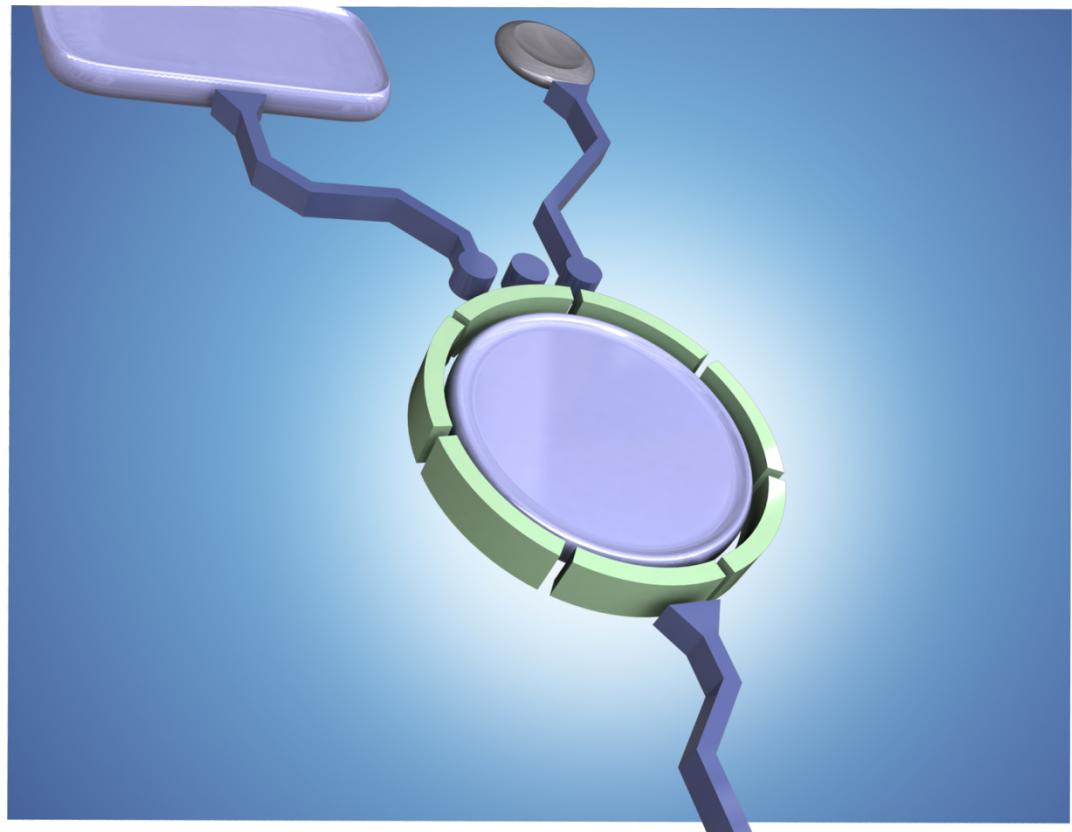


# LONI Pipeline Neuroimaging Solutions

A practical, hands-on guide to using the LONI Pipeline for neuroimaging analysis



LONI Pipeline version 5.4.3



LONI Pipeline training is funded by the National Institutes of Health  
through the NIH Roadmap for Medical Research  
Grant U54 RR021813 and P41 RR013642.

# Table of Contents

<b>Getting Started</b>	<b>1</b>
<b>Overview: About LONI Pipeline</b>	<b>1</b>
<b>Benefits</b>	<b>1</b>
<b>Requirements</b>	<b>1</b>
<b>Installation</b>	<b>2</b>
<b>Software Download</b>	<b>2</b>
<b>LONI Cluster Usage Account</b>	<b>2</b>
<b>Platform-Specific Installation</b>	<b>2</b>
<b>OS X</b>	<b>2</b>
<b>Windows</b>	<b>3</b>
<b>Linux/Unix</b>	<b>3</b>
<b>Interface</b>	<b>4</b>
<b>Overview</b>	<b>4</b>
<b>Data Sources &amp; Sinks</b>	<b>6</b>
<b>Overview</b>	<b>6</b>
<b>How-to</b>	<b>6</b>
<b>Find &amp; Replace</b>	<b>7</b>
<b>LONI Pipeline Viewer</b>	<b>8</b>
<b>Overview</b>	<b>8</b>
<b>Beginning Example Workflows</b>	<b>9</b>

<b>Registration Using AIR</b>	<b>10</b>
<b>BrainSuite Processing</b>	<b>29</b>
<b>Automated Image Registration &amp; BrainSuite</b>	<b>41</b>
<b>MNC-BrainSuite Heterogeneous Workflow</b>	<b>53</b>
<b>Intermediate Topics</b>	<b>61</b>
<b>LONI Image Data Archive</b>	<b>62</b>
<b>Overview</b>	<b>62</b>
<b>Change Server for Entire Workflow</b>	<b>65</b>
<b>Overview</b>	<b>65</b>
<b>How-to</b>	<b>65</b>
<b>Intermediate Example Workflows</b>	<b>67</b>
<b>Skullstripping</b>	<b>68</b>
<b>fMRI First Level Analyses</b>	<b>80</b>
<b>DTI Analyses</b>	<b>97</b>
<b>DSI Analyses</b>	<b>103</b>
<b>Advanced Topics</b>	<b>111</b>
<b>Building a Module</b>	<b>112</b>
<b>Overview</b>	<b>112</b>
<b>How-to</b>	<b>112</b>
<b>Optional Actions</b>	<b>113</b>
<b>Required Options</b>	<b>114</b>

<b>Adding additional parameters</b>	<b>116</b>
<b>Specify dependencies and manipulate file names</b>	<b>117</b>
LONI Pipeline Provenance	119
<b>    Overview</b>	<b>119</b>
<b>    How-to</b>	<b>119</b>
Study Module	122
<b>    Overview</b>	<b>122</b>
<b>    How-to</b>	<b>122</b>
Conditional Module	125
<b>    Overview</b>	<b>125</b>
<b>    How-to</b>	<b>125</b>
<b>Distributed Pipeline Server Installation</b>	<b>129</b>
<b>Introduction</b>	<b>129</b>
<b>GUI Installation</b>	<b>129</b>
<b>Command Line Installation</b>	<b>140</b>
<b>Glossary</b>	<b>145</b>
<b>Bibliography</b>	<b>150</b>

# Getting Started

## Overview: About LONI Pipeline

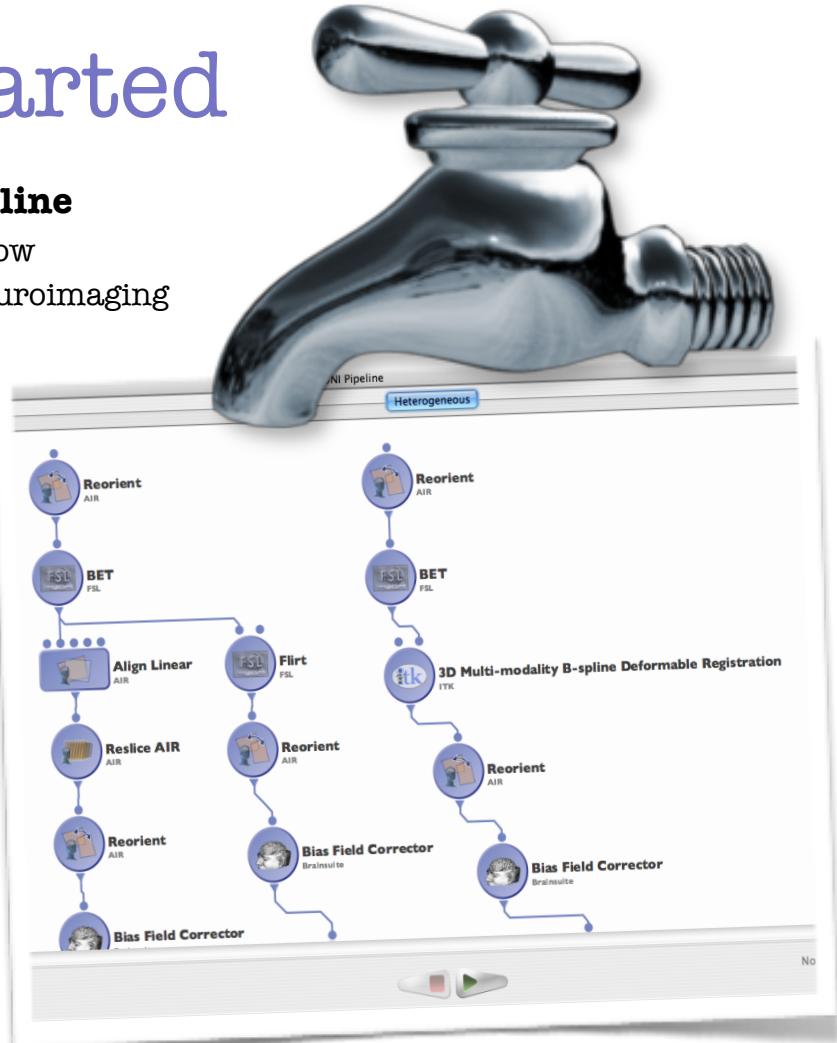
The LONI Pipeline is a free workflow application primarily aimed at neuroimaging researchers. With the LONI Pipeline, users can quickly create workflows that take advantage of any and all neuroimaging tools available. They can also gain access to the supercomputing cluster housed at LONI, free of charge, or install and use the LONI Pipeline entirely locally.

## Benefits

The LONI Pipeline is a workflow application that allows users to easily describe executables in a graphical user interface (ie. create a module) and connect them to create complex analyses, without having to code a single line in a scripting language. Instead of manually managing intermediate data in a script, the LONI Pipeline handles the passing of data between programs and stores it in a cache directory. Once you've created a module for use in the LONI Pipeline, you can save it into your personal library and reuse it in other workflows you create by simply dragging and dropping it in. The same is also true of entire workflows.

## Requirements

The LONI Pipeline is platform independent, thus the only computer requirement of the Pipeline client is an installation of JRE 1.5 or higher, which can be downloaded from Sun (<http://www.java.com>). In terms of memory consumption, it's unlikely that you'll need to worry about having sufficient RAM to run the Pipeline. However, you will need to ensure sufficient RAM is available if you choose to run the Pipeline from a virtual machine.



# Installation

## Software Download

To get the latest version of the LONI Pipeline, go to the Pipeline web site (<http://pipeline.loni.ucla.edu>) and click on the download link in the menu at the top.

If you do not already have a LONI User Account, you will be asked to create an account before you download.

## LONI Cluster Usage Account

When downloading the LONI Pipeline, you are given the option to connect to the server

running on LONI's supercomputing resources (available free of charge). By choosing this option, you are prompted to complete an online form. This form is then reviewed and you are notified of the status of your account through email.

**Please note that the approval process can take up to one working day.**

## Platform-Specific Installation

### OS X

To install the program, double click the disk image file you downloaded, and drag the LONI Pipeline application into the Applications folder. Once the program is done copying you can unmount (eject) the disk image and throw it in the trash. To start the Pipeline, just go to your Applications folder and double-click on the LONI Pipeline application.

The image contains three screenshots of the LONI Pipeline software interface:

- Screenshot 1: LONI PIPELINE DOWNLOAD REQUEST**  
This screenshot shows the "RETURNING USERS SIGN IN" section. It includes fields for "E-mail" and "Password", a "Forgot your password?" link, and a "SIGN-IN" button. Below this, the "NEW USERS" section is visible with a "Create a new account" link and a note about temporary passwords.
- Screenshot 2: ADDITIONAL OPTION: LONI CLUSTER AND SERVER LIBRARY**  
This screenshot shows a dialog box asking if the user has previously obtained credentials for accessing LONI servers. It includes a "Pipeline Account:" radio button group (set to "no") and an "Accept" button. A black arrow points from this screen down to the third screenshot.
- Screenshot 3: LONI PIPELINE ACCOUNT APPLICATION**  
This screenshot shows a detailed account application form. Fields include:
  - Principal Investigator Information:** First Name, Last Name, MI, E-mail Address, Phone, Fax.
  - Academic Position:** M.D., Ph.D., M.S., M.A., B.S., B.A., Faculty, Postdoc, Researcher, Other.
  - Insitution/Company:** UCLA.
  - Address:** Street Address, City, State, Zip/Post.
  - Project Information:** Project Title, a checkbox for AIDS research.All fields marked with an asterisk (\*) are mandatory.

## Windows

To install on Windows, double-click the installer and follow the on-screen instruction. Once it finishes installing, you can throw away the installer and launch the program by going to the Start menu->Programs->LONI Pipeline and start the program.

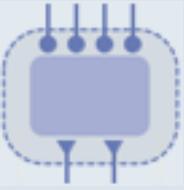
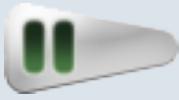
## Linux/Unix

Extract the contents of the file to a location on disk, and execute the PipelineGUI script. Make sure you have the java binary in your path.

# Interface

## Overview

The LONI Pipeline has an easy to use interface. Putting together workflows only requires knowledge of your tools and your goal, instead of programming languages and scripts. While your workflow is executing you can see exactly what step the Pipeline is currently on, and even see the output of a particular step when it finishes instead of waiting for the entire workflow to finish.

<b>Data Source</b> 	The data to be processed is input here. Double click to set the data path.	<b>Software Module Group</b> 	Software modules can be grouped together to simplify the look of your workflow.
<b>Play Button</b> 	Clicking play will initiate processing. The workflow will connect to the computer cluster and process the workflow.	<b>Pause</b> 	Clicking on Pause will temporarily pause the workflow. All running jobs/instances will be stopped and their output files will be deleted, but not the outputs of completed jobs.
<b>Stop</b> 	Clicking on Stop will permanently stop the execution of the workflow.	<b>Reset</b> 	Clicking on reset will reset the workflow and all temporary output files will be deleted.
<b>Data Sink</b> 	Workflow results will output to this module's destination. Double click to set the output path.	<b>Empty Output Parameter</b> 	Modules also have Output Parameters, this one is empty.

<b>Software Module</b> 	Each module holds a unique piece of software.	<b>Specified Output Parameter</b> 	After an Output Parameter is specified it will appear as a darker triangle.
<b>Empty Input Parameter</b> 	Modules often require Input Parameters. An empty parameter has no value specified.	<b>Connections</b> 	Lines that connect the output of one module to the input of another module.
<b>Input Parameter</b> 	A darker circle denotes an Input Parameter with a value specified.	<b>Smartlines</b> 	A Smartline connects two modules and will automatically perform the necessary file conversion on the output of the first module in order to ensure that the input to the second module is of the proper type.
<b>Optional Input Parameter</b> 	Occasionally, Input Parameter do not require arguments but offer them optionally.		
<b>Flow Control</b> 	Flow Control parameter is used to ensure the sequential execution of modules. Connecting flow control parameters between two modules enforces a strict ordering on the execution of these modules.	<b>Conditional Module</b> 	Conditional Module is used when the execution path of various inputs to a workflow is dependent on some criteria. Use of Conditional Modules makes the workflow more dynamic.
<b>Study Module</b> 	Study Module incorporates imaging data and non-imaging meta-data, and enables queries, groupings and construction of study designs based on user-specified criteria.	<b>Label</b> 	Software modules and module groupings have been labeled to describe the processing.

# Data Sources & Sinks

## Overview

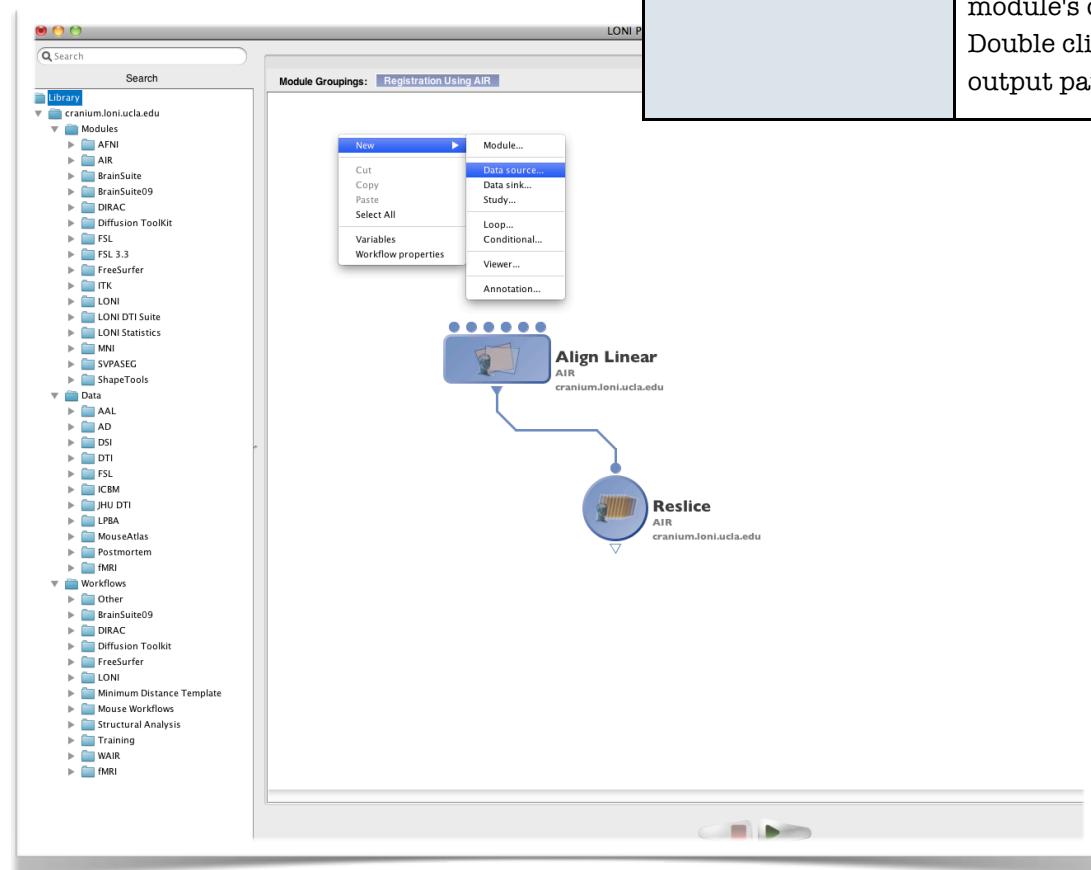
Data Sources and Sinks are a simple way to take advantage of parallelization of data processing. Users may specify all the files to be processed in much the same way one would specify all the subjects in a simple loop. What's more, the files can reside locally or remotely. Please note that in order to specify remote files, or to remotely browse for files, a Pipeline server must be running on the specified server.

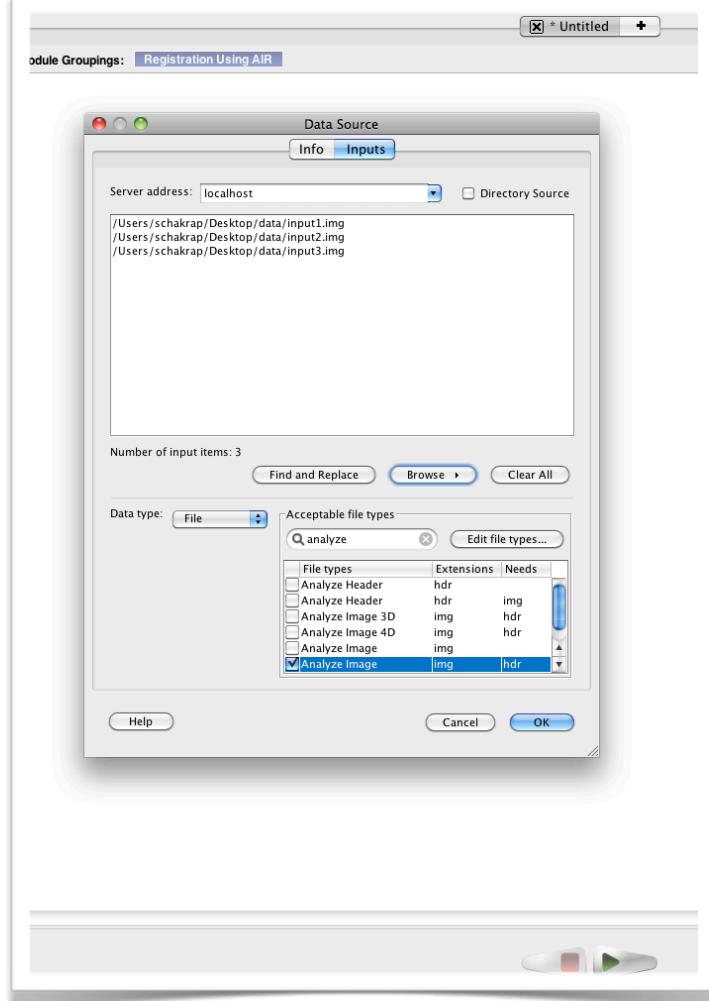
## How-to

Right click on the canvas

Choose new data source

<b>Data Source</b>	The data to be processed is input here. Double click to set the data path.
<b>Data Sink</b>	Workflow results will output to this module's destination. Double click to set the output path.





All fields under the Module's tab are optional, including Name, Package, Package version, Tags, Description, website and comments

Click the Inputs tab

In the Input Data field, choose localhost (for local files) or use the drop down to specify remote server

Choose Browse

Choose Local Files to browse for local files, or Remote files to browse for remote files

Browse for file, click open

Repeat for all input files

Choose Data type (Directory, File, Number, String, Enumerated)

For File, choose File Type

Click OK

Connect to input parameter

### Find & Replace

This feature is particularly useful for data sinks as it allows you to cut and paste input file names, and manipulate them, thus creating explicit outputs. This feature is similar to find and replace in any text editing program.

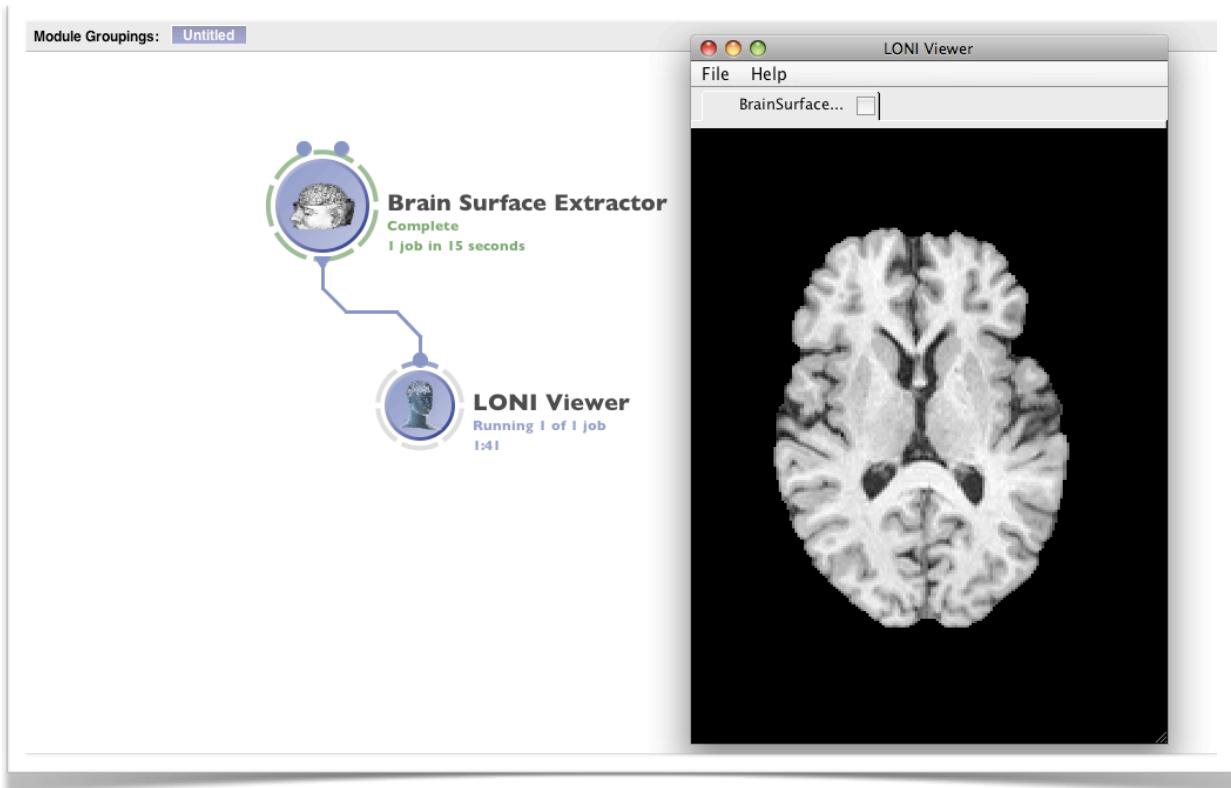
- Specify files to be processed using instructions above
- Choose Find and Replace
- In the Find field, specify string to search for (eg, .img)
- In the Replace field, specify string to replace the previous instance (eg \_SkullStripped.img)
- Choose Replace All

**NOTE:** The above process is applicable to data sources as well as data sinks, simply right click on the canvas and specify data sink.

# LONI Pipeline Viewer

## Overview

The LONI Pipeline now has a built-in visualization tool (LONI Viewer) that lets you view results as soon as the execution completes. Simply choose Insert -> New Viewer on the top menu or right click on the workflow canvas and select “New Viewer...”. You can give inputs to this viewer by connecting the input parameter of the viewer to the output of other modules. When the file is ready, a separate Viewer window will pop up with the file to display. You can exit the LONI Pipeline, the Viewer will keep running until you close it.



# Beginning Example Workflows

## **Registration Using AIR**

This is a simple registration workflow. The structural MRI images are aligned to a standard brain image by using AIR registration tools. To build this workflow follow the steps mentioned on the following pages.

## **BrainSuite Processing**

BrainSuite tools are used to perform common pre-processing steps.

## **AIR-BrainSuite Heterogeneous Workflow**

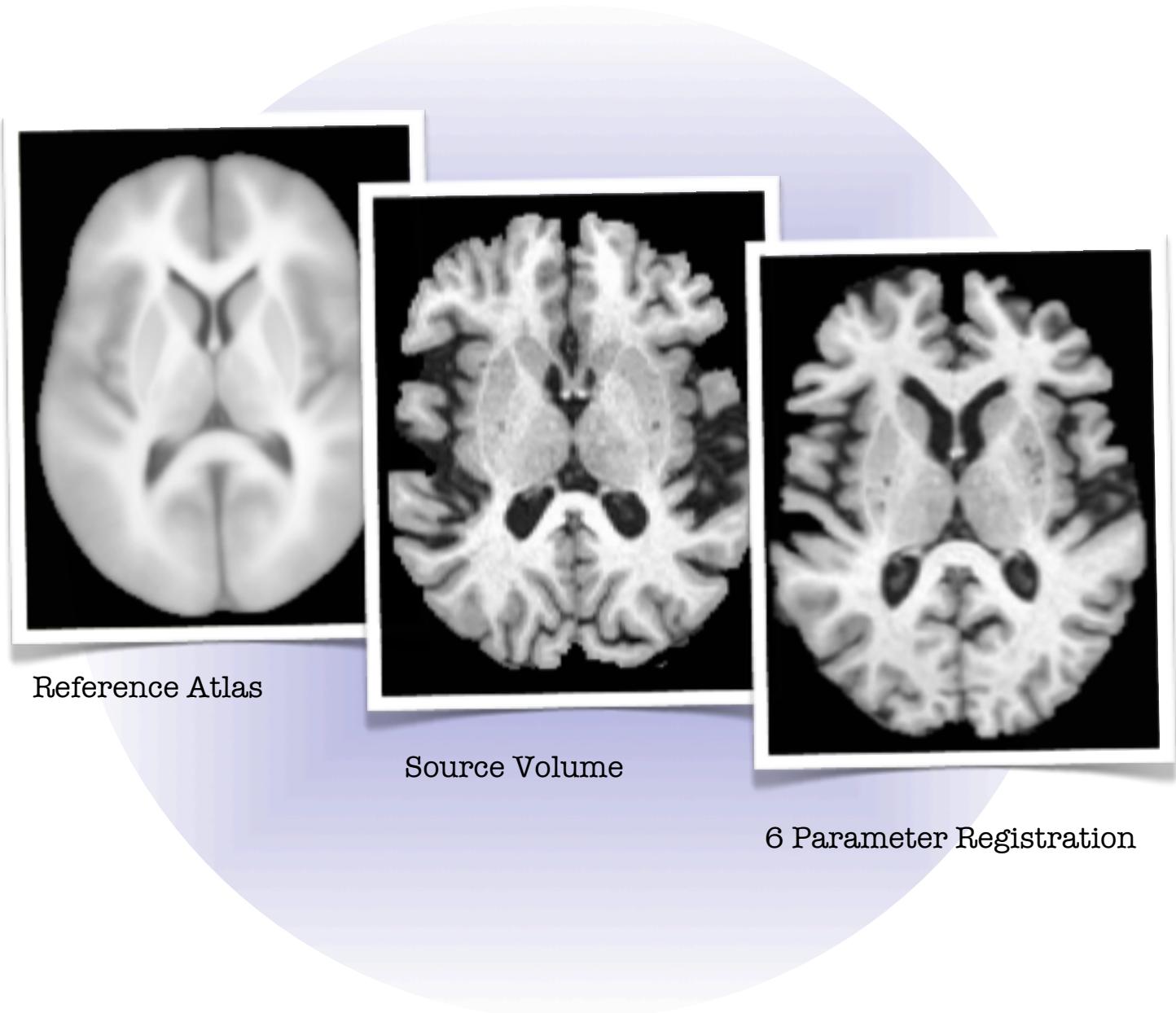
Tools from the AIR and BrainSuite packages are used to register and skull-strip a brain volume.

## **MNC-BrainSuite Heterogeneous Workflow**

Tools from the MNC and BrainSuite packages are used skull-strip a brain volume in MNC file format.

# Registration Using AIR

## Automated Image Registration

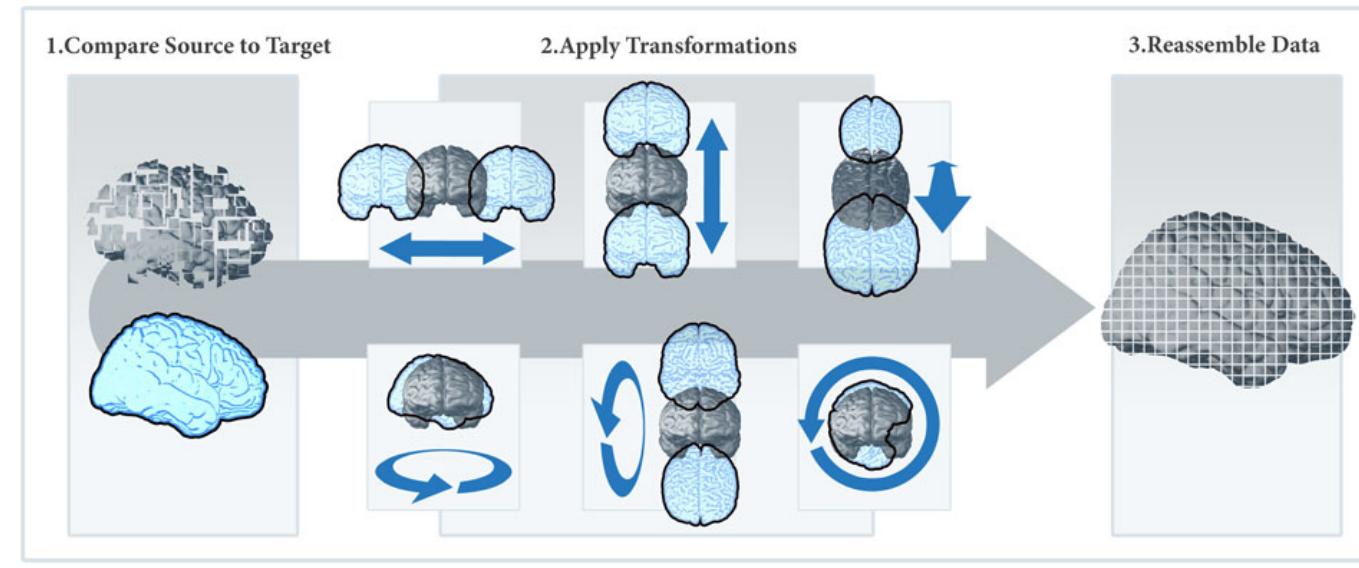


# Registration Using AIR

## Introduction

In this simple workflow, a registration algorithm generates a transformation matrix and applies it to move the source into the standard space.

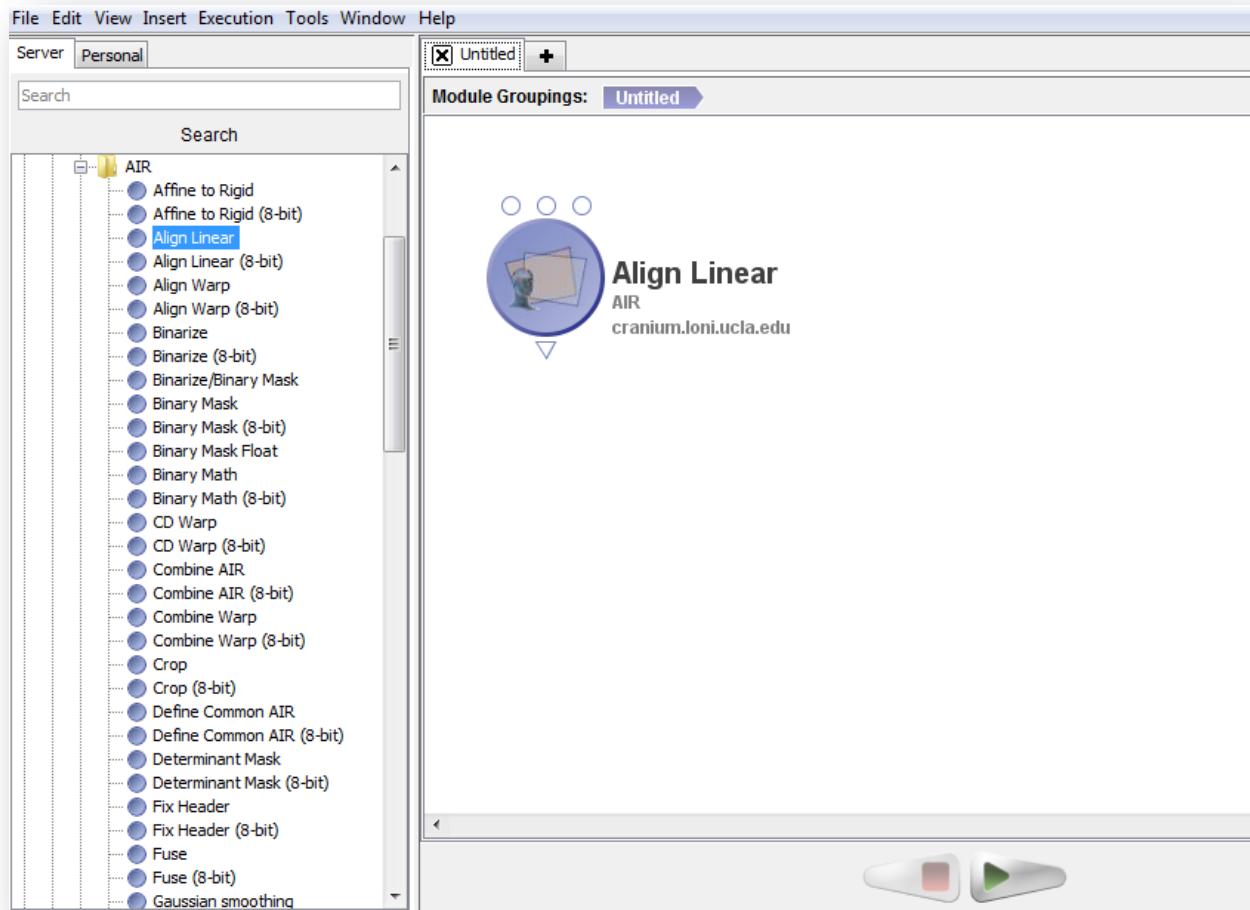
### Overview



This is a simple registration workflow. The structural MRI images are aligned to a standard brain image by using AIR registration tools. To build this workflow follow the steps mentioned on the following pages.

# Registration Using AIR

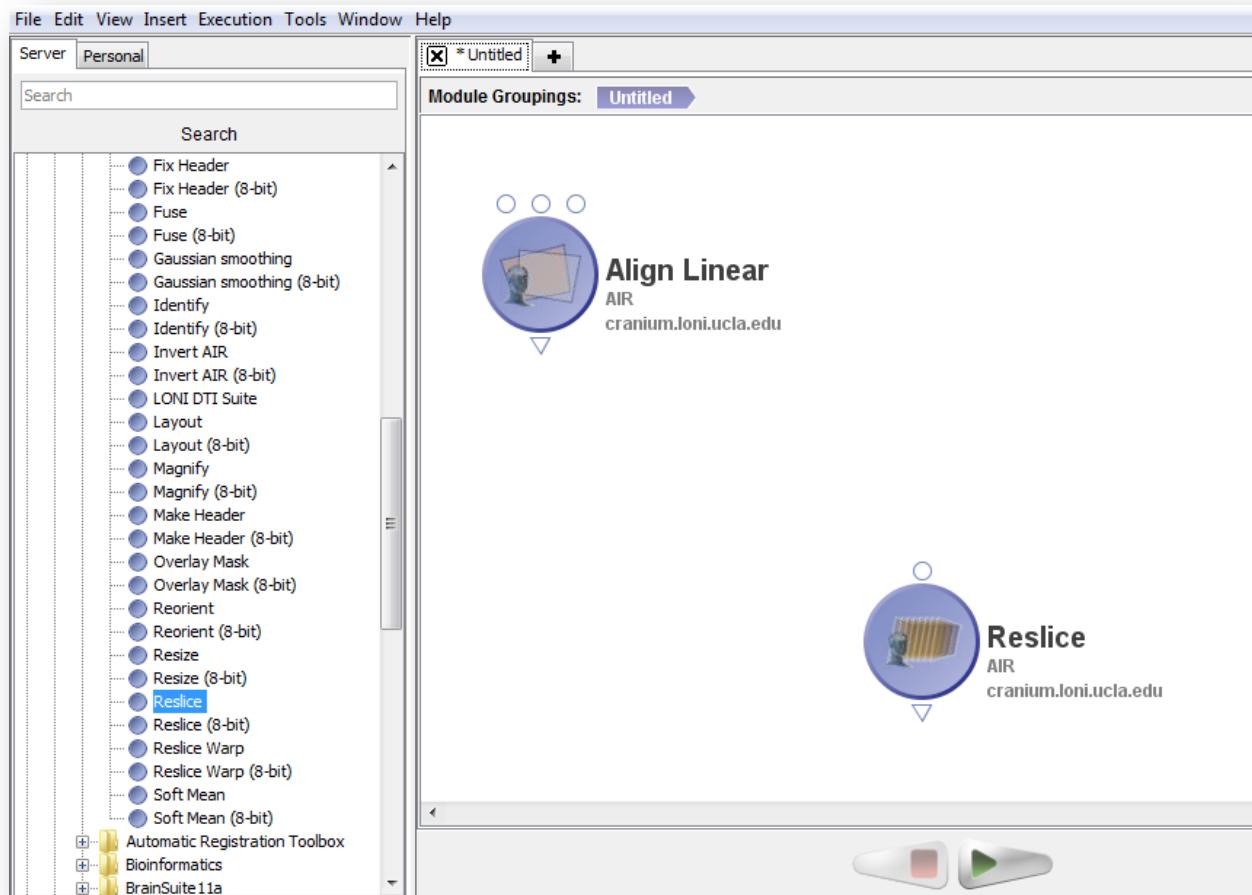
## Select Align Linear



- 1) On the LONI Pipeline window, open the AIR package on the left side panel. Drag the module labeled Align Linear onto the canvas within the larger right panel of the Pipeline window.

# Registration Using AIR

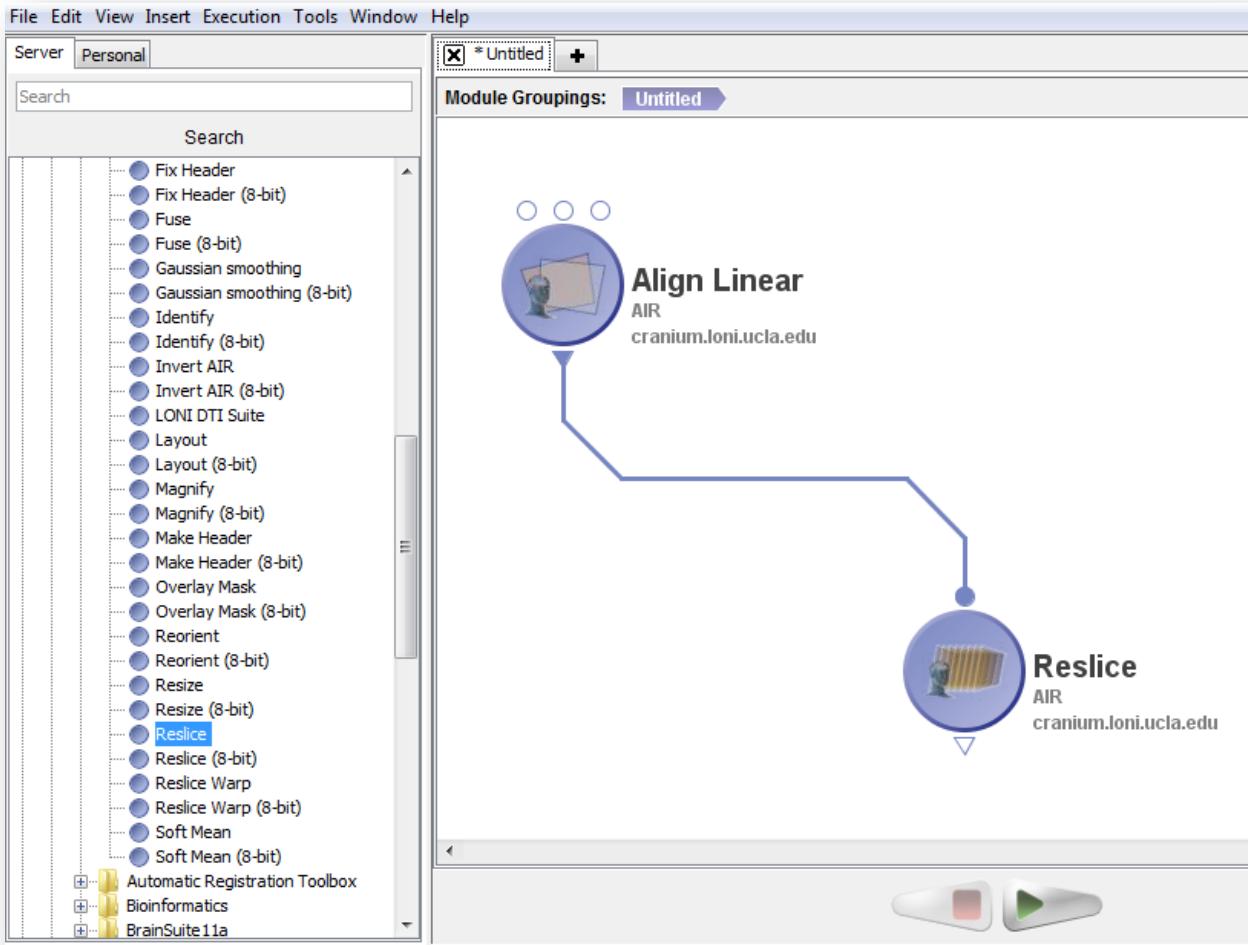
## Select Reslice



- 2) In much the same way we added Align Linear to the canvas, drag across the Reslice module so that it is positioned below Align Linear.

# Registration Using AIR

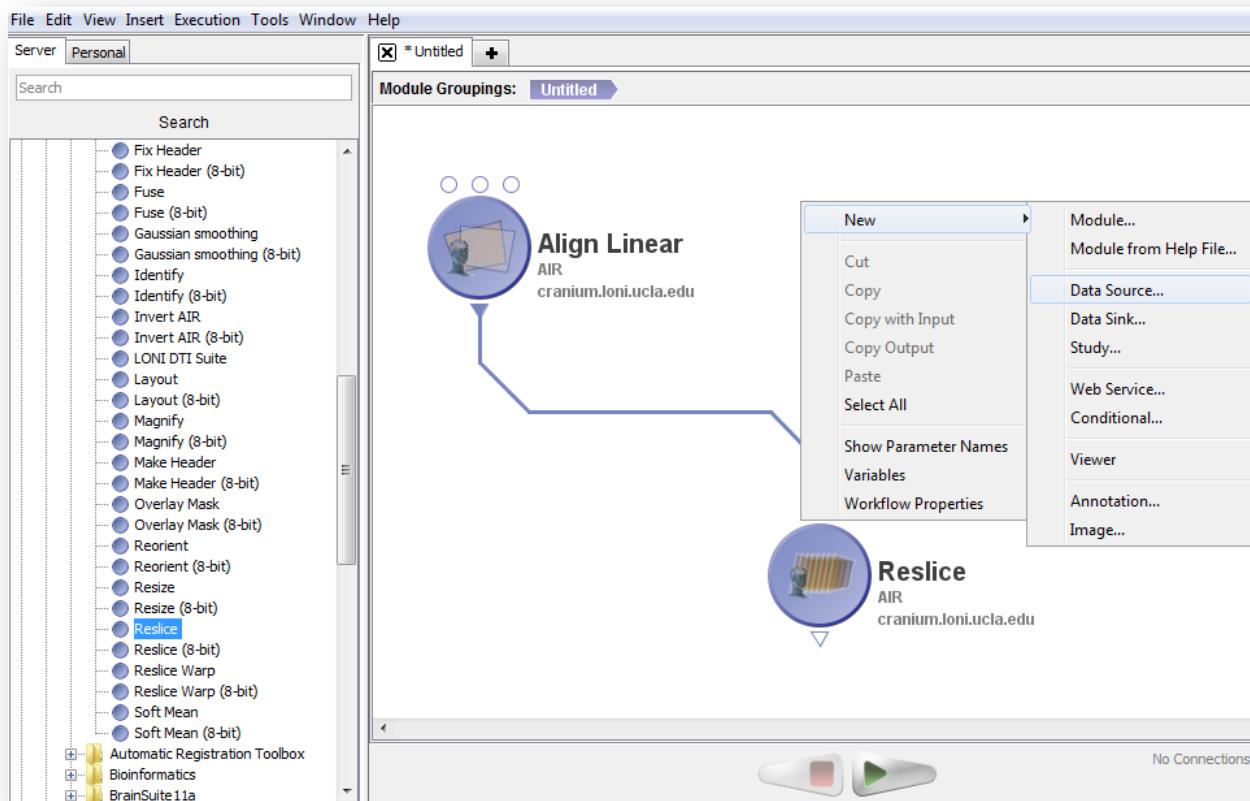
## Connect Align Linear to Reslice



- 3) Connect the two modules by left-clicking on the output triangle of Align Linear and dragging the connecting line to the input circle of Reslice.

# Registration Using AIR

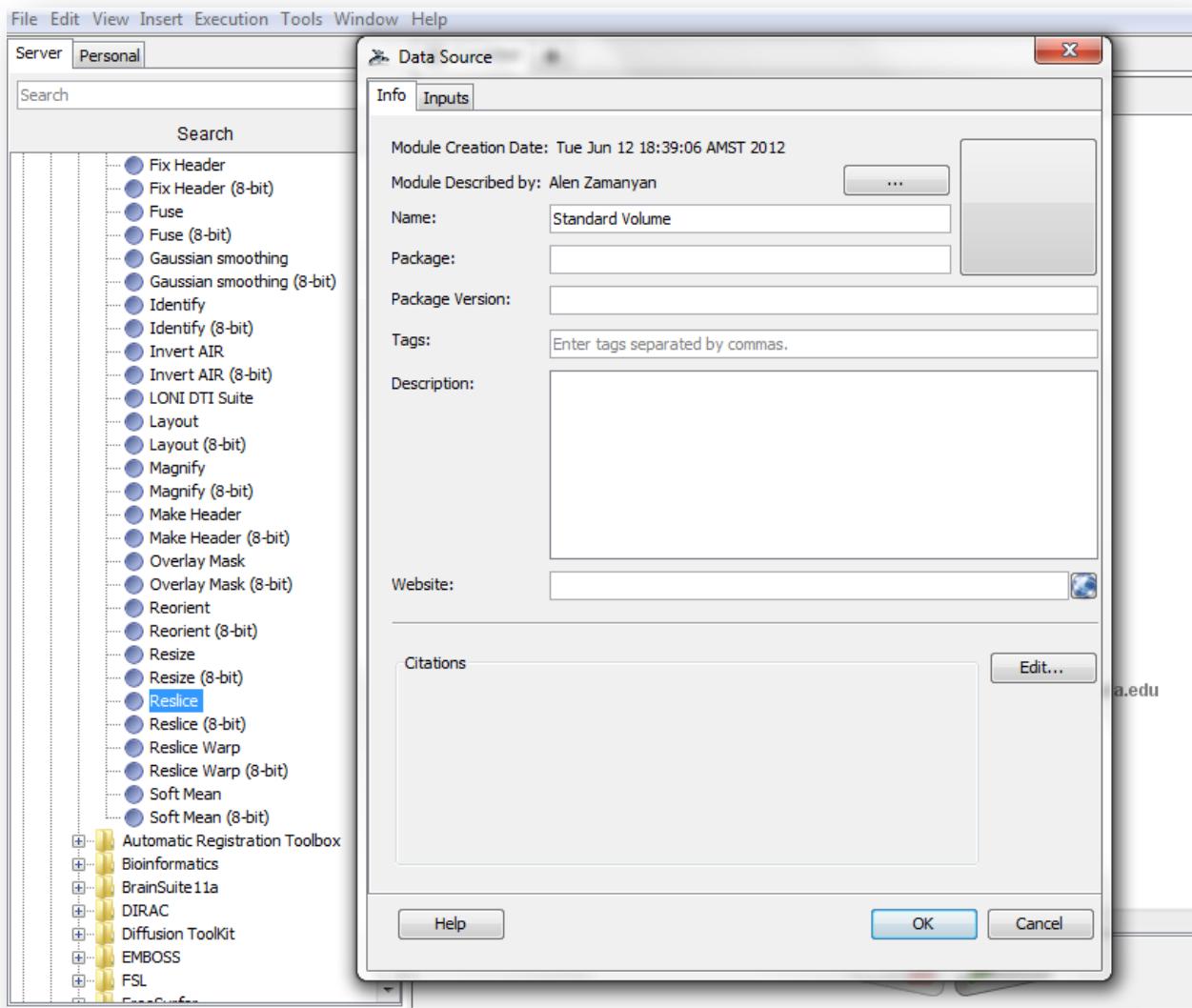
## Create Data Source



- 4) The two modules are connected, but we have yet to specify any names of input files or destinations of output files. Let's begin with the inputs. Right clicking on the canvas will display workflow options. Select "New data source..."

# Registration Using AIR

## Name Data Source



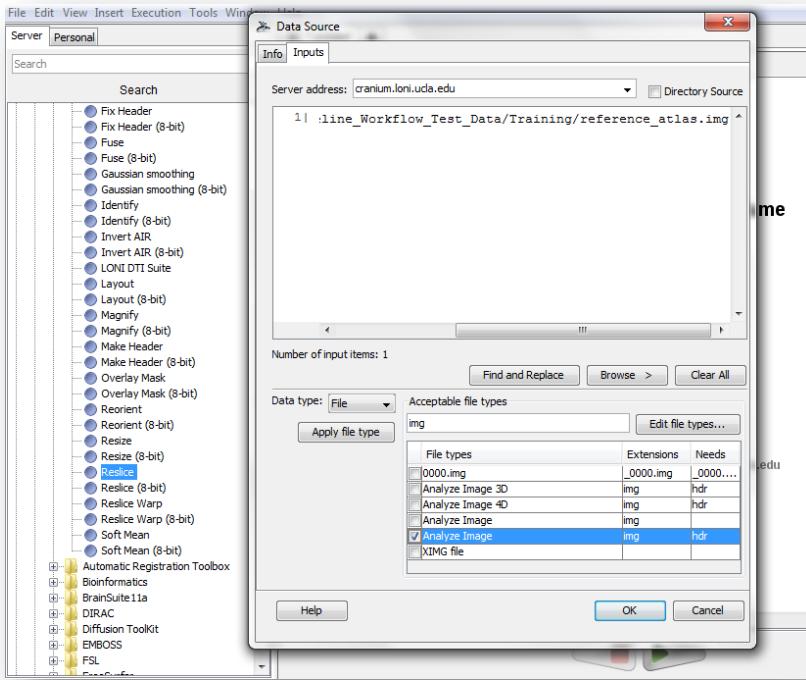
- 5) A dialog box "Data Source" appears on the screen, which has two tabs, namely Module and Data.

### Module

In this tab, we can specify the **optional** information about the input data. In order to distinguish this source from others we will create, it is a good idea to complete the "Name:" field. Type in "Standard Volume".

# Registration Using AIR

## Specify Input Data



### Data

There are three components that are **required** to be completed in this tab.

**Input Data Field:** First we will need to specify that the data is stored on the servers at LONI. There is a dropdown menu that appears for this field, which has two values, Local host and [cranium.loni.ucla.edu](http://cranium.loni.ucla.edu).

### Choose

**cranium.loni.ucla.edu**

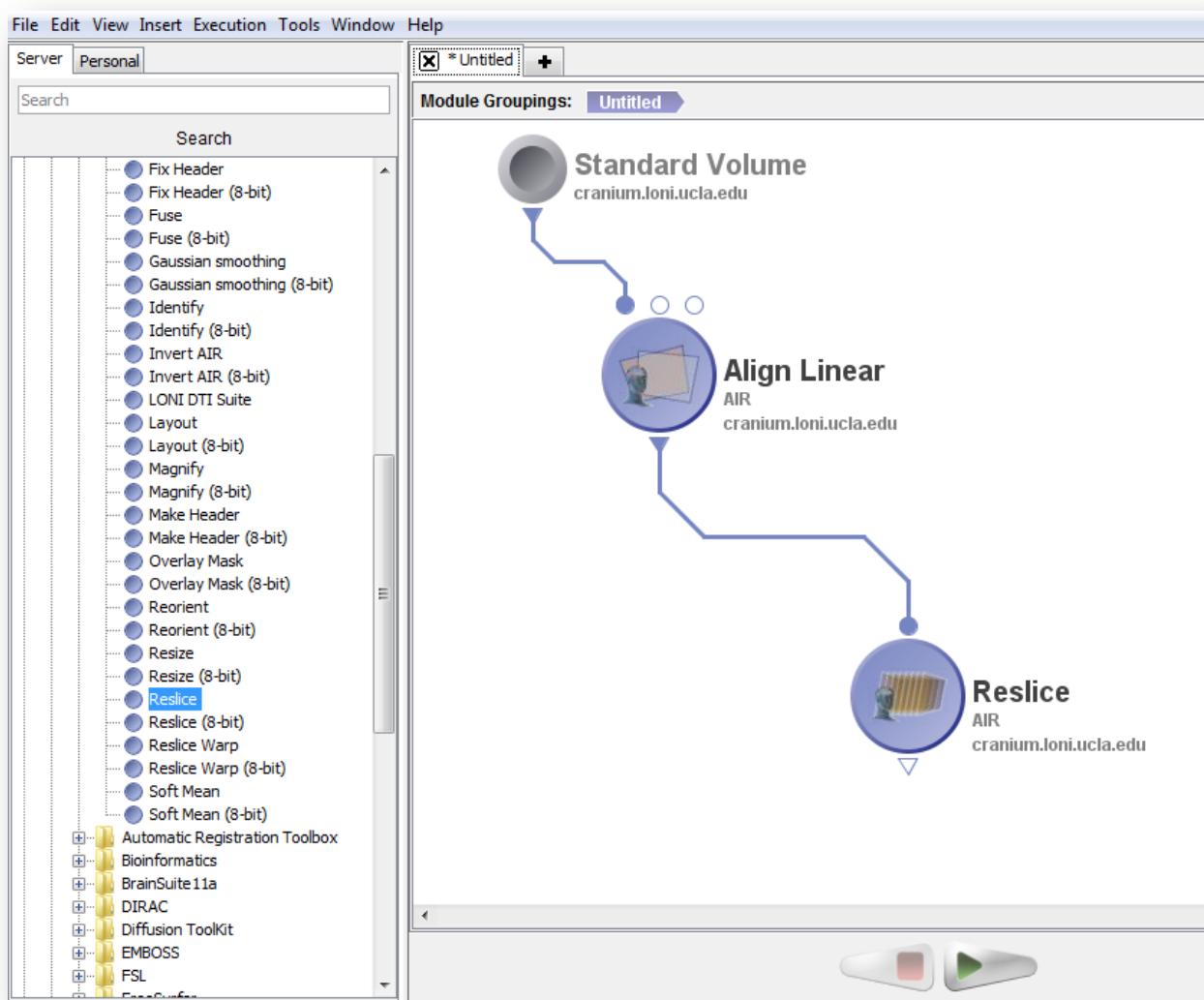
**Test Field (Below the Input Data Field):** Type in the full path to the file in the text area. In this case, we will use the file stored at

**/usr/local/loniData/Pipeline\_Workflow\_Test\_Data/Training/reference\_atlas.img**

**Input file Type:** We need to select the input file type. Select (by checking) "Analyze Image" and uncheck the file option (which is the default). Hitting the OK button will return us to the workflow canvas.

# Registration Using AIR

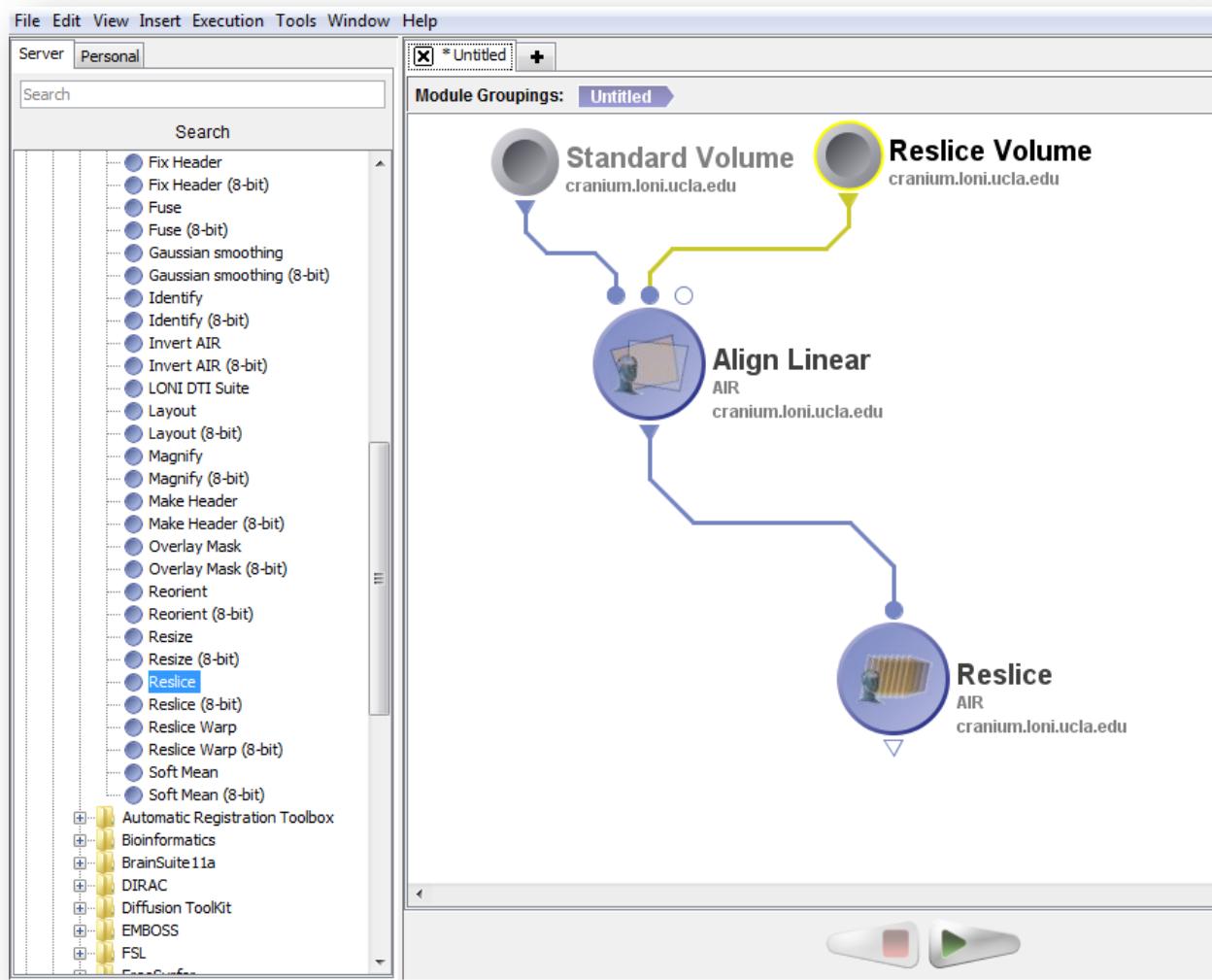
## Connect Standard Volume to Align Linear



- 6) Connect the data source that we just created to the first input parameter of Align Linear, labeled "Standard Volume." Do this in much the same way we connected Align Linear to Reslice. Click on the empty triangle of the data source and drag into the first empty circle above the Align Linear module.

