

<b>I.</b>	<b>MANIPULATING THE FILE TYPES.....</b>	<b>2</b>
<b>A.</b>	<b>Analyze Image .....</b>	<b>2</b>
1.	Analyze image format .....	2
a)	Header files: .....	2
b)	Image files:.....	3
2.	Reading the parameters of the Analyze image header file.....	3
a)	SHIVA .....	3
<b>B.</b>	<b>Converting Commonly used File Types to Analyze image.....</b>	<b>4</b>
1.	2dseq and Reco.....	4
2.	CIVM and MINC .....	4
<b>C.</b>	<b>Reorienting image files .....</b>	<b>4</b>
<b>D.</b>	<b>Resampling image files .....</b>	<b>4</b>
<b>II.</b>	<b>MOUSE BRAIN MRI IMAGES .....</b>	<b>5</b>
<b>A.</b>	<b>Getting started.....</b>	<b>6</b>
<b>B.</b>	<b>processing of Mouse MRI.....</b>	<b>7</b>
1.	Image processing prior to atlasing .....	7
a)	Skull stripping .....	7
b)	Bias field correction .....	8
2.	Generating a Minimum Deformation Atlas .....	9
a)	Linear alignment of two files .....	10
a)	Nonlinear alignment of two files.....	11
b)	Alignment of multiple files .....	12
<b>C.</b>	<b>labeling Brain Areas .....</b>	<b>15</b>
<b>D.</b>	<b>Applying Deformable atlases .....</b>	<b>15</b>
<b>E.</b>	<b>Volume Analysis.....</b>	<b>16</b>

# I. MANIPULATING THE FILE TYPES

## A. ANALYZE IMAGE

The Analyze image format is a commonly used format by LONI tools.

### 1. ANALYZE IMAGE FORMAT

In order to use many of the LONI tools, your files will need to be in the Analyze image format. The Analyze image format includes two files that must be associated with each other. One is a header file (.hdr) and the other is an Analyze image file (.img). The image files contain pixel values without any associated header information, and the header file contains all the information necessary to interpret the data in the image file (metadata).

#### a) Header files:

The following pieces of information are stored in the header files:

- the number of bits per pixel in the image (8 bit and 16 bit images are supported), called filetype, see more information below
- the image matrix x-dimension
- the image matrix y-dimension
- the image matrix z-dimension (distance between slices)
- the image voxel x-size
- the image voxel y-size
- the image voxel z-size
- the global maximum for the image
- the global minimum for the image
- the size of the .hdr file (used to detect the need for byte swapping)

The units for the image voxel sizes are not specified in the AIR package or the Analyze file format (millimeters are recommended), and **identical units must be used for all three dimensions.**

For AIR the filetype designation can be chosen using the following list; however, keep in mind that Analyze image format can be much more extensive:

0. 8 bits/pixel (values 0 to 255)
1. 16 bits/pixel unsigned short ints (values 0 to 65535)
2. 16 bits/pixel short ints (values 0 to 32767)
3. 16 bits/pixel short ints (values -32767 to 32767)

If you plan on comparing images, the X, Y, Z planes must be correctly specified. It is probably best to use SHIVA to check that the images are in the proper orientation.

## b) Image files:

A single image file contains all of the data for the entire three dimensional volume stored row after row, plane after plane. The image file consists of "raw" voxel intensity values (8 bit and 16 bit images are supported) that are stored sequentially.

Analyze image file format defines the voxel order as follows:

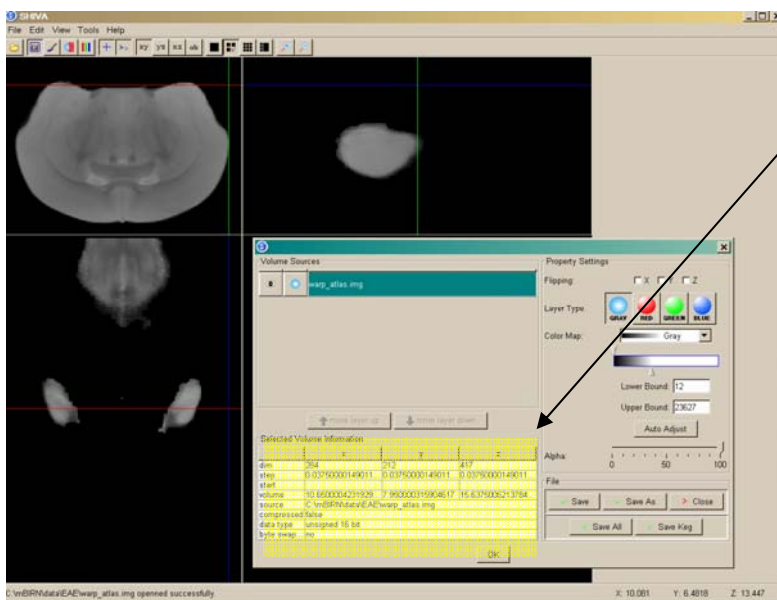
- The file *x-dimension* is defined as the dimension that changes most rapidly (i.e., each sequential voxel will fall in a different column and will therefore have a different x coordinate).
- The file *y-dimension* changes more slowly than the x-dimension and more quickly than the z-dimension.
- The file *z-dimension* is defined as the dimension that changes most slowly (e.g., all of the voxels for a given z-plane are stored before any of the voxels of the next z-plane).

