



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

Programming into Slicer3

Sonia Pujol, Ph.D.

Surgical Planning Laboratory
Harvard University





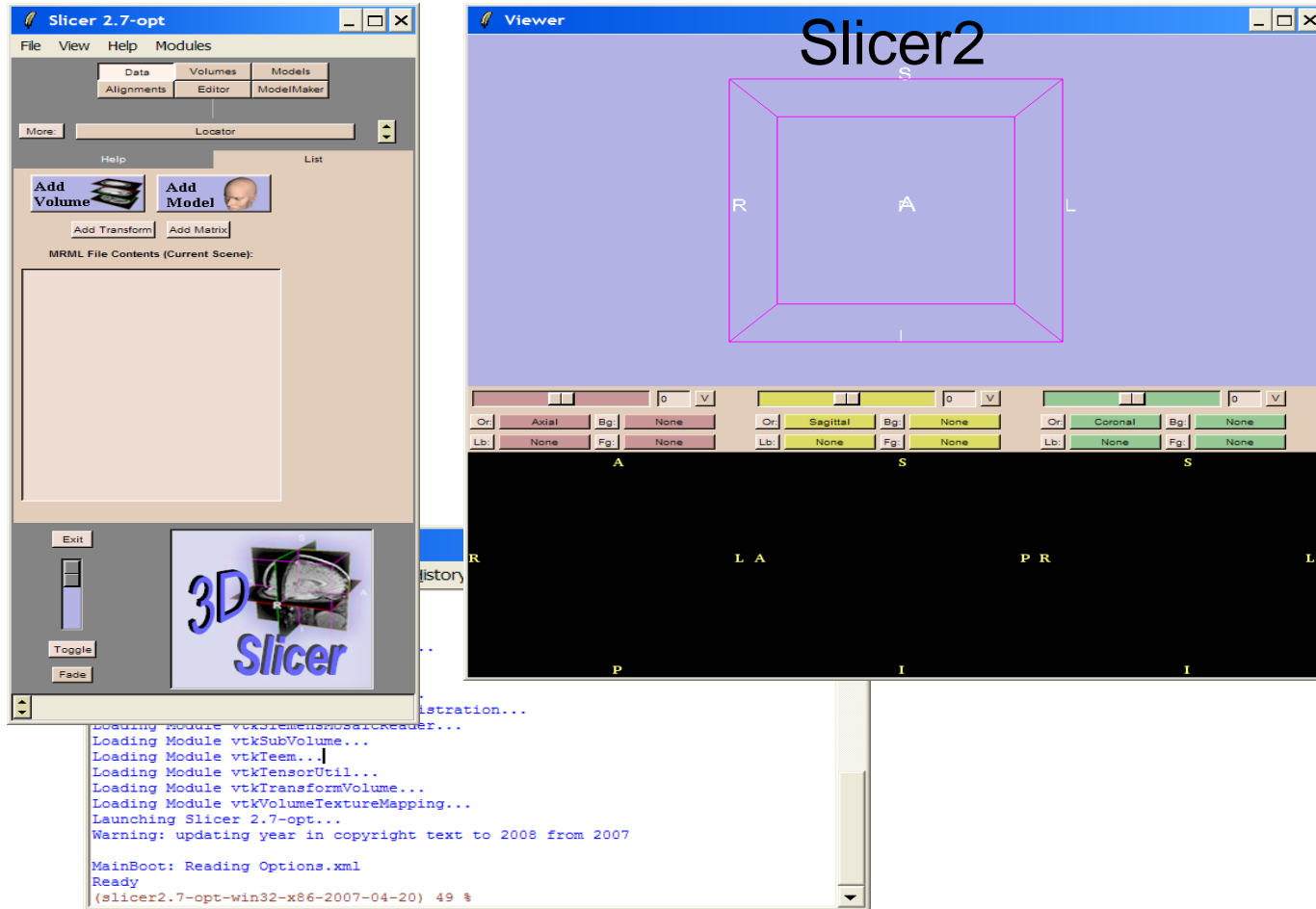
The NA-MIC Kit



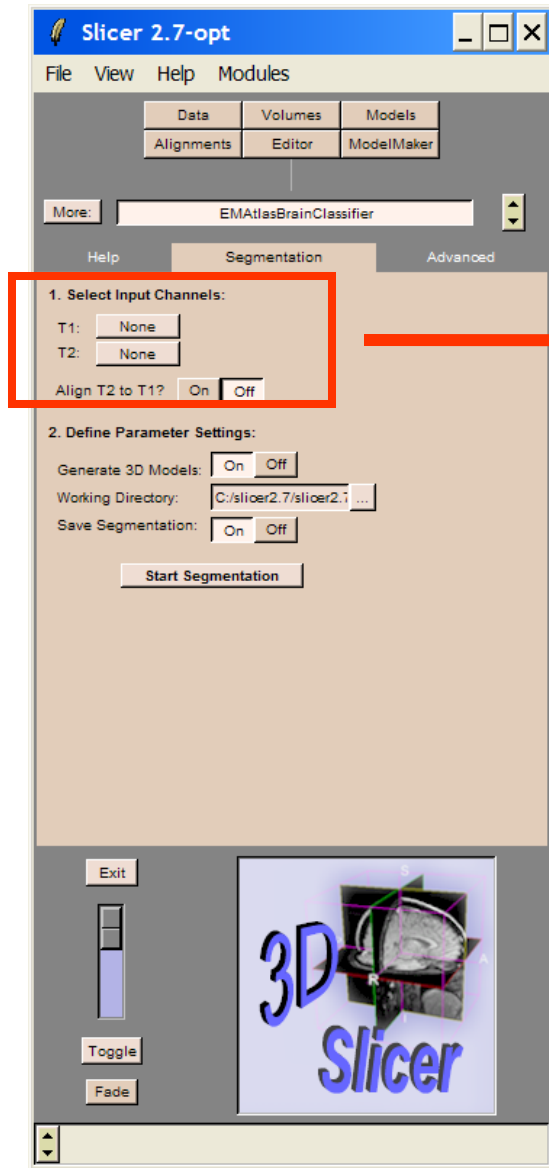


- An **end-user application** for image analysis
- An **open-source environment** for software development
- A software platform that is both **easy to use** for clinical researchers and **easy to extend** for programmers

Before Slicer3



Programming into Slicer2

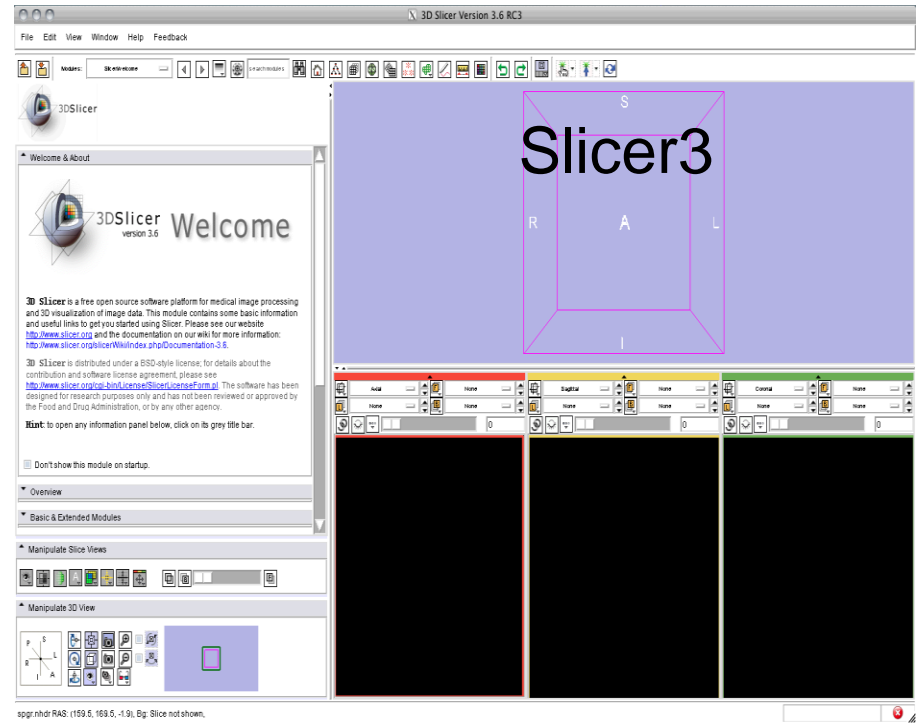
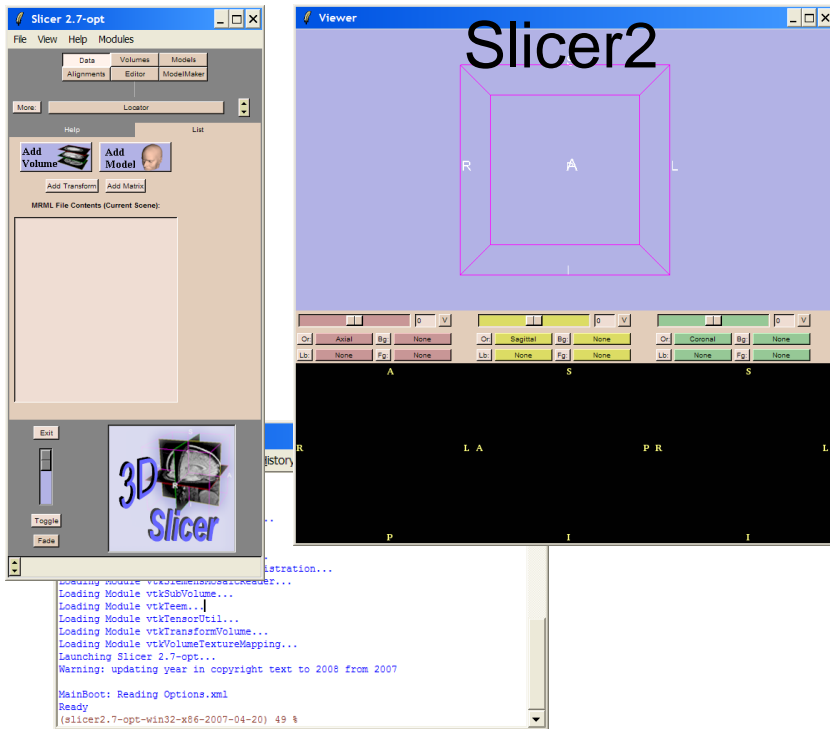


```

#-----
# 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.InInput -bg $Gui(activeWorkspace)
pack $f.InInput -side top -padx 0 -pady 0 -anchor w
foreach frame "Left Right" {
    frame $f.InInput.f$frame -bg $Gui(activeWorkspace)
    pack $f.InInput.f$frame -side left -padx 0 -pady $Gui(pad) }
foreach LABEL "T1 T2" Input "SPGR T2W" {
    DevAddLabel $f.InInput.fLeft.I$Input " ${LABEL}:"
    pack $f.InInput.fLeft.I$Input -side top -padx $Gui(pad) -pady 1 -anchor w
    set menubutton $f.InInput.fRight.m${Input}Select
    set menu $f.InInput.fRight.m${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 }
  
```

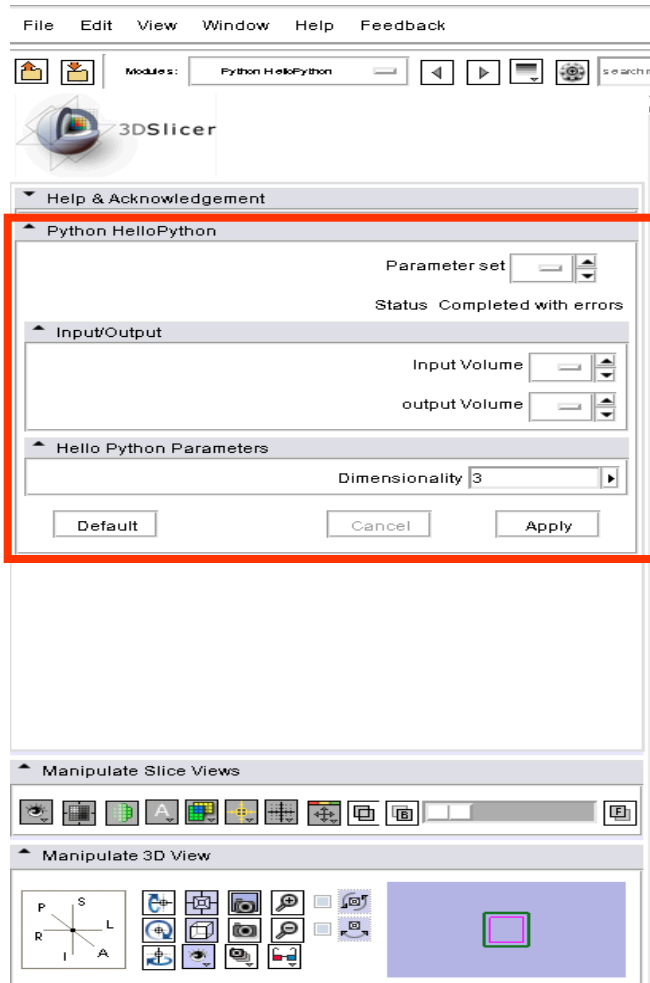


From Slicer2 to Slicer3





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>
```



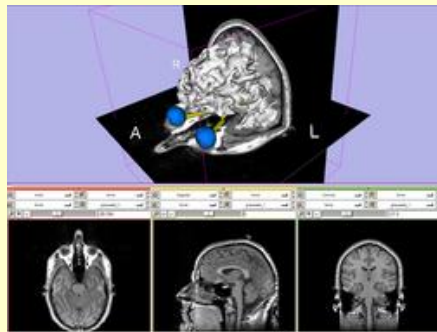
Slicer3 Execution Model

- 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

- 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>

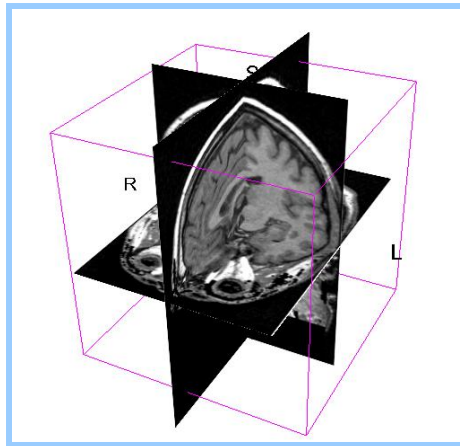
- HelloPython.zip

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

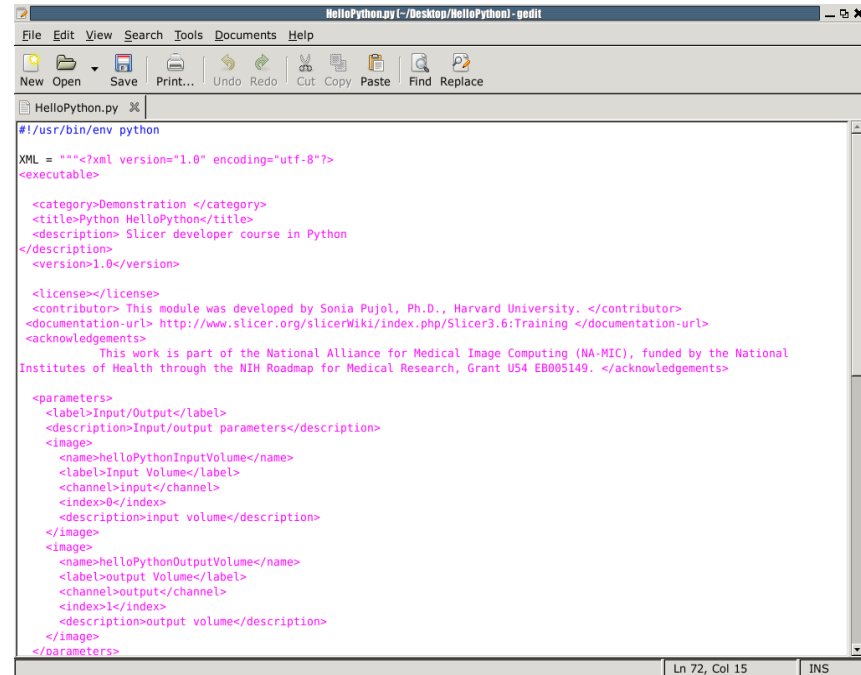
Disclaimer

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.

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>

