Contents

I Official Transmittal Letter 1

II Executive Summary 4

III Executive Summary Slide 6

IV Goals and Impact 7
- IV.A Overview . . . 7
- IV.B Justification . . . 7
- IV.C Raw Data . . . 8
- IV.D Use Cases . . . 10
  - IV.D.1 Use Case 1: Mesoscale Functional and Structural Brain Data . . . 10
  - IV.D.2 Use Case 2: Microscale Functional and Structural Brain Data . . . 10
- IV.E Task 1: Mathematical Formalism . . . 11
- IV.F Task 2: Computational Infrastructure . . . 12
- IV.G Task 3: Datafication . . . 14
- IV.H Task 4: Discovery . . . 15

V Technical Plan 16
- V.A Task 1: Mathematical Framework . . . 19
  - V.A.1 RAG Embedding . . . 20
  - V.A.2 FlashRAG . . . 22
  - V.A.3 RAG Testing . . . 23
- V.B Task 2: Computational Infrastructure . . . 24
  - V.B.1 Data Management . . . 24
  - V.B.2 Remote Access and Visualization . . . 25
  - V.B.3 Local Access and Visualization . . . 27
- V.C Task 3: Datafication . . . 28
  - V.C.1 Data Ingest . . . 28
  - V.C.2 Data Registration . . . 29
  - V.C.3 Quality Control . . . 30
- V.D Task 4: Discovery . . . 31
  - V.D.1 Graph Construction . . . 31
  - V.D.2 RAG Summary Statistics . . . 32
  - V.D.3 RAG Predict . . . 33

VI Management Plan 34

VII Personnel, Qualifications, and Commitments 36

VIIICapabilities 38
- VIII.A Open Connectome Project . . . 38
- VIII.B Graph Analytics . . . 39
- VIII.C FlashGraph . . . 39
- VIII.D GRAPHS vs. SIMPLEX . . . 39
- VIII.E Specialized Facilities . . . 39

IX Statement of Work 41
- IX.A Phase I . . . 41
  - IX.A.1 Task 1: Mathematical Formalism . . . 41

IX.A.2 Task 2: Computational Infrastructure . . . 41
IX.A.3 Task 3: Datafication . . . 42
IX.A.4 Task 4: Discovery . . . 42
IX.A.5 Task 5: Program Management . . . 42
IX.B Phase II . . . 42
IX.B.1 Task 1: Mathematical Formalism . . . 42
IX.B.2 Task 2: Computational Infrastructure . . . 42
IX.B.3 Task 3: Datafication . . . 43
IX.B.4 Task 4: Discovery . . . 43
IX.B.5 Task 5: Program Management . . . 43
IX.C Phase III . . . 43
IX.C.1 Task 1: Mathematical Formalism . . . 44
IX.C.2 Task 2: Computational Infrastructure . . . 44
IX.C.3 Task 3: Datafication . . . 44
IX.C.4 Task 4: Discovery . . . 44
IX.C.5 Task 5: Program Management . . . 45

X Schedule and Milestones 45

XI Cost Summary 45

XII Administrative and National Policy Requirements 48

List of Figures

1 Schematic listing the four technical tasks of our proposal. . . . 8
2 Illustration of the raw image data, datafication process, and resulting graphs. The different imaging modalities have different spatiotemporal scales. . . . 9
3 Example microscale structural and functional images. . . . 10
4 A schematic pipeline of our proposed computational infrastructure. . . . 13
5 Schematic illustration of our fast surface rendering tool. . . . 26
6 Schematic illustrating an example cortical graph with repeated motifs. . . . 33
7 Orgchart. . . . 35
8 Datascope. . . . 41
9 Schematic listing all milestones for this proposal. . . . 46

List of Tables

1 Table of disparate experimental paradigms our system will operate on. . . . 9
2 Statistical methods to be used for quality assurance . . . 31

II Executive Summary

TA1

What is the proposed approach and how will it accomplish the stated TA1 milestones and objectives?

The proposed approach is to convert multiple disparate datasets, spanning spatiotemporal scales, modalities, and species, into richly attributed graphs (RAGs). The RAG representation is sufficiently rich to represent a wide variety of forms of knowledge, including simple graphs, attributed graphs, labeled graphs, weighted graphs, and directed graphs. Once all the datasets are stored in a common format, they will also be accessible via a single API, to enable researchers and other interested parties to make novel discoveries, both within and across data types.

What are the key innovative concepts in your technical approach?

The key innovative concept is that we can embed RAGs into a much lower dimensional representative space, by optimizing these embeddings with respect to the downstream inference task. Moreover, we can encode qualitative knowledge in RAG construction and embedding by adding both soft and hard constraints. By implementing our algorithms in software optimized for solid state disks, many of the classical bottlenecks for quantitative analytics are significantly mitigated.

How does this approach surpass the capabilities of existing representations and/or data structures?

Most current graph representations either only store simple binary graphs, or use an excessively heavy-weight representation. By representing complex multi-graphs, for example, by RAGs, we can surpass the data representation capabilities, significantly compressing data storage, enabling much more efficient processing.

To what degree is your representation modular and extensible?

The RAG representation is inherently extensible. For example, if we have stored a RAG representing only two modalities, so that there are two aligned graphs, and a third modality is obtained, it may straightforwardly be added to the RAG. Moreover, attribute types can be arbitrarily extended.

To what degree is this approach practical, scalable and domain-agnostic, and what are its limitations?

Insofar as this approach is an extension and unification of previous approaches, it is practical. Our implementations are scalable, indeed, we have already process billion vertex 100 billion edge graphs with it. Moreover, as long as the domain of interest has multiple entities, each can be thought of as a node with attributes, then the RAG representation is applicable. RAGs cannot easily represent arbitrary simplicial complices, such as faces, which require arbitrary third-order relationships to be stored.

How much will it cost, and how long will it take?

Our total requested budget for Task 1 and Task 2, which focus on TA1, is $1,199,371. Task 5, which is program management, is half devoted to TA1, and half devoted to TA2. Therefore, the TA1 portion of Task 5 will cost $78,688.50. Together, this totals $1,278,059.50, for a 39 month period.

TA2

What is the proposed domain and use case and why is it appropriate for SIMPLEX? What are the technical challenges associated with this use case, and how are they being addressed today?

The proposed domain is massive neuroscience image stacks, including static and dynamic data, at multiple scales, from multiple measurement modalities. These data are the most exciting and complex datasets available in the world today; yet, novel scientific discoveries fusing these disparate data modalities has remained a challenge. The main technical challenge limiting progress on these datasets is scale. For our mouse data, each mouse yields approximately 1 terabyte (TB) of image data. For our human data, each dataset yields only 50 gigabytes (GB), but we have thousands of subjects, totaling 50 TB. Today, scientists are typically extensively downsampling or subsampling.

What data and domain knowledge will be used, and how will it serve your use case? Is this data and domain knowledge readily available?

The PI on this program has a PhD in neuroscience, and the remaining co-I's and key personnel have been working in this domain together already for many years. This knowledge leads us to determine the specific goals of our two use cases, and will be utilized throughout. For example, the datafication process we process is only possible with extensive domain knowledge. Thus, while this domain knowledge is not readily available without years of study, the data products that we will build will encode our knowledge, so that users of our services will not need to.

What datafication approach will you be using and why is it an appropriate choice?

The datafication approach we will take includes deep processing pipelines converting tera-voxel images into tera-edge richly attributed graphs. This builds on our previous images to graphs pipelines, unifying previously disparate methods, and extending them to support a larger number of data modalities.

What discovery tools do you intend to build and why are they appropriate?

The discovery tools that we will build including constructing RAGs utilizing qualitative and quantitative knowledge, providing summary statistics such as moments, motifs, and modes, and predictive capabilities, all optimized for RAGs. These are appropriate as they all enable discovery and hypothesis generation and prediction. Moreover, they are foundational statistical tools readily applied to answer questions for essentially any domain of interest.

What capabilities will you assume in a TA1 Knowledge Representation for your particular use case?

The requisite TA1 knowledge representation capabilities will include low-dimensional embedding of graphs. To the extent that a TA1 knowledge representation enables scalable inference, and more expressive knowledge representation (such as attributed graphs), our use cases can be further explored.

How much will it cost, and how long will it take?

Our total requested budget for Task 3 and Task 4, which focus on TA2, is $619,115. Task 5, which is program management, is half devoted to TA1, and half devoted to TA2. Therefore, the TA2 portion of Task 5 will cost $78,688.50. Together, this totals $697,803.50, for a 39 month period.

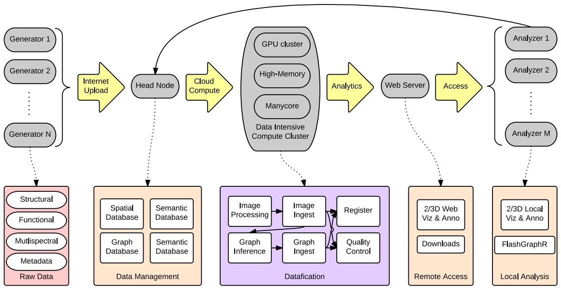
III Executive Summary Slide

From RAGs to Riches: Utilizing Richly Attributed Graphs to Reason from Heterogeneous Data

PI: JT Vogelstein, JHU

Executive Summary Slide: DARPA-BAA-14-59

RESEARCH OVERVIEW



Multiple, large, multifarious brain imaging datasets are rapidly becoming standards in neuroscience. Yet, we lack the tools to analyze individual datasets, much less populations thereof. Therefore, we will develop theory and methods to analyze and otherwise make such data available. Discovery from these complex datasets can answer age old questions in neuroscience, such as whether repeated motifs existence, normal and abnormal connectivity profiles, and psychiatric taxonomies.

TECHNICAL APPROACH

Key ideas

(1) Richly attributed graphs can represent a great variety of kinds of information,

(2) Low-dimensional embeddings of such representations will yield representations amenable to discovery.

Comparison to alternative approaches

Existing approaches focus on a simple graphs, or lack the theoretical and practical ability to fuse across these domains at scale.

QUANTITATIVE BENEFITS

The problems that we address are fundamentally NP-hard, or worse. Therefore, no system can guarantee optimal performance in suitable time. We will provide theoretical guarantees of our performance given the computational trade-offs that we employ to facilitate achieving approximate answers. Moreover, our analytics will run on commodity machines, easing the implementation of these tools for myriad further applications at a fraction of the cost on high-performance clusters.

IV Goals and Impact

IV.A Overview

There are over 40,000 neuroscientists and neurologists, 7 billion people, and trillions of mammals, each with a brain composed of $> 10^6$ neurons and $> 10^9$ connections between them. The answers to age old questions, ranging from what makes us human, to how do we generate music or learn a new skill, lie within them. While we have recently been developing technologies that enable us to collect data at sufficient spatiotemporal resolution to start obtaining answers, we lack the tools to adequately make use of such data. We propose to build a knowledge representation system that would enable anybody—ranging from expert scientists and clinicians, to curious citizens—to access, annotate, and analyze the images and derived graph-valued data on their own. To do so, we focus on four tasks; for each task, we define one sub-task per phase of the program (see Figure 1).

First, **we will build foundational theory and methods for statistical and computational pattern recognition in populations of richly attributed graphs (RAGs).** As we will show, point clouds, directed graphs, weighted graphs, multi-graphs, time-varying graphs, and populations of graphs can all be represented as RAGs. We will develop theory and methods for estimation and testing of RAGs, with a special emphasis on joint embedding designed specifically for downstream inference tasks. Our implementations of these functions will scale to billion vertex and trillion edge RAGs. Moreover, our embeddings will utilize prior and qualitative knowledge, encoding them as soft and hard constraints.

Second, we will **build a computational infrastructure to support (i) conversion of multiple, multifarious, multi-scale tera-voxel image datasets into registered databases of tera-edge RAGs, (ii) visualization of the raw data and RAGs with various overlays to gain understanding, (iii) analysis of these massive graphs to generate and test novel scientific theories.** By developing both open source libraries and Web-services, we will significantly lower the barrier to entry, enabling anybody with Web-access the ability to visualize or analyze these data, either remotely on cloud services, or locally on their machines.

Third, the data of interest always arrives as tera-voxel images, whereas the desired representation is richly attributed graphs. Therefore, we will build and apply pipelines to **datafy both microscale and mesoscale tera-voxel brain imaging data and associated meta-data into richly attributed brain-graphs.** This task will result in (i) open source code for implementing our ingestation process, (ii) a Web-service for data ingestion of many heterogeneous data types, and (iii) data derived products, including registered images and RAGs. Heterogeneous data will be registered, and quality control scripts will be in place, both of which will incorporate qualitative and domain-specific knowledge.

Finally, we will develop **discovery techniques to construct RAGs, provide summary statistics, and make predictions.** We will also apply these methods to both microscale and mesoscale data to ensure validity and utility. All of these inferences will be available for download in a variety of formats via our Web-services for further analysis and juxtaposition, and all code will be open source on github.

IV.B Justification

In myriad scientific disciplines today, we can collect an unprecedented amount of heterogeneous data. Indeed, the scientific bottleneck has moved from data collection, to data computation. We will develop a system to push the bottleneck further along to data interpretation. To do so, we will

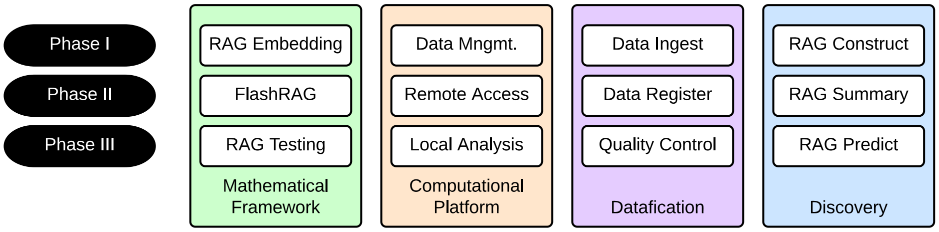


Figure 1: Schematic listing the four technical tasks of our proposal. To achieve the program's goals, for each task, we have one sub-task per phase.

develop a number of tools, both computational and mathematical, both theoretical and practical. All of our tool development will be driven by a particular domain—brain sciences—both because we are experts in the field, and because it is one of the best examples of a scientific discipline in need of such development. On the flip side, the data types that we deal with, data representations that we construct, theory we develop, and methods we implement are almost all entirely domain agnostic (with the exception of our datafication procedures; see below for details). Thus, we fully expect to utilize these deliverables with several performer partners.

Our overarching goal for this proposal is to develop a comprehensive framework to enable scientists, clinicians, and curious individuals to discover novel scientific principles governing the brain and its relationship to our experiences. It is from this vantage point that we decide on the chief operating specifications for our system. Indeed, with finite resources, it is infeasible to optimize *everything*; thus, instead, we consider a few exemplar exploitation tasks and work backgrounds to design a system to meet its needs. By breaking down these tasks into core access patterns and mathematical abstractions, we can then build tools that not only enable us and others to address the particular exploitation tasks we have in mind, but also a great variety of additional questions that we have not even conceived us. We conjecture that by virtue of creating an infrastructure sufficiently flexible and nimble to support vastly disparate bodies of knowledge, we can be a platform for otherwise inconceivable juxtapositions and bouts of creativity.

Prior to providing a detailed set of goals, we first lay out the kinds of data upon which our system will operate, the mathematical abstraction we will utilize to investigate the data, and the two use cases we will investigate.

IV.C Raw Data

Raw data never comes in a format designed for ease of analysis; they data we will operate on in this proposal is no exception to this rule. We will observe n entities, $\widetilde{\mathcal{D}}_n = \{\widetilde{\boldsymbol{\xi}}_1, \ldots, \widetilde{\boldsymbol{\xi}}_n\}$, which collectively comprise our data corpus. Each entity can be observed from m different modalities, so $\widetilde{\boldsymbol{\xi}}_i = (\widetilde{\xi}_i^{(1)}, \ldots, \widetilde{\xi}_i^{(m)}) \in (\widetilde{\Xi}_1, \ldots, \widetilde{\Xi}_m) \triangleq \widetilde{\boldsymbol{\Xi}}$. These modalities may span different orders of magnitude of spatiotemporal scales, may be photons and electrons, may be text and speech. For example, $(\widetilde{\Xi}_1, \widetilde{\Xi}_2, \widetilde{\Xi}_3)$ might be the space of structural, functional, and diffusion magnetic resonance imaging (MRI) data, $(\widetilde{\Xi}_4, \widetilde{\Xi}_5, \widetilde{\Xi}_6)$ could be the space of scanner sequence information for each scan type, respectively, $\widetilde{\Xi}_7$ could be the space of demographic information for the subject, $\widetilde{\Xi}_8$ could be the space of whole genomes, and $\widetilde{\Xi}_9, \ldots, \widetilde{\Xi}_{m-1}$ could be the space of different psychological evaluations, and $\widetilde{\Xi}_m$ could be the time and date stamp. As is evident, many of these spaces are not Euclidean, and even those that are, it is not *a priori* obvious how to combine data across modalities. For example, how does one add a subject's structural MRI image, $\widetilde{\xi}_j^{(i)}$, to her functional MRI image, $\widetilde{\xi}_j^{(i')}$? Similarly, how does one compare a functional MRI image of a human brain, with a

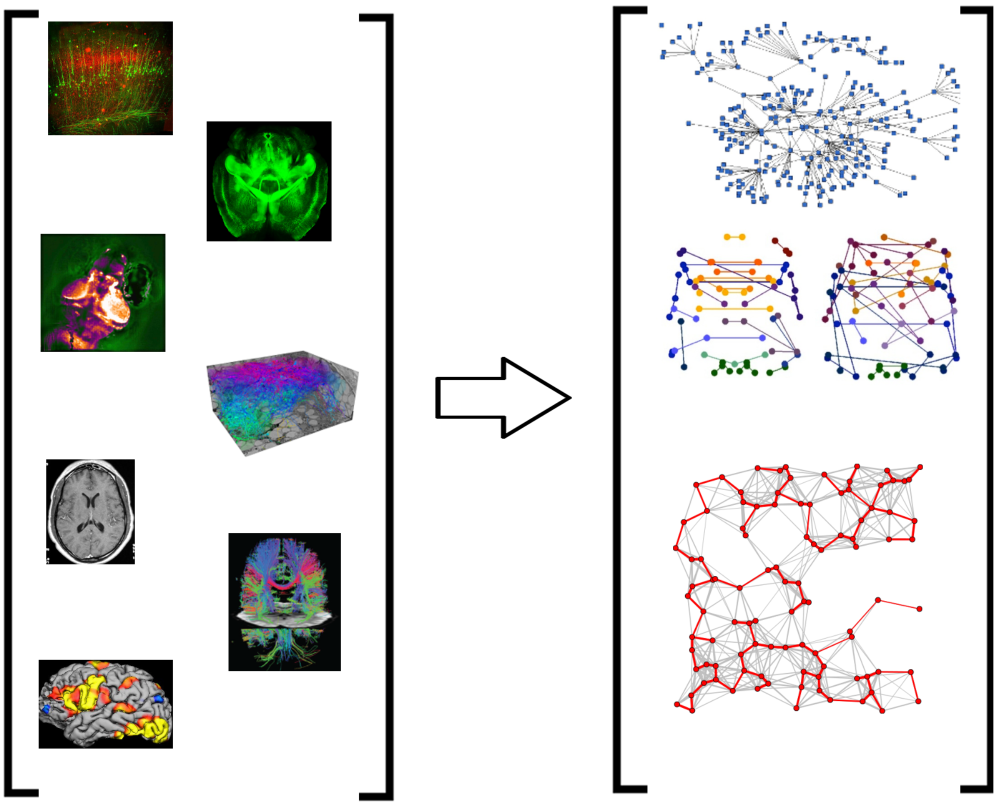


Figure 2: Illustration of the raw image data, datafication process, and resulting graphs. The different imaging modalities have different spatiotemporal scales.

two-photon image of a zebrafish brain? These complications motivate our mathematical abstraction: richly attributed graphs. Figure 2 and Table 1 provide examples of the various experimental paradigms which will generate data that our system will be able to operate on (that is, experimental paradigms for which this project will explicitly build and/or utilize datafication pipelines for; of course, any other imaging modality could also be used in theory).

Table 1: Table of disparate experimental paradigms that generate data upon which our knowledge representation system will operate on. Importantly, although the different paradigms generate data with different representations and file types, upon ingesting into our system, there will be a common API.

	resolution (μm^3)	image rate (Hz)	invasiveness
serial EM	$0.004 \times 0.004 \times 0.04$	0	terminal
Array Tomography	$0.2 \times 0.2 \times 0.05$	0	terminal
2P Calcium	0.2^3	30	chronic
CLARITY	0.2^3	0	terminal
Light Field	2^3	30	chronic
Light Sheet	$0.2 \times 0.2 \times 20$	8	chronic
structural MRI	200^3	0	non-invasive
diffusion MRI	200^3	0	non-invasive
functional MRI	200^3	2	non-invasive

Our datafication pipelines (see §IV.G) will convert these image data, as well as all associated meta-data (for example, time/date stamp, acquisition parameters, experimentalist collecting the data, etc.) into subject-level graphs and RAGs (see §IV.E and Figure 2).

