

BiImage Suite Web

Xenophon Papademetris and Dustin Scheinost
Dept. of Radiology & Biomedical Imaging
Yale School of Medicine

Adapted from a talk given at Yale, October 2018

What is It?

- BioImage Suite ported to run in a browser
- No SERVER – everything is running locally
- Desktop version (with essentially embedded browser) also available
- Ported over some of the functionality from the “legacy” version

BiImage Suite Web

Applications - Download - Documentation - Developer Info - Help -

https://biimagesuiteweb.github.io/unstableapp/

test2

ame

- sampledata_anat.nii.gz
- sampledata_overlay.nii.gz

4. The orthogonal viewer and editor.

These applications are still in 'beta' (development) stage. Use with care.

Version: 0.9.9 (Unstable 2018/10/12)

Welcome to BiImage Suite Web.

0:00 / 7:29

<https://youtu.be/zWS68aOWUBQ>

Biolmage Suite Web

https://www.bioimagesuite.org

Applications ▾ Download ▾ Documentation ▾ Developer Info ▾ Help ▾

6. A dual registration viewer with linked cursors.

○○○○●

These applications are still in 'beta' (development) stage. Use with care. ✕

Version: 0.9.9 (2018/10/12)

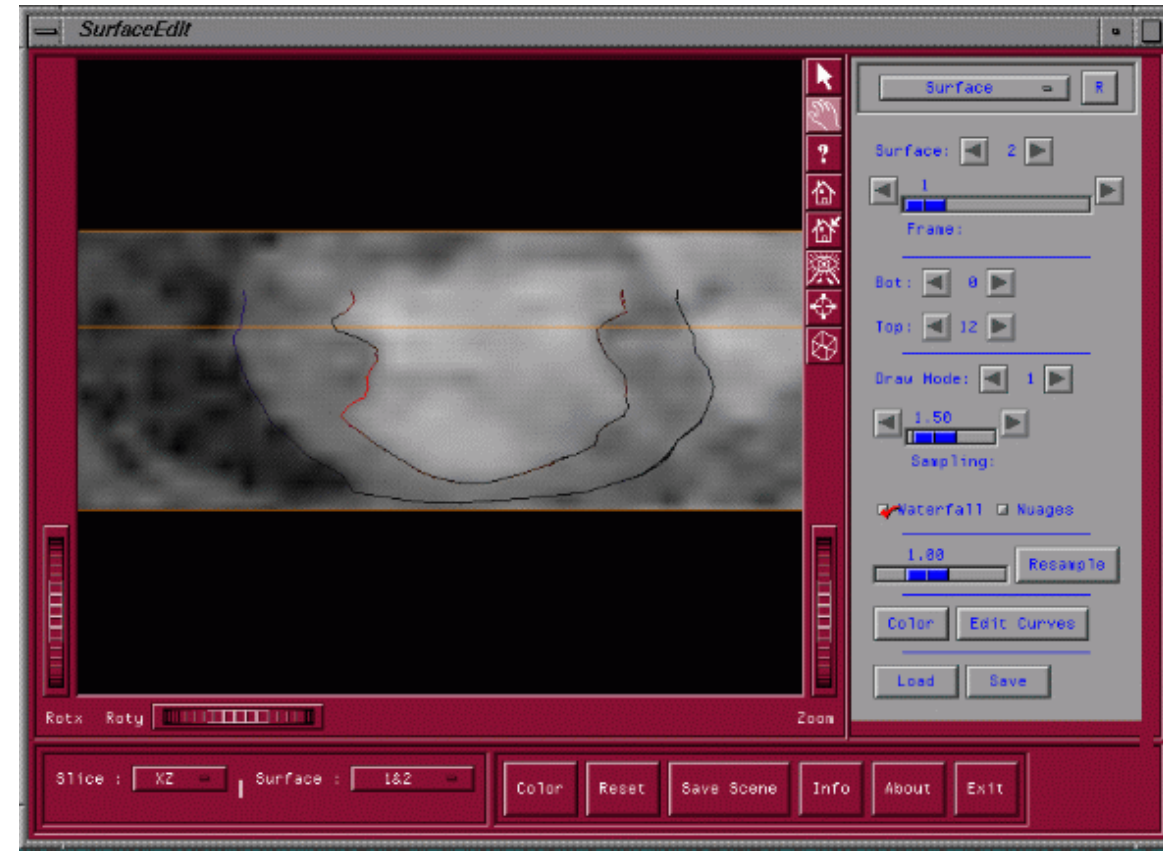
Welcome to Biolmage Suite *Web*.

- Select any tool from the [Applications menu](#) above, or simply click on the application picture, to open the application. Please visit [our download site](#) for commandline and desktop versions of the software.
- **This software is not approved for clinical use. Use at your own risk.**
- Biolmage Suite *Web* is written in Javascript and has been mostly tested on Google Chrome. All applications are running on your local machine. **No data** is ever uploaded to any server.
- We gratefully acknowledge support from the [NIH Brain Initiative](#) under grant R24 MH114805 (*Papademetris X. and Scheinost D. Pls, Dept. of Radiology and Biomedical Imaging, Yale School of Medicine.*)

If you are looking for the old desktop based Biolmage Suite software you may download it from the [Legacy Biolmage Suite Webpage](#).

Biolmage Suite Versions

- 1995 – “Surface Editor” – cardiac segmentation on Silicon Graphics (MOTIF, Open Inventor)
- 2001 – Moved to multiplatform and adapted to brain use (with Todd) (VTK, Tcl)
- 2009 – 2017 : Many failed attempts to port to web
- 2017 – Biolmage Suite Web (JavaScript/WebGL)



Why Web?

- Nothing to install – just navigate to www.bioimagesuite.org
- Works on almost anything (best on Chrome) (try it on your phone!)
- Software is always up-to-date
- But web based does not preclude desktop – we provide
 - Command line apps
 - Desktop Apps
 - Web Applications
- Same code, (almost) same functionality

What can it do?

- Provide six applications (plus two hidden ones!)
- Overlay images to create image visualizations
- Connectivity Visualizations
- Interactive Image Segmentation
- Run various image processing, segmentation and registration algorithms
- MNI2TAL Conversions

MNI ↔ Talairach Converter with Brodmann Areas (1.3)

A MNI Y=0

B MNI X=0

C MNI Z=0

D Show Brodmann areas overlay

E MNI: 0 0 0

F TAL: 0 -2 3

G Area: Outside defined BAs

H **I**

MNI ↔ Talairach Converter with Brodmann Areas (1.3)

MNI Y=-21

MNI X=-12

MNI Z=4

Show Brodmann areas overlay

MNI: -12 -21 4

TAL: -11 -22 7

Area: Left-Thalamus (50)

Connectivity Viewer

Inputs= Parcellation + Matrices

The screenshot shows the BiImage Suite Connectivity Viewer web application. The browser address bar displays the URL: <https://bioimagesuiteweb.github.io/webapp/connviewer.html>. The application has a dark blue header with a menu: File, Edit, View, Parcellations, Advanced, and Help. The main content area is divided into several sections:

- Top Left:** Text reads "Using node definitions from Shen et al. Neuroimage 2013 with 268 nodes." Below this are two small square plots labeled "Positive" and "Negative" showing connectivity matrices.
- Center:** A circular network graph with nodes colored by brain region. A legend titled "'Lobes'" lists: Prefrontal (red), MotorStrip (orange), Insula (yellow), Parietal (green), Temporal (blue), Occipital (purple), Limbic (cyan), Cerebellum (brown), Subcortical (dark orange), and Brainstem (dark purple).
- Right:** A 3D brain model with a network graph overlaid on it, showing connections between nodes.
- Bottom:** Three brain slices: "Sagittal-jk", "Coronal-ik", and "Axial-ij", each showing the brain's parcellation and a crosshair indicating the current node's location.

At the bottom of the main content area, a tooltip for the selected node reads: "Node:238 (L-Cerebellum, unknown, Cerebellum). MNI=(-37,-52.9,-31.1). (Degree: p=34, n=32, s=68) (sorted=238)".

On the right side, there is a "Viewer Controls" panel with a "Viewer Snapshot" section. Below that is the "Connectivity Control" panel, which includes:

- Mode:** Single Node (dropdown)
- Node:** 238 (input field)
- Lobe:** L-Cerebellum (dropdown)
- Network:** Default Mode (dropdown)
- Degree Threshld:** 10 (slider)
- Lines to Draw:** Both (dropdown)
- Display:** Toggle Legends, Toggle 3D Mode
- Buttons:** Create Lines, Clear Lines

The footer of the application contains the BiImage Suite logo and the text: "This application (still in beta!) is part of Yale BiImage Suite. (2018/10/12)".

Overlay Viewer

Inputs=
Anatomical+
Functional
Images

BiImage Suite Overlay Viewer

https://bioimagesuiteweb.github.io/webapp/overlayviewer.html

File Overlay Edit Help

Orthogonal Mosaic

Cluster Analyzer

#	Coordinates of Peak	Size (voxels)	Value At Peak
1	MNI: 66, -6, 6	1256	5720.294
2	MNI: 9, -12, 3	353	-5516.678
3	MNI: 54, -45, 39	263	-5096.590
4	MNI: 42, -66, -3	262	5764.263
5	MNI: 3, 33, 33	249	-5490.896
6	MNI: -33, -60, 42	179	-5405.376
7	MNI: -57, -9, 24	177	5157.129
8	MNI: 30, 18, 51	115	-4545.311
9	MNI: -45, -66, 0	108	5255.873
10	MNI: 39, 30, 27	103	-4375.811

Image: images/samplefunc.nii.gz dim=61,73,61,1,1,1, sp=3,3,3,0,1 orient=LPS type=float

Save Table

Coronal-ik S

Sagittal-jk S

Axial-ij A

Img (27,34,25) =278.98, Ovr: -5516.68

Mode Slices

I-Coord 27

J-Coord 34

K-Coord 25

Labels

Disable Mouse

Image Color Mapping

Overlay Color Mapping

Opacity 1

Overlay Type: Overlay

Overlay Show: Both

Min Overlay 2882.1

Max Overlay 5188

Cluster Size 103

Reset Slices Z- Z+ ?

Viewer Snapshot

Atlas Tool

BiImage Suite Atlas (MNI)

Coordinates (mm)	81, 102, 75
MNI Coordinates	9, -12, 3
Lobes	Right Subcortical
Networks	Subcortical
Brodman Areas	Thalamus (50)

This application (still in beta!) is part of Yale BiImage Suite. (2018/10/12)

Image Editor

Interactive Segmentation

The screenshot displays the Image Editor software interface. At the top, a menu bar includes File, Objectmap, Edit, Tools, and Help. The Tools menu is open, showing options: Paint Tool, Create Objectmap, Morphology Operations, Deface Head Image, Regularize Objectmap, Mask Image, and Landmark Editor. The main workspace shows three brain image slices: Coronal-ik (top left), Axial-ij (bottom left), and Sagittal-ik (top right). Each slice has a red horizontal line, a blue vertical line, and a green vertical line. The Coronal-ik slice is labeled with 'R' (Right) and 'S' (Superior). The Axial-ij slice is labeled with 'R' (Right), 'L' (Left), 'A' (Anterior), and 'P' (Posterior). The Sagittal-ik slice is labeled with 'S' (Superior) and 'P' (Posterior). On the right side, there is a control panel with sections: Viewer Controls, Viewer Snapshot, and Paint Tool. The Paint Tool section includes checkboxes for Enable, Overwrite, 3D Brush, Threshold, and Connect. Below these are Brush Parameters: Brush Size (3), Min Threshold (0), and Max Threshold (2612). At the bottom of the Paint Tool section is a color palette with buttons labeled 0, 1, 2, 3, 4, 5, 6, and -. Below the color palette are Undo and Redo buttons. At the bottom left of the interface, the status bar shows 'Img (90,108,90) =0, Ovr:0'.

