



Paul Cézanne, Moulin sur la Coulevre à Pontoise, 1881, Staatliche Museen zu Berlin, Nationalgalerie

Programming into
Slicer3:

**The “Hello Python”
Tutorial**

Sonia Pujol, Ph.D.

Surgical Planning Laboratory
Harvard University

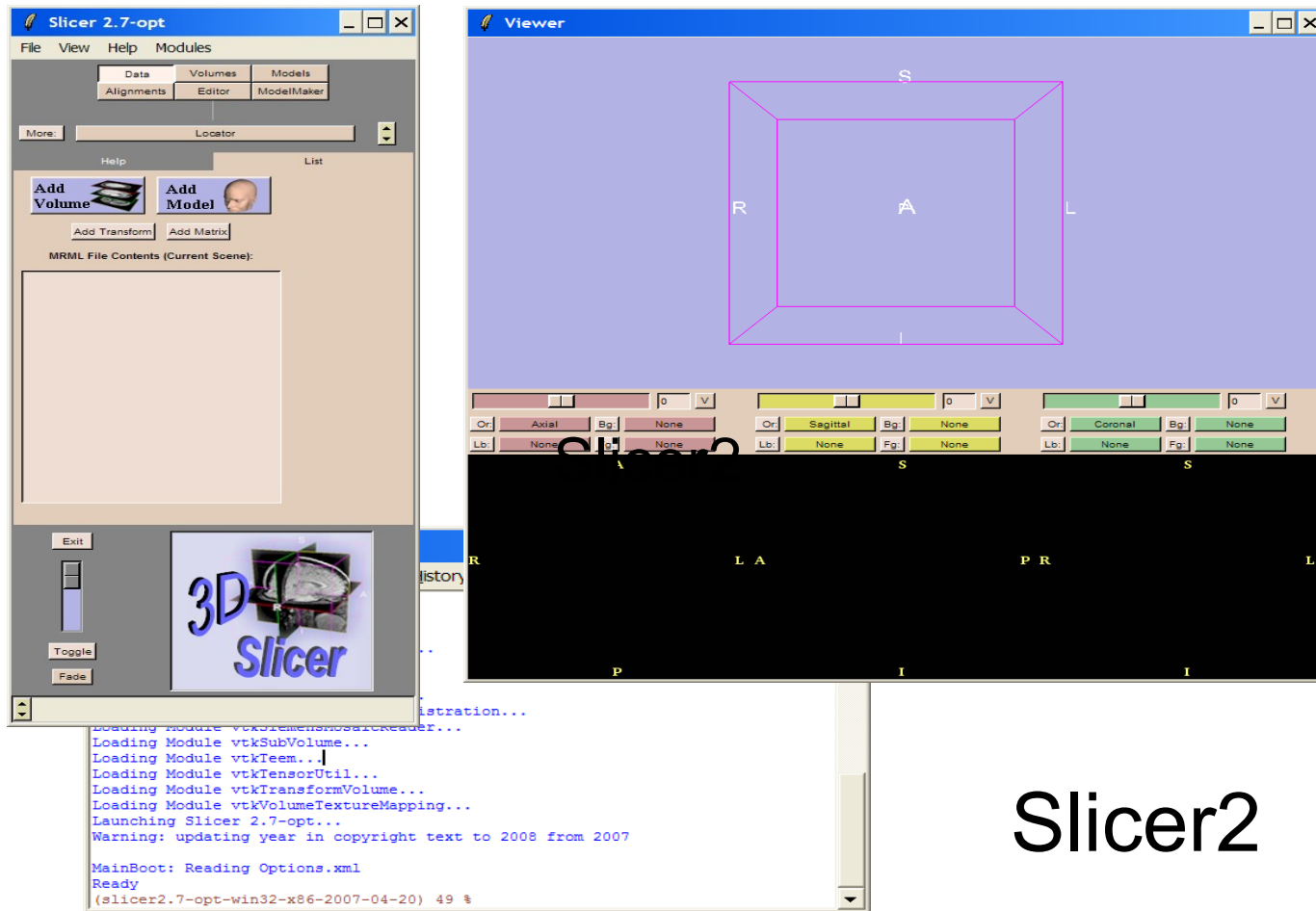
The NA-MIC Kit





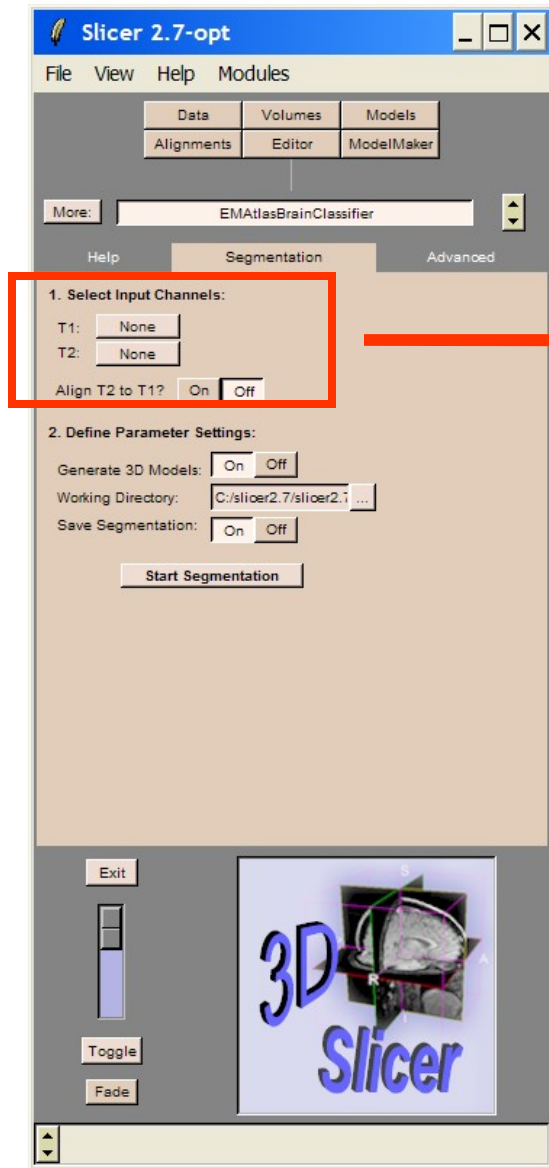
- An **end-user application** for image analysis
- An **open-source environment** for software development
- A software platform that is both **easy to use** for clinical researchers and **easy to extend** for programmers

Before Slicer3



Slicer2

Programming into Slicer2

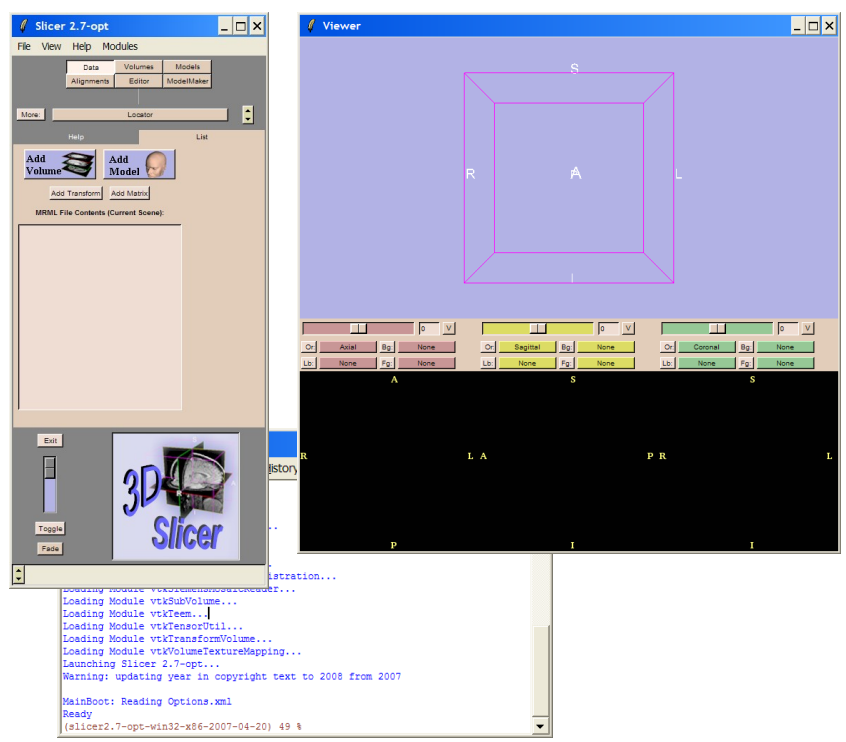


```

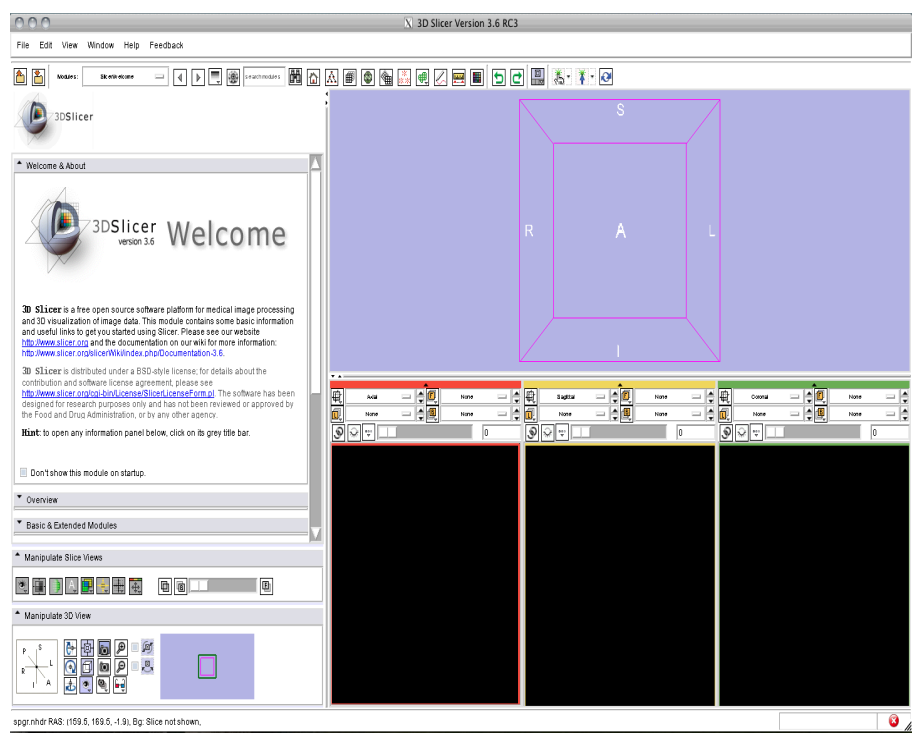
#-----
# 1. Step
#-----
set f $fSeg.fStep1
DevAddLabel $f.ITitle "1. Select Input Channels: " WTA
pack $f.ITitle -side top -padx $Gui(pad) -pady 1 -anchor w
frame $f.flInput -bg $Gui(activeWorkspace)
pack $f.flInput -side top -padx 0 -pady 0 -anchor w
foreach frame "Left Right" {
    frame $f.flInput.f$frame -bg $Gui(activeWorkspace)
    pack $f.flInput.f$frame -side left -padx 0 -pady $Gui(pad) }
foreach LABEL "T1 T2" Input "SPGR T2W" {
    DevAddLabel $f.flInput.fLeft.I$Input " ${LABEL}:"
    pack $f.flInput.fLeft.I$Input -side top -padx $Gui(pad) -pady 1 -anchor w
    set menubutton $f.flInput.fRight.m${Input}Select
    set menu $f.flInput.fRight.m${Input}Select.m
eval {menubutton $menubutton -text [Volume($EMAtlasBrainClassifier(Volume,${Input}),node) GetName]
relief raised -bd 2 -width 9 -menu $menu} $Gui(WMBA)

    eval {menu $menu} $Gui(WMA)
    TooltipAdd $menubutton "Select Volume defining ${Input}"
    set EMAtlasBrainClassifier(mbSeg-${Input}Select) $menubutton
    set EMAtlasBrainClassifier(mSeg-${Input}Select) $menu
    # Have to update at UpdateMRML too
    DevUpdateNodeSelectButton Volume EMAtlasBrainClassifier Seg-${Input}Select Volume,$Input
    pack $menubutton -side top -padx $Gui(pad) -pady 1 -anchor w }
frame $f.Align -bg $Gui(activeWorkspace)
TooltipAdd $f.Align "If the input T1 and T2 are not aligned with each other set flag here"
pack $f.Align -side top -padx 0 -pady 2 -padx $Gui(pad) -anchor w
DevAddLabel $f.Align.IAlign "Align T2 to T1? "
pack $f.Align.IAlign -side left -padx $Gui(pad) -pady 1 -anchor w
foreach value "1 0" text "On Off" width "4 4" {
    eval {radiobutton $f.Align.r$value -width $width -indicatoron 0}
        -text "$text" -value "$value" -variable EMAtlasBrainClassifier(AlignInput) } $Gui(WCA)
    pack $f.Align.r$value -side left -padx 0 -pady 0 }
  
```

From Slicer2 to Slicer3



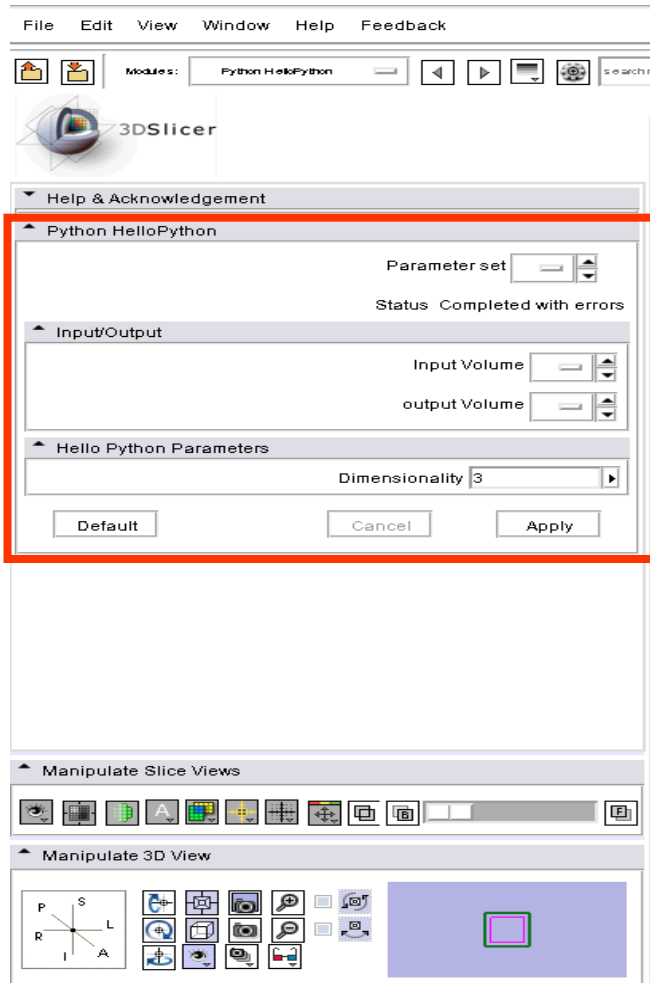
Slicer2



Slicer3



The New Execution Model



```
<?xml version="1.0" encoding="utf-8"?>
<executable>
  <category> Demonstration </category>
  <title> Python HelloPython </title>
  <description> Slicer Developer Course </description>
  <version> 1.0 </version>
  <documentation-url> </documentation-url>
  <license></license>
  <contributor>
    Sonia Pujol, Ph.D., Surgical Planning Laboratory, Harvard Medical School
  </contributor>
  <acknowledgements> National Alliance for Medical Image Computing (NAMIC), Grant
  U54 EB005149. </acknowledgements>
  <parameters>
    <label>Input/Output</label>
    <description>Input/output parameters</description>
    <image>
      <name>helloPython</name>
      <label>Input Volume</label>
      <channel>input</channel>
      <index>0</index>
      <default>None</default>
      <description>Input volume</description>
    </image>
    <image>
      <name>helloPythonOutputVolume</name>
      <label>Output Volume</label>
      <channel>output</channel>
      <index>1</index>
      <default>None</default>
      <description>Output filtered</description>
    </image>
  </parameters>
</executable>
```



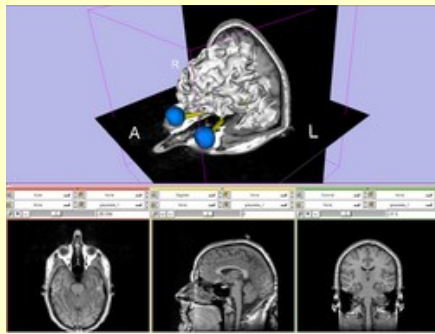
Slicer3 Execution Model

- This course is based on the [Execution Model](#) which provides a mechanism for incorporating command line programs as Slicer modules.
- Jim Miller, Dan Blezek, Bill Lorensen (GE)
- This course uses the Python interpreter that has been integrated to Slicer.



