Cobraz Instructions

The Cobraz software is for whole brain voxel-intensity and deformation-field analysis in zebrafish.

The basic workflow is:
1. Use registration software of your choice to register brain scans to one or more reference brains.
2. Use CobraZ, either running on a desktop computer or in a slurm-based computing cluster, to detect differences between groups of brains.

1. Acquire confocal images

Acquire 16 bit brain image stacks at 2 × 2 × 2 µm resolution. This is half the native resolution of ZBB, but downsampling is necessary for reasonable computing time.

2. Register confocal scans to reference brain(s)

Any software can be used for brain registration. Output files need to be either in tif (.tif) or NiFTy format (.nii or .nii.gz).

We use ANTs with settings described in Marquart GD, Tabor KM, Horstick EJ, Brown M and Burgess HA (2017) High precision registration between zebrafish brain atlases using symmetric diffeomorphic normalization. *GigaScience*. 6:1-15. To use ANTs in a slurm based computing cluster, please contact Harry Burgess for bash scripts.

The /references directory contains three references that we use for multi-channel registration:
vglut-mini.nii.gz - vGlut:DsRed pattern
tuba-mini.nii.gz - pan-neuronal mCardinal
gad1b-mini.nii.gz - gad1b:GFP pattern

Optional (1). To measure brain volume, you will need the regions map file reg_SP2c.nii.gz, which is available in the cobraz /regions folder. You will need to apply the inverse of the transformation matrix to the regions map, so it is aligned with the original brain. An example command with ANTS:

```
/usr/local/apps/ANTs/antsbin/bin/antsApplyTransforms -d 3 -v 0 --float -n MultiLabel -i
/data/BurgessLab/ANTS/regions/reg_SP2c.nii.gz -r /lscratch/n300_larva25-01.nii.gz -o
/data/burgessha/vp/iwarped/n300_larva25_iwarp.nii.gz -t [/lscratch/refx/vglut-ref-01-
mini/n300_larva25_0GenericAffine.mat,1] -t /lscratch/refx/vglut-ref-01-
mini/n300_larva25_1InverseWarp.nii.gz
```

Optional (2). For deformation analysis, you need to generate the log jacobian image of the transformation matrix. An example command with ANTS:

```
/usr/local/apps/ANTs/antsbin/bin/CreateJacobianDeterminantImage 3 /lscratch/refx/vglut-
ref-01-mini/n300_larva25_1InverseWarp.nii.gz
/data/burgessha/vp/ljd/n300_larva25_ljd.nii.gz 1
```

3. Run cobraz

Cobraz can run either on desktop PCs (no IDL license needed), or on a slurm-based computing cluster with an IDL license. If you want to run Cobraz on a cluster, contact Harry Burgess for code and bash scripts.

On networked lab computers running cobraz

You do not need an IDL license to run Cobraz on desktops. You just need to install the IDL virtual machine, which is part of the full IDL download. So download and install IDL, and just ignore everything connected to entering a license code. The desktop version works on Windows (tested 7 and 10) and linux (tested Ubuntu).

1. Install the IDL virtual machine

2. Unpack the cobraz folder

3. Run cobraz.sav - this should open in the IDL virtual machine

4. Set the parameters (see section 3.2).
   The configs menu contains default values for voxel-intensity (vox), and deformation-field (ljd) analysis, for images registered to ZBB reference brains (full, 1x1x1 µm), or half-size (2x2x2 µm) registered to half-size ZBB references

