



NA-MIC

National Alliance for Medical Image Computing

<http://www.na-mic.org>

Hello CLI: contributing an algorithm into Slicer 4

Nadya Shusharina, Greg Sharp
Massachusetts General Hospital
nshusharina@partners.org

NA-MIC Tutorial Contest: Summer 2013



Learning Objective

This step by step tutorial leads you through developing command line interface (CLI) for Slicer 4 (<http://www.slicer.org>)

- Getting ready
- Building a template module
- Building module for image thresholding



Pre-requisite

- Slicer is an open-source software for segmentation, registration and visualization of medical imaging data
- The platform is developed through a multi-institution effort of several NIH funded large-scale consortia
- Slicer is for medical research only, and is not FDA approved
- For the general information and “How to” tutorials please visit <http://www.slicer.org/slicerWiki/index.php/Documentation/Nightly/Training>



Material

- We recommend to build Slicer 4 from source
- Refer to the following page:
http://www.slicer.org/slicerWiki/index.php/Documentation/Nightly/Developers/Build_Instructions
- Read prerequisites and platform specific instructions and install all required tools
- Checkout Slicer source
- Configure and build
- Become Slicer community member:
<http://www.slicer.org/slicerWiki/index.php/Documentation/Nightly/Developers/StartHere>

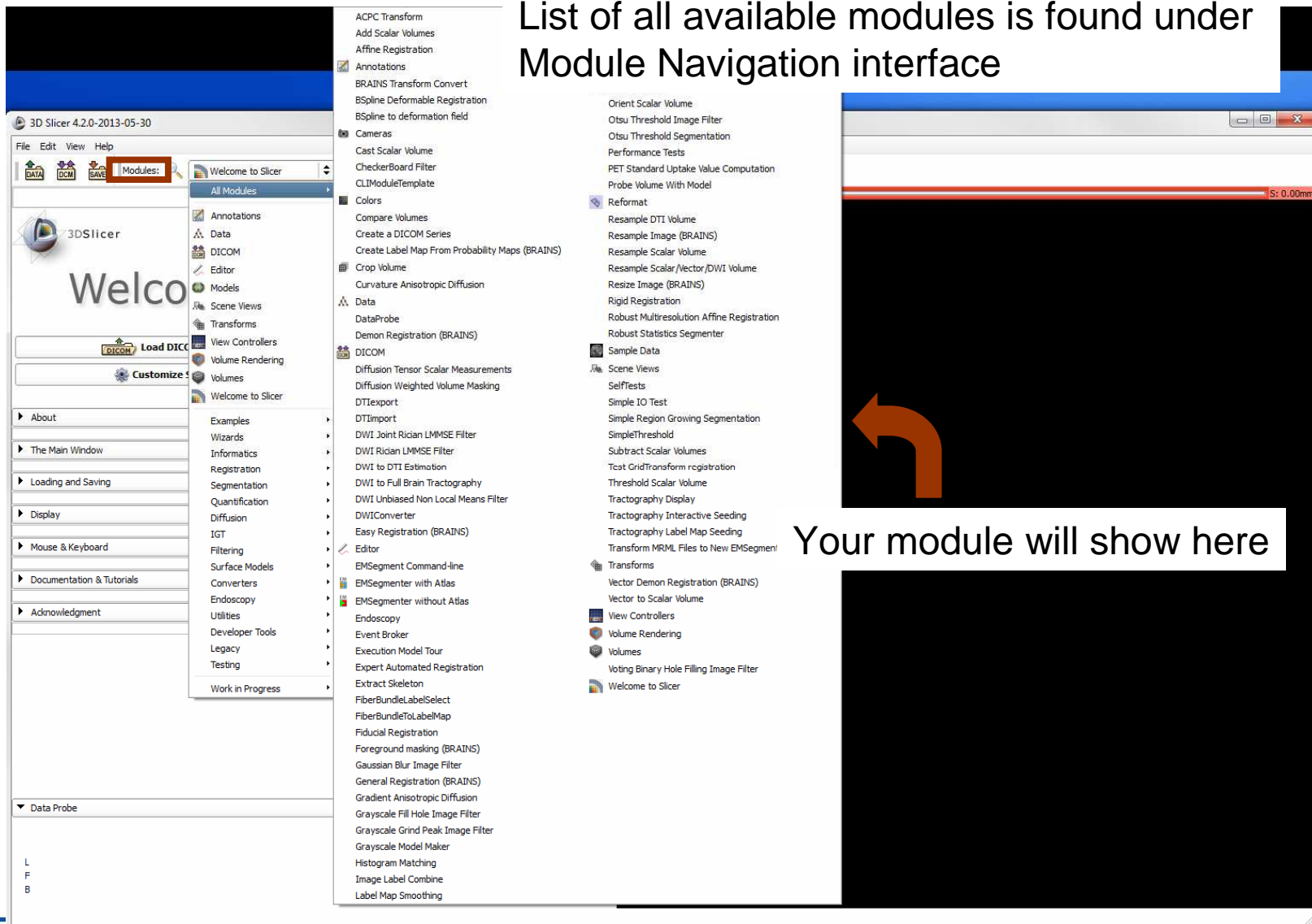


Platforms

- [Add the results of the tests on Linux, Mac and Windows]

Run Slicer: *path-to-SlicerSuperbuild/Slicer-build/Slicer.exe*

List of all available modules is found under Module Navigation interface



ACPC Transform
Add Scalar Volumes
Affine Registration
Annotations
BRAINS Transform Convert
BSpline Deformable Registration
BSpline to deformation field
Cameras
Cast Scalar Volume
CheckerBoard Filter
CLI/ModuleTemplate
Colors
Compare Volumes
Create a DICOM Series
Create Label Map From Probability Maps (BRAINS)
Crop Volume