```

HelloPython.py

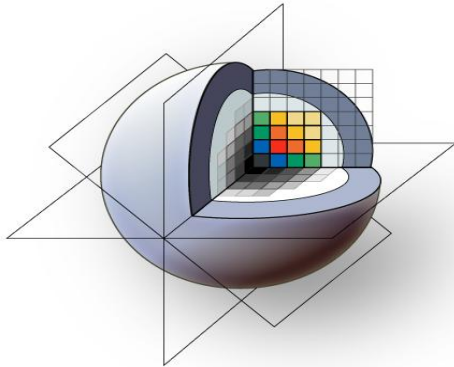


Overview

- 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

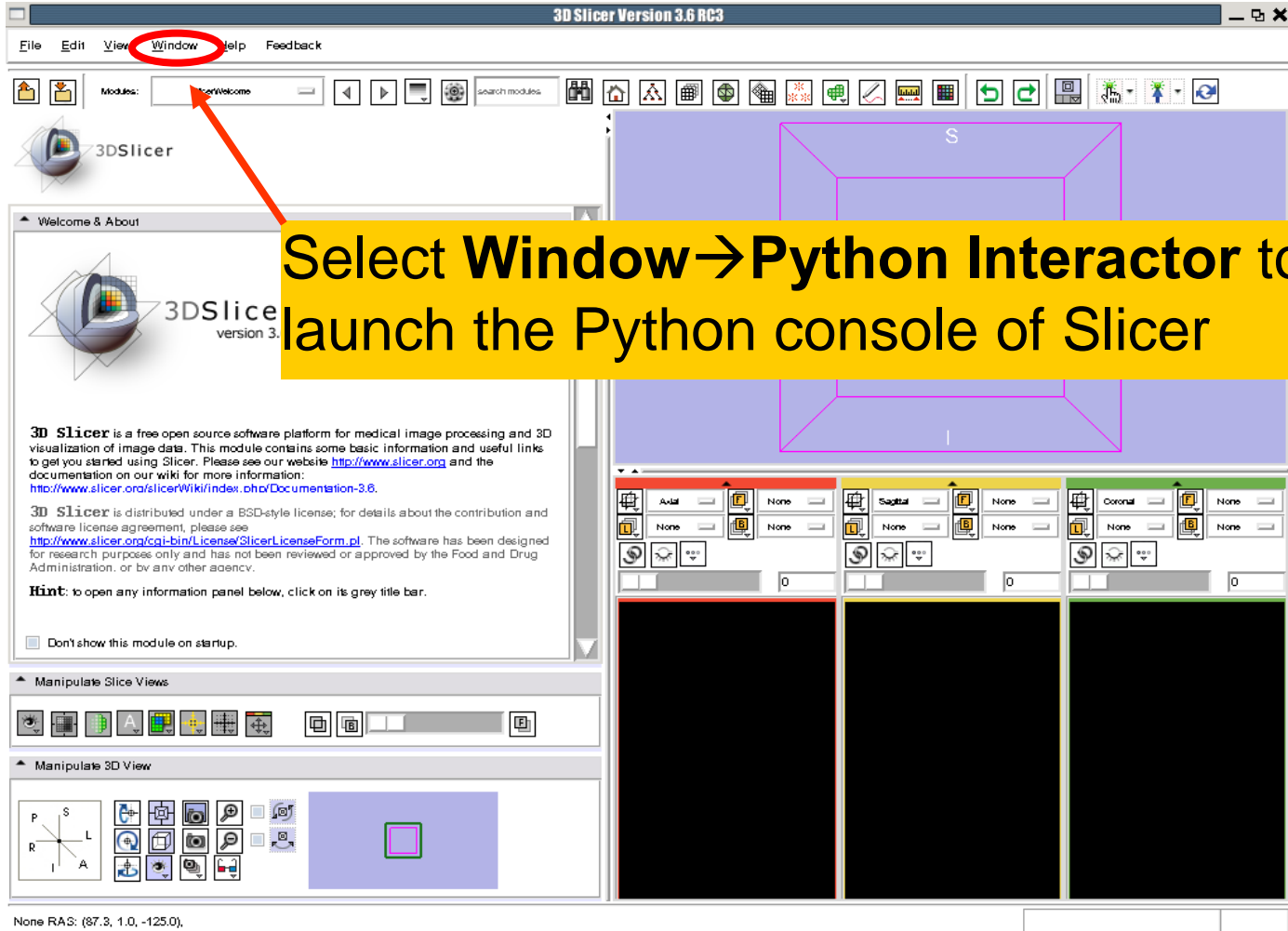


3DSlicer

Part A: Integrating HelloPython into Slicer3

```
>HelloPython.py [C:/Desktop/HelloPython]-gedit
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace
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 US4 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>
```

Python Console



Select Window → Python Interactor to launch the Python console of Slicer

3D Slicer Version 3.6 RC3

File Edit View **Window** Help Feedback

Modules: Python Welcome

Welcome & About

3DSlice version 3.

3D Slicer is a free open source software platform for medical image processing and 3D visualization of image data. This module contains some basic information and useful links to get you started using Slicer. Please see our website <http://www.slicer.org> and the documentation on our wiki for more information: <http://www.slicer.org/slicerWiki/index.php/Documentation-3.6>.

3D Slicer is distributed under a BSD-style license; for details about the contribution and software license agreement, please see <http://www.slicer.org/cgi-bin/License/SlicerLicenseForm.pl>. The software has been designed for research purposes only and has not been reviewed or approved by the Food and Drug Administration, or by any other agency.

Hint: to open any information panel below, click on its grey title bar.

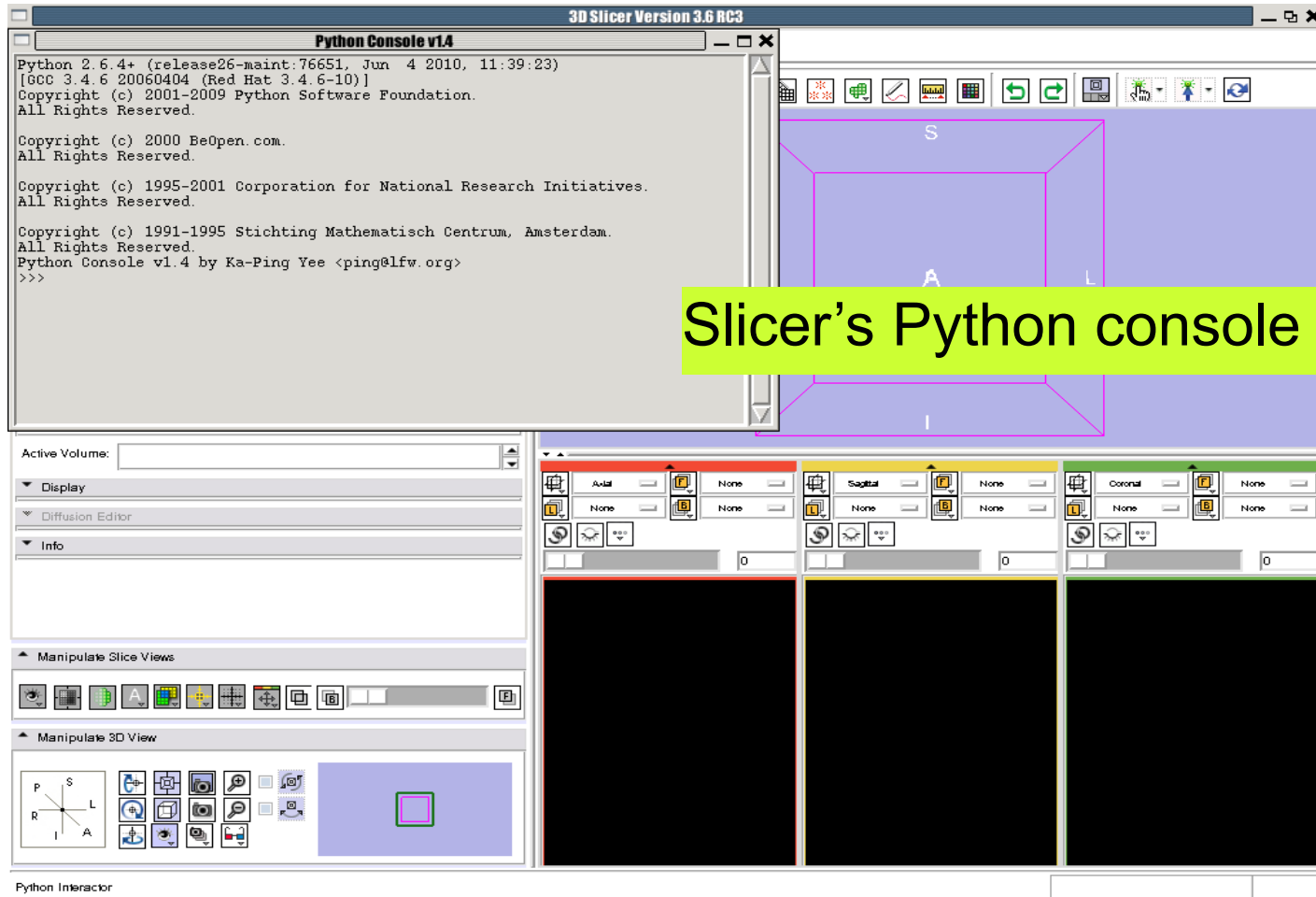
Don't show this module on startup.

Manipulate Slice Views

Manipulate 3D View

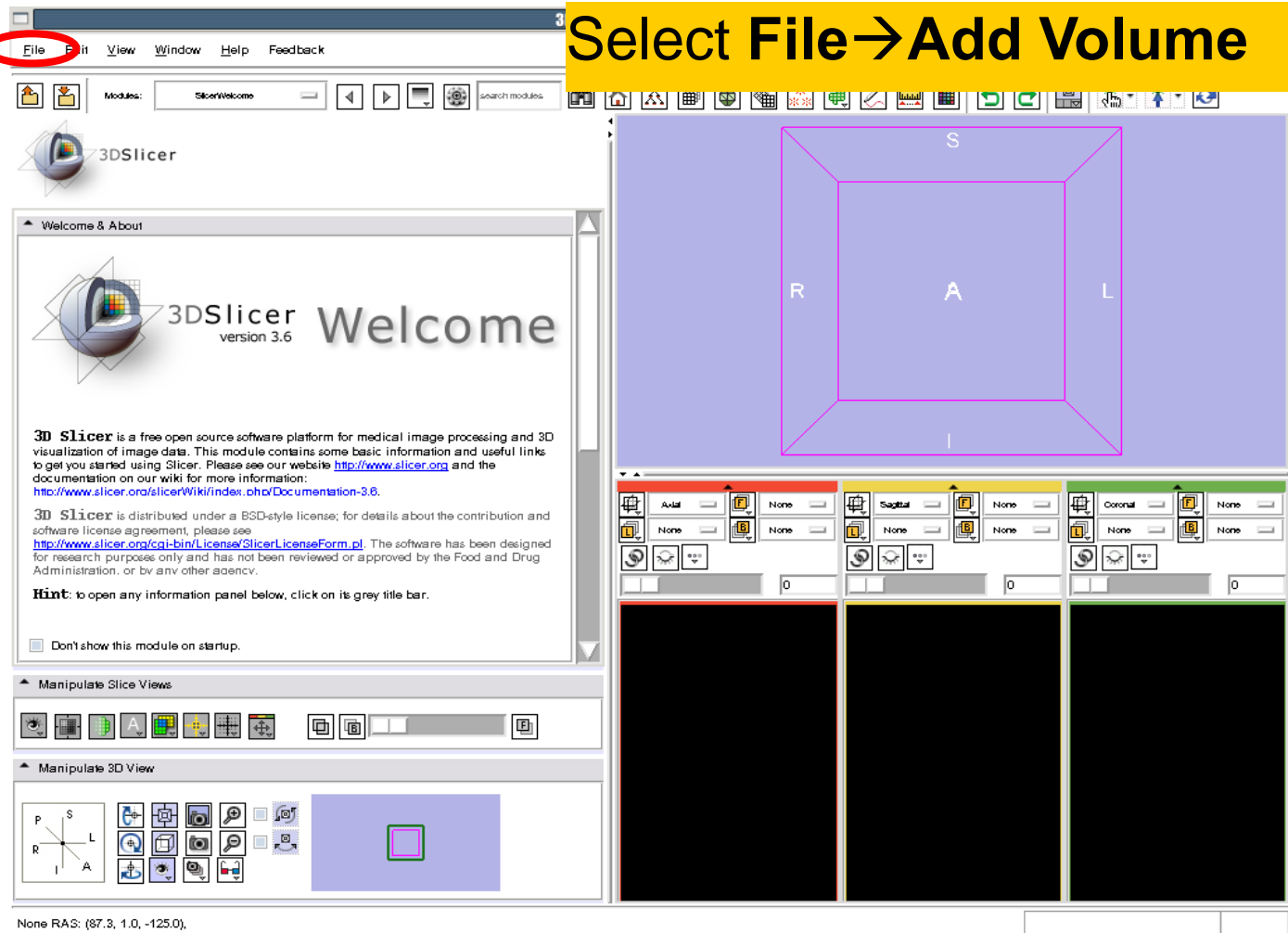
None RAS: (87.3, 1.0, -125.0)

Python Console



Python Console

Select File → Add Volume



File Edit View Window Help Feedback

Modules: SlicerWelcome

3DSlicer

Welcome & About

3DSlicer version 3.6 **Welcome**

3D Slicer is a free open source software platform for medical image processing and 3D visualization of image data. This module contains some basic information and useful links to get you started using Slicer. Please see our website <http://www.slicer.org> and the documentation on our wiki for more information: <http://www.slicer.org/SlicerWiki/index.php/Documentation-3.6>.