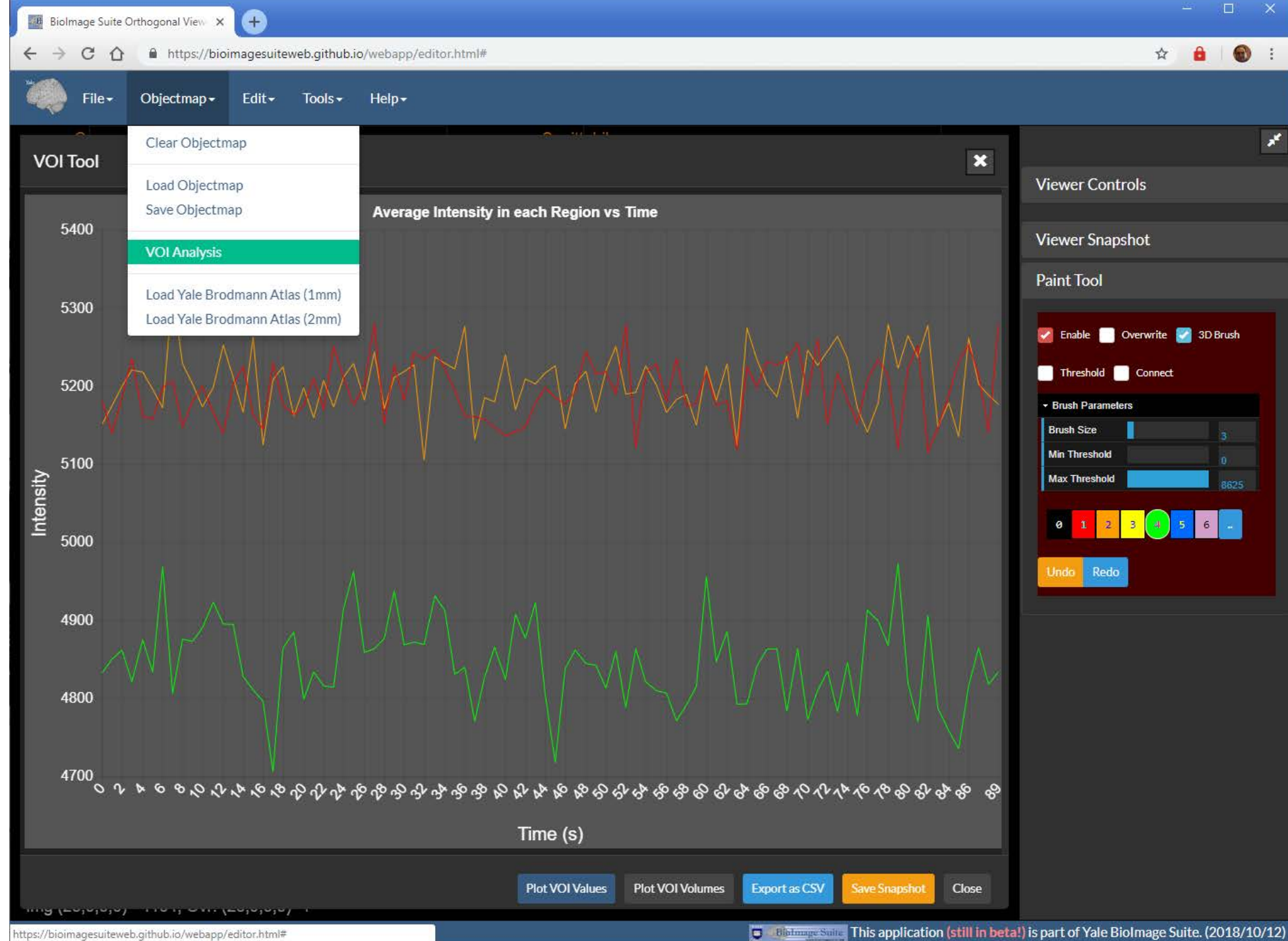
Image Editor II

VOI Analysis

The screenshot displays the BiImage Suite Orthogonal View web application. The interface is divided into three main viewports: Coronal-ik (top-left), Sagittal-jk (top-right), and Axial-ij (bottom-left). Each viewport shows a grayscale brain scan with colored regions of interest (VOI) overlaid. The Coronal-ik view shows a green square VOI. The Sagittal-jk view shows two green squares and one orange square VOI. The Axial-ij view shows a red square, an orange square, and a green square VOI. The application includes a menu bar (File, Objectmap, Edit, Tools, Help) and a toolbar with navigation and editing icons. A control panel on the right side contains 'Viewer Controls', 'Viewer Snapshot', and 'Paint Tool' settings. The Paint Tool settings include checkboxes for 'Enable', 'Overwrite', and '3D Brush', as well as 'Threshold' and 'Connect' options. The 'Brush Parameters' section shows 'Brush Size' set to 3, 'Min Threshold' set to 0, and 'Max Threshold' set to 8625. A color palette with buttons for 0, 1, 2, 3, 4, 5, 6, and a range button is also present. The bottom status bar displays 'Img (25,9,6,0) =4104, Ovr: (25,9,6,0)=4'. The footer text reads: 'This application (still in beta!) is part of Yale BiImage Suite. (2018/10/12)'

Image Editor II

VOI Analysis



Single Viewer

Image Display + Processing

The screenshot displays the BiImage Suite Viewer Tool interface. The main area shows three brain image slices: Coronal-ik, Sagittal-jk, and Axial-ij. A context menu is open over the Coronal-ik slice, listing various processing options. The 'Crop Image' option is highlighted in green. A red arrow points from this option to the 'Crop Image' section in the 'Viewer Controls' panel on the right. The 'Viewer Controls' panel includes a 'Core' section with 'Mode' set to 'Slices', coordinate sliders for I-Coord (45), J-Coord (54), and K-Coord (45), and checkboxes for 'Labels' and 'Disable Mouse'. The 'Viewer Snapshot' section shows the 'Crop Image' parameters: I-Start (0), I-End (90), J-Start (0), J-End (108), K-Start (0), and K-End (90). At the bottom, there are 'Crop', 'Undo', and 'More' buttons. The status bar at the bottom left shows 'Img (45,54,45) =67'.

BiImage Suite Viewer Tool

https://bioimagesuiteweb.github.io/unstableapp/viewer.html

File Overlay Edit Image Processing Segmentation Help

Coronal-ik Sagittal-jk Axial-ij

Smooth Image
Normalize Image
Threshold Image
Cluster Threshold
Correct Bias Field
Resample Image
Shift+Scale(+Cast) Image
Reorient Image
Flip Image
Crop Image
Blank Image
Extract Frame
Combine Images
Process 4D Image

Viewer Controls

Core

Mode Slices

I-Coord 45

J-Coord 54

K-Coord 45

Labels

Disable Mouse

Image Color Mapping

Overlay Color Mapping

Reset Slices Z- Z+ ?

Viewer Snapshot

Crop Image

Inputs

Outputs

Parameters

I-Start 0

I-End 90

J-Start 0

J-End 108

K-Start 0

K-End 90

Advanced

Crop Undo More

Img (45,54,45) =67

https://bioimagesuiteweb.github.io/unstableapp/viewer.html#

BiImage Suite This application (still in beta!) is part of Yale BiImage Suite. (2018/10/22)

Dual Viewer

Image Display
+ Processing
+ Registration
+ ...

The screenshot displays the Dual Viewer software interface. The main workspace is divided into four quadrants, each showing a different view of a brain slice. A context menu is open over the top-right quadrant (C), listing options: 'Both Viewers', 'Viewer 1 Only', 'Viewer 2 Only', 'Viewer 1 Info', and 'Viewer 2 Info'. The bottom status bar shows 'V1: Img (108,36,94) =2079' and 'V2: Img (54,18,47) =136'. On the right side, there is a control panel with sections for 'Viewer 1 Controls', 'Viewer 2 Controls', and 'Viewer Snapshot'. The 'Viewer 1 Controls' section includes a 'Mode' dropdown set to 'Slices', and sliders for 'I-Coord' (51), 'J-Coord' (18), and 'K-Coord' (47). The 'Viewer Snapshot' section includes a 'Scale' dropdown set to 'x2', checkboxes for 'White Bkgrd' and 'Crop', and a 'Take Snapshot' button.

Both Viewers
Viewer 1 Only
Viewer 2 Only

Viewer 1 Info
Viewer 2 Info

Viewer 1 Controls
Viewer 2 Controls

Core
Mode: Slices
I-Coord: 51
J-Coord: 18
K-Coord: 47
Labels:
Disable Mouse:
Image Color Mapping
Overlay Color Mapping

Reset Slices Z- Z+ ?

Viewer Snapshot
Scale: x2 White Bkgrd Crop
Take Snapshot

V1: Img (108,36,94) =2079
V2: Img (54,18,47) =136

Registration I

The screenshot displays a medical image registration software interface. The main workspace is divided into five viewports showing brain slices in different planes: Coronal-ik, Sagittal-ik, Coronal-k, Sagittal-k, and Axial-ik. Each viewport shows a reference image (white outline) and a target image (gray) with a green registration line. The interface includes a menu bar at the top with options: File, Overlay, Image2, Overlay2, Edit, Display, Image Processing, Segmentation, Registration, and Help. On the right side, there is a control panel for the registration process, including 'Viewer 2 Controls', 'Viewer Snapshot', and 'Linear Registration' settings. The 'Linear Registration' panel has sections for 'Inputs', 'Parameters', and 'Advanced'. The 'Parameters' section includes sliders for 'Levels' (set to 3), 'Iterations' (set to 10), and 'Resolution' (set to 1.5), along with a 'Mode' dropdown set to 'Affine' and a checked 'Header Orient' checkbox. At the bottom of the control panel are buttons for 'Run', 'Undo', 'Redo', and 'More'. The status bar at the bottom left shows 'V1: Img (54,63,63) =390' and the status bar at the bottom right shows 'V2: Img (108,128,128) =311'.