Curvature Anisotropic Diffusion
Data
DataProbe
Demon Registration (BRAINS)
DICOM
Diffusion Tensor Scalar Measurements
Diffusion Weighted Volume Masking
DTIexport
DWI Joint Rician LMMSE Filter
DWI Rician LMMSE Filter
DWI to DTI Estimation
DWI to Full Brain Tractography
DWI Unbiased Non Local Means Filter
DWIConverter
Easy Registration (BRAINS)
Editor
EMSegment Command-line
EMSegmenter with Atlas
EMSegmenter without Atlas
Endoscopy
Event Broker
Execution Model Tour
Expert Automated Registration
Extract Skeleton
FiberBundleLabelSelect
FiberBundleToLabelMap
Fiducial Registration
Foreground masking (BRAINS)
Gaussian Blur Image Filter
General Registration (BRAINS)
Gradient Anisotropic Diffusion
Grayscale Fill Hole Image Filter
Grayscale Grind Peak Image Filter
Grayscale Model Maker
Histogram Matching
Image Label Combine
Label Map Smoothing
Orient Scalar Volume
Otsu Threshold Image Filter
Otsu Threshold Segmentation
Performance Tests
PET Standard Uptake Value Computation
Probe Volume With Model
Reformat
Resample DTI Volume
Resample Image (BRAINS)
Resample Scalar Volume
Resample Scalar/Vector/DWI Volume
Resize Image (BRAINS)
Rigid Registration
Robust Multiresolution Affine Registration
Robust Statistics Segmenter
Sample Data
Scene Views
SelfTests
Simple IO Test
Simple Region Growing Segmentation
SimpleThreshold
Subtract Scalar Volumes
Test GridTransform registration
Threshold Scalar Volume
Tractography Display
Tractography Interactive Seeding
Tractography Label Map Seeding
Transform MRML Files to New EMSegmenter
Transforms
Vector Demon Registration (BRAINS)
Vector to Scalar Volume
View Controllers
Volume Rendering
Volumes
Voting Binary Hole Filling Image Filter
Welcome to Slicer

Your module will show here



CLI module

- Standalone executables, shared libraries or scripts
- Introduced via plugin mechanism
- XML description produces UI
- Command line parsing code
- Link: http://www.na-mic.org/Wiki/index.php/File:Slicer4_CLI.ppt