# Registration Using AIR

## Reslice Volume Data Source

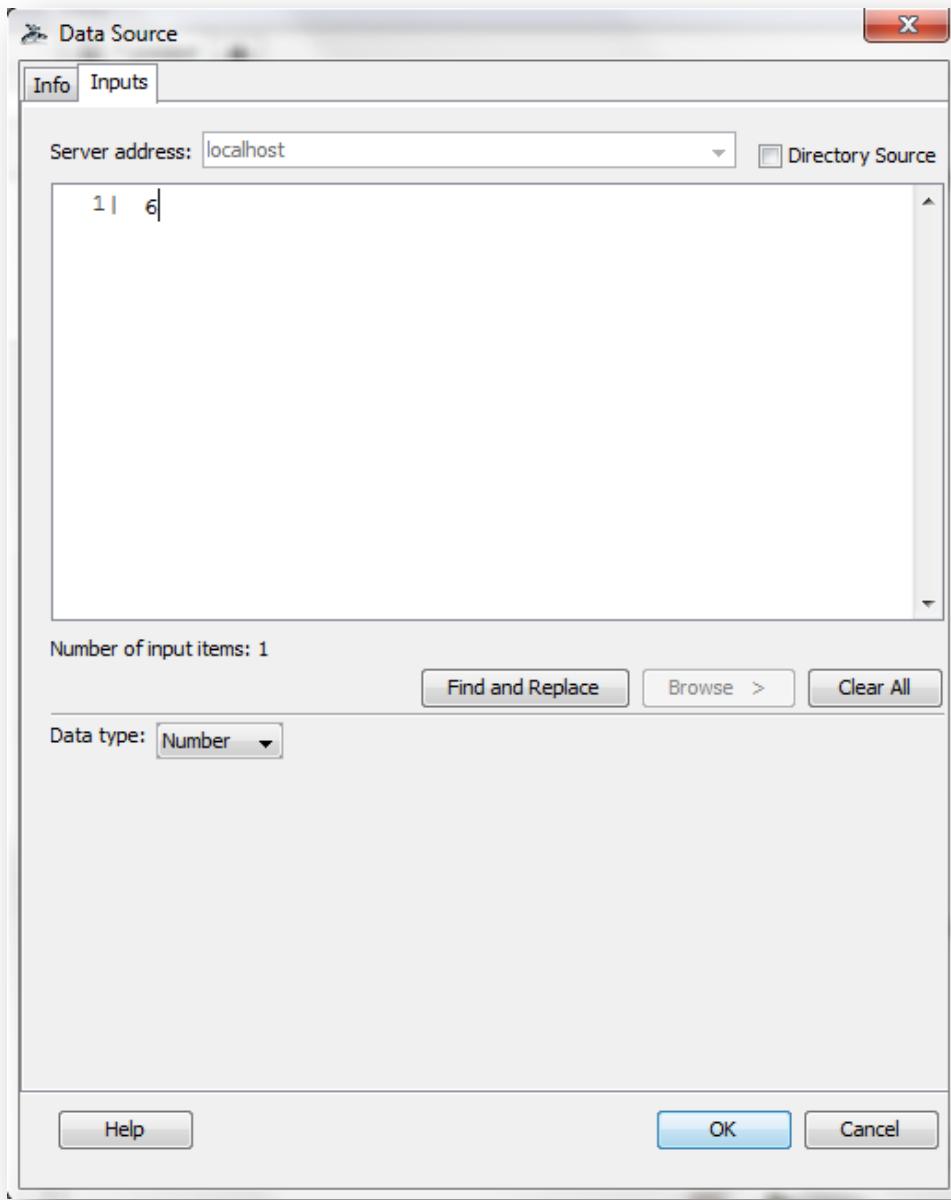


- 7) Repeat steps four through seven to create a data source for the file that is to be aligned to the standard. Call the data source "**Reslice Volume**" and specify the input filename as

**/usr/local/loniData/Pipeline\_Workflow\_Test\_Data/Training/source\_volume.img**. Make sure to switch from **localhost to cranium.loni.ucla.edu** and change the **filetype to Analyze Image**. Connect the input to the second empty circle above Align Linear (the parameter name is "Reslice Volume").

# Registration Using AIR

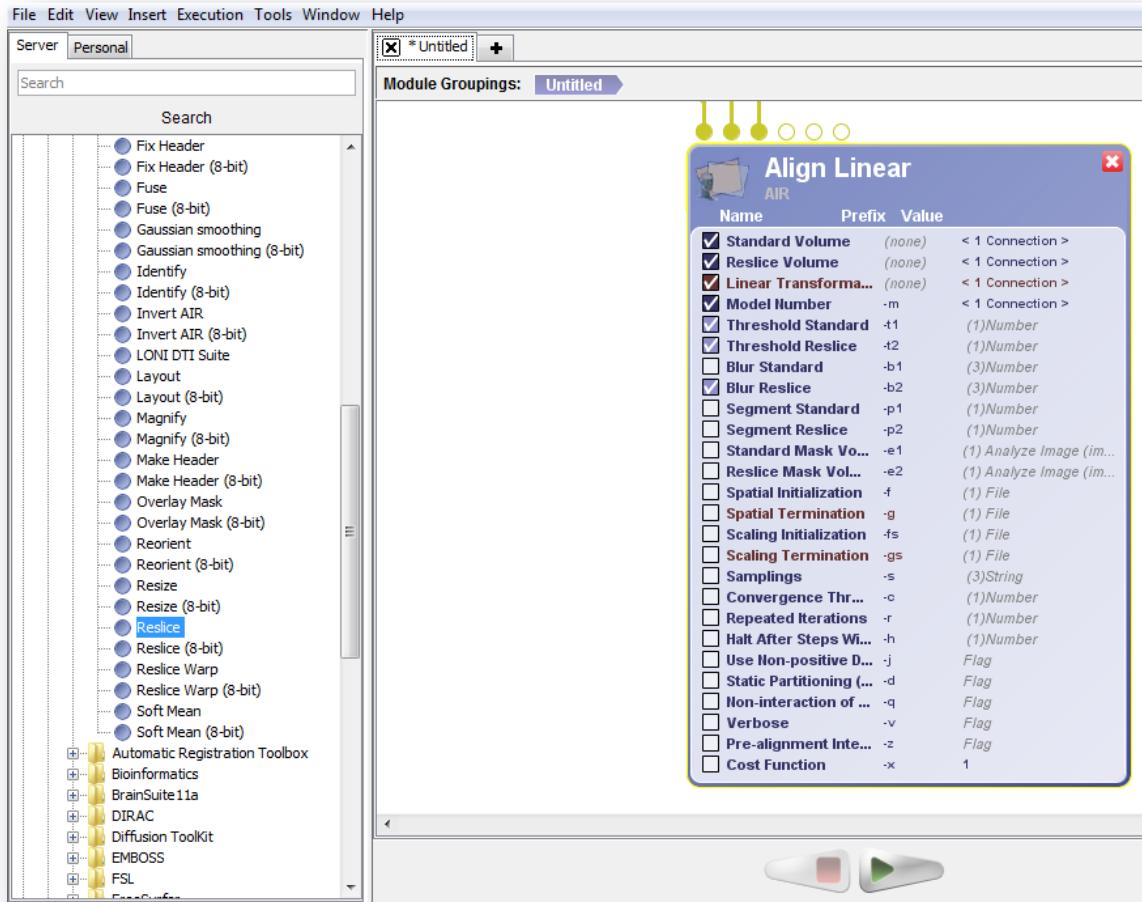
## Model Number Data Source



- 8) Create a third data source by right clicking on the canvas and selecting "New data source..." Name the data source "**Model Number**". In the Data tab, change the "**Data type**" to "**Number**". In the text area, enter the number 6.

# Registration Using AIR

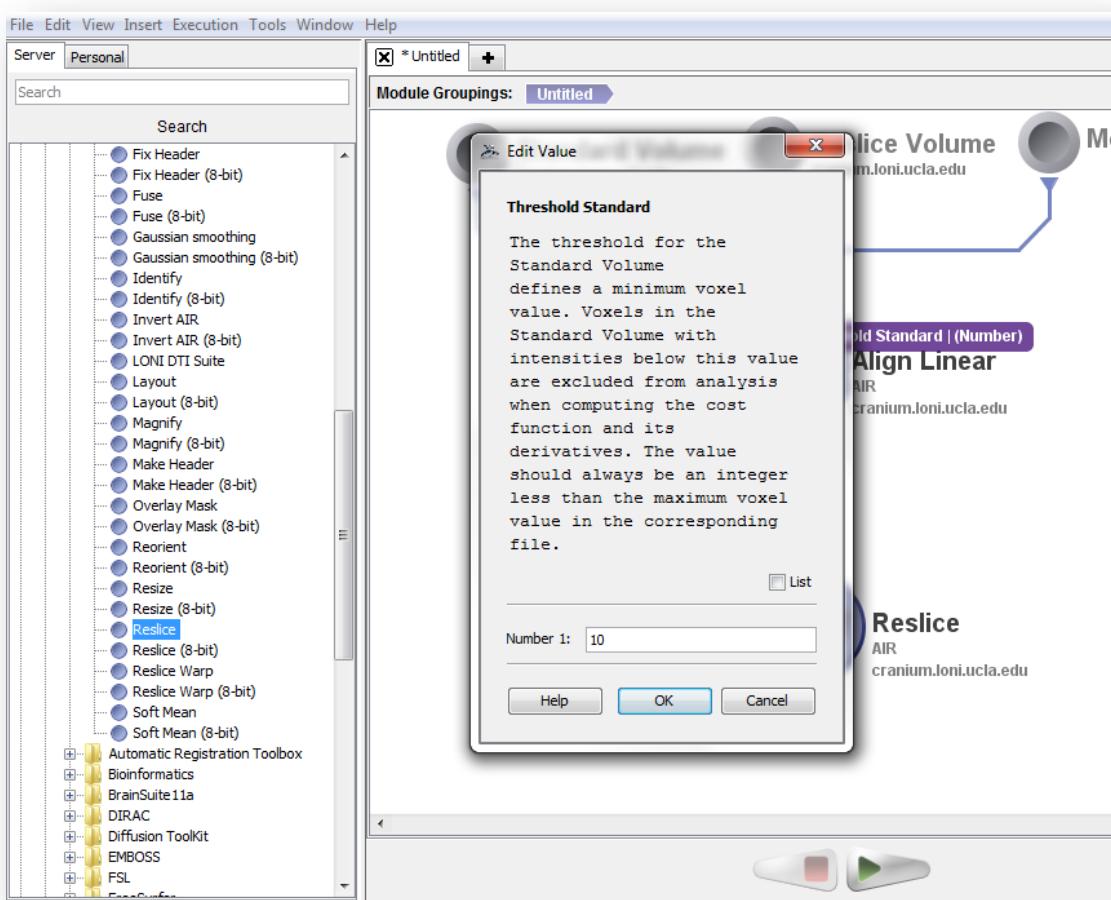
## Select Optional Parameters



- 9) For some modules, there are a number of options, which are optional, but highly recommended. In the case of Align Linear, we will threshold the standard and reslice volumes, along with blurring the reslice volume in order to achieve optimal results. To do this, double-click on the Align Linear module to bring up a panel with the complete list of parameters. In order to enable the optional parameters mentioned, click the box on the left of the Threshold Standard, Threshold Reslice, and Blur Reslice.

# Registration Using AIR

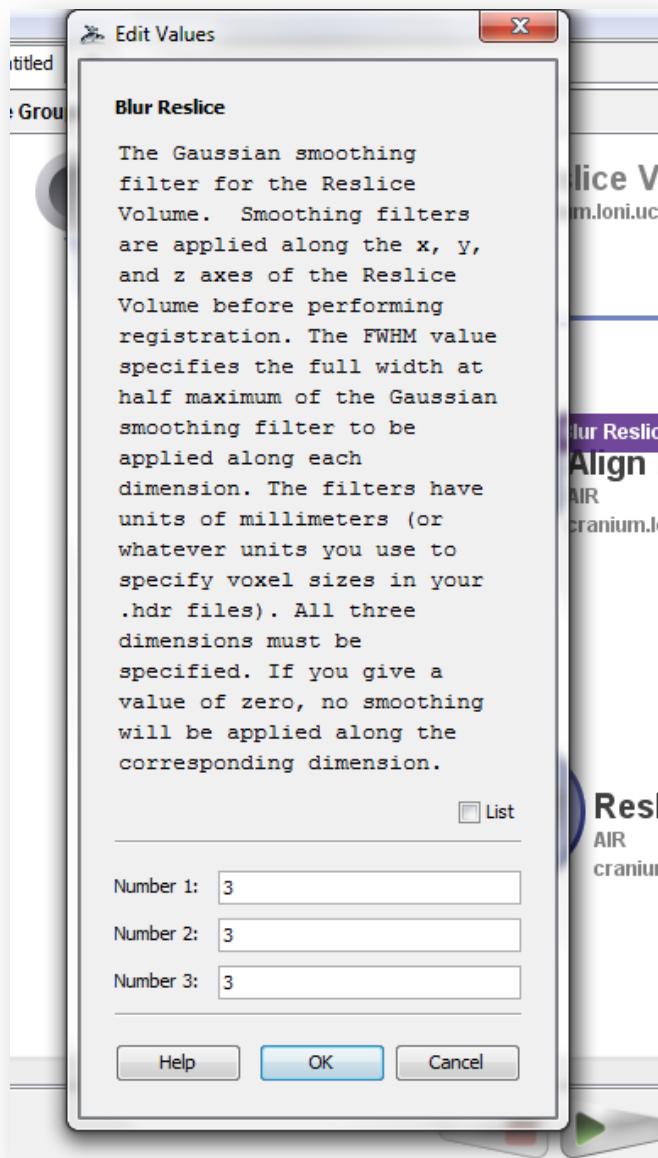
## Arguments to Threshold Standard/Reslice Parameters



- 10) Three new empty circles will appear above the Align Linear rectangular module. Double clicking on these will produce dialog boxes, which we will use to specify arguments. In the threshold dialogs, the parameters are described as such: "The threshold for the Standard/Reslice Volume defines a minimum voxel value. Voxels in the Standard/Reslice Volume with intensities below this value are excluded from analysis when computing the cost function and its derivatives. The value should always be an integer less than the maximum voxel value in the corresponding file." If we were to be meticulous about it, we could produce a histogram of the MRI data and find the threshold below which the values would be considered noise. For most MRI data, 10 is a reasonable value. Specify this value for both the standard and reslice files.

# Registration Using AIR

## Arguments to Blur Reslice Parameter

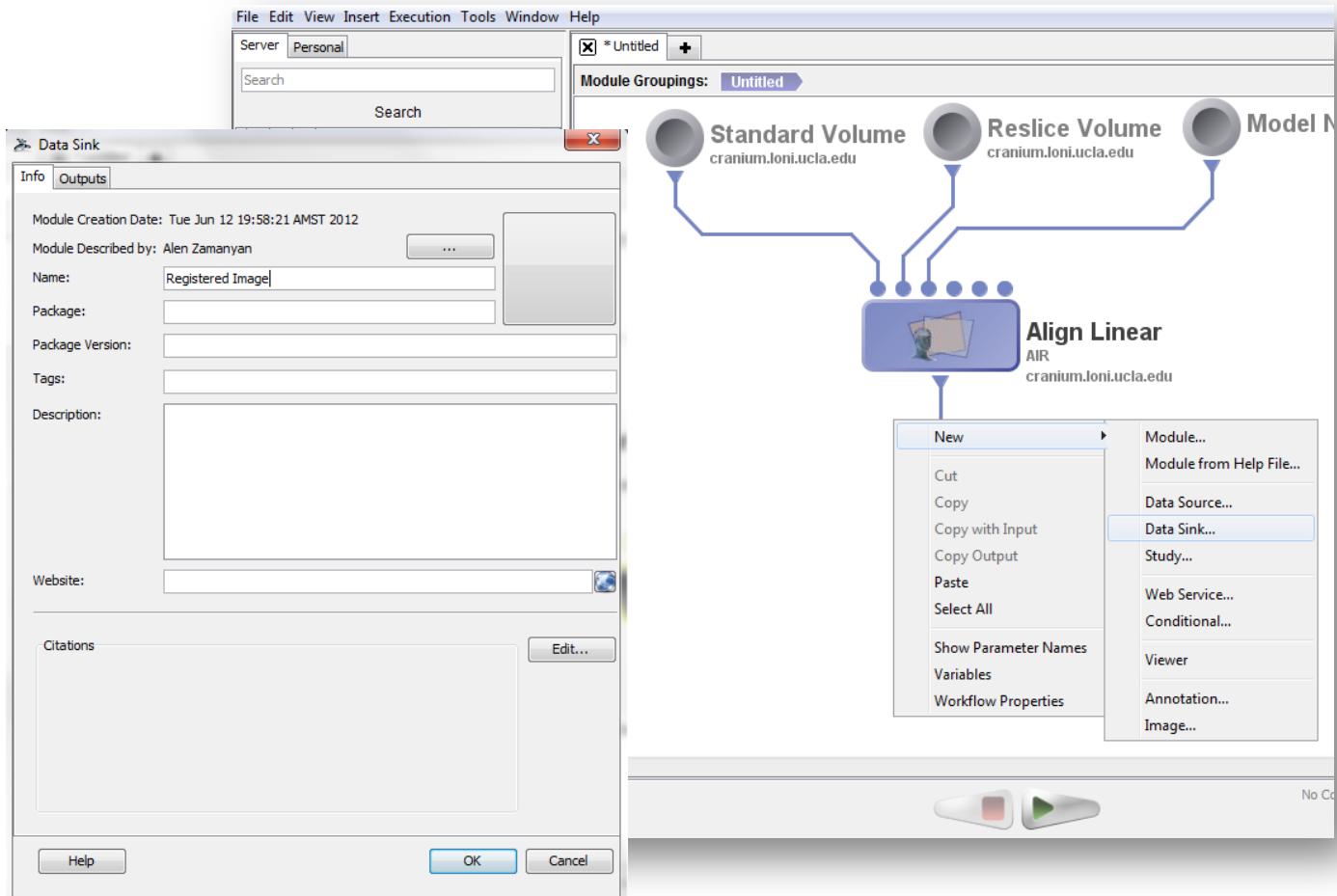


11) The description of the blurring option states: "The Gaussian smoothing filter for the Reslice Volume. Smoothing filters are applied along the x, y, and z axes of the Reslice Volume before performing registration. The FWHM value specifies the full width at half maximum of the Gaussian smoothing filter to be applied along each dimension. The filters have units of millimeters (or whatever units you use to specify voxel sizes in your .hdr files). All three dimensions must be specified. If you give a value of zero, no smoothing will be applied along the corresponding dimension." The concept is a simple one. In structural MRI, the atlas is typically an average of many subject MRIs, hence, if visualized, it looks blurry. A single subject's MRI is much crisper. In this instance, simply blurring the subject MRI may provide better registration. So for a voxel size of 1x1x1, blurring at a rate of 3x3x3

should be appropriate. Enter in the value 3 for each of the three text boxes in the edit parameter dialog box.

## Data Sink for Registered Image

# Registration Using AIR



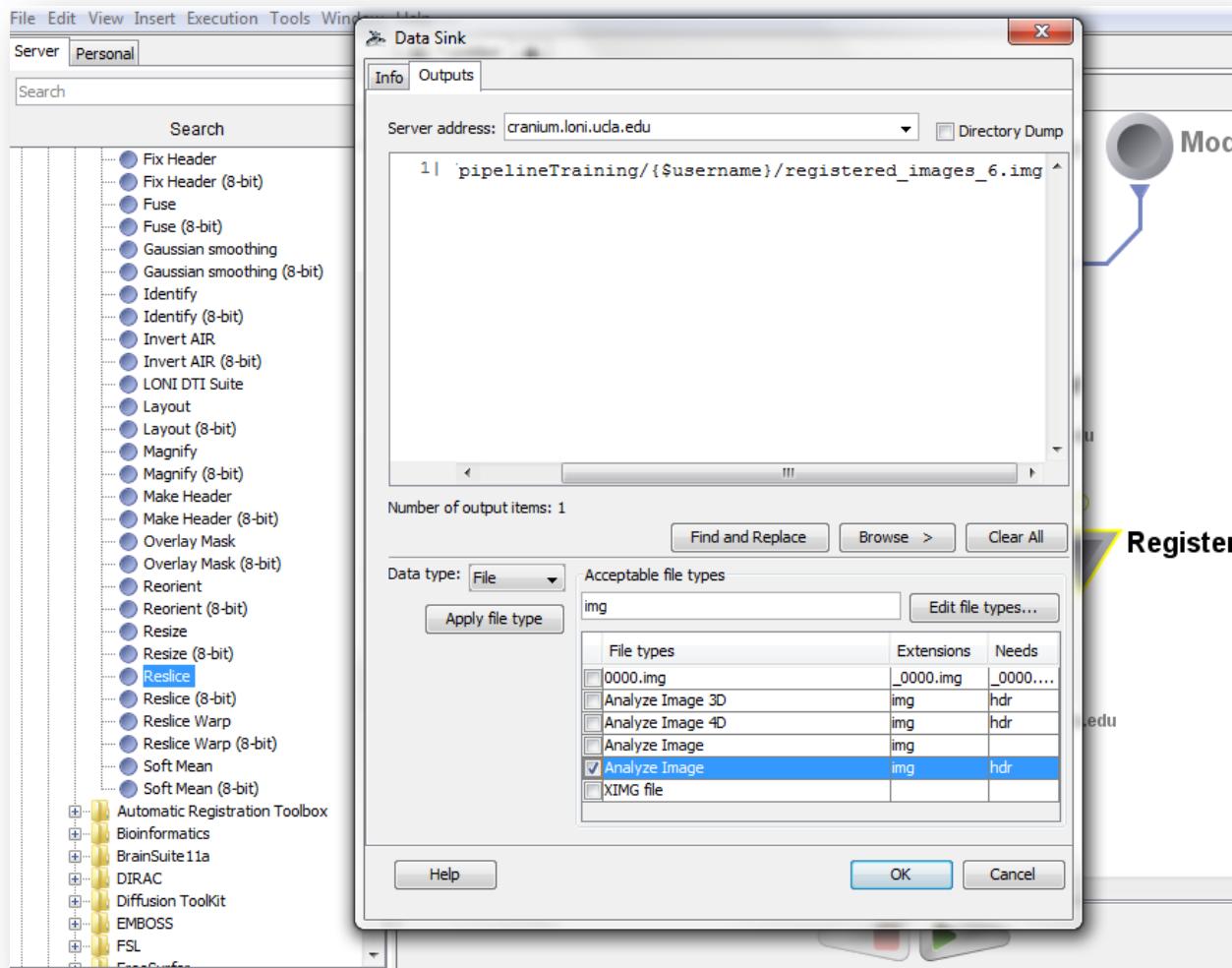
- 12) We can now proceed to specify the destination of the output of Reslice. Right click on the workflow canvas, and select "New data sink..."

## Module tab

Fill in the name field with the value "Registered Image."

# Registration Using AIR

## Destination of Output File

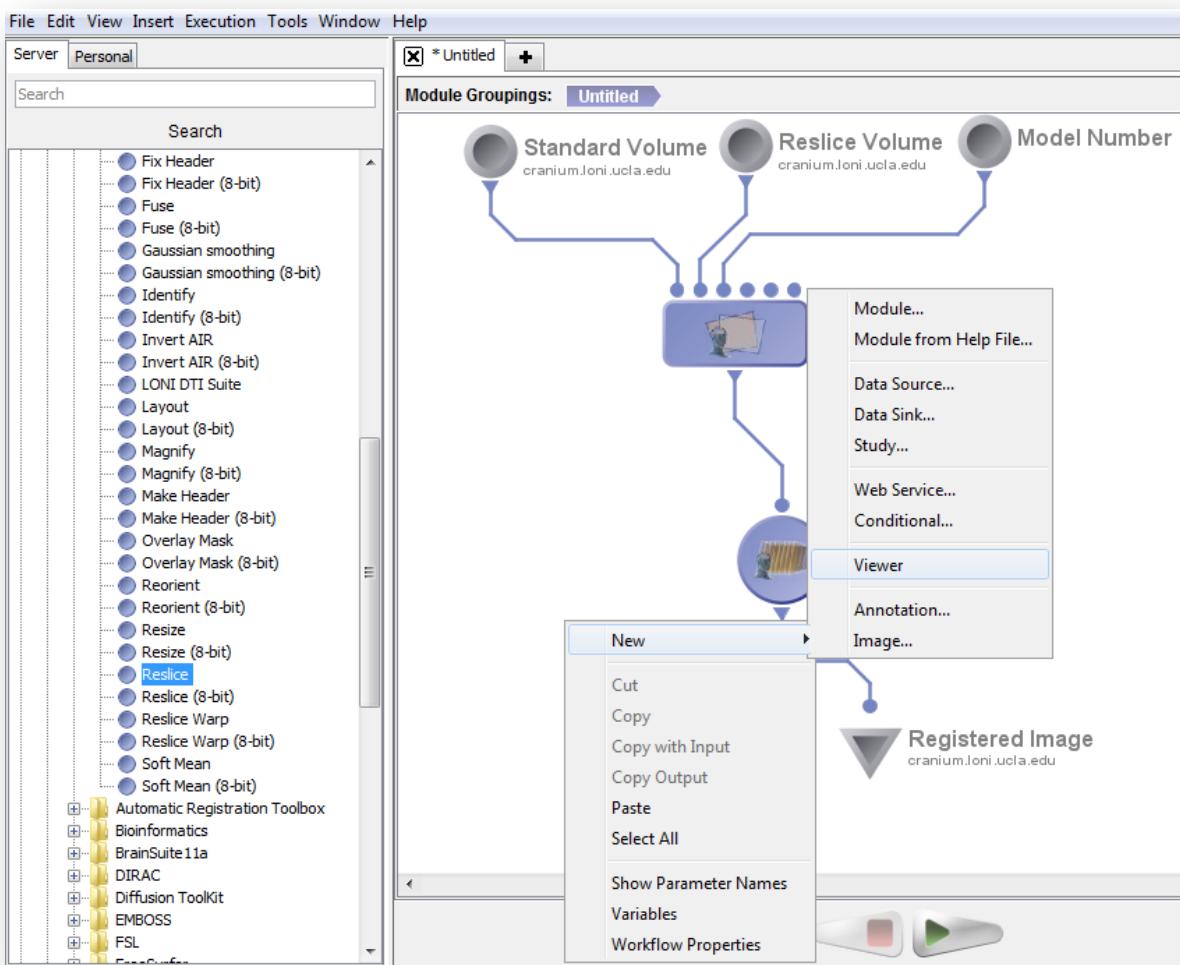


### Data tab

Select cranium.loni.ucla.edu from the dropdown at the top of the window. The output filetype is Analyze Image. In the text area, specify the output file as /ifs/tmp/pipelineTraining/{\$username}/registered\_images\_6.img. Hit OK and connect the output of Reslice to the input of the data sink.

# Registration Using AIR

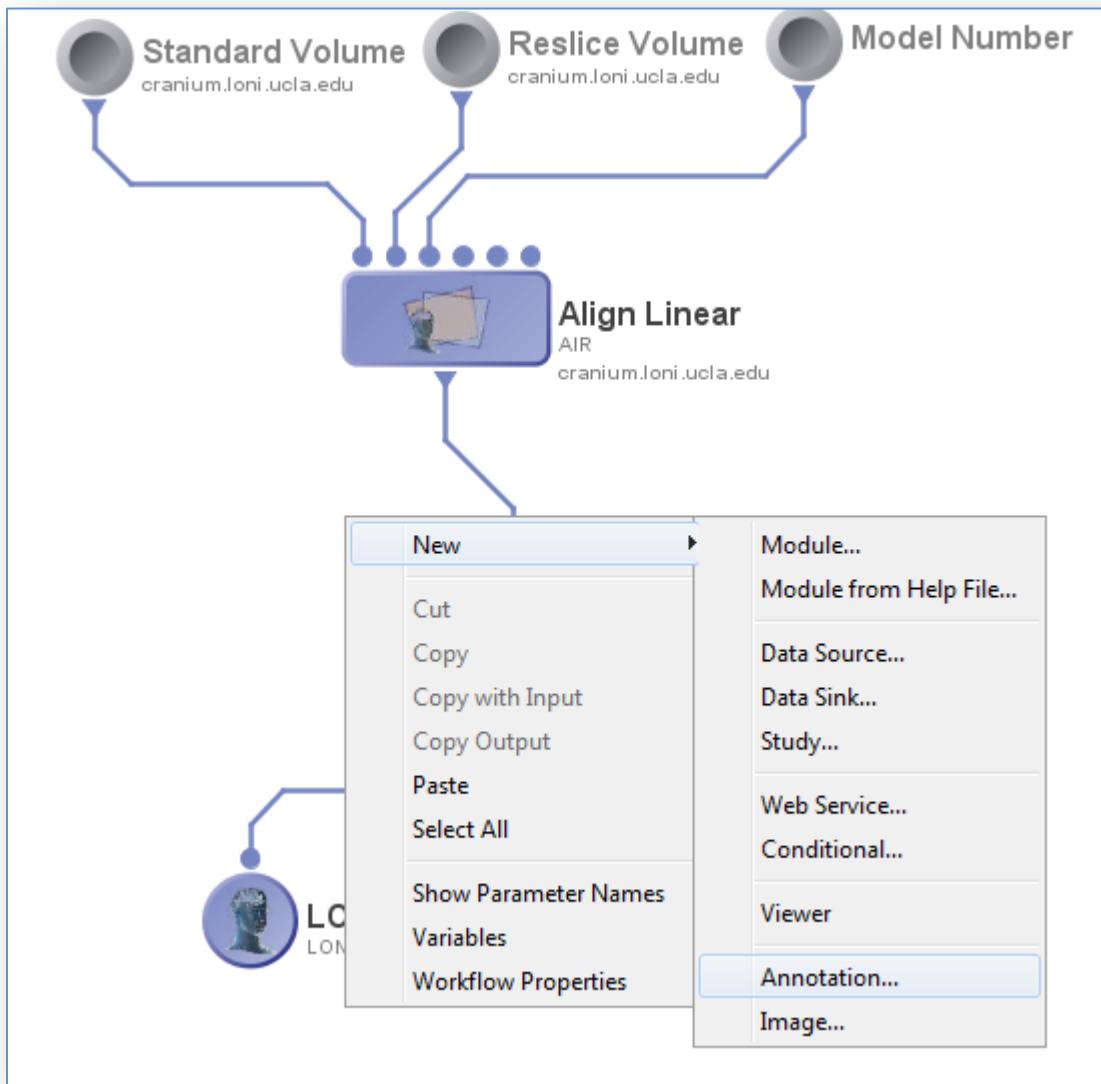
## LONI Viewer



- 13) Finally, for visualization purposes, we will create a LONI Viewer module. Right-click on the canvas and select "LONI Viewer..." Hit the OK button and connect the output of Reslice to the input of the viewer. Optionally, we can create another viewer and connect it to the input volumes.

# Registration Using AIR

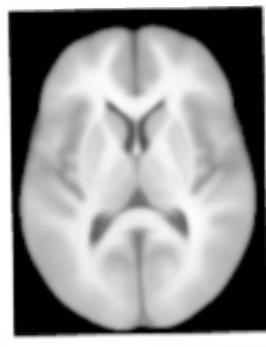
## Annotation Option



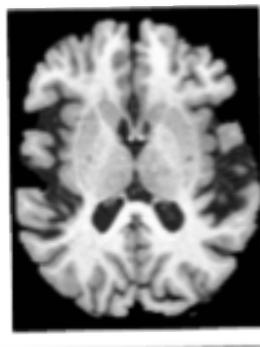
- 14) Optionally, you can create annotations by right-clicking on the canvas and selecting "New annotation..."

# Registration Using AIR

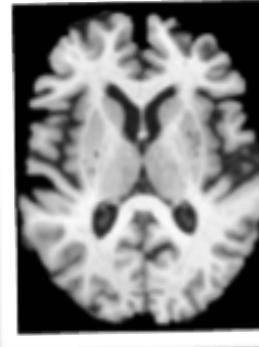
## Results



Reference  
Atlas



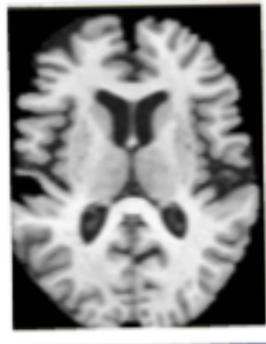
Source  
Volume



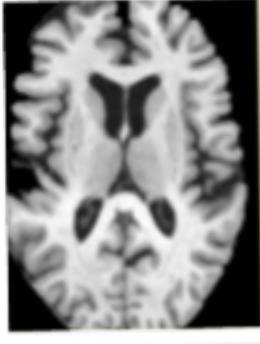
6 Parameter  
Registration

The results of this Pipeline Workflow would look like the 6 Parameter Registration image. The Reference and Source Volumes are included for comparison.

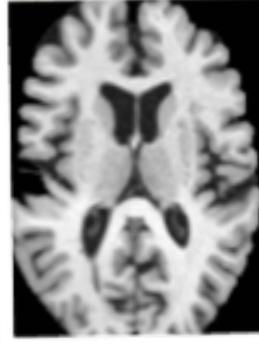
Optionally, you could also increase the degrees of freedom to



7 Parameter  
Registration



9 Parameter  
Registration



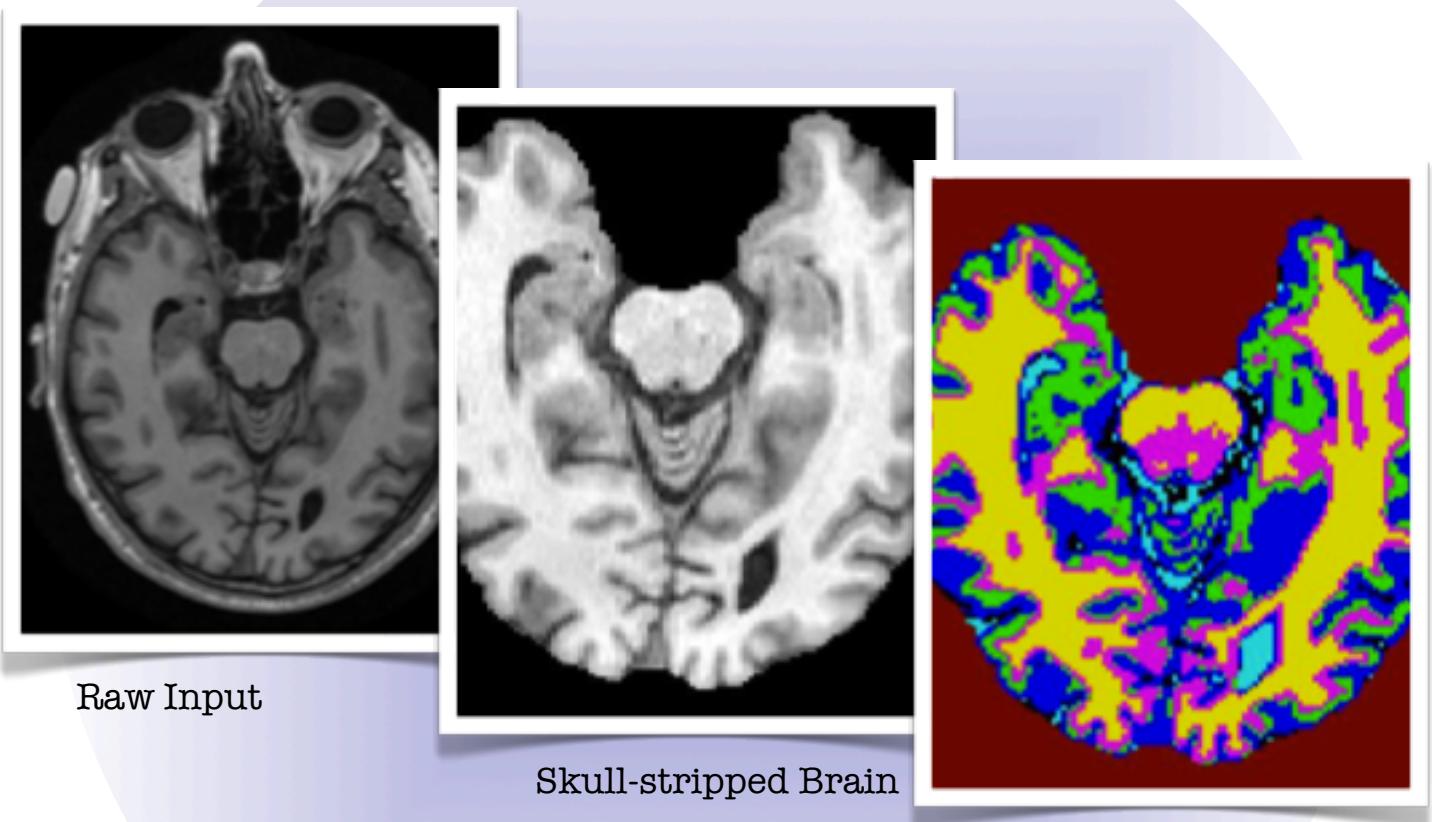
12 Parameter  
Registration

fit the source volume to the reference even tighter. To get these optional results, in step 8, list the numbers 6, 7, 9, and 12, each on its own line.

## Other Options...

# BrainSuite Processing

BrainSuite



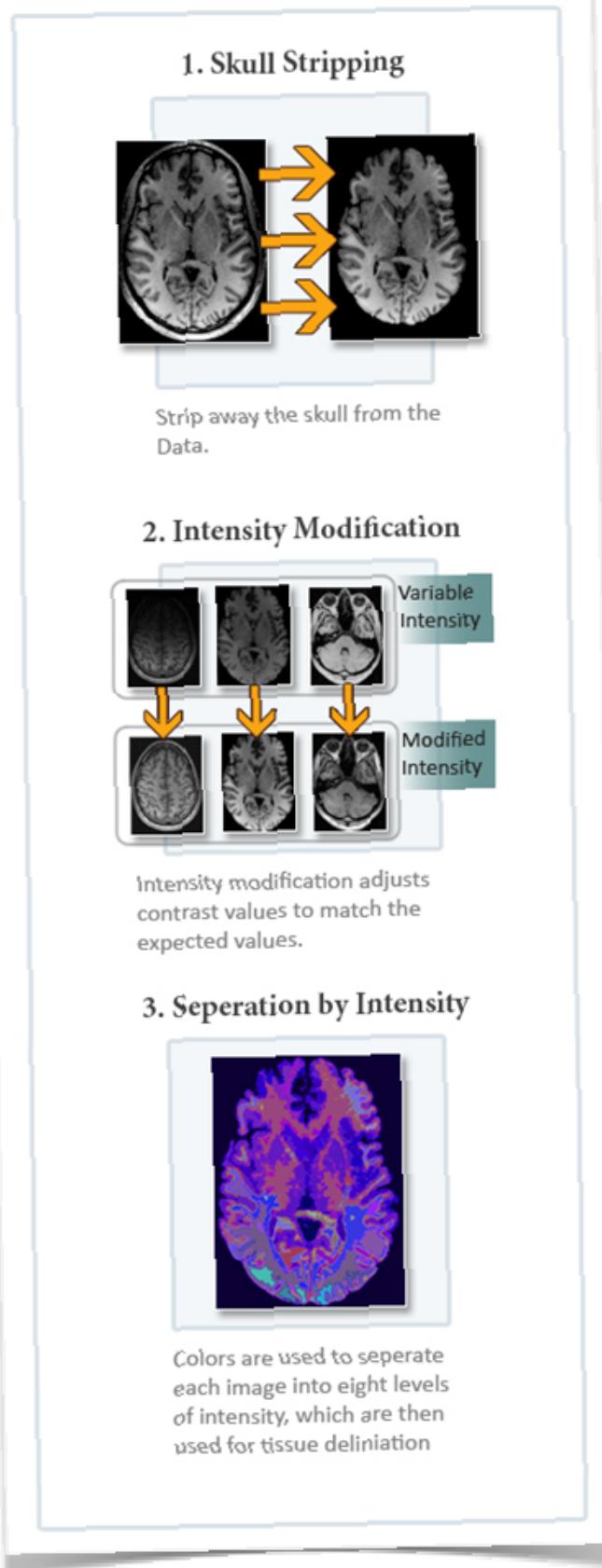
Raw Input

Skull-stripped Brain

Tissue Classes

# BrainSuite Processing

## Introduction



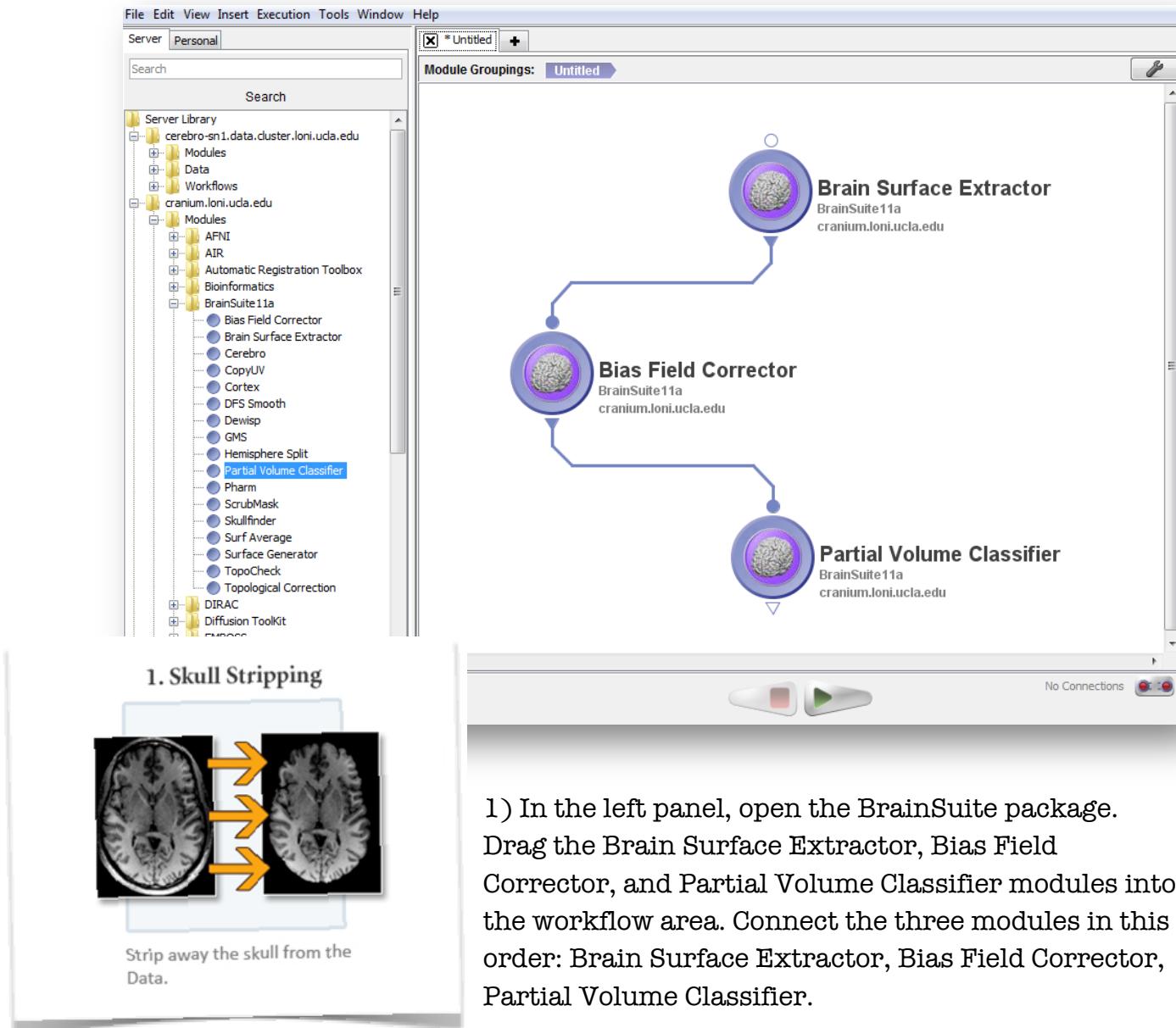
The sequence of steps needed to perform skull-stripping, inhomogeneity correction, and tissue classification.

### Overview

This example utilizes three tools from the BrainSuite package. Given a raw brain image in Analyze format, the objective of the workflow is to generate a file containing labels for each of the three tissue types, namely gray matter, white matter, and cerebrospinal fluid. The initial steps skull-strip the brain volume and perform inhomogeneity correction, prepping the image for the tissue classifier.

# BrainSuite Processing

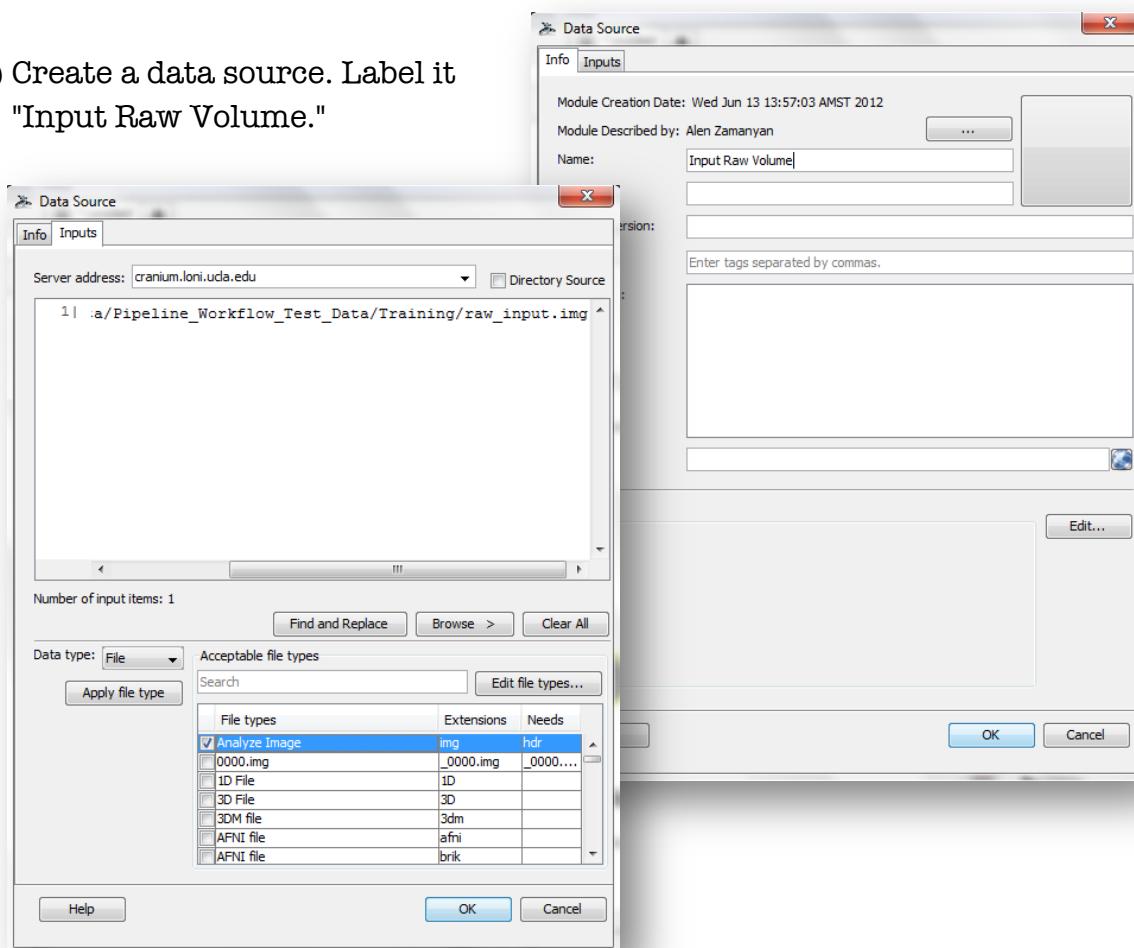
## Select BrainSuite Modules



# BrainSuite Processing

## Specify Input File

- 2) Create a data source. Label it "Input Raw Volume."

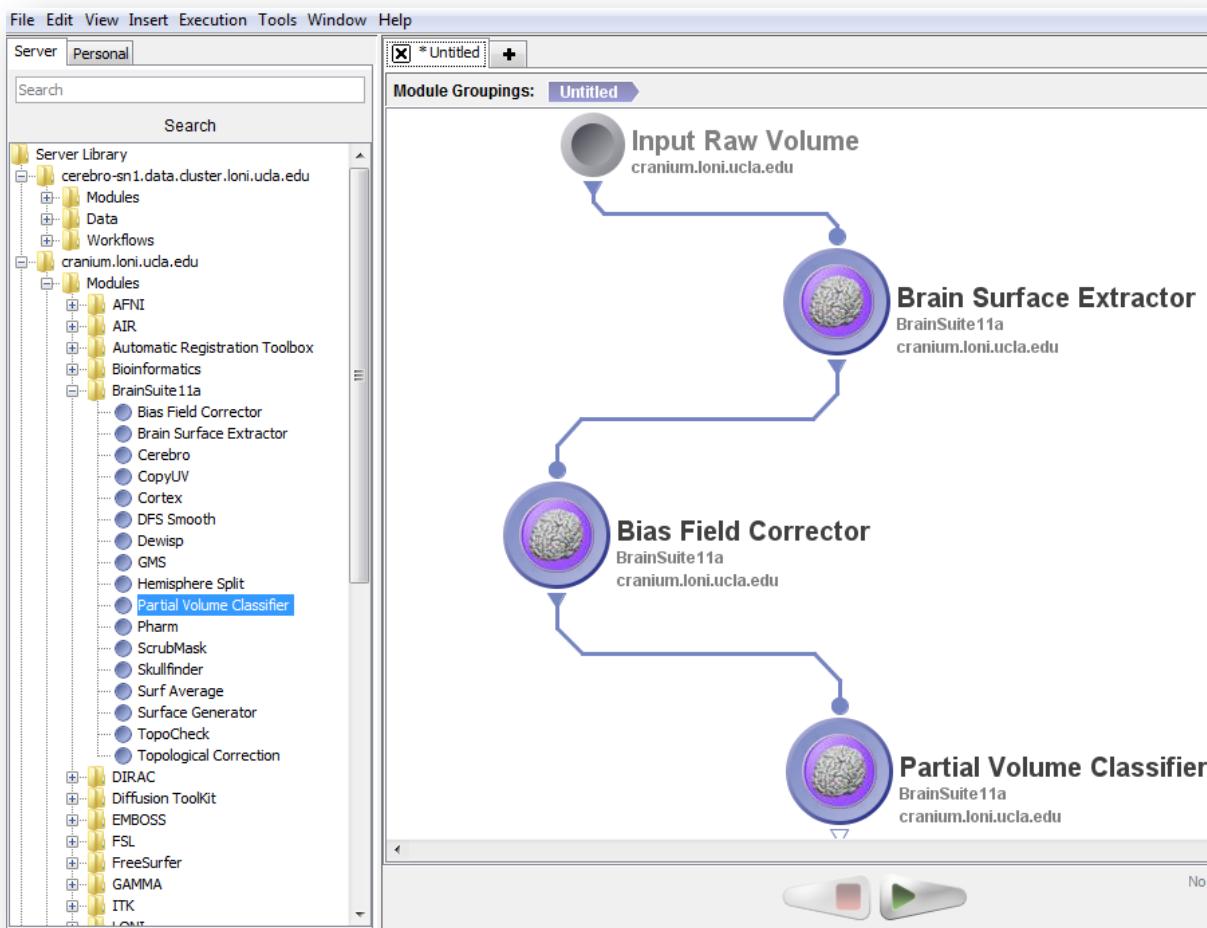


The path of the input is

/usr/local/loniData/Pipeline\_Workflow\_Test\_Data/Training/raw\_input.img. Make sure to select cranium.loni.ucla.edu as the server and to check Analyze Image as the file type. Hit OK.

# BrainSuite Processing

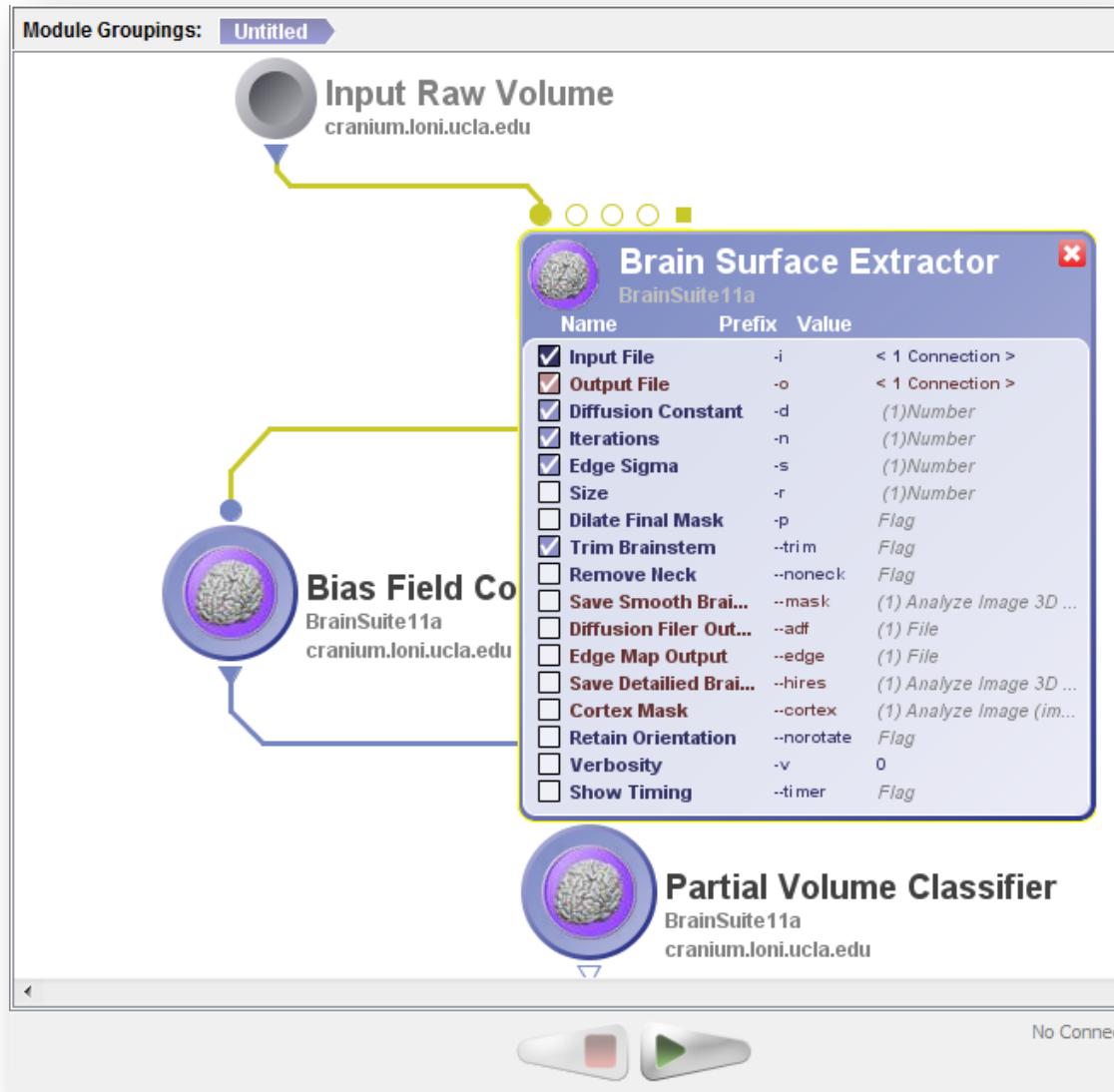
## Connect Data Source to BSE



3) Connect the data source to Brain Surface Extractor.

# BrainSuite Processing

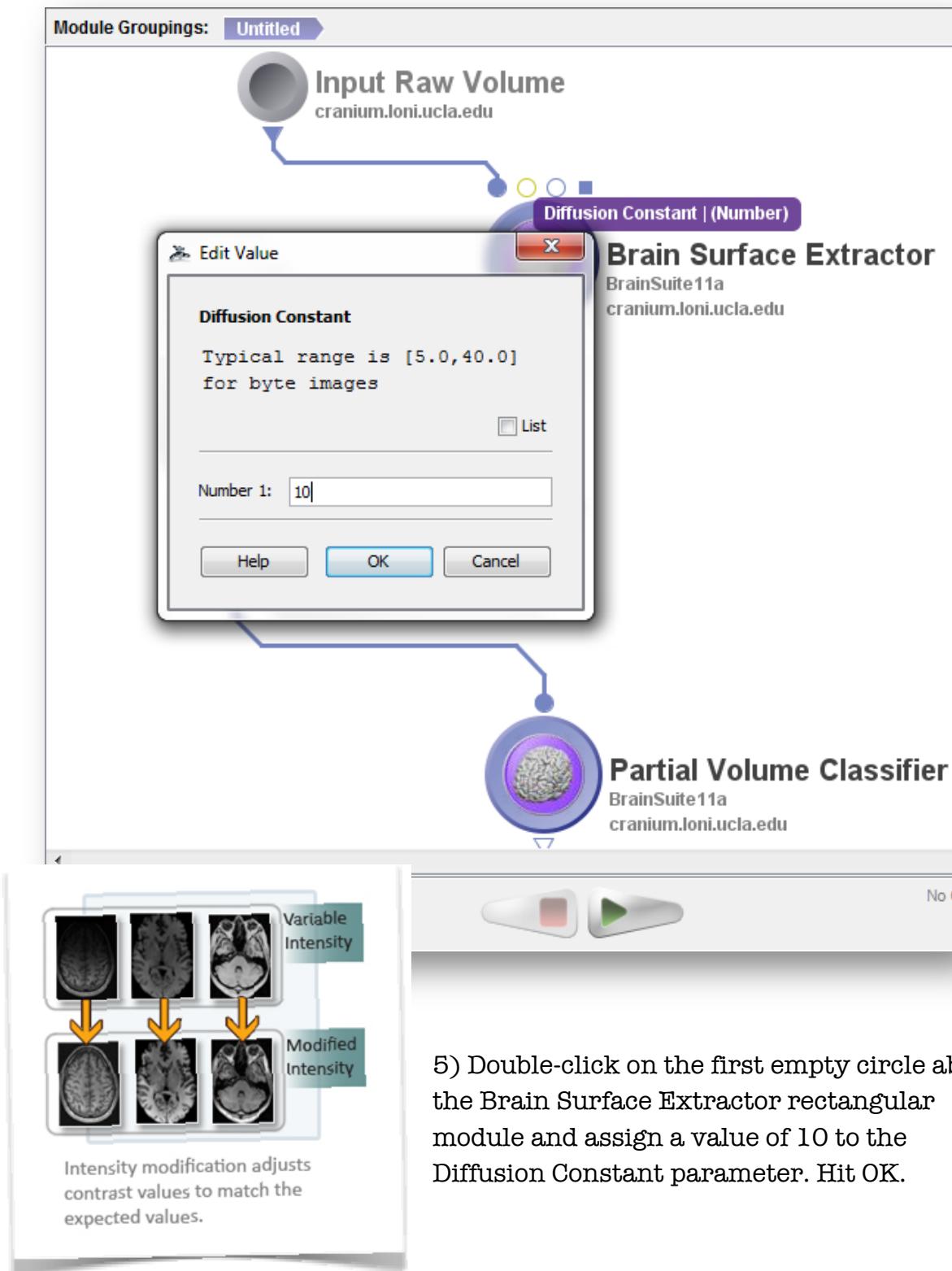
## Optional Parameters



- 4) Double-clicking on Brain Surface Extractor will bring up a panel allowing us to choose optional parameters. Select Iterations, Diffusion Constants, Edge Sigma, and Trim Brainstem.

