

LONI PIPELINE DEMO DAY

Software Training Event and
Ice Cream Social

Friday October 2nd, 2015
9:00 A.M. - 12:00 P.M.

BIG DATA
ANALYSIS using the LONI Pipeline

Sponsored by LONIR P41EB015922 and BDDS U54EB020406

Rosen Family Screening Theatre, TCC 227
Tutor Campus Center, University Park Campus



Advanced Neuroimaging and Informatics and Genomics Computing

<http://pipeline.loni.usc.edu>

USC Laboratory of Neuro Imaging



Welcome to LONI Pipeline Demo Day!

Big Data Analysis using the LONI Pipeline

Greetings and thank you for attending this unique event for the USC research community! We hope that you will find it informative and helpful for getting a greater sense of how you might utilize emerging scientific workflow design and execution tools – being developed here at USC – to automate and streamline your big data analytic needs.

Modern science is comprised of ever larger sets of data. Indeed, a typical scientist is likely to be collecting more data now than scientists ever did only a decade ago. To extract useful information from these data sets, efficient analytic tools are needed as are the means to link processing steps into comprehensive, end-to-end workflows. Yet, the algorithmic resources for processing data are often diverse and constantly evolving. What can one do to optimize their data processing and more rapidly obtain results?

To provide an extensible framework for interoperability of such resources, we introduce the LONI Pipeline* – a 6th generation workflow design and execution environment. Developed by USC-based computer scientists and engineers, LONI Pipeline utilizes a graphical user interface driven means for linking data processing “modules” into complete workflows. LONI Pipeline employs a decentralized compute infrastructure model, where tools, services, and data are linked via an external inter-resource-mediating layer. Several servers around the USC campus and the world now support LONI Pipeline processing. Many of these are FREELY available for your use. If you would like to have LONI Pipeline running on a CPU cluster in your laboratory, setting up the LONI Pipeline server is straightforward and provides a range of useful job and user management tools.

To integrate existing data and computational tools within the LONI Pipeline environment, no modification to them is required. An XML resource (a “module”) description is all that is needed and includes all the important information about the resource’s location, the proper innovation protocol, run-time controls and acceptable data types. The usage descriptors are complemented by a variety of auxiliary information about the resource state, specifications, history, authorship, and bibliography. These can be created in just a few minutes by even the novice programmer and then used over and over again. As a result, LONI Pipeline is ideal for the sharing of data processing methods, the independent re-analysis of data, and provides for the long-term recording of data provenance. For complete details about LONI Pipeline, all of its specifications, usage, and documentation, please visit <http://pipeline.loni.usc.edu>.

In this workshop, entitled “Big Data Analysis using the LONI Pipeline”, we will showcase LONI Pipeline and present a series of brief summaries of how LONI Pipeline is being used by USC investigators on a range of big data problems. While many of the examples focus on neuroimaging and genetics as examples of large data, this by no means suggests that LONI Pipeline is limited to only these biomedical data types. In fact, LONI Pipeline is data type agnostic and is suitable for engineering, computer science, physics, information science, as well as any other discipline which requires automated methods for large-scale data analytics. We think that LONI Pipeline might be right for you!

Once again, we are delighted to have you take part and look forward to presenting applications of LONI Pipeline for big data analysis to you. Please feel free to ask questions and, once back in your lab, try out the guest-access LONI Pipeline web-start server. We look forward to your own contributions to the further use and development of LONI Pipeline!

Yours sincerely,
John Darrell Van Horn, Ph.D.
Arthur W. Toga, Ph.D.

*LONI Pipeline development is supported through NIH grants to the LONI Resource (P41EB015922) and the Big Data for Discovery Science project (U54EB020406) from the NIH’s Big Data to Knowledge (BD2K) program.

AGENDA

8:30 a.m.	Sign in	
9:00 a.m.	Jack Van Horn	Introduction
9:15 a.m.	Petros Petrosyan, Sam Hobel, and Ivo Dinov	Hippocampal Meta-Analysis Workflow
9:45 a.m.	Dogu Baran Aydogan	On the convergence of human connectome graph properties
10:00 a.m.	Junning Li	Rotational Induced Riemannian Manifolds for Neural Fiber Orientation Distributions
10:15 a.m.	Sumiko Abe	Surface generation for visualizing large-scale brain imaging data sets by using LONI pipeline
10:30 a.m.	Carinna Torgerson	Off the map: using LONI Pipeline to process and analyze brain regions that do not appear in automated label maps
10:45 a.m.	Andrei Irimia	Analysis of connectomic changes due to traumatic brain injury using the LONI Pipeline
11:00 a.m.	Anjani Bhargavi	Computing Brain Connectivity and Shape through the LONI Pipeline
11:15 a.m.	Joshua Faskowitz and Faisal Rashid	The LONI Pipeline: Solving challenges to Big Data collection through the use of a high-throughput computational engine
11:30 a.m.	Marc Harrison	Computing Multi-Resolution Structural Connectomes with the LONI Pipeline
11:45 a.m.	Ice Cream Social	

BIG DATA ANALYSIS using the LONI Pipeline
Friday October 2nd, 2015
9:00 A.M. - 12:00 P.M.

Rosen Family Screening Room, TCC 227
Tutor Center, University Park Campus

TABLE OF CONTENTS

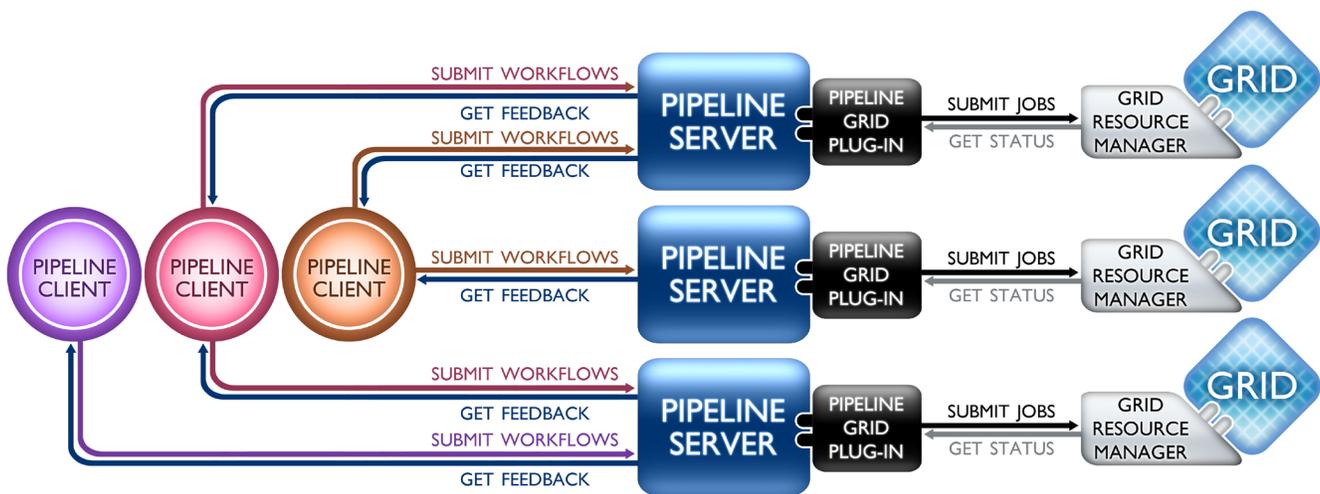
Introduction	4
Hippocampal Meta-Analysis Workflow	8
On the convergence of human connectome graph properties	9
Rotational Induced Riemannian Manifolds for Neural Fiber Orientation Distributions	10
Surface generation for visualizing large-scale brain imaging data sets by using LONI pipeline	11
Off the map: using LONI Pipeline to process and analyze brain regions that do not appear in automated label maps	12
Analysis of connectomic changes due to traumatic brain injury using the LONI Pipeline	13
Computing Brain Connectivity and Shape through the LONI Pipeline	14
The LONI Pipeline: Solving challenges to Big Data collection through the use of a high-throughput computational engine	15
Computing Multi-Resolution Structural Connectomes with the LONI Pipeline	16
USC Campus Maps	17

