

Cortical Computing Circuits, Connectomics, and CAJAL3D

R. Jacob Vogelstein, PhD
Program Manager, Applied Neuroscience
Johns Hopkins University Applied Physics Laboratory

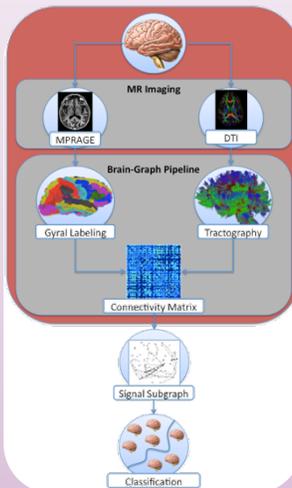
Today's Objectives

1. Initiate a conversation about the key missing ingredients in today's neuromorphic computing systems
2. Describe how connectomics may inform the next generation of neuromorphic computing systems
3. Introduce CAJAL3D and OpenConnecto.me as an "open-science" platform for applied research in connectomics

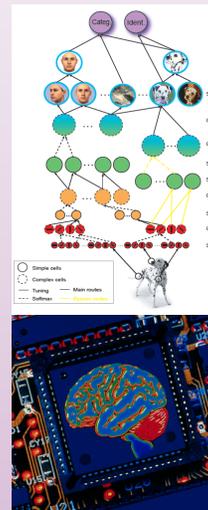
Applied Neuroscience @ APL

- APL's Applied Neuroscience programs aim to provide real-world solutions to our sponsors' critical challenges through applied research in three related disciplines:

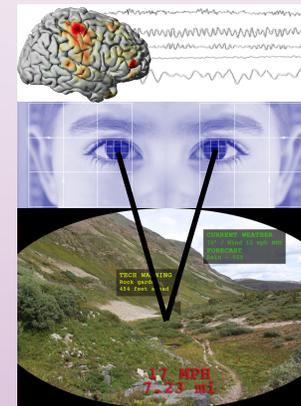
Neural Basis of Excellence



Neuromimetic Computing



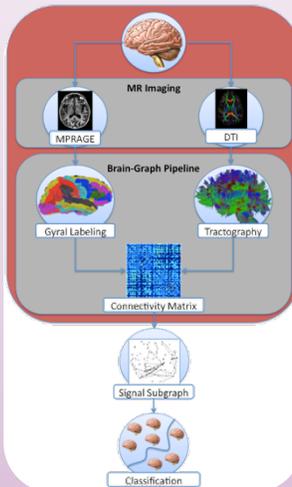
Brain-Computer Interfaces



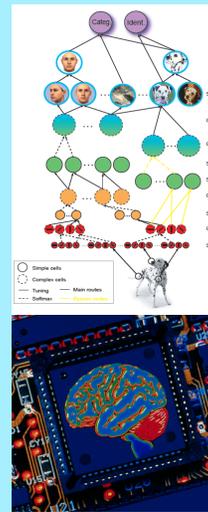
Applied Neuroscience @ APL

- APL's Applied Neuroscience programs aim to provide real-world solutions to our sponsors' critical challenges through applied research in three related disciplines:

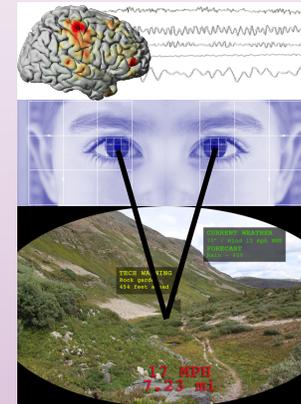
Neural Basis of Excellence



Neuromimetic Computing



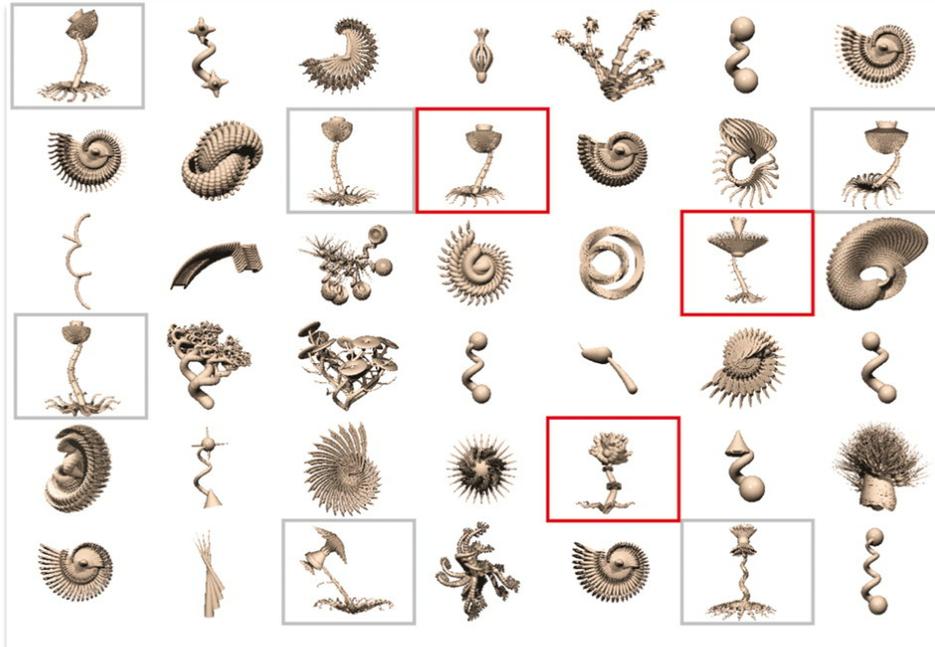
Brain-Computer Interfaces



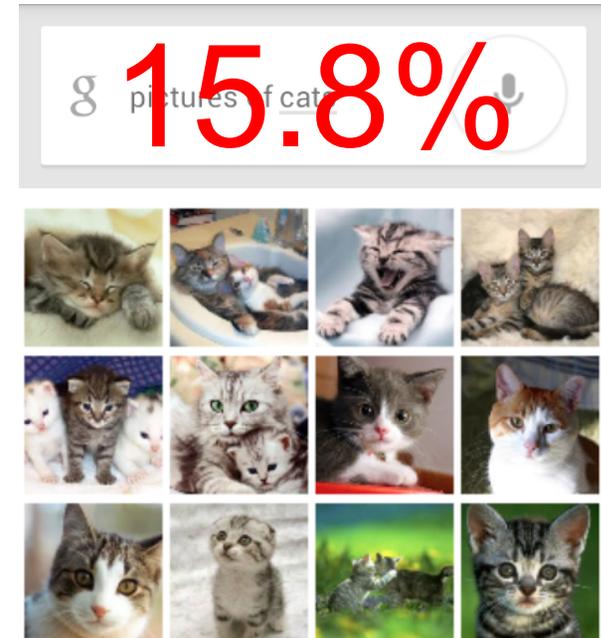
- Neuromimetic computing research objective: Build novel machine learning and pattern recognition systems using the algorithms employed by the brain

