



# Deliverables: 23 October 2017



—  
AVATR



# From Last Week

---

- Demo on how people currently annotate ✓
- Documentation on current annotation workflow ✓
- Jupyter Notebook on Ingest Large Vol ✓
- Help annotators setup code ✓

# This Week:

---

- Improve Documentation ✓
- Fix viewing annotations and potentially other bugs ✓
- Tech Eval of NDM ✓
- Work with NOMADs to use this ✓ (DEMO)
- Get feedback on Annotation workflow ✓

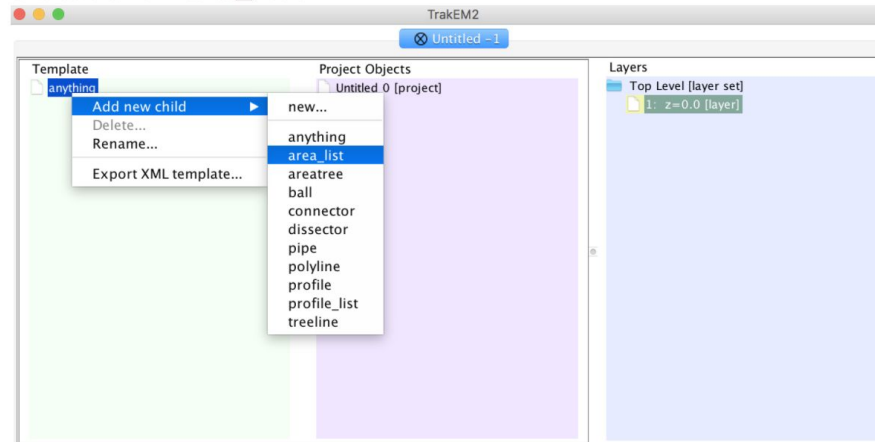
# Feedback from Annotation Squad

---

- Liked how easy it is to pull data from the boss
  - No “weird stuff in python”
  - Only a few bash commands
  - Store in computer easily
- Easy to annotate stuff and save
  - Good instructions on how to use FIJI
- Things happen quickly
- Kind of hard to tell what xyz parameters to use
  - Some sort of helpful tool?

# Documentation updated in Sphinx: FIJI annotations

- d. In your TrakEM2 properties, right click on "anything" in the template column and add a new "area\_list".



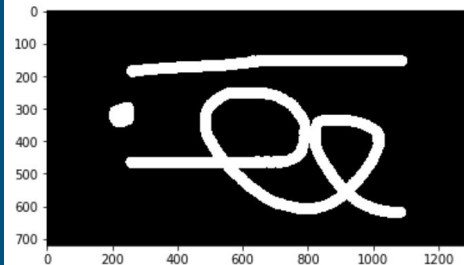
- e. Drag the entire "anything" folder into "Untitled 0" in the middle column.  
f. Right click the nested "anything" folder and add a "new area list"  
g. This should have changed your ImageJ canvas. Now, drag your volume from your folder into the canvas.  
h. In the popup window, make sure that "Resize canvas to fit stack" is checked.

# JNB + Script for Sparse Annotations

Next week, upload to datamanager formatted as BSON

1.8MB -> 3KB, saved as .npz for now

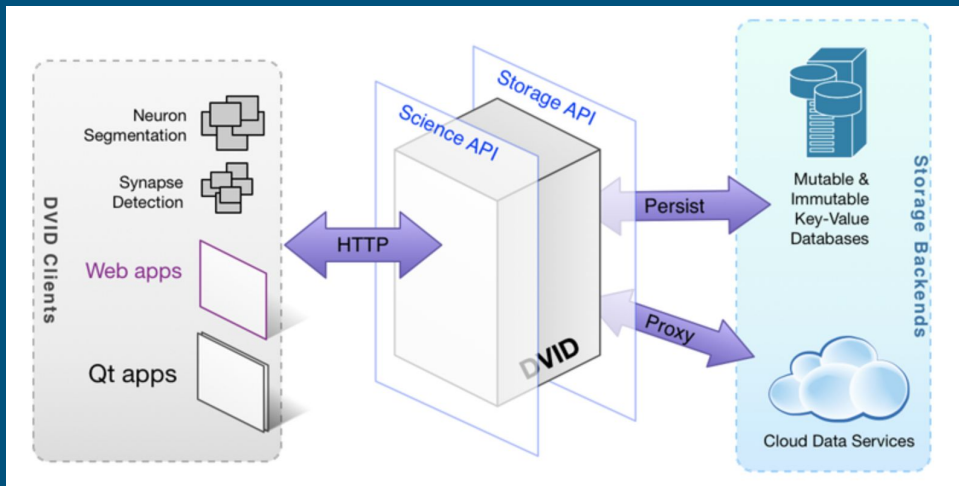
(720, 1280)



```
[<720x1280 sparse matrix of type '<type 'numpy.uint8'>'
  with 921600 stored elements in Compressed Sparse Row format>, <720x1280 sparse matrix of type '<type 'numpy.u
int8'>'
  with 921600 stored elements in Compressed Sparse Row format>, <720x1280 sparse matrix of type '<type 'numpy.u
int8'>'
  with 15124 stored elements in Compressed Sparse Row format>, <720x1280 sparse matrix of type '<type 'numpy.ui
nt8'>'
  with 116277 stored elements in Compressed Sparse Row format>]
```

# NeuroDataManager for Storing Annotations and Pipeline Outputs

- Supports data ingest and pulling cutouts from a data storage.
- Currently only supports .tif files.
- Requires NeuroGlancer precomputed datastore (should be able to work for S3 and Google Cloud).



# NeuroDataManager for Storing Annotations and Pipeline Outputs (Notes)

---

## Spatial access to large data volumes

- Pros:
  - Does what we want and need eventually, we think
    - Need more investigation since it is dependent on NeuroGlancer
      - Next week demo
- Cons:
  - Poor documentation (incomplete/not updated)
  - Building is hard cuz not python
  - Can it handle more than images?
  - Can't trace errors in code at all
  - Not ready?

```
(venv) Richard's MacBook Pro:build richardguo$ bin/ndm -datadir ./DATA/ -input hellowo
*** Aborted at 1508704814 (unix time) try "date -d @1508704814" if you are using GNU d
PC: @ 0x107ce0af4 dataArray_namespace::TiffArray32::load()
*** SIGBUS (@0x108b86ac0) received by PID 89331 (TID 0x7fffe64b63c0) stack trace: ***
@ 0x7fffd7b1b3a _sigtramp
@ 0x1110fe0cb (unknown)
@ 0x107cc16ff main
@ 0x7fffd5a2235 start
@ 0xe (unknown)
Bus error: 10
```



# Bugs and Quirks Discovered in Annotator

---

1. Sparse matrices aren't properly formatted with Scipy to be used outside of a .npz file format
  - a. Script of custom sparse annotation class in the works
2. Haven't solved z-indexing issue pulling data from BOSS
  - a. No "Issue" section in Intern so we're not sure what to do
3. Some tif stacks are read incorrectly in ingest\_large\_vol

# For Next week:

---

Wednesday: Talk to annotators to keep them up to date

- Make Documentation for NDM in Sphinx
  - Using NDM
- Demo of storing annotations in NeuroDataManager
- Tech Eval for NeuroGlancer