Creating module: Step 1

- Make sure that any version of Python is installed on your computer
- From Slicer source directory run the command:
`./Utilities/Scripts/ModuleWizard.py --template
./Extensions/Testing/CLIExtensionTemplate --target
../My_Module My_Module`
- This command created a new directory “My_Module” parallel to Slicer source directory



Build extension: Windows

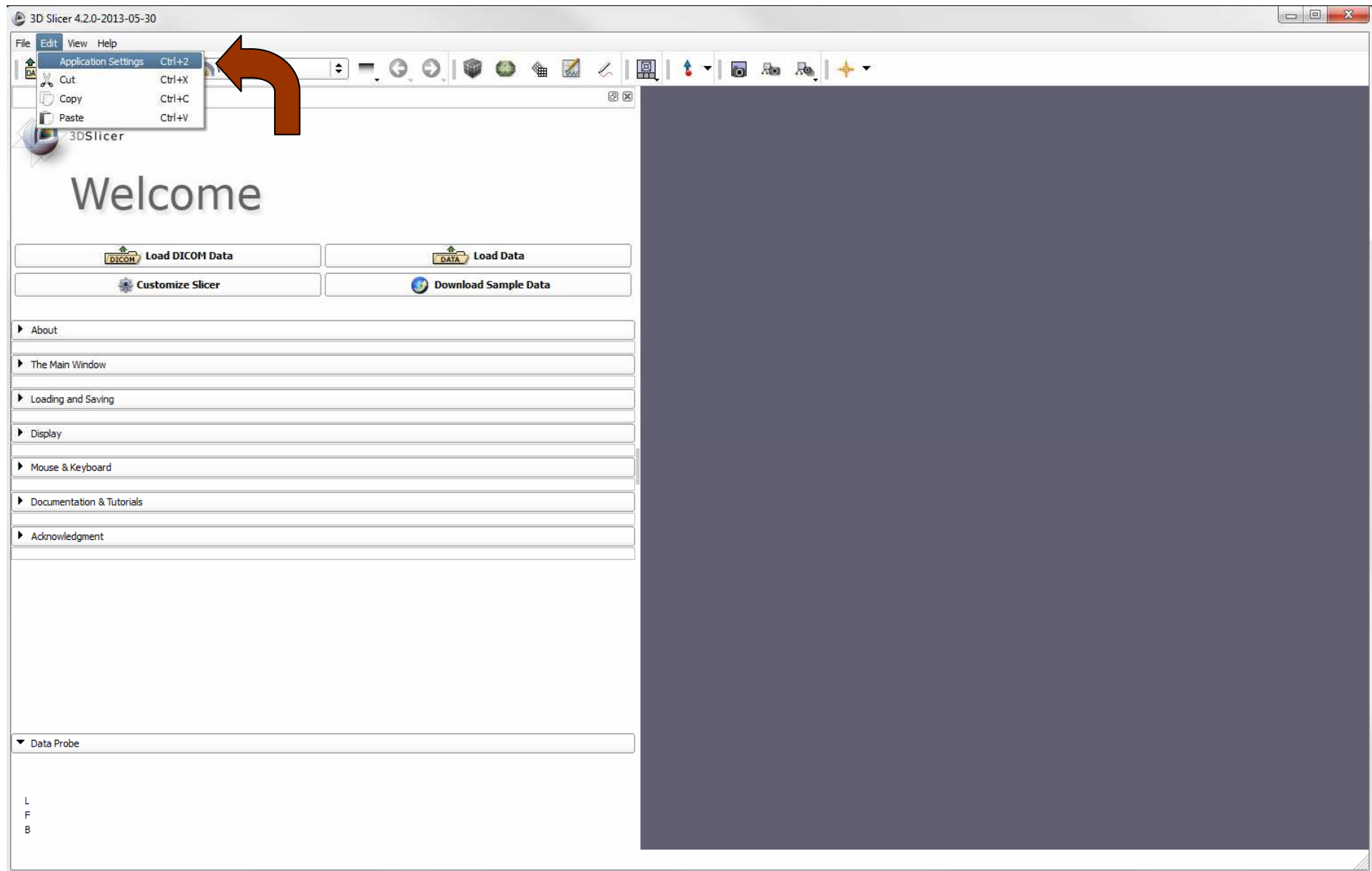
- Run cmake with *../My_Module* as a source directory and *path-to-SlicerSuperbuild/Slicerbuild/Modules/CLI/My_Module* as a build directory
- Use default settings
- Compile using VC