IV.D Use Cases

IV.D.1 Use Case 1: Discovery of Psychiatric Populations and Subpopulations from Multimodal Mesoscale Brain Imaging Data

Internationally, there are two standard and reasonably well accepted taxonomies for mental health: the American Psychiatric Association's DSM-V[1] and the World Health Organization's International Classification of Disease (Ch. 5)[2]. Both of these taxonomies are decided by committee, similar to how the tree of life was decided prior to the genetic revolution. The big data revolution in brain imaging now enables us to construct novel, data driven, algorithmic taxonomies, utilizing brain imaging data, as well as cognitive assays, demographics, and other side information. Our knowledge representation system will enable learning a taxonomy by fusing information across modalities and datasets. To learn these taxonomies, we will use the NKI/Rockland Enhanced Dataset.[3] This is the hallmark dataset for our mesoscale analyses, already consisting of 200 subjects, with 800 more planned. Study participants are sampled from Rockland County across lifespan. The sampling strategy matches participant demographics to US demographics [1].

To achieve our discovery goals for this mesoscopic data, we will first build tools to estimate population means. Then, we will discover motifs that repeat across individuals. Finally, we will use these two tools to build data-driven multi-level clusterings. These steps will employ and extend the RAG embedding, testing, and clustering methodologies, developed in Task 1 and Task 4. This new taxonomy, by virtue of utilizing brain imaging data, unlike previous taxonomies, will potentially reveal etiologies and pathologies of mental disorders, pointing the way to novel treatments.

IV.D.2 Use Case 2: Discovery of Primitives of Neural Processing from Multimodal Microscale Brain Imaging Data

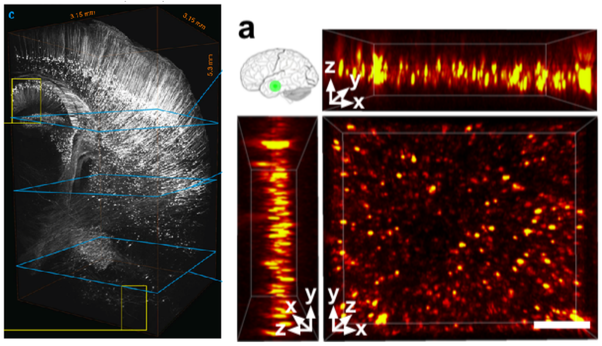


Figure 3: Example microscale structural and functional images. Courtesy of Karl Deisseroth Laboratory.

While neuroscientists have amassed a great deal of knowledge about various scales of information processing, we have not yet discovered many primitives at the circuit level. Until recently, this dearth in knowledge stemmed from a lack of data. Now, however, with both static and dynamic volumetric imaging becoming mainstream, we have the data that will enable answering these century old questions. What we still lack are knowledge presentation systems adequate for answering these questions.

Our framework will enable us to extract detailed connectivity and activity information at the cellular level. More specifically, for structural data, we will utilize data from Karl Deisseroth's recent extension to CLARITY called COLM [2]. From this, we obtain sparse images of each brain; and each brain has GFP tagged to an immediate early gene, so that cell activity leads to expression. This data is not yet published or publicly available. From the same tissue, we will obtain light field microscopy data [3], recorded at >4 Hz, imaging a volume of $\mathcal{O}$ (100) μm per side, with a resolution approaching a cubic micron. From

[1] http://www.dsm5.org/Pages/Default.aspx

[2] http://apps.who.int/classifications/icd10/

[3] http://fcon_1000.projects.nitrc.org/indi/enhanced/

such data, we will search for joint motifs, that is, patterns of connectivity and/or activity that repeat more frequently than one would expect by chance. Because our system represents data across modalities, scales, and species, we will be able to search for motifs within a dataset, and evaluate whether such motifs persist across conditions, scales, species, etc.

IV.E Task 1: Mathematical Formalism

The ultimate power of our knowledge representation system will be our ability to perform inferential analysis across populations of richly attributed graphs (RAGs; see V.A for detailed definition). For example, by storing estimated graphs from a variety of species, we will be able to implement comparative connectomics inference procedures. Similarly, we will be able to test for differences between two different populations of humans, or humans and mice, or two populations of mice. Having all the data co-localized and stored in a consistent format will enable such inferences that are otherwise quite difficult, and therefore, not previously implemented.

Our primary goal for this task is to develop a unified computational and statistical framework for reasoning and inference on populations of RAGs. To achieve this goal, we will develop theory, methods, and scalable implementations for a collection of inference tasks. More specifically, we consider three sub-tasks that collectively span a large set of statistical pattern recognition problems. The deliverables for each of the tasks will include technical reports containing proofs of the validity of the procedures under certain model assumptions, as well as detailed open source code for scalable implementations. We propose to achieve one sub-task per phase on the control, the sub-tasks are organized according to increased complexity and difficulty.

1. *RAG Embedding* will focus on learning low-dimensional representations of the RAGs. Because in our setting the number of ambient dimensions is typically significantly higher than the number of observations, discovery of low-dimensional structure will be crucial for subsequent inference. Moreover, because the space of low-dimensional embeddings is huge, **we will search for embeddings that are approximately optimal for the downstream joint inference tasks, such as testing, clustering, or prediction**. We will encode qualitative knowledge by incorporating both soft and hard constraints.
2. *FlashRAG* will define computational tools for processing the RAG construction by enhancing the FlashGraph system. FlashGraph is a semi-external memory graph-processing engine that builds on the random I/O capabilities of solid-state disks (SSDs) to achieve unprecedented scale and performance [4]. FlashRAG will add the capability to compute on the dual representation of a graph as a sparse matrix of edges. This includes developing novel, semi-external memory formulations of matrix algorithms. We will also encode attribute data in a scalable, set-associative data structure that spans SSDs and memory in order to query attributes in parallel on non-uniform memory, many-core systems. Together, this will enable us to combine matrix and graph algoritms on massive RAGs on commodity machines and evaluate attribute metadata within the inner loops of computations.
3. *RAG Testing* will focus on developing techniques for conducting valid hypothesis tests on RAGs. When data samples live in Euclidean space, there are many standard tests, such as tests for independence, or model fit, etc. **We will extend the classical hypothesis testing formalism to the RAG domain, developing scalable tests for independence, model fit, and more.** Notably, these tests will often first employ the above described estimation and embedding procedures. This enables the existent machinery of classical hypothesis testing to be employed and increases power when the sample size is small and the number of vertices is large.

Our work will generalize and integrate much prior art. For example, there is a extensive literature on graphs and random graphs, from mathematics [5–8], to computer science [9–12]. More recently, the statistics community began to get involved, developing more interesting and complicated random graph models [13–19]. Yet, the vast majority of that work focused on a single graph, typically lacking the attribute structure (such as time, labels, vertex colors, etc.) that we require for our scientific questions.

More recently, we have been developing theory and methods for certain kinds of attributed graphs [20–44] and time-varying graphs [45–54]. Similarly, we have begun to develop theory and methods for populations of graphs [55–63]. We and others, however, lack a comprehensive framework for inference on populations of RAGs, which we will begin to assemble with the proposed work.

The product of our work will take several forms. First, we will build an open source library of graph inference tools, all of which can be run in R, to perform inferences on graphs (see below for more details). These tools will build on two existing libraries which we either created (FlashGraph[4]) or are currently developing (igraph[5]). Second, we will provide a collection of theorems proving approximate accuracy/optimality for our methods. These proofs will be documented in a series of open access technical reports which will mature into peer-reviewed manuscripts in either high-impact statistics journals, computer science conferences, or some equivalent location.

Because we will develop all of our code in an open source environment, we will not seek to directly commericalize any of these products. However, we will be open to consulting for, or otherwise working with companies who desire to utilize our methods. We have a history of companies utilizing our methods in commercial products, and expect this trend to continue. Moreover, because our code will be deposited on github, others will be able to continue utilizing it after the program ends.

IV.F Task 2: Computational Infrastructure

We will build the computational infrastructure to serve as the back-bone of our knowledge representation system. This infrastructure will support the following:

1. **Upload** all the different data types that we support (including images, graphs, annotations, and intermediate data products).
2. **Visualize** images with a wide variety of overlays, including derived graphs, manual annotations, region information, etc. Visualization capabilities will include both two- and three-dimensional (2D and 3D) views, both remotely via Web-services and locally via client services.
3. **Annotate** images and graphs with a rich ontology of pre-defined terms, as well as arbitrary additional information. Annotation will be enabled by both humans and machines, both locally and remotely.
4. **Query** the system to deliver arbitrary subsets of the data. For example, one user may wish to find all the synapses on excitatory cortical neurons across datasets, another may want all the hippocampi of all human subjects over 30.
5. **Download** all raw data and data derived products, including graphs and quality metrics.
6. **Analyze** the resulting graphs using our graph analytics library. This library will support a wide variety of statistically justified algorithms on populations of graphs utilizing our R bindings.

Figure 4 provides a schematic diagram of the computational infrastructure to enable this process; there are three sub-goals in this task, reflecting rather disparate development efforts, as we

[4] https://github.com/icoming/FlashGraph

[5] http://igraph.org/index.html

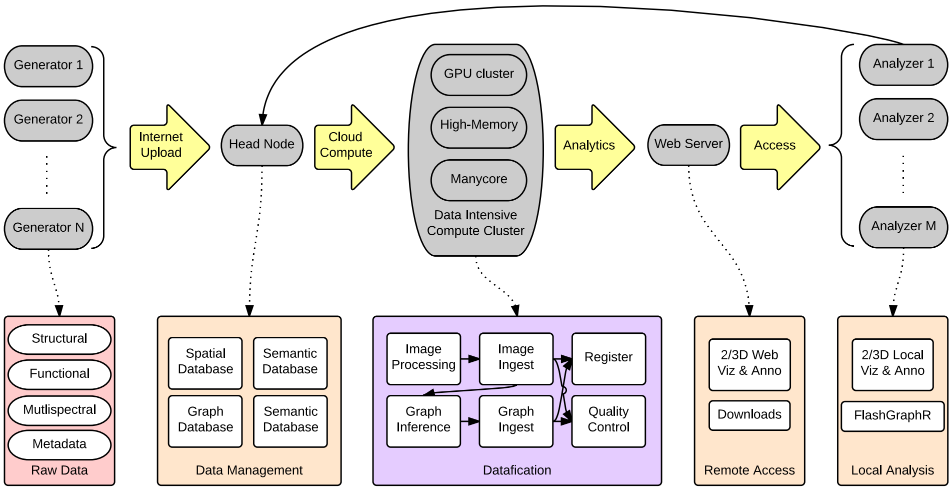


Figure 4: A schematic pipeline of the computational infrastructure supporting our knowledge representation system. Note that different users or partners could enter into this infrastructure at different phases.

describe below. This infrastructure will allow scientific (and clinical) workflows that are currently inaccessible. For example, imagine that a researcher develops a new mouse model of depression, and collects CLARITY data with a certain subtype of cells expressing GFP. This experimentalist could upload her data to our server, which would include his experimentalist providing a host of experimental details, such as age and conditions of mouse, imaging resolution and hardware, etc. Our services would then automatically align this new mouse brain to the Allen Institute for Brain Sciences (AIBS) Mouse Atlas, and provide the user with a quality control Web-page that reports on a number of metrics, enabling this user to immediately assess the quality of her data relative to other similar datasets. Then, this user could overlay the mouse atlas to see all the different brain regions labeled in her new brain. Finally, she could choose several regions, or subregions, to label as “important” in her opinion. A statistician might then go to our website, notice that a new dataset is available, and download just the subvolumes that the experimentalist listed as important, both in the new animal, and in several reference datasets. He could then run a bunch of local analyses of each to compare the different conditions. Once he gets results, he can upload the annotations back to the server so others can use them, while he starts writing his new high-impact publication!

The above workflow is, of course, not currently possible, but is highly desirable. In fact, as we are already working on both use cases, the above described workflow is something that is painstakingly missing from our process for use case 1. The current workflow is that the people who collect the data mail us a hard drive. Then, we *manually* ingest it into our database. It is *never* aligned to any other data, as we lack such alignment tools. When the experimentalist wants to give us more detailed or qualitative information, he sends an email with some screen shots showing him circling areas vaguely in Adobe Photoshop. This whole process so far has taken us three months. Once we have this computational infrastructure in place, most of that process will be automatic, not to mention much improved. More specifically, we target the following three sub-tasks for this task, one per phase, achieving increased flexibility over the course of the program:

1. *Data Management* The datasets of interest demand different computational properties from much extant data. Specifically, we require large memory machines, fast input/output (I/O), data locality, etc. **To meet our quantitative goals for performance, we build a comprehensive hardware and software solution to enable processing with necessary performance characteristics.** These hardware designs will typically not be available in the cloud,

therefore, we will provide details for assembling local clusters, as well as package our software suite onto cloud services, such as Amazon.

2. *Remote Access* Because the data of interest is big, sometimes exceeding 100 terabytes (TB) per dataset, and the resulting graphs are similarly scaled (up to 1 billion vertices and 100 billion edges), many interested individuals will lack the computational resources to navigate the data to make scientific discoveries. Therefore, **we will build a suite of tools to enable remote access to the data, including both 2D and 3D visualization, overlays, and arbitrary annotations, as well as a querying tool to search and juxtapose different data sets.** All this code will be developed open source and will be available as Web-services for any user.
3. *Local Analytics* For some questions, it will benefit the user to download subsets of the data for local visualization and manipulation. To support this, **we will build a suite of tools for users to use locally on their workstations, including GPU enable visualization and annotation tools.** This code will also be developed open source.

Our whole software stack will be developed open source. Moreover, we will deploy cloud services, such as Amazon EC2, mirroring our infrastructure, so that other people can deploy their own versions if they want. Our cloud deployments will also help keep the software stack alive, even after we cycle our hardware or move on to other projects (we won't). We do not plan to commercialize any of our software, though we do suspect we might work with companies that want to utilize some of our technology; indeed, Google is already building infrastructure modeled off of ours to store massive neuroscience image stacks.

IV.G Task 3: Datafication

The goal of the datafication task is to take multiple kinds of multi-modal data, and associated metadata, including annotations, prior confidence levels, etc., and convert them into richly attributed graphs. This will enable domain scientists, clinicians, and "citizen scientists" access to the data that has been appropriately pre-processed to make it amenable for subsequent discovery. This drastically lowers the barrier to entry, as we will remove the necessarily complex and domain specific aspects of datafication from the required expertise to discover new principles from these data.

The datafication process that we propose for this project is well aligned with our previous datafication efforts, as we will describe below. We sub-divide our datafication process into three sub-tasks, one for each phase of the proposal. By proceeding along this three step procedure, we will make the data increasingly accessible:

1. *Data Ingest* We currently have several distinct pipelines [64–66], one for each different data modality, all using the LONI pipelining infrastructure [67]. While these operate on a variety of data modalities, including serial electron microscopy and multi-modal magnetic resonance imaging, they are not unified nor consistent, nor do they span the set of modalities of interest. Therefore, **we will extend and unify our set of pipelines to include CLARITY and light field microscopy (LFM) data.** Moreover, each pipeline will output a RAG per dataset, including spatially downsampled RAGs where appropriate, to enable multi-scale analysis and discovery.
2. *Data Register* Having multiple heterogeneous datasets processed and stored in the same format, enabling the same access patterns, will significantly simplify scientific discovery. However, without registering those datasets to common atlases, their utility is somewhat limiting. Therefore, **for each whole brain dataset, we will register both its image, and its**

RAG, to a standard template, for example, the Allen mouse brain atlas[6] or MNI space for the human data[7]. This will enable queries such as, find all the hippocampi, across species, modalities, and subjects; a query which would otherwise be unavailable.

3. *Quality Control* The data we obtain will be heterogeneous in terms of data quality too. Therefore, **we will also deploy quality control scripts on all datasets**, extending the quality control routines currently available from CPAC [68]. This will enable us and others to select datasets according to quality, in addition to modality, species, and other criteria.

Our approach is innovative in several respects. First of all, most groups develop sub-routines that could potentially be incorporated into a pipeline, but not actually pipelines (for example, [69–71]). If they do generate a pipeline, it is typically for a single modality (for example, [64; 65; 68; 72]). Thus, ours will be the first set of pipelines to operate across such a wide variety of data modalities and scales. Second, of the extant pipelines, they typically output the results in a single scale; our pipelines automatically spatially downsample and therefore provide data products at multiple different scales, enabling a multilevel analysis that is otherwise quite difficult. Third, other groups develop pipelines open source, but still require users to download, install, and run the pipeline. Our pipelines will run a Web-service, so that data collectors can literally upload their data (arranged according to our specifications), and then download the data products, multi-level RAGs, in a format convenient for them.

The result of our datafication process will mean that anybody in the world can upload their raw data, and download RAGs at different scales, regardless of their computational capabilities or resources. The number of people with access to connectomes, for example, will change from the few thousand in the most successful labs, to billions of people. In addition to the RAGs, people will be able to download a variety of intermediate data products, which they can then use to improve our RAG inferences. As always, all the code that we develop will be open source, and available for collaborative coding via our development on github[8].

Because our code will be licensed open source, we will not pursue commercialization, but will be amenable to acting as consultants for companies that commercialize our code. By virtue of depositing all of our code online, with the most open license (Apache 2.0), anybody who chooses will be able to further our work. We will incorporate contributions from the community as appropriate.

IV.H Task 4: Discovery

The goal of this task is to develop methods that explicitly can and are applied to make novel scientific discoveries. More concretely, we will apply the mathematical formalism developed in Task 1, with the computational implementation deployed in Task 2, operating on the RAGs estimated in Task 3, to discover novel properties and principles of brain function and dysfunction in normal and impaired mice and humans. We subdivide our discovery goals into three sub-tasks, one per phase of the program. Each task in increasingly complex, utilizing increasingly sophisticated machinery:

1. *RAG Construction* The data will rarely, if ever, arrive to us as RAGs, therefore, we will often need to convert it to RAGs. **We develop the tools to scalably convert a collection of n points in arbitrary measurable metric spaces into RAGs.** This will require choosing a distance function, utilizing *qualitative* domain knowledge. The result will be that we can represent our data as RAGs and then implement various embedding strategies to find low

[6] http://mouse.brain-map.org/

[7] http://www.bic.mni.mcgill.ca/ServicesAtlases/ICBM152NLin2009

[8] https://github.com/

dimensional representations that capture the variance, even for arbitrarily complex observations and distance functions. We will apply this idea to both the micro- and meso-time-series. For both, we will choose distance metrics based on qualitative knowledge of the system, resulting in RAGs for each.

2. *RAG Summary Statistics* With the foundation of RAG embedding established, we can further these methods to **learn embeddings that are optimized to obtain summary statistics characterizing the RAGs**. Three desiderata can be viewed as three different summary statistics of the high-dimensional non-Euclidean RAGS: moments, motifs, and modes. First, given a collection of RAGs, we desire to know what is the population average RAG. Second, for each RAG, or for the population, we desire to find subgraphs that repeat more frequently than one would expect under a null model. Third, we desire to obtain multi-level clusters of the RAGs, which will enable us to determine multiple moments of the RAGs. This summarization makes more sense than a simple mean when the RAGs are multi-modal. We will estimate summary statistics such as these from both the micro- and meso-RAGs.
3. *RAG Prediction* Beyond summarization and testing, we will often want to make predictions using our RAGs. Therefore, we will **develop tools to make predictions of graph, vertex, and/or edge attributes using graph structure**. This will potentially include classification, regression, and multivariate regression. In each case, we will employ the embedding utilized to determine independence. We will apply these tools to both the mouse micro-RAGs and the human meso-RAGs, to classify (in mouse) and predict personality (in human).

Given satisfactory achievement of the above goals, we will be able to efficiently explore and exploit RAG-valued data in a coherent and principled framework. Currently, no such framework exists. In a certain sense, our work can be considered parallel to functional data analysis (FDA) [73]. FDA focuses on estimation, summarizing, and predicting using functions, rather than Euclidean vectors. RAGs can be considered either special cases of functions, or generalizations of functions. In either case, given these tools, discovery on a wide variety of data sets with different complexities will be enabled, much like FDA enables inference and pattern recognition on typical functions.

Each of the above tasks will utilize and extend the methods developed in the previous tasks. All the code we develop for data analysis will be open source and provided in github repositories. Moreover, all the data derivatives (for example, estimates of population means, motifs, and clusters), will be available for download, analysis, and annotation, using the tools we develop in Task 2. While we will not seek to commercialize any of these products, we will work closely with clinicians and hospitals as appropriate when the technologies reach sufficient maturity. As our methods get more refined, so will our estimates, so the database storing them will continue to be revised.

V Technical Plan

TA1

What are your research and technology objectives?

Our research and technology objectives are to build (1) foundational theory and methods for the analysis of populations of richly attributes graphs (RAGs), and (2) scalable implementations thereof to enable discover on multiple domains and use cases.

What are the key innovative concepts in your technical approach?

There are two key innovative concepts in our technical approach. First, the realization that we can obtain joint embeddings of populations of RAGs via using a a generalization of canonical correlation analysis and/or tensor analysis. Second, that RAG algorithms can operate by storing only the vertex state in main memory, and the edge list and attributes on solid state disks, to enable fast I/O.

What types of knowledge will your system use, and how will it be represented?

Our system will operate on teravoxel images from a wide variety of experimental paradigms and spatiotemporal scales, as well as meta-data of all kinds, including atlases, priors, and ontologies. All information will be stored in RAGs, where various attributes will encode different kinds of information.

