Breakout Session 6: Track B

Retinal Circuitry - Improving AI Readiness of Existing Retinal Connectomes

Dr. James Anderson
Senior Software Design Engineer, University of Utah - Moran Eye Center

Retinal Circuitry - Improving AI Readiness of Existing Retinal Connectomes

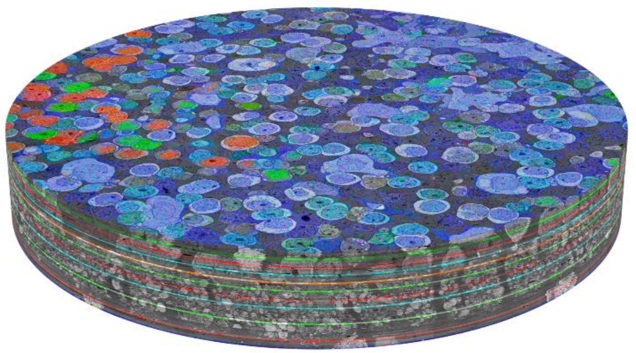
Presenter: James Anderson

PI: Bryan Jones

NOT-OD-22-67

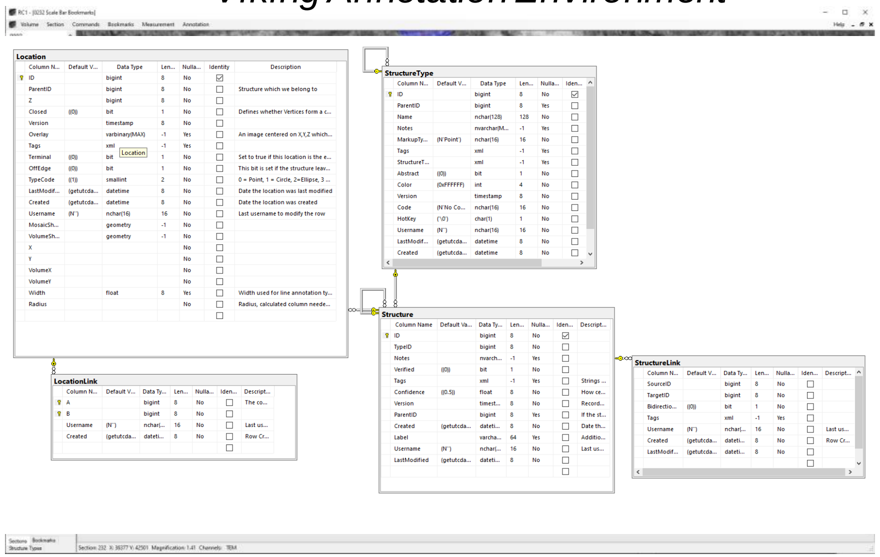
Connectomics Workflow

Volume Construction

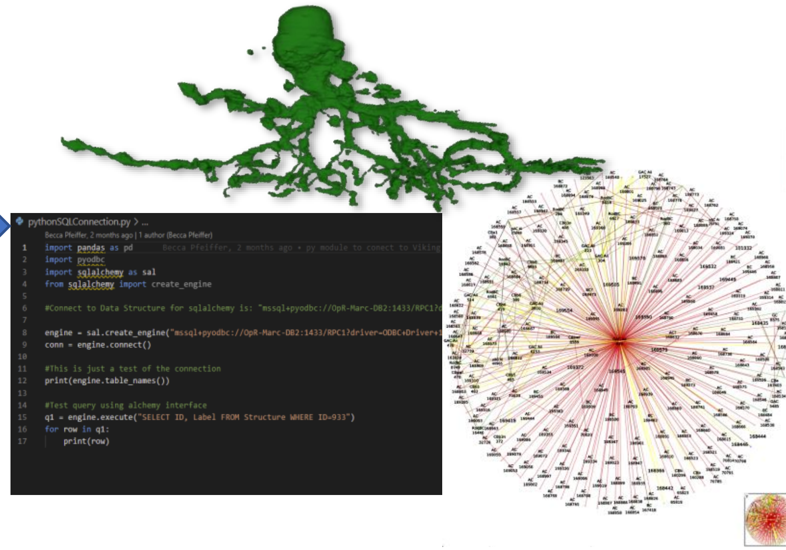


- ssTEM (70-90nm sections)
- 2.18nm/px resolution
- Interleaved small molecule labeling

Viking Annotation Environment



Analysis



Comparative Neuroanatomy

RC1
Rabbit retinal connectome
Status: Open
Database Annotations: ~1.3 million

RC2
Mouse retinal connectome
Status: Active
Database Annotations: ~500k

RC3
Primate (Macaque) retinal connectome
Status: Captured
Database Annotations: N/A