File Overlay Image2 Overlay2 Edit Display Image Processing Segmentation Registration Help

Viewer 2 Controls

Viewer Snapshot

Linear Registration

Inputs

- Reference Image
- Target Image
- Initial Xform: identity
- Resliced Image

Parameters

- Levels: 3
- Iterations: 10
- Mode: Affine
- Header Orient:
- Resolution: 1.5

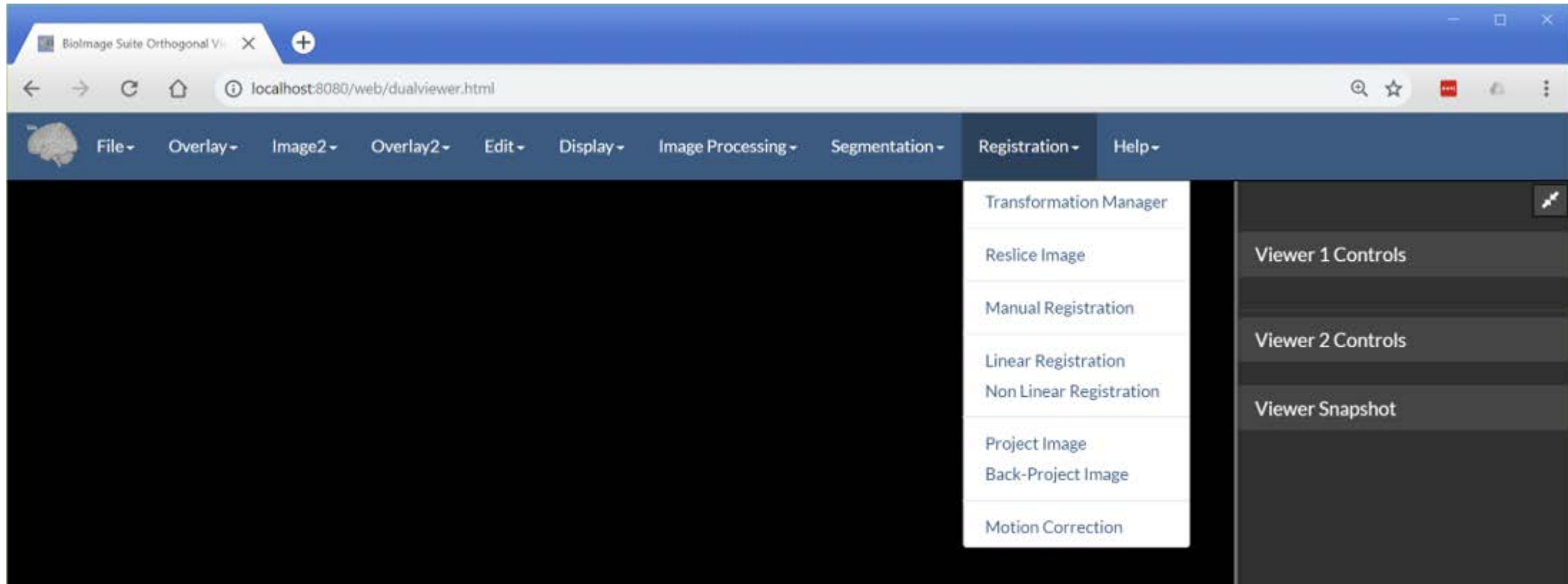
Advanced

Run Undo Redo More

V1: Img (54,63,63) =390

V2: Img (108,128,128) =311

Registration II

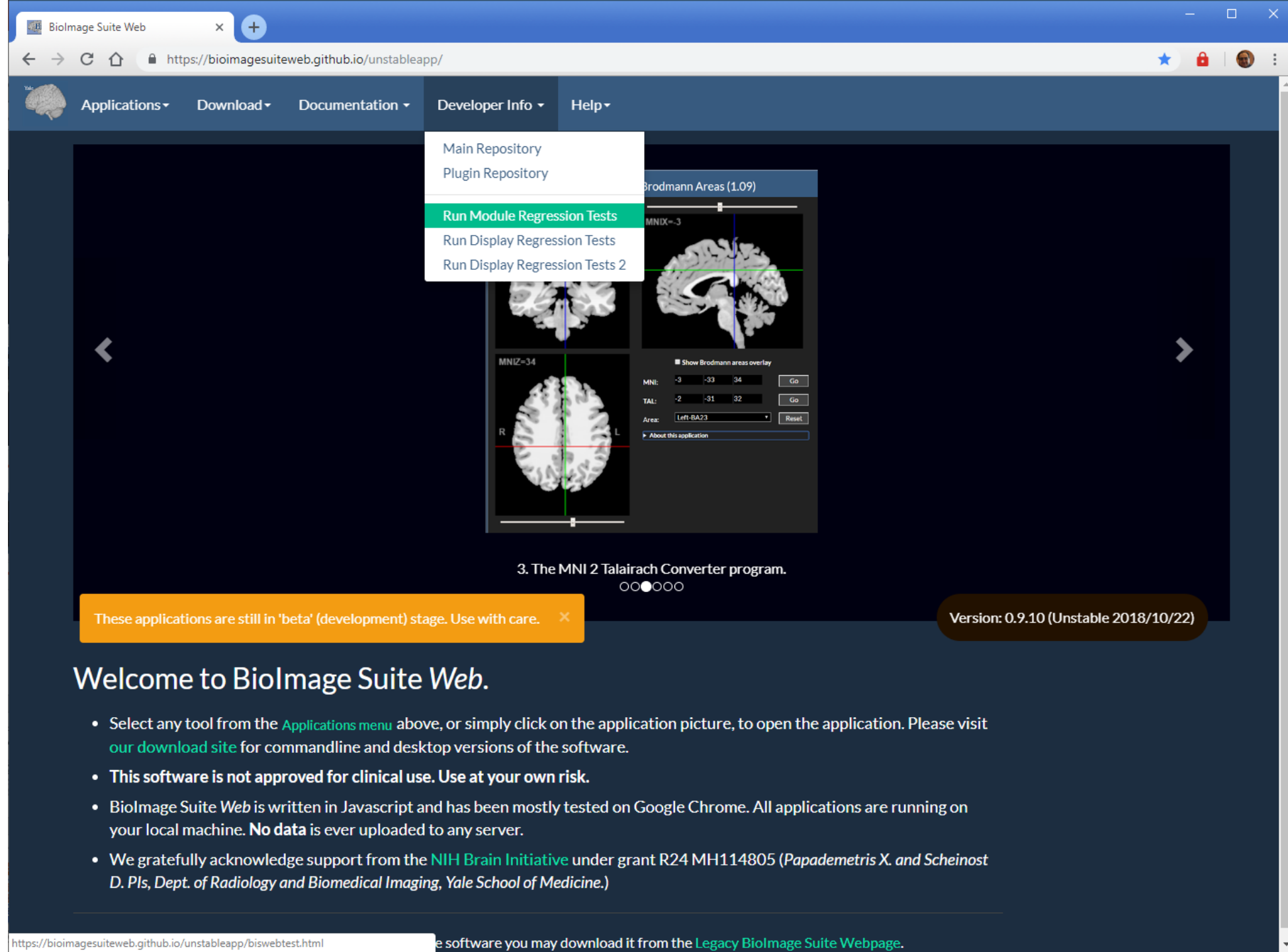


Registration III

The interface features a top menu bar with options: File, Overlay, Image2, Overlay2, Edit, Display, Image Processing, Segmentation, Registration, and Help. The main workspace is divided into two vertical panels, V1 and V2, each containing two image windows. V1 shows a reference image (left) and a target image (right) with crosshairs. V2 shows the reference image (left) and the resliced target image (right) with crosshairs. A central control panel titled 'Linear Registration' includes sections for 'Inputs' (Reference Image, Target Image, Resliced Image), 'Parameters' (Levels: 3, Iterations: 10, Mode: Affine, Resolution: 1.5), and 'Advanced' options. At the bottom of the control panel are buttons for 'Run', 'Undo', 'Redo', and 'More'.

```
***** bislib.js:845
*****
** Beginning to compute linear registration at          bislib.js:845
level=1, numsteps=1, tolerance=0.001 centeronrefonly=1
***** bislib.js:845
** Initializing level =1.500, rate=2.000 resolu= 1.500 bislib.js:845
**
** Preprocessing Step1 linear
registration:level_ref_image:
** Extracting frame =0 dim=(128,128,88) spa=          bislib.js:845
(2.000,2.000,2.000) rng=(0.000:880.000)
** Smoothing (1.274,1.274,1.274) --> dim=(128,128,88) bislib.js:845
spa=(2.000,2.000,2.000) rng=(0.000:738.320) vx->sigma=(0.637,0.637,0.637)
** Resampling (3.000,3.000,3.000) --> dim=(85,85,59) bislib.js:845
spa=(3.000,3.000,3.000) rng=(0.017:728.177)
** Normalizing (0.010:0.990) 63 --> dim=(85,85,59) bislib.js:845
spa=(3.000,3.000,3.000) rng=(0.000:63.000) robust 1:99 %,
info=0.017,554.142 numbins=64
**
** Preprocessing Step1 linear
registration:level_targ_image:
** Extracting frame =0 dim=(216,256,256) spa=          bislib.js:845
(1.000,0.977,0.977) rng=(0.000:646.000)
** Smoothing (0.622,0.622,0.622) --> dim=(216,256,256) bislib.js:845
spa=(1.000,0.977,0.977) rng=(0.000:631.582) vx->sigma=(0.622,0.637,0.637)
** Resampling (1.465,1.465,1.465) --> dim=(147,170,170) bislib.js:845
spa=(1.465,1.465,1.465) rng=(0.000:626.669)
** Normalizing (0.010:0.990) 63 --> dim=(147,170,170) bislib.js:845
spa=(1.465,1.465,1.465) rng=(0.000:63.000) robust 1:99 %,
info=0.000,405.492 numbins=64
***** bislib.js:845
*****
** Beginning level=1 resolution=3 numdof=12          bislib.js:845
current_step=3
**
** In step = 1. Iterations = 10, optimization=2,      bislib.js:845
current=3.000.
**** CG: Beginning (it=10), tol=0.001, stepsize=3:    bislib.js:845
**** CG:Init 0 : (-13.95936 12.95625 -6.92582 -0.80411 bislib.js:845
-1.06547 3.93905 100.00000 100.00000 100.00000 0.00000 0.00000 )
-0.13928
**** CG:Lmn 1 : (-15.56428 11.68796 -5.72948 0.68109 bislib.js:845
-0.85405 3.88245 99.37441 98.55619 102.74436 -0.01889 0.01587 -0.01591 )
-0.14200
**** CG:Lmn 2 : (-16.56783 10.19819 -6.51256 0.67115 bislib.js:845
0.20496 3.39537 99.20675 97.13290 105.01382 -0.03828 0.04152 -0.03307 )
+0.14410
**** CG:Lmn 3 : (-16.37077 10.02462 -6.72674 0.40214 bislib.js:845
0.33843 2.89729 99.54864 97.00649 105.08165 -0.04760 0.04458 -0.04030 )
-0.14450
**** CG:Done 4 : (-16.37077 10.02462 -6.72674 0.40214 bislib.js:845
0.33843 2.89729 99.54864 97.00649 105.08165 -0.04760 0.04458 -0.04030 )
-0.14410
**** CG Stats: Nfunc=25, Ngrad=3, vx=0.14450          bislib.js:845
*****
** Stats : total_time 8                               bislib.js:845
*****
```

Testing and Verification



The screenshot shows the BiImage Suite Web application interface. At the top, there is a navigation bar with a logo and menu items: Applications, Download, Documentation, Developer Info, and Help. A dropdown menu is open under 'Developer Info', listing options: Main Repository, Plugin Repository, Run Module Regression Tests (highlighted in green), Run Display Regression Tests, and Run Display Regression Tests 2. The main content area features a brain image viewer with a control panel on the right. The control panel includes a 'Show Brodmann areas overlay' checkbox, coordinate input fields for MNI (-3, -33, 34) and TAL (-2, -31, 32), an 'Area' dropdown menu set to 'Left-BA23', and 'Go' and 'Reset' buttons. Below the viewer, there is a progress indicator for '3. The MNI 2 Talairach Converter program.' with five circles, the second of which is filled. A yellow warning box at the bottom left states: 'These applications are still in "beta" (development) stage. Use with care.' A dark brown box at the bottom right shows the version: 'Version: 0.9.10 (Unstable 2018/10/22)'. The footer contains the URL 'https://bioimagesuiteweb.github.io/unstableapp/biswebtest.html' and a link to the 'Legacy BiImage Suite Webpage'.

