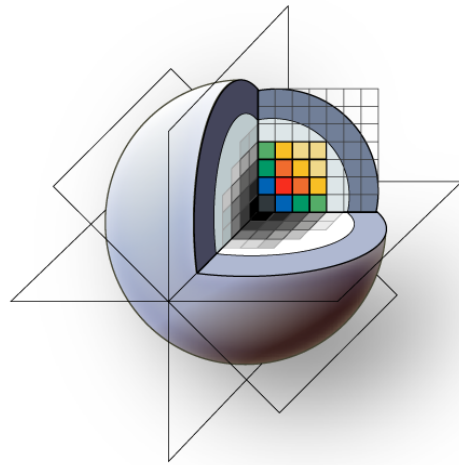
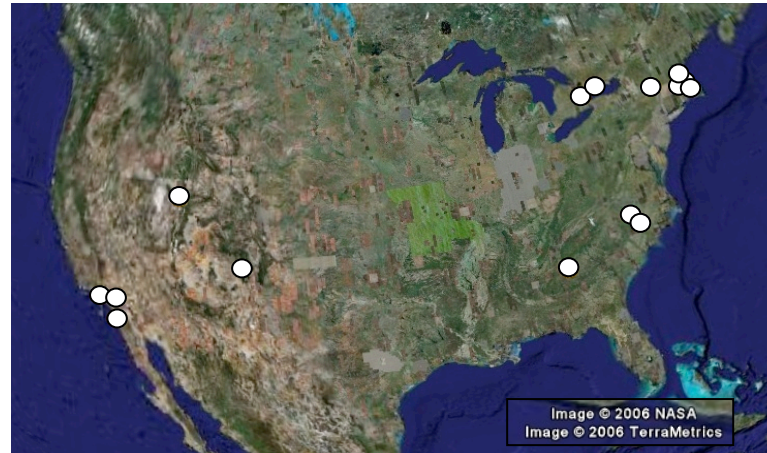
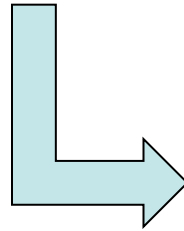
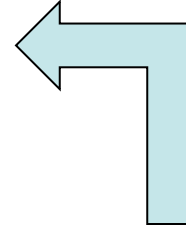




Integrating with Slicer3



3DSlicer





NA-MIC Kit Goals

- Software and Methodologies for Medical Image Computing
 - Facilitate Research
 - Promote Interoperability
- Stable, Cross-Platform Run Time Environment
 - Full Set of Core Features
 - Avoid Duplicated Effort
- Flexible Module Architecture
 - Plug-ins should be As Simple As Possible

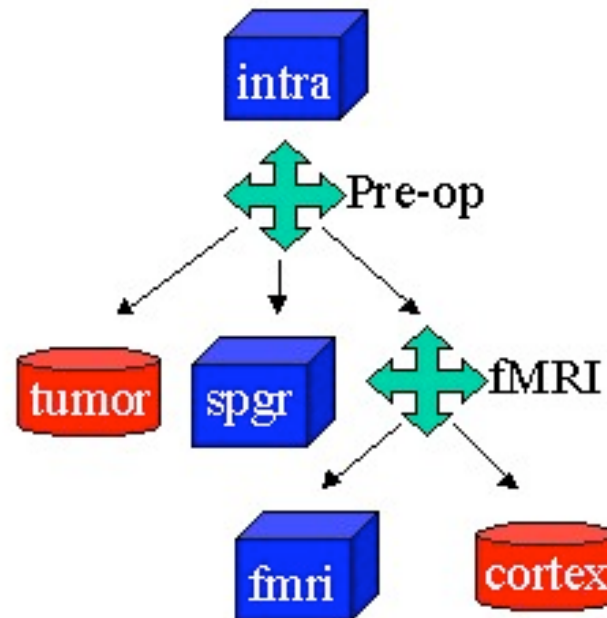


Slicer3 data model

MRML File:

