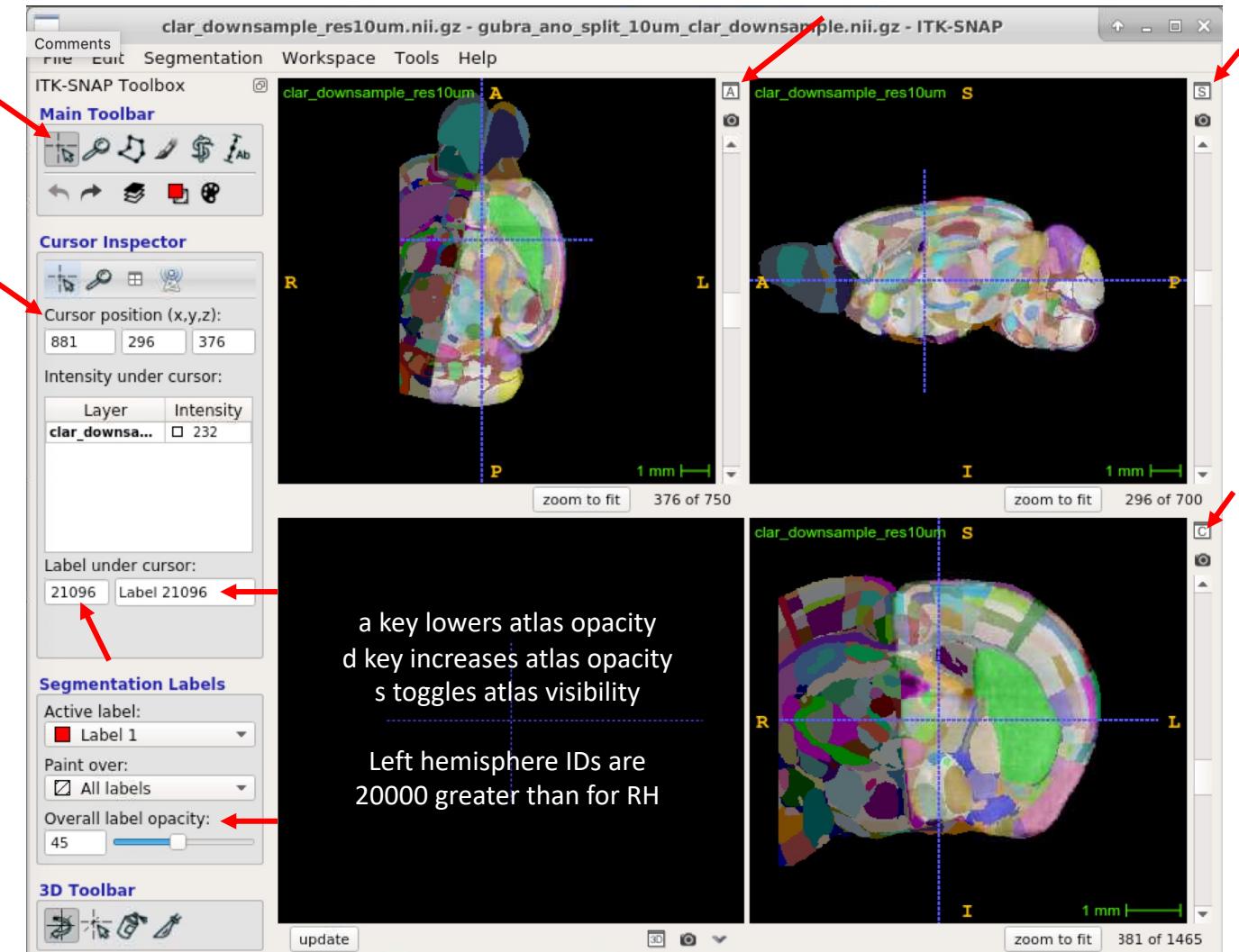
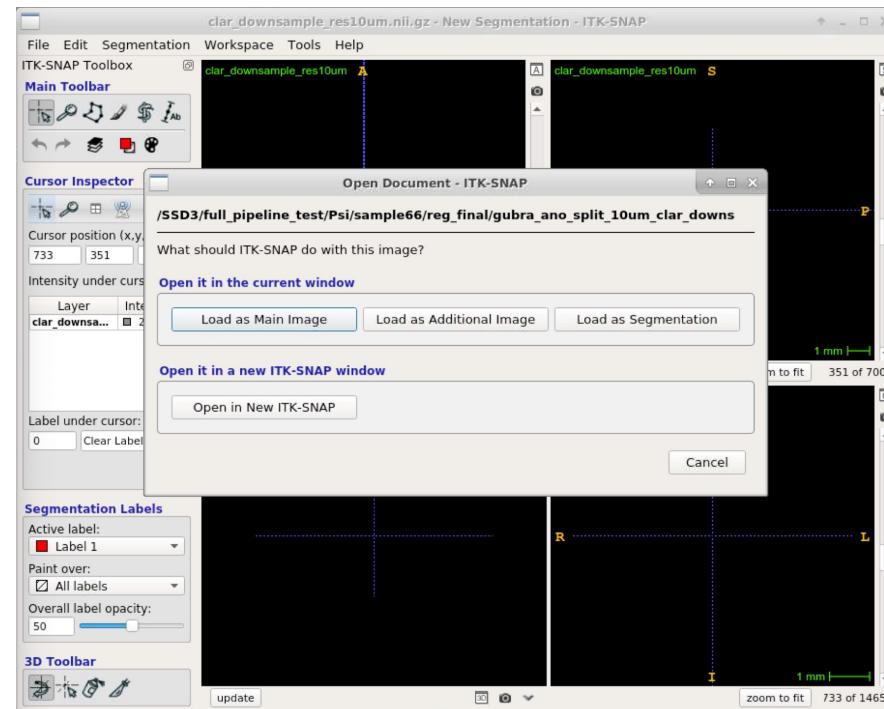
Custom FSL LUT