What types of data will your representation be able to include? Are there limitations?

In this proposal, our representation will include images, semi-structured text, time-series, point clouds, and more. As long as the number of entities represented as vertices, and their associated attributes, can be stored in main memory of our cluster, and the collection of edges and associated attributes can be stored on disk, our representation can manage the data.

How will your representation link qualitative and quantitative knowledge and data?

Yes. For example, users will be able to assign a confidence score to all aspects of the representation, including the graph, vertex, or edge, of interest. Moreover, our RAGs could represent relative ranks subjectively evaluated, as we will employ non-metric based embedding methodologies.

How does this approach surpass the capabilities of existing representations and/or data structures? What are the quantifiable benefits over other approaches?

Existing representations either handle either Euclidean data, or simple graphs, but typically not richly attributed graphs. Moreover, existing algorithms for obtaining low-dimensional embeddings on large graphs or populations thereof do not scale adequately. For example, our representation enables implementing various graph traversal algorithms on one billion vertex and one hundred billion edge graphs, using commodity parts. There does not exist another system with such capabilities, to our knowledge.

To what degree is this approach practical, modular, scalable and domain-agnostic?

As our approach extends our previous methods, it is demonstrably practical. Its modularity is apparent by virtue of realizing that a RAG can be thought of as a collection of graphs, and therefore, one can easily add additional graphs, as desired. It is scalable because we only represent the vertex state in main memory. And it is domain agnostic because RAGs can represent relationships between any arbitrary collection of objects (assuming they reside in a measurable metric space).

What methods and analysis will need to be developed in order to make your representation useful?

To fully exploit our representation, we will need to develop scalable embedding strategies that operate on RAGs.

How do you plan to interact with TA2 performers?

Any TA2 performer who can represent her information as richly attributed graphs, will be able to utilize our representation and algorithms. We will provide open source code, specifications for storing data in a format amenable to ingesting into our system, and Web-services for converting their data into RAGs and providing downloads in various formats and scales.

TA2

What are your research and technology goals?

We have two research and technology goals. First, to develop and apply tools to convert multiple disparate kinds of tera-voxel neuroscience image stacks into richly attributed graphs. Second, to develop and apply tools for analysis of these tera-edge RAGs.

What is the proposed domain and use case, and why is it an appropriate choice for the SIMPLEX program?

The proposed domain is neuroscience; we consider two use cases. First, a microsale dataset, containing both static whole brain volumes and dynamic subvolumes. For this use case, we have $\mathcal{O}(10)$ such brains, and we seek to understand the similarity and differences across experimental conditions. Second, a mesoscale dataset, containing hundreds or thousands of multi-modal magnetic resonance imaging experiments, as well as phenotypic and other graph attributes. These choices are appropriately because they both exhibit considerable complexity, and multi-modality. Together, the represent a set of especially challenging questions, those across species, rather than merely across individuals within a species.

What are the technical challenges associated with this use case, and how are they being addressed today?

The main technical challenge limiting progress on these datasets is scale. For our mouse data, each mouse yields approximately 1 terabyte (TB) of image data. For our human data, each dataset yields only 50 gigabytes (GB), but we have thousands of subjects, totaling 50 TB. Today, scientists are typically extensively downsampling or subsampling.

What data and domain knowledge will be employed in your effort and how will it serve your use case?

The PI on this program has a PhD in neuroscience, and the remaining co-I's and key personnel have been working in this domain together already for many years. This knowledge leads us to determine the specific goals of our two use cases, and will be utilized throughout. For example, the datafication process we process is only possible with extensive domain knowledge.

Is this data and domain knowledge readily and openly available? Are there any restrictions?

While this domain knowledge is not readily available without years of study, the data products that we will build will encode our knowledge, so that users of our services will not need to.

What datafication approach will you be using and why is it an appropriate choice?

The datafication approach we will take includes deep processing pipelines converting tera-voxel images into tera-edge richly attributed graphs. This builds on our previous images to graphs pipelines, unifying previously disparate methods, and extending them to support a larger number of data modalities.

What discovery tools do you intend to build and why are they appropriate?

The discovery tools that we will build including estimating means, finding repeated patterns, and clustering RAGs. These are appropriate as they all enable discovery and hypothesis generation. Moreover, they are foundational statistical tools readily applied to answer questions for essentially any domain of interest.

How do you intend to accomplish the stated TA2 milestones and objectives?

The formalism and framework that we develop in our TA1 efforts are geared directly at enabling our TA2 efforts, and in particular, to meet our TA2 milestones and objectives. For example, the low-dimensional embedding of RAGs will be utilized in each of the discovery sub-tasks that we pursue.

What capabilities will you assume in a TA1 Knowledge Representation in order to solve your particular use case?

The requisite TA1 knowledge representation capabilities will include low-dimensional embedding of graphs. To the extent that a TA1 knowledge representation enables scalable inference, and more expressive knowledge representation (such as attributed graphs), our use cases can be further explored.

How do you plan to interact with TA1 performers?

First, we will be a TA1 performer, so we will naturally interact with ourselves. Second, our discovery methodologies could be modified to utilize different representations and low-dimensional algorithms.

V.A Task 1: Mathematical Framework

As mentioned previously, our choice of data representation is the richly attributed graph. Formally, let $G = (V, E)$ be a graph; V is a collection of vertices, and E is a collection of edges amongst them, i.e., $E = \{u \sim v | u, v \in V\}$. Let $\mathcal{G} = \{G : V \in \mathcal{V}, E \in \mathcal{E}\}$ be the set of all graphs under consideration, where $\mathcal{V}$ and $\mathcal{E}$ represent the set of possible vertices and edges, respectively. Already, this representation is quite rich, as vertices can represent any entity (for example, a voxel, region, subject, etc.) and edges can represent any kind of communication or relationship between two vertices (for example, a structural connection, high correlation, physical proximity, etc.).

A simple graph, $G \in \mathcal{G}_s = (\mathcal{V}, \mathcal{E}_s)$ is a graph that lacks self-loops (so for all $E \in \mathcal{E}_s$, we have $u \sim u \notin E$) is undirected (so $u \sim v$ implies $v \sim u$), and has only binary edges (so $E(u, v) \in \{0, 1\}$). Although a simple graph is a rich abstract representational form, for the purposes of the motivating scientific discoveries awaiting, it is not sufficiently general. We therefore consider a number of generalizations:

Digraph (Directed Graph) allows $u \sim v$ and not $v \sim u$.

Weighted Graph allows real-valued edge weights $E(u, v) \in \mathbb{R}$.

Time-varying Graph allows edges to be attributed with a time-stamp, $G = (V, E, \gamma)$, where $\gamma \in \mathbb{R}_+$ is a set of time stamps, one per edge (so that $|E| = |\gamma|$).

Labeled Graph assigns a name or label to each vertex, $G = (V, E, \beta)$, where β is a set of labels, one per vertex, so that $|\beta| = |V|$. Slightly abusing notation, we also will use β to denote the bijective naming function $V \mapsto \beta$.

Multi-Graph allows edges to be of different kinds, so $\gamma \in \{\widetilde{\gamma}_1, \widetilde{\gamma}_2, \ldots\}$, where $\widetilde{\gamma}_i$ is a categorical label, such as chemical or electrical.

To unify the above generalizations (and more), we introduce the notion of Richly Attributed Graphs (RAGs). A RAG is characterized by $RAG = (V, E, \alpha, \beta, \gamma)$, where $\alpha \in \mathcal{A}$ is a global graph attribute, $\beta \in \mathcal{B}$ is a set of vertex attributes, and $\gamma \in \mathcal{C}$ is a set of edge attributes. These are *rich* in the sense that $\mathcal{A}, \mathcal{B}$, and $\mathcal{C}$ can be spaces of essentially arbitrary complexity. For example, α might include the subject identity, time and date stamp of data acquisition, all the data provenance, etc. Similarly,

β might include three-dimensional (3D) voxel position, or the entire 3D morphology of a process, etc., and γ could include the synaptic physiology, all dynamical measurements, etc.

Note that a large fraction of RAGs can be represented by K-dimensional arrays. Consider, for a moment, a simple graph. While graphs live in graph space—not Euclidean space—they can be computationally represented and mathematically manipulated as 2-dimensional arrays. Note that while it is convenient to think of 2-dimensional arrays as matrices, graphs are not matrices (for example, what is $G_1 + G_2$?). Nonetheless, it is often convenient to represent a graph by an adjacency matrix. To do so, however, we require not just the vertex and edge lists, but also an ordering of the vertices, so that we know which vertex is the first row/column pair of the matrix, which is second, etc. Therefore, an adjacency matrix, $A \colon \mathcal{G} \times \mathcal{B} \to \mathbb{R}^{n \times n}$, where $n = |V|$, and $\mathcal{B} = \{\beta : [n] \times [n]\}$ is a matrix representation of a labeled graph for which $A_{uv} = 1$ if and only if $u \sim v$. If we stack multiple adjacency matrices next to one another, we get a 3-dimensional array (sometimes called a tensor). We can use this tensor representation for any of the above generalizations. For example, in a time varying graph, each matrix could represent a single time step, for a multi-graph, each matrix can represent a single edge type.

In addition, sometimes the graphs will have more sophisticated vertex labels, such as three-dimensional (3D) position, morphology, or dynamics. In such a case, we can define a dissimilarity function, $\Delta \colon \beta \times \beta \to \mathbb{R}_+$, which yields a dissimilarity matrix representation of the associated vertex property. Intuitively, $\Delta(\beta(u), \beta(v))$ is small is u and v are similar (according to β) vertices and large if $\beta(u)$ and $\beta(v)$ are significantly different. We can then convert this dense dissimilarity matrix Δ into a sparse dissimilarity matrix (see §V.A.1), thereby obtaining a convenient graph representation for the associated vertex feature. In all our work, we will therefore observe m RAGs (which we will often simply call graphs), $\mathcal{D}_m^{(j)} = \{G_1^{(j)}, \ldots, G_m^{(j)}\}$ (j indexing the data modality), on which we will pursue scientific discoveries via statistical inference.

Because we are interested in using populations of RAGs for statistical pattern recognition and scientific discovery, it is important for us to include a notion of noise or error. Therefore, we construct RAG-valued random variables, $\boldsymbol{G} := \boldsymbol{RAG} \colon \Omega \to \mathcal{RAG} := \mathcal{G}$, which is governed by a RAG-valued distribution, $\boldsymbol{G} \sim F_G$, and takes realizations $\boldsymbol{G}(\omega) = G \in \mathcal{G}$. For this task, we consider three sub-tasks, of increasing complexity. Each sub-task comes with its own technical challenge, approach to overcome it, potential pitfalls, and milestones, as described below.

V.A.1 RAG Embedding

Tensor Factorization Any RAG can be stored as a collection of simpler graphs. When they are aligned, then can be stored as a 3-way tensor. In such scenarios, we can use tensor factorization techniques to recover a lower dimensional representation of the RAG. In comparison to matrix factorization, tensor factorization is still in its infancy despite the fact that there are plethora of numerical algorithms for performing tensor factorizations. In particular, we lack good theory for model selection (choosing the number of factors in the factorization) in tensor factorization.

Unfortunately, model selection is often left to heuristics. As much of popularity of tensor factorization comes from the fact that each factor of tensor factorization is amenable to scientific interpretation, it is critical to make a best choice. This problem can only be exacerbated when the tensor is large and sparse. We propose handling this issue via cross-validation and adding a penalty term to the minimization problem for the tensor factorization. To accommodate noisy data, it is of theoretical and practical interest to develop a framework for understanding the stochastic risk involved with a wide variety of penalty functions and also to develop a methodology for performing such tasks with large, sparse, noisily observed tensors.

Fast JOFC and Missing Data It will often be the case that some data or information is missing. In that case, rather than constructing a tensor, we can generate an omnibus matrix, concatenating the individual graphs. In this representation, we can implement a "joint optimization of fidelity and commensurability" (JOFC) for missing data. SMACOF is an algorithm designed to solve multidimensional scaling with missing data, which we can show is equivalent to this problem. To implement SMACOF, we first construct an omnibus dissimilarity matrix $D \in \mathbb{R}^{mn \times mn}$ which compactly represents the entirety of the within and across modality relationships provided by the Δ_i's. Towards this end, we set the diagonal blocks of D equal to the Δ_i's and identically set each off-diagonal block of D to a block matrix whose diagonal blocks are zero's and whose off-diagonal blocks are missing. From D, the SMACOF algorithm produces a collection of mn points in $\mathbb{R}^d$ representing the n data objects across all m modalities. The particular form of D produces an embedding which simultaneously seeks to preserve the inter-modality dissimilarities as well as the across modality matchedness of the data objects.

While embedding very large data sets via missing data multidimensional scaling is often computationally expensive, our JOFC implementation is highly scalable and can utilize parallel computational architecture for further algorithmic speed up. Indeed, with a bounded number of JOFC iterates, our implementation has runtime of order $O(mn^3)$ (versus $O((mn)^3)$ for a naive missing data MDS implementation).

Benchmarking We will evaluate the above mentioned joint embedding and out-of-sample embedding methodologies to assess the relative strengths and weaknesses of each, as well as comparing newly developed methods to previously described approaches in the literature. Our evaluation will include theoretical analysis of appropriately simple scenarios, as well as numerical evaluation on more complicated simulations, and finally, applications on our two use cases. Each approach will be evaluated on a number of considerations, including computational scalability and accuracy. These evaluations will be included in manuscripts developing each of the methods, as well as open source code running the evaluations and benchmarking the results.

The main challenge in simultaneously embedding multiple RAGs for subsequent joint inference is scalability. To overcome this challenge, we will develop out-of-sample embedding strategies. Out-of-sample (OOS) embedding is classically a difficult problem. In the case wherein the original multidimensional scaling for JOFC is carried out with respect to the strain error criterion, i.e., the error criterion induced by inner products in Euclidean space, the out-of-sample extension is essentially solved via the previous formulations[74]. This leads to a simple procedure for out-of-sample embedding whose computational cost scales linearly with the number of points in the original embedding. On the other hand, JOFC using the strain error criterion is much less flexible when compared to JOFC using the stress error criterion induced by Euclidean distance, e.g., MDS using strain does not allow for missing entries in the omnibus matrix. Implementations of the out-of-sample extension for MDS with the stress error criterion is currently still lacking. The most common formulation for out-of-sample extension with the stress criterion proceeds via an iterative minimization approach. This approach suffers from multiple local minima in the feasible region and may also require computational costs that scales quadratic with the number of points in the original embedding. An efficient implementation for out-of-sample extension with the stress criterion is therefore of great theoretical and practical interests for JOFC embedding on large graphs. Even without an OOS extension for stress, we can utilize OOS for strain, even when the sample size n is arbitrarily large. **The milestone for completing this task will be demonstration that embedding RAGs is amenable to subsequent inference tasks which meet TA1 goals.**

V.A.2 FlashRAG

RAGs demand new compute capabilities, specifically the ability to seamlessly execute matrix and graph traversal algorithms on a unified data representation and to leverage attribute metadata within graph computations. The distributed data representations of parallel graph engines [75; 76] prevent operating on graphs as sparse matrixes of edge lists. To do so, one needs to export the graph and materialize it in a different data structure to build a matrix and invoke a parallel matrix library. To convert back to graphs, one must reingest the data. We will develop a compute environment that allows one to build workflows that combine matrix computation and graph traversals and perform queries against complex metadata within algorithms. As an example, one might induce a subgraph based on a combination of predicates against edge and vertex metadata, compute strongly connected components on the subgraph (a graph traversal algorithm), and perform spectral clustering on the resulting components (a matrix algorithm). All of this will be done within commodity hardware, on a single data representation, and on graphs that scale well beyond memory capacity. Users will be able to initiate these computations from their desktop analysis environments, the R shell and the iPython notebook.

The building block for FlashRAG is the FlashGraph semi-external memory graph processing framework [77]. FlashGraph allows users to express graph algorithms in a vertex-centric fashion. Each vertex maintains algorithmic state and performs computation on local state in parallel. FlashGraph keeps the algorithmic state in memory and accesses edge lists of vertices from solid-state storage devices (SSD). Only storing algorithmic state in memory increases the scalability of FlashGraph in proportion to the ratio of edges to vertices in a graph. For example, breadth-first search in FlashGraph only uses 22 GB of memory when running on a graph of 3.4 billion vertices and 129 billion edges.

FlashMatrix We will extend FlashGraph to natively support sparse matrix computations on graph edge lists. This capability is currently lacking and critically needed to perform tasks, such as eigen-decompositions and graph clustering. Our approach uses a thin translation layer on FlashGraph's edge list representation to present data to matrix algorithms in compressed-spare rows and compressed-sparse column formats. FlashGraph redundantly maintains both in-edges and out-edges for each vertex, which allow us to choose between row and column format for an individual algorithm. Similarly, output matrixes can be operated on as graphs.

FlashAttributes Rich attributes define RAGs and giving algorithms access to attributes is critical. Our goal is to allow individual vertexes or edges to access any attribute on demand. However, the attributes exceed the graph structure in size by orders of magnitude. Attributes will be stored on SSDs and accessible by key/value lookup on vertex/edge id and attribute name. We adopt two design points to make this efficient. First, we will group data by attribute, using column-store principles [78]. This will localize access to attributes and make attributes more cacheable; algorithms tend to access one or few attributes and grouping makes individual attributes sequential on disk and compact in memory. Second, we will use the set-associative cache that underlies FlashGraph [4] to partition the attribute store so that all memory and SSD accesses are asynchronous and local to processors in a non-uniform memory multicore architecture. Avoiding remote memory accesses will realize the full potential of the hardware, allowing us to randomly access attribute data.

FlashR We have implemented several graph and matrix algorithms in FlashGraph that are able to scale to billion-node graphs and packed these graph algorithms in a C++ library. We will create a R, Matlab, and iPython Notebook binding for this library, so that users can perform massive graph computations in their preferred analysis environment. All library code and bindings will be freely available, released open-source through GitHub.

The main technical challenge associated with this task is providing semi-external memory implementations that compete with in-memory performance. The team has already achieved this goal for graph traversal algorithms in prior work. For matrix algorithms, we have demonstrated this capability for computational building block, such as sparse matrix-dense vector multiplication. In all cases, there is good evidence that careful I/O and caching can bridge the gap between SSD and memory. An potential pitfall is that semi-external memory is not well-defined for sparse matrixes. Graph algorithms have a formal notion in which the vertices are in memory and edges on disk. We are developing an analog for matrixes in which algorithmic state is in memory and matrix data on disk. **The milestone for completing this task will be a combined workflow that performs strongly connected components (graph traversal) and spectral clustering (matrix) on a trillion edge graph.**

V.A.3 RAG Testing

RAG 1-sample test An important step in statistical inference on a richly attributed graph involves discovering whether of not the graph is a result of single population of vertices or a mixture of multiple populations of vertices. For an analog in the classical statistic theory, consider deciding if a sequence of random variables are observed from a single Gaussian distribution or a mixture of multiple Gaussian distributions. We propose to develop an equivalent notion of this task where a these samples are now observed as the vertices of a single RAG. Here, the RAG embedding provides an invaluable tool, allowing for the development and deployment of classical testing methodologies on our graph data.

RAG 2-sample test There is an extensive literature devoted to 2-sample hypothesis testing in classical statistics. For our scientific discovery objectives, classical techniques such as a paired t-test are insufficient because they often obfuscates significant amount of network signals which we believe to be critical to our goal. Extending these results to RAGs is an essential task for our goal. For instance, a connectome for C. elegans can be constructed using either the electrical or chemical synapses, and one can perform our 1-sample test on each modality. On the other hand, it is plausible that both connectomes are manifestation of the same underlying network structure, and our solution to the 2-sample hypothesis testing problem for paired graphs [63] will enable the scientist to draw a conclusion about whether or not there is sufficient evidence to support the hypothesis that the network structures are stochastically equivalent. We will consider two approaches: embedding the graphs and employing more classical testing methodologies, and tensor factorization followed by testing procedures developed for the tensor factors.

RAG Independence The value of RAGs over simple graphs is that RAGs can encode graph, vertex, and edge attributes, whereas graphs do not. Given the richness of their expressive power, it is natural to desire to test whether different classes of attributes are independent of graph structure. We will conduct this test, extending the embedding methodologies to this domain to obtain test statistics that we can use to generate valid p-values. This extends the one-sample and two-sample tests developed in Task 1 of Phase II, as here we are also dealing with attributes, rather than merely graph structure. The procedure we will develop is a permutation test. Pseudocode 1 provides an example of how we will conduct such a test.