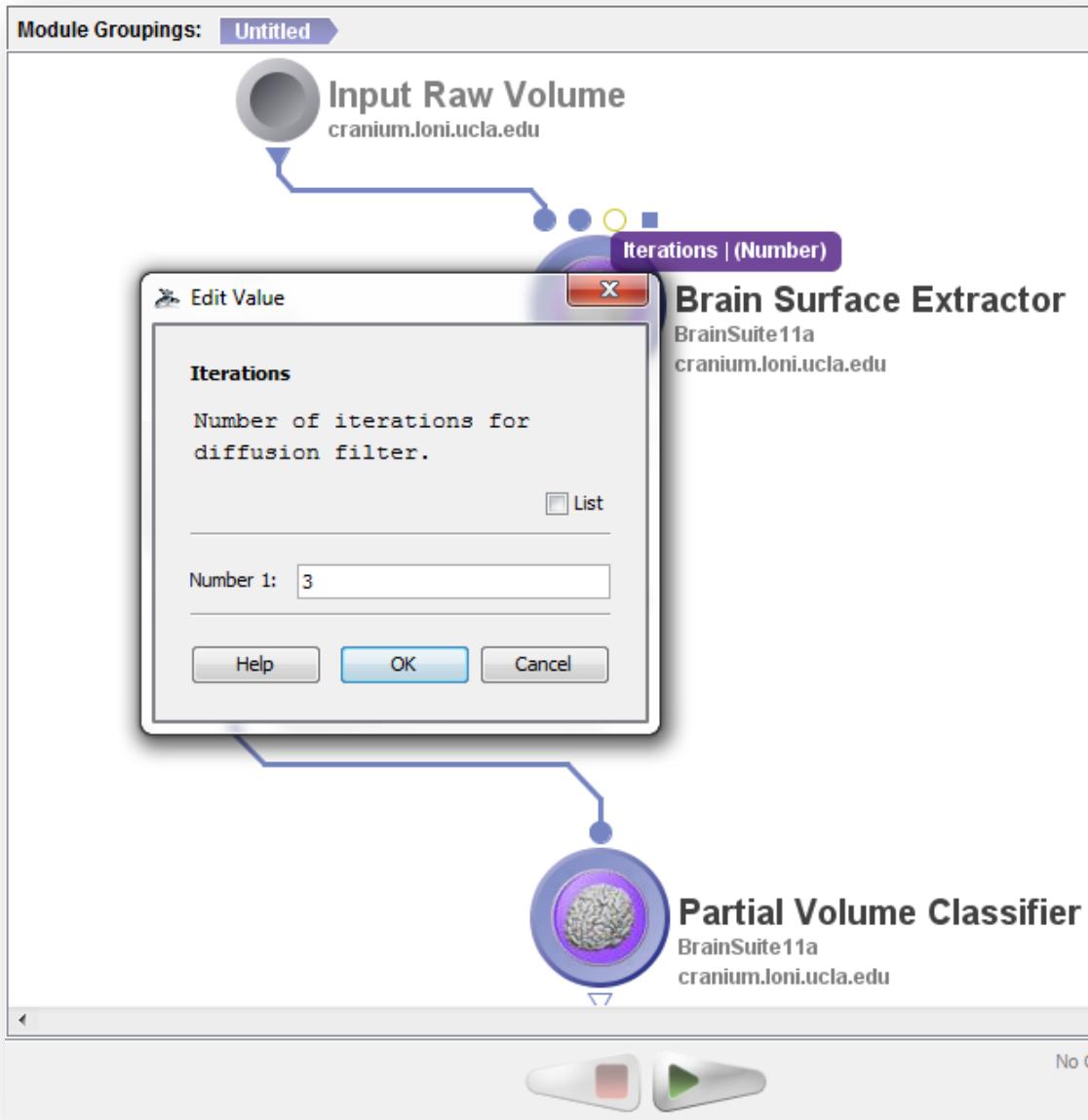
# BrainSuite Processing

## Diffusion Constant



# BrainSuite Processing

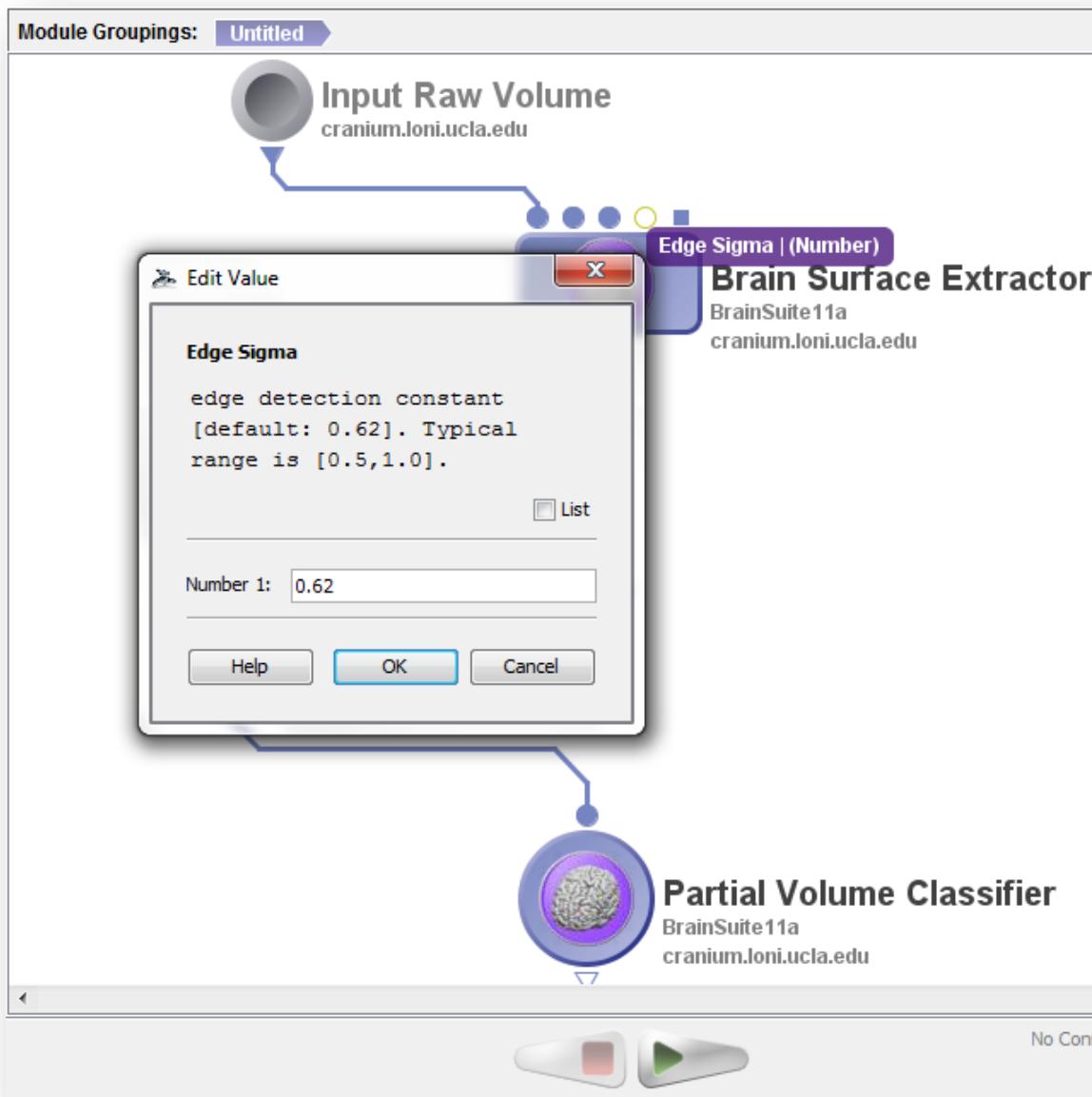
## Iteration of Diffusion Filter



- 6) Double-click on the Iterations parameter and assign a value of 3. Hit OK.

# BrainSuite Processing

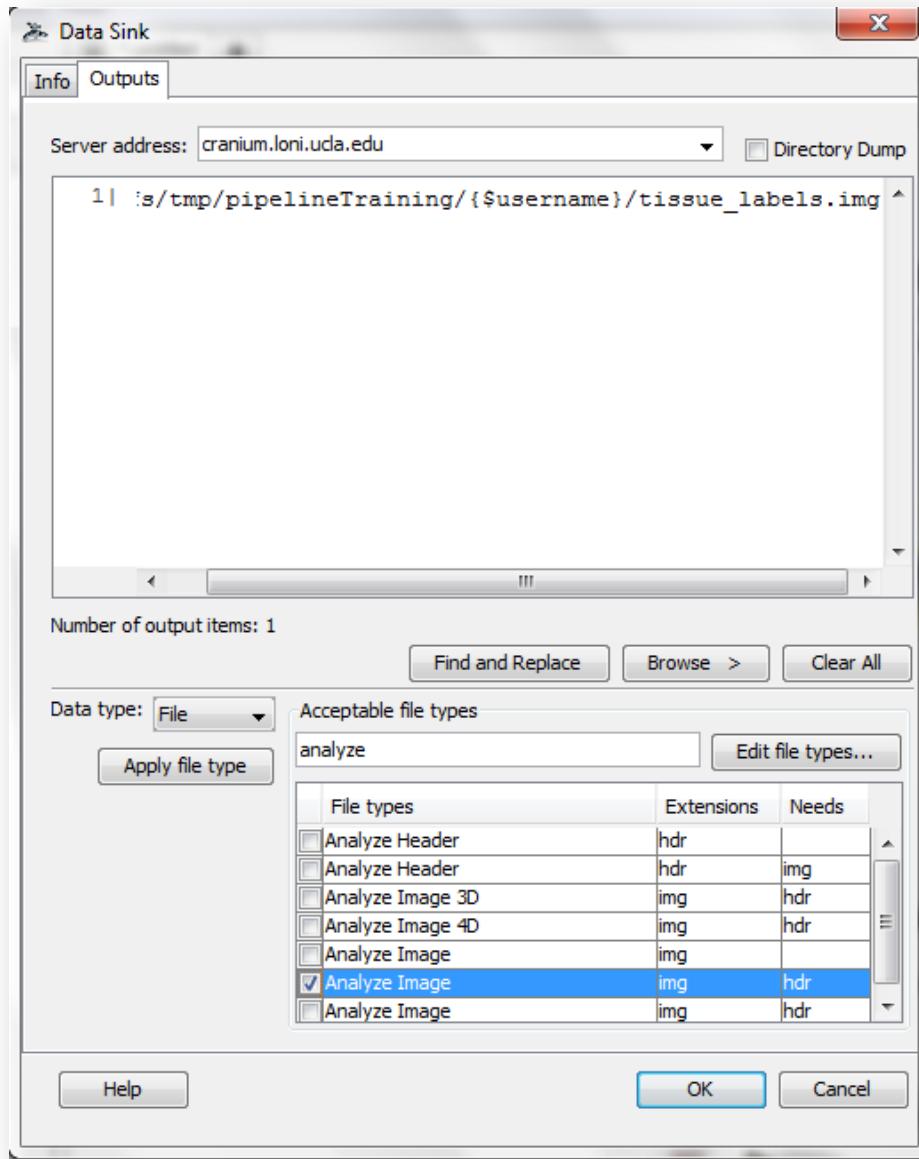
## Edge Scale



7) Double-click on the Edge Sigma parameter and assign a value of 0.62. Hit OK.

# BrainSuite Processing

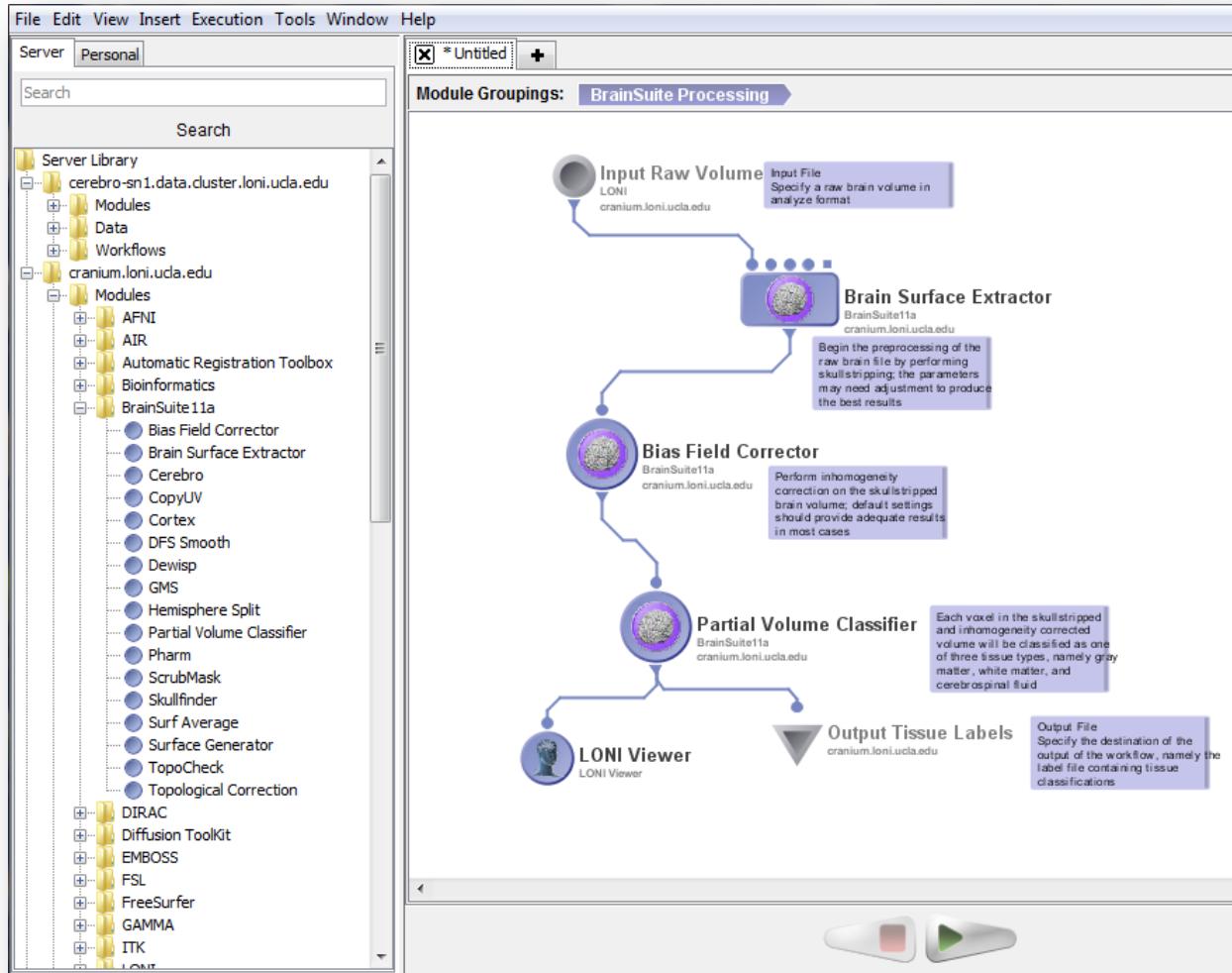
## Specify Output in Data Sink



- 8) Create a data sink and label it "Output Tissue Labels."
- 9) Make sure that under the Data tab, the "Output locations:" dropdown says cranium.loni.ucla.edu. Specify the path of the output file as /ifs/tmp/pipelineTraining/{\$username}/tissue\_labels.img. Make sure to select Analyze Image in the dropdown at the bottom of the panel.

# BrainSuite Processing

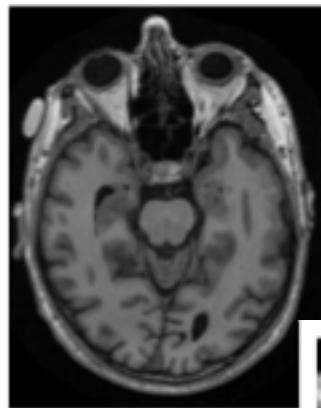
## LONI Viewer/Optional Annotations



10) Create a viewer by right-clicking on the canvas and selecting "New viewer..." Hit OK and connect the viewer's input to the data source and the output of Partial Volume Classifier.

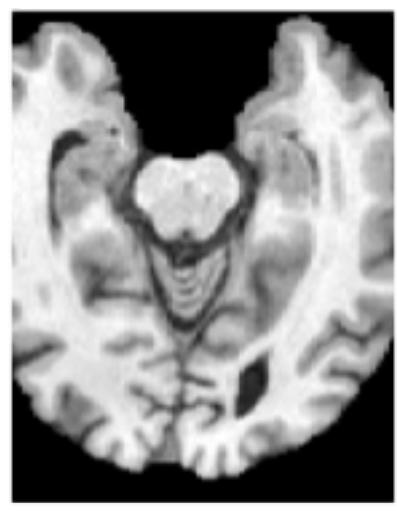
11) Optionally, you can create annotations by right-clicking, selecting "New annotation..." and typing in the message that you want to include.

# BrainSuite Processing

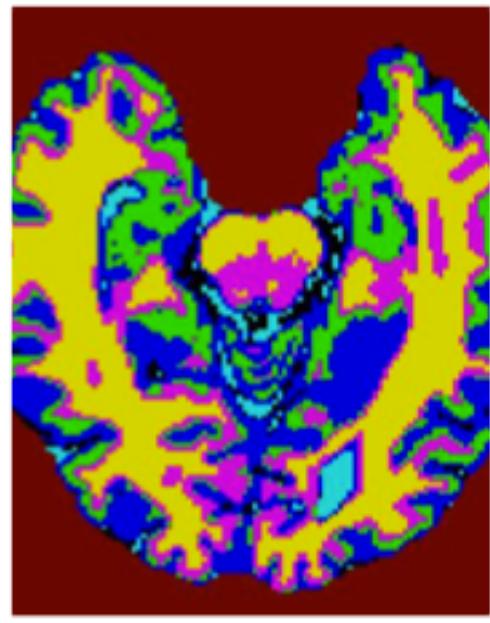


Raw Input

## Results



Skull-stripped Brain

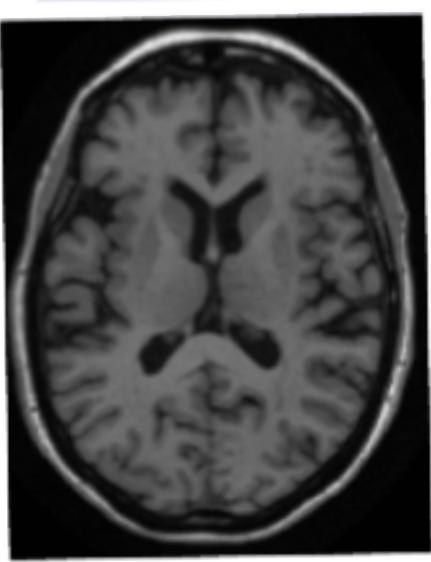


Tissue Classes

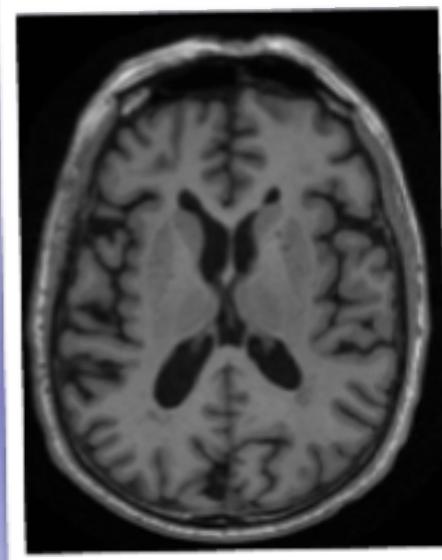
The Raw Input is skull-stripped and then tissue classified. The resultant image shows eight colors of tissue:

- red = CSF
- yellow = white matter
- green = grey matter
- blue = grey matter/CSF
- cyan = partial volume CSF/other
- purple = grey matter/white matter
- brownish red = background

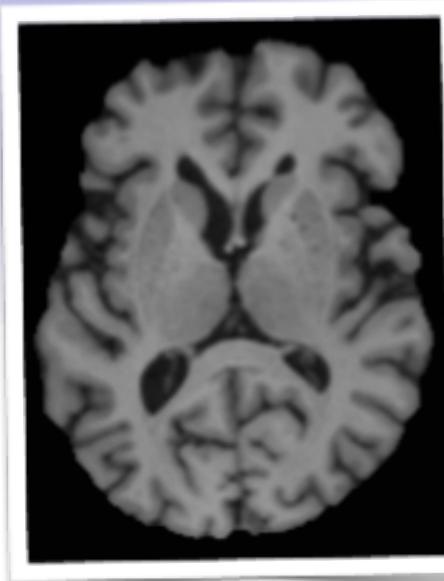
# Automated Image Registration & BrainSuite



Raw Input



Target Volume



Registered and  
Skull-stripped

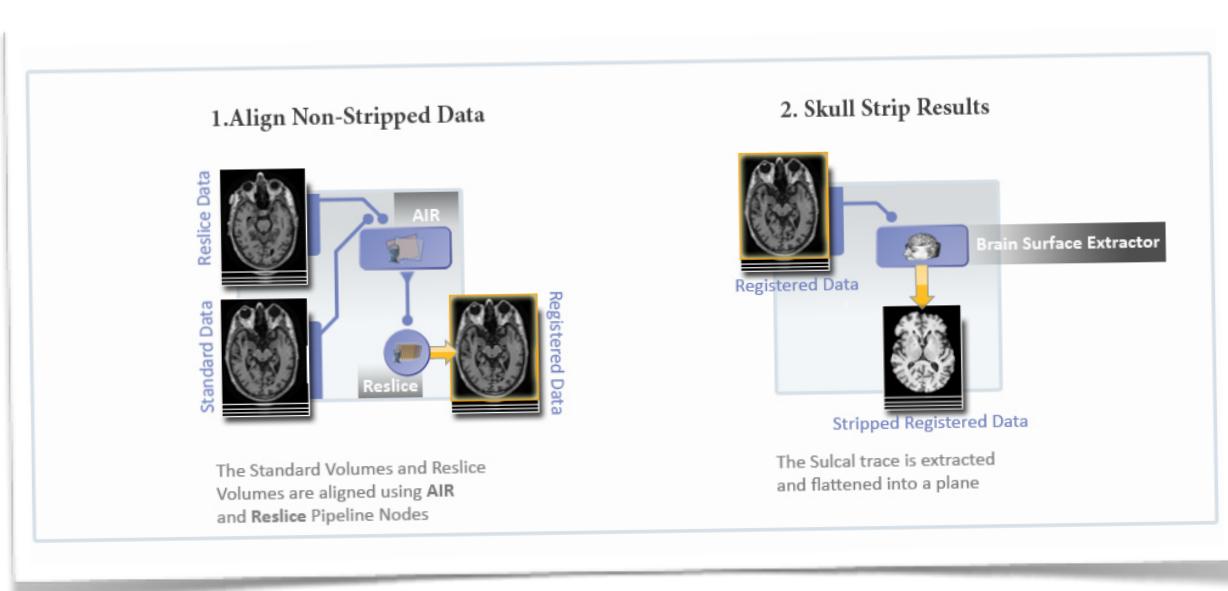
# AIR-BrainSuite Heterogeneous Workflow

## Introduction

The sequence include registering the source volume to the target space and skull-stripping the source data.

### Overview

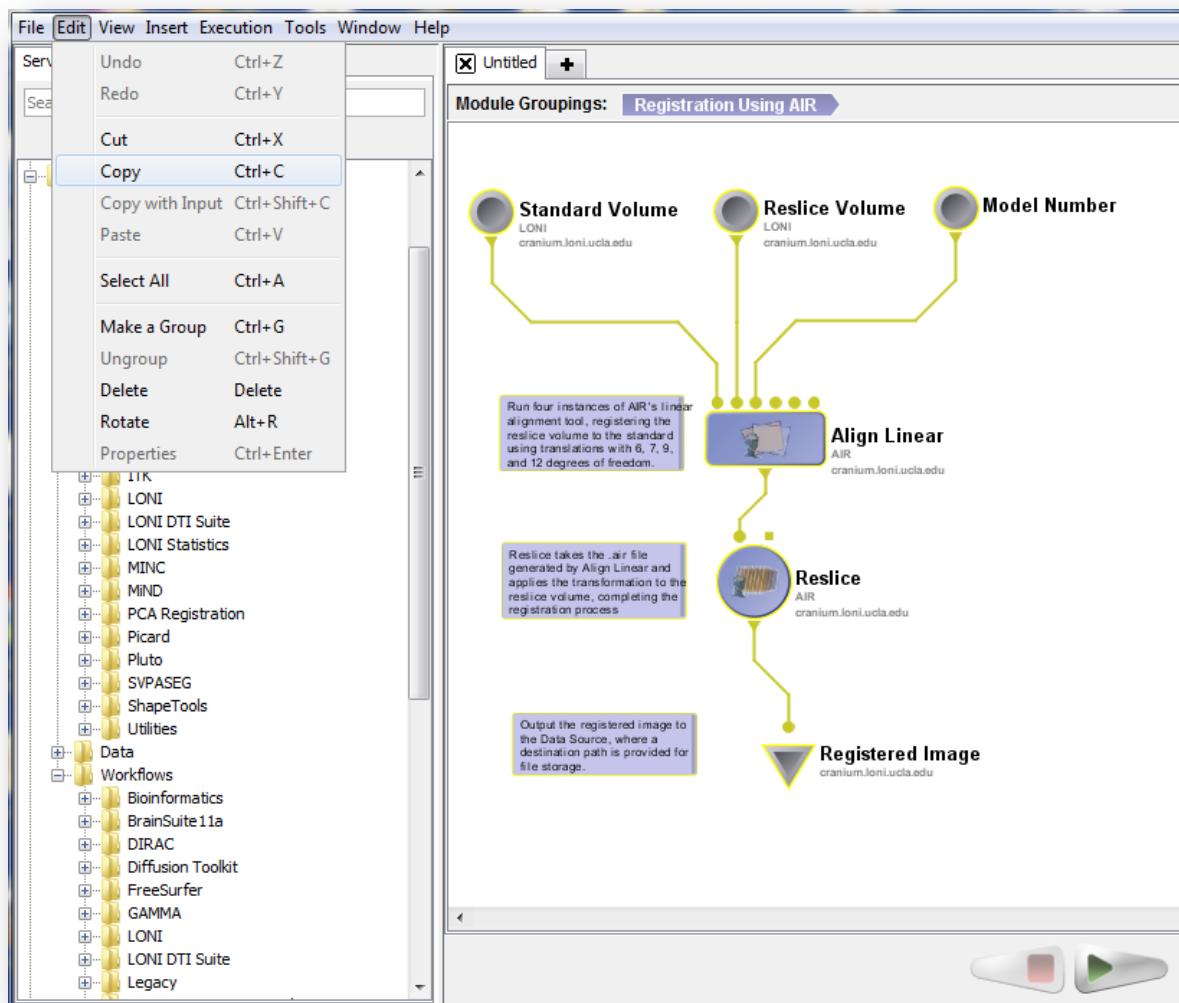
In order to demonstrate the versatility of the Pipeline, we will now build a hybrid of the previous two workflows. This example will showcase the potential for heterogeneity of image analysis tools within a Pipeline workflow. Also, it will present an often useful approach to putting together a workflow, in that we will take parts of already existing workflows and combine them in a meaningful way to perform a new type of analysis. The steps below will further clarify this notion.



# AIR-BrainSuite Heterogeneous Workflow

## Copy AIR Workflow

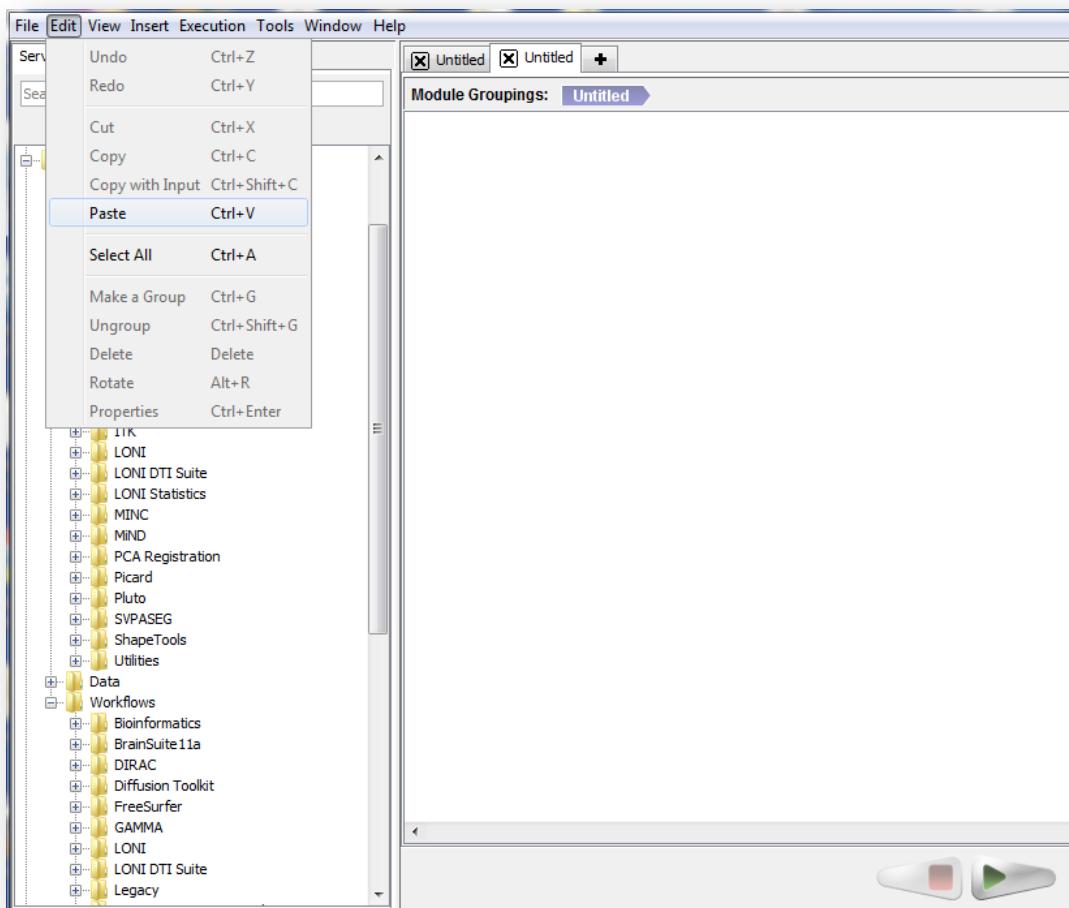
- 1) Open the AIR Registration workflow. Select all elements of this workflow by clicking and dragging across the entire canvas.  
From the Edit menu, select Copy.  
From the File menu, select New.



# AIR-BrainSuite

## Heterogeneous Workflow

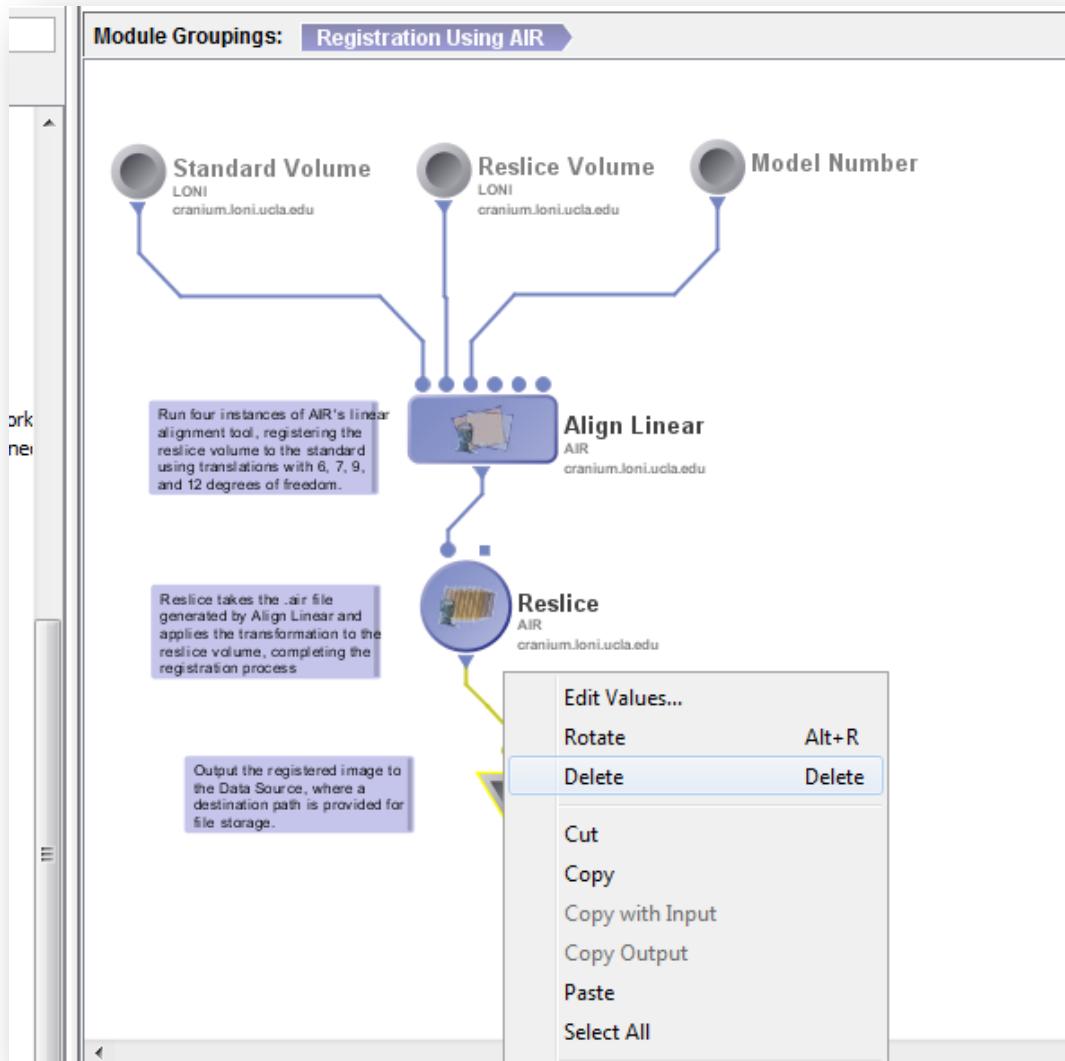
### Paste into Empty Canvas



From the Edit menu, select Paste. A shortcut to these four steps is: ctrl-a (select all), ctrl-c (copy), ctrl-n (new workflow), ctrl-v (paste)

# AIR-BrainSuite Heterogeneous Workflow

## Remove Unnecessary Components

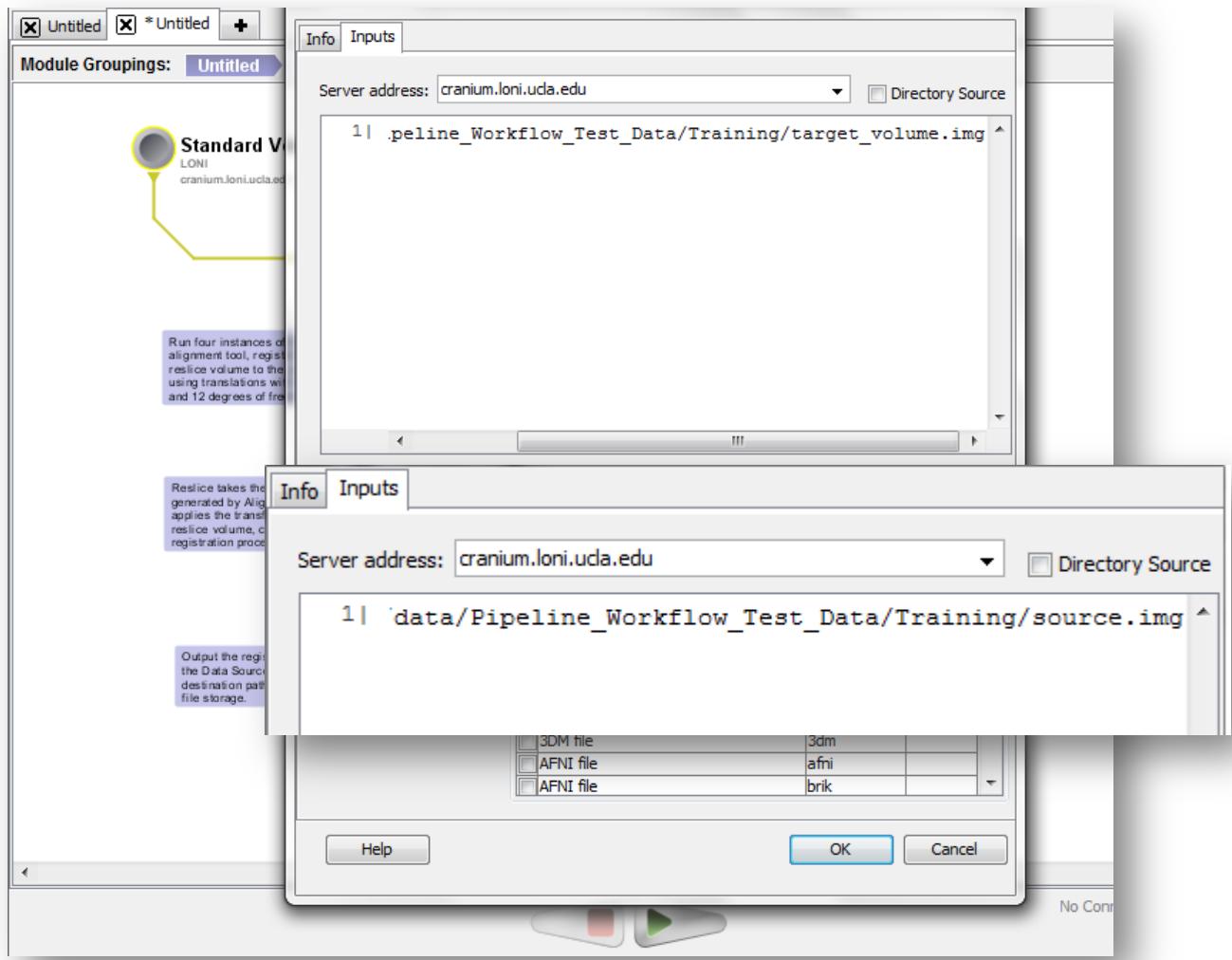


- 2) We can remove the LONI Viewer and the data sink, since we will be adding another component at the end of this workflow. Select these two modules by clicking and dragging across them. Hit delete to remove.

# AIR-BrainSuite

## Heterogeneous Workflow

### Change Inputs

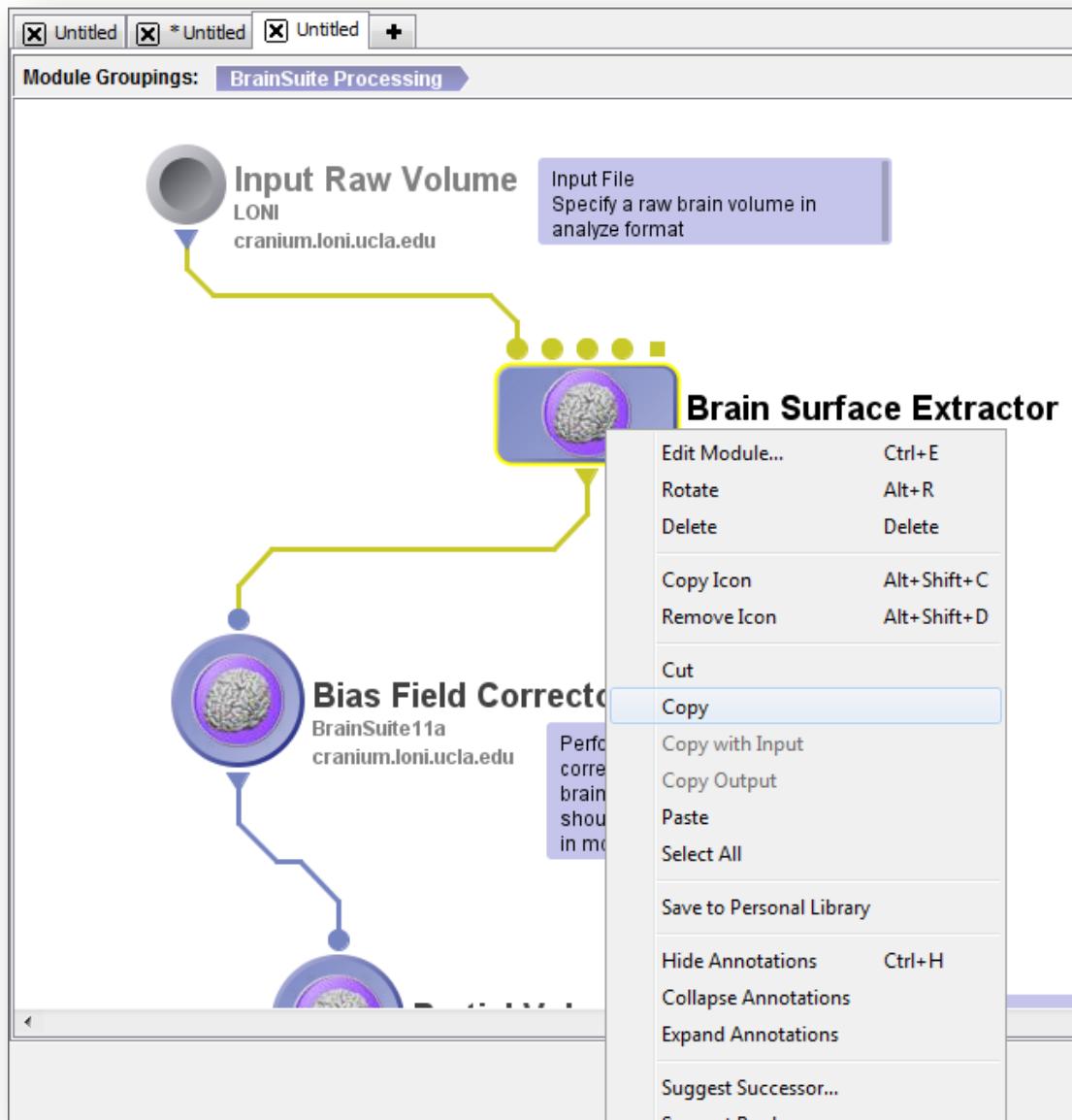


- 3) Also, we will change the inputs to the workflow. The standard volume will be changed to  
/usr/local/loniData/Pipeline\_Workflow\_Test\_Data/Training/target\_volume.img  
and the Reslice Volume will now be  
/usr/local/loniData/Pipeline\_Workflow\_Test\_Data/Training/source.img. The Model Number data source will now contain the number 6.

# AIR-BrainSuite

## Heterogeneous Workflow

### Copy Brain Surface Extractor

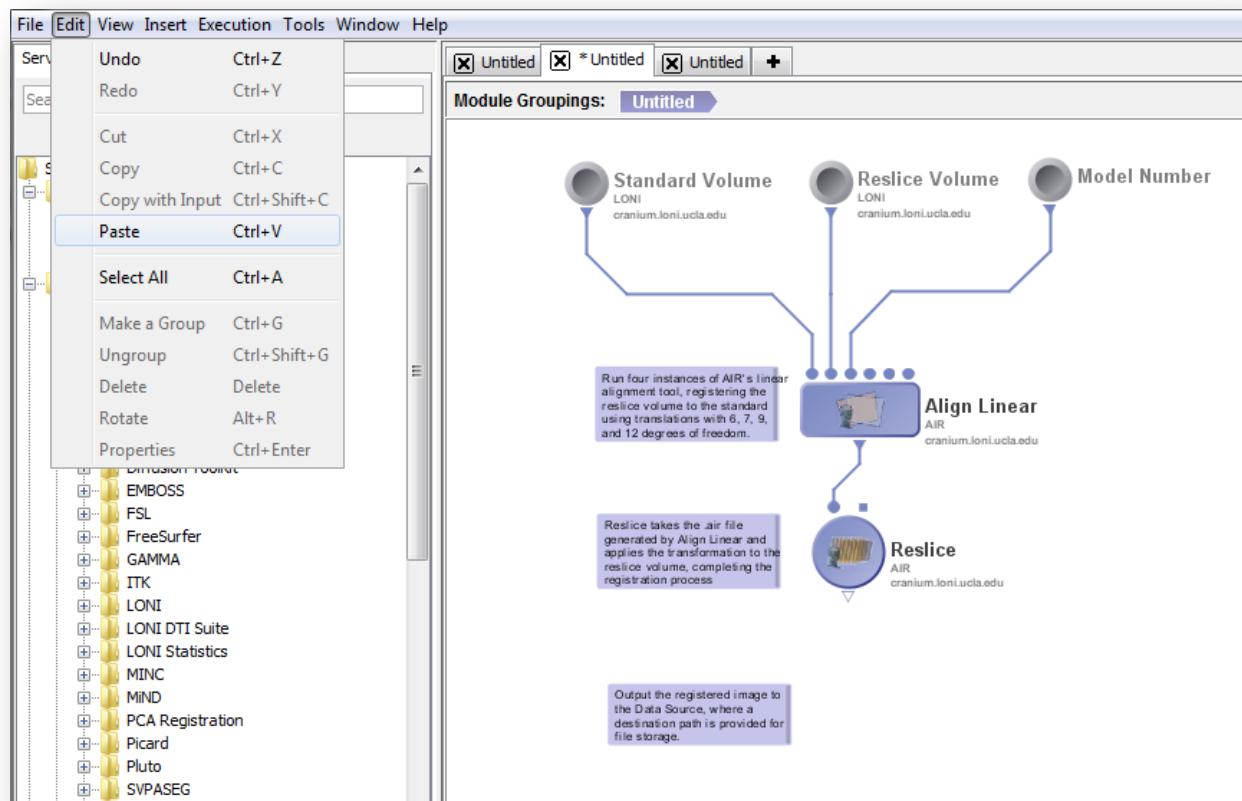


- 4) Open the BrainSuite Processing workflow. Click on the Brain Surface Extractor and select Edit -> Copy (alternatively, ctrl-c).

# AIR-BrainSuite

## Heterogeneous Workflow

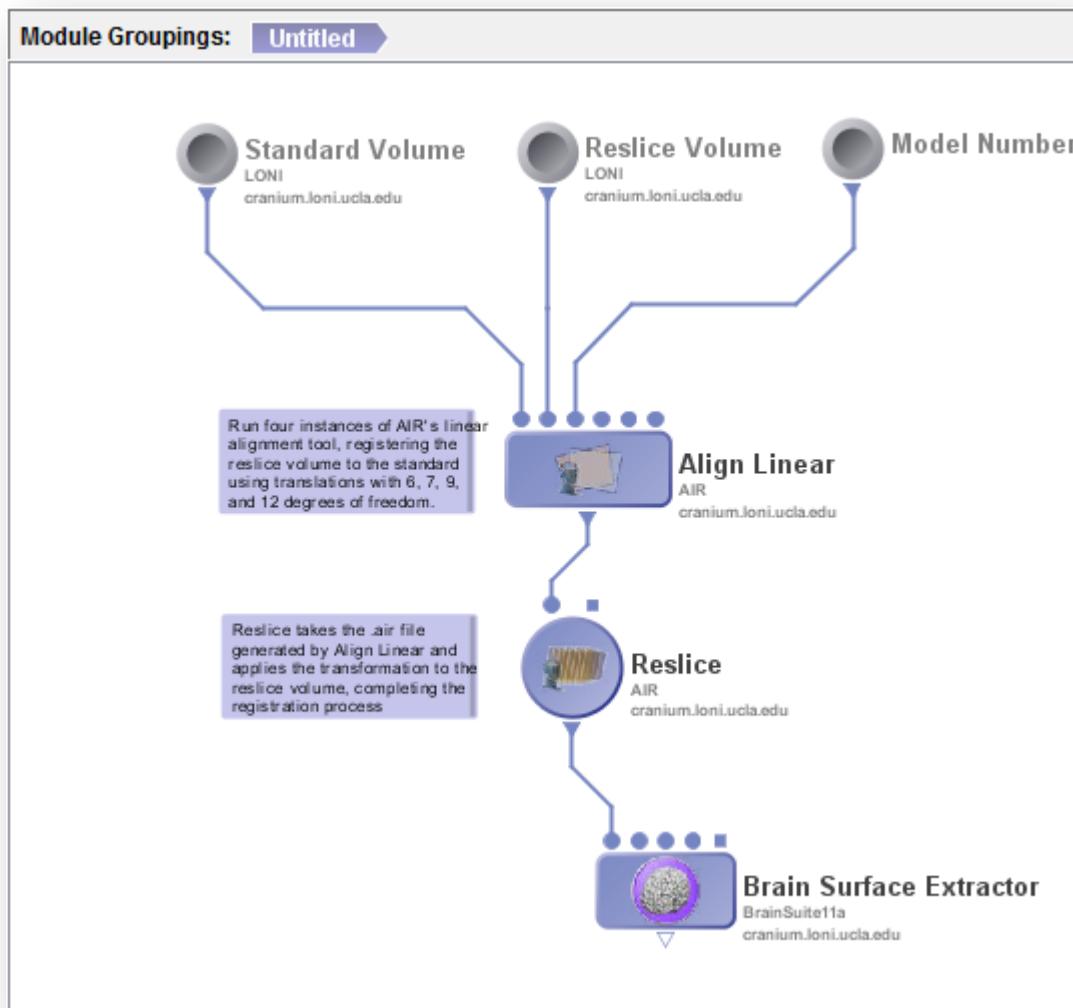
### Paste BSE Module into New Workflow



- 5) Switch to the new workflow window (labeled Untitled).  
Select Edit -> Paste (or ctrl-v).

# AIR-BrainSuite Heterogeneous Workflow

## Connect Reslice and BSE

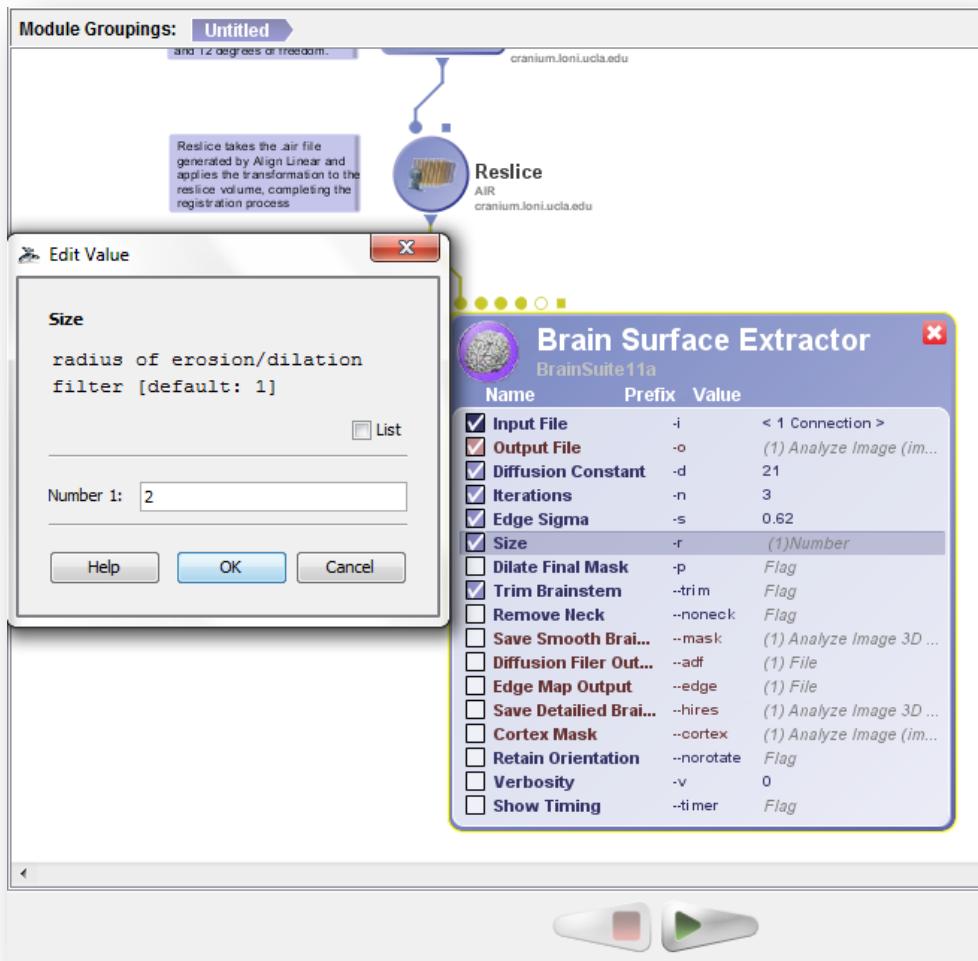


- 6) Connect the output of Reslice to the Input File parameter of Brain Surface Extractor.
- 7) Double-click on the third circle above the Brain Surface Extractor. Change the argument to the Diffusion Constant parameter to 21.

# AIR-BrainSuite

## Heterogeneous Workflow

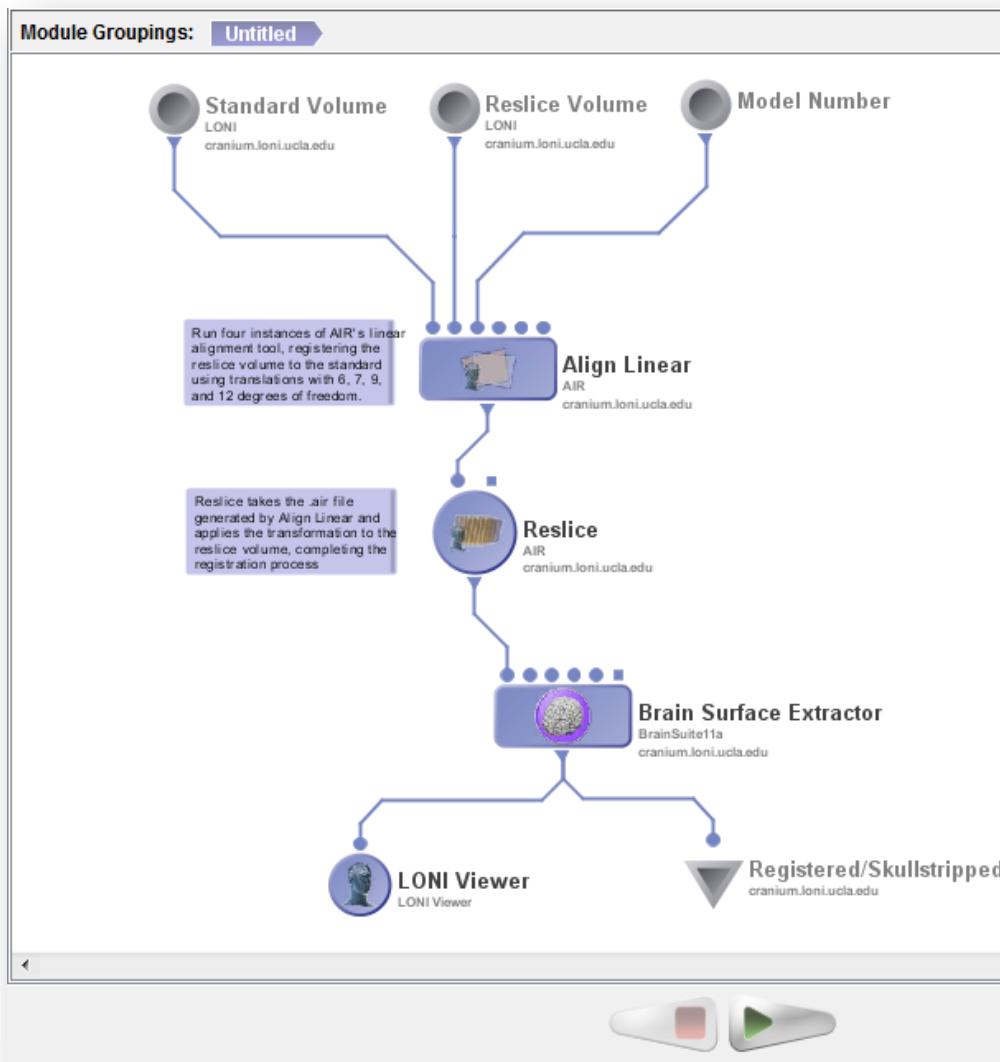
### Change BSE Parameters



- 8) Double-click on the Brain Surface Extractor module and enable the Size parameter. Set the argument to this parameter to 2.

# AIR-BrainSuite Heterogeneous Workflow

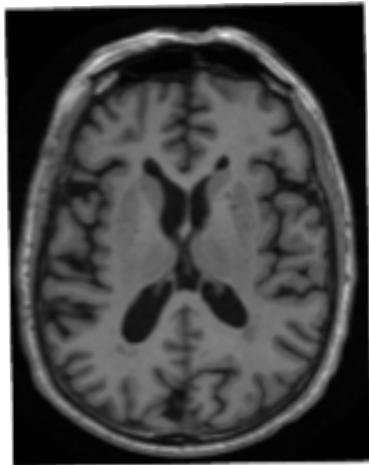
## LONI Viewer and Data Sink



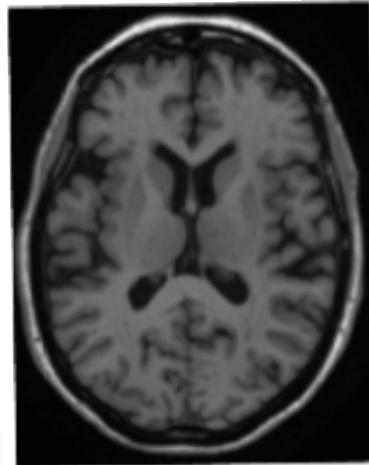
- 9) Create a LONI Viewer module and connect it to Brain Surface Extractor.
- 10) Create a data sink called "Registered/Skullstripped" and specify /ifs/tmp/pipelineTraining/{\$username}/registered\_skullstripped.img as the path of the output. Make sure to select cranium.loni.ucla.edu as the server and Analyze Image as the acceptable filetype.

# AIR-BrainSuite Heterogeneous Workflow

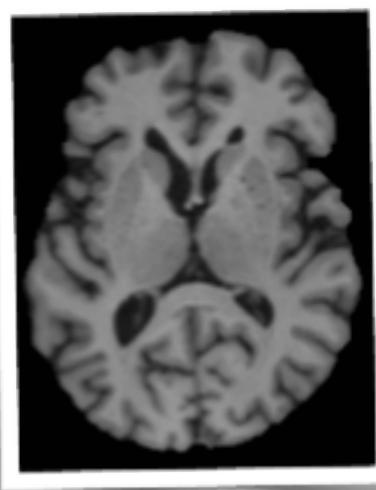
## Results



Raw Input



Target Volume



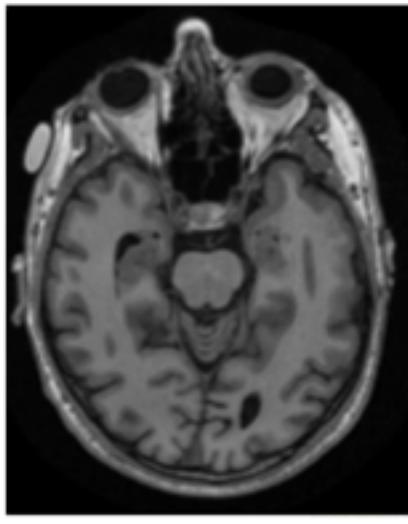
Registered and  
Skull-stripped  
Results

The Raw Input has been aligned to the Target Volume and skull-stripped. The result appears to the left.