PRESENTER EMAILS

Sumiko Abe	sumi888@gmail.com
Dogu Aydogan	baran.aydogan@loni.usc.edu
Ivo D Dinov	dinov@umich.edu
Joshua Faskowitz	Joshua.Faskowitz@loni.usc.edu
Marc Harrison	Marc.Harrison@loni.usc.edu
Samuel Hobel	Samuel.Hobel@loni.usc.edu
Andrei Irimia	andrei.irimia@gmail.com
Junning Li	Junning.Li@loni.usc.edu
Petros Petrosyan	Petros.Petrosyan@loni.usc.edu
Anjanibhargavi Ragothaman	Anjanibhargavi.Ragothaman@loni.usc.edu
Faisal Rashid	Faisal.Rashid@loni.usc.edu
Carinna Torgerson	carinnat@gmail.com

Why you should use LONI Pipeline

The LONI Pipeline environment (<http://pipeline.loni.usc.edu>) is a distributed infrastructure model for mediating communications between different data resources, software tools and web-services. No software redesign or rebuilding modifications of the existing resources are necessary for their integration with other computational components within the Pipeline. The Pipeline Environment employs eXtensible Markup Language (XML) schema to enable the inter-resource communication, data transfers and execution instructions. Each XML resource description contains important information about the tools location, the proper invocation protocol (i.e., input/output types, parameter specifications, etc.), run-time controls and data-types. The Pipeline XML schema (<http://pipeline.loni.usc.edu/support/xml-overview/>) also includes auxiliary metadata about the resource state, specifications, history, authorship, licensing, and bibliography. Using this resource metadata, the Pipeline infrastructure facilitates the integration of disparate resources and provides a complete and comprehensive protocol provenance for the data, tools, hardware and results. Individual module descriptions and entire protocol XML objects are managed as .PIPE files, facilitate the broad dissemination of resource metadata descriptions via web services, and promote constructive

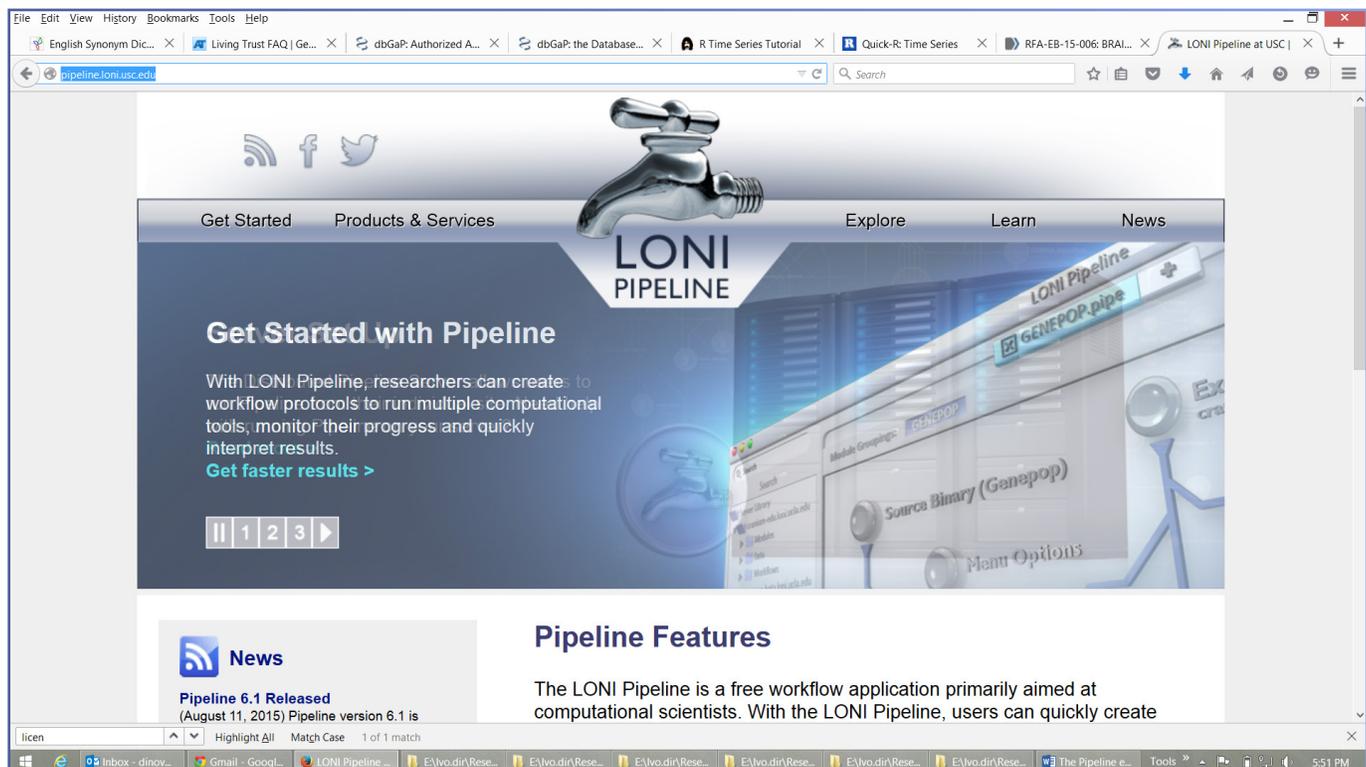


High-level schematic representation of the communication between multiple local Pipeline clients connected to multiple remote Pipeline servers.