The main technical challenge associated with this task is extending classical testing statistical testing methodologies to the RAG setting. To overcome this challenge, we will first embed our RAGs into lower dimensional Euclidean spaces and then adapt classical methodologies to these embedded graphs. If our embedding procedures are not suitably robust to errorfully observed RAGs, we will explore alternate procedures in which we will use robust tensor factors for subsequent hypothesis testing. **The milestone for completing this task will be procedures for**

Algorithm 1 Pseudocode for testing for independence between graph structure and vertex attributes.

```
Require: (i) a richly attributed graph, $G = (V, E, \beta)$, where $|V| = n$ and $\beta \in \mathcal{B}^n$, where $\mathcal{B}$ is some metric measurable space, and (ii) $\delta \colon \mathcal{B} \times \mathcal{B} \to \mathbb{R}_+$.
Ensure: p-value
Compute $\Delta$, the $n$ by $n$ distance matrix, where $\delta_{uv} = \delta(\beta(u), \beta(v))$.
Form the omnibus matrix, $D = [E, W; W, \Delta]$, where $W$ is a matrix as determined by the methods described in Task 1.
Embed $D$ into $d$ dimensional space, yielding $2 \times n$ points in that space.
Compute the average distance of $\|\widetilde{x}_u - \widetilde{y}_u\|$, where $\widetilde{x}_u$ is the latent vertex embedded position of vertex $u$, and $\widetilde{y}_i$ is the attributed vertex embedded position of vertex $u$. Call this $t$.
for $b \in [B]$ do
    Permute the labels of $\beta$, and repeat the above steps
end for
Let the p-value equal the cumulative distribution function of the average distance distributions evaluated at $t$.
```

testing whether a graph is heterogeneous and also whether multiple heterogeneous graphs are statistically different from one another.

V.B Task 2: Computational Infrastructure

V.B.1 Data Management

OCP is deployed on datascope which is a petabyte data cluster built for data-intensive analysis. The architecture consists of a scalable distributed spatial NoSQL database which stores all of the image and annotation data. We use multiple web-servers as a load balancing proxy to access these data nodes. Concurrent workloads are directed towards different data nodes to achieve full optimal I/O and avoid interference. SSD nodes are also deployed to handle high write intensive workloads. The cluster is capable, using simple RESTful services, of easily ingesting data from a variety of formats and provide efficient data cutouts which can used for further analysis. Data management of this data includes building various kinds of databases to enable efficient processing and querying, as described below.

Dense Spatial Multi-way Arrays The image data are stored in multi-dimensional spatial arrays which are partitioned across all dimensions into cuboids. Space filling curves are used to store these cuboids in the database for good performance whenever contiguous regions are accessed. Data is stored at multi-resolution hierarchy to enable ease analyses and visualizations at smaller scales. This database is capable of storing multiple data types which includes electron microscopy, light microscopy, multispectral CLARITY and array tomography data, as well as multi-modal MRI data.

Semi-external memory computation model stores a small portion of data for computation in memory and a large portion of data in external memory. For general graph analysis, we can keep all algorithmic vertex state in memory and edge lists in external memory. For sparse matrix vector multiplication (SpMV), a key routine in scientific computing, we can keep the vector in memory and the sparse matrix in external memory. These settings are able to increase the scalability of data analysis significantly while still having performance comparable to pure in-memory solution.

Sparse Multi-way Arrays Sparse mutli-way arrays are used to store the RAGs that we derive from the image data. FlashGraph stores edge and edge attributes of a graph on an SSD array.

It accesses the SSD array through Set-Associative Filesystem (SAFS), a user-space filesystem optimized for accessing a large SSD array. SSDs are connected to a machine through host bus adapters (HBA) instead of RAID controllers, so each SSD is exposed to the operating system. We deploy Linux filesystem (such as XFS and Ext4) on each SSD and deploy SAFS on top of Linux filesystems. SAFS evenly distributes data across all SSDs and refactors I/O from FlashGraph to maximize the performance of an SSD array.

FlashGraph stores edge lists on SSDs in a compact format to avoid reading unnecessary data from SSDs. It orders edge lists on SSDs by vertex ID. Edges and edge attributes are stored separately, so that graph applications that do not require edge attributes can avoid reading attributes from SSDs. For a directed graph, FlashGraph stores in-edge and out-edge lists separately, based on the observation that many graph applications require only one type of edge. Storing both in-edges and out-edges of a vertex together would cause FlashGraph to read more data from SSDs.

For time-series graphs, FlashGraph stores timestamp of an edge as the edge attribute. To locate the edges within the specified time period efficiently, all edges of a vertex are sorted by their timestamps.

Sparse Cutouts To make semantic/spatial/connectivity queries efficiently, in addition to building databases, we also construct a "cutout" query that can join and export the data at desired scale. Our scalable data cluster provides publicly-accessible Web-services for statistical analysis, computer vision, data mining, machine learning, and search on high resolution datasets. We provide a cutout service to extract spatially contiguous regions of the data, including projections to lower dimensions. We use indexes derived from space-filling curves to partition data which makes cutout queries efficient and (mostly) uniform across lower dimensional projections. Vision algorithms can read blocks of data, run analysis on this data and output descriptions of neural connectivity. We capture these in a relational database of neural object metadata linked to a spatial annotation database that stores the structures. We support queries across images and annotations. Annotation databases are spatially registered to images. Finally, we support spatial queries for individual objects and regions that are used in analysis to extract volumes, find nearest neighbors, and compute distances. This data model allows algorithms to run in multiple phases, viewing, analyzing and assembling structure from the output of previous stages, e.g. fusing previous segmentations into neurons thereby enabling the images to graphs pipeline.

The main technical challenge associated with each of these tasks is interoperability and heterogeneous computational demands. The different databases will need to interact, and even be stored local to another another. Moreover, the computations span external memory, cache, and everything in between. We will therefore mitigate these concerns via buying appropriate hardware to optimize each step as necessary. **The milestone for completion of this task is the existence of more than one example of each of the databases on the same dataset.**

V.B.2 Remote Access and Visualization

2D Web Visualization We will integrate existing visualisation tools like CATMAID into our pipeline to view data stored on the remote data nodes. Using these tools, we will allow users to navigate the data and visualize random two-dimensional image planes in a multi resolution hierarchy. These image planes (xy, xz, or yz) are fetched from the data nodes and are rendered dynamically. To perform tile access at memory speed, we will maintain a tile cache in a memory cluster using memcached. The cache contains the most popular tiles. Tiles that are not in cache are rendered on-demand. To service such a cache miss, we will render the requested tile and return it to the viewer immediately. We will then load neighboring tiles into the memory cache as an asynchronous background process thereby populating the cache with all the nearby data without slowing down the user interface. Subsequent accesses to data from this region access tiles from memcache. In practice, when browsing to a new area of data, the system will make a few cache misses, taking less than a second. Subsequent access to related data, scrolling, panning, or zooming, accesses data that are available immediately in memcache.

Surface Extraction We will develop fast visualization tools to extract surface representations of annotation data from petavoxel images (see Figure 5). We will use a modified version of Marching Cubes, a surface extraction algorithm that we will accelerate by running on a Graphics Processing Unit (GPU). However, the limited memory of any GPU, currently 12GB, limits the size of the model that can be built. To realize scale, we will develop a novel external memory implementation of Marching Cubes that decomposes a large model into parts that can be run independently on the GPU and then assembles, integrates, and corrects the partial models on the host CPU system. The performance goal of the system is to realize GPU acceleration at scale. We will realize near linear scaleup (90% efficiency on a four-times larger model on four GPUs when compared with the single GPU implementation). Additionally, we will use the surface extraction tool to power our 3D web visualization tool.

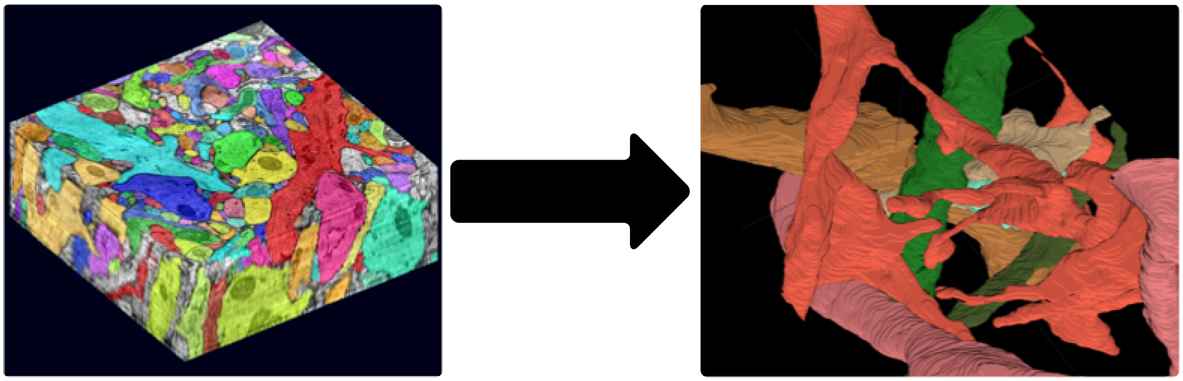


Figure 5: Schematic illustration of our fast surface rendering tool.

3D Web Visualization We will provide interactive 3D visualization of brain structures extracted as surfaces from our Web service. In addition to perspective transformation, such as rotate, pan, and zoom, the user will be able to selectively highlight objects to view the relationship among structures and selectively dim structures to reduce visual complexity. The user will also be able to view object metadata, to include TODO, within the visualization tool. We will utilize the Surface Extraction tool to extract surfaces for the viewer, possibly in real time.

Graph Web Visualization We will support web visualization of tera-edge graphs. We have worked with software packages similar to CATMAID designed for graph visualization, and will adapt one of them to read data from our database. The graph web viewer will be deployed in conjunction with our quality control pages for generated graphs.

The main challenge associated with each of these tasks is efficiency. Ideally, remote access should feel like local access, the target is to achieve video rate when visualizing one megavoxels (a single screen). If the above strategies do not achieve this rate, the bottleneck could be the result of any number of deficiencies. We will benchmark performance all the way up and down the stack, and look for places that can support further parallelism, for example by implementing on GPUs, or faster I/O by deploying an intermediate SSD layer. **The milestones for completion of this task will include an operational software stack implementing all the above functionality, and providing all source code open source.**

V.B.3 Local Access and Visualization

API To facilitate local analysis and the development of computer vision algorithms of sufficient quality to enable exploitable graph inference, we developed an Application Programming Interface (API). The initial version of this tool has been publicly deployed (http://openconnecto.me/api) and is used by the connectomics community to facilely download, analyze, visualize, and upload large scale image and annotation datasets. The API is written in MATLAB and is divided into a collection of classes and enumerations that provide core functionality, and an additional set of functions and tools that perform common tasks. This API is readily extensible to new datatypes and straightforwardly allows users to perform multimodal analysis. We will extend this API to support multi-channel data (to support CLARITY, for example), as well as the SSD array and improved OCP back-end.

Downloads To facilitate the dissemination of human (and other) biologically derived graphs we will expand and further publicize our http://openconnecto.me/graph-services/download portal. This open science resource is currently the largest collection of publicly available human brain-graphs. The collection will grow as we (a) obtain new raw scan data, and (b) estimate and derive graphs from multiple imaging modalities. We will continue to extend the capabilities of this portal. For example, downloads will be available in several common formats in addition to multiple sizing scales. Scaling permits those without the computational/algorithmic resources to conduct analysis on down-sampled graphs. The site will support fast search, filter-by querying and tutorials for quick use cases in various languages such as C, Python and R.

GPU 3D Visualization 3D visualization in large datasets is computationally expensive. A client could not download raw image dataset and visualize it on their laptop. We will develop algorithms to process image data server side. Specifically, we convert our raw image data (voxels) to a surface representation, where multiple points in the same plane are combined and neighboring planes are smoothly connected. We are investigating both preprocessing and storing data as well as processing and passing to the client on demand, to limit storage costs.

Converting image data to surface data is expensive because each individual voxel must be processed. However, voxels in close proximity can be processed together, and most image processing algorithms are data parallel. We use Graphics Processing Units (GPU), which typically have more than 2,000 cores, to preprocess image data in parallel on a single server. However, GPUs have very limited memory compared to a traditional CPU server configuration. For the GPU to process data, it must be loaded into GPU memory (device memory) and then loaded back into main memory (on the server, typically called the "host"). Typically GPU memory is between 6GB and 12GB, which is only enough to process a roughly 1,000 voxel cubed dataset. We are developing external memory algorithms to dynamically load data into and out of GPU memory, ensuring the GPU consistently has data to work on and is not spending time waiting for data from the CPU.

Remote Image Annotation We will develop a system to support manual annotation of image data already stored in our database. While automated methods are becoming more common, manual annotation of EM image data is still widely performed, and is also important for correcting automated methods. Our system will enable end users to seamlessly download annotation data from our database to their local computer, annotate the data in a widely supported software package (e.g. ITK-SNAP), and then re-upload to the database.

The main technical challenge in supporting local analysis and visualization is to enable memory efficient implementations of the desired functionality in simple GUIs that interface with the OCP infrastructure. To that end, we have been working closely with our data contributors and collaborators for the last several years, and we will continue to keep the closely in the loop. As we deploy new features, we will ensure that they meet or exceed the standards required for utility for discovery science. **The milestone for completion of this task is to have an operational complete ecosystem enabling downloading, 3D visualization, and annotation onto remote work-stations, with all the open source code downloadable from github.**

V.C Task 3: Datafication

Recent advances in neuroimaging have enabled our development of connectomics pipelines across modalities ranging from mesoscale (e.g., MRI) to nanoscale (e.g., electron microscopy), which promise revolutionary advancements in healthcare, brain understanding, and biofidelic algorithms. However, the computer vision space is broad and typically optimized at a modular level, rather than considering the overall graph exploitation task. Here we leverage our graph generation pipelines and tools to perform a hyper-parameter search of the best available algorithms, and determine the best operating point to estimate brain networks. We evaluate performance using both existing error metrics (e.g., adjusted rand, variation of information), and also develop novel graph-aware metrics, which can be used to optimize processing.

V.C.1 Data Ingest

Since the data we will operate on is so vastly different, ranging across scales and imaging modalities, we will build pipelines for each that will run on the computational infrastructure that we have developed and are extending in Task 2. Moreover, this effort will extend and fuse our previous datafication efforts. In particular, we have a datafication pipeline operational to convert serial electron microscopy data into richly attributed graphs [66], a pipeline we refer to as “images to graphs (I2G)”. Based on this project, we will build the following datafication pipelines.

CLARITY CLARITY datasets of whole mouse brains typically yield approximately 500 GB of data. Therefore, we will extend our I2G pipeline, which employs data parallelism, and supports multiple channels [79]. We will enable “point cloud extraction” at multiple scales. In other words, rather than merely enabling investigators to download image volumes, we will deploy level-set thresholding to enable extracting point clouds, collections of N triples, (x, y, z), which provide a relative location of the N brightest points in the image. Such a representation enables an immense assortment of analyses, for example, using the Point Cloud Library [80]. From these point clouds, we can efficiently build proximity graphs [59], thereby enabling our graph representation of these datasets. The graph attributes will include all experimental conditions, vertex attributes will include relative 3D coordinates, as well as intensity, edge attributes will include the distance between the pair of vertices. There does not exist any comparable efforts to convert such data to RAGs, although several other groups are building large scale informations pipelines for analysis of high-throughout volumetric microscopy data [72].

Light Field Microscopy (LFM) LFM datasets are time-varying volumes, typically a few hundred microns per side, yielding datasets on the order of 500 GB. In such experiments, in a mouse, we observed up to 10,000 cells. To convert these volumes to RAGs, we first automatically detect all cell bodies, using either our algorithms [69], or more recently developed extensions thereof [70; 71]. Given the set of cell body locations, we will then use our software to estimate the most likely spike trains [81; 82], and then estimate the most likely connectivity pattern, or graph [83]. The graph, vertex, and edge attributes listed above will also be encoded in these RAGs. Moreover, we will also include, for each vertex, a time-varying attribute of its activity over the duration of the experiment, and an edge attribute of the estimated strength of connection (e.g., degree of correlation) between the pair of vertices. We are working in close collaboration with Karl Deisseroth’s laboratory to develop this pipeline, which will be the first of its kind.

Diffusion MRI We will utilize and extend our multi-modal diffusion and structural magnetic resonance imaging (M^3RI) data to graphs pipelines in numerous ways [64; 65]. First, we will upgrade processing modules from JIST [84], which has been problematic when scaling up to large compute clusters, with AFNI [85] and FSL [86]. This will include incorporating state-of-the-art subroutines for estimating orientation distribution functions and probabilistic tractography [87]. Note that a few other groups have more recently developed dMRI processing pipelines, for example, [88–90]. These pipelines all lack the scalability of ours pipeline, for example, PANDA is a collection of MATLAB routines, running MATLAB on a cluster requires too much RAM overhead, and is not open source.

Functional MRI We will utilize and extend our functional MRI pre-processing pipeline [68]. Currently, it uses NiPype [91] to scale to distributed clusters. However, we have found various computational bottlenecks. We will therefore incorporate several of CPAC's key workflows into our LONI infrastructure [67]. Moreover, CPAC's graph extraction functionality is quite limited, using only correlation matrices. Preliminary analyses demonstrate one can dramatically improve reliability by introducing sparsity and other constraints, which we will do.

The main technical challenge associated with each of these pipelines is scalability. Each mouse experiment is large, on the order of 1 TB. By virtue of our scalable spatial database [79], we will mitigate this problem, enabling cutouts of subvolumes of arbitrary size for analysis. Our "block-merging" scripts will enable us to merge the output from each subvolume into a coherent whole. The human data is smaller per subject (only around 50 GB), but has orders of magnitude more subjects (around 1,000). Our pipelines for both will utilize the LONI pipelining environment [67], which we have been extending for several years to support our requisite scalability. Moreover, the facilities that we will utilize have up to a couple thousand cores. Because we will also port our computational platform to the cloud, if we discover that we need more computational power than we have available to us, we will be able to expand. Regardless of the computational resources that we utilize, **the milestone indicating that we have completed this task is to have, for each subject, a multi-level RAG.**

V.C.2 Data Registration

Having all the data stored in a common format enables a consistent access pattern across the disparate datasets. However, without registering the data into a common reference frame, queries such as: "find all frontal cortices", would not be possible. We will therefore register the brains into a common species specific atlas.

Human Multimodal Alignment We will align our structural, functional, and diffusion data to one another. As this is not typically done, the details for how to optimally align images from such disparate modalities remains an open question. We will try several state-of-the-art methods and choose the most reliable one to run on all of our datasets. In our extent pipelines, we already deploy several different algorithms, and preliminary results suggest that ANTS [92] outperforms the other competitors. For all subjects, we will align the data to the MNI152 Atlas[9]. Because we and others have put a number of atlases in alignment with the MNI152 Atlas, including, for example, the Desikan Atlas [93], all human subjects will also be aligned with these atlases. We will enable users to upload their own atlases as well, so that users can then automatically use any atlas they choose to query the data.

Mouse Multimodal Alignment For the mouse brains, the Allen Mouse Brain Atlas[10] is quickly

[9] http://www.bic.mni.mcgill.ca/ServicesAtlases/ICBM152NLin2009

[10] http://mouse.brain-map.org/

becoming the standard brain atlas. To date, there are no alignment tools that operate on the 500 GB brain volumes that we will obtain via CLARITY data. Moreover, the mouse atlas is only available at a much lower resolution. Because our native storage representation of the brain images downsamples several times to obtain the volumes at a variety of scales, we will ensure that we obtain, for each brain, a volume at the same resolution as the mouse brain. Once at the appropriate resolution, we will deploy the set of different alignment tools, including ANTS and Large Deformation Diffeomorphic Metric Mapping (LDDMM) [94]. Already, LDDMM has become a standard for nonlinear mappings, which will be required for the mouse brains, as expression is highly variable across brains. LDDMM has already been demonstrated to be useful in mouse models [95]. Moreover, it is the cornerstone algorithm for collaborative computational neuroanatomy at the millimeter scale [96], having been demonstrated to provide clinically useful results for psychiatric diseases [97]. Therefore, it is a natural choice to extend to these data.

Multi-Graph-Match We will extend our seeded graph matching algorithm [27; 37; 41; 42; 98; 99] to simultaneously match multiple weighted directed graphs of potentially different orders. Suitably modifying the penalty for incorrect matches in our objective function and padding the smaller graph with dummy vertices to make it of commensurate size allows us to match graphs of different orders. To simultaneously match a collection of multiple graphs, we first match each graph to a randomly selected graph from our collection. From this initial alignment, we create a sample mean graph and iteratively alternate between matching each graph to this sample mean and updating the sample mean. This procedure enables the implementation of statistical inference procedures on graphs which require an a priori known vertex correspondence. We will measure the performance of Multimatch on benchmark graph matching problems as well as the effect of the matching on subsequent inference.

The primary technical challenge for this task is also scalability. All existing tools are developed for data orders of magnitude smaller, and the existing atlases are at scales orders of magnitude smaller. Careful analysis of the algorithms will reveal with subroutines can be parallelized, or implemented on a GPU, or are otherwise separable to enable scalability. If these options still do not enable sufficient computational efficiency, we have two other options. First, we can implement scalable linear alignment, indeed we already have a working prototype. Second, we can always downsample to align, and then propagate the alignment up to the higher resolution. Regardless of the path we eventually take, **the milestone indicating that we have completed this task will be having *aligned* all the mice to a common atlas, and all the humans to another common atlas.**

V.C.3 Quality Control

To accurately represent the performance of our pipeline, a variety of statistical measures will be implemented for each of the data types: anatomical MRI, functional MRI, diffusion MRI, CLARITY and LFM. Many of the quality control algorithms implemented are implemented in the preprocessed connectome pipeline Quality Assessment Protocol[11], which operate on anatomical and functional MRI. We will extend these tools to also operate on diffusion MRI, as well as our microscale data (CLARITY and LFM). Table 2 summarizes the statistical methods which will be used to process the MRI data.

