Python Scripting in 3D Slicer
Greg Sharp @ MGH
July 19, 2019
Disclosures

No conflict of interest
Disclaimer

“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”
Development in Slicer

• Slicer modules
  – Command line module
  – Scripted module
  – Loadable module

• Matlab bridge
Python scripting

- Python only
- Very easy and powerful
  - You can make a GUI
  - Or just run commands
- Included: numpy, VTK, CTK, PythonQt, SimpleITK
>>> t0 = slicer.util.loadVolume('t0.mha')
>>>
Python Interactor

Python 3.6.7 (default, Jul 10 2019, 09:34:30)
[GCC 6.3.0 20170516] on linux2

>>> Loading Slicer RC file [PATH/MYRC/0.slicerrc.py]

>>> pip install ('scipy')

Collecting scipy
  Downloading https://files.pythonhosted.org/packages/72/4c/5f81e7264b0a7a8bd570810f48cd346ba36fadb2ba255c873ad556de76/scipy-1.3.0-cp36-cp36m-manylinux1_x86_64.whl (25.2MB)
Requirement already satisfied: numpy>=1.13.3 in ./build/slicer-4/
Slicer-build/python-install/lib/python3.6/site-packages/
numpy-1.16.2-py3.6-linux-x86_64.egg (from scipy) (1.16.2)
Installing collected packages: scipy
Successfully installed scipy-1.3.0

>>> slicer.util.load

loadAnnotationFiducial()
loadAnnotationROI()
loadAnnotationRuler()
loadColorTable()
loadFiberBundle()
loadFiducialList()
loadLabelVolume()
MRML Nodes

- MRML Scene
- MRML Node
- Image, Markups, Segments, Etc.
- Attributes
- Display Node
- Storage Node
Help & Acknowledgement

Subject hierarchy | Transform hierarchy | All nodes

Nodes
- View1
- Red
- Yellow
- Green
- Camera
- SubjectHierarchy
- t0

Python Interactor
>>> t0 = slicer.util.loadVolume('t0.mha')
>>>
Python Interactor

```python
>>> t0 = slicer.util.loadVolume('t0.mha')
>>> t0
```

- AddAndObserveDisplayNodeID()
- AddAndObserveNodeReferenceID()
- AddAndObserveStorageNodeID()
- AddDefaultStorageNode()
- AddNodeReferenceID()
- AddNodeReferenceRole()
- AddObserver()
```python
>>> t0 = slicer.util.loadVolume('t0.mha')
>>> t0dn = t0.GetDisplayNode()
>>> t0dn.AutoScaleWindowOff()
>>> t0dn.SetWindow(50)
>>> t0dn.SetLevel(100)
>>> t0
```
Python Interactor

```python
>>> t0 = slicer.util.loadVolume('t0.mha')
>>> t0dn = t0.GetDisplayNode()
>>> t0dn.AutoScaleWindowLevelOff()
>>> t0dn.SetWindow(50)
>>> t0dn.SetLevel(100)
>>> t0dn.AutoScaleWindowLevelOn()
```
Python Interactor

```python
>>> a = slicer.util.array('t0')
>>> a[:] = a * 20
>>> t0.Modified()
```
Python Interactor

```python
>>> a = slicer.util.array('t0')
>>> a[:] = a * 20
>>> t0.Modified()
>>> slicer.mrmlScene.Clear(0)
>>>```

```
Node references

- Nodes can refer to other nodes using the concept of a node reference
# Create model node
model = slicer.vtkMRMLModelNode()

# Create model node
transform = slicer.vtkMRMLLinearTransformNode()

# Transform model by reference to transform node
model.SetAndObserveTransformNodeID(transform.GetID())
Node attributes
SimpleITK

- An easy way to use ITK functions in python
- But, requires extra step to move images between SimpleITK and Slicer
Python Interactor

```python
>>> import SimpleITK
>>> import sitkUtils
>>> input = sitkUtils.PullVolumeFromSlicer('MRBrainTumor1')
>>> output = SimpleITK.DiscreteGaussian(input, 1.0, 5)
>>>```

>>> node = sitkUtils.PushVolumeToSlicer(output)
>>>
Python Interactor

