



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

Programming into Slicer3

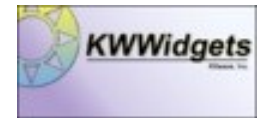
Sonia Pujol, Ph.D.

Surgical Planning Laboratory
Harvard University





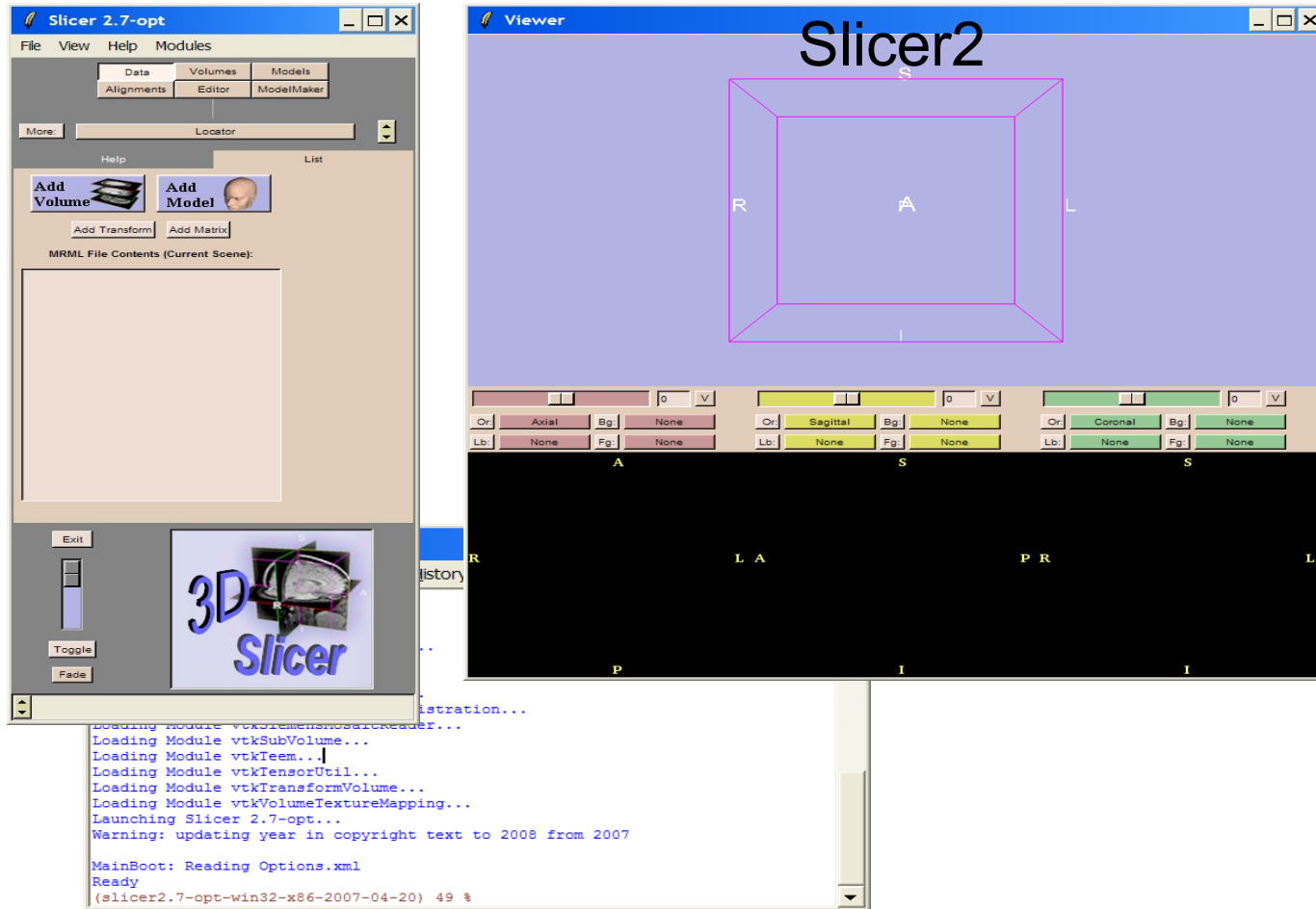
The NA-MIC Kit



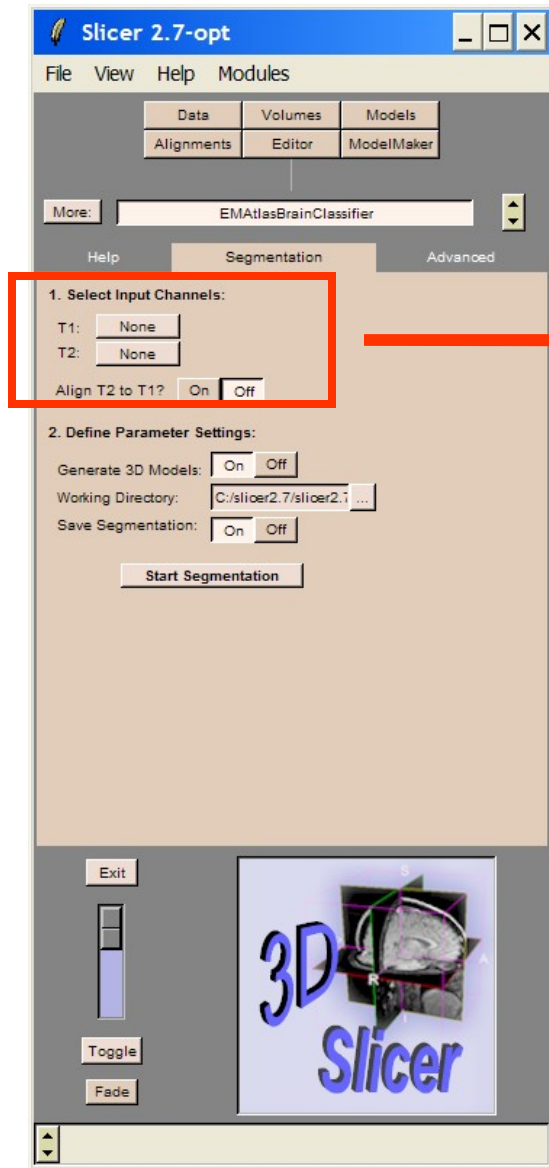


- An **end-user application** for image analysis
- An **open-source environment** for software development
- A software platform that is both **easy to use** for clinical researchers and **easy to extend** for programmers

Before Slicer3



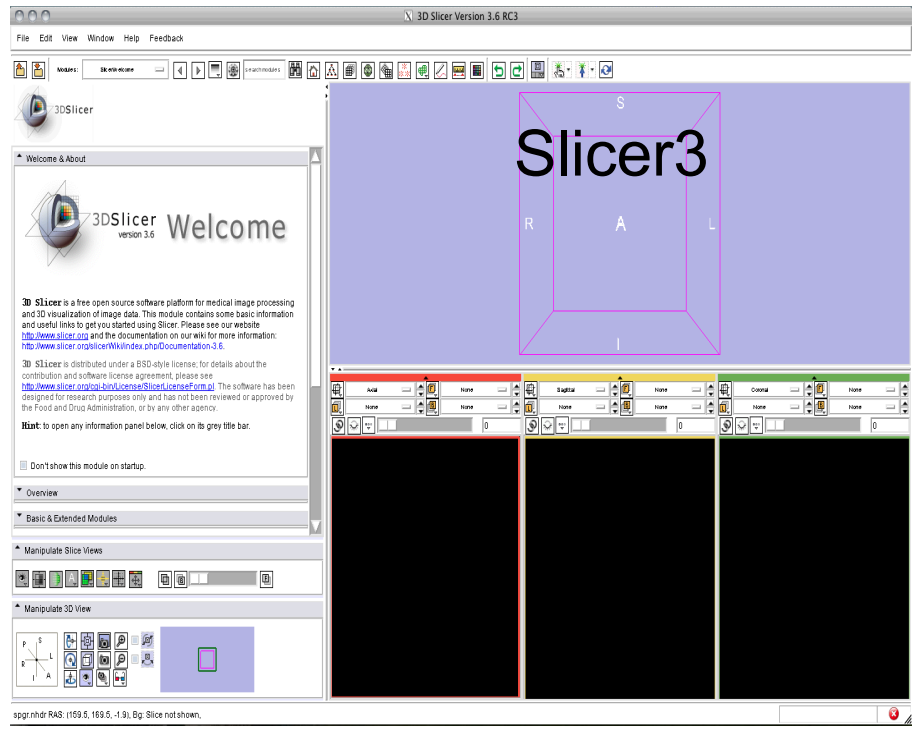
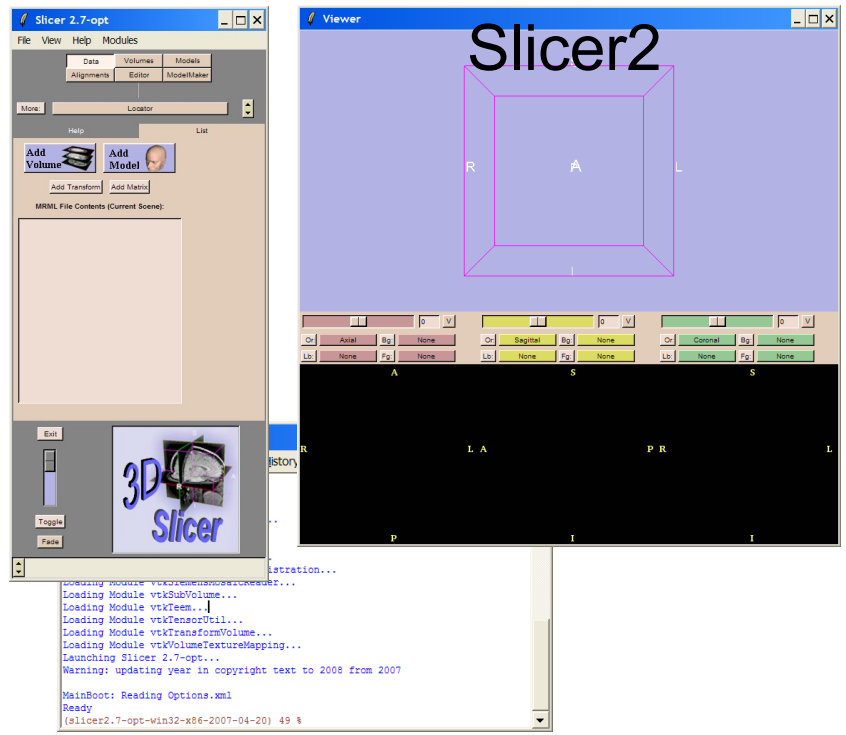
Programming into Slicer2



```

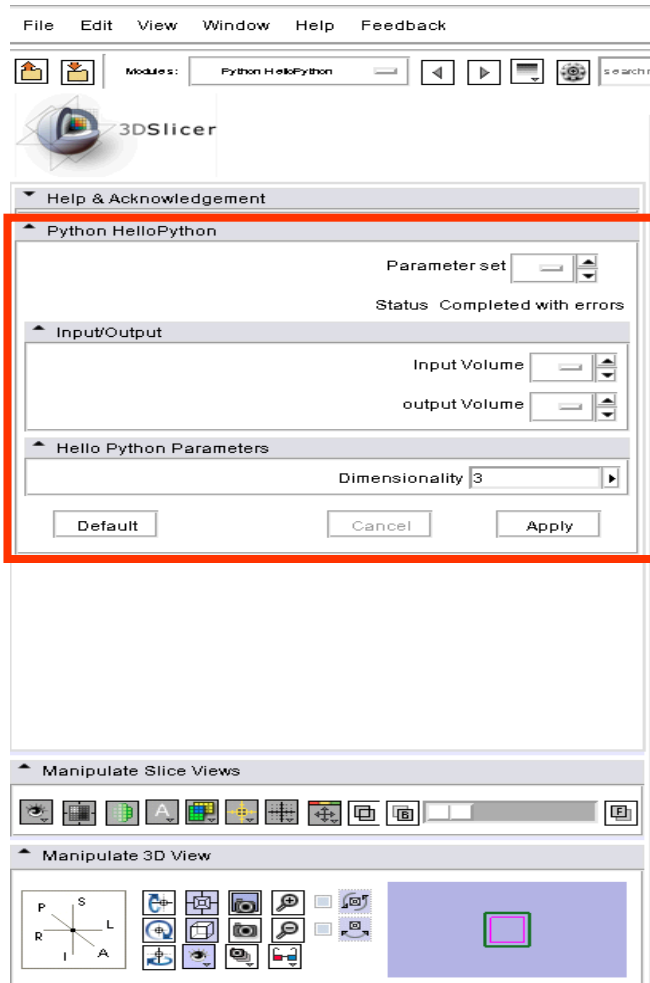
#-----
# 1. Step
#-----
set f $fSeg.fStep1
DevAddLabel $f.ITitle "1. Select Input Channels: " WTA
pack $f.ITitle -side top -padx $Gui(pad) -pady 1 -anchor w
frame $f.fInput -bg $Gui(activeWorkspace)
pack $f.fInput -side top -padx 0 -pady 0 -anchor w
foreach frame "Left Right" {
    frame $f.fInput.f$frame -bg $Gui(activeWorkspace)
    pack $f.fInput.f$frame -side left -padx 0 -pady $Gui(pad) }
foreach LABEL "T1 T2" Input "SPGR T2W" {
    DevAddLabel $f.fInput.fLeft.l$Input " ${LABEL}:"
    pack $f.fInput.fLeft.l$Input -side top -padx $Gui(pad) -pady 1 -anchor w
    set menubutton $f.fInput.fRight.m${Input}Select
    set menu $f.fInput.fRight.m${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.fAlign -bg $Gui(activeWorkspace)
TooltipAdd $f.fAlign "If the input T1 and T2 are not aligned with each other set flag here"
pack $f.fAlign -side top -padx 0 -pady 2 -padx $Gui(pad) -anchor w
DevAddLabel $f.fAlign.l$Align "Align T2 to T1? "
pack $f.fAlign.l$Align -side left -padx $Gui(pad) -pady 1 -anchor w
foreach value "1 0" text "On Off" width "4 4" {
    eval {radiobutton $f.fAlign.r$value -width $width -indicatoron 0
        -text "$text" -value "$value" -variable EMAtlasBrainClassifier(AlignInput) } $Gui(WCA)
    pack $f.fAlign.r$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-uri> </documentation-uri>
  <license></license>
  <contributor>
    Sonia Pujol, Ph.D., Surgical Planning Laboratory, Harvard Medical School
  </contributor>
  <acknowledgements> National Alliance for Medical Image Computing (NAMIC), Grant
  J54 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>
```