The main technical challenge for this task is finding metrics that are meaningful and informative about the downstream inference task. To end, we will utilize test-retest data, examples where the subject was imaged twice, to find maximally reliable metrics.[12] In preliminary work, we

[11] http://preprocessed-connectomes-project.github.io/abide/quality_assessment.html

[12] http://fcon_1000.projects.nitrc.org/indi/CoRR/html/

Measure	MRI	fMRI	dMRI	CLARITY	LFM
Contrast-to-Noise Ratio	✓			✓	✓
Entropy Focus Criterion	✓	✓	✓	✓	✓
Foreground to Background Energy Ratio	✓	✓	✓	✓	✓
Smoothness of Voxels	✓	✓	✓		
Percent of Artifact Voxels	✓				
Signal to Noise Ratio	✓			✓	✓
Standardized DVARS		✓	✓	✓	✓
Fraction of Outlier Voxels		✓	✓	✓	✓
Mean Distance to Median Volume		✓	✓		
Mean Framewise Displacement		✓			
Number/Percent FD greater than 0.2 mm		✓			
Ghost to Signal Ratio		✓	✓		

Table 2: Statistical methods to be used for quality assurance

have mathematically proven the utility of this approach. If however, our preliminary list of metrics does not yield *any* that are reliable, we will coordinate with the data contributors to investigate what information they use subjectively to evaluate, and work together to make such subjective knowledge quantitative. **The milestone we will achieve upon completion of this task is that each dataset will have a Web-page "report", collecting the above metrics, as well as various interactive images to provide examples of the quality of the dataset.**

V.D Task 4: Discovery

V.D.1 Graph Construction

Note that some of the data of interest to us will not be offered as graphs, but rather, as point clouds. A common practice is to define a metric between pairs of points, and then compute the $(N-1)N/2$ pairwise comparisons between all points. The result is a dense dissimilarity matrix. However, graphs encode *similarity*, rather than dissimilarity. So, prior to obtaining a graph (or RAG) representation, we must convert the dense dissimilarity matrix into a sparse similarity matrix.

Random Walk A popular approach for constructing dissimilarity measures given a graph whose edge weights are similarities is via the notion of a random walk on the graph. This approach give rise to dissimilarity measures such as expected commute time, diffusion distances at time t, and forest distances, among others. When t is small, diffusion distance at time t usually corresponds to a sparse dissimilarity measure when the original graph is sparse. In general, however, many of the random walk based dissimilarities are global in nature, i.e., they may yield dense dissimilarity matrices. In light of this, a characterization of when a random walk based dissimilarity measure will be sparse is then of primary interest. We will search for settings in which such approaches are sparse on both the microscale and mesoscale data.

Graph Sparsification Another principled approach to obtaining sparse dissimilarity measure is via sparsification (e.g., graph sparsification of [100]) or thresholding of the constructed dissimilarity measure. We are interested in principled procedures for determining the optimal sparsification procedure and/or thresholding parameter for both the mesoscale and microscale data.

Benchmarking The above two strategies will have different performance characterisitics in different settings. We will conduct a number of simulation experiments to evaluate the relative advantages and disadvantages of each, in terms of accuracy and efficiently for downstream inference tasks. Given such an understanding, we will also apply these methods to both the microscale and mesoscale data.

The primary challenge with both of these approaches is that they lack the theoretical underpinnings to justify current heuristic techniques. We propose theory and methodology for the construction of dissimilarity matrices from similarity matrices that preserves sparsity in the original similarity matrices. Moreover, our approach unifies graph distances as different parameterizations of a larger family, enabling theoretical tractability of the relative trade-offs of different approaches. If we are ineffective in establishing the theoretical underpinnings for robust dissimilarity representations for general RAGs, we will develop our theory in more tractable random graph regimes and empirically explore the robustness of our procedures when employed on real data. The result, is that for each domain, we obtain a sparse graph. The collection of such graphs comprise a RAG, which we can then embed utilizing the above strategies. **The milestone for completing this task will be theoretical and empirical demonstration that RAGs are able to successfully encode quantitative and qualitative knowledge**, and express functional relationship among entities in complex systems, and that these features are captured in our dissimilarity RAG representations. Moreover, we will build graphs from the functional time-series data, both the microscale from mouse and the mesoscale from human.

V.D.2 RAG Summary Statistics

A fundamental aspect of all exploratory data analysis is the construction of summary statistics. In classical statistics, these include moments, repeated patterns, and modes. Similarly, for RAG valued data, we would like to be able to compute the analogs of each.

RAG Moments Estimating the mean of a population is a fundamental statistical desiderata. While estimating means of one- and two-dimensional Euclidean data is straightforward, once we even have three dimensional data, the optimal estimation procedure changes [101]. While this result does not hold in finite sample spaces [102], optimal procedures for estimating the population mean of a graph are not yet available in the literature. *We will evaluate different approaches, utilizing our various embedding methodologies, to obtain optimal estimates of RAG means.* We will estimate the mean RAG for various sub-populations of human RAGs, as well as the mouse RAGs. One of the most useful diagnostic tools is a growth chart, where doctors can look at weight and height distributions as a function of age. If a child is developing atypical, this can be an early warning sign, which can lead to preventative therapies. It is widely believed that psychiatric disorders are neurodevelopmental disorders [103]; thus, being able to estimate RAG moments, so that we can track neural development, will potentially provide pediatricians with their most useful tool.

We will develop a graph analogue of the strong law of large numbers for estimating a population average graph from a collection of sample graphs. Our strong law will be developed in the space of graph embeddings and will build off recently developed theory in which we established a central limit theorem for graph embeddings [36]. When the across graph vertex correspondences are known, we will explore the convergence behaviors of at least three mean estimation procedures: embedding the edge averaged graph, separately embedding the graphs and subsequently averaging the embedded points, and jointly embedding the graphs and subsequently averaging the embedded points. Additionally, we will measure the viability of all three approaches when the vertex correspondence is errorfully observed or imputed via graph matching.

RAG Motifs Finding repeated subgraphs (or “motifs”) is known to be an NP-hard problem [104], even in simple graphs. **We will develop techniques to find such motifs in RAGs**, utilizing a novel hierarchical stochastic block model (see Figure 6), and tensor factorization methods developed in Task 1. We will find motifs from both the human and mouse RAGs. Since Vernon Mountcastle postulated the existence of a cortical column [105], many have searched for the existence of repeated processing motifs [106]. Their discovery would therefore answer a half-century year old question.

RAG Modes Multilevel clustering can yield insight into large populations of high-dimensional and/or non-Euclidean observations. We and others have recently published some preliminary work on clustering simple graphs [40; 52; 54; 61]. **We will extend this work to obtain multi-level clusters on large RAGs**, by utilizing the embedded representation developed in Task 1. We will learn multi-level clusterings of the human RAGs. The National Institutes of Mental Health recently announced that the current psychiatric taxonomy is inadequate, and they are in search of new methods for discovering psychological clads via brain-based methods[13]. By searching for multi-level clusters of RAGs, we will discover potentially informative groupings of subjects, which could lead to preventative and targeted therapeutics.

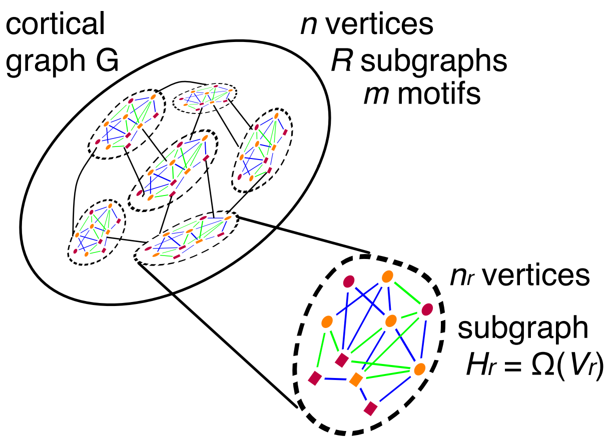


Figure 6: Schematic illustrating an example cortical graph with repeated motifs.

The primary technical challenge associated with estimating all of these summary statistics are scale. Naïvely, each would require an exponential search. Therefore, we utilize approximation strategies that we prove are approximately optimal under certain simplifying assumptions. Notably, we use the embedding methodologies described in Task 1 of Phase I to harness these results. If the embeddings fail to yield fruitful inferences, one can always use them to initialize a constrained, discrete, and possibly brute force technique, which is guaranteed to improve performance. **The milestone for completion of this task will include having operational FlashGraph code to implement these functions operational on tera-edge RAGs**, with R bindings and Web-services to enable easy use in the community. Moreover, we will apply and test these methods on both the micro-RAGs and meso-RAGs, and provide all the resulting data derived products via our Web-services.

V.D.3 RAG Predict

We will develop theory and methods to perform classification, regression, and structured prediction on populations of large graphs. More specifically, we will learn the characteristics of joint graph embeddings that are amenable for adapting classical generalized regression procedures. We will consider both tensor-based and network-of-network (NoN) based methods. Additionally, we will study the consistency of linear and nearest neighbor classifiers on our embedded RAGs, building on the work of [35]. These classifiers will provide insight into taxonomic brain structure within and across subject, and will also help us understand the consistency of our dissimilarity representations across modalities; indeed, the vertices being perfectly classified across modality is strong evidence of the across modality dissimilarities. We will prove optimality in certain settings.

Nearest Neighbor Approach Building on our classification work [44; 60; 107], we will extend this to work on RAGs, rather than merely graphs. Note that the collection of methods will include para-

[13] http://www.nimh.nih.gov/research-priorities/rdoc/index.shtml

metric, semi-parametric, and non-parametric methods. For example, our non-parametric approach will be to define a metric on RAGs. This metric will be informed by qualitative domain knowledge. Given this metric, we will find the distance to all RAGs. Now, using a nearest neighbor strategy, we readily have a classifier, multivariate regression, or structured prediction algorithm. More formally, given collections, $\{(x_i, y_i)\}_{i \in [n]}$, and new observation x, we find the k nearest neighbors of x under our RAG metric (all the x's are RAGs). Let $\mathcal{I}$ be the set of indices of the k nearest neighbors. Now, let $\hat{y}$ be the average of $\{y_i\}_{i \in \mathcal{I}}$. Thus, when $y \in \mathbb{R}^p$, computing this average is trivial. when y is a graph, or another RAG, we can utilize a metric on y to find the mean in that space.

Network of Networks Approach The above strategy, while computationally relatively efficient when n (the number of RAGs) is large, does not make use of all available data. Instead, it only computes the distance from x to all x_i's. If we compute the distance between all pairs of x_i as well, we obtain a network of networks, or a RAG-of-RAGs. Given this, we can embed it utilizing the above developed embedding strategies to obtain low-dimensional embeddings of each RAG. Finally, once we have a low-dimensional embedding representation, we can apply any classification or regression technique. For structured prediction, we can apply a k-nearest neighbor approach, but now measuring Euclidean distance in the low-dimensional space.

Tensor Factorization Approach Given the collection of RAGs, we can also employ a tensor factorization to obtain low-dimensional embeddings, rather than the RAG-of-RAGs approach. The advantage to this approach over the other is that we need not compute all pairwise distances. The disadvantage, however, is that we do not get to incorporate as much qualitative knowledge to embed the representation. We can change the norm, and add penalties and constraints, to the tensor embedding to recover some of that qualitative information. Adding these constraints, however, will typically increase the computational complexity of the algorithms. Thus, there will be an optimal accuracy/efficiency trade-off that we will find. Once embedded using that strategy, we proceed as above, using standard classification, regression, and structured prediction methods applied to the embedded representations.

Benchmarking The above three strategies will have different performance characterisitics in different settings. We will conduct a number of simulation experiments to evaluate the relative advantages and disadvantages of each, in terms of accuracy and efficiently for downstream inference tasks. Given such an understanding, we will also apply these methods to both the microscale and mesoscale data.

The main technical challenge associated with these task is adapting classical clustering, regression and classification procedures to RAG data. As before, the key to our approach is first embedding the graphs and then adapting the classical approaches to these embeddings. If this strategy proves ineffective, we will modify the embedding procedures as necessary, potentially changing the norm that we minimize, or the constraints, or the RAG construction itself. All these modifications will utilize qualitative knowledge, where ever possible. The milestone for completing this task will be having an operational prediction method on RAGs implemented in FlashGraph, with R bindings. Moreover, we will apply these methods to both the microscale mouse RAGs and the mesoscale human RAGs, and provide the open source code as well as the data derived products via our Web-services.

VI Management Plan

PI Joshua Vogelstein has assembled a vertically intergrated team of individuals with complementary expertise and a long successfully history of working well together. Importantly, PI Vogelstein, co-I Priebe, and key personnel, although in different departments with different backgrounds, all

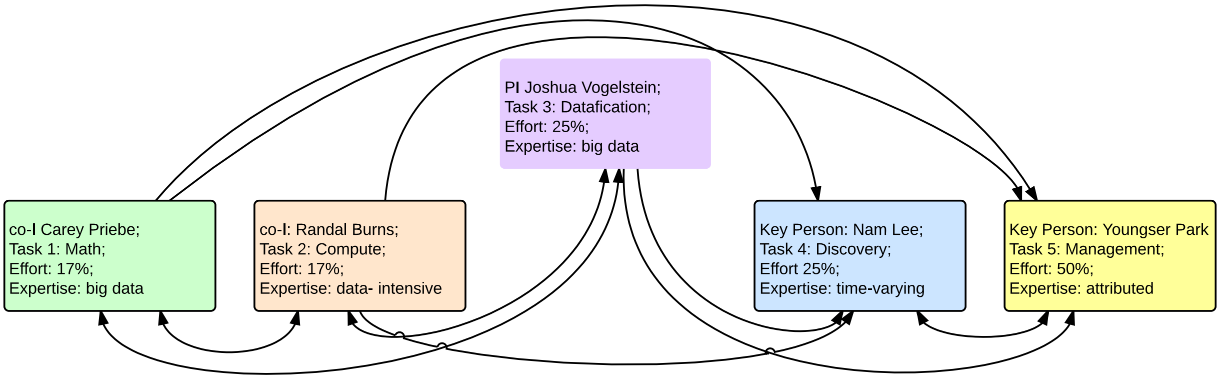


Figure 7: From RAGs to Riches Org-Chart.

work on the same floor in the same building at Johns Hopkins University. Co-I Burns's office is just across the quad. The PI and the two co-I's share graduate students and post-docs, in fact, some of PI Vogelstein's students work in co-I Burns's lab space. The three investigators are jointly funded through both XDATA and GRAPHS; separately, co-I Burns and PI Vogelstein are jointly funded on several NIH awards, as well as private funding.

The RAGs to Riches team, as illustrated in Figure 8, is tightly integrated. Each member of the team will be responsible for a single task throughout all three phases of the program:

1. Co-I Priebe will manage Task 1, all aspects of mathematical framework, working closely with co-I Burns to ensure scalability of the developed methodology. Priebe will also work with PI Vogelstein, and oversee both of the key personnel, Nam Lee and Youngser Park.
2. Co-I Burns will manage all Task 2 activities, including design and administration of computational resources, and implementation of all scalable algorithms, databases, and interfaces.
3. PI Vogelstein will manage Task 3, datafication, extensive utilizing the ideas of co-I Priebe and implementations of co-I Burns.
4. Key Personnel Nam Lee will be responsible for Task 4, discovery. This will largely entail taking the registered data products from PI Vogelstein, and deploying the methodologies developed by co-I's Priebe and Burns.
5. Key Personnel Youngser Park will contribute to all Task 1 efforts with co-I Priebe, as well a work closely on discovery with Dr. Lee. Dr. Park will also assist Dr. Vogelstein with project management duties and his task responsibilities will include assisting with the coordination of meetings, technical reports. He has served as Project Manager for Dr. Carey Priebe in various projects in the past.

The team will collectively oversee three graduate students for the duration of the program. The three investigators will each be responsible for the activity of one of the graduate students. The close-knitness of the team will mean that all three graduate students have extensive interactions with each of the team members.

Dr. Vogelstein will serve as the main point of contact with the DARPA PM. For Task 5, he will 1) oversee all of the internal JHU efforts for the proposed project, 2) manage the budget, 3) oversee writing of reports and research papers, and maintenance of project websites, 4) oversee participation in face-to-face meetings involving travel. The PI will also manage Task 3, datafication, extensively utilizing the ideas of co-I Priebe and implementations of co-I Burns.

The team will communicate through regular weekly meetings, which will be conducted in person or by teleconference and include all team members. Overall and individual task progress will be assessed monthly and recommendations for any necessary effort focus or modifications to meet program milestones will be issued by the PI. **Identifying risks due to technical issues and establishing mitigation strategies will be addressed at the weekly teleconference.** This will include mitigation of dependency risk, for example, FlashGraph implementations of context-aware algorithms must follow from the key development of the intial algorithms in Phase I. In the event of impending risk, PI Vogelstein will re-assign teammates appropriately to ensure successful completion of all milestones in the proposed timeline. The PI and other project staff as appropriate will participate in all DARPA meetings and collaboration meetings with TA1 partners, as determined by DARPA. Reports and software will be shared between all team members though websites hosted at JHU. Publication of all research work will be encouraged.

Importantly, the groups that are contributing to the data that the team will use for datafication and discovery are all in regular contact with the team. Twice annually, members of the team will meet members from the data contributors laboratory for extensive face-to-face time. Already the team has a day long scheduled meeting with Karl Deisseroth's data collection guru.

Finally, we note that this team already publishes together extensively in a wide range of venues, including the top computer science, applied mathematics, statistics, and neuroscience journals. Moreover, we also have already co-published with the data generator groups, and have several additional publications at various stages of acceptance. PI Vogelstein will ensure that everything runs smoothly, and will intervene, as necessary, to guarantee successful program completion.

VII Personnel, Qualifications, and Commitments

Joshua T. Vogelstein received a BS degree in Biomedical Engineering from Washington University in St. Louis in 2002, a MS in Applied Mathematics and Statistics (AMS) and a PhD in Neuroscience in 2009 from Johns Hopkins University. He then spent a couple years as a Post-Doctoral Fellow with Carey Priebe, followed by a brief appointment as Research Faculty in AMS, and then to Duke University for another Research Faculty appointment at Duke's big data center. He recently came back home to Johns Hopkins University, where he is an Assistant Professor in Biomedical Engineering, as well as core faculty at the Institute for Computational Medicine and the Center for Imaging Science, with joint appointments in the works for Computer Science, Neuroscience, and Biostatistics. He is also a member of the Institute for Data Intensive Engineering and Sciences and the Human Language Technology Center of Excellence. He has published in a wide variety of top venues, ranging from Science and Science Translation Medicine, and Nature Methods, to Annals of Applied Statistics and Neural Information Processing Systems, to Neuron, Nature Neuroscience, and Journal of Neurophysiology.

As evidenced from his list of publications, Dr. Vogelstein works extensively on all aspects of this proposal, from mathematical theory to machine learning and signal processing, to neuroscience.

Levels of Effort. Dr. Vogelstein will commit 3 months (25%) of effort to this grant per year. He has the following sources of funding, noted with months of effort and project role.

3.0, co-PI	NIH PAR-12-806 (NIBIB), CRCNS: Data Sharing: The EM Open Connectome Project. 09/01/12–08/31/15.
3.0, co-PI	DARPA N66001-14-1-4028, Scalable Brain Graph Analyses using Big-Memory High-IOPS Compute Architectures. 05/01/2014-11/31/2015.
3.0, co-PI	NIH 1R01NS092474-01, Synaptomes of Mouse and Man. 09/30/14–06/30/19. .
0.5, co-PI	NIH NIDA 1R01DA036400-01, BIGDATA: Small: DCM: ESCA: DA: Computational infrastructure for massive neuroscientific image stacks. 03/15/13–03/14/16.

Carey E. Priebe received the BS degree in mathematics from Purdue University in 1984, the MS degree in computer science from San Diego State University in 1988, and the PhD degree in information technology (computational statistics) from George Mason University in 1993. From 1985 to 1994 he worked as a mathematician and scientist in the US Navy research and development laboratory system. Since 1994 he has been a professor in the Department of Applied Mathematics and Statistics, Whiting School of Engineering, Johns Hopkins University, Baltimore, Maryland. At Johns Hopkins, he holds joint appointments in the Department of Computer Science, the Department of Electrical and Computer Engineering, the Center for Imaging Science, the Human Language Technology Center of Excellence, and the Whitaker Biomedical Engineering Institute. He is a past President of the Interface Foundation of North America - Computing Science & Statistics, a past Chair of the American Statistical Association Section on Statistical Computing, a past Vice President of the International Association for Statistical Computing, and on the editorial boards of Journal of Computational and Graphical Statistics, Computational Statistics and Data Analysis, and Computational Statistics. His research interests include computational statistics, kernel and mixture estimates, statistical pattern recognition, statistical image analysis, dimensionality reduction, model selection, and statistical inference for high-dimensional and graph data. He is a Senior Member of the IEEE, a Lifetime Member of the Institute of Mathematical Statistics, an Elected Member of the International Statistical Institute, and a Fellow of the American Statistical Association. Professor Priebe is a Research Professor in the National Security Institute at the Naval Postgraduate School, and was named one of six inaugural National Security Science and Engineering Faculty Fellows. He was the 2010 recipient of the Distinguished Achievement Award from the American Statistical Association Section on Statistics in Defense and National Security. He is a member of the NSA Advisory Board Mathematics Panel, and holds various security clearances.

