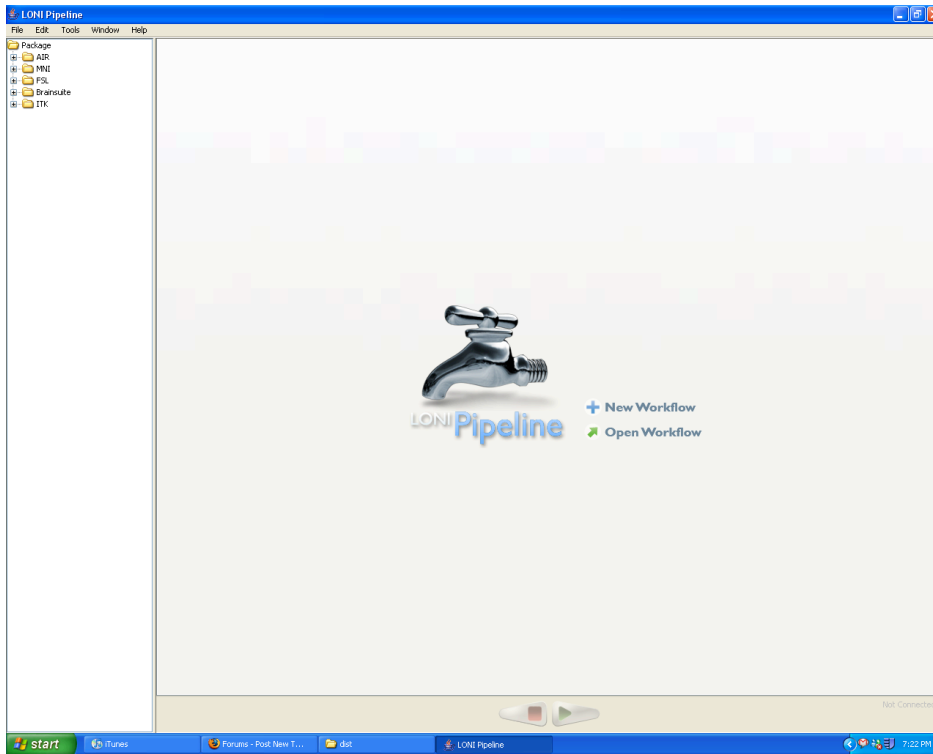
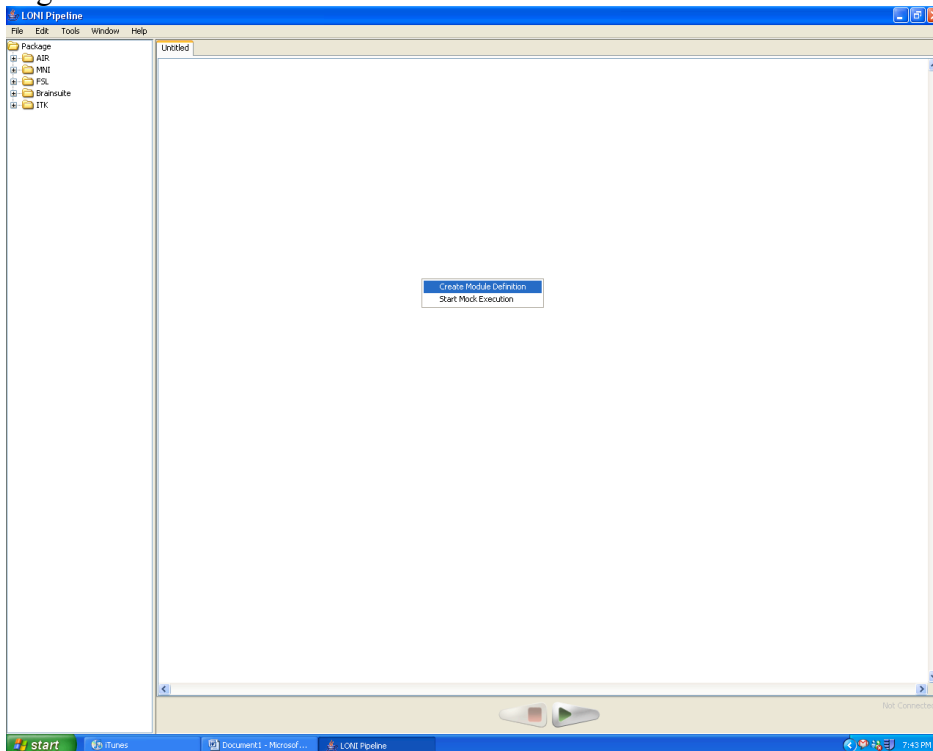