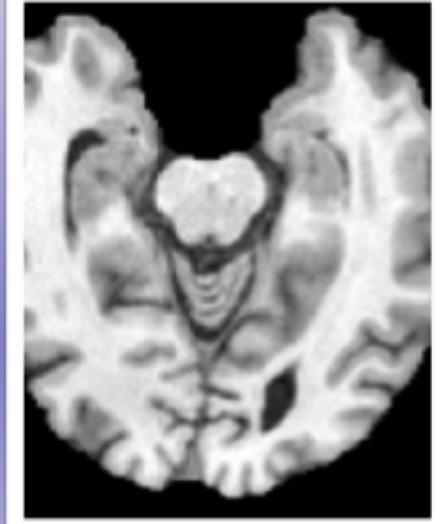
# MNC-BrainSuite Heterogeneous Workflow

## Introduction

This is a simple skull-stripping workflow for data that are in MINC format.



Raw Input (MINC)



Skull-stripped Brain (MINC)

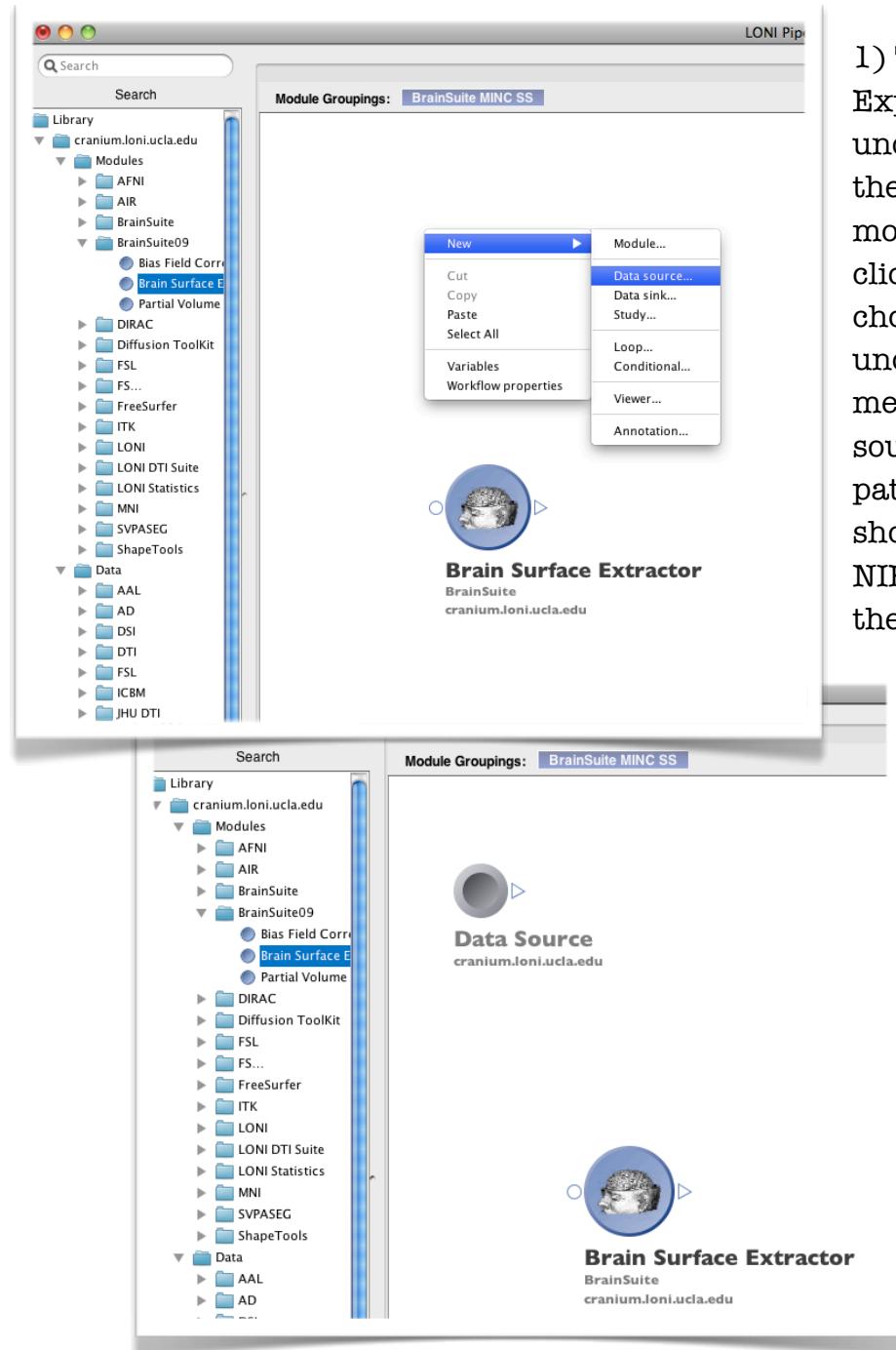
### Overview

Brain Surface Extractor (BSE) skull stripping algorithm and the new Smartline feature is used in this workflow. Brain Surface Extractor (BSE) requires the input data to be either a NIfTI or an analyze file format. MRI volumes that are in MINC file format are automatically converted between different file formats using the new Smartline feature to match the requirement of BSE.

# MNC-BrainSuite

## Heterogeneous Workflow

### Copy Brain Surface Extractor and Specify the Input



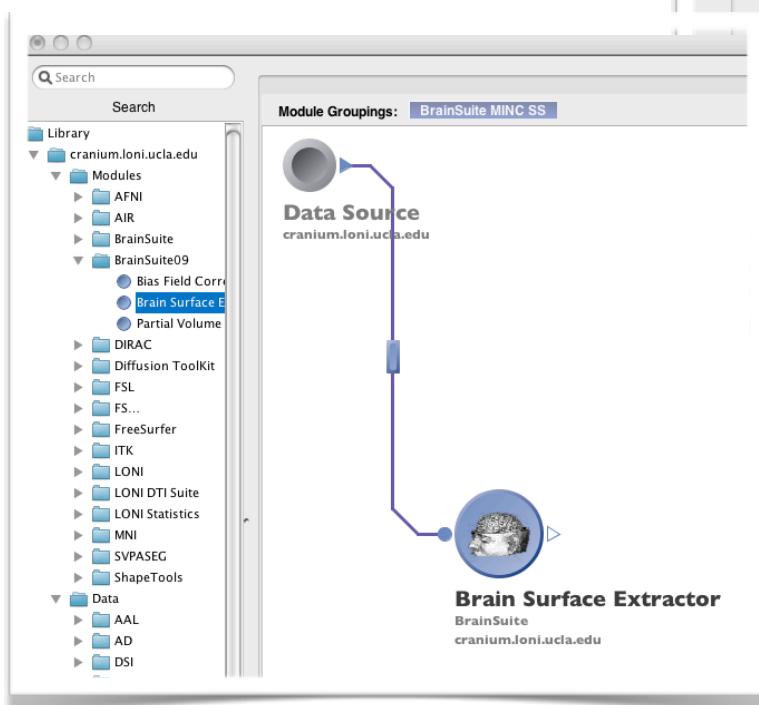
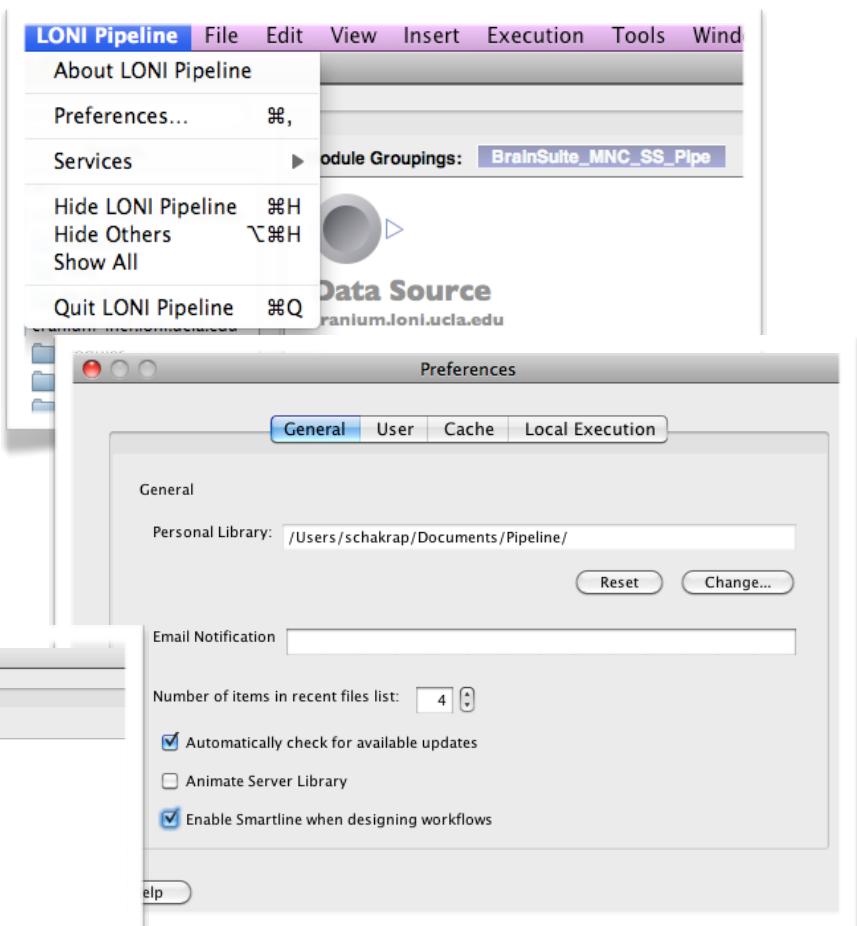
1) To begin, open a new workflow. Expand the BrainSuite folder under the library, click and drag the Brain Surface extractor module to the workflow. Right click on the empty canvas and choose the “data source” option under new from the drop down menu. Double click on the data source and browse or type in the path to the input files. These files should be in ANALYZE, MINC, or NIFTI format. Specify which of these formats you are using by checking the appropriate box under ‘File types’.

# MNC-BrainSuite

## Heterogeneous Workflow

### Enable Smartline Feature

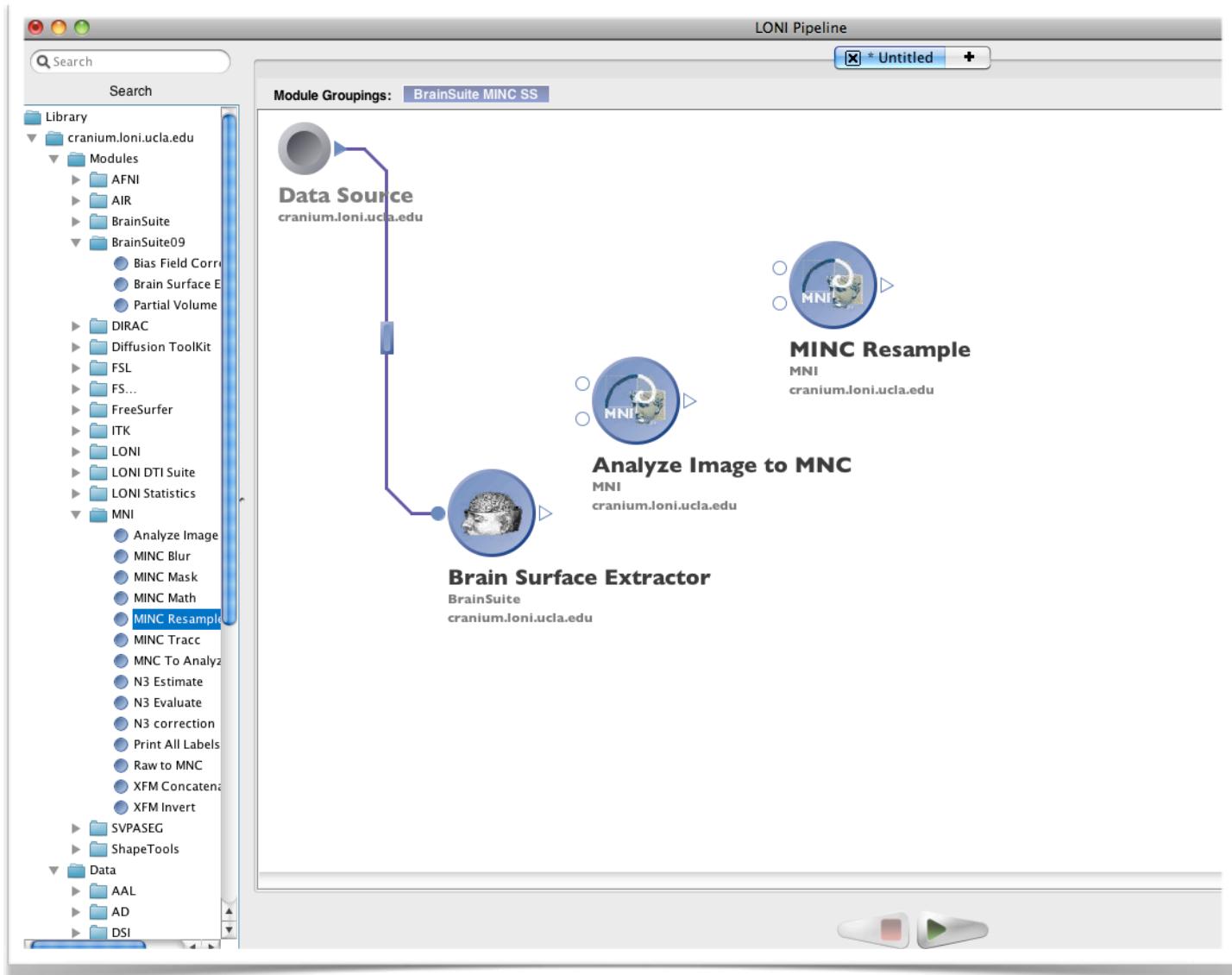
- 2) Connect the data source to the input of BSE. To enable the Smartline feature, choose Preferences under the LONI Pipeline option (on the top tool bar). Under the general tab, check the box that is next to “Enable Smartline when designing workflows” option. You could also enable the Smartline connection by holding on the shift button



while drawing a connection between the modules. Smartline recognizes the input file format that BSE expects and it automatically converts the raw input data to the desired format.

# MNC-BrainSuite Heterogeneous Workflow

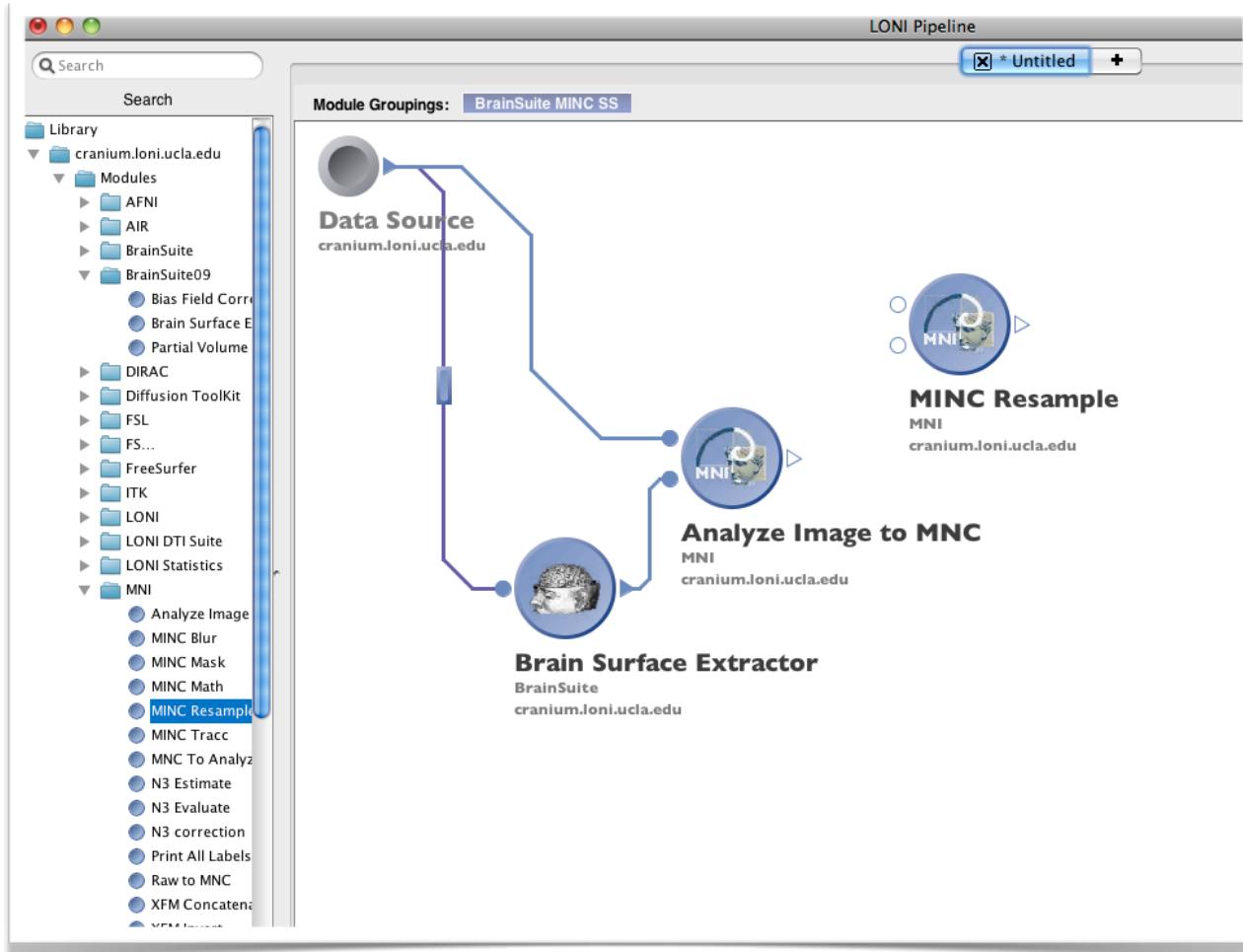
Copy the MINC tools



- 3) Once the raw input files are skull stripped, it can be converted back to MINC file format. To do this, expand the MNI folder under the library. Click and drag Analyze Image to MNC and MINC Resample modules.

# MNC-BrainSuite Heterogeneous Workflow

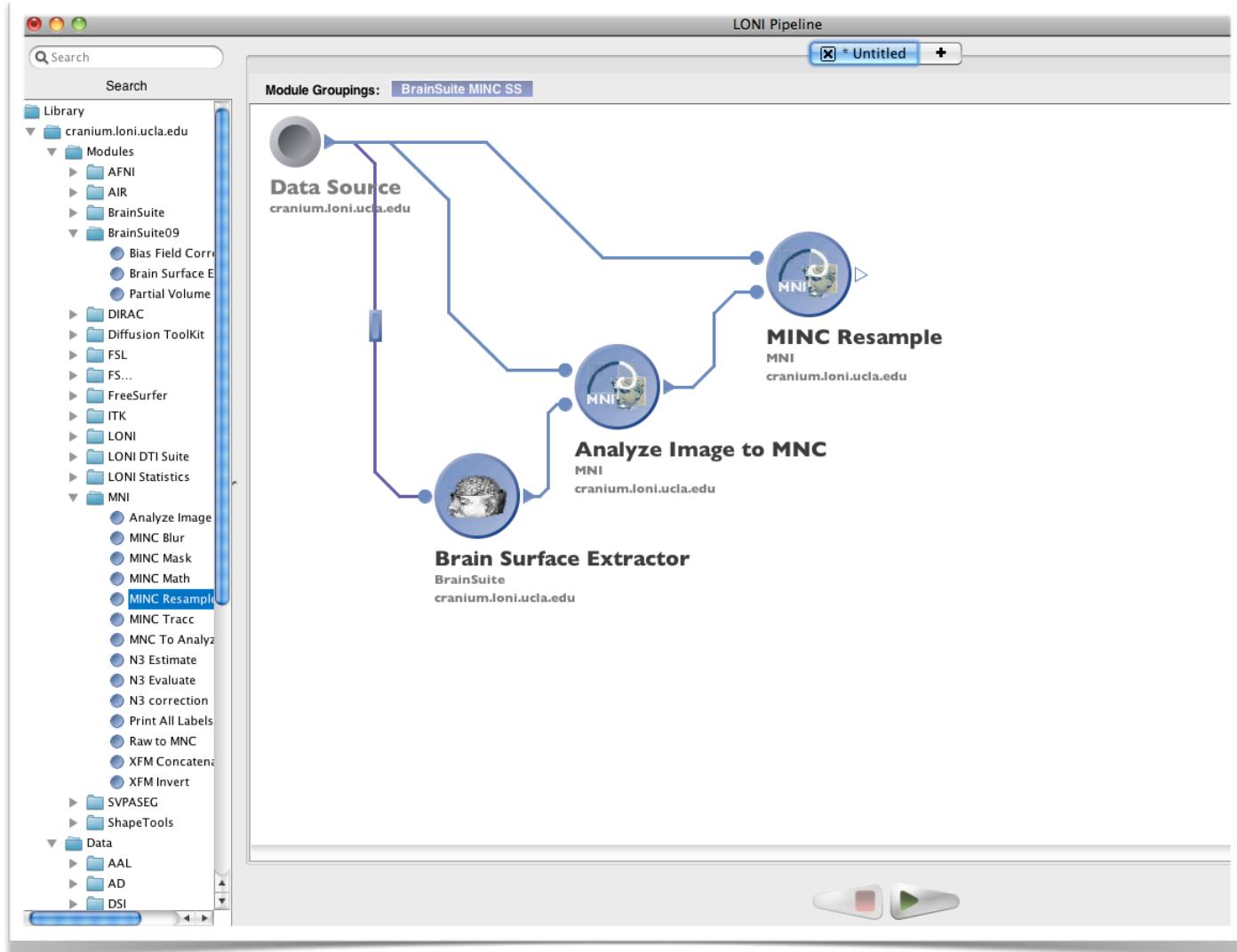
Connect BSE to Analyze Image to MNC  
Module



- 4) Connect the output of BSE to the 'Input Analyze image File' parameter and the Data Source to the 'Model Minc File' parameter of the 'Analyze Image to MNC' module.

# MNC-BrainSuite Heterogeneous Workflow

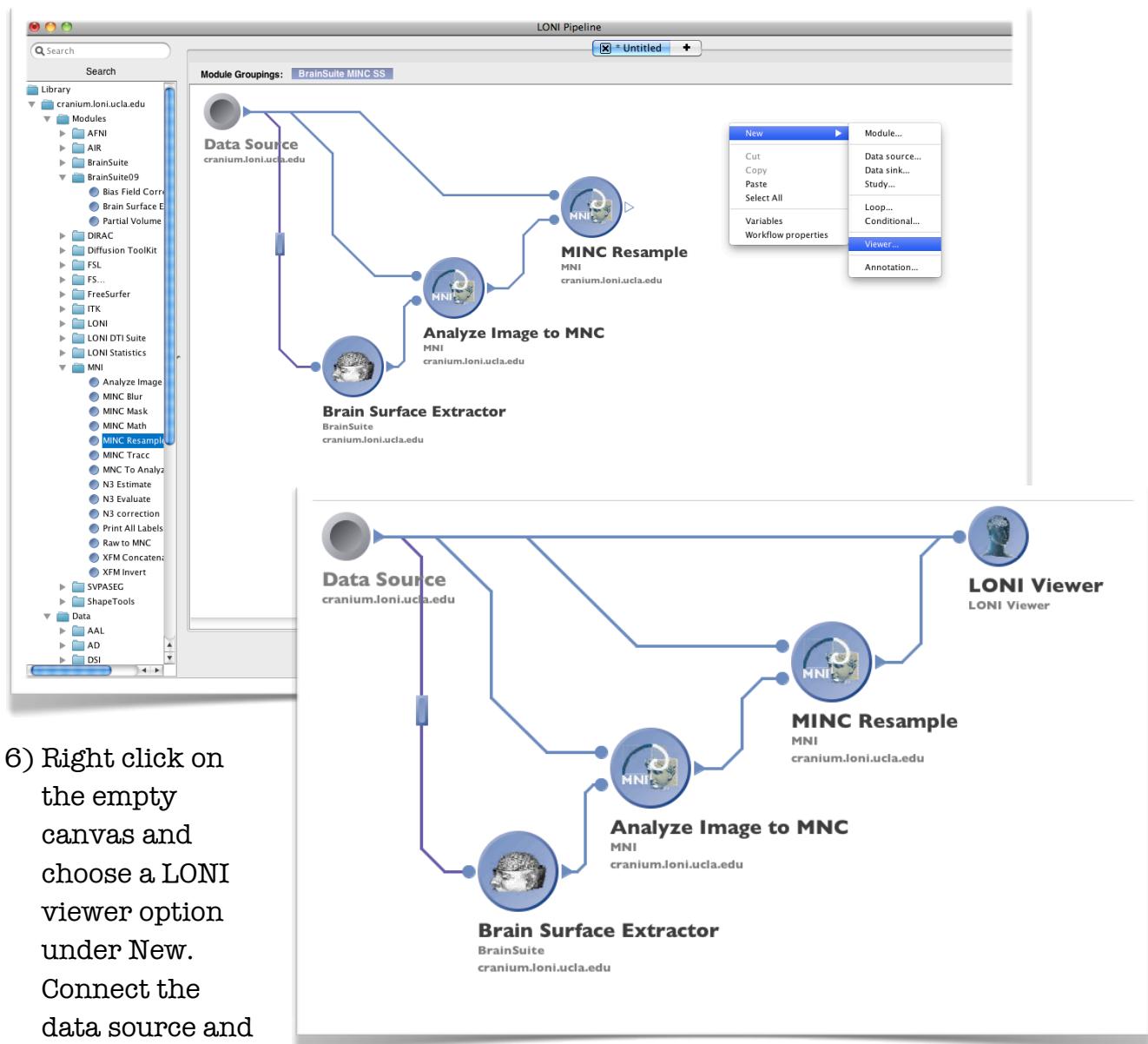
## Connect the Data source to MINC Resample Module



- 5) Similarly, the ‘Analyze Image to MNC’ module and the Data Source are connected to the ‘Input’ and ‘Like’ parameters, respectively, of the ‘MINC Resample’ module.

# MNC-BrainSuite Heterogeneous Workflow

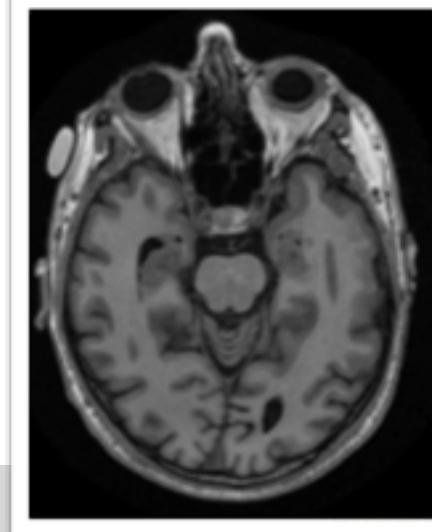
## LONI Viewer



- 6) Right click on the empty canvas and choose a LONI viewer option under New. Connect the data source and the output of the MINC Resample modules to the LONI Viewer. This will enable us to view the input raw file and the output skull stripped image.

# MNC-BrainSuite Heterogeneous Workflow

## Results



Raw Input (MNC)



Skull-stripped Brain (MNC)

The result of this workflow is a skull stripped image that is in the MNC file format. MINC tools are used to preserve all the header information from the raw input file.

# Intermediate Topics

## **LONI Image Data Archive**

The LONI Image Data Archive or IDA (<http://ida.loni.ucla.edu>) is a user-friendly environment for archiving, searching, sharing, tracking and disseminating neuroimaging and related clinical data.

## **Change Server for Entire Workflow**

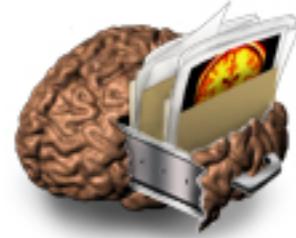
The Server Changer option in the Pipeline allows one to reset the server associated with any workflow.

# LONI Image Data Archive

## Overview

The Pipeline has the capability to utilize data from the LONI Image Data Archive or IDA (<http://ida.loni.ucla.edu>).

The Pipeline takes advantage of our cluster nodes to download files in parallel from the IDA database. This



improves download time drastically, and you don't have to keep connected to the server during the download. You can also enable metadata that metadata files will be downloaded along with data as a Study design module.

In order to establish a connection to the database, go to Tools > IDA Database. Enter your IDA username and password and click Connect. You will see on the right hand side the data that you have access to through the IDA.

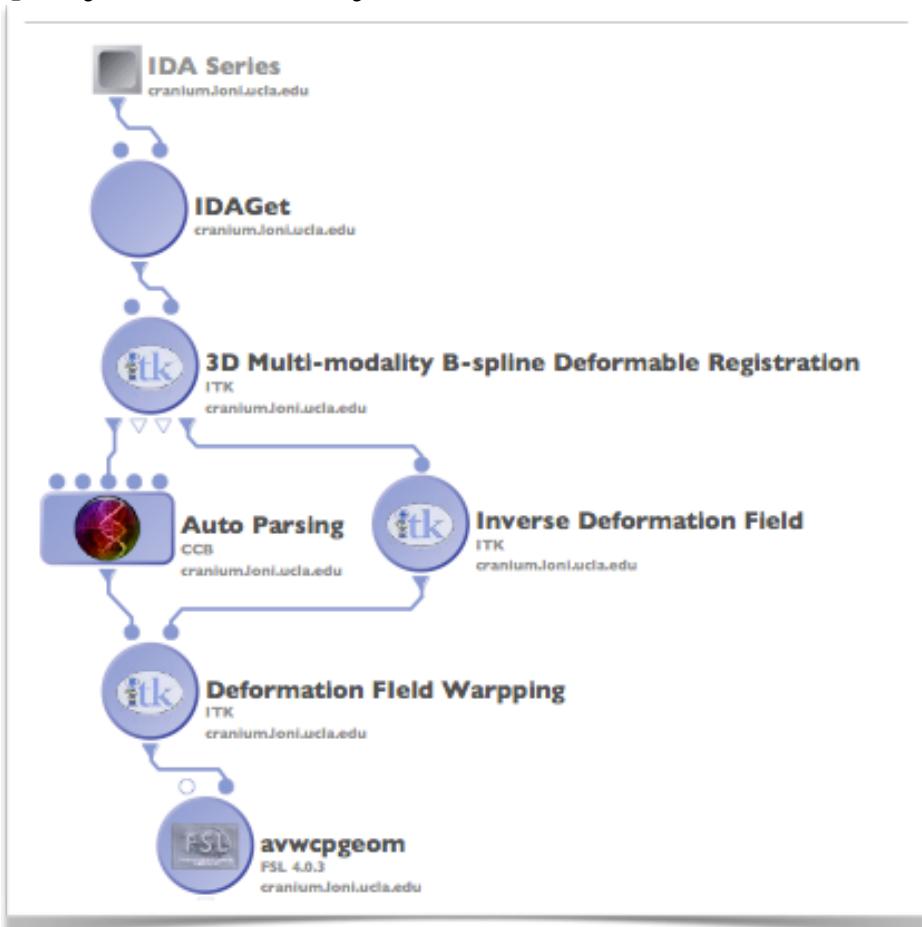
Select the files that you want to

The screenshots illustrate the process of connecting to the IDA database and selecting files for download. In the top screenshot, the user has entered their IDA username ('zhizhongliu') and selected the 'ANALYZE' file format. They have chosen to download to a 'Remote server' ('cranium.loni.ucla.edu') and checked the 'Include metadata (use Study module)' option. In the bottom screenshot, a modal dialog box titled 'IDA request successful' is displayed, indicating that the download was completed successfully. The background shows the same IDA interface with the selected files highlighted in the collection list.

process with the Pipeline, desired file format and path for the files to be downloaded. Data can be downloaded on to remote server or locally to your machine. If destination is remote, check the Remote box and specify server name. If you want to download metadata as well, check the Include metadata option and select destination server.

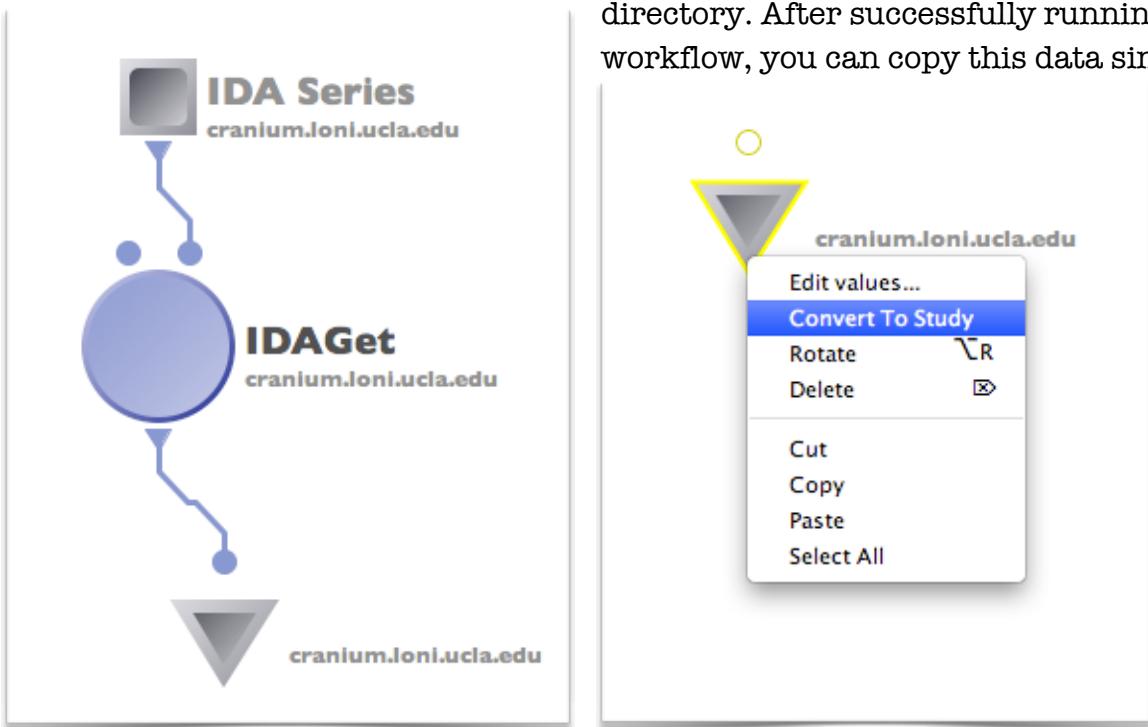


Click on Create Module, a new workflow is created. This workflow will have an IDAGet module and either a data source (if metadata is not downloaded) or a study module (if metadata is also being downloaded). Once you create this workflow the metadata files alone are downloaded to a temporary location.



You will notice that the output of the IDAGet module has the file type you specified. You can now connect this output to the input parameters of any module in your workflow and execute the workflow. As the first module of the workflow begins to execute, the data will be downloaded and will be directly passed on to the following modules. The data and metadata files downloaded are temporarily stored and will be deleted if you reset the workflow.

The data can also be downloaded to a permanent location from IDA database. To do this create a data sink and connect it to the output of the IDAGet module. You can either list the output items or use directory dump and specify an output directory. After successfully running the workflow, you can copy this data sink

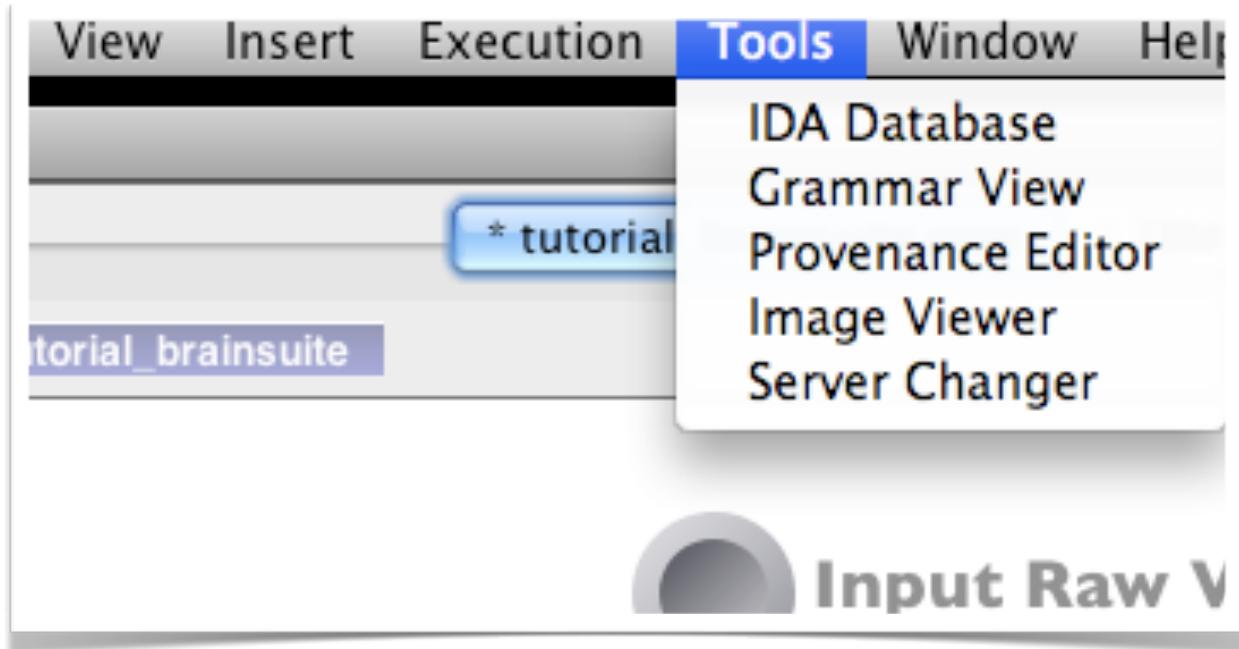


and paste it to any workflow, right click on it and choose “Convert to study” option. The data sink will be automatically converted to a study module with proper inputs. To reuse this data set, you can simply copy this study module to your workflow.

# Change Server for Entire Workflow

## Overview

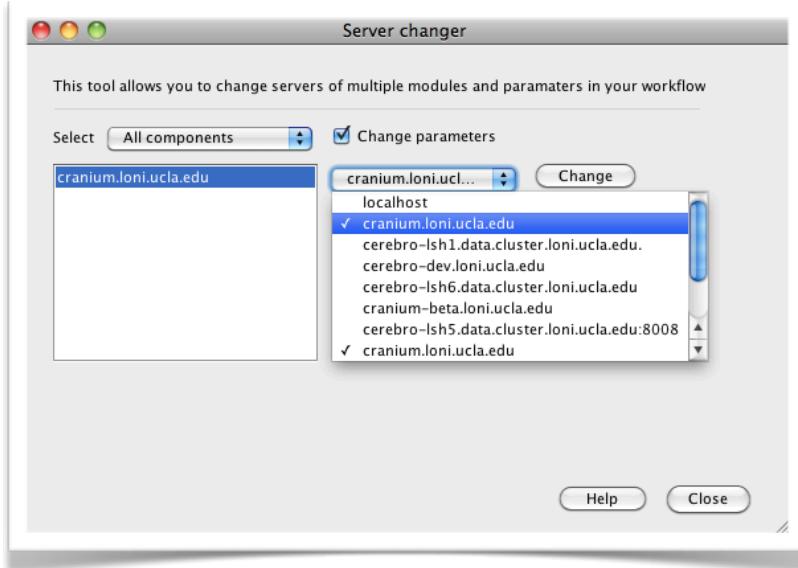
The **Server Changer** option in the Pipeline allows one to reset the server associated with any workflow. All data, binaries and workflows have servers associated with them. One of the core features of the Pipeline is the ability to have input & output files and binaries on remote servers as well as your local computer; all within a single workflow.



## How-to

- Under Tools, choose Server Changer
- Under Select, choose either All Servers, Modules, Data Sources or Data Sinks to narrow which servers to reset
- In the Select window, choose the server to be replaced

- Choose whether you'd like the change to be applied to parameters as well
- Use the drop-down menu to specify the server to be changed to
- Click Change, followed by Close



**NOTE:** You may change servers to and from servers you have connected to at least once.

# Intermediate Example Workflows

## **Skull-stripping**

Combine brain surface extraction algorithms from two independent suites in order to robustly skull-strip raw structural data.

## **fMRI First Level Analyses**

Generate first level analyses fMRI results on a single subject.

## **DTI Analyses**

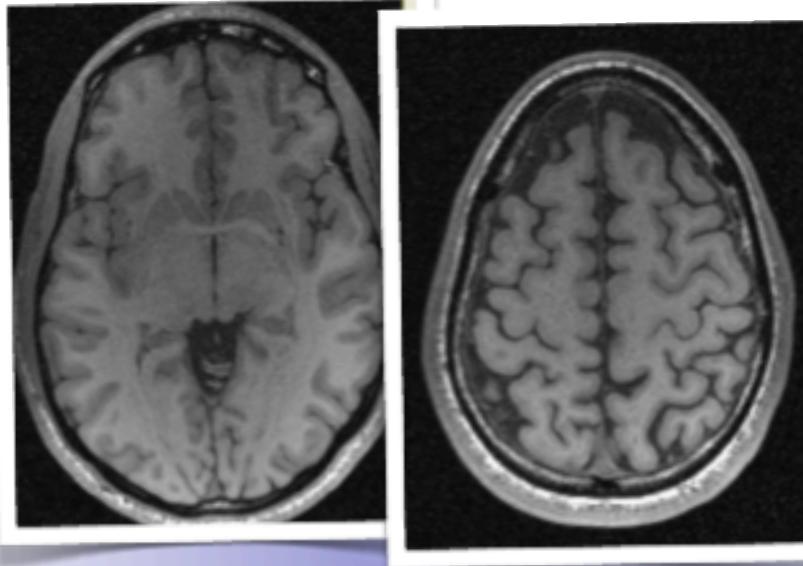
DTI (DICOM/NIfTI) images are reconstructed and fiber tracks are generated for TrackVis Software.

## **DSI Analyses**

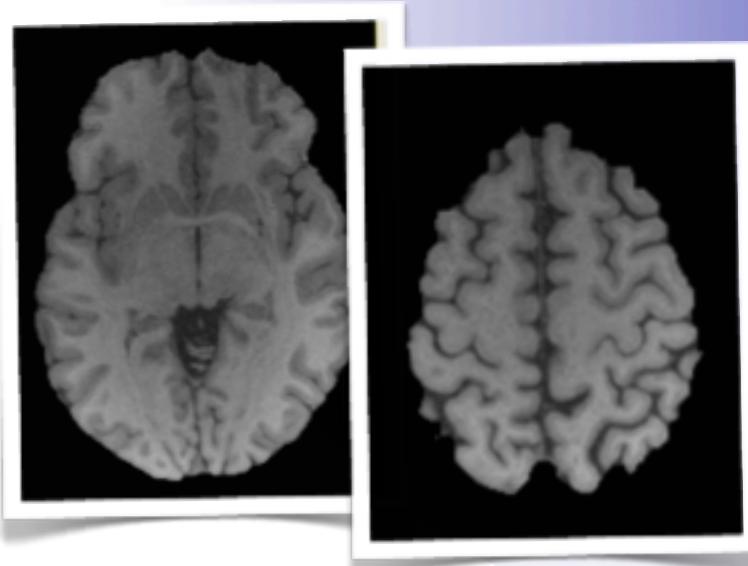
DSI (DICOM/NIfTI) images are reconstructed and fiber tracks are generated for TrackVis.

# Skullstripping

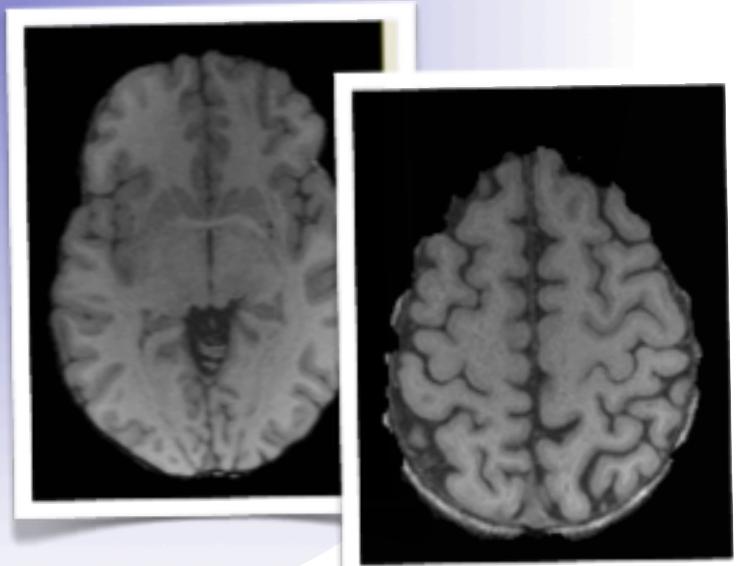
Automated Image Registration &  
BrainSuite



Raw Input



Strict

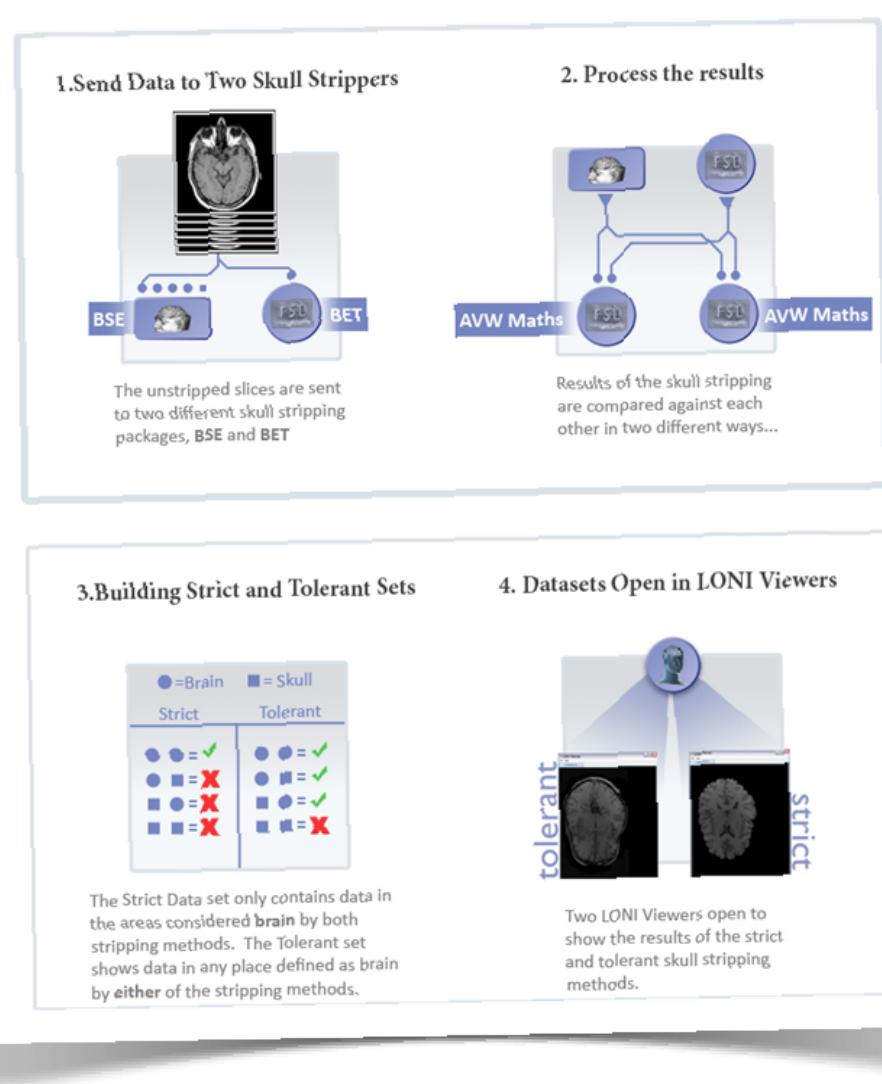


Tolerant

# Skullstripping

## Introduction

Generate whole-brain masks by running MR data through BSE and BET, then use FSL Maths to combine the results of these skull-stripping algorithms in two distinct ways.



BET, which execute unique skullstripping algorithms. The process then splits into two independent segments. In one branch, the idea is to intersect the whole-brain masks generated by these tools in order to generate a tightly skullstripped result. In the other, we union the masks to achieve more tolerant skullstripping. In this workflow, we will also make use of Smartlines, a Pipeline feature which takes care of any file format compatibility issues.

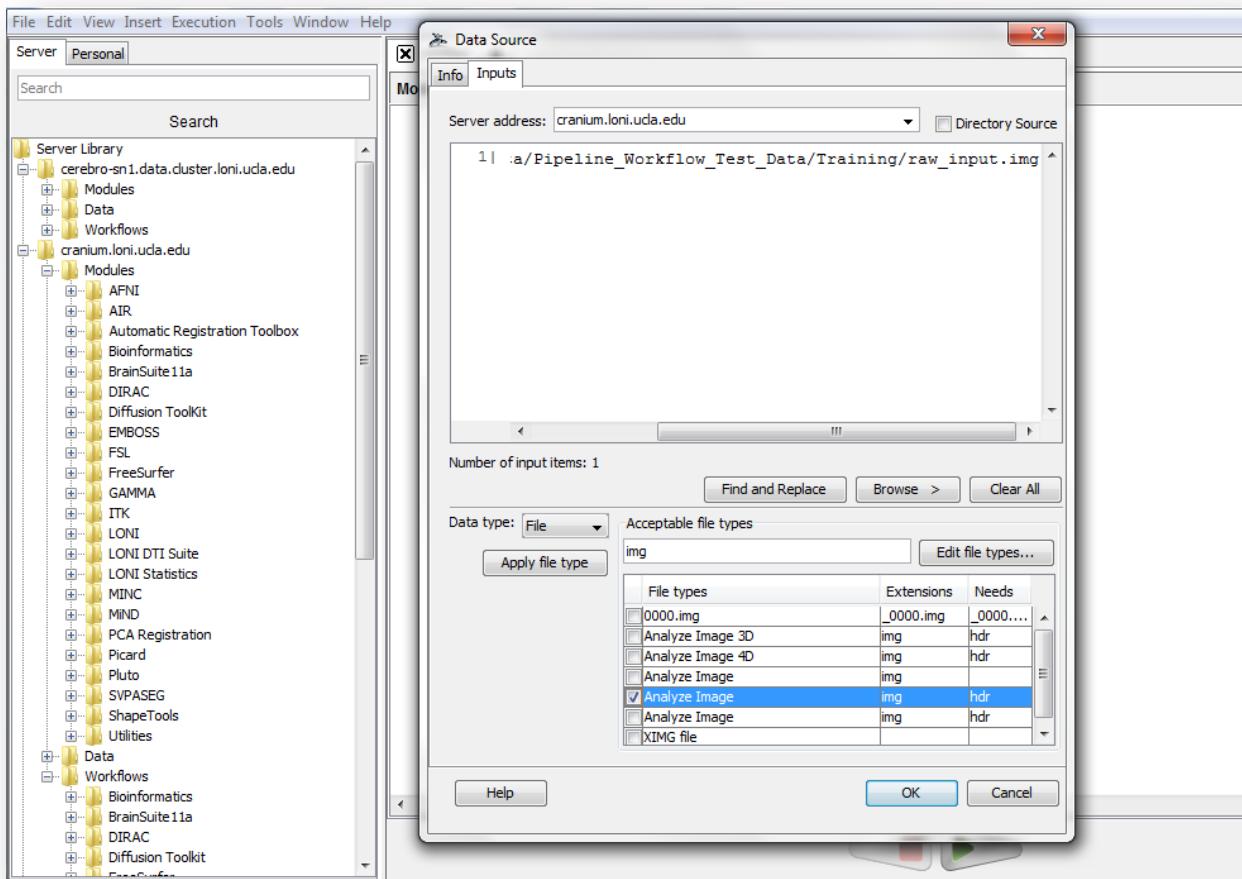
## Overview

Next, let's take a look at a slightly more involved example. This workflow combines components from the BrainSuite and FSL packages. The aim of the process is to perform skullstripping along two distinct paths and to compare the results in order to determine the better of the two methods. As a first step, we run the raw brain volume through both Brain Surface Extractor and

# Skullstripping

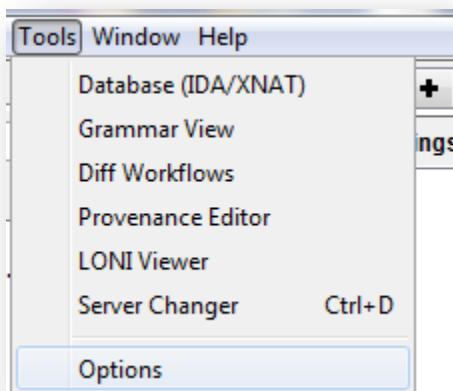
## Specify Input Data

- 1) Let's begin by defining the input file, which will be a brain volume which has not been skull stripped. We will specify this in a data source that we can label "Raw File." You can use your own file in Analyze Image format or you can make use of the one we designated, namely /usr/local/loniData/Pipeline\_Workflow\_Test\_Data/Training/raw\_input.img. If you use this file, make sure to select cranium.loni.ucla.edu from the Input Data dropdown menu.

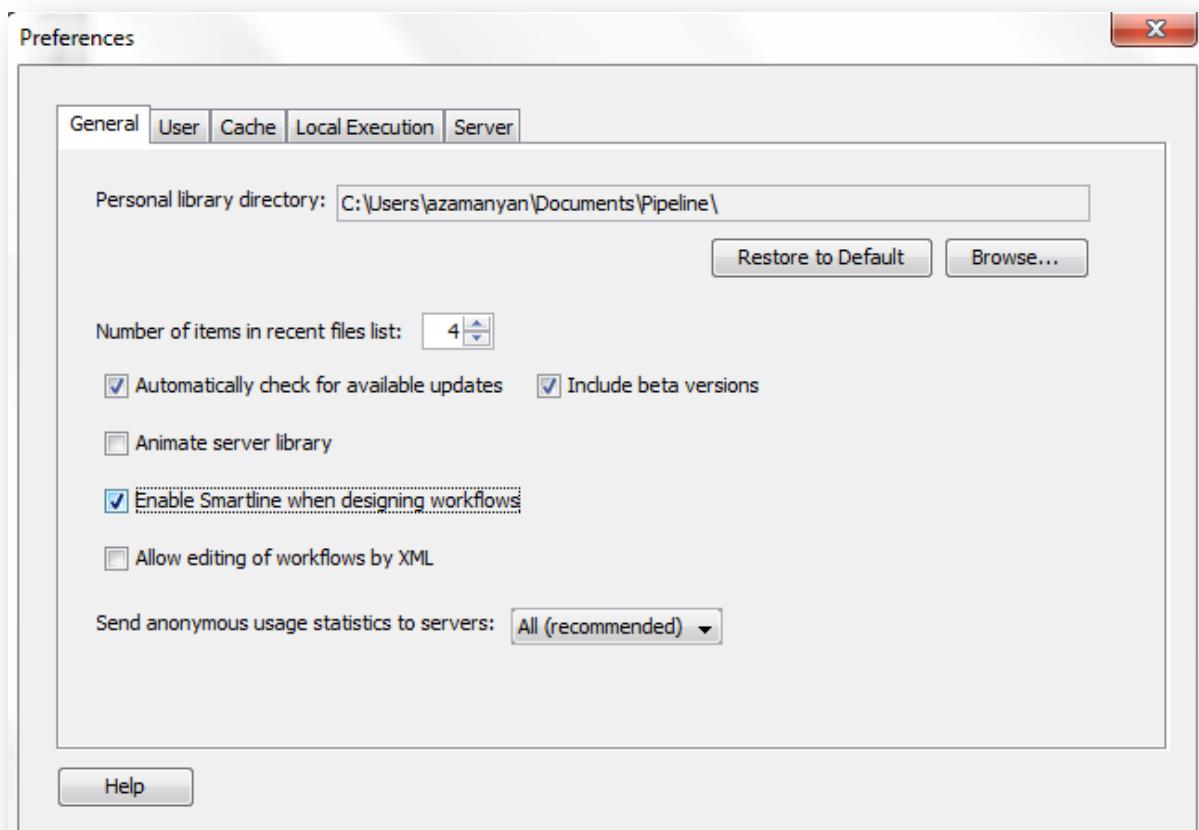


# Skullstripping

## Enable Smartlines



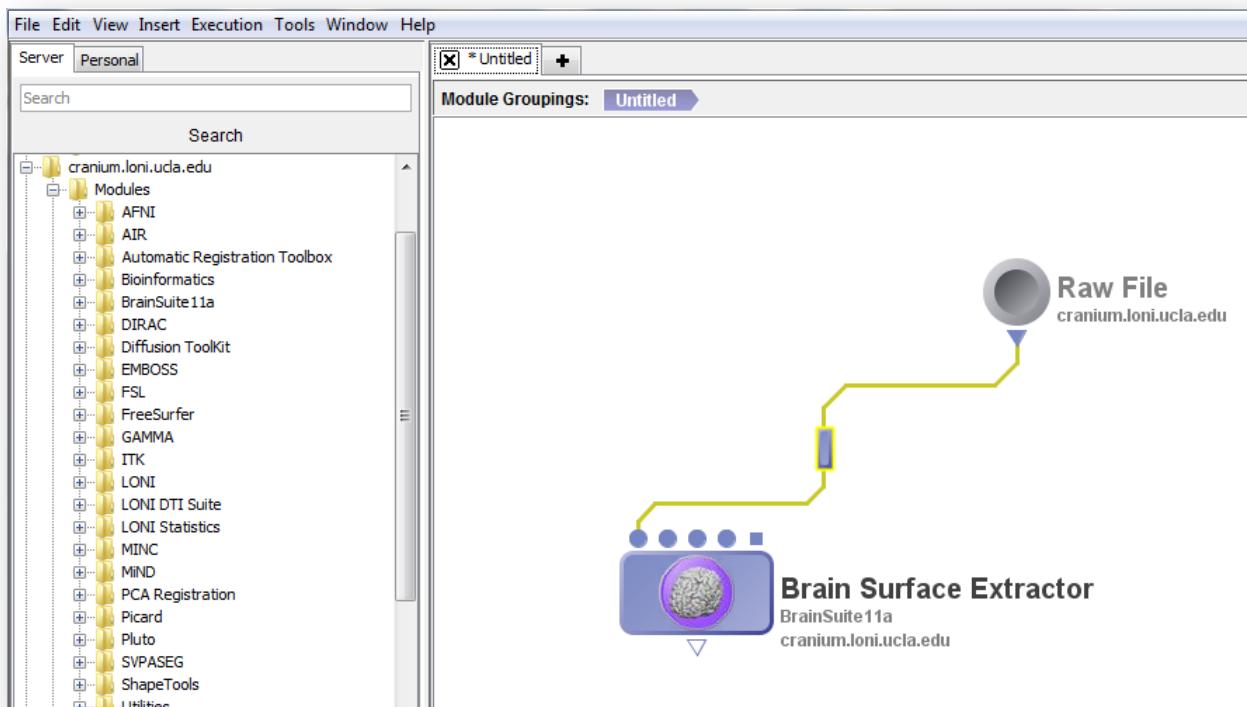
2) Next, we will make sure that Smartlines have been enabled. Open "Options" under the "Tools" menu. Check "Enable Smartline when designing workflows", if it has not already been checked.