Applications ▾ Download ▾ Documentation ▾ Developer Info ▾ Help ▾

- Main Repository
- Plugin Repository
- Run Module Regression Tests
- Run Display Regression Tests
- Run Display Regression Tests 2

Brodmann Areas (1.09)

MNI: -3 -33 34 Go

TAL: -2 -31 32 Go

Area: Left-BA23 Reset

About this application

3. The MNI 2 Talairach Converter program.

○○●○○○

These applications are still in "beta" (development) stage. Use with care. ✕

Version: 0.9.10 (Unstable 2018/10/22)

Welcome to BiImage Suite Web.

- Select any tool from the [Applications menu](#) above, or simply click on the application picture, to open the application. Please visit [our download site](#) for commandline and desktop versions of the software.
- **This software is not approved for clinical use. Use at your own risk.**
- BiImage Suite Web is written in Javascript and has been mostly tested on Google Chrome. All applications are running on your local machine. **No data** is ever uploaded to any server.
- We gratefully acknowledge support from the [NIH Brain Initiative](#) under grant R24 MH114805 (*Papademetris X. and Scheinost D. Pls, Dept. of Radiology and Biomedical Imaging, Yale School of Medicine.*)

<https://bioimagesuiteweb.github.io/unstableapp/biswebtest.html> e software you may download it from the [Legacy BiImage Suite Webpage](#).

Module Testing I

BiImage Suite Web Regression Test Runner

First: 0 Last: 87 Testname

WebWorker FileServer Run Tests Run Memory Test

Welcome to the BiImage Suite Web R

Select the tests you would like to run from the co

► Additional Help

Note: You may, before running tests, like to open
observe the print outs from the algorithms.


Developer Tools (on Chrome simply press 'Control-Shift-I' or 'Apple-Option-I') to

Test all modules

- approximateField
- backProjectImage
- binaryThresholdImage
- blankImage
- butterworthFilter
- clusterThreshold
- combineImages
- computeCorrelation
- computeGLM
- computeROI
- cropImage
- defaceImage
- displacementField
- extractFrame
- extractSlice
- flipImage
- linearRegistration
- manualRegistration
- maskImage
- morphologyFilter
- motionCorrection
- motionReslice
- nonlinearRegistration
- normalizeImage
- prepareRegistration
- process4DImage
- projectImage
- regressGlobal
- regressOut
- regularizeObjectmap
- reorientImage
- resampleImage
- resliceImage
- segmentImage
- shiftScaleImage
- sliceBiasCorrect
- smoothImage
- thresholdImage

Module Testing II

← → ↻ 🏠 🔒 https://bioimagesuiteweb.github.io/webapp/biswebtest.html 🔍 ☆

 **BioImage Suite Web Regression Test Runner**

First: Last: Testname: WebWorker

Running Tests

Executing tests 0:87 (Max index=87). Only running tests with name=computeCorrelation

Test 3: computeCorrelation

- Command: `computeCorrelation -i testdata/ButterWorthOutput.csv --zscore false`
- Test details: `--test_target testdata/newtests/goldcorrelation.matr --test_type matrix --test_comparison ssd --test_threshold 0.01`
- Should pass: true

.... test execution time=0.21s
.... WASM memory size=16 MB.

Test completed, now checking results.

Comparing matrix using ssd and threshold=0.01 Module computeCorrelation test **passed**.
deviation (ssd) from expected: $2.537576312709809e-7 < 0.01$

Tests for version=2018/09/03: completed=1/88, passed=1/88, failed=0/88, skipped=87/88

```
"command" : "computeCorrelation -i testdata/ButterWorthOutput.csv --zscore false",  
"test"    : "--test_target testdata/newtests/goldcorrelation.matr --test_type matrix"
```

Image Display Testing

The screenshot displays the BiImage Suite Web Display Regression Test Runner interface. The browser address bar shows the URL: `https://bioimagesuiteweb.github.io/unstableapp/biswebdisplaytest.html`. The application has a menu bar with `File`, `Overlay`, `Edit`, and `Help`. The main header reads "BiImage Suite Web Display Regression Test Runner" and includes controls for "First: 0" and "Last: 8", along with "Run Single Test" and "Run Multiple Tests" buttons. Below the header, there are tabs for "Orthogonal" and "Mosaic". The central area shows a 4x4 grid of brain slices with yellow and red overlays. To the right of the grid is a "Viewer Controls" panel with settings for "Plane" (Axial), "Rows", "Columns", "First", "Increment", and "Labels". Below this is an "Image Color Mapping" and "Overlay Color Mapping" section, and a "Reset Slices" button. Further down is a "Viewer Snapshot" section with a "Scale" dropdown (set to x1), checkboxes for "White Bkgd" and "Crop", and a "Take Snapshot" button. At the bottom of the grid is a color scale legend with values: -5188, -4035, -2882, 2882, 4035, 5188. On the right side, there are two panels: "Gold 8" and "Result 8", each showing a 4x4 grid of brain slices. Below these is a log window titled "Starting Test 8" with the following content:

```
Starting Test 8
Reading app state from images/testdata/overlay2.biswebstate
Reading result from: images/testdata/test8.png
Result: {"testresult":true,"value":0.9999736178960362,"metric": "cc"}
```

At the bottom of the interface, a status bar reads: "Tests for version=2018/10/22: completed=9/9, passed=9/9, failed=0/9".

Connectivity Display Testing

Biomed Suite Display Test 2

https://bioimagesuiteweb.github.io/unstableapp/biswebdisplaytest2.html

File Edit View Parcellations Advanced Help

Biomed Suite Web Display Regression Test Runner (2)

First: 1 Last: 5 Run Single Test Run Multiple Tests

Using node definitions from Shen et al. Neuroimage 2013 with 268 nodes.

Positive Negative

Lobes: Prefrontal, MotorStrip, Insula, Frontal, Temporal, Occipital, Limbic, Cerebellum, Subcortical, Brainstem

Node: 108 (R-Cerebellum, unknown, Cerebellum).
MNI: (16.2, -47.2, -52.3), (Degree: p=1, n=2, s=3) (sorted=108)

Viewer Controls

Viewer Snapshot

Connectivity Control

Core

Mode: Single Node

Node: 108

Lobe: R-Cerebellum

Network: Default Mode

Degree Threshld: 10

Lines to Draw: Both

Display

Toggle Legends Toggle 3D Mode

Create Lines Clear Lines

Gold 0 Result 0

Test: 0 cc=0.994 PASSED

Starting Test 0

Reading app state from images/testdata/connstate.biswebstate

Reading result from: images/testdata/otest1.png

Result: {"testresult":true,"value":0.993808477333613,"metric":"cc"}

Tests Completed

2: completed=1/1, passed=1/1, failed=0/1

Manual Testing (Cheryl, Dana, Haley)

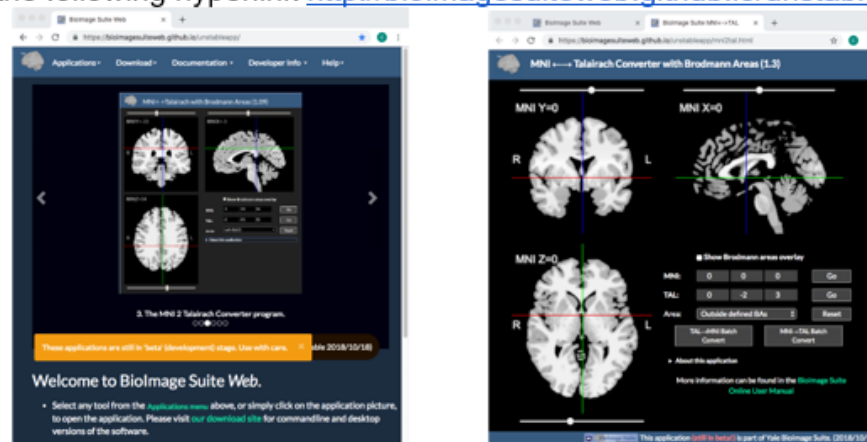
Testing Protocol for Bioimage Suite: MNI2TAL

Background: Imaging of the brain is a relatively new technique in medicine, neuroscience, and psychology. The Montreal Neurological Institute (MNI) template is used to identify reference spaces in imaging studies to identify structures and regions within the brain. The Talairach (TAL) Atlas is a mapping technique used to reduce the intersubject anatomic variability in mapping studies and allows for comparisons between coordinate-based data. Each point in the brain has both a set of MNI and Talairach coordinates.

About BioImage Suite: BioImage Suite is a collection of image analysis programs available as a software and online that can be used to view brain regions. One of the applications in Bioimage Suite, MNI2TAL, is used to provide the MNI and Talairach coordinates of a given point in the brain. Given any MNI coordinate, the program outputs the Talairach equivalent coordinate, and for any Talairach coordinate, the program provides the corresponding MNI.

Testing MNI2TAL on a Mac:

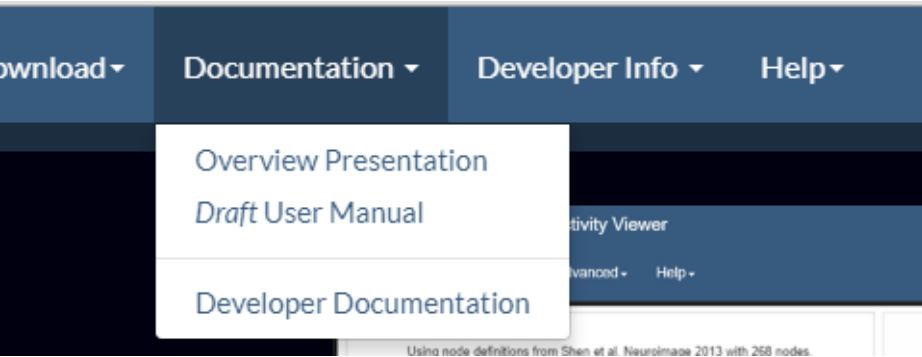
1. Open the following hyperlink <http://bioimagesuiteweb.github.io/unstableapp/>,



- a. Find and click on “3. The MNI 2 Talairach Converter program”