Creating New Module Definition: Launch Pipeline

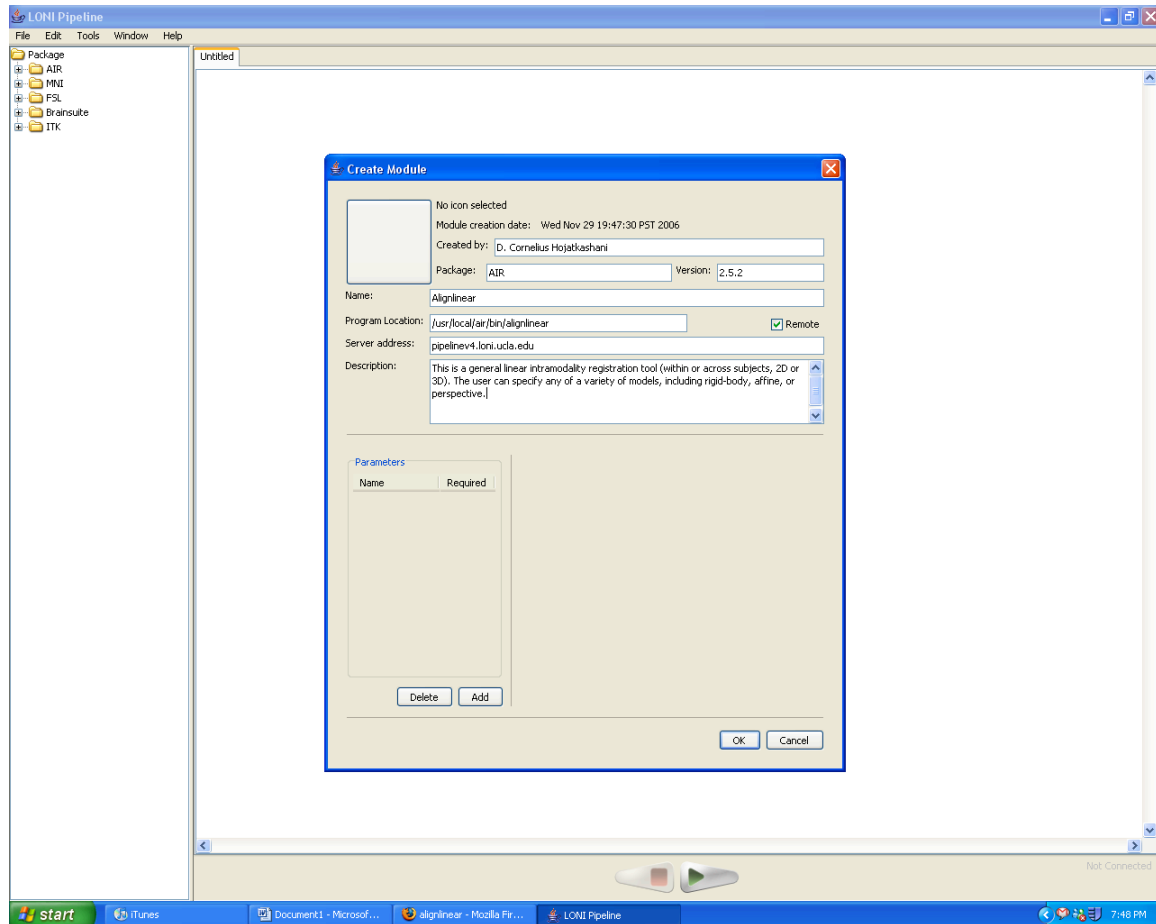
Click on New Workflow



Right click and choose Create Module Definition



Created by: The author of the definition/executable
Package: The suite to which the executable belongs
Program Location: The location where the executable can be found
Server Address: The IP address of the Pipeline server
Description: The text which will appear when the mouse hovers over the module



Note: Order of parameters being added must be dictated by the expectations of the executable.