```python
>>> node = sitkUtils.PushVolumeToSlicer(output)
>>> slicer.util.setSliceViewerLayers(node)
>>> 
```
class ExampleModule(ScriptedLoadableModule):
    def __init__(self, parent):
        ScriptedLoadableModule.__init__(self, parent)
        self.parent.title = "Hello Python"
        self.parent.categories = ["Examples"]
        self.parent.dependencies = []
        self.parent.contributors = ["Jean-Christophe ..."]
        self.parent.helpText = ...

class ExampleModuleWidget(ScriptedLoadableModuleWidget):
    def __init__(self, parent = None):
        ScriptedLoadableModuleWidget.__init__(self, parent)
        self.logic = ExampleModuleLogic()

    def setup(self):
        ScriptedLoadableModuleWidget.setup(self)
        inputCollapsibleButton = ctk.ctkCollapsibleButton()
        inputCollapsibleButton.text = "A Collapsible Button"
        self.layout.addWidget(inputCollapsibleButton)
Python modules

Common module attributes

Module UI
Calling Slicer modules

- Slicer modules can be called from python
- CLI modules follow this pattern:
  - Create any needed output nodes
  - Set parameters
  - Call CLI function
Python Interactor

```
>>> parameters = {}
>>> parameters["plmslc_fixed_volume"] = getNode('t0');
>>> parameters["plmslc_moving_volume"] = getNode('t5');
>>> out_vol = slicer.vtkMRMLScalarVolumeNode()
>>> slicer.mrmlScene.AddNode(out_vol)
>>> (vtkCommonCorePython.vtkMRMLScalarVolumeNode)0x7ff36e646b48
>>> parameters["plmslc_output_warped"] = out_vol
>>> parameters["enable_stage_2"] = True
>>> ```
>>> slicer.cli.run(slicer.modules.plastimatch_slicer_b spline,None,parameters)
(vtkCommonCorePython.vtkMRMLCommandLineModuleNode) 0x7ff36e546870

>>>
```python
>>> node = slicer.mrmlScene.GetFirstNodeByClass('vtkMRMLMarkupsNode')
>>> node.GetNumberOfControlPoints()
300
>>> ```
Python Interactor

```python
>>> position=[0,0,0]
>>> node.GetNthControlPointPosition(249, position)
>>> position
[-65.96, -112.52, -115.0]
```
```python
>>> seg_node = slicer.mrmlScene.GetFirstNodeByClass('vtkMRMLSegmentationNode')
>>> seg = seg_node.GetSegmentation()
>>> for i in range(0, seg.GetNumberOfSegments()):
...    seg.GetNthSegment(i).GetName()
...    'Insert'
...    'External'
...    'GTV'
...    'OAR'
>>> ```
Python Interactor

```python
>>> s = seg.GetNthSegment(1)
>>> representations = []
>>> s.GetContainedRepresentationNames(representations)
>>> representations
['Closed surface', 'Planar contour']
```
```python
>>> def onMouseMoved(observer, eventid):
...     ras=[0,0,0]
...     crosshairNode.GetCursorPositionRAS(ras)
...     print(ras)
...
>>> crosshairNode=slicer.util.getNode('Crosshair')
>>> crosshairNode.AddObserver(slicer.vtkMRMLCrosshairNode.CursorPositionModifiedEvent, onMouseMoved)
15
>>> 
```
Python Interactor

[-135.52327037773358, -179.21376739562623, -114.99999999999999]
[-136.51062624254473, -180.20112326043736, -114.99999999999999]
[-136.51062624254473, -180.20112326043736, -114.99999999999999]
[-137.00430417495028, -180.68480119284293, -114.99999999999999]
[-137.00430417495028, -180.68480119284293, -114.99999999999999]
[-137.49798210735585, -181.18847912524848, -114.99999999999999]
[-137.49798210735585, -181.18847912524848, -114.99999999999999]
[-141.4474055666004, -181.68215705765408, -114.99999999999999]
[-141.4474055666004, -181.68215705765408, -114.99999999999999]
[-149.839930417495, -184.64422465208747, -114.99999999999999]
[-149.839930417495, -184.64422465208747, -114.99999999999999]
[-169.933697813121, -190.07468190854868, -114.99999999999999]
[-169.933697813121, -190.07468190854868, -114.99999999999999]
[-195.7519781312172, -202.41663021868786, -114.99999999999999]
[-195.7519781312172, -202.41663021868786, -114.99999999999999]
[-220.43587475149104, -212.78386679920476, -114.99999999999999]
[-220.43587475149104, -212.78386679920476, -114.99999999999999]
[-220.43587475149104, -212.78386679920476, -114.99999999999999]
[-220.43587475149104, -212.78386679920476, -114.99999999999999]
Python scripting