Volume:	intra
Transform:	Pre-op
Volume:	spgr
Model:	tumor
Transform:	fMRI
Volume:	fmri
Model:	cortex

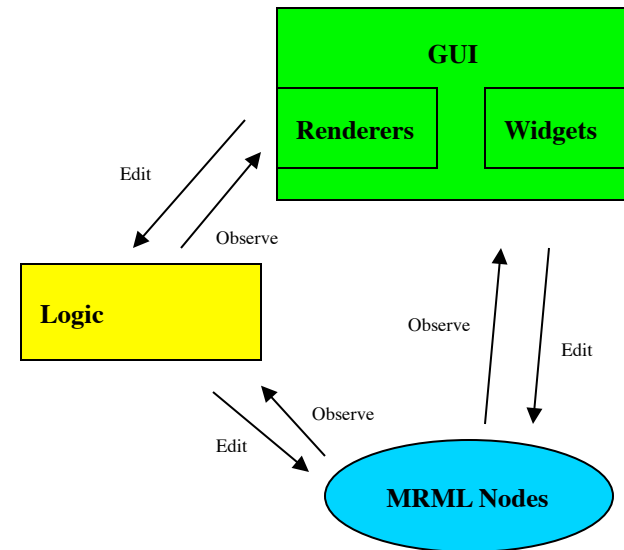
MRML Tree:





Slicer3 “Observer MVC” Pattern

- **MRML** (Model)
 - For Scene Description and Application State
 - MRML Nodes are Persistent and Undoable
 - Scene and Nodes are Observable
- **Logic** Encapsulate VTK and ITK Pipelines (Controller)
 - Observe MRML to Configure Pipelines
 - Help Create/Manage Nodes
 - No UI Components (no Widgets, Actors, Mappers, Renderers or RenderWindows)
- **GUI** (View)
 - Observe and Edit MRML
 - Interact with User and Display Hardware
- Modules Should Follow Same Conventions



“**Observe**” means generic event mechanisms are used to pass information.

“**Edit**” means code can directly call methods.

Example: GUI can call methods in Logic classes, but Logic *cannot* call GUI methods.

MRML *cannot* call Logic or GUI methods.

There can be many observers for any event.



Variety of levels of integration

Slicer Libs	<ul style="list-style-type: none">• ModuleDescriptionParser• GenerateCLP• vtkITK• MRML	Non-slicer specific support libraries
Slicer Base	<ul style="list-style-type: none">• Application logic• Widgets	Common infrastructure for Slicer applications
Built in modules	<ul style="list-style-type: none">• Slice viewers• Models• Fiducials• Transforms	Full access to Slicer internals
Loadable modules	<ul style="list-style-type: none">• Query Atlas• QDEC• Volume rendering• ChangeTracker• EMSegment	Full access to Slicer internals
Scripted modules	<ul style="list-style-type: none">• Editor• Teem Two Tensor Tractography• VMTK	Limited access to Slicer internals
Command line modules	<ul style="list-style-type: none">• Registration	Restricted access to Slicer internals
Daemon	<ul style="list-style-type: none">• OpenIGTLink• Stochastic Tractography	Access to MRML