As example, we cover AIR's alignlinear. Usage: alignlinear [standard-file](#) [reslice-file](#) [air-out](#) -m [model-menu-number](#) [options]

Click Add under Parameters

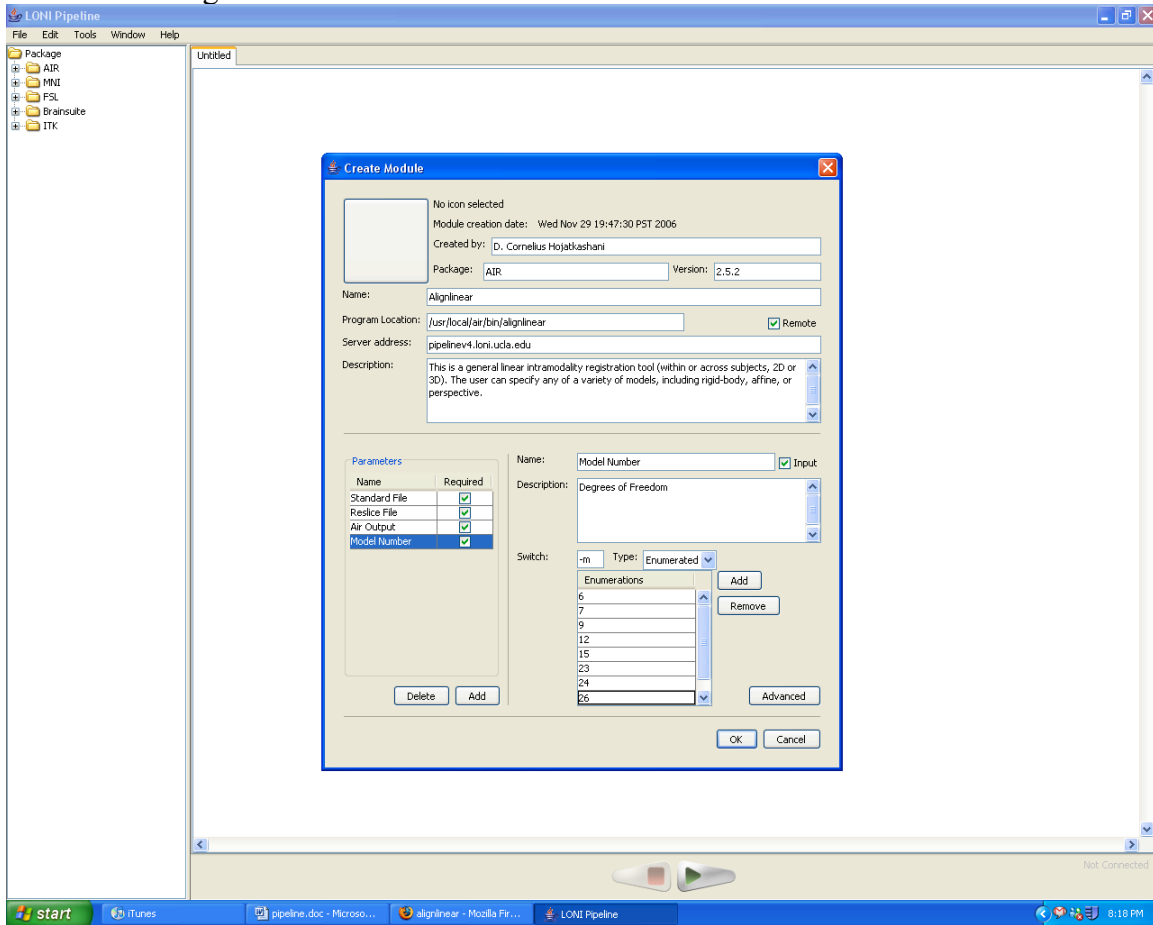
Change 'New parameter' to 'Standard File' and activate the 'Required' box

Change 'File Type' from 'File' to 'Analyze Image 3D'

Click Add under Parameters

Repeat for the reslice-file

Click Add under Parameters
 Change Name to Air Output
 Uncheck the 'Input' switch next to the Name: field as this is the output
 Change File Type to AIR file
 Click Add under Parameter
 Change the name to Model Number
 In the Switch File, type '-m' as illustrated above in 'Usage'
 Change 'Type' to 'Enumerated' and click 'Add' next to 'Enumerations'
 Continue adding until all valid values have been entered



Repeat process for all other options.