Pre-requisite

- This course supposes that you have taken the following tutorial:



Slicer3 Data Loading and Visualization, Sonia Pujol Ph.D.

- The tutorial is available on the Slicer3.6 101 compendium:
http://www.slicer.org/slicerWiki/index.php/Slicer3.6:Training#Software_tutorials



Material

This course requires the following material

- Slicer3-3.6.1 release version

<http://www.slicer.org/pages/Special:SlicerDownloads>

- HelloPython.zip

http://www.slicer.org/slicerWiki/index.php/Slicer3.6:Training_Software_tutorials

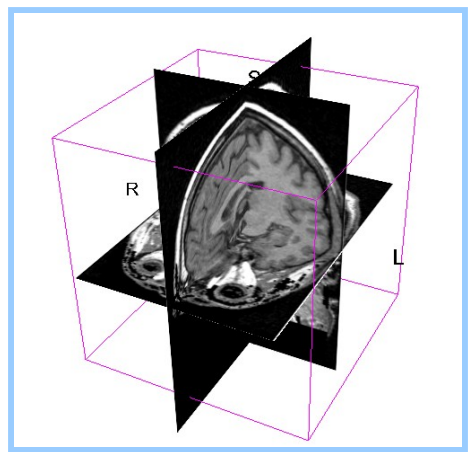
Disclaimer

It is the responsibility of the user of 3DSlicer to comply with both the terms of the license and with the applicable laws, regulations and rules.



HelloPython Course Material

Unzip the HelloPython.zip archive



spgr.nhdr

spgr.raw.gz

(124 SPGR images)

```
#!/usr/bin/env python
XML = """<?xml version="1.0" encoding="utf-8"?>
<executable>
  <category>Demonstration </category>
  <title>Python HelloPython</title>
  <description>Slicer developer course in Python
</description>
  <version>1.0</version>

  <license></license>
  <contributor> This module was developed by Sonia Pujol, Ph.D., Harvard University. </contributor>
  <documentation-url> http://www.slicer.org/slicerWiki/index.php/Slicer3.6:Training </documentation-url>
  <acknowledgements>
    This work is part of the National Alliance for Medical Image Computing (NA-MIC), funded by the National
    Institutes of Health through the NIH Roadmap for Medical Research, Grant U54 EB005149. </acknowledgements>

  <parameters>
    <label>Input/Output</label>
    <description>Input/output parameters</description>
    <image>
      <name>helloPythonInputVolume</name>
      <label>Input Volume</label>
      <channel>input</channel>
      <index>0</index>
      <description>input volume</description>
    </image>
    <image>
      <name>helloPythonOutputVolume</name>
      <label>output Volume</label>
      <channel>output</channel>
      <index>1</index>
      <description>output volume</description>
    </image>
  </parameters>
</executable>
"""
```

HelloPython.py

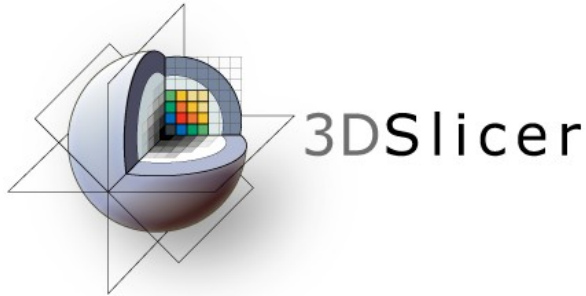


Overview

- Part A: Integration of the HelloPython.py program into Slicer3
- Part B: Implementation of the Laplace operator in the HelloPython module
- Part C: Image Sharpening using the Laplace operator



3DSlicer



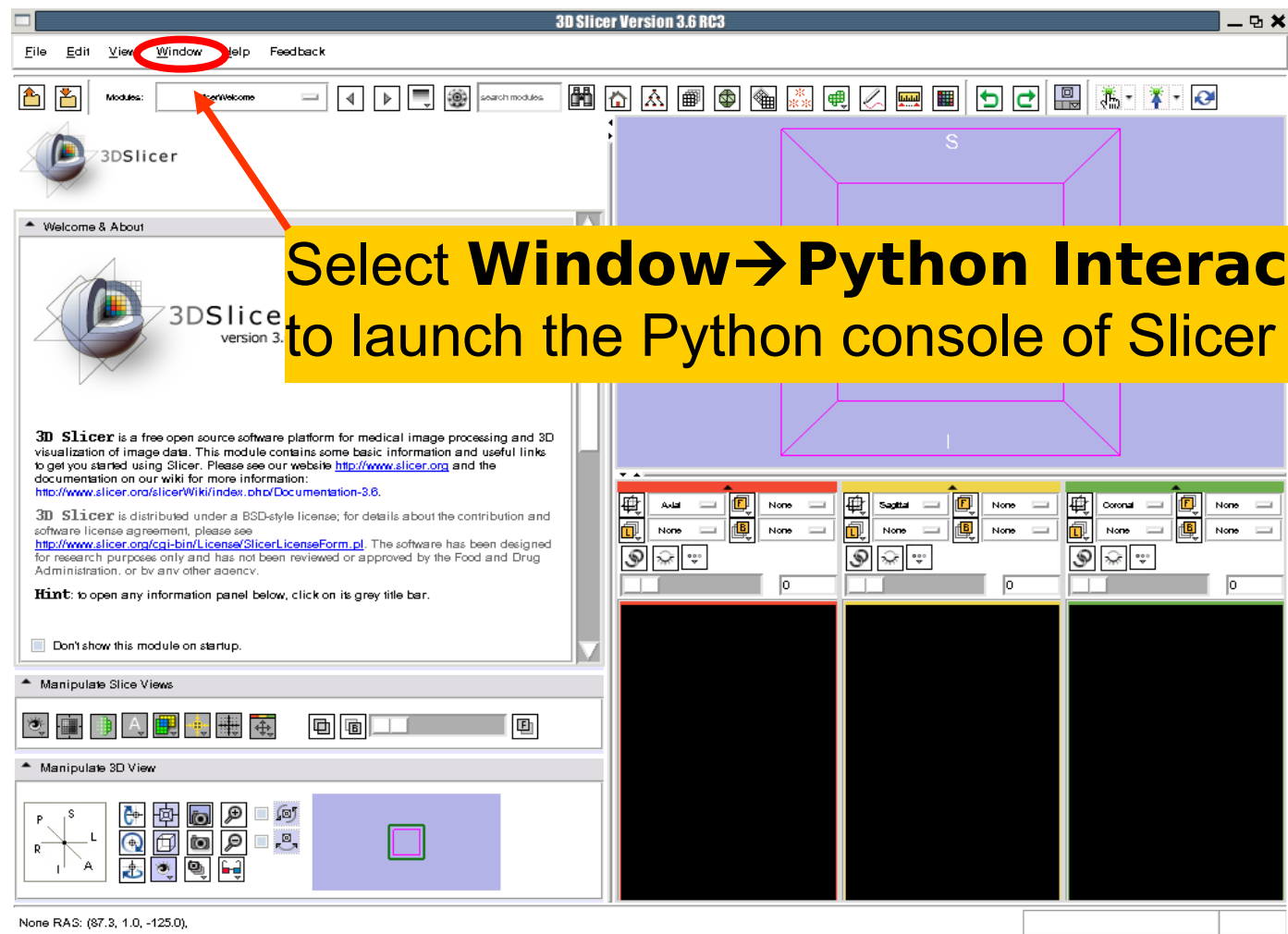
```
#!/usr/bin/env python
XML = """<?xml version="1.0" encoding="utf-8"?>
<executable>
  <category>Demonstration </category>
  <title>Python HelloPythons</title>
  <description> Slicer developer course in Python
  </description>
  <version>1.0</version>

  <license></license>
  <contributors> This module was developed by Sonia Pujol, Ph.D., Harvard University. </contributor>
  <documentation-url> http://www.slicer.org/slicerWiki/index.php/Slicer3.6:Training </documentation-url>
  <acknowledgements>
    This work is part of the National Alliance for Medical Image Computing (NA-MIC), funded by the National
    Institutes of Health through the NIH Roadmap for Medical Research, Grant U54 EB005149. </acknowledgements>

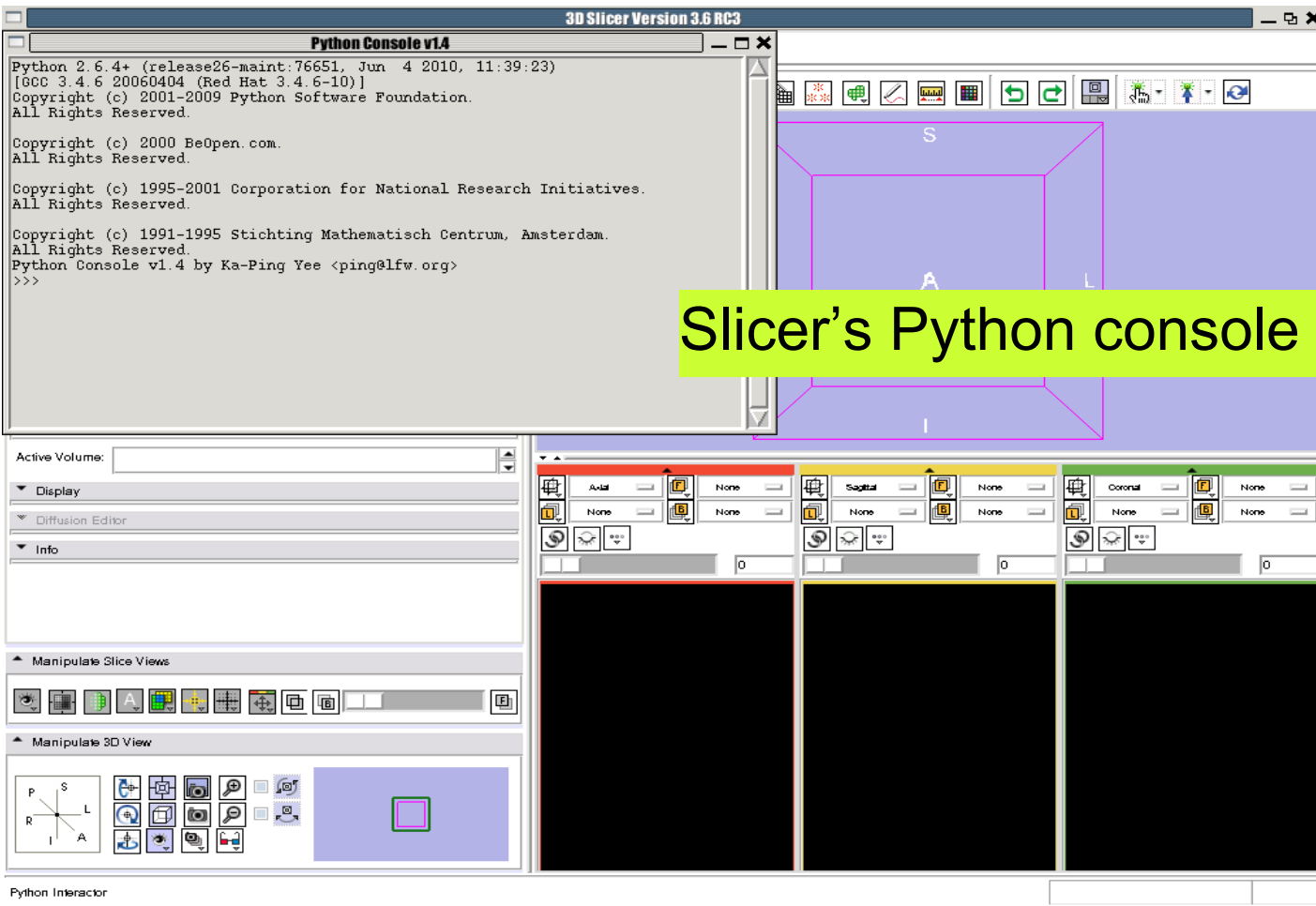
  <parameters>
    <label>Input/Output</label>
    <description>Input/output parameters</description>
    <image>
      <name>helloPythonInputVolume</name>
      <label>Input Volume</label>
      <channel>input</channel>
      <index>0</index>
      <description>input volume</description>
    </image>
    <image>
      <name>helloPythonOutputVolume</name>
      <label>output Volume</label>
      <channel>output</channel>
      <index>1</index>
      <description>output volume</description>
    </image>
  </parameters>
</executable>
"""
```