## 2. READING THE PARAMETERS OF THE ANALYZE IMAGE HEADER FILE

There are many steps that require you understand the data in your file. In order to do this, you will need to be able to read your header file.

### a) SHIVA

1. SHIVA must be installed on your computer.
2. If you are using SHIVA on a unix machine, call SHIVA using a command line  
*java -jar shiva.jar*
3. If you are using SHIVA on a PC or Mac, open it by double clicking on the SHIVA.jar file.
4. Open the file in SHIVA, you can do this from the menu once SHIVA is open
5. From the tools menu or the toolbar select "Volume management"
6. Select the file from the list of files
7. The header information for will be displayed in the Volume information section (see figure).



## B. CONVERTING COMMONLY USED FILE TYPES TO ANALYZE IMAGE

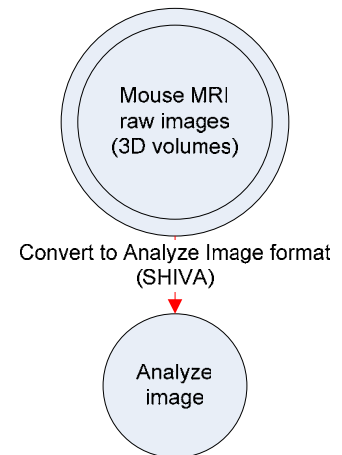
Most processing steps for mouse MRI data require the Analyze image format.

### 1. 2DSEQ AND RECO

- You must have SHIVA installed on your computer
- Open your file within SHIVA
- Save your file as a Data volume

### 2. CIVM AND MINC

- You must have SHIVA installed on your computer
- The header file and the individual images should be in the same directory
- Open the headfile within SHIVA
- Save your file as a Data volume



## C. REORIENTING IMAGE FILES

For the registration process to work its best, the scans need to be aligned in the same orientation within 45 degrees of each other.

- Open your files in SHIVA
- Open the Reorient Tool from the tool menu
- If necessary flip the axis or axes necessary to reorient your file, make sure all the images are within 45 degrees of each other
- Save them in this configuration

## D. RESAMPLING IMAGE FILES

If data files exceed 500 MB, the files will need to be downsampled to a size of 512 x 256 x 256 in order for skull stripping to succeed using BrainSuite 2.

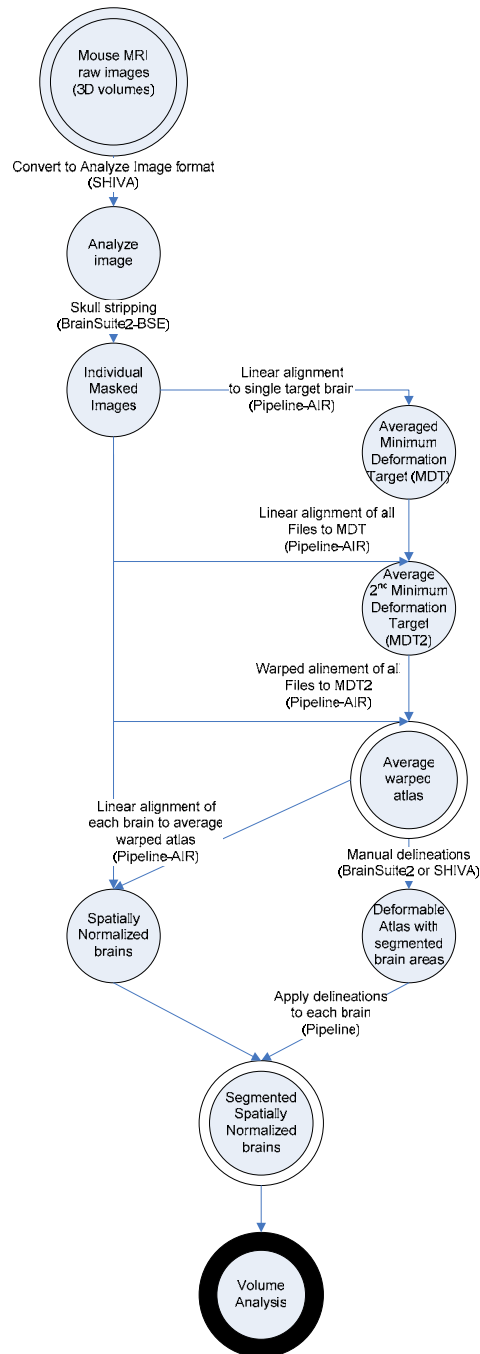
- This may be done manually using a command line program called manual reslice
- This may also be done using the program ImageJ

If you wish to use the higher image resolution for further data analysis, you will need to do the following:

- Run most of your analyses using the lower resolution image data
- Upsample both the image data as well as their masks
- Apply the high resolution masks to their respective images

## II. MOUSE BRAIN MRI IMAGES

Several different types of image data from the mouse brain can be processed and registered to a similar space (atlasing the image data), which allows an investigator to examine several different types of information from a common space and potentially a single interface. The processing steps discussed in this tutorial are with this goal of putting mouse MRI images into this space in order to allow for this type of analysis. The figure to the right shows the steps needed to convert MRI image into data that can be atlased and analyzed for volume information.