Motivation (1 of 2)

- Brain as existence proof of a better learning and pattern recognition system than today's state of the art computing systems (for at least some types of problems)
 - One-shot learning
 - Transfer learning
 - Learning in noise or distortion



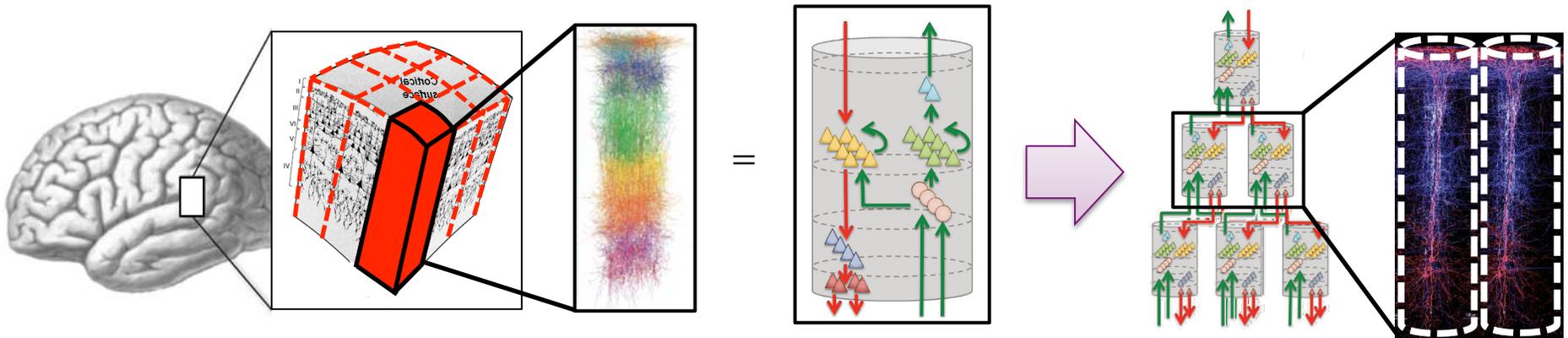
Tenenbaum et al., Science, 2011



Google, 2012

Motivation (2 of 2)

- Not a black box: converging evidence (from electrophysiology, neuroanatomy, information theory, and computational modeling) that the brain employs algorithms composed of hierarchical networks of repeated computing “primitives”
 - Primitives: defined here as recurring neural circuit motifs consisting of $10^2 - 10^4$ neurons and occupying approximately $0.1 - 1.0 \text{ mm}^2$ of cortical surface area
 - Presumed to represent basic and repeated operations and transformations that a particular brain region uses as building blocks for more complex computations
 - NB: This is not a “uniform cortical sheet” or “cortical column” hypothesis
- Although examples of these primitives have been measured and observed for over 50 years, there is still no comprehensive and definitive description of their structure or function (but good theories abound)
 - Primarily limited by technology: until recently, no tools existed to interrogate $10^2 - 10^4$ neurons simultaneously with single-synapse, single-spike resolution

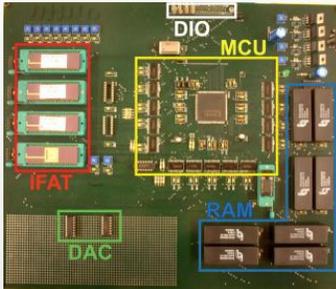


Working Hypothesis

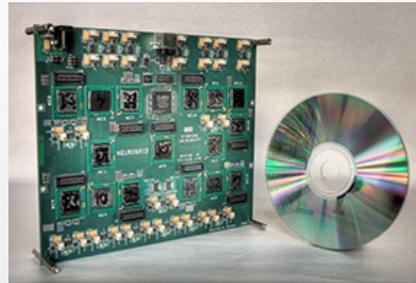
- Extensive information exists about brain function at the macro- and micro-scales; a critical missing piece in our ability to reconstruct neural algorithms *in silico* and build useful biofidelic models of cortical computation (to solve real-world problems) is quantitative information about the function of *mesoscale* cortical computing primitives
- Implications
 - Current approximations are insufficient: existing neuromimetic algorithms do not exhibit brain-like performance
 - Ignoring or abstracting these primitives fails to reproduce critical aspects of cortical algorithms such as data representation, dimensionality reduction, and dynamic evolution / tuning
 - Existing models of cortical primitives are insufficiently constrained by the empirical data available to-date (most of the $10^4 - 10^8$ possible connections remain plausible)
 - If the detailed “schematics” (attributed graphs) of these cortical computing primitives could be identified, interpreted, and instantiated, we could more closely approximate neural algorithms *in silico*
 - Devise new or constrain existing models of cortical primitives (eg predictive coding)
 - Construct better machine learning and pattern recognition algorithms that employ these primitives as their basis of operation
 - **We need innovations in neuromimetic “software”, not hardware (yet)**

“Software” vs. Hardware

- The development of neuromorphic hardware has far outpaced the development of neuromorphic “software” (aka networks) that we intend to deploy on on it



IFAT: $O(10^4)$ neurons
Vogelstein et al., 2002



NeuroGrid: $O(10^6)$ neurons
Boahen et al., 2007

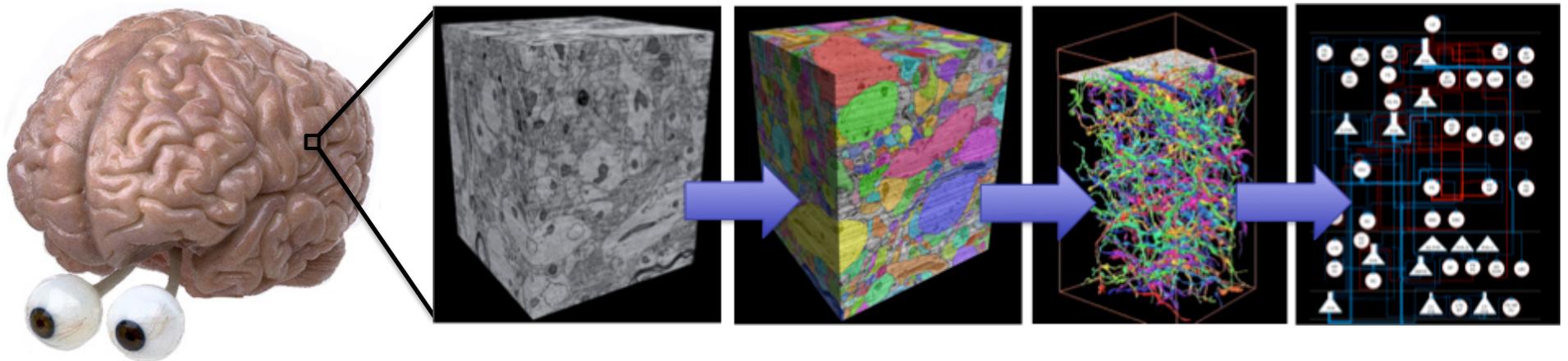


Blue Gene: $O(10^9)$ neurons
IBM et al., 2009

- Even a decade ago, with 9,600 neurons, we had limited knowledge with which to constrain the $9,600^2$ possible connections; with 1,000,000,000 neurons, the problem is much harder
 - Wiring diagram is not all we need to know, but from a practical perspective, it is a minimum requirement for building neuromimetic systems out of individual neurons
 - Question is not many neurons we can emulate, but how much fidelity (e.g. in synaptic weights, polarities; cell topology; parameterization of synapses, axons, dendrites, etc.) is required to reproduce functionality of cortical circuits?
- How do we go from neurons, to primitives, to algorithms, to solving problems?