Base Features vs. Modules

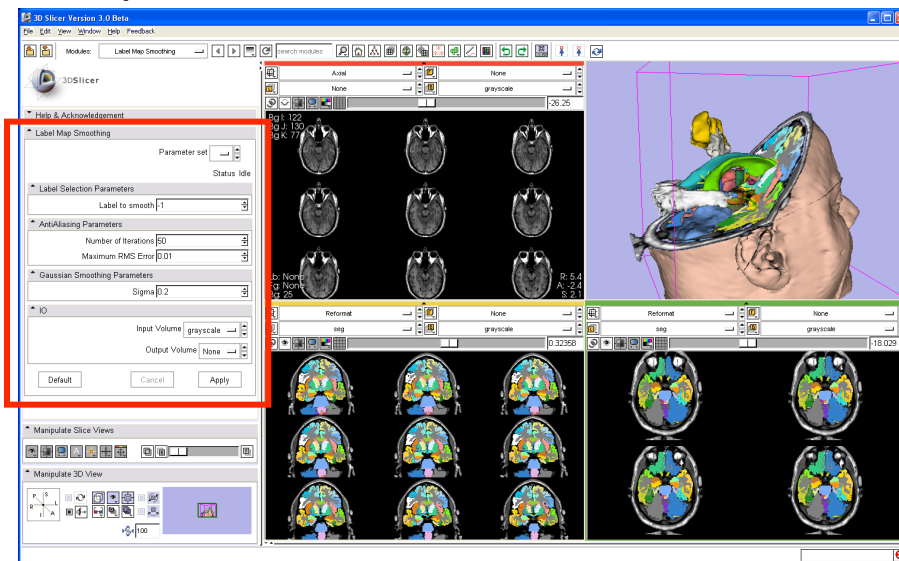
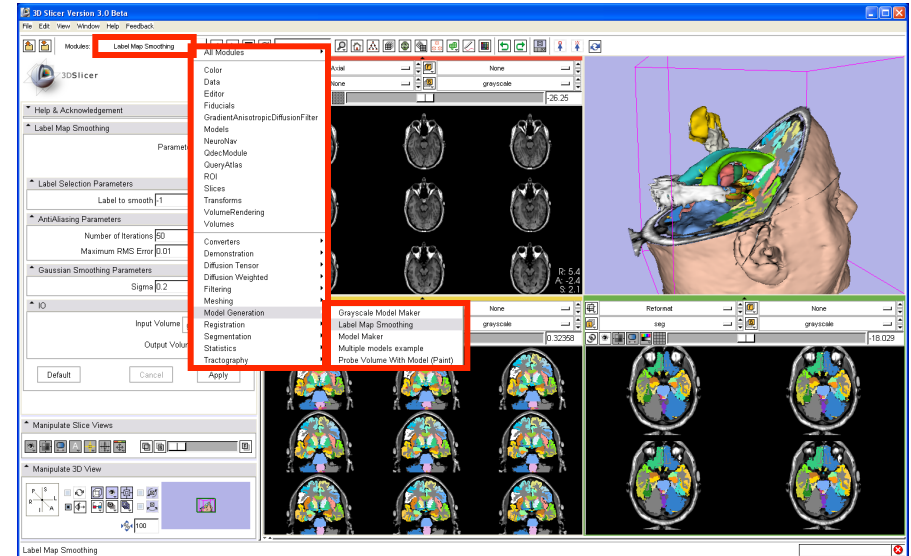
Features	Base	Modules
Visualization	MRML: Models, Volumes, Volume Rendering, Fiducials, Continuum Meshes, Labeled Data	Can create custom MRML Nodes and behavior
Filtering	None	Implemented as Modules using ITK or other Libraries
Registration	Transform Display and Edit, Save/Restore	Calculate Transforms, Resample Data
Segmentation	Label Maps, Parcellated Surfaces	Segmentation Algorithms in ITK or other Libraries
Quantification	Label, Image, Volume Statistics; Numpy access to MRML	Applications in Python or MATLAB
Real-time Integration	VTK Rendering, KWWidgets framework, Tracker Support (as Transforms)	Direct Manipulation of the MRML Scene; 2D/3D Widgets; Device Interfaces, OpenIGTLink Module
Diffusion Imaging	DWI, DTI, Fiber Bundles	Tractography, Clustering, Atlases
Applications	“Bundles” of Modules in Distribution: Registration, Editor, some Filters,	Customized Extensions, Domain-specific code, Optimized Interfaces...



Each module has ...