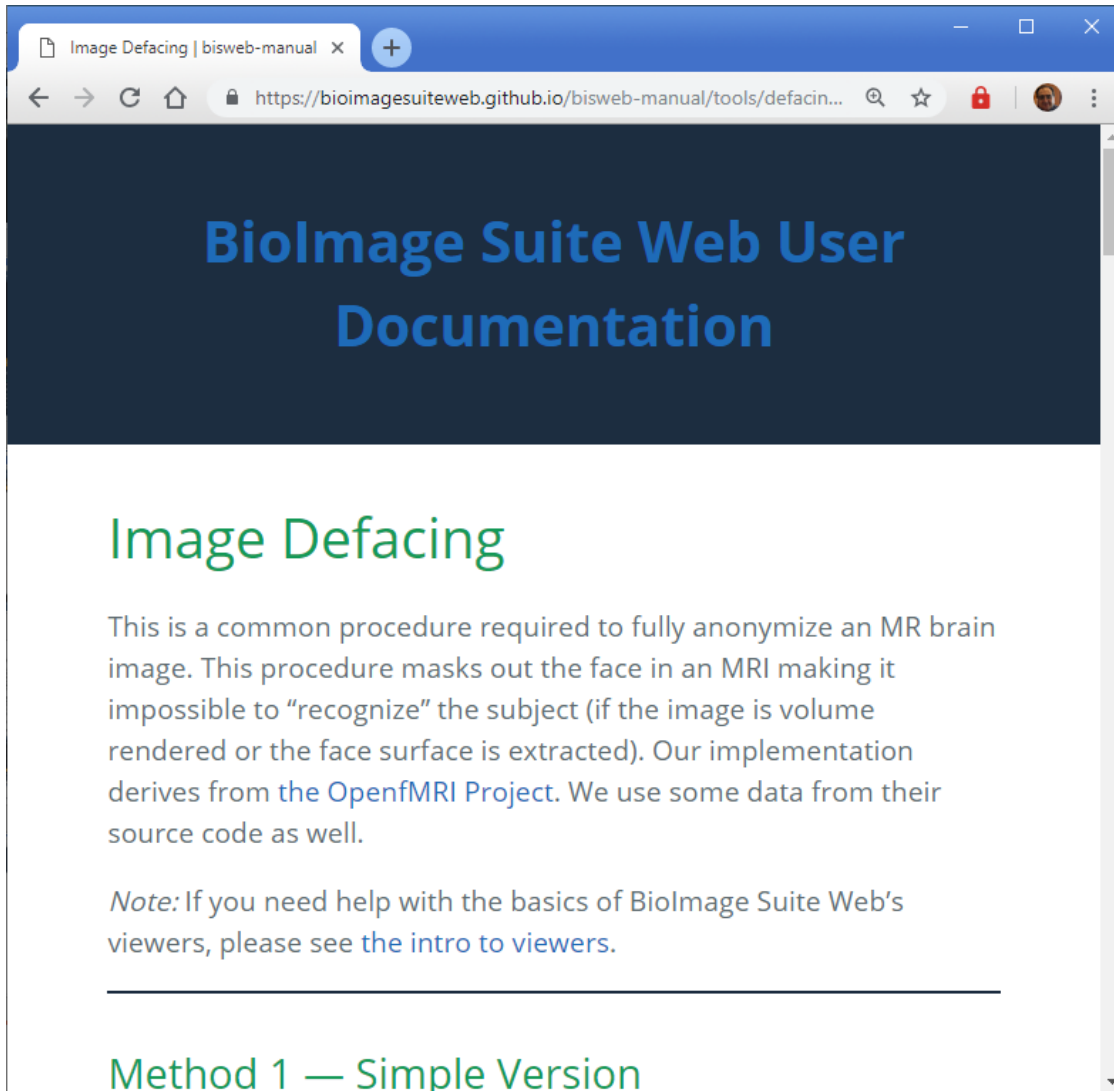
10/19/2018	Cheryl	MNI2TAL	9.1 - 10/18/18	MNI2TAL	Mac	Chrome	All	error	running TAL > MNI batch conversion yields the wrong coordinates. MNI > TAL batch conversion works fine.	Fixed it was using x
------------	--------	---------	----------------	---------	-----	--------	-----	-------	---	----------------------

Documentation



A screenshot of a web browser displaying the 'Biolmage Suite Web User Documentation' page. The browser's address bar shows the URL 'https://bioimagesuiteweb.github.io/bisweb-manual/'. The page has a dark blue header with the title 'Biolmage Suite Web User Documentation' in white. Below the header, the page content is white. The main heading is 'Table Of Contents' in green. The text reads: 'First we cover some core information in this document:' followed by a bulleted list: 'Starting Biolmage Suite Web - how to get and run the software.', 'Some Key Information - information about configuring your browser and default image orientations.', and 'The following documents describe the different applications in Biolmage Suite Web.' followed by another bulleted list: 'Displaying Images - we discuss here how the core viewer in Biolmage Suite web works (including colormapping etc.). This is shared by the almost all the applications.', 'The Overlay Viewer Application - an application optimized for displaying functional overlays, including a brief description of the Mosaic Viewer Tool included here.', 'The Image Editor Application - an application that can be used for interactive segmentation and VOI analysis of images.', 'The Connectivity Visualization Tool - an application to visualize connectome matrices.', 'The MNI2TAL Tool - an application to map MNI to Talairach coordinates and vice-versa.', and 'The Dual Viewer Tool - an advanced dual viewer application. This is intended to be primarily used for image registration tasks.' Below this, the text reads: 'These documents describe how to perform specific tasks in one or more of the applications in the software.' followed by a final bulleted list: 'Image Processing Tasks - here we describe how to perform basic image processing tasks (e.g. image thresholding, cropping, resampling) using the tools in Biolmage Suite Web.', 'Image Segmentation - this describes image segmentation functionality in our software.', and 'Image Registration - the image registration tools are described in the document describing the'.

Text + Video (Videos in Progress)



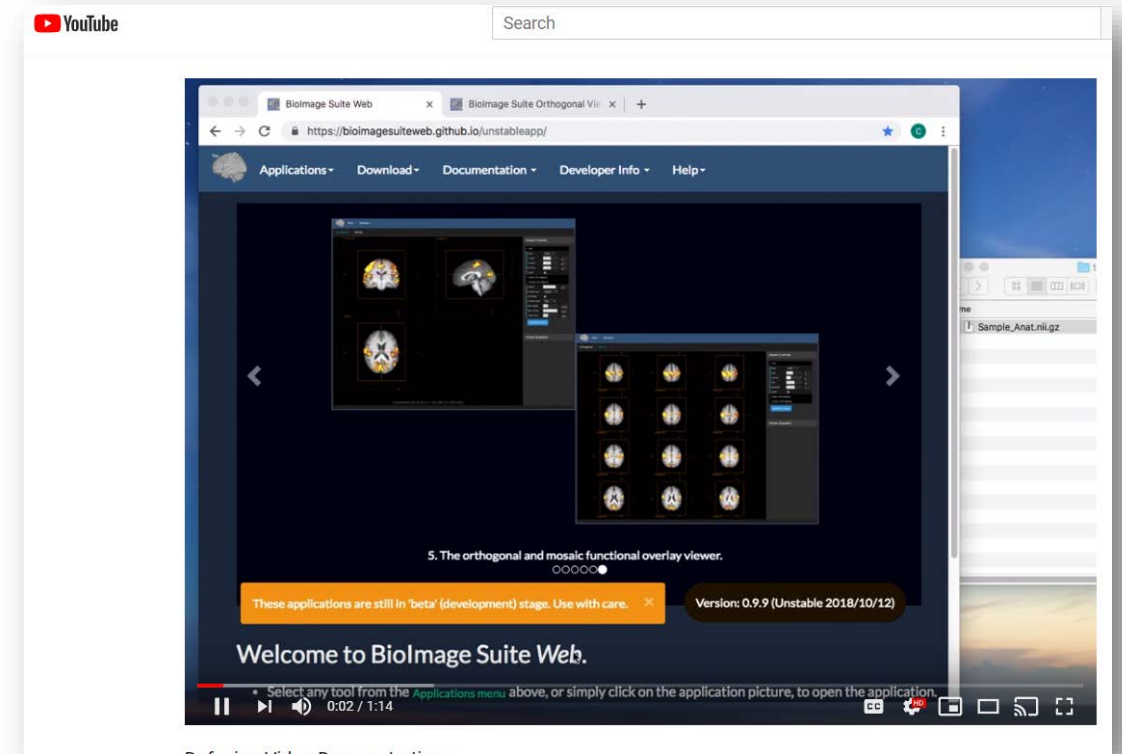
The screenshot shows a web browser window with the URL <https://bioimagesuiteweb.github.io/bisweb-manual/tools/defacin...>. The page title is "Image Defacing | bisweb-manual". The main heading is "BioImage Suite Web User Documentation" in blue. Below it, the section "Image Defacing" is written in green. The text explains that image defacing is a common procedure to anonymize MR brain images by masking the face. It mentions that the implementation is based on the OpenfMRI Project. A note suggests consulting the "intro to viewers" for help with the basics. At the bottom, "Method 1 — Simple Version" is listed in green.

Image Defacing

This is a common procedure required to fully anonymize an MR brain image. This procedure masks out the face in an MRI making it impossible to “recognize” the subject (if the image is volume rendered or the face surface is extracted). Our implementation derives from the [OpenfMRI Project](#). We use some data from their source code as well.

Note: If you need help with the basics of BioImage Suite Web’s viewers, please see the [intro to viewers](#).

Method 1 — Simple Version



The screenshot shows a YouTube video player. The video content is a screenshot of the BioImage Suite Web application interface. The interface has a dark blue header with navigation links: "Applications", "Download", "Documentation", "Developer Info", and "Help". The main area displays a grid of application thumbnails. A tooltip or overlay is visible, showing a 3x3 grid of brain slices. Below the grid, there is a text box that says "5. The orthogonal and mosaic functional overlay viewer." and a version indicator "Version: 0.9.9 (Unstable 2018/10/12)". At the bottom of the video frame, it says "Welcome to BioImage Suite Web." and "Select any tool from the Applications menu above, or simply click on the application picture, to open the application." The video player controls at the bottom show a play button, a progress bar at 0:02 / 1:14, and other standard controls.