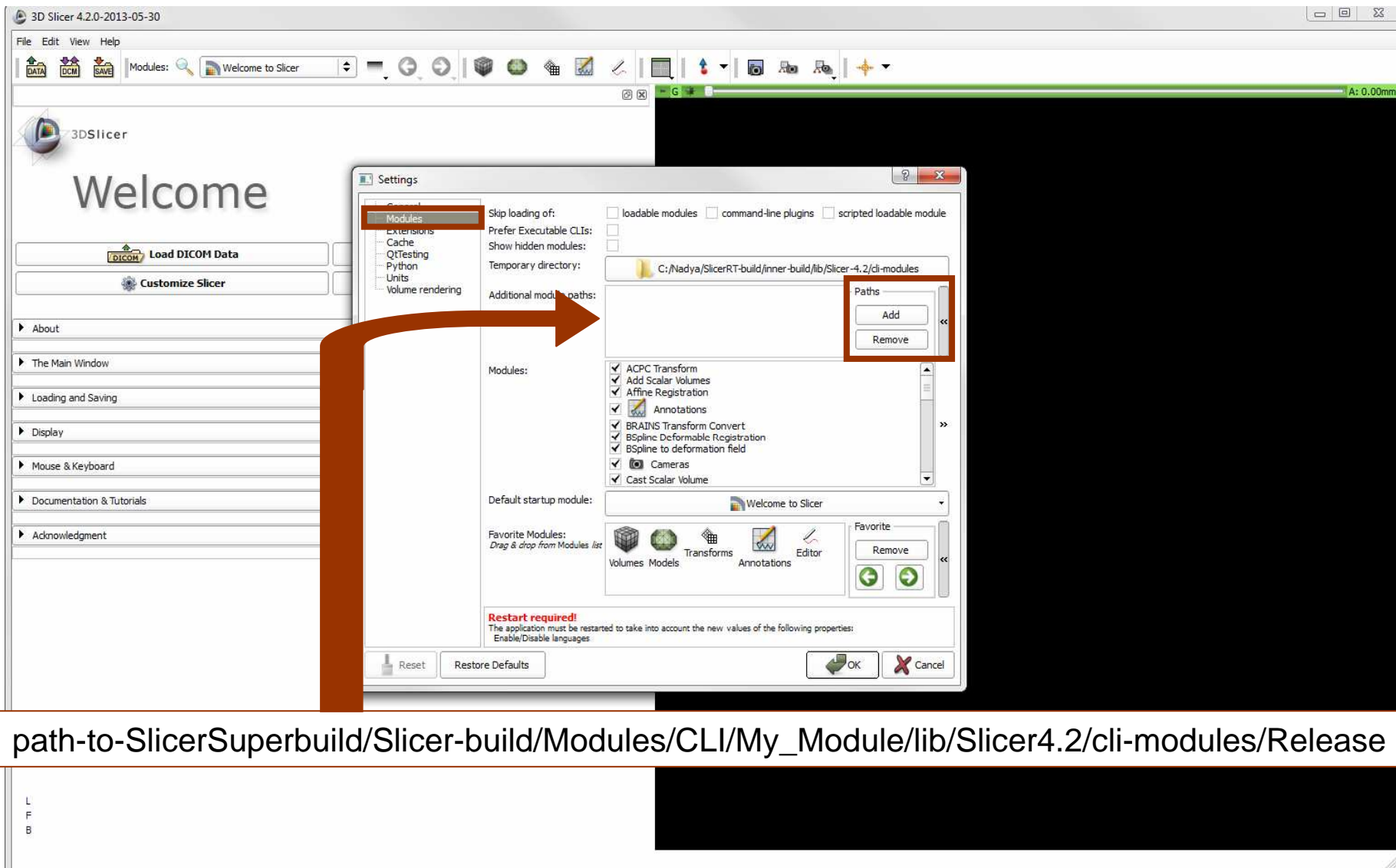


Build extension: Linux

- `$ mkdir MyModule-build`
- `$ cd MyModule-build`
- `$ cmake -DSlicer_DIR:PATH=/path-to-Slicer-Superbuild/Slicer-build ../MyModule`
- `$ make`