Part A: Integrating HelloPython into Slicer3

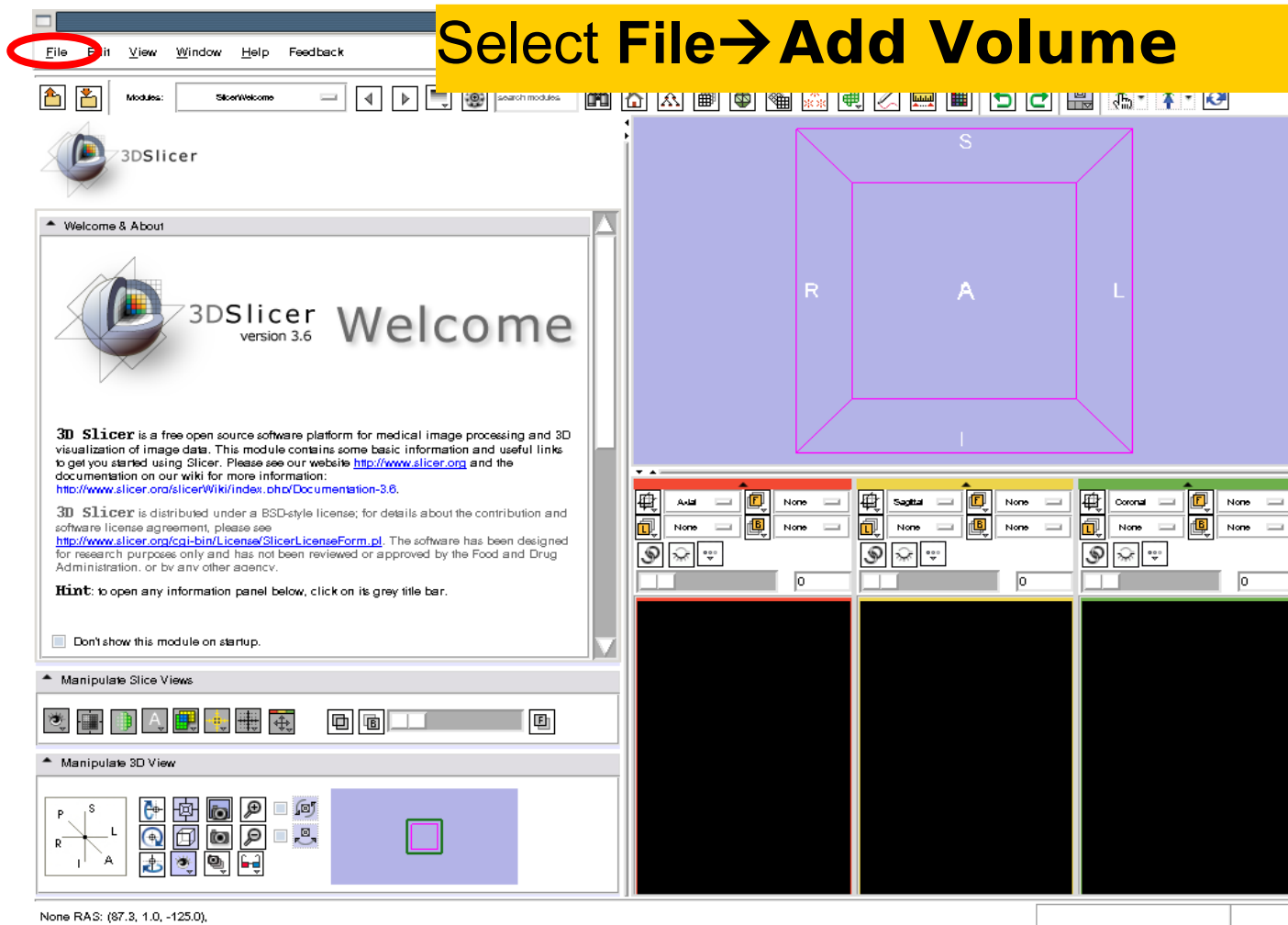
Python Console



Python Console

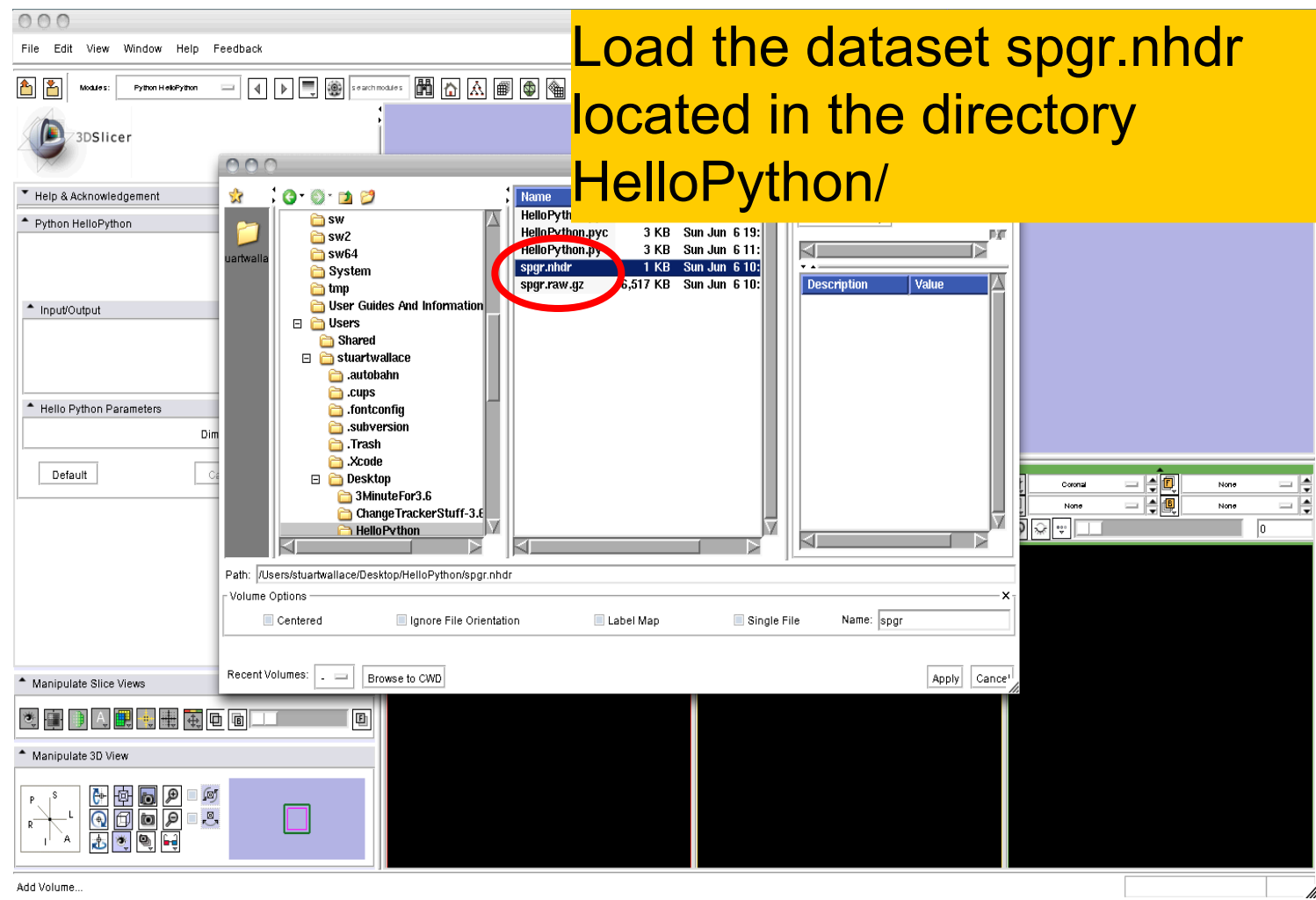


Python Console

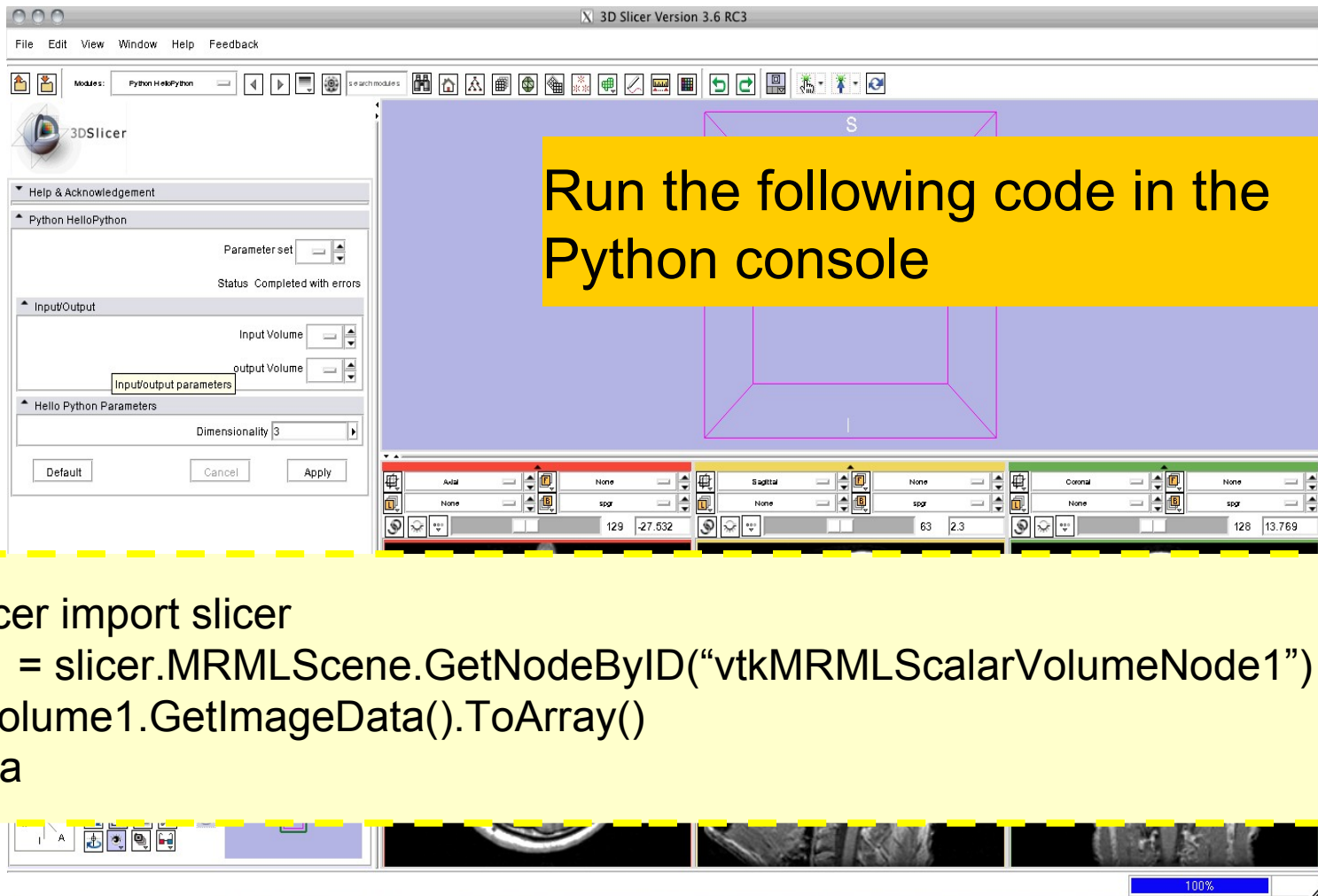


Python Console

Load the dataset spgr.nhdr located in the directory HelloPython/



Python Console

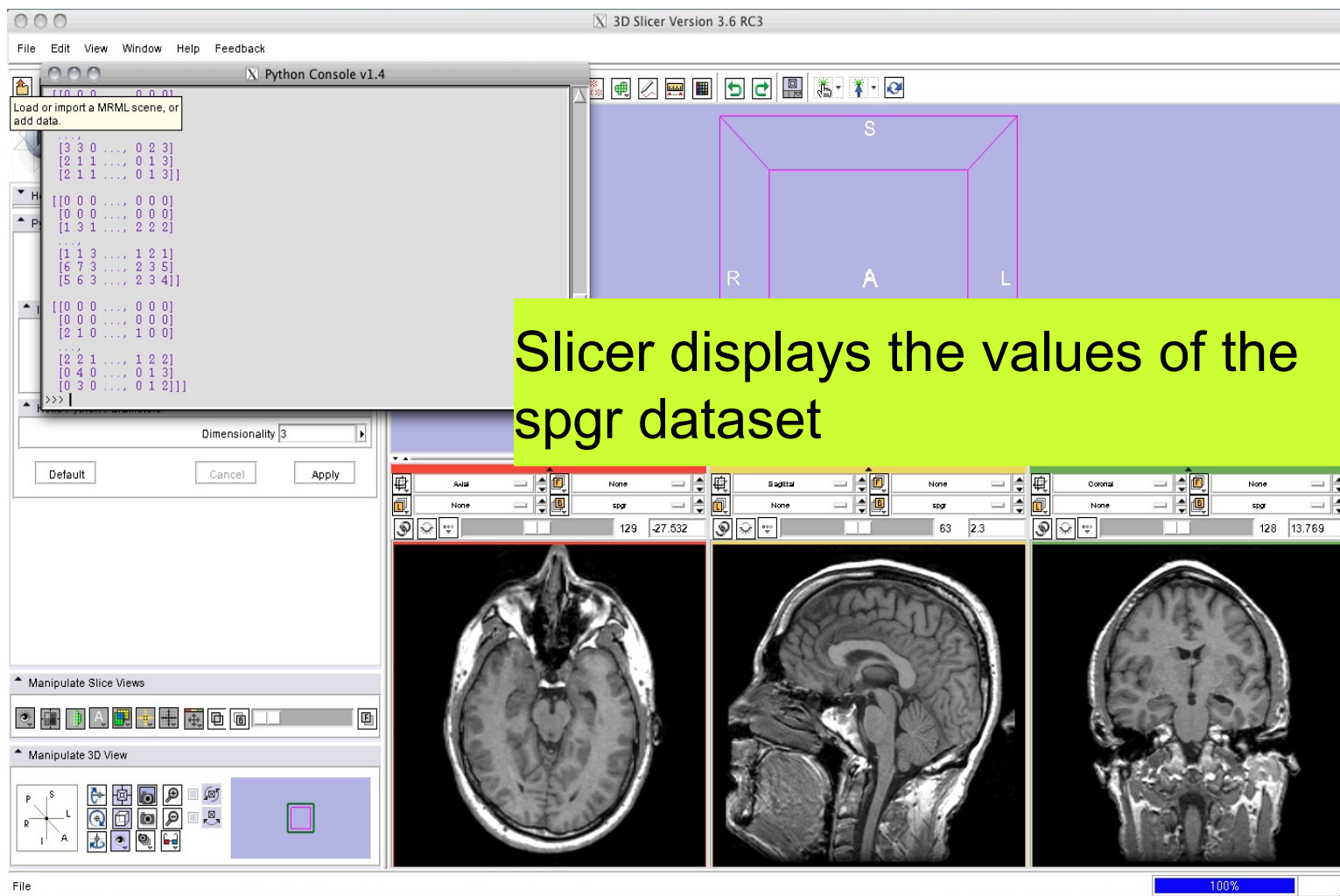


The screenshot shows the 3D Slicer 3.6 RC3 interface. A yellow text box is overlaid on the main 3D view, containing the instruction: "Run the following code in the Python console". Below this, a yellow dashed box contains the following Python code:

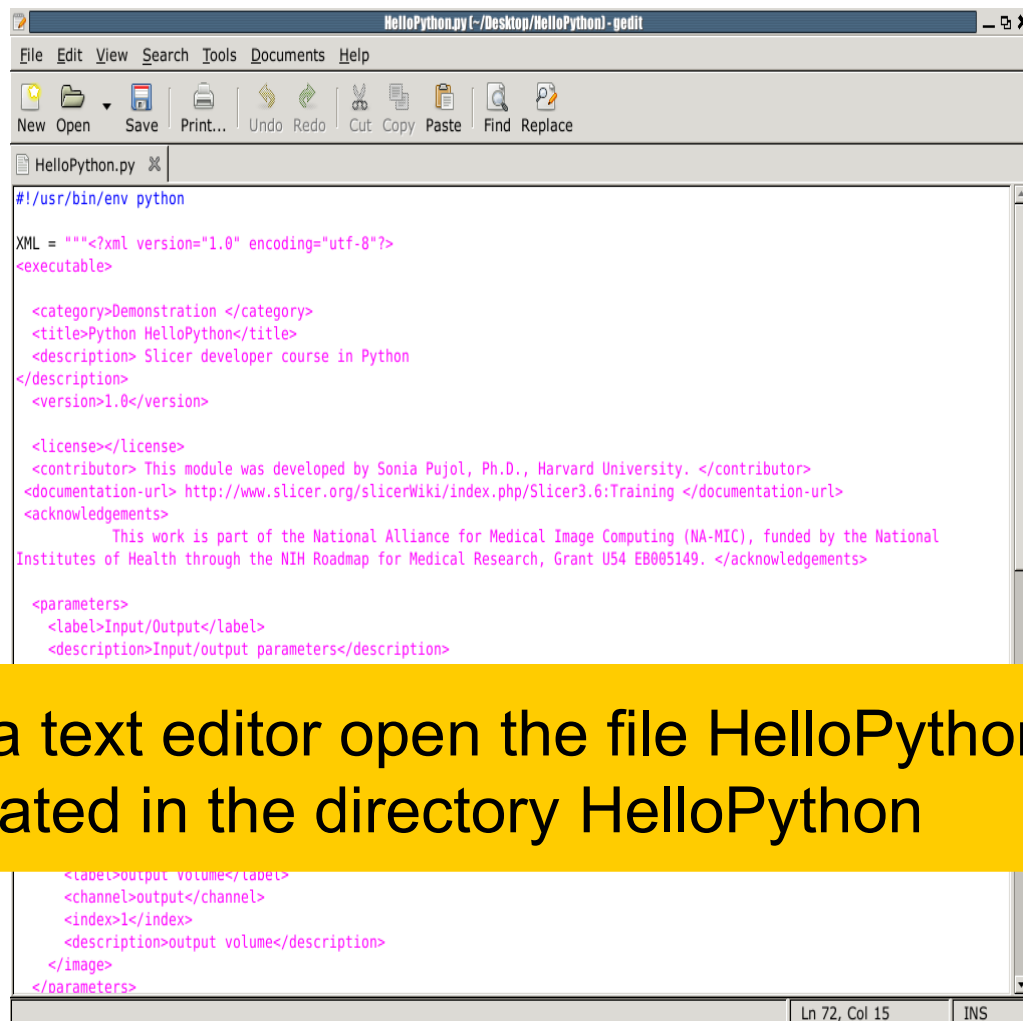
```
from Slicer import slicer
volume1 = slicer.MRMLScene.GetNodeByID("vtkMRMLScalarVolumeNode1")
data = volume1.GetImageData().ToArray()
print data
```

The interface also shows a parameter panel on the left for "Python HelloPython" and a bottom toolbar with various icons.

Python Console



HelloPython.py



```
#!/usr/bin/env python

XML = """<?xml version="1.0" encoding="utf-8"?>
<executable>

  <category>Demonstration </category>
  <title>Python HelloPython</title>
  <description> Slicer developer course in Python
</description>
  <version>1.0</version>

  <license></license>
  <contributor> This module was developed by Sonia Pujol, Ph.D., Harvard University. </contributor>
  <documentation-url> http://www.slicer.org/slicerWiki/index.php/Slicer3.6:Training </documentation-url>
  <acknowledgements>
    This work is part of the National Alliance for Medical Image Computing (NA-MIC), funded by the National
    Institutes of Health through the NIH Roadmap for Medical Research, Grant U54 EB005149. </acknowledgements>

  <parameters>
    <label>Input/Output</label>
    <description>Input/output parameters</description>

    <label>output volume</label>
    <channel>output</channel>
    <index>1</index>
    <description>output volume</description>
  </image>
</parameters>

```

In a text editor open the file HelloPython.py located in the directory HelloPython

Module Description

Module Parameters

Execute function

```
#!/usr/bin/env python
XML = """<?xml version="1.0" encoding="utf-8"?>
<executable>
  <category>Demonstration </category>
  <title>Python HelloPython</title>
  <description> Slicer developer course in Python
</description>
  <version>1.0</version>

  <license></license>
  <contributor> This module was developed by Sonia Pujol, Ph.D., Harvard University. </contributor>
  <documentation-url> http://www.slicer.org/slicerWiki/index.php/Slicer3.6:Training </documentation-url>
  <acknowledgements>
    This work is part of the National Alliance for Medical Image Computing (NA-MIC), funded by the National Institutes of
    Health through the NIH Roadmap for Medical Research, Grant U54 EB005149. </acknowledgements>

  <parameters>
    <label>Input/Output</label>
    <description>Input/output parameters</description>
    <image>
      <name>helloPythonInputVolume</name>
      <label>Input Volume</label>
      <channel>input</channel>
      <index>0</index>
      <description>input volume</description>
    </image>
    <image>
      <name>helloPythonOutputVolume</name>
      <label>output Volume</label>
      <channel>output</channel>
      <index>1</index>
      <description>output volume</description>
    </image>
  </parameters>
</executable>

<parameters>
  <label>Hello Python Parameters</label>
  <description>Parameters of the Python Hello Python module </description>
  <integer>
    <name>dimensionality</name>
    <longflag>dimensionality</longflag>
    <description>Dimensionality of the Laplace operator</description>
    <label>Dimensionality</label>
    <default>3</default>
    <constraints>
      <minimum>2</minimum>
      <maximum>3</maximum>
    </constraints>
  </integer>
</parameters>

</executable>
"""

def Execute ():
    Slicer = __import__("Slicer")
    slicer = Slicer.slicer
    scene = slicer.HRHLScene

    return
```



Module Description

```
#!/usr/bin/env python
XML = """<?xml version="1.0" encoding="utf-8"?>
<executable>
<category>Demonstration </category>
<title>Python HelloPython</title>
<description> Slicer developer course in Python </description>
<version>1.0</version>
<license></license>
<contributor> This module was developed by Sonia Pujol, Ph.D., Harvard University. </contributor>
<documentation-url> http://www.slicer.org/slicerWiki/index.php/Slicer3.6:Training </documentation-url>
<acknowledgements>
    This work is part of the National Alliance for Medical Image Computing (NA-MIC),
    funded by the National Institutes of Health through the NIH Roadmap for Medical Research,
    Grant U54 EB005149.
</acknowledgements>
```

Module Parameters

```
<parameters>  
  <label>Input/Output</label>  
  <description>Input/output parameters</description>
```

Input
Volume

```
<image>  
  <name>HelloPythonInputVolume</name>  
  <label>Input Volume</label>  
  <channel>input</channel>  
  <index>0</index>  
  <description>input volume</description>  
</image>
```

A file that
specifies
the image

Output
Volume

```
<image>  
  <name>HelloPythonOutputVolume</name>  
  <label>Output Volume</label>  
  <channel>output</channel>  
  <index>1</index>  
  <description>output volume</description>  
</image>  
</parameters>
```

Execute Function

```
def Execute ():  
  
    Slicer = __import__("Slicer")  
    slicer = Slicer.slicer  
    scene = slicer.MRMLScene  
  
    return
```




Integrating HelloPython to Slicer3

3D Slicer Version 3.6 RC3

Click on the View → Application Settings in the main menu

Application Settings

100%

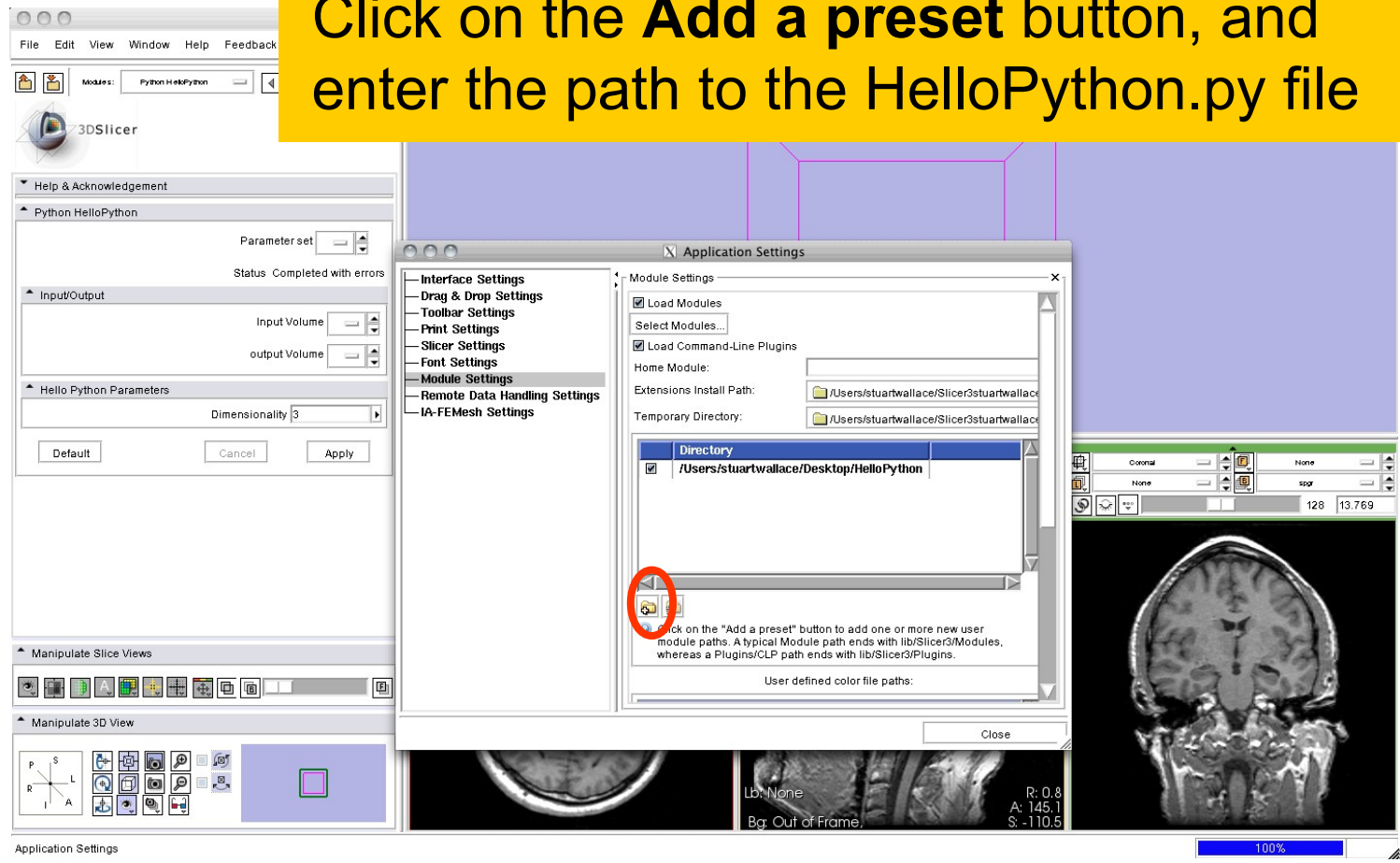


Integrating HelloPython to Slicer3

The screenshot displays the 3D Slicer 3.6 RC3 interface. A yellow callout box with the text "Select Module Settings from the Application Settings GUI" is overlaid on the top right. The "Application Settings" dialog is open, showing the "Module Settings" tab. Under "Module Settings", the "Select Modules..." button is visible. Below it, a list of directories is shown, with "/Users/stuartwallace/Desktop/HelloPython" selected. The "Interface Settings" tab is also visible on the left side of the dialog. In the background, the "Python HelloPython" module settings are visible, showing a "Parameter set" dropdown and a "Status" field that says "Completed with errors". The main 3D view shows a coronal MRI slice of a brain.

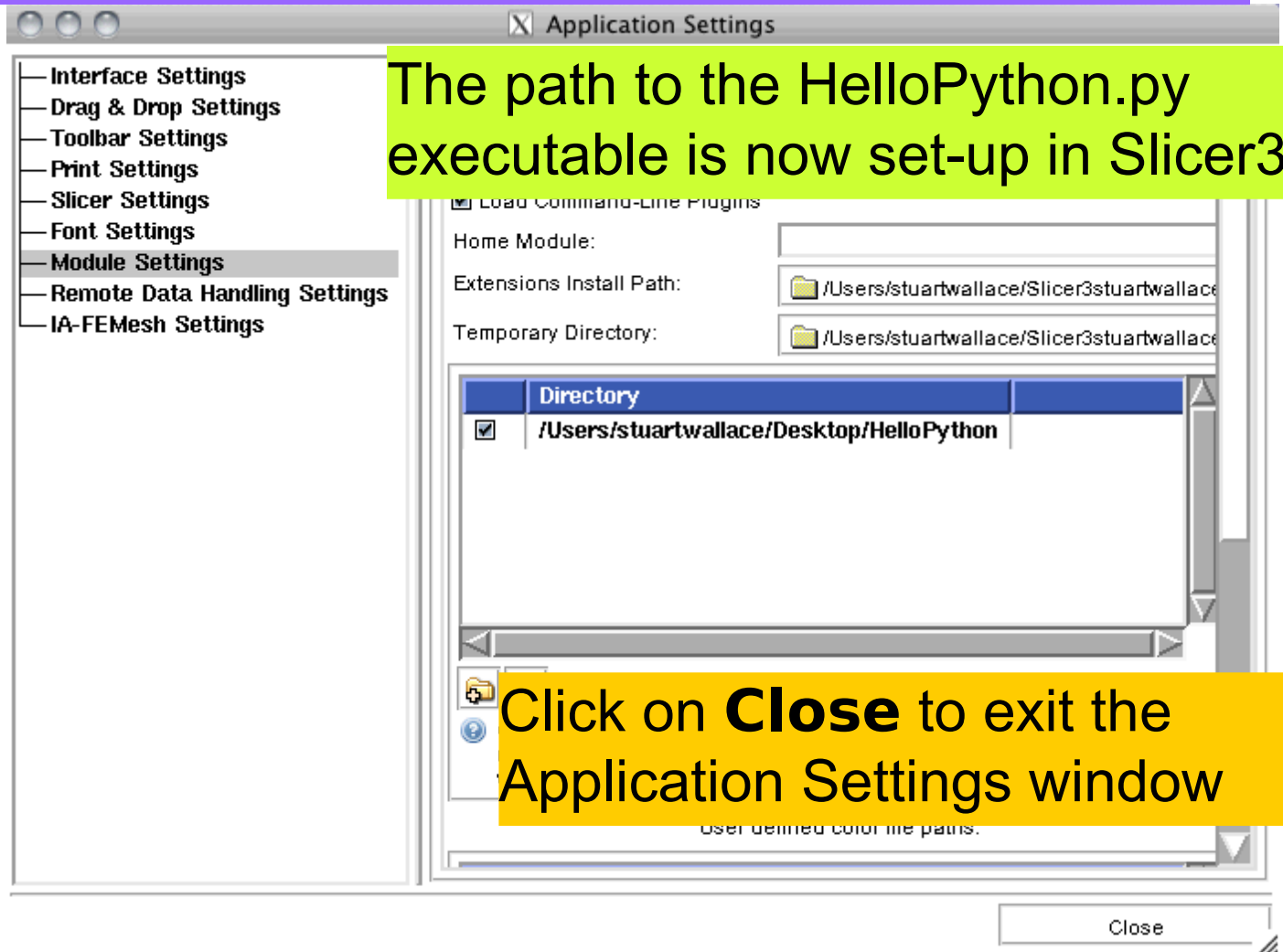
Integrating HelloPython to Slicer3

Click on the **Add a preset** button, and enter the path to the HelloPython.py file