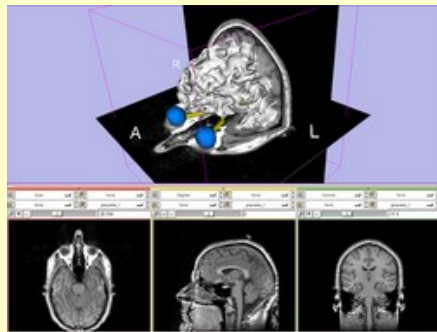
Slicer3 Execution Model

- 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

This course requires the following material

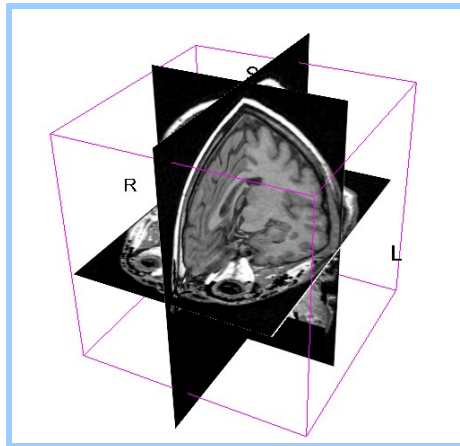
- Slicer3.6
- HelloPython.zip

Disclaimer

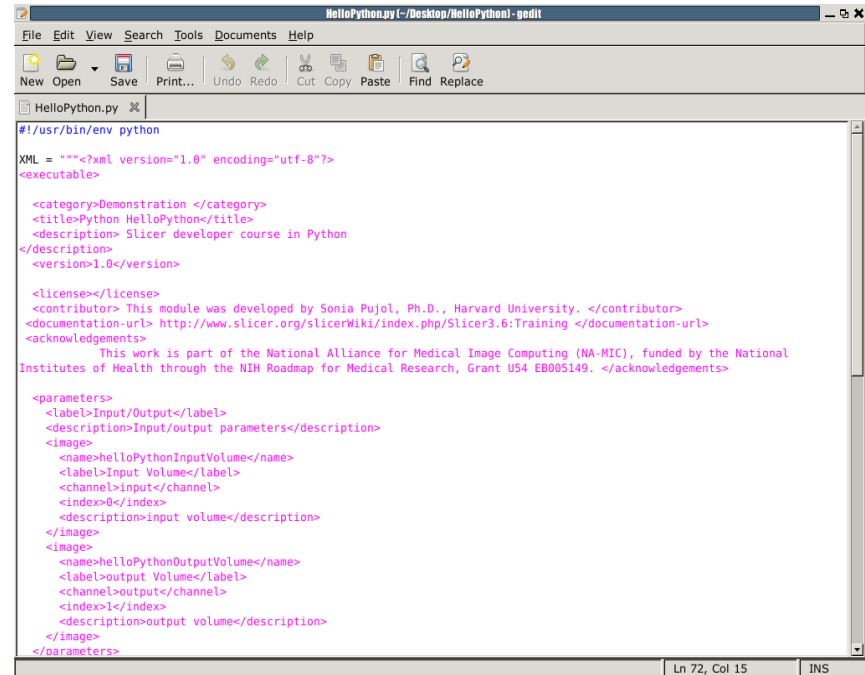
It is the responsibility of the user of 3DSlicer to comply with both the terms of the license and with the applicable laws, regulations and rules.

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>

```

HelloPython.py

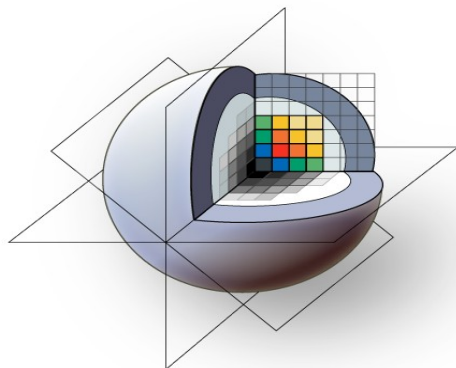


Overview

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



3DSlicer

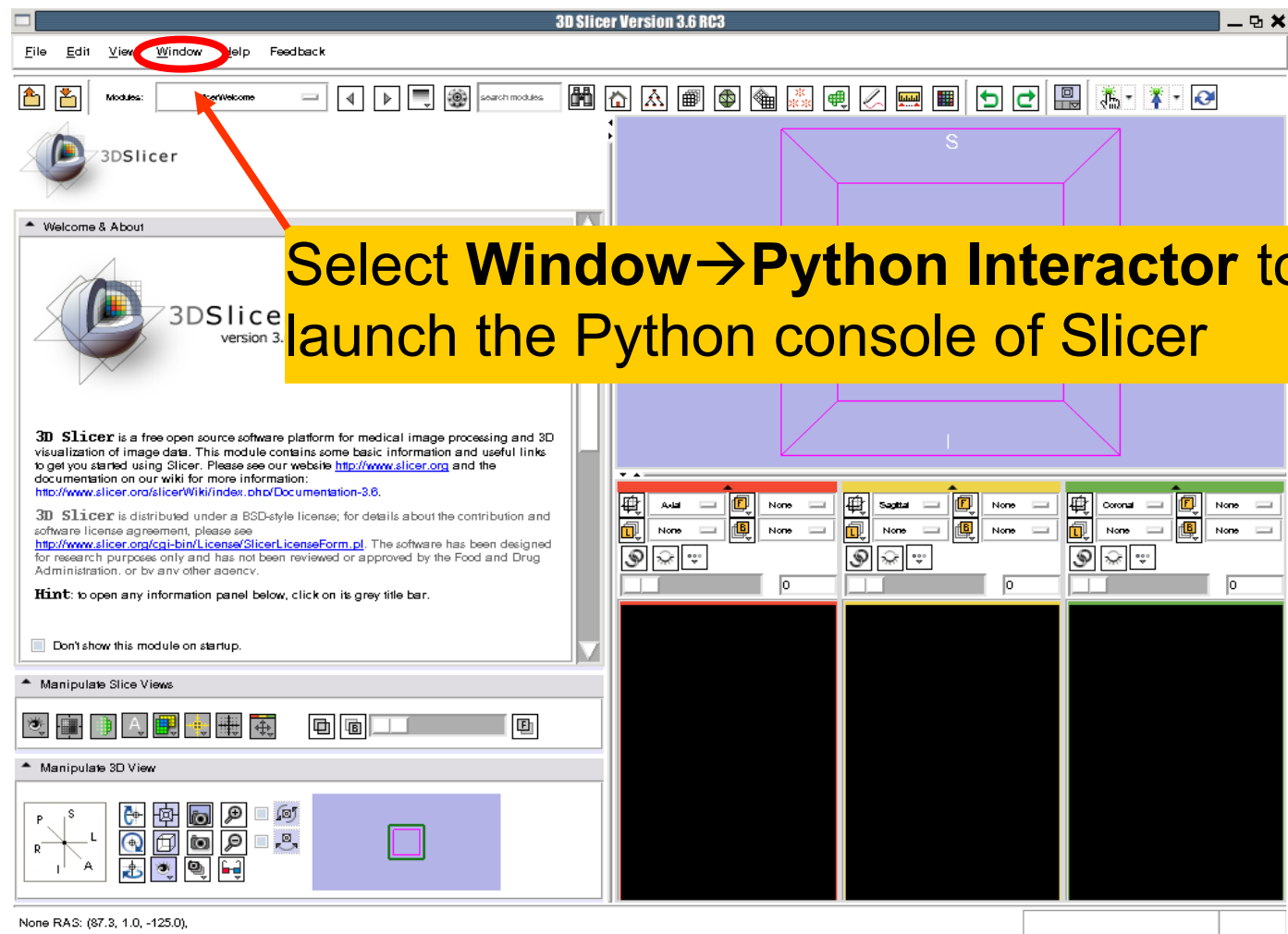


3DSlicer

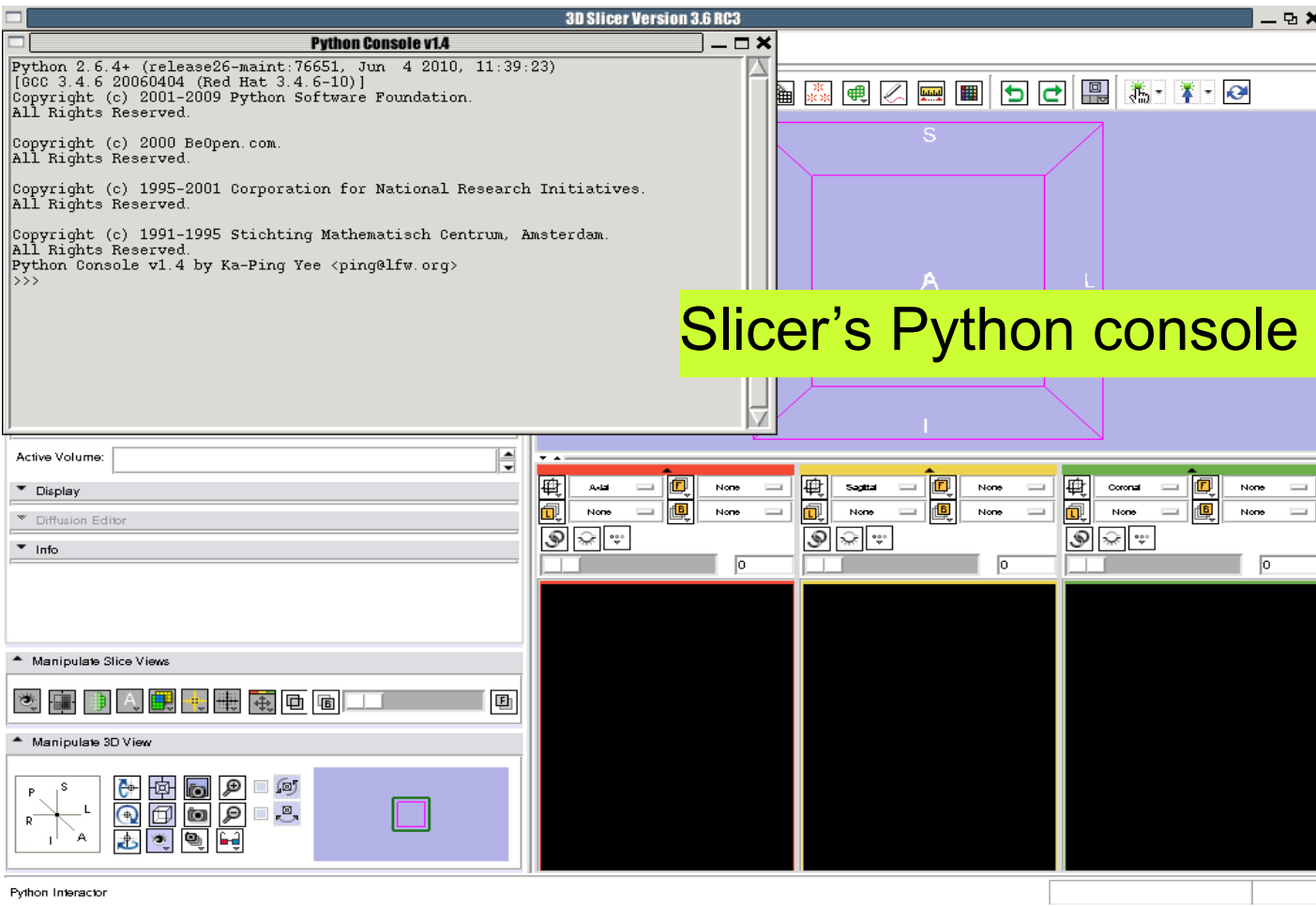
Part A: Integrating HelloPython into Slicer3