3D Slicer is distributed under a BSD-style license; for details about the contribution and software license agreement, please see <http://www.slicer.org/cgi-bin/License/SlicerLicenseForm.pl>. The software has been designed for research purposes only and has not been reviewed or approved by the Food and Drug Administration, or by any other agency.

Hint: to open any information panel below, click on its grey title bar.

Don't show this module on startup.

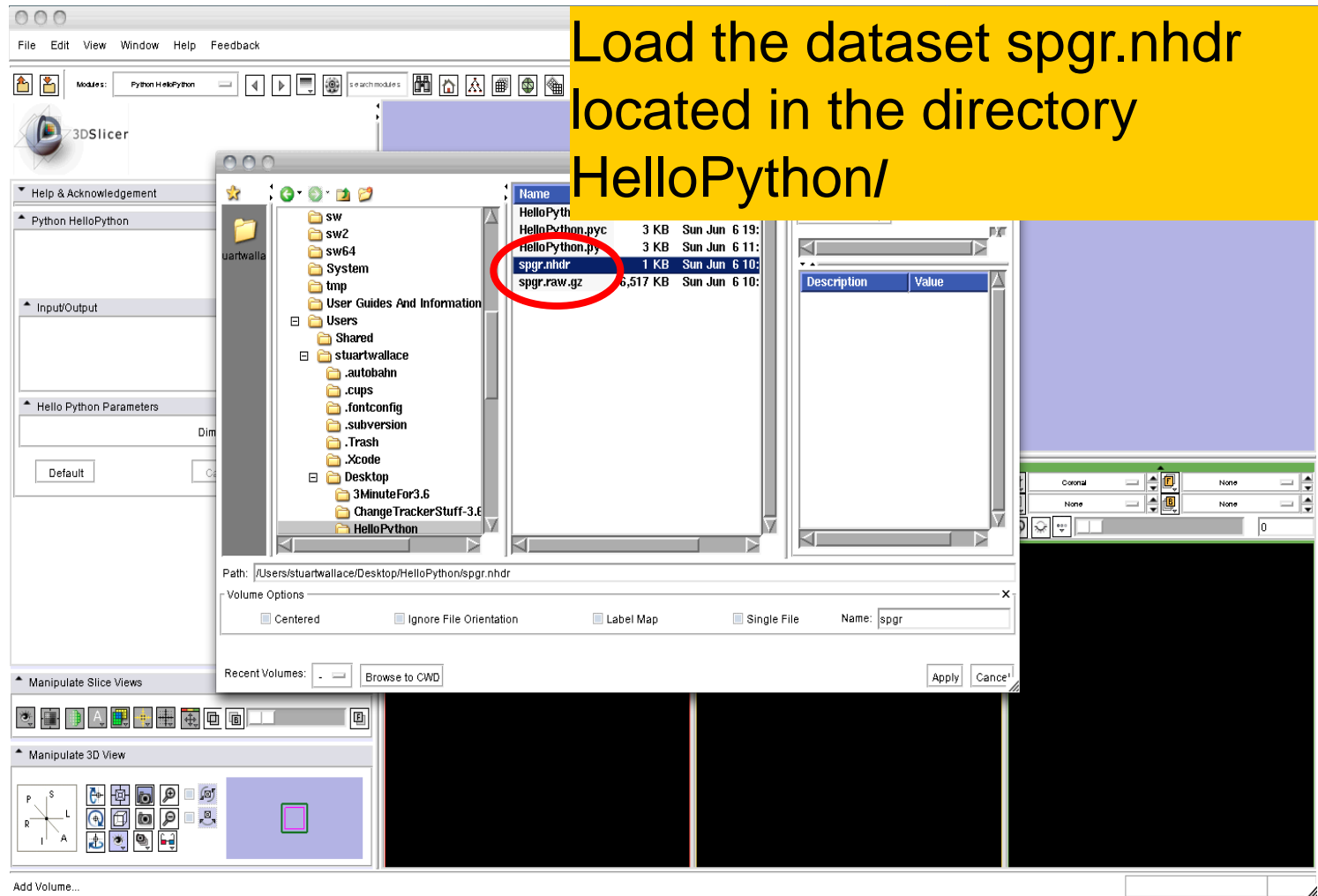
Manipulate Slice Views

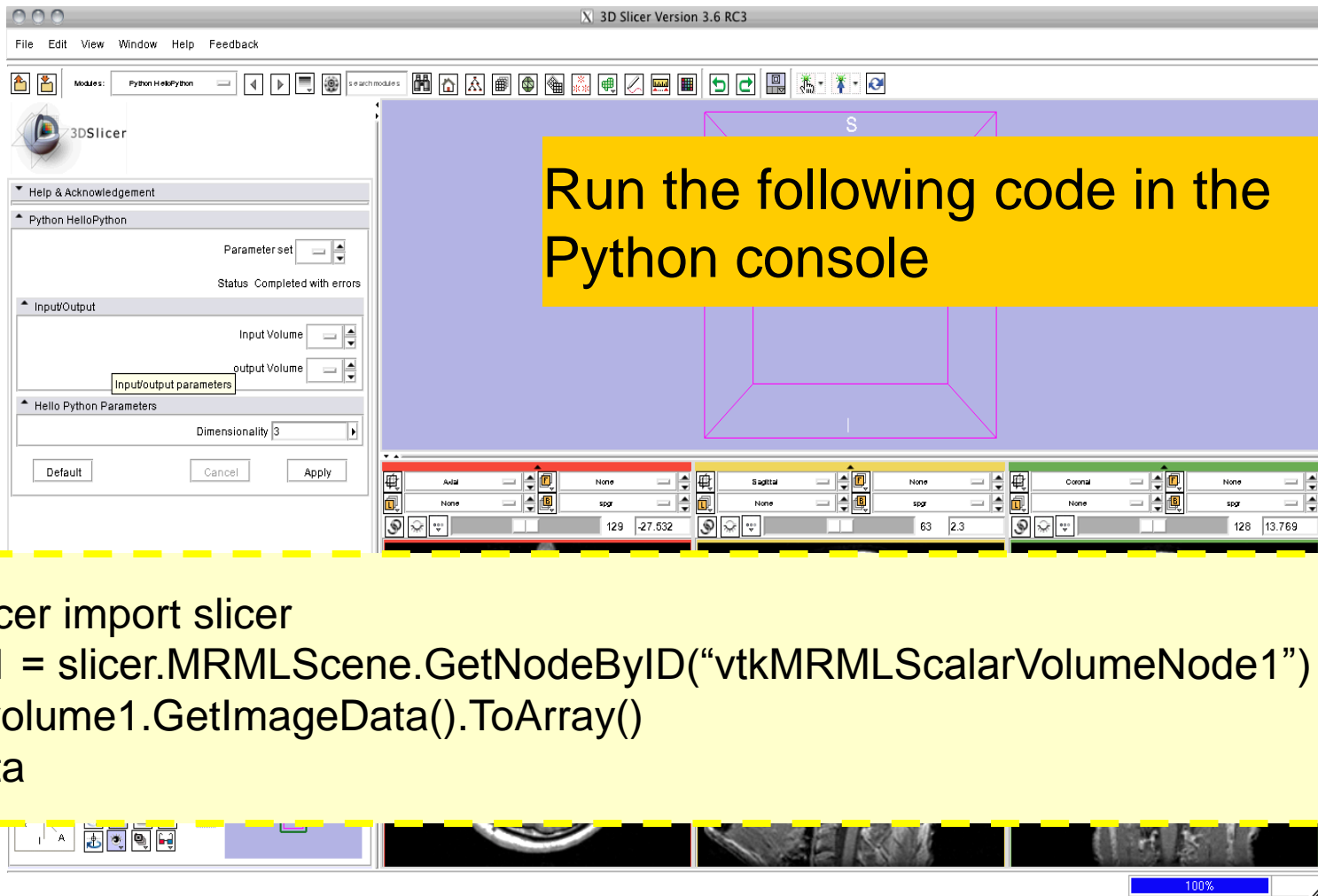
Manipulate 3D View

None RAS: (87.3, 1.0, -125.0)

Python Console

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

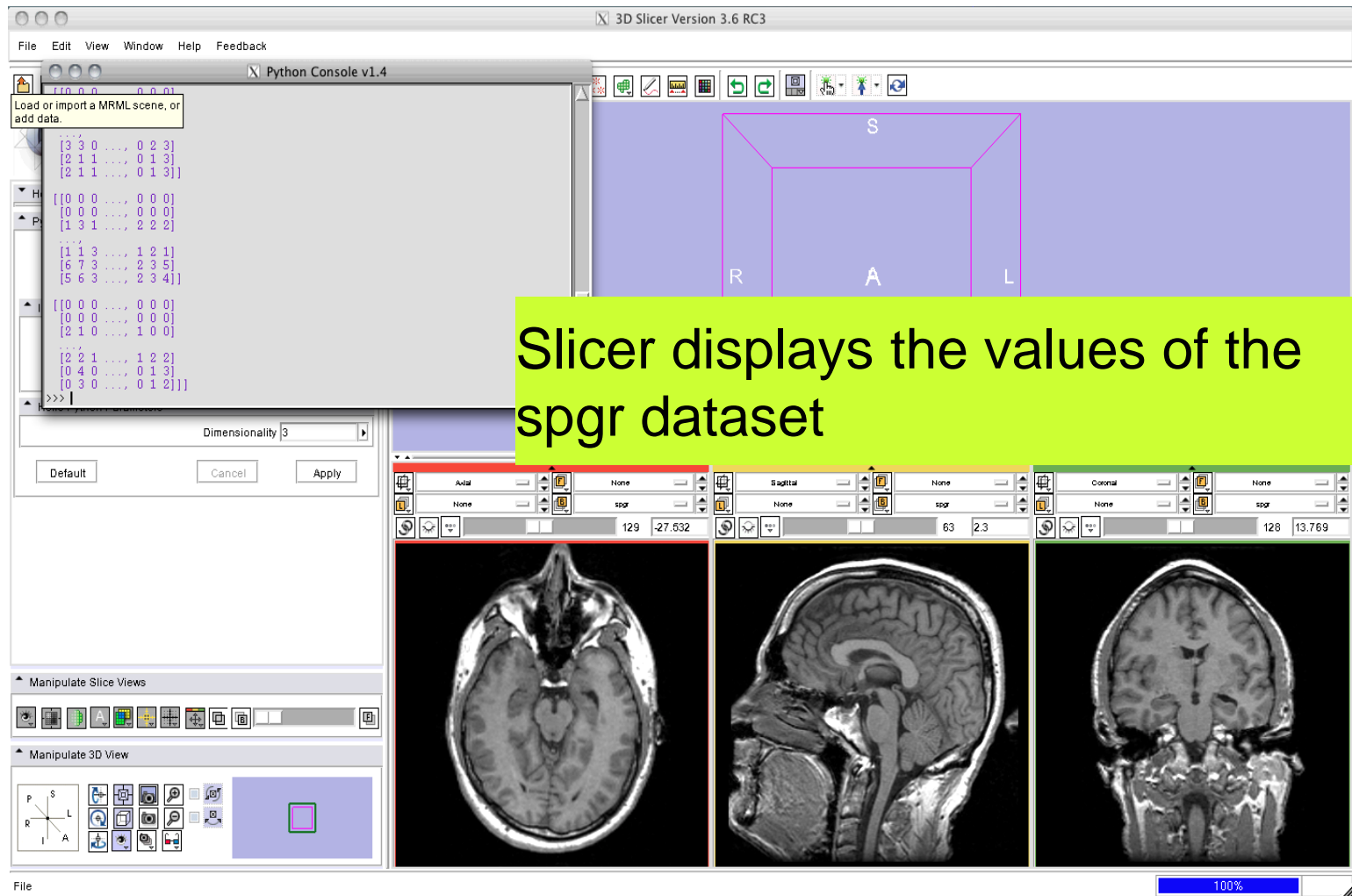




The screenshot shows the 3D Slicer 3.6 RC3 interface. The main window displays a 3D view of a volume with a yellow box overlaid containing the text "Run the following code in the Python console". The left sidebar shows the "Python HelloPython" module with a "Parameter set" dropdown and a "Status" field indicating "Completed with errors". Below this are "Input/Output" parameters for "Input Volume" and "Output Volume", and "Hello Python Parameters" with a "Dimensionality" dropdown set to "3". The bottom of the interface shows the "Layout" panel with three views: Axial, Sagittal, and Coronal. The Axial view shows a slice with coordinates 129 and -27.532. The Sagittal view shows a slice with coordinates 63 and 2.3. The Coronal view shows a slice with coordinates 128 and 13.769. The bottom right corner of the interface shows a "100%" zoom level.


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

Python Console



The screenshot displays the 3D Slicer software interface. At the top, the title bar reads "3D Slicer Version 3.6 RC3". Below it is a menu bar with "File", "Edit", "View", "Window", "Help", and "Feedback". A "Python Console v1.4" window is open, showing a list of matrices. A yellow text box is overlaid on the console area with the text "Slicer displays the values of the spgr dataset". The main 3D view shows a brain MRI scan with a purple wireframe box overlaid, labeled with "S" (Superior), "R" (Right), "A" (Anterior), and "L" (Left). Below the 3D view are three slice views: Axial, Sagittal, and Coronal. The Axial view shows a brain slice at 129, -27.532. The Sagittal view shows a brain slice at 63, 2.3. The Coronal view shows a brain slice at 128, 13.769. The interface also includes a "Manipulate Slice Views" panel and a "Manipulate 3D View" panel.