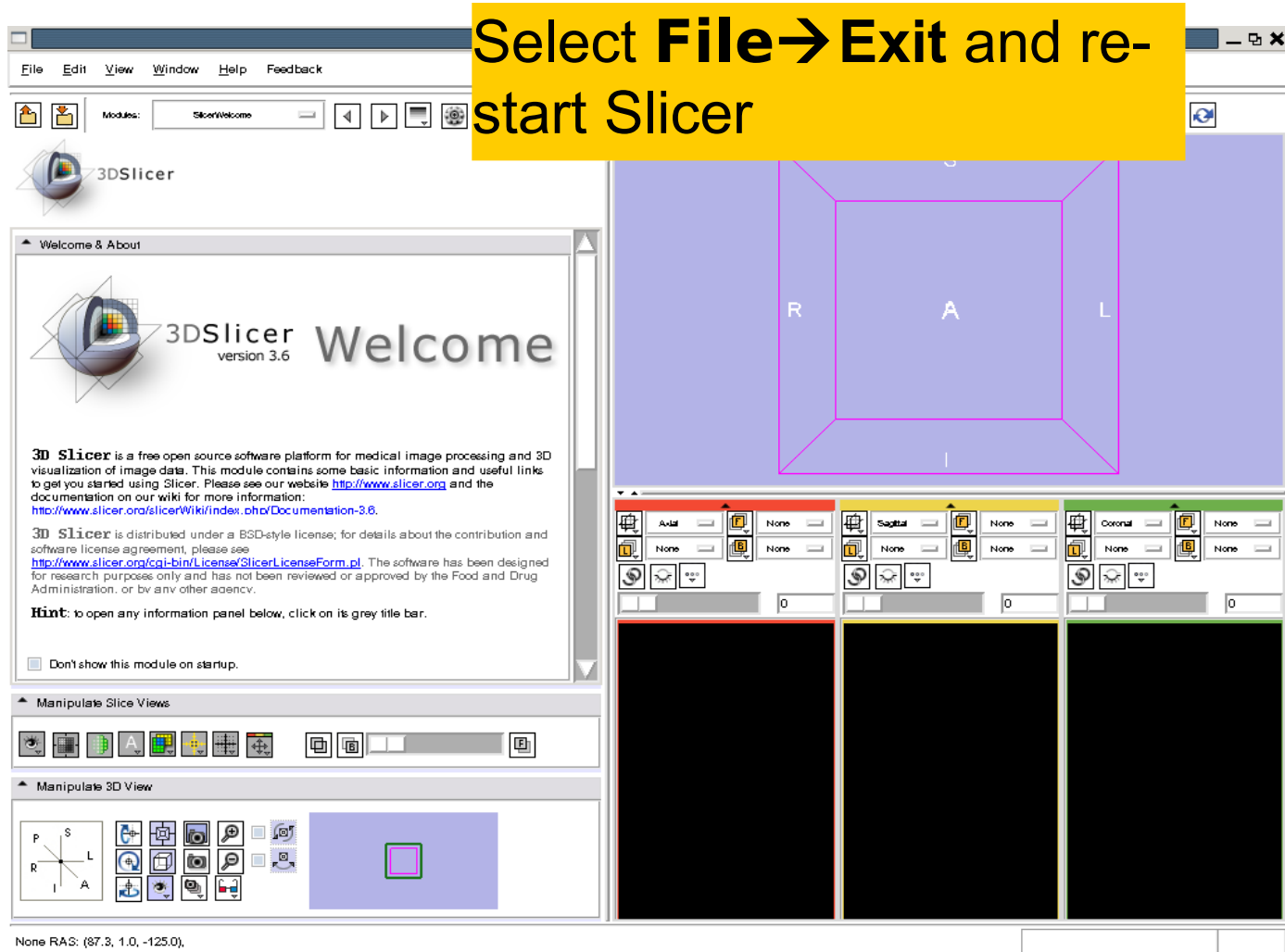


Integrating HelloPython to Slicer3



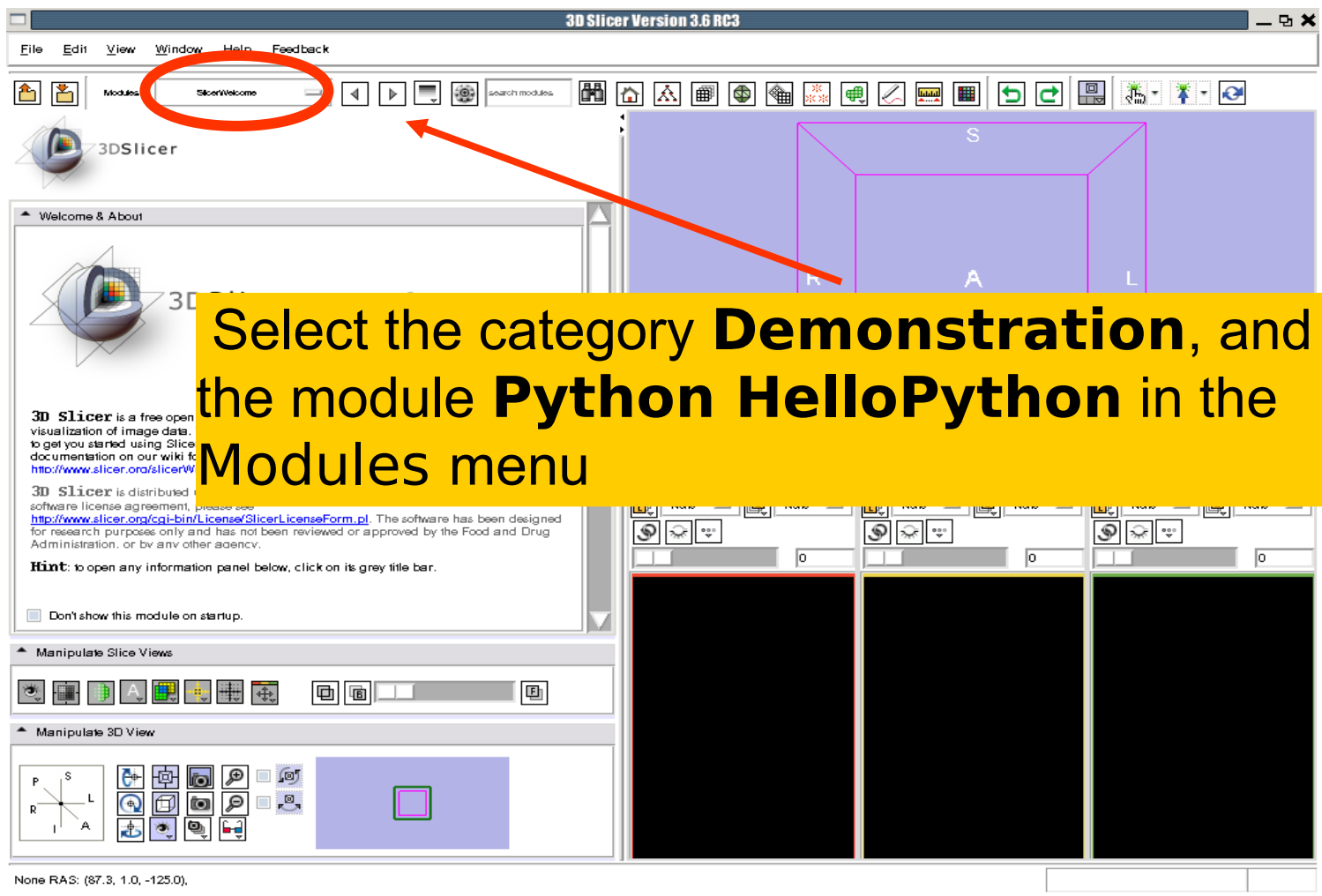


Integrating HelloPython to Slicer3

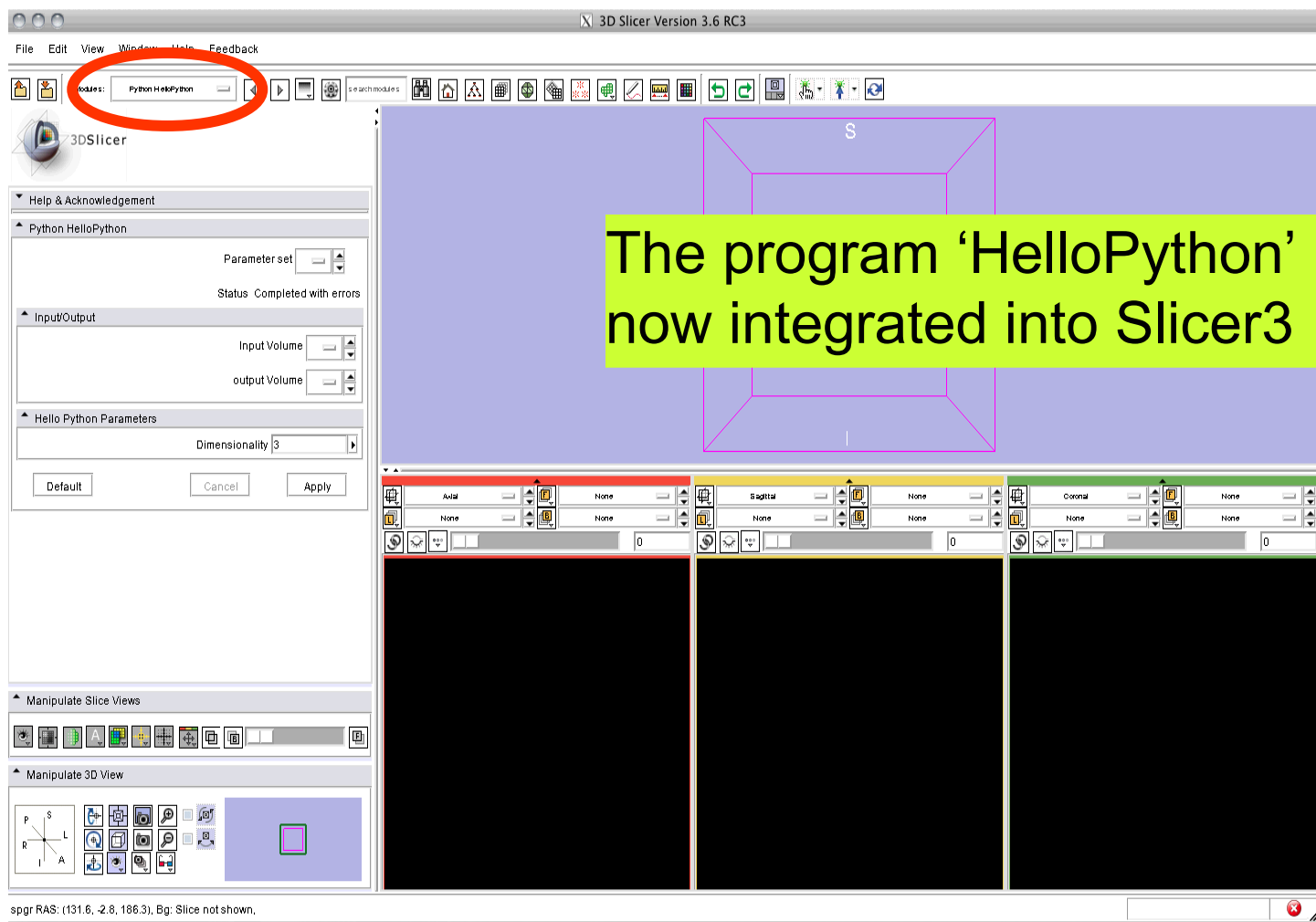


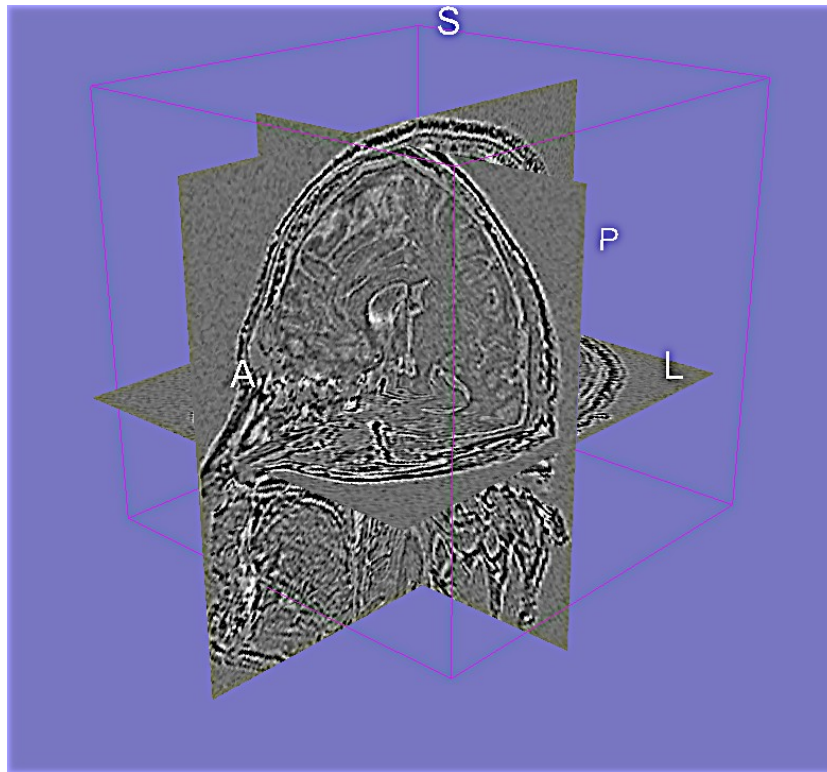


HelloPython module



HelloPython Module





Part B: Implementing the Laplace* Operator

*named after Pierre-Simon, Marquis de Laplace (1749-1827)