# Skullstripping

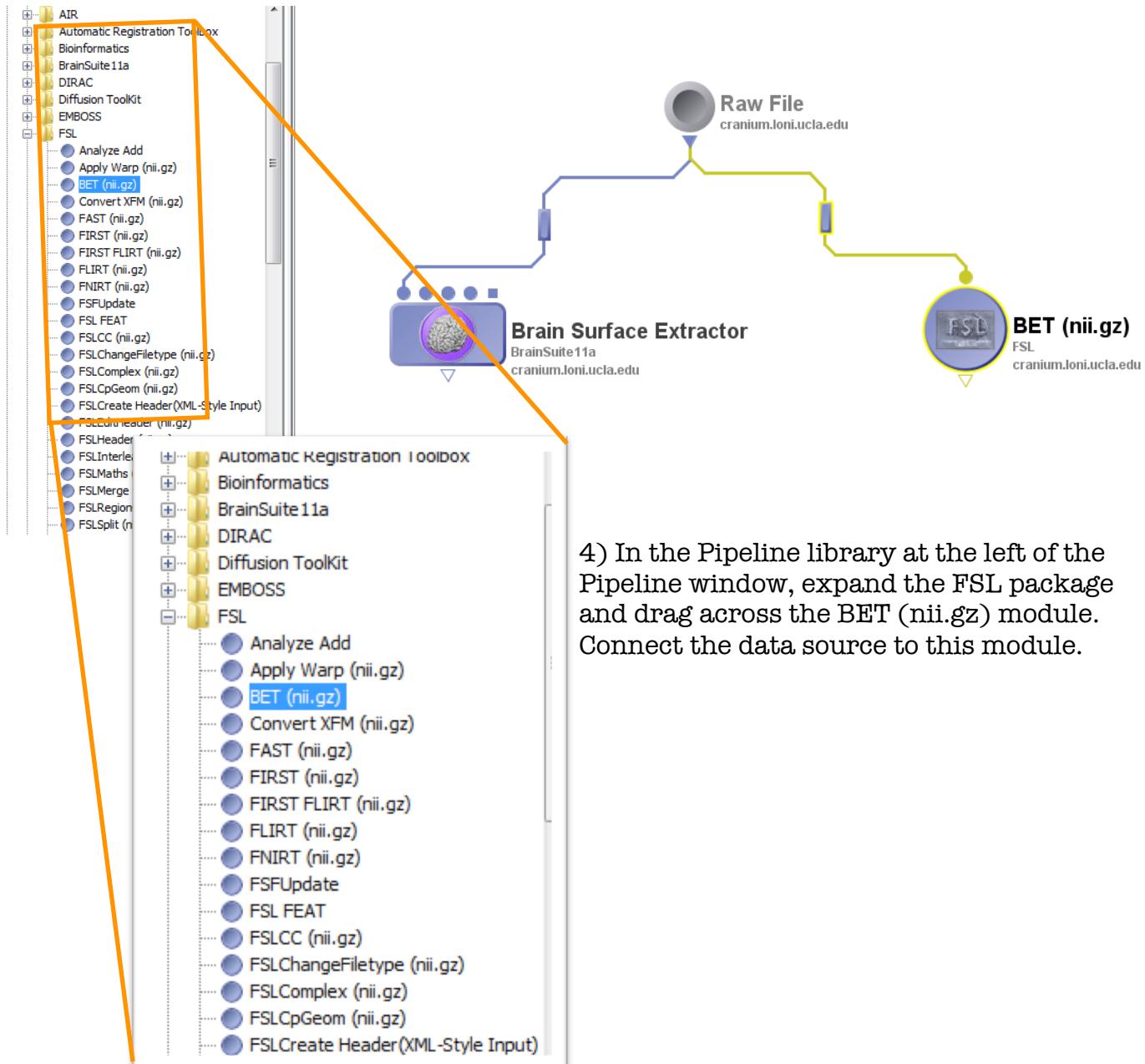
## Copy Brain Surface Extractor

3) Copy the Brain Surface Extractor module from the BrainSuite Processing workflow. Connect the data source to the Input File parameter of this module. You will see the two modules linked by a Smartline, which will handle any necessary conversions.



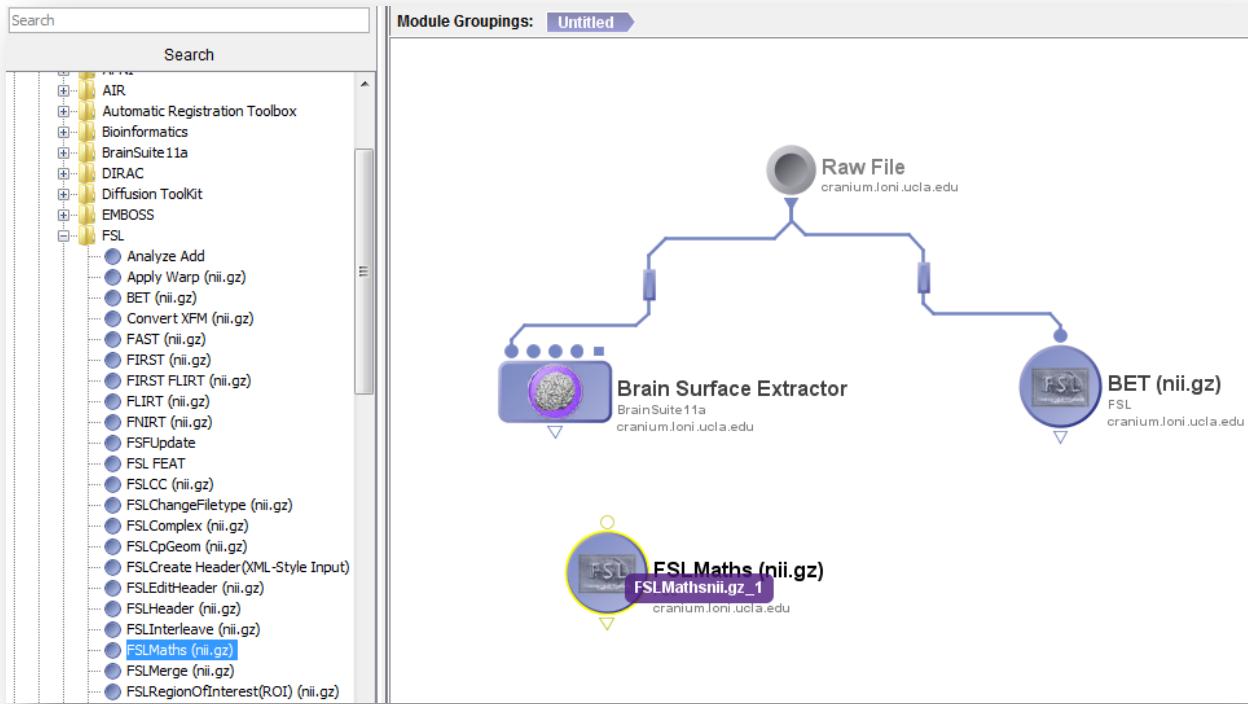
# Skullstripping

## Select BET



# Skullstripping

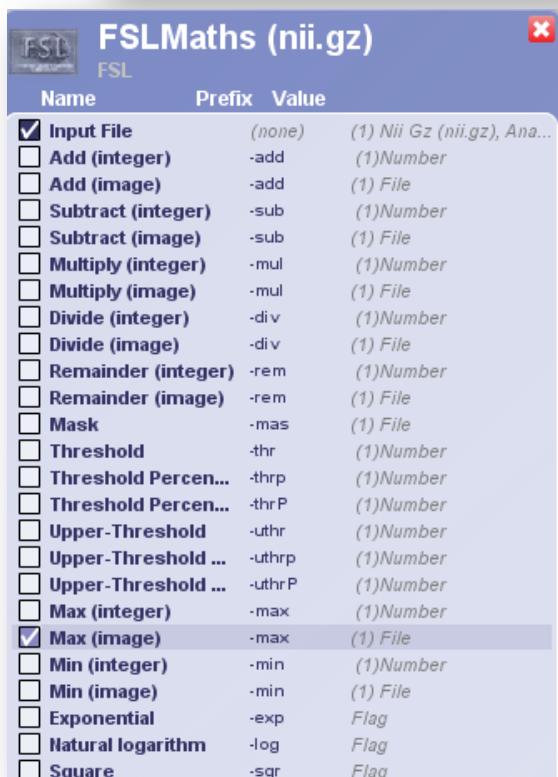
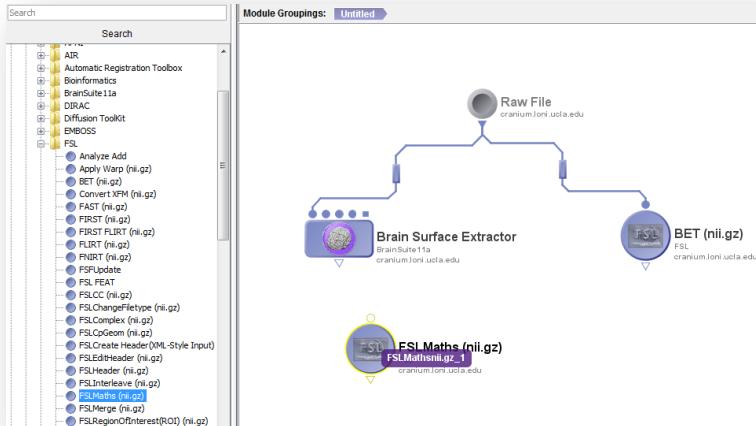
## Drag in FSL Maths



5) Next, drag in the module labeled FSL Maths (nii.gz) in the FSL package.

# Skullstripping

## Select Max (image) Parameter



6) There are many ways in which we can compute a union of skullstripped images generated in the previous step. In this case, we will use the voxelwise maximum operator of the fslmaths program. At every voxel, the greater intensity value of the two inputs will be written to the output. In this way, if a voxel is labeled as brain by at least one of the algorithms, it will be considered brain. An alternate approach would be to use two instances of Binarize, in combination with Binary Math and Binary Mask, all tools available in the Pipeline library's AIR package.

Double-click on the FSL Maths (nii.gz) module and select the parameter called "Max (image)."

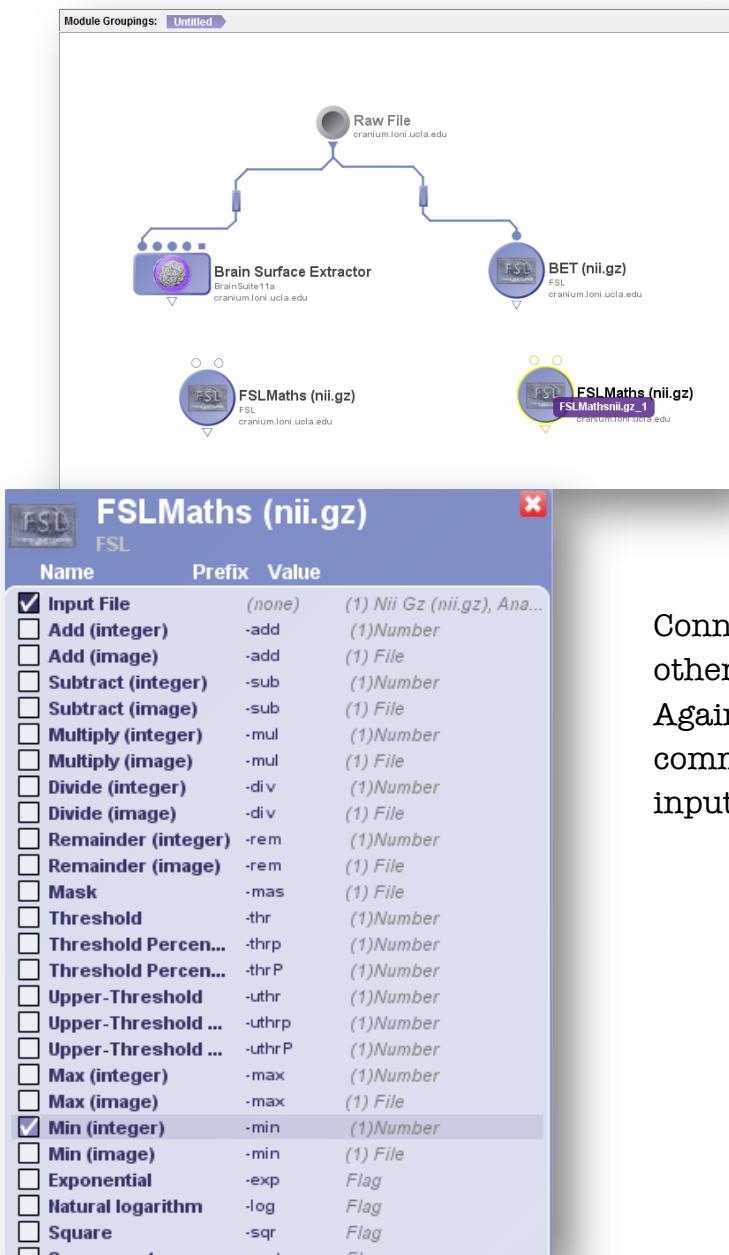
Connect the output of Brain Surface Extractor to one of the two inputs of FSL Maths (nii.gz).

Connect the output of BET (nii.gz) to the other input of FSL Maths (nii.gz).

Notice that the operation we have chosen is commutative, so the order in which the inputs are connected is irrelevant.

# Skullstripping

## Select Min (Image) Parameter

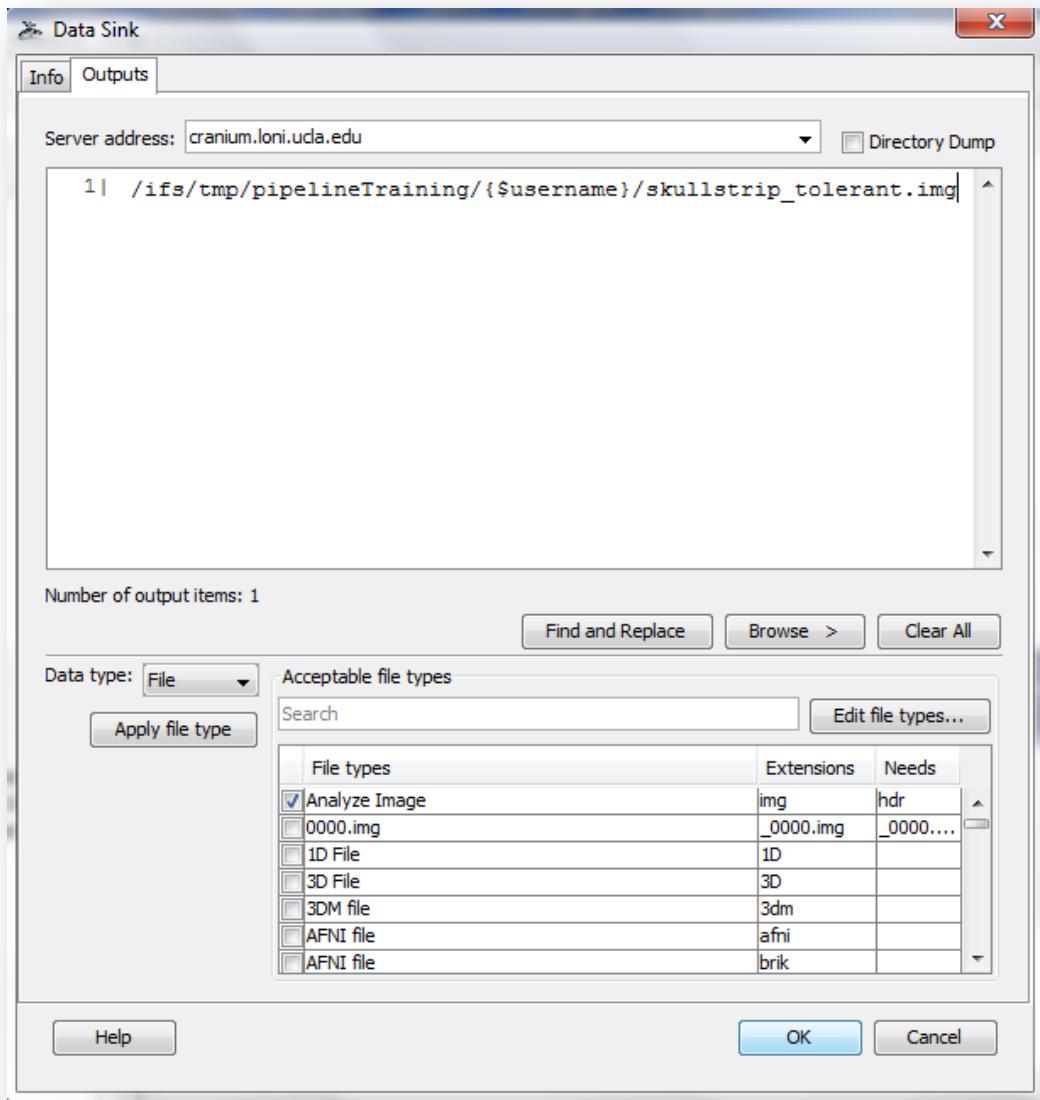


7) We will now drag in a second instance of the FSL Maths module, but will use it to perform an intersection of the two skullstripping programs instead of a union. Double-click on this FSL Maths (nii.gz) module and select the parameter called "Min (image)." Connect the output of Brain Surface Extractor to one of the two inputs of FSL Maths (nii.gz).

Connect the output of BET (nii.gz) to the other input of FSL Maths (nii.gz). Again, note that the Min operator is commutative, so the order in which the inputs are connected is irrelevant.

# Skullstripping

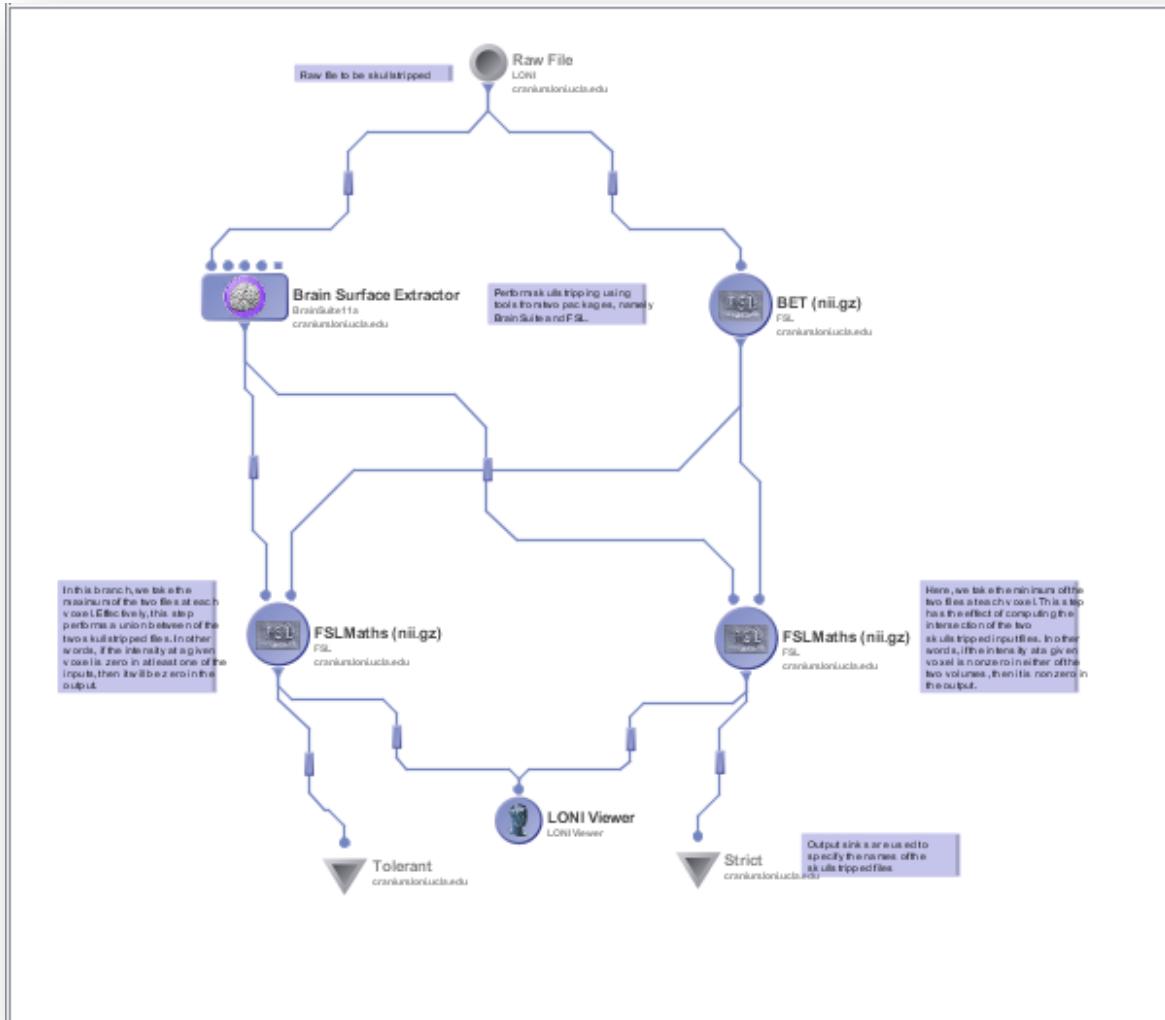
## Data Sinks/LONI Viewer



- 8) Create two data sinks, one labeled 'Tolerant' and the other 'Strict'. In the former, specify the remote path as `/ifs/tmp/pipelineTraining/{$username}/skullstrip_tolerant.img` and in the latter, use the value `/ifs/tmp/pipelineTraining/{$username}/skullstrip_strict.img`. Don't forget to select Analyze Image (with 'hdr' in the 'Needs' column) as the filetype of the output and `cranium.loni.ucla.edu` as the Output location.
- 9) Next, create a new LONI Viewer and connect it to the two FSLMaths modules.

# Skullstripping

## Connect Outputs

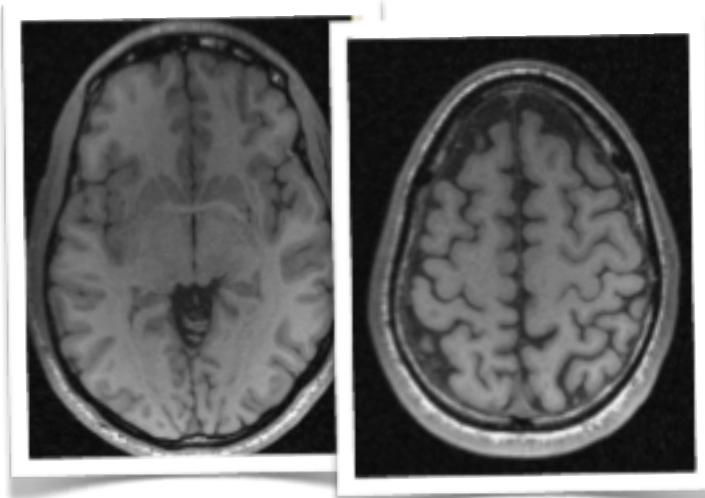


10) Connect the output of the FSL Maths (nii.gz) module running with the '-max' option to the Tolerant data sink and the one with the '-min' option to the Strict sink. Also connect both outputs to the viewer.

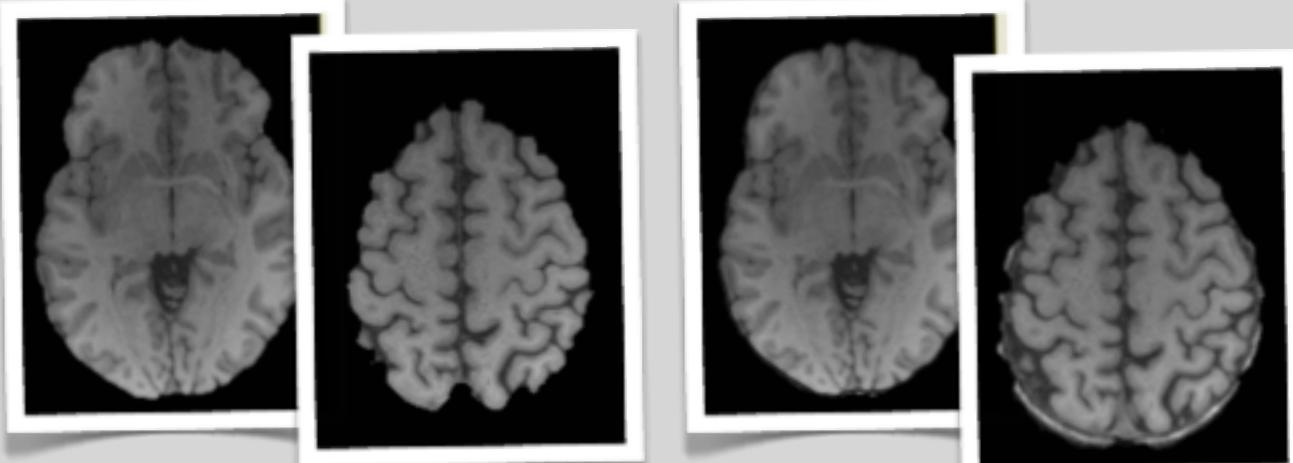
11) Optionally, you can also connect a LONI Viewer to the data source in order to visualize the input.

# Skullstripping

## Results



Raw Input



Strict Results

Tolerant Results

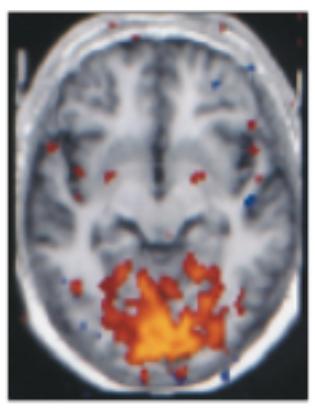
The results show that the Strict method shows some loss of brain matter, whereas the Tolerant method shows some skull remaining.

# fMRI First Level Analyses

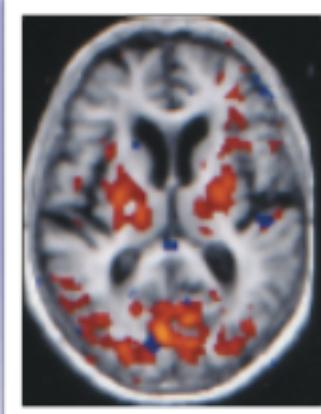
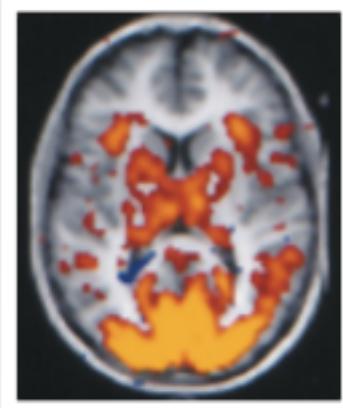
FSL FEAT



Raw Input



Results



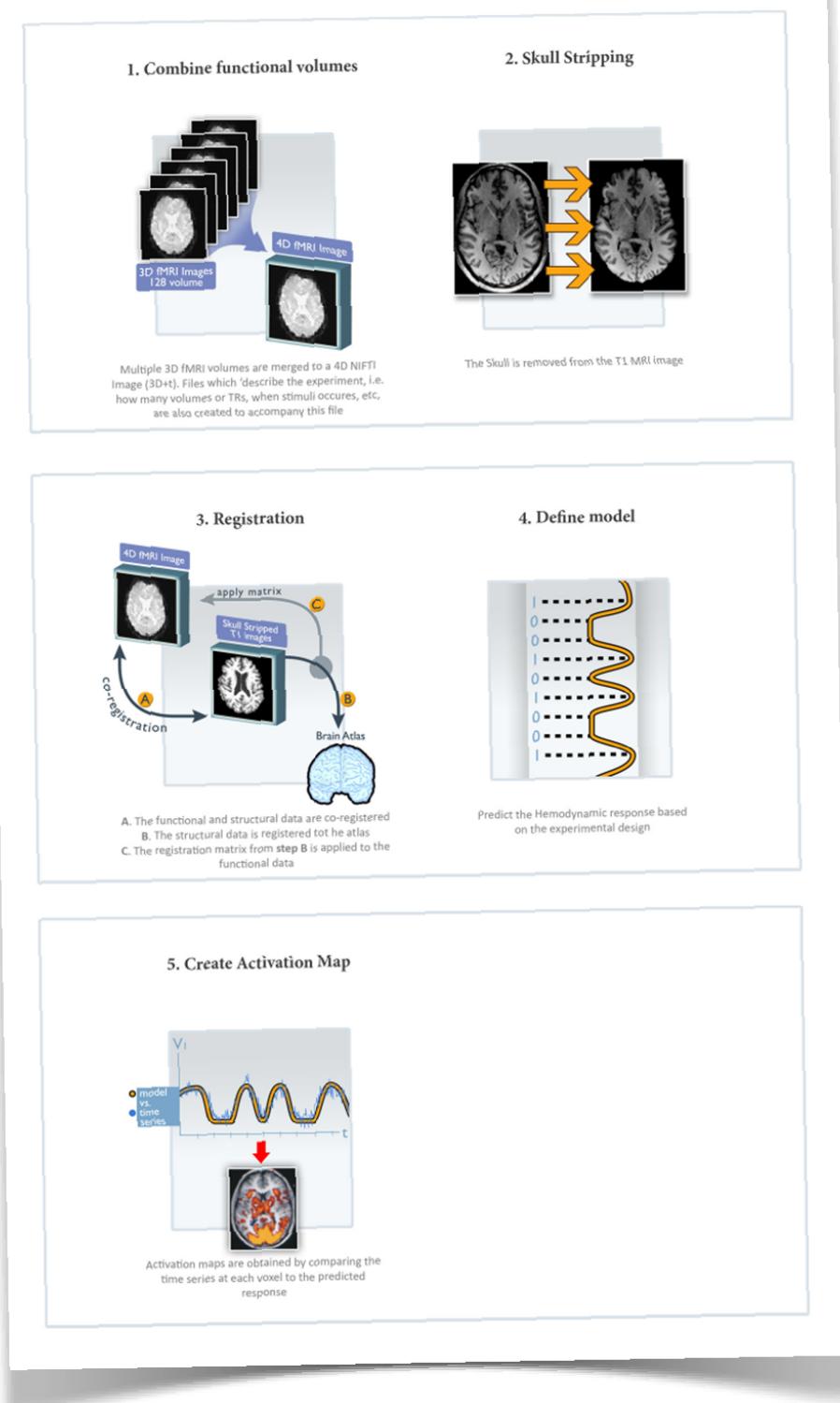
# fMRI First Level Analyses

## Introduction

Generate first level analyses fMRI results on a single subject.

### Overview

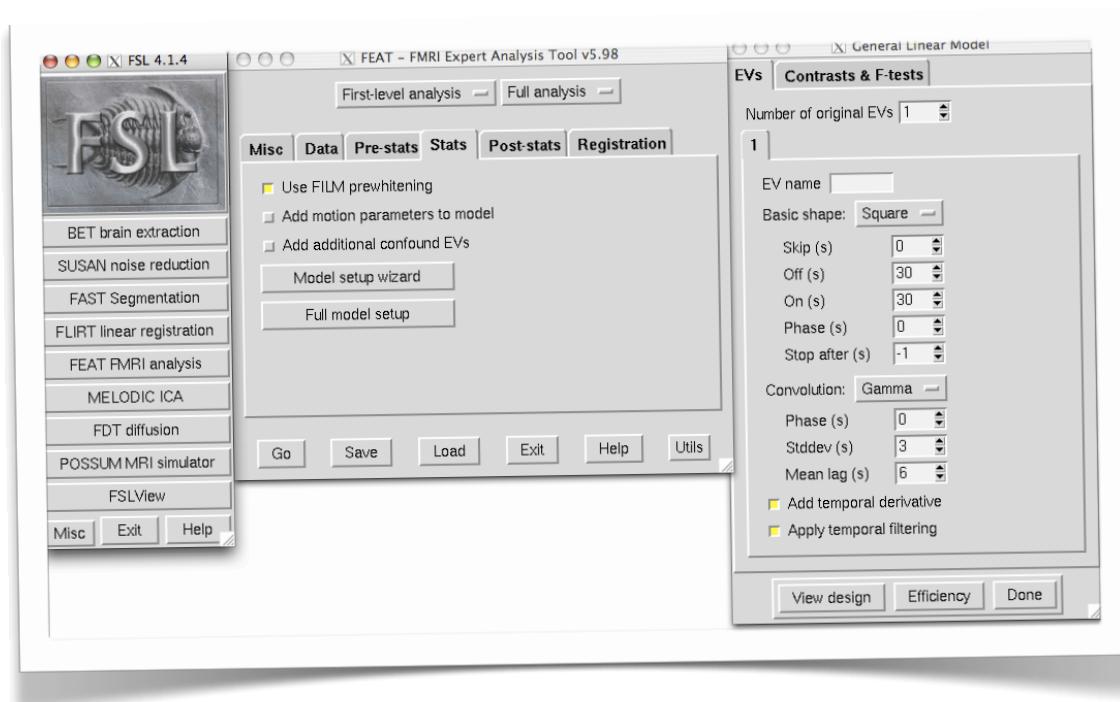
This workflow is a simple single subject, first level fMRI analysis. The workflow features FSLmerge, BET (in case the data is not already skull stripped for non-skullstripped data) and first level FEAT analysis.



# fMRI First Level Analyses

Before you begin...

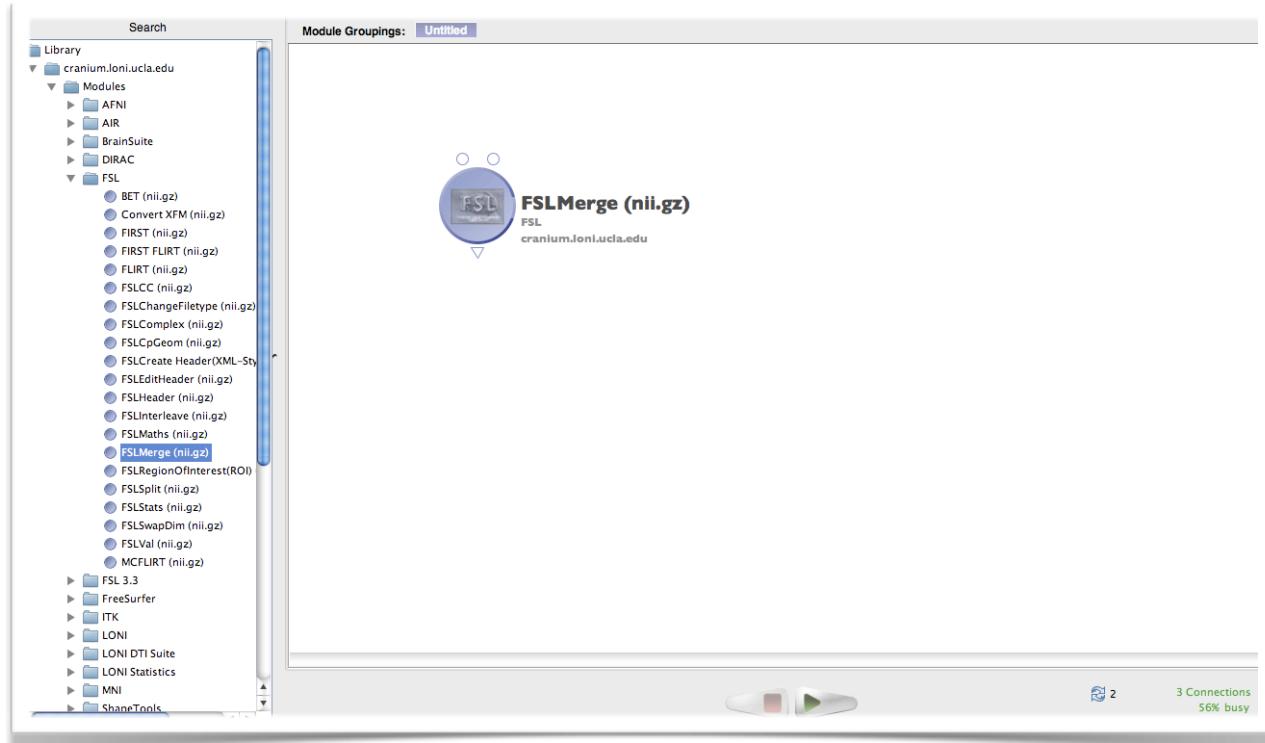
Before we begin, a design file is required (design.fsf) from FSL FEAT. You may need to launch FSL FEAT to create a design file—the LONI Pipeline can edit items in an existing file but cannot add new items, such as more experimental variables to a design file. The included, pre-made design.fsf file is a single subject with 2 experimental variables. If your experiment matches, you can just use the template included. However, if your experiment differs from this, you will have to create your own design file in FSL FEAT first.



# fMRI First Level Analyses

## 3D to 4D FSLMerge

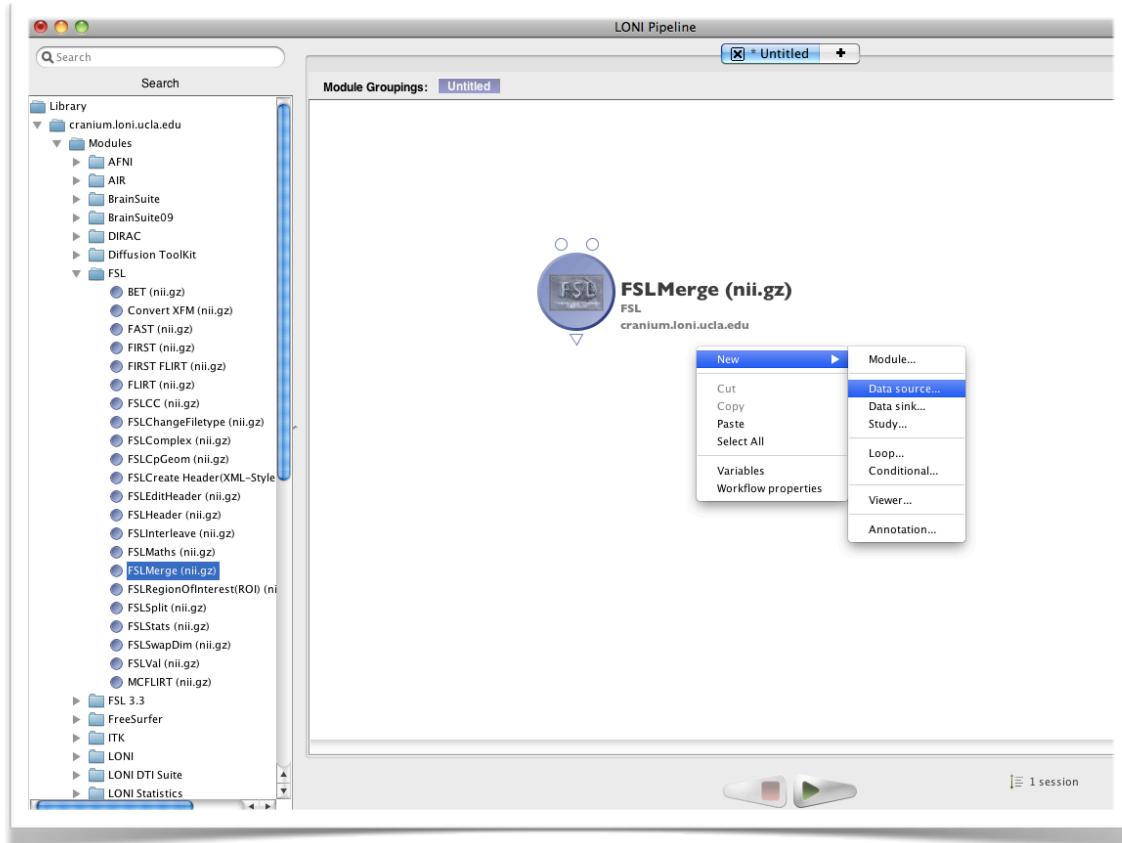
- 1) To begin, open the LONI Pipeline. Click on new workflow. Expand the FSL folder in the Library on the left and select FSLMerge. Drag the module to the empty canvas.



# fMRI First Level Analyses

## Specify 3D Files

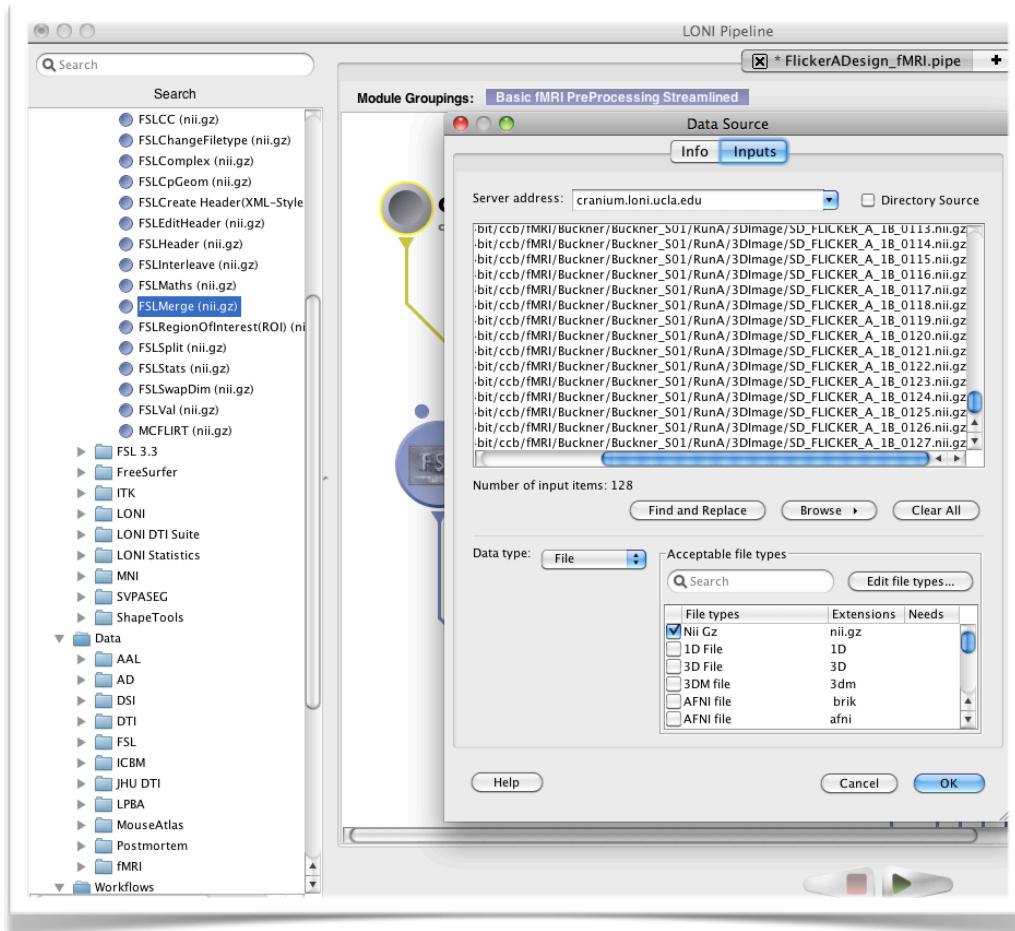
- 2) Right click on the canvas and select a data source from the Pipeline Library “One Functional Run”.



# fMRI First Level Analyses

## Specify 3D Files

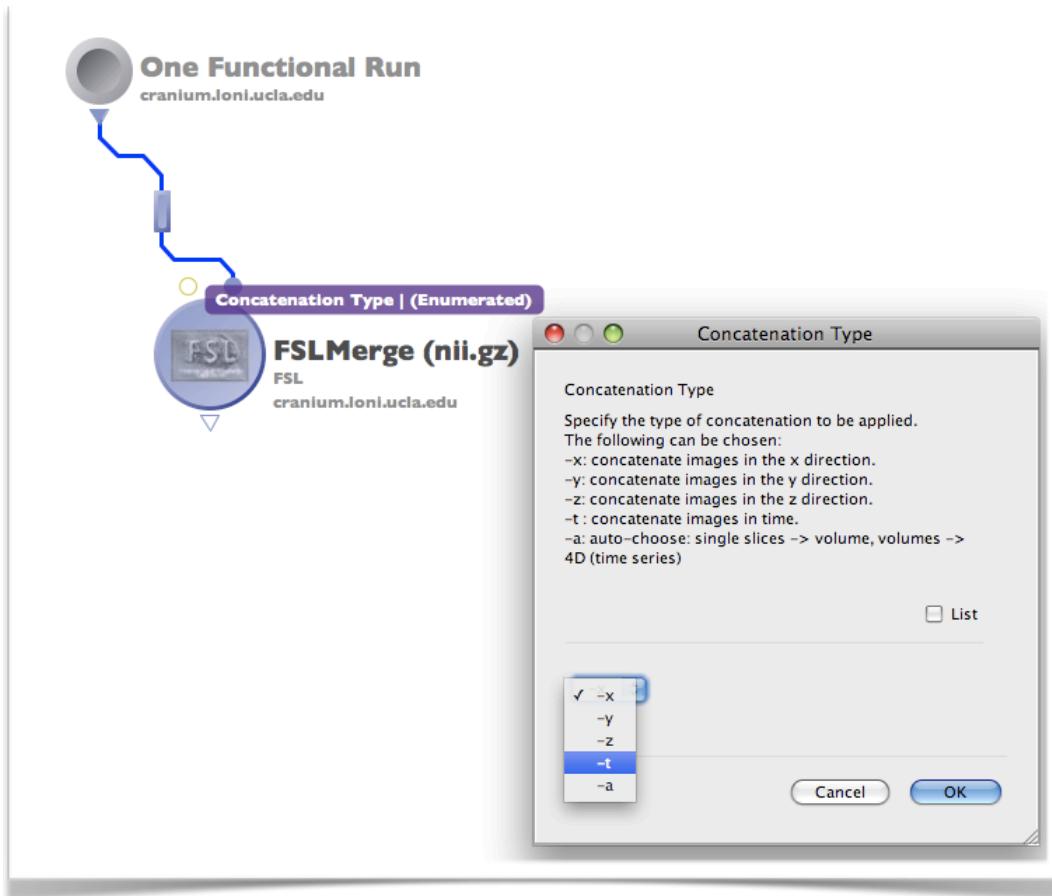
- 3) Choose all the 3D fMRI files for the subject. Connect the output of the data source to the input of FSLMerge.



# fMRI First Level Analyses

## Concatenate 3D Files Across Time

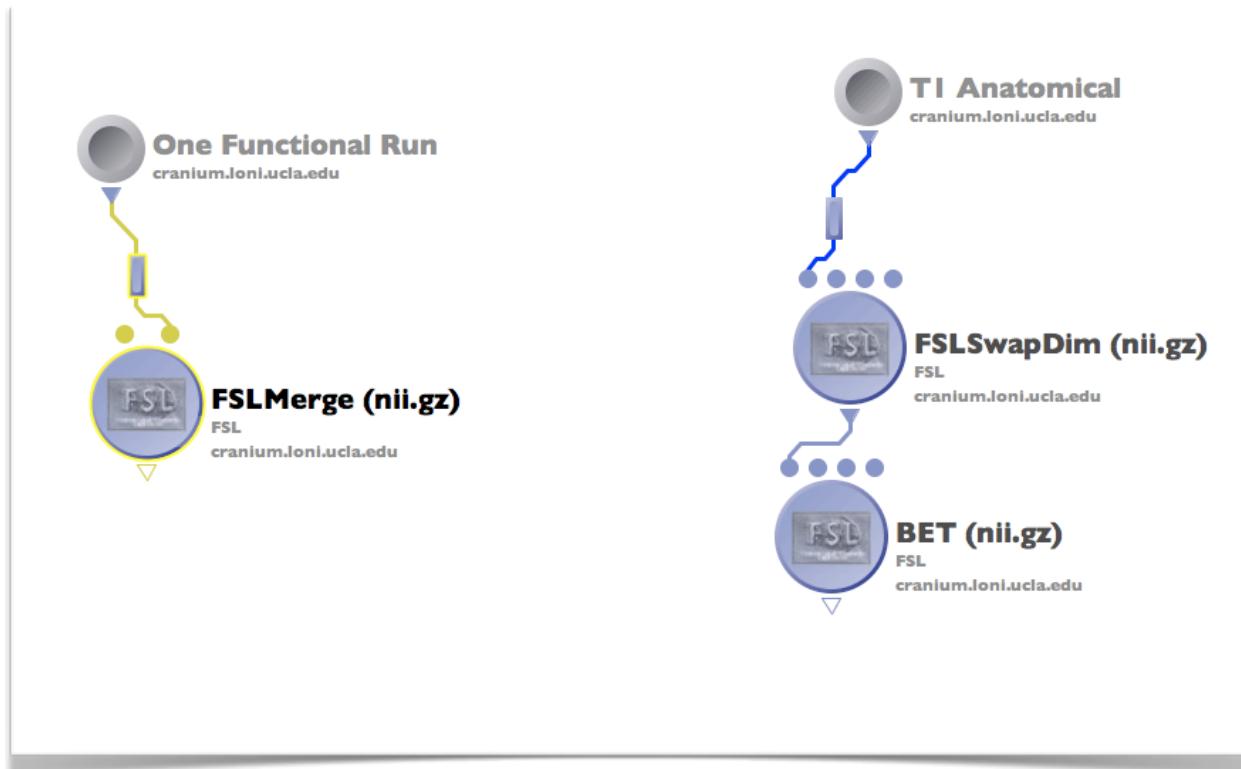
- Double click on the “Concatenation Type.” Select -t in the drop down menu.



# fMRI First Level Analyses

## Brain Extraction

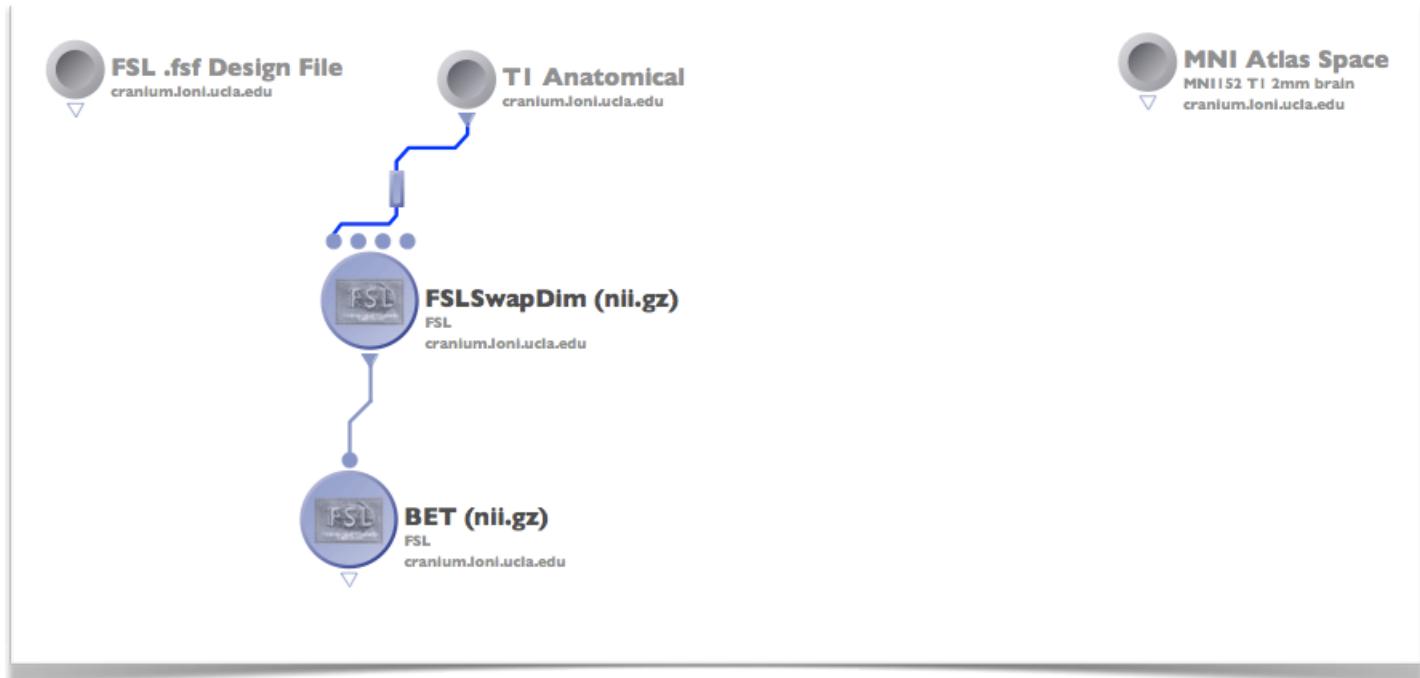
- 5) Choose FSLSwapDim and BET (nii.gz) modules. Drag them to the canvas. Right click and select New data source, name it "T1Anatomical". Select the T1 Anatomical data of the corresponding subject. Connect all the 3 modules, Data source to FSLSwapDim and FSL SwapDim to BET



# fMRI First Level Analyses

## Specify the Atlas and the Template Design File

- 6) Right click on the canvas and create two new data sources. Name one “MNI Atlas Space”. Double click on it a Select the MNI Atlas, which will be used as a reference. Name the second source, “FSL .fsf Design file.” Select the design file created using the FSL FEAT interface.

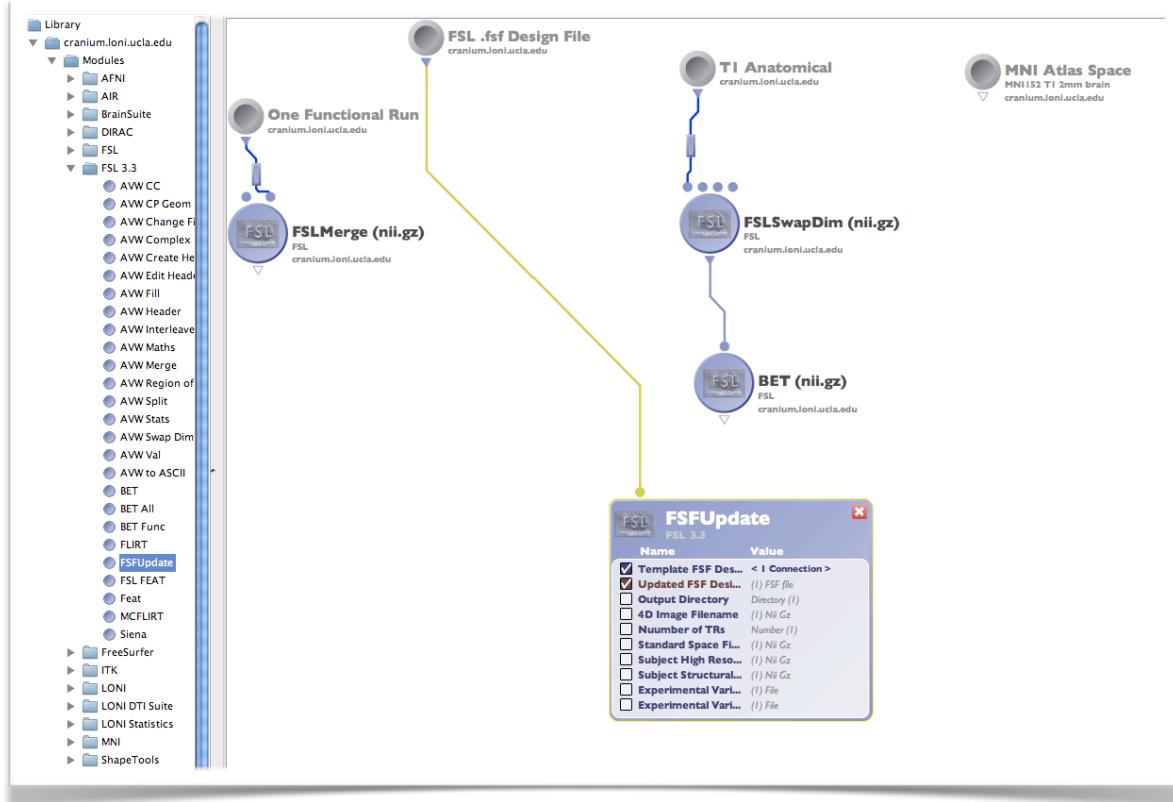


# fMRI First Level Analyses

## Update Design File

Select the FSF Update Module under the FSL library. Drag it to the canvas.

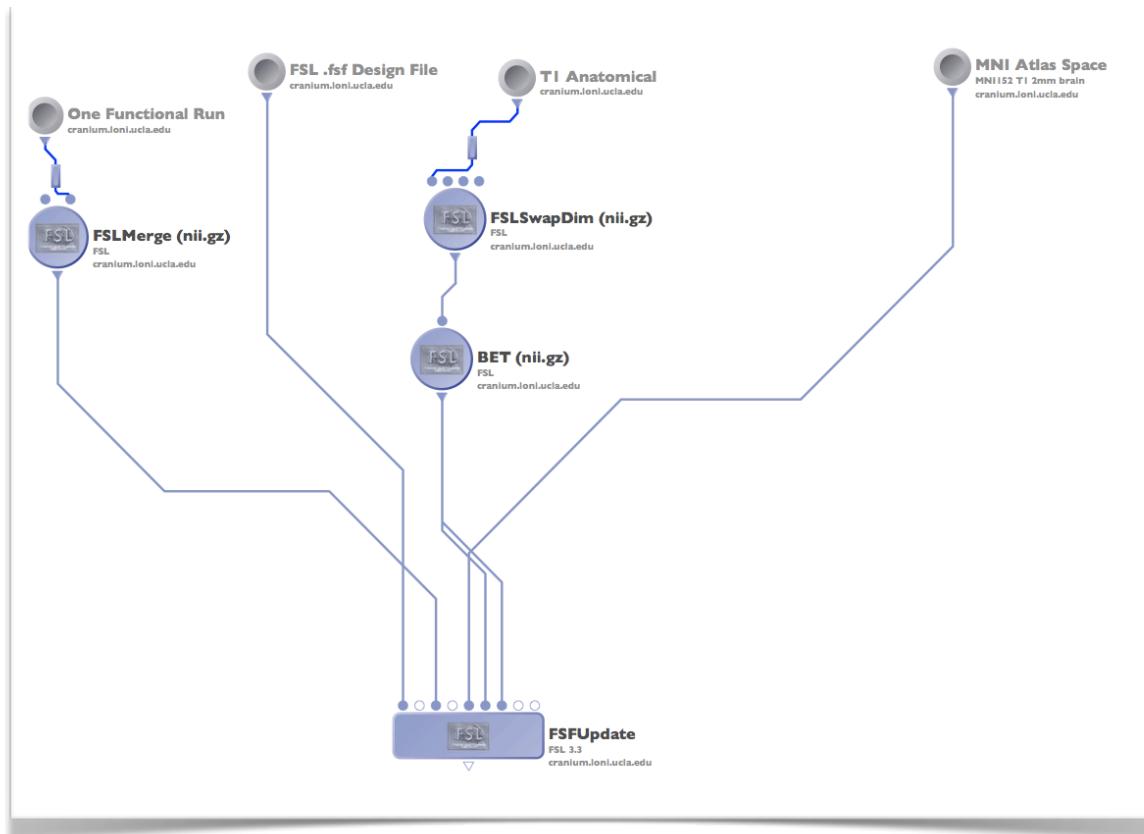
Connect the data source “FSL .fsf Design file” to the empty circle that says “Template FSF design file”. Double click on the FSF Update Module and select all the other optional parameters.



# fMRI First Level Analyses

## Update Design File

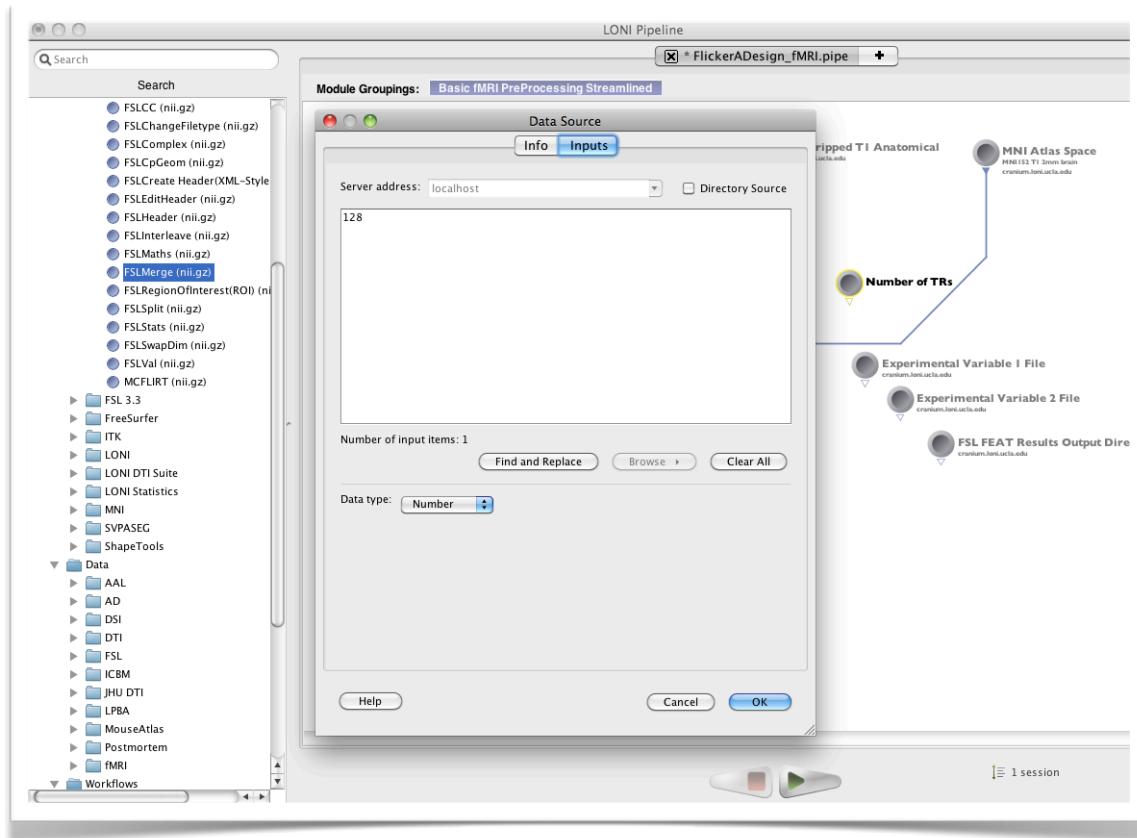
Connect the “MNI Atlas Space” data source to the corresponding empty circle on the FSF Update Module. Connect the output of BET to the circles labeled, Subject High Resolution Anatomical and Subject Structural Image for Analysis. Connect the output of FSLMerge to the corresponding circle.