LONI Pipeline Architecture

utilization of multidisciplinary tools and expertise by professionals, novice users and trainees. The Pipeline software architecture design is domain and hardware independent which makes the environment useful in different computational disciplines and on diverse hardware infrastructures. The Pipeline environment may be utilized in three synergistic mechanisms. The first one ([http:// pipeline.loni.usc.edu /downloads](http://pipeline.loni.usc.edu/downloads)) involves the local use of the Pipeline client via connection to a remote server running natively on a hardware system which includes all appropriate plug-ins for system-specific grid managers, file systems, network, and communication protocols. The second type of Pipeline server distribution relies on virtualization technology. The virtualized Pipeline infrastructure provides end-users with the latest stable pre-compiled environment including all pre-installed open-source informatics tools. The resulting Pipeline Virtual Environment (PNVE, <http://pipeline.loni.usc.edu/PNVE>), contains the complete self-contained execution environment that can be run locally or on remote grid computing environment. The third Pipeline distribution mechanism is called Distributed Pipeline Server (DPS, <http://pipeline.loni.usc.edu/DPS>). This distribution includes a user-friendly graphical user interface (GUI) for automated native system configuration, installation and deployment of the Pipeline server, the available XML computational library, and back-end software tools. The Pipeline environment uses a clientserver architecture, but each Pipeline client may also act as

a local server and manage job submission and execution. Following proper authentication, the process of a client submitting a workflow for execution to a specified server prompts the server to translate (break) the workflow into parallel jobs and send them to the grid resource manager which in turn farms these to the back-end grid (or multiple cores). When a job is complete, the server retrieves the results from the grid resource manager and sends out subsequent jobs from the active workflow. The client receives status updates from the server at regular intervals. Currently, the Pipeline server supports Distributed Resource Management Application API (DRMAA, www.DRMAA.org) interface and Java GridEngine Database Interface (JGDI) to communicate to the grid resource manager. These include many of the popular grid resource managers, including Sun/Oracle Grid Engine (http://en.wikipedia.org/wiki/Oracle_Grid_Engine), GridWay (www.gridway.org), PBS/Torque (www.ClusterResources.com). The Pipeline Environment has been employed



Website: <http://pipeline.loni.usc.edu>

in a wide range of imaging, brain mapping, neuroscientific and genomics computing applications.

Project home pages

- LONI: <http://pipeline.loni.usc.edu>
- NITRC: <http://www.nitrc.org/projects/pipeline>
- BIRN: <http://www.birncommunity.org/tools-catalog/loni-pipeline>
- Bioinformatics.org: <http://www.bioinformatics.org/pipeline>
- Try the LONI Pipeline Informatics and Genomics Workflows online without any software installation using anonymous guest account: <http://pipeline.loni.usc.edu/PWS>

Operating system(s)

Pipeline clients and servers are platform-independent, while some features (e.g. privilege escalation, failover) require the server run on Linux/UNIX OS. The Distributed Pipeline Server (DPS) graphical user interface, which installs the Pipeline server, Grid Engine, and computational imaging and informatics software tools, require standard Linux OS kernels. The Pipeline Web Start (PWS) allows users to start the Pipeline application directly from the web browser and run it locally without any installation. It has all the features

and functionality of the down loadable stand-alone Pipeline application and allows anonymous guest access or user authentication to connect to remote Pipeline servers.

Pipeline Navigator

Explorer of available end-to-end biomedical computing solutions:

- <http://pipeline.loni.ucla.edu/services/library-navigator>

Training, Help and Support

- Training Events: <http://pipeline.loni.usc.edu/training/>
- Forum: <http://pipeline.loni.usc.edu/forum/>
- Documentation: <http://pipeline.loni.usc.edu/support/>
- Handbook (PDF): <http://pipeline.loni.usc.edu/support/handbook/>
- Video Tutorials: <http://pipeline.loni.usc.edu/support/screencasts/>

Programming language

Pure Java

Other requirements

- Requirements Summary: The Pipeline client and server can run on any system that is supported by Java Runtime Environment (JRE) 1.5 or higher. Windows Pipeline servers will not be able to use privilege escalation. Three-tier Failover feature is only supported by Unix/Linux systems. All other features are available for all platforms. Most Distributed Pipeline Servers require 300-1,000MB memory, which may depend on the load and garbage collection preferences.
- Distributed multicore deployment: the Distributed Pipeline Server (DPS) requires a Grid manager (e.g., Distributed Resource Management Application API, DRMAA), which is provided with the DPS distribution. The Pipeline server will still work on a platform without a Grid manager, however, jobs may not be processed in parallel and performance on multicore machines may be suboptimal.
- Complete requirements:
 - Client: <http://pipeline.loni.usc.edu/support/user-guide/installation/>
 - Server: <http://pipeline.loni.usc.edu/support/server-guide/installation/>
 - DPS: <http://pipeline.loni.usc.edu/DPS>
 - PWS: <http://pipeline.loni.usc.edu/PWS>

License

Apache-derived software license:

- <http://loni.usc.edu/Software/license.php>

Caution

There are some potential limitations of the Pipeline environment and its current collection of data, tools services and computational library (module XML meta-data descriptions):

- Each new informatics tool which needs to be accessible as a processing module within the Pipeline environment needs to be described manually by an expert using the Pipeline GUI or automatically using a properly configured XML exporter. Then the Pipeline XML module description can be shared with other users.
- To run available Pipeline workflows (* .pipe workflow files) on remote Pipeline-servers, users need to have accounts on the remote Pipeline servers. In addition, 2 types of updates may be necessary in the PIPE files -the server-name references of data sources (inputs), data sinks (results), and executables, as well as the path references to the data sources, sinks and executables. The server-name can be easily updated using server changer tool in Pipeline (Tools menu -> Server Changer). User has to edit path references on some or all of the data sources, sinks and executables for their server. No workflow modifications are necessary for executing these pipeline workflows on the LONI Pipeline Cranium server; however

this requires a LONI Pipeline useraccount (http://pipeline.loni.usc.edu/Collaboration/Pipeline/Pipeline_Application.jsp). A proper administrator configuration of the Distributed Pipeline Server (DPS, <http://pipeline.loni.usc.edu/DPS>) will resolve the need for such revisions by the user.

- Some computational tools may require wrapper scripts that call the raw executable binaries. These scripts (not the raw binaries) are then invoked via the Pipeline environment. Example situations include tools that have implicit outputs, or if the tools routinely return non-trivial exit codes, distinct from zero. Such problems may cause the Pipeline environment to halt execution of subsequent modules, because of a broken module-to-module communication protocol.
- Smartlines, which auto-convert between different informatics data formats, need to be extended to handle informatics and genomics data (currently, Smartlines handle mostly image file format conversions).
- Access to external informatics databases may need to be customized - e.g., PDB (<http://www.rcsb.org>), SCOP (<http://scop.mrc-lmb.cam.ac.uk/scop>), GenBank (www.ncbi.nlm.nih.gov/genbank), etc.
- Native vs. Virtual Pipeline server
- The fully Distributed Pipeline Server (DPS) architecture (which allows anyone to locally download, configure and deploy the complete Pipeline server) provides (natively) both the Pipeline middleware as well as installers for all computational tools available on the LONI Cranium Pipeline Grid Server (<http://pipeline.loni.usc.edu/learn/server-guide/>).