Region IDs /
intensities

| | ABA RGB values | | | |
|----|--|---------|---------|---|
| 1 | 1.00000 0.29804 0.24314 TMv - Tuberomammillary nucleus, ventral part | | | |
| 2 | 0.09412 | 0.50196 | 0.39216 | SSp-m6b - Primary somatosensory area, mouth, layer 6b |
| 6 | 0.80000 | 0.80000 | 0.80000 | int - internal capsule |
| 7 | 1.00000 | 0.68235 | 0.43529 | PSV - Principal sensory nucleus of the trigeminal |
| 9 | 0.09412 | 0.50196 | 0.38824 | SSp-tr6a - Primary somatosensory area, trunk, layer 6a |
| 10 | 1.00000 | 0.56471 | 1.00000 | SCig - Superior colliculus, motor related, intermediate gray layer |
| 12 | 1.00000 | 0.65098 | 1.00000 | IF - Interfascicular nucleus raphe |
| 15 | 1.00000 | 0.56471 | 0.62353 | PT - Parataenial nucleus |
| 17 | 1.00000 | 0.56471 | 0.99608 | SCIw - Superior colliculus, motor related, intermediate white layer |
| 19 | 0.49412 | 0.81569 | 0.29412 | IG - Induseum griseum |
| 20 | 0.19608 | 0.72157 | 0.14510 | ENTl2 - Entorhinal area, lateral part, layer 2 |
| 22 | 0.00000 | 0.62353 | 0.67451 | PTLp - Posterior parietal association areas |
| 23 | 0.50196 | 0.75294 | 0.88627 | AAA - Anterior amygdalar area |
| 26 | 1.00000 | 0.56078 | 1.00000 | SCdg - Superior colliculus, motor related, deep gray layer |
| 27 | 1.00000 | 0.56471 | 0.61961 | IGL - Intergeniculate leaflet of the lateral geniculate complex |
| 28 | 0.19608 | 0.72157 | 0.14118 | ENTl6a - Entorhinal area, lateral part, layer 6a |
| 30 | 1.00000 | 0.36471 | 0.31373 | PVa - Periventricular hypothalamic nucleus, anterior part |
| 33 | 0.03137 | 0.52157 | 0.54902 | VISp6a - Primary visual area, layer 6a |
| 35 | 0.99608 | 0.56471 | 1.00000 | III - Oculomotor nucleus |
| 36 | 0.00000 | 0.61176 | 0.45882 | GU1 - Gustatory areas, layer 1 |
| 38 | 1.00000 | 0.36471 | 0.30980 | PVH - Paraventricular hypothalamic nucleus |
| 41 | 0.03137 | 0.52157 | 0.54510 | VISpm2/3 - posteromedial visual area, layer 2/3 |
| 42 | 1.00000 | 0.56078 | 0.99608 | SCdw - Superior colliculus, motor related, deep white layer |
| 50 | 0.99608 | 0.56471 | 0.99608 | PRC - Precommissural nucleus |
| 52 | 0.19608 | 0.71765 | 0.14510 | ENTl3 - Entorhinal area, lateral part, layer 3 |
| 54 | 0.79608 | 0.80000 | 0.79608 | mfb - medial forebrain bundle |
| 56 | 0.50196 | 0.80392 | 0.97255 | ACB - Nucleus accumbens |

Registration of the template/atlas to the brain can be checked with itksnap

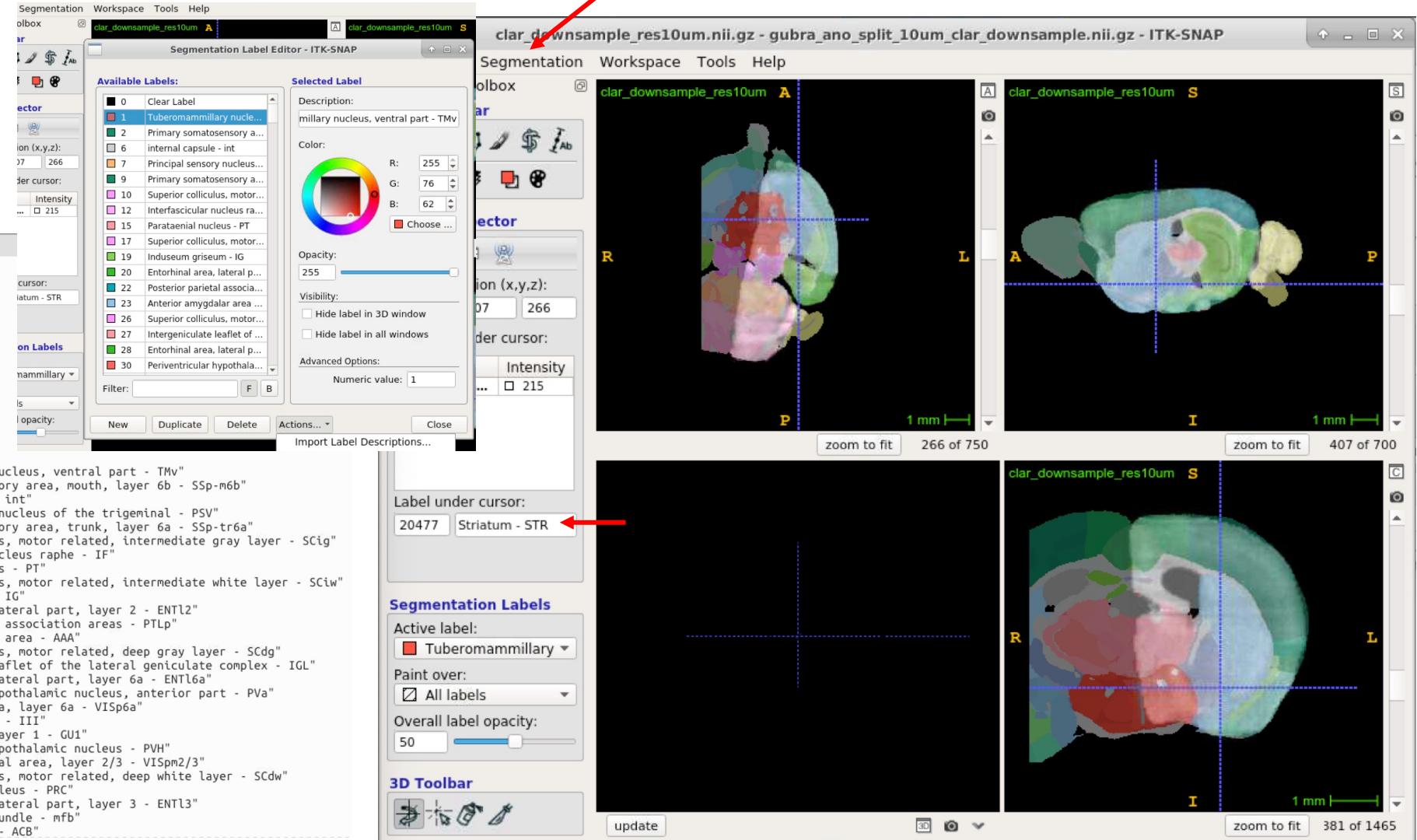
- Launch itksnap (e.g., run: itksnap #if alias set up)
- Drag/drop in ./sample??/reg_final/clar_downsample_res10um.nii.gz
- Drag/drop in ./sample??/reg_final/gubra_ano_split_10um_clar_downsample.nii.gz and Load as Segmentation
- Control+j to auto-adjust brightness/contrast or Tools → Image Contrast → Contrast Adjustment... → Adjust Maximum: (display range)