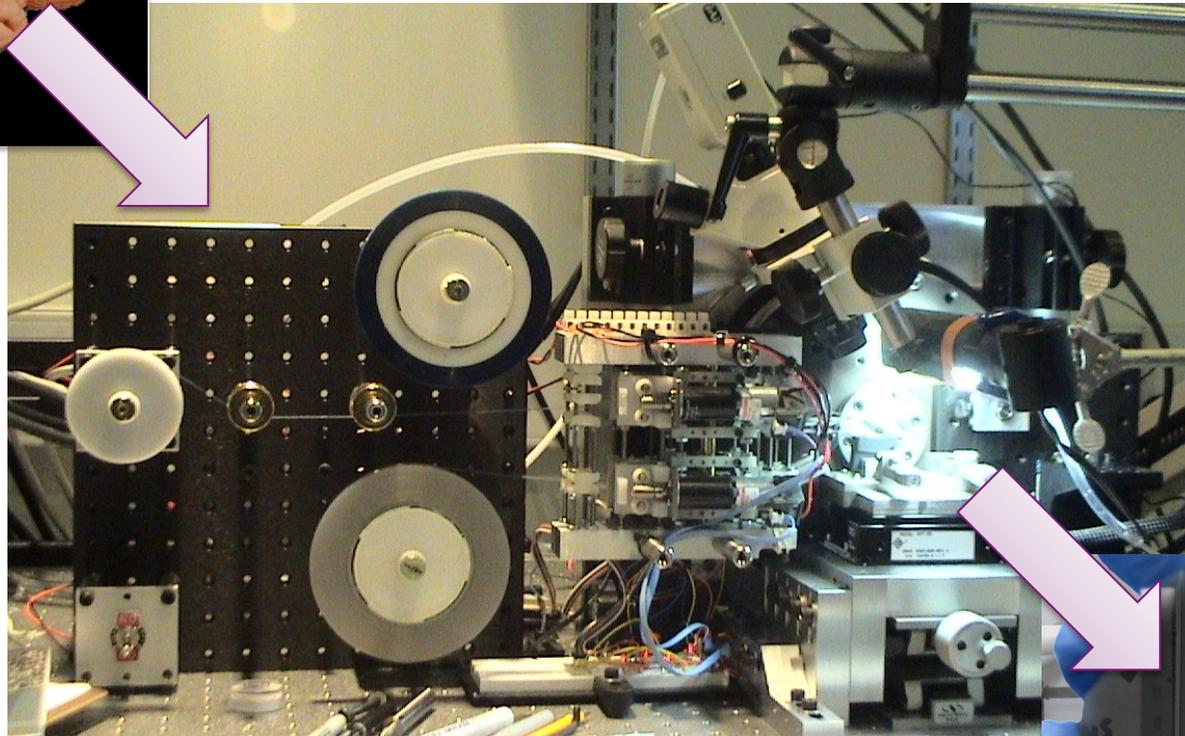
Connectomics: A Way Forwards?

- Connectomics: the science of neural circuits
 - From *connectome*: a comprehensive description of the networks in the brain
 - Relatively new field of research, enabled by new techniques in high-throughput, high-resolution imaging (e.g. serial electron microscopy)
- Provides access to full-detail “as-built” drawings of the human brain, in the form of three-dimensional images of cortical tissue with voxels as small as $5 \times 5 \times 5 \text{ nm}^3$
 - Single-synapse resolution
 - Wide field of view (on the order of cubic millimeters)
- Can be used to “reverse-engineer” (RE) cortical microcircuits, *in combination with knowledge about macro-scale connectivity and neural dynamics*