Levels of Effort. Dr. Priebe will commit 2 months (17%) of effort to this grant per year. He has the following sources of funding, noted with months of effort and project role.

1.0, PI DARPA FA8750-12-2-0303, Fusion and Inference from Multiple and Massive Disparate Distributed Dynamic Data Sets. 09/10/12 - 03/09/17.

2.0, co-PI DOD H9823007C0365, Human Language Technology Center of Excellence. 01/13/07 - 01/12/14.

1.0, PI NSF DBI-1451081, Brain Eager: Discovery and Characterization of Neural Circuitry from Behavior. 09/01/14 - 08/31/16.

0.25, co-PI DARPA N66001-14-1-4028, Scalable Brain Graph Analyses using Big-Memory High-IOPS Compute Architectures. 05/01/2014-11/31/2015.

Randal Burns builds the high-performance, scalable data systems that allow scientists to make discoveries through the exploration, mining, and statistical analysis of big data. These include the Open Connectome Project (openconnecto.me) [108] and the JHU Turbulence Databases (turbulence.pha.jhu.edu) [109]. His research contributions tear down the barriers to using massive amounts of data either by making data access more efficient or improving the performance of I/O and memory systems [4; 110]. Burns' research approach embeds his group (students, postdocs, and programmers) in multi-disciplinary research teams with domain scientists in order to create the data systems and analysis tools that they use on daily basis. This approach ensures that research results create new analysis capabilities that transform scientists' ability to extract knowledge from data. For example, Eyink et al. [111] exploited a parallel database search to show that a 70-year-old belief about high-conductivity plasmas—magnetic flux freezing—fails in the presence of MHD turbulence, explaining why solar flares can erupt in minutes or hours rather than the millions of years predicted by flux freezing.

Levels of Effort. Burns will commit 2 months (17%) of effort to this grant per year. He has the following sources of funding, noted with months of effort and project role.

1.0, PI NIH PAR-12-806 (NIBIB), CRCNS: Data Sharing: The EM Open Connectome Project. 09/01/12–08/31/15.

0.72, co-PI CIF21 DIBBs: Long Term Access to Large Scientific Data Sets: The SkyServer and Beyond. 10/01/13–09/30/18.

1.0, PI DARPA N66001-14-1-4028, Scalable Brain Graph Analyses using Big-Memory High-IOPS Compute Architectures. 05/01/2014-11/31/2015.

1.0, PI NIH 1R01NS092474-01, Synaptomes of Mouse and Man. 09/30/14–06/30/19.

0.12, co-PI NIH NIDA 1R01DA036400-01, BIGDATA: Small: DCM: ESCA: DA: Computational infrastructure for massive neuroscientific image stacks. 03/15/13–03/14/16.

Youngser Park received the B.E. degree in electrical engineering from Inha University in Seoul, Korea in 1985, the M.S. and Ph.D. degrees in computer science from The George Washington University in 1991 and 2011 respectively. From 1998 to 2000 he worked at the Johns Hopkins Medical Institutes as a senior research engineer. From 2003 until 2011 he worked as a senior research analyst, and has been an associate research scientist since 2011 in the Center for Imaging Science at the Johns Hopkins University. At Johns Hopkins, he holds joint appointments in the Department of Applied Mathematics and Statistics and the Human Language Technology Center of Excellence. He has reviewed papers for ACM Transactions on Knowledge Discovery in Data, Statistical Analysis and Data Mining, and WIREs Computational Statistics. His current research interests are clustering algorithms, pattern classification, and data mining for high-dimensional and graph data.

Levels of Effort. Dr. Park will commit 6 months (50%) of effort to this grant per year. He has the following sources of funding, noted with months of effort and project role.

1.0, PI DARPA FA8750-12-2-0303, Fusion and Inference from Multiple and Massive Disparate Distributed Dynamic Data Sets. 09/10/12 - 03/09/17.

0.72, co-PI DOD H9823007C0365, Human Language Technology Center of Excellence. 01/13/07 - 01/12/14.

1.0, PI NSF DBI-1451081, Brain Eager: Discovery and Characterization of Neural Circuitry from Behavior. 09/01/14 - 08/31/16.

Nam Lee received the B.A., M.S., and Ph.D. degrees in mathematics from the University of California, San Diego, CA, USA, in 2002, 2004, and 2008, respectively. He is currently an Assistant Research Professor in the Department of Applied Mathematics and Statistics, Whiting School of Engineering, the Johns Hopkins University, Baltimore, MD, USA. His research interests include stochastic processes, statistical pattern recognition, and their applications to network science.

Levels of Effort. Dr. Nam Lee will commit 3 months (25%) of effort to this grant per year. He has the following source of funding, noted with months of effort and project role.

4.0, co-PI DOD H9823007C0365, Human Language Technology Center of Excellence. 01/13/07 - 01/12/14.

VIII Capabilities

VIII.A Open Connectome Project

We have previously built a petascale data-intensive computing infrastructure designed around serial electron microscopy images [79]. As we will explain in greater detail below, in this proposal, we will extend existing capabilities in a number of ways. First, we will greatly expand the different modalities and scales of data. This will enable us to ingest and query data from worms, flies, mice, monkeys, and humans, all using the exact same API. Second, we will greatly exapnd visualization capabilities, including both two- and three-dimensional (2D and 3D) local and remote capabilities, to further lower the barrier to entry, enabling citizen scientists to interact with the data as we do. Third, we will significantly enhance support for adding annotations of a wide variety of types to each data product. This includes image annotations (for example, different atlases or local statistics), text annotations (such as names, relavant citations, etc.), and connectivity annotations. Fourth, we will add significant analytical capabilities to the users.

Collectively, these modifications will enable workflows such as the following: (i) search for all connections to hippocampus in any brain of any animal, (ii) visualize each brain (only those parts connected to hippocampus) in 3D, (iii) grade each in terms of quality, (iv) compute an average neighborhood of hippocampus, weighted by quality score, (v) push such annotation back to the server so that others may now query against it.

VIII.B Graph Analytics

We and others have begun foundational statistical theory for the analysis of single graphs [15; 19; 20; 25; 28–36; 39; 44; 46; 47; 50–53; 56–58; 60; 61; 107; 112–123]. Similarly, we have worked on computing distances between graphs [62], as well as matching a pair of graphs [27; 37; 41; 98; 99], including weighted graphs, graphs with different numbers of vertices, etc. [42]. However, theory for jointly embedding populations of graphs, foundational for testing as well as both supervised and unsupervised learning remains completely absent. This includes both matched graphs (where we need not align the vertices), and unmatched graphs (where we must either align or otherwise be invariant to alignment).

VIII.C FlashGraph

FlashGraph was developed to enable efficient processing of massive individual graphs on commodity machines [77]. Specifically, it was designed for graph traversal algorithms, such as breadth-first-search and connected components. We will extend it to support semi-external memory implementations of linear algebra subroutines (including sparse matrix operations), as well as operate on populations of graphs

VIII.D GRAPHS vs. SIMPLEX

With respect to the above background subsections, our GRAPHS proposal does not address §VIII.A at all. In GRAPHS we do develop a few statistical methods on graphs, but we do not build a foundational theory of richly attributed graphs upon which a great variety of further applications may be developed. In terms of FlashGraph, our GRAPHS proposal only addresses graph traversal style algorithms for single simple graphs, versus here where we propose to extend it to include populations of graphs and a whole suite of dense and sparse matrix operations on graphs. Thus, this proposal is entirely complementary to our SIMPLEX proposal, extending the deliverables significantly. Specifically, this proposal will enables us to fuse previously totally disparate methods and code bases under a single coherent roof to enable previously unconceptualized capabilities, both internally and for the community.

VIII.E Specialized Facilities

This proposal is heavily computational, and hence we describe the computational resources available to us at this time. Dr. Vogelstein and Dr. Priebe, as core faculty in the Center for Imaging Science (CIS), have access to the CIS cluster; Dr. Vogelstein, as core faculty also in the Institute for Computational Medicine (ICM), as access to the ICM cluster; and Dr. Vogelstein and Dr. Burns, as members of the Institute for Data Intensive Engineering and Sciences (IDIES), have access to those facilities. Moreover, the three faculty are utilizing resources from Dr. Vogeslstein's startup package to build a new dedicated "Bruster" for doing brain computations. Finally, Dr. Priebe and Dr. Vogelstein, as members of CIS, have an allocation of one million compute hours on XSEDE[14], a nationally supported compute cluster. The faculty and personnel on this proposal have been utilizing these resources in preliminary work for this proposal, as well as the funded CRCNS grant (CRCNS-1208044; co-PIs Vogelstein, Burns, et al.) and Big Data grant (BIGDATA-1251208; co-PIs Vogelstein, Burns, et al.), a CRCNS grant on computational infrastructures upon which the methods developed herein can operate at accelerated rates (NSF Proposal ID-1311505), and several others. Below we briefly describe current resources.

The GrayWulf Cluster GrayWulf is a distributed database cluster at JHU consisting of 50 database nodes with 22TB and an 8-core server each, for a total of 1.1PB. The cluster was purchased on funds from the

[14] https://www.xsede.org/

Gordon and Betty Moore Foundation, the Pan-STARRS project and Microsoft Research. The cluster already hosts several large datasets (Pan-STARRS, turbulence, SDSS, various Virtual Observatory catalogs and services, environ- mental sensor data, computer security datasets, network traffic analysis data, etc). Currently about 800TB is already utilized. The cluster has an IO performance exceeding many supercomputers: the aggregate sequential read speed is more than 70 Gbytes/sec. The internal connectivity has recently been upgraded to 10Gbps Ethernet.

The HHPC Cluster The same computer room hosts a 1400 core BeoWulf cluster, a computational facility shared among several JHU faculty. The HHPC and the GrayWulf share a common 288-port DDR Infiniband switch for an extremely high-speed interconnect. There is an MPI interface under development that will enable very fast peer-to-peer data transfers between the compute nodes and the database nodes. The Deans of the JHU Schools provide funds to cover the management and operational costs of the two connected clusters. The second generation of the cluster is about to be purchased, adding another 2,400 cores to the system.

NSF-MRI NVIDIA cluster JHU has received an NSF MRI grant (CMMI-0923018), to purchase a large GPU cluster. We have 100 Fermi C2050's in production. Burns is a co-PI on this grant. There is a natural cohesion between the Data-Scope and the GPU cluster, and coordination between the principal architects of the two systems.

Bloomberg 156 Data Center The NSF has awarded a $1,337,272 grant for "Advanced CyberInfrastructure for High Performance Data Intensive Computing" (OCI-0963185). This infrastructure project renovated room 156 in Bloomberg Hall to create a flexible, stable environment for a high density of computing equipment that supports research and research training on the Homewood Campus. The 3100 square foot room is covered with a raised floor fed with cold air from seven Liebert air conditioners, and a dedicated chilled water line is available for water-cooled racks. Bloomberg 156 supports a steady load of at least 450kVA, with potential expansion to 750kVA. To ensure a stable environment for data repositories, 150kVA of power has both battery and generator backup. The grant also upgraded the network infrastructure supporting the space from 1GigE to 10GigE to insure that users throughout campus can access the data center effectively and that data can be streamed to and from the outside world through Internet2. This network infrastructure includes a Cisco Nexus 7000 chassis that accommodate the 100GigE uplink to Internet 2. The room has collocated the GPU cluster, HHPC, and Data-Scope. This has created a tightly coupled, heterogeneous clusters with compute, data, and GPU components, allowing computational science to be done in new ways. The center serves as a focal point for interdisciplinary activities in computational science and engineering.

Data-Scope The NSF has recently awarded a $2M MRI grant (OCI MRI-1040114, co-PI Burns) to build a 6.5PB cluster for data-intensive computations. The system components have arrived and are being commissioned. The system will have an aggregate sequential throughput in excess of 500GBps, and will also contain 90 GPU cards, providing a substantial floating point processing capability. The Data-Scope will be connected entirely through 10Gbps Ethernet. There is an ongoing NSF-funded effort to bring 100G connectivity to JHU, through a collaboration with the Mid-Atlantic Crossroads (MAX). The system components are currently evaluated and tested, and deployment is expected to happen in June.

The driving goal behind the Data-Scope design is to maximize stream processing throughput over TB- size datasets while using commodity components to keep acquisition and maintenance costs low. Performing the first pass over the data directly on the servers' PCIe backplane is significantly faster than serving the data from a shared network file server to multiple compute servers. This first pass commonly reduces the data significantly, allowing one to share the results over the network without losing performance. Furthermore, providing substantial GPU capabilities on the same server enables us to avoid moving too much data across the network as it would be done if the GPUs were in a separate cluster.

Open Connectome Project Mini-Cluster By virtue of a Dean's graph to Professor Burns, Professor Burns and Dr. Vogelstein have established an Open Connectome Project Mini-Cluster. This cluster contains four "braincrunch" machines, each 4-core development boxes, along with two "braingraph" machines, each currently serving about 20TB of brain imaging data, and finally one "awesome" machine, which has 500GB of RAM and 16 128GB solid state hard drives. These machines are all actively used already for this project by PI Vogelstein and co-I Burns.

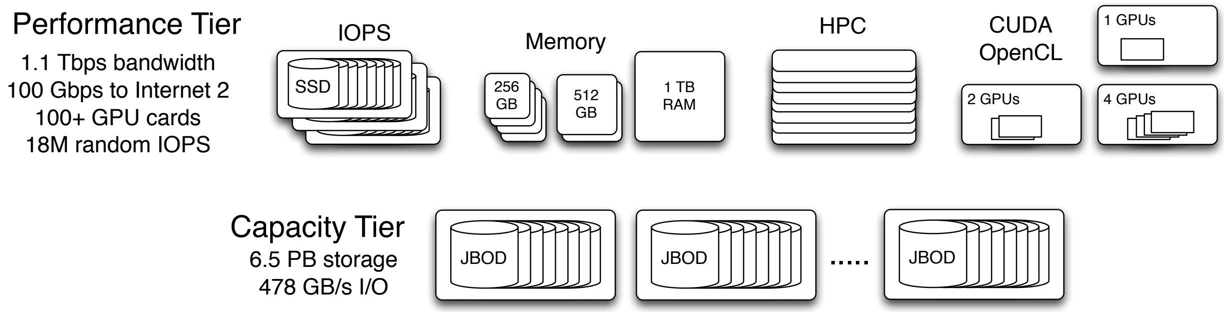


Figure 8: Schematic illustration of our DataScope compute cluster.

Center for Imaging Science Computing Resources The Center for Imaging Science (CIS) offers extensive computing resources. The major computational machine within CIS is the Intel Itanium2 cluster, which is a 32 processor cluster that is part of the TeraGrid, which includes over 20 teraflops of computing power distributed at 9 sites, facilities capable of managing and storing over 1 PB of data and high-resolution visualization environments. The CIS infrastructure also includes two 32GB/8cpu and one 128GB/16cpu computational servers; 120TB of data storage; three tape libraries with a backup capacity of over 210TB; and over 35 visualization workstations.

Institute for Computational Medicine Computing Resources The Institute for Computational Medicine (ICM) offers additional computing resources, including 250 node, 2000 core, IBM iDataPlex cluster attached to a 1 Petabyte Storage Area Network, and an IBM TotalStorage 3584 UltraScalable tape library with 4 LTO4 drives and 250 slots.

IX Statement of Work

IX.A Phase I

IX.A.1 Task 1: Mathematical Formalism

- **Goal:** *RAG Embedding:* Completion of the theoretical development and associated data structures of our RAG representation system. This includes establishing baseline methods for embeddings RAGs and populations thereof. Specifically, we will explore both JOFC and tensor factorization methodologies, to enable understanding of the computational and statistical advantages and disadvantages of each for embedding high-dimensional non-Euclidean RAGS.
- **Primary Site:** JHU
- **Milestone:** Demonstration that RAGs are able to meet TA1 goals, including encoding quantitative and qualitative knowledge, and express functional relationship among entities in complex systems.
- **Deliverables:** Description of mathematical framework and preliminary benchmarks evaluating performance on open access data sets and simulations.

IX.A.2 Task 2: Computational Infrastructure

- **Goal:** *Data Management:* Implementation of baseline algorithms for context-aware reasoning and inference using the representation. Development of an initial computational and data management platform, including establishing common data formats, common methods and format for query and analysis of results, and a common API through which all domain-specific users will access the framework. We will also extend our dense spatial and semantic databases, as well as our graph data format to support time-varying data, along with multi-modal data.
- **Primary Site:** JHU
- **Milestone** Completion of Phase I prototype software and services.
- **Deliverables:** Open source software and documentation for end-to-end prototype.

IX.A.3 Task 3: Datafication

- **Goal:** *Data Ingest:* Completion of data ingest techniques. Completion of research and design into microscopic and mesoscopic specific analysis tools. Demonstration of auto-data ingestion and registration of multiple different modalities (functional to structural for both microscopic and mesoscopic data). More specifically, we will have ingested CLARITY, LFM, and M^{3}RI data into the same database schema; all M^{3}RI data will be co-registered.
- **Primary Site:** JHU
- **Milestone:** Demonstration of operational auto-ingestion and registration on two different use-cases.
- **Deliverables:** Open source software and documentation for datafication techniques, as well as image datasets ingested and RAGs estimated from all different data modalities.

IX.A.4 Task 4: Discovery

- **Goal:** *RAG Construction:* Completion of research and design into microscopic and mesoscopic specific analysis tools, by designing metrics appropriate for the different data modalities. This includes both the microscale and mesoscale functional time-series, converting into RAGs via utilizing qualitative information.
- **Primary Site:** JHU
- **Milestone:** Demonstration of RAG construction on both use cases.
- **Deliverables:** Open source software and documentation RAG construction techniques, as well as the derived RAGs available via our Web-services.

IX.A.5 Task 5: Program Management

- **Goal:** *Phase I:* Ensure successful execution of the effort. Manage the proposed effort using a proven methodology for project planning, resource allocation, task specification, and monitoring. Establish a baseline project plan with a list of tasks, specifications, requirements, and timelines; update plan periodically; document updates to the plan and share them with the project team and DARPA PM.
- **Primary Site:** JHU
- **Milestone** Meet Phase I goals.
- **Deliverables:** (1) Comprehensive quarterly technical reports that include updates systems architecture and progress made on milestones for Phase I; (2) Brief month reports, including preprints of technical reports; (3) Final Technical Report; (4) Monthly Financial Reports.

IX.B Phase II

IX.B.1 Task 1: Mathematical Formalism

- **Goal:** *FlashRAG:* Implementation of all embedding and construction methodologies in FlashGraph to enable scalable implementations and processing. Moreover, all constructed RAGs will obtain multilevel representations. We will build R bindings to enable easy use of FlashGraph for data scientists. We will check that our implementations and bindings yield approximately the same answer as benchmark methods, on data sufficiently small that benchmark methods can run.
- **Primary Site:** JHU
- **Milestone** Fully operational FlashGraph and R bindings for embedding and constructing methodologies.
- **Deliverables:** Open source software and documentation for end-to-end prototype for embedding and constructing RAGs. This includes an R package for FlashGraph.

IX.B.2 Task 2: Computational Infrastructure

- **Goal:** *Remote Access:* Implementation of prototype platform for remote access. This will include Web-services for uploading the raw data, and downloading the derived data products (RAGs and intermediate data products), as well as both 2D and 3D visualization and annotation tools, which will support multiple kinds of analytic overlays, all of which will support multiple data scales. Moreover, we will have made theoretical and practical refinements to the representation to enable scalable implementation of several foundational algorithms on RAGs, implementing the embedding methodologies developed in Task 1 of Phase I into our semi-external memory formalism.

- **Primary Site:** JHU
- **Milestone** Fully operational Web-services supporting uploading, visualizing, annotation, querying, downloading, and analyzing the data.
- **Deliverables:** Open source software and documentation for end-to-end prototype. This includes an R package for FlashGraph which extends it capabilities to RAGs, rather than simply graphs.

IX.B.3 Task 3: Datafication

- **Goal:** *Data Register:* Integration of domain-specific computational models across modalities and scales. This includes completion of statistical multi-modal referencing, including alignment of structural and functional imaging data, for both microscopic and mesoscopic data sets. We will also complete functional inference capabilities. We will align data both via scaling up multidimensional out-of-core image alignment algorithms, and RAG matching, which extends graph matching by incorporating attributes. This will enable us to determine optimal alignments using data priors and known topological structure, rather than relying on images to align well.
- **Primary Site:** JHU
- **Milestone:** Demonstration of multi-modal registration for both microscopic and mesoscopic use cases.
- **Deliverables:** Open source software and documentation for datafication techniques, as well as registered multi-modal image datasets ingested and aligned RAGs from both microscale and mesoscale.

IX.B.4 Task 4: Discovery

- **Goal:** *RAG Summary Statistics:* Utilize RAG knowledge representation to estimate population moments, motifs, and/or modes from both micro- and meso-scale RAGs. More specifically, we will utilize the various joint embedding methodologies developed in Task 1 to estimate these summary statistics. The different approaches, JOFC versus tensor factorization, will enable incorporating different kinds of prior knowledge and constraints, so they will therefore lead to different bias/variance trade-offs. We will explore these options empirical on the real data, to complement our experiments in Task 1, to discover both (i) the best methods for estimation these summary statistics, and (ii) the best estimates of the summary statistics for the two different scales.
- **Primary Site:** JHU
- **Milestone:** Demonstration utility of RAG representation of data for estimating summary statistics for multi-modal data.
- **Deliverables:** Estimated summary statistics from micro- and meso-scale RAGs available for download in various formats, as well as open source code for the different estimators.