```
#!/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

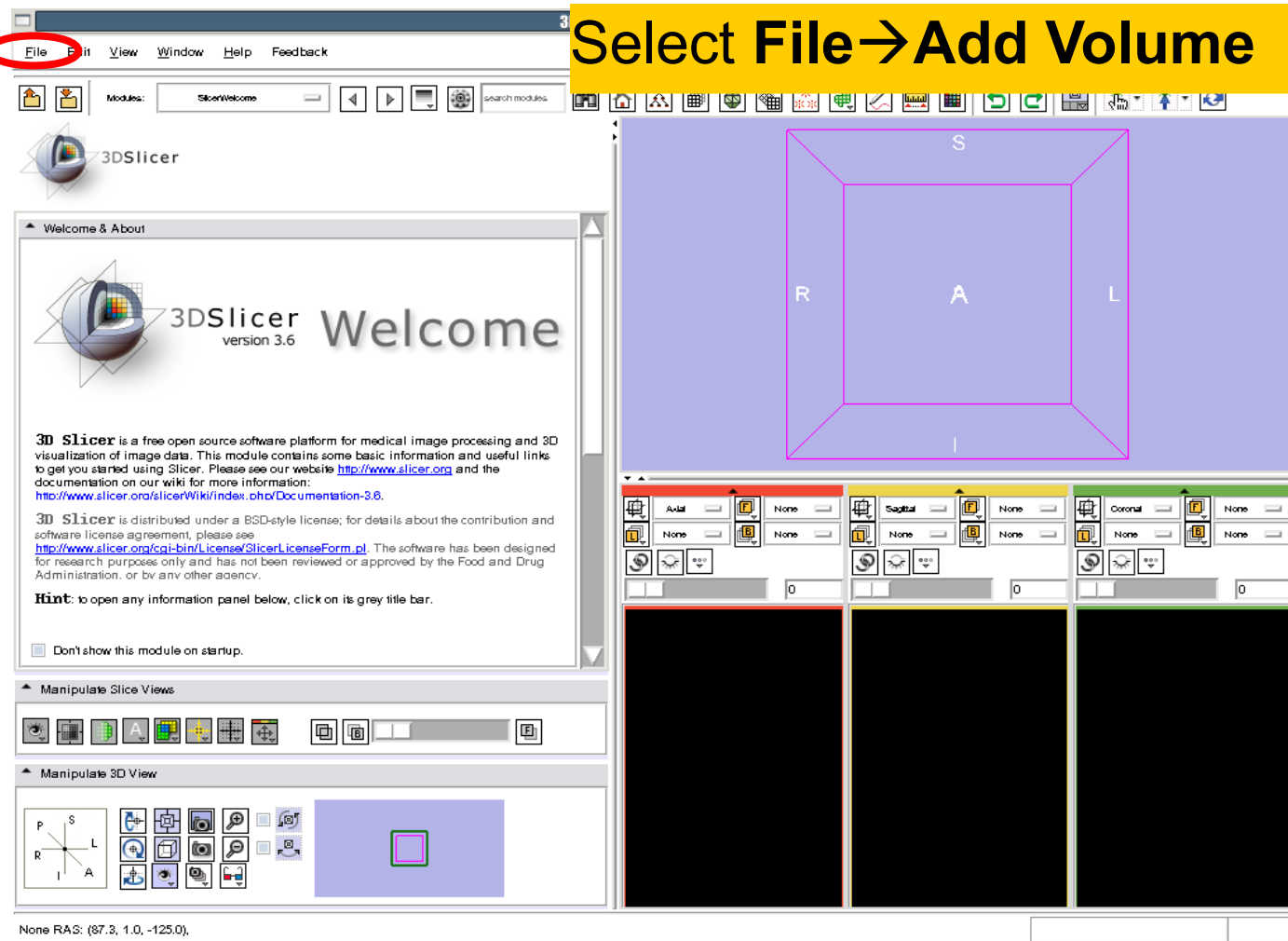


Python Console



Python Console

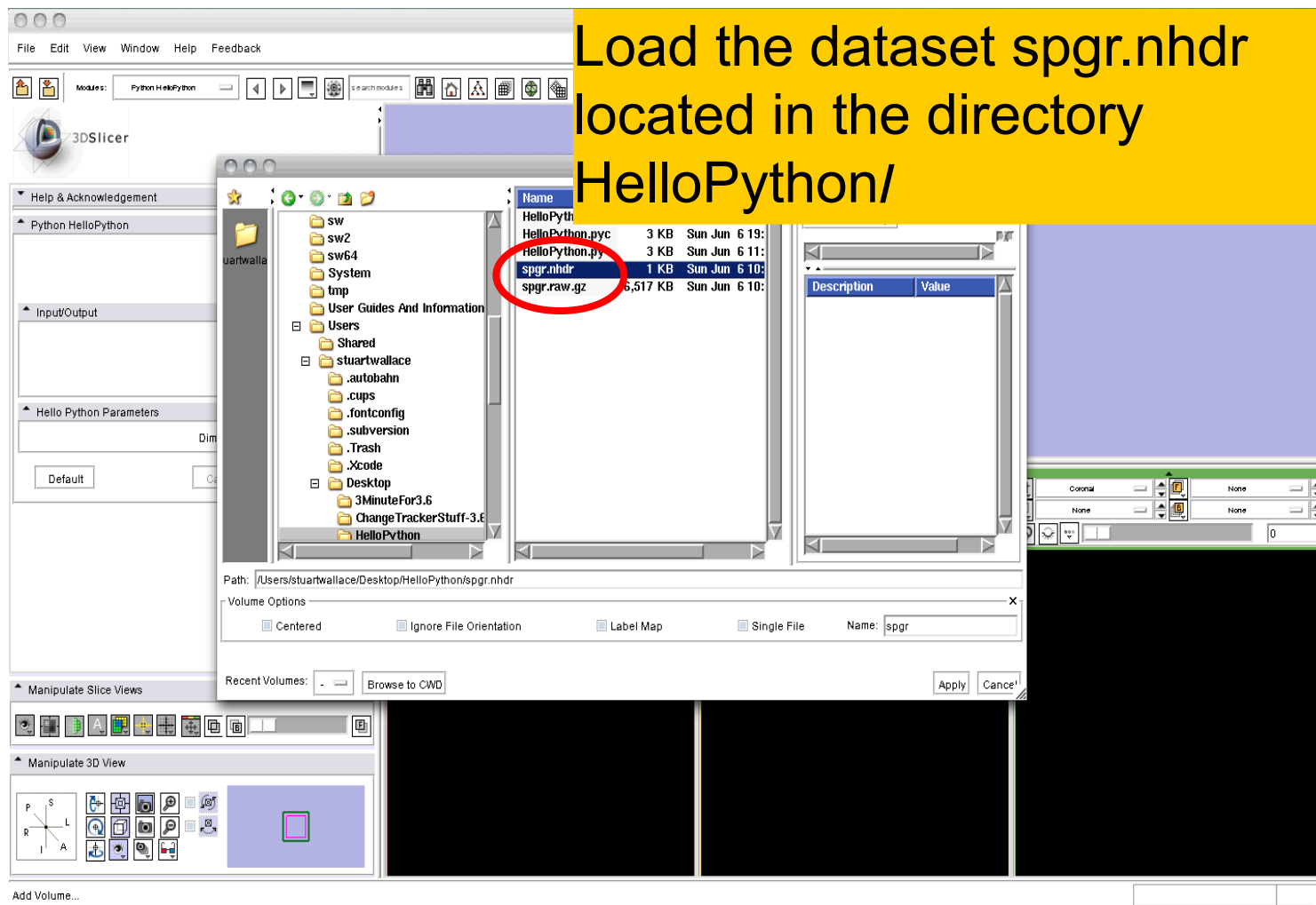
Select File → Add Volume



The screenshot displays the 3DSlicer 3.6 software interface. The top menu bar includes 'File', 'Edit', 'View', 'Window', 'Help', and 'Feedback'. The 'File' menu is highlighted with a red circle. A yellow banner at the top right contains the text 'Select File → Add Volume'. The main interface is divided into several panels: a 'Welcome & About' panel on the left, a central 3D view area with a purple wireframe box labeled 'S', 'R', 'A', 'L', 'I', and a bottom toolbar with 'Axial', 'Sagittal', and 'Coronal' view options. The status bar at the bottom indicates 'None RAS: (87.3, 1.0, -125.0)'.

Python Console

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



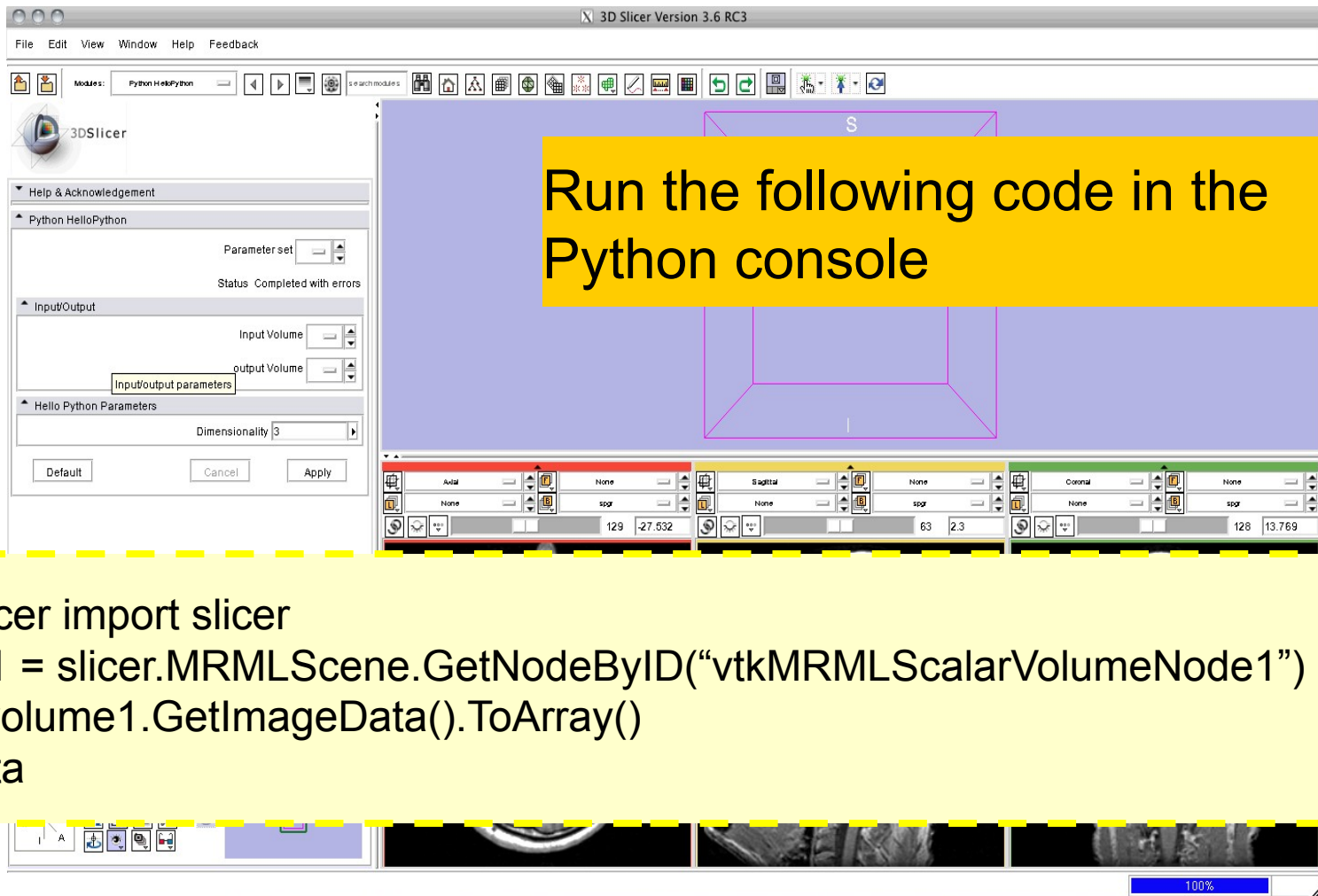
The screenshot shows the 3DSlicer application window with a file explorer window open. The file explorer window displays the contents of the directory `/Users/stuartwallace/Desktop/HelloPython/`. The file `spgr.nhdr` is highlighted with a red circle. The file explorer window also shows a table of files with the following data:

Name	Size	Modified
HelloPyth		
HelloPython.pyc	3 KB	Sun Jun 6 19:
HelloPython.py	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 3DSlicer interface shows the Python HelloPython module loaded, and the file explorer window is open to the directory `/Users/stuartwallace/Desktop/HelloPython/`. The file `spgr.nhdr` is selected, and the path is shown as `/Users/stuartwallace/Desktop/HelloPython/spgr.nhdr`. The file explorer window also shows a table of files with the following data:

Description	Value

Python Console



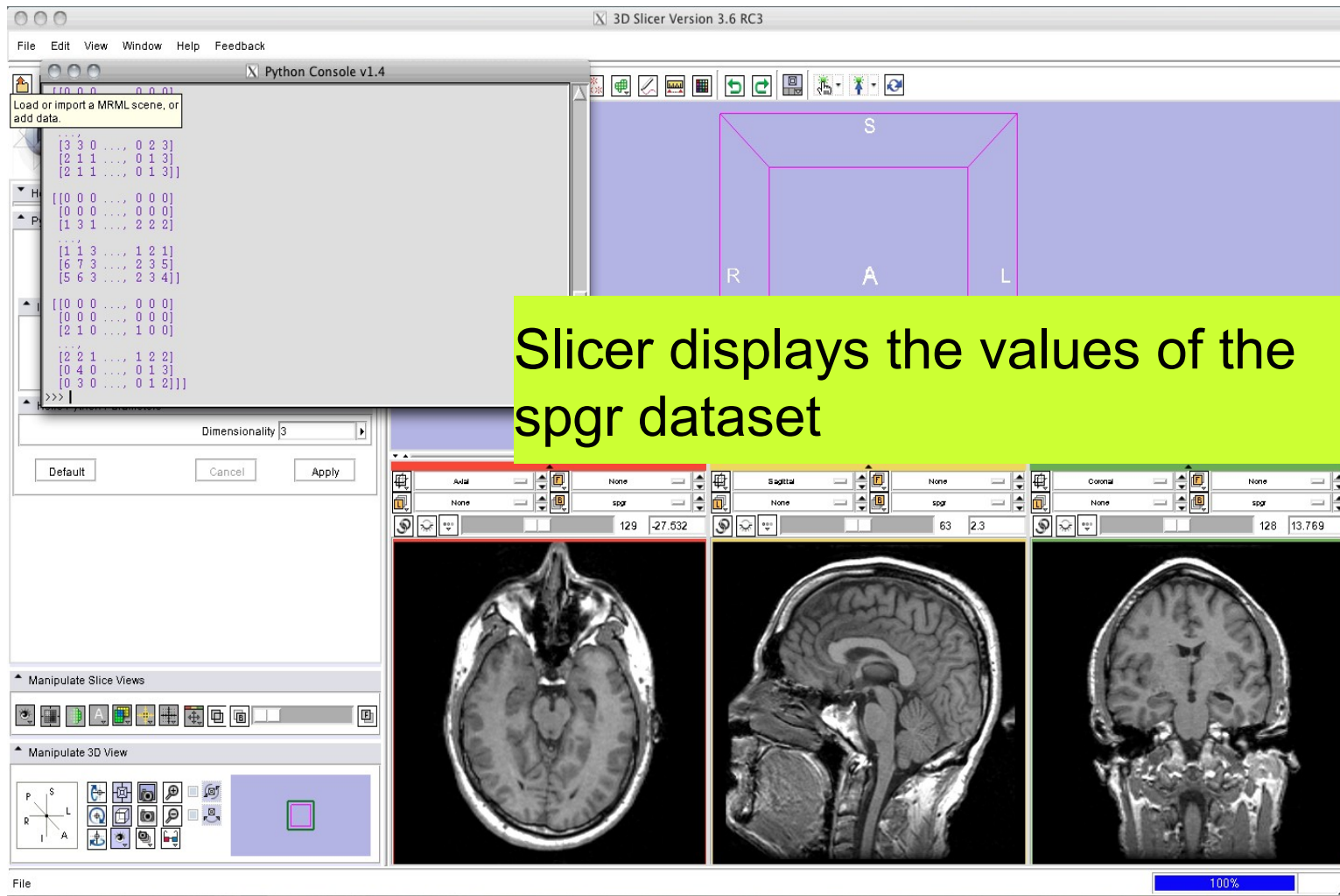
The screenshot shows the 3D Slicer 3.6 RC3 interface. The main window displays a 3D view of a brain slice with a yellow box overlaid on it containing the text "Run the following code in the Python console". The Python console is open on the left side of the interface, showing the following code:

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

The interface also shows various toolbars, a parameter set for "Python HelloPython", and a status bar at the bottom indicating 100% zoom.

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

Python Console



The screenshot shows the 3D Slicer 3.6 RC3 interface. The Python Console (v1.4) displays a matrix of values:

```

[[0 0 0 ..... 0 0 0]
 [3 3 0 ..... 0 2 3]
 [2 1 1 ..... 0 1 3]
 [2 1 1 ..... 0 1 3]]

[[0 0 0 ..... 0 0 0]
 [0 0 0 ..... 0 0 0]
 [1 3 1 ..... 2 2 2]]

[[1 1 3 ..... 1 2 1]
 [6 7 3 ..... 2 3 5]
 [5 6 3 ..... 2 3 4]]

[[0 0 0 ..... 0 0 0]
 [0 0 0 ..... 0 0 0]
 [2 1 0 ..... 1 0 0]]

[[2 2 1 ..... 1 2 2]
 [0 4 0 ..... 0 1 3]
 [0 3 0 ..... 0 1 2]]
  
```

The 3D view shows a purple bounding box with axes labeled S (Superior), R (Right), A (Anterior), and L (Left). The slice views at the bottom show Axial, Sagittal, and Coronal slices of a brain MRI. The Python Console also shows a 'Dimensionality 3' dialog box.

Slicer displays the values of the spgr dataset

HelloPython.py

<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>""";

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>
  <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.HIFILScene

