Visualization and Analysis of Large Medical Image Collections Using Pipelines

by

Ramesh Sridharan
B.S. Electrical Engineering and Computer Sciences, UC Berkeley, 2008
S.M. Electrical Engineering and Computer Science, MIT, 2011

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

June 2015

Signature of Author: ______________________________

Department of Electrical Engineering and Computer Science
May 20 2015

Certified by: ______________________________

Polina Golland
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by: ______________________________

Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Visualization and Analysis of Large Medical Image Collections Using Pipelines

by Ramesh Sridharan

B.S. Electrical Engineering and Computer Sciences, UC Berkeley, 2008

S.M. Electrical Engineering and Computer Science, MIT, 2011

Submitted to the Department of Electrical Engineering
and Computer Science on May 20 2015
in Partial Fulfillment of the Requirements for the Degree
of Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

Medical image analysis often requires developing elaborate algorithms that are implemented as computational pipelines. A growing number of large medical imaging studies necessitate development of robust and flexible pipelines. In this thesis, we present contributions of two kinds: (1) an open source framework for building pipelines to analyze large scale medical imaging data that addresses these challenges, and (2) two case studies of large scale analyses of medical image collections using our tool.

Our medical image analysis pipeline construction tool, PipeBuilder, is designed for constructing pipelines to analyze complex data where iterative refinement and development are necessary. We provide a lightweight scripting framework that enables the use of existing and novel algorithms in pipelines. We also provide a set of tools to visualize the pipeline's structure, data processing status, and intermediate and final outputs. These visualizations enable interactive analysis and quality control, facilitating computation on large collections of heterogeneous images.

We employ PipeBuilder first to analyze white matter hyperintensity in stroke patients. Our study of this cerebrovascular pathology consists of three main components: accurate registration to enable data fusion and population analysis, segmentation to automatically delineate pathology from the images, and statistical analysis to extract clinical insight using the images and the derived measures. Our analysis explores the relationship between the spatial distribution, quantity, and growth of white matter hyperintensity.

Our next application of PipeBuilder is to a neuroimaging study of Alzheimer's patients, where we explicitly characterize changes over time using longitudinal data. As with the previous application, we introduce a workflow that involves registration, segmentation, and statistical analysis. Our registration pipeline aligns the large, heterogeneous group of populations while still accurately characterizing small changes in each patient over time. The statistical analysis exploits this alignment to explore the change in white matter hyperintensity over time.

Thesis Supervisor: Polina Golland
Associate Professor of Electrical Engineering and Computer Science

Acknowledgments

Officially, this thesis is the product of many hours spent staring at problem sets, whiteboards, and computer screens, and the rest of the thesis will describe that portion in detail. But the real value of the graduate school experience has been the growth, learning, and enrichment that came with the process over time. This value at the heart of the PhD program has been primarily due to the amazing people I've interacted with and learned from, and this section is devoted to expressing my gratitude for that. Unfortunately this section cannot be as long as the thesis itself, which means that this listing must be woefully incomplete. Still, the lessons I've learned from peers and mentors alike are too great to not describe some of them here.

Perhaps the person from whom I've learned the most has been my advisor, Polina Golland. Polina has taught me how to do research by providing insight and support at all levels of the research process, from working through derivations to evaluating and choosing different research directions. She has also taught me how to be a better scholar by example: her capacity to think of and ask incisive, insightful questions during meetings and talks helped shape my thinking for the better. Polina has also devoted countless hours to editing papers and providing detailed feedback. While my writing skills have seen modest improvements since entering grad school, I know my skills as an editor have grown by leaps and bounds thanks to Polina's example. Finally, Polina's genuine interest in her students' well-being, both personally and professionally, has manifested itself repeatedly, from our discussions about career choices to her encouragement to seek out professional opportunities. For all these and more over all my years in grad school, I am deeply grateful to Polina.

As I sought a thesis topic, I had the opportunity to learn from my committee members, Rob Miller and Ron Kikinis. Through discussions with Rob and members of the UID group, I learned to constantly question and seek out the simplest, most effective design. Ron has provided a much-needed perspective and drive toward broader usefulness of the tools in the thesis. I appreciate his constant push to ensure the things I build are truly useful. I am certain that I will hold on to these lessons as I move beyond my graduate studies, and I thank Rob and Ron for their insights and influences.

I was fortunate to learn a great deal through a collaboration with Natalia Rost and members of her stroke research group. This collaboration served as the motivation for much of the work in this thesis. Natalia Rost, Lisa Cloonan, and Cathy Zhang not only supplied data and interesting and challenging problems, but more importantly taught me a great deal through our collaboration. A collaboration of this nature is about so much more than the exchange of data and results. While I learned about stroke, neurology, and neuroimaging through our collaboration, much more importantly, I learned the fundamentals of interdisciplinary collaboration: communicating, listening to, and learning from different viewpoints. For these lessons and experiences, I thank them.

Among the best parts of my grad school experience have been the by numerous teaching opportunities I've been afforded. I would like to thank the instructors I've had the pleasure of working with and learning from. In particular, I want to thank Polina, Greg Wornell, and Vivek Goyal. Polina and Greg taught me the value of discipline and timeliness in running a class like a well-oiled machine. Similarly, Vivek taught me by example the importance of enthusiasm and attention to detail in developing assignments. Finale Doshi-Velez entrusted to me the teaching of 6.s085, an introductory statistics class, first as a co-instructor and then as an instructor as she moved on to bigger and better things. This class provided me with tremendous room for experimentation and growth in both my teaching and statistics abilities. For these lessons and countless others, I am indebted to all of them.

I am lucky to have had an amazing set of peers in the Golland group over the years. In particular, Danial Lashkari, Adrian Dalca, and George Chen have been stupendous collaborators. I begain working with Danial early in my studies. He taught me the nuts and bolts of statistical modeling research and guided me through my first successful project. Several years later, after a slew of false starts following Danial's departure, I began collaborating with Adrian. We worked together on many projects, including the work in this thesis. I will be eternally grateful for our marathon whiteboard sessions, late nights before deadlines, and weekly Monday meetings (even if we didn't always meet every week). Adrian has taught me to soldier on through disappointments and to step back and see the big picture when I got lost in the details. I am certain that I could not have asked for a better collaborator, let alone collaborator and friend. George has been my partner in crime in many teaching endeavors, from TAing the pilot offering of 6.s080, a class on statistical inference, to working as co-instructors of 6.s085. George's dedication to teaching has inspired me, and working together with him has been a delightful and edifying experience. Beyond these three close collaborators, the various members of the group over the years have provided many hours of work and play, from daily lunches to heated discussions over papers to endless sessions of invaluable feedback on talks and posters. In rough chronological order, I've been lucky to overlap with and learn from Thomas Yeo, Wanmei Ou, Archana Venkataraman, Michal Depa, Mert Sabuncu, Tammy Riklin-Raviv, Bjoern Menze, Georg Langs, René Donner, Andrew Sweet, Gabriel Tobon, Amelia Arbisser, Christian Wachinger, Kayhan Batmanghelich, Danielle Pace, and Polina Binder. It has truly been a pleasure working with and learning from these amazing people.

Outside of my own group, it has been a pleasure to work with and learn from many insightful people in and around the lab. In my first few years, conversations with Matt Johnson and Roger Grosse about machine learning and computer vision inspired me to consider research in this area, and they have been a continual source of wisdom and knowledge ever since. In addition to being a close friend whose endless drive and work ethic has truly inspired me, Matt was also the first to encourage me to work in cafes, which has contributed enormously to the work in this thesis (indeed, the writing alone has happened in at least 15 different cafes in three different time zones). My officemates over the years have provided me with fun, distraction, answers to technical questions, and long philosophical discussions. Danial, Adrian, George, Bjoern, Ilya Baran, Giorgos Papachristoudis, and Zoran Dzunic have enriched my time in the office. Other members of the computer vision group were a delightful source of conversation, both technical and nontechnical. In addition to those already mentioned, Katie Bouman has been a dear friend. I am truly fortunate to have had coworkers who were also true friends; I am eternally grateful to all of these wonderful people.

Of course, grad school is about much more than the lab, and the last seven years have been made incomprehensibly richer due to the friendships I've been lucky enough to make. As this section is already too long, I'll avoid listing again the people above, many of whom have been close friends as well as colleagues. Over the years, I've been fortunate to be a part of many groups through which I met a diverse group of people. I will cherish my time in the Sidney-Pacific house government, the EECS Graduate Student Association, and the EECS Resources for Easing Friction and Stress (REFS) program. The REFS program in particular gave me the opportunity to work with many caring, dedicated fellow students. I would also like to give special thanks to Harr Chen. Harr and I met early in my grad school experience, and he sparked my nascent love of travel and inspired me to seek out opportunities and experiences outside the sphere of everyday life, and I will always treasure our friendship. In my later years, the spark of my interest in cooking and baking was kindled by my friendship with Audrey Fan, Carrie Cai, William Li, and Sophie Chou. Our many hours of cooking, eating, and laughing have been among the highlights of my grad school experience. Just as the rest of this section, this paragraph in particular is characterized more by its omissions than its inclusions: I have met so many incredible people during my journey through grad school, and I humbly apologize for all exclusions here.

While grad school has been an incredible journey, it is still only a small part of a much greater journey that began many years ago. I would like to thank my parents and my brother and sister, who for years have pushed me to dream big and work hard. Their truly boundless unconditional support and encouragement have made me what I am today, and they will always be a source of inspiration and love.

Contents

Chapter 1

Introduction

The field of medical image computing, or medical image analysis, is broadly concerned with extracting insights from images of biological tissue using computational analysis. Its aims are ambitious and varying in scope, from clinical analyses that seek to characterize links between imaging and disease in patient cohorts to neuroscientific studies that pursue scientific truths such as the structure and function of the human brain. These analyses, while myriad in application, are largely built upon a core set of computational methods from a range of fields, from signal processing to machine learning to biophysics. As an inherently computational discipline, the foundation of medical image analysis is in building computational structures that take in data—in the form of both imaging data and auxiliary data ranging from demographic information to genetic markers to the results of clinical tests—and produce insights about the task at hand.

A fundamental aim of these methods—indeed, a fundamental aim of most applications of statistics and data analysis—is to draw conclusions about populations of people, whether they are healthy subjects or patients afflicted with some disease, based on a representative sample. To better enable these conclusions, and fueled by increasingly more affordable imaging technology and collaborative acquisition efforts (Bots et al., 1997; Hubert et al., 1983; Paulsen et al, 2008; Regan et al, 2011; Weiner et al, 2012), a growing trend has emerged of large collections of images. While some of these collections are the result of large, well funded research studies (Paulsen et al, 2008; Weiner et al, 2012), others incorporate images acquired during standard clinical practice into large scale research studies (Meschia et al., 2013; Regan et al, 2011; Rost et al, 2010b). This trend toward larger image collections is also partially fueled by a rising interest in genetics and its interaction with imaging (J. Liu and Calhoun, 2014). In these problems, the high complexity of systems under study, relatively small effect sizes, and high intrinsic data dimensionality necessitate large sample sizes in order to draw meaningful conclusions.

A related challenge in medical image computing is the analysis of data acquired in clinical settings. Medical imaging is used predominantly in clinical rather than research settings, and a large fraction of existing image acquisition efforts are geared toward acquiring images to serve a particular clinical or diagnostic purpose. For example, fast imaging may be essential to facilitate urgent followup treatment or to avoid artifacts due to uncontrollable patient motion. In contrast, many medical images used in computational research settings are acquired with the goal of image analysis in mind, and are therefore acquired using imaging protocols that yield higher resolution and contrast but take longer to acquire (Lu et al., 2005). When clinical images are brought into the research setting, they typically present unique challenges due to lower resolution and contrast. These challenges often translate to failures of many traditional analysis techniques and algorithms, which are usually designed with high resolution, high contrast research data in mind. Clinical images are often part of large collections that include broad patient cross sections and hold great promise to provide answers to important questions about populations. While these challenges are nontrivial, so too are the benefits of using this kind of data.

■ 1.1 Contributions

This challenge motivates a primary focus of this thesis: robust and scalable analyses of medical images. In particular, we solve these problems through the use of workflows/pipelines[1], which compose a series of algorithmic steps to form a robust computational data processing structure. Indeed, medical image computing research frequently involves elaborate analysis algorithms that are built in this fashion. These workflows offer a superior alternative to *ad hoc* scripts for such analysis tasks, providing improved scalability and robustness. Iterative refinement and adjustment are imperative components of analytical research into method development and implementation. The development of novel methods to solve problems is almost always preceded by a failure of existing approaches to solve that problem. Thus, any method that can expedite this process of experimentation has the potential to make research more efficient.