Additional Features:

Dependencies:

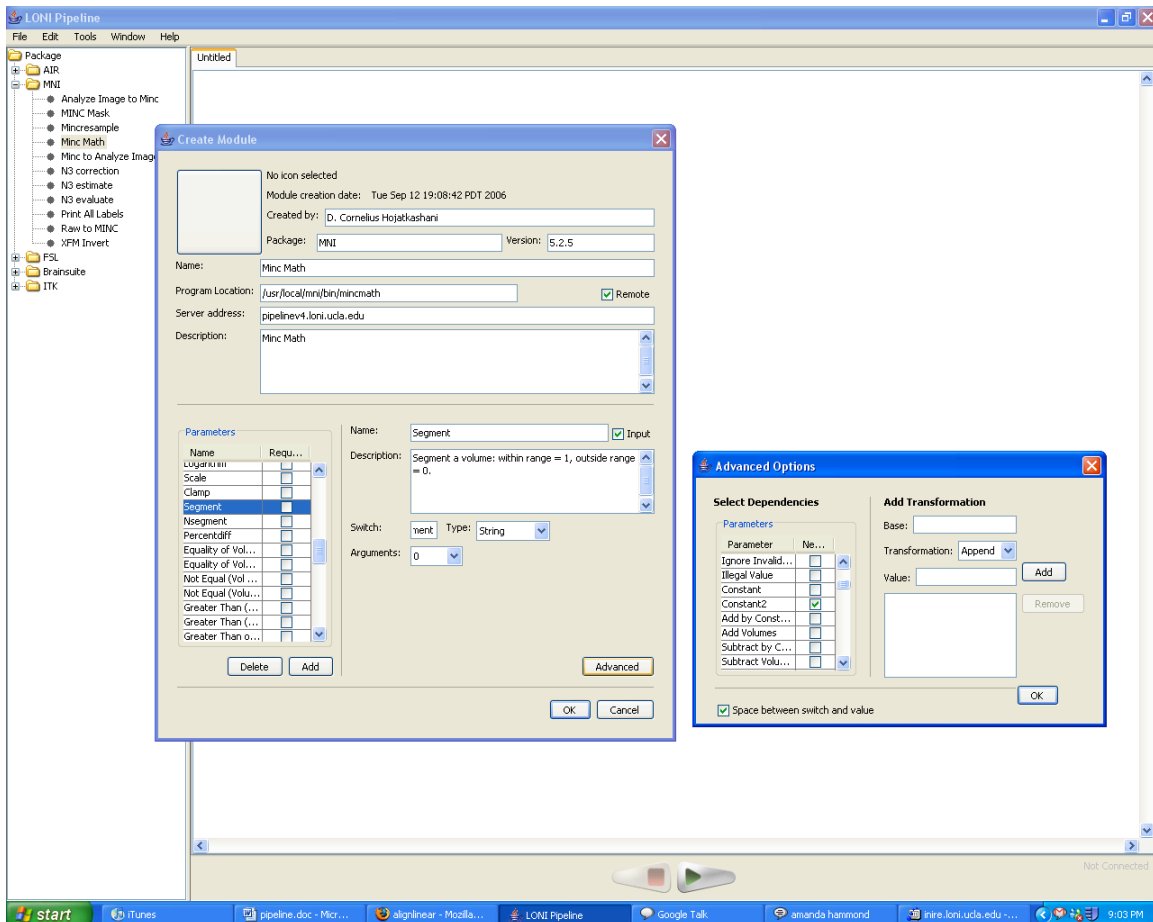
Example: Mincmath, Usage: mincmath [options] [<in1.mnc> ...] <out.mnc>

Focus: '-segment' option: Segment a volume using range of -const2: within range = 1, outside range = 0, i.e, when -segment is activated, so much -const2 so as to specify values ranges to be extracted

Create new parameters for both `-segment` and `-const2`, where `-segment` is of type 'String' with 'Argument' set to '0' and `-const2` is of type 'Number' with 'Argument' set to '2'

Once done with `-segment`, click 'Advanced'

Under 'Select Dependencies' activate the 'Necessary' switch belonging to `-const2` parameter



Dependencies:

Example: FSL's BET, Usage: `bet <input fileroot> <output fileroot> [options]`

Focus: bet expects the input file's root only, meaning, no file extensions, as when exercising other options, strings are attached to the root, eg, if `-m` is activated `_mask.img` is appended to the rootname.