# A. GETTING STARTED

Before processing MRI data, you need to make sure your data are in place and the proper tools are installed on your system (this list is for the PC and Mac).

Install the following software tools:

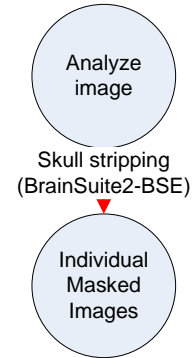
- BrainSuite 2 (for Windows only)
  - Download this software at [http://www.loni.ucla.edu/Software/Software\\_Detail.jsp?software\\_id=19](http://www.loni.ucla.edu/Software/Software_Detail.jsp?software_id=19)
  - Download and install this software into your preferred directory
  - This will create an application named BrainSuite 2 on your PC
  
- Java
  - Both Pipeline and SHIVA require JAVA version 1.4.2 or greater
  - Download this software at <http://java.sun.com/j2se/1.4.2/download.html>
  
- Pipeline Version 1
  - Download this software at [http://www.loni.ucla.edu/Software/Software\\_Detail.jsp?software\\_id=2](http://www.loni.ucla.edu/Software/Software_Detail.jsp?software_id=2)
  - The LONI Pipeline is distributed as a single Java jar program, pipeline.jar.
  - Create the directory, C:\mouseBIRN\Pipeline and place the jar file in that directory
  - To run the program, double click the pipeline.jar file.
  - If you wish to have the program on your desktop, create a shortcut that points to the C:\mouseBIRN\Pipeline\pipeline.jar file
  - For advanced installing options see [http://www.loni.ucla.edu/Software/Installing\\_Detail.jsp?software\\_id=2](http://www.loni.ucla.edu/Software/Installing_Detail.jsp?software_id=2)
  - If running the program in Client/Server mode, you are allowed access to LONI modules, pipelets, and pipelines on the LONI server. However, for this mode you will need a LONI Pipeline account, to get this account apply for an account using Pipeline
  
- SHIVA
  - Download this software at [http://www.loni.ucla.edu/Software/Software\\_Detail.jsp?software\\_id=12](http://www.loni.ucla.edu/Software/Software_Detail.jsp?software_id=12)
  - Create the directory, C:\mouseBIRN\SHIVA and place the jar file in that directory
  - To run the program, double click the SHIVA.jar file.
  - If you wish to have the program on your desktop, create a shortcut that points to the C:\mouseBIRN\SHIVA\shiva.jar file

For hands on practice of this whole processing step, download the data set from the SRB at /BIRN/mouse/Sharing/Tutorials/MRI/AHM session to your computer in C:\mouseBIRN\data

## B. PROCESSING OF MOUSE MRI

### 1. IMAGE PROCESSING PRIOR TO ATLASING

The image of the brain should be as clean as possible. Removing the skull from the image by masking and adjusting for spatial variations in the image using a bias field correction both facilitate the process of atlasing the mouse brain.



#### a) Skull stripping

Masking is the process of selecting a portion/region of the raw MRI scan by creating a binary file that covers the region of interest. Here we discuss how to we use the masking process to remove the skull from the image.

*Image type:*

Analyze Image format

*Required preprocessing steps:*

In order to use BrainSuite 2, image data cannot be larger than 512 x 256 x 256. If necessary, resample your image files (see section I.D).

*General tips before beginning:*

A good approximation of a stripped skull can be created semi-automatically using BSE (run within BrainSuite 2). You will then want to check and clean up the image volume manually using BrainSuite 2. See mask.avi on <http://www.loni.ucla.edu/twiki/bin/view/MouseBIRN/MouseBIRNTools> for a demonstration of this process.

Keep your masks consistent across brains by defining the edges of your brain before you begin.

*Processing steps:*

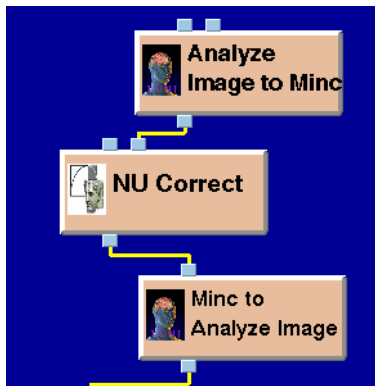
1. You must have BrainSuite 2 installed on your PC
2. Open the Analyze image data volume within BrainSuite
3. On the BrainSuite 2 toolbar, select the BSE option
4. Click Do All. When it is finished, close the BSE window.
5. If you wish, you can change some of the values in the BSE window to see if your mask improves
6. This will generate a mask file that will mask out most of the non-neural tissue. You will want to save this mask separate from your original file.
7. Click File, Save, Mask, and name your mask file (*original\_name.mask.img*)
8. Further editing will be require manual manipulation of your mask file, so make sure both the original image and the mask file are open in BrainSuite 2.
9. Open the Mask tool and click *Apply edits to mask*. The Mask tool must be kept open while editing the mask, and you may wish to change the brush size frequently, so often it's handy to keep that open as well.