... an entry in the module menu

... a panel of user interface controls





Built in module

 Slicer provided
 Module writer provided

vtkMRMLNode.h

vtkMRML**Your**ModuleNode.h
•GetNodeTagName()
•Copy()
•ReadXMLAttributes()
•WriteXML()

vtkSlicerModuleGUI.h

vtkSlicer**Your**ModuleGUI.h
•BuildGUI()
•ProcessGUIEvents()
•ProcessLogicEvents()
•ProcessMRMLEvents()

vtkSlicerModuleLogic.h

vtkSlicer**Your**ModuleLogic.h
•ProcessMRMLEvents()
•ProcessLogicEvents()



Loadable module

▭ Slicer provided
▭ Module writer provided

vtkMRMLNode.h

vtkMRML**Your**ModuleNode.h
•GetNodeTagName()
•Copy()
•ReadXMLAttributes()
•WriteXML()

vtkSlicerModuleGUI.h

vtkSlicer**Your**ModuleGUI.h
•BuildGUI()
•ProcessGUIEvents()
•ProcessLogicEvents()
•ProcessMRMLEvents()

vtkSlicerModuleLogic.h

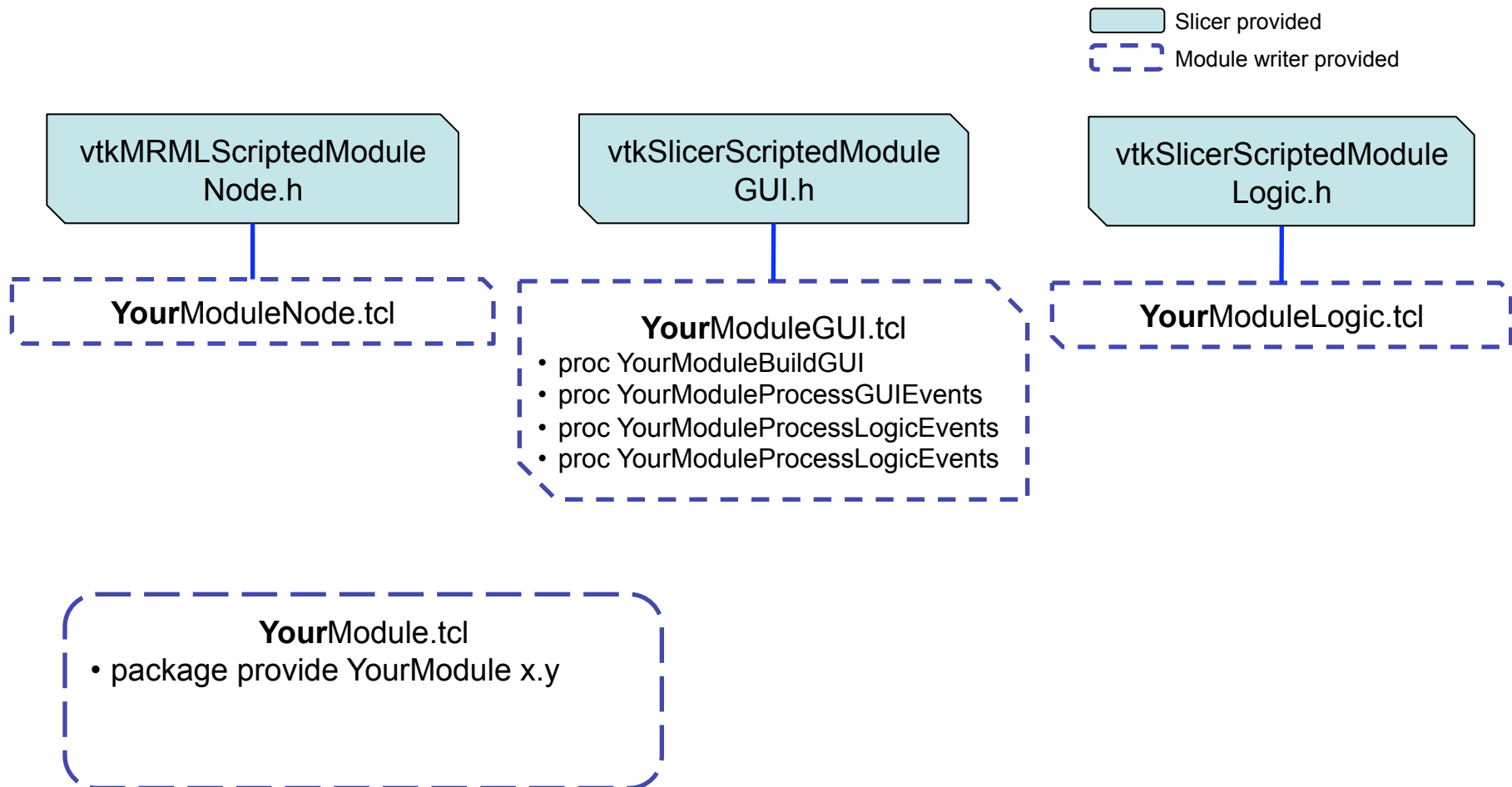
vtkSlicer**Your**ModuleLogic.h
•ProcessMRMLEvents()
•ProcessLogicEvents()