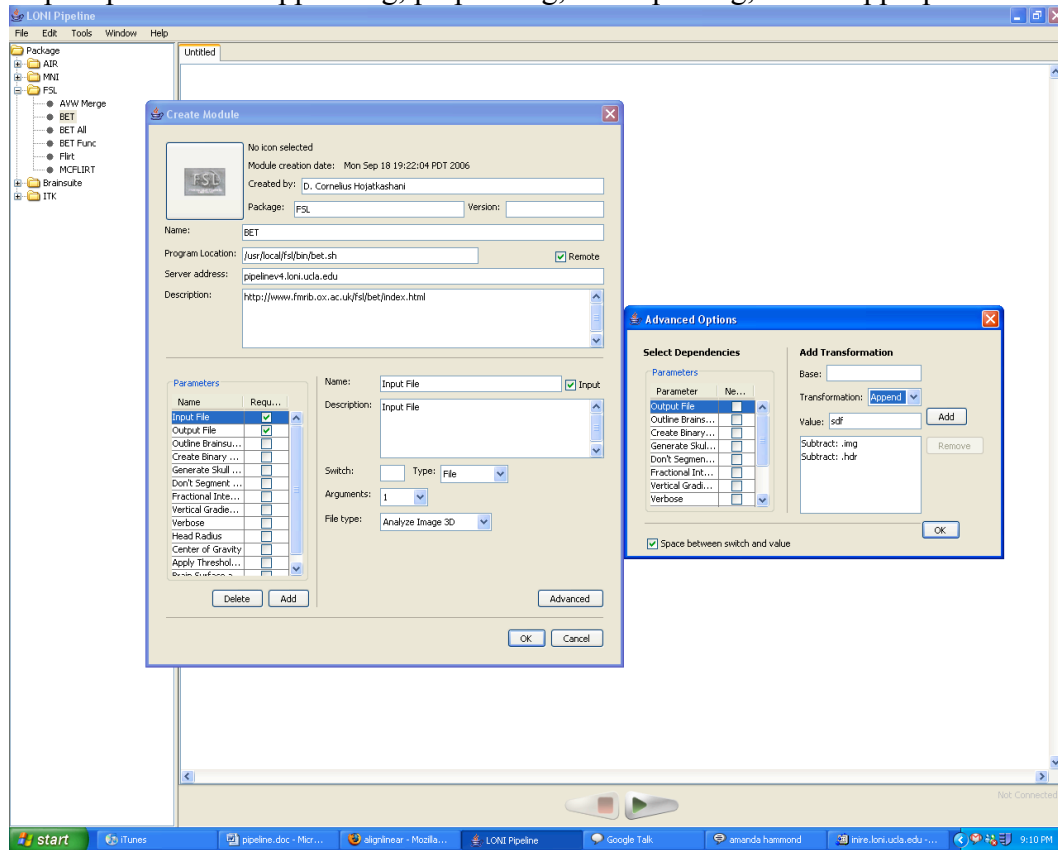
Click Add under Parameters

Create a description for the input analyze image file

Click advanced

Under Transformation, choose Subtract

In the Value box, type .img and select 'Add'
 Repeat for .hdr
 Repeat process for Appending, prepending, and replacing, where appropriate



Parameter Type:

Directory: Choose this type only when new directories will be created by the executables. If a directory name is need for a program just as a place holder, ie, set_working_dir, this parameter is of type string, not directory.

Enumerated: Allows for enumerated types, ie, valid values to be used in the context of an executable, whether strings or integers (eg, 1 2 3 4, A B C, or trilinear nearest_neighbor)

File: The most generic of types, but can be further categorized by choosing file type, currently limited to the following, but easily expandable (NOTE: choosing file types allows the pipeline to establish connections between complementary parameters, and appends appropriate extension to intermediate files being created between nodes, which some programs rely on):

- File: Most generic
- Analyze Header: .hdr file
- Analyze Image: .img file
- Analyze Image 3D: .img file
- Analyze Image 4D: .img file (typically fMR files)

- Brainsuite2 Contour File: ucf file
- Brainsutie2 Surface File: .dfs 3D surface file
- IMP File: .imp file
- MNC File: .mnc image file
- NII File: .nii nifti imge file
- TIFF File: .tif image file
- Vector File:
- Warp File: .warp AIR file
- XFM File: .xfm MINC transformation file

Number: Integers and floats, all number types required by parameters

String: All characters required by parameters