10. Once the skull has been automatically stripped, you have the option of dilating and eroding (either in a diamond or cube format) this mask using the buttons in the Mask tool.
11. Since you will need to go through every slice of the brain, you may choose to edit all the images in one plane and then check them in another.
12. For additional information on manually editing this mask, see [cleaningamask.doc](http://www.loni.ucla.edu/twiki/bin/view/MouseBIRN/MouseBIRNTools) on <http://www.loni.ucla.edu/twiki/bin/view/MouseBIRN/MouseBIRNTools> for a demonstration of this process.
13. Save your file often as you are editing

## b) Bias field correction

Nu\_correct will run a RF correction on a MINC File format, in order to do this, the image must be converted to MINC file format and converted back after the correction.

You can run this program using the Pipeline module NU Correct.



Key Variables: *(Suggestions in parenthesis)*

Analyze Image to MINC:

In put: Analyze image volume (*input\_file.img*)

Output: MINC image volume (*output\_file.mnc*)

NU Correct:

Input: MINC image volume (*input\_file.mnc*)

Output: RF\_corrected\_MINC\_image (*output\_file.mnc*)

MINC to Analyze Image:

Input: MINC image volume (*output\_file.mnc*)

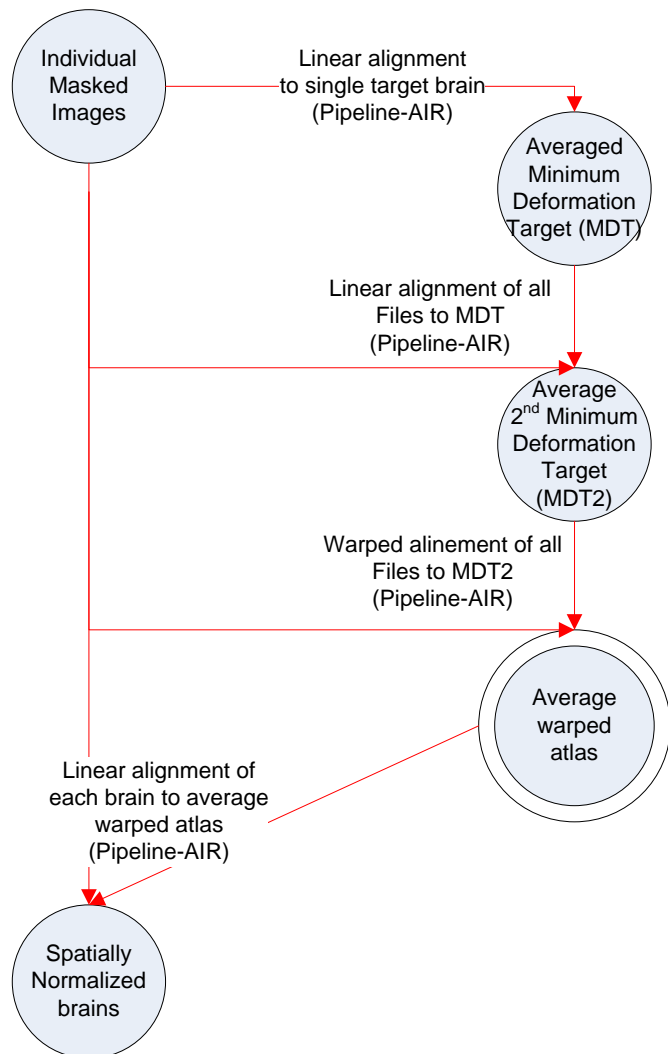
Output: Analyze image volume (*output\_file.img*)

## 2. GENERATING A MINIMUM DEFORMATION ATLAS

There is no single representative brain, nor a simple method of generating an “average” anatomy to represent 3D anatomic variations, let alone variations across strains, genetically manipulated animals, and disease states. Probabilistic atlasing is a research strategy for generating anatomical templates, expert diagnostic systems, and knowledge-based imaging tools that retain quantitative information on intersubject variations in brain architecture. Intensity-based approaches concentrate on generating “average” representations of anatomy by intensity averaging multiple MRI scans.

A fundamental problem for brain-mapping studies, when it is necessary to integrate data from many different subjects, is that there are significant anatomic variations in the size, shape, and position of neuroanatomical structures. Registration is the image-processing tool used in brain-mapping research to reduce inter-individual anatomic variance by matching homologous spatial features of a “source brain” to those of a “target brain.”

Registration is the process of aligning an image to another image. You will be using the LONI Pipeline tool to call the AIR modules in the AIR package to register mouse MRI images. You may register two images, or you can register a group of images to a single image, or create an average image. This registration process may involve a linear alignment, which scales the size and orientation of the images and places them in the same coordinate frame (shown in the first step) or a nonlinear alignment, which actually warps the images into a common space (shown in the second step).



*Image type:*

Analyze image format

*Required preprocessing steps:*

Skull stripping

Bias Field Correction

### *General tips before beginning:*

To use Pipeline properly, you will need to give the modules the names and directories of your files and often some options. To learn more detailed information than what's discussed here, see the Pipeline tutorial and the specific help on each module.

The AIR package includes two programs for automated registration of Analyze image files.

#### 1. **Alignlinear**

- This module includes 2D and 3D variations of all linear spatial transformation models. It generates an .air file that contains the linear transformation parameters that when applied, can be used to resample one of the images to match the other.
- The AIR module **reslice** must be used after alignlinear to apply the transformation parameters. It resamples the reslice file to match the standard file for a linear alignment.