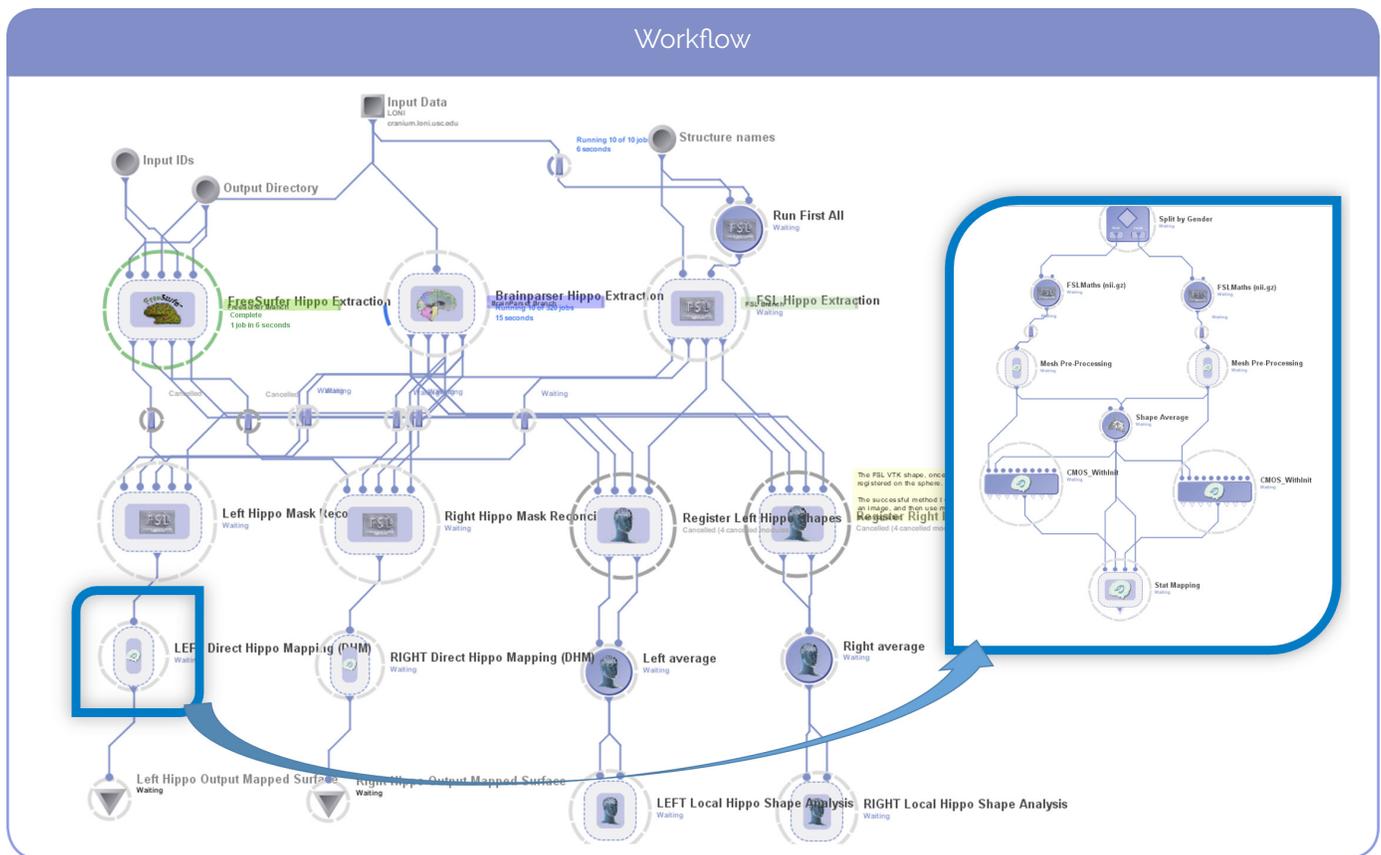
Restrictions for use by non-academics

Free for non-commercial research purposes.

HIPPOCAMPAL META-ANALYSIS WORKFLOW

9:15 a.m. - Pipeline Team

Petros Petrosyan, Sam Hobel, and Ivo Dinov



Step 1:

A list of input subjects is run through 3 different protocols (FreeSurfer's ReconAll, BrainParser, and FSL's RunFirstAll) to extract hippocampus 2D images and 3D shapes, then aligned/registered on the sphere to create an average image and shape for each subject.

Step 2:

The image results from the above step are divided into 2 groups using a metadata value and collectively run through the Direct Hippocampal Mapping protocol. In the DHM protocol, the images are turned into shapes and statistical differences are computed between each group and the average of both groups. The p-values are mapped onto the average shape.

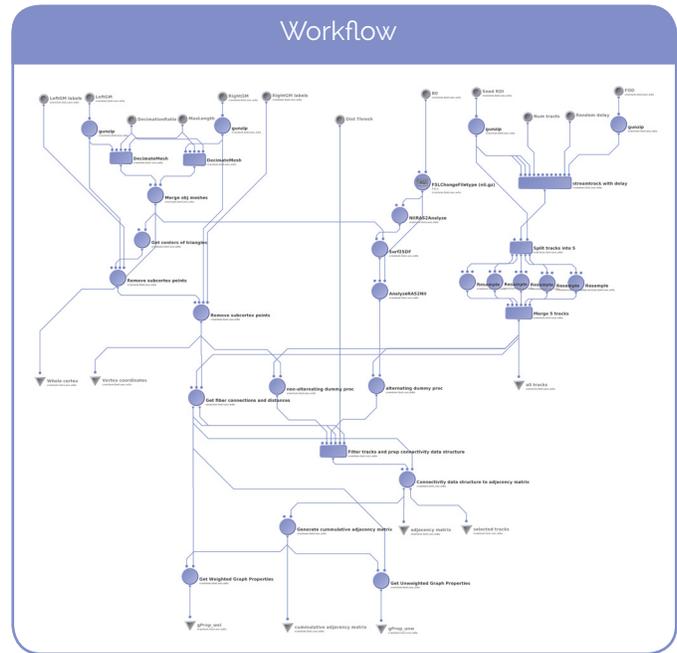
Step3:

A variation of the above step is run on the 3D shapes from step 1. The shape results from step 1 are divided (again by a metadata value) to create 2 groups. Radial distances and displacement features are computed for each group and mapped onto an average shape.