In this thesis, we provide contributions of two kinds: (1) software tools that enable rapid iterative refinement for constructing pipelines for medical image computing, and (2) applications of this tool to two studies of cerebrovascular pathology using magnetic resonance (MR) images of the brain. Our first software tool is a system for constructing pipelines for medical image computing. It is designed to enable analysis of low resolution clinical data through convenient structures for refinement and quality control. Our second tool is an integrated visualization system that produces a visualization of a pipeline's structure, outputs, and status from code. The applications include a study of stroke and cerebrovascular pathology in stroke patients and a study of longitudinal progression of similar pathologies in a large cohort of Alzheimer's patients and healthy controls.

[1] Throughout this thesis, we use the terms “pipeline” and “workflow” interchangeably.

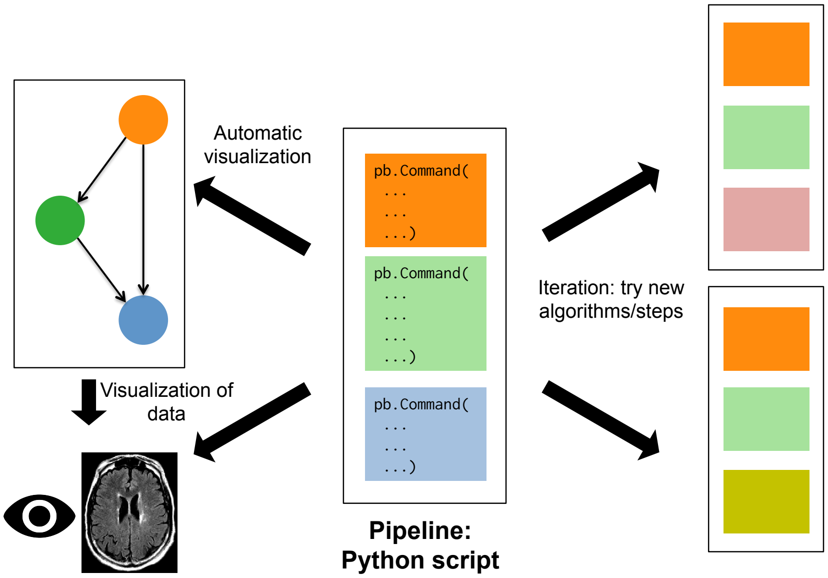


Figure 1.1: The main features of PipeBuilder. Pipelines are specified using scripts (center), and enable rapid iteration and refinement (right). PipeBuilder also provides automatic visualization of pipelines and output data (left).

■ 1.2 PipeBuilder: a pipeline construction tool

We present PipeBuilder, a tool for the construction of pipelines to analyze large medical image collections. Our tool offers a flexible scripting interface for the use of novel algorithms in pipelines. This environment provides scalability and the ability to easily analyze and debug the resulting computational pipeline. Our design choices enable the use of pipelines to process not only well structured, high quality data, but also heterogeneous low quality data. To ensure ease of adoption and compatibility, our open source tool is freely available.

PipeBuilder provides a scripting interface that enables developers and researchers to build and iteratively refine computational pipelines. Users construct pipelines as a series of computational steps whose interconnections are determined automatically. Our interface also provides a simple mechanism for extensions through creation of novel computational nodes. This framework enables rapid iteration and the ability to investigate many approaches to a problem.

We also provide a powerful browser-based visualization tool that automatically computes a visualization of the underlying computational pipeline. Given the code, we determine the connections between computational steps and produce a graph that represents the pipeline. This visualization provides the ability to inspect the status of an active or completed task, as well as an interactive viewer for image-valued intermediate

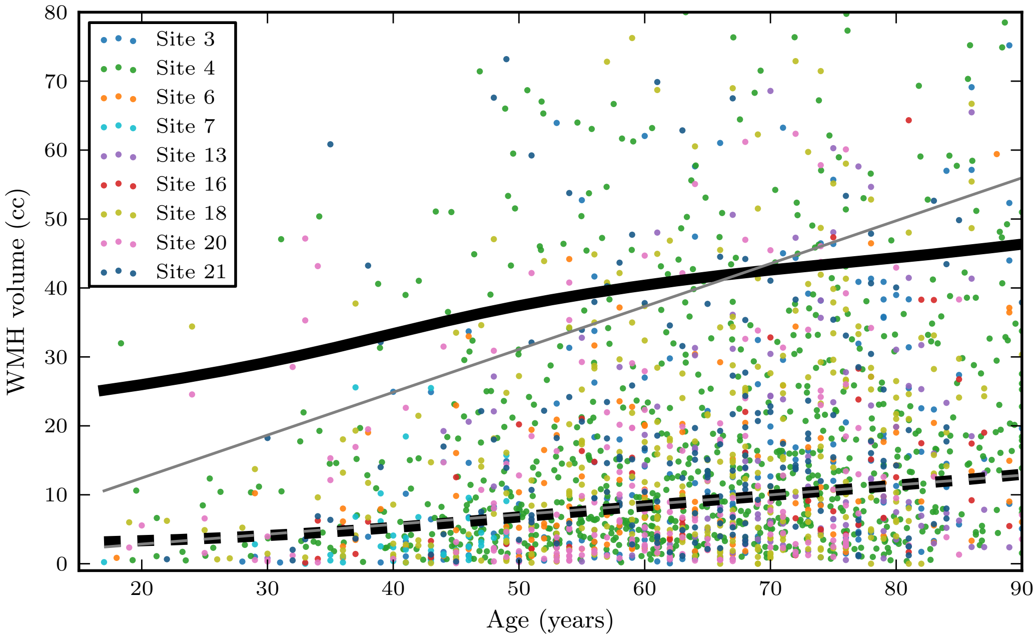


Figure 1.2: Results of our analysis of WMH in stroke patients, showing the relationship between WMH and patient age with two subgroups of patients and their disease trajectories.

and final results, enabling visual verification of the results. Additionally, aggregated values can be inspected directly from our interface to find outliers in numerical outputs.

■ 1.3 Study of cerebrovascular pathology in stroke patients

We demonstrate the strengths of our framework by using it for analysis of white matter hyperintensity in a large cohort of stroke patients. White matter hyperintensity (WMH) is a range of pathologies that manifests as bright areas on T2-FLAIR MR images of the brain and has been linked to a number of other diseases (Rost et al, 2010b; Thanprasertsuk et al., 2014). Segmentation and statistical analysis of WMH pathology promises to provide better understanding of the clinical underpinnings of cerebrovascular health. In this thesis, we present a set of insights for analysis of large clinical image collections such as this one, and employ these insights to construct a robust pipeline for analyzing this data.

The computational pipeline we present consists of three main components: registration, segmentation, and statistical analysis. Our ultimate goal is in-depth study of disease progression in the clinical population, which will deliver insights into the structural and functional changes associated with the disorder from noisy, low resolution images. Accurate registration is a critical prerequisite for such an investigation, as it brings all imaging modalities into a common coordinate frame and enables data fusion across subjects and modalities. Quantifying regions of interest such as WMH requires accurate segmentation. Manual segmentation is infeasible for larger datasets with thousands of images, motivating the development of automatic methods. Population analysis of heterogeneous data requires improved models to capture trends, variability, and statistics.

We present two methods for large scale multimodal registration. The first uses an indirect registration technique that is useful when several modalities may be available, but the modality of the template (i.e., the image in the common coordinate frame) differs from the modality of the images being analyzed. The second requires only a single modality and includes a technique for template synthesis to enable this.

Given accurate alignment, we present an algorithm for multimodal segmentation of WMH and stroke lesions, as well as an in-depth analysis of WMH development as a function of age. Our segmentation algorithm employs a MAP framework for segmentation using image intensities. Our analysis of WMH identifies patterns in the disease progression as a function of patient age. This development is characterized both by its amount and by its spatial distribution across the brain. Our exploratory analysis of the spatial extent of WMH in this cohort indicates distinct patterns across patient groups. Figure 1.2 previews the results of our pipeline.

■ 1.4 Study of cerebrovascular pathology in Alzheimer's disease patients

We continue our exploration of large medical image collections using PipeBuilder with an application to quantifying WMH in a longitudinal cohort of Alzheimer's patients and healthy controls, where multiple images are acquired over time for each subject. Longitudinal data promises to provide information about the change in anatomy and disease over time. However, data fusion of this information requires not only alignment of the images, but also alignment of a measure of change as well.

We present a pipeline for registration that enables such an analysis. In particular, we align the first acquired scan for each patient to a template to account for the large variability in populations. Then, to take advantage of smaller variability within multiple images of the same patient, we align subsequent scans to the first one. By composing different components of these transformations, we can compute transformations between initial scans and subsequent scans in a common coordinate frame across subjects, enabling data fusion. We apply this method to study WMH changes in this cohort. In particular, we examine not only the quantity and distribution across the population, but also the change within a particular patient. This registration frame-

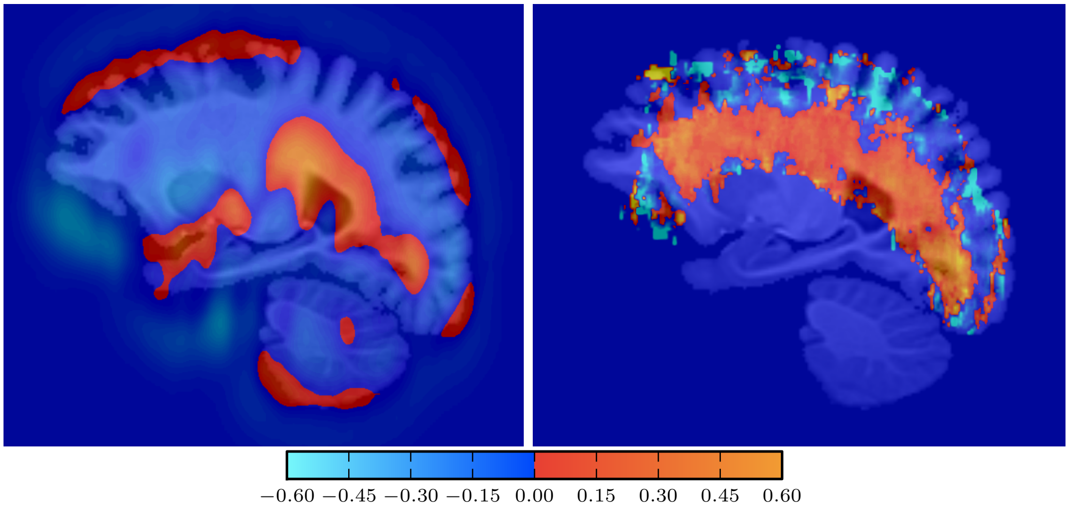


Figure 1.3: Results of our longitudinal analysis of WMH in a large cohort, showing the change (as a function of patient age) in healthy tissue (left) and WMH tissue (right).

work also enables prediction of subsequent images from a single initial scan. Figure 1.3 previews the results of our analysis of longitudinal change.

■ 1.5 Outline

The remainder of this thesis is organized as follows. Chapter 2 reviews existing approaches to pipeline construction. Chapter 3 describes PipeBuilder, our tool for constructing medical image computing pipelines. Chapter 4 provides a necessary background on the disease under study and existing approaches to analyzing the pathology we seek to understand. Chapter 5 and Chapter 6 present our analyses of cerebrovascular pathology in stroke and Alzheimer's patient cohorts, respectively. Chapter 7 concludes the thesis and discusses directions for future work. Appendix A presents a collaborative study enabled by the methods developed in Chapter 6.

Chapter 3 contains material previously presented in Sridharan et al., n.d. This chapter, as well as Chapter 5, contains material from a publication currently under review. The material in Chapter 6 is part of a publication currently under review. Chapter 5 also contains material previously presented in Sridharan et al., 2013.

Chapter 2

Background on pipelines and workflow tools

In this chapter, we review prior work relevant to PipeBuilder. We describe existing pipeline/workflow construction systems, discussing both general-purpose tools and domain-specific tools for medical image analysis and other domains.

■ 2.1 Workflow and visualization systems

Workflow construction tools are typically built on an explicit or implicit graph structure. Within such a graph, nodes or vertices represent actions, computational steps, or data input/output, and edges represent the flow of data from one step to the next. Figure 2.1 illustrates the graph structure used by several common workflow systems. Building a workflow using any of these frameworks entails the construction of such a graph. Beyond this underlying commonality, the tools vary across many dimensions, such as what nodes (computational steps) they provide, the user interface, the mechanics of connecting various computational steps, the pipeline construction mechanism, and the auxiliary features for pipeline execution and data integration.

Workflow and visualization tools fall broadly into two groups: general-purpose systems designed to work across a range of application areas, and domain-specific systems that are tailored toward a particular application area. In this section, we review general-purpose systems, and then focus on domain-specific systems related to our work. In particular, we examine pipeline tools for image processing, visualization, and analysis for their relevance to the work in this thesis.