    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

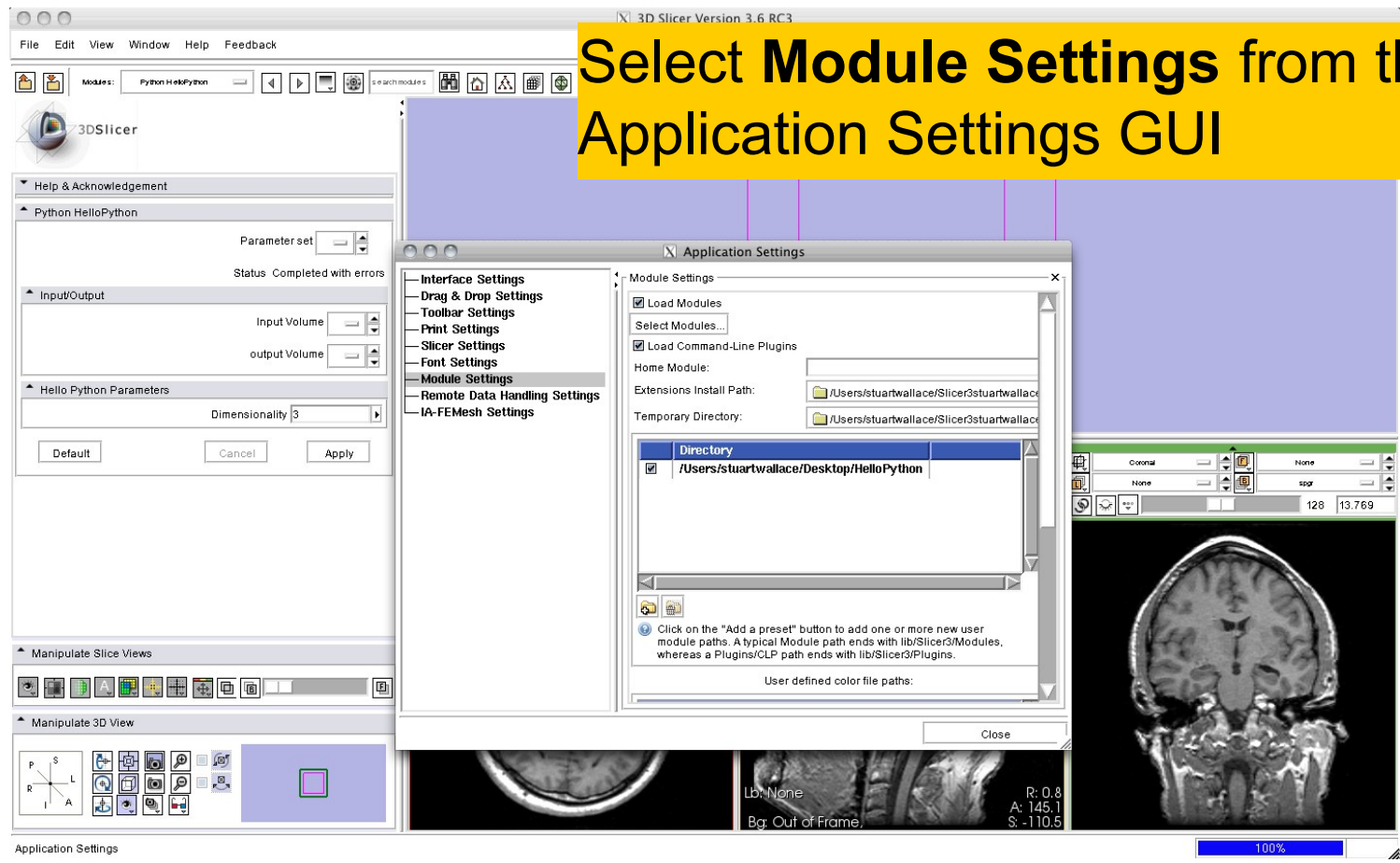
3D Slicer Version 3.6 RC3

Click on the View → Application Settings in the main menu

Application Settings

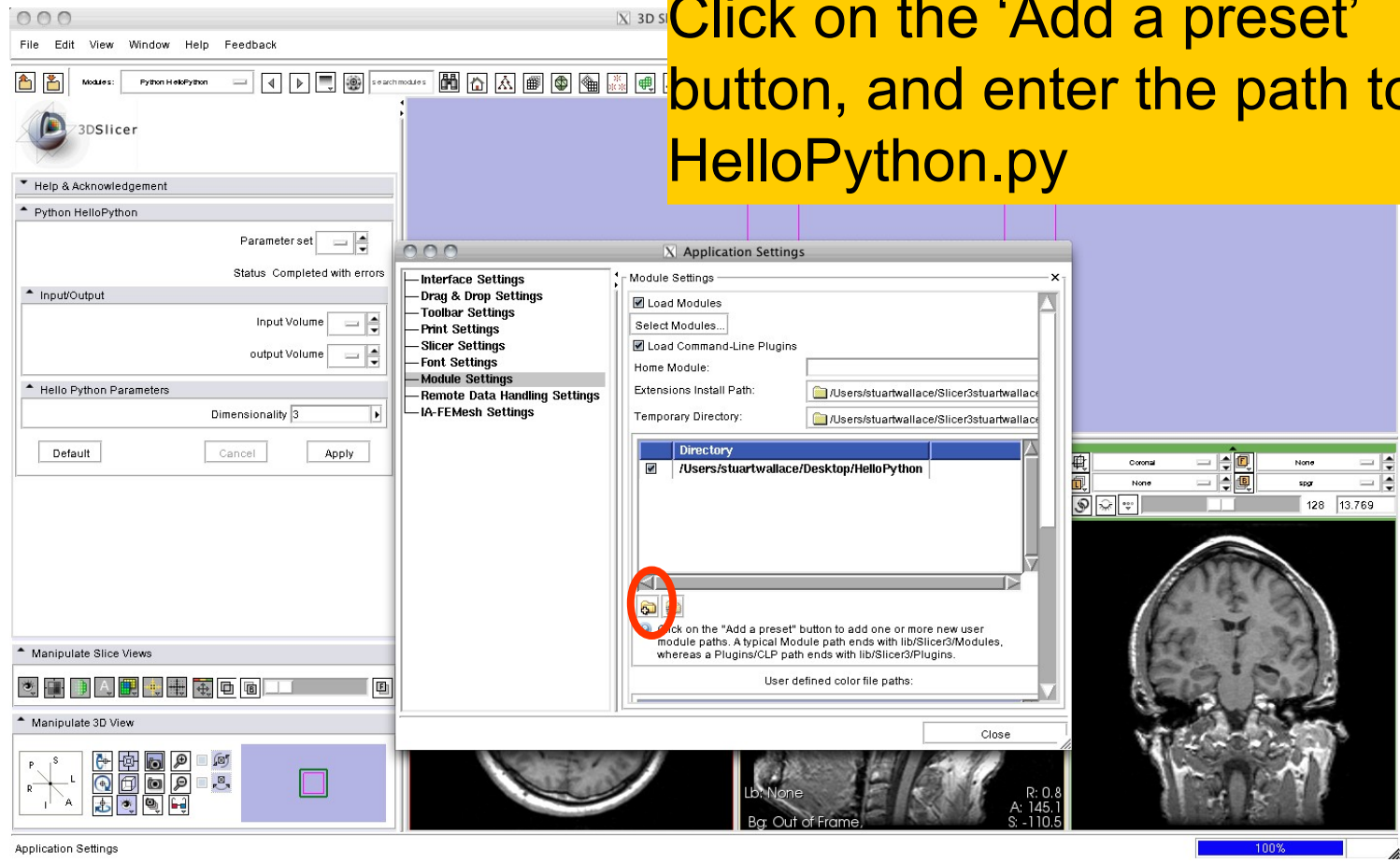
100%

Integrating HelloPython to Slicer3



Integrating HelloPython to Slicer3

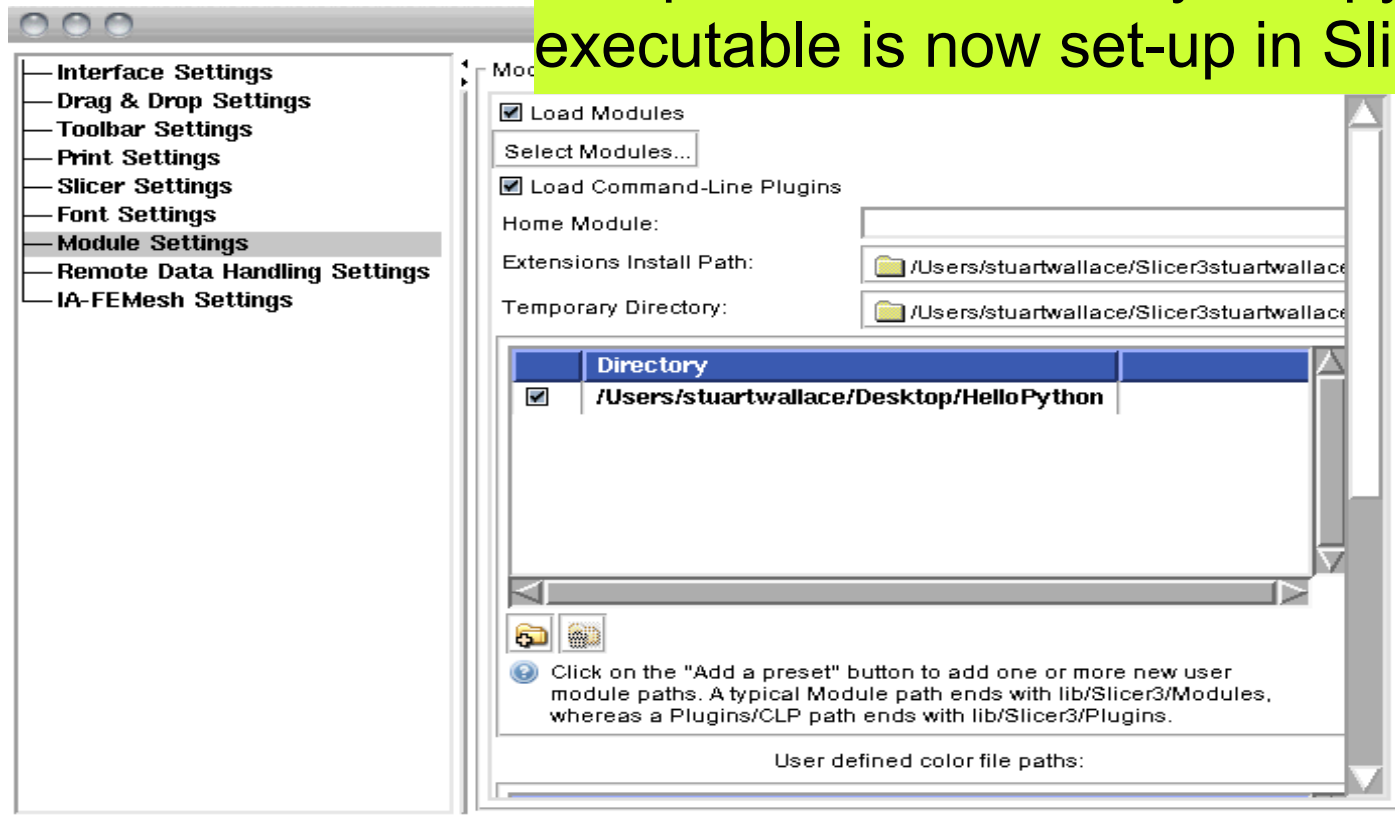
Click on the 'Add a preset' button, and enter the path to HelloPython.py



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 'Directory' list shows the path `/Users/stuartwallace/Desktop/HelloPython`. The 'Add a preset' button is located at the bottom left of the dialog box. The main 3DSlicer interface is visible in the background, showing the 'Python HelloPython' module settings and a 3D view of a brain scan.

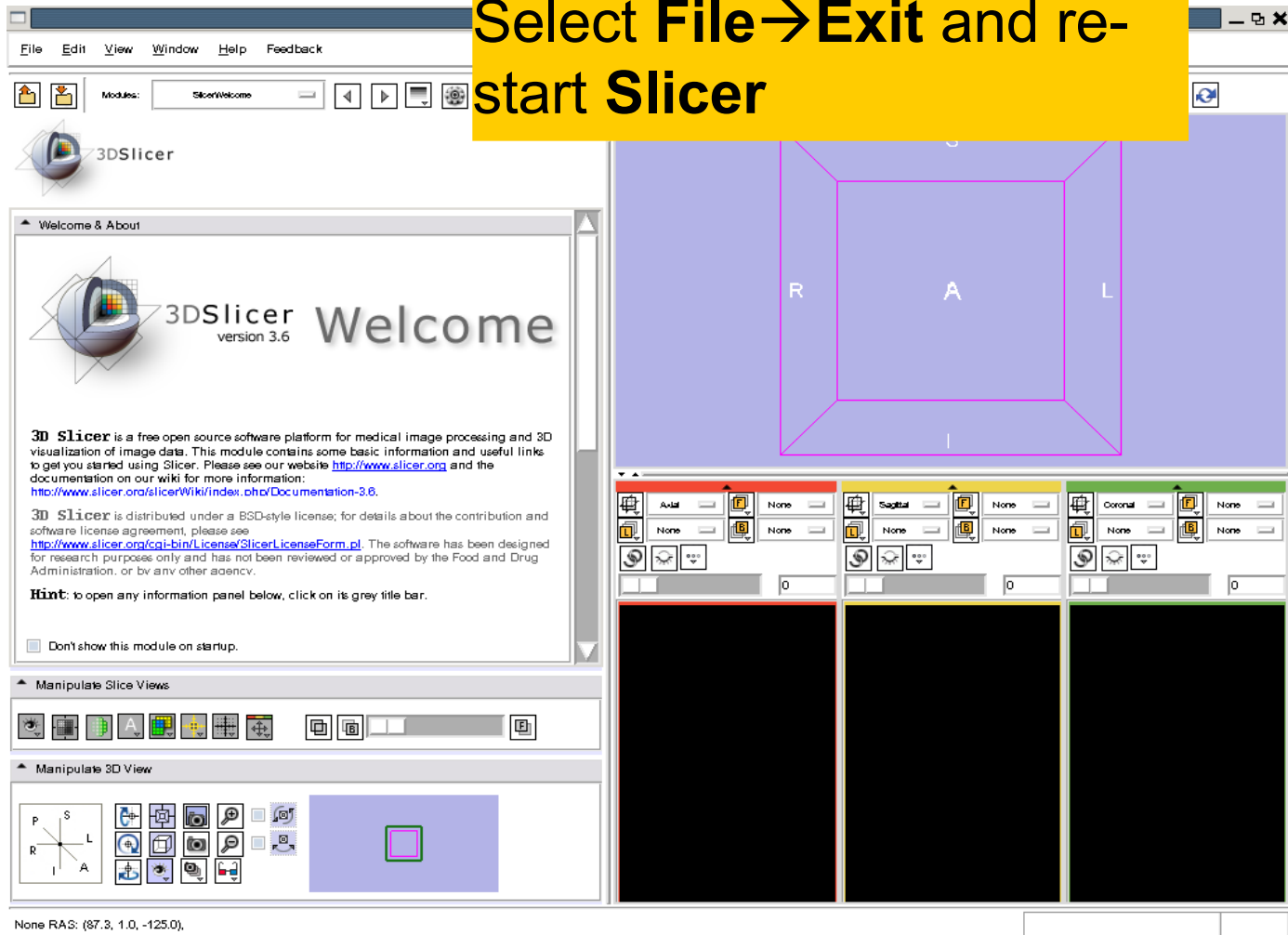
Integrating HelloPython to Slicer3

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



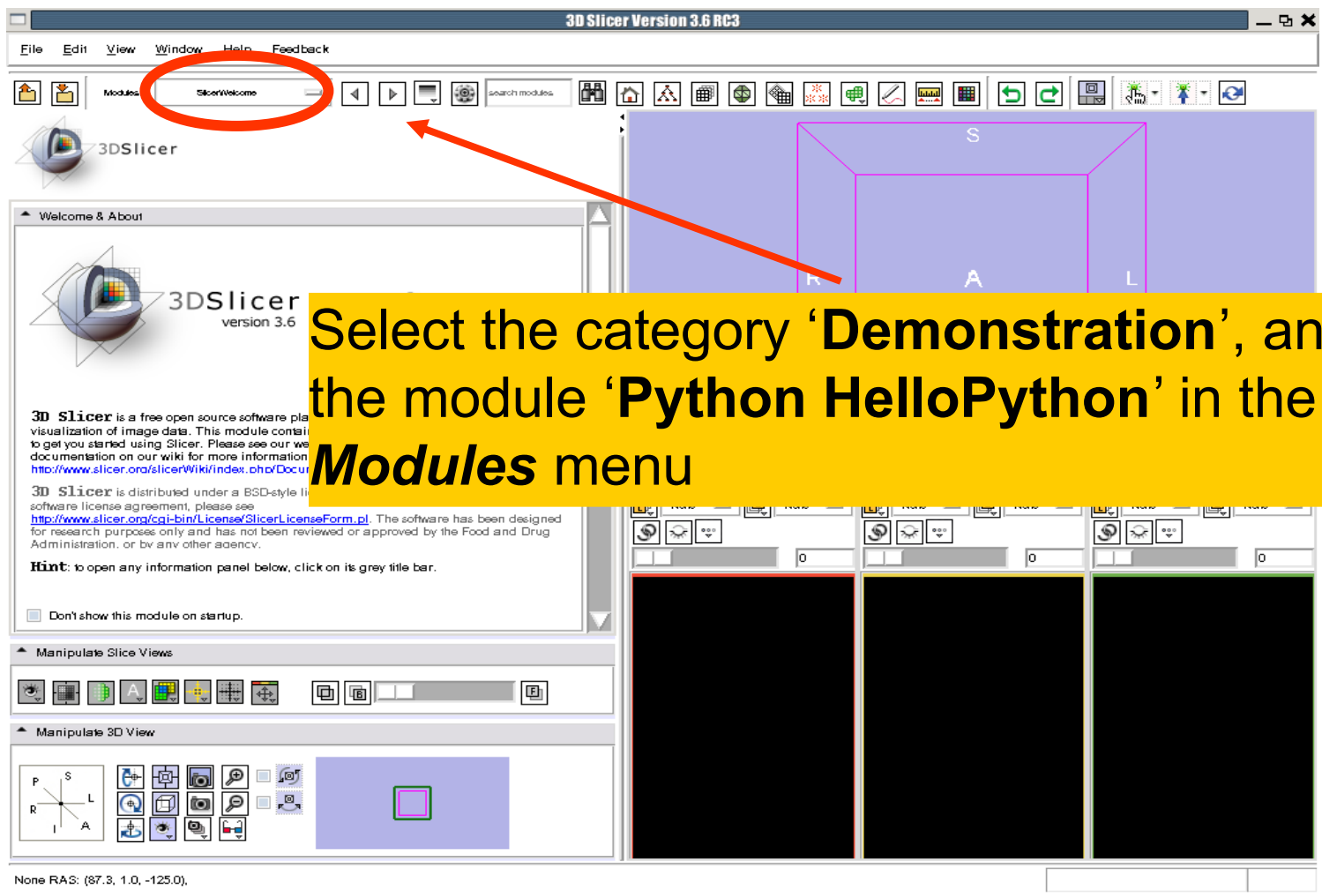
Click on the **Close** to exit the Application Settings window.