HelloPython.py



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>
    <name>helloPythonOutputVolume</name>
    <label>output Volume</label>
    <channel>output</channel>
    <index>1</index>
    <description>output volume</description>
    </image>
  </parameters>
  <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.HIFMLScene

    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

Click on the **View** → **Application Settings** in the main menu

The screenshot shows the 3D Slicer 3.6 RC3 interface. The main window displays three orthogonal MRI slices: Axial, Sagittal, and Coronal. The Application Settings dialog is open on the left, showing the configuration for the Python HelloPython extension. The dialog has sections for Input/Output and Hello Python Parameters. The main window shows the following data for the slices:

View	Background	Spacing	Other Data
Axial	None	spg	129 -27.532
Sagittal	None	spg	63 2.3
Coronal	None	spg	128 13.769

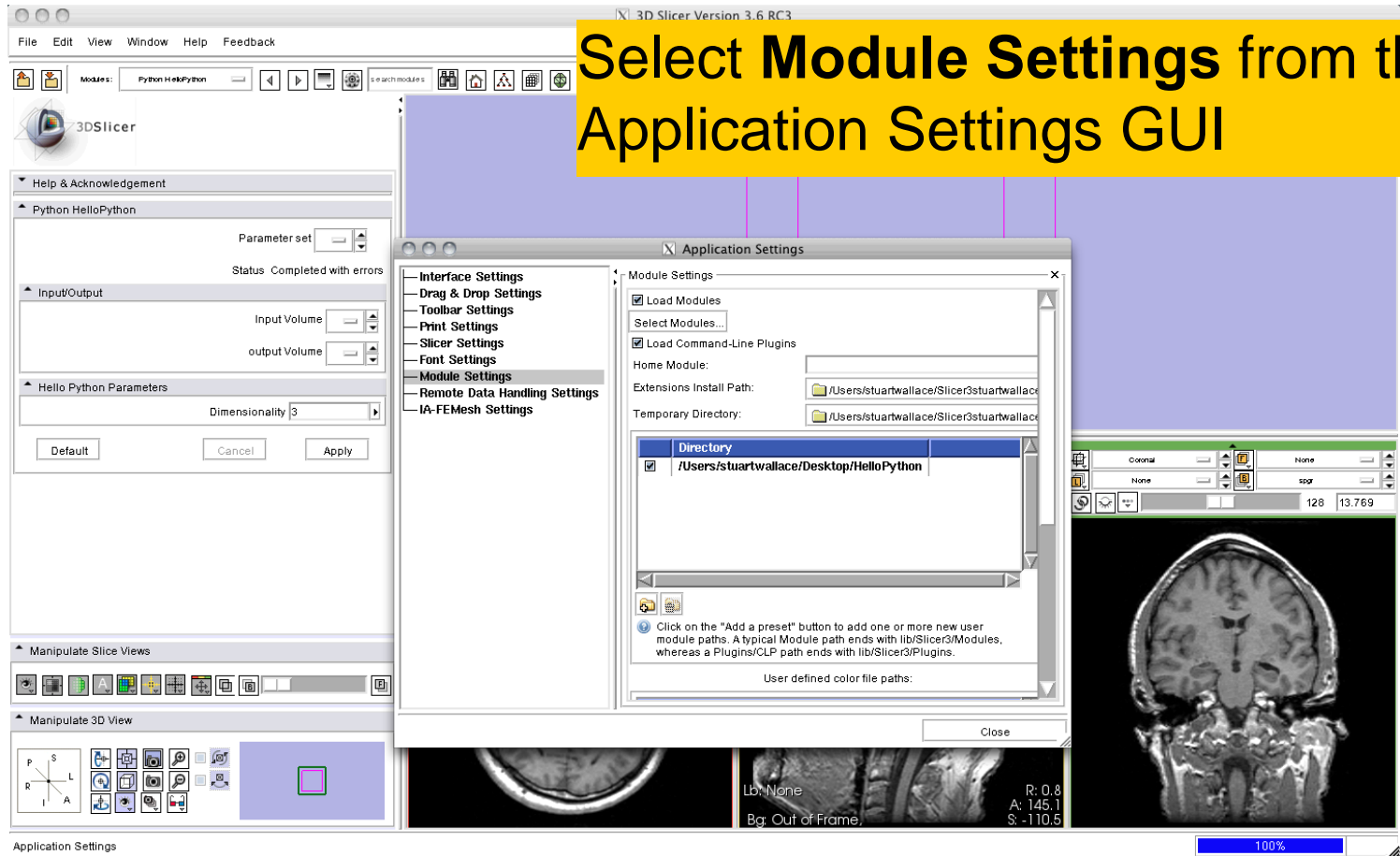
Additional data for the Sagittal slice:

- Bg I: -13
- Bg J: 216
- Bg K: 61
- Sagittal Sp: 1.5mm
- Lb: None
- Bg: Out of Frame
- R: 0.8
- A: 145.1
- S: -110.5

The Application Settings dialog shows the following parameters:

- Parameter set: [None]
- Status: Completed with errors
- Input/Output: Input Volume [None], output Volume [None]
- Hello Python Parameters: Dimensionality [3]
- Buttons: Default, Cancel, Apply

Integrating HelloPython to Slicer3



Select Module Settings from the Application Settings GUI

3D Slicer Version 3.6 RC3

File Edit View Window Help Feedback

Modules: 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

Application Settings

Interface Settings

Drag & Drop Settings

Toolbar Settings

Print Settings

Slicer Settings

Font Settings

Module Settings

Remote Data Handling Settings

IA-FEMesh Settings

Module Settings

Load Modules

Select Modules...

Load Command-Line Plugins

Home Module:

Extensions Install Path: /Users/stuartwallace/Slicer3stuartwallace

Temporary Directory: /Users/stuartwallace/Slicer3stuartwallace

Directory	
<input checked="" type="checkbox"/>	/Users/stuartwallace/Desktop/HelloPython

Click on the "Add a preset" button to add one or more new user module paths. A typical Module path ends with lib/Slicer3/Modules, whereas a Plugins/CLP path ends with lib/Slicer3/Plugins.

User defined color file paths:

Close

Coronal None

None spgr

128 13.769

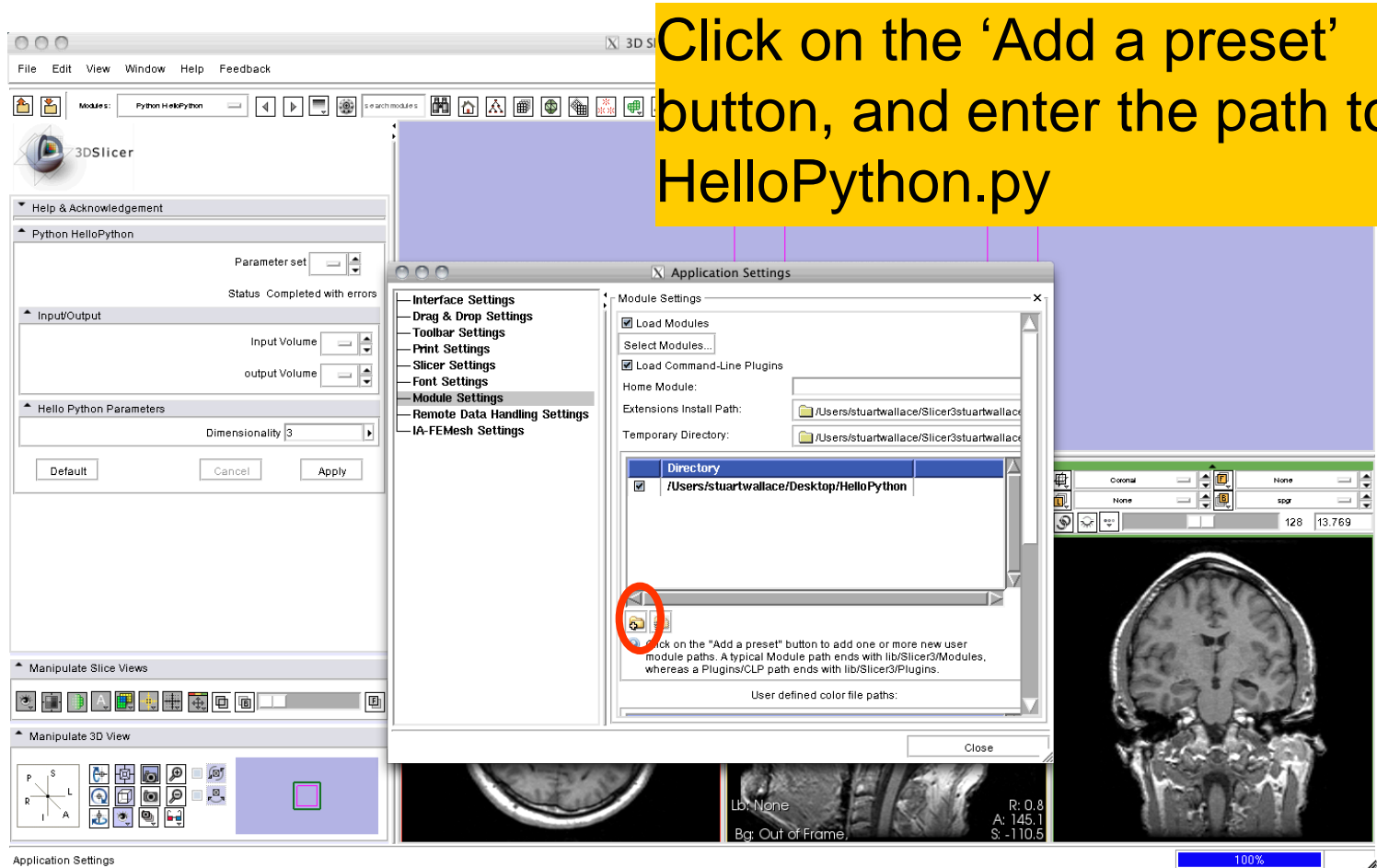
R: 0.8
A: 145.1
S: -110.5

Lb: None
Bg: Out of Frame

100%

Integrating HelloPython to Slicer3

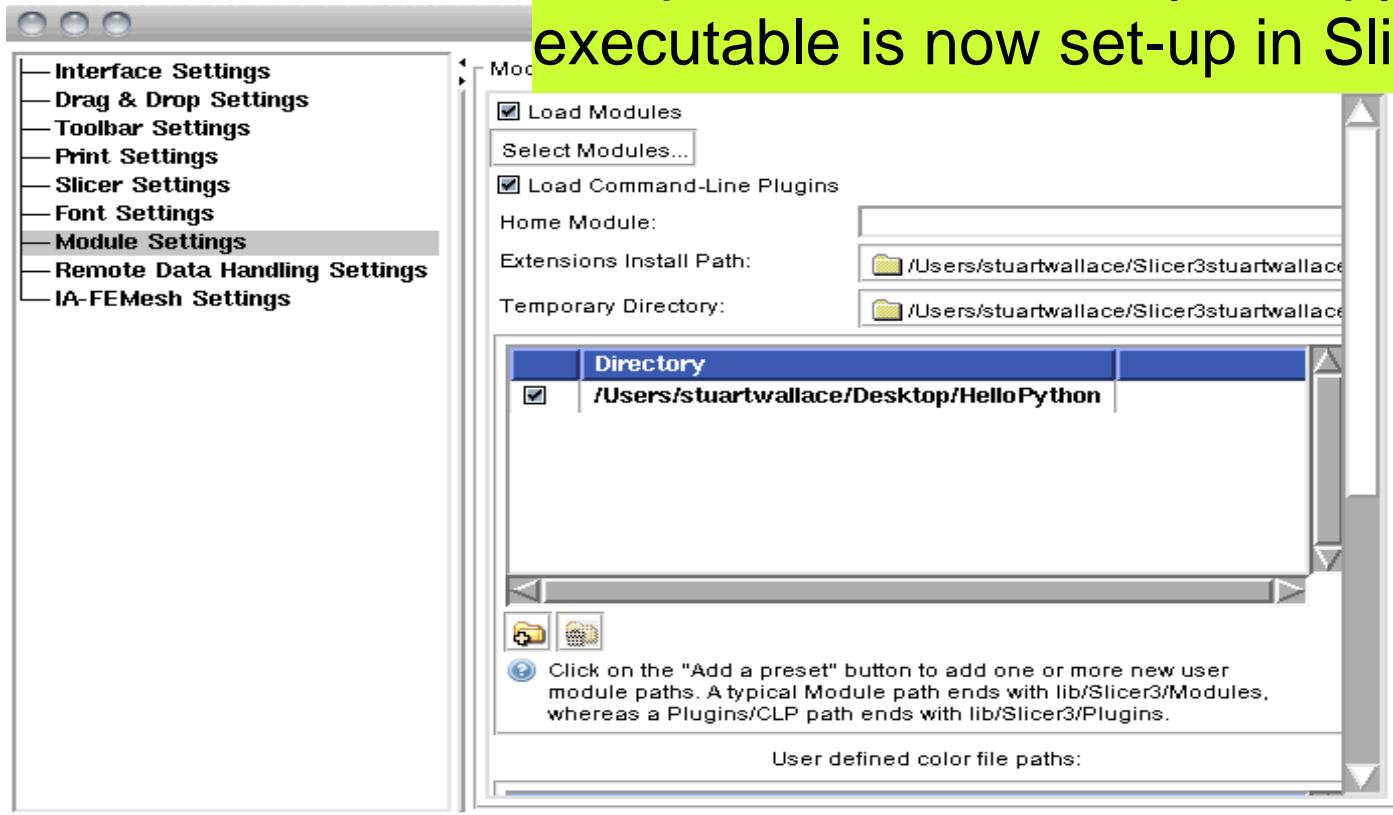
Click on the 'Add a preset' button, and enter the path to HelloPython.py



The screenshot shows the 3DSlicer application window with the 'Application Settings' dialog box open. The 'Module Settings' tab is selected, and the 'Add a preset' button is circled in red. The dialog box displays a list of directories, with '/Users/stuartwallace/Desktop/HelloPython' selected. Below the list, there is a text box with instructions: 'Click on the "Add a preset" button to add one or more new user module paths. A typical Module path ends with lib/Slicer3/Modules, whereas a Plugins/CLP path ends with lib/Slicer3/Plugins.' The 'Close' button is visible at the bottom right of the dialog box.