Connectomics for Reverse Engineering

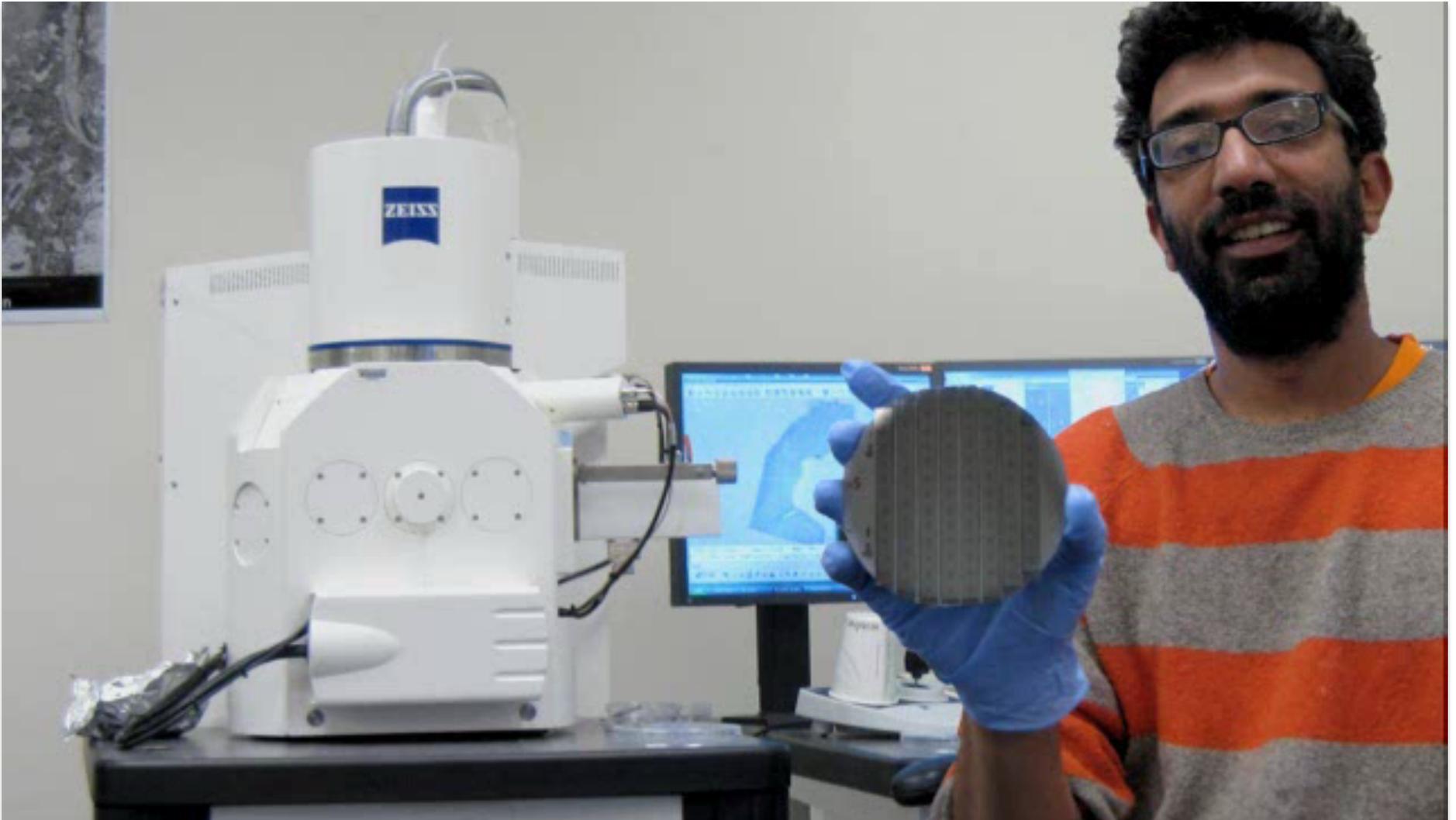
Step 1: Slice Brain



Lichtman et al.