YourModule.so

• GetLoadableModuleDescription()
• GetLoadableModuleLogic()
• GetLoadableModuleGUI()





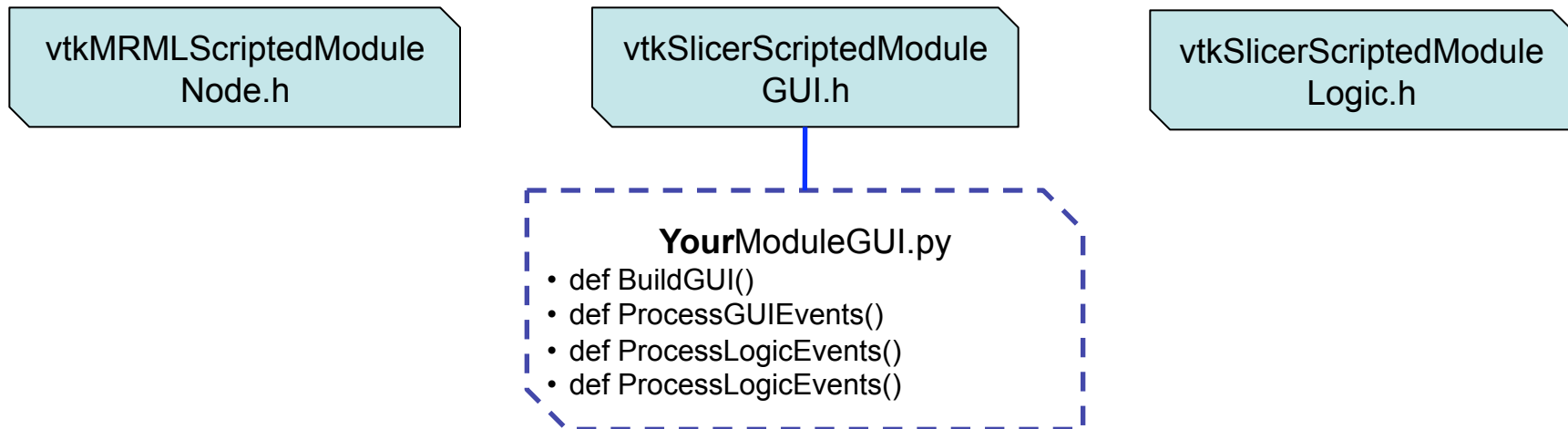
Tcl scripted module







Python scripted module

 Slicer provided
 Module writer provided





Command line module

 Slicer provided
 Module writer provided

vtkMRMLCommandLine
ModuleNode.h

vtkCommandLine
ModuleGUI.h

vtkCommandLine
ModuleLogic.h

YourModule.xml

- <image>
- <geometry>
- <integer>
- <double>
- <file>

YourModule.so

- XMLModuleDescription
- GetXMLModuleDescription()
- ModelEntryPoint()

or

YourModule (exe)