# fMRI First Level Analyses

## Update Design File

- 9) For this analysis, we have two experimental variables. our Number of TRs is equal to 128. We will create four data sources to update the fsf design file. Name the first data source “Number of TRs.” Enter 128. Connect the Number of TRs data source to the FSF Update module’s input parameter that also reads



“Number of TRs”.

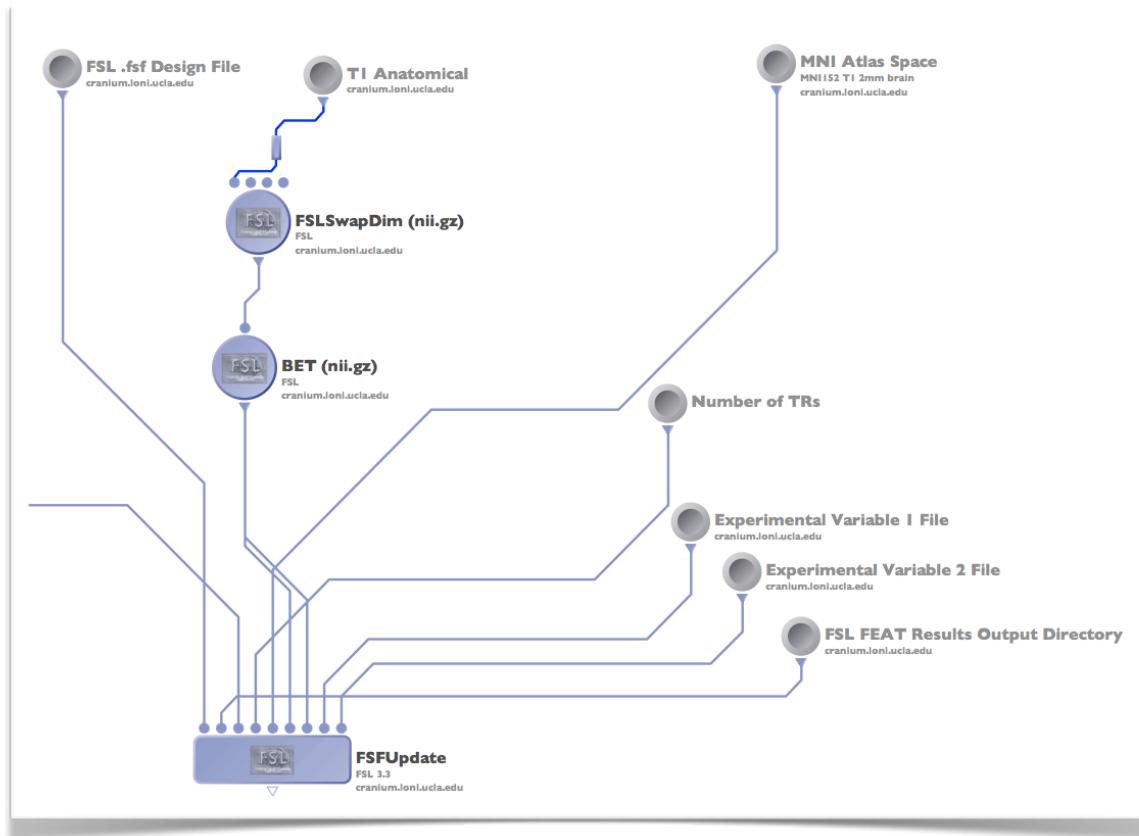
# fMRI First Level Analyses

## Update Design File

10) The next two data sources can be named Experimental Variable 1 and Experimental Variable 2. Double click on each of the Experimental Variables and select the corresponding timing file (a text file which has information about the onset times and duration of the stimuli).

The third data source can be named “Output Directory”. Use this data source to set the path location to your final results directory.

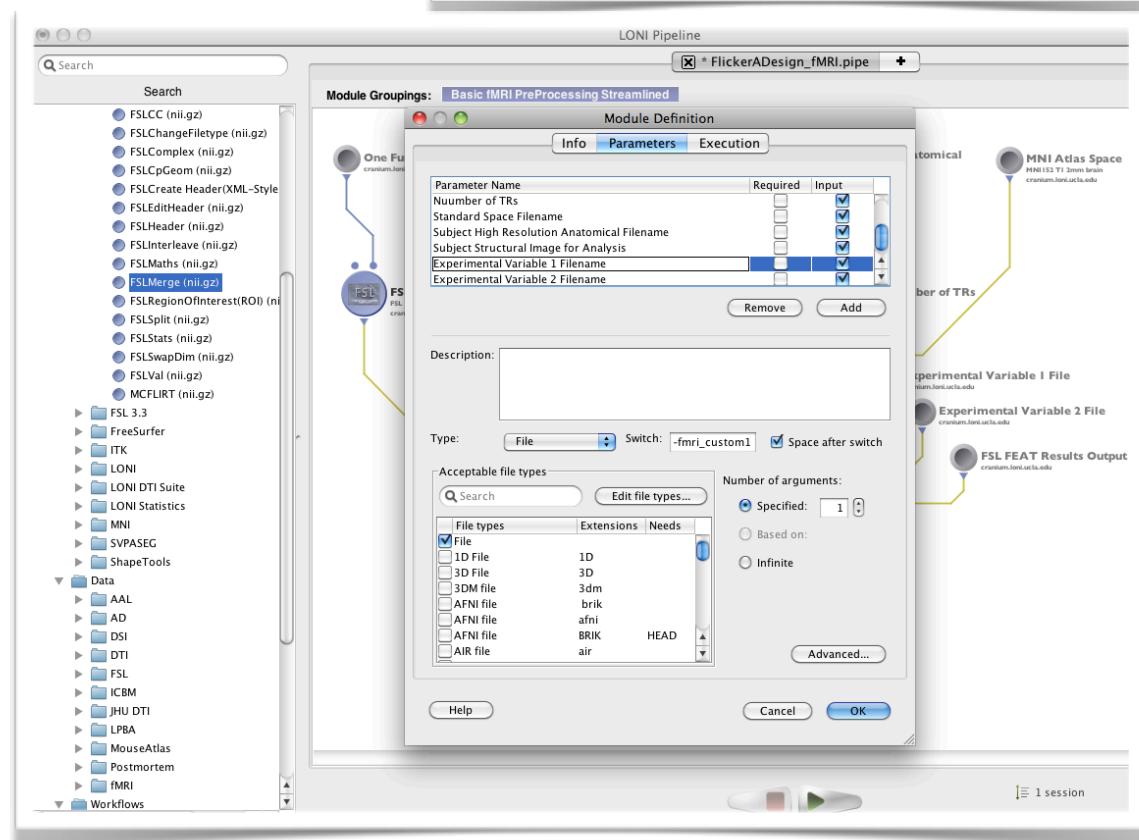
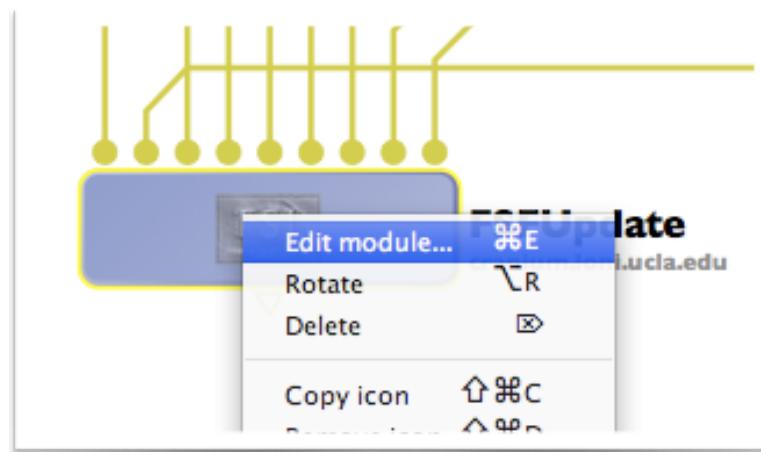
Connect all the modules to the respective empty input parameters in the FSF Update module. To do this, drag a line connecting each of the above-mentioned data source output parameters to the corresponding input parameters on the FSF Update module.



# fMRI First Level Analyses

## Add Optional Parameters

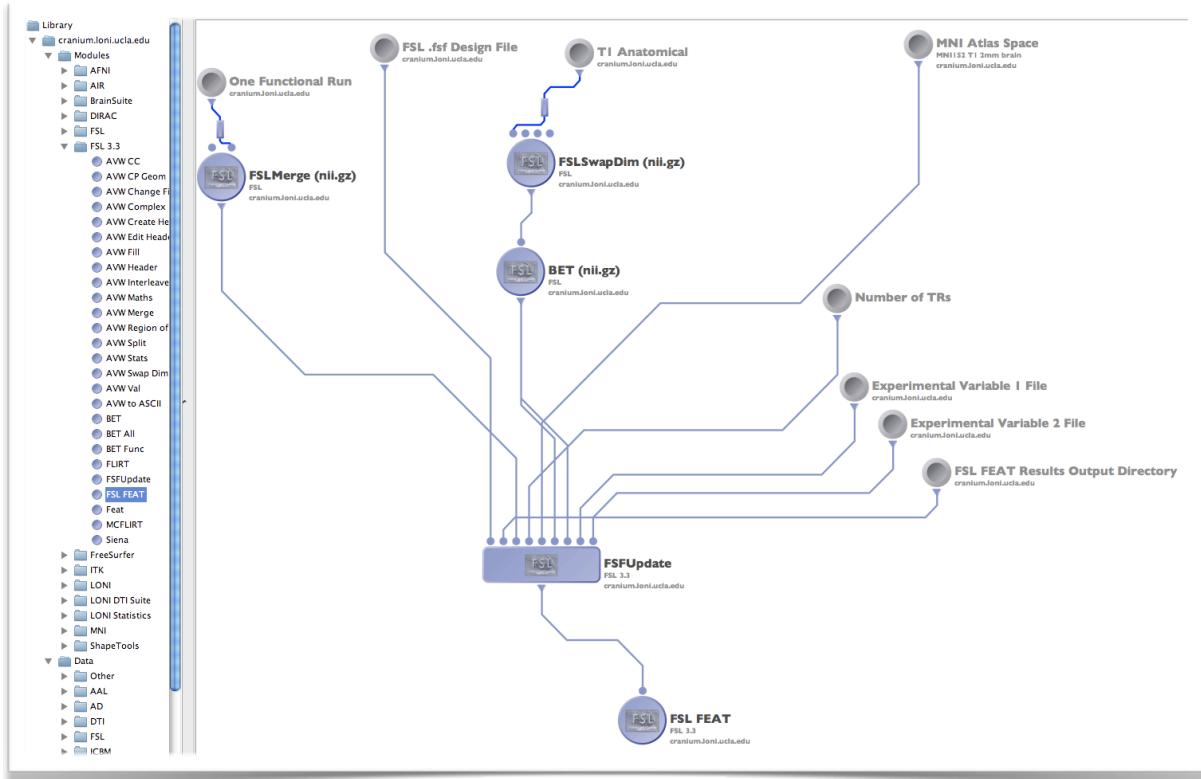
- 11) If you are using a new design file rather than the template we provided (see the section before step 1, “Before you begin...”) you may now have to add a parameter to the FSUpdate module. For instance, if your design file has three Experimental Variables, rather than two, you would need to right click and edit the FSUpdate module. Under the parameters tab, add a new parameter and name it “Experimental Variable 3 File Name”.



# fMRI First Level Analyses

## Setup FEAT

12) Select the FSL FEAT module under the FSL suit and drag it to the canvas. Connect the output of the FSFUpdate module to the FSL FEAT module.



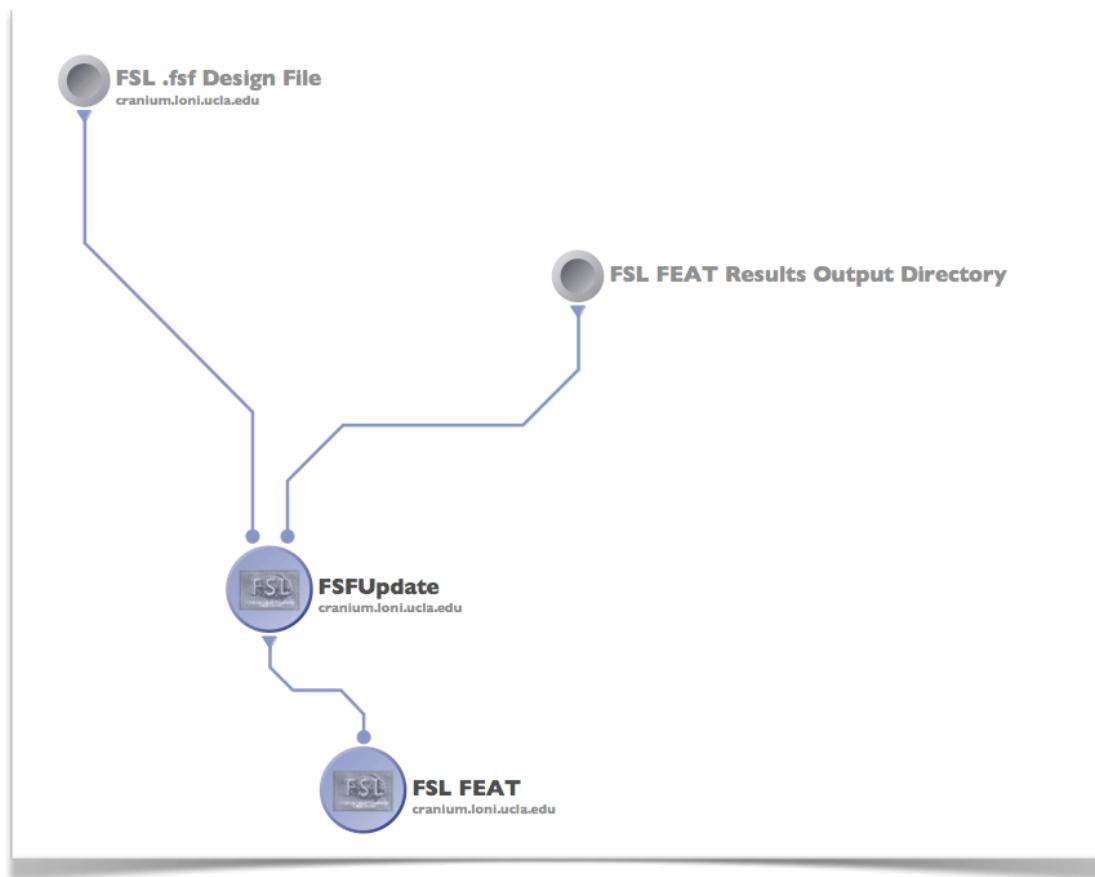
# fMRI First Level Analyses

## Re-run FEAT

- 13) If you would like to re-run the analysis on the same subject with the same parameters, you can simplify the workflow as shown in the picture below and described as follows.

Double click on the FSFUpdate Module and uncheck all the parameters, except Template FSF Design File, Updated FSFDesign File and Output Directory. Delete all the modules that are not connected to the FSFUpdate module.

The input to the FSL Design data source will now be the updated design file created in the previous workflow (this has been saved as a separate design.fsf file in the results directory). Next specify the output directory path and run the new workflow. This simplified workflow will save you from having to run through the steps of updating the template design file again.

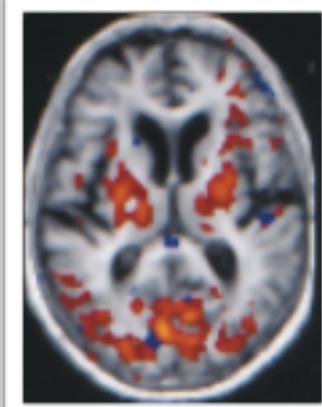
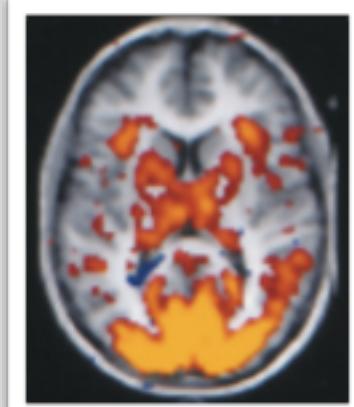
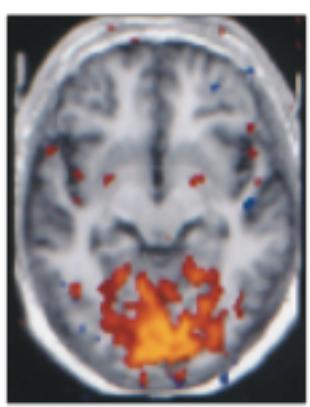


# fMRI First Level Analyses

## Results



Raw Input



Results

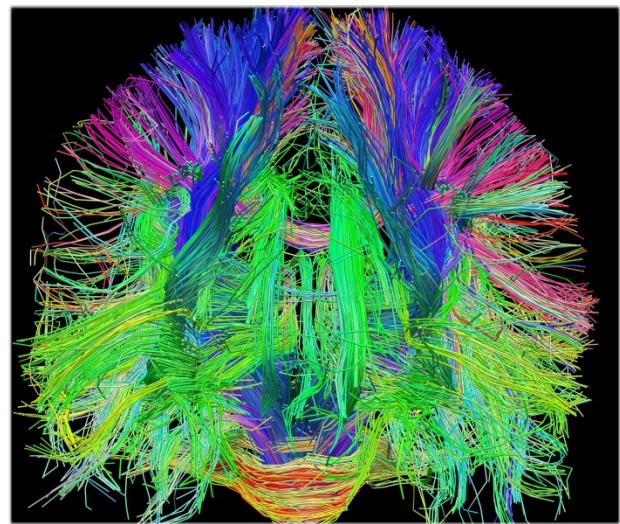
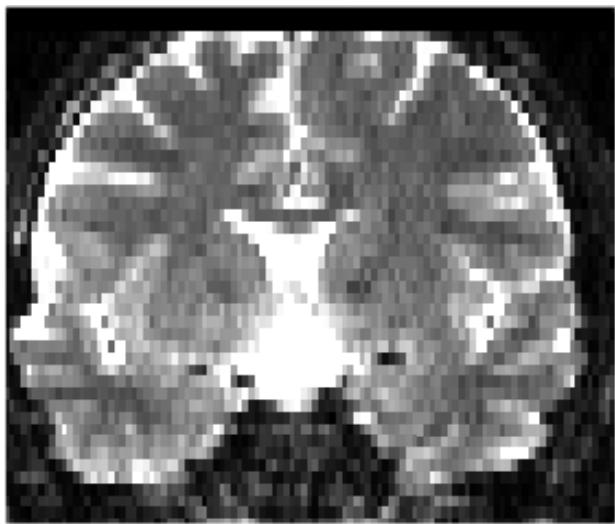
**Data for this workflow was graciously provide from:**

Buckner RL, Snyder AZ, Sanders AL, Raichle ME, Morris JC (2000) "Functional Brain Imaging of Young, Nondemented, and Demented Older Adults", Journal of Cognitive Neuroscience, 12 Supplement 2, pp. 24-34.

# DTI Analyses

## Introduction

This workflow creates white matter tracts for viewing in TrackVis from DTI files.



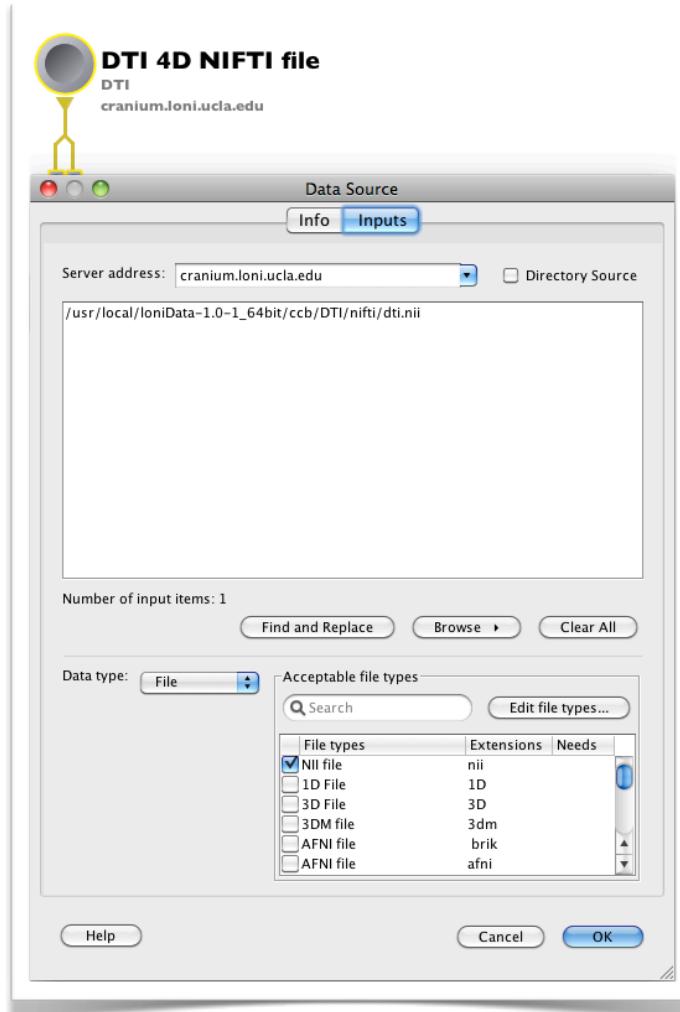
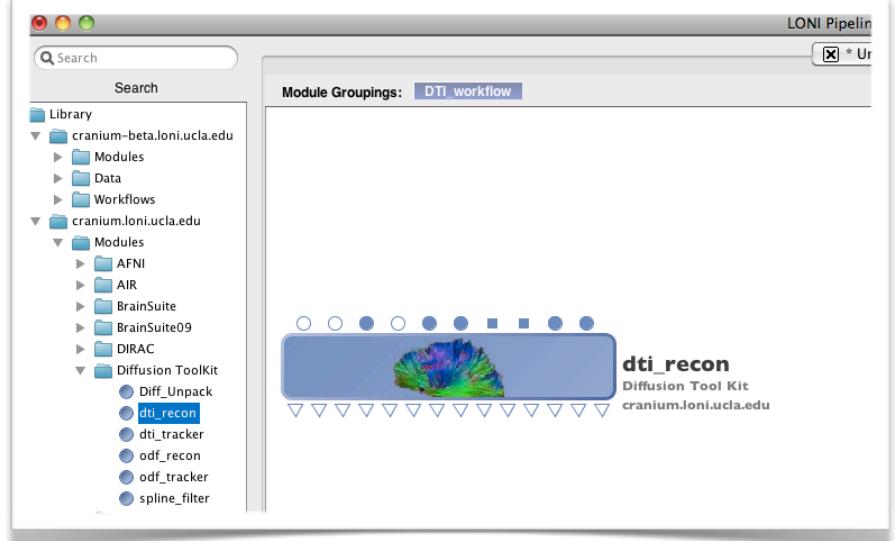
### Overview

In this workflow, DTI (DICOM/NIfTI) images are reconstructed and fiber tracks are generated for TrackVis. Diffusion toolkit command-line tools are used to build the workflow. Individual command-line tools are available as modules in the pipeline library. To build this workflow follow the steps below.

# DTI Analyses

## Copy DTI Recon and specify the input

- 1) Start by opening a new workflow. Expand the Diffusion Toolkit folder under the Library and choose dti\_recon module.



Create a new data source and choose the 4Ddti.nii file.

# DTI Analyses

## DTI Recon Optional Parameters

- 2) Connect the data source to the RAW\_DATA and the OUTPUT\_FILE\_PREFIX input parameters in the dti\_recon module. Double click on the module and choose the following optional parameters (Pic3).

output type: this is always set to nii

b-value: this is set to 1000 based on the data that is used.

number\_of\_b0: 1 for this dataset

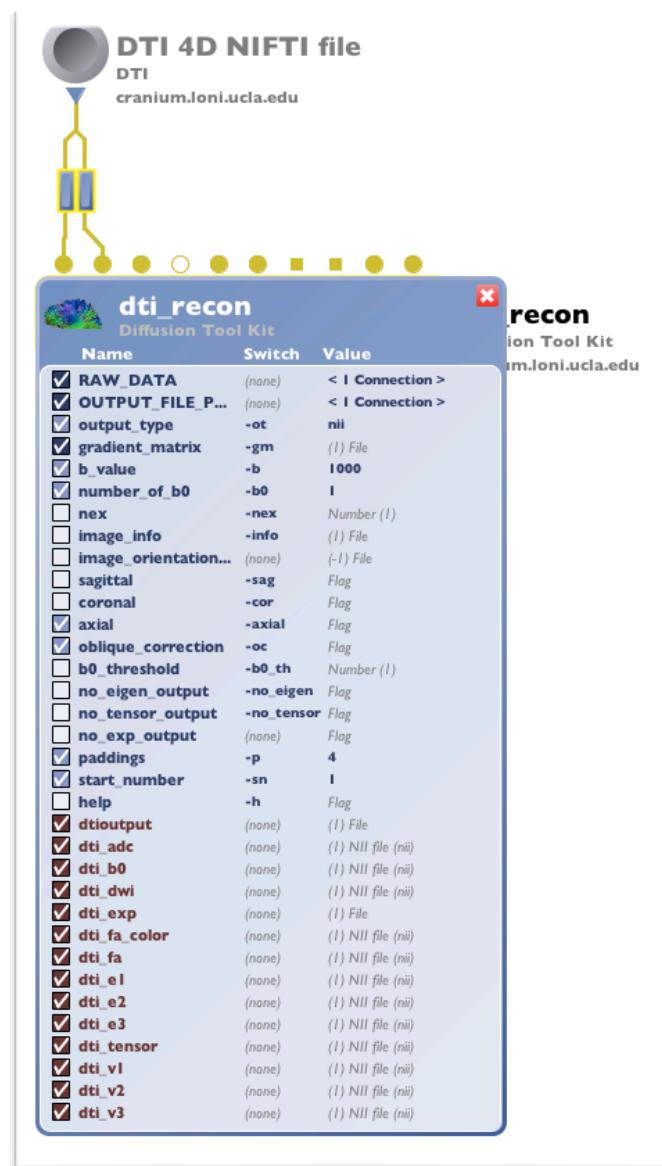
axial: flag

oblique\_correction: flag

padding: is set to 4

start\_number: is set to 1

**NOTE:** These options are for the dataset that is used in this example. It can be changed according to any dataset.



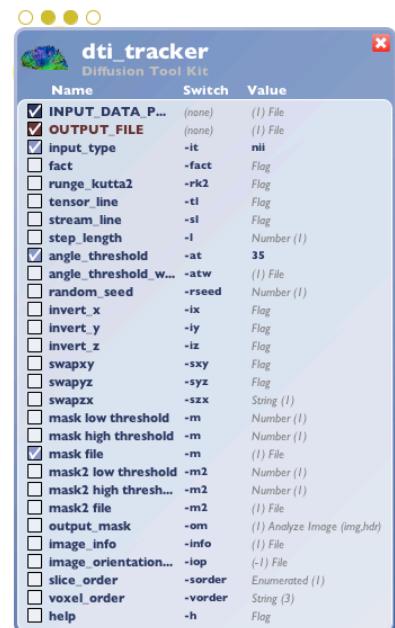
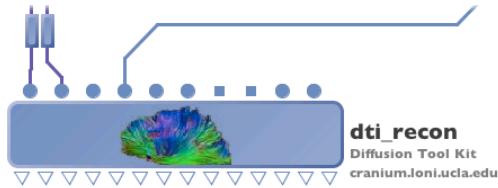
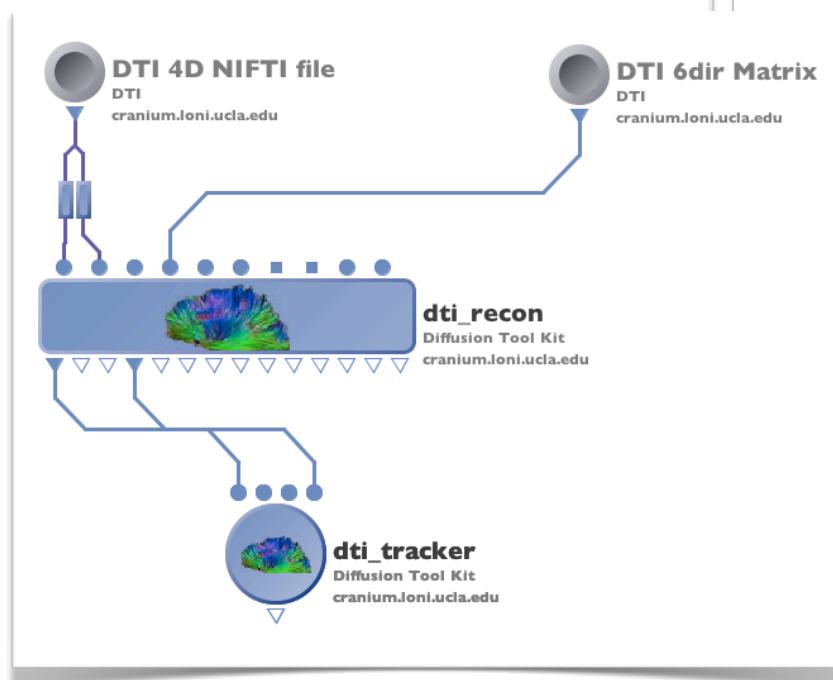
# DTI Analyses

## DTI Tracker

- 3) Click on dti\_tracker module in the library and drag it to the empty canvas. Double click on the module and enable the following optional parameters (Pic5).

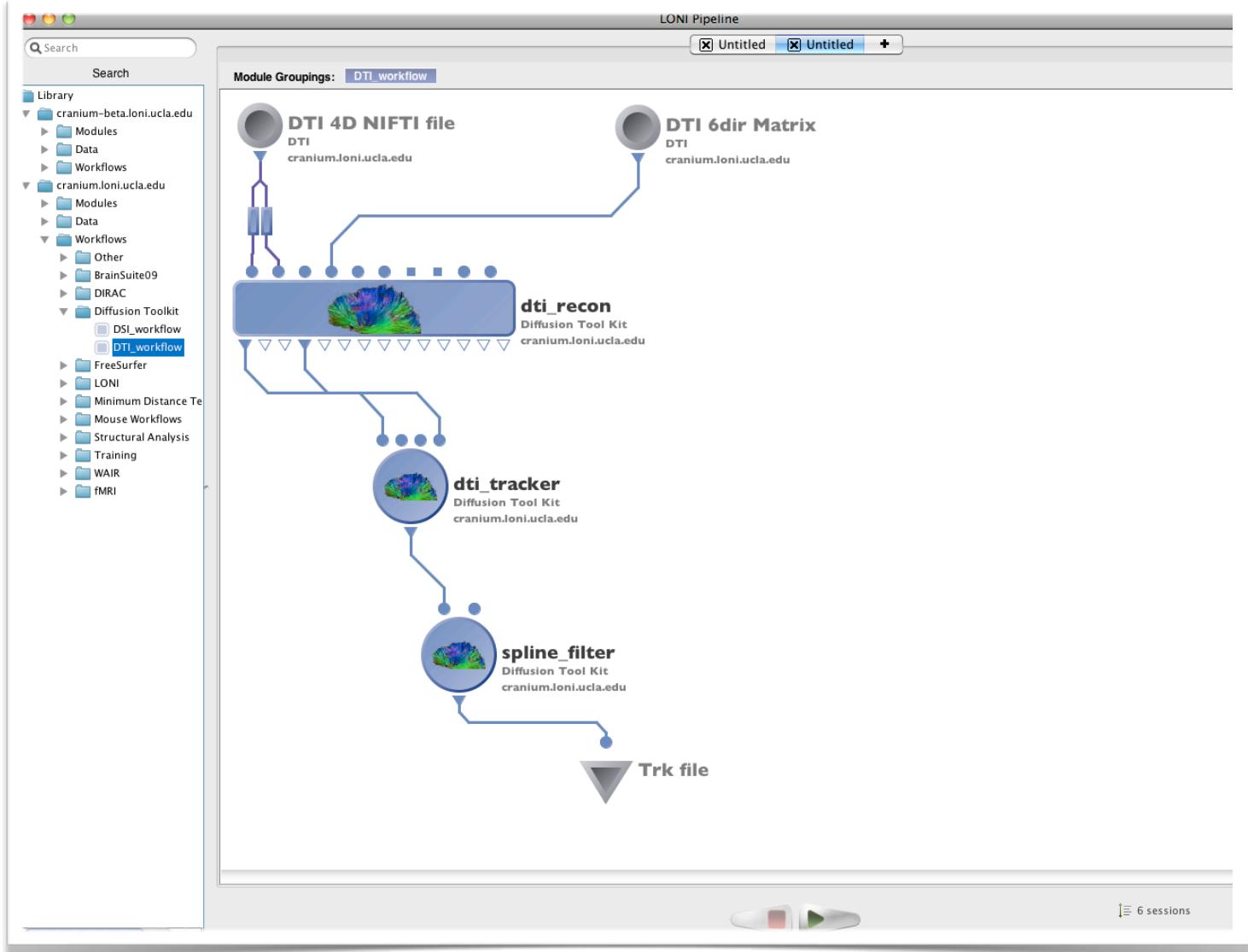
Input\_type: choose nii option only.  
angle\_threshold: enter the value 35  
mask file

Connect the output parameters, dtioutput and dti\_dwi in the dti\_recon module to the input parameters, INPUT\_DATA\_PREFIX and mask file in the dti\_tracker module.



# DTI Analyses

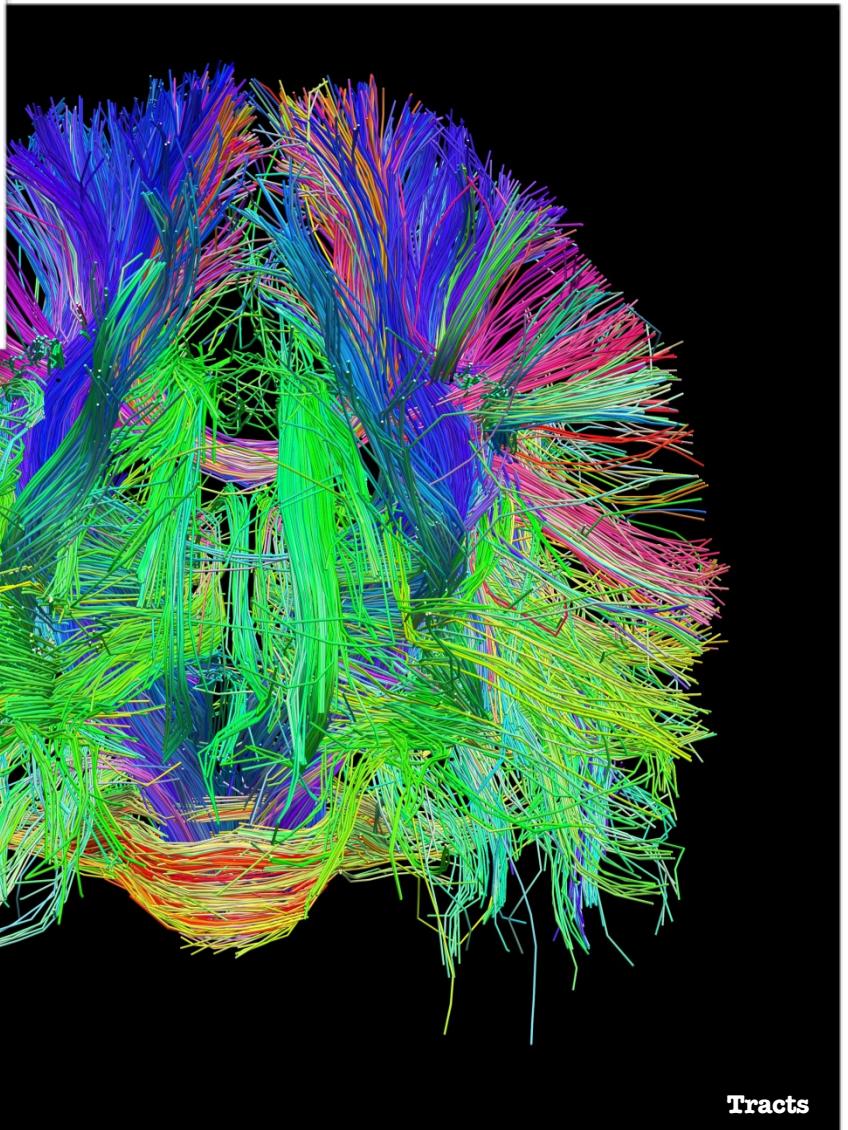
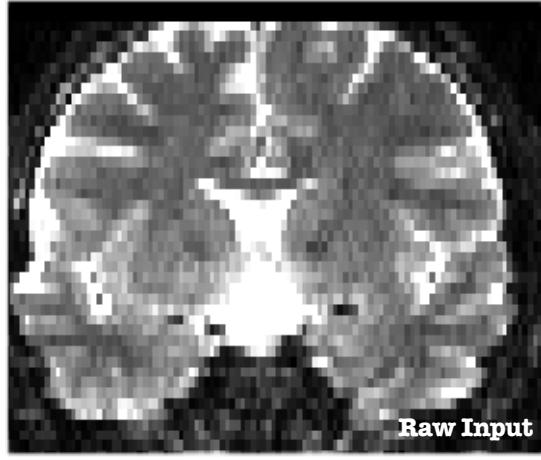
## Spline Filter and Output TRK file



- 4) Choose and drag the spline\_filter module and connect the output of the dti\_tracker module to the INPUT\_TRACK\_FILE parameter in the spline\_filter module. Choose value 1 for the STEP\_LENGTH option. Save the output of the spline\_filter as a data sink or simply specify the file path. To save outputs at intermediate steps you can add data sinks and specify the file path.

# DTI Analyses

## Results

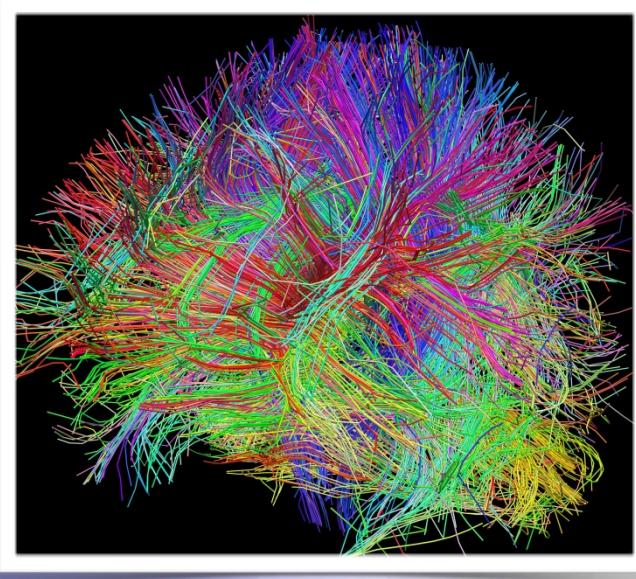
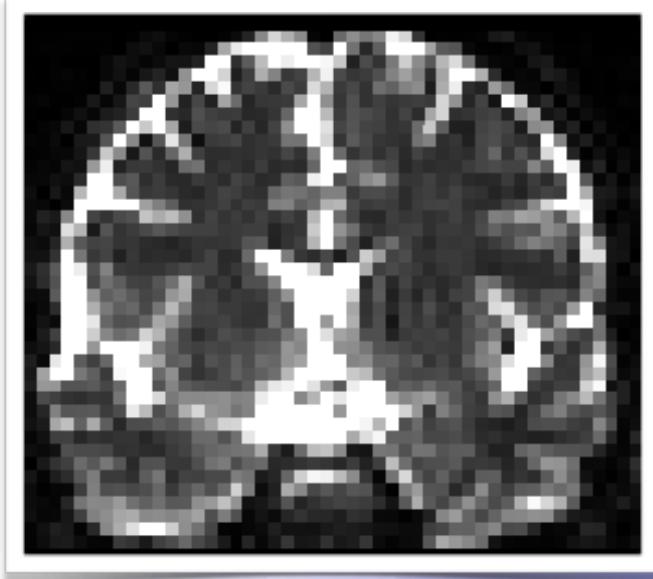


The results of this workflow is a DTI Track file that can be used in TrackVis program for further analysis.

# DSI Analyses

## Introduction

This workflow creates white matter tracts for viewing in TrackVis from DSI files.

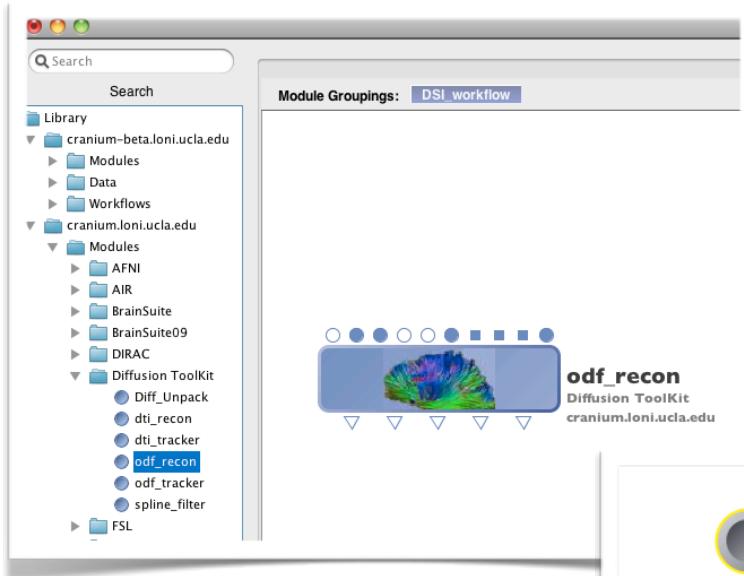


### Overview

In this workflow, DSI (DICOM/NIfTI) images are reconstructed and fiber tracks are generated for TrackVis. Diffusion toolkit command-line tools are used to build the workflow. Individual command-line tools are available as modules in the Pipeline library.

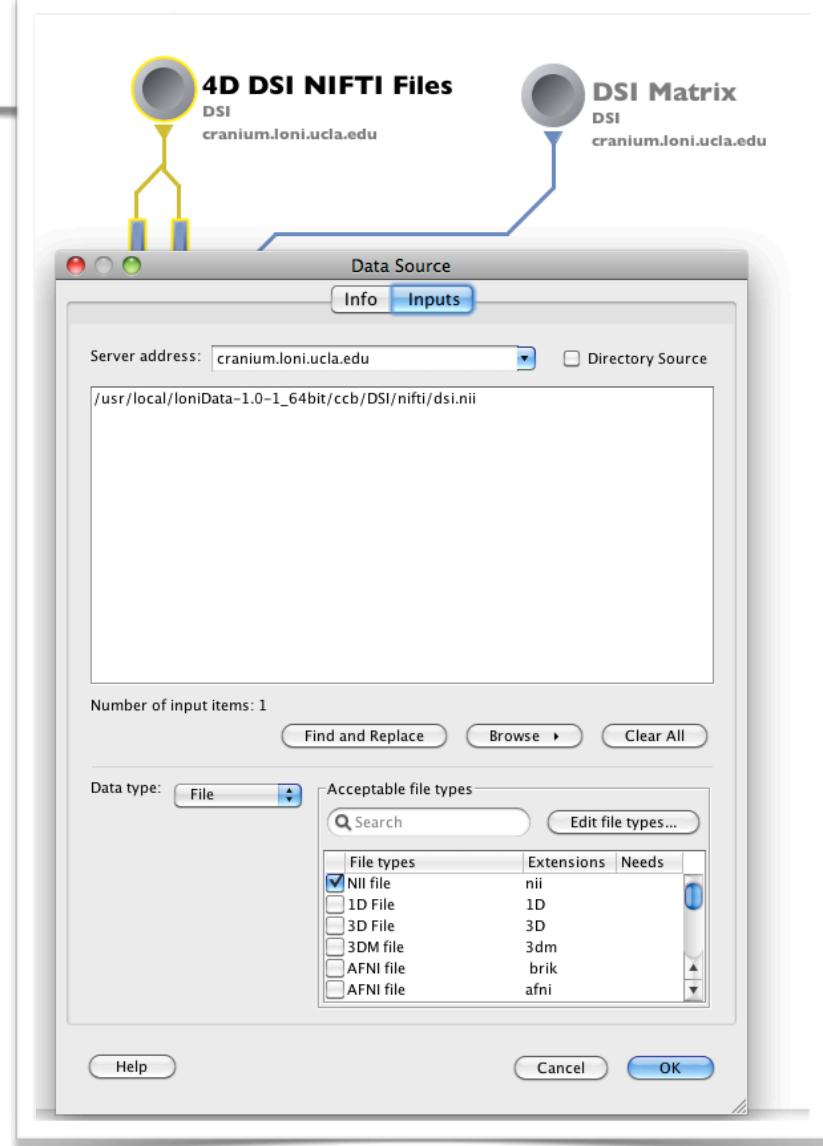
# DSI Analyses

## Copy ODF Recon and Specify the Input



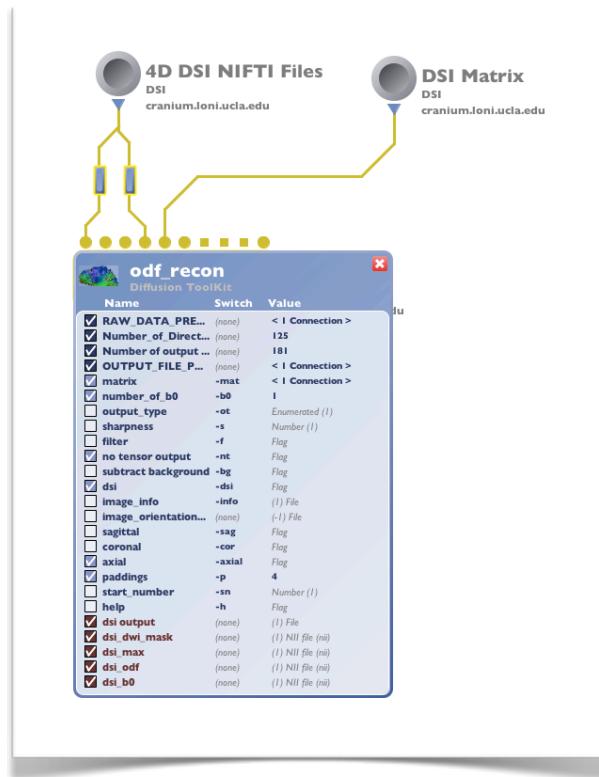
Create a new data source and choose the 4Ddsi.nii file.

- 1) Start by opening a new workflow. Expand the Diffusion Toolkit folder under the Library and choose, odf\_recon module



# DSI Analyses

## ODF Recon Optional Parameters



- 2) Connect the data source to the RAW\_DATA and the OUTPUT\_FILE\_PREFIX input parameters in the dti\_recon module. Double click on the module and choose the following optional parameters,

Number\_of\_Directions: 125 (for the data set being used)

Number of output directions: 181

- 3) To enable optional parameters, double click on the module and choose the following,

matrix

Number\_of\_b0: enter value 1 (this is for the data that is used here)

No tensor output flag

dsi: flag

axial: flag

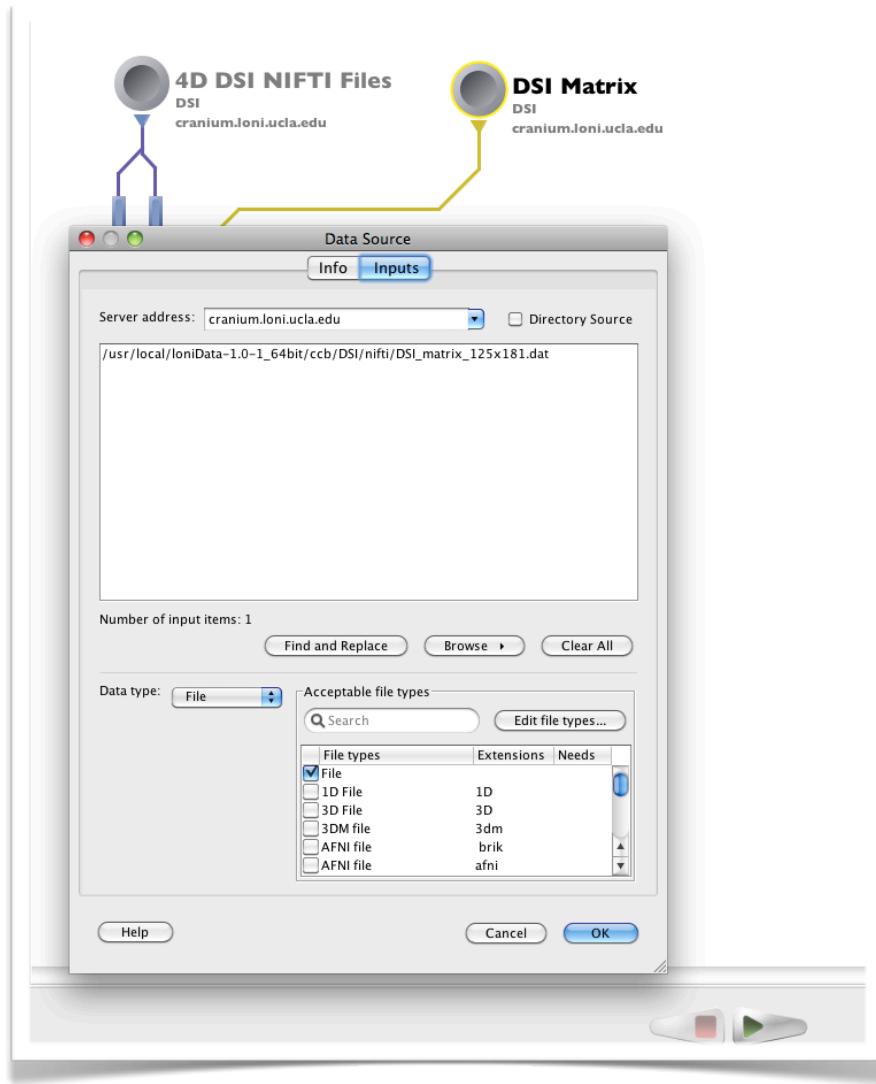
paddings: enter value 4

NOTE: These options are based on the dataset that is used in this example.

# DSI Analyses

## Specifying the Gradient Matrix

- 4) One of the input parameters in the odf\_recon module is called the gradient\_matrix. This matrix contains the directional information of the data that is used. This matrix can be saved or copied from the diffusion toolkit interface Gradient table option. For this example we use 125x181 directional matrix. Create a data source or double click on the parameter and enter the path to this file.

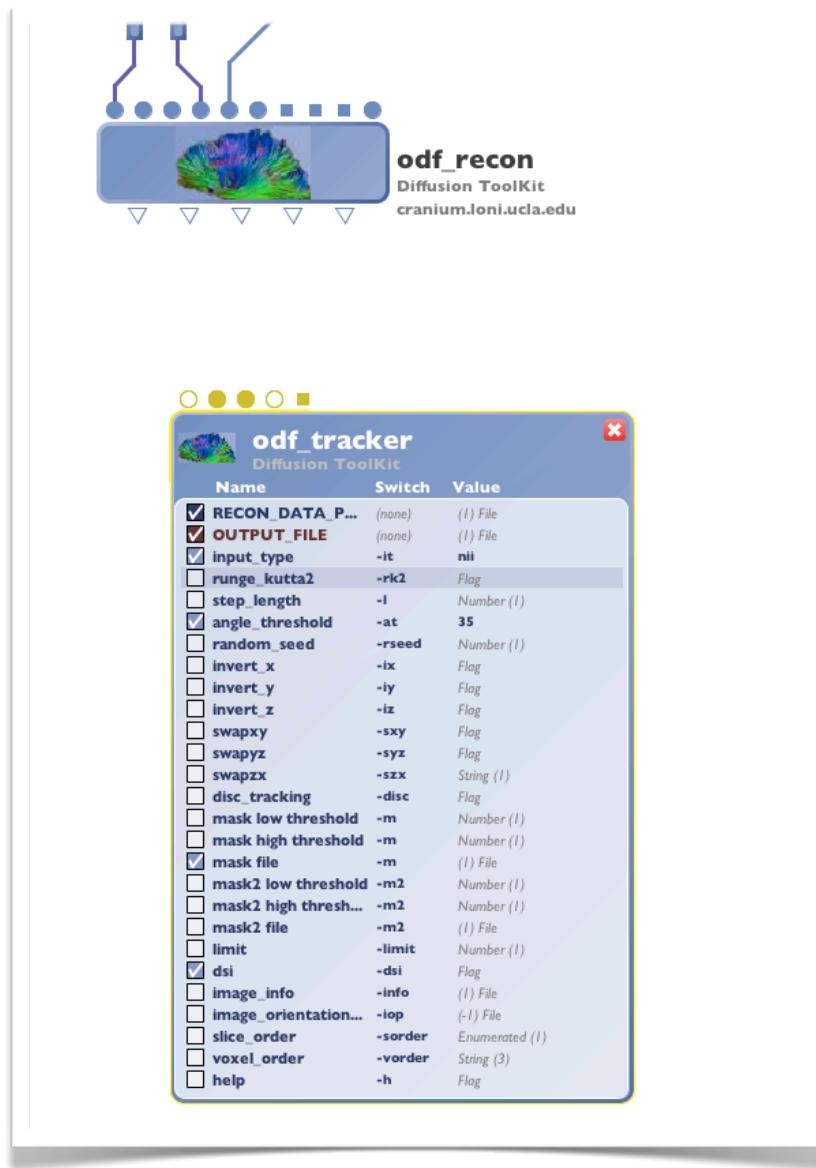


# DSI Analyses

## ODF Tracker Optional Parameters

- 5) From the library click on odf\_tracker module and drag it to the empty canvas.  
Double click on the module and enable the following optional parameters.

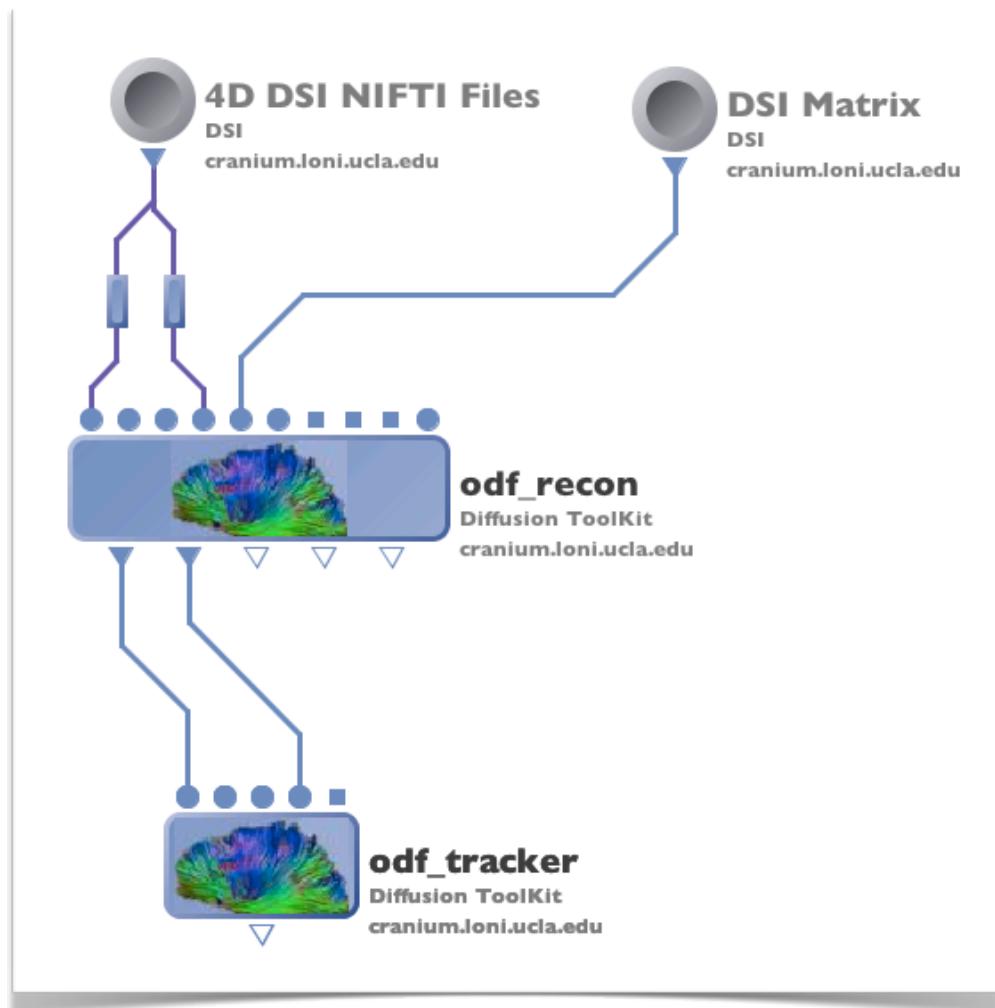
- Input\_type: choose nii option only.
- angle\_threshold: enter the value 35
- mask file
- dsi



# DSI Analyses

## Connect the ODF Recon to the ODF Tracker

- 6) Connect the output parameters, dsioutput and dsi\_dwi\_mask in the odf\_recon module to the input parameters, RECON\_DATA\_PREFIX and mask file in the odf\_tracker module.