IX.B.5 Task 5: Program Management

- **Goal:** *Phase II Goals:* Ensure successful execution of the effort. Manage the proposed effort using a proven methodology for project planning, resource allocation, task specification, and monitoring. Establish a baseline project plan with a list of tasks, specifications, requirements, and timelines; update plan periodically; document updates to the plan and share them with the project team and DARPA PM.
- **Primary Site:** JHU
- **Milestone** Meet Phase II goals.
- **Deliverables:** (1) Comprehensive quarterly technical reports that include updates systems architecture and progress made on milestones for Phase II; (2) Brief month reports, including preprints of technical reports; (3) Final Technical Report; (4) Monthly Financial Reports.

IX.C Phase III

IX.C.1 Task 1: Mathematical Formalism

- **Goal:** *RAG Testing:* Demonstration of capabilities and objectives on both microscale and mesoscale data, as well as one additional SIMPLEX performer. To achieve this, we will extend our embedding methodologies, to derive provably approximately optimal embeddings for conducting one-sample and two-sample tests on RAGs; two fundamental testing procedures in statistics. Our tests will leverage our ability to efficiently sample RAGs, as they will be resampling based tests, analogs to the classic parametric and non-parametric bootstrap. We will apply these tests to test, for example, whether our data are sampled from relatively simple RAGs statistical models, and whether population means that we obtained in Phase I are significantly different from one another.
- **Primary Site:** JHU
- **Milestone:** Demonstrate our mechanism for relating qualitative and quantitative knowledge, and relating multiple heterogeneous datasets, on both heterogeneous scales (use cases). Specifically, testing whether multiple heterogeneous datasets are statistically different from one another.
- **Deliverables:** Description of capabilities, emphasizing generalizability to multiple use cases, open source code from implementing our tests, visualizations and numerical summaries of test results.

IX.C.2 Task 2: Computational Infrastructure

- **Goal:** *Local Analysis:* Completion of an integrated system on both heterogeneous scales (as well as additional domains). This system ingests and registers imaging data and semantic and qualitative knowledge, converts them into RAGs, allows query and recall, visualization, hypothesis generation, and analysis. We will also release open source packages containing all of the key resources, such as an R package (which calls igraph or FlashGraph) containing all of the developed methods, and GPU optimized visualization and annotation tools. This will enable anybody to implement analyses locally by running the code on their machine.
- **Primary Site:** JHU
- **Milestone** Fully operation Web-services to auto-ingest, register, store in compact representation, and query, as well as operate locally.
- **Deliverables:** Open source software and documentation for end-to-end prototype. including FlashGraphR and GPU optimized visualization and annotation tool.

IX.C.3 Task 3: Datafication

- **Goal:** *Quality Control:* Completion of quality control of all datasets. For each different modality, and each different scale, we will have already converted the raw data into RAGs. Now, we will build automatic quality controls, so that with each dataset, we automatically generate a quality control report, quantifying the key quality metrics appropriate for that data (see Table 2).
- **Primary Site:** JHU
- **Milestone:** All datasets have been checked for quality.
- **Deliverables:** Open source software and documentation for datafication techniques, including quality control scripts, and resulting outputs.

IX.C.4 Task 4: Discovery

- **Goal:** *RAG Prediction:* Completion of toolset for analysis, modeling, and data-driven hypothesis generation and testing in both microscale and mesoscale heterogeneous use cases. This will include multiscale prediction; prediction of mouse status from microscale RAGs, and human personality from human RAGs.
- **Primary Site:** JHU
- **Milestone:** Successful integration with TA1 technology and end-of-program demonstrations of our integrated system on both microscale and mesocale.
- **Deliverables:** All data-derived products available for visualization utilizing our Web-services and quality control pages, as well as for download and further analysis using our open source code.

IX.C.5 Task 5: Program Management

- **Goal:** *Phase III:* Ensure successful execution of the effort. Manage the proposed effort using a proven methodology for project planning, resource allocation, task specification, and monitoring. Establish a baseline project plan with a list of tasks, specifications, requirements, and timelines; update plan periodically; document updates to the plan and share them with the project team and DARPA PM.
- **Primary Site:** JHU
- **Milestone** Meet Phase III goals.
- **Deliverables:** (1) Comprehensive quarterly technical reports that include updates systems architecture and progress made on milestones for Phase III; (2) Brief month reports, including preprints of technical reports; (3) Final Technical Report; (4) Monthly Financial Reports.

X Schedule and Milestones

Figure 9 provides the list of milestones organized by time and task. Please see §IX for details.

XI Cost Summary

Phase	Task	#	Sub-Task	Phase I					Phase II				Phase III				Responsible
	GFY		Milestone	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Agent
I	RAG Embedding	I.1.A	Tensor Factorization	■	■												Priebe
		I.1.B	JOFC			■	■										Priebe
		I.1.C	Benchmarking					■									Priebe
II	FlashRAG	II.1.A	FlashMatrix						■	■							Priebe
		II.1.B	FlashAttributes							■	■						Priebe
		II.1.C	FlashR									■					Priebe
III	Rag Testing	III.1.A	1-sample										■	■			Priebe
		III.1.B	2-sample											■	■		Priebe
		III.1.C	independence													■	Priebe
I	Data Management	I.2.A	Dense Arrays	■	■												Burns
		I.2.B	Sparse Arrays		■	■											Burns
		I.2.C	Sparse Cutouts				■	■									Burns
II	Remote Access	II.1.A	2D Web Viz						■	■							Burns
		II.1.B	Surface Extraction							■	■						Burns
		II.1.C	3D Web Viz							■	■						Burns
		II.1.D	Graph Viz								■	■					Burns
III	Local Analysis	III.1.A	API										■	■			Burns
		III.1.B	Downloads											■	■		Burns
		III.1C	GPU 3D Viz												■	■	Burns
		III.1.D	Remote Annotation												■	■	Burns
I	Data Ingest	I.3.A	diffusion MRI	■	■												Vogelstein
		I.3.B	functional MRI		■	■											Vogelstein
		I.3.C	CLARITY			■	■										Vogelstein
		I.3.D	LFM				■	■									Vogelstein
II	Data Register	II.3.A	Human Align						■	■							Vogelstein
		II.3.B	Mouse Align							■	■						Vogelstein
		II.3.C	Multi-Graph-Match								■	■					Vogelstein
III	Quality Control	III.3.A	diffusion MRI										■	■			Vogelstein
		III.3.B	functional MRI											■	■		Vogelstein
		III.3.C	CLARITY												■	■	Vogelstein
		III.3.D	LFM												■	■	Vogelstein
I	RAG Construct	I.4.A	Random Walks	■	■												Lee
		I.4.B	Graph Sparsitifcation		■	■											Lee
		I.4.C	Benchmarking				■	■									Lee
II	RAG Summary Statistics	II.4.A	Moments						■	■							Lee
		II.4.B	Motifs							■	■						Lee
		II.4.C	Modes								■	■					Lee
III	RAG Predict	III.4.A	Nearest Neighbor										■	■			Lee
		III.4.B	Network of Networks											■	■		Lee
		III.4.C	Tensor Factorization											■	■		Lee
		III.4.D	Benchmarking												■	■	Lee
	Monthly Tech Reports	5		X	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	Vogelstein
	Monthly Financial Reports	5		X	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	Vogelstein
	Quarterly Reports	5		X	X	X	X	X	X	X	X	X	X	X	X	X	Vogelstein
	Final Report	5						X				X				X	Vogelstein

Figure 9: Schematic listing all milestones for this proposal. Black elements of this array denote quarters during which the work for that specific subtask will be completed. Deliverables will be delivered in the final quarter for the subtasks. Key milestone for each task in each phase correspond to the names of those tasks in §IX, and §V lists all subtasks. For Task 5 (bottom rows), the number of 'X's per element denotes the number of times the reports will be generated and sent.

GY 15 03/01/15-09/30/15	**Task 1**	**Task 2**	**Task 3**	**Task 4**	**Task 5**	**Total**
Labor Costs	62,141	33,662	28,296	21,143	417	145,660
Fringe	15,793	5,968	4,117	7,294	144	33,315
Travel					6,927	6,927
Other Direct Costs					3,733	3,733
Tuition	8,538	8,538	8,538			25,613
Total Direct Costs	**86,472**	**48,168**	**40,951**	**28,437**	**11,221**	**215,250**
F&A	48,319	24,571	20,097	17,631	6,957	117,575
Total Costs	**134,791**	**72,739**	**61,048**	**46,068**	**18,178**	**332,825**

GFY 16 10/01/15-09/30/16	**Task 1**	**Task 2**	**Task 3**	**Task 4**	**Task 5**	**Total**
Labor Costs	109,145	59,125	49,700	37,136	732	255,837
Fringe	27,304	10,048	6,796	12,812	253	57,212
Travel					21,315	21,315
Other Direct Costs					7,013	7,013
Tuition	14,925	14,925	14,925			44,775
Total Direct Costs	**151,374**	**84,097**	**71,420**	**49,948**	**29,314**	**386,153**
F&A	84,598	42,887	35,027	30,968	18,175	211,655
Total Costs	**235,972**	**126,984**	**106,447**	**80,916**	**47,489**	**597,807**

GFY 17 10/01/16-09/30/17	**Task 1**	**Task 2**	**Task 3**	**Task 4**	**Task 5**	**Total**
Labor Costs	112,419	60,899	51,191	38,250	754	263,512
Fringe	28,124	10,349	7,000	13,196	260	58,929
Travel					22,506	22,506
Other Direct Costs					8,323	8,323
Tuition	15,762	15,762	15,762			47,285
Total Direct Costs	**156,304**	**87,010**	**73,952**	**51,446**	**31,843**	**400,555**
F&A	87,136	44,174	36,078	31,897	19,742	219,027
Total Costs	**243,440**	**131,184**	**110,030**	**83,343**	**51,585**	**619,582**

GFY 18 10/01/17-05/31/18	**Task 1**	**Task 2**	**Task 3**	**Task 4**	**Task 5**	**Total**
Labor Costs	76,620	41,506	34,889	26,070	514	179,598
Fringe	18,771	6,657	4,374	8,994	177	38,974
Travel					18,417	18,417
Other Direct Costs					5,659	5,659
Tuition	10,852	10,852	10,852			32,557
Total Direct Costs	**106,244**	**59,015**	**50,116**	**35,063**	**24,767**	**275,206**
F&A	59,142	29,861	24,343	21,739	15,358	150,443
Total Costs	**165,386**	**88,876**	**74,459**	**56,802**	**40,125**	**425,649**

Cost Summary	**Task 1**	**Task 2**	**Task 3**	**Task 4**	**Task 5**	**Total**
Labor Costs	360,324	195,191	164,076	122,598	2,417	844,607
Fringe	89,992	33,022	22,287	42,296	834	188,431
Travel					69,166	69,166
Other Direct Costs					24,728	24,728
Tuition	50,077	50,077	50,077			150,230
Total Direct Costs	**500,393**	**278,290**	**236,440**	**164,895**	**97,145**	**1,277,163**
F&A	279,195	141,493	115,545	102,235	60,232	698,700
Total Costs	**779,588**	**419,783**	**351,985**	**267,130**	**157,377**	**1,975,863**

XII Administrative and National Policy Requirements

(1). Team Member Identification(Key = Key Personnel):

Individual Name	Role	Organization	Non-US?		FFRDC or Govt?
			Org.	Ind.	
Joshua Vogelstein	Prime PI	JHU	N/A	N/A	N/A
Carey Priebe	Prime co-I	JHU	N/A	N/A	N/A
Randal Burns	Prime co-I	JHU	N/A	N/A	N/A
Youngser Park	Prime Key	JHU	N/A	N/A	N/A
Nam Lee	Prime Key	JHU	N/A	N/A	N/A

(2). Government or FFRDC Team Member Proof of Eligibility to Propose: NONE

(3). Government or FFRDC Team Member Statement of Unique Capability: NONE

(4). Organizational Conflict of Interest Affirmations and Disclosure: NONE

(5). Intellectual Property (IP): NONE

(6). Human Subjects Research (HSR): NONE

(7). Animal Use: NONE

(8). Representations Regarding Unpaid Delinquent Tax Liability or a Felony Conviction under Any Federal Law:

(a) The proposer represents that it is [ ] is not [X] a corporation that has any unpaid Federal tax liability that has been assessed, for which all judicial and administrative remedies have been exhausted or have lapsed, and that is not being paid in a timely manner pursuant to an agreement with the authority responsible for collecting the tax liability.

(b) The proposer represents that it is [ ] is not [X] a corporation that was convicted of a felony criminal violation under a Federal law within the preceding 24 months.

(9). Cost Accounting Standards (CAS) Notices and Certification: NONE.

Literature Cited

[1] "NKI/Rockland Country Sampling Strategy," *http://fcon_1000.projects.nitrc.org/indi/enhanced/recruit.html*.

[2] R. Tomer, L. Ye, B. Hsueh, and K. Deisseroth, "Advanced CLARITY for rapid and high-resolution imaging of intact tissues.," *Nature protocols*, vol. 9, pp. 1682–1697, July 2014.

[3] M. Broxton, L. Grosenick, S. Yang, N. Cohen, A. Andalman, K. Deisseroth, and M. Levoy, "Wave optics theory and 3-D deconvolution for the light field microscope.," *Optics express*, vol. 21, pp. 25418–25439, Oct. 2013.

[4] D. Zheng, R. Burns, and A. S. Szalay, "Toward millions of file-system IOPS on low-cost, commodity hardware," in *Supercomputing*, 2013.

[5] F. R. K. Chung and C. C. on Recent Advances in Spectral Graph Theory, "Spectral graph theory," 1997.

[6] B. Bollobas, *Modern Graph Theory*. Springer, 1998.

[7] B. Bollobás, *Random graphs*. Springer, 1998.

[8] D. B. West and Others, *Introduction to graph theory*, vol. 2. Prentice hall Upper Saddle River, 2001.

[9] P. Fjallstrom, "Algorithms for graph partitioning: A survey," *Computer and Information Science*, vol. 3, no. 10, 1998.

[10] B. Bollobás, *Extremal graph theory*. Courier Dover Publications, 2004.

[11] D. Conte, P. Foggia, C. Sansone, M. Vento, D. Conte, P. Foggia, C. Sansone, and M. Vento, "Thirty Years of Graph Matching in Pattern Recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 18, no. 3, pp. 265–298, 2004.

[12] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[13] T. A. B. Snijders and K. Nowicki, "Estimation and prediction for stochastic blockmodels for graphs with latent block structure," *Journal of Classification*, vol. 14, no. 1, pp. 75–100, 1997.

[14] J. G. DeVinney and C. E. Priebe, "A new family of proximity graphs: class cover catch digraphs," *Discrete Applied Mathematics*, vol. 154, no. 14, pp. 1975–1982, 2006.

[15] E. Ceyhan, C. E. Priebe, and D. J. Marchette, "A New Family of Random Graphs for Testing Spatial Segregation," *Canadian Journal of Statistics*, vol. 35, no. 1, pp. 27–50, 2007.

[16] S. J. Young, E. R. Scheinerman, S. Young, and E. Scheinerman, "Random dot product graph models for social networks," in *Proceedings of the 5th international conference on algorithms and models for the web-graph*, pp. 138–149, Springer, 2007.

[17] E. M.˜Airoldi, D. M.˜Blei, S. E.˜Fienberg, E. P.˜Xing, D. M. Blei, and E. P. Xing, "Mixed membership stochastic blockmodels," *The Journal of Machine Learning Research*, vol. 9, pp. 1981–2014, 2008.

[18] S. Chatterjee, P. Diaconis, and A. Sly, "Random graphs with a given degree sequence," *The Annals of Applied Probability*, vol. 21, pp. 1400–1435, Aug. 2011.

[19] N. H. Lee and C. E. Priebe, "A Latent Process Model for Time Series of Attributed Random Graphs," *Statistical Inference for Stochastic Processes*, vol. 14, pp. 231–253, June 2011.

[20] H. Pao, G. A. Coppersmith, and C. E. Priebe, "Statistical inference on random graphs: Comparative power analyses via Monte Carlo," *Journal of Computational and Graphical Statistics*, pp. 1–22, 2010.

[21] C. E. Priebe, G. A. Coppersmith, and A. Rukhin, "You say graph invariant, I say test statistic," *Statistical Computing Statistical Graphics Newsletter*, vol. 21, no. 2, pp. 11–14, 2010.

[22] Z. Ma, D. J. Marchette, and C. E. Priebe, “Fusion and Inference from Multiple Data Sources in Commensurate Space,” *Statistical Analysis and Data Mining*, pp. 1–7, 2011.

[23] C. E. Priebe, J. T. Vogelstein, and D. D. Bock, “Optimizing the quantity/quality trade-off in connectome inference,” *Communications in Statistics Theory and Methods*, p. 7, 2011.

[24] C. E. Priebe, J. T. Vogelstein, and D. D. Bock, “Optimizing the quantity/quality trade-off in connectome inference,” *Communications in Statistics Theory and Methods*, p. 7, 2011.

[25] A. Rukhin and C. E. Priebe, “A Comparative Power Analysis of the Maximum Degree and Size Invariants for Random Graph Inference,” *Journal of Statistical Planning and Inference*, vol. 141, no. 2, pp. 1041–1046, 2011.

[26] G. A. Coppersmith and C. E. Priebe, “Vertex Nomination via Content and Context,” *Technology*, pp. 1–21, 2012.

[27] D. E. Fishkind, S. Adali, and C. E. Priebe, “Seeded Graph Matching,” *arXiv preprint*, p. 1209.0367v1, 2012.

[28] D. E. Fishkind, D. L. Sussman, M. Tang, J. T. Vogelstein, and C. E. Priebe, “Consistent adjacency-spectral partitioning for the stochastic block model when the model parameters are unknown,” *Journal of the American Statistical Association*, vol. 107, pp. 1119–1128, 2012.

[29] D. S. Lee and C. E. Priebe, “Bayesian Vertex Nomination,” *Arxiv preprint*, 2012.

[30] C. E. Priebe, D. L. Sussman, M. Tang, and J. T. Vogelstein, “Statistical inference on errorfully observed graphs,” *under revision at Journal of Computational and Graphical Statistics*, 2013.

[31] L. Chen, J. T. Vogelstein, and C. E. Priebe, “Robust Vertex Classification,” *Under review*, p. 27, Nov. 2013.

[32] Y. Qin, D. Mhembere, S. Ryman, R. E. Jung, R. J. Vogelstein, R. Burns, J. T. Vogelstein, and C. E. Priebe, “Robust Clustering of Adjacency Embeddings of Brain Graph Data via Lq-likelihood,” in *Organization of Human Brain Mapping*, 2013.

[33] A. Rukhin and C. E. Priebe, “On the Limiting Distribution of a Graph Scan Statistic,” *Communications in Statistics - Theory and Methods*, vol. 41, no. 7, pp. 1151–1170, 2012.

[34] D. L. Sussman, M. Tang, D. E. Fishkind, and C. E. Priebe, “A consistent dot product embedding for stochastic blockmodel graphs,” *Journal of the American Statistical Association*, vol. in press, 2013.

[35] D. L. Sussman, M. Tang, and C. E. Priebe, “Universally Consistent Latent Position Estimation and Vertex Classification for Random Dot Product Graphs,” *arXiv preprint*, p. 1212.1182, 2012.

[36] A. Athreya, V. Lyzinski, D. J. Marchette, C. E. Priebe, D. L. Sussman, and M. Tang, “A limit theorem for scaled eigenvectors of random dot product graphs,” *arXiv preprint*, p. 1305.7388, May 2013.

[37] V. Lyzinski, D. L. Sussman, D. E. Fishkind, H. Pao, and C. E. Priebe, “Seeded graph matching for large stochastic block model graphs,” *arXiv preprint*, p. 1310.1297, Oct. 2013.

[38] V. Lyzinski, D. L. Sussman, M. Tang, A. Athreya, and C. E. Priebe, “Perfect clustering for stochastic blockmodel graphs via adjacency spectral embedding.” 2013.

[39] M. Tang, Y. Park, and C. E. Priebe, “Out-of-sample Extension for Latent Position Graphs,” *Arxiv preprint*, May 2013.

[40] D. Durante, D. B. Dunson, and J. T. Vogelstein, “Nonparametric Bayes Modeling of Populations of Networks,” *Arxiv preprint*, June 2014.

[41] V. Lyzinski, D. Fishkind, M. Fiori, J. T. Vogelstein, C. E. Priebe, and G. Sapiro, “Graph Matching: Relax at Your Own Risk,” May 2014.

[42] V. Lyzinski, S. Adali, J. T. Vogelstein, Y. Park, and C. E. Priebe, “Seeded Graph Matching Via Joint Optimization of Fidelity and Commensurability,” *Under review*, 2014.

[43] D. Koutra, N. Shah, J. T. Vogelstein, B. J. Gallagher, and C. Faloutsos, “DELTACON: A Principled Massive-Graph Similarity Function and Applications,” *in preparation*.