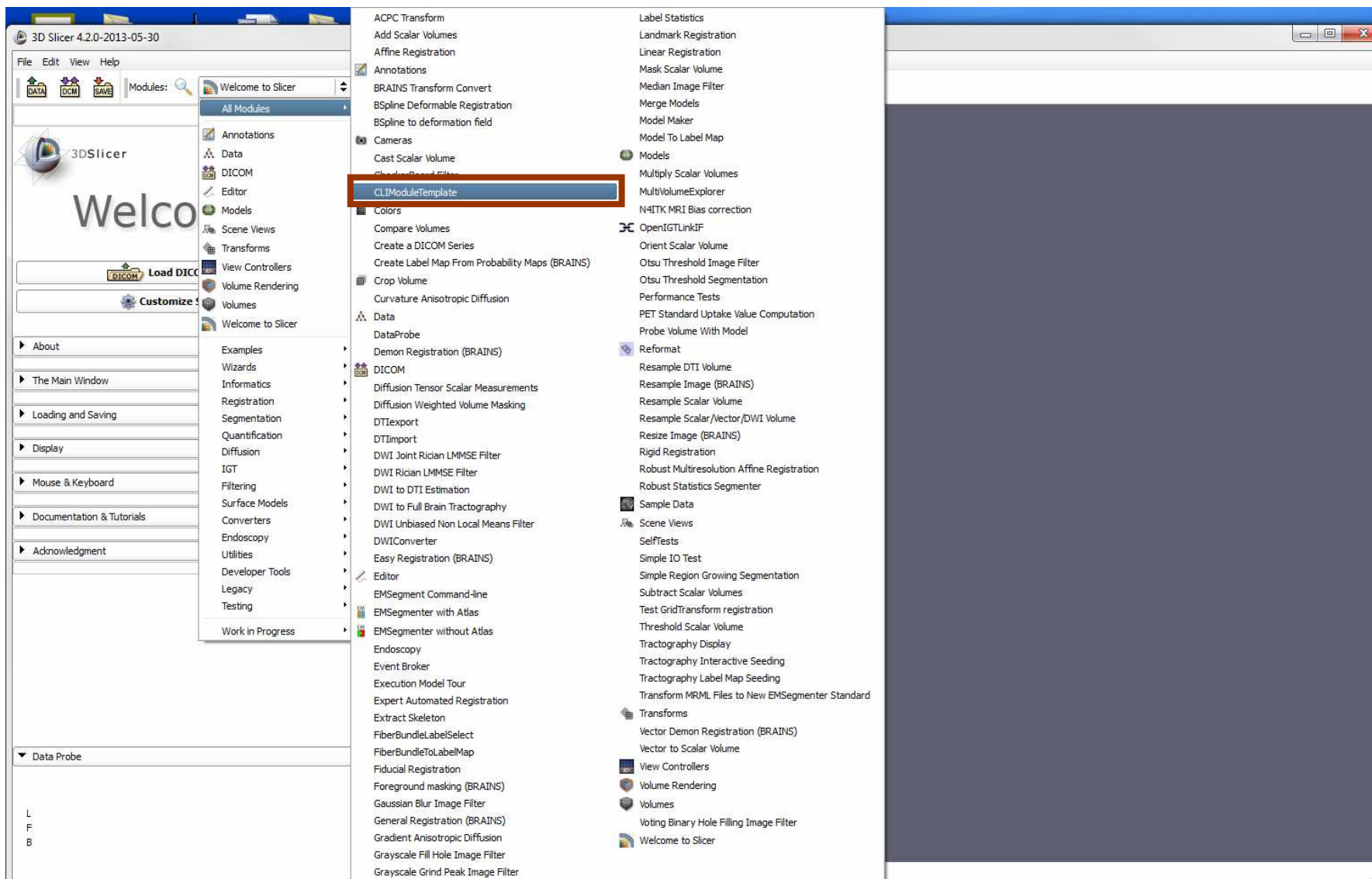
Set the path to My_Module in the Application Settings