5. File → Save parameters

6. Hit the 'All' button to start running.

7. Continue with section 3.

Running parallel instances of Cobraz to reduce computing time

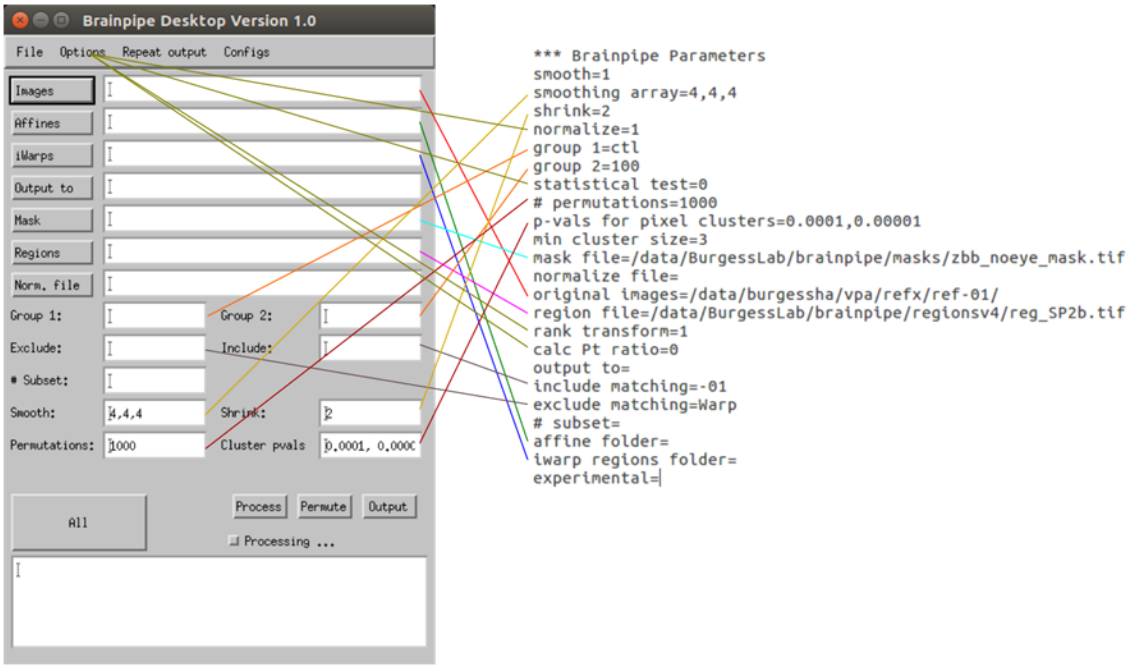
A single instance of Cobraz will take around 30 hours to complete 1000 permutations. To increase throughput, start Cobraz, then wait until the image processing step has completed, and permutation analysis has started, then:

1. Open 2-3 additional instances of Cobraz on the same computer. In each instance, use File → Load parameters and read the params.txt file from the output folder. The same parameters as in the original instance should now populate all the fields. In each instance, press the 'permute' button. The instance will then start running and contribute to the permutation analysis.

2. If the output directory is on a share drive you can open additional instances of cobraz on other networked lab computers, and set them up to run in parallel as above.

3.2. Cobraz parameters

The desktop and swarm based versions of Cobraz use the same set of parameters. In the desktop app, you set the parameters in the interface. In the swarm version, the parameters are in a text file in the form <parameter name>=<value>



smooth= 0/1 - to smooth the input images (default 1)

smoothing array= size in pixels of the boxcar smoothing function (default 6,6,3 - because cells appear to be about 8 x 8 x 8 pixels in size before shrinking by a factor of 2, and imaging dimensions are usually 1x1x2 um)

shrink= value to shrink the input image by (default 2)

normalize= 0-3 - to normalize intensity of each input image

0 No normalization
1 Normalize range of pixels in each brain 0-1000 (default)
2 Normalize (histogram equalization) each brain to the file specified in 'normalize file='
3 Normalize each brain, plane-by-plane to the normalize file.
4 Normalize each brain 0-19999, using the max and min intensity range across all files

group 1= a string that matches filenames for first group of images. The string must match the beginning of the filenames (for example, 'sib' for sibfish_00, sibfish_01 ; but 'fish' would not work)

group 2= as above, for the second group of images (for example 'mut')

statistical test=

0 - unpaired t-tests (default)
1 - paired t-tests

2 - paired t-tests, absolute values (for identifying assymetric areas that are not consistently left or right)

 # permutations= the number of random permutations of images into 'group 1' and 'group 2' that should be performed in order to rank the pixel t-scores and cluster-scores of the actual group 1/2 (default 1000).

pvals for pixel clusters= Comma separated p-value thresholds that are used to find clusters of significant pixels. Clusters will be computed separately for each p-value threshold. Lower p-values are result in smaller and fewer clusters (default 0.0001, 0.00001, 0.000001).

min cluster size= Minimum number of pixels that can constitute a cluster (default 3).

mask file= must be either

(A) full path to a tif file of the same dimensions as the original images that contains a mask of the brain (background pixels=0 ; non-background > 0) (default: /data/BurgessLab/brainpipe/masks/zbb_maskv01.tif).

(B) a threshold number that will be applied to the average of all brains to make a mask (ie all pixels below threshold set to zero, all above set to 255).

original images= the directory that contains the tifs or nifti files to be compared. The image path must end in a '/'. Example /data/BurgessLab/tifs/<directory>/

region file=full path to a tif that contains neuroanatomical segmentation

rank transform= 0/1 - to convert the shrunk/smoothed/normalized image files to per pixel rank transformed values (default 0)

calc Pt ratio= 0/1 - if set, then, following normalization, the code will calculate the ratio of the second channel (-02.tif) and the first channel (-01.tif). Specifically, the result is (100*02-tif/01-tif). Any value greater than 65535 is then set to 65535.