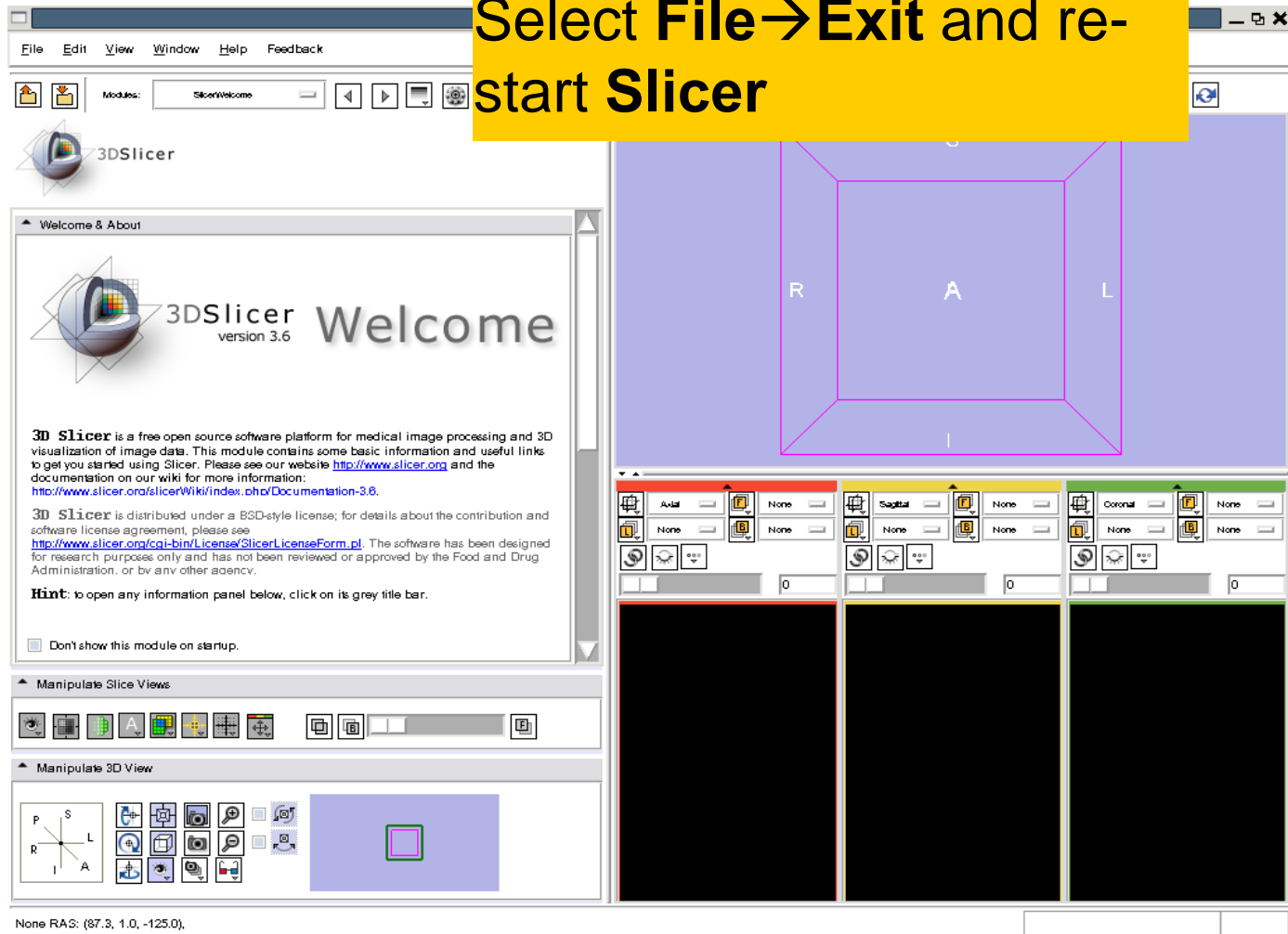
Integrating HelloPython to Slicer3

The path to the HelloPython.py executable is now set-up in Slicer3.



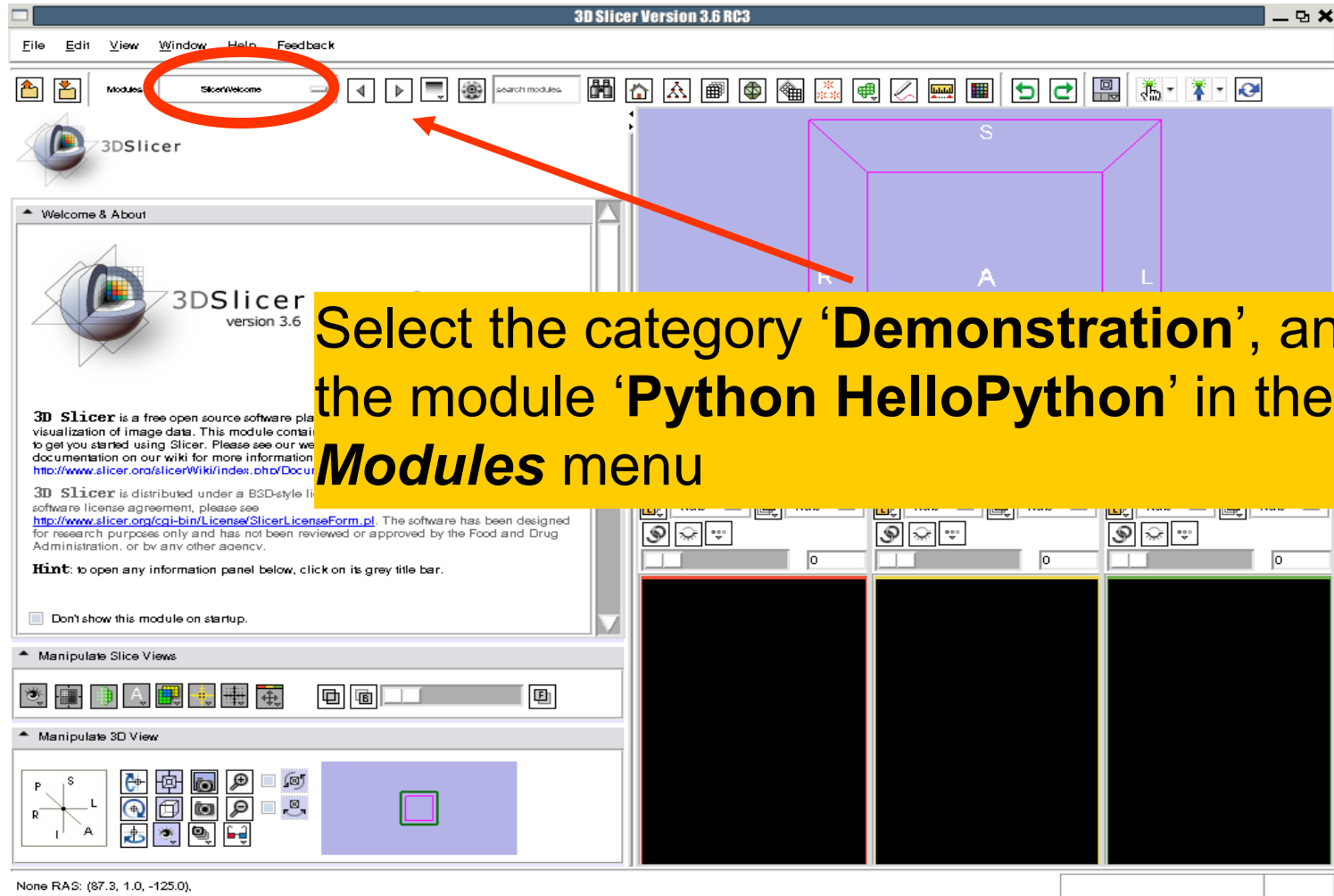
Click on the **Close** to exit the Application Settings window.

Integrating HelloPython to Slicer3

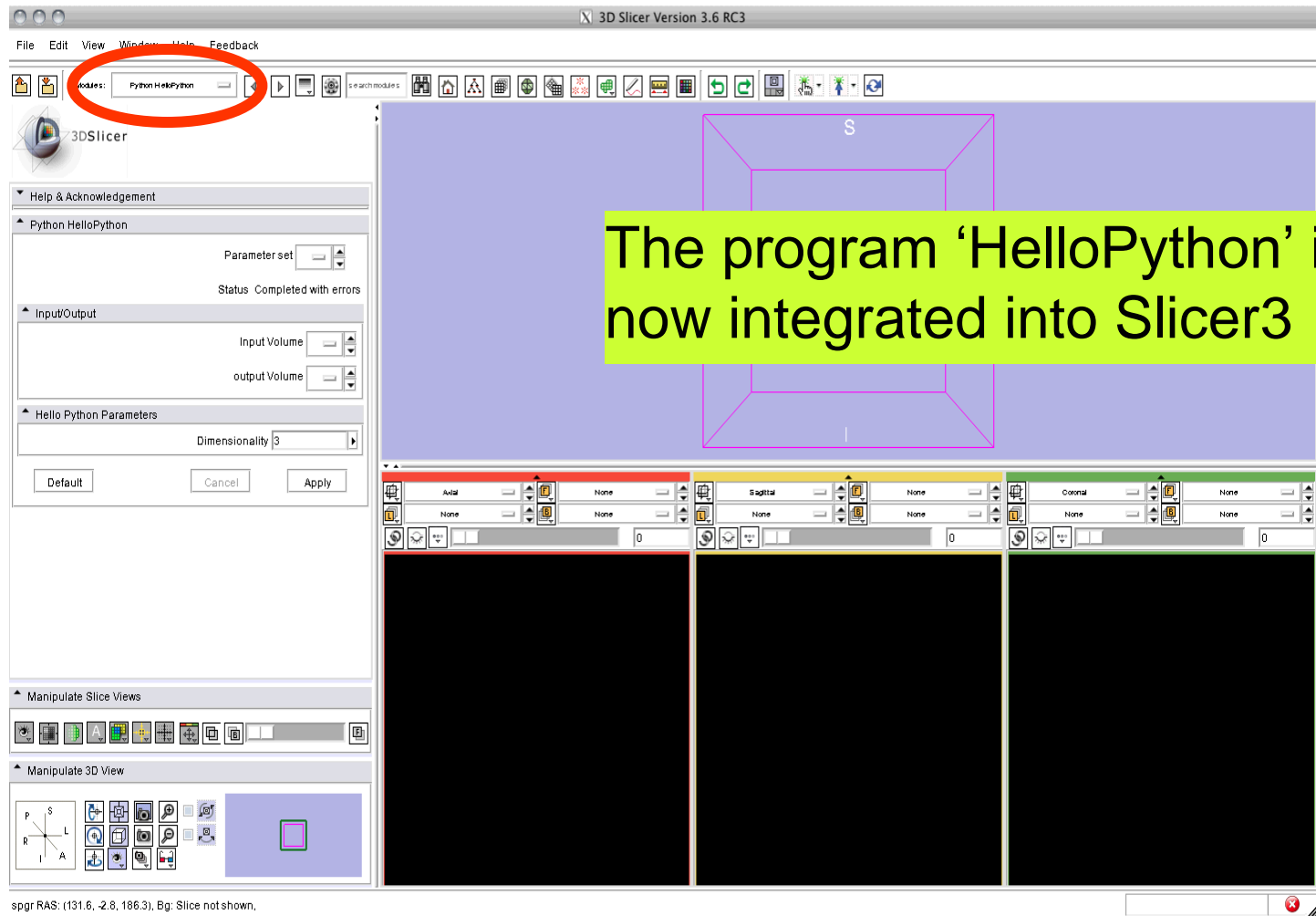


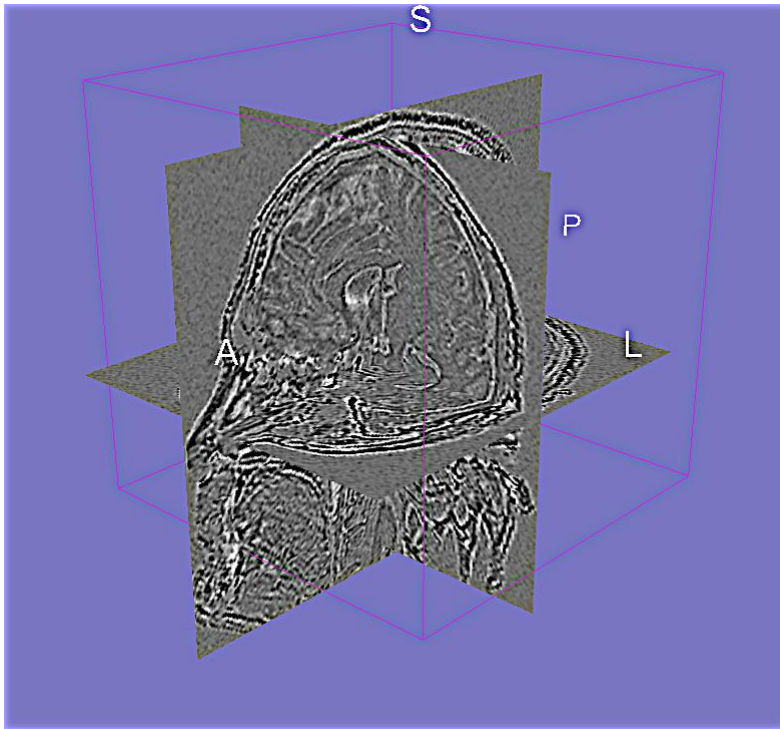


HelloPython module



HelloPython Module





Part B: Implementing the Laplace* Operator

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



Execute Function

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

Add the I/O code

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

```
slicer = Slicer.slicer
```

```
scene = slicer.MRMLScene
```

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

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

```
return
```



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>
```

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

Add a new parameter group for the Laplace operator



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
```

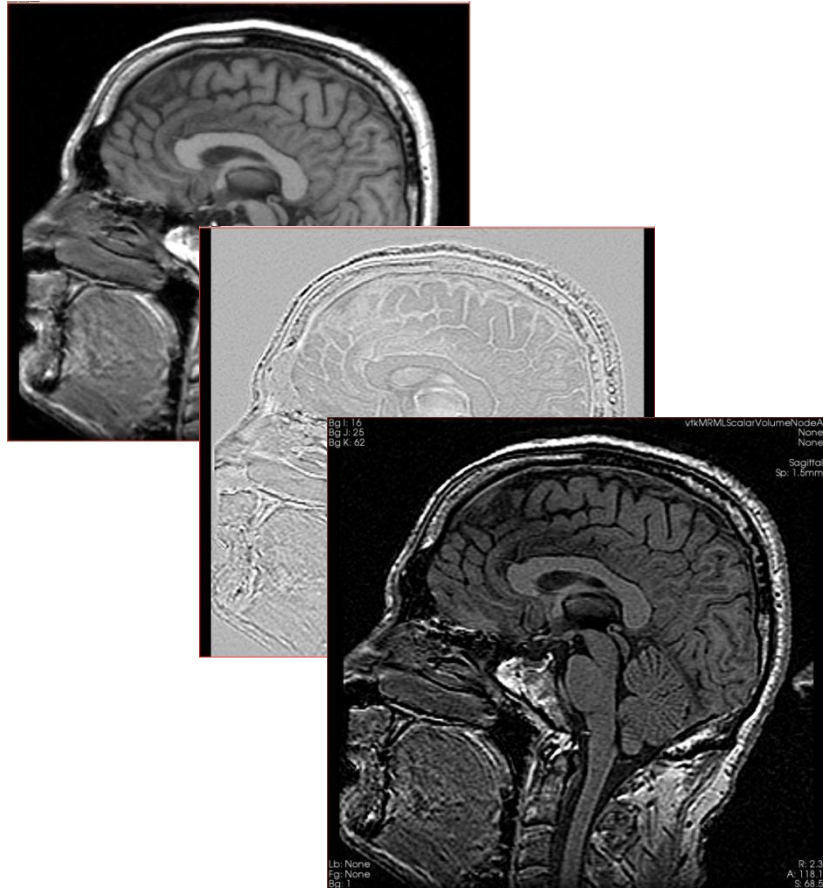
Place back the Laplacian of the image in the RAS reference system.