Registration of the template/atlas to the brain can be checked with itksnap (w/ ABA LUT)

- Segmentation → Label Editor... → Actions... → Import Label Descriptions... → browse to /home/bear → open custom ABA LUT: itksnap_Gubra_LUT.txt → OK → Close
- When region name is cut off, select text & drag cursor to view more (or copy/paste)

```
itksnap_Gubra_LUT.txt
#####
# ITK-SnAP Label Description File
# File format:
# IDX -R- -G- -B- -A-- VIS MSH LABEL
# Fields:
# IDX: Zero-based index
# -R: Red color component (0..255)
# -G: Green color component (0..255)
# -B: Blue color component (0..255)
# -A-: Label transparency (0.00 .. 1.00)
# VIS: Label visibility (0 or 1)
# IDX: Label mesh visibility (0 or 1)
# LABEL: Label description
#####
0 0 0 0 0 0 "Clear Label"
1 255 76 62 1 1 1 "Tuberomammillary nucleus, ventral part - TMv"
2 24 128 100 1 1 1 "Primary somatosensory area, mouth, layer 6b - SSp-m6b"
6 204 204 204 1 1 1 "internal capsule - int"
7 255 174 111 1 1 1 "Principal sensory nucleus of the trigeminal - PSV"
9 24 128 99 1 1 1 "Primary somatosensory area, trunk, layer 6a - SSp-tr6a"
10 255 144 255 1 1 1 "Superior colliculus, motor related, intermediate gray layer - SCig"
12 255 166 255 1 1 1 "Interfascicular nucleus raphe - IF"
15 255 144 159 1 1 1 "Parataenial nucleus - PT"
17 255 144 254 1 1 1 "Superior colliculus, motor related, intermediate white layer - SCiw"
19 126 208 75 1 1 1 "Induseum griseum - IG"
20 50 184 37 1 1 1 "Entorhinal area, lateral part, layer 2 - ENTL2"
22 0 159 172 1 1 1 "Posterior parietal association areas - PTLp"
23 128 192 226 1 1 1 "Anterior amygdalar area - AAA"
26 255 143 255 1 1 1 "Superior colliculus, motor related, deep gray layer - SCdg"
27 255 144 158 1 1 1 "Intergeniculate leaflet of the lateral geniculate complex - IGL"
28 50 184 36 1 1 1 "Entorhinal area, lateral part, layer 6a - ENTL6a"
30 255 93 80 1 1 1 "Periventricular hypothalamic nucleus, anterior part - PVa"
33 8 133 140 1 1 1 "Primary visual area, layer 6a - VISP6a"
35 254 144 255 1 1 1 "Oculomotor nucleus - III"
36 0 156 117 1 1 1 "Gustatory areas, layer 1 - GU1"
38 255 93 79 1 1 1 "Paraventricular hypothalamic nucleus - PVH"
41 8 133 139 1 1 1 "posteromedial visual area, layer 2/3 - VISPm2/3"
42 255 143 254 1 1 1 "Superior colliculus, motor related, deep white layer - SCdw"
50 254 144 254 1 1 1 "Precommissural nucleus - PRC"
52 50 183 37 1 1 1 "Entorhinal area, lateral part, layer 3 - ENTL3"
54 203 204 203 1 1 1 "medial forebrain bundle - mfb"
56 128 205 248 1 1 1 "Nucleus accumbens - ACB"
```



After checking brain/atlas alignment for all samples, set up folder for voxel-wise stats



- Make folder for voxel-wise stats (e.g., `glm_ttest_<EXP>_rb*_z`) in the experiment summary folder
 - Add sample??_ochann_rb*_z_gubra_space.nii.gz images & use mirror.sh as needed to overlay hemispheres (e.g., to pool hemispheres of wholebrains)
 - Add prefixes (e.g, <group1/2>_sample??_ochann_rb4_z_gubra_space.nii.gz). Group order is alphabetical.
 - For quickly renaming files, click magnifying glass → Bulk Rename → drag/drop in files to rename → click broom to clear
- For more info, run: `glm.sh`



```
(miracl) bear@cb-Precision-7920-Tower:/SSD3/full_pipeline_test/Psi_summary$ glm.sh help
```

glm.sh [non-template mask in 25 um Gubra atlas space]

if glm_folder/stats/design.fts exists, run ANOVA, else run t-test

With whole brains, left and right sides can 1st be overlayed with mirror.sh, then use the left template mask for non-lateral data

For a two-sample unpaired t-test based on permutation testing (takes a few days to run):

- 1) make glm folder named sufficiently (e.g., glm_<EX>_rb<4>_z_<contrast> for t-test or anova_<EX>_rb<4>_z)
- 2) add <condition1/2>_sample*_gubra_space.nii.gz files and follow prompts
- 3) open inputs in fsleyes & the atlas to check alignment and that sides are correct:
fsleyes.sh <display range min> <display range max> [leave blank to process all .nii.gz files or enter specific files separated by spaces]
- 4) run glm.sh from glm folder and follow prompts (outputs to ./stats)
tstat1 = group 1 intensities > group 2 (group order is alphabetical [use ls to check order])
tstat2 = group 1 < group 2
vox_p images are uncorrected p value maps
<https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/GLM>
<https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/Randomise/UserGuide>

For a 2x2 ANOVA, before running this script make ./anova_<EX>_rb<4>_z/stats/design

open terminal from ./stats and run: fsl

Misc -> GLM Setup

GLM Setup window:

Higher-level / non-timeseries design
inputs: <total # of samples>

EVs tab in GLM window:

of main EVs: 4

Name EVs (e.g., EV1 = group 1)