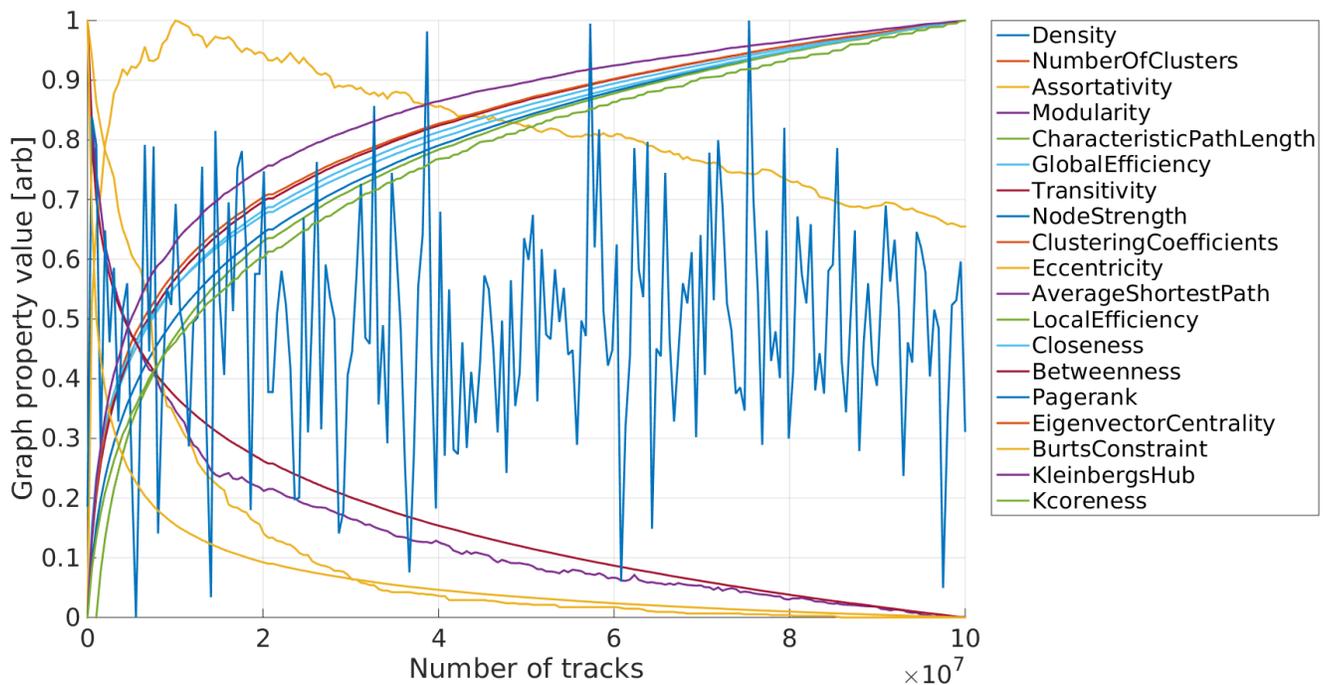
ON THE CONVERGENCE OF HUMAN CONNECTOME GRAPH PROPERTIES

9:45 a.m. - Dogu Baran Aydogan

Using graph properties to understand the structure and function of connectome has become a major research area. This work presents a critical assessment on how to generate and use connectomes. In particular, we are investigating the relationship between the number of brain regions and the number of tracks used to form a connectome. We use Pipeline to generate large scale connectomes, with vertex counts varying from 500 to 5000 and track counts from 500K to 100M. Our workflow computes more than 30 graph properties for each connectome and provides valuable information about their convergence. The results give us directions on how to generate reliable connectomes and which graph features we should be careful about.



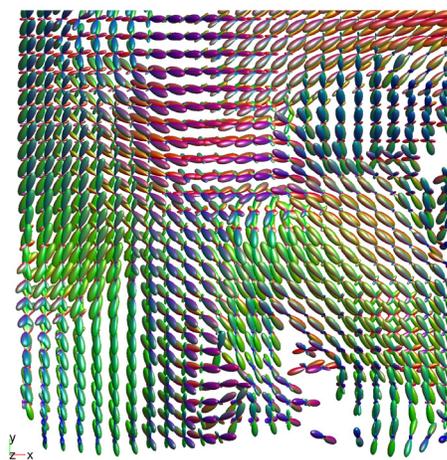
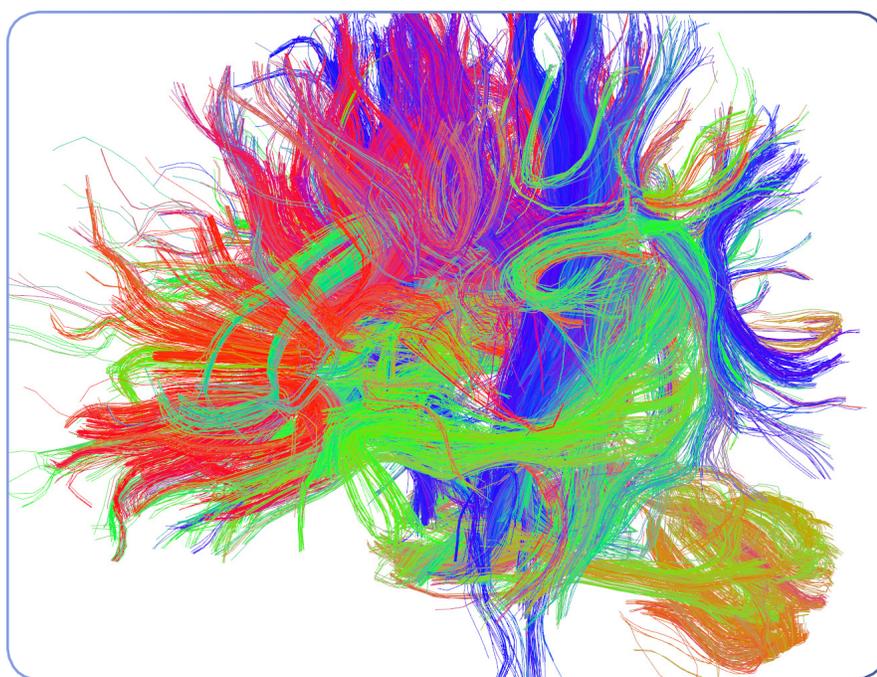
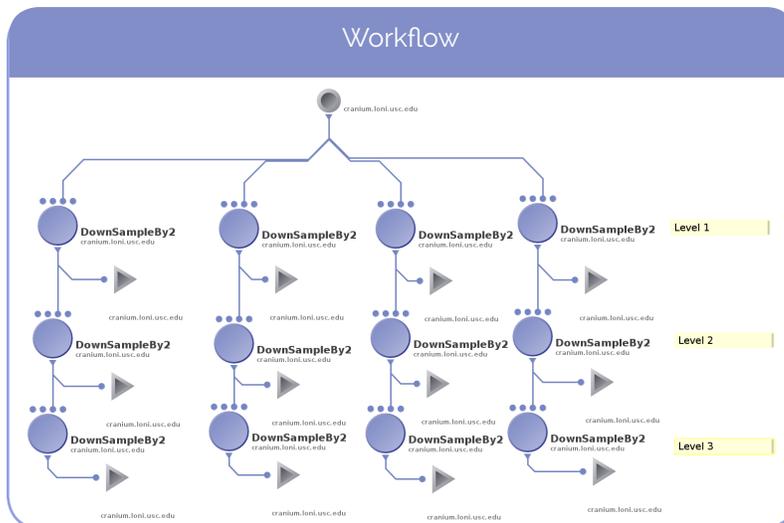
Scaled unweighted graph properties - 712 vertices



ROTATIONAL INDUCED RIEMANNIAN MANIFOLDS FOR NEURAL FIBER ORIENTATION DISTRIBUTIONS

10:00 a.m. - Junning Li

Interaction between voxel values is the driving force of many image processing methods. For example, in smoothing, neighboring voxels exert forces on the central one to make its value gradually coherent with theirs. By defining different driving forces, various effects can be achieved. In image registration, voxel-interaction forces between two images drive them to spatially deform and gradually become similar to each other. Processing fiber orientation distribution (FOD) images are challenging because each voxel is a spherical function whose mathematical properties are much more complicated than a simple intensity scalar. Even fundamental tasks such as interpolation, smoothing, segmentation and registration must be redesigned. To utilize well-developed image processing frameworks, it is crucial to develop suitable manifold structures for FODs. As neuronal fibers extend in the brain, they do not take straight paths, but turn gradually to make curves. We proposed a rotation-induced manifold to directly handle FODs' orientation difference. We show this Riemannian manifold can be used for smoothing, interpolation and building image-pyramids, yielding more accurate or intuitively more reasonable results than the linear or the unit hyper-sphere manifold.