- YourModule -xml

or

YourModule.py

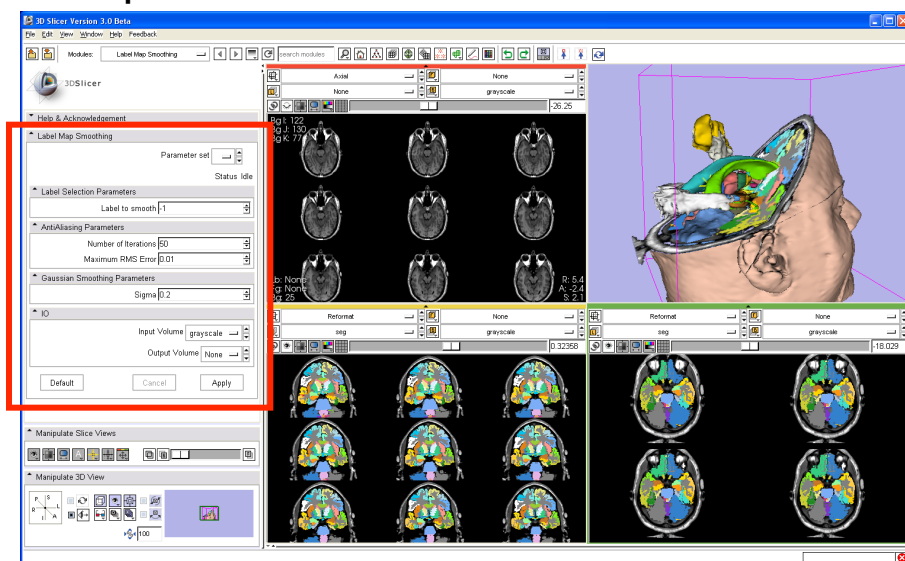
- XML
- def toXML()
- def Execute()



All modules are created equal...

... an entry in the module menu

... a panel of user interface controls





Modularity Goals

- Keep the base package “lean and mean”
- Modules have individual identity
 - Per-module web site, svn, downloads, mailing lists, wiki...
- Allow users to assemble their own set of tools
 - Customized ‘Bundles’ by task or application
- Easy to download compatible extensions
 - Analogous to Firefox extensions
 - Integrate extension builds into developer/nightly/release process
- NITRC Supplement to NA-MIC helping to pay for needed infrastructure (Neuroimaging Informatics Tools and Resources Clearinghouse)
 - NITRC can host neuroimaging projects (gforge implementation)



Module distribution*

Module Management Wizard

icon This wizard lets you search for modules to add to 3D Slicer, download and install them, and uninstall existing modules.

Find & Install Uninstall Either

Download (cache) directory:

Where to search:

Module Management Wizard

Select modules, then click uninstall to remove them from your version of 3D Slicer, or click download to retrieve them.

Select	Status	Module	Category	Description	Homepage	BinaryURI
<input checked="" type="checkbox"/>		Module1	Registration	Registers 2...	http://...	http://...
<input type="checkbox"/>		Module2	Registration	Registers a...	http://...	http://...
<input type="checkbox"/>		Module3	Resgistration	Simple wiz...	http://...	http://...

* Under development



NITRC

The screenshot shows the NITRC website in a Mozilla Firefox browser window. The browser title is "NITRC: Welcome - Mozilla Firefox" and the address bar shows "http://www.nitrc.org/". The page features a navigation menu with "Home", "Community", "Tools & Resources", "Log In", "Register Now", and "Help". The main content area is titled "The Neuroimaging Informatics Tools and Resources Clearinghouse" and includes a paragraph about the organization's mission. Below this, there are four columns of links: "Participate", "Find and Compare", "Rate and Review", and "Topical Communities". A "Latest News" section is also visible. On the right side, there are two boxes: "NITRC Statistics" showing 141 hosted tools/resources and 661 registered users, and "Most Active This Week" listing various tools and their usage percentages.