#### 2. **Align\_warp**

- This module includes 2D and 3D variants of nonlinear polynomial spatial transformation models. It generates a .warp file that contains nonlinear transformation parameters that when applied, can be used to resample one of the images to match the other.
- **Reslice\_warp** must be used after alignwarp to apply the transformation parameters a nonlinear alignment.

In many instances, the procedure that provides the closest fit involves first running a linear alignment (either once or twice) followed by a nonlinear alignment. It should be noted, that while you can reverse the results of a linear alignment, you cannot do the same for a nonlinear alignment.

For these modules, you need to specify which image will be your standard (your atlas or the image to which you want to align the other file) and which will be the reslice image (the file which is resampled to match the standard). It is also possible to align multiple images to your standard with a single pipeline.

To maximize the effectiveness of mouse brain alignments, first use the “crop” module to remove empty planes around the edges of the image. After being cropped, these images get passed to alignlinear, which creates an .air file that contains the linear transformation parameters required to align the file (or files) with the standard.

In general, you can use a list of files instead of single file names to run the same process on each file in the list. Instead of passing the individual file name to the Pipeline, you simply pass the list file. To begin, make a text file (name it something that makes sense to you) that holds a list of your .img files with the directory. An example for the PC is shown to the right, however the same concept holds for the UNIX or the Mac.

*Contents of “input\_files.list”:*

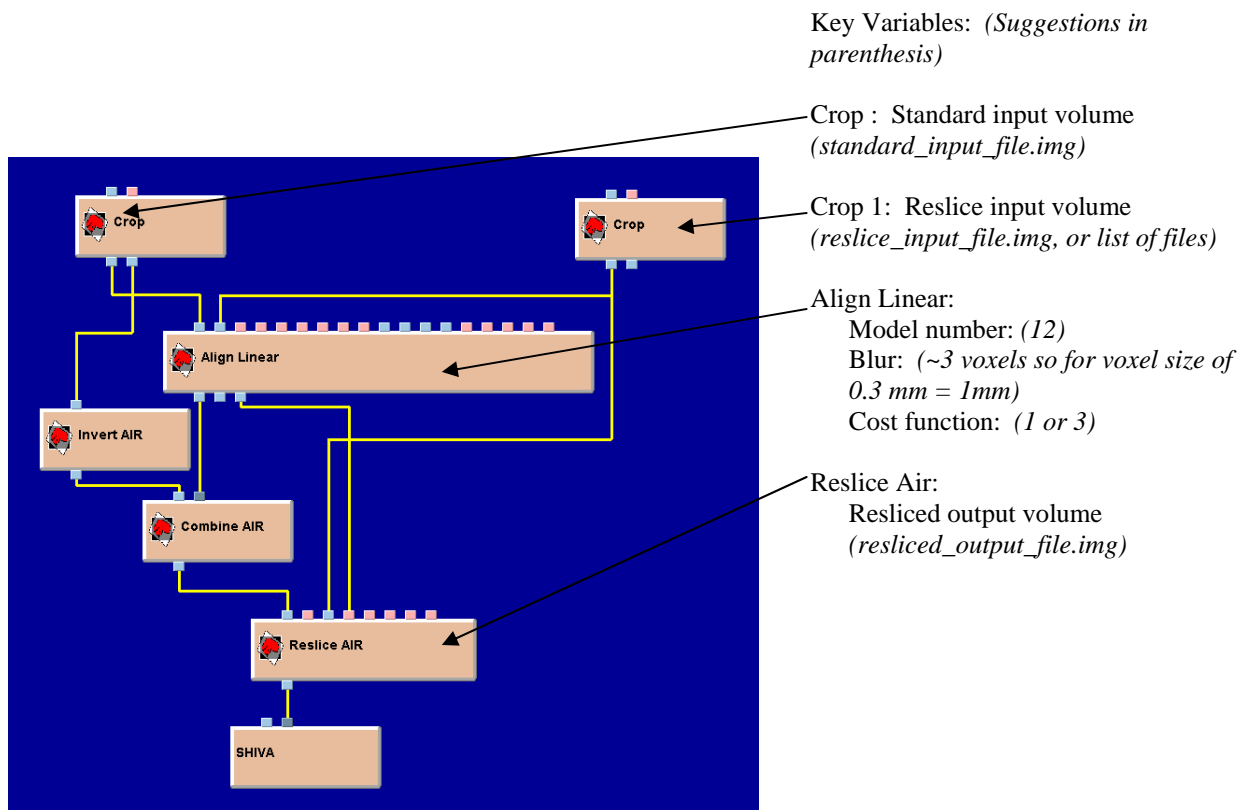
```
C:\mBIRN\data\MRI_001.img  
C:\mBIRN\data\MRI_002.img  
C:\mBIRN\data\MRI_003.img  
C:\mBIRN\data\MRI_004.img
```

## **a) Linear alignment of two files**

### *Processing steps:*

1. Crop your standard and reslice files
2. Send them to the alignlinear module to generate an .air transformation file. The cost function argument should be assigned 3 if you are trying to align similar images, but 1 if they are a different modality.