- As of June 2019, python binary packages can be used
Python Interactor

Python 3.6.7 (default, Jul 10 2019, 09:34:30)
[GCC 6.3.0 20170516] on linux2

>>> Loading Slicer RC file [/PHS/home/gcs6/.slicerrc.py]
>>> pip_install('scipy')
Collecting scipy
  Downloading https://files.pythonhosted.org/packages/72/4c/
  5f81e7264b0a7a8bd570810f48cd346ba36faedbd2ba255c873ad556de76/
  scipy-1.3.0-cp36-cp36m-manylinux1_x86_64.whl (25.2MB)
Requirement already satisfied: numpy>=1.1.3 in /build/slicer-4/
Slicer-build/python-install/lib/python3.6/site-packages/
numpy-1.16.2-py3.6-linux-x86_64.egg (from scipy) (1.16.2)
Installing collected packages: scipy
Successfully installed scipy

>>>
New Developers: Welcome! Check these instructions!

Getting involved
Start Here
- New community member checklist
- Developer meetings
  - It is open to everyone, feel free to join.
- Discussion Forum
  - The most effective way to get help from the community
- FAQ
  - Set of common development questions/answers

Resources
- Roadmap
- Source code API documentation
- Source code repository
  - Github
- CMake Quality Dashboard: Slicer-Stable, Slicer-Preview
  - Nightly, continuous and experimental dashboards
- Bug tracking system
  - Configuration
- Slicer Style Guidelines
  - Consistency and readability for a manageable code base
- Script repository
  - Collection of python scripts manipulating various Slicer components
- Change logs and release details
- Information about Slicer releases
- List Slicer resources and who to contact in case of problem

Build instructions
Create Slicer extensions
- Build, test, package and distribute extensions
Build Slicer application
- Compiling and installing Slicer from source
Build Module
- Compiling slicer modules outside of the slicer source tree

How-to's

Migration Guide
- How to update Slicer extension code following breaking changes in Slicer or dependent tools like ITK, MI, Qt...

Source code Howtos
- How to checkout the source and contribute patches

Module Howtos
- How to create or build modules, how to add or update remote modules

Testing Howtos
- How to write tests and manage testing data

Quick Links

Stable Dashboard
- Preview Dashboard
- Source
- C++ API

Contribute
- Style Guide
- Issue Tracker

Wiki Cheat Sheet, Wiki Editing Guide

Design & Implementation

Modules
- Comparison between the different supported module types

Data Model / MRML
- Objects and their organization. MRML library provides API for managing medical image data type

Logos
- Details the role of MRML/Slicer/Applications/Modules logos and Displayable managers

Layouts
- How to control the layout of the views

Slice Orientation Presets
- How to manage slice orientation presets

Slicelets
- Create simple standalone applications (slicelets)

Qt Mechanism
- How to read or write nodes from Numpy

Python Scripting
- Presents the underlying infrastructure

Charts
- Description of the Charting (qPlot) architecture

Plots
- Description of the Plotting (VTK) architecture

Directory Structure
- Files location in the build and install tree

Qt Plugins
- How to build and load Qt plugins

Build system / Release process / Factory description
- Details how Slicer is built and packaged

Testing
- Testing framework to test Slicer application. It complements unit tests
## Python script repository

More than 100 simple examples

### Documentation

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.1</td>
<td>Network</td>
</tr>
<tr>
<td>1.1.2</td>
<td>Image</td>
</tr>
<tr>
<td>1.1.3</td>
<td>Image Filtering</td>
</tr>
<tr>
<td>1.1.5</td>
<td>Image Alignment</td>
</tr>
</tbody>
</table>

### References

1.2.1 Save the script into a single .txt file
1.2.2 Center the 3D view on the Scene
1.2.3 Save the 3D view
1.2.4 Display text in a 3D view or slice view
1.2.5 Make slice viewing annotations (DataProbe)
1.2.6 Turn off interoposition
1.2.7 Custom observer layout
1.2.8 Customize keyboard shortcuts
1.2.9 Disable certain viewer interactions in slice views
1.2.10 Change default slice view orientation
1.2.11 Set all slice views to be transparent
1.2.12 Change a slice view inside the view layout
1.2.13 Show a 3D view outside the view layout
1.2.14 Running as TKinter in Python using Cytkinter
1.2.15 Get current mouse coordinates in a slice view
1.2.16 GetDataNode tool
1.2.17 Get a reformatted image from a slice viewer as an numpy array
1.2.18 Think back reconstruction and mean-maximum intensity volume projections
1.2.19 Change default file type for nodes that have never been saved yet
1.2.20 Change the type for converting all volumes with existing storage nodes