■ 2.1.1 General-purpose workflows

General-purpose pipeline construction tools are typically highly extensible and flexible so that they can be used across a range of domains. In order to achieve this universality, they often provide a common pipeline format that can be understood across domains. One of the earliest examples of such a system provided for the construc-

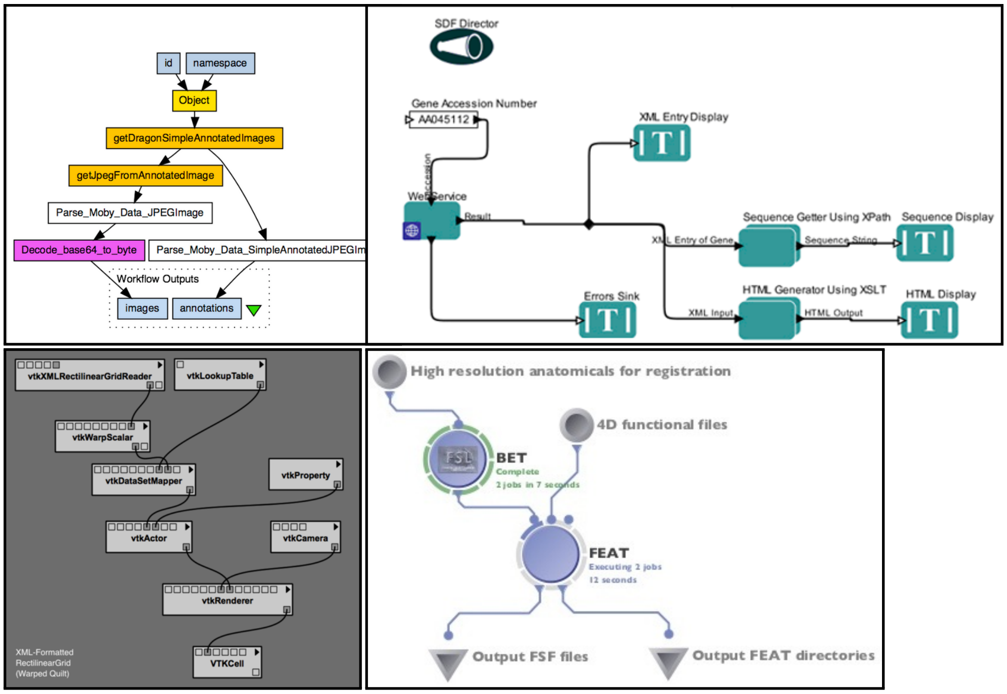


Figure 2.1: Four sample workflows from systems described in this section: Taverna (top left), Kepler (top right), VisTrails using VTK (bottom right), and LONI Pipeline (bottom right).

tion of simple arithmetic operations using a graphical interface (Sutherland, 1966). A precursor to most modern tools is IBM Data Explorer (Lucas et al., 1992), which introduced the notion of a graphical workflow framework with interoperable nodes whose inputs and outputs shared a common format. Recent general-purpose tools such as Kepler (Ludäscher et al., 2006), Taverna (Oinn et al., 2004), and VisTrails (Callahan et al., 2006) provide similar graphical interfaces for constructing a broad range of scientific computing workflows, and remain in widespread use. The building blocks for these workflows can come from a wide range of sources, ranging from simple arithmetic operations to pre-built executables to web services that process data remotely. Indeed, some general-purpose tools such as Triana (Taylor et al., 2007) are dedicated to providing access to services that run on middleware such as remote servers or computing grids.

Modern workflow tools often provide a number of useful and beneficial features that make them superior to the simplest alternative of using *ad hoc* scripts. The features typically fall into the category of supporting infrastructure, and include the ability to share workflows, track provenance of data, provide fault tolerance, and easily distribute computation of a predefined workflow across a computational cluster. While each system typically presents its own set of strengths and weaknesses (Barker and Van Hemert, 2008; Curcin and Ghanem, 2008; Yu and Buyya, 2005), these features are commonly found across a range of modern tools, and are a key factor differentiating them from their predecessors (G. Fox and Gannon, 2006).

While the primary interface to most of these tools is graphical, many of them, including Kepler (Ludäscher et al., 2006) and Taverna (Oinn et al., 2004), provide hybrid interfaces that also allow workflows to be specified using a script. Graphical interfaces can be simpler and more interpretable, but scripting approaches typically provide a finer level of control. This is especially true when a broad range of building blocks or complex set of operations is available, since designing usable interfaces becomes more difficult as more complex operations are supported by the system.

■ 2.1.2 Domain-specific tools

The general-purpose tools described above provide a powerful abstraction. However, they may not take domain-specific considerations into account. For example, within a particular research domain, there is typically a set of common algorithms or software packages that are widely used within that community. Because convenient access to such computational building blocks is an important component of a workflow construction framework, many domain-specific systems have been developed. We note that this distinction between domain-specific and general-purpose is not a dichotomy, but rather a spectrum. Many of the domain-specific tools described here typically aim to provide an interface for using custom components in the computational process, just like their general-purpose counterparts.

Domain-specific systems typically aim to provide general, extensible interfaces while making common tasks within a domain as simple and straightforward as possible. An early example is Khoros and its accompanying visual programming language Cantata (Konstantinides and Rasure, 1994), which focus on signal and image processing applications. They therefore provide built-in support for signal processing tasks such as computing FFTs and filtering, and image processing tasks such as texture extraction and medial axis transformations. In a different domain, the Application Visualization System (AVS) (Upson et al., 1989) focuses on applications involving computer graphics, rendering, and data visualization.

Many modern domain-specific tools exist to enable construction of tailored workflows in a wide range of domains including machine learning (Berthold et al., 2008; Mierswa et al., 2006; Patel et al., 2010; *RapidMiner* 2012), astronomy (Berriman et al., 2004), cell modeling (Lloyd et al., 2004), and seismology (Maechling et al., 2005). As with their earlier counterparts, these frameworks typically make specific considerations for particular domains. For example, systems such as RapidMiner (*RapidMiner* 2012) and Knime (Berthold et al., 2008) provide specialized interfaces for machine learning tasks, featuring specialized interfaces to commonly used algorithms such as k-means and support vector machines. In addition, the idea of extracting a workflow from a set of operations has been explored in domains from spreadsheet processing (Fisher et al., 2006) to commercial image editing software such as Adobe Lightroom.

While most of these methods provide detailed views of workflows, they typically do so using monolithic interfaces that make strong assumptions about how the data is distributed (or not) across computer systems and institutions. In particular, they are targeted toward research applications with open data that is freely available, at least to all who can access the workflow system. However, interdisciplinary collaborative fields such as medical image computing often require sharing of data between two distinct types of researchers: domain experts such as clinicians, who provide data, motivation, interpretation, and domain expertise; and technical experts, who develop algorithms and process data. This data is often subject to restrictions and controls. Additionally, the results should be accessible to users with and without technological expertise. Such interdisciplinary collaboration is frequently facilitated by *boundary objects* (Star, 1989; Star, 2010; Walenstein, 2003), or tools that bridge an interdisciplinary gap and enable communication between groups. Most workflow construction and visualization frameworks are not designed to be boundary objects. In contrast, we argue that this is a critical role for such systems to fill within medical image computing research due to its interdisciplinary nature. This motivates the design and accessibility goals of our contributions. In particular, we argue that many of the visualization and interaction features would be most useful if available in a lightweight, cross-platform manner. This motivates our development in this thesis of a browser-based visualization of pipelines that can be accessed outside the computer system where the data is stored and processed with the appropriate protections for sensitive data.

■ 2.1.3 Visualization

As part of the graphical interface, workflow construction tools typically provide several different kinds of visualization. First, they often support visualization of the data flow via the graph representation. For graphical interfaces such as the ones provided by the systems discussed earlier, the visualization and pipeline construction occur simultaneously. Workflow construction frameworks sometimes also include mechanisms dedicated to visualization of pipeline outputs. In fact, systems such as VTK (Schroeder et al., 2000) provide scripted interfaces for visualization and pipeline rendering of images and other graphics. Additionally, a central feature of VisTrails (Callahan et al., 2006) is its ability to construct pipelines for data transformation for the purpose of visualization using a graphical interface, typically using VTK. Finally, workflow systems can also visualize the version history of the workflow being developed (Callahan et al., 2006): sometimes referred to as design provenance, this enables the evolution of the pipeline itself to be tracked and evaluated.

■ 2.1.4 Medical image computing workflows

Medical image computing shares a great deal in common with the specialized domains described above. Analysis of the images requires the use of signal processing and image analysis blocks as provided by frameworks such as Khoros (Konstantinides and Rasure, 1994). It also requires visualization and rendering of data that is often three-dimensional, as provided by libraries such as VTK (Schroeder et al., 2000) and AVS (Upson et al., 1989). It also often requires the use of machine learning algorithms, as provided by systems like Knime (Berthold et al., 2008) and RapidMiner (*RapidMiner* 2012). But beyond these, it also requires interfacing with a range of special-purpose algorithms and software packages, which we discuss in this section.

In medical image computing, researchers and engineers approaching an algorithm design or a problem involving pipeline construction can often choose the algorithmic components from a wide assortment of toolkits for solving individual steps. Some toolkits such as 3D Slicer (Pieper et al., 2004) and GIMIAS (Larrabide et al., 2009) enable interactive application of algorithms on a single image or small collection of images. While excellent for iterative refinement or small scale studies, such toolkits are not designed for use with large multi-subject studies, where once an algorithm has been developed, it must be applied and tested across a large dataset. Many other toolkits provide simple building blocks that must be combined into an application-specific solution (Avants et al., 2011a; Fischl, 2012; Jenkinson et al., 2012; T. Yoo et al., 2002). These building blocks enable a wide range of capabilities, from broadly applicable tasks such as image registration and image segmentation to important domain-specific preprocessing tasks such as motion correction. Such software is typically modular and flexible, but must be integrated in order to perform meaningful tasks.

Pipeline systems for composing individual computational components in medical image computing can be grouped into two broad categories: monolithic frameworks and flexible platforms for designing custom pipelines. Monolithic tools are typically designed for a particular task (Ashburner et al., 2008; Fischl, 2012), and impose rigid requirements on data organization and the task being performed to avoid the challenges presented by heterogeneous data and computation structures. For example, FreeSurfer (Fischl, 2012) provides both a modular set of programs for processing MRI data and a pipeline for surface-based and volumetric analysis of T1-weighted brain MRI. That is, the pipeline is a rigid structure that conducts a particular set of analyses on a specific type of data. While these analyses address a common use case within medical image research, they are not suited for every application. For example, a study that acquires images of a different modality or aims to conduct a different analysis would require a different algorithmic pipeline.

In contrast, flexible platforms are more similar to the workflow systems described earlier, and typically enable users to construct custom pipelines for a specific task at hand, providing an alternative to *ad hoc* scripts. These methods typically either provide a graphical interface a developer can use to construct a pipeline visually by combining building blocks (Parker and Johnson, 1995; Rex et al., 2003; Koenig et al., 2006) as with the general-purpose tools discussed above, or allow the construction to be programmed as a script within a software environment (Bellec et al., 2012; Gorgolewski et al., 2011). Graphical interfaces provide a natural visualization of the underlying pipeline as a graph, but are more difficult to scale to complex analyses and lack the expressive power of scripting approaches. As described earlier, general-purpose workflow frameworks such as Taverna and Kepler allow hybrid approaches, integrating scripting capabilities within the graphical platform. While both types of systems offer a more scalable and feature-rich alternative to *ad hoc* scripts, they often do so at the cost of requiring more overhead both in learning how to use the tool and in developing a pipeline.

In contrast, the work presented in this thesis aims to combine the best of both visual and scripting methods. In particular, we present a twofold contribution in this regard. First, we present PipeBuilder, a lightweight tool for medical image analysis workflow construction that focuses on enabling algorithm development using iterative refinement. Second, we present a visualization and interaction system that enables better understanding of workflows for large scale image analysis. PipeBuilder aims to visualize any pipeline represented by an underlying graph and data structure, further enabling iterative refinement.

We also provide several integrations of PipeBuilder into Nipype, an existing framework for scripting medical image analysis pipelines (Gorgolewski et al., 2011). Motivated by its existing user base and collection of auxiliary features, we provide both a mechanism to import PipeBuilder pipelines into Nipype. In addition, our visualization tool aims to be general-purpose, providing visual representations of computational structures and input and output data for both PipeBuilder and Nipype pipelines.

■ 2.1.5 Databases and storage

Pipeline systems typically interact with input and output data, and many methods have been proposed for organization and structure of such data. They address a set of challenges related to a different aspect of the same underlying problem addressed in this thesis, namely, handling large collections of medical imaging data. As most pipelines must interact with this data, we briefly review the current state of medical image storage architecture.