Group should be 1 for all

Make design matrix:

Under EV1 enter 1 for each subject in group 1 (1 row/subject). EV2-4 are 0 for these rows

Under EV2 enter 1 for each subject in group 2, starting w/ row after the last row for group 1

Follow this pattern for EV3 and EV4

Contrasts & F-tests tab in GLM window:

Contrasts: 3

C1: Main_effect_<e.g., drug> 1 1 -1 -1 (e.g., EV1/2 are drug groups and EV3/4 are saline groups)

C2: Main_effect_<e.g., context> 1 -1 1 -1 (e.g., EV1/3 were in context1 and EV2/4 were in context2)

C3: Interaction 1 -1 -1 1

GLM Setup window:

Save -> click design -> OK

run: glm.sh from anova folder

vox_p_fstat1=1st main effect 1-p values

vox_p_fstat2=2nd main effect

vox_p_fstat3=interaction

If outside of Heifets lab, update path to template mask: /usr/local/miracl/atlas/ara/gubra/

Run glm.sh in glm folder for voxel-wise stats and respond to prompts (outputs to ./stats/)

```
inal_GLMs/test$ glm.sh
```

Enter side of the brain to process (l, r, both) or (m) for mask specified by 1st positional argument: l

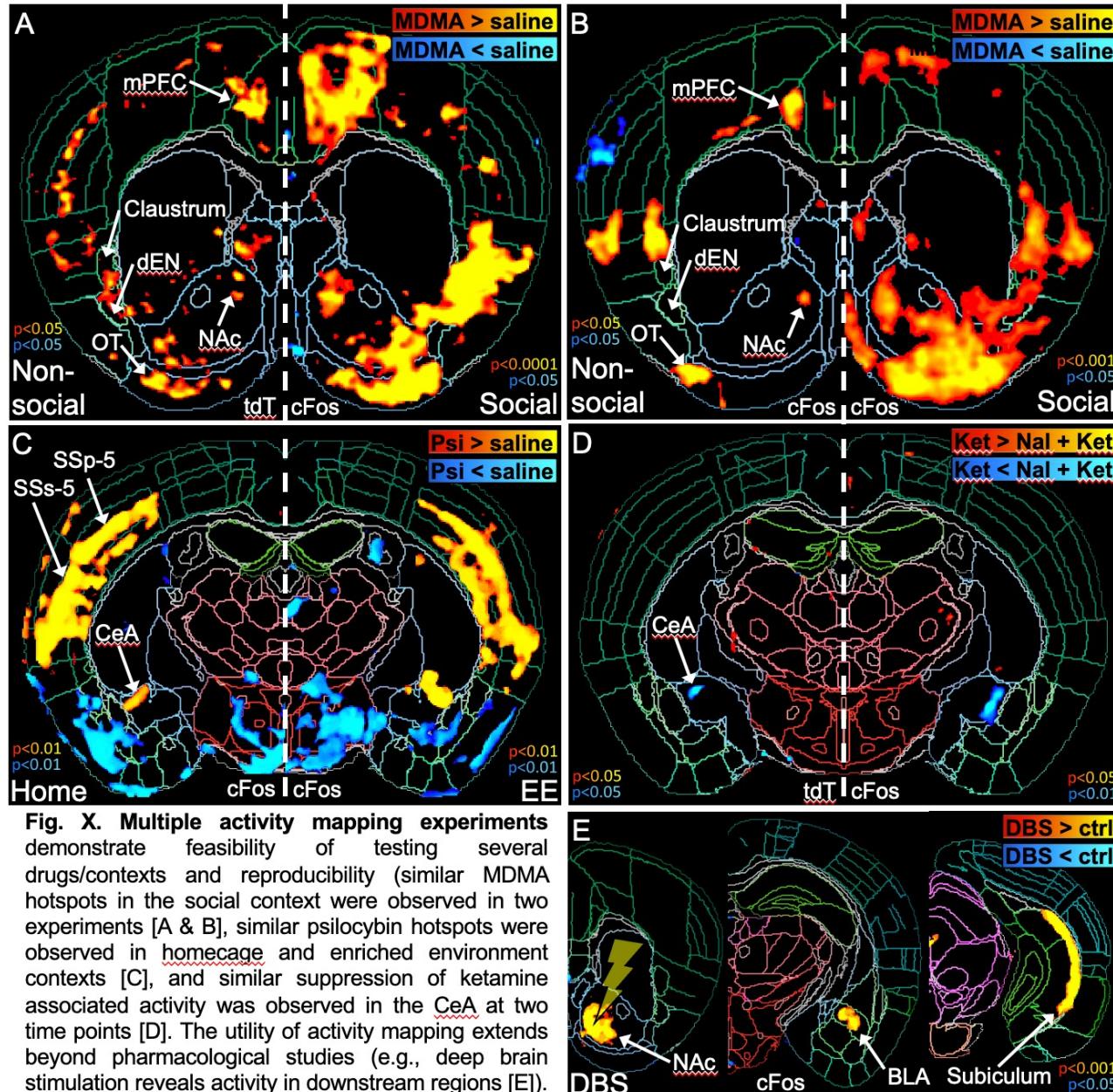
Enter additional randomise options (aside from --uncorrp -x) or just press enter (for help run: randomise -h):

Enter # of permutations (e.g., 6000 [must be divisible by 300]): 6000

Enter kernel radius for smoothing w/ fslmaths in mm (e.g., 0.05): 0.05 █

For more randomise options, run: randomise

Explore vox_p (uncorr 1-p value) maps in fsleyes



Adjust display range to threshold significance
(e.g., for uncorr $p<0.001$ max dr $\rightarrow 0.999$) or use
fdr.sh for voxel-wise
correction or ez_thr.sh for
cluster-wise correction

Run: `validate_clusters2.sh` #1) segment cells w/ ilastik (start when
`./sample/ochann/tifs` exists) & refine by consensus (see `ilastik.sh` and `consensus.sh`)
 and 2) get cell densities etc., in hot/cold spots (clusters)

```
(miracl) bear@cb-Precision-7920-Tower:/SSD3/full_pipeline_test/Psi_summary$ validate_clusters.sh
```

Run `validate_clusters.sh` from experiment summary folder and follow prompts in terminal.