[44] J. T. Vogelstein and C. E. Priebe, “Shuffled Graph Classification: Theory and Connectome Applications,” *Journal of Classification*, vol. in press, 2014.

[45] C. E. Priebe, “Automatic Stream Characterization via Joint Exploitation of Content and Externals: Anomaly detection in a time series of edge-attributed graphs,” vol. October, no. 1, pp. 1–4, 2009.

[46] J. Grothendieck, C. E. Priebe, A. L. Gorin, Y. Park, D. J. Marchette, and J. C. M. Conroy, “Statistical Inference on Attributed Random Graphs: Fusion of Graph Features and Content: An Experiment on Time Series of Enron Graphs,” *Computational Statistics & Data Analysis*, vol. 54, pp. 1766–1776, July 2010.

[47] M. Tang, Y. Park, N. H. Lee, and C. E. Priebe, “Attribute fusion in a latent process model for time series of graphs,” *2011 IEEE Statistical Signal Processing Workshop SSP*, pp. 513–516, 2011.

[48] N. H. Lee, C. E. Priebe, and M. Tang, *An implied latent position process for doubly stochastic messaging activities*. Annual International Conference on Computational Mathematics, Computational Geometry & Statistics ({CMCGS} 2012). January, 2012.

[49] N. H. A. M. H. Lee, M. Tang, J. Yoder, and C. E. Priebe, “On latent position inference from doubly stochastic messaging activities,” pp. 1–25, May 2012.

[50] Y. Park, C. E. Priebe, and A. Youssef, “Anomaly Detection in Time Series of Graphs using Fusion of Graph Invariants,” *Submitted for publication*, pp. 1–20, Oct. 2012.

[51] L. F. Robinson and C. E. Priebe, “Detecting Time-dependent Structure in Network Data via a New Class of Latent Process Models,” *arXiv*, vol. 1212.3587v.

[52] N. H. Lee, C. E. Priebe, R. Tang, and M. Rosen, “Using non-negative factorization of time series of graphs for learning from an event-actor network,” *Under review*, Dec. 2013.

[53] H. Wang, M. Tang, Y. Park, and C. E. Priebe, “Locality statistics for anomaly detection in time series of graphs,” *arXiv preprint*, 2013.

[54] N. H. Lee, I.-J. Wang, R. Tang, M. Rosen, and C. E. Priebe, “A rank estimation criterion using an NMF algorithm under an inner dimension condition,” June 2014.

[55] J. T. Vogelstein, W. R. Gray, J. L. Prince, L. Ferrucci, S. M. Resnick, C. E. Priebe, and R. J. Vogelstein, “Graph-Theoretical Methods for Statistical Inference on MR Connectome Data,” *Organization Human Brain Mapping*, 2010.

[56] C. E. Priebe, J. T. Vogelstein, and D. D. Bock, “Optimizing the quantity/quality trade-off in connectome inference,” *Communications in Statistics Theory and Methods*, p. 7, 2013.

[57] D. Mhembere, S. Ryman, D. L. Sussman, R. E. Jung, J. T. Vogelstein, R. J. Vogelstein, C. E. Priebe, and R. Burns, “Multivariate Invariants from Massive Brain-Graphs,” in *Organization of Human Brain Mapping*, 2013.

[58] D. Mhembere, W. G. Roncal, D. L. Sussman, C. E. Priebe, R. E. Jung, S. Ryman, R. J. Vogelstein, J. T. Vogelstein, and R. Burns, “Computing Scalable Multivariate Glocal Invariants of Large (Brain-) Graphs,” in *Global Conference on Signal and Information Processing*, 2013.

[59] N. Sismanis, D. L. Sussman, J. T. Vogelstein, W. Gray Roncal, R. J. Vogelstein, E. Perlman, D. Mhembere, S. Ryman, R. E. Jung, R. Burns, C. E. Priebe, N. Pitsianis, and X. Sun, “Extracting Proximity for Brain Graph Voxel Classification,” in *5th Panhellenic Conference on Biomedical Technology*, 2013.

[60] J. T. Vogelstein, W. Gray Roncal, R. J. Vogelstein, and C. E. Priebe, “Graph Classification using Signal Subgraphs: Applications in Statistical Connectomics,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1539–1551, 2013.

[61] N. H. Lee, I.-J. Wang, Y. Park, C. E. Priebe, and M. Rosen, “Automatic Dimension Selection for a Non-negative Factorization Approach to Clustering Multiple Random Graphs,” *Under review*, June 2014.

[62] M. Tang, A. Athreya, D. L. Sussman, V. Lyzinski, and C. E. Priebe, “A nonparametric two-sample hypothesis testing problem for random dot product graphs,” Sept. 2014.

[63] M. Tang, A. Athreya, D. L. Sussman, V. Lyzinski, and C. E. Priebe, “A semiparametric two-sample hypothesis testing problem for random dot product graphs.” 2014.

[64] W. Gray Roncal, J. A. Bogovic, J. T. Vogelstein, B. A. Landman, J. L. Prince, and R. J. Vogelstein, “Magnetic resonance connectome automated pipeline: an overview.,” *IEEE pulse*, vol. 3, pp. 42–48, Mar. 2010.

[65] W. Gray Roncal, Z. H. Koterba, D. Mhembere, D. M. Kleissas, J. T. Vogelstein, R. Burns, A. R. Bowles, D. K. Donavos, S. Ryman, R. E. Jung, L. Wu, V. D. Calhoun, and R. J. Vogelstein, “MIGRAINE: MRI Graph Reliability Analysis and Inference for Connectomics,” *Global Conference on Signal and Information Processing*, 2013.

[66] W. Gray Roncal, D. M. Kleissas, G. D. Hager, C. E. Priebe, J. T. Vogelstein, R. J. Vogelstein, and R. Burns, “Images to Graphs: Quantification and Optimization of Connectomics Graph Error,” *in preparation*, 2014.

[67] D. E. Rex, J. Q. Ma, and A. W. Toga, “The LONI Pipeline Processing Environment,” *NeuroImage*, vol. 19, pp. 1033–1048, July 2003.

[68] S. Sikka, J. T. Vogelstein, and M. P. Milham, “Towards Automated Analysis of Connectomes: The Configurable Pipeline for the Analysis of Connectomes (C-PAC),” in *Organization of Human Brain Mapping*, Neuroinformatics, 2012.

[69] E. A. Pnevmatikakis, T. A. Machado, L. Grosenick, B. Poole, J. T. Vogelstein, and L. Paninski, “Rank-penalized nonnegative spatiotemporal deconvolution and demixing of calcium imaging data,” in *Computational and Systems Neuroscience Meeting*, 2013.

[70] E. A. Pnevmatikakis, Y. Gao, D. Soudry, D. Pfau, C. Lacefield, K. Poskanzer, R. Bruno, R. Yuste, and L. Paninski, “A structured matrix factorization framework for large scale calcium imaging data analysis,” Sept. 2014.

[71] B. Haeffele, E. Young, and R. Vidal, “Structured Low-Rank Matrix Factorization: Optimality, Algorithm, and Applications to Image Processing,” in *Proceedings of The 31st International Conference on Machine Learning*, pp. 2007–2015, 2014.

[72] J. W. Bohland, C. Wu, H. Barbas, H. Bokil, M. Bota, H. C. Breiter, H. T. Cline, J. C. Doyle, P. J. Freed, R. J. Greenspan, S. N. Haber, M. J. Hawrylycz, D. G. Herrera, C. C. Hilgetag, Z. J. Huang, A. Jones, E. G. Jones, H. J. Karten, D. Kleinfeld, R. KÃtter, H. A. Lester, J. M. Lin, B. D. Mensh, S. Mikula, J. Panksepp, J. L. Price, J. Safdieh, C. B. Saper, N. D. Schiff, J. D. Schmahmann, B. W. Stillman, K. Svoboda, L. W. Swanson, A. W. Toga, D. C. Van Essen, J. D. Watson, P. P. Mitra, B. Hermant, R. Kotter, and D. Essen, “A Proposal for a Coordinated Effort for the Determination of Brainwide Neuroanatomical Connectivity in Model Organisms at a Mesoscopic Scale,” *PLoS Comput Biol*, vol. 5, no. 3, p. 0901.4598, 2009.

[73] J. Ramsay and B. W. Silverman, *Functional Data Analysis*. Springer, 2005.

[74] M. W. Trosset, C. E. Priebe, Y. Park, and M. I. Miller, “Semisupervised learning from dissimilarity data.,” *Computational statistics data analysis*, vol. 52, pp. 4643–4657, June 2008.

[75] J. E. Gonzalez, Y. Low, H. Gu, D. Bickson, and C. Guestrin, "Powergraph: Distributed graph-parallel computation on natural graphs," in *Proceedings of the 10th USENIX Conference on Operating Systems Design and Implementation*, pp. 17–30, 2012.

[76] G. Malewicz, M. H. Austern, A. J. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski, "Pregel: A system for large-scale graph processing," in *SIGMOD*, pp. 135–146, 2010.

[77] D. Zheng, D. Mhembere, R. Burns, and A. S. Szalay, "FlashGraph: Processing Billion-Node Graphs on an Array of Commodity SSDs," Aug. 2014.

[78] D. J. Abadi, S. R. Madden, and N. Hachem, "Column-stores vs. row-stores: how different are they really?," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 967–980, ACM, 2008.

[79] R. Burns, K. Lillaney, D. Berger, K. Deisseroth, M. Kazhdan, A. S. Szalay, W. Gray Roncal, P. Manavalan, D. D. Bock, L. Grosenick, J. W. Lichtman, J. T. Vogelstein, D. M. Kleissas, E. Perlman, K. Chung, N. Kasthuri, R. C. Reid, and R. J. Vogelstein, "The Open Connectome Project Data Cluster: Scalable Analysis and Vision for High-Throughput Neuroscience," in *Scientific and Statistical Database Management*, 2013.

[80] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2011.

[81] J. T. Vogelstein, A. M. Packer, T. A. Machado, T. Sippy, B. Babadi, R. Yuste, and L. Paninski, "Fast non-negative deconvolution for spike train inference from population calcium imaging.," *Journal of neurophysiology*, vol. 104, p. 22, Dec. 2010.

[82] J. T. Vogelstein, B. O. Watson, A. M. Packer, R. Yuste, B. M. Jedynak, and L. Paninski, "Spike inference from calcium imaging using sequential Monte Carlo methods.," *Biophysical Journal*, vol. 97, pp. 636–655, July 2009.

[83] J. T. Vogelstein, A. M. Packer, R. Yuste, and L. Paninski, "Towards inferring neural circuits from population calcium imaging," *Frontiers in Systems Neuroscience. Conference Abstract: Computational and systems neuroscience*, 2009.

[84] B. C. Lucas, J. a. Bogovic, A. Carass, P.-L. Bazin, J. L. Prince, D. L. Pham, and B. a. Landman, "The Java Image Science Toolkit (JIST) for rapid prototyping and publishing of neuroimaging software.," *Neuroinformatics*, vol. 8, pp. 5–17, Mar. 2010.

[85] R. W. Cox, "AFNI: software for analysis and visualization of functional magnetic resonance neuroimages.," *Computers and biomedical research, an international journal*, vol. 29, pp. 162–173, June 1996.

[86] M. Jenkinson, C. F. Beckmann, T. E. J. Behrens, M. W. Woolrich, and S. M. Smith, "FSL," 2012.

[87] S. N. Sotiropoulos, S. Jbabdi, J. Xu, J. L. Andersson, S. Moeller, E. J. Auerbach, M. F. Glasser, M. Hernandez, G. Sapiro, M. Jenkinson, D. A. Feinberg, E. Yacoub, C. Lenglet, D. C. Van Essen, K. Ugurbil, and T. E. J. Behrens, "Advances in diffusion MRI acquisition and processing in the Human Connectome Project.," *NeuroImage*, vol. null, May 2013.

[88] D. W. Shattuck and R. M. Leahy, "BrainSuite: An automated cortical surface identification tool," *Medical Image Analysis*, vol. 6, pp. 129–142, June 2002.

[89] A. Daducci, S. Gerhard, A. Griffa, A. Lemkaddem, L. Cammoun, X. Gigandet, R. Meuli, P. Hagmann, and J.-P. Thiran, "The connectome mapper: an open-source processing pipeline to map connectomes with MRI.," *PloS one*, vol. 7, p. e48121, Jan. 2012.

[90] Z. Cui, S. Zhong, P. Xu, Y. He, and G. Gong, "PANDA: a pipeline toolbox for analyzing brain diffusion images.," *Frontiers in human neuroscience*, vol. 7, p. 42, Jan. 2013.

[91] K. Gorgolewski, C. D. Burns, C. Madison, D. Clark, Y. O. Halchenko, M. L. Waskom, and S. S. Ghosh, "Nipype: a flexible, lightweight and extensible neuroimaging data processing framework in python.," *Frontiers in neuroinformatics*, vol. 5, p. 13, Jan. 2011.

[92] B. B. Avants, N. J. Tustison, G. Song, P. A. Cook, A. Klein, and J. C. Gee, "A reproducible evaluation of ANTs similarity metric performance in brain image registration," *NeuroImage*, vol. 54, no. 3, pp. 2033–2044, 2011.

[93] R. S. Desikan, F. Ségonne, B. Fischl, B. T. Quinn, B. C. Dickerson, D. Blacker, R. L. Buckner, A. M. Dale, R. P. Maguire, B. T. Hyman, M. S. Albert, and R. J. Killiany, "An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest.," *NeuroImage*, vol. 31, pp. 968–980, July 2006.

[94] J. Glaun, A. Qiu, M. I. Miller, L. Younes, and J. Glaunès, "Large Deformation Diffeomorphic Metric Curve Mapping.," *International journal of computer vision*, vol. 80, pp. 317–336, Dec. 2008.

[95] M. Aggarwal, W. Duan, Z. Hou, N. Rakesh, Q. Peng, C. A. Ross, M. I. Miller, S. Mori, and J. Zhang, "Spatiotemporal mapping of brain atrophy in mouse models of Huntington's disease using longitudinal in vivo magnetic resonance imaging.," *NeuroImage*, vol. 60, pp. 2086–2095, May 2012.

[96] M. I. Miller, C. E. Priebe, A. Qiu, B. Fischl, A. Kolasny, T. Brown, Y. Park, J. T. Ratnanather, E. Busa, J. Jovicich, P. Yu, B. C. Dickerson, and R. L. Buckner, "Collaborative computational anatomy: an MRI morphometry study of the human brain via diffeomorphic metric mapping.," *Human Brain Mapping*, vol. 30, no. 7, pp. 2132–2141, 2009.

[97] J. G. Csernansky, L. Wang, S. C. Joshi, J. T. Ratnanather, and M. I. Miller, "Computational anatomy and neuropsychiatric disease: probabilistic assessment of variation and statistical inference of group difference, hemispheric asymmetry, and time-dependent change.," *NeuroImage*, vol. 23 Suppl 1, pp. S56—-68, Jan. 2004.

[98] J. T. Vogelstein, J. C. M. Conroy, L. J. Podrazik, S. G. Kratzer, D. E. Fishkind, R. J. Vogelstein, and C. E. Priebe, "(Brain) Graph Matching via Fast Approximate Quadratic Programming," *Under review*, 2014.

[99] M. Fiori, P. Sprechmann, J. T. Vogelstein, P. Muse, and G. Sapiro, "Robust Multimodal Graph Matching: Sparse Coding Meets Graph Matching," in *Neural Information Processing Systems (Spotlight Presentation)*, 2013.

[100] D. A. Spielman and S.-H. Teng, "Spectral sparsification of graphs," *SIAM Journal on Computing*, vol. 40, no. 4, pp. 981–1025, 2011.

[101] C. M. Stein, "Inadmissibility of the usual estimator of the mean of a multivariate normal distribution," in *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 197–206, University of California Press, 1956.

[102] S. Gutmann, "Stein's Paradox is Impossible in Problems with Finite Sample Space," *The Annals of Statistics*, vol. 10, pp. 1017–1020, Sept. 1982.

[103] A. Di Martino, D. A. Fair, C. Kelly, T. D. Satterthwaite, F. X. Castellanos, M. E. Thomason, R. C. Craddock, B. Luna, B. L. Leventhal, X.-N. Zuo, and M. P. Milham, "Unraveling the Miswired Connectome: A Developmental Perspective," *Neuron*, vol. 83, pp. 1335–1353, Sept. 2014.

[104] C. Jiang, F. Coenen, and M. Zito, "A survey of frequent subgraph mining algorithms," *The Knowledge Engineering . . .* , vol. 00, pp. 1–31, 2013.

[105] V. Mountcastle, "Modality and topographic properties of single neurons of cat's somatic sensory cortex.," *Journal of Neurophysiology*, vol. 20, no. 4, pp. 408–434, 1957.

[106] J. C. Horton and D. L. Adams, "The cortical column: a structrue without a function," *Philosophical transactions of the royal society B*, vol. 360, pp. 837–862, Apr. 2005.

[107] J. T. Vogelstein, R. J. Vogelstein, and C. E. Priebe, “Are mental properties supervenient on brain properties?,” *Scientific Reports*, vol. 1, no. 100, p. 11, 2009.

[108] R. Burns, W. G. Roncal, D. Kleissas, K. Lillaney, P. Manavalan, E. Perlman, D. Berger, D. D. Bock, K. Chung, K. Deisseroth, L. Grosenick, N. Kasthuri, M. Kazhdan, J. Lichtman, R. C. Reid, A. S. Szalay, J. T. Vogelstein, and R. J. Vogelstein, “The Open Connectome Project data cluster: Scalable analysis and vision for high-throughput neuroscience,” in *Scientific and Statistical Databases Management Conference*, 2013.

[109] Y. L. Et al., “A Public Turbulence Database Cluster and Applications to Study {Lagrangian} Evolution of Velocity Increments in Turbulence,” *Journal of Turbulence*, vol. 9, no. 31, pp. 1–29, 2008.

[110] K. Kanov, E. Perlman, R. Burns, Y. Ahmad, and A. S. Szalay, “I/O Streaming Evaluation of Batch Queries for Data-Intensive Computational Turbulence,” in *Supercomputing*, 2011.

[111] G. Eyink, E. Vishniac, C. Lalescu, H. Aluie, K. Kanov, K. Bürger, R. Burns, C. Meneveau, and A. Szalay, “Flux-freezing breakdown in high-conductivity magnetohydrodynamic turbulence,” *Nature*, vol. 497, no. 7450, pp. 466–469, 2013.

[112] N. Borges, G. A. Coppersmith, G. G. L. Meyer, and C. E. Priebe, “Anomaly detection for random graphs using distributions of vertex invariants,” in *2011 45th Annual Conference on Information Sciences and Systems*, pp. 1–6, IEEE, Mar. 2011.

[113] C. E. Priebe and D. J. Marchette, “Information Fusion: Inference from Graphs & Feature Matrices,” in *Information Fusion*, 2009.

[114] N. H. Lee, T. S. T. Leung, and C. E. Priebe, *Random Graphs Based on Self-Exciting Messaging Activities*. International Conference on Business Intelligence and Financial Engineering, December, 2011.

[115] D. L. Sussman, D. Mhembere, S. Ryman, R. E. Jung, R. J. Vogelstein, R. Burns, J. T. Vogelstein, and C. E. Priebe, “Massive Diffusion MRI Graph Structure Preserves Spatial Information,” in *Organization of Human Brain Mapping*, 2013.

[116] J. L. Solka, B. T. Clark, and C. E. Priebe, “A Visualization Framework for the Analysis of Hyperdimensional Data,” *International Journal of Image and Graphics*, vol. 2, no. 1, pp. 145–161, 2002.

[117] R. S. Pilla, P. Tao, and C. E. Priebe, “Adaptive Methods for Spatial Scan Analysis via Semiparametric Mixture Models,” *Journal of Computational and Graphical Statistics*, vol. 12, no. 2, pp. 332–353, 2003.

[118] W. L. Poston, E. J. Wegman, C. E. Priebe, and J. L. Solka, “A Deterministic Method for Robust Estimation of Multivariate Location and Shape,” *Journal of Computational and Graphical Statistics*, vol. 6, no. 3, pp. 300–313, 1997.

[119] D. Q. Naiman and C. E. Priebe, “Computing Scan Statistic p Values Using Importance Sampling, With Applications to Genetics and Medical Image Analysis,” *Journal Of Computational And Graphical Statistics*, vol. 10, no. 2, pp. 296–328, 2001.

[120] E. Ceyhan, C. E. Priebe, and J. C. Wierman, “Relative Density of the Random r-Factor Proximity Catch Digraph for Testing Spatial Patterns of Segregation and Association (Technical Report),” *Computational Statistics and Data Analysis*, vol. 50, no. 8, pp. 1925–1964, 2006.

[121] L. A. Abrams, D. E. Fishkind, and C. E. Priebe, “A proof of the spherical homeomorphism conjecture for surfaces.,” 2002.

[122] D. J. Marchette and C. E. Priebe, “Predicting Unobserved Links in Incompletely Observed Networks,” *Computational Statistics and Data Analysis*, vol. 52, pp. 1373–1386, Jan. 2008.

[123] C. E. Priebe, J. L. Solka, D. J. Marchette, and B. T. Clark, “Class Cover Catch Digraphs for Latent Class Discovery in Gene Expression Monitoring by DNA Microarrays,” *Computational Statistics and Data Analysis*, vol. 43, no. 4, pp. 621–632, 2003.