output to=<not used when running under Biowulf - the output path is the directory that zb4 is run from>

include matching=<match string for files to include in the analysis>. For instance, if the image directory contains files that are XXXX-01.tif and XXXX-02.tif and you only want to analyze the XXXX-01.tif files, then use 'filter=-01.tif'

exclude matching=<match string for files to exclude from analysis>. This is useful if the image directory points to the ANTS output folder, and you want to exclude the transformation matrixes, by setting this value to 'Warp'

 # subset=<number of group 1 and group 2 files to sample>. For example: you have 20 group 1 files, and 21 group 2 files. You set the # subset to 8. Then Cobraz will randomly select 8 file from group 1, and 8 files from group 2 to compare.

affine folder=full path to folder that contains ANTs output .mat files

iwarp regions folder=full path to folder that contains back-transformed neuroanatomical segmentation onto the original brain scans

experimental= for debugging purposes.

Notes:
- In params.txt, the '=' after the name of the parameter is essential because it is used to separate parameter name from value.

An example of a Cobraz parameter file

```
*** Brainpipe Parameters
smooth=1
smoothing array=2,2,2
shrink=1
normalize=1
group 1=gpa
group 2=gpb
statistical test=1
# permutations=1000
p-vals for pixel clusters=0.0001,0.00001
min cluster size=3
mask file=/data/cobraz/masks/zbb_half.tif
normalize file=
original images=/data/myexpt/refx/ref-01/
region file=/data/cobraz/regions/reg_SP2c.tif
rank transform=1
calc Pt ratio=0
output to=
include matching=-01
exclude matching=Warp
# subset=
affine folder=
iwarp regions folder=
experimental=
```

4. Interpreting the brain comparison pipeline output files

The output directory will contain these files:

av-all.tif - tif stack with the average for all smoothed/shrunk/normalized brains

av-<group 1 name>.tif - tif stack with average for all smoothed/shrunk/normalized group 1

av-<group 2 name>.tif - tif stack with average for all smoothed/shrunk/normalized group 2

max-stack.tif - max projection for all brains (useful for finding brains that have abnormal appearance)

mask_file.tif - a shrunk version of the maskfile specified, or the mask produced using the use-specified threshold applied to the average brain

Analysis steps. You should open the following files:

1. **max-stack.tif** in ImageJ. Check that all images analyzed look similar.

2. **out_status.txt** in notepad. Check that no errors were reported.
tip: If you ran on unix and are analyzing in windows, then you need to open with WordPad not NotePad to see line breaks.

3. **out_voxels.txt** in notepad. This contains voxel-wise and cluster-wise analysis, for voxels and clusters that are significant across the whole brain.

4. **out_region_intensity.tsv** in excel. This lists the number of significant pixels for each brain region.

5. **out_region_size.tsv** in excel. This lists the size of each region in each brain and whether any regions is significantly different in size between group 1 and group 2.

4. 1. out_voxels.txt

This file has three sections.

(1) During permutation analysis, the brainpipe randomly allocates brains to group 1 and group 2, calculates the t-score (ie the t value from the students t test) for every pixel within the mask and finds the pixel with the highest t-score. It then ranks all the permutations from low to high according to their greatest t-score. Pixels in the actual groups are then significant if their t-score is 'highly ranked', it better than 95% (p=0.05) or 99% (p=0.01) of the greatest t-score for the 1000 permutations.
tip: there is another file called 'scores_from_permutations.txt' that gives the actual permutation values.Useful for making figures illustrating the degree of significance of the t-score.

(A) Positive t-scores (ie where the mean of pixels in group 1 was greater than group 2)
- The highest t-score for pixel comparisons between the actual groups, and the rank of this t-score compared to all the permutations (ie 1000/1000 means the actual highest t-score beat all of the permutations)
- 95th percentile permutations: a t-score value greater than 95% of permutations. Brackets: the number of pixels in the actual comparison with a t-score higher than this value.
- 99th percentile: as above for the 99th highest t-score
- Top t-score: the greatest t-score seen in any of the permutations. Brackets, the number of pixels in the actual comparison with an even higher t-score.

For every threshold (95th, 99th, top) where there are pixels in the actual comparison with a greater t-score, there is also an image file, with those pixels color coded. The name of the file indicates whether the mean of group 1 or group 2 was higher, and what the t-score threshold was according to the filename structure: pix-map_<group name>_hi_t*****.tif
eg pix-map_sib_hi_t0009.74.tif
- A multi-image tif stack with pixels that are higher intensity in sibs than in mutants at the t-score threshold 9.74

B. Negative t-scores (mean of group 2 pixels is greater than group 1 pixels).
- As for section A

(2) Bonferroni corrected significant pixels

This section reports the t-score corresponding to a p value of 0.05 divided by the number of pixels within the mask. Next is the number of pixels that beat this threshold:
T-Scores > threshold (group A > group B)
T-Scores < threshold (group B > group A)

Bonferroni correction is too rigorous, so next is the number of pixels that are significant using the Holms-Bonferroni procedure.