Connectomics for Reverse Engineering

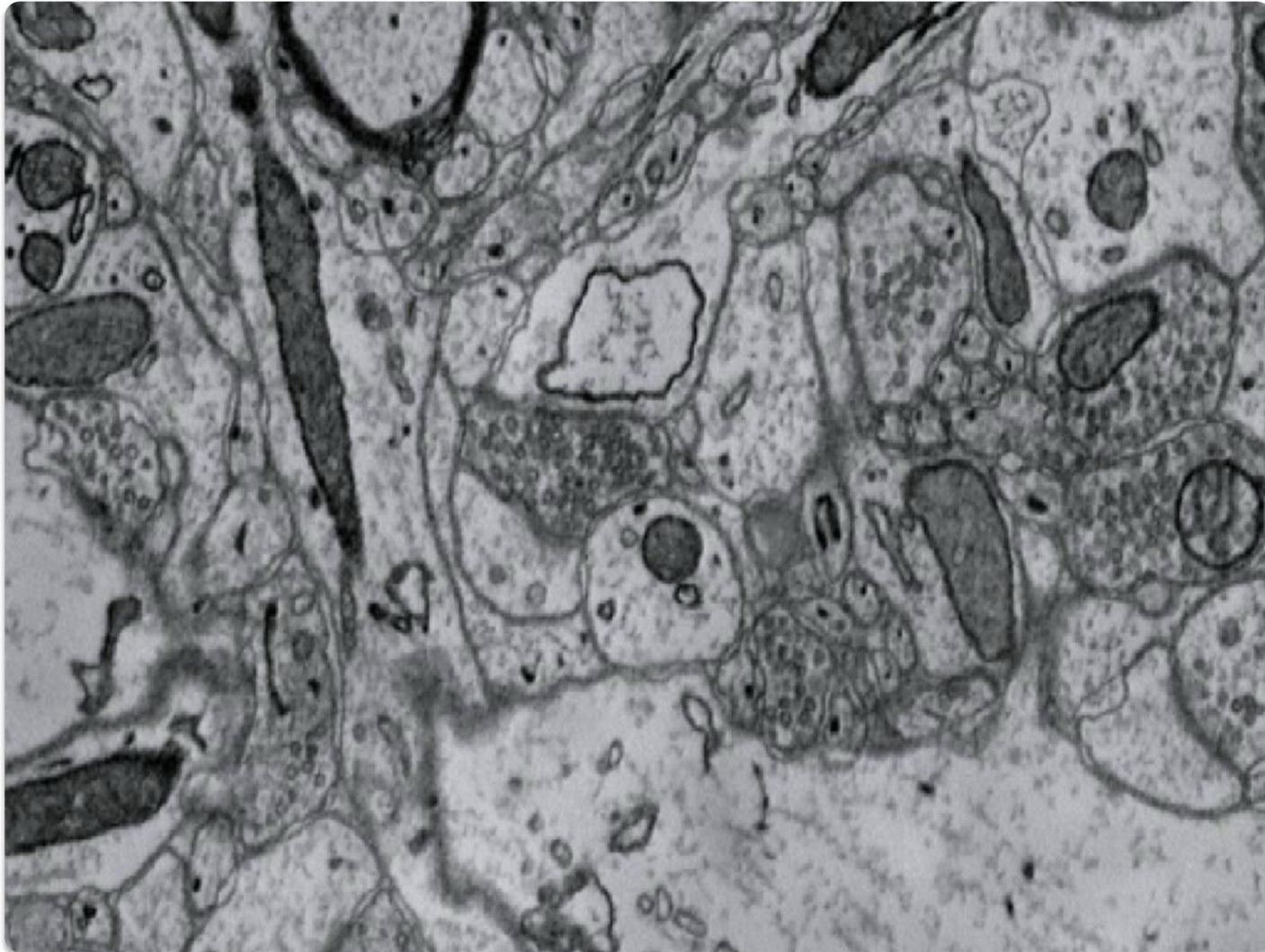
Step 2: Image Stack



Lichtman and Kasthuri

Connectomics for Reverse Engineering

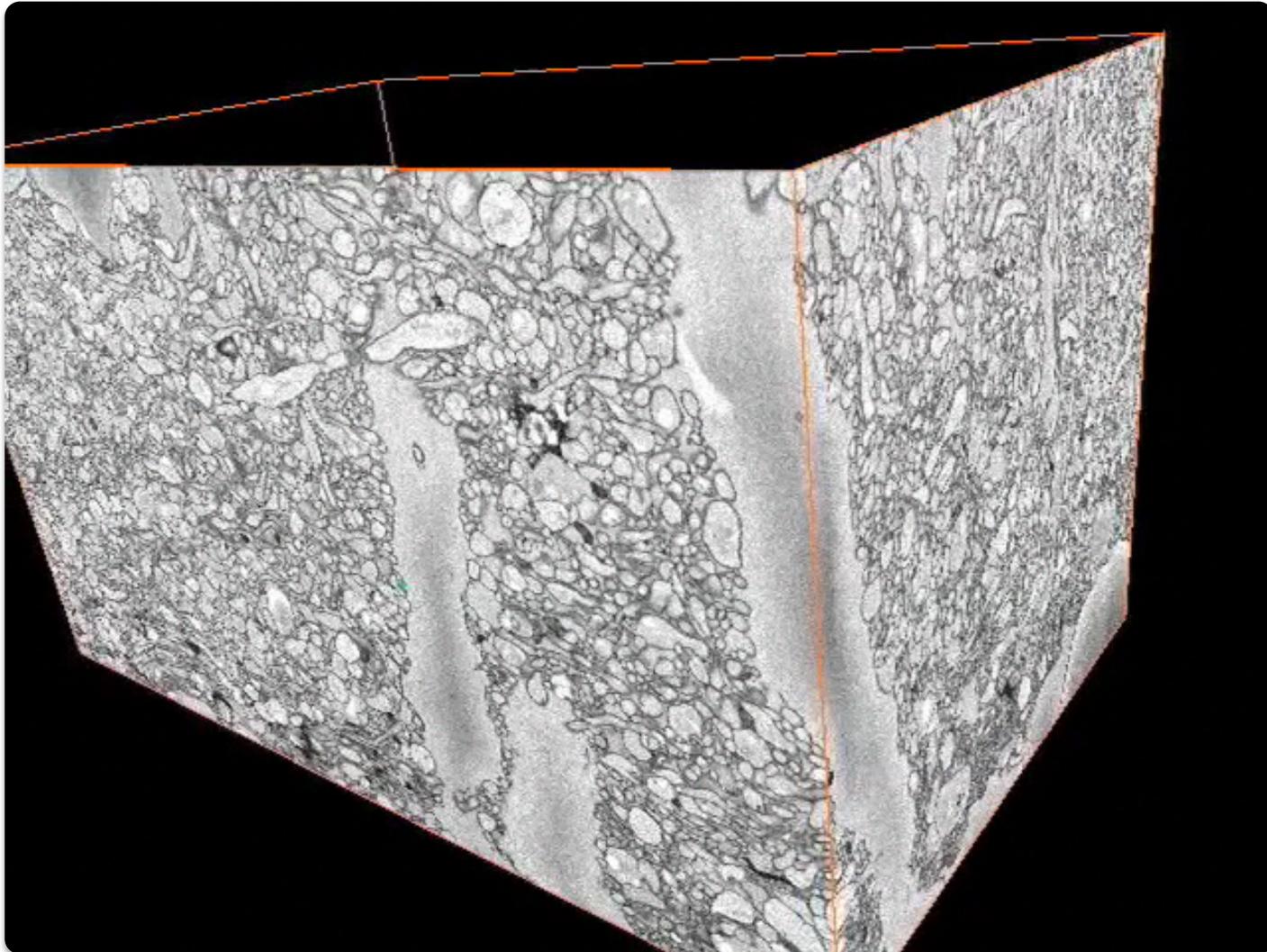
Step 3: Segment Images



Lichtman, Seung, Berger et al.

Connectomics for Reverse Engineering

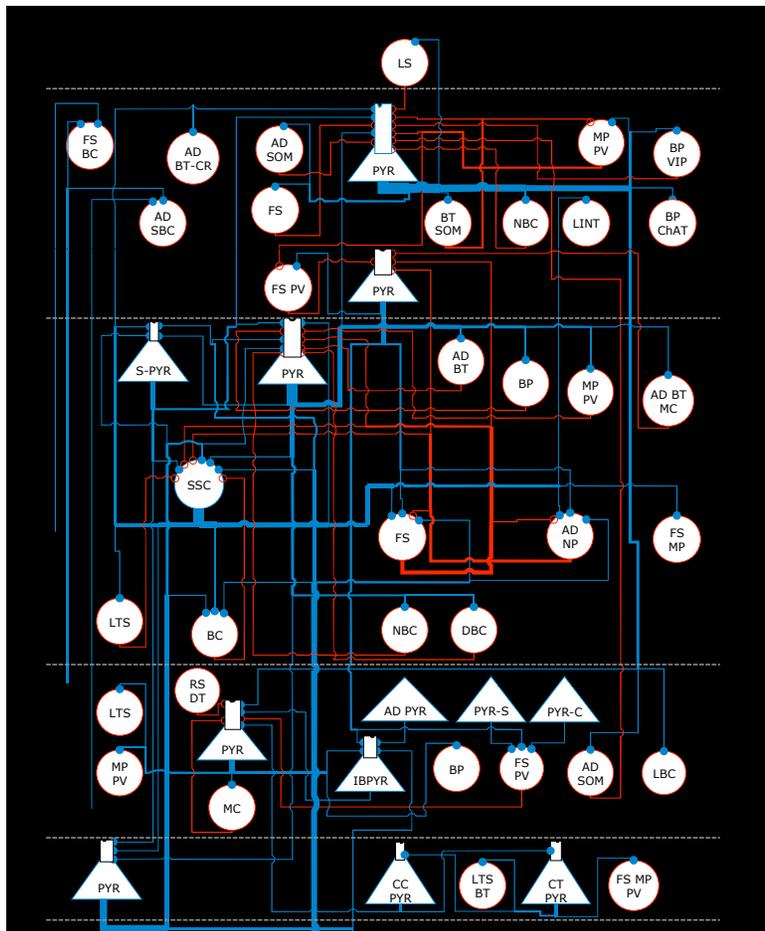
Step 4: Extract Network



Jain, Briggman, Denk et al.

Connectomics for Reverse Engineering

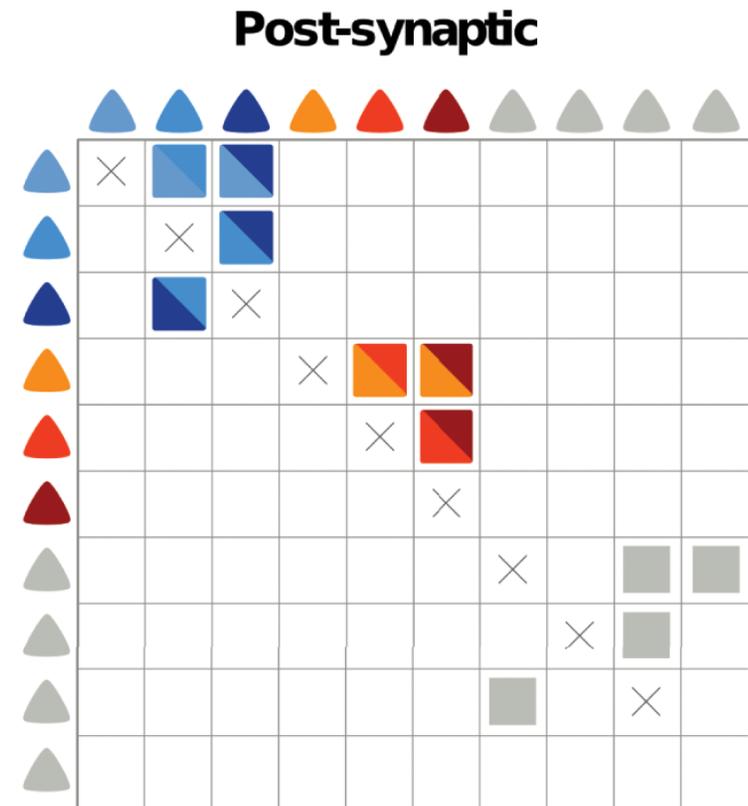
Step 5: Identify Connectivity



Yuste et al.