Medical images are acquired and stored in picture archiving and communication systems (PACS), which provide access to medical imaging data to clinicians for medical practice (Huang, 2011). However, these platforms are designed for clinical practice, not for large scale image analysis. Toward this end, many solutions have been proposed to address the problem of storing and providing convenient access to medical imaging data for computational research. Many publicly available large medical collections employ *ad hoc* solutions to distribute their data (Van Horn and Toga, 2009). These include sophisticated systems employed by large, widely used repositories such as the Human Connectome Project (Van Essen et al., 2012) and the Alzheimer's Disease Neuroimaging Initiative (Jack et al., 2008), as well as simpler systems employed by a range of smaller studies. Many methods arise as publicly available versions of these *ad hoc* solutions. All of these methods provide separation of data storage and processing. Each emphasizes a different aspect of data storage and distribution, such as data entry and quality control as in the eXtensible Neuroimaging Archive Toolkit, or XNAT (Marcus et al., 2007); multi-site longitudinal studies as in Longitudinal Online Research and Imaging System, or LORIS (Das et al., 2011); or ease of integration with workflows using the DICOM format, as in Chronicle (Pieper, 2014). In the domain of neuroimaging, specifically functional neuroimaging, many platforms exist for aggregating and publishing data in raw and processed forms (Gorgolewski et al., 2013; Laird et al., 2005).

In the next chapter, we present PipeBuilder, our solution for constructing medical image analysis pipelines, and detail its design goals and salient features.

Chapter 3

PipeBuilder: A tool for medical image computing workflows

As discussed in Chapter 1, pipeline development is an important component of medical image computing research. In this chapter, we describe the PipeBuilder framework, its key features and design choices, and their relation to key challenges and issues facing medical image computing researchers. We explain how PipeBuilder provides the scalability and robustness necessary for large datasets.

The remainder of this chapter is structured as follows. Section 3.1 describes our design goals and motivations for PipeBuilder. Section 3.2 describes the main features of PipeBuilder and provides several examples illustrating them. Section 3.3 describes the relationship between PipeBuilder and several other tools for medical image analysis pipeline construction, including comparisons and a discussion of our integration with one of these tools. Section 3.4 discusses how various kinds of user feedback influenced the design of PipeBuilder. Finally, Section 3.5 summarizes the design goals and main contributions.

■ 3.1 Design goals

Iterative refinement

Analysis of low resolution clinical quality data often requires the use of novel steps and algorithms as part of the computational pipeline. In choosing and adapting these algorithms, substantial refinement and iteration is often necessary to obtain clinically relevant results. This process of attempting various methods, examining their outcome, and trying new methods is a key component of successful research, and PipeBuilder is designed to enable this.

Adaptability and robustness

Large collections of low resolution data are often heterogeneous and present high variability. Algorithms to analyze and extract insights from such data must be robust to both variations in quality and to variations in other intrinsic properties of interest in the data. For example, in a multi-site imaging study of a disease, patients from one site may exhibit variability requiring a different pipeline. One of PipeBuilder's design goals is to enable the construction of such robust and adaptable pipelines.

Visualization

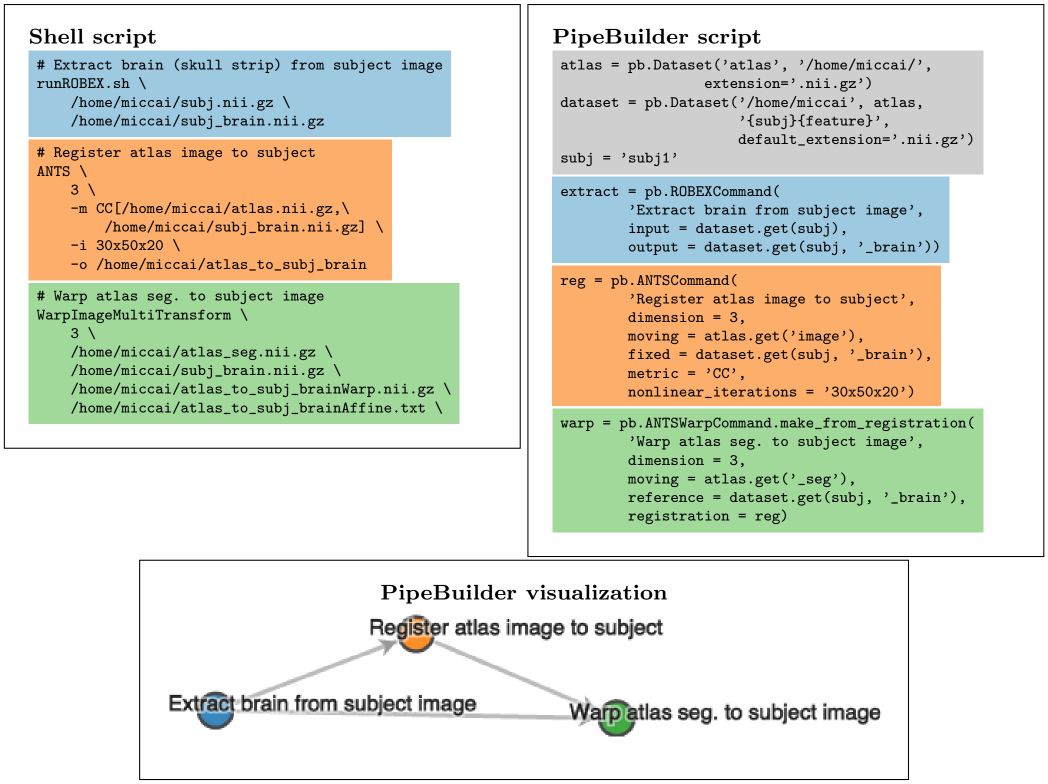
Visualization and interaction are an important component of algorithm design, especially for large imaging datasets. Flexible visualization of a pipeline's numerous intermediate results and possible failure modes during the development process makes it easier to perform careful examination and subsequent refinement. In addition, providing a lightweight visualization interface that is easily accessible to both pipeline developers and collaborators with medical or other domain expertise can facilitate collaboration and sharing of results and algorithmic design. These factors motivate the inclusion of a visualization component in PipeBuilder.

Quality control

Closely related to these three goals is our focus on making it easy to perform extensive quality control (QC), which is often required at each computational step while processing large scale clinical quality data. While iterative refinement and development can produce a robust pipeline that can be adapted to handle variation at a large scale, the reality of such image collections is that some images will inevitably fail to be processed. The ability to identify such outliers and exclude them from subsequent analysis is an important component that we emphasize in the design of PipeBuilder.

■ 3.2 Overview

PipeBuilder pipelines are constructed by users as Python scripts, where commands representing algorithmic tools operate on data represented by filenames. These commands are specified according to their command line interface, and PipeBuilder produces a shell script or series of shell scripts that, when executed, will carry out the computational process. PipeBuilder automatically infers the underlying workflow structure from the Python script. As is conventional for workflow construction systems, we represent a pipeline as a directed graph whose nodes correspond to data processing steps and algorithms, and whose edges correspond to flow of data between different steps. In particular, a directed edge between two nodes indicates that an output of the parent node is used as the input to the child node. Figure 3.1 illustrates a simple pipeline constructed in PipeBuilder, with each section highlighted to improve readability, as well as the automatically constructed visualization. The remainder of this section describes in detail the various components of this pipeline and their relationship to PipeBuilder.



Shell script

```
# Extract brain (skull strip) from subject image
runROBEX.sh \
    /home/miccai/subj.nii.gz \
    /home/miccai/subj_brain.nii.gz
```

```
# Register atlas image to subject
ANTS \
    3 \
    -m CC[/home/miccai/atlas.nii.gz,\
        /home/miccai/subj_brain.nii.gz] \
    -i 30x50x20 \
    -o /home/miccai/atlas_to_subj_brain
```

```
# Warp atlas seg. to subject image
WarpImageMultiTransform \
    3 \
    /home/miccai/atlas_seg.nii.gz \
    /home/miccai/subj_brain.nii.gz \
    /home/miccai/atlas_to_subj_brainWarp.nii.gz \
    /home/miccai/atlas_to_subj_brainAffine.txt \
```

PipeBuilder script

```
atlas = pb.Dataset('atlas', '/home/miccai/',
                 extension='.nii.gz')
dataset = pb.Dataset('/home/miccai', atlas,
                     '{subj}{feature}',
                     default_extension='.nii.gz')
subj = 'subj1'
```

```
extract = pb.ROBEXCommand(
        'Extract brain from subject image',
        input = dataset.get(subj),
        output = dataset.get(subj, '_brain'))
```

```
reg = pb.ANTSCommand(
        'Register atlas image to subject',
        dimension = 3,
        moving = atlas.get('image'),
        fixed = dataset.get(subj, '_brain'),
        metric = 'CC',
        nonlinear_iterations = '30x50x20')
```

```
warp = pb.ANTSWarpCommand.make_from_registration(
        'Warp atlas seg. to subject image',
        dimension = 3,
        moving = atlas.get('_seg'),
        reference = dataset.get(subj, '_brain'),
        registration = reg)
```

PipeBuilder visualization

Register atlas image to subject
Extract brain from subject image
Warp atlas seg. to subject image

Figure 3.1: Example application: simple brain MRI atlas-based segmentation that involves three steps: skull removal (blue), registration to an atlas (orange), and deformation of the atlas labels to the new image (green). Top left: shell script for this pipeline with colored boxes added to visually separate the three components. Top right: PipeBuilder pipeline with similar boxes added. The small additional overhead (gray) enables the features described in the text, in particular the automatic browser-based visualization of the pipeline (bottom left).

■ 3.2.1 Data input and output

The primary method for data flow in PipeBuilder is through interaction with files in a filesystem. Data input and output are managed by a `Dataset` object, which maps application-specific identifiers to filenames. For example, in a multimodal imaging study, files might be parametrized by subject identifier, modality (T1, T2-FLAIR, DWI, etc.), and image feature (image, manually segmented regions of interest, etc.). The `Dataset` abstraction would in this case then require users to specify a mapping between $(\text{subject}, \text{modality}, \text{feature})$ tuples and filenames using a template. An example of a `Dataset` specification is shown in Figure 3.2.

This construct uses templates to map between filenames and named tuples representing the abstract data. PipeBuilder can therefore store, for every node, both (1)

```
dataset = pb.Dataset(
            base_dir = ’/path/to/data’,
            # How are the inputs to the pipeline stored?
            input_template = ’inputs/{subj}_{modality}_{feature}.{extension}’),
            # How should intermediate/output files be stored?
            processing_template = ’results/{subj}_{modality}_{feature}_{description}.{extension}’,
            log_prefix = ’logs/’)
dataset.get_input(subj = 1, modality = ’T1’, feature = ’image’, extension = ’nii.gz’)
dataset.get(subj = 1, modality = ’T1’, feature = ’image’, description = ’preproc’, extension = ’nii.gz’)
```

Figure 3.2: Example `Dataset` for PipeBuilder that represents input and output data. The `input_template` argument provides a template for data used as input to the pipeline. In this example, the fields are `subject`, `modality`, and `feature`. Each must be provided when accessing data, as illustrated in the lines of code at the bottom. If the templates do not have any ambiguity, then PipeBuilder will automatically infer the fields from the resulting filenames, inverting the filename construction.

```
class ROBEXCommand(Command):
    def __init__(self, desc, **keyword_args):
        self.cmd = ’/path/to/ROBEX/runROBEX.sh %(input)s %(output)s’
        Command.__init__(self, desc, **keyword_args)
```

Figure 3.3: Example code for specifying a new computational node that corresponds to the skull removal step in Figure 3.1 using ROBEX (Iglesias et al., 2011). The third line specifies the details of the interface: a sample command line usage with templates for the argument files suffices to enable the use of this tool in PipeBuilder. The remaining lines are Python object-oriented boilerplate.

information about the input and output files for ease of visualization and quality control as well as (2) a more abstract representation for understanding the structure of the pipeline. Additionally, while a specification of the `Dataset` enables PipeBuilder to produce an executable script for a command, a pipeline can easily be adapted to new underlying data by modifying the `Dataset` or creating a new one.

```
class ComplexCommand(Command):
    def __init__(self, desc, cond_val,  **keyword_args):
        # Handle argument differently if negative/positive
        if cond_val >= 0:
            keyword_args[’val’] = str(cond_val)
        else:
            keyword_args[’val’] = ’--negate ’ + str(cond_val)

        self.cmd = ’/path/to/command’
            ’%(input)s %(output)s %(val)’

        Command.__init__(self, desc, **keyword_args)
```

Figure 3.4: Example code for specifying a more complex interface. As with the simpler example, the core of the interface is a specification of the command line with templates filled in using input arguments, but here the arguments are determined based on tool-specific logic for constructing the command line.