3. After the files are aligned, resample the reslice file to the space of the original standard file.
4. Since the standard file has been cropped, you need to invert the crop transformation to put your standard file back into its original space. This can be accomplished by running the invertAIR module on the standard file, which will create an .air transformation file that holds these parameters.
5. Run the combine\_AIR module with the aligned transformation and the “uncropped” standard transformation to generate an .air transformation file that allows for placing the aligned reslice image into the standard space.
6. Use this .air file with the module resliceAIR on the reslice image file to apply these transformations to the image file and create a new realigned file.
7. The SHIVA module at the end will allow you to visualize these files, and you can check their alignment, or you can manually open the file within SHIVA or another visualization program to check the registration.



### a) Nonlinear alignment of two files

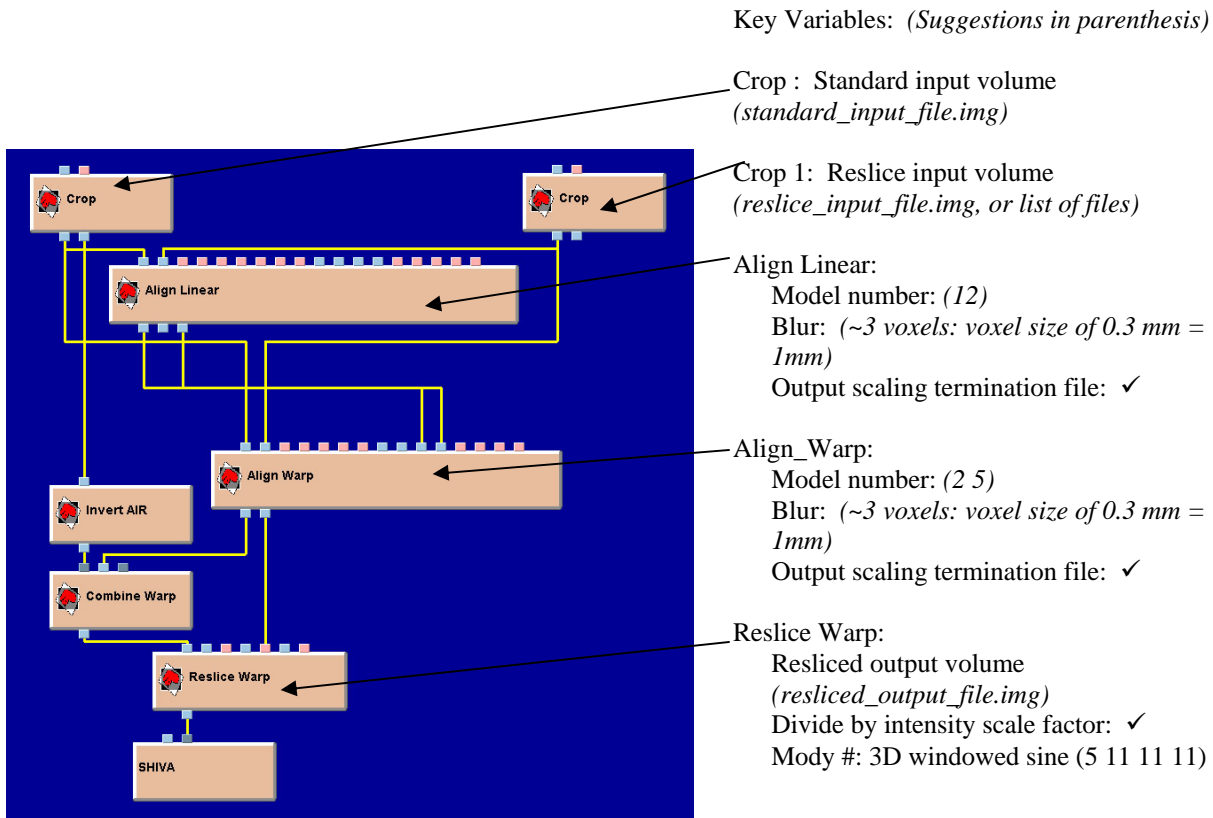
This process is slightly different from the linear alignment of two files, but basically it follows the same idea.

*Processing steps:*

1. Send the cropped files are sent to alignlinear
2. The parameters from this module are sent to the alignwarp module (.air transformation file and the scaling termination file, which specify the transformation initialization and scaling

initialization for the alignwarp module). The alignwarp module uses these parameters as the starting point before beginning a nonlinear alignment of the two volumes.

3. Again, the idea is to register the reslice volume to the standard volume, so invert\_air reverses the crop and puts the standard volume back into standard space.
4. Combining the output of alignwarp with the original generates a .warp transformation file that allows for placing the aligned reslice image into the standard space.
5. Using this .warp file with the module reslicewarp on the reslice image file will now apply these transformations to the image file and creates a new realigned file.
6. The resulting image files can be sent to SHIVA to visualize the registration.