Integrating HelloPython to Slicer3

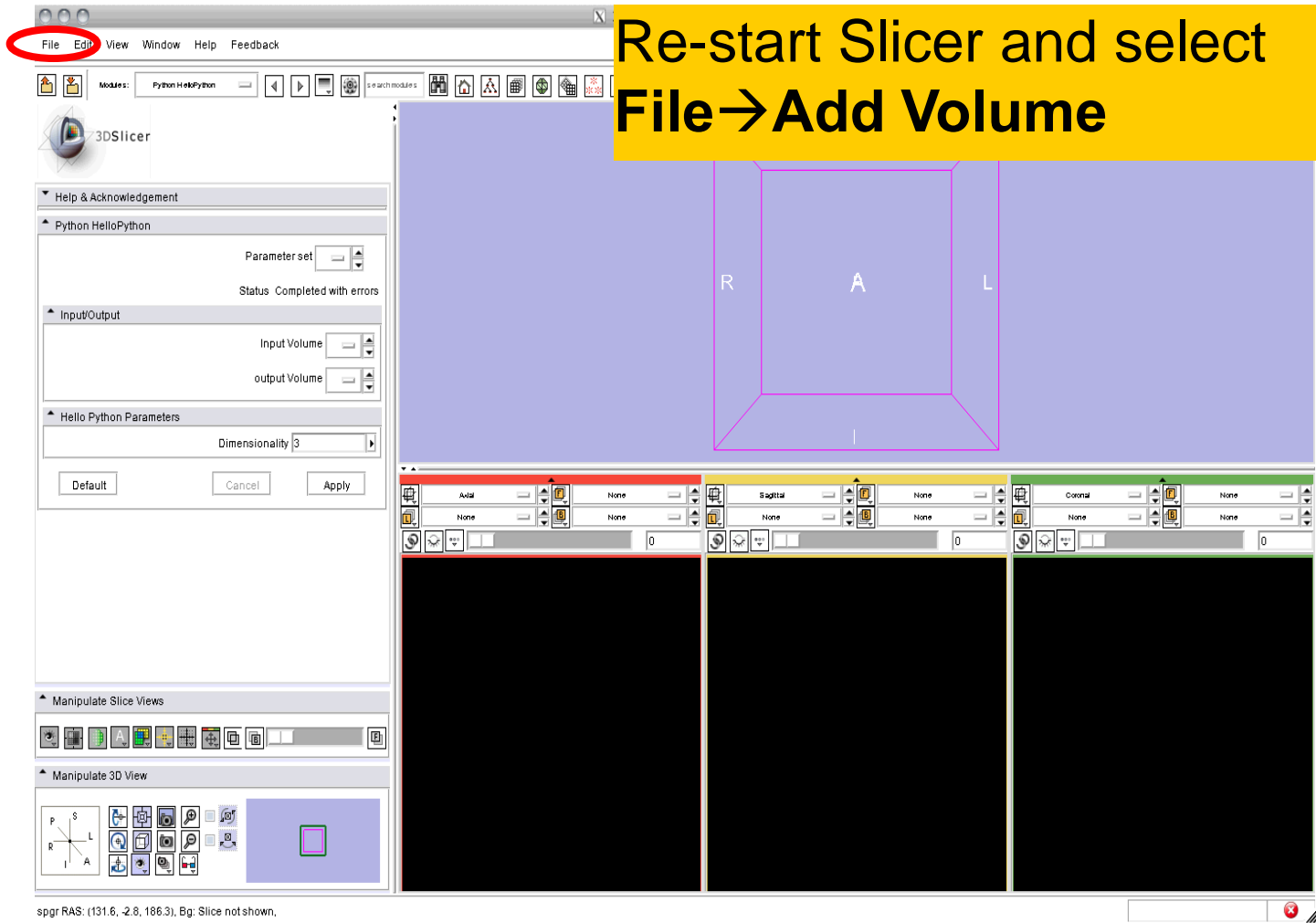
```
def Execute (HelloPythonInputVolume, HelloPythonOutputVolume,
dimensionality=3):
    Slicer = __import__("Slicer")
    slicer = Slicer.slicer
    scene = slicer.MRMLScene
    inputVolume = scene.GetNodeByID(HelloPythonInputVolume)
    outputVolume = scene.GetNodeByID(HelloPythonOutputVolume)
   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

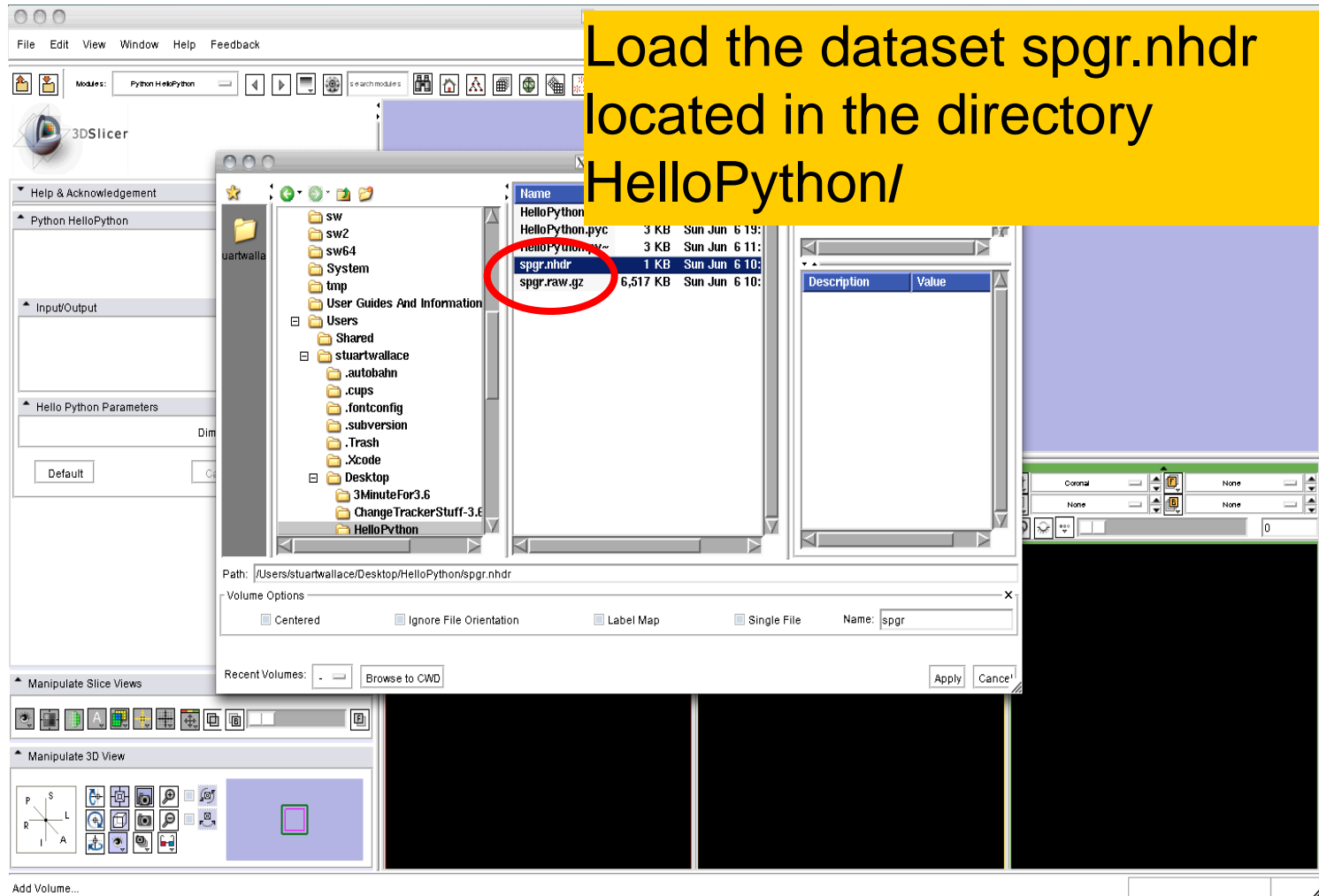
Running the Laplace Operator



The screenshot shows the 3DSlicer application window. The menu bar at the top includes 'File', 'Edit', 'View', 'Window', 'Help', and 'Feedback'. The 'File' menu is circled in red. A yellow callout box with black text reads: 'Re-start Slicer and select File → Add Volume'. The main 3D view area is a light purple color with a pink wireframe box in the center, labeled with 'R', 'A', 'L', and 'I' at its corners. Below the 3D view are three panels for slice views: Axial (red header), Sagittal (yellow header), and Coronal (green header). Each panel has a 'None' button and a numerical value of '0'. The left sidebar contains several panels: 'Help & Acknowledgement', 'Python HelloPython' (with 'Parameter set' and 'Status Completed with errors'), 'Input/Output' (with 'Input Volume' and 'output Volume' fields), 'Hello Python Parameters' (with 'Dimensionality' set to '3'), 'Manipulate Slice Views' (with various icons), and 'Manipulate 3D View' (with a 3D orientation diagram and manipulation icons). At the bottom left, the status bar shows 'spgr RAS: (131.6, -2.8, 186.3), Bg: Slice not shown.'

Running the Laplace Operator

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

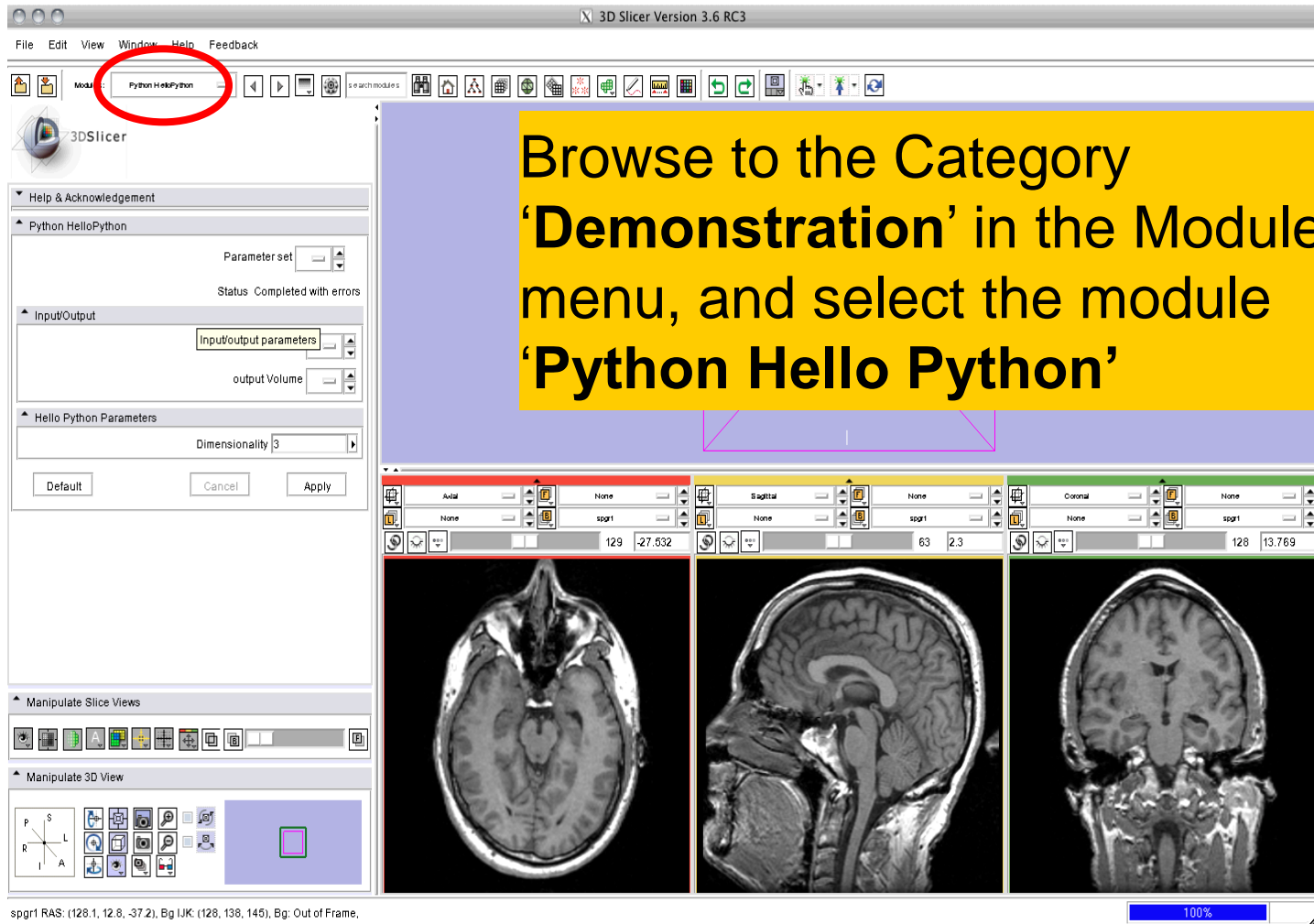


The screenshot shows the 3DSlicer interface with a file selection dialog open. The dialog displays a directory tree on the left, with 'HelloPython' selected. The right pane shows a list of files:

Name	Size	Date	Time
HelloPython			
HelloPython.pyc	3 KB	Sun Jun 6 19:	
HelloPython.pyw	3 KB	Sun Jun 6 11:	
spgr.nhdr	1 KB	Sun Jun 6 10:	
spgr.raw.gz	6,517 KB	Sun Jun 6 10:	

The file 'spgr.nhdr' is circled in red. The path at the bottom of the dialog is: /Users/stuartwallace/Desktop/HelloPython/spgr.nhdr. The 'Volume Options' section shows 'Name: spgr'.

Running the Laplace Operator

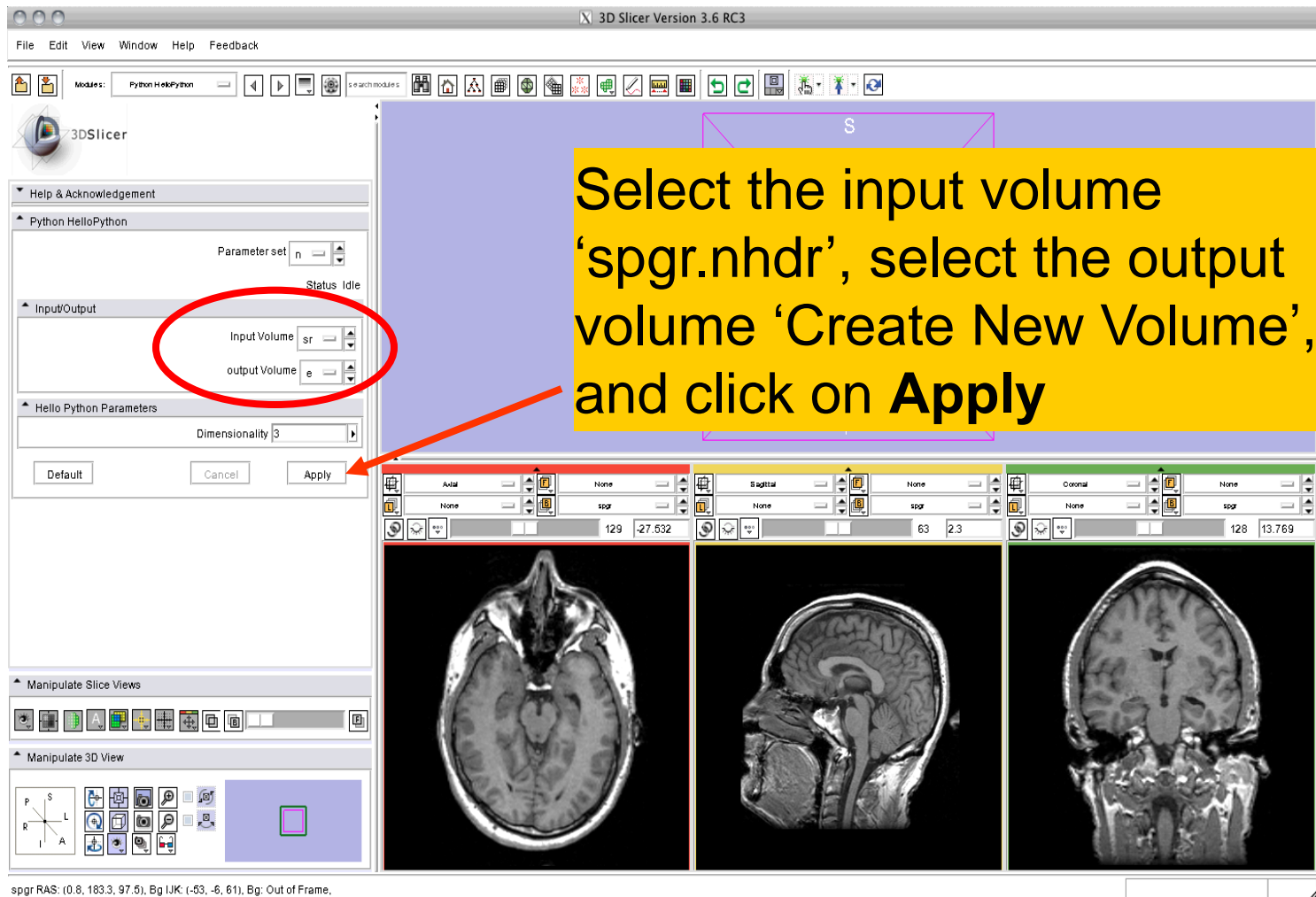