This script aims to validate clusters of significance (hot/cold spots) identified from voxel-wise stats (`find_clusters.sh -> glm.sh`). `validate_clusters.sh` consists of two main phases (segmentation of cells in raw data and measuring cell density in clusters). See `ilastik.sh` help and `consensus.sh` help for how to set up phase 1 (ochann/tif series is required). For more info on phase 2 (requires output(s) from `glm.sh`, `reg.sh`, `consensus.sh` as well as 488/tifs), run: `validate_clusters.sh` help

Enter path/exp_dir list (process all samples) or path/sample?? list (for specific samples) separated by spaces: '/media/bear/14Tb-1/Psi_NE/Psi_NE_LH_cfos/Final_GLMs/glm_z_rb4_acute_drug_effect/stats/test'

Enter s to segment ochann & run `consensus.sh`, v for cluster validation, or both: v

Enter path/stat_image.nii.gz for cluster validation: '/media/bear/14Tb-1/Psi_NE/Psi_NE_LH_cfos/Final_GLMs/glm_z_rb4_acute_drug_effect/stats/glm_z_rb4_acute_drug_effect_psilocybin-15_saline-18_vox_p_tstat1.nii.gz'

Enter q value for voxel-wise FDR correction (e.g., 0.05 or 0.01) or n for using cluster correction: 0.01

Enter min cluster size in voxels: 100

Enter side of the brain (l, r, or both): l

Enter xy voxel size (um), s to get once from `sample_overview.csv`, or m for metadata for each sample: s

Which clusters to process? all, '{1..4}' (range), or '1 2 4' (select clusters): all

Enter y for regional cell counts/volumes in clusters or n for just total cell counts: y

Enter y to warp stats to native, crop it, and find/get most sig slices for montage (else: n): y

For raw/rb* montage tiles, list folders separated by spaces (.e.g, 'ochann ochann_rb4') (else: n): ochann ochann_rb4

```
:/Psi_NE_LH_cfos/sample90/clusters/glm_ANOVA_Psi_cFos_rb4_z_vox_p
glm_ANOVA_Psi_cFos_rb4_z_vox_p_fstat1_FDR0.01_MinCluster100
Name
ABAcluster_cropped
ABAcluster_masks
ABAconsensus_cropped
bounding_boxes
cluster_masks
clusters_cropped
cluster_volumes
native_atlas
native_cluster_index
```

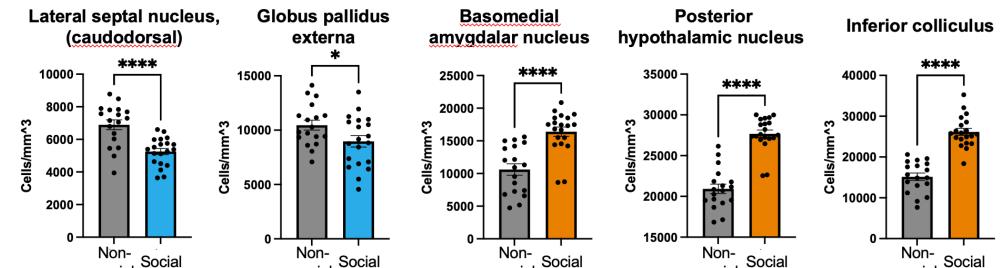
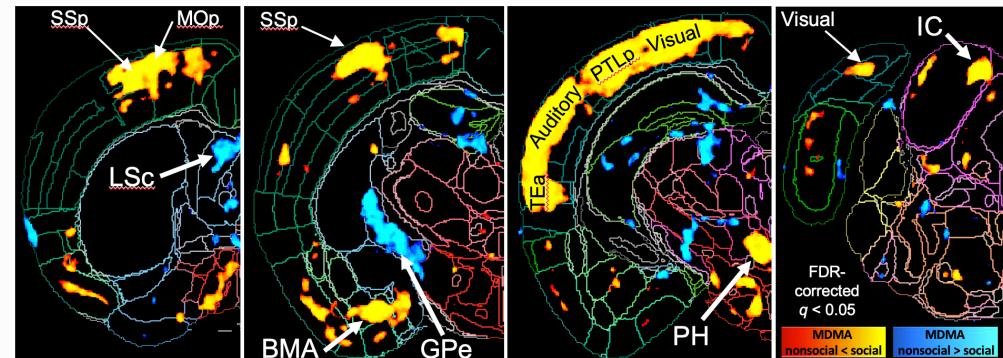


Fig. X. Novel regions modulated by MDMA in a social vs non-social context. For example, activity was increased in the BMA, PH, & IC (FDR $q<0.01$ for cluster validation) and decreased in the LSc & GPe (FDR $q<0.05$ for cluster validation)

(miracl) bear@cb-Precision-7920-Tower:/SSD3/full_pipeline_test/Psi_summary\$ validate_clusters.sh help

Run validate_clusters.sh from experiment summary folder and follow prompts in terminal.

This script aims to validate clusters of significance (hot/cold spots) identified from voxel-wise stats (find_clusters.sh -> glm.sh). validate_clusters.sh consists of two main phases (segmentation of cells in raw data and measuring cell density in clusters). See ilastik.sh help and consensus.sh help for how to set up phase 1 (ochann/tif series is required). For more info on phase 2 (requires output(s) from glm.sh, reg.sh, consensus.sh as well as 488/tifs), run: validate_clusters.sh help

This script can run ilastik.sh and consensus.sh as part of the segmentation phase and can separately or serially run subscripts for cell density measurements etc. Scripts for the second phase include:

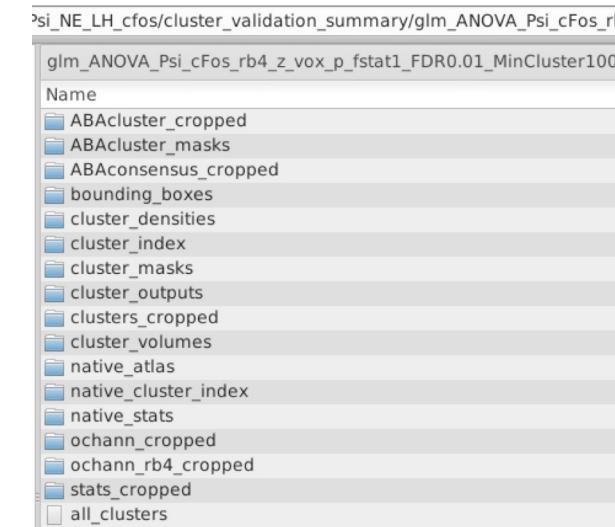
fdr.sh or ez_thr.sh (voxel-wise or cluster-wise correction for multiple comparisons, respectively)
to_native.sh (warp cluster index image [w/ IDs reversed] and the thresholded stats map to tissue space and scale to full res. Also scales warped atlas to full res)
cp_prior_clusters.sh (to avoid redundant processing, relevant prior data copied to new cluster analysis folder if using same p value thresh and new min cluster size)
cluster_masks.sh (makes binary masks for each cluster at full res from native rev_cluster_index)
bounding_box.sh (determines location/extent of cluster for cropping)
crop_cluster.sh (crops data to bounding box of cluster for ochann, ochann_rb*, [ABA]consensus, thresholded stats, & ABACluster data, speeding up cell counting/cluster volume measurements and making of montage tiles)
ABAconsensus.sh (multiplies the warped atlas by the consensus image, converting intensities of cells into the unique brain region intensity where they are located)
3d_count_cluster.sh (fast 3D cell counting on the GPU. Cell intensities can be used for region specific counts)
get_most_sig_slice.sh (finds the 'most significant' slice in the thresholded stats image for a cluster by measuring integrated density of each slice)
extract_most_sig_slice.sh (saves the 'most sig' slice for raw, rb*, consensus, and thresholded stats data)

Phase 1 outputs saved in sample folders

Phase 2 outputs saved in ./sample??/clusters/ & copied to ./cluster_validation_summary/

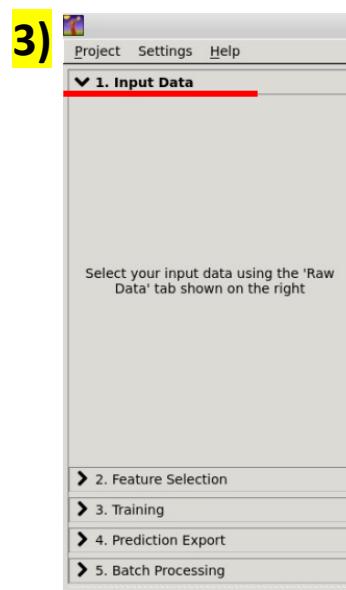
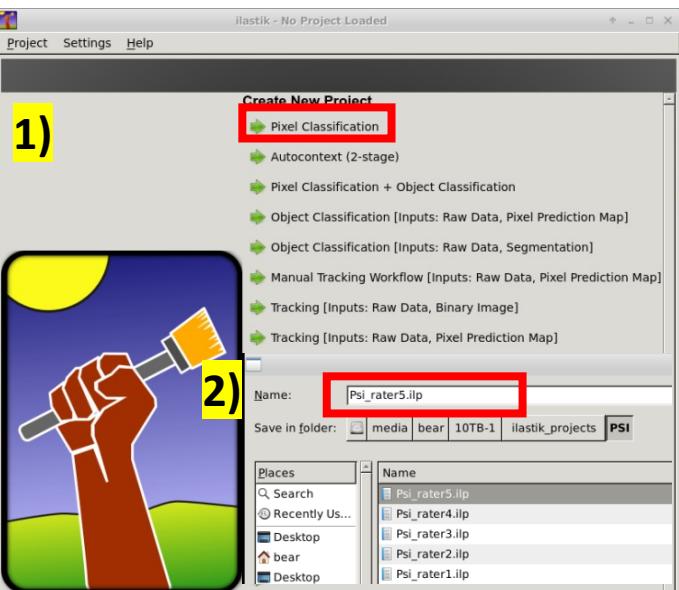
Notes: This requires stats map, reg.sh outputs, sample??/consensus/sample??_consensus.nii.gz and sample??/488/tifs. If you stop processing (control+c), delete partial files if any (e.g., in ./sample??/clusters/unique_cluster_folder/...). If rerunning warp to native with inputs that have been fixed, delete an intermediate file (same name as input) in clar_allen_reg. If viewing image in FIJI and content looks black, control+shift+c -> change min to 0 and max to 1 (don't click Apply). If sample is flipped, copy and flip rev_cluster_index & stat_thresh. Rename flipped images as the original before warping to native. Just process the flipped sample s and, after warping, revert input names to avoid warping a flipped image to native for non-flipped samples. If outside of Heifets lab, update path to template masks in script or use custom masks.

Outputs in ./samples??/clusters/"clusters_folder"/ &
exp_summary/cluster_validation_summary/"clusters_folder"/

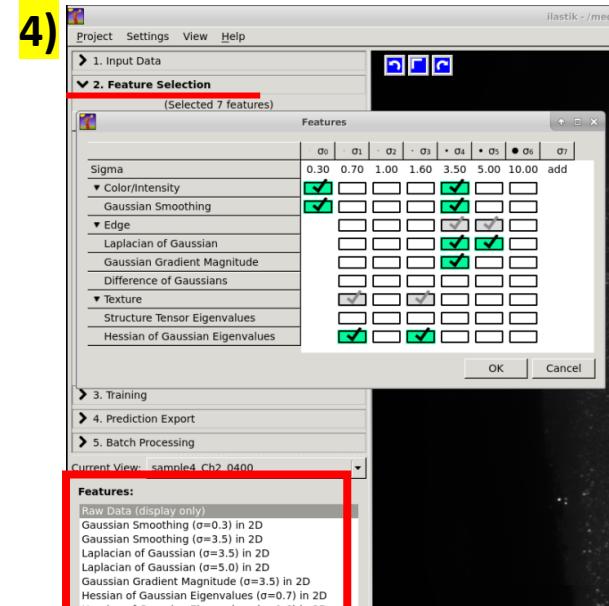


Ilastik Training Tutorial

- Run: ilastik
 - 1) Select “Pixel Classification”
 - 2) Name .ilp project as <Exp>_rater*.ilp and save in an experiment specific ilastik_projects directory
 - 3) Input data: 3 tif files spanning the z stack for 3 samples in each condition
 - 4) Feature selection: use same features for each rater, refine as needed