■ 3.2.2 Construction of new nodes

Workflows are constructed using a series of `Command` objects, each of which provides access to an underlying tool or algorithm. Each of these objects is constructed using a lightweight interface as illustrated in Figure 3.3. In particular, a single line specifying a template for the command line interface to the tool is enough to facilitate the use of that tool in PipeBuilder. This simple lightweight interface facilitates iterative refinement, which is critical to finding robust solutions to processing of large, heterogeneous image collections. Keyword arguments provided to the `Command` are used to fill in the corresponding entries of the template command line. These command line arguments can either be passed in directly according to the template, as shown in Figure 3.3, or computed dynamically based on other arguments, as shown in the more complex example in Figure 3.4.

In an attempt to provide a flexible interface with as few restrictions as possible, the only constraint on the specification of a `Command` is that the outputs must be annotated as such. In most cases, where the underlying command line tool produces only a single output, this requirement is easy to satisfy, as any template argument named `output` is inferred to be the output. In cases where the command line tool produces multiple outputs, a single extra line should be added to enable the automatic visualization. Similarly, in cases where a command line tool takes a template or prefix as input, PipeBuilder cannot automatically infer the resulting filenames from the template or prefix; these can be specified manually in a similar fashion.

Once a tool's usefulness has been verified through application and quality control, the `Command` interface to that tool can be refined to include type checking and other more complex features that are easily available in the underlying scripting language. In addition, while many tools such as the one in Figure 3.3 allow for a simple command line specification, others may feature idiosyncrasies that require special handling. The interface provided by PipeBuilder facilitates such special handling as an easy extension, again facilitated by the ability to use arbitrary features of the underlying language.

■ 3.2.3 Workflow construction and execution

After specification of a pipeline in this fashion, PipeBuilder produces a shell script or series of shell scripts. These can either be executed directly in an interactive session for ease of debugging, or can be distributed across a computational cluster for scalability. PipeBuilder provides an interface to Oracle Grid Engine to parallelize across a particular parameter of interest, which is typically the subject index.

PipeBuilder can also be robust to partial or incomplete data, as is common with large imaging studies. In particular, a `Dataset` can specify that certain partial input formats are mandatory, which causes any attempt to access a missing file to fail fast.

For example, in the example given in Figure 3.2, the entire processing may depend on the existence of a specific modality. In this case, the following line of code would mark the “image” feature of the modality “T1” as being mandatory:

```
dataset.add_mandatory_input(modality = 'T1', feature = 'image')
```

If subsequent data access for any particular subject encounters a missing instance of such a file, the pipeline will not be run for that subject. In contrast, if a non-mandatory missing input is detected, only the steps of the pipeline that depend on that input (as well as their descendants) are skipped. PipeBuilder also avoids rerunning steps whose inputs are the same and whose outputs already exist.

■ 3.2.4 Visualization

We provide a visualization in PipeBuilder that is constructed automatically from the pipeline code, offering a rich set of features for interacting with the pipeline and its results. Its use requires no additional software download or installation beyond a modern web browser, and can be viewed not only on the computer where the pipeline is executed, but from any computer that is able to successfully authenticate. In addition to the improved ability to understand and debug a pipeline, this enables sharing of pipeline results between collaborators.

Our tool enables visualization of the pipeline itself in addition to detailed information about the nodes and edges of the underlying workflow graph. As illustrated in Figure 3.5, clicking on a node in the graphical representation provides access to information about the underlying process, including standard output, standard error, command line arguments, and return status. In addition to text, image-valued outputs can be shown using browser-based viewers for medical imaging data, such as Slice:Drop (Haehn, 2013), as shown in Figure 3.5, and Papaya (Lancaster and Martinez, n.d.). These visualizations facilitate quality control and pave the way to constructing robust, adaptable pipelines. This visualization tool is not constrained to the system where the computation is being performed. PipeBuilder enables visualization of pipelines in the browser, so that both the structure of the computational pipeline as well as its results can easily be communicated. This is particularly useful for medical imaging applications, where results, both individual and aggregate, can be shared with collaborators who may be unfamiliar with the technical details of implementation of the computational pipeline.

The visualization provided by PipeBuilder also facilitates easy quality control for each step of the data processing. In addition to the visualization of text and image-valued data described above, PipeBuilder also provides an aggregation mode for the common case where the same pipeline is applied across a large image collection. While the figures and examples in this section all aggregate by subject, this can be applied to

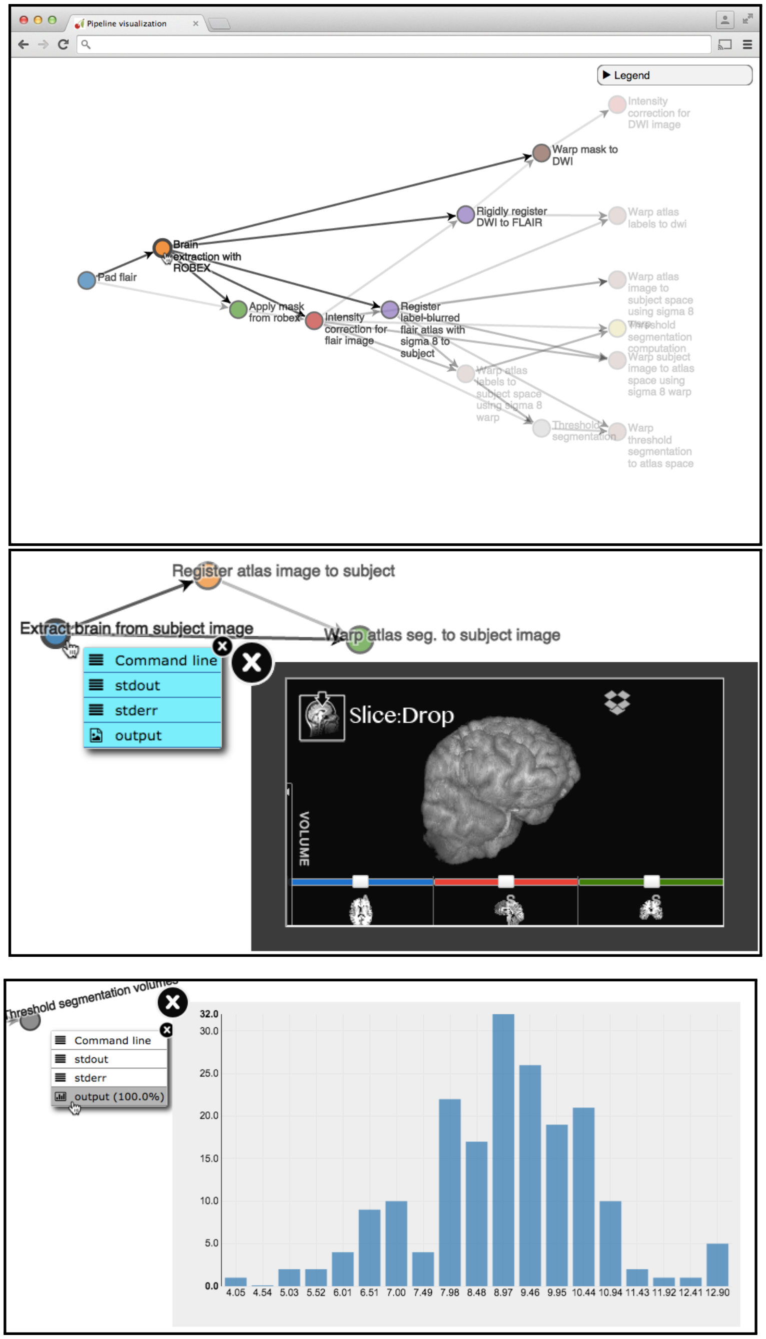

Pipeline visualization
Legend
Intensity correction for DWI image
Warp mask to DWI
Rigidly register DWI to FLAIR
Warp atlas labels to dwi
Brain extraction with ROBEX
Pad flair
Apply mask from robex
Intensity correction for flair image
Register label-blurred flair atlas with sigma 8 to subject
Warp atlas image to subject space using sigma 8
Threshold segmentation computation
Warp subject image to atlas space using sigma 8 warp
Warp atlas labels to subject space using sigma 8 warp
Threshold segmentation
Warp threshold segmentation to atlas space
Register atlas image to subject
Extract brain from subject image
Warp atlas seg. to subject image
Command line
stdout
stderr
output
Slice:Drop
VOLUME
Threshold segmentation volumes
Command line
stdout
stderr
output (100.0%)
32.0
30.0
25.0
20.0
15.0
10.0
5.0
0.0
4.05
4.54
5.03
5.52
6.01
6.51
7.00
7.49
7.98
8.48
8.97
9.46
9.95
10.44
10.94
11.43
11.92
12.41
12.90

Figure 3.5: Illustration of PipeBuilder's visualization features. *Top*: visualization of a complex pipeline (see Chapter 5 for details on the underlying computational process). Mousing over a single node in the graph highlights that node, focusing on its parents and children. *Middle*: clicking on a node reveals a menu that can be used to display status information about a node as well as visualize image outputs in the same window. *Bottom*: in PipeBuilder's aggregation mode, clicking on a node with scalar output collects the data across a population, producing a histogram visualization of the data. Both the image viewer and the histogram are displayed in the browser, overlaid on the pipeline diagram.

any variable for which the pipeline iterates over many values, such as parameter values in parameter sweeps. The user need only annotate this variable as being iterated over in this fashion.

Many QC steps involve the computation and evaluation of quantitative measures for each image relative to the rest of the data. This is often a laborious and time-consuming procedure that requires substantial back-and-forth between loading image files from disk into viewers and matching them to QC output values. To streamline this process, the aggregation feature in PipeBuilder automatically detects scalar-valued outputs at each pipeline step, and displays these values in a histogram for in-depth analysis of failure modes, as illustrated in Figure 3.5. This feature also displays a list of outlier parameter values (e.g., subjects) in the browser console to enable special handling or exclusion as needed, reducing the researcher time needed for analyzing data. In addition, our tool is open source, inviting the development of even more visualization plugins for efficient summarization and analysis of large medical image collections.

Implementation details

The visualization is computed automatically from the code. Each `Command` has its outputs and inputs inferred by the program based on command line inputs and potentially on auxiliary information as described in Section 3.2.2. A directed edge is inferred between two nodes if the output of the parent is used as an input to the child. Auxiliary information about each execution is written to disk at the same time as the shell script corresponding to workflow execution, and each computational step is wrapped by an auxiliary script that stores standard input and output appropriately. This information is retrieved by the server as needed as the user interacts with the corresponding display element.

The visualization itself is implemented using a server-client architecture. The server, implemented using CherryPy (Delon, 2014), interacts with the workflow and underlying data. The underlying data, represented by JSON, is served to the client, which then visualizes it using d3.js (Bostock et al., 2011), relying on the server to retrieve data such as images or aggregate statistics.

■ 3.3 Relation and integration to other workflow tools

■ 3.3.1 Comparison to existing tools

PipeBuilder stands apart from other medical image computing workflow construction tools in large part due to the focus on the design goals described in Section 3.1. Of the existing pipeline construction systems for medical image computing described in detail in Chapter 2, Nipype (Gorgolewski et al., 2011) and the LONI Pipeline (Rex et al.,

Nipype

```
import nipype.interfaces.utility as util
import nipype.interfaces.ants as ants
import nipype.pipeline.engine as pe
data_src = pe.Node(
        nio.DataGrabber(infields=[’subj’, ’feature’]),
        name=’Subject data source’)
data_src.inputs.base_directory = ’/home/miccai’
data_src.inputs.template = ’%s%s.nii.gz’

atlas_src = pe.Node(
        util.IdentityInterface(
            fields=[’atlas’, ’atlas_seg’]),
        name=’Atlas data source’)
atlas_src.inputs.atlas = ’/home/miccai/atlas.nii.gz’
atlas_src.inputs.atlas_seg = ’/home/miccai/atlas_seg.nii.gz’

workflow = pe.Workflow(
        name=’Registration workflow’)
workflow.base_dir=’/home/miccai’
```

```
extract = pe.Node(
        interface=ROBEXTask(),
        name=’Extract brain from subject image’)
```

```
reg = pe.Node(
        ants.ANTS(dimension=3),
        ’Register atlas image to subject’)
reg.inputs.metric = [’CC’]
reg.inputs.metric_weight = [1]
reg.inputs.radius = [4]
```

```
combiner = pe.Node(
        interface=util.Merge(numinputs=2),
        name=’Combine warp + affine’)
warp = pe.Node(
        ants.WarpImageMultiTransform(dimension=3),
        name=’Warp atlas segmentation to subject image’)
```

```
workflow.connect(data_src, ’outfiles’, extract, ’input’)
workflow.connect(extract, ’output’, reg, ’fixed_image’)
workflow.connect(atlas_src, ’atlas’, reg, ’moving_image’)
workflow.connect(reg, ’affine_transform’, combiner, ’in1’)
workflow.connect(reg, ’warp_transform’, combiner, ’in2’)
workflow.connect(combiner,’out’, warp,’transformation_series’)
workflow.connect(atlas_src, ’atlas_seg’, warp, ’input_image’)
workflow.connect(extract, ’output’, warp, ’reference_image’)
```

Figure 3.6: Example code for Nipype that implements the same atlas-based brain MRI segmentation pipeline from Fig. 3.1. While Nipype provides a similar interface for pipeline construction (colored), it requires substantially more overhead (gray).

```
class ROBEXInputSpec(CommandLineInputSpec):
    input = File(exists = True, desc = ’input file’,
            argstr = ’%s’, position = 0, mandatory = True)
    output = File(name_source = [’input’], hash_files = False,
            name_template = ’%s_brain’, position = 3,
            argstr = ’%s’)

class ROBEXOutputSpec(TraitedSpec):
    output = File(exists=True, desc=’Name of output file’)

class ROBEXTask(CommandLine):
    input_spec = ROBEXInputSpec
    output_spec = ROBEXOutputSpec
    cmd = ’/path/to/ROBEX/runROBEX.sh’
```

Figure 3.7: A new node specification for skull stripping In contrast to the PipeBuilder interface in Figure 3.1, the Nipype interface requires a detailed specification of all the inputs and outputs to provide more error checking.