Pathoconnectomics

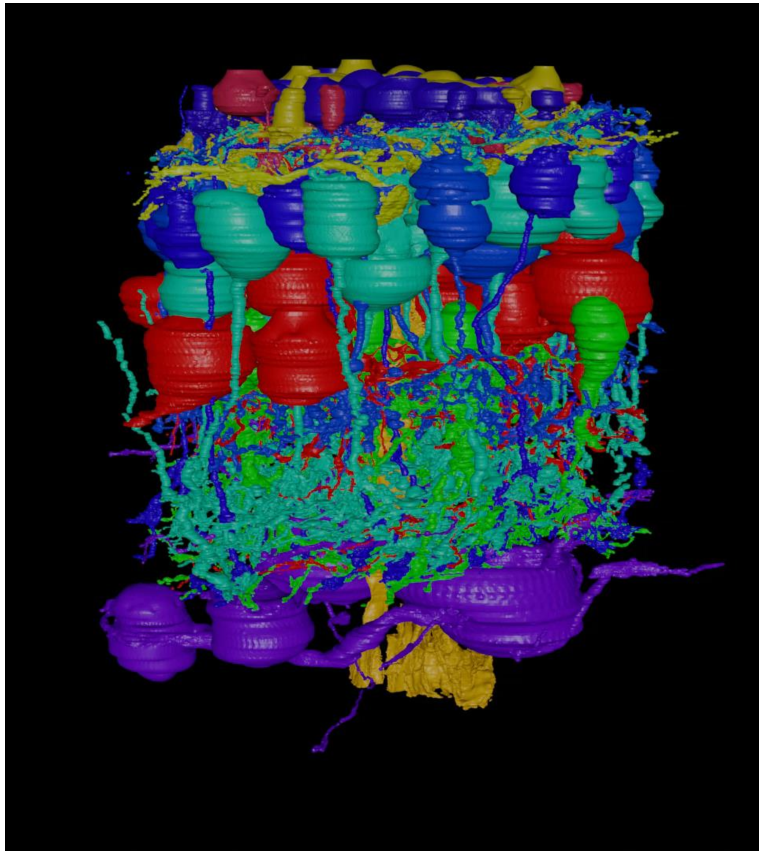
RPC1
Pathoconnectome from 10mo rabbit model of Retinitis Pigmentosa (Phase 1 retinal degeneration)
Status: Open
Database Annotations: ~280k

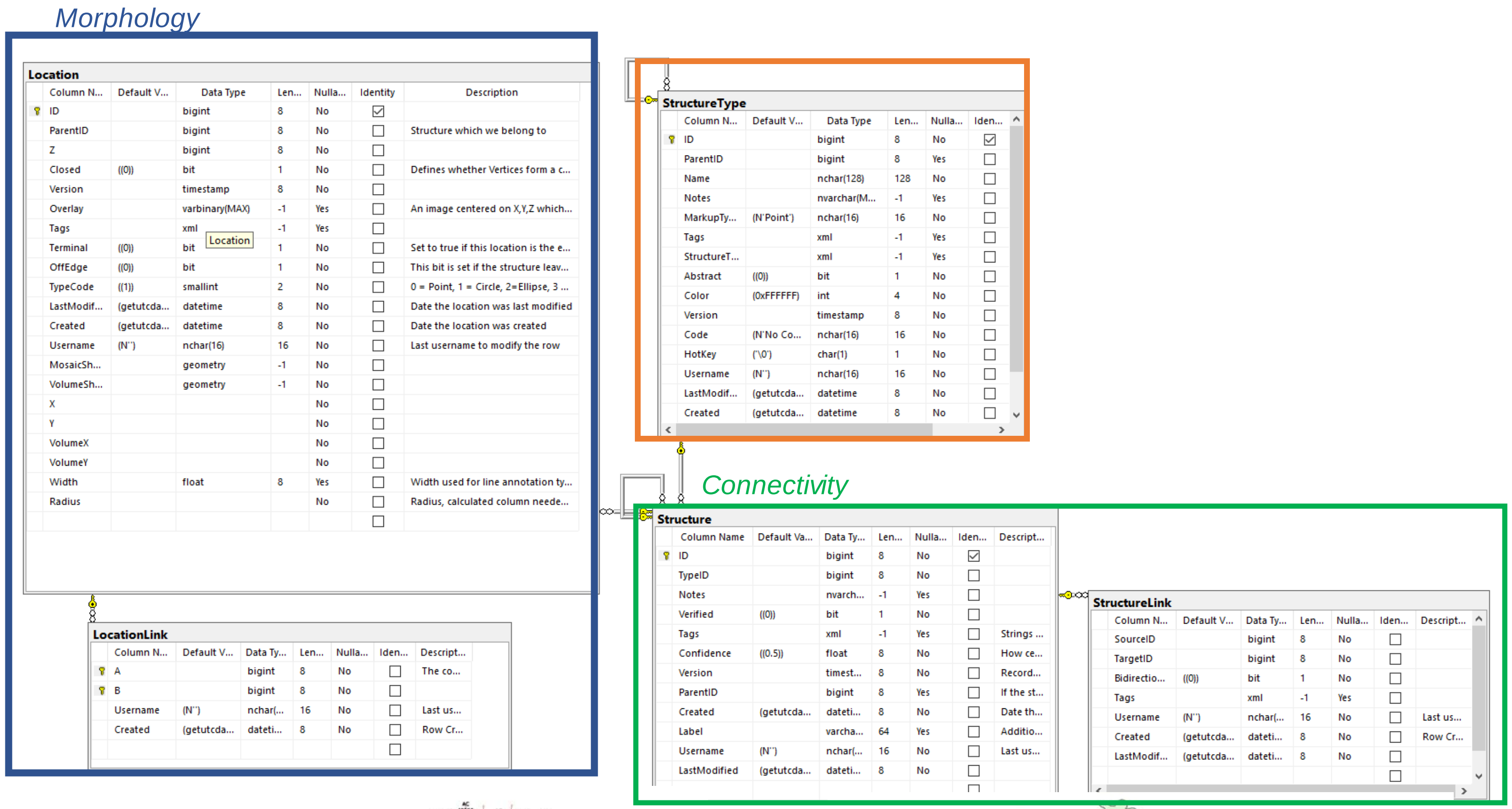
RPC2
Pathoconnectome from 2yo rabbit model of Retinitis Pigmentosa (Phase 2 retinal degeneration)
Status: Active
Database Annotations: ~100k

