**3D-ARG image analysis pipeline [Multigauge, Matlab, Fiji, Ilastik and R]**

*using Fiji plug-ins ‘Trakem2’, ‘BUnwarpJ’, ‘Volume Viewer’*

**Get ARG files and set all ARG images to same grey scale [Multigauge, Fiji]**

* Multigauge:
  + Export in tiff file
* Fiji:
  + set ROI on microscale highest value and from low microscale the 5th value and measure separately [ctrl+m]
  + Select the complete image (so the measured square is not selected anymore), set the brightness contrast according to these values: Image > Adjust > Brightness/Contrast > Set. Save in separate folder
  + repeat when necessary (measured area might not be homogeneous)

**Crop and resize images (Nissl and photos resized to ARG size) [Fiji]**

* ARG images: open & crop slides separately:
  + Use save\_separate\_slides.ijm [non-automated script, with type = “ARG”]
* Nissl images:
  + Change to RGB format: save\_as\_RGB.ijm
  + Resize: resize\_nissl.ijm
  + Crop slides: save\_separate\_slides.ijm [non-automated script, type = “Nissl”]
* Camera images:
  + Flip all images: rotate\_all\_images.ijm
  + Crop and resize:
    - Take a photo with ruler and measure the size of a glass slide in pixels to calculate the pixel to mm ratio (use analyze > set scale)
    - Crop boxes with the size of the glass slide (based on above determined size), crop\_photo.ijm (adjust rectangle per slice)
    - Resize to ARG slide size: resize\_photo.ijm

**Crop slice outline [Fiji]**

* Nissl staining:
  + Threshold out background and create masks from slice on duplicate image:
    - If contrast is not high: process > enhance contrast (1%), or for stronger contrast use enhance\_contrast.ijm
      * Convert to ilastik file: fiji2ilastik\_conversion.ijm
      * Segment in Ilastik (feature selection: high on edges) in one file, batch process for all files and save in separate folder
      * Import files back in Fiji and use wand to select the area of the slice and create a mask: create\_masks.ijm
    - Adjust manually or draw manually completely
  + Crop slides: nissl\_crop.ijm
* ARG images:
  + Crop slides using Nissl masks (adjust where necessary, adjustment of Nissl image might also be necessary): use arg\_crop.ijm – double check if all 3 type of files have the same dimensions, if not adjust this!
* Camera images:
  + Enhance contrast: photo\_contrast.ijm
  + Create mask: photo\_cropandmask.ijm [non-automatic script]
  + Crop slices: photo\_crop.ijm [photo contrast is also slightly enhanced]

**Repair images [Fiji]**

* Repair images when necessary, in Nissl image, applying to ARG: use slice\_repair.ijm
  + Only make slight adjustments, e.g. (re)move parts that have been ripped off
* For photo’s use: photo\_repair.ijm

**Create stack of camera images [Fiji (Trakem2 plug-in), R]**

* File > New > Trakem2 (blank)
* Drag folder with cropped camera images to right black part, save
* Manual alignment (try to be precise):
  + right click > Link > Unlink all to unlock patches,
  + Go to layers, scroll to 2nd image and right click > red channel on first image.
  + Double click on image, right click > transform (or press t)
  + Repeat transform for all slices (when retransforming, apply transformation till last image). [if photos are taken without minimal movement this step doesn’t have to be repeated, in that case use ‘apply transformation till last image right away].
  + Complement with automatic alignment (right click on image > align layers)
  + Click on image patch (left bar) 🡪 Export > Make flat image
  + Save flat image (right click > export, rgb image) and save trakem2-template
* Run script save\_stack\_images\_separately.ijm on the image stack to create separate images
* Separate slices in folders for T and NS: use save\_T\_NS.ijm
* Open image sequence of T slices in Trakem2, perform automatic alignment, save flat image and run save\_stack\_images\_separately.ijm [in case filesize is changed: use adjust\_size.R
* Open image sequence of NS slices in Trakem2 (use template of T image stack)
  + Add first image of T slice stack (right click on top level: add on z=-1, drag T image file)
  + Adjust transformation of T slice to start position, use change transf\_between.R
  + Open new xml file in Fiji, align NS to T slice (propagating to last layer)
  + Adjust size (use adjust\_size.R or: right click > display > resize canvas)
  + Open new xml file in Fiji, remove T slice, save flat image and run script save\_stack\_images\_seperately.ijm

**Create stack of Nissl, roughly aligned with photo’s [Fiji (Trakem2 plug-in), R]**

* In case extra slices were used in experiment, choose the ones used for analysis.
* Separate slices in folders for T and NS: save\_T\_NS.ijm
* For T slices and NS slices separately:
  + Open new Trakem2 from template
  + Drag folder with Nissl files to canvas (black part)
  + Save xml and unzip xml.gz
  + Adjust size of slices to those of camera images, use: change\_stack\_size.R
  + In trakem2, add first image of camera stack, resize canvas back to previous setting
  + Adjust transformation of T slice to start position, use change transf\_between.R
  + Open the new xml file in Fiji, align first slice to camera slice and remove T slice
  + Adjust size if necessary (right click > display > resize canvas)
  + Manual alignment from last to first, like above, but does not have to be precise. (propagate to first patch/layer)
  + Save flat image and run script save\_stack\_images\_seperately.ijm