2003) represent two broad categories of tools within medical image workflow construction: scripting interfaces and graphical interfaces, respectively. Similar to PipeBuilder, Nipype provides a scripting environment, while LONI Pipeline provides a graphical environment.

Due to the focus on the design goals described above, PipeBuilder provides several important contributions beyond what is available in existing work. For example, the specification for building a new interface in PipeBuilder is simpler than the corresponding interface one would provide in Nipype. Figure 3.7 demonstrates this contrast: Nipype requires significantly more overhead. PipeBuilder provides an extensible interface that enables both the minimal interface shown in Figure 3.3 as well as more sophisticated input verification of the kind available in Nipype.

Figure 3.6 provides code using Nipype for the atlas-based segmentation example from Figure 3.1. Compared to the PipeBuilder pipeline in Figure 3.1, this pipeline requires substantially more overhead. While each step is only slightly longer in the Nipype version, the relationships between them require more overhead, both directly through the flow of data and also through the initial setup.

Additionally, most pipeline construction tools provide references to the data primarily through the steps that produce it. This is especially true of graphical tools, where visually, the edges serve to connect nodes. An example of this can be seen in Figure 3.6, where the intermediate values are never explicitly instantiated in the code, but are rather constructed implicitly through the connections between steps. This is in contrast to a typical program, which names intermediate outputs or assigns temporary variables. To facilitate this direct interaction with the data at the time of pipeline construction as well as when interacting with the data after pipeline completion, the `Dataset` abstraction in PipeBuilder effectively provides a second way to access the data as it flows through the processing pipeline.

The graphical interface provided by our tool resembles GUI-based medical image analysis pipeline building environments such as the LONI Pipeline (Rex et al., 2003). We emphasize that PipeBuilder computes this visualization automatically from the underlying code, enabling developers to use an expressive and flexible scripting environment for specifying pipelines and a rich visual interface for interacting with them. In addition, our tool's visualization of data and capacity to enable easy-to-use quality control set it apart from existing workflow visualizations.

■ 3.3.2 Integration with Nipype

Nipype is a well-established open source software project with a large user base and a rich collection of auxiliary features, including integration with medical image data storage systems such as XNAT (Marcus et al., 2007), the ability to distribute computation across different cluster systems, and a broader set of interfaces to commonly used medical image analysis tools. To take advantage of these features, we provide several components to integrate PipeBuilder and its visualization with Nipype.

Importing PipeBuilder pipelines into Nipype

The first such feature is the ability to import PipeBuilder pipelines into Nipype to take advantage of the feature set. Nipype provides a mechanism for constructing computational nodes that execute arbitrary Python functions. We construct a Python function that executes the command line associated with a `Command`. This dynamically constructed Python function can then be associated with the correct inputs and outputs using the metadata associated with the PipeBuilder `Command`. The command can then be used with Nipype just like any other node such as the ones in Figure 3.6. As the experimental research pipelines facilitated by PipeBuilder mature to stable algorithms, they can then be imported into Nipype using this mechanism to better take advantage of Nipype's additional features.

Interactive visualization for Nipype pipelines

The second such feature is the application of the visualization described in Section 3.2.4 to Nipype pipelines. While Nipype pipelines are constructed in code explicitly as graphs and represented in the same way, Nipype provides only a static visualization of the pipeline (Ellson et al., 2002). However, we provide a dynamic visualization based on the same client-server architecture as the PipeBuilder visualization. The client is largely the same, while the server accounts for the relevant differences between the structuring of the pipeline and intermediate data between PipeBuilder and Nipype. At the time of writing of this thesis, the interactive visualization will appear in the next release of Nipype, and has already been used by several users.

■ 3.4 User feedback and development

The development of PipeBuilder has been influenced by a range of user feedback, including a heuristic evaluation with human-computer interaction experts and some small usability studies with small scale usage by target users. A full user study to quantitatively and qualitatively evaluate the performance of users using PipeBuilder would be difficult, as proper evaluation would ideally measure the performance of users as they complete a full research project.

The usability studies and heuristic evaluation guided several decisions. In particular, the focus on enabling iterative improvement was driven by a need identified in early usability studies. These studies also identified a need for example-based documentation, as full documentation may not be readily available as the pipeline is being written.

Instead, reference pipelines with good usage patterns were helpful in constructing new ones. Finally, the visualization was heavily influenced by feedback. Most notably, the ability to aggregate information and perform quality control was driven by feedback.

■ 3.5 Summary

To enable the creation of computational pipelines for medical image analysis, we present PipeBuilder. Our work is driven by several key needs in this domain. First, experimentation and iteration are critical to developing pipelines, especially when working with clinical quality data. Second, large image collections require solutions that are both adaptable to account for heterogeneity in the data and robust to variations in quality. Third, visualization of a pipeline's computational structure, as well as data inputs and outputs, is invaluable in understanding both high-level organization of the system and the details of its results. Finally, while iterative development can produce pipelines that are fairly robust and adaptable, large collections of images, especially from clinical settings, will often present outliers that require some degree of manual intervention to identify and remove.

These needs inform the design and creation of PipeBuilder. We provide a range of features to enable the construction of robust pipelines. These include an intuitive interface for interacting with data, a simple interface to construct novel pipeline steps, a browser-based visualization that is computed automatically from code, and integration with existing pipeline creation tools to enable interoperability of both our visualization tool and PipeBuilder pipelines.

Chapter 4

Background on medical image analysis and white matter hyperintensity

In this chapter, we provide a necessary background for each of the applications described in the remainder of the thesis. We also briefly review each of the algorithmic components used in the applications.

■ 4.1 Cerebrovascular Pathology

We now turn to providing a background for the stroke and Alzheimer's applications presented in the thesis. In particular, we focus on cerebrovascular pathologies, as these motivate our analysis of the images.

■ 4.1.1 White matter hyperintensity

Magnetic resonance imaging (MRI) provides a means for imaging the brain in research and clinical settings, revealing a variety of different pathologies. In particular, white matter hyperintensity (WMH) is a broad category of pathology visible as bright (hyperintense) regions in T2-weighted FLuid-Attenuated Inversion Recovery (FLAIR or T2-FLAIR) MR images, as illustrated in Figures 4.1 and 4.2. While the precise cause and mechanism of WMH development is not fully understood, a great deal is known about its links to other clinical and demographic factors. For example, WMH has been shown to increase with age, as well as be linked to tissue damage, decline in cognitive function, structural brain changes, stroke, dementia, genetic markers linked to dementia and Alzheimer's, and indicators of vascular health such as hypertension (Debette, Markus, et al., 2010; Fazekas et al., 1993; Jeerakathil et al., 2004; Raz et al., 2012). Additionally, among patient cohorts afflicted with pathology, WMH burden has been shown to be linked to worsened clinical outcomes in diseases including stroke, Alzheimer's disease, and cerebral amyloid angiopathy (Rost et al, 2010a; Rost et al, 2010b; E. Smith et al.,

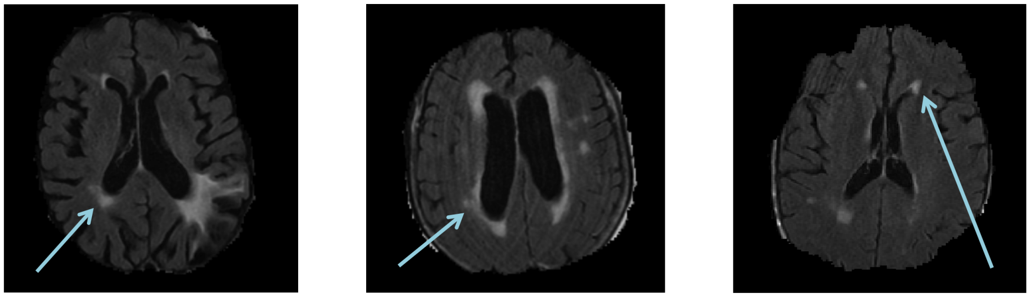


Figure 4.1: Sample two-dimensional axial slices of T2-FLAIR brain MRI showing white matter hyperintensity (highlighted by blue arrows) in varying degrees.

2002; Thanprasertsuk et al., 2014). In addition to quantity, the spatial distribution of WMH has been shown to differ in healthy and diseased patients (Gootjes et al., 2004). These factors motivate the explorations presented in Chapter 5 and Chapter 6 of this thesis.

White matter hyperintensity segmentation

Segmentation of white matter hyperintensity is a well-studied problem. It also presents several challenges: while WMH is characterized by high intensity, so are other brain structures such as ependymal tissue (ventricle lining) and the skull. Stroke lesions and other pathologies can sometimes appear bright in T2-FLAIR as well. This presents challenges in segmentation, as simple thresholding of normalized intensities will include these undesired structures.

Most segmentation techniques for WMH are either semiautomatic or automatic. Both categories typically involve preprocessing steps, particularly intensity normalization to ensure consistent segmentations. Bias field correction (Van Leemput et al., 1999; Tustison and Gee, 2009) is also often used to correct inhomogeneities in the image caused by acquisition artifacts.

Semiautomatic techniques vary in the degree of expert interaction required; fully manual approaches are rare. Many semiautomatic methods involve an automatic threshold within a expert-defined region of interest followed by adjustment and correction (Gurol et al., 2006; Y. Chen et al., 2006; Nandigam et al., 2007; Rost et al., 2010; Tanabe et al., 1997). Semiautomatic segmentation of WMH of this kind typically takes 10 to 30 minutes per patient scan. As the distinction between semiautomatic and automatic is a spectrum rather than a dichotomy, several methods exist that require less interaction, relying on expert input only for limited image-specific pre- and post-processing (Gibson et al., 2010; Zijdenbos et al., 1994).

Automatic techniques vary in complexity, and this variation typically depends on the

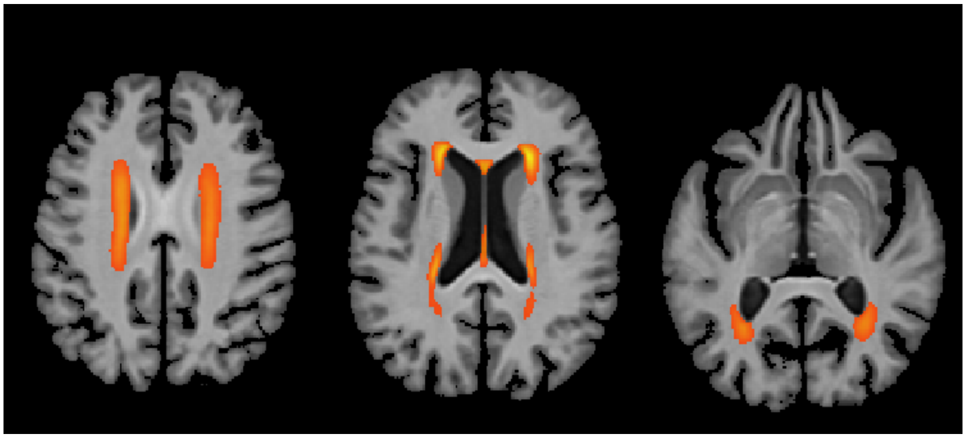


Figure 4.2: An average map of white matter hyperintensity (in color) overlaid on a template brain in three axial slices.

resolution of the input images. While higher resolution scans can typically be segmented using existing approaches (Admiraal-Behloul et al, 2005; P. Schmidt et al., 2012), lower resolution data presents a challenge for such algorithms. Elaborate pipelines have shown promise when applied to smaller sets of such low resolution data (Tsai et al., 2014; B. Yoo et al., 2014), motivating our use of such a pipeline in Chapter 5. The scalability of computational pipelines has also been demonstrated in large scale studies of segmentation of other pathologies (Zijdenbos et al., 2002).