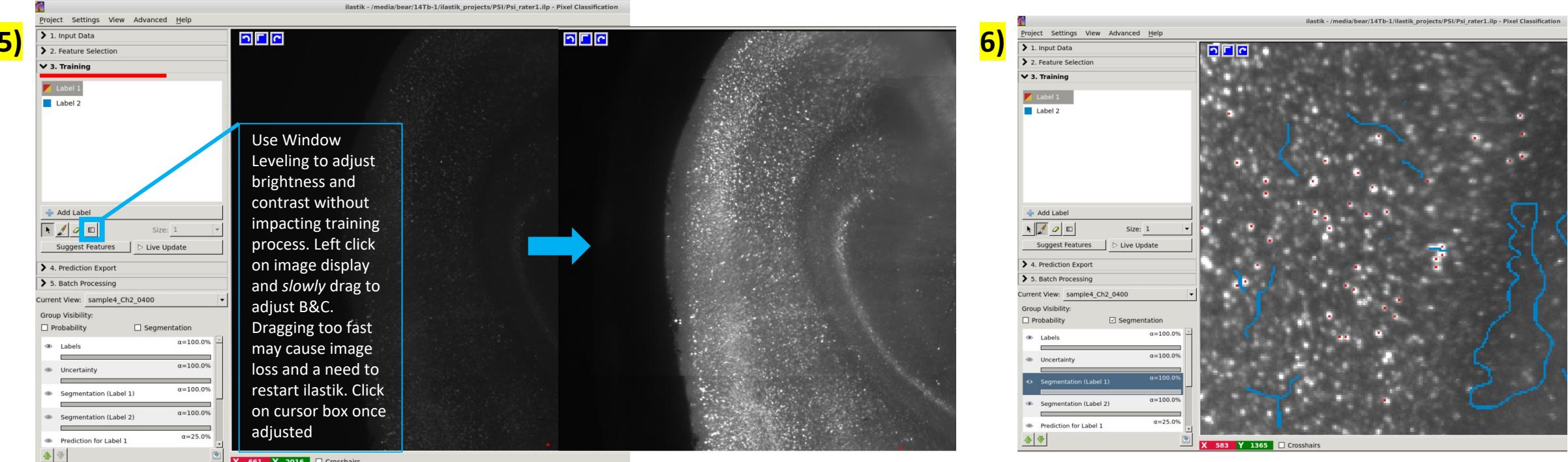


| | Nickname | Location | Internal Path | Axes | Shape | Data Range |
|----|-------------------|--------------------|---------------|------|----------------|------------|
| 1 | sample4_Ch2_0400 | Project Int... [] | | yxc | (3180, 156...) | |
| 2 | sample4_Ch2_0800 | Project Int... [] | | yxc | (3180, 156...) | |
| 3 | sample4_Ch2_1200 | Project Int... [] | | yxc | (3180, 156...) | |
| 4 | sample14_Ch2_0400 | Project Int... [] | | yxc | (3188, 158...) | |
| 5 | sample14_Ch2_0800 | Project Int... [] | | yxc | (3188, 158...) | |
| 6 | sample14_Ch2_1200 | Project Int... [] | | yxc | (3188, 158...) | |
| 7 | sample28_Ch2_0400 | Project Int... [] | | yxc | (3194, 157...) | |
| 8 | sample28_Ch2_0800 | Project Int... [] | | yxc | (3194, 157...) | |
| 9 | sample28_Ch2_1200 | Project Int... [] | | yxc | (3194, 157...) | |
| 10 | sample40_Ch2_0400 | Project Int... [] | | yxc | (3200, 154...) | |
| 11 | sample40_Ch2_0800 | Project Int... [] | | yxc | (3200, 154...) | |
| 12 | sample40_Ch2_1200 | Project Int... [] | | yxc | (3200, 154...) | |
| 13 | sample44_Ch2_0400 | Project Int... [] | | yxc | (3188, 153...) | |
| 14 | sample44_Ch2_0800 | Project Int... [] | | yxc | (3188, 153...) | |
| 15 | sample44_Ch2_1200 | Project Int... [] | | yxc | (3188, 153...) | |
| 16 | sample46_Ch2_0400 | Project Int... [] | | yxc | (3188, 154...) | |



Ilastik Training Tutorial

- 5) Under “Training”, adjust Brightness and Contrast as necessary
- 6) For each same, use “Label 1” to place a red dot at the center of each cell, and use “Label 2” to define background
 - Dedicate ~ 10 minutes to each image
 - Save project after adjusting and training each image

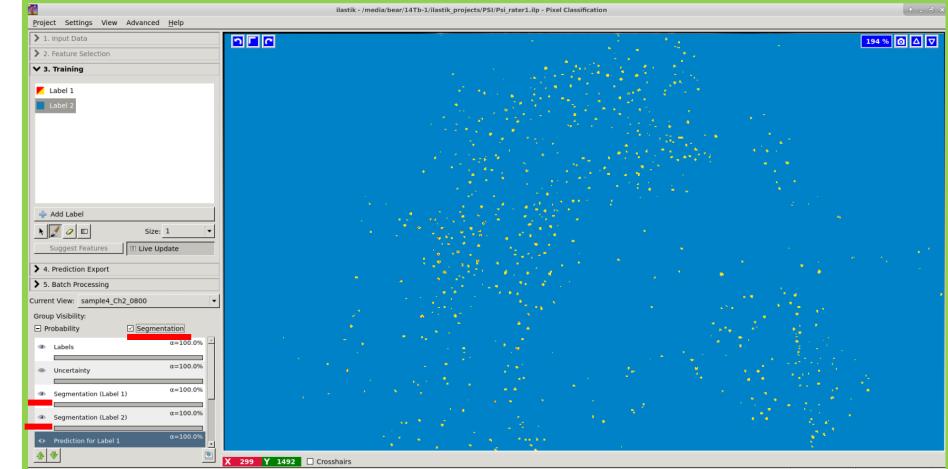
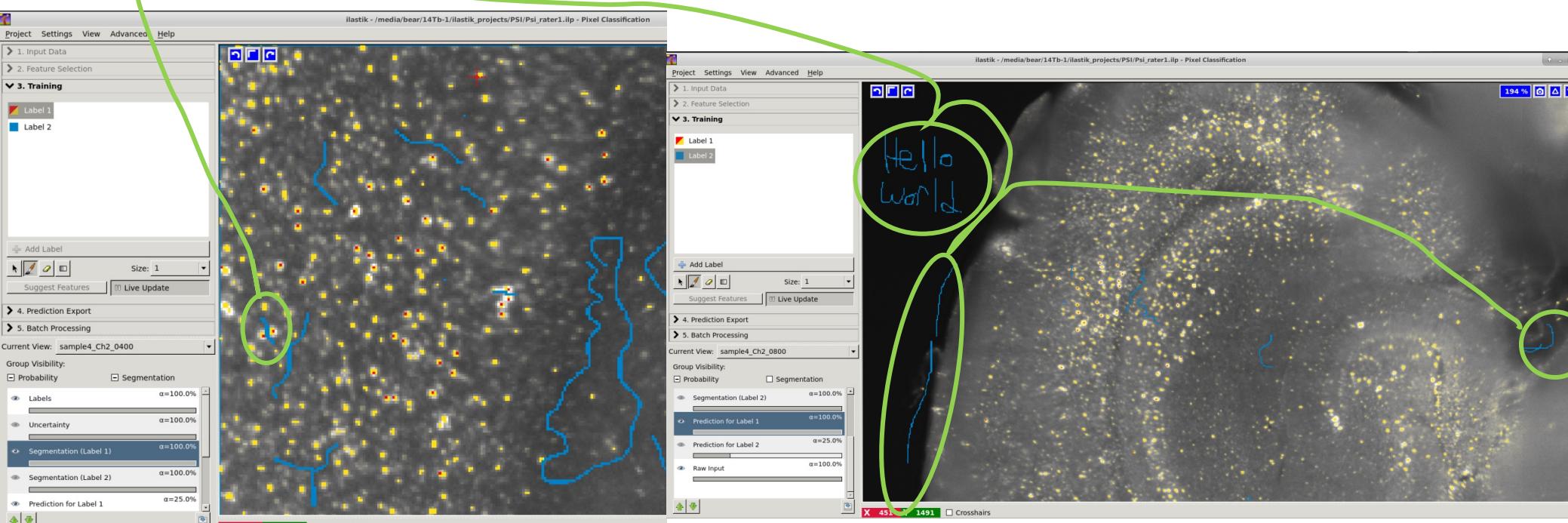