For both bonferroni and holm significant pixels, there are pix-map_XXXX_bonferroni.tif, and pix-map_XXX_holms.tif files that contain the location of these pixels.

C. Permutation tests for significant clusters

For each of the thresholds specified in the 'p-vals for pixels clusters=' parameter, the code finds clusters of adjoining pixels where every pixel has a p-value lower than the threshold. The code then calculates the total t-score for all the adjoined pixels in each cluster. Each brain will likely have several such clusters. The permut_analysis.txt file then specifies
- The highest total t-score for a cluster in the actual comparison, and the rank of this total t-score compared to permutations
- 95th percentile permutations: the cluster-total t-score that is greater than the maximal value seen for 95% of permutations. Brackets: the number of pixels in the actual comparison in clusters that have a higher cluster-total t-score higher than this value.
- 99th percentile: as above for the 99th highest t-score
- Top cluster t-score permutations: the cluster, in all of the permutations, with the highest total t-score.

For every threshold (95th, 99th, top) where the actual comparison has a cluster with a t-score greater than the threshold, there is a corresponding image stack containing the cluster. That is:
- Clusters are shown if:
(1) Each pixel in the cluster is significant at the specified pval threshold
(2) The sum t-score for all pixels is above the rank percentile threshold.
- all pixels belonging to a single cluster given the same intensity code. Even numbered and odd numbered values designate clusters higher and lower for groups 1/2.
The filename contains information on the pvalue and the cluster total t-score threshold, using the system clus-map_pixp-*****_clusT*****.tif
eg clus-map_pixp-1.0e-07_clusT-00004820.tif
- each pixel in the cluster is significant at < 0.0000001
- the sum t-score for all pixels in the cluster is > 4820

4. 2. out_region_intensity.tsv

This file lists each region (in a single row) followed by statistical tests of whether pixel intensities were differernt between group 1 and group 2 in that region.

Column descriptions:

A. Region number
B. Region name. Some are listed as !outside_brain - these are not analyzed

C-E: Simple tests of mean pixel intensity between group 1 and group 2.

C. T-val. Result of student t-test comparing the mean intensityof pixels in group 1 brains in the region, against mean intensity for group 2 brains.
D. p-val. The p-value corresponding to the T-val in C.
E. gp2/gp1. The ratio of mean intensity for group 2 brains to the mean intensity of group 1 brains.

F-J: Permutation analysis of pixels in group 1 and group 2. These columns report pixels in the actual comparisons where group 1 had intensities greater than group 2, and where the significance scores (T-values) beat 999/1000 permutations

F: The maximum T-value for t-tests between pixels in the actual group 1 and group 2 brains, in this region

G: Where that maximum T-value ranks among the 1000 permutation tests in which brains were randomly assigned to 2 groups.

H: The T-value that corresponds to rank 950 out of 1000 random permutations
I: The T-value that corresponds to rank 990 out of 1000 random permutations
J: The highest T-value seen in all 1000 random permutations.
K: The number of pixels in the actual group 1 v group 2 comparisons, with a T-value exceding that reported in column I (ie, pixels significant at a level of 0.01).

L-Q: As for F-J, but looking for pixels with lower intensity in group 1 than in group 2.

There are also pixel-map files produces for the pixels in columns K and Q. These are the files

region_code_by_intensity_XXX-hi.tif - all pixels where group 1 intensity > group 2 (column K) across all regions, color coded by significance level

4. 3. out_region_size.tsv

This file is only produced if your params file has entries for the iwarp regions folder

For each brain (a single row), there are columns describing
(1) The overall brain size (in pixels)
(2) The total size of the mostly-cellular area (in pixels)
(3) The total size of the mostly-neuropil area (in pixels)
(4) The relative size of each region (pixels divided by the total brain size)

There are 4 rows at the bottom. There are:
(1) Means for group 1
(2) Means for group 2
(3) p-value for t-test result comparing means of group 1 and group 2.
(4) the Holm's corrected p-value. If not significant, this is given just with a dash.

5. Import into the brain browser

If your input files were registered to the brain browser, you can directly load the brainpipe output files into the browser. After running zbb.sav:

To load a pix-map file:
1. File→Auto re-size user stack
2. File→Select user 1 stack

This should load the pixel-map into the User-1 file (at the bottom of the transgenic window). You may need to increase the contrast to see the above threshold pixels.

To load a cluster-map file:
1. File→Auto re-size user stack
2. File→Binarize user stack
3. File→Select user 1 stack

The cluster map will be loaded into the User-1 file. Cluster number information is discarded and all pixels in any cluster are assigned the value '125'.