### Sequences

- 2.1.1 Consider all sequences in the scene into a new sequence

### Segmentation

- 2.5.1 Create a segmentation from a legend map and display in 3D
- 2.5.2 Export legend map node from segmentation node
- 2.5.3 Export legend node from segmentation node
- 2.5.4 Show a segmentation in 3D
- 2.5.5 Get a representation of a segment
- 2.5.6 Convert all segments using default path and conversion parameters
- 2.5.7 Convert all segments using custom path or conversion parameters
- 2.5.8 Rename, using a modified conversion parameter
- 2.5.9 Set central of a segment in world RMS coordinates
- 2.5.10 How to cut segmentation edgel with effects from a script

### Accessing views, renderers, and cameras

- 2.7.1 Hide views, cut plane, and cameras
- 2.7.2 Hide views, cut plane, and cameras
- 2.7.3 Hide views, cut plane, and cameras
- 2.7.4 Hide views, cut plane, and cameras
- 2.7.5 Hide views, cut plane, and cameras
- 2.7.6 Hide views, cut plane, and cameras
- 2.7.7 Hide views, cut plane, and cameras
- 2.7.8 Hide views, cut plane, and cameras
- 2.7.9 Hide views, cut plane, and cameras
- 2.7.10 Hide views, cut plane, and cameras

### More than 100 simple examples
Welcome to the Slicer forum!

Slicer is an open-source extensible software platform for image visualization and analysis. Slicer has a large community of users in medical imaging and surgical navigation, and is also used in fields such as astronomy, paleontology, and 3D printing.

- Click here for answers to common questions
- Click here for tutorials and documentation
- Click here to post a new question.

To receive updates about new software releases and major events, sign-in with Google, GitHub, Facebook, or email.

<table>
<thead>
<tr>
<th>Category</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Announcements</td>
<td>1/week</td>
</tr>
<tr>
<td>Low-traffic category for 3D Slicer, extension, and community news and announcements.</td>
<td></td>
</tr>
<tr>
<td>- Release Notes</td>
<td></td>
</tr>
<tr>
<td>- Jobs</td>
<td></td>
</tr>
<tr>
<td>Support</td>
<td>36/week</td>
</tr>
<tr>
<td>The Support category is for all usage questions and general discussion of Slicer and extensions.</td>
<td></td>
</tr>
<tr>
<td>- Feature requests</td>
<td></td>
</tr>
<tr>
<td>Development</td>
<td>9/week</td>
</tr>
<tr>
<td>The Development category is for discussion of Slicer application and extension programming, software testing, and related topics - similarly to the former Slicer-devel mailing list.</td>
<td></td>
</tr>
<tr>
<td>- SlicerCMF</td>
<td></td>
</tr>
<tr>
<td>- SlicerSALT</td>
<td></td>
</tr>
<tr>
<td>- SlicerDMRI</td>
<td></td>
</tr>
<tr>
<td>- ProjectWeek</td>
<td></td>
</tr>
<tr>
<td>- SlicerCIP</td>
<td></td>
</tr>
<tr>
<td>- Hangout</td>
<td></td>
</tr>
<tr>
<td>Community</td>
<td>2/week</td>
</tr>
<tr>
<td>Community information and project/topic sub-forums.</td>
<td></td>
</tr>
<tr>
<td>- SlicerCMF</td>
<td></td>
</tr>
<tr>
<td>- SlicerSALT</td>
<td></td>
</tr>
<tr>
<td>- SlicerDMRI</td>
<td></td>
</tr>
<tr>
<td>- ProjectWeek</td>
<td></td>
</tr>
<tr>
<td>- SlicerCIP</td>
<td></td>
</tr>
<tr>
<td>- Hangout</td>
<td></td>
</tr>
</tbody>
</table>

| Latest                     |        |
| Welcome to the 3D Slicer Forum! |
| - Announcements            | Aug 18 |
| Simplify Slicer or Remove some modules from Slicer |
| - Support                  | 35m    |
| Update 3D view if node changes |
| - Development              | 1h     |
| rebuild the surface of skull bone |
| - Support                  | 1h     |
| Ensegmentation cannot found |
| - Support                  | 1h     |
| Slicer Tractography ROI Selection Not Working |
| - Support                  | 1h     |
| Python codes for diffusion module |
| - Development              | 3h     |
Links

- https://www.slicer.org/
- https://discourse.slicer.org/