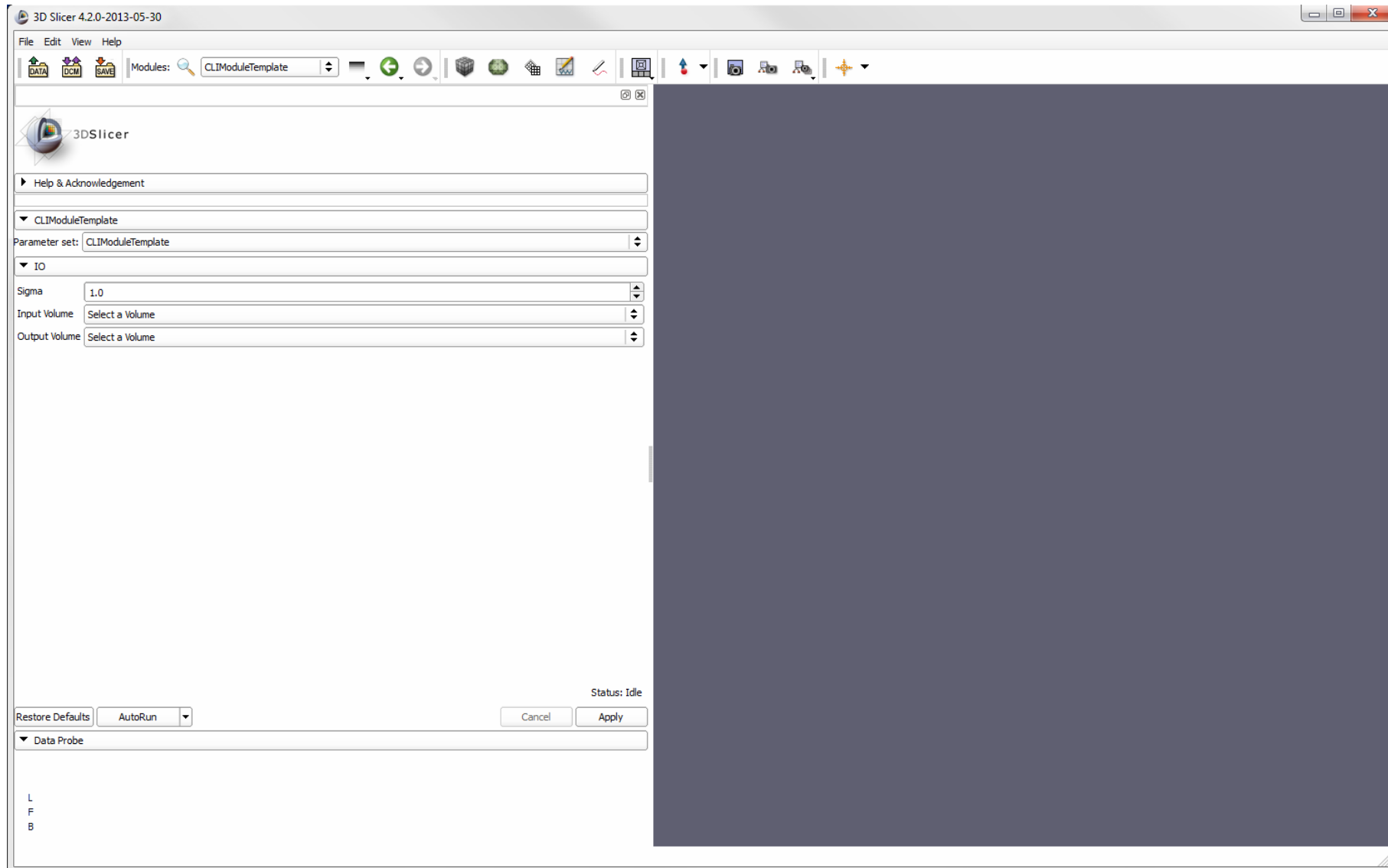


Restart Slicer!