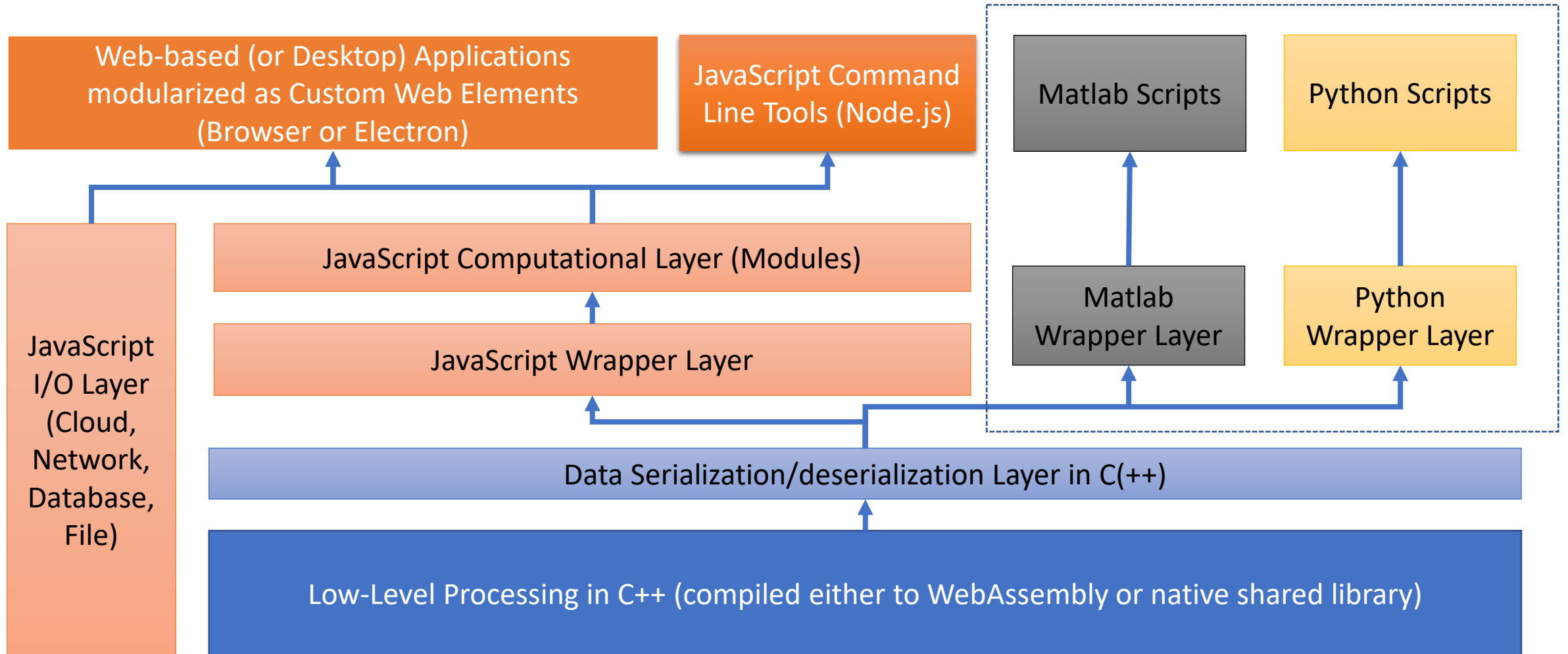
<https://youtu.be/a8Cbs4SJABs>

<https://bioimagesuiteweb.github.io/bisweb-manual/tools/defacing.html>

Key Technological (Software) Innovations

- Multi-context: same code commandline (via Node.js), desktop (via Electron) and Web
 - Each context has unique strengths
 - Tune to technical sophistication of the user and her needs
- Formal module architecture for automatic GUI and command line implementations of algorithmic modules.
- WebAssembly implementation of computationally expensive code (C++ compiled to WASM) e.g. Non Linear Registration ~90% native performance
- Custom Web Elements to modularize the applications
- No server, all computation is done in the client (your browser)
- A lot of this work builds on our previous experience with the Yale BioImage Suite (www.bioimagesuite.org) software package → This is really the web version of this.

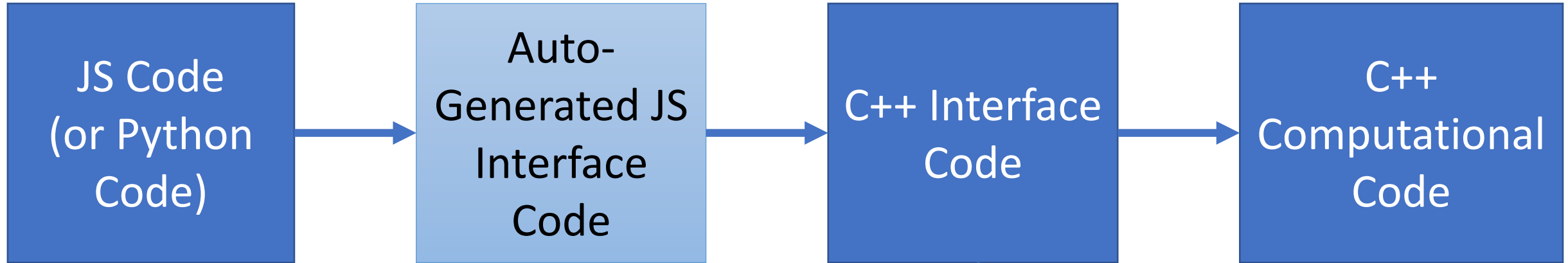
Software Architecture



C++/Web Assembly

- ~18,000 lines of C++ code (per wc) (much of this adapted from Biolmage Suite but stripped of dependencies to VTK/ITK/..)
- Only external dependency is Eigen numerical library
- Compile into bytecode using Emscripten
- Serialization/Transfer layer
 - JS \leftrightarrow C++ data transfer is restricted to simple “C”-style objects, essentially numbers and arrays

C++ JS Integration



```
/** Compute butterworthFilter output
 * @param input the input matrix to filter (time = rows)
 * @param jsonstring the parameters { 'type': "low", "cutoff": 0.15, 'sampleRate': 1.5 };
 * @param debug if > 0 print debug messages
 * @returns a pointer to the filtered matrix (rows=frames,cols=rois)
 */
// BIS: { 'butterworthFilterWASM', 'Matrix', [ 'Matrix', 'ParamObj', 'debug' ] }
BISEXPORT unsigned char* butterworthFilterWASM(unsigned char* input, const char* jsonstring, int debug);
```

JS Code

- 45,000 lines of JS Code
- Abstraction of File I/O
- Computational Modules
 - Automatic user interface generation
 - Automatic commandline generation
- User Interface packaged as Custom Web Elements

Formal Module Architecture

(Inputs, Outputs, Parameters specified using JSON)

JS

```
class SmoothImageModule extends BaseModule {  
  
  constructor() {  
    super();  
    this.name = 'smoothImage';  
  }  
  
  execute(vals) {  
    console.log('oooo executing: smoothImage with vals',vals);  
    return new Promise( (resolve, reject) => {  
      let input = this.inputs['input'];  
      let s = parseFloat(vals.sigma);  
  
      biswrap.initialize().then(() => {  
        this.outputs['output'] =  
          biswrap.gaussianSmoothImageWASM(input, {  
            "sigmas": [s, s, s],  
            "inmm": super.parseBoolean(vals.inmm),  
            "radiusfactor": parseFloat(vals.radiusfactor)  
          }, super.parseBoolean(vals.debug));  
        resolve();  
      }).catch( (e) => {  
        reject(e);  
      });  
    });  
  }  
}
```

Python

```
class smoothImage(bis_basemodule.baseModule):  
  
    def __init__(self):  
        super().__init__();  
        self.name='smoothImage';  
  
    def execute(self,vals):  
        print('oooo executing: smoothImage with vals', vals);  
  
        input = self.inputs['input'];  
        s = (vals['sigma']);  
  
        try:  
            self.outputs['output']=  
                libbis.gaussianSmoothImageWASM(input,  
                    paramobj={  
                        "sigmas": [s, s, s],  
                        "inmm": self.parseBoolean(vals['inmm']),  
                        "radiusfactor": vals['radiusfactor'],},  
                        debug=self.parseBoolean(vals['debug']))  
        except:  
            return False  
  
        return True
```


Web Components

```
<body oncontextmenu="return false;" ondragstart="return false;" ondrop="return false;">
  <bisweb-topmenubar id="viewer_menubar">
    </bisweb-topmenubar>
  <div id="viewerwidget">
    <bisweb-viewerlayoutelement
      id="viewer_layout"
      bis-sidewidth="310"
      bis-coreopen="false"
      bis-wholescreen="1"
      bis-defaulttext="">
    </bisweb-viewerlayoutelement>
    <bisweb-colormapcontrollerelement id="viewer_cmap">
      </bisweb-colormapcontrollerelement>
    <bisweb-orthogonalviewer
      id="viewer"
      bis-layoutwidgetid="#viewer_layout"
      bis-colormapeditorid="#viewer_cmap">
    </bisweb-orthogonalviewer>
    <bisweb-snapshotelement
      bis-layoutwidgetid="#viewer_layout"
      bis-dowhite="false"
      bis-viewerid="#viewer">
    </bisweb-snapshotelement>
```

```
<bisweb-simplealgorithmcontrollerelement
  id="algorithmelement"
  bis-viewerid="#viewer">
</bisweb-simplealgorithmcontrollerelement>

<bisweb-painttoolelement
  id="painttool"
  bis-layoutwidgetid="#viewer_layout"
  bis-viewerid="#viewer"
  bis-algorithmcontrollerid="#algorithmelement">
</bisweb-painttoolelement>

<bisweb-console id="bisconsole"></bisweb-console>

<bisweb-viewerapplication
  bis-menubarid="#viewer_menubar"
  bis-painttoolid="#painttool"
  bis-consoleid="#bisconsole"
  bis-viewerid="#viewer">
</bisweb-viewerapplication>

</div>

<bisweb-botmenubar></bisweb-botmenubar>
</body>
</html>
```

Web Components

```
<body oncontextmenu="return false;" ondragstart="return false;" ondrop="return false;">
  <bisweb-topmenubar id="viewer_menubar">
    </bisweb-topmenubar>
  <div id="viewerwidget">
    <bisweb-viewerlayoutelement
      id="viewer_layout"
      bis-sidewidth="310"
      bis-coreopen="false"
      bis-wholescreen="1"
      bis-defaulttext="">
    </bisweb-viewerlayoutelement>
    <bisweb-colormapcontrollerelement id="viewer_cmap">
      </bisweb-colormapcontrollerelement>
    <bisweb-orthogonalviewer
      id="viewer"
      bis-layoutwidgetid="#viewer_layout"
      bis-colormapeditorid="#viewer_cmap">
    </bisweb-orthogonalviewer>
    <bisweb-snapshotelement
      bis-layoutwidgetid="#viewer_layout"
      bis-dowhite="false"
      bis-viewerid="#viewer">
    </bisweb-snapshotelement>
```

```
<bisweb-simplealgorithmcontrollerelement
  id="algorithmelement"
  bis-viewerid="#viewer">
</bisweb-simplealgorithmcontrollerelement>

<bisweb-painttoolelement
  id="painttool"
  bis-layoutwidgetid="#viewer_layout"
  bis-viewerid="#viewer"
  bis-algorithmcontrollerid="#algorithmelement">
</bisweb-painttoolelement>

<bisweb-console id="bisconsole"></bisweb-console>

<bisweb-viewerapplication
  bis-menubarid="#viewer_menubar"
  bis-painttoolid="#painttool"
  bis-consoleid="#bisconsole"
  bis-viewerid="#viewer">
</bisweb-viewerapplication>

</div>

<bisweb-botmenubar></bisweb-botmenubar>
</body>
</html>
```

Data Provenance

- Store data provenance information as JSON-extensions to files
- Who generated this file, when, where, with what and how?
- For images embed the JSON info as extensions to NIFTI Headers
- For matrices and transformations we are formalizing new JSON based file formats
- Same applies to higher level outputs (e.g. motion parameters)