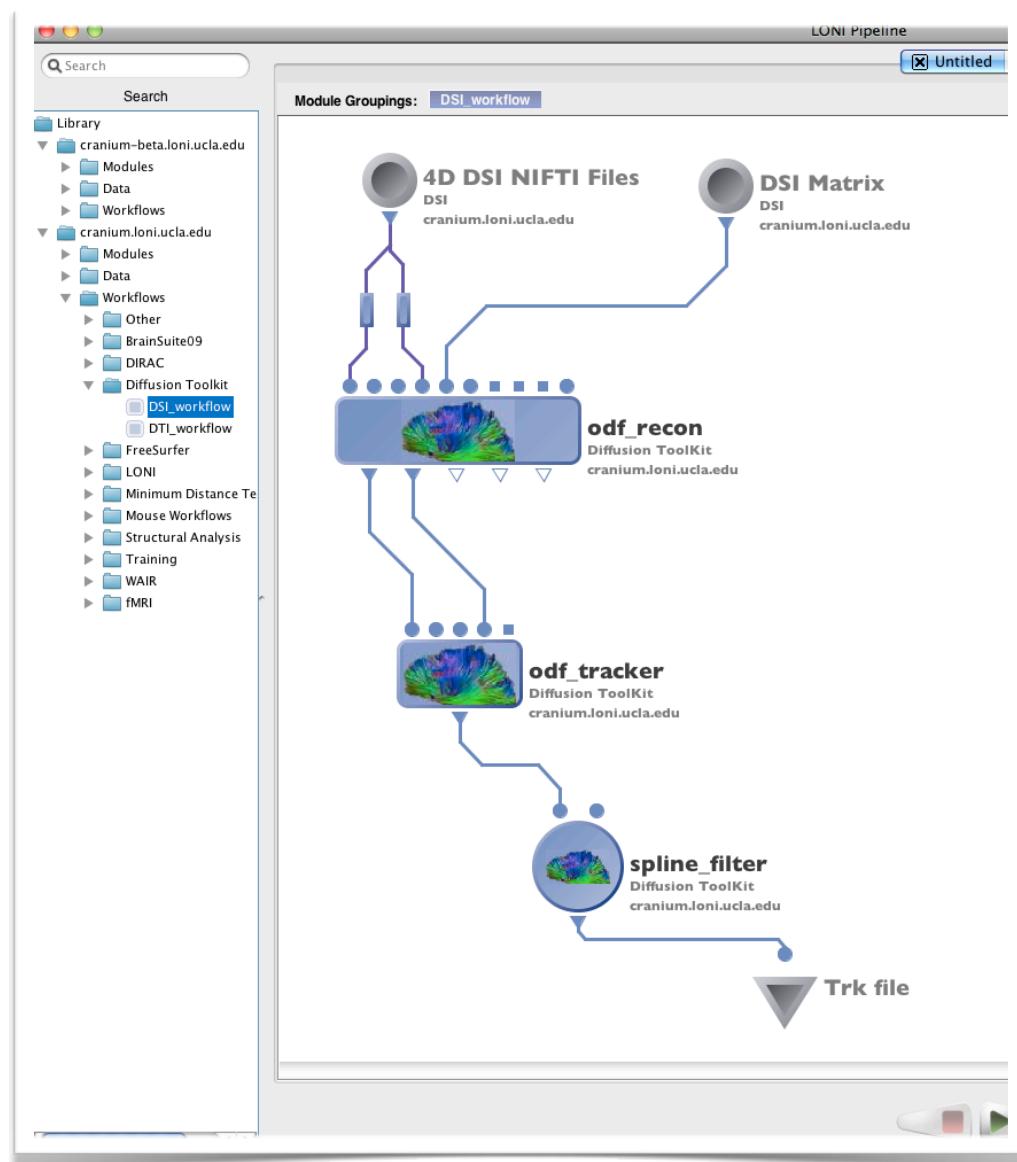
# DSI Analyses

## Spline Filter and Output TRK file

7) Choose and drag the spline\_filter module and connect the output of the dti\_tracker module to the INPUT\_TRACK\_FILE parameter in the spline\_filter module. Choose value 1 for the STEP\_LENGTH option.

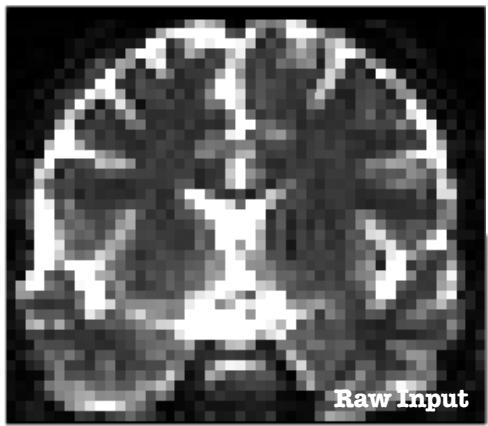
8) Save the output of the spline\_filter a data sink or simply specify the file path.

**NOTE:** To save outputs at intermediate steps you can add data sinks and specify the file path.

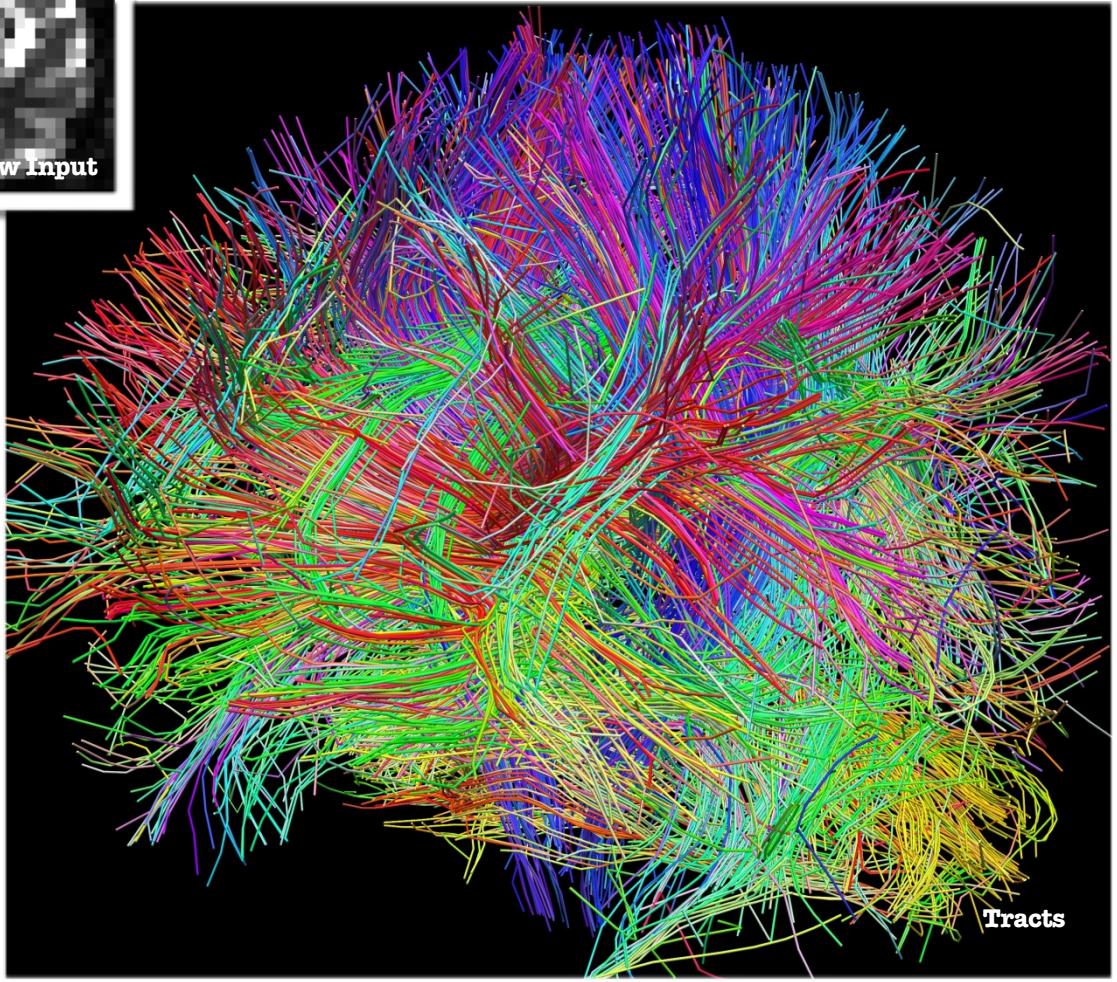


# DSI Analyses

## Results



Raw Input



Tracts

The results of this workflow is a DSI Track file that can be used in TrackVis program for further analysis.

# Advanced Topics

## Building a Module

A module is a simple XML descriptor used to define a binary. When connected to LONI's computer cluster resources, the LONI Pipeline will automatically download the library of modules available on the cluster.

## LONI Pipeline Provenance

Provenance in the Pipeline context equates to history. There are 3 categories of provenance in the Pipeline: Data provenance, binary provenance, and execution provenance.

## Study Module

The Study Module incorporates imaging data and non-imaging meta-data, and enables queries, groupings and construction of study-designs based on user-specified criteria.

## Conditional Module

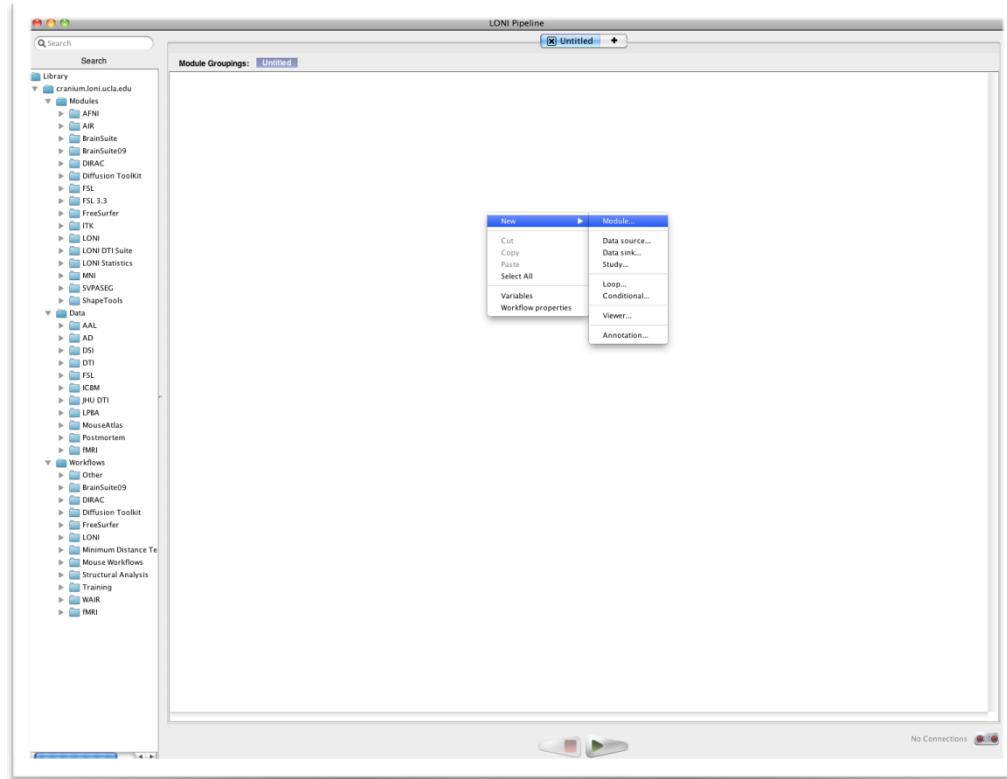
This module is used when the execution path of various inputs to a workflow is dependent on some criteria.

# Building a Module

## Overview

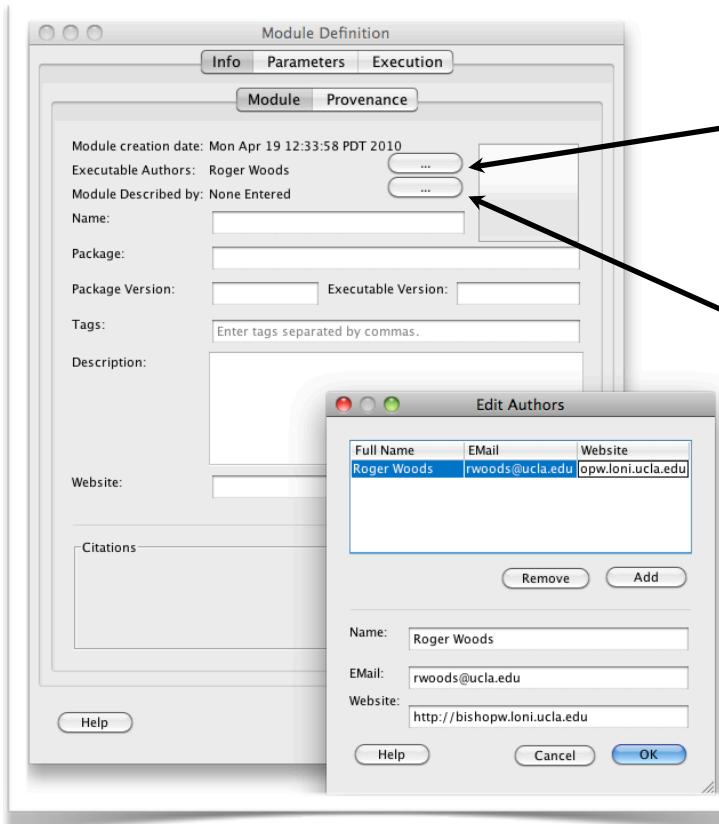
A module is a simple XML descriptor used to define a binary. When connected to LONI's computer cluster resources, the LONI Pipeline will automatically download the library of modules available on the cluster.

Although advanced users may prefer to create their own modules run their executables locally or on their own computing resources, having this list of modules is a great place to begin.



## How-to

- Right click on the canvas
- Choose New Module



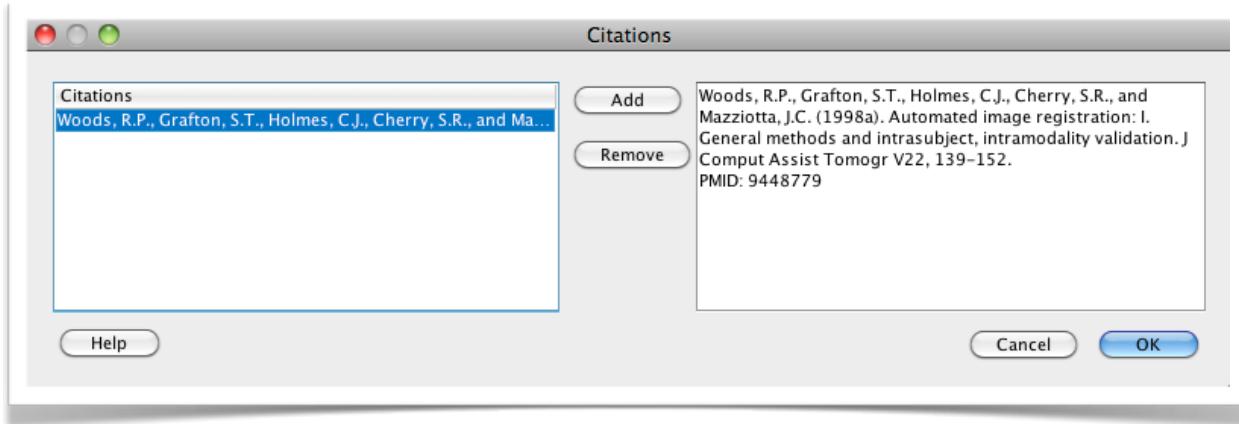
### Optional Actions

- Click Executable Authors button
- Click Add
- Enter Name, email, and website information
- Repeat for all Authors
- Repeat for ‘Module Described by’ button
- Click OK
- In the Name field, enter the name of the binary
- In the Package field, enter the name of the software suite
- In the Package Version, enter the version number of the software
- In the Executable Version, enter the version number of the



### executable

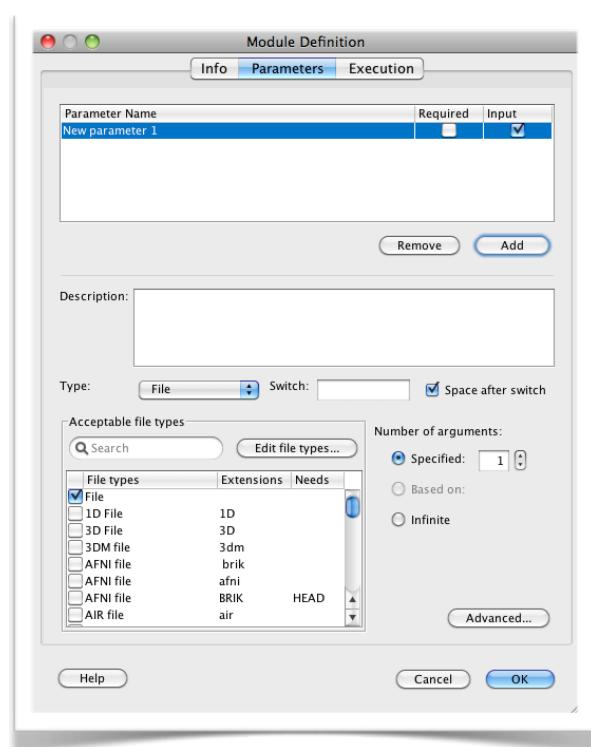
- In the Tags field, enter keywords for the software (eg registration, skull-stripping, tissue classification)
- In the Description field, enter the description of the functionality of the binary
- In the website field, enter relevant website



- Click Edit in the Citations field
- Click Add
- Enter either the Digital Object Identifier or the PubMed ID
- Click OK
- Click the Executable tab
- Enter information for Binary, BinaryConfigOptions, Compiler, SystemConfig, Operating System, Library, Library Compiler, Library System Config, Library Operating System & Library Config Options

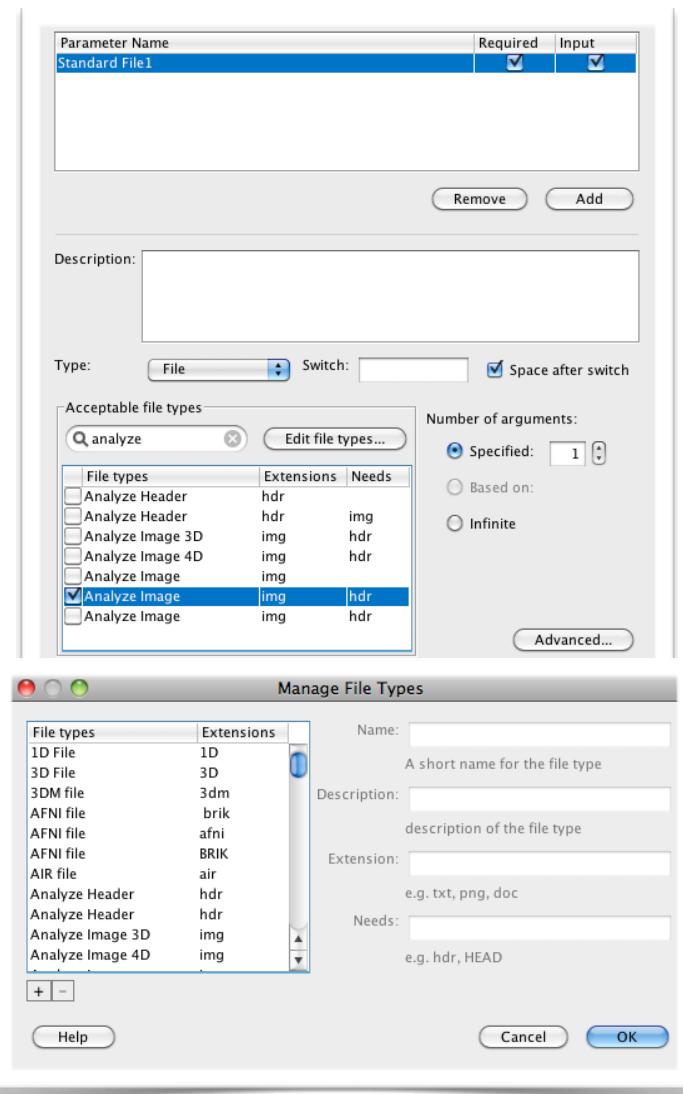
### Required Options

- Click Add under the Parameter Name field
- Double Click on New Parameter 1 and specify name of parameter (eg Standard File) Choose Required if this a required parameter, else, it is optional
- Choose Input if the parameter is of type input, else, it will be output
- In the Description field, describe the functionality of the parameter
- In the Switch field, specify the required flags (eg -m)
- Choose whether the switch required a space to follow or not
- Choose the type of parameter being described, the options are Directory, Enumerated, File, Number, and String

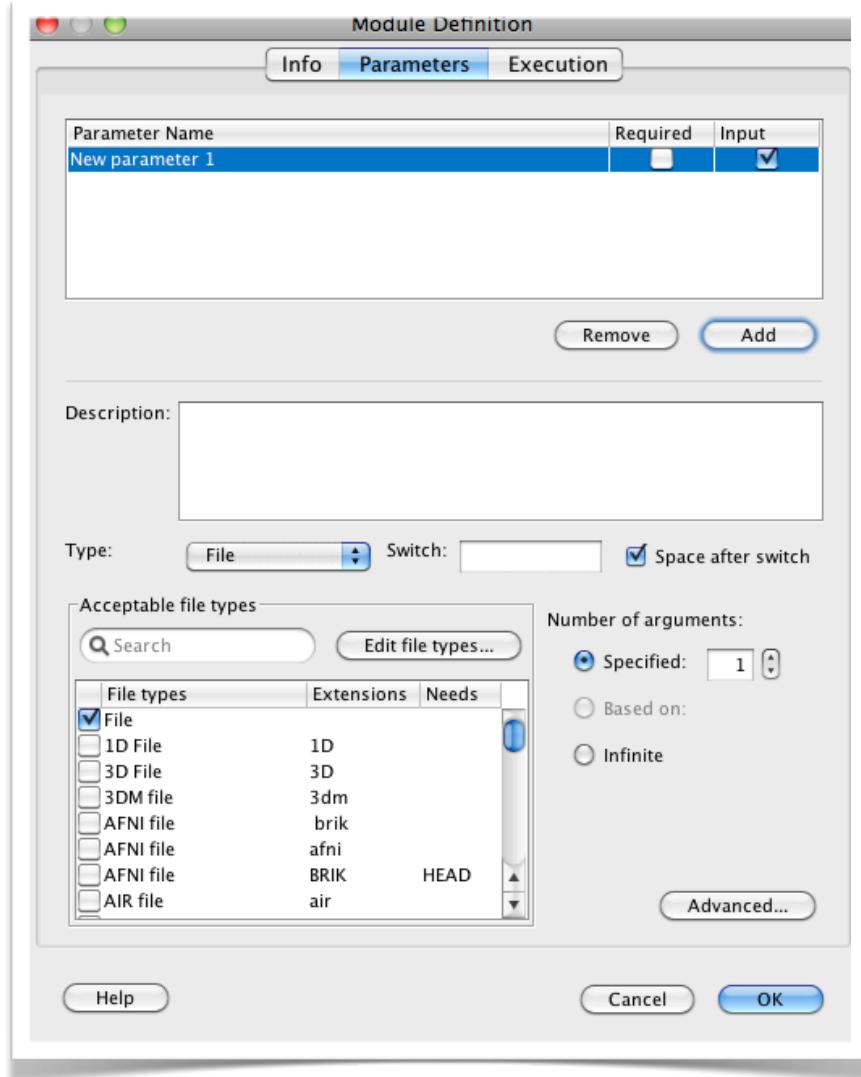


Specify the number of Arguments the parameter expects

- For Enumerated, click Add, double click on value, specify value (NOTE character, strings, or numbers are acceptable; enumeration equates to a drop down menu when selecting the parameter with predefined values)
- For File, please choose the Acceptable file types for the parameter by using the search option (eg search analyze and choose one of the file types) NOTE: You may create your own file type.
- Choose Edit file types, click the + icon
- Type the name of file type (eg, Analyze Image)
- In the Description field, include a description
- In the Extension field, included appropriate extension (eg .img)
- In the Needs filed, include dependancies (eg .hdr)



## Adding additional parameters



header (.hdr) is preserved. This is required when moving files from and to various directories for execution.

Choose **Add** in the Parameter Name box. If the file is required for the binary to execute, choose **Required**, else, it's considered optional, and users may elect to use the parameter. If the parameter is an input, choose **input**, else, the parameter is considered an output. As with each parameter, one must specify the type of the parameter. The acceptable types are **File**, **Directory**, **String**, **Character** or **Enumerated**. With type **File**, please specify the expected file type; for example, analyze image, as the dependency of an analyze file with a

Specify dependencies and manipulate file names

The LONI Pipeline allows you to specify dependancies between parameters. For example, if using MNI's **-segment** (part of mincmath) one must also specify **-const2** to specify a min and max range to extract. Select the parameter, then click **Advanced**, the in the

### Select Dependencies

window, choose **Needed** for the required parameter, click **OK**.

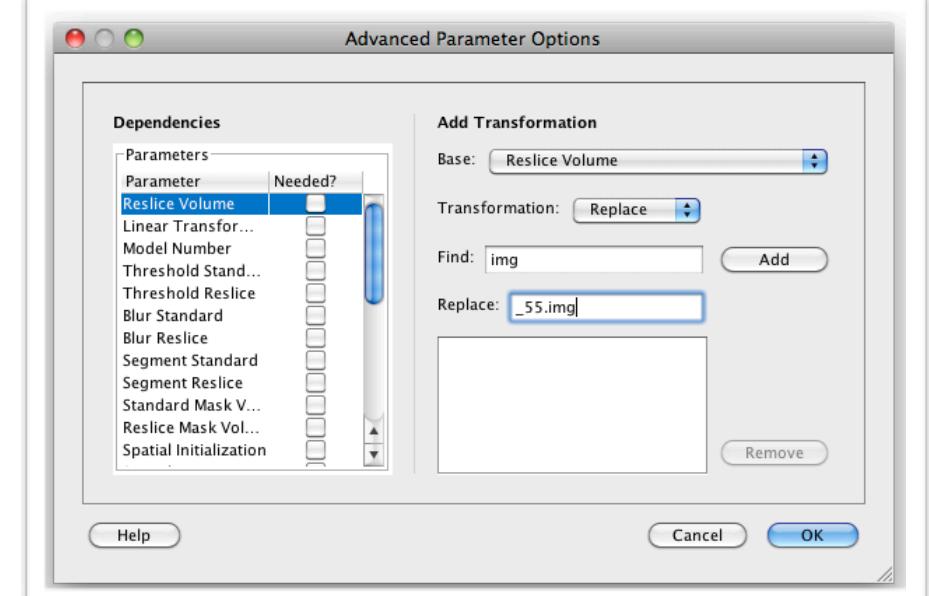
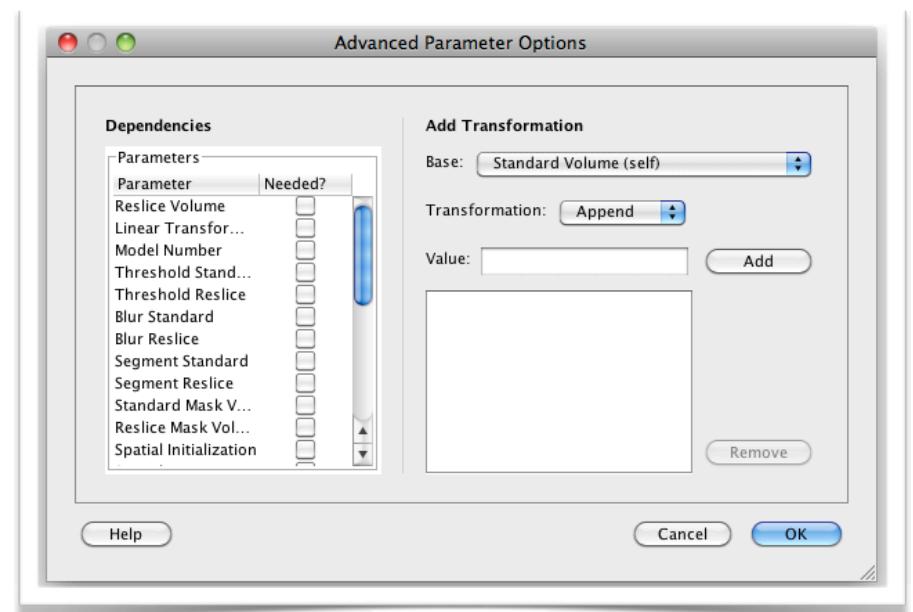
Additionally, the Pipeline allows for file name manipulations. For example, using

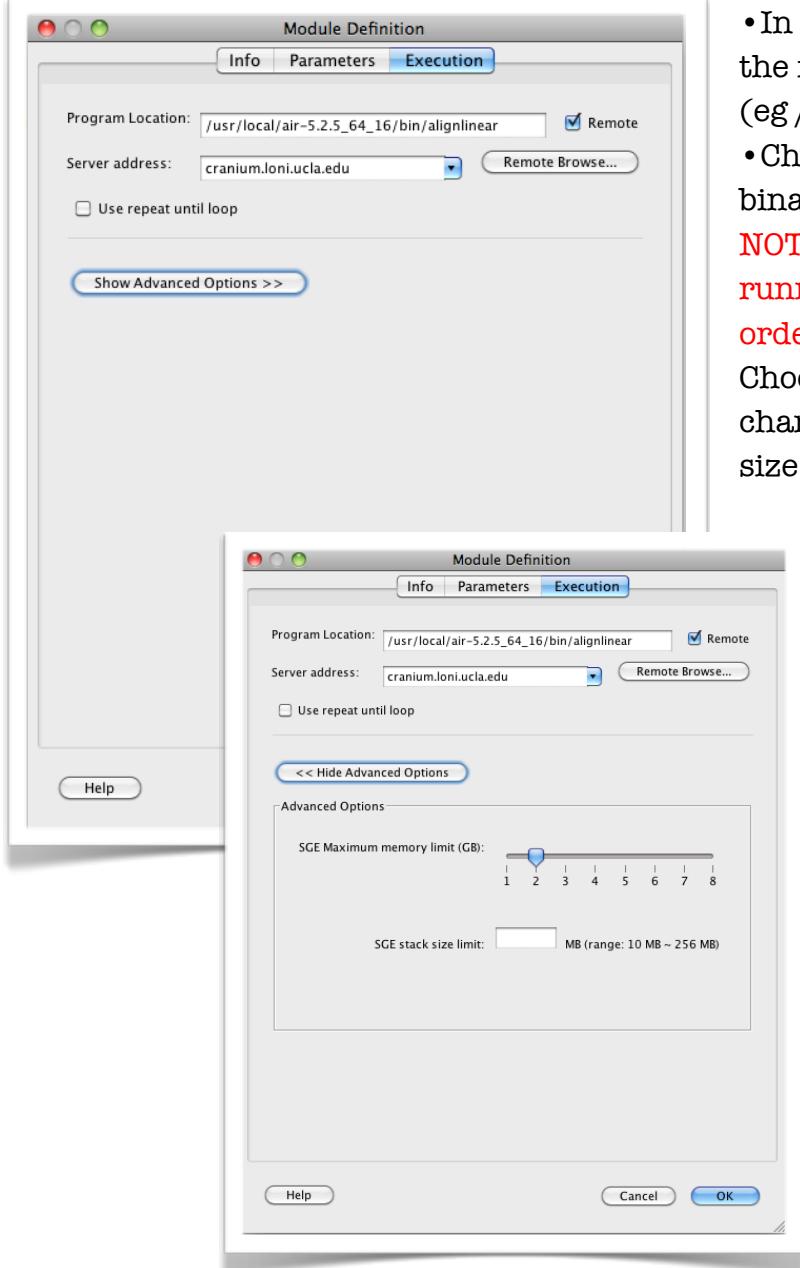
**-m** (provide skull stripped mask) which is a part of FSL BET will result in

<root\_name>\_mask.img & <root\_name>\_mask.hdr files being generated in the same directory. If one were then to connect the output to another module, and execution error would occur.

In this instance, choose the parameter, click Advanced, provide the Base (this is the parameter name whose base you'd like to manipulate, choose the Transformation (**append**, **prepend**, **replace**, **subtract**), choose the text to **Find**, and the text to **Replace** the text with, and

hit **OK**. This will result in the type of transformation specified.





- In the Program Location field enter the full path and binary name (eg /usr/local/air/bin/alignlinear)

- Choose remote and specify server if binary resides on a remote server

**NOTE: A Pipeline server must be running on the specified server in order for the client to connect**

Choose the Show Advanced Options to change the memory limit or the stack size limit.

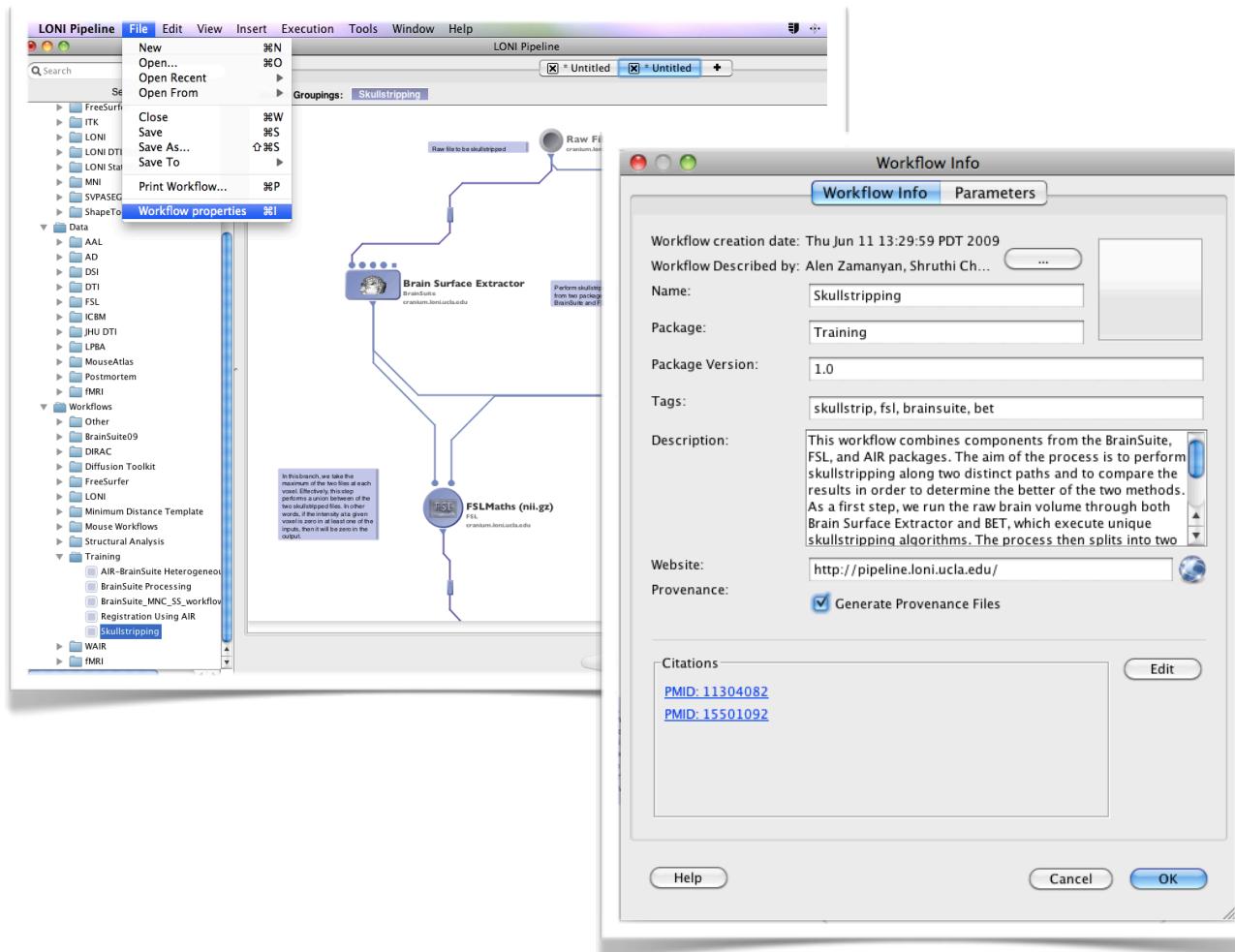
# LONI Pipeline Provenance

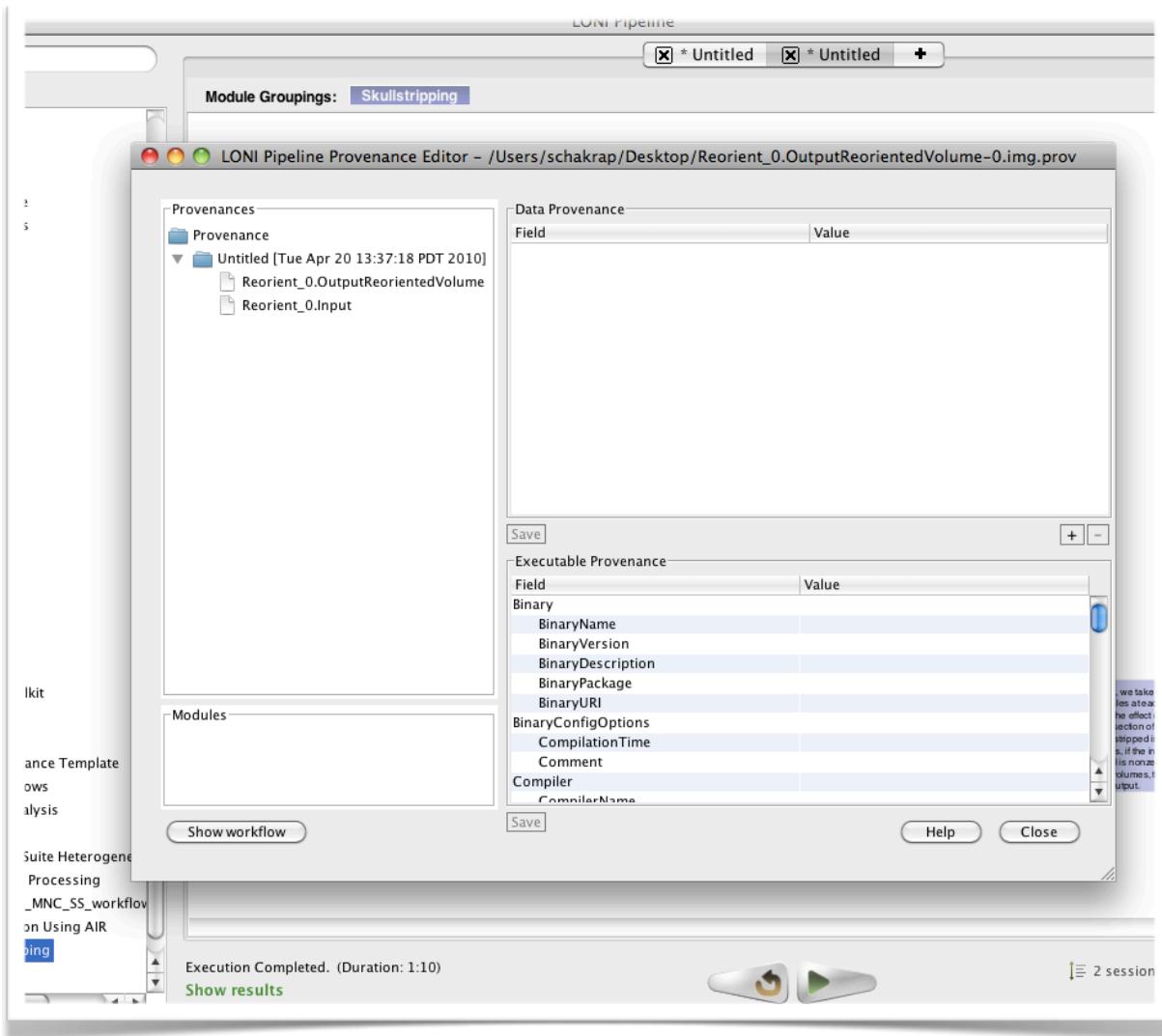
## Overview

Provenance in the Pipeline context equates to history. There are 3 categories of provenance in the Pipeline: Data provenance, binary provenance, and execution provenance. Data provenance is a snapshot of the state of the data pre & post manipulation by a binary. Binary provenance is a record of all the factors influencing the creation of the binary, examples of which include the OS, the compiler used, the library dependancies, and the optimization flags. The execution provenance is a record of the list of executables the data has gone through.

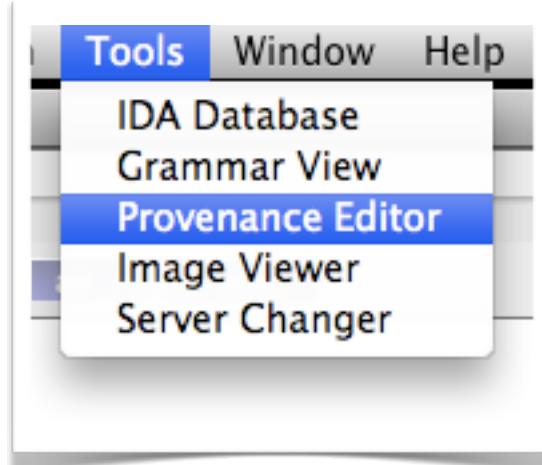
## How-to

- To generate a provenance file, go to File, choose Workflow Properties
- Under Provenance, choose Generate Provenance Files

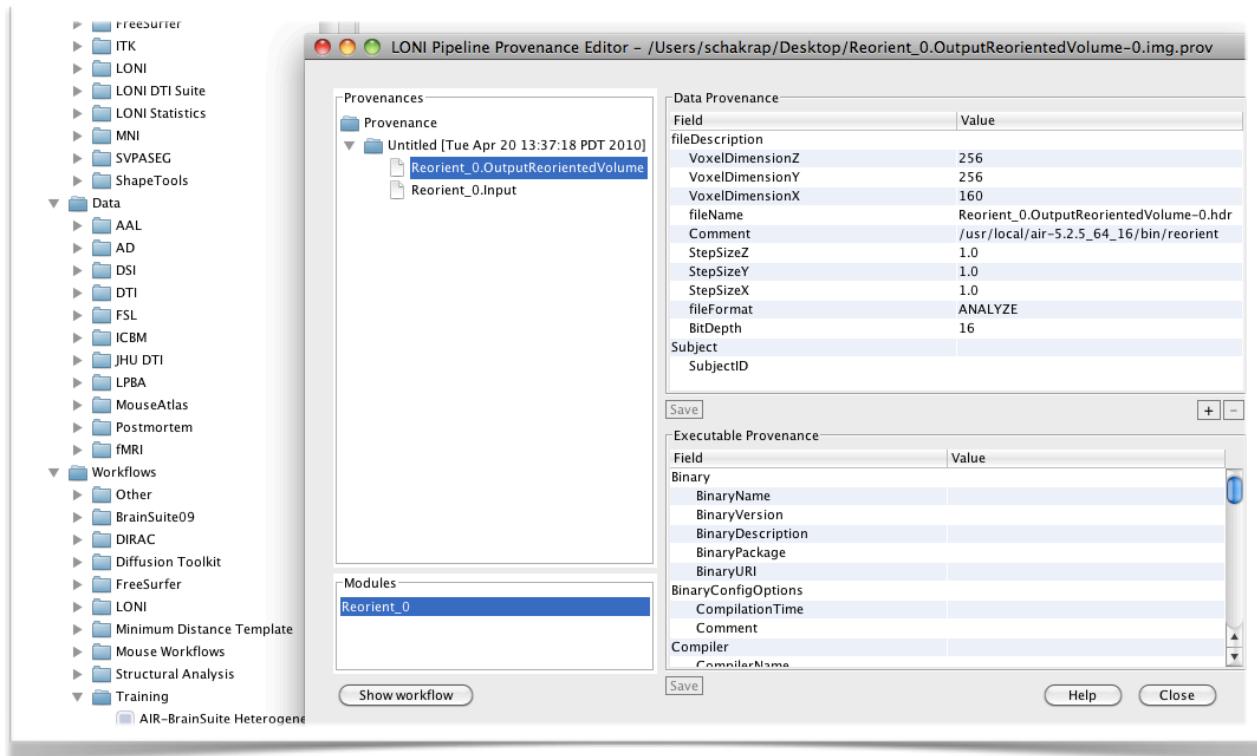




- Execute workflow
- The Pipeline now generates a .prov file for each file generated by a binary
- Download this file
- Under Tools, choose Provenance Editor



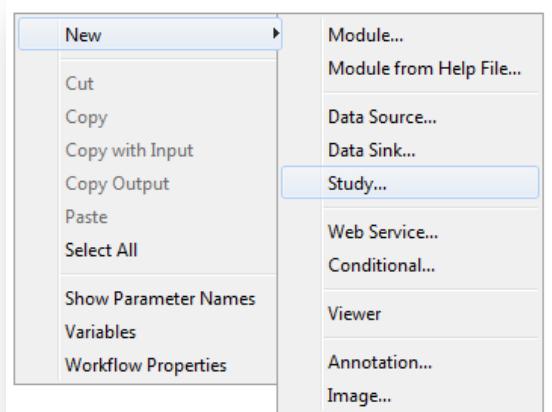
- Under provenance, expand the directory
- Under Data provenance, you will find all metadata stored in the header
- To edit or add any additional field, choose the + icon
- Choose Category or field, edit accordingly
- Under Executable Provenance, you'll find all binary provenance
- Click on Show workflow, this will generate the Pipeline used for processing the data



# Study Module

## Overview

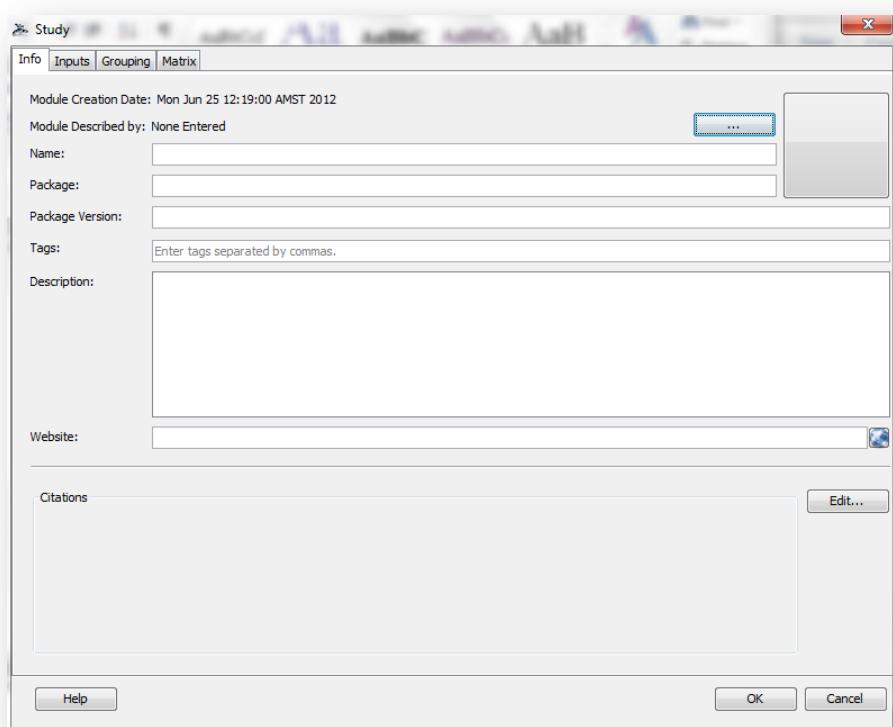
Study Module is a new feature in the Pipeline 5.0 version. It incorporates imaging data and non-imaging meta-data, and enables queries, groupings and construction of study-designs based on user-specified criteria. A study module is similar to a data source and can be connected to the input parameters of other modules in the same workflow. It allows passing both imaging data and the metadata information to subsequent modules and all of this study information may be passed along throughout the Pipeline workflow. The metadata may be used for setting up various conditional criteria in conditional modules. The metadata information may be represented as an XML file, as long as its schema is valid (well-formed) and consistent (uniform for every subject in the study), or as a tabular spreadsheet (CSV).



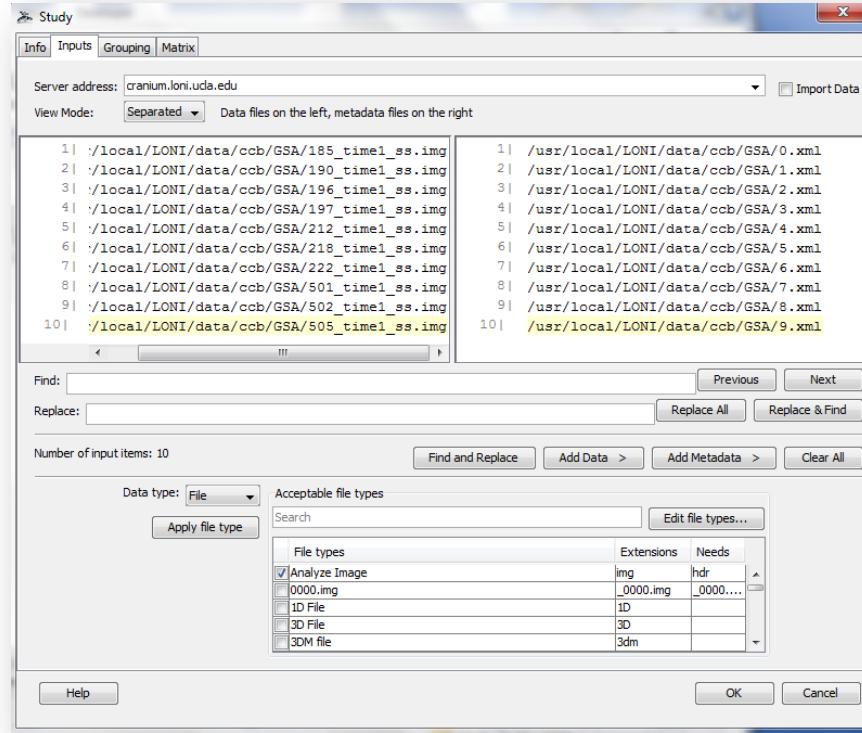
## How-to

A study module can be created by right clicking on the canvas and selecting “Study” under the “New” option.

A new dialog will appear which has four tabs. Under the Info tab, basic information about this module is added, such as module name, package, description etc, similar to other modules.



Under the Input tab, data, metadata file and file formats are specified.



Under the Grouping tab, different groups can be created based on some criteria (conditional expressions) on the metadata.

The screenshot shows the 'Study' window with the 'Grouping' tab selected. On the left, a tree view shows three groups: 'All Groups' (containing 10 items), 'Group 1 (4)' (containing 4 items), 'Group 2 (3)' (containing 3 items), and 'Group 3 (3)' (containing 3 items). At the bottom are buttons for 'Make each group as an output parameter', 'Reload', 'New Group', 'Generate Groups', 'Help', 'OK', and 'Cancel'. On the right, an 'XML Tree View' window is open, showing the XML structure of a file named 'C:\Users\AZAMAN~1\AppData\Local\Temp\17.xml'. An 'Add Variable' dialog box is also open, prompting for a variable name ('group') with 'group' entered. Buttons for 'OK', 'Cancel', 'Add XPath as matrix column', 'Add XPath as variable', and 'Close' are visible.

Under the Matrix tab, a table displays the associated meta-data. The columns relate to your specifications. The rows represent values for each subject.

The screenshot shows the LONI Pipeline software interface with the 'Study' module selected. The 'Matrix' tab is active. At the top, there is a 'Columns' input field containing '{sex}, {sn}, {age}, {educ}, {group}' and a 'Generate Matrix' button. Below this is a 'Result Matrix' table with 10 rows and 7 columns. The columns are labeled 'Index', 'Metadata', '{sex}', '{sn}', '{age}', '{educ}', and '{group}'. The data in the table is as follows:

Index	Metadata	{sex}	{sn}	{age}	{educ}	{group}
1	0.xml	1	185	72.6	16	1
2	1.xml	0	190	66.7	17	2
3	2.xml	1	196	63.3	17	1
4	3.xml	1	197	64.1	17	2
5	4.xml	0	212	72.2	18	1
6	5.xml	1	218	63.5	16	1
7	6.xml	0	222	63.4	16	2
8	7.xml	0	501	57.5	14	3
9	8.xml	1	502	56.8	13	3
10	9.xml	1	505	56.6	16	3

At the bottom of the window, there are buttons for 'Save metadata as CSV...', 'Export matrix to CSV...', 'Help', 'OK', and 'Cancel'.

NOTE: For more detailed information about the Study Module, please see the Pipeline website:

<http://pipeline.loni.ucla.edu/support/user-guide/building-a-workflow/#Study>

# Conditional Module

## Overview

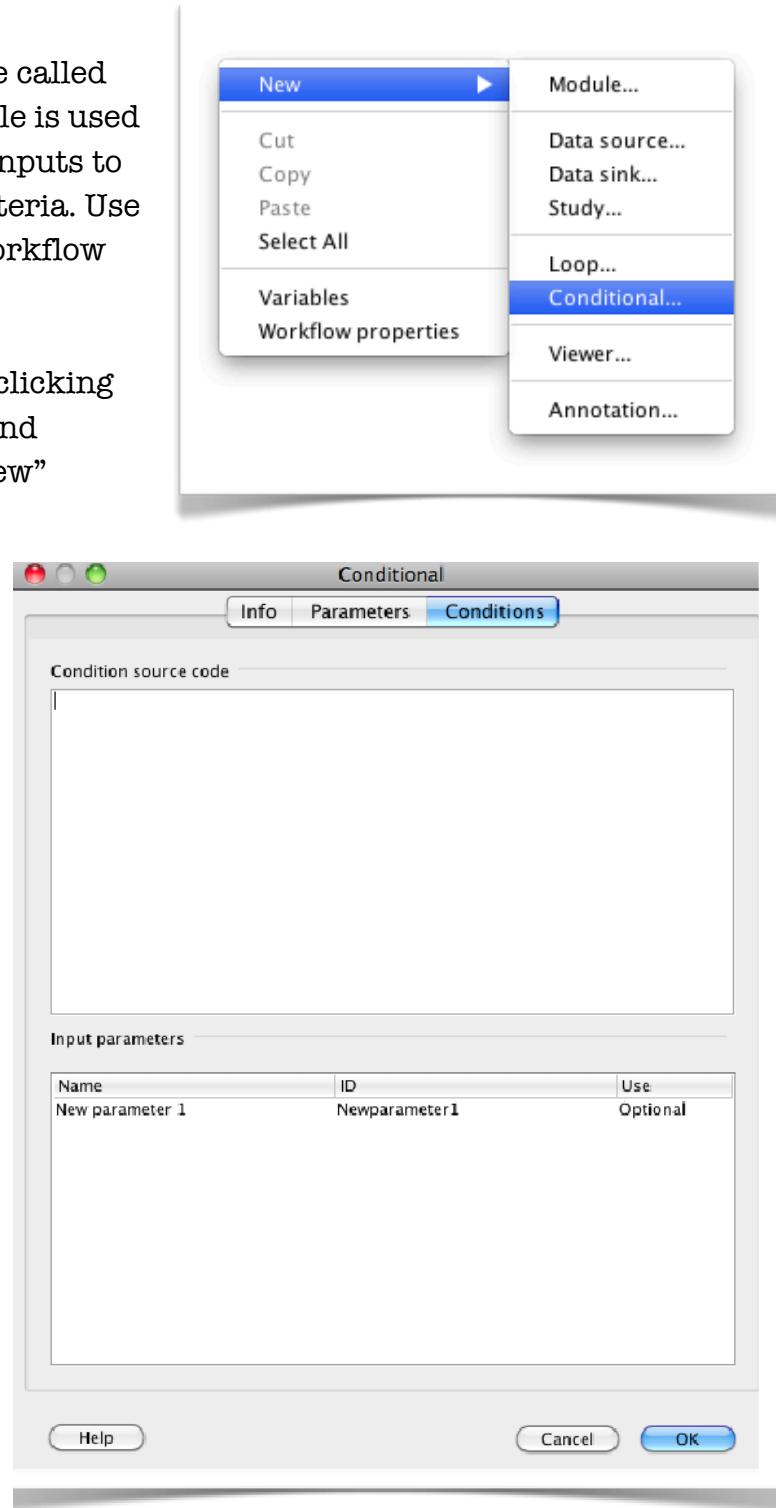
Pipeline 5.0 introduces a new feature called the “Conditional Module”. This module is used when the execution path of various inputs to a workflow is dependent on some criteria. Use of Conditional Modules makes the workflow more dynamic.

This module can be created by right clicking on the empty area in any workflow and choosing “Conditional” under the “New” option.

A new dialog will appear that has three tabs. The first and second tab is similar to what is seen in other type of modules. The third tab is different and is called "Conditions". Under this tab there is a “Condition source code” section where the conditional criteria should be entered. The syntax of the code entered is the same as the Pipeline Programming Language (PPL), which is similar to Java/C. Pipeline programming language is very simple and easy to learn.

## How-to

The following examples will help better understand the functionality of the Conditional module.