Integrating HelloPython to Slicer3

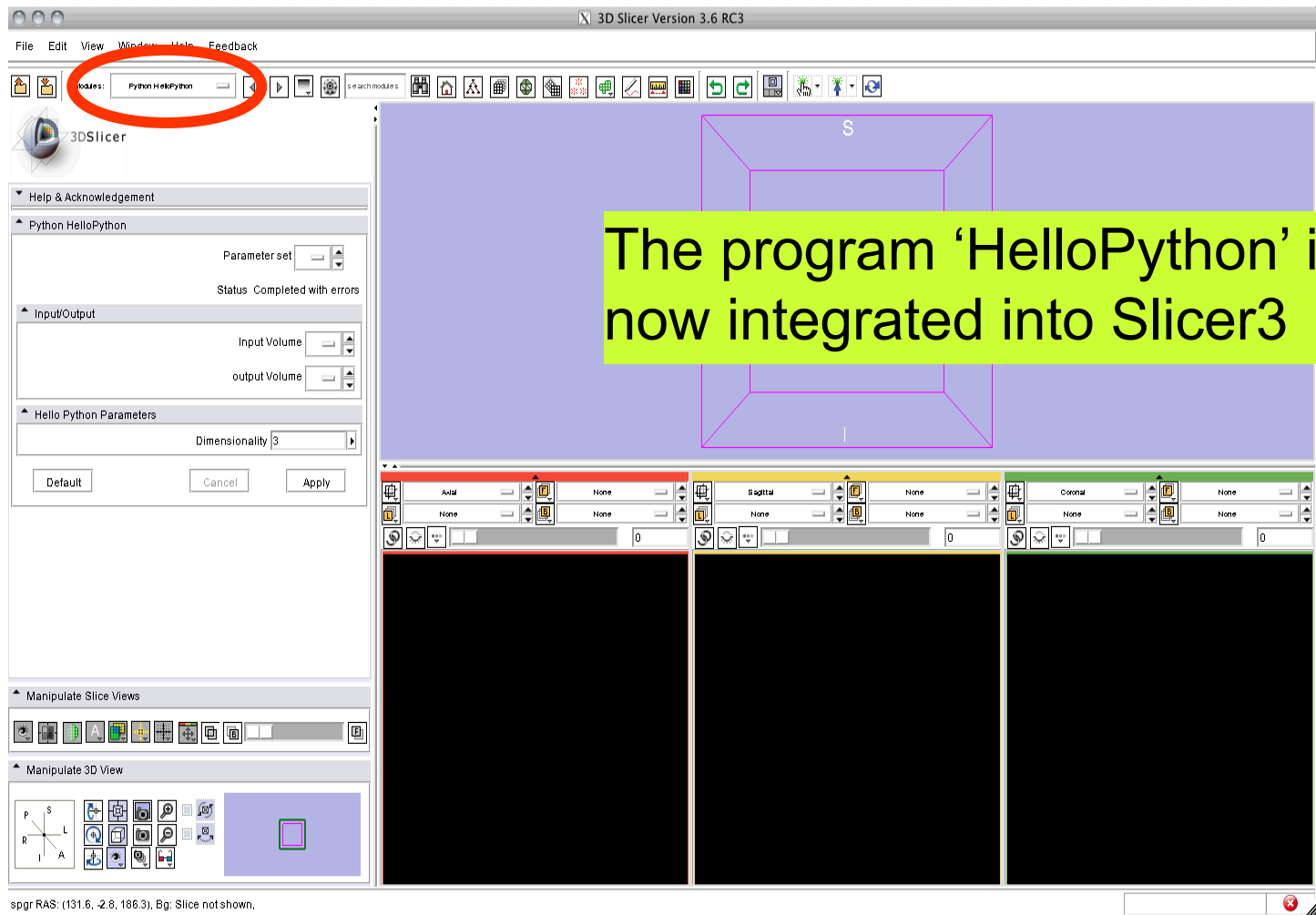


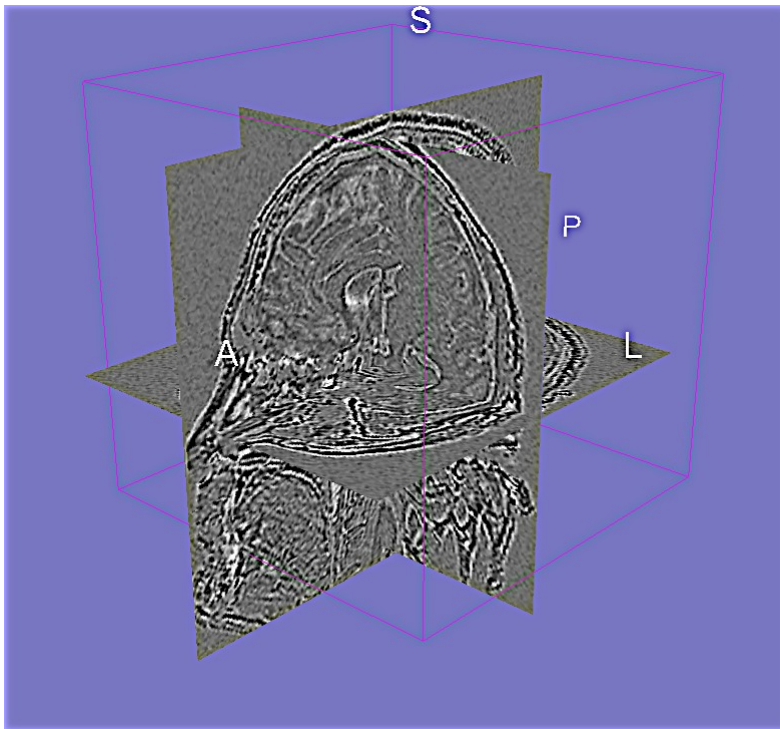


HelloPython module



HelloPython Module





Part B: Implementing the Laplace* Operator

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



Execute Function

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

```
    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>
```

```
<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
```

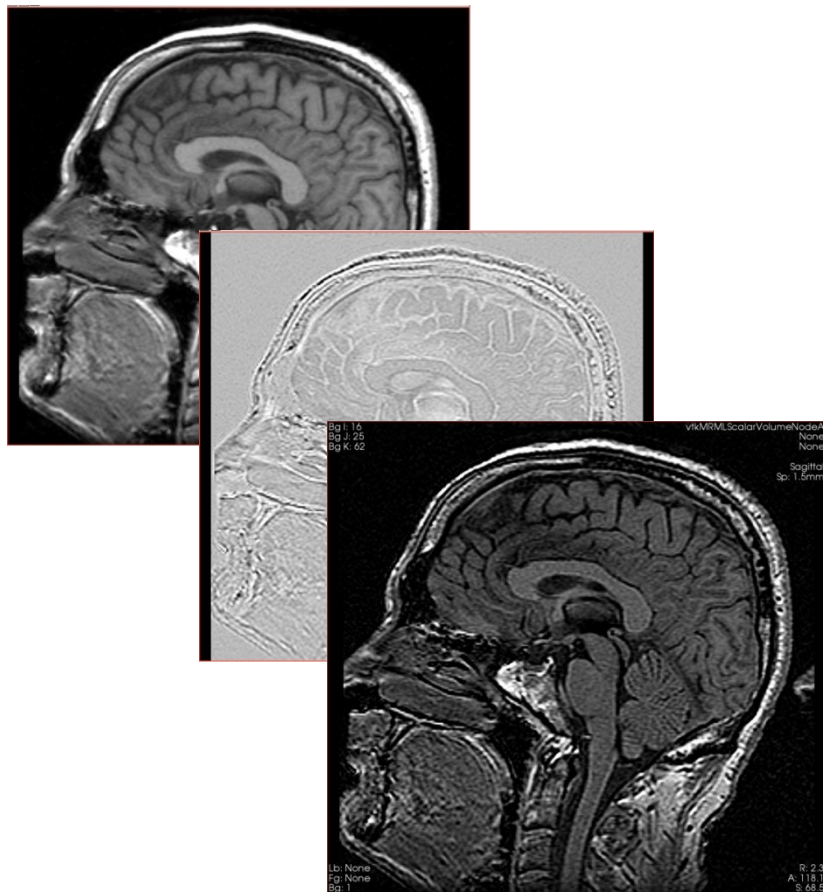
Place back the Laplacian of the image in the RAS reference system.