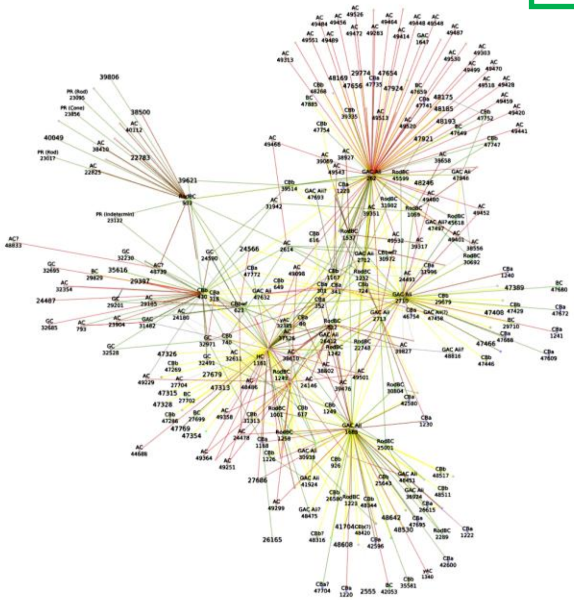
RPC3
Pathoconnectome from 4yo rabbit model of Retinitis Pigmentosa (Phase 3 retinal degeneration)
Status: Captured
Database Annotations: N/A

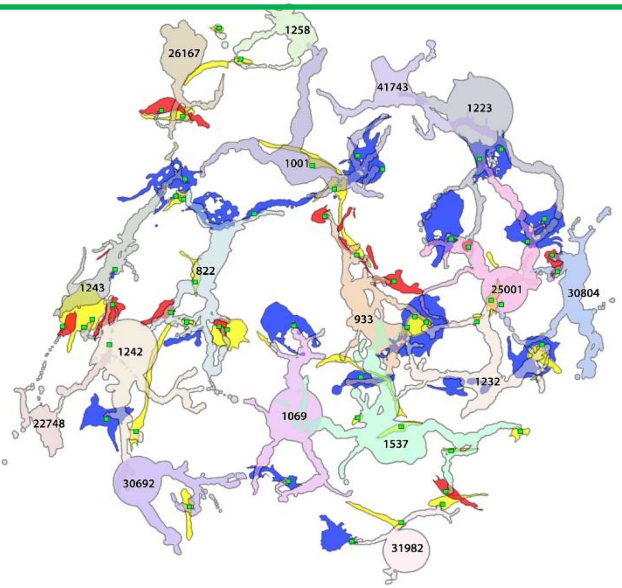
Database Features:

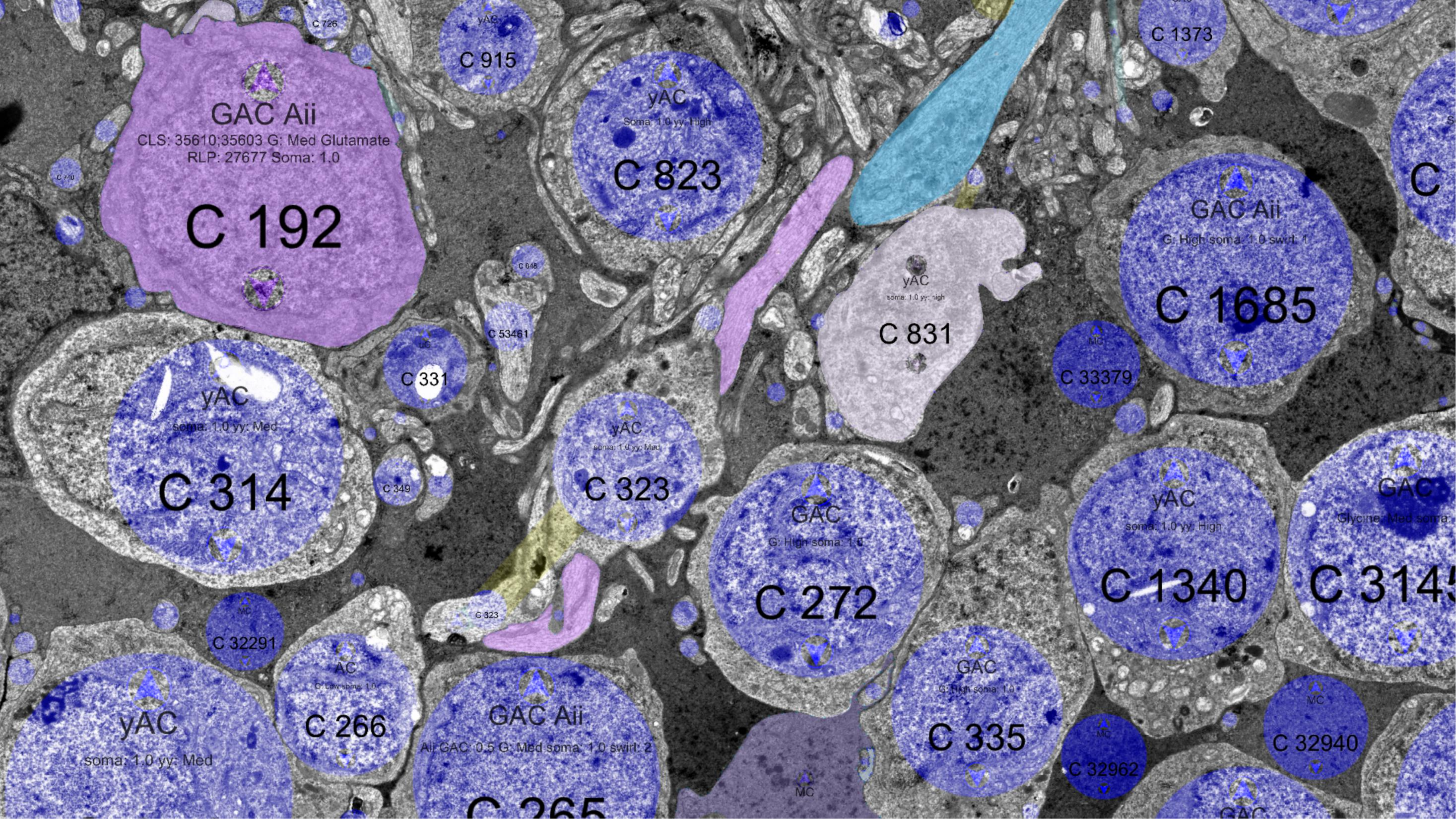
- SQL reduces development cost and maintenance. Very mature tools.
- Flexible addition of new structures
- Geometry columns encode annotations. Spatial queries within SQL. Enables encoding any geometric shape.
- Hierarchy describes relationships of biological structures (Parent=Cell, Child=Subcellular structures)
- Size, shape, and position are encoded in every annotation
- Morphology and connectivity are encoded in separate but relatable graphs. Allowing efficient querying.