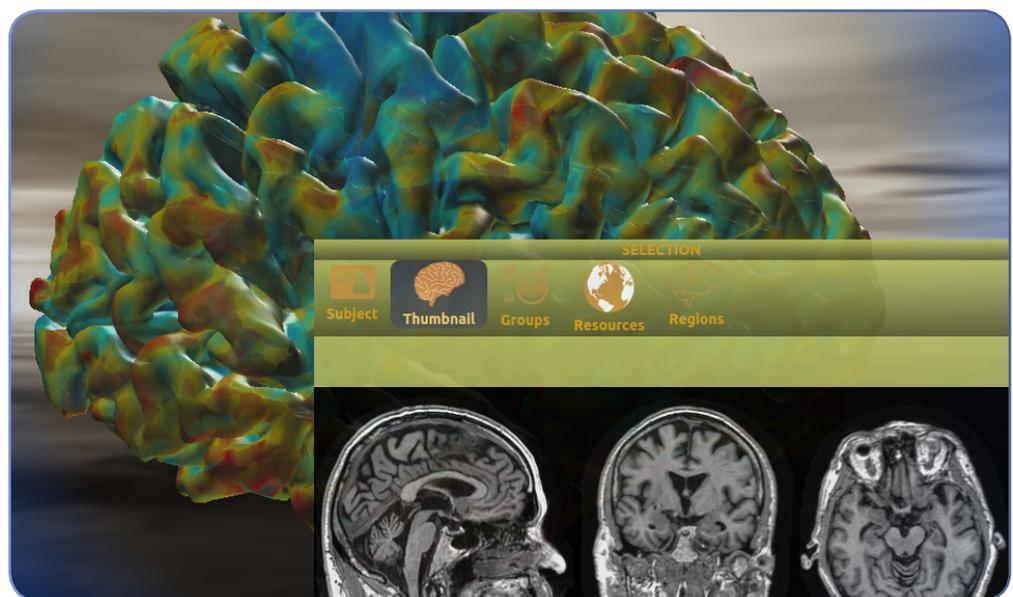
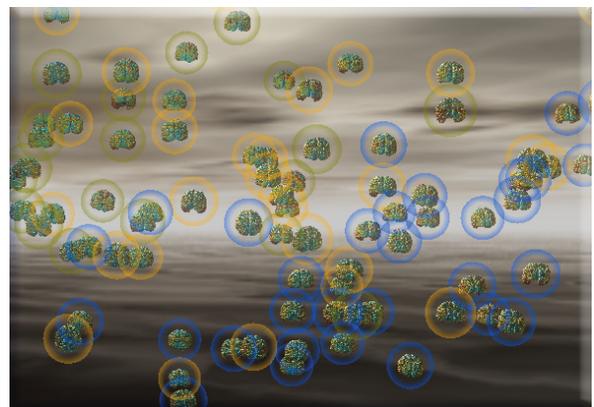
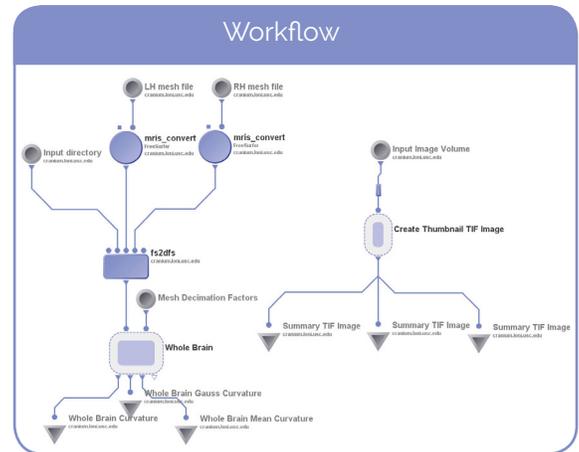
SURFACE GENERATION FOR VISUALIZING LARGE-SCALE BRAIN IMAGING DATA SETS BY USING LONI PIPELINE

10:15 a.m. - Sumiko Abe, Andrei Irimia, and Jack Van Horn

The “Informatics Visualization in Neuroimaging” (INVIZIAN) project seeks to introduce a brand new concept for exploring large brain imaging archives in 3D space. Using T1-weighted neuroimaging data, INVIZIAN provides a powerful graphic user interface to allow users to change the visualization effects such as lighting color, light positions, surface shininess, brain color maps, background sky box map, etc.

INVIZIAN relies on the LONI Pipeline to create files which can specify cortical thickness, curvature and other properties for each subject whose brain is visualized. Such information is generated by FreeSurfer software following automatic segmentation and morphometry, where after it is converted into the INVIZIAN-compatible DFS format as illustrated by the LONI workflow shown here. Machine learning methods can then be employed to identify new coordinate systems describing the differences between subjects.

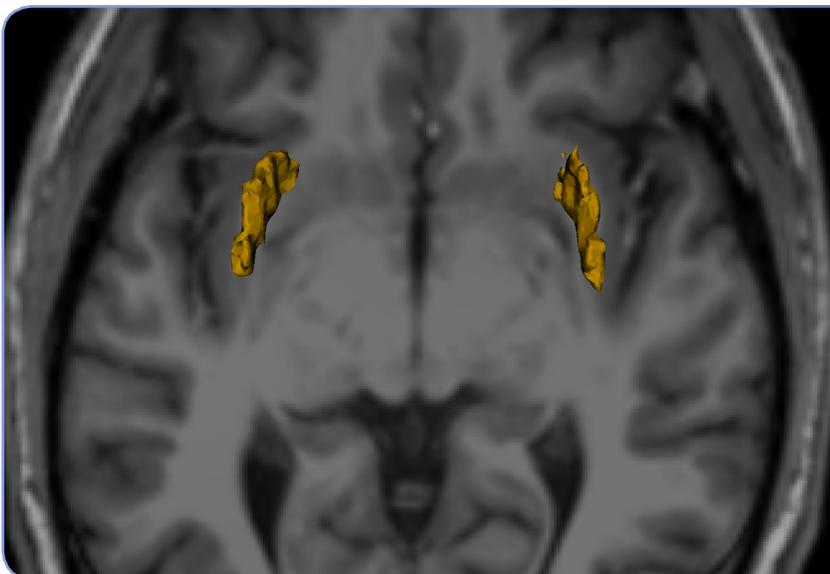
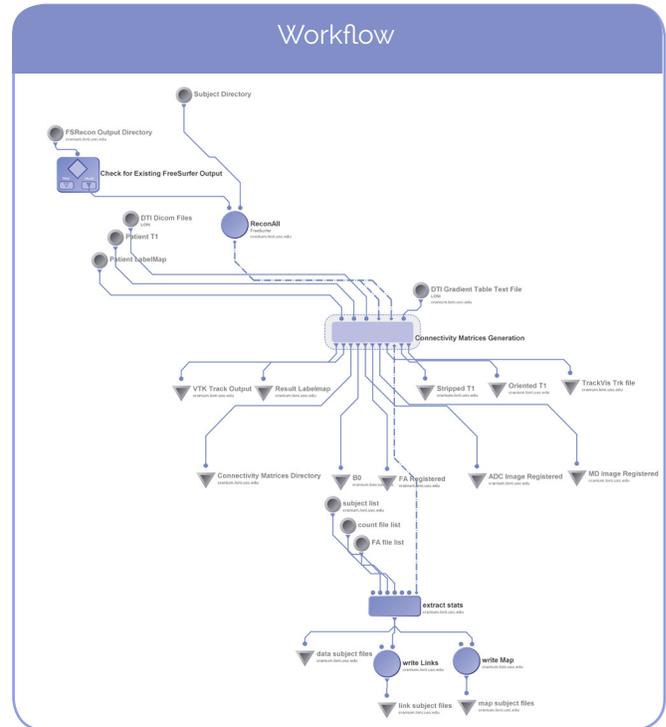
INVIZIAN then plots each brain surface in the accurate location in 3D space, centered and drawn at the machine learning derived coordinates. This is illustrated below. Users then can easily translate, rotate, zoom in on the cluster of brain surfaces and select individual cases. Right clicking any individual brain reveals subject-specific meta-data, image thumbnails, as well as helps to keep track of user-defined groups of brain surfaces (also illustrated). For established researchers as well as students, INVIZIAN provides the capabilities to identify novel or unanticipated differences between patient groups. LONI Pipeline can be used to craft any suitable arrangement of brain surface data – in a configuration best suited to highlighting differences between brains. All in all, LONI Pipeline is a critical element for enabling INVIZIAN to be a 21st century visualization environment for discovery neuroscience.