||

Pre-synaptic



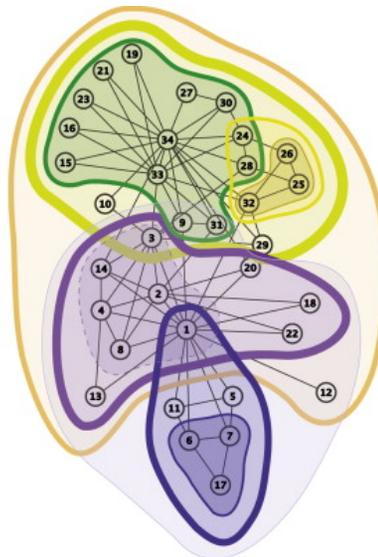
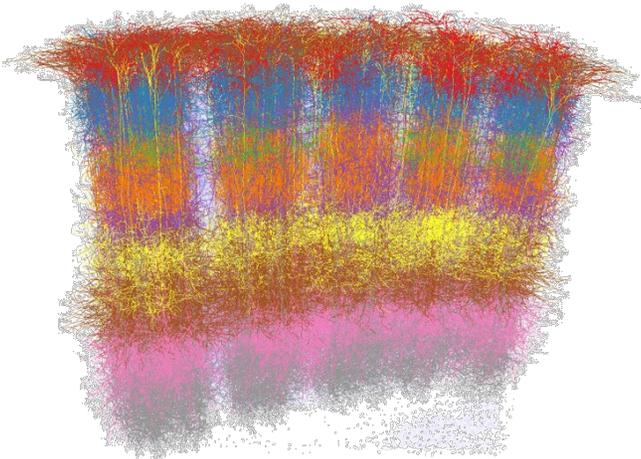
Attributed brain graph

Zador et al., 2012

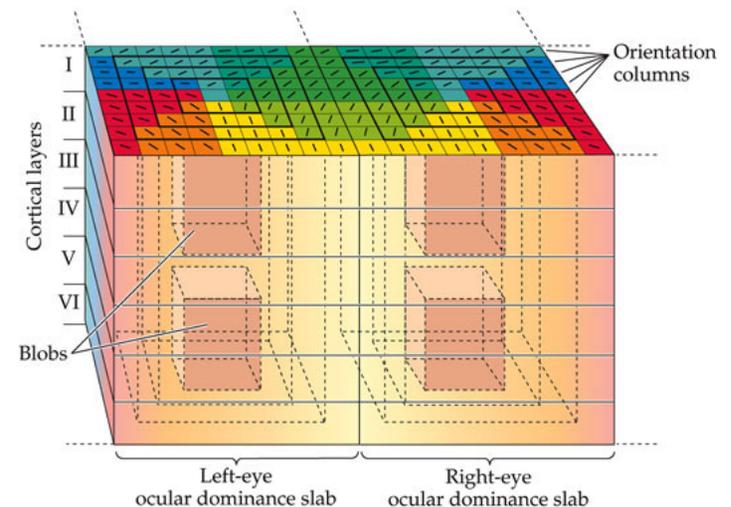
Connectomics for Reverse Engineering

Step 6: Find Conserved Motifs

- Cortex is composed of repeated neural circuits at multiple scales (hundreds to tens of thousands of neurons)
- Cortical computing primitives can be identified from the relative occurrence statistics of identified motifs



Tibely, 2012

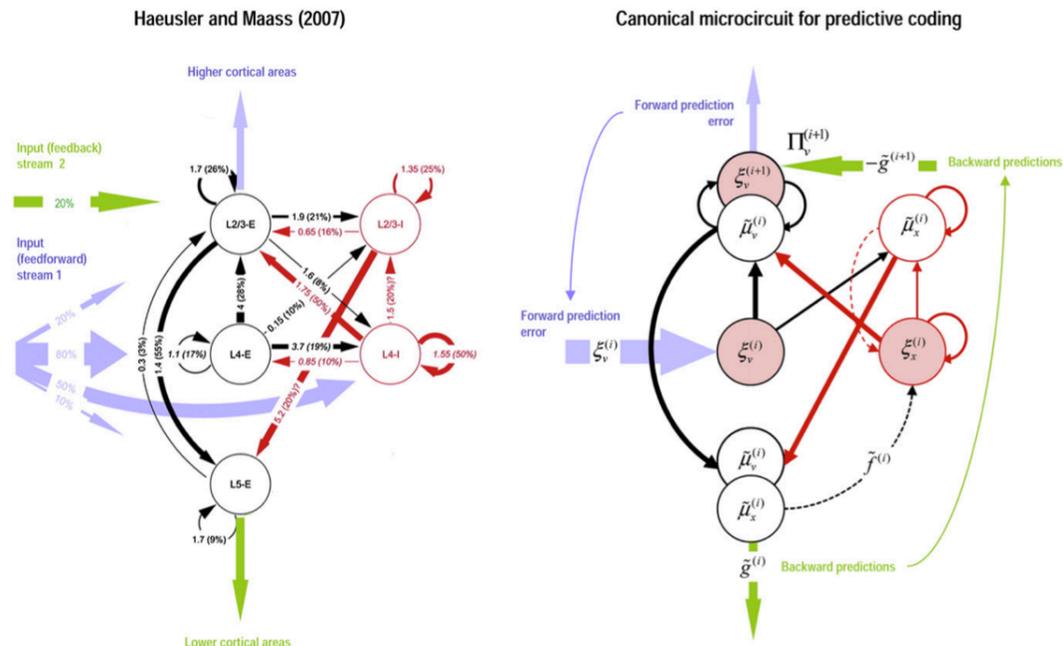


Biological Psychology

Connectomics for Reverse Engineering

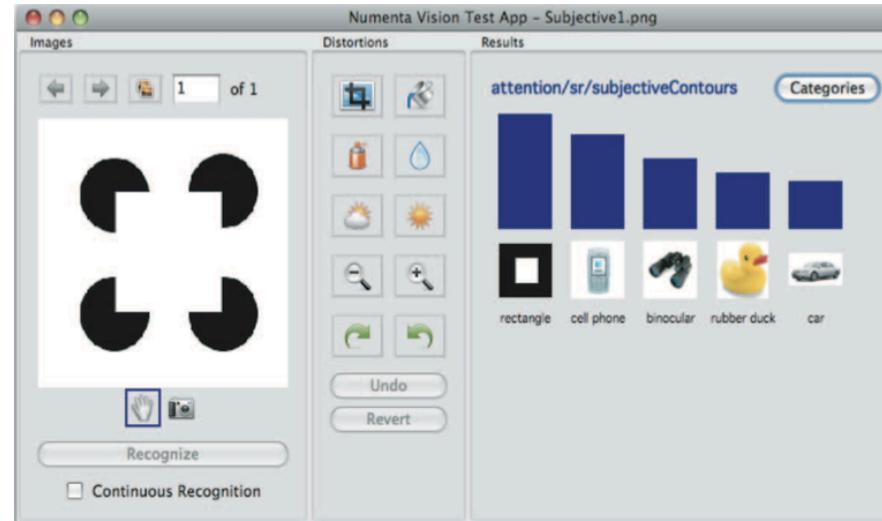
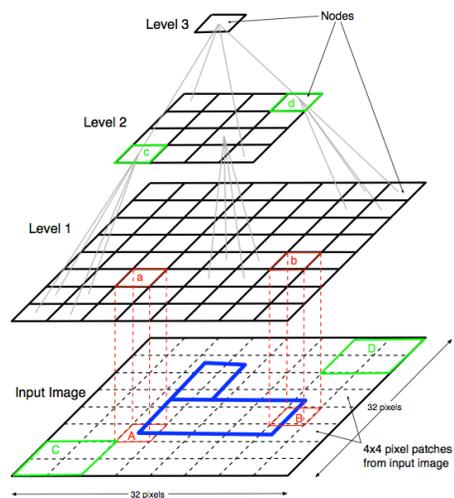
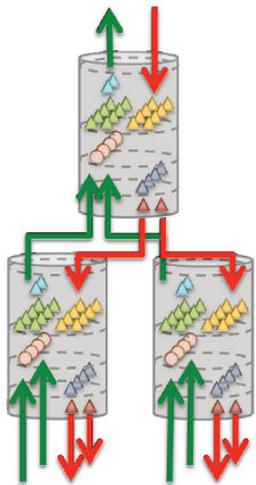
Step 7: Characterize Function(s)