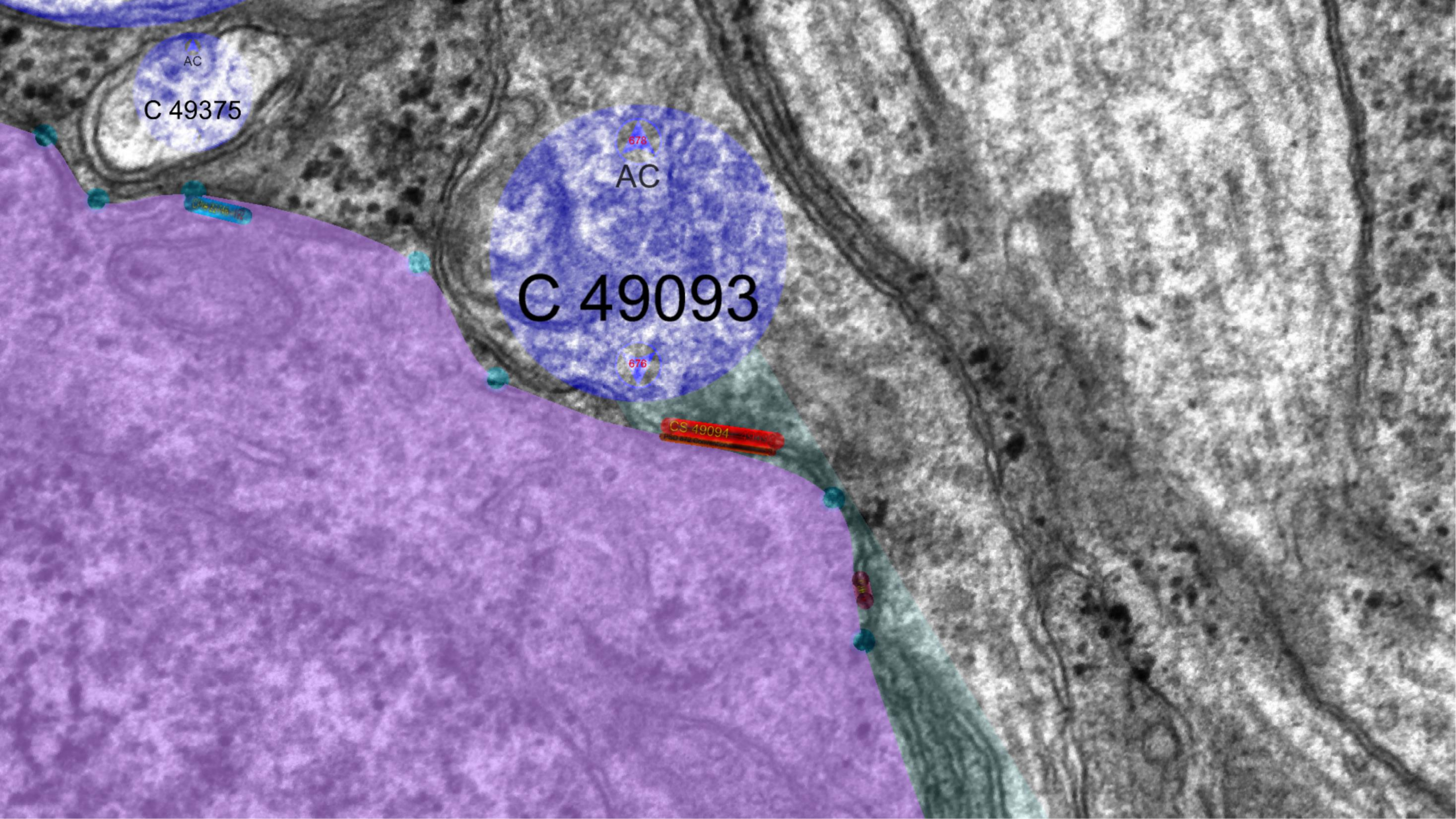




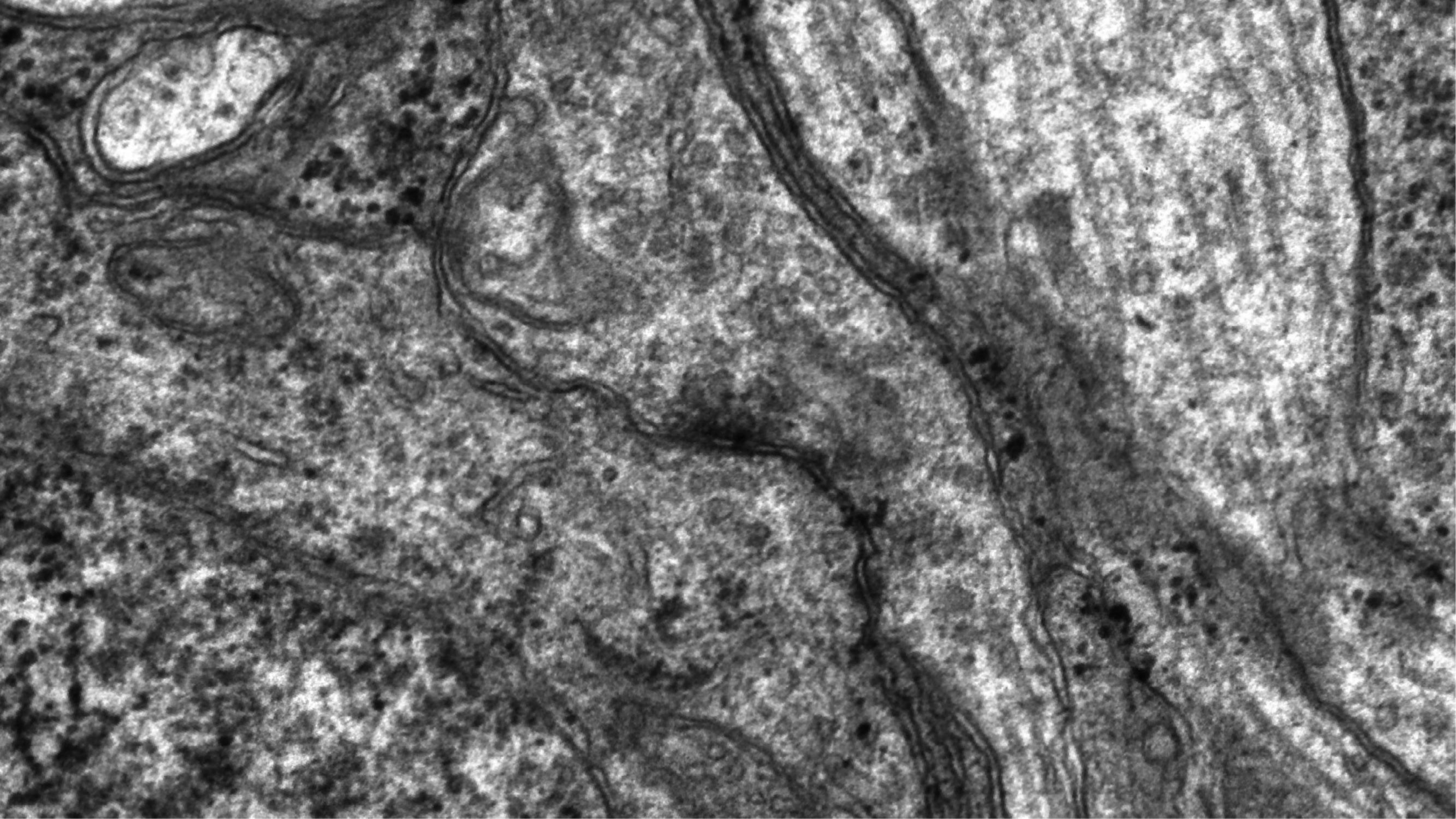




GAC Aii
CLS: 35610;35603 G: Med Glutamate
RLP: 27677 Soma: 1.0
C 192
C 726
yAC
C 915
yAC
Soma: 1.0 yy: High
C 823
C 1373
C 648
yAC
C 831
GAC Aii
G: High soma: 1.0 swirl: 1
C 1685
C 53461
C 331
C 33379
yAC
soma: 1.0 yy: Med
C 314
C 349
yAC
C 323
GAC
G: High soma: 1.0
C 272
yAC
soma: 1.0 yy: High
C 1340
GAC
Glycine: Med soma
C 3145
C 323
C 32291
AC
C 266
yAC
soma: 1.0 yy: Med
GAC Aii
Aii GAC: 0.5 G: Med soma: 1.0 swirl: 2
GAC
G: High soma: 1.0
C 335
C 32962
C 32940
MC



AC
C 49375
678
AC
C 49093
676
CS 49094



Data Sharing

Primary Goal: Make it simple for collaborators to access our combined annotation and image data to create machine learning algorithms to augment volume annotation efforts.