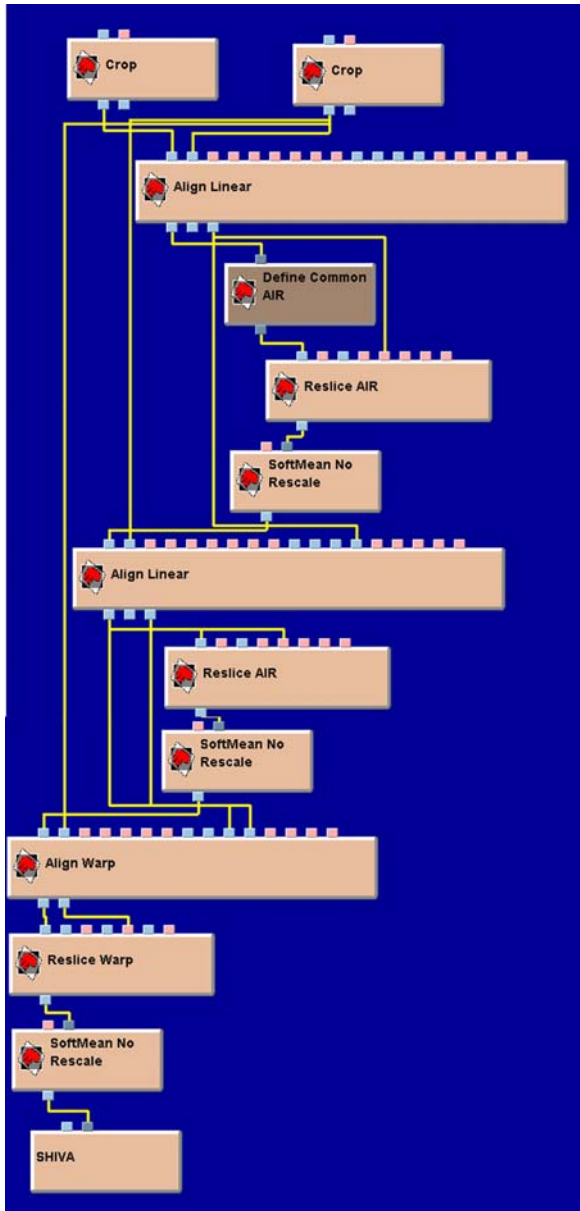


## b) Alignment of multiple files

This process is slightly different from single files. With a group of data files from several subjects, you more than likely want to align them to a standard space and average them. Also, you may want to compare this average to an atlas. If set up properly, the pipeline will run the modules on each file in order and then you can create an average from these transformations. For the best possible registration, you will probably want to first perform a linear alignment of all the files to an arbitrary one and create an average from these alignments. Then, do a second linear alignment of your files, only this time to the first average and create a new average from this second alignment. Finally, you will want to do a nonlinear alignment to compare your average to a standard.

*Processing steps:*

1. Create a list file (text) that holds a list of your .img files including the path. An example is shown at the beginning of this section.
2. Your input to one of the crops will be your first “standard” image and the input to the other crop will be the list of the rest of the files.
3. Then send standard and reslice files to the alignlinear module to generate .air transformation files for each reslice file.
4. The output of these transformations then gets sent to define\_common\_AIR which will find the average shape, size, and position of the reslice files and create a new standard based on this average, then generates new .air transformation files for each of the reslice files based on this new standard.
5. The reslice files now get resliced to the MDT (*min\_def\_target.img*) in reslice\_AIR.
  - a. Note that reslice\_AIR is sent something from the alignlinear module, this is an intensity scaling parameter file that contains the intensity scaling parameter identified as optimal by the algorithm.
  - b. This can be combined with a spatial transformation initialization file to restart the algorithm at the same location in parameter space where it left off.
  - c. Reslice can use this information to create a final image that is intensity corrected as well as spatially corrected.
  - d. In addition, the scaling parameter can be used as an intensity normalization factor for subsequent statistical analysis of the registered data.
6. At this point running the module soft mean no rescale creates an average of the resliced files.
7. This new average MDT2 (*min\_def\_target2.img*) will now be used as the standard for the reslice files in a second linear alignment, again the files will get resliced and averaged.
8. Finally this second average will be used as the standard to which the reslice files are nonlinearly aligned. The files are resliced to match this average and averaged again to create a final warped average of all the files MDA (*min\_def\_atlas.img*).
9. SHIVA can be used to visualize this final average file.



Key Variables: (Suggestions in parenthesis)

Crop : Standard input volume (*standard\_input\_file.img*)

Crop 1: Reslice input volume (*reslice\_input\_file.img*)

Align Linear 1:

Model number: (12)

Blur: (~9 voxels: voxel size of 0.3 mm = 3mm)

Output scaling termination file: ✓

Reslice\_Air 1:

Divide by intensity scale factor: ✓

Mody #: 10 (chirpz)

SoftMean No Rescale 1:

Average volume (*min\_def\_target.img MDT*)

Align Linear 2:

Model number: (12)

Blur: (~9 voxels: voxel size of 0.3 mm = 3mm)

Scaling initialization file: ✓

Output scaling termination file: ✓

Reslice\_Air 2:

Divide by intensity scale factor: ✓

Mody #: 10 (chirpz)

SoftMean No Rescale 2:

Average volume (*min\_def\_target2.img MDT2*)

Align\_Warp:

Model number: (2 5)

Blur: (~2 voxels: voxel size of 0.3 mm = 1mm)

Output scaling termination file: ✓

Reslice\_Warp:

Divide by intensity scale factor: ✓

Model: 3D windowed sine (5 11 11 11)

SoftMean No Rescale 2:

Resliced output volume (*min\_def\_atlas.img MDA*)

This final image file is basically an averaged atlas and you may wish to compare this atlas to some other file or atlas. In order to put these files in the same space as each other, use the linear alignment of two files as discussed in that section.