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

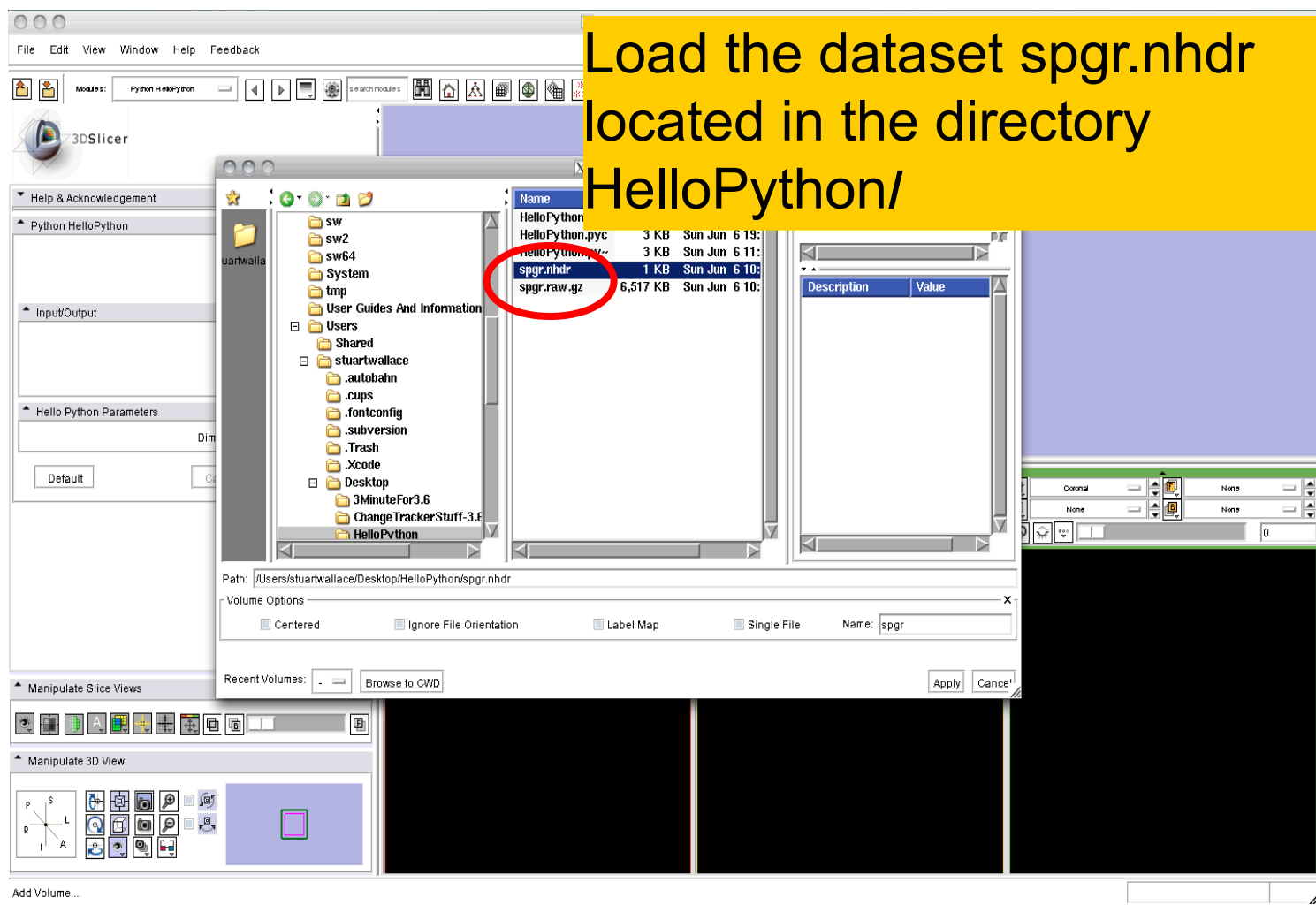


Running the Laplace Operator

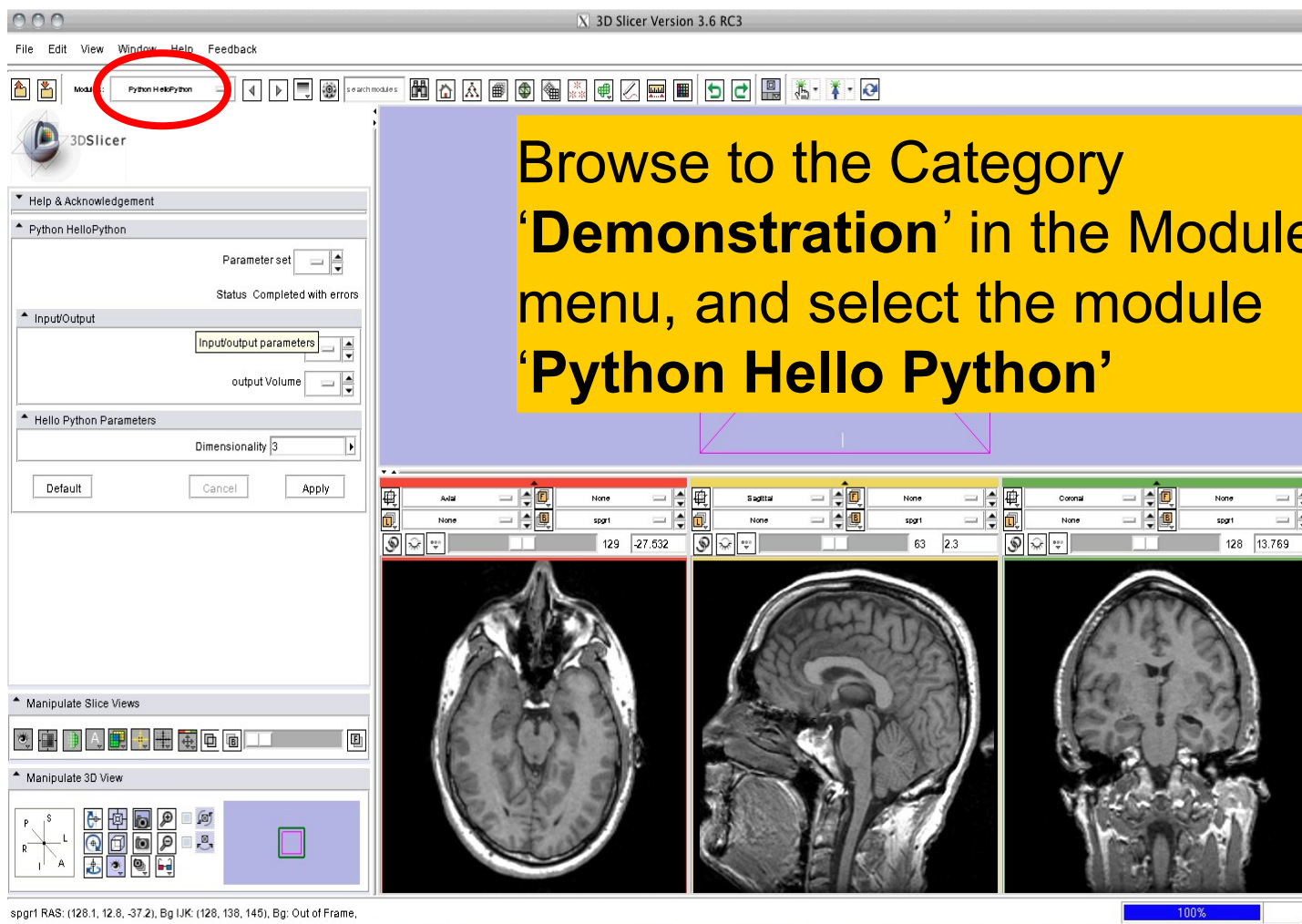
Re-start Slicer and select **File** → **Add Volume**

Running the Laplace Operator

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



Running the Laplace Operator

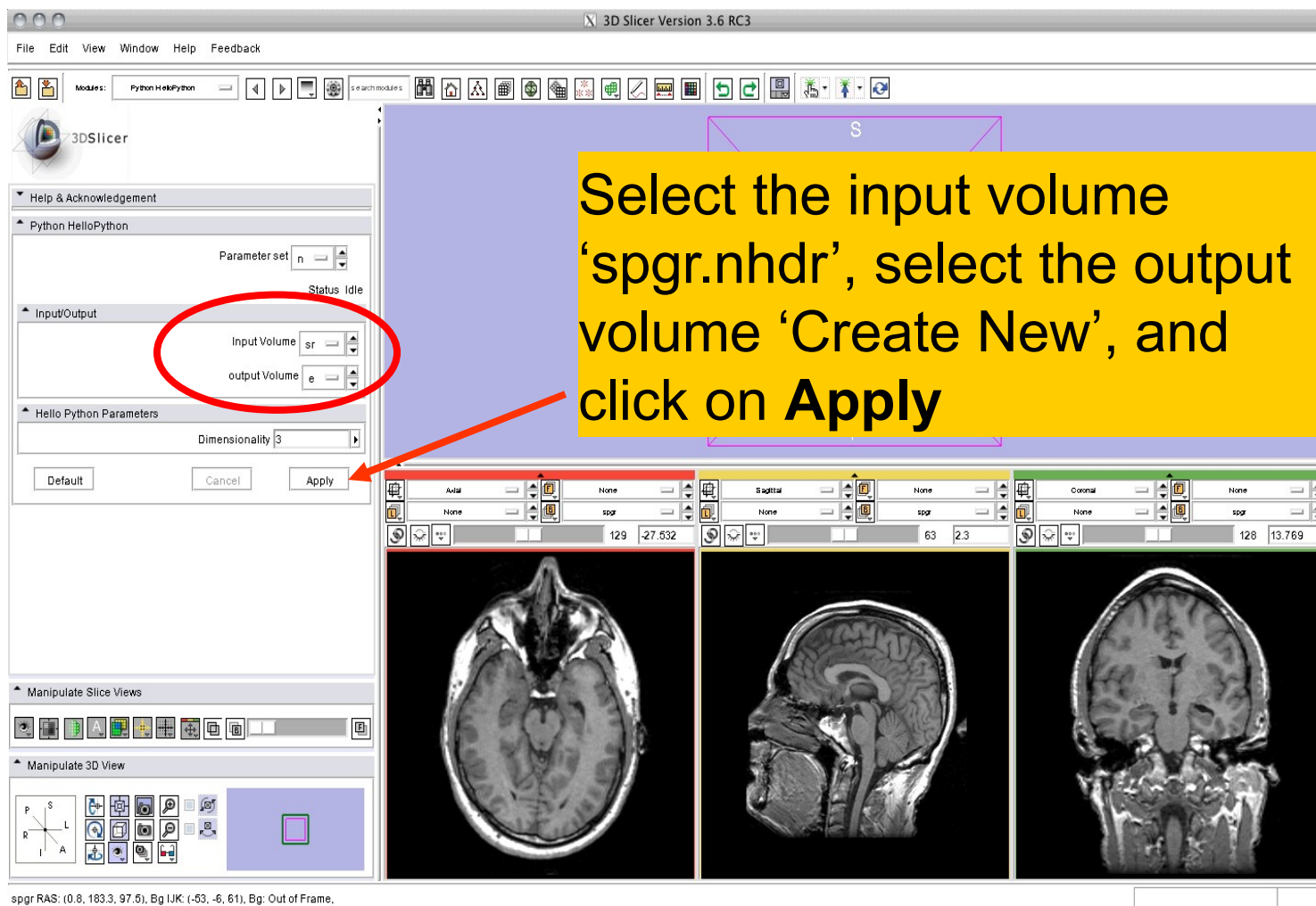


The screenshot shows the 3D Slicer 3.6 RC3 interface. The 'Modules' menu is open, and 'Python Hello Python' is highlighted with a red circle. A yellow text box on the right contains the following text:

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 panel on the left, which includes a 'Parameter set' dropdown, a 'Status' field indicating 'Completed with errors', and various input/output parameters. The main view displays three orthogonal MRI slices (Axial, Sagittal, and Coronal) of a brain scan, with a status bar at the bottom showing coordinates and a 100% zoom level.