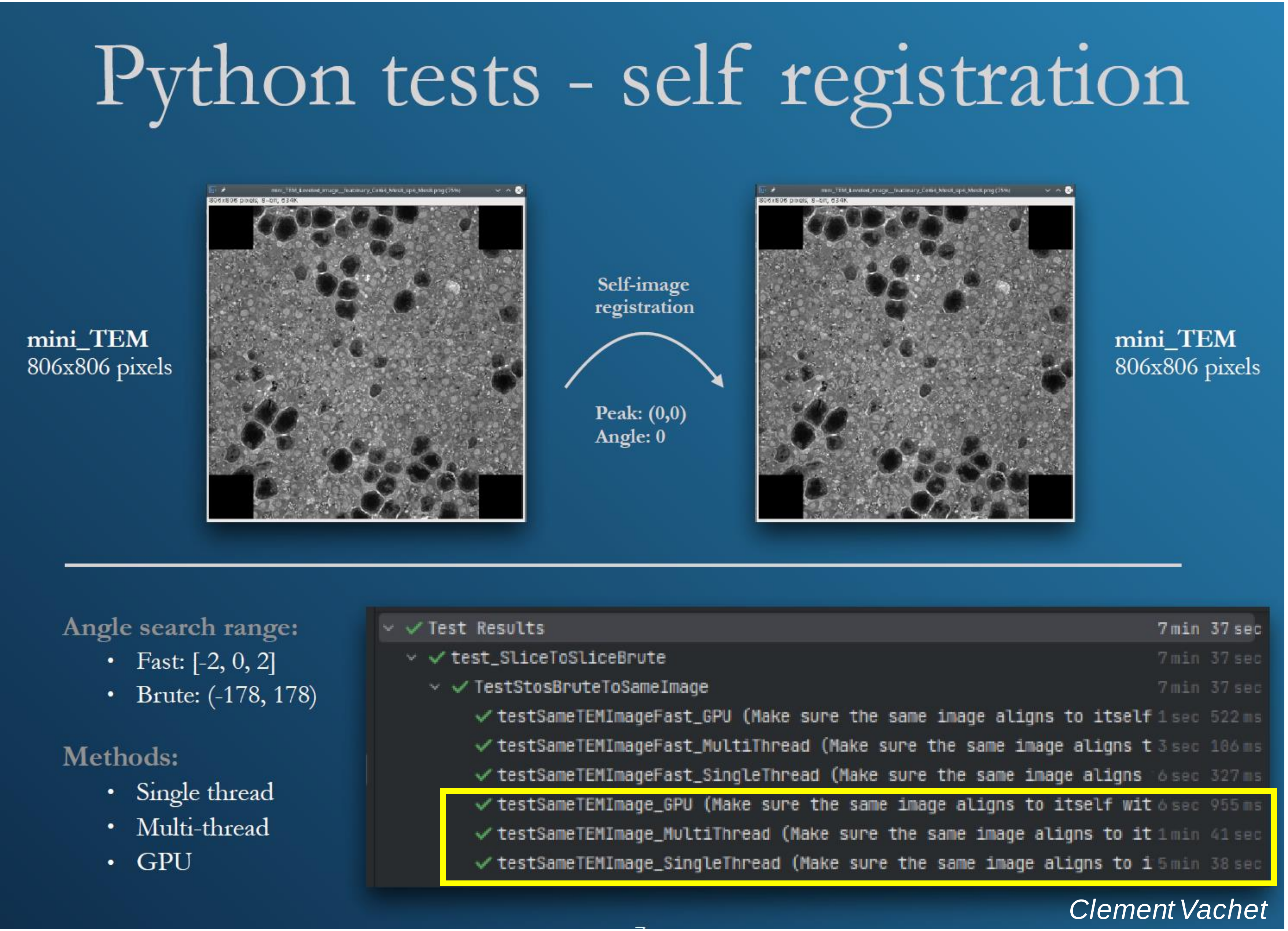
Annotations:

- OData works great for sharing annotations:
  - http://websvc1.connectomes.utah.edu/RC1/OData/
- Queryable, Readable directly into a Browser, Spreadsheet, or programming API
- Spatial data is plain text Open Geospatial Consortium format. Libraries exist to interpret it

Images:

- The goal is to ease accessibility for the image processing lab.
- Viking uses client GPU to perform transforms. Moving processing from client to server simplifies the client experience.
- Real-time transformation has been an advantage for adjusting volume registration during annotation phase.
- Attempts to share images:
  - 1st Approach : Client exports region-of-interest to disk
  - 2nd Approach : Cached RC1 volume available via HTTP, ex: http://storage1.connectomes.utah.edu/RC1VolumeRegisteredV2/RC1/234/Tiles/001/X160_Y049.png
  - 3rd Approach : Web API (Stopped at prototype due to performance issues)
  - 4th Approach : A stack of numpy arrays, either shipped in the mail or a long download

Progress Towards Goal: CuPy proof-of-concept



Registration requirements:
- FFT
- Random Number Generation
- Image Labeling
- Arithmetic
- Convolution

Assembly Requirements:
- Delaunay triangulation
- Texture mapping

Registration Speed

GPU	< 7 sec
Multi-core	101 sec

GPU is 14x faster at Registration!