■ 4.1.2 Stroke

Stroke is an acute cerebrovascular affliction characterized by insufficient blood flow to the brain leading to cell death. Strokes can be characterized into a number of subtypes. These are linked to patient outcome. In the study motivating this thesis, there are four classes of abnormal hyperintense regions we observe in the various modalities: leukoariosis, a particular type of white matter hyperintensity found in the periventricular (around the ventricles) region of the brain with a distinct spatial pattern; acute stroke, which has occurred no more than 48 hours before scan acquisition; chronic stroke, which has occurred in the past; and imaging artifacts, which appear hyperintense due to idiosyncrasies of the imaging acquisition process. In neuroimaging studies of stroke patients, several imaging modalities can be acquired; the analysis in this thesis focuses on three modalities of MR imaging: T1 scans, T2-FLAIR scans, and diffusion-weighted imaging (DWI) scans.

Each imaging modality measures different tissue properties and serves a different purpose in the context of the clinical and research goals of the study. In DWI images,

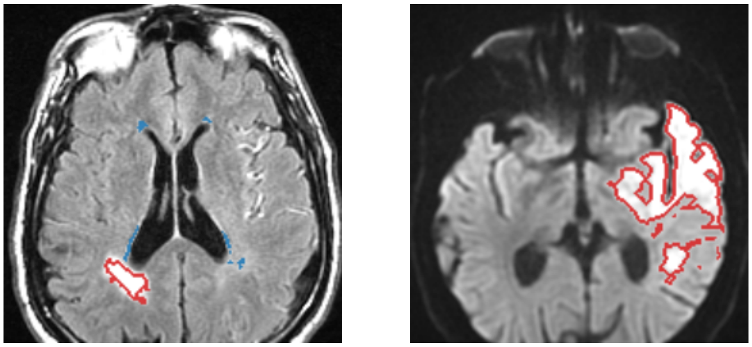


Figure 4.3: Illustration of various cerebrovascular pathologies in a T2-FLAIR scan (left) and a DWI scan (right). On the left, WMH is outlined in blue, while chronic stroke is outlined in red. Neither of these are visible on DWI. On the right, acute stroke is labeled in red.

	Bright on T2-FLAIR?	Bright on DWI?
Healthy tissue	No	No
WMH	Yes	No
Acute stroke	Yes	Yes
Chronic stroke	Yes	No

Table 4.1: A description of the various cerebrovascular pathologies relevant to stroke and their intensity profiles in T2-FLAIR and DWI scans.

stroke appears bright and, aside from miscellaneous artifacts, is the only such bright region. In T2-FLAIR images, white matter hyperintensity, especially leukoariosis, appears bright, as do both acute and chronic stroke lesions. T1 images typically do not provide high contrast for identifying any of the pathologies described above, but are often acquired initially as an anatomical localizer. Table 4.1 summarizes the relationship between the different modalities acquired and the pathologies visible in those modalities. Figure 4.3 illustrates these pathologies on representative T2-FLAIR and DWI images.

4.2 Medical image analysis algorithms for neuroimaging

In this section, we briefly review highlights from the relevant literature on the algorithms used in this thesis, focusing on the aspects and prior work most relevant to our analyses and derived insights.

4.2.1 Registration

Image registration is the process of aligning two different images in order to draw correspondences between them. Image registration techniques have been widely studied, and generally include a distance or similarity metric, a transformation model, and an optimization procedure (Hartkens et al., 2002; Rueckert and Schnabel, 2011; Viola and Wells III, 1997). Three of the most popular metrics used in registration are sum of squared differences (SSD), cross correlation (CC), and mutual information (MI). SSD and CC are used when the intensity distributions are directly comparable between the two images. MI is typically used for multimodal registration when the intensity profiles differ between scans, e.g., when registering a T1-weighted image to a T2-weighted image (Viola and Wells III, 1997). Optimizing over nonrigid transformations is usually only effective after an initialization based on simpler rigid or affine alignment. As smaller deformations are easier to estimate accurately than large ones, registration between similar images (e.g., images of the same patient at different time points), tends to be more robust. This motivates the strategy used in many popular registration algorithms (Vercauteren et al., 2008; Avants et al., 2011a) of iterative registration performed as a sequence of small deformations.

4.2.2 Atlases and atlas construction

A fundamental component of population analysis in medical image computing is the use of deformable templates, or atlases. These, along with a technique for computing deformations or correspondences between the template and images in the population, are essential to population analysis, providing a common coordinate system across patients and modalities in a study and enabling data fusion across patients and modalities (Fonov et al., 2009; Gee et al., 1993; Grenander and Miller, 1998; Mazziotta et al., 1995; Talairach and Tournoux, 1988). While some approaches use a single brain as a template (Warfield et al., 2002), atlases constructed from multiple brains can provide a better representation of average anatomy (Bhatia et al., 2004; Fonov et al., 2011; Guimond et al., 2000; Joshi et al., 2004).

The simplest algorithms for atlas construction are based on iterating between two steps until convergence. In the first step, a collection of images is aligned to a common coordinate frame using an image registration algorithm. In the second step, the aligned images are combined to create a new template for a better-defined common coordinate frame. While the registration used is typically nonrigid, the degree of regularization of this registration affects the resulting atlas quality: less regularized warps produce sharper atlases whose anatomical regions are more precisely defined, while more regularized (or even affine or rigid) registrations produce blurrier atlases that may capture a broader range of anatomy at the cost of less precisely defined anatomical boundaries to which to align images (Yeo et al., 2008).

4.2.3 Longitudinal analysis

Longitudinal analysis is a well-studied area in medical image computing that aims to characterize a population both in terms of inter-subject variability across a cohort, and in terms of within-subject variability over time using longitudinal data. Typically, at least two scans per patient are available. As with most population analysis studies, accurate registration is also a prerequisite here. Many approaches exist that find correspondences between an atlas and each template image by aligning each scan to the atlas separately (L. Wang et al., 2003). Others note that the changes between the first longitudinal scan and subsequent scans of the same patient are likely to be small, and model these explicitly (Ehrhardt et al., 2008; Qiu et al., 2009). Many methods are designed for the case of registering exactly two scans per subject (Gerig et al., 2006; Aljabar et al., 2008). Most closely related to the analysis in this thesis is work by Durrleman et al., 2013, which models both a low-dimensional correspondence between subsequent scans from the same subject as well as a more complex correspondence between pairs of different subjects and subjects and an atlas.

In the next chapter we present a case study that employs PipeBuilder to create a computational pipeline for WMH analysis and to apply this pipeline to a large cohort of stroke patients.

Chapter 5

Case study 1: WMH burden in stroke patients

In this chapter, we demonstrate the application of PipeBuilder to analysis of pathology in a large cohort of stroke patients. We present a study of white matter hyperintensity analysis in this multi-site cohort. The analysis is enabled by our processing pipeline constructed using PipeBuilder. In particular, we demonstrate two registration pipelines with different goals, and describe our algorithm to automatically segment WMH. This enables our analysis of the relationship between WMH and age in this population.

This chapter is organized as follows. Section 5.1 provides two approaches to registration of large, multimodal collections of clinical quality images. Section 5.2 describes our framework for analysis and segmentation of white matter hyperintensity in scans that have been aligned to a common coordinate frame. Section 5.3 details the role of PipeBuilder in constructing pipelines for our analysis. Section 5.4 describes the data from a multi-site neuroimaging study of stroke patients. Section 5.5 demonstrates our methods on this study.[1]

■ 5.1 Registration

Registration of low resolution clinical images of the brain presents several challenges. For example, patient images contain many irrelevant structures: our goal is brain analysis, but the images include the skull and large portions of the neck, and may even crop structures of interest, as illustrated in Fig. 5.1. The optimization procedure in typical registration algorithms treats all regions uniformly, and aligning these bright, highly variable structures may drive the registration and may result in an inaccurate transformation of the brain. Another concern is caused by the fact that images in large studies are acquired at multiple sites with different scanners and different acquisition parameters, both within sites and across sites in multi-site studies. Therefore, the range

[1] The work presented in this chapter has been conducted in collaboration with Adrian Dalca from the Computer Science and Artificial Intelligence Laboratory at MIT.

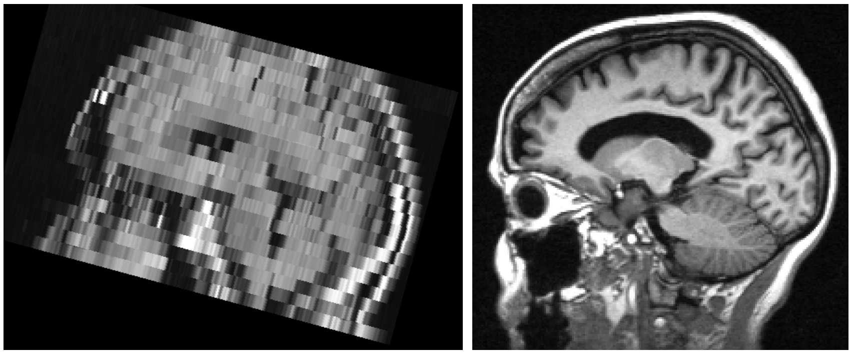


Figure 5.1: A sagittal slice of a representative low resolution T2-FLAIR scan (left) compared to the sagittal slice of a representative high resolution T1 scan (right). The top of the T2-FLAIR image is cropped, and due to the thick slices, there is no clear intensity boundary between the brain and the skull.

of values and the intensity distributions of specific tissue can vary greatly even within images of the same modality. While these challenges have been studied extensively in the context of high resolution consistent research scans, in this section we address these challenges by proposing general strategies applicable to low resolution clinical images.

■ 5.1.1 Multimodal registration using a T1 template

In this section, we present a registration pipeline that aligns multimodal patient images to a template of a particular modality. This work is motivated by the fact that although the analysis we wish to perform is on T2-FLAIR images, commonly used templates such as the Talairach average template (Talairach and Tournoux, 1988) and the FreeSurfer brain atlas (Fischl, 2012) are only available in one modality, T1-weighted. Another commonly used template, the MNI152 brain atlas (Mazziotta et al., 1995), is available in multiple modalities but not in T2-FLAIR, which is our modality of interest in this work. Using one of these templates or a similar one provides distinct advantages, as they are well-annotated with anatomical labels. More generally, the specific registration techniques, as well as the underlying insights in this section, are relevant to any analysis of multimodal data in which the modality of the data being analyzed and the modality of an existing template differ. The work presented in this section was published in Sridharan et al., 2013.

Atlas construction

We use an atlas constructed from 39 T1-weighted brain MRI scans and corresponding manual delineations that are part of the FreeSurfer brain atlas (Daly et al, 2000; Johnson et al, 1998; Killiany et al, 2000). The 39 subjects span a wide age range, reflect

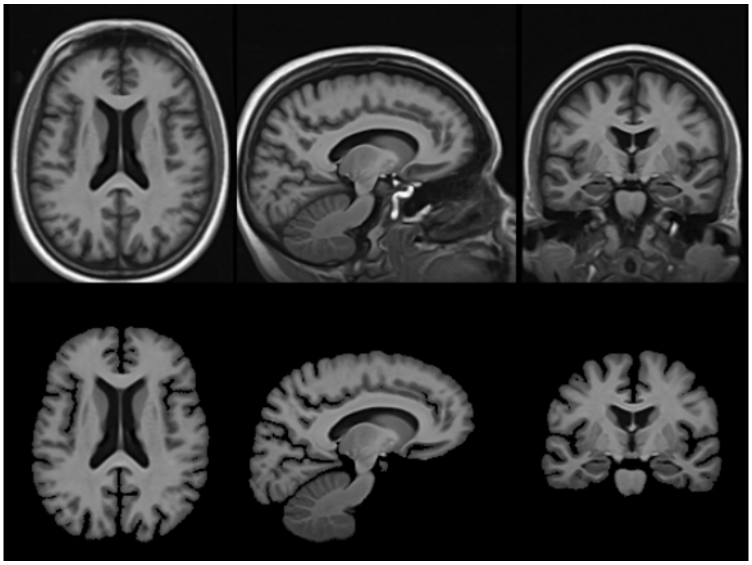


Figure 5.2: The T1-weighted atlas used by the registration pipeline in Section 5.1.1. The top row shows the atlas including the skull, which is useful for affine alignment of images that have not been skull stripped; the bottom row shows the same image with manual segmentations of the brain used to remove the skull, enabling more accurate registration in the brain.

significant anatomical variation, and include some Alzheimer's patients. Figure 5.2 presents the atlas.

Intra-modal initialization with MI

When registering images of the same modality, the standard practice of first computing an initial rigid registration (i.e., rotation and translation only) is relatively insensitive to the problem of extraneous structures. Inconsistent intensity distributions in images of the same modality in clinical datasets render the usual intra-modality metrics such as cross-correlation (CC) and sum of squared differences (SSD) ineffective for alignment since the assumption of direct intensity matching for the same tissue type across different images is violated. Standard methods for matching intensity profiles, such as histogram equalization, cannot be used either, since they would be dominated by highly variable non-brain regions such as the neck. We employ mutual information (MI) in performing this rigid registration since the difference of tissue intensities between these images is more similar to the difference of tissue intensities between images of different modalities. We build on this initial registration to solve the problems of inconsistent field of view and intensity profiles.