OFF THE MAP: USING LONI PIPELINE TO PROCESS AND ANALYZE BRAIN REGIONS THAT DO NOT APPEAR IN AUTOMATED LABEL MAPS

10:30 a.m. - Carinna Torgerson

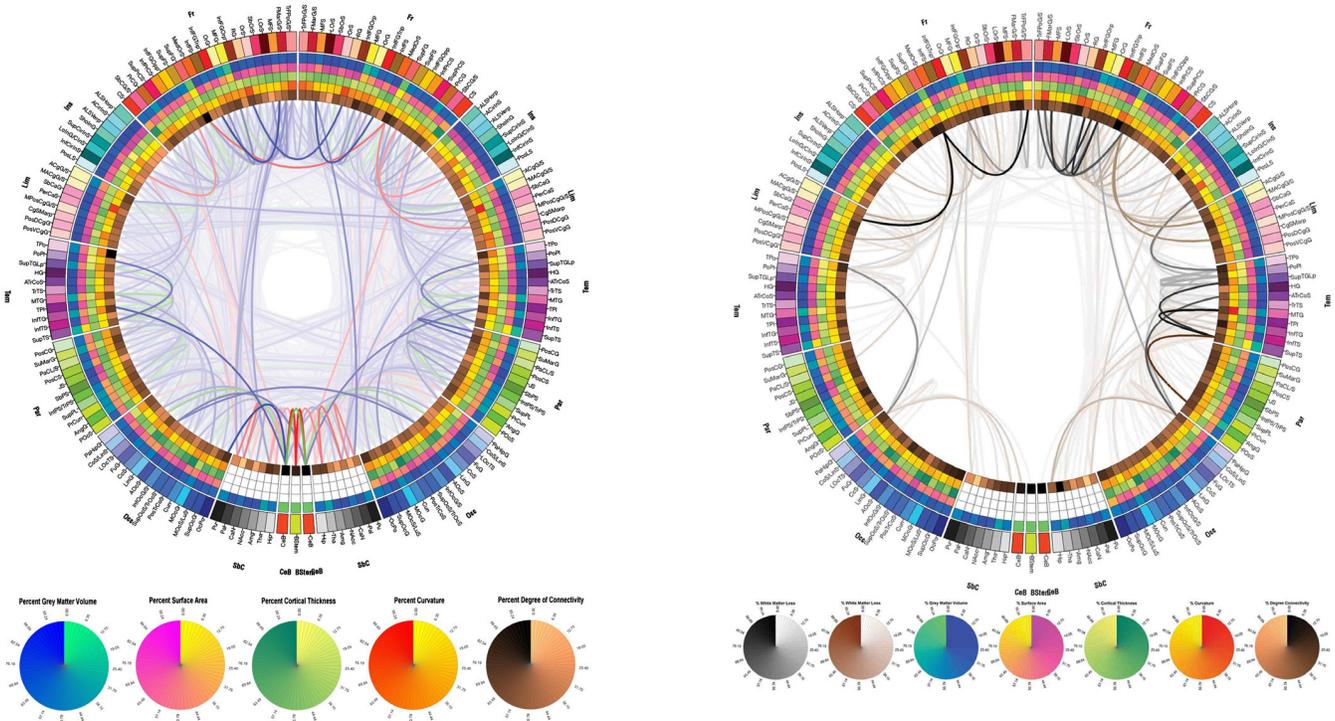
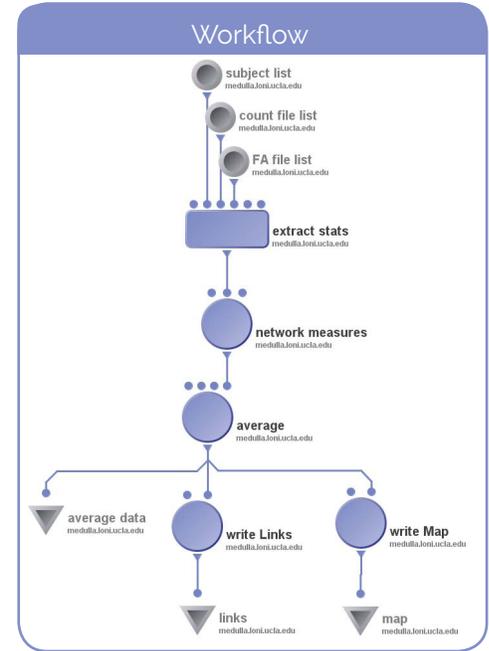
The claustrum seems to have been waiting for the science of connectomics. Due to its tiny size, the structure has remained remarkably difficult to study until modern technological and mathematical advancements like graph theory, connectomics, diffusion tensor imaging, HARDI, and excitotoxic lesioning. The enigmatic structure has become somewhat legendary for its complex and widespread connectivity; Francis Crick and his colleague Christof Koch even theorized that the claustrum may be the center of human consciousness. Mapping the structure and connectivity of such a small, well-connected structure in a cohort of 100 subjects requires computations that would have been impossible a decade ago, and could be prohibitively time-consuming today. The LONI Pipeline processing environment enables large-scale analyses of cortical architecture and connectivity via local, cluster, or “cloud”-based computing resources. As such, our team was able to create image masks of the region specific to each subject’s brain space, add these masks to each subject’s FreeSurfer labelmap, calculate the number, destination, and FA of the fibers originating in each claustrum, and calculate network metrics across each subject simultaneously, then also generate averages of these statistics. Ultimately, the pipeline workflows we designed also output the data tables we then used to generate a “Connectogram” image, which displays white matter connections and cortical characteristics (such as grey matter (GM) volume, surface area, GM thickness, curvature, and degree of connectivity) for a subject or population.