**Align ARG image to Nissl image & create stack [Fiji (Trakem2 plug-in), R]**

* Separate slices in folders for T and NS: save\_T\_NS.ijm
* For T slices and NS slices separately:
  + Open new Trakem2 from template (which is the nissl one created above)
  + Drag folder with ARG files to canvas (black part)
  + Save xml and unzip xml.gz
  + Adjust size of slices to those of camera images, use: change\_stack\_size.R
  + In trakem2, add first image of camera stack (add z layer, right click > import image)
  + Open the new xml file in Fiji, align first slice to camera slice and remove T slice
  + Adjust size of stack with adjust\_size.R
  + Run R-script change\_stack\_transf.R (don’t try to do this in MATLAB, it will change the xml-file when using xmlread) with ARG file as source and nissl file as target
  + Open new xml-file in Fiji, adjust size of canvas and save flat image
  + Run script save\_stack\_images\_seperately.ijm on the flat arg stack to have all the images separately as well
* In case there are slices without photo:
  + T Nissl:
    - Copy and open Nissl\_T xml - copy
    - Add z-layers (7) and add the files
    - Manually align them
    - Save xml, save flat image and save separate images
  + NS Nissl:
    - Copy and open Nissl\_NS.xml – copy
    - Add z-layers (7) and add the old T files (so not the aligned ones
    - Transform the T slices to the same position as above (change\_transf\_extraslices.R)
    - Manually align NS with T and remove T slices [they will be warped, so manually is sufficient]
    - Save xml, save flat image and save separate images
  + ARG:
    - Copy and open ARG\_T xml - copy
    - Add z-layers (7) and add the files
    - Run R-script change\_stack\_transf.R to align arg with nissl
    - Save new xml, open in Trakem2, save flat image and save separate images
    - Repeat for NS Nissl

**Transformations & Substract ARG NS from ARG T binding [Fiji (BUnwarpJ plug-in)]**

*In case different nissl slices will be used than ARG slices: warp nissl slice first to both NS and T ARG slice, before proceeding as below.*

* Create transformations (images and transformation files). Use Fiji script batch\_transform\_images.ijm
  + Warp 1: Source: nissl NS; Target: nissl T
  + Warp 2: Source: nissl T; Target: camera T
* QC the warps, if they are not good, some might need landmarks to give extra weighting for certain areas (set landmarks). *Sometimes the warp is not properly processed, so trying again manually can also help.*
* Apply transformation to ARG NS slices [warp based on nissl, so based on anatomy + shape, not based on ligand binding 🡪 nissl NS and T show similar ‘signal’]:

[note that the ARG images first have to aligned with the nissl images, both T and NS!]

* + Warp 1 on ARG NS, with target ARG T. script: warp\_arg\_NS2T
  + QC
* Subtract ARG NS [transformed] from ARG T
  + Process > Image Calculator (and select 32 bit float image as output), worked better. Script: batch\_subtract.ijm
  + In case an important part is not present in NS slide: apply subtraction to non inverted image as well and add missing part on the difference to the result.
  + Alternative: only subtract segmented part after segmentation
* Apply transformation to corrected ARG T slices (i.e. ARG specific binding)
  + Warp 2 on ARG SB, with target camera T. script: warp\_arg2camera.ijm
  + QC
* Adjust alignment in Trakem2 if necessary
  + Apply this alignment to nissl slices change\_stack\_transf.R
    - Save new xml, open in Trakem2, save flat image

**Segmentation [Fiji, Ilastik] - OPTIONAL**

*Note that in next step the radioactivity values corresponding to the segmented areas will be calculated. To make this process easier, the tritium scales can be directly included in the to be segmented files.*

* Export to Ilastik file: export\_h5.ijm
* Ilastik pixel classification:
  + Open images
    - Add a single 3D/4D volume from sequence
    - Click on first file, shift click for all files
    - Check the z-axis box and open
  + Set features (high on intensity and edge)
  + Make labels (for example: high, medium, low, background)
  + Check live and check segmentation
  + Save segmentation, export
  + Import in Fiji with File > Import > Ilastik HDF5, tzyxc, apply LUT
    - Save stack as tiff
    - QC and adjust segmentation if necessary