Execute Function

```
def Execute (HelloPythonInputVolume, HelloPythonOutputVolume):
```

```
    Slicer = __import__("Slicer")
```

```
    slicer = Slicer.slicer
```

```
    scene = slicer.MRMLScene
```

```
    inputVolume = scene.GetNodeByID(HelloPythonInputVolume)
```

```
    outputVolume = scene.GetNodeByID(HelloPythonOutputVolume)
```

```
    return
```

Add the I/O code

Laplace Operator

```
def Execute (HelloPythonInputVolume, HelloPythonOutputVolume):
```

```
    Slicer = __import__("Slicer")
```

```
    slicer = Slicer.slicer
```

```
    scene = slicer.MRMLScene
```

```
    inputVolume = scene.GetNodeByID(HelloPythonInputVolume)
```

```
    outputVolume = scene.GetNodeByID(HelloPythonOutputVolume)
```

```
    laplacian = slicer.vtkImageLaplacian()
```

```
    laplacian.SetInput(inputVolume.GetImageData())
```

```
    return
```

Add the Laplace operator

Laplace Operator

```
<parameters>
  <label>Input/Output</label>
  <description>Input/output parameters</description>
  <image>
    <name>HelloPythonInputVolume</name>
    <label>Input Volume</label>
    <channel>input</channel>
    <index>0</index>
    <description>input volume</description>
  </image>
  <image>
    <name>HelloPythonOutputVolume</name>
    <label>Output Volume</label>
    <channel>output</channel>
    <index>1</index>
    <description>output volume</description>
  </image>
</parameters>
```

Add a new parameter group for the Laplace operator

```
<parameters>
  <label>Hello Python Parameters</label>
  <description> Parameters of the Python Hello Python module </description>
</parameters>
```

Laplace Operator

```
<parameters>
  <label>Hello Python Parameters</label>
  <description>Parameters of the Python Hello Python module</description>
  <integer>
    <name>dimensionality</name>
    <longflag>dimensionality</longflag>
    <description>Dimensionality of the Laplace operator</description>
    <label>Dimensionality</label>
    <default>3</default>
    <constraints>
      <minimum>2</minimum>
      <maximum>3</maximum>
    </constraints>
  </integer>
</parameters>
```

Add the Laplace
operator's
dimensionality

Laplace Operator

```
def Execute (HelloPythonInputVolume, HelloPythonOutputVolume,
dimensionality=3):
    Slicer = __import__("Slicer")
    slicer = Slicer.slicer
    scene = slicer.MRMLScene
    inputVolume = scene.GetNodeByID(HelloPythonInputVolume)
    outputVolume = scene.GetNodeByID(HelloPythonOutputVolume)
    laplacian = slicer.vtkImageLaplacian()
    laplacian.SetInput(inputVolume.GetImageData())
    laplacian.SetDimensionality(dimensionality)

return
```

Set-up the corresponding dimensionality parameter in the Python code

Laplace Operator

```
def Execute (HelloPythonInputVolume, HelloPythonOutputVolume,
dimensionality=3):
    Slicer = __import__("Slicer")
    slicer = Slicer.slicer
    scene = slicer.MRMLScene
    inputVolume = scene.GetNodeByID(HelloPythonInputVolume)
    outputVolume = scene.GetNodeByID(HelloPythonOutputVolume)
    laplacian = slicer.vtkImageLaplacian()
    laplacian.SetInput(inputVolume.GetImageData())
    laplacian.SetDimensionality(dimensionality)
    laplacian.Update()
    outputVolume.SetAndObserveImageData(laplacian.GetOutput())
    return
```

Add code to get the output of the Laplace operator

Laplace Operator

```
def Execute (HelloPythonInputVolume, HelloPythonOutputVolume,
dimensionality=3):
    Slicer = __import__("Slicer")
    slicer = Slicer.slicer
    scene = slicer.MRMLScene
    inputVolume = scene.GetNodeByID(HelloPythonInputVolume)
    outputVolume = scene.GetNodeByID(HelloPythonOutputVolume)
    laplacian = slicer.vtkImageLaplacian()
    laplacian.SetInput(inputVolume.GetImageData())
    laplacian.SetDimensionality(dimensionality)
    laplacian.Update()
    outputVolume.SetAndObserveImageData(laplacian.GetOutput())
    matrix = slicer.vtkMatrix4x4()
    inputVolume.GetIJKToRASMatrix(matrix)
    outputVolume.SetIJKToRASMatrix(matrix)
    return
```

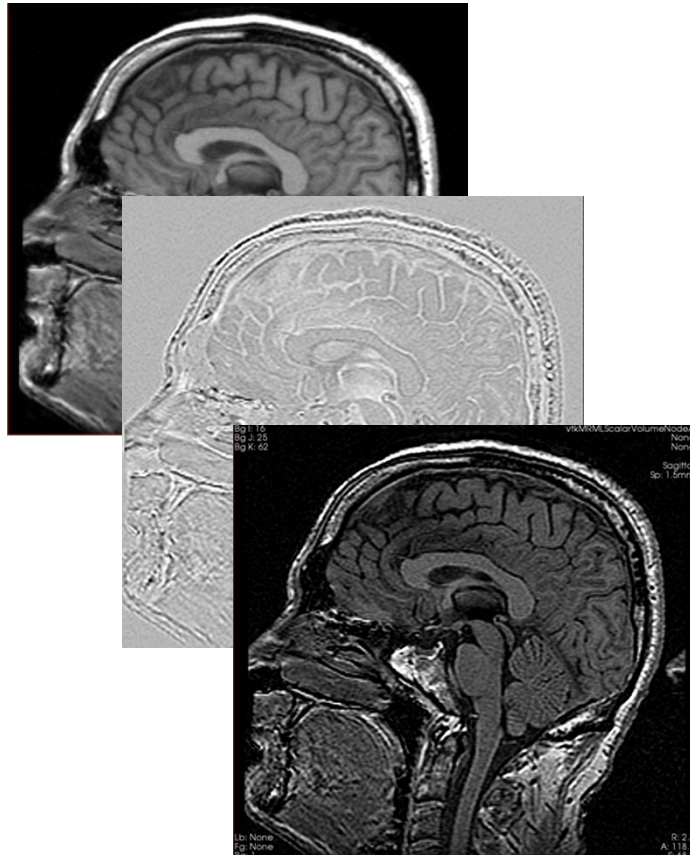
Place back the Laplacian of the image in the RAS reference system.



Integrating HelloPython to Slicer3

```
def Execute (HelloPythonInputVolume, HelloPythonOutputVolume,
dimensionality=3):
    Slicer = __import__("Slicer")
    slicer = Slicer.slicer
    scene = slicer.MRMLScene
    inputVolume = scene.GetNodeByID(HelloPythonInputVolume)
    outputVolume = scene.GetNodeByID(HelloPythonOutputVolume)
    laplacian = slicer.vtkImageLaplacian()
    laplacian.SetInput(inputVolume.GetImageData())
    laplacian.SetDimensionality(dimensionality)
    laplacian.Update()
    outputVolume.SetAndObserveImageData(laplacian.GetOutput())
    matrix = slicer.vtkMatrix4x4()
    inputVolume.GetIJKToRASMatrix(matrix)
    outputVolume.SetIJKToRASMatrix(matrix)
    return
```

Save the HelloPython.py file and exit Slicer.



Part C: Image Sharpening with the Laplace Operator



Running the Laplace Operator

File Edit View Window Help Feedback

Python HelloPython

3DSlicer

Help & Acknowledgement

Python HelloPython

Parameter set

Status Completed with errors

Input/Output

Input Volume

output Volume

Hello Python Parameters

Dimensionality 3

Default Cancel Apply

Manipulate Slice Views

Manipulate 3D View

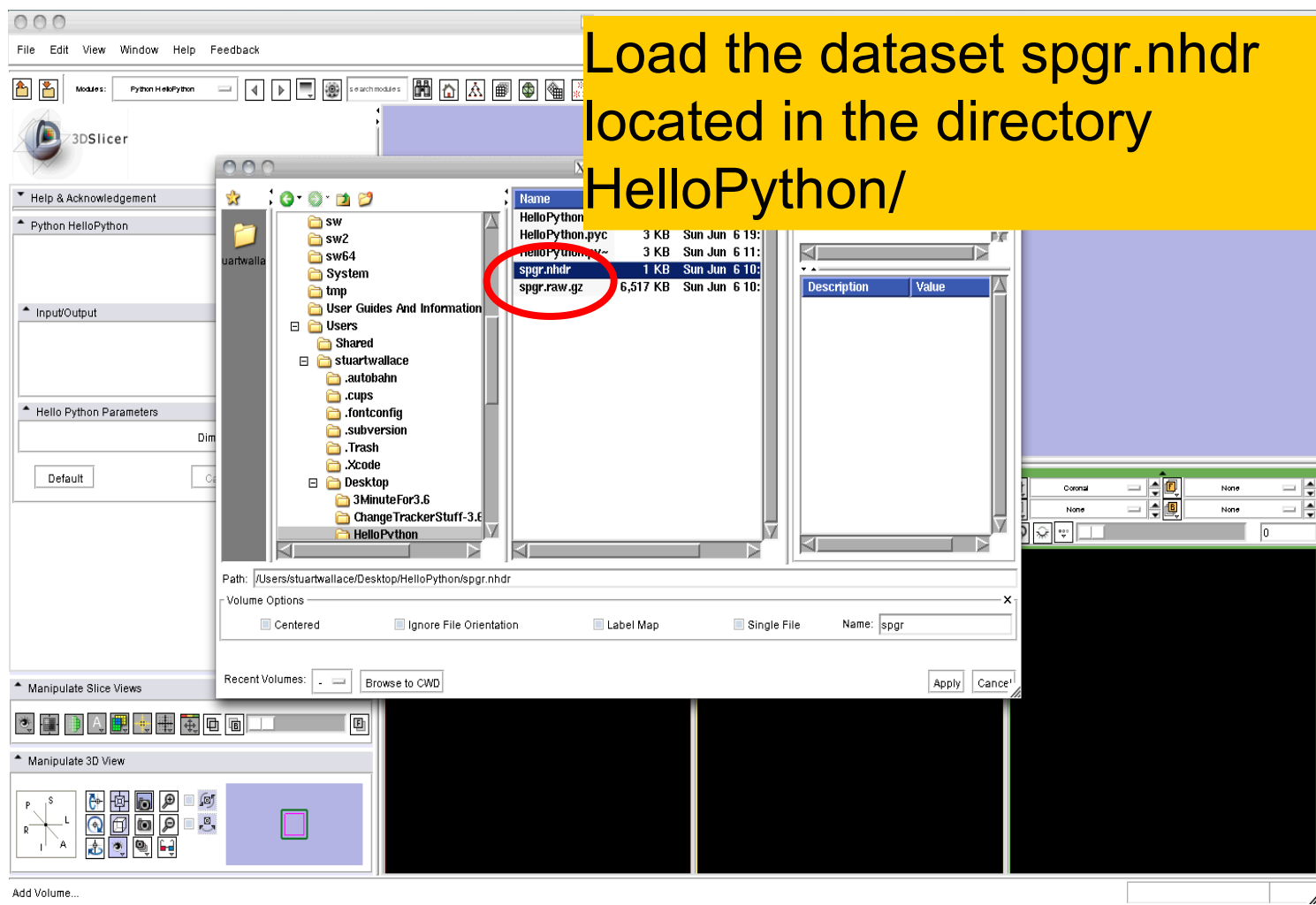
spgr RAS: (131.6, -2.8, 186.3), Bg: Slice not shown.