The screenshot shows the 3D Slicer 3.6 RC3 interface. The 'Modules' menu is open, and 'Python Hello Python' is selected, highlighted with a red circle. A yellow text box with black text is overlaid on the right side of the interface, containing the following instructions:

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 fields. The bottom of the interface displays three slice views (Axial, Sagittal, Coronal) and a 3D view of a brain MRI scan.

Running the Laplace Operator



3D Slicer Version 3.6 RC3

File Edit View Window Help Feedback

Modules: Python HelloPython

3DSlicer

Help & Acknowledgement

Python HelloPython

Parameter set: n

Status: Idle

Input/Output

Input Volume: **sr**

Output Volume: **e**

Hello Python Parameters

Dimensionality: 3

Default Cancel Apply

Manipulate Slice Views

Manipulate 3D View

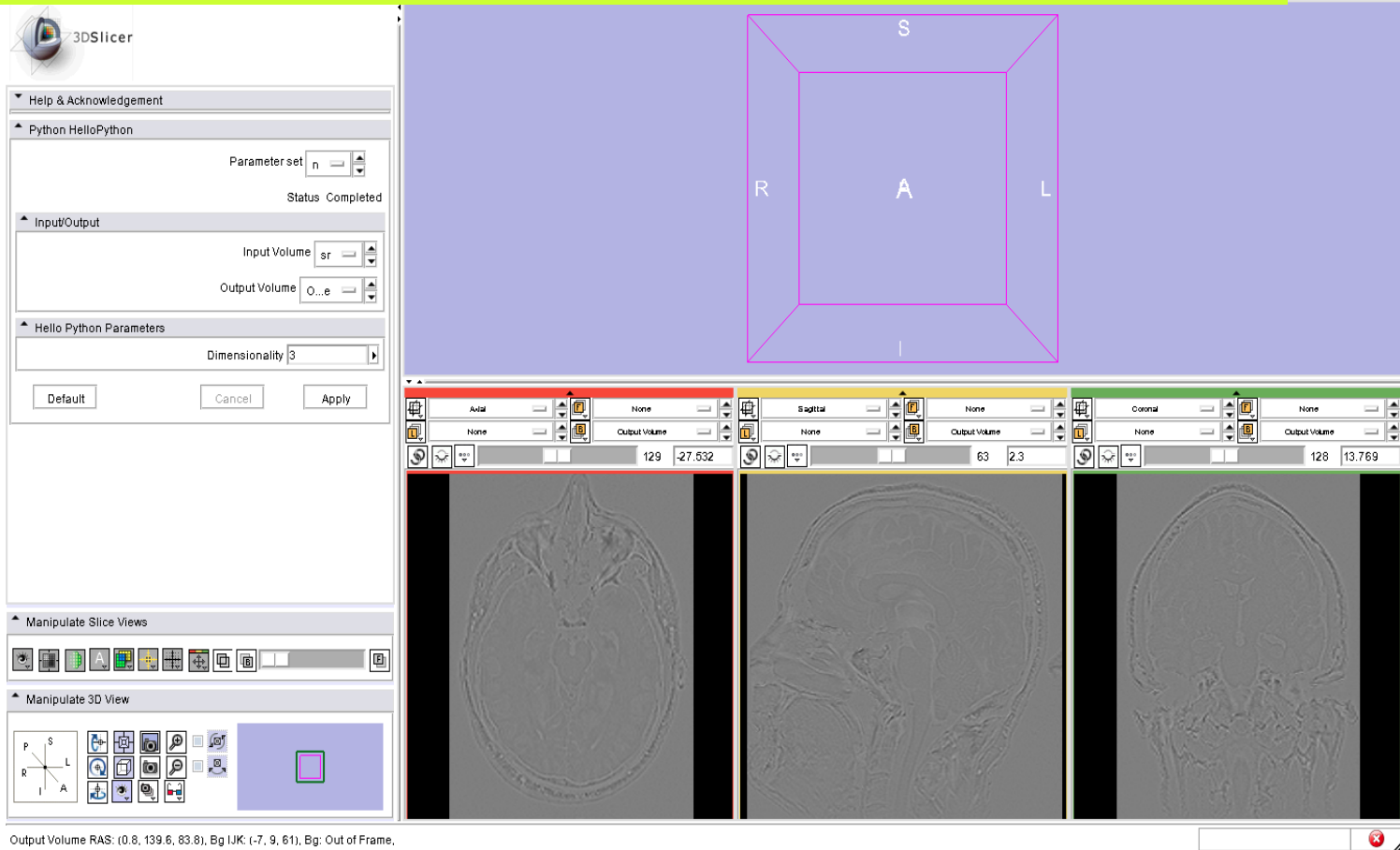
spgr RAS: (0.8, 183.3, 97.5), Bg IJK: (-63, -6, 61), Bg: Out of Frame.

Axial	None	Sagittal	None	Coronal	None
None	spgr	None	spgr	None	spgr
129	-27.632	63	2.3	128	13.769

Select the input volume 'spgr.nhdr', select the output volume 'Create New Volume', and click on **Apply**

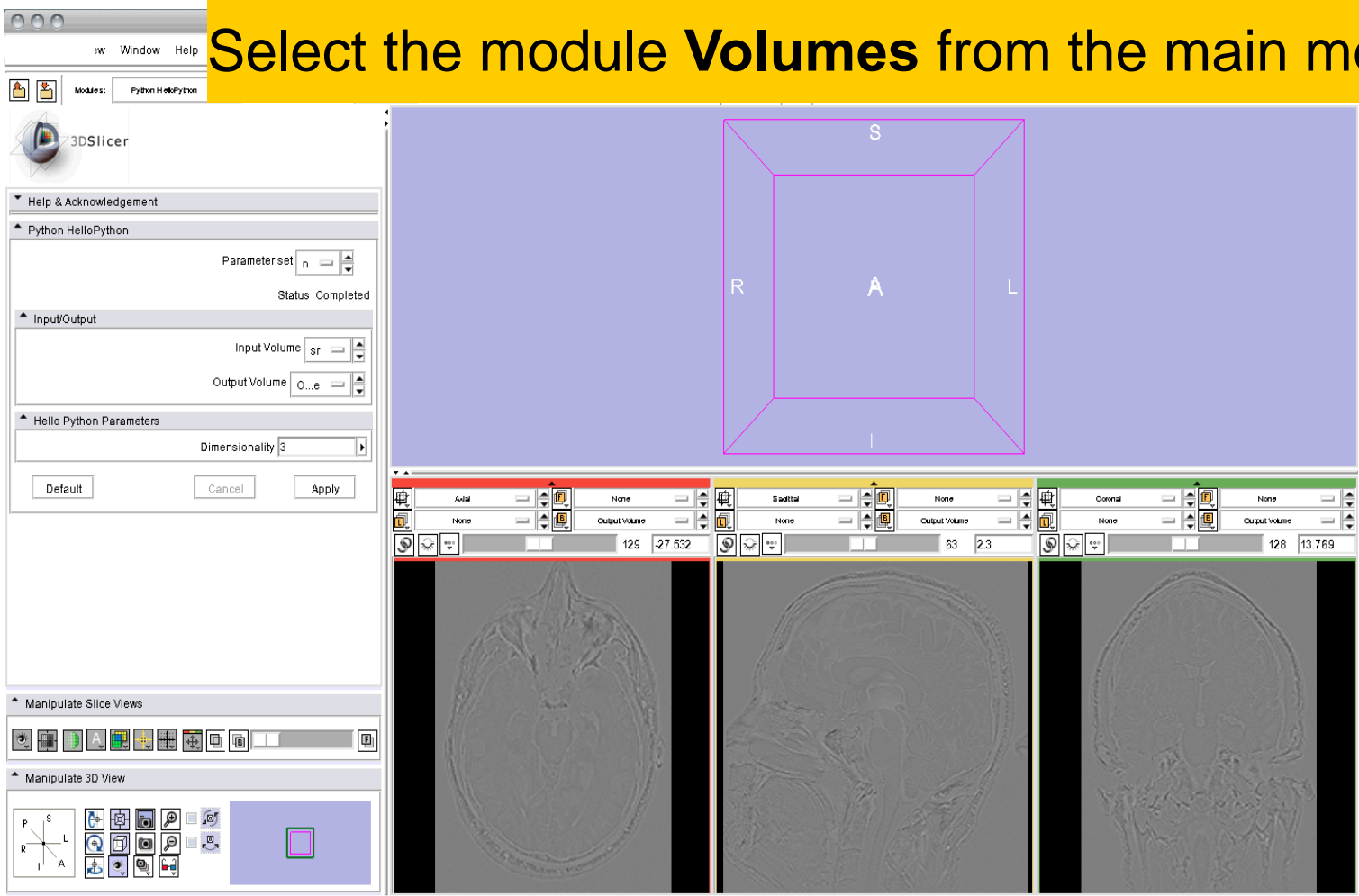
Running the Laplace Operator

Slicer displays the Laplacian of the spgr image.



Laplacian of the image

Select the module **Volumes** from the main menu



The screenshot shows the 3DSlicer software interface. On the left, the 'Python HelloPython' module is active, showing parameters for 'Parameter set' (n), 'Input Volume' (sr), 'Output Volume' (O...e), and 'Dimensionality' (3). The main 3D view area is a light purple color with a white wireframe box overlaid, labeled with 'S' (Superior), 'I' (Inferior), 'R' (Right), and 'L' (Left). Below the 3D view, three orthogonal slice views are displayed: Axial (top), Sagittal (middle), and Coronal (bottom). Each slice view shows a grayscale image of a brain with a white outline representing the Laplacian of the image. The status bar at the bottom indicates 'Output Volume RAS: (0.8, 139.6, 83.8), Bg IJK: (-7, 9, 61), Bg: Out of Frame.'

Laplacian of the image

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

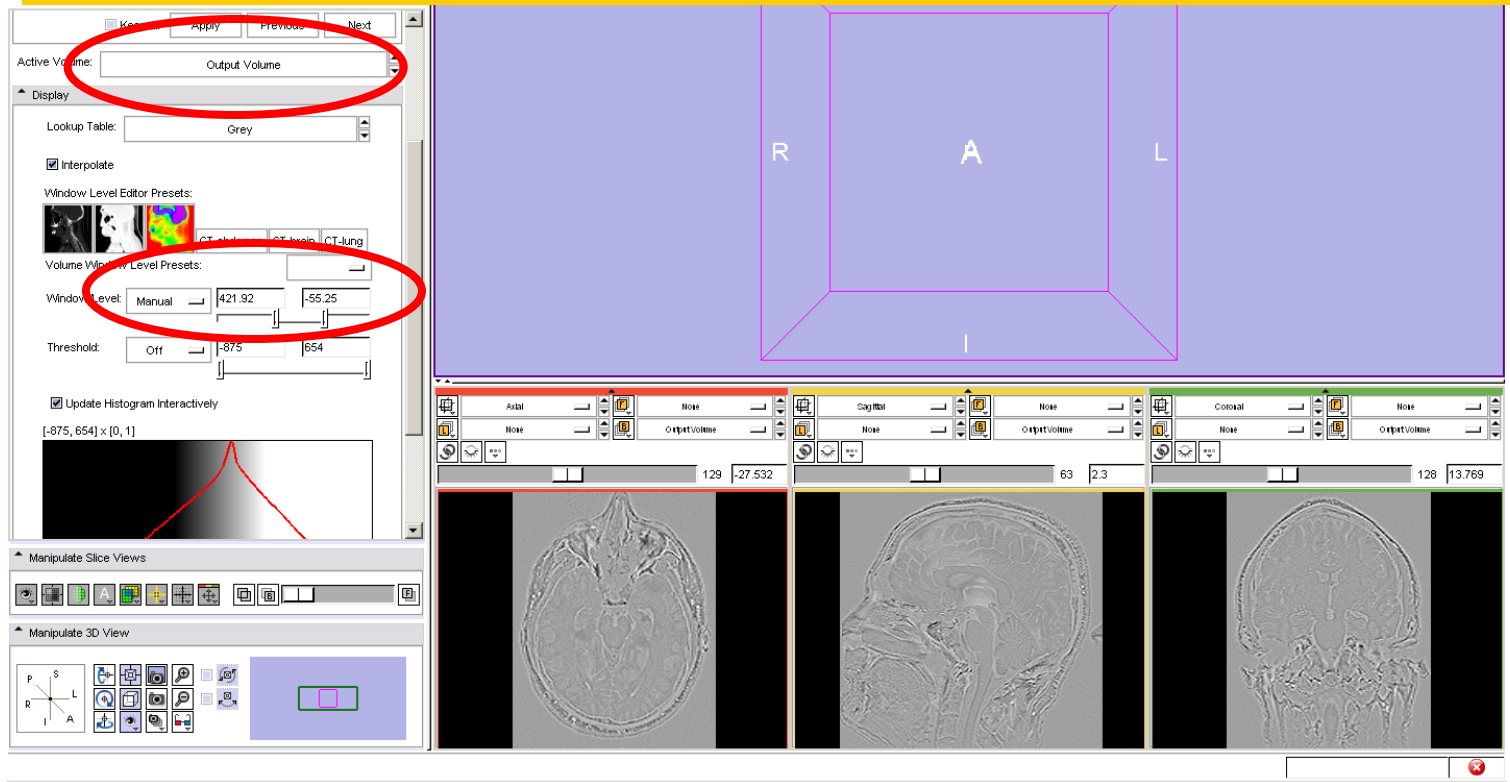
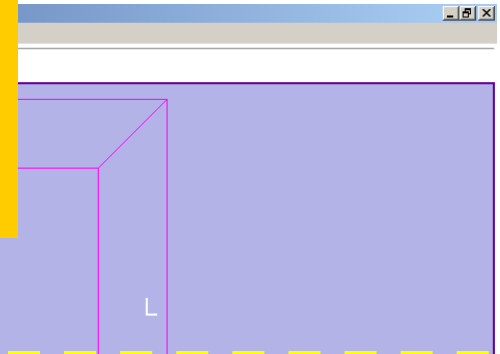


Image Sharpening

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)
```