**Calculate radioactivity values for segmented areas [Ilastik and/or Fiji] - OPTIONAL**

* Checking radioactivity values of segmentation:
  + Export Ilastik labels of the SB slices (per slice)
  + Create a new folder including the SB slices, include the calibration scales
  + Export to Ilastik file: export\_h5.ijm
  + Ilastik pixel classification:
    - Open images
      * Add a single 3D/4D volume from sequence
      * Click on first file, shift click for all files
      * Check the z-axis box and open
    - Import labels per slice (from above exported labels)
    - Export simple segmentation files
  + Import in Fiji with File > Import > Ilastik HDF5, tzyxc, apply LUT
  + Compare radioactivity values of tritium scales with colours the scales are now segmented in
* Checking radioactivity values of ROIs:
  + Set the calibration curve in Fiji:
    - Open the scanned tritium scales
    - Select the separate scales using rectangle
    - Measure density value using ctrl+M
    - Open calibration plug-in: Analyze > Calibrate
    - Set the radioactivity values, corresponding the measured density values
    - Create the calibration line (linear in case linear tritium scale is used)
    - Note down the formula (values of a & b in y = ax + b)
  + Open segmented stack in FIJI
  + Create separate stacks for all the segmented areas (so one stack with only highest binding, one file with only medium binding, etc). Use Image > Adjust > Color Threshold
  + Create ROIs using wand on the segmented areas and apply this ROI to the specific binding slides (unsegmented, press shift+E)
  + Measure the density value in this ROI by pressing ctrl+m
  + Continue to measure several ROIs in several slices for each segmented colour
  + Calculate for each segmented colour the radioactivity corresponding to the density measured, using the above derived formula

**Volume rendering [Fiji (Volume Viewer plug-in)] - OPTIONAL**

* Open segmented stack in Fiji
  + Adjust LUT: Image > Color > Edit Lut…, open LUT
* Create stack with mirrored images and get axial slices: reslice\_all.ijm
* Volume Viewer (Plugins > Volume Viewer)
  + Define and set z-aspect = z/x
    - Z: distance between slices
    - X: slice width / pixels on slice width [use Analyze > Set scale in picture with ruler and end slice image]
  + Set mode on volume, interpolation: tricubic smooth
  + In the menu on the right adjust alfa’s
    - 1D: set the alpha for the tissue background to 0 (low alfa means more transparent). The alpha graph shows higher intensity colours in peaks?? Most right peak is lighest colour (in this case background)
    - 3D-fill: set the gradient to a specific height and add all the different colours separately, make the tissue colour transparent
  + Create an animation, manually: make 36 snapshots with 10 degrees rotation difference, save all images, create stack/video
* Create an animation of non-volume rendered image: volume\_viewer\_animation.ijm
* Export as avi: File > Save as > AVI

**Prepare ARG slices for transformation to MNI space [Fiji]**

* Get slices in right colour scale: adj\_colour\_arg.ijm
* Get mirrored images of the arg slices: get\_mirror\_stack.ijm
* Get file in nifty and in 16 bit grey scale, [resize to approximate size in MNI space]:
  + Compare size of nissl slice with MNI brain:
    - Reslice MNI brain to saggital plane: Image > stacks > reslice. Start at left, flip vertically
    - Open ARG slices, use wand to select the slice, copy and press shift E on MNI brain image to compare ARG slice size with MNI brainstem size
    - Adjust size of ARG image. Image > Adjust > Size. And repeat until approx the right size.
  + Use: reslice\_and\_resize.ijm which includes:
    - Resize canvas (x = 218 \* A, y = 182 \* A, [saggital plane] in which A = scaling factor found above)
    - Downscaled to x = 218 \* 10, y = 182 \* 10 [reslicing with bigger images tends to leave pc out of memory]. Then resliced to axial plane
    - Add empty black images before and after (ARG slices are in 0.5 mm space > in ‘full brain space’ this would be 364 slices, so images to be added = 364 – nr of ARG slices)
    - Downscaled to 218 \* 182 \* 182 mm
    - Resliced
  + Save slices in 16 bit nifti files (stack and separate images): save\_nifti\_grey.ijm

**Transformation of ARG to MNI space [Matlab (SPM)]**

* Align ARG stack with MR image in SPM, resize if necessary
  + Create nifti file in matlab to create the .mat file (use function createnifti)
  + Open brainstem nifti file in SPM (type ‘spm’ + enter, then click on ‘Display’)
  + Rotate brainstem to position as in MNI space
  + Estimate location of anterior commissure and move crosshair to this location
  + Note coordinates and type invert of these coordinates in ‘right’/’forward’/’up’
  + File > generate m-file
  + Check if the image stack is in the right position
    - Create a mask of the stack, use create\_mask.m
    - In SPM open MRI image in MNI-space (using ‘display’)
    - Right click > ROI tool > Launch > open mask file
  + Adjust if necessary (it’s just an estimation)
* Get a brainstem mask
  + Get an MRI average in MNI space
  + Mask out brainstem: mask\_out\_ROI.m
  + Get a SN weighted brainstem, use SN\_weighted\_BS.m
    - Get the SN out of template with multiple ROIs
    - Adjust the (voxel)size to that of the brainstem in MNI space
    - Adjust the intensity so that it equals high intensity for the SN
    - Subtract the SN from the brainstem
* Do further affine transformation in SPM to match the ARG slices better to the MNI brainstem:
  + Linear: rotate, translate in 3 directions
  + Non-linear: scale
* Warp Nissl brainstem to MRI brainstem in MNI-space
  + Use reg\_f3d
  + Alternative via Fiji: batch\_transform\_images2MNI.ijm