- Detailed microcircuit structure (i.e. attributed graph) is the biggest driver of functionality, but the schematics need to be infused with known neural dynamics, macro-scale connectivity and mechanisms of plasticity to infer function
 - NB: in the brain, the *attributed* graph is both the hardware and the software
- Circuit functions may be derived from the bottom-up, or the schematic may be used to constrain existing models of cortical primitives



Bastos et al., 2012

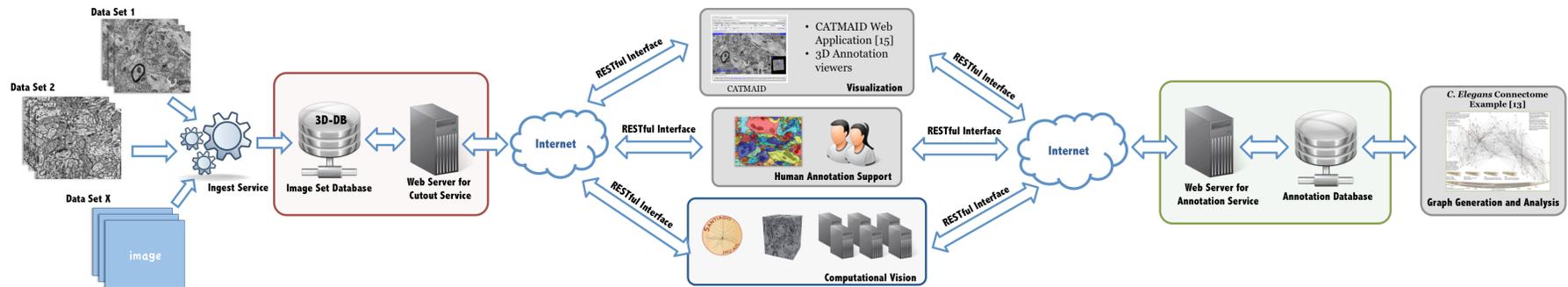
Application (Step 8) Define Architecture; Deploy



Our Contribution

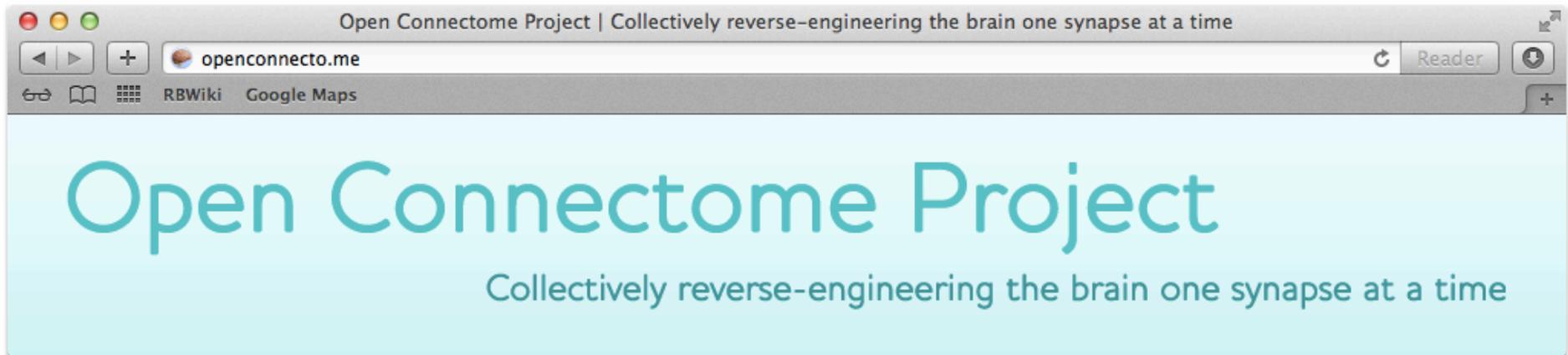
- Immediately evident that no single lab, and probably no single institution, would have the capabilities or resources to perform all components of this work
- Most likely way to get from today's state of the art to the next-generation of neuromimetic algorithms based on cortical computing primitives discovered through connectomics is through collaborative science and engineering
- We have therefore developed the CAJAL3D (Connectome Analysis through Joint Annotation of Large 3D Data) ecosystem to facilitate the large-scale, multi-institution collaboration required to reverse engineer significant quantities of cortical tissue and identify the cortical computing primitives

What is CAJAL3D?



- The CAJAL3D ecosystem is a collection of software tools, data standards, interfaces, and databases designed to facilitate large-scale, collaborative annotation and reverse-engineering of cortical circuits
- Focus on architecture, not algorithms
 - We're trying to create a destination for people to easily implement, evaluate and share the results of their own algorithms for analyzing connectome data
 - Intended to be flexible and inclusive of multiple different approaches that share a common goal of identifying structures within EM connectomics datasets
- Currently focused on sharing image data and data products, as opposed to computing resources for server-side computations

CAJAL3D and the Open Connectome Project (OCP)

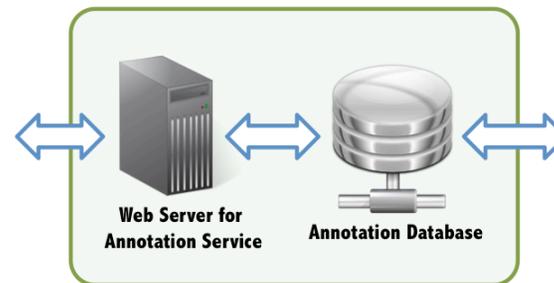
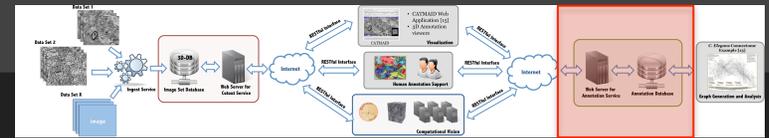