## C. LABELING BRAIN AREAS

You may wish to label certain brain areas in order to perform further analyses. This is done by painting labels over your MRI images and saving these labels as a separate, but related file. See label.avi on <http://www.loni.ucla.edu/twiki/bin/view/MouseBIRN/MouseBIRNTools> for a demonstration of this process.

*Image type:*

Analyze Image format

*Suggested preprocessing steps:*

Skull Stripping

Bias field correction

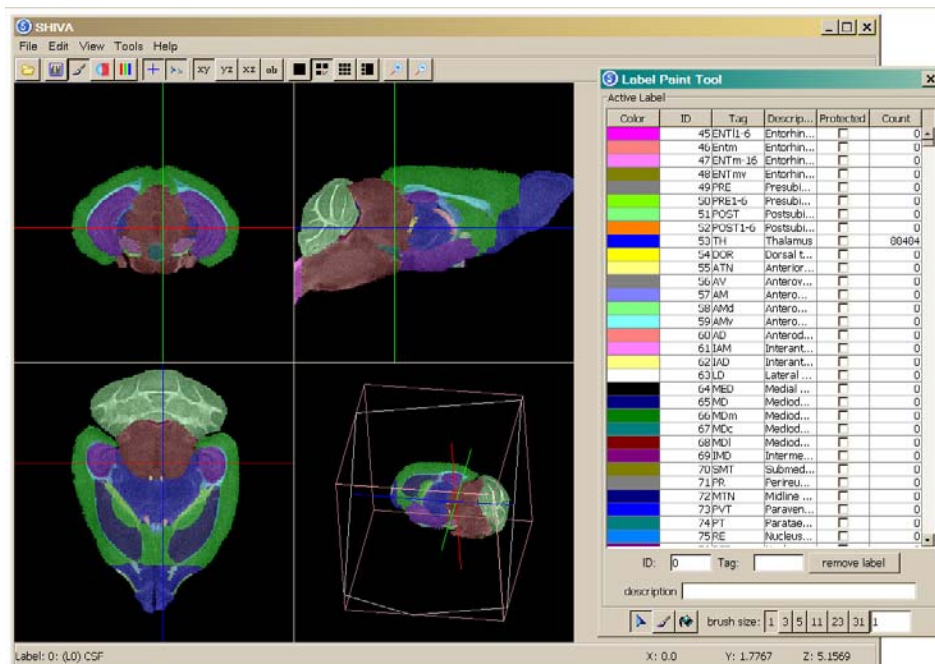
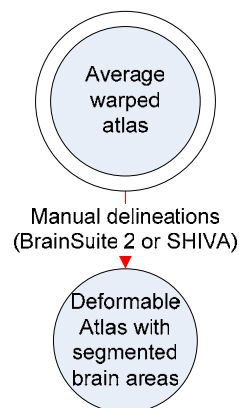
Generation of a minimum deformable atlas

*Processing steps:*

In order to segment the brain into separate brain areas, you will need to manually draw the outlines on the data volume. This can be accomplished using BrainSuite 2.

1. Open your data volume in SHIVA or BrainSuite 2.
2. Open the Label Painter and select the “Edit Labels” box on the label painter.
3. See the BrainSuite 2 tutorial for information on how to draw these labels.

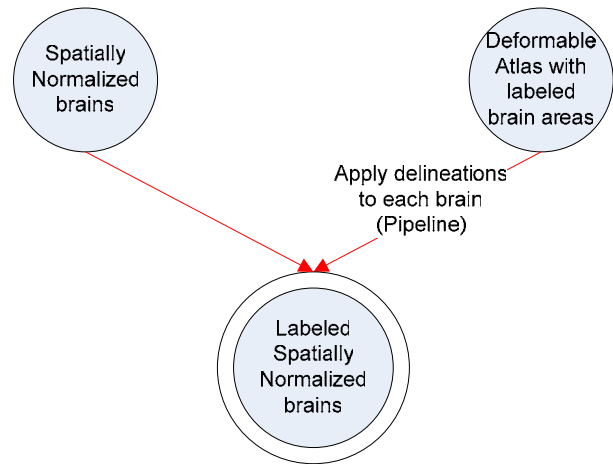
Once the areas have been drawn, it is straightforward to count the number of voxels in each label. Knowing the size of the pixels, you can easily compute the volume of this are from this information.



D.

# APPLYING DEFORMABLE ATLASES

After you have a segment atlas, you may wish to apply these delineations to other MRI images. This is done by warping the delineated atlas onto each MRI image.



## Image type:

Analyze Image format

## Suggested preprocessing steps:

- Skull Stripping
- Manual correction of masks
- Bias field correction
- Generate an MDA
- Segment deformable atlas

## Processing steps:

## E. VOLUME ANALYSIS

After you have segmented your data using SHIVA or BrainSuite 2, it will automatically tell you how many voxels are in each delimited area. These values can be recorded and compared to the volumes in another atlas. In order to make valid comparisons, the brains should be registered to each other, which in essence should “normalize” the atlases.

1. Open your label volume in SHIVA or BrainSuite 2.
2. Open the Label Painter tool
3. The number of voxels in each area are listed in the “count” column of the Label Painter tool
4. The volumes of the labeled areas are listed in the “volume” column of the label painter tool