Re-start Slicer and select File → Add Volume

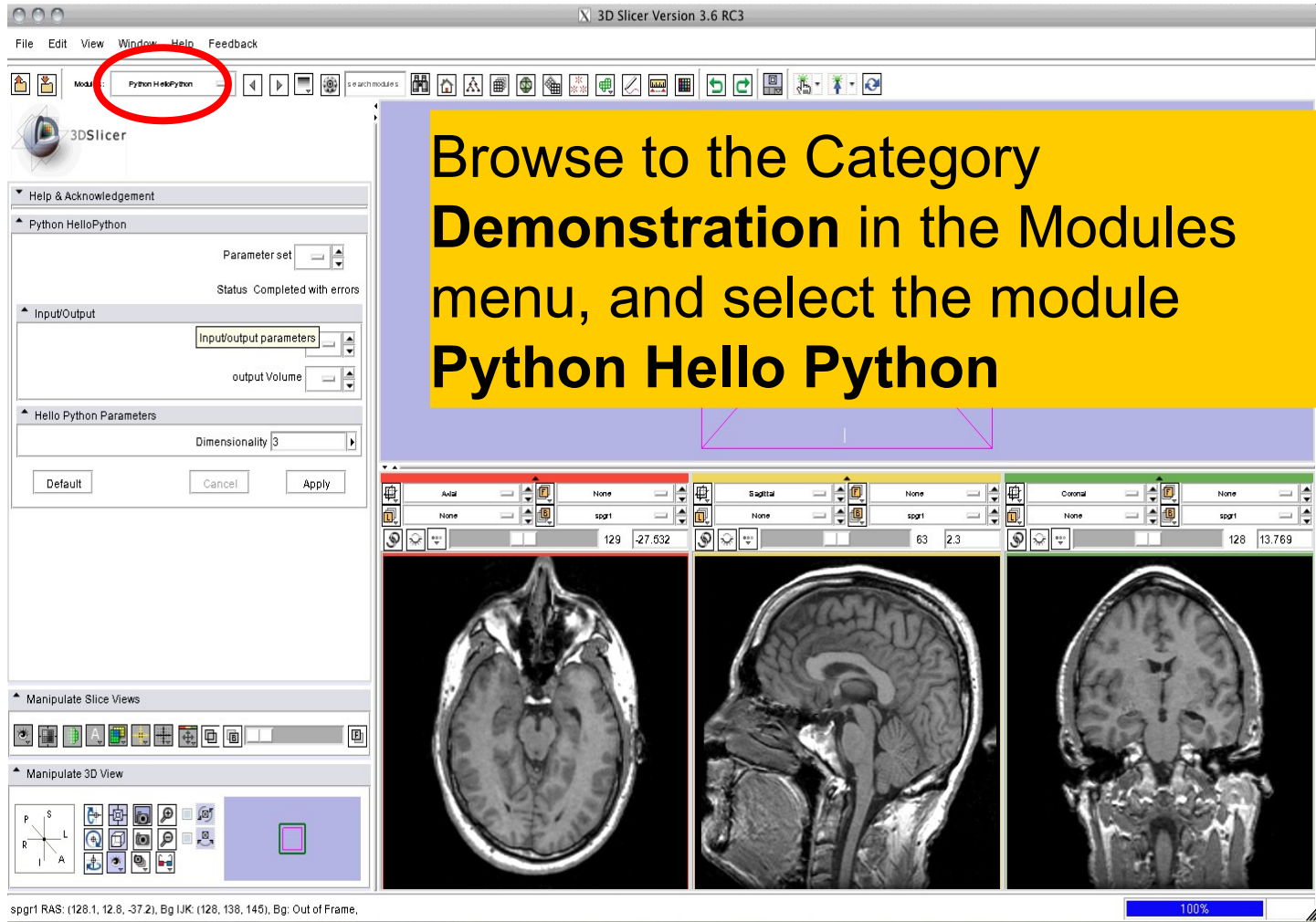


Running the Laplace Operator

Load the dataset spgr.nhdr located in the directory HelloPython/

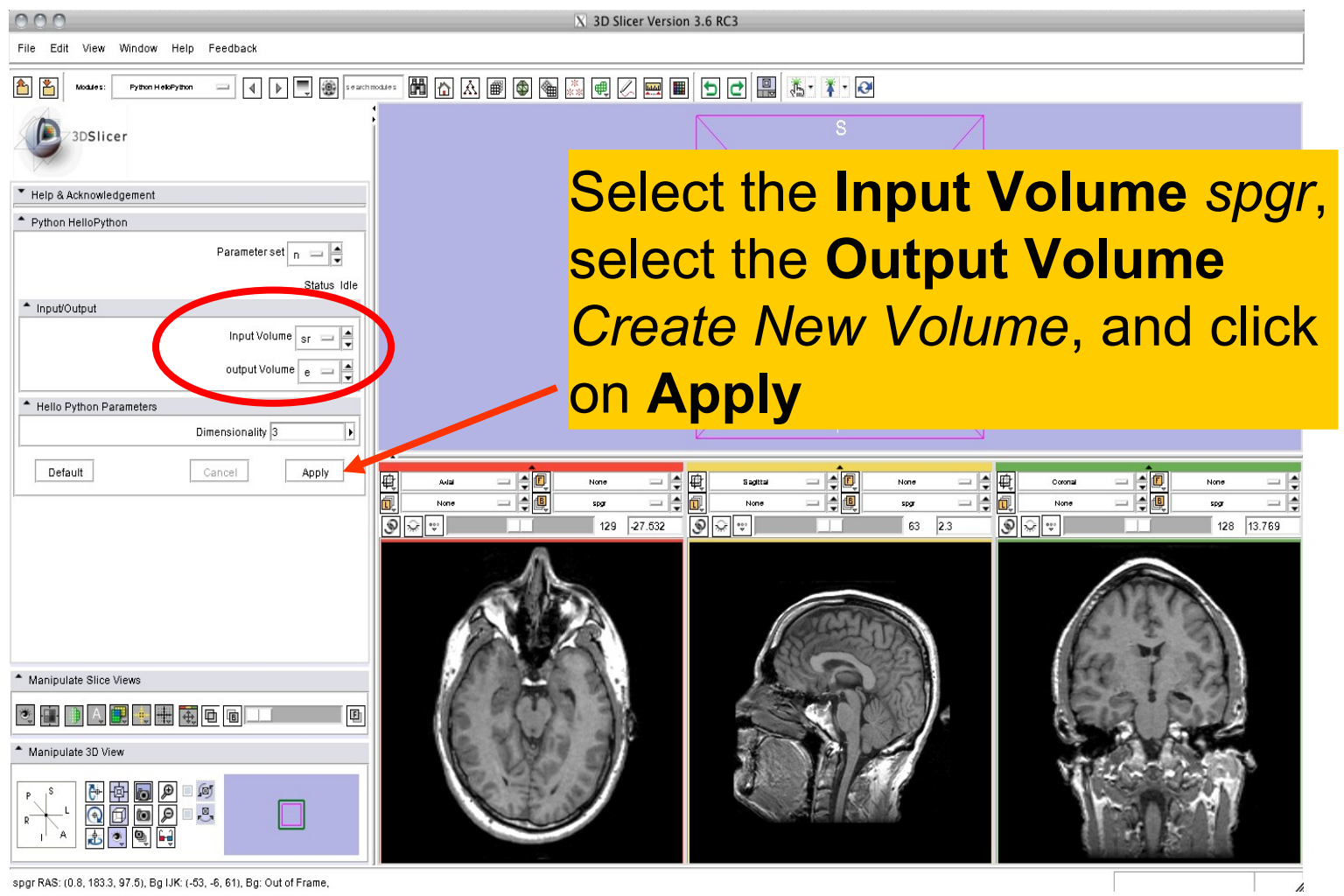


Running the Laplace Operator



The screenshot shows the 3D Slicer 3.6 RC3 interface. The 'Modules' menu is open, and 'Python Hello Python' is selected, highlighted with a red circle. A yellow text box on the right contains the instruction: 'Browse to the Category Demonstration in the Modules menu, and select the module Python Hello Python'. The interface also shows the 'Python Hello Python' module's parameter set, status, and input/output options. The main view displays three orthogonal MRI slices: Axial, Sagittal, and Coronal.

Running the Laplace Operator

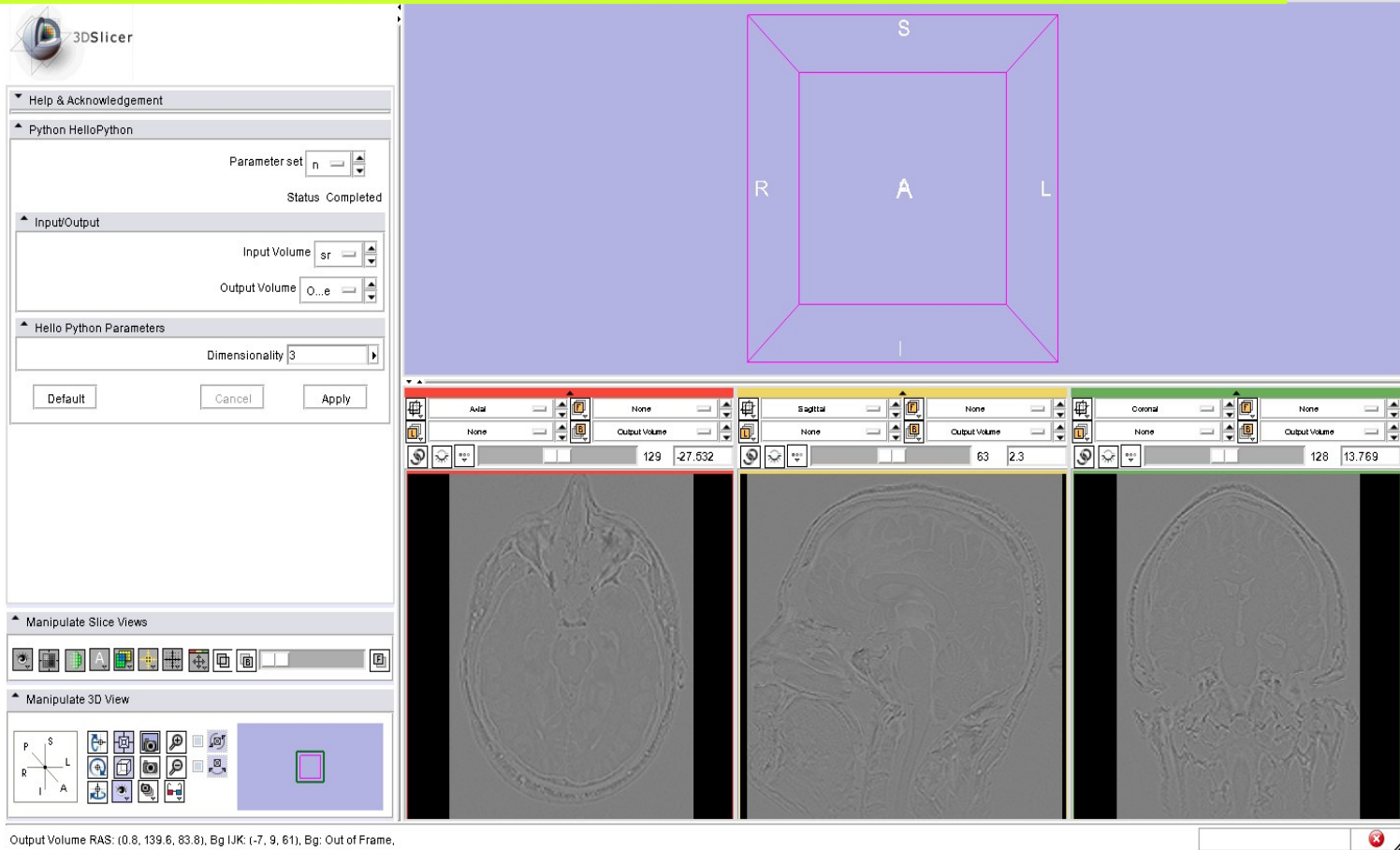


Select the **Input Volume** *spgr*, select the **Output Volume** *Create New Volume*, and click on **Apply**

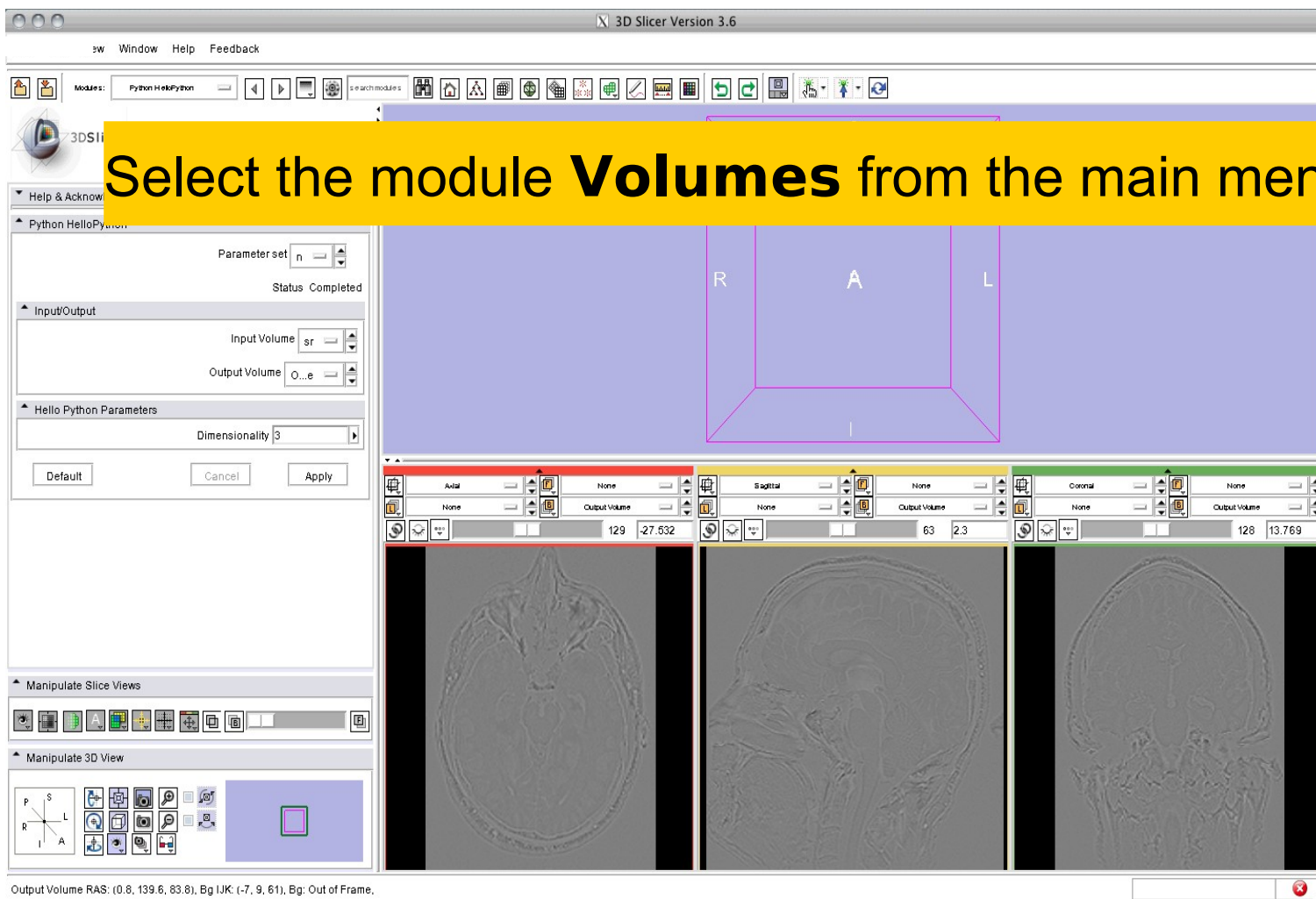
spgr RAS: (0.8, 183.3, 97.5), Bg IJK: (-63, -6, 61), Bg: Out of Frame.

Running the Laplace Operator

Slicer displays the Laplacian of the spgr image



Laplacian of the image



Laplacian of the image

Set the **Active Volume** to *Output Volume* and adjust the **Window/Level** parameters

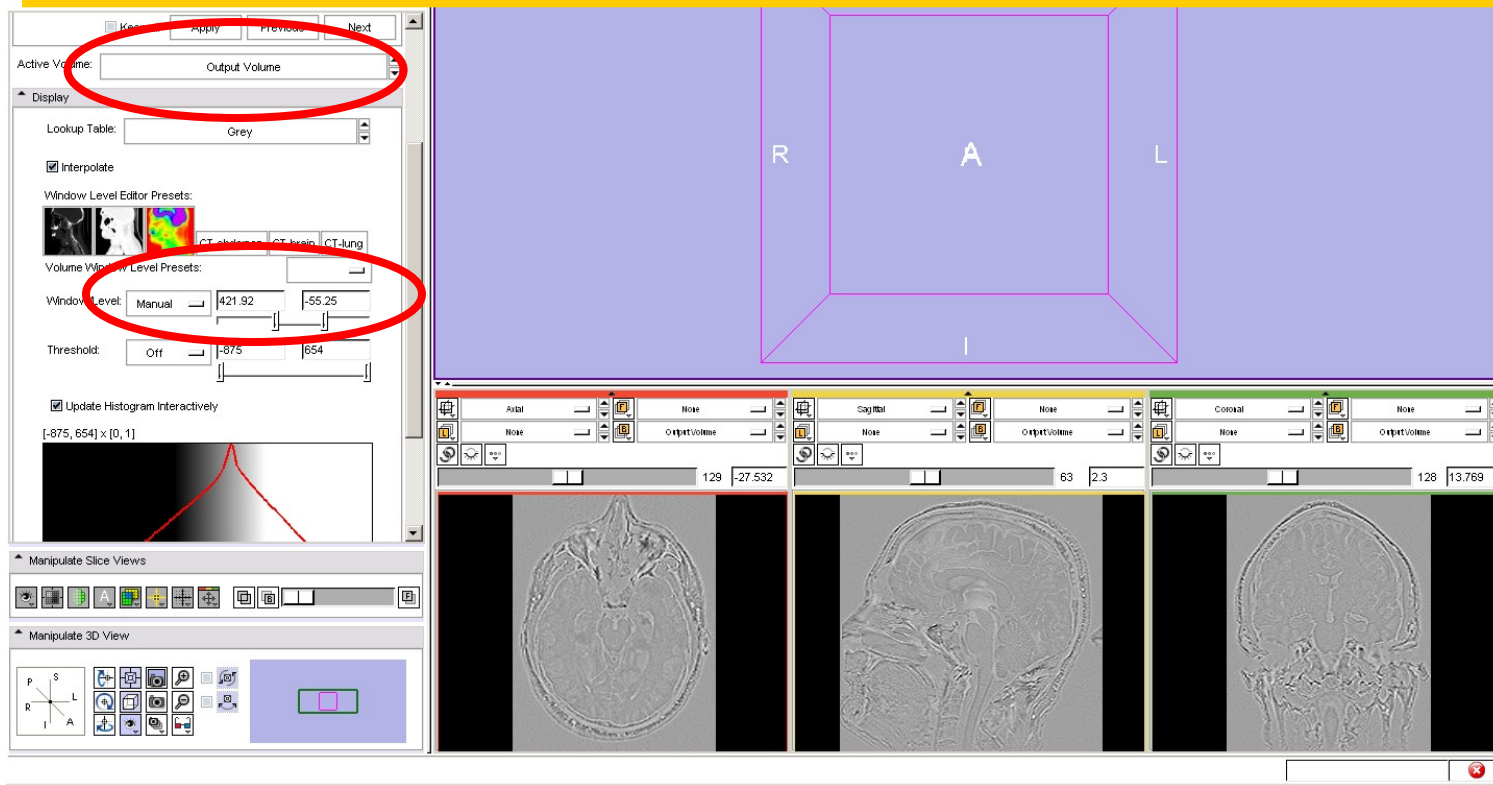
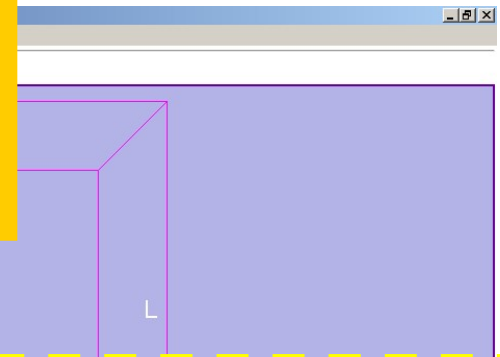