ANALYSIS OF CONNECTOMIC CHANGES DUE TO TRAUMATIC BRAIN INJURY USING THE LONI PIPELINE

10:45 a.m. - Andrei Irimia

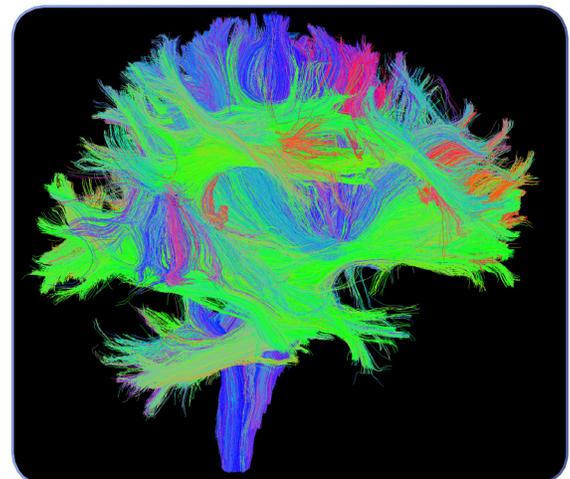
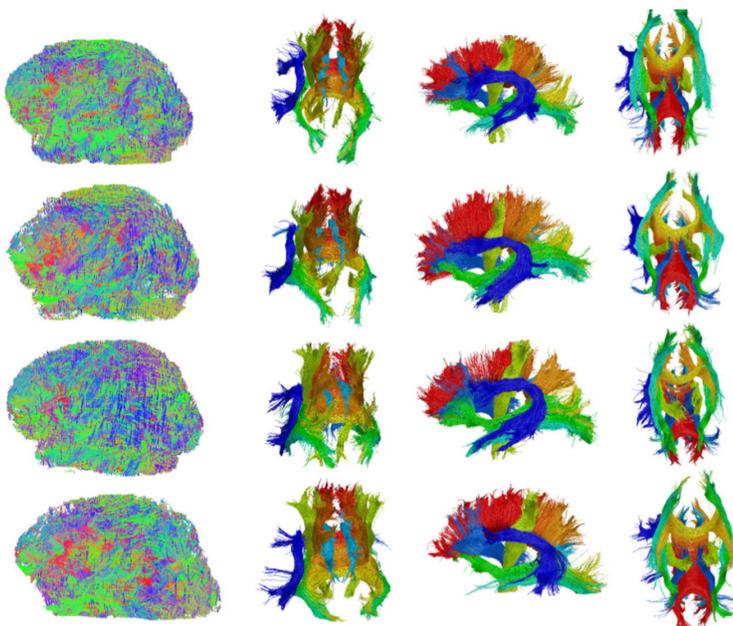
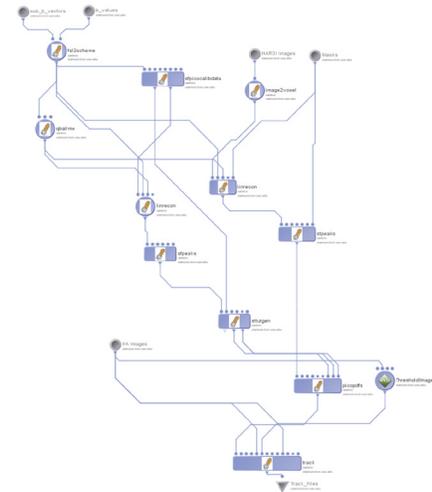
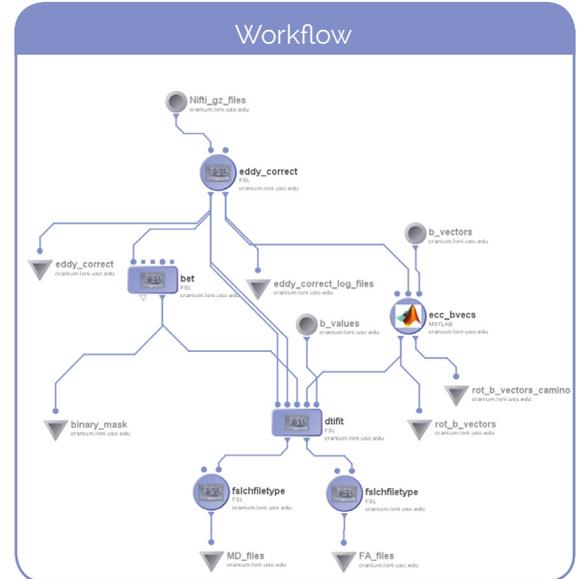
Cortical network architecture has predominantly been investigated visually using graph theory representations. In the context of human connectomics, such representations are not however always satisfactory because canonical methods for vertex–edge relationship representation do not always offer optimal insight regarding functional and structural neural connectivity. Using the LONI Pipeline environment, we have introduced an innovative framework for the depiction of human connectomics by employing a circular visualization method which is highly suitable to the exploration of central nervous system architecture. This type of representation, which we name a ‘connectogram’, has the capability of classifying neuroconnectivity relationships intuitively and elegantly. Within the LONI Pipeline, a multimodal protocol for MRI/DTI neuroimaging data acquisition is combined with automatic image segmentation to (1) extract cortical and non-cortical anatomical structures, (2) calculate associated volumetrics and morphometrics, and (3) determine patient-specific connectivity profiles to generate subject-level and population-level connectograms. The scalability of the approach is demonstrated for a population of 50 adults whose MRI/DTI volumes are processed within the Pipeline. Two essential advantages of the connectogram are (1) the enormous potential for mapping and analyzing the human connectome, and (2) the unconstrained ability to expand and extend this analysis framework to the investigation of clinical populations and animal models.



THE LONI PIPELINE: SOLVING CHALLENGES TO BIG DATA COLLECTION THROUGH THE USE OF A HIGH-THROUGHPUT COMPUTATIONAL ENGINE

11:15 a.m. - Joshua Faskowitz and Faisal Rashid

Recent advancements in neuroimaging have led to an exponential growth in Big Data collection. Neuroimagers collect more data within a few days now than data collected within an entire year just over a decade ago (Van Horn et al., 2013). Many research institutions are not fully equipped to handle large data collection, or to systematically and efficiently implement state-of-the-art computational software and hardware to process the vast amount of data being generated. The Laboratory of Neuro Imaging (LONI) and the Stevens Neuroimaging and Informatics Institute at the University of Southern California house the LONI Pipeline, an easy to use graphical user-interface (GUI) for designing and executing complex data analysis protocols. The Pipeline incorporates a wide range of analysis tools (Dinov et al., 2014). It allows for analysis of large quantities of neuroimaging, genetic, and other modes of data in a visual workflow environment, where data is transferred and distributed to remote high performance servers for computing. Here we demonstrate the Pipeline workflow as an application for automatic clustering of white matter fibers in brain diffusion MRI (Yan et al., 2014). We are able to systematically process raw diffusion weighted imaging (DWI) data through several steps, including linear and non-linear registration, tractography, fiber bundle clustering, and tract visualization and statistical regression analysis using the LONI Pipeline environment.



COMPUTING MULTI-RESOLUTION STRUCTURAL CONNECTOMES WITH THE LONI PIPELINE

11:30 a.m. - Marc Harrison

Creating structural connectomes involves combining diffusion and structural modalities of MRI to understand how each region in the brain is connected to all others. In addition, there is now a great deal of research on how to best choose the shape and number of regions or nodes in the brain to optimally study connectivity. One way to understand different choices of these nodes is to look at connectomes of increasing resolution by refining the regions on the surface of the brain to get a more detailed perspective. These are heavy computational problems that involve clustering and matching surfaces of the brain together using nonlinear deformations. This complexity along with its easy parallelizability makes it ideal for use with the LONI Pipeline where we are able to send each of these computations to a different compute node and create an efficient workflow to get results quickly.