- CAJAL3D is open-source and nominally deployable anywhere
- The Open Connectome Project (OCP) at <http://openconnecto.me> provides community access and a centralized repository for connectome data (e.g. images) and data products (e.g. annotations, graphs)
- CAJAL3D ecosystem is currently integrated with and hosted by OCP
 - Image and annotation databases
 - Web services
 - Visualization tools

CAJAL3D Technical Objectives

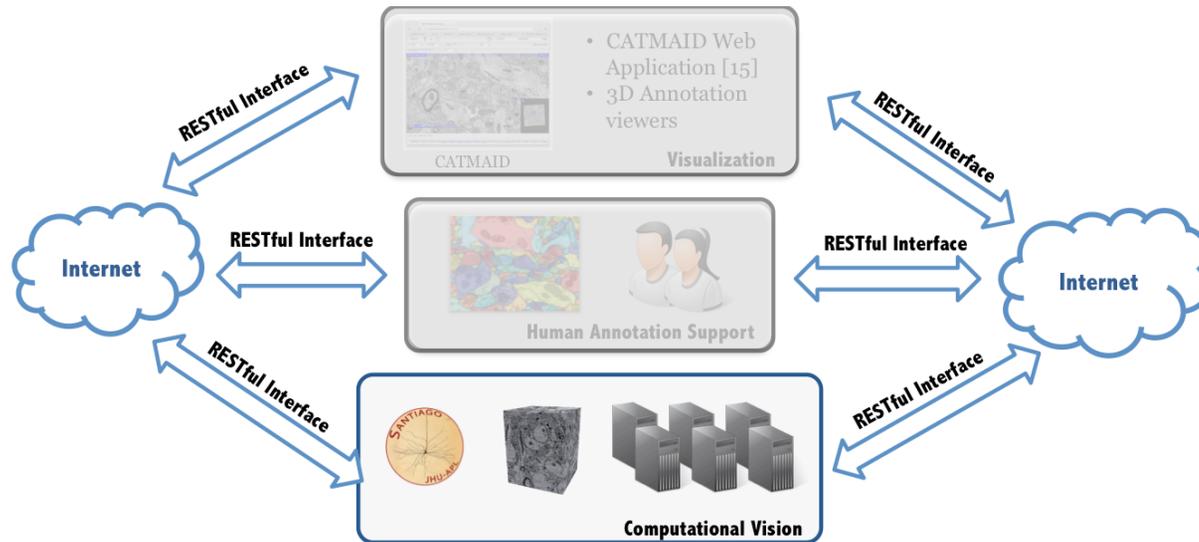
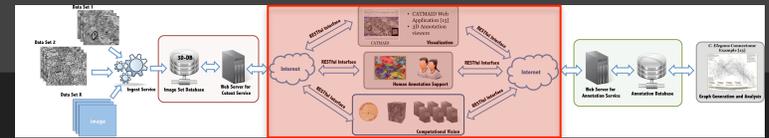
- Universal access to all system components and capabilities via Internet(2)
- High-speed and automated upload and ingest from microscope to database
- Efficient storage and retrieval of very large (petabyte+) image datasets
- Common language (aka data types) for annotation data products and intermediate artifacts to facilitate information exchange between people and/or algorithms
 - Support for manual, semi- and fully-automatic annotation
 - Application Programmer's Interface (API) for key system functions
 - Modular distributed computing pipeline for automated image processing
- Read/write access to shared (and access-controlled) annotation databases
- Fast, query-based retrieval of annotation and/or image data (and associated API)
- Two- and three-dimensional visualization tools
- Support for data-intensive spatial analyses of annotations and derived graphs

CAJAL3D Component: Annotation Database



- **Current capabilities**
 - Standard data types (“RAMON” objects) are defined to provide universal language for sharing annotation data products and intermediate artifacts
 - Manual skeletonized and/or dense 2D or 3D annotations can be ingested and converted into standard data types
 - Annotations may be kept private or shared by distributing unique access key
 - Multiple sets of annotations may be linked to the same source image data
 - Annotation database supports rich queries against standard data types to facilitate distributed processing, data exploration, and data-intensive analysis, such as:
 - Find all annotations of type “synapse” within the volume bounded by $(X,Y,Z)-(X,Y,Z)$
 - Find all annotations labeled as “spine” associated with “Neuron #1234”
 - Return all annotations within the volume bounded by $(X,Y,Z)-(X,Y,Z)$
 - Return the minimal image volume containing “Neuron #5678”
 - Annotations and queries are exposed through RESTful web service

CAJAL3D Component: Automated Annotation Pipeline



- **Current capabilities**
 - We have developed a modular framework (“pipeline”) and reference implementation in LONI for scalable, automated, distributed, large-scale computer annotation of EM images
 - Inputs to and outputs from the pipeline are exchanged over RESTful web service interfaces
 - Input: two and/or three-dimensional image data
 - Output: RAMON objects
 - Information exchange between stages of the pipeline is facilitated by the annotation database, allowing for modularity and cross-institutional collaboration using RAMON standard datatypes
 - Convenient wrapper and helper functions are provided via MATLAB API to abstract away the REST URLs and HTTP responses under standard MATLAB function calls

Summary

- We want to build a new generation of machine learning and pattern recognition systems that solve real-world problems using the strategies employed by the brain
- I believe that the critical missing piece in our understanding and ability to implement biofidelic neural algorithms *in silico* is quantitative information about the function of *meso-scale* cortical computing primitives
 - “Software” (in the form of attributed graphs), not hardware, is the current bottleneck, and has been for at least a decade
 - Employing biofidelic cortical computing primitives is essential to achieving brain-like function in artificial machine learning and pattern recognition systems
- Connectomics provides a method for obtaining this missing information
 - Both structure and function are critical, but structure currently lags function
 - Let’s start a conversation on how and whether to prioritize
- CAJAL3D and OpenConnectome.me provide tools and infrastructure for exploiting connectomics data and extracting neural circuit structures
- We want to collaborate! Please contact me at jacob.vogelstein@jhuapl.edu

THANK YOU!!!

Contributors and Collaborators

- JHU/APL
 - Dean Kleissas
 - Will Gray
 - James Burck
 - Philippe Burlina
- JHU
 - Randal Burns
 - Misha Kazdhan
 - Carey Priebe
 - Daniel Naiman
 - Greg Hager
- Harvard University
 - Jeff Lichtman lab
 - Bobby Kasthuri
 - Hanspeter Pfister lab
- Janelia Farm Research Campus
 - Davi Bock lab
- Allen Institute
 - Clay Reid
- Duke University
 - Joshua Vogelstein

APL

JOHNS HOPKINS UNIVERSITY

Applied Physics Laboratory

Applied Physics Laboratory

JOHNS HOPKINS UNIVERSITY

APL