Data Format Example

```
{
  "bisformat": "BisDataObjectCollection",
  "filename": "testdata/test_motion_correction__test_motion_correction_
  _mot.json",
  "comments": [
    {
      "ModuleOutput": {
        "command": "/usr/bin/node /home/xenios/javascript/biscomp
        lib/js/bin/bisweb.js motionCorrection -t testdata/test_motion_correctio
        n.nii.gz -r testdata/test_motion_correction.nii.gz --doreslice true -o /
        home/xenios/winhome/Desktop/motionparams.json --resliced /home/xenios/wi
        nhome/Desktop/motioncorrect.nii.gz",
        "output": "output",
        "parameters": {
          "doreslice": true,
          "norm": true,
          "intscale": 1,
          "numbins": 1024,
          "extrasmoothing": 0,
          "metric": "CC",
          "optimization": "Hillclimb",
          "stepsize": 0.25,
          "levels": 3,
          "iterations": 32,
          "resolution": 1.01,
          "debug": false,
          "steps": 4,
          "refno": 0
        },
        "systeminfo": {
          "os": "linux",
          "arch": "x64",
          "hostname": "z230pc",
          "user": "xenios",
          "date": "2018-04-02T20:13:52.377Z",
          "nodeversion": "v8.9.4",
          "biswebversion": "04/02/2018"
        }
      }
    }
  ],
  "numitems": 5
```

```

  "numitems": 5,
  "itemlist": [
    {
      "type": "transform",
      "data": "{\"bisformat\":\"BisLinearTransformation\",\"filen
      ame\":\"identity.matr\",\"comments\":[],\"matrix\":[[1,0,0,0],[0,1,0,0]
      ,[0,0,1,0],[0,0,0,1]],\"parameters\":[0,0,0,0,0,0]}",
      "metadata": {
        "frame": 0
      }
    },
    {
      "type": "transform",
      "data": "{\"bisformat\":\"BisLinearTransformation\",\"filen
      ame\":\"identity.matr\",\"comments\":[],\"matrix\":[[0.9049245715141296
      ,-0.425572007894516,0,61.54688262939453],[0.425572007894516,0.904924571
      5141296,0,-39.15327453613281],[0,0,1,0],[0,0,0,1]],\"parameters\":[2.65
      1249885559082,-1.7674999237060547,0,0,0,-25.186874389648438]}",
      "metadata": {
        "frame": 1
      }
    },
    {
      "type": "transform",
      "data": "{\"bisformat\":\"BisLinearTransformation\",\"filen
      ame\":\"identity.matr\",\"comments\":[],\"matrix\":[[0.9049245715141296
      ,0.425572007894516,0,-39.15327453613281],[-0.425572007894516,0.90492457
      15141296,0,61.54688262939453],[0,0,1,0],[0,0,0,1]],\"parameters\":[-1.7
      674999237060547,2.651249885559082,0,0,0,25.186874389648438]}",
      "metadata": {
        "frame": 2
      }
    },
    {
      "type": "transform",
      "data": "{\"bisformat\":\"BisLinearTransformation\",\"filen
      ame\":\"identity.matr\",\"comments\":[],\"matrix\":[[0.9081797003746033
      ,0,-0.41858047246932983,34.94537353515625],[0,1,0,0],[0.418580472469329
      83,0,0.9081797003746033,-43.92396545410156],[0,0,0,1]],\"parameters\":[
      0.8837499618530273,0,-1.7674999237060547,0,-24.744998931884766,0]}",
      "metadata": {
        "frame": 3
      }
    }
  ]
}
```

Data Format Example

```
{
  "bisformat": "BisDataObjectCollection",
  "filename": "testdata/test_motion_correction__test_motion_correction__mot.json",
  "comments": [
    {
      "ModuleOutput": {
        "command": "/usr/bin/node /home/xenios/javascript/biscliplib/js/bin/bisweb.js motionCorrection -t testdata/test_motion_correction.nii.gz -r testdata/test_motion_correction.nii.gz --doreslice true -o /home/xenios/winhome/Desktop/motionparams.json --resliced /home/Desktop/motioncorrect.nii.gz",
        "output": "output",
        "parameters": {
          "doreslice": true,
          "norm": true,
          "intscale": 1,
          "numbins": 1024,
          "extrasmoothing": 0,
          "metric": "CC",
          "optimization": "Hillclimb",
          "stepsize": 0.25,
          "levels": 3,
          "iterations": 32,
          "resolution": 1.01,
          "debug": false,
          "steps": 4,
          "refno": 0
        }
      },
      "systeminfo": {
        "os": "linux",
        "arch": "x64",
        "hostname": "z230pc",
        "user": "xenios",
        "date": "2018-04-02T20:13:52.377Z",
        "nodeversion": "v8.9.4",
        "biswebversion": "04/02/2018"
      }
    }
  ],
  "numitems": 5
}
```

```
      "ModuleOutput": {
        "command": "/usr/bin/node /home/xenios/javascript/biscliplib/js/bin/bisweb.js motionCorrection -t testdata/test_motion_correction.nii.gz -r testdata/test_motion_correction.nii.gz --doreslice true -o /home/xenios/winhome/Desktop/motionparams.json --resliced /home/Desktop/motioncorrect.nii.gz",
        "output": "output",
        "parameters": {
          "doreslice": true,
          "norm": true,
          "intscale": 1,
          "numbins": 1024,
          "extrasmoothing": 0,
          "metric": "CC",
          "optimization": "Hillclimb",
          "stepsize": 0.25,
          "levels": 3,
          "iterations": 32,
          "resolution": 1.01,
          "debug": false,
          "steps": 4,
          "refno": 0
        }
      },
      "systeminfo": {
        "os": "linux",
        "arch": "x64",
        "hostname": "z230pc",
        "user": "xenios",
        "date": "2018-04-02T20:13:52.377Z",
        "nodeversion": "v8.9.4",
        "biswebversion": "04/02/2018"
      }
    }
  ],
  "numitems": 5
}
```

Data Format Example

```
{
  "bisformat": "BisDataObjectCollection",
  "filename": "testdata/test_motion_correction__test_motion_correction__mot.json",
  "comments": [
    {
      "ModuleOutput": {
        "command": "/usr/bin/node /home/xenios/javascript/bisweb.js motionCorrection -t testdata/test_motion_correction.nii.gz -r testdata/test_motion_correction.nii.gz --doreslice true -o /home/xenios/winhome/Desktop/motionparams.json --resliced /home/Desktop/motioncorrect.nii.gz",
        "output": "output",
        "parameters": {
          "doreslice": true,
          "norm": true,
          "intscale": 1,
          "numbins": 1024,
          "extrasmoothing": 0,
          "metric": "CC",
          "optimization": "Hillclimb",
          "stepsize": 0.25,
          "levels": 3,
          "iterations": 32,
          "resolution": 1.01,
          "debug": false,
          "steps": 4,
          "refno": 0
        }
      },
      "systeminfo": {
        "os": "linux",
        "arch": "x64",
        "hostname": "z230pc",
        "user": "xenios",
        "date": "2018-04-02T20:13:52.377Z",
        "nodeversion": "v8.9.4",
        "biswebversion": "04/02/2018"
      }
    }
  ],
  "numitems": 5
}
```

```
      "ModuleOutput": {
        "command": "/usr/bin/node /home/xenios/javascript/bisweb.js motionCorrection -t testdata/test_motion_correction.nii.gz -r testdata/test_motion_correction.nii.gz --doreslice true -o /home/xenios/winhome/Desktop/motionparams.json --resliced /home/Desktop/motioncorrect.nii.gz",
        "output": "output",
        "parameters": {
          "doreslice": true,
          "norm": true,
          "intscale": 1,
          "numbins": 1024,
          "extrasmoothing": 0,
          "metric": "CC",
          "optimization": "Hillclimb",
          "stepsize": 0.25,
          "levels": 3,
          "iterations": 32,
          "resolution": 1.01,
          "debug": false,
          "steps": 4,
          "refno": 0
        }
      },
```

```
      "systeminfo": {
        "os": "linux",
        "arch": "x64",
        "hostname": "z230pc",
        "user": "xenios",
        "date": "2018-04-02T20:13:52.377Z",
        "nodeversion": "v8.9.4",
        "biswebversion": "04/02/2018"
      }
    }
  ],
  "numitems": 5
}
```

```
      "metadata": {
        "frame": 3
      }
    }
  ],
  "numitems": 5
}
```

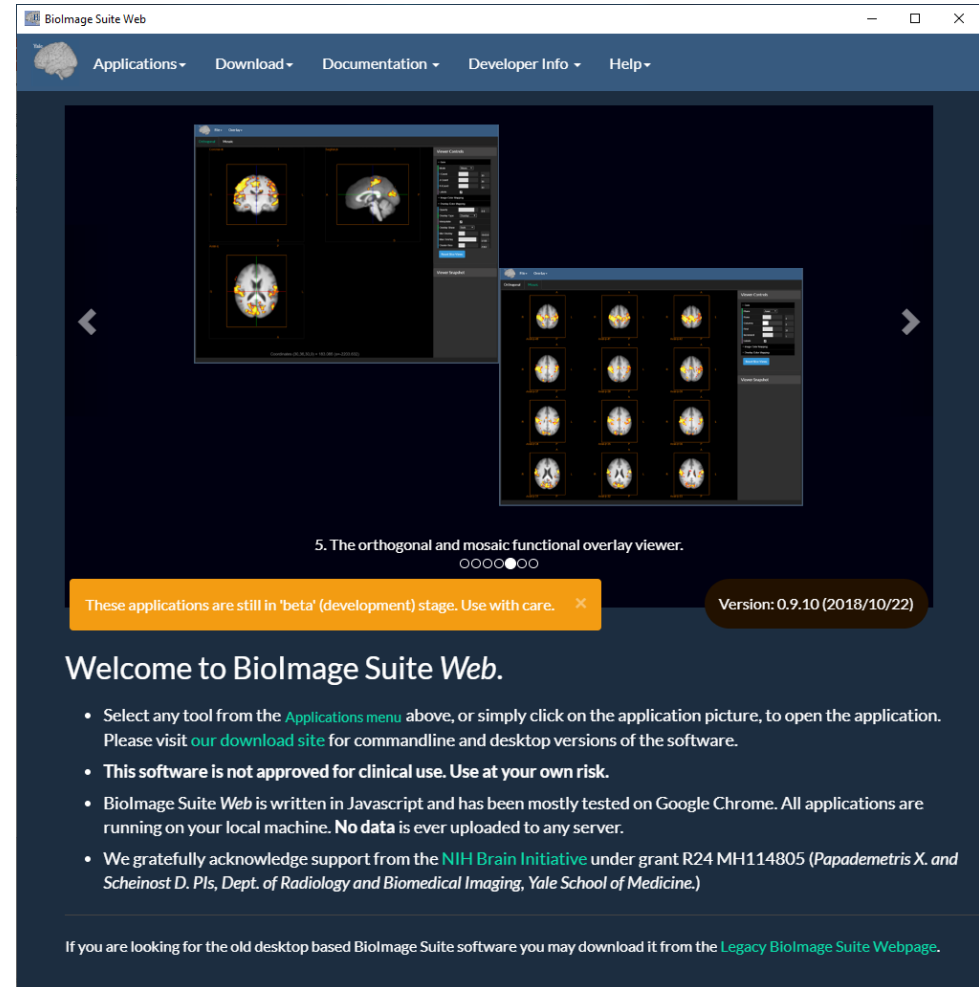
Some Limitations

- There are some cases where the web (alone) is not enough
- Reading multiple files from disk at the same time is not easy to do in a browser
 - Each file | open and file | save operation must be approved by the user (security)
- Solution I – Cloud Storage (Amazon S3 and Local Server)
 - We provide a simple script to allow browser to access your filesystem directly
- Solution II – Desktop Version
 - Electron-based Desktop Apps

Desktop Apps

- Must be downloaded + installed
- Essentially package a version of Chrome (Electron) and our code
- Can access filesystem directly
- Looks more or less same as web-based version

Desktop Version



5. The orthogonal and mosaic functional overlay viewer.

These applications are still in 'beta' (development) stage. Use with care. ✕

Version: 0.9.10 (2018/10/22)

Welcome to Biolmage Suite Web.

- Select any tool from the [Applications menu](#) above, or simply click on the application picture, to open the application. Please visit [our download site](#) for commandline and desktop versions of the software.
- This software is not approved for clinical use. Use at your own risk.
- Biolmage Suite Web is written in Javascript and has been mostly tested on Google Chrome. All applications are running on your local machine. **No data** is ever uploaded to any server.
- We gratefully acknowledge support from the [NIH Brain Initiative](#) under grant R24 MH114805 (*Papademetris X. and Scheinost D. Pls, Dept. of Radiology and Biomedical Imaging, Yale School of Medicine.*)

If you are looking for the old desktop based Biolmage Suite software you may download it from the [Legacy Biolmage Suite Webpage](#).