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

Axial: None, spgr, 129, -27.632

Sagittal: None, spgr, 63, 2.3

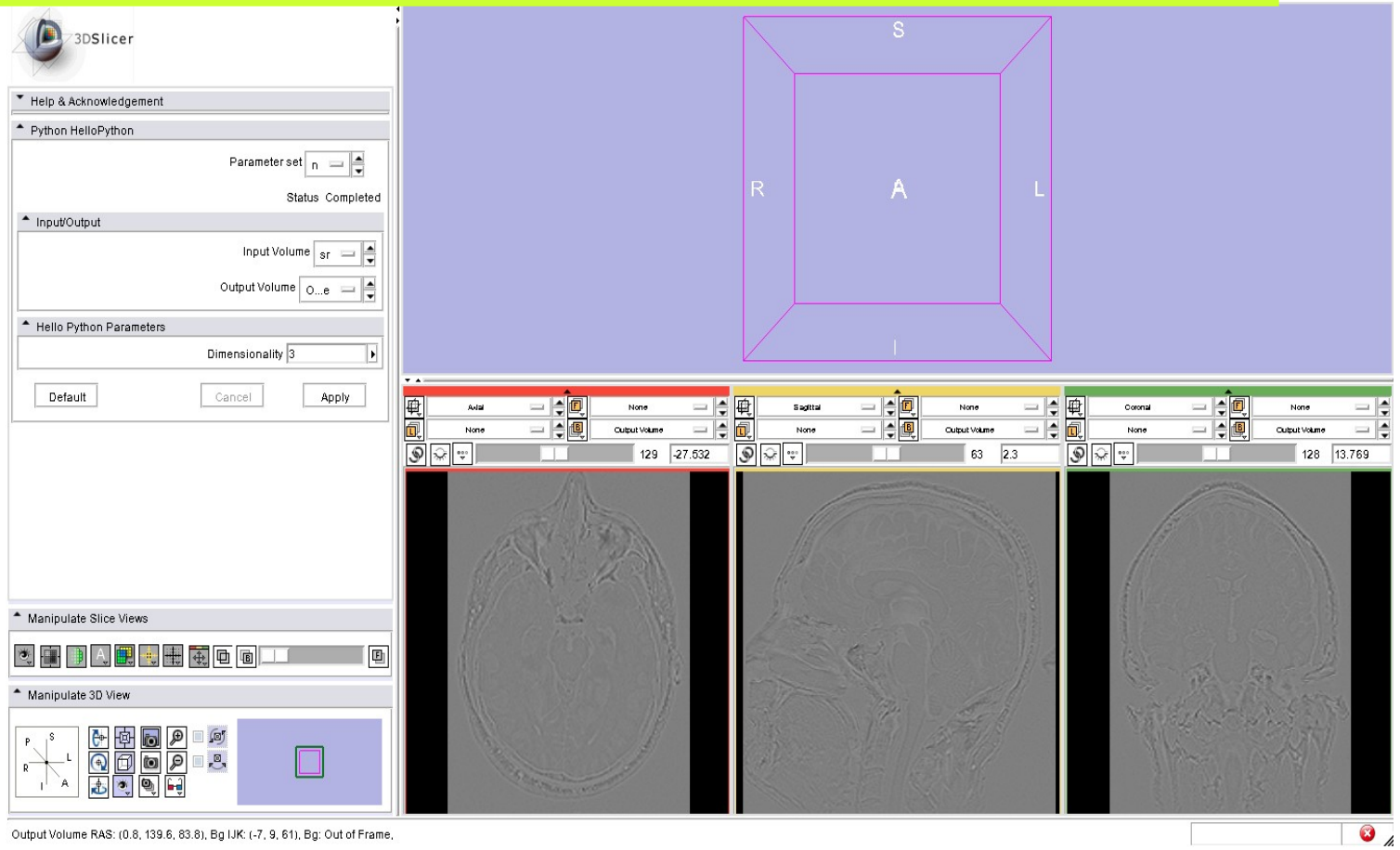
Coronal: None, spgr, 128, 13.769

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

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

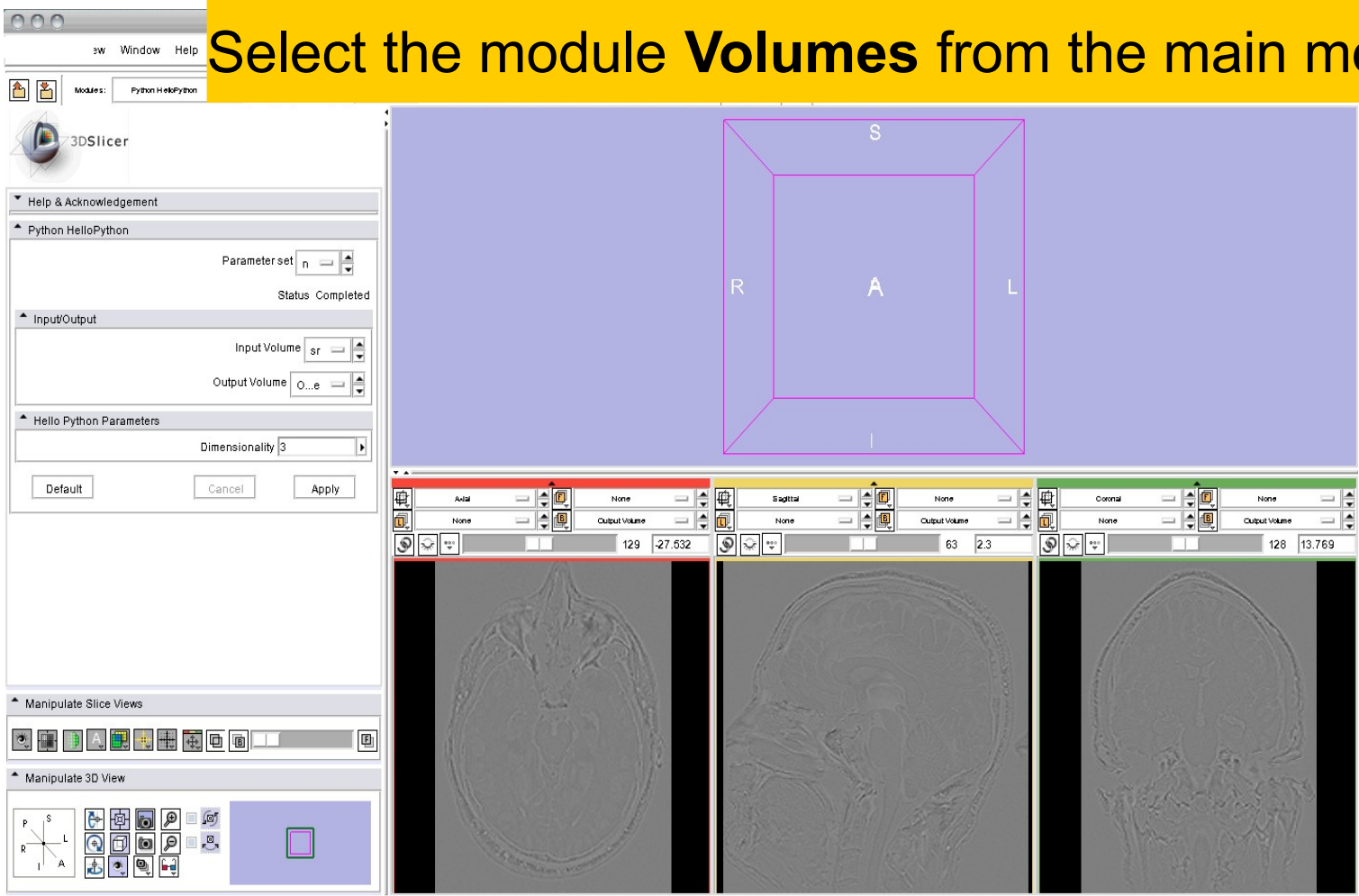
Running the Laplace Operator

Slicer displays the Laplacian of the spgr image.



Laplacian of the image

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



The screenshot displays the 3DSlicer software interface. On the left, the 'Python HelloPython' module is active, showing parameters for 'n' (set to 3), 'Input Volume' (sr), and 'Output Volume' (O...e). The 'Manipulate 3D View' panel shows a 3D perspective of the brain volume with a purple wireframe box indicating the current slice plane, labeled with 'S' (Superior), 'I' (Inferior), 'R' (Right), and 'L' (Left). The main window shows three orthogonal slice views: Axial, Sagittal, and Coronal. Each view displays the original brain image and its Laplacian (edge detection) result. The status bar at the bottom indicates the output volume coordinates: '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

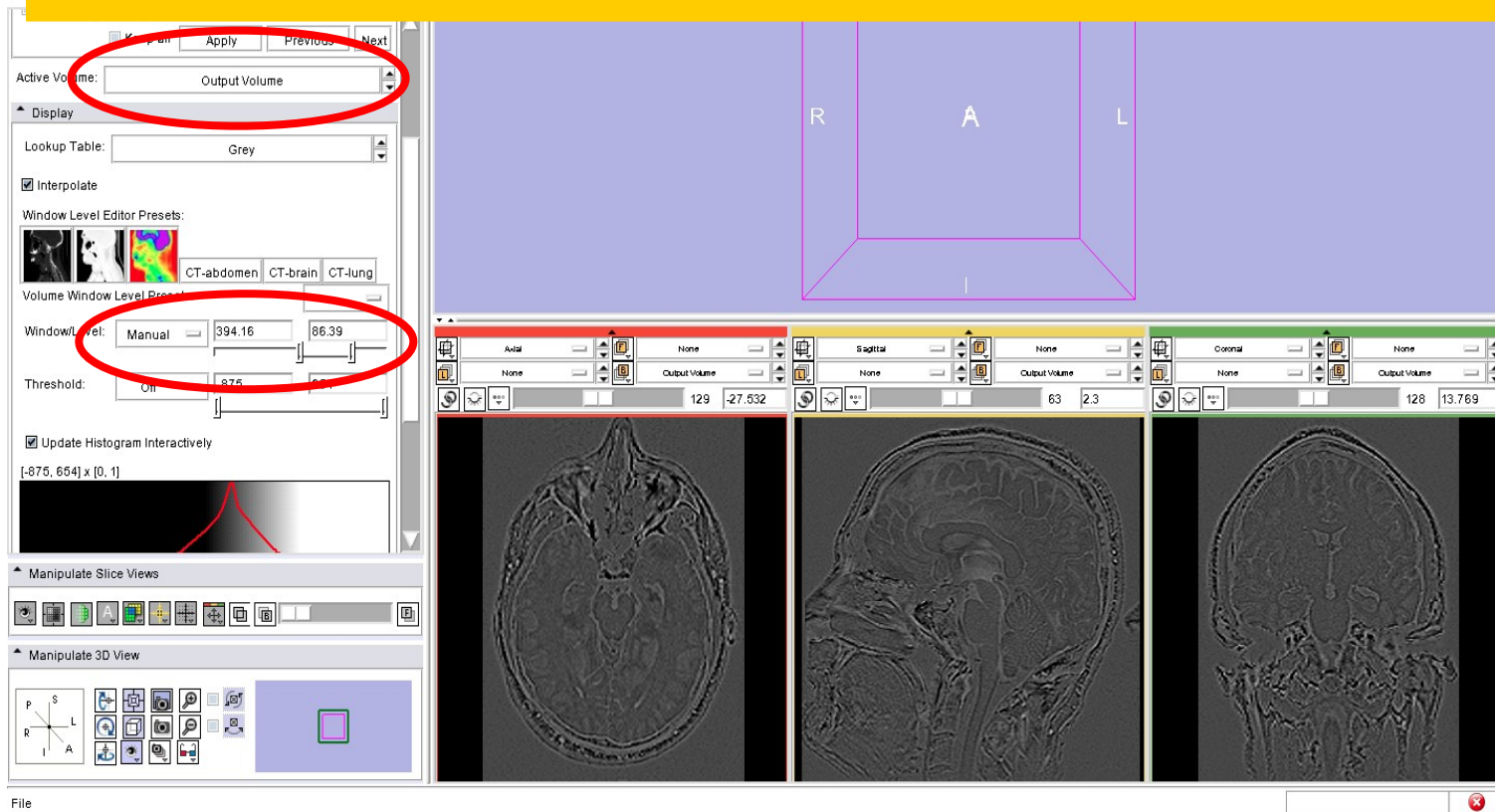
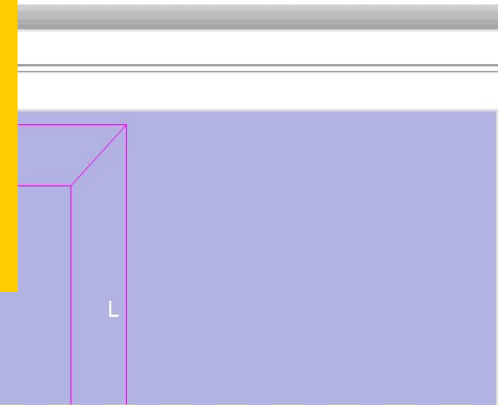


Image Sharpening

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)
```

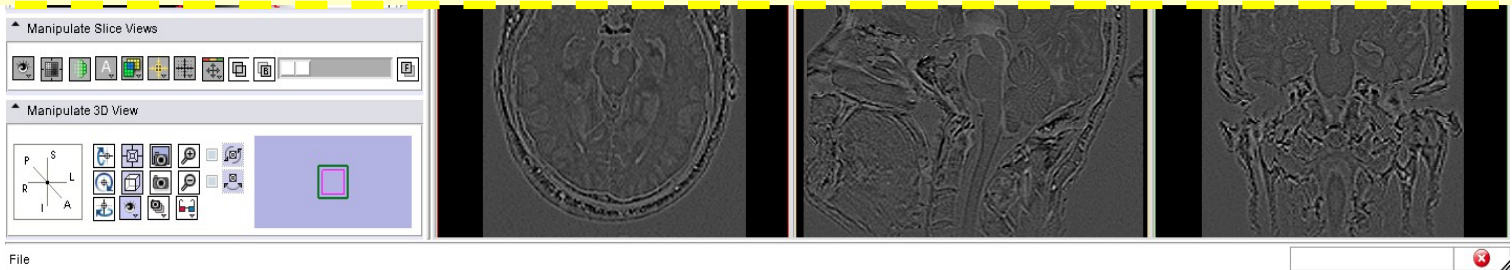
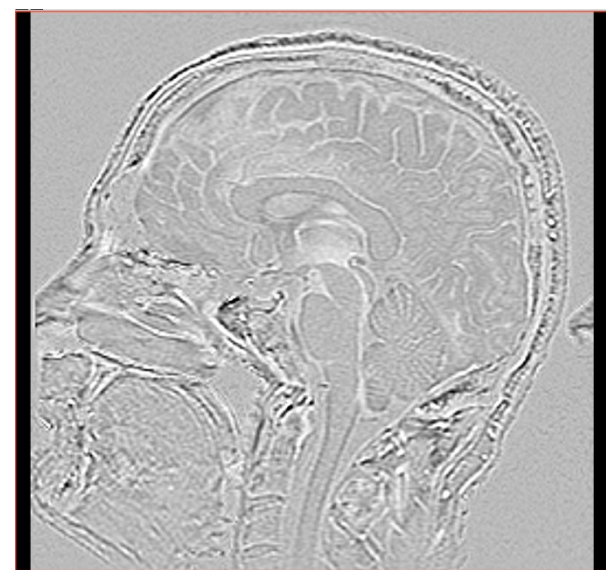


Image Sharpening

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