Progress Towards Goal: GPU Implementation with CuPy

CuPy

CuPy is a NumPy/SciPy open-source array library
for GPU-accelerated computing with Python

Transforms	single CPU	multi CPU	partial GPU	full GPU
Rigid_NoRotation	✓	✓		✓
Rigid	✓	✓		✓
Centered Similary	✓	✓		✓
MeshWithRBFFallback	✓	✓	✓	
GridWithRBFFallback	✓	✓	✓	

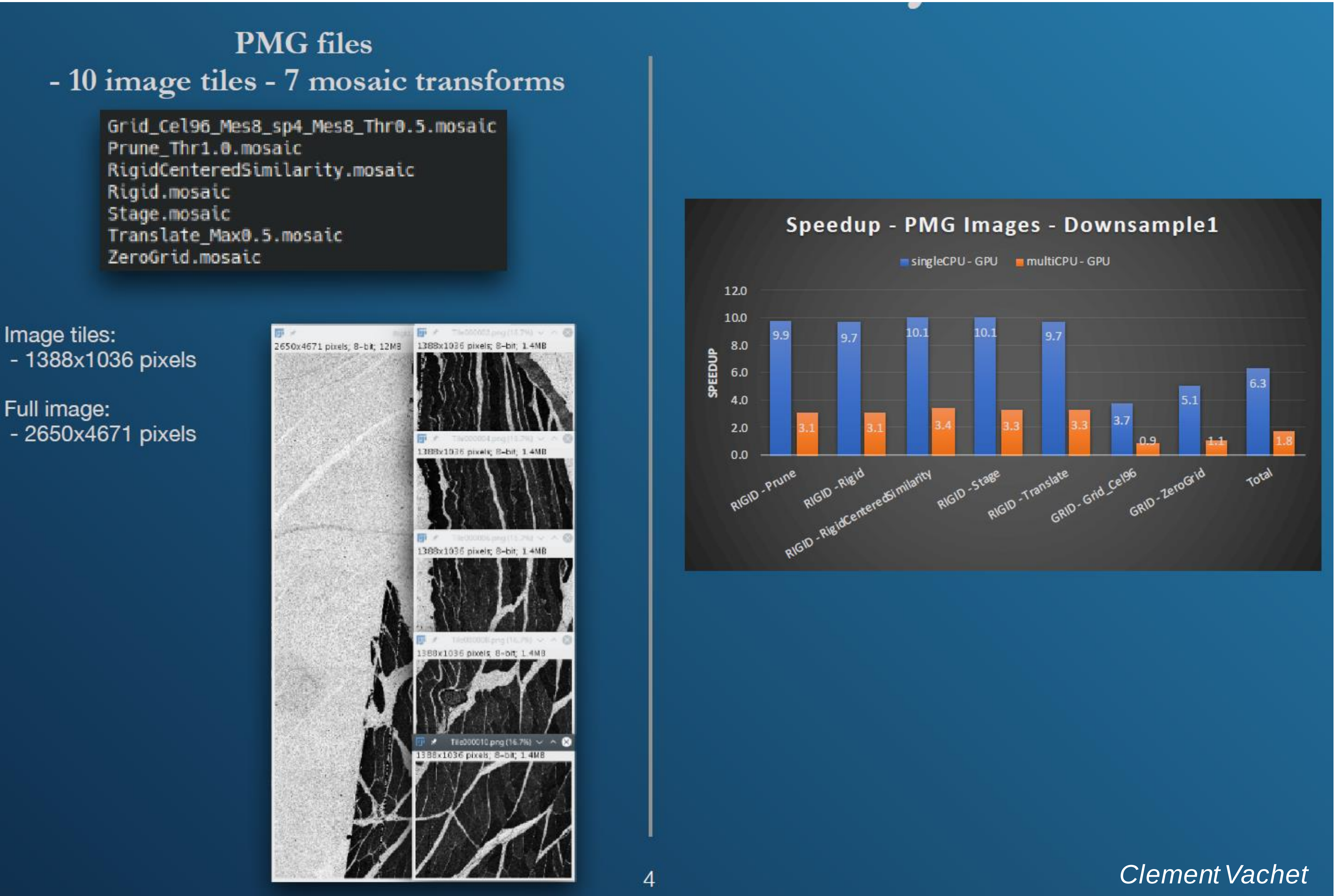
Implementation for assembly transforms

Note: some transform components (e.g. triangulation) haven't yet been implemented in CuPy

Clement Vachet

- CuPy lacked the LinearNDInterpolator function necessary for a full GPU implementation.
- The function was pulled into the next CuPy release two weeks ago
- Our CuPy version will be fully implemented upon next official cupy release

Progress Towards Goal: Tile Assembly



- Ignore the blue bar results.
- Yellow multi-core comparison ran on an 8-core desktop system.
- Beyond missing LinearNDInterpolate I suspect moving memory between CPU and GPU is a significant time cost. Most likely optimizable.