Command Line Apps

- Some times it is useful to “batch” process data
- We provide command line applications of the same modules as in the web-based version (same exact code)
- Download and Install
- Also need node.js (commandline JS interpreter)

```
Bash on Ubuntu on Windows
bw_thresholdImage -h
.... Using node.js version 8.11.2 (OK)
.... This program is part of the commandline suite of tools from BioImage Suite Web.
.... See https://github.com/bioimagesuiteweb/bisweb for more information.
....
,,,,
+++++ Setting forcing orientationOnLoad to: None (from None), None
,,,, bisweb commandline user preferences loaded from /home/xenios/.bisweb
,,,, {"orientationOnLoad":"None","snapshotScale":2,"snapshotDownWhite":true,"fileSource":"local","showWelcome":true,"favoriteFolders":[],"internal":false}
,,,,
Usage: bisweb.js thresholdImage [options]

Options:
  -V, --version          output the version number
  --low [n]              The threshold below which values will be classified as 'low'
  --high [n]             The value above which values will be classified as 'high'
  --replacein [s]        If true, values classified as 'in' will be replaced with 's'
  --replaceout [s]       If true, values classified as 'out' will be replaced with 's'
  --inval [n]            Value to replace 'in' values with
  --outval [n]           Value to replace 'out' values with
  --outtype [s]          Output Type
  --debug [s]           Toggles debug logging
  -i --input <s>         The image to be processed
  -o --output <s>        Save the output image
  --paramfile [s]       Specifies that parameters should be read from the file 's'
  --silent               Run in silent mode (no output on the console)
  -h, --help            output usage information
```

Threshold Image

Inputs

Outputs

Parameters

Low Threshold 21

High Threshold 208

Advanced

Replace 'in'

Replace 'out'

'in' Value

'out' Value

Output Type

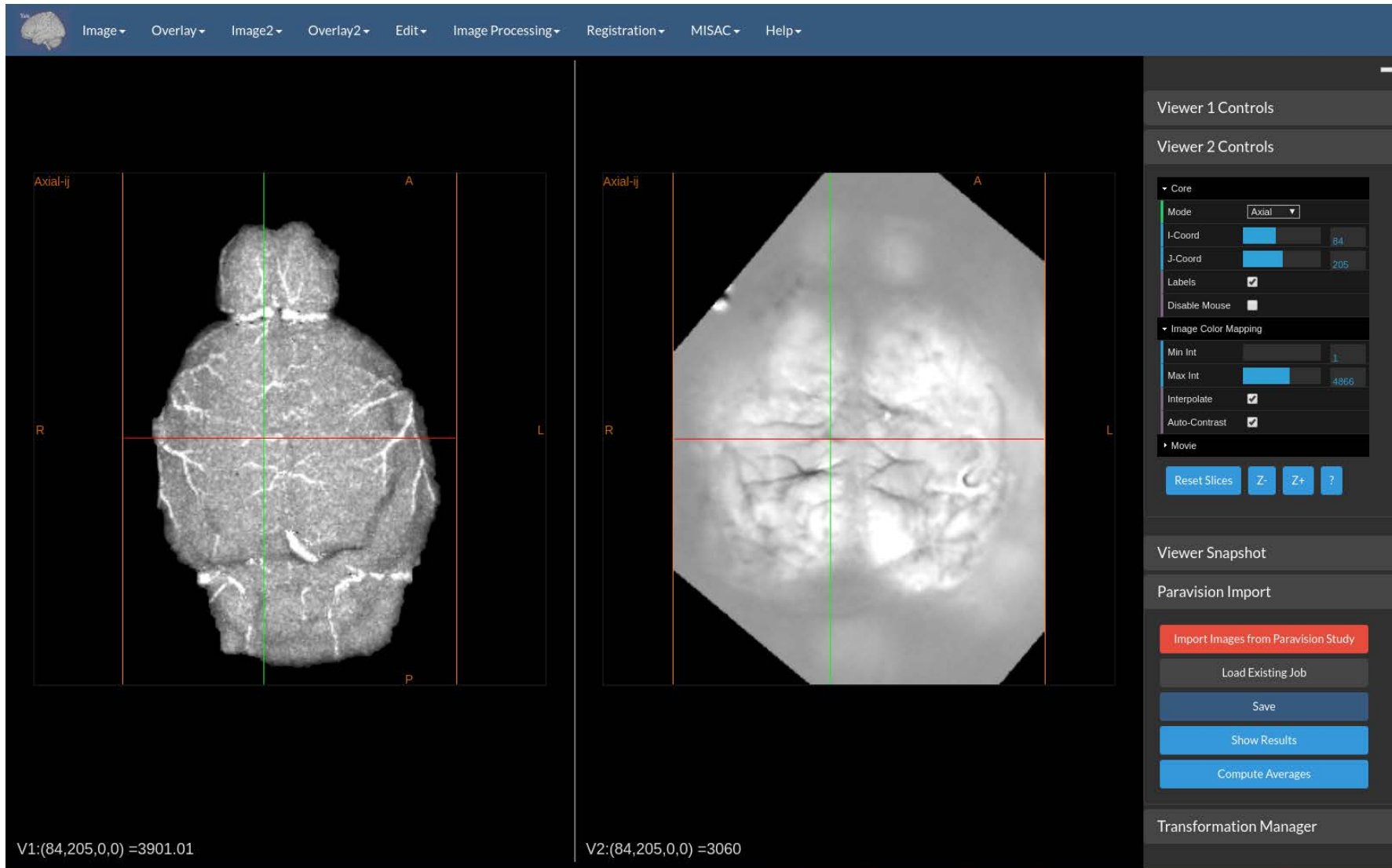
Debug

Threshold Undo More ▾

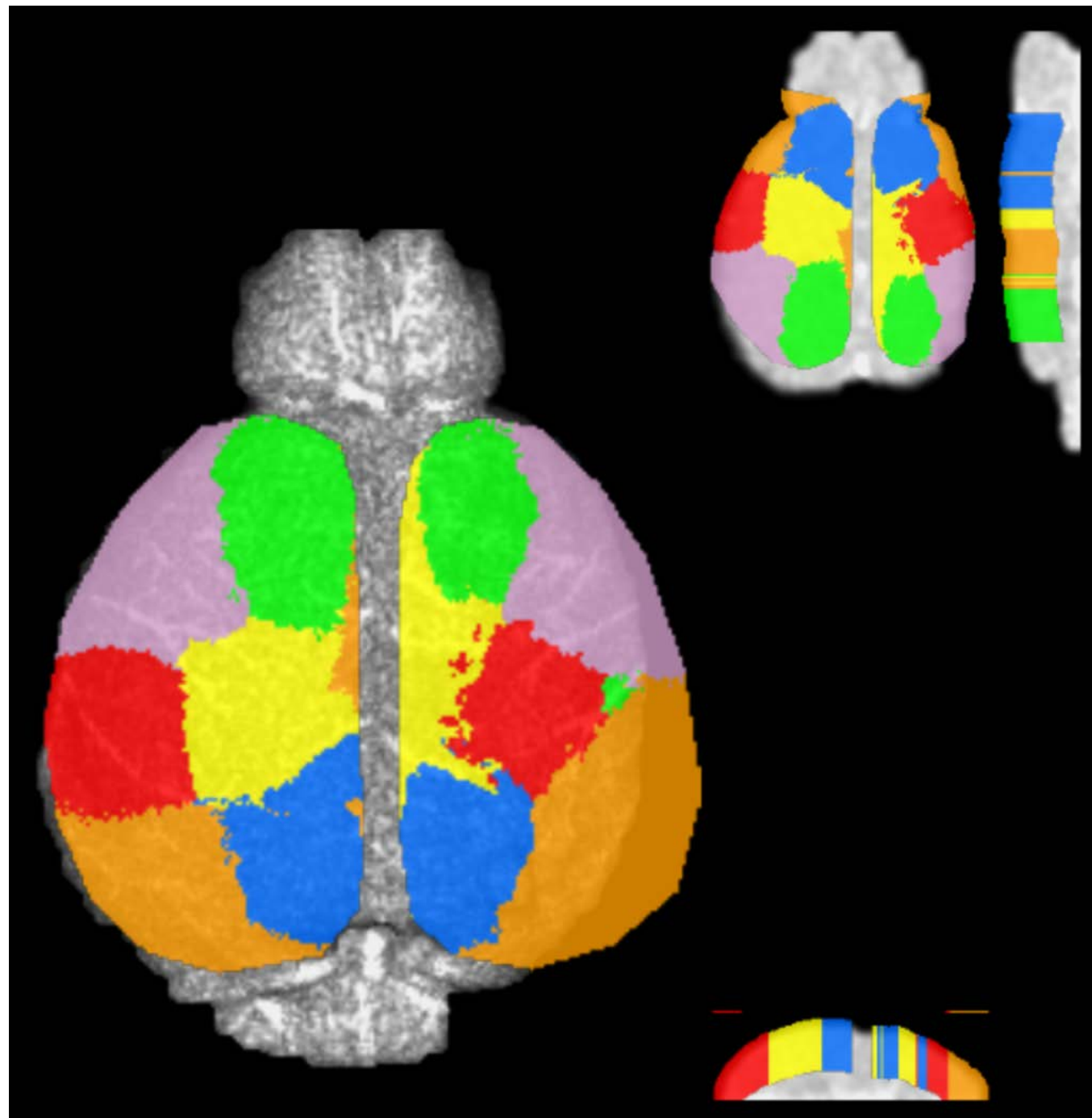
Some Cool Figures

- Simultaneous fMRI Ca(2+) Imaging Project (Crair, Constable PIs)
- Images using data from Eve Lake and Xinxin GE
- Custom code to merge 2D Projected Images and 3D MRI

Mapping 2D Optical to fMRI (via Angio MRI)



Mapping Parcellations from Optical to MRI



Lots more

- Google Drive and Dropbox integration (read only)
- Progressive Web Application (offline use via caching)
- All code on github
- Lots of programmer documentation on JS/WebAssembly etc. on Github (under docs).
- Python and Matlab (early) integration of computational code.
- Paravision Import functionality for small animal MRI
- Port of old diff-SPECT code for Epilepsy
- Landmark placement and editing
- Interactive Registration
- Copy & Paste Viewers across applications (via browser cache)
- Arbitrary Image Orientation and Registration (plus force “on load” if desired)

Current State

- Software is available (web, desktop, source code etc.)
- Closing in on v1.0 release (0.99!) – probably next week
 - Focus has been on getting low level functionality in place
- Next
 - Data organization (DICOM, BIDS)
 - Processing pipeline for resting state fMRI and optical Ca 2+ images
 - Better cloud integration
 - Machine learning via tensorflow.js
 - Form a basis for all future development work

Acknowledgments

- Zach Saltzman
- John Onofrey
- Cheryl Lacadie
- Haley Garbus
- Dana Casetti
- Daniel Coman
- Basav Ganganna
- Pranav Chandrasekaran
- Users, testers, ...
- Eigen
- CMake/CTest/CPack
- Emscripten
- Node.js
- Gulp
- Webpack
- Bootstrap
- JQuery
- dat.gui
- Three.js
- Electron
- Electron-Packager