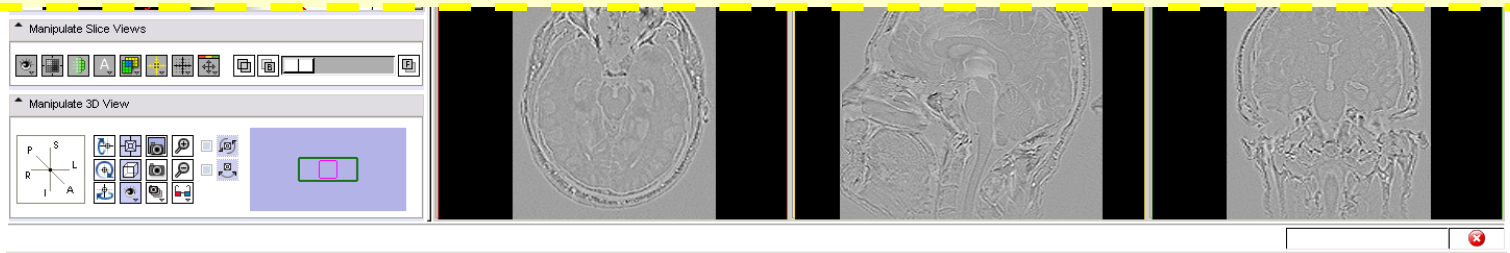
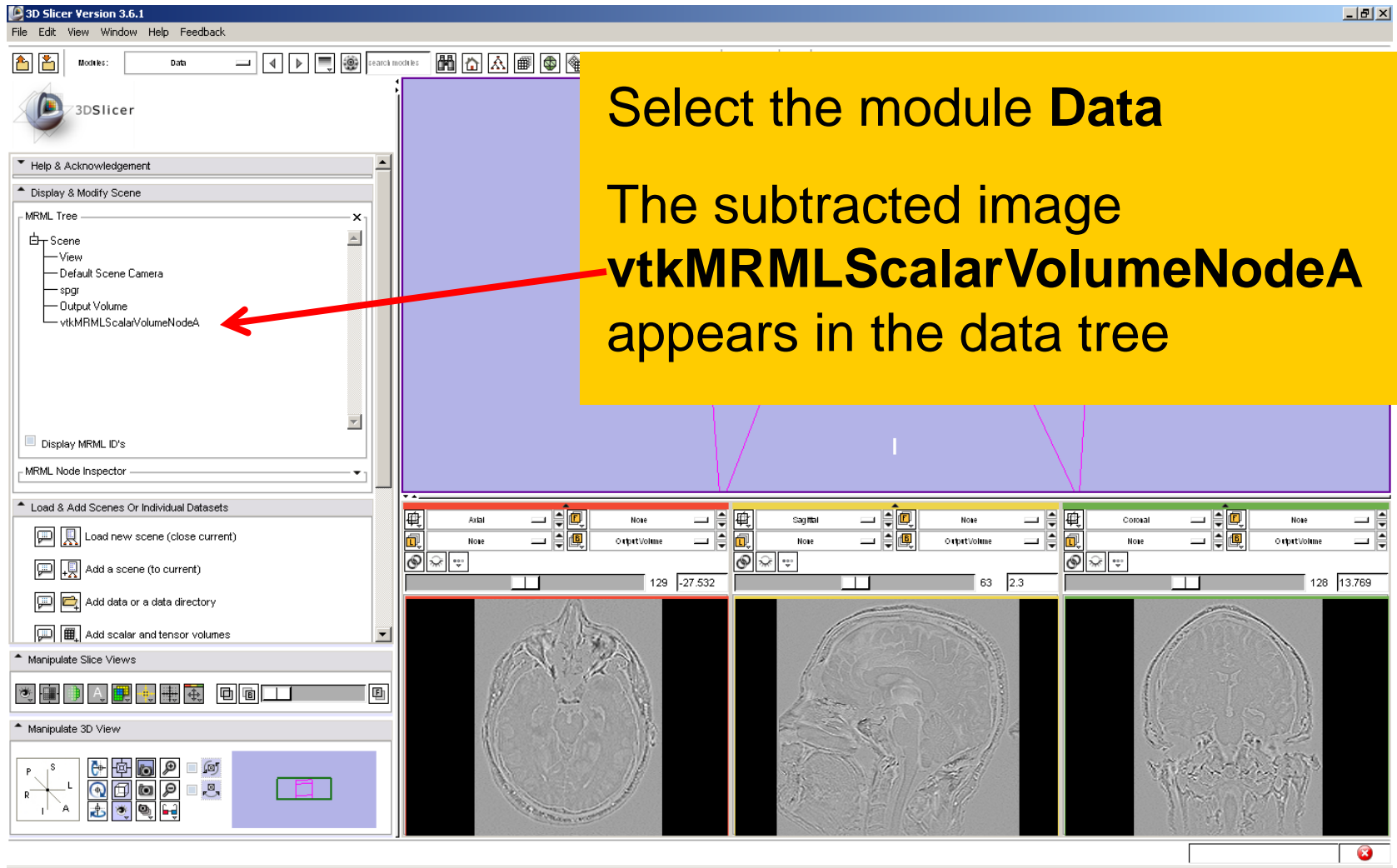


Image Sharpening



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 with the following structure:

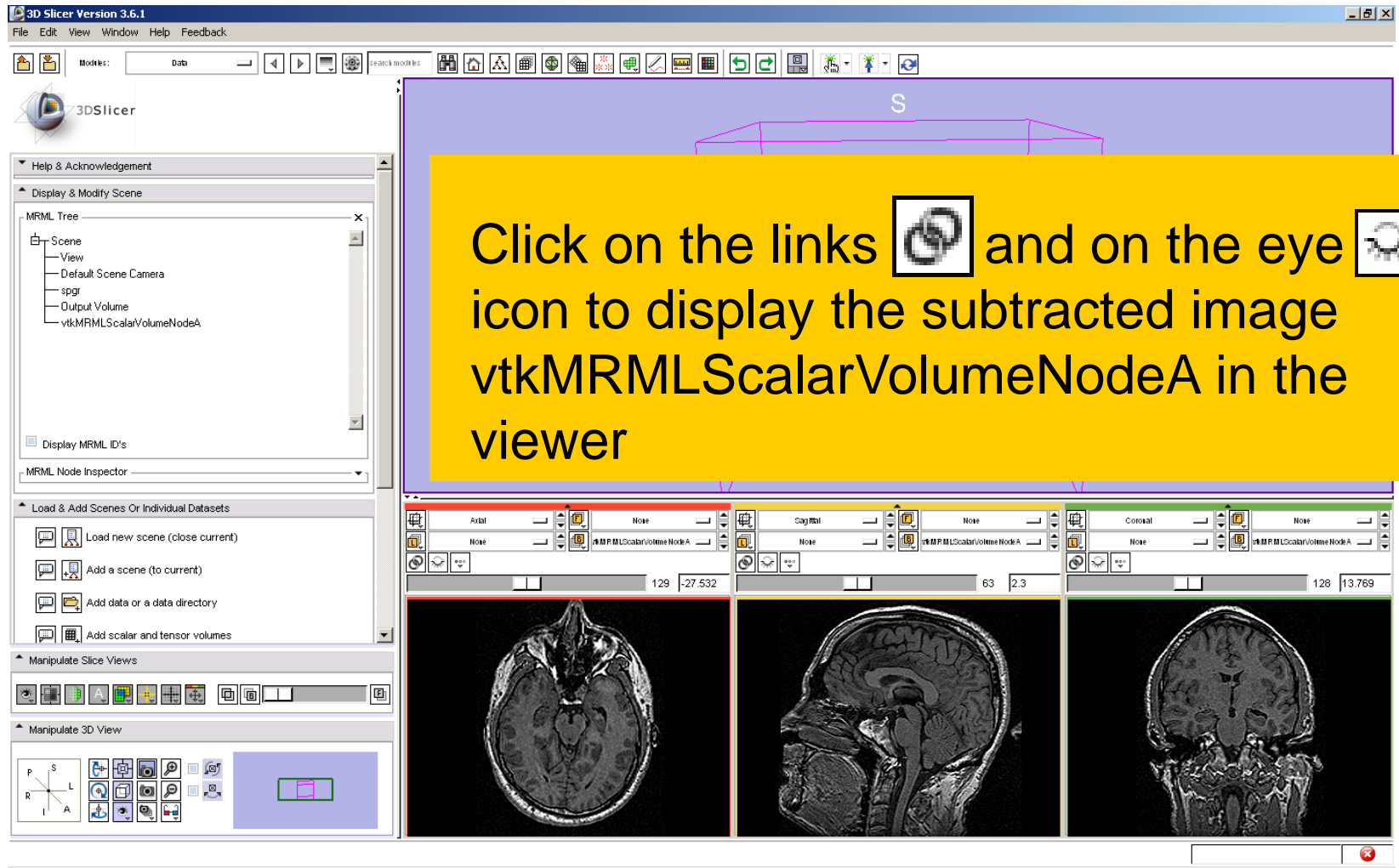
- Scene
 - View
 - Default Scene Camera
 - spgr
 - Output Volume
 - vtkMRMLScalarVolumeNodeA

A red arrow points to the 'vtkMRMLScalarVolumeNodeA' node in the tree. The main 3D view area is divided into three panels: Axial, Sagittal, and Coronal. Each panel shows a grayscale image of a brain slice. The 'Data' module is selected in the top toolbar.



Select the module **Data**

The subtracted image **vtkMRMLScalarVolumeNodeA** appears in the data tree

Image Sharpening



The screenshot shows the 3D Slicer 3.6.1 interface. A yellow box with black text is overlaid on the top right, containing the following instructions:

Click on the links  and on the eye icon  to display the subtracted image vtkMRMLScalarVolumeNodeA in the viewer

The interface includes a menu bar (File, Edit, View, Window, Help, Feedback), a toolbar, and several panels:

- MRML Tree:** Shows a hierarchy with 'Scene' expanded to 'View', 'Default Scene Camera', 'spgr', 'Output Volume', and 'vtkMRMLScalarVolumeNodeA'.
- Load & Add Scenes Or Individual Datasets:** Contains buttons for 'Load new scene (close current)', 'Add a scene (to current)', 'Add data or a data directory', and 'Add scalar and tensor volumes'.
- Manipulate Slice Views:** Features icons for axial, sagittal, and coronal views.
- Manipulate 3D View:** Includes a 3D orientation compass and manipulation tools.

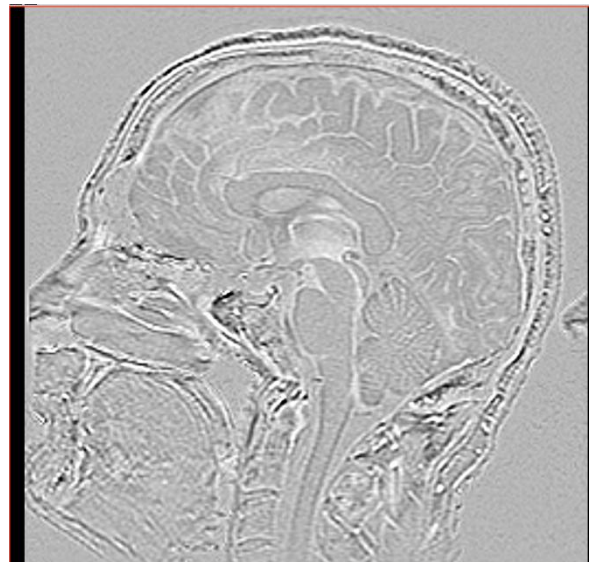
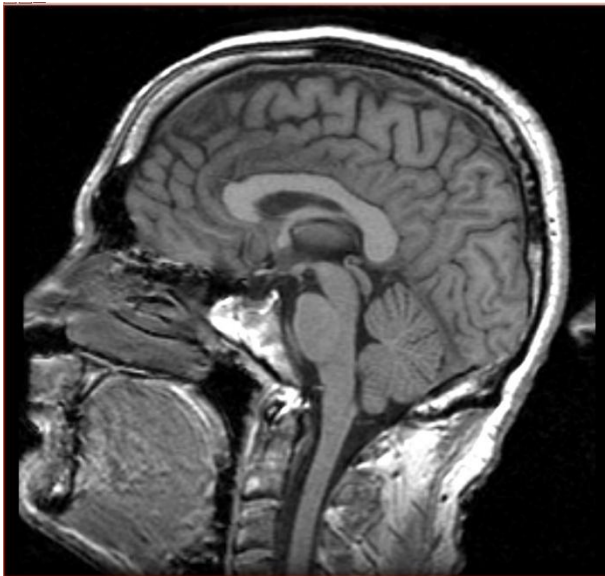
 The main viewer area displays three MRI slices: Axial (z-axis: 129, -27.532), Sagittal (z-axis: 63, 2.3), and Coronal (z-axis: 128, 13.769). Each slice has a corresponding control bar with icons for visibility and zoom.

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