Progress and Ongoing Efforts

Tools

- Every critical path now utilizes CuPy (with the one exception discussed). Passes unit tests.
- Have support to export a stack of full resolution numpy files
- Waiting for next cupy version to fully test with GPU at scale

Docker Distribution

- Docker is simple to deploy and gives us easy access to our university high-performance computing center. This provides easier access to the command-line tool that can export registered volumes and will allow clients to run the Web Service locally for performance.
- All but one tool required to take images from capture to a 3D numpy array now run in a single docker image.
- We have contracted Kitware to port the remaining legacy C++ ITK-based tool to a web image to complete the docker image.

Web Service

- In progress.
- GPU performance determines if we revisit the original on-the-fly web service prototype or rely upon pre-generated images.

Legacy Data

- Original registered RC1 images available via HTTP are being copied into a stack of numpy arrays.
- RC1 raw data is now compatible with the latest code. Going from imported data to a registered volume is a great stress test for the new GPU code.

MarcLab for Connectomics:

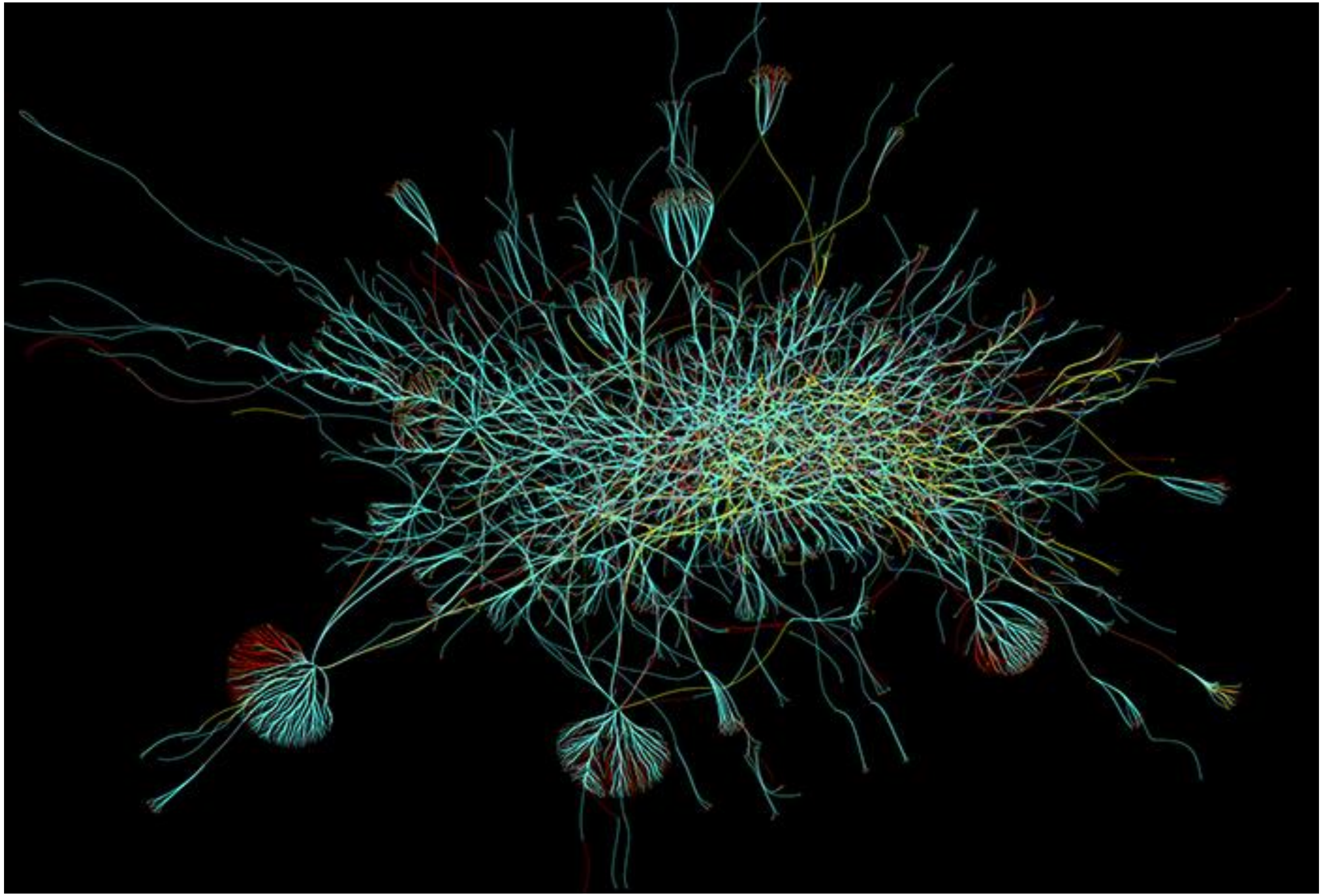
Bryan W. Jones (Director)
James Anderson
Rebecca Pfeiffer
Crystal Sigulinsky
Megan Croom
Jia Hui Yang
Matt Berardy
Selena Wirthlin
Taylor Otterness

Collaborators:

Clement Vachet *(Scientific Computing Institute, University of Utah)*
KitWare

Unmet CuPy Contributor:

Edgar Andrés Margffoy Tuay
(LinearNDInterpolator Author)



National Eye Institute

Supported By
Research to Prevent Blindness

NEI EY015128
NSF NeuroNex2 Grant (2014862)
NEI EY14800 Vision Core
Research to Prevent Blindness CDA
Unrestricted Grant from Gabe Newell