Find the module CLIModuleTemplate in the Module Navigation interface



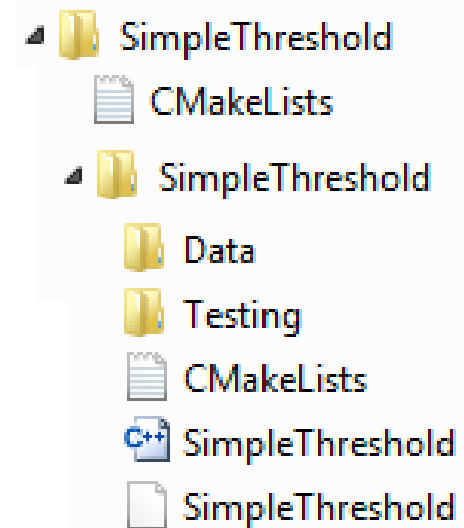
Open the module. Congratulations!





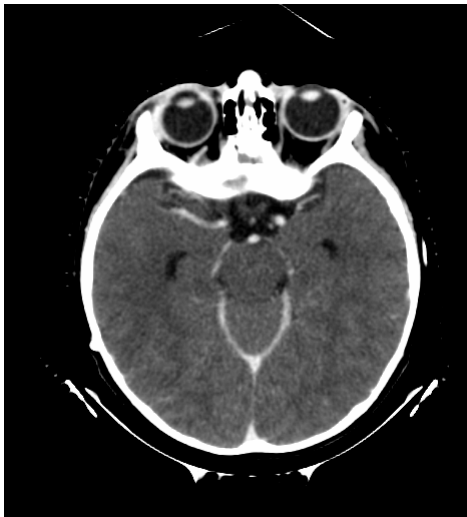
Creating module: Step 2

- Download sample data:
https://forge.abcd.harvard.edu/gf/download/frsrelease/85/2851/hello_cli.zip
- The name of directory is the name of the Module as it appears in the list of modules





Module function



Input image

processing



Output image



Module description

GUI

▼ IO

Input Volume	Select a Volume
Output Volume	Select a Volume

▼ ThresholdParameters

Lower Threshold	10
Upper Threshold	50

XML

```
<image>
  <name>inputVolume</name>
  <label>Input Volume</label>
  <channel>input</channel>
  <index>0</index>
  <description><![CDATA[Inputvolume]]></decription>
</image>
```

```
<integer>
  <name>lowerThreshold</name>
  <longflag>--lowerThreshold</longflag>
  <description><![CDATA[The lower threshold]]></description>
  <label>Lower Threshold</label>
  <default>10</default>
</integer>
<integer>
```



Module coding

```
typedef itk::ImageFileReader<InputImageType> ReaderType;
typedef itk::ImageFileWriter<OutputImageType> WriterType;
typename ReaderType::Pointer reader = ReaderType::New();
reader->SetFileName( inputVolume.c_str() );
itk::GetImageType(inputVolume, pixelType, componentType);
```

```
typedef itk::BinaryThresholdImageFilter<
    InputImageType, OutputImageType> FilterType;
typename FilterType::Pointer filter = FilterType::New();
filter->SetLowerThreshold(lowerThreshold);
```



Compile the module

- From Slicer source directory run the command:
`./Utilities/Scripts/ModuleWizard.py --template
../SimpleThreshold --target ../SimpleThreshold
SimpleThreshold`
- Build extension (pp 9,10)
- Start Slicer
- Set the path to SimpleThreshold in the Application Settings (pp 11,12)
- Restart Slicer