Ilastik Training Tutorial

7) To view accuracy of training, click “Live Update”

Tips:

- To separate fused cells, draw a background line in the darkest pixels separating two cells
- Define some background in non-tissue areas surrounding the brain and in ventricles
- See overall prediction accuracy by clicking “segmentation” and ensuring that segmentation for “label 1” and “label 2” are activated



```
(miracl) bear@cb-Precision-7920-Tower:/SSD3/full_pipeline_test/Psi_summary$ ilastik.sh help
```

From experiment folder containing ./sample??/ochann/tifs, run:

```
ilastik.sh <path/><EXP>_rater1.ilp (trained ilastik project) <'{1..5}' (range for raters 1-5) or '1 2 4' (for specific rater(s))> [leave blank to process all samples or enter sample?? separated by spaces]
```

This script creates the active cell segmentation using Ilastik's pixel classification workflow

To train an Ilastik project, organize training slices into folder (e.g., 3 slices from 3 samples per condition)

launch ilastik (e.g., by running: ilastik)

Pixel Classification -> save as <EXP>_rater1.ilp

<https://www.ilastik.org/documentation/pixelclassification/pixelclassification>

1. Input Data

Raw data -> Add New... -> Add Separate Image(s)... -> select training slices -> Open
ctrl+A -> right click -> Edit shared properties -> Storage: Copy into project file -> Ok

2. Feature Selection

Select Features... -> select same predefined features for each rater
(To choose a subset of features, initially select all [control+a], then refine later)

3. Training

Double click yellow square -> click yellow rectangle (Color for drawing) -> click in triangle and drag to the right to change color to red -> ok

Adjust brightness and contrast as needed (select gradient button and click and drag slowly in the image as needed)

Use control + mouse wheel scroll to zoom, press mouse wheel and drag image to pan (faster if zoomed in)

With label 1 selected, paint on cells

With label 2 selected, paint on background

Turn on Live Update to preview pixel classification (faster if zoomed in) and refine training.

If label 1 fuses neighboring cells, draw a thin line in between them with label 2.

Toggle eyes show/hide layers

The segmentation will be exported, so turn off the Prediction for Label1/2.

Turn alpha (opacity) for Segmentation (Label 2) to 0% and then you can presss on the keyboard to toggle on and off Segmentation (Label 1).

If you accidentally pressa and add an extra label, turn off Live Updates and press X to delete the extra label

If you want to go back too steps 1 & 2, turn off Live Updates off

ChangeCurrent view to see other training slices. Check segmentation for these and refine as needed.

Optional: With fewer features segmentation is faster and less RAM intensive but less accurate

To speed up processing find a subset of suggested features to use by:

Train ilastik initially with all features, turn off Live Updates, click Suggest Features, choose number of features used for pixel classification (7 features is default, but to select number of features automatically, enter 0), Run Feature Selection, from menus at bottom of window note the error and computation time, also record features from Show Feature Names, and click Select Feature Set. Selected features can also be noted by going by to step 2 (Feature Selection). Use the same set of features for all raters (people that indepdently train ilastik) for all samples for a specific experiment. If it is diffi-
cult to train ilastik to distinguish cells from non-cells, adding features might improve performance.

Refine training with subset of features.

Save the project in experiment summary folder and close if using this script to run ilastik in headless mode for segmenting all images.

If using the GUI, proceed to steps 4 and 5.

4. Prediction Export

Source: Simple Segmentation

Choose Export Image Settings... -> Format: -> tif -> if desired, replace {dataset_dir} with path where you want to export segmented images and leave /{nickname}_{result_type}.tif or /{nickname}.tif

5. Batch Processing

Select Raw Data Files... -> Select all ochann images to process

Process all files (if you cancel processing, Clear Raw Data Files and again Select Raw Data Files..., otherwise the project will become corrupted).

```
(miracl) bear@cb-Precision-7920-Tower:/SSD3/full_pipeline_test/Psi_summary$ consensus.sh help
```

Run this from experiment folder:

```
consensus.sh [leave blank to process all samples or enter sample?? separated by spaces]
```

If a pixel was classified as a cell by at least 3/5 raters using Ilastik, then preserve it as a cell

Input (requires 5 raters): EXP/sampleX/seg_ilastik_?/IlastikSegmentation/tifs (from ilastik.sh)

main output: <EXP>/sample??_consensus.nii.gz

montage.sh