Image Sharpening

Run the following code in the Python console to subtract the Laplacian of the image to the original image



```
import Slicer
volume1 = Slicer.slicer.MRMLScene.GetNodeByID("vtkMRMLScalarVolumeNode1")
volume2 = Slicer.slicer.MRMLScene.GetNodeByID("vtkMRMLScalarVolumeNode2")
plugin = Slicer.Plugin("Subtract Images")
plugin.Execute(volume1,volume2)
```

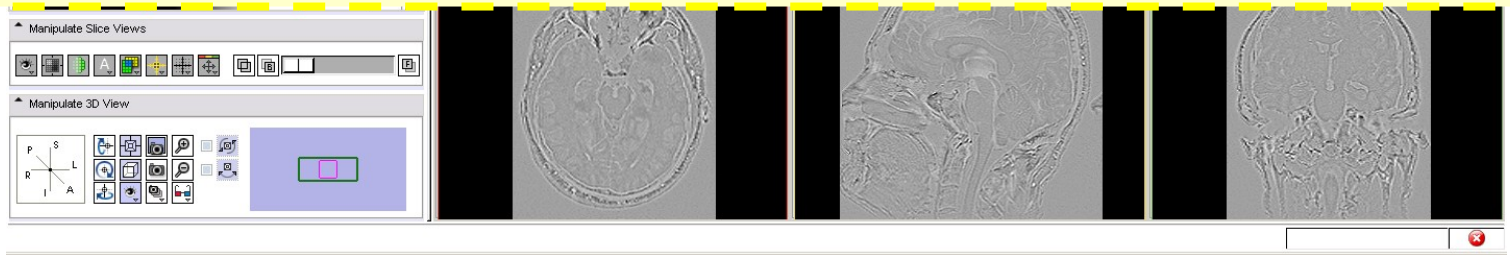
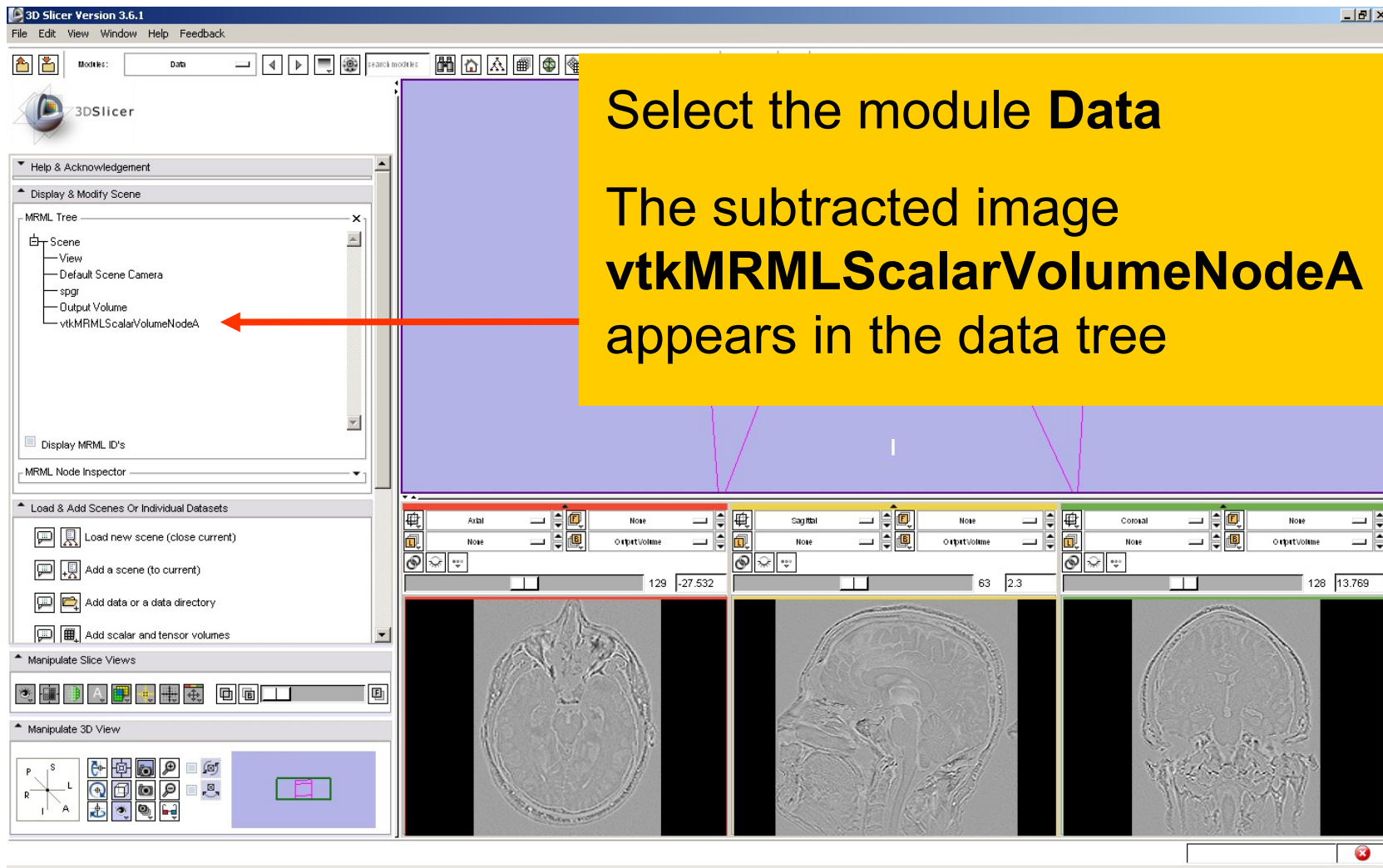
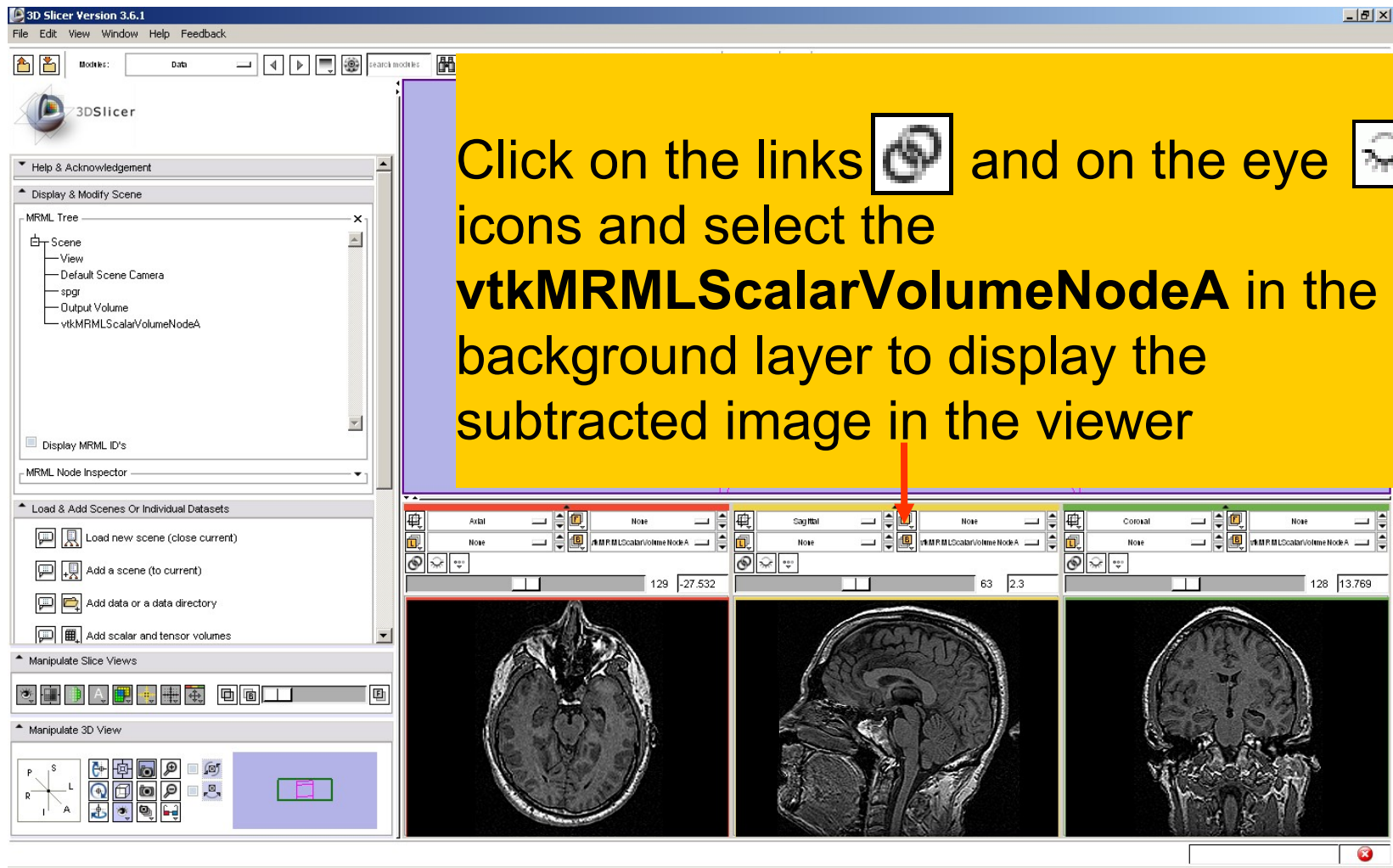


Image Sharpening



The screenshot shows the 3D Slicer 3.6.1 interface. The top menu bar includes File, Edit, View, Window, Help, and Feedback. Below the menu is a toolbar with various icons. The left sidebar contains several panels: 'Help & Acknowledgement', 'Display & Modify Scene', 'Load & Add Scenes Or Individual Datasets', 'Manipulate Slice Views', and 'Manipulate 3D View'. The 'Display & Modify Scene' panel shows the MRML Tree, which is expanded to show the 'Scene' folder containing 'View', 'Default Scene Camera', 'spgr', 'Output Volume', and 'vtkMRMLScalarVolumeNodeA'. A red arrow points to 'vtkMRMLScalarVolumeNodeA'. The 'Load & Add Scenes Or Individual Datasets' panel has buttons for 'Load new scene (close current)', 'Add a scene (to current)', 'Add data or a data directory', and 'Add scalar and tensor volumes'. The 'Manipulate Slice Views' panel has icons for 'Axial', 'Sagittal', and 'Coronal' views, each with 'None' and 'Output Volume' options. The 'Manipulate 3D View' panel has a 3D view icon and a small 3D view window. The main 3D view area shows three orthogonal slices: Axial, Sagittal, and Coronal. The Axial slice shows a brain cross-section with a value of 129 and a range of -27.532. The Sagittal slice shows a brain sagittal section with a value of 63 and a range of 2.3. The Coronal slice shows a brain coronal section with a value of 126 and a range of 13.769. A yellow text box on the right side of the screenshot contains the text: 'Select the module Data' and 'The subtracted image vtkMRMLScalarVolumeNodeA appears in the data tree'.

Image Sharpening



3D Slicer Version 3.6.1

File Edit View Window Help Feedback

3DSlicer

Help & Acknowledgement

Display & Modify Scene

MRML Tree

- Scene
 - View
 - Default Scene Camera
 - spgr
 - Output Volume
 - vtkMRMLScalarVolumeNodeA

Display MRML ID's



MRML Node Inspector

Load & Add Scenes Or Individual Datasets

- Load new scene (close current)
- Add a scene (to current)
- Add data or a data directory
- Add scalar and tensor volumes

Manipulate Slice Views

Manipulate 3D View

Click on the links  and on the eye  icons and select the **vtkMRMLScalarVolumeNodeA** in the background layer to display the subtracted image in the viewer

Abial None 129 -27.532

Sagittal None 63 2.3

Coronal None 128 13.769

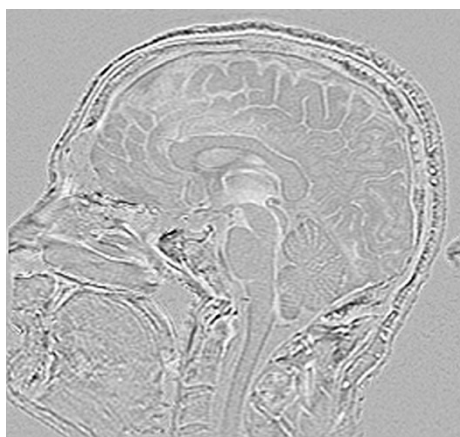
None vtkMRMLScalarVolumeNodeA

Image Sharpening

Original



Laplacian



Laplacian filtered





Conclusion

- This course demonstrates how to integrate an external program in Python within Slicer3
- The **Execution Model** of Slicer3 provides a simple mechanism for incorporating command line programs as Slicer modules in Python.





Acknowledgments



National Alliance for Medical Image Computing
NIH U54EB005149



Neuroimage Analysis Center
NIH P41RR013218