The Neuroimaging Informatics Tools and Resources Clearinghouse

Funded by the National Institutes of Health Blueprint for Neuroscience Research, NITRC facilitates finding and comparing structural and functional neuroimaging tools and resources. Collecting and pointing to standardized information about tools, this site can help you find the right structural or functional neuroimaging tool or resource and help you decide whether it can help in your research. [Read more about NITRC.](#) [Stay connected with RSS Feeds.](#)

Participate
[Register as a NITRC User \(Why Register?\)](#)
[Add your Tool or Resource](#)
[Discuss \(easiest tool, best practices, open publications\)](#)

Find and Compare
[Find Software or Web sites](#)
[View All Resources](#)
[View more QuickLinks](#)
[Compare Tools or Resources](#)

Rate and Review
[Rate or Review Tools/Resources](#)
Supplemental Funding
[Seek Funding Opportunities](#)

Topical Communities
[Task Independent Fluctuations](#)
[Functional Connectivity Events](#)
[Others](#)

Latest News

Updates to NITRC
Christian Haselgrove - 2008-12-22 10:49 - NITRC Community
In addition to adding quick links on the front page, we have removed the "fMRI" from the tool tree link and around

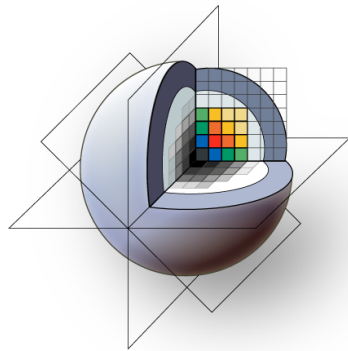
NITRC Statistics
Hosted Tools/Resources: 141
Registered Users: 661

Most Active This Week

- (100.0%) 3DSlicerLupusLesionModule
- (96.8%) BRAINSFit
- (90.3%) MRICron
- (87.1%) DSI Studio
- (83.9%) BrainMap Database
- (80.6%) Automatic Registration Toolbox
- (77.4%) ProbabilisticBiasCorrection
- (74.2%) BrainMask Volume Processing Tool
- (71.0%) Resources for MIPAV Plugin Development
- (67.7%) Functional Imaging BIRN
- (64.5%) Talairach Daemon
- (61.3%) Human Imaging Database (HID)
- (58.1%) BRAINSCut
- (54.8%) Brain lesion segmentation tool using SVM

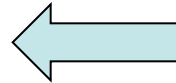


Daemon



3DSlicer

```
slicerget.tcl name > image.nrrd
```



```
cat image.nrrd | slicerput.tcl name
```

YourTool



Variety of levels of integration

Slicer Libs	<ul style="list-style-type: none">• ModuleDescriptionParser• GenerateCLP• vtkITK• MRML	Non-slicer specific support libraries
Slicer Base	<ul style="list-style-type: none">• Application logic• Widgets	Common infrastructure for Slicer applications
Built in modules	<ul style="list-style-type: none">• Slice viewers• Models• Fiducials• Transforms	Full access to Slicer internals
Loadable modules	<ul style="list-style-type: none">• Query Atlas• QDEC• Volume rendering• ChangeTracker• EMSegment	Full access to Slicer internals
Scripted modules	<ul style="list-style-type: none">• Editor• Teem Two Tensor Tractography• VMTK	Limited access to Slicer internals
Command line modules	<ul style="list-style-type: none">• Registration	Restricted access to Slicer internals
Daemon	<ul style="list-style-type: none">• OpenIGTLink• Stochastic Tractography	Access to MRML