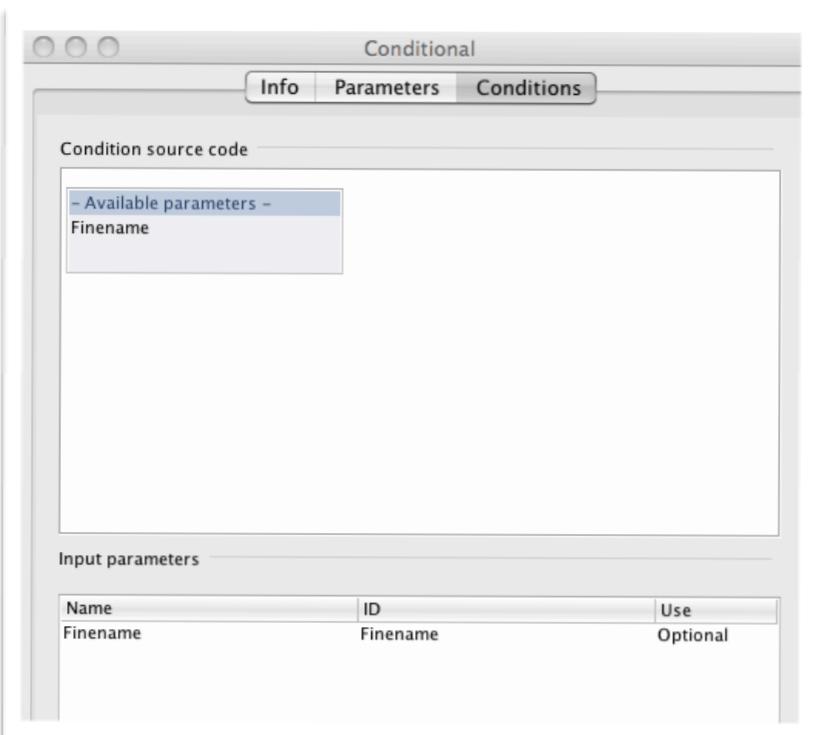
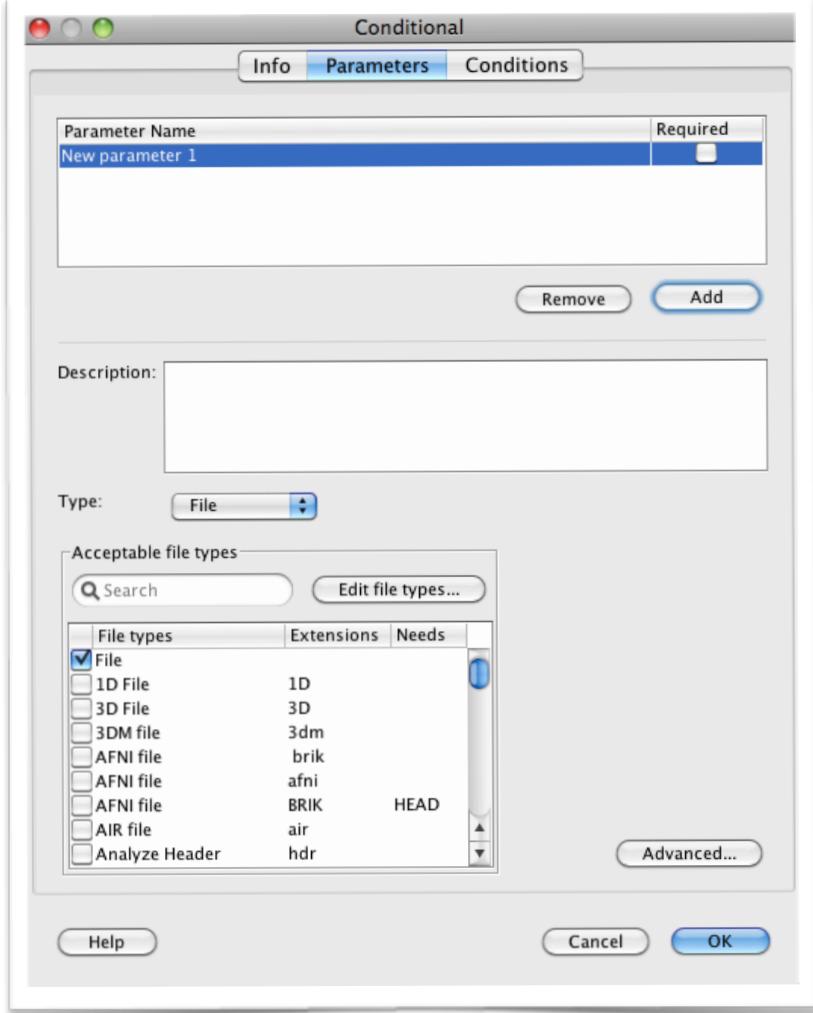


## File Conditions

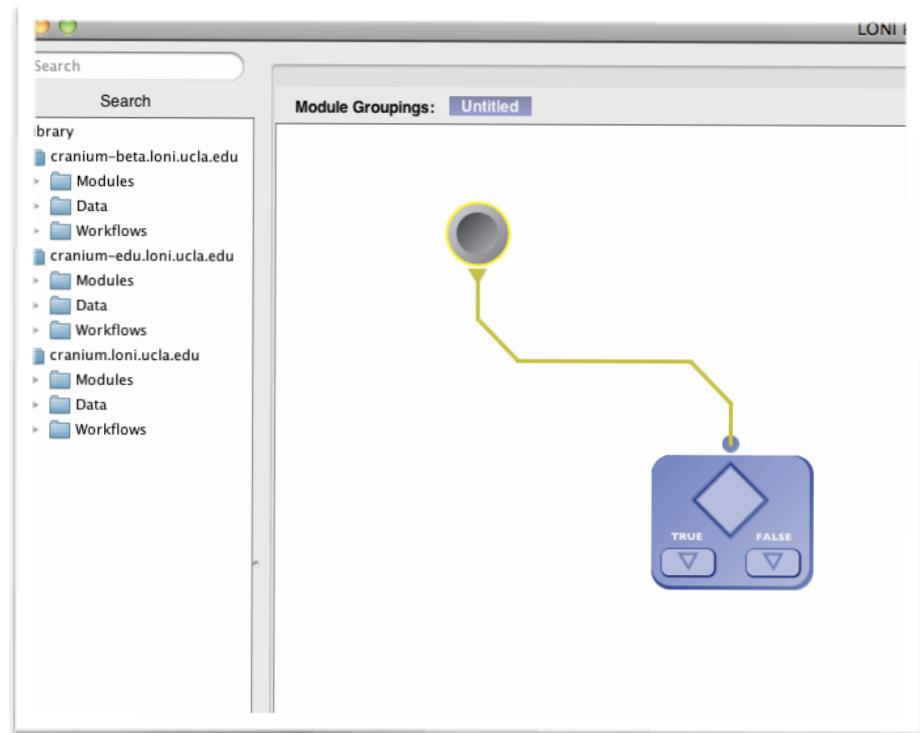
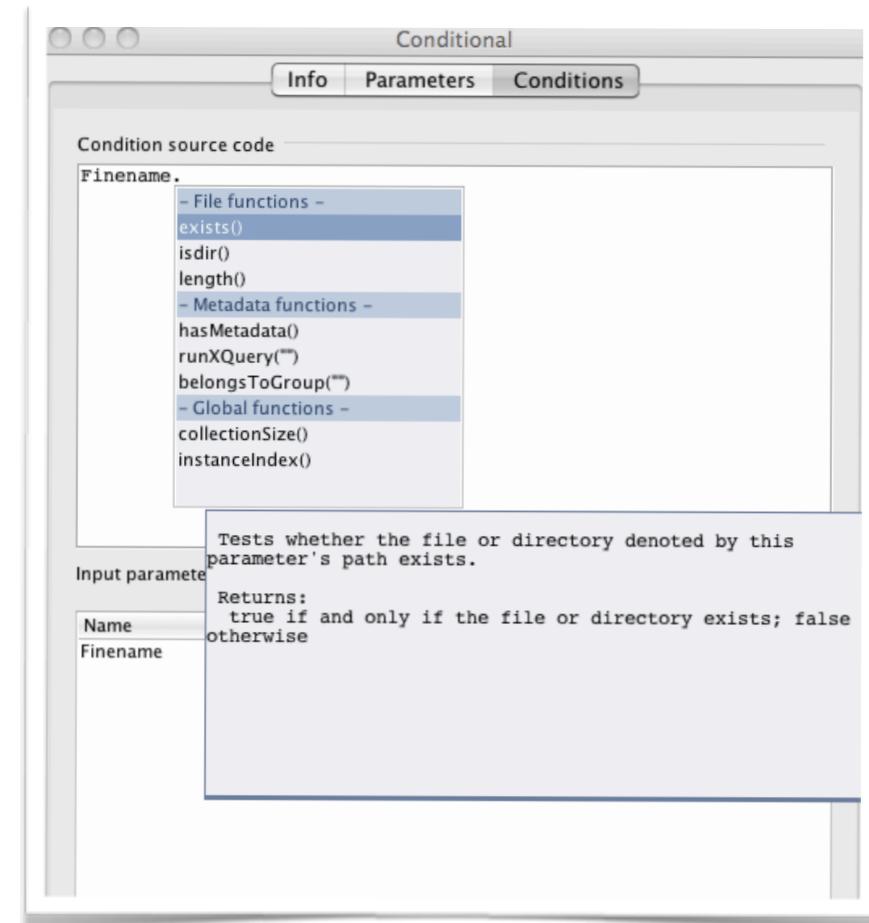
This example will help understand how to set up a conditional module that chooses the execution path based on whether a file exists at a specific module output.

To create this conditional module follow the steps below:

- 1) Right click on the empty area in any workflow and select "New->Conditional"
- 2) Click on the "Parameters" tab and click on "Add" button to create a new parameter. Name the parameter, for example as "inputFile". Choose the file type if needed and click "OK".
- 3) Go to the "Conditions" tab and click on the "Condition source code" area and press the F1 Key to see a list of available parameters (**NOTE: If there are no parameters declared, then there will be no parameters displayed when F1 key is pressed. New parameters have to be defined for the current conditional module before the conditional source code is specified**). Choose the "inputFile" parameter by double clicking on this option.



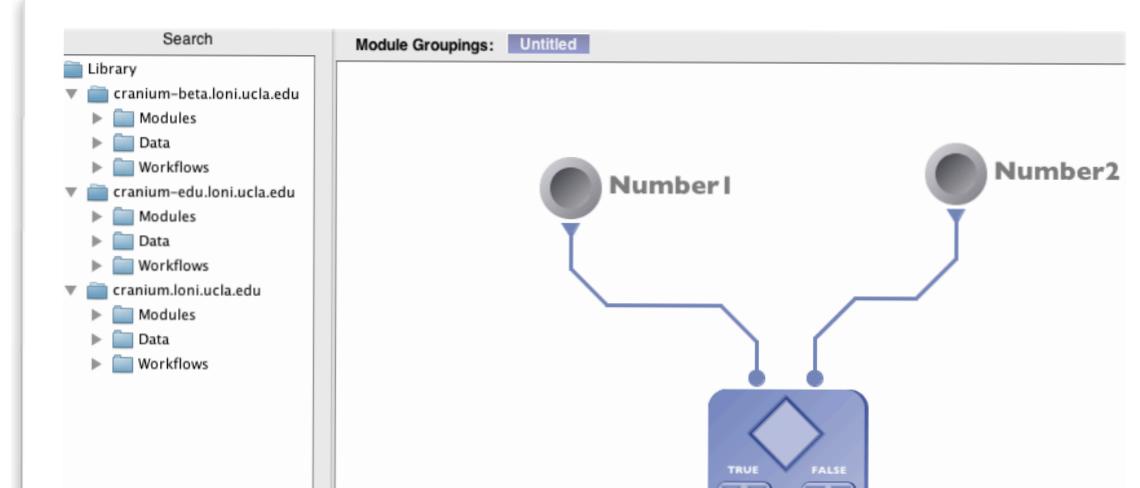
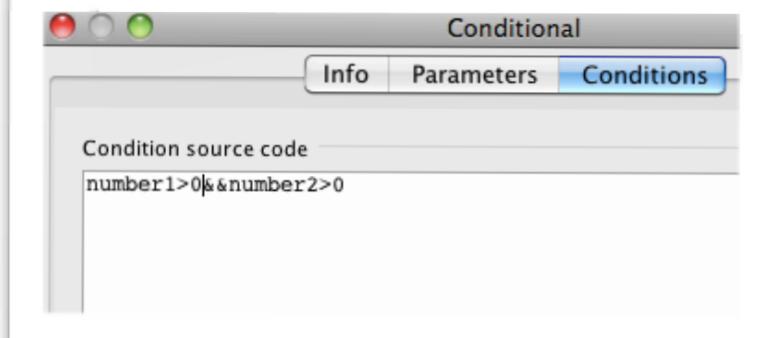
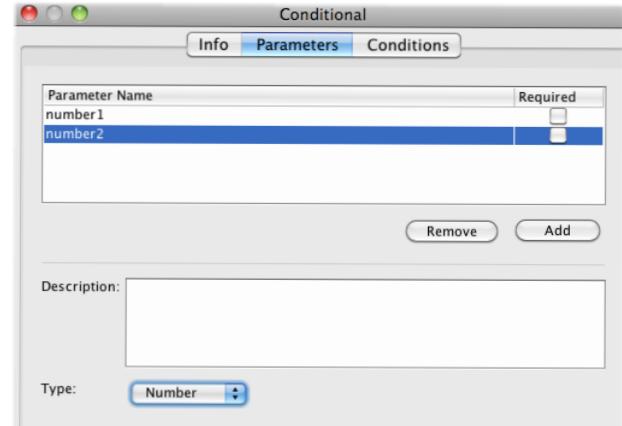
- 4) Enter a “.” after the inputFile (inputFile.) to access the various functions. Choose “exists()” under the “File functions” option by double clicking on it (inputFile.exists()). This condition checks if the parameter “inputFile” exists.
- 5) Click OK and a new conditional module is created with one input and two outputs, “TRUE” and “FALSE”. If the parameter “inputFile” exists then the conditional will feed the inputFile to the “TRUE” output Parameter else to the “FALSE” output parameter.
- 6) Other functional modules can be connected to the outputs of TRUE/FALSE accordingly. If one output is always used, the other output could be disabled like any other module output.



## Arithmetical/Comparison and Metadata Conditions

This example will demonstrate how to check the value of a parameter and determine if it has positive value (this number could be output of a previous module or information in the metadata).

- 1) Follow the previous example until the step where the parameters are defined. Click on edit and create 2 input parameters, number1, number2 and both are of type "Number". Click "OK".
  - 2) In the "Condition source code" area type the following "Number1 > 0 && Number2 > 0".
- This condition will check if both the inputs, Number1 and Number2 are greater than zero i.e. positive values. The conditional module created thus will have two inputs (Number1 and Number2) and four outputs (2 True and 2 False).



# Distributed Pipeline Server Installation

## Introduction

Setting up your own Pipeline server is a great way to remotely take advantage of the power of a cluster or a just a dedicated computer with many helpful programs installed on it. More importantly, you can enable many people to take advantage of all this power all through the easy to use interface of the Pipeline client.

## GUI Installation

The Distributed Pipeline Server Installer is a GUI installer that allows you to install and configure 3 types of resources – backend grid management resources (Oracle Grid Engine), the Pipeline server, and a number of computational imaging and informatics software tools. After successfully running the installer, you will have a running Pipeline server with Oracle Grid Engine managing jobs on your machine(s), imaging and informatics software tools installed, as well as a set of predefined workflows and modules in your server library.

## Requirements

The requirements for the Pipeline server installation:

One or more machines running CentOS Linux operating system version 5.4 or above

OpenJDK or Sun/Oracle JDK 1.5 or above (If Sun/Oracle JDK is not installed, our installer will guide you to install it)

NFS V.3 server and network connected nodes with a shared partition

At least 30 GB available space for full installation (including tools and data)

Remote Root Access (with SSH credentials) to all hosts (to install Grid Engine)

A user that will run the Pipeline server process (a dedicated user is recommended)

Static IP Addresses for all the hosts, and their hostnames properly configured on DNS

Internet connection is needed during the installation

Warning: if any of the requirements is not met, there may be unexpected behavior (e.g. hanging, crashing of the installer). If you have any questions, please contact [pipeline@loni.ucla.edu](mailto:pipeline@loni.ucla.edu)

A complete installation (including Grid Engine, the Pipeline server, and download and install all the tools) takes approximately 6-7 hours. Some of the tools take a long time to download (e.g. FSL takes 6 hours to download). If you skip the tools or have all the tools downloaded, the total installation time is less than 30 minutes.

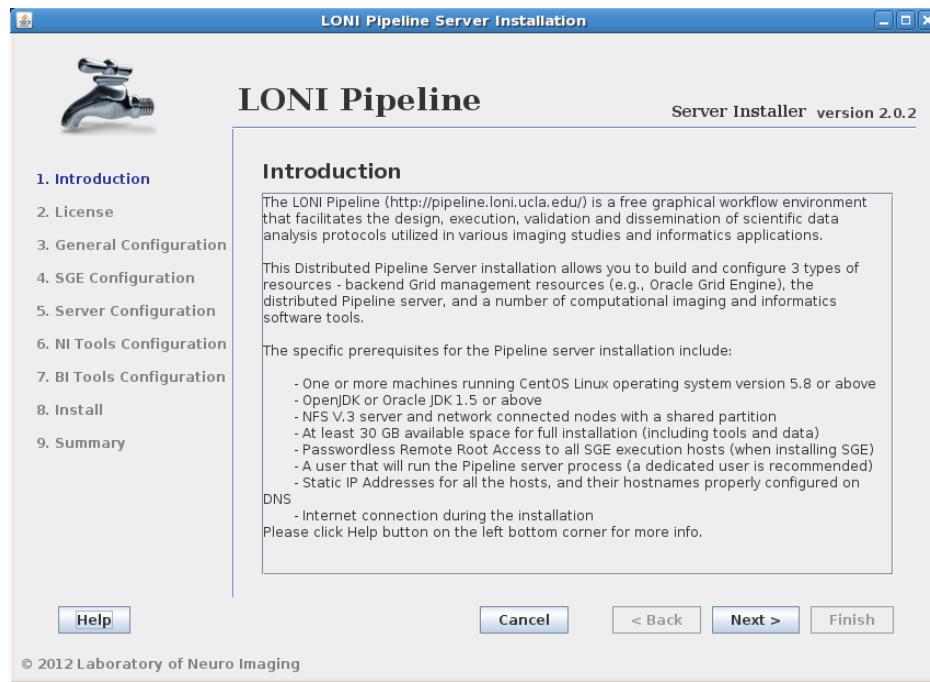
## Downloading

Download the installer from the Pipeline website, under Downloads > Distributed Pipeline Server Installer.

## Start the Installer

To start the installer, open a Terminal, sudo as root, go to the directory where the installer file is located, and type

```
tar -zxvf pipelineServerInstaller.tar.gz  
cd pipelineServerInstaller  
../launchInstaller.sh
```



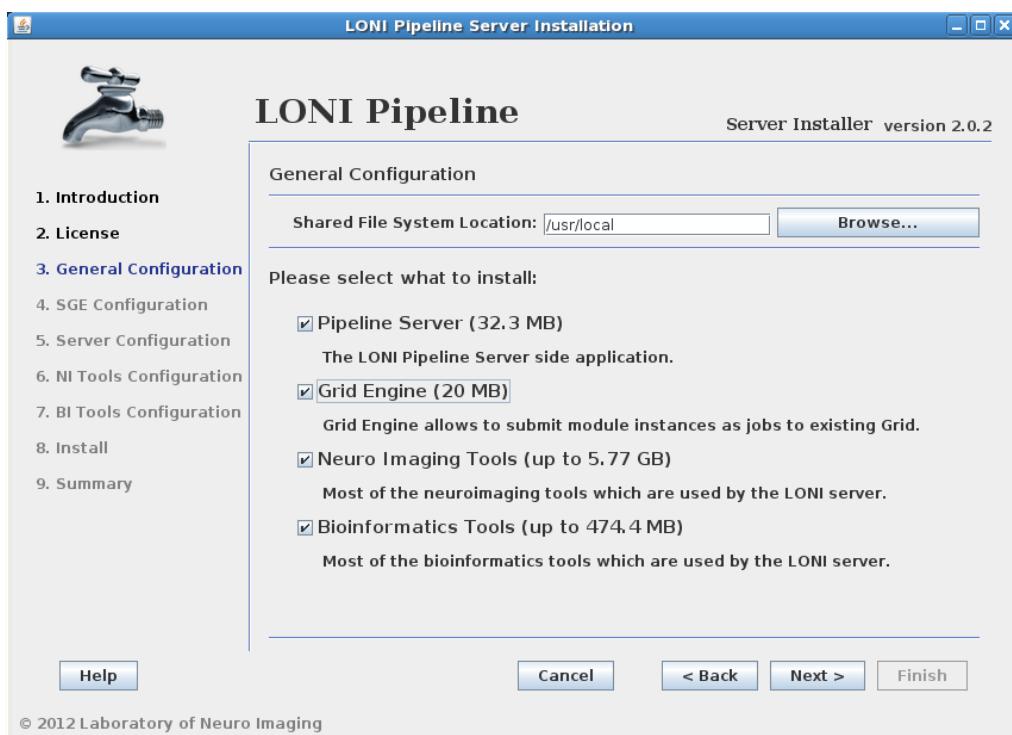
## Select Components

After you read and agree the license, it asks you the installation location and what components you want to install:

You can select **any\*** or all of the components. It will guide you through all the steps needed for the installation.

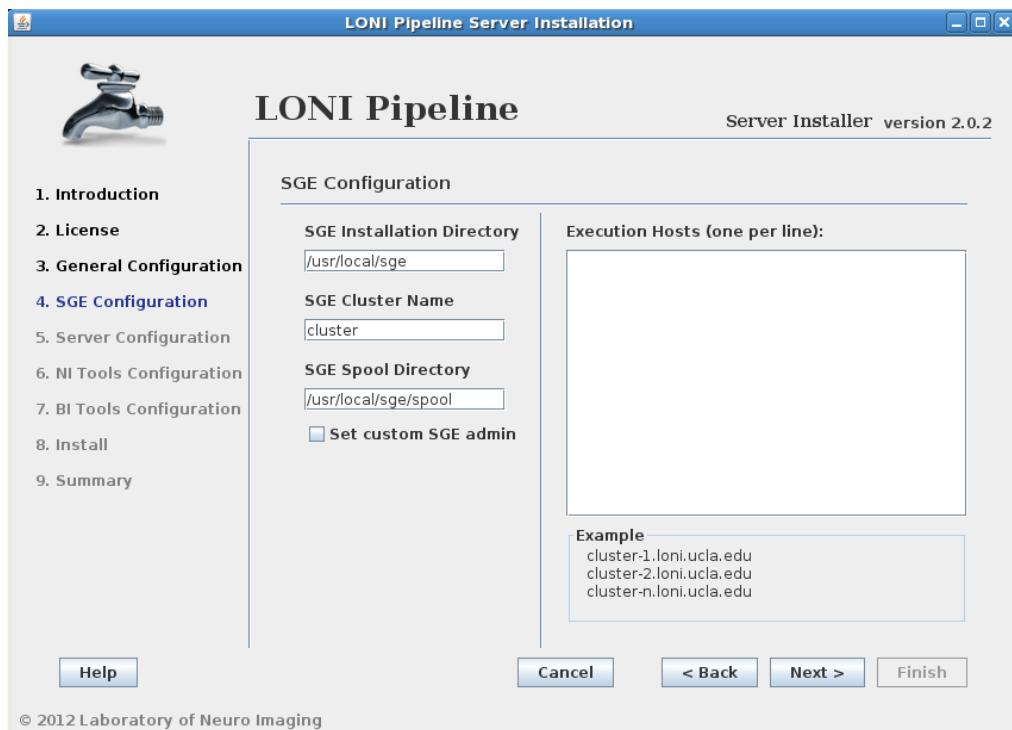
\* For example, if you have already installed SGE before launching this installer, then deselect the Sun Grid Engine component. Likewise, if you only want to install the latest tools, you can select the Neuro Imaging Tools component and uncheck the rest.

The installer will verify the Shared File System Location given. It is required to have it on NFS if the server is set to use a grid. The shared file system is used for the Pipeline server to store intermediate files of workflows and to install Grid Engine and Neuro Imaging Tools.



## Install Grid Engine

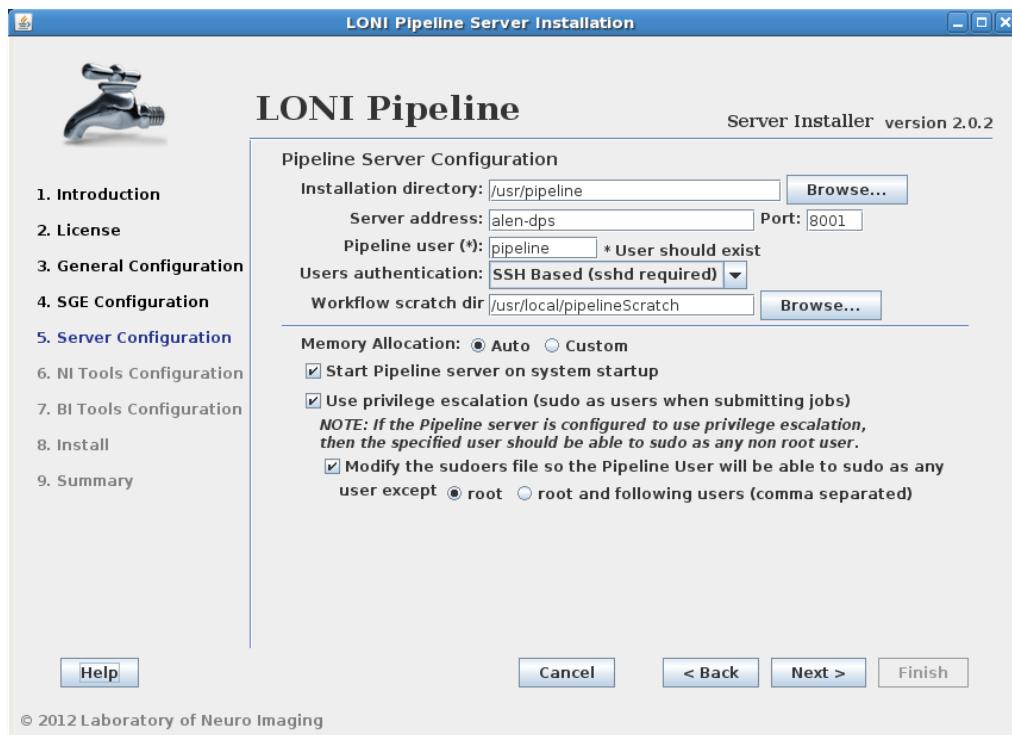
In this section you can configure Grid Engine installation. You can specify installation location, cluster name, spool directory, and execution hosts. You can leave installation location, cluster name and spool directory as they are, but you must provide a list of hostnames. You must provide fully qualified domain names, so something like “host1”, “localhost” or “127.0.0.1” is not allowed.



## Install Pipeline

In this section you can configure the Pipeline server. You can specify installation directory, Pipeline server address, port and user to run the Pipeline server process. The user has to be created and you can have the option to have its sudo file modified to accommodate privilege escalation.

If modify sudo option is selected, the installer will modify operating system's sudoers file so that the Pipeline server user will be able to sudo as any user, except root and a list of users provided. For example, if you have some user that can sudo as root, then this user should be listed in the exception list, so that the Pipeline user will not be able to sudo to and ultimately gain root access.

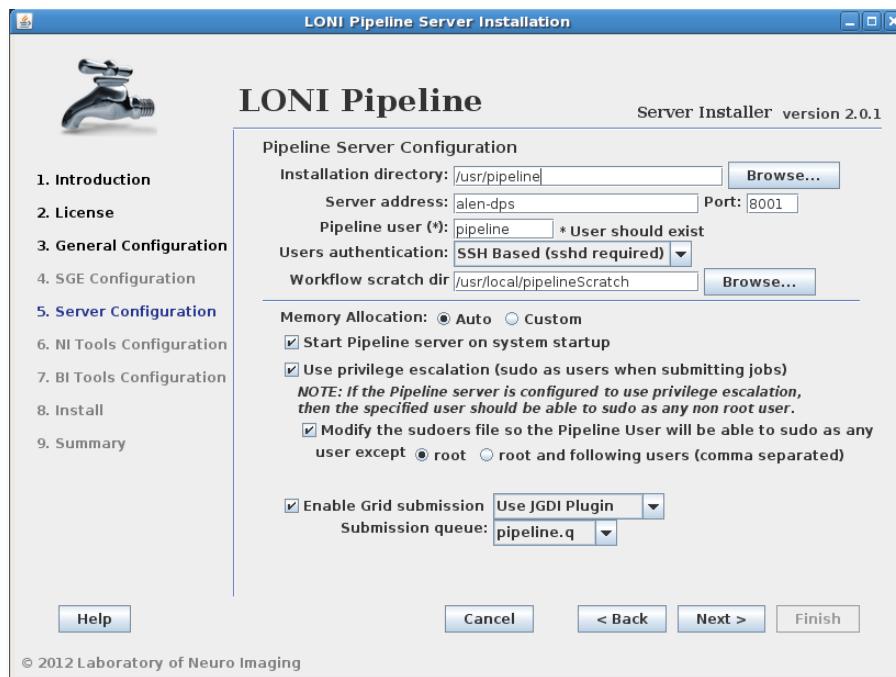


## Install Pipeline with already installed SGE

If you have deselected the Sun Grid Engine checkbox from step 3 (General Configuration), then the Pipeline configuration window will have slightly different look. There should be a checkbox “Enable Grid Submission” which needs to be checked if you want to use Pipeline with your pre-installed SGE.

Upon checking the “Enable Grid Submission” checkbox, a combobox with two options should appear. In order to communicate with SGE, Pipeline uses Grid Plugins. LONI Provides two plugins for SGE and those are JGDI Plugin and DRMAA Plugin. If you are using SGE we highly recommend to use JGDI Plugin as it supports more Pipeline features and is more reliable. You can choose DRMAA Plugin if you have other DRMAA supported Grid Manager installed and want to integrate Pipeline with it.

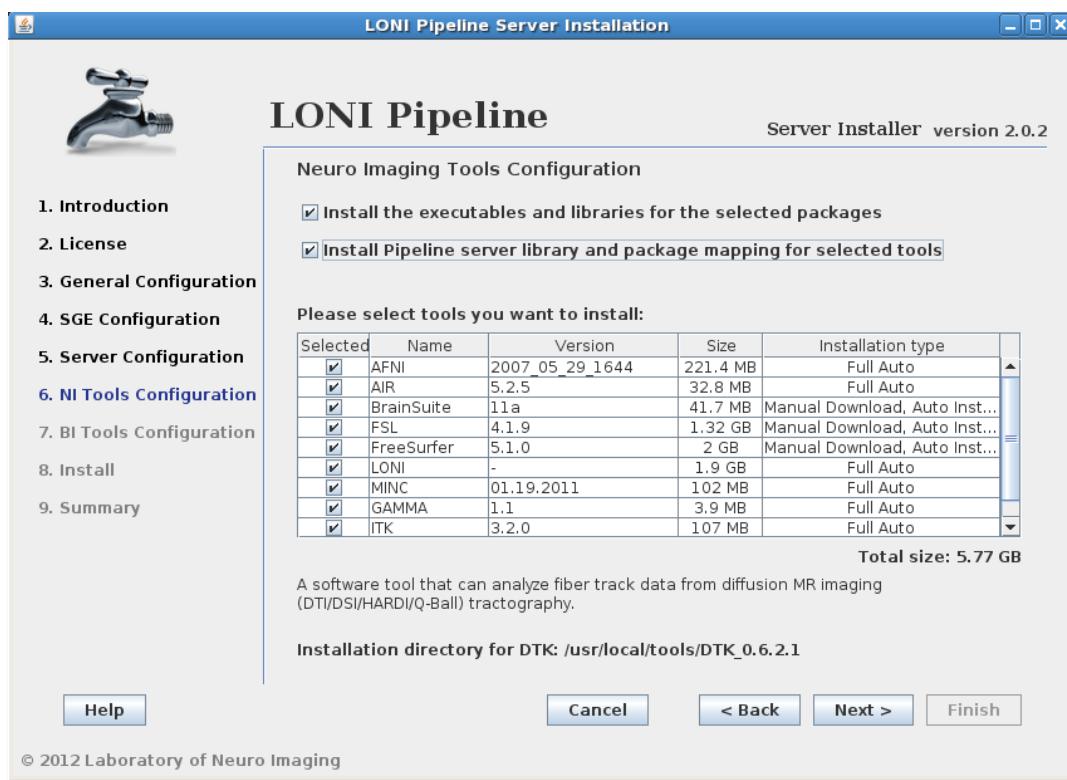
And the last step is to choose the submission queue. The installer will list all the queues you have and you have to pick one for the Pipeline. If you don't have a special queue for Pipeline then you can use the default queue of SGE (SGE's default queue is all.q). If you do not have any queue defined in SGE, you have to create one yourself and run the installer again.



## Install Pipeline without SGE

It is also possible to configure Pipeline to not use any queue. In this case all the jobs will be executed locally on the server and you have to be careful with number of jobs submitted to the server as high number of jobs will affect on server's performance. Please see "Maximum Number of threads for active jobs" if you want to set limits to number of parallel running jobs.

## Install Neuro Imaging Tools

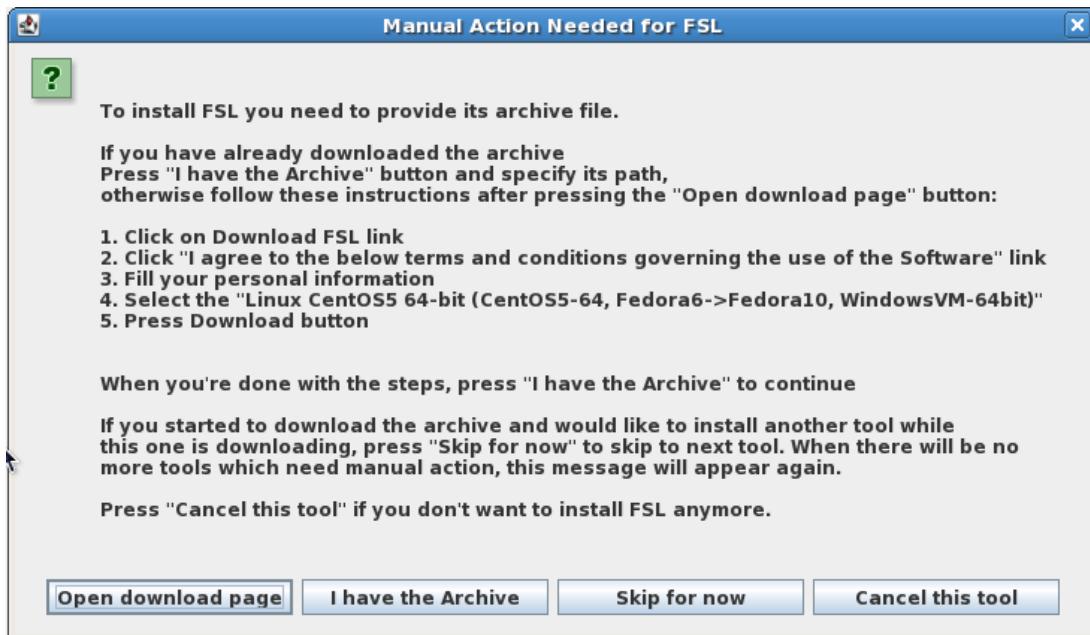


In this section you have can select the imaging and informatics software tools and server library files for these tools.

If the installation type for a tool is "Full Auto", it will be installed automatically. For some tools, it is marked as "Manual Download", this means you will be asked to go to a website to download (e.g. FSL, FreeSurfer). This is because the licensing restriction imposed on the software.

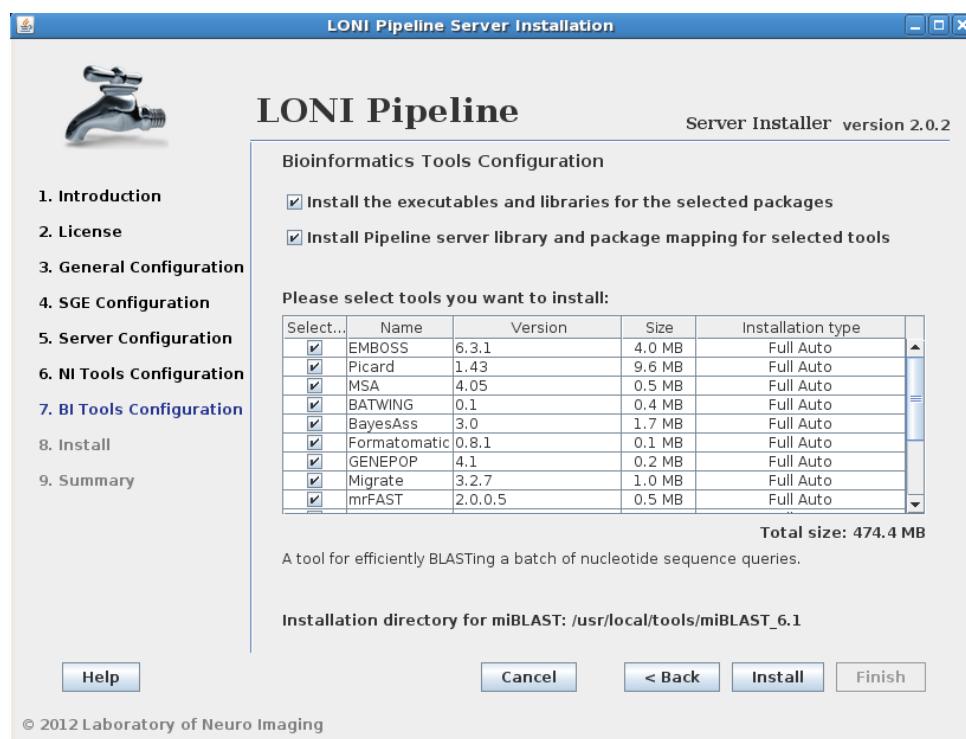
Click on Open download page to open the browser, the page will be loaded for you, follow the instructions on the dialog and download the tool to your computer.

After it is done, click “I have the archive” to locate the file. If you already downloaded the file, you can also click on “I have the archive” to provide the path of the file. Click on Skip for now will go to the next tool. If some other tool needs manual download, it will show instructions for that tool as well, so that you can download all the tools in parallel. All the message dialogs will cycle through if you click “Skip for now”, until your download is complete and you provide the file by clicking “I have the archive.”



## Install Bioinformatics Tools

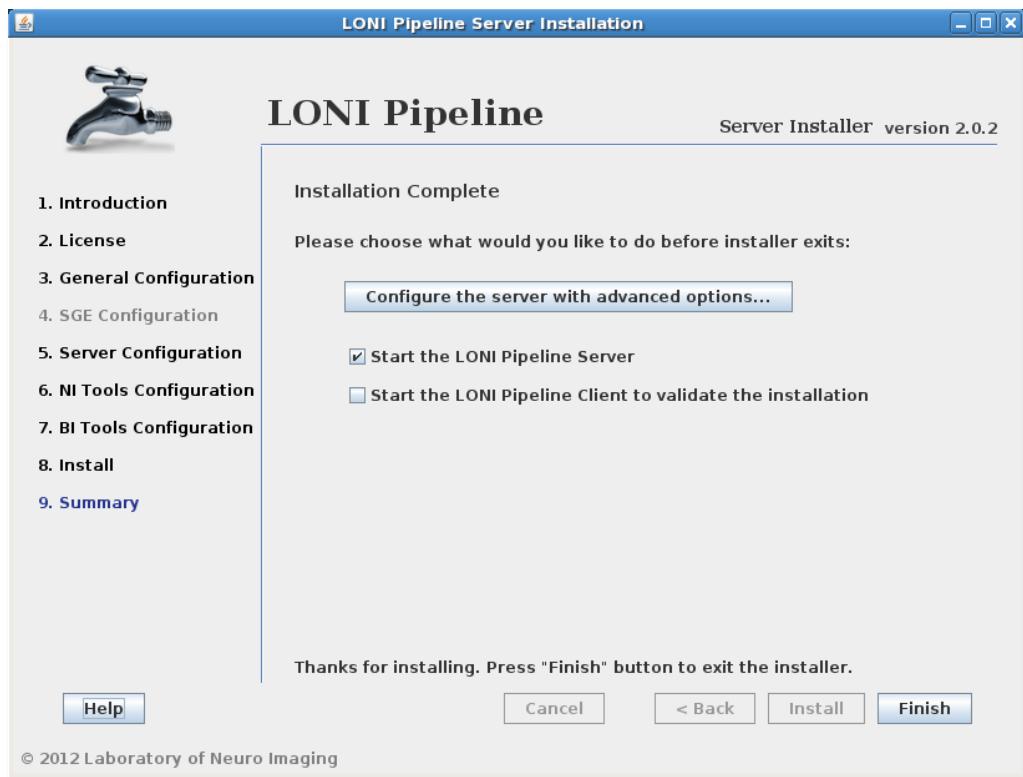
Follow the same instructions given above for neuro imaging tools. In particular, select the packages that you would like to install. Also, you can choose to only install the executables, the Pipeline server library files, or both. After selecting the packages that you need, press the Next button to proceed with the installation. Note that this is the final step before the Pipeline installation utility takes over and starts to download/install files.



## Install the tools without installing Pipeline or SGE

If at a later time, you want to install updated version of some tool, you can have it installed without installing the Pipeline and SGE. Simply check only the Neuro Imaging Tools in the general configuration section of the installer, then click Next, it will skip the Pipeline and SGE installation steps and go directly to the tools installation step.

## Finish Install



After the installation is successfully done, it will show a summary screen. Click Finish with Start Server checked will exit the installer and launch the Pipeline server. You can also check Start client option to launch the client to validate installation.

You can also configure advanced server preferences using the server configuration tool by clicking on “Configure the server with advanced option...”.

If you have any questions, please contact [pipeline@loni.ucla.edu](mailto:pipeline@loni.ucla.edu)

## Start the Server

If you checked “Start the LONI Pipeline Server” option on the summary page of the installation, the Pipeline server process should be started. To check the logs of the Pipeline server, go to the Pipeline server’s directory (`/usr/pipeline` by default), specified in the “Install Pipeline” step. You will find files called `outputStream.log` and `errorStream.log`, which stores output stream and error stream. You can verify if the server started successfully by checking the contents of the `outputStream.log` file, it should look something like this:

```
[ 1/6 ] Connecting to Persistence Database.....DONE [117ms]
[ 2/6 ] Starting server on port 8008.....DONE [1152ms]
[ 3/6 ] Loading server library.....DONE [31ms]
[ 4/6 ] Loading server packages info.....DONE [7ms]
[ 5/6 ] Checking to resume backlogged workflows....DONE [0ms]
[ 6/6 ] Checking to resume active workflows.....DONE [0ms]
[ SUCCESS ] Server started.
```

You can stop and start the Pipeline server by calling (root access required)

```
/etc/init.d/pipeline stop
/etc/init.d/pipeline start
```

If you don’t have root access, you can stop and start the Pipeline server by going to the Pipeline server’s directory and type

```
./killServer.sh
./launchServer.sh
```

Always check if the server has started successfully by viewing the `outputStream.log` file. If it shows error on persistence database, you can stop and start the persistence database process by typing:

```
./db/stopDB.sh
./db/startDB.sh
```

## Troubleshoot

The following is a list of error messages and explanation:

- **The provided directory seems not to be a network file shared (NFS) directory.**

The installer will verify the Shared File System Location given. It is required to have it on NFS if the server is set to use a grid. The shared file system is used for the Pipeline server to store intermediate files of workflows and to install Grid Engine and Neuro Imaging Tools.

- **It is not supported for a Grid Engine installation that the local hostname is “localhost” and/or the IP address is like 127.0.\*.\***

You must provide fully qualified domain names as hostnames, so something like “[host1]”, “[localhost]” or “[127.0.0.1]” is not allowed.

- **Cannot enable Grid submission as SGE doesn’t have any queue.**

If you do not have any queue defined in SGE, you have to create one yourself and recheck “Enable Grid submission” checkbox and select the queue.

## Command Line Installation

Alternative to DPS installer, you can also install the Pipeline server using an automated method that relies on a preference file. All of the fields that would otherwise have to be entered via the GUI can be specified within a hierarchical XML file. First of all, you need to download the same archive that contains the graphical version of the installer (you can get it here). A default configuration file is included in the distribution (in the dist/install\_files directory). You can run the installation in automatic mode by typing the following into your shell:

```
tar -zvxf pipelineServerInstaller.tar.gz  
cd pipelineServerInstaller  
. ./launchInstaller.sh -auto  
dist/install_files/DefaultInstallationPreferencesFile.xml
```

A complete template for the XML file can be found here. If you use this template as a starting point, note that it has a lot of placeholders and is not set up to run “as is”, so you would have to make many modifications. For reference, each of the tags is documented below:

- `DistributedPipelineServerInstaller`: root tag, contains all other tags
- `SharedFileSystemPath`: path to a directory that is shared (via NFS) between the host running the Pipeline server and qmaster, admin, and execution hosts of SGE
- `JDKLocation`: only include this tag if you don't already have Oracle JDK running on the host where you're installing the Pipeline server; the value should be the path to the JDK RPM, which you can install from the Oracle page
- `PipelineServer`: use attribute `enabled="true"` to indicate that you would like to install the Pipeline server; the children of this element will specify information about the server installation
  - `InstallLocation`: specifies location where Pipeline server is to be installed
  - `Hostname`: specifies the hostname of the host where Pipeline server is being installed
  - `Port`: specifies port on which the Pipeline server will be accepting connections from clients
  - `Username`: specifies user that will be running the Pipeline server
  - `TempDir`: specifies a directory where Pipeline modules will write intermediate files
  - `ScratchDir`: specifies a scratch directory where sample workflows will write their outputs; this value then becomes available to users through the pre-defined  `${tempdir}` variable, documented here
  - `GridSubmission`: use attribute `enabled="true"` to indicate that you would like the Pipeline to submit jobs via grid engine to execution hosts; otherwise, the jobs will be run locally on the host running the Pipeline server
    - `GridPlugin`: options are JGDI or DRMAA
    - `GridSubmissionQueue`: the SGE queue where Pipeline should submit its jobs
    - `UsePrivilegeEscalation`: options are true or false; privilege escalation is documented here
  - `DBInstallLocation`: path to a directory where you would like to install the Pipeline database; if it doesn't exist, it will be created by the installer
  - `StartPipelineOnSystemStartup`: set value to true if you would like to configure the system to start the Pipeline server on startup; false, otherwise
  - `AuthenticationModule`: options are SSH, NIS, and NoAuth; these are documented here
  - `ModifySudoers`: use attribute `enabled="true"` to indicate that you want to add the Pipeline user to the sudoers list

- SuperUsers: comma-separated list of users that you don't want the Pipeline server to sudo as (default: root)
- MemoryAllocation: specify the amount of memory you would like to allocate to the Pipeline server/database, in megabytes
- SGE: use attribute enabled="true" to indicate that you would like to install Son of Grid Engine; the tags that follow will describe some of the preferences for the installation; you can find documentation on SGE here
  - SGERoot: path to directory where you would like to install SGE (default: /usr/local/sge)
  - SGECcluster: name of cluster that you would like to install (default: cluster)
  - SubmitHosts: specify hostnames of machines which will be configured to handle job submission and control; you can do this using one hostname per Host element, as children of the SubmitHosts element
  - ExecHosts: specify hostnames of machines which will be execution hosts; use same format as for SubmitHosts
  - AdminHosts: specify hostnames of machines that will be used for SGE administration purposes; use same format as for SubmitHosts
  - AdminUsername: user that will serve as SGE administrator
  - SpoolDir: path to a directory that will be used for spooling during installation
  - Queue: use attribute configure="true" to indicate that you would like to configure a queue at the end of SGE installation; this is documented here
    - Name: the name of the new queue that you would like to configure
    - Hosts: the hosts that you would like to add to the queue
    - Slots: the slots that you would like to add to the queue (the difference between hosts and slots is documented here)
- Tools: use the attribute enabled="true" to indicate that you would like to install some tools; also use the path attribute to specify the directory where you would like to install the tools (note that this should be in an NFS-shared directory)
  - NeuroImagingTools: use the attribute enabled="true" to indicate that you would like to install one or more neuro imaging tools; true/false values for the executables/serverLib attributes indicate whether you need only the executables, the .pipe files, or both for the neuro imaging tools
    - Available neuroimaging tools: AFNI, AIR, BrainSuite, FSL, FreeSurfer, LONI, MINC, ITK, DTK, GAMMA; for each of these, the enabled="true" attribute is used to activate the tool installation; note that BrainSuite, FSL, FreeSurfer, and DTK require that the user specify a sub element, namely ArchivePath, whose value is the path to the archive file,

- downloaded manually from the software website; FreeSurfer additionally requires a LicensePath element with the path to the FreeSurfer license
- BioinformaticsTools: same attributes as NeuroImagingTools tag
  - Available bioinformatics tools: EMBOSS, Picard, MSA, BATWING, BayesAss, Formomatic, GENEPOP, Migrate, GWASS, MrFAST, Bowtie, SamTools, PLINK, MAQ, miBLAST; again, the enabled attribute can be used to indicate activation or deactivation of installation for each of these elements

## Requirements

The Pipeline server can run on any system that is supported by JRE 1.5 or higher, so the first thing to do is head over to the official Java website to download the latest JRE/JDK. If you run the server on Windows, you will not be able to use privilege escalation (you might not even need/want it). Also the Failover feature is only supported by Unix/Linux systems. All other features are available for all platforms.

The amount of memory required varies based on the load you will expect on the server, but for a reference point, the Pipeline server running on cranium.loni.ucla.edu has been set to accept a max load of 620 jobs, and its memory footprint hovers between 50-300MB depending on the load and garbage collection scheme.

## Downloading

Head over to the Pipeline download page (<http://pipeline.loni.ucla.edu/downloads/>) and download the latest version of the program for Linux/Unix. The server and the client are both in the same jar file, so you only need to change the Main entry point when starting up the server. Extract the contents of the download to the location you want to install the server at.

## Starting the Server

Now let's start the server for the first time. Get to a prompt and switch to the directory where you copied the Pipeline.jar and lib directory and type:

```
$ java -classpath Pipeline.jar server.Main
```

Assuming you have java in your path, you should have received the following message back in your terminal window:

```
[ 1/6 ] Connecting to Persistence Database.....DONE [ 61ms ]
[ 2/6 ] Starting server on port 8001.....DONE [ 747ms ]
[ 3/6 ] Loading server library.....DONE [ 336ms ]
[ 4/6 ] Loading server packages info.....DONE [ 2ms ]
[ 5/6 ] Checking to resume backlogged workflows.....DONE [ 46ms ]
[ 6/6 ] Checking to resume active workflows.....DONE [ 0ms ]
[ SUCCESS ] Server started.
```

That's not enough to have a fully functional server yet, but we're a step closer, so go ahead and break out of the process by hitting Ctrl-C.

# Glossary

**AFNI** - (Analysis of Functional NeuroImages) is a Neuroimaging tool that is used for processing and viewing of structural and functional MRI images. There are many tools that are available within this program, like; file conversion tools (AFNI to Analyze, MINC, NIfTI etc), Statistical analysis tools (3D ANOVA, 3D ANOVA2 etc), Surface Tools (Surface clustering, Surface Measures etc) and so on. For more information about the tools and its application please visit the website, <http://afni.nimh.nih.gov/afni>

**AIR** - Automated Image Registration is an automated Image registration tool (3D or 2D images) that is used to register images of different modalities either between or across subjects. There are many AIR programs that can be used to manipulate the images. A list of tools can be found at,

<http://bishopw.loni.ucla.edu/AIR5/programsabc.html>

**AlignLinear** - This is an intramodality registration tool (within or across subjects). Users can specify the models that includes, rigid-body, affine or perspective. For more information on usage and examples follow the link,

<http://bishopw.loni.ucla.edu/AIR5/alignlinear.html>

**Reslice** - This program takes a .air file and uses the information that it contains to load the corresponding image file and generate a new, realigned file. The link below has more information on the usage of this program.

<http://bishopw.loni.ucla.edu/AIR5/reslice.html>

**AVW Maths** - Simple but powerful FSL program to allow mathematical (add, subtract, divide, multiply, square) manipulation of images.

**BET** - FSL's Brain Extraction Tool Deletes non-brain tissue from an image of the whole head. It can also estimate the external skull surface.

**Bias Field Corrector** - Computes local estimates of gain variation in the image using an adaptive partial volume tissue model. Uses the estimates to compute a tri-cubic B-spline. Corrects the extracted brain image using values of the spline.

**BrainSuite** - An Image analysis tool that is used for MR Images. This tool is used for identifying the tissue type and surfaces in the MR images. BrainSuite was specifically designed for surface extraction.

**Brain Surface Extraction** - Removes non-brain tissue from the MRI using a combination of anisotropic diffusion filtering, Marr-Hildreth edge detection, and mathematical morphology.

<http://brainsuite.usc.edu/>

**Cache** - A directory in which the application creates intermediate output files, streams, and log files.

**Command String** - The exact command that was submitted by the Pipeline to the underlying operating system for execution.

**Connection Manager** - The dialog box which holds connections that you have created to Pipeline servers.

**Data Sink** - A special module that takes one or more output values and can be used as the output destination of one or more modules.

**Data Source** - A special module that takes one or more input values and can be used as the input source of one or more modules.

**DRMAA** - Distributor Resource Management Applications API. A library, or specifications, which allow applications to interact, or submit control to jobs on one or more DRM systems.

**Executable** - A file whose contents are meant to be interpreted as a program by a computer.

**FSL** - FMRIB Software Library is a set of tools, which can be used for analyzing the MRI, fMRI and DTI brain imaging data. The various tools that are available through the FMRIB library are categorized based on the type of analyses that needs to be performed (e.g. functional MRI analyses, Structural analyses, DTI analyses and other miscellaneous tools).

<http://www.fmrib.ox.ac.uk/fsl/index.html>

**FSF Update** - A special module used to update the design.fsf setup file. It is a very flexible module that is used to add/update information to the design file (adding experimental variables and/or updating the T1 image, 3D images etc).

**FSL Design.fsf** - This is a setup file that has information about various options and design specified using the FSL FEAT interface.

**FSL FEAT** - FEAT is part of FSL (FMRIB's Software Library). It is a software tool for high quality model-based fMRI data analysis, with an easy-to-use graphical user interface (GUI). FEAT automates as many of the analysis decisions as possible, and allows easy (though still robust, efficient and valid) analysis of simple experiments whilst giving enough flexibility to also allow sophisticated analysis of the most complex experiments. **FreeSurfer** - This is a group of software programs that are used to process the MRI images in two ways, one volumetric analyses and the other surface analyses. The freesurfer pipeline is completely automated which is very helpful for large sets of data.

**FSL Merge** - FSL's miscellaneous utility that is used to concatenate files into a single file. Concatenation can be along time or X, Y or Z-axis. All image dimensions (except for the one being concatenated over) must be the same in all input images. For example, this can be used to take multiple 3D files (eg as output by SPM) and create a single 4D image file.

**FSLSwapDim** - This program re-orders the data storage to permit changes between axial, sagittal and coronal slicing. When used in this mode no warning should be printed and the output files will maintain the same left-right order.

**Image Data Archive (IDA)** - An acronym that stands for Image Database Archive. Along with being one of the protocol which the LONI Pipeline support, IDA offers the following benefits:

- **De-identification** - Addresses government regulations for protection of human subject privacy
- **Data Transmission** - Data is transmitted over the internet using Hyper-Text Transfer Protocol with SSL encryption (HTTPS)
- **Storage** - Data is archived on a fault-tolerant storage area network (SAN), providing near 24/7 availability
- **Execution Dialog** - A dialog which shows important messages printed by the different times during its execution.

**ITK** - Insight Segmentation and registration toolkit is a set of programs that employs leading edge algorithms for segmentation and registration of images. There are 3 main programs that are used in the ITK toolkit, 3D Deformable Model Segmentation, 3D Deformable Registration, 3D Multi-modality B-spine Deformable Registration

**LONI** - Laboratory of Neuro Imaging offers a set of tools that are used for structural, functional and DT Image data manipulation. This tool kit is available through the LONI Pipeline library.

**LONI Viewer** - An Image Viewer that allows viewing of shape, surface, volume and geometric image data.

**MiND** - Metadata in NIfTI for DWI is a set of extensions to NIfTI header, which serve as a mechanism for preserving the metadata in direct association with the Diffusion Weighted datasets.

**Module** - The smallest unit of a Pipeline workflow. Specifically, it is a chunk of XML that describes an executable and its inputs and outputs. It can be created by a user and placed directly into a workflow, or a user can drag and drop predefined modules from the library of any server to which they are connected.

**Module Definition** - The collection of information including the executable author, name, package, version, description and parameter names that must be specified for each executable to create a module that can be used in the Pipeline. Once a module definition has been created for an executable, it can be saved in the library and reused by other users.

**Module Group** - A collection of modules. The Pipeline can abstract a Module Group to be represented as a single module in a workflow.

**MINC** - MINC toolkit was developed at the Brain Imaging Centre at Montreal. They are a set of applications and tools that are used for medical image processing and data management. There are five categories of software programs available, Visualization Tools, Basic Tools, Pipelining Tools and Statistical Analysis Tools.

**Output Log** - A collection of messages which are printed out to the output stream of the application.

**Package** - A suite of module definitions which are interrelated.

**Parameter** - An input or output to a module.

**Partial Volume Classifier** - Each voxel is classified according to the tissue type (WM, GM, CSF, and partial volume mixture) by combining the partial volume tissue model with a Gibbs spatial prior to produce a classifier that encourages contiguous region of similar tissue type.

**Personal Library** - A place to store or save workflows and modules for easy access through the Pipeline.

**Play Button** - This is used to validate and execute the workflows by connecting to the computer cluster.

**Pipeline** - An environment to develop workflows for data processing, independent of data location, program location, and platform.

**Provenance** - LONI Pipeline includes a new feature that enables tracking data, workflow and execution history of all processes. This functionality improves the communication, reproducibility and validation of newly proposed experimental designs, scientific analysis protocols and research findings.

**ShapeTools** - Shape Tool programs are a set of portable tools that support the modeling, analysis and display of interesting geometric shapes. Nearly all Shape Tools applications are written in the Java programming language.

**Smartlines** - A connection between two modules, which will automatically convert the output file type of the first module to be compatible with the input file type of the second module.

**Validation** - Occurs automatically when the user requests that a workflow be executed. Validation entails:

- Verifying the existence of inputs, outputs and executables
- Cycle detection
- File cardinality checks

- File type checking

Depending on workflow content, connection status and other variables, the validation may be more or less complex. Please refer to Pipeline documentation on validation for more information.

**Workflow** - A set of connected modules that performs analysis or simply processes input data.



# Bibliography

- Woods, R. P., Grafton, S. T., Holmes, C. J., Cherry, S. R. and Mazziotta, J. C. (1998). "Automated image registration: I. General methods and intrasubject, intramodality validation." *J Comput Assist Tomogr* 22(1): 139-52.
- Woods, R. P., Grafton, S. T., Watson, J. D., Sicotte, N. L. and Mazziotta, J. C. (1998). "Automated image registration: II. Intersubject validation of linear and nonlinear models." *J Comput Assist Tomogr* 22(1): 153-65.
- Smith, S. M., Jenkinson, M., Woolrich, M. W., Beckmann, C. F., Behrens, T. E., Johansen-Berg, H., Bannister, P. R., De Luca, M., Drobnjak, I., Flitney, D. E., Niazy, R. K., Saunders, J., Vickers, J., Zhang, Y., De Stefano, N., Brady, J. M. and Matthews, P. M. (2004). "Advances in functional and structural MR image analysis and implementation as FSL." *Neuroimage* 23 Suppl 1: S208-19.
- MacKenzie-Graham, A., Payan, A., Dinov, I., Van Horn, J. D. and Toga, A. W. (2008). Neuroimaging Data Provenance Using the LONI Pipeline Workflow Environment. Provenance and Annotation of Data International Provenance and Annotation Workshop, IPAW 2008. , Salt Lake City, UT, University of Utah.
- Mackenzie-Graham, A. J., Van Horn, J. D., Woods, R. P., Crawford, K. L. and Toga, A. W. (2008). "Provenance in neuroimaging." *Neuroimage* 42(1): 178-95.
- Cox, R. W. (1996). "AFNI: software for analysis and visualization of functional magnetic resonance neuroimages." *Comput Biomed Res* 29(3): 162-73.
- Saad, Z. S., Chen, G., Reynolds, R. C., Christidis, P. P., Hammett, K. R., Bellgowan, P. S. and Cox, R. W. (2006). "Functional imaging analysis contest (FIAC) analysis according to AFNI and SUMA." *Hum Brain Mapp* 27(5): 417-24.
- Rex, D. E., Ma, J. Q. and Toga, A. W. (2003). "The LONI Pipeline Processing Environment." *Neuroimage* 19(3): 1033-48.

Toga, A., Rex, D. E. and Ma, J. (2001). A graphical interoperable processing pipeline. Organization for Human Brain Mapping Annual Meeting, Brighton, England, Academic Press.

Shattuck, D. W., Mirza, M., Adisetiyo, V., Hojatkashani, C., Salamon, G., Narr, K. L., Poldrack, R. A., Bilder, R. M. and Toga, A. W. (2008). "Construction of a 3D probabilistic atlas of human cortical structures." Neuroimage 39(3): 1064-80.

Mazziotta, J., Toga, A. W., Evans, A., Fox, P., Lancaster, J., Zilles, K., Woods, R. P., Paus, T., Simpson, G., Pike, B., Holmes, C., Collins, L., Thompson, P. M., MacDonald, D., Iacoboni, M., Schormann, T., Amunts, K., Palomero-Gallagher, N., Geyer, S., Parsons, L. M., Narr, K., Kabani, N., LeGoualher, G., Boomsma, D., Cannon, T., Kawashima, R. and Mazoyer, B. (2001). "A probabilistic atlas and reference system for the human brain: International Consortium for Brain Mapping (ICBM)." Philos Trans R Soc Lond B Biol Sci 356: 1293-1322.

Mazziotta, J., Toga, A., Evans, A., Fox, P., Lancaster, J., Zilles, K., Woods, R., Paus, T., Simpson, G., Pike, B., Holmes, C., Collins, L., Thompson, P., MacDonald, D., Iacoboni, M., Schormann, T., Amunts, K., Palomero-Gallagher, N., Geyer, S., Parsons, L., Narr, K., Kabani, N., Le Goualher, G., Feidler, J., Smith, K., Boomsma, D., Pol, H. H., Cannon, T., Kawashima, R. and Mazoyer, B. (2001). "A four-dimensional probabilistic atlas of the human brain." J Am Med Inform Assoc 8(5): 401-30.

Dale, A. M., Fischl, B. and Sereno, M. I. (1999). "Cortical surface-based analysis. I. Segmentation and surface reconstruction." Neuroimage 9(2): 179-94.

Fischl, B., Sereno, M. I. and Dale, A. M. (1999). "Cortical surface-based analysis. II: Inflation, flattening, and a surface-based coordinate system." Neuroimage 9(2): 195-207.

Tu, Z., Narr, K. L., Dinov, I., Dollar, P., Thompson, P. M., & Toga, A. W. (2008). Brain Anatomical Structure Segmentation by Hybrid Discriminative/Generative Models. IEEE Transactions on Medical

Imaging. Volume 27, Issue 4, April 2008 Page(s): 495-508, Digital Object Identifier 10.1109/TMI.2007.908121.

Shi, Y.; Thompson, P.M.; Dinov, I.; Toga, A.W. (2008) Hamilton-Jacobi Skeleton on Cortical Surfaces. IEEE Transactions on Medical Imaging. Volume 27, Issue 5, May 2008 Page(s):664 - 673. Digital Object Identifier 10.1109/TMI.2007.913279

Yonggang Shi, Paul M. Thompson, Greig I. de Zubicaray, Stephen E. Rose, Zhuowen Tu, Ivo Dinov and Arthur W. Toga, (2007). Direct mapping of hippocampal surfaces with intrinsic shape context, NeuroImage, 37(3), September 2007, Pages 792-807.

Buckner RL, Snyder AZ, Sanders AL, Raichle ME, Morris JC (2000) "Functional Brain Imaging of Young, Nondemented, and Demented Older Adults", Journal of Cognitive Neuroscience, 12 Supplement 2, pp. 24-34.

Shattuck DW, Sandor-Leahy SR, Schaper KA, Rottenberg DA, and Leahy RM (2001) Magnetic Resonance Image Tissue Classification Using a Partial Volume Model," [NeuroImage 13\(5\):856-876.](#)