Loading data

3D Slicer 4.2.0-2013-05-30

File Edit View Help

Modules: SimpleThreshold

3DSlicer

Help & Acknowledgement

SimpleThreshold

Parameter set: SimpleThreshold

IO

Input Volume: Select a Volume

Output Volume: Select a Volume

ThresholdParameters

Lower Threshold: 10

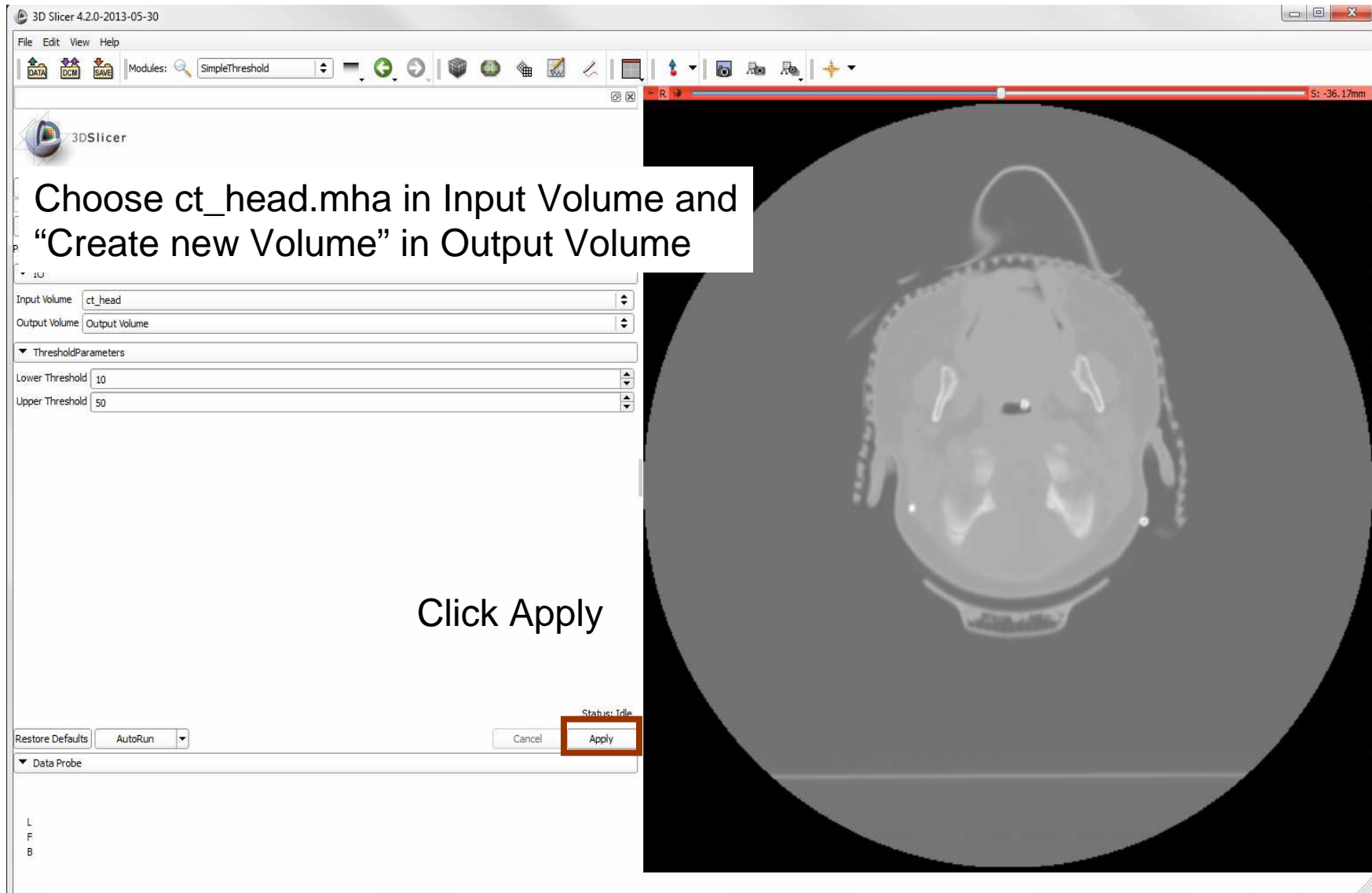
Upper Threshold: 50

Reset

OK Cancel

Click OK

Navigate to the location of ct_head.mha





Saving data

In "Save" menu choose volume and format to save

Click Save



Contact information

If you have questions please contact Greg Sharp at

E-mail: gcsharp@partners.org

Tel: (617) 724 3866



Acknowledgments



National Alliance for Medical Image Computing

NIH U54EB005149



National Institutes of Health

NIH / NCI 6-PO1 CA 21239

Federal share of program income earned by MGH on C06CA059267



Progetto Rocca Foundation

A collaboration between MIT and Politecnico di Milano