Skull stripping and brain masking

Since Neuroimaging analysis typically focuses on the brain, we seek an accurate transformation in the brain, and restrict the region where the registration metric is evaluated accordingly. In research-quality images, skull stripping or brain mask extraction is achieved via any number of available techniques, including registration to a template, deformable balloon models, watershed-based methods that assume that the brain consists of a single connected component, and graph-based segmentation algorithms (Boesen et al., 2004; Iglesias et al., 2011; S. M. Smith, 2002). Most of these methods only work on T1-weighted MR images. Additionally, many model-based algorithms that assume a particular shape or spatial model fail on low resolution data, where the low resolution of the image may result in unintended spatial contiguity effects. The registration method described in this section employs a simple rigid registration to the atlas, which we use to propagate a brain mask from the atlas via the estimated rigid transformation. While not a perfect brain mask, it enables intensity correction and constrains the subsequent nonrigid registration to a region that reasonably approximates the brain.

Intensity correction

In our experiments with clinical images of stroke patients, MI failed when used in nonrigid registration, resulting in inconsistent deformations that did not match the images. Differences in intensity profiles of patient images prevent us from using intensity-based measures such as CC and SSD directly. Using the approximate brain mask, we adjust the intensity separately for each image to solve this problem. Histogram equalization (Hummel, 1977) still cannot be used due to the approximate nature of the brain mask and variable intensity profiles, illustrated in Fig. 5.3. We choose to restrict our intensity correction to global scaling. Specifically, we match the intensity of the white matter while not altering the shape of the intensity profiles. As one of the largest structures in the brain, the white matter is important to match well between the two images in registration. We estimate the mode of white matter intensity for each patient as the mode of the component with higher intensity in a two-component mixture model for intensity values within the brain mask.

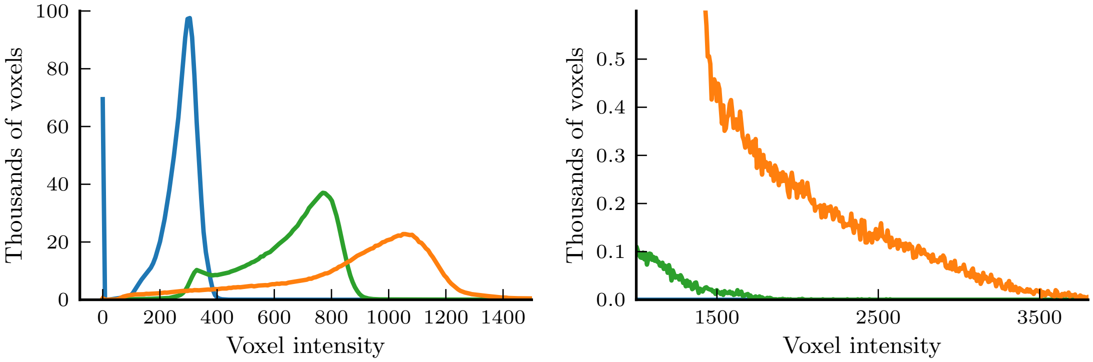


Figure 5.3: Voxel intensity histograms of T1-weighted images from three different patients (shown in three different colors), illustrating typical differences in intensity distributions within the approximate brain mask obtained via rigid registration from the atlas. Left: intensity histograms. Right: difference at the high end of intensity values (see axes).

Non-rigid registration

Once the image intensity distribution of the patient image has been matched to that of the atlas image, non-rigid registration can then be performed with CC as a metric. In order to prevent non-brain structures from dominating the optimization, we continue to use the approximate brain mask in computing this registration. Once the registration is concluded, we propagate a more accurate mask of the brain to be used for multimodal registration within each patient.

Intra-patient multimodal registration

In order to align other modalities (such as T2-FLAIR and DWI in the stroke study) into the atlas coordinate system, we first estimate the rigid transformation to the atlas-modality image (in our case, this is T1) using MI, and compose it with the final nonrigid transformation between the patient and the atlas.

■ 5.1.2 Multimodal registration using a T2-FLAIR template

In this section, we present an alternative method for registration based on synthesis of an atlas from low resolution data. The pipeline described in this section is more broadly applicable, as it provides for a method for construction of the atlas from data. The work presented in this section was used in Dalca et al., 2014 as well as in subsequent analysis. We present a pipeline for registration to a common coordinate frame using an atlas constructed from that modality, as well as a description of our atlas synthesis

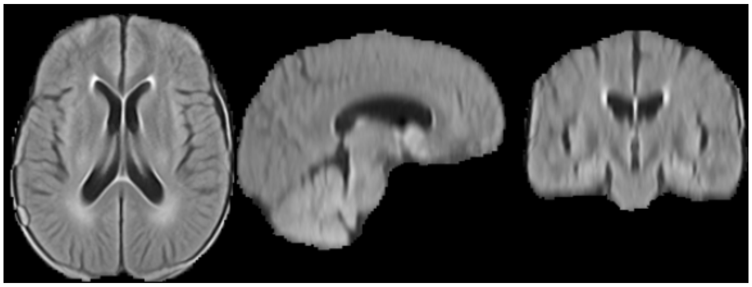


Figure 5.4: The initial T2-FLAIR atlas constructed from low resolution stroke patient images. Note the white matter hyperintensity and interpolation artifacts visible near and in the ventricles.

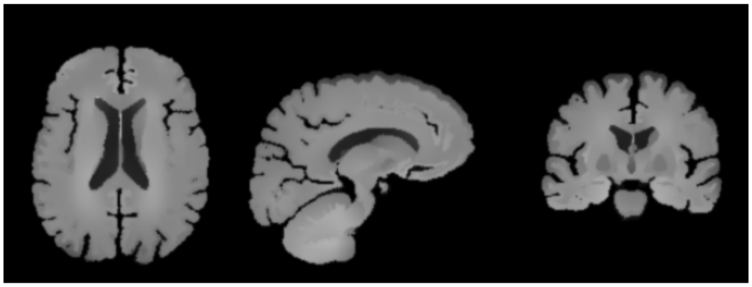


Figure 5.5: The synthesized T2-FLAIR template used for the registration pipeline of Section 5.1.2.

technique.

Construction of T2-FLAIR atlas

Due to the low resolution of our data, applying a simple atlas construction as described in Section 4.2.2 to the clinical scans in this data result in atlases that are blurrier in one direction than the others. In particular, the typical ratio of 7 between the in-plane resolution and the slice thickness of the T2-FLAIR images in the stroke study led to atlases with interpolation artifacts, as illustrated in Figure 5.4. In contrast to the atlases constructed from high resolution data shown earlier in Figure 5.2, the anatomical structure of the atlas is incorrect. Additionally, we have no manual labels of the anatomical structures are available for the low resolution images used to construct the atlas. In order to solve these two problems, we computed a registration to the

T1 atlas created earlier. Due to the differing modalities between the two images, we used MI. To counteract the inaccuracies due to the metric, since this was only a single registration, we swept over a broad range of parameters and chose the parameter that produced the registration we determined to have the best appearance visually. Given this mapping, we used the fact that within each anatomical region, image intensities are fairly homogeneous. Motivated by this, we computed a new image in the coordinates of the T1 atlas, where for each anatomical label, we used the intensities of the warped rough T2-FLAIR atlas, but blurred the values within each label while not blurring across label boundaries. The blur kernel was chosen by qualitatively evaluating its performance on registration to 10 randomly chosen images. The resulting atlas is shown in Figure 5.5: while visually somewhat smoother than a typical image, this image still maintains sharp gradients. We found in our experiments that this produced more accurate registrations, likely achieving a better tradeoff between registration smoothness and image sharpness (Yeo et al., 2008).

Preprocessing

As with the pipeline described in Section 5.1.1, accurate registration depends on skull removal and intensity correction.

Most of the T2-FLAIR images in our data have higher contrast than the corresponding T1-weighted localizer images. Motivated by this, and driven by further exploration of existing algorithms, we can successfully skull strip the majority of the T2-FLAIR images in our collection using ROBEX (Iglesias et al., 2011). As the majority of skull stripping algorithms are only designed to work with T1-weighted images, this algorithm performed the best among the few that work with images of other modalities.

As with the first registration pipeline, variable amounts of pathology in the images presented additional challenges for intensity normalization, a critical step to ensure robust registration with intensity-based metrics. After investigating a variety of algorithms, we normalized the intensities using global scaling as before, this time using mean shift clustering to align the modes of intensity (Mayer and Greenspan, 2009). We could not reliably correct the bias field due to the presence of WMH, stroke, and other pathologies.

■ 5.1.3 Evaluation of registration

Verifying registration quality is an important prerequisite to further analysis, as images that fail to register should be excluded from further analysis. Since visual inspection is difficult to perform for thousands of patients, we employ automatically computed measures of registration quality to detect when registration failed. We construct a (voxelwise) median image of registered patients for each modality in the atlas space, compute SSD of each patient from this median image within the brain mask, and isolate patients whose measures are substantially higher than the rest using the Tukey fence (more than 1.5 times the interquartile range above the third quartile) (Tukey, 1977). This threshold was chosen based primarily on visual inspection: most images above the threshold were misaligned, while most images below the threshold were reasonably well-aligned.

5.2 Segmentation and analysis of white matter hyperintensity

5.2.1 Segmentation

We employ MAP classification to label hyperintense voxels within a region of interest. In particular, we use the same framework to automatically segment WMH in T2-FLAIR images and stroke in DWI images, as described in Section 4.1.2.

Given intensity $I(x)$ at voxel x, we choose label $L(x) \in \{H, \bar{H}\}$ (where H represents hyperintense regions and $\bar{H}$ represents healthy tissue) to maximize the posterior probability of the label $p(L(x)|I(x))$:

$$
\begin{aligned}
L^*(x) &= \operatorname*{arg\,max}_{L \in \{H, \bar{H}\}} p(L \mid I(x)) && (5.2.1)\\
&= \operatorname*{arg\,max}_{L \in \{H, \bar{H}\}} p(I(x) \mid L)\, p(L). && (5.2.2)
\end{aligned}
$$

We use intensity-corrected T2-FLAIR images as described in Section 5.1.2 to avoid inconsistent intensities. As discussed in Section 4.1.2, other pathology such as stroke can also appear as hyperintense on T2-FLAIR images. In order to compensate for this effect, we first segment stroke and other artifacts using hyperintense voxels in DWI using Equation (5.2.1). This provides an exclusionary mask of voxels to remove from the analysis of WMH in T2-FLAIR images. We note that due to the presence of artifacts in these images, the automatic segmentations of DWI scans contain both stroke and artifacts. As WMH is the quantity of greater clinical interest, here we focus our analysis on WMH rather than analyzing both stroke volume and WMH volume.

We use 50 patient images to construct the likelihood models $p(I|L = H)$ and $p(I|L = \bar{H})$ as histograms of intensity for WMH in T2-FLAIR and stroke in DWI. These training images were visually inspected to have accurate manual segmentations, and intensity normalized as described earlier.

5.2.2 Analysis

While the overall WMH *volume* of each patient can be compared and analyzed from just the manual segmentations for each patient, we use the registration framework to evaluate and visually inspect the *spatial distribution* of WMH and to understand its evolution across the brain as a function of age. Since WMH volume varies dramatically across different patients, we choose to first cluster the patients into more homogeneous sub-populations and then investigate the change of WMH distribution with age separately in each sub-population.

We use a two-component regression mixture model to capture variability in WMH volume growth (Jacobs et al., 1991; Quandt and Ramsey, 1978). Each mixture component is characterized by a different dependency of WMH burden on age. To determine the assignment of patients to sub-populations associated with components, we alternate between (1) assigning the cluster membership of each patient and (2) estimating the regression coefficients for WMH volume as a function of age in each cluster, until convergence.

Formally, let v_i and z_i be the scalar total WMH volume and cluster assignment of patient i ($i \in \{1, \dots, N\}$) respectively. We let X_i be a p-dimensional feature vector associated with patient i. Specifically, we use age and a constant to account for the intercept (i.e., $p = 2$). Let $\boldsymbol{v}$ be the vector of all volume values and $\boldsymbol{X}$ be the $N \times p$ matrix of features. We assume i.i.d. multinomial priors for cluster membership. Given p-dimensional regression coefficient vectors $\boldsymbol{\beta}_c$ for each cluster c and fixed variance σ^2, we assume that WMH volume v_i in patient i is normally distributed with mean $X_i\boldsymbol{\beta}_c$ and fixed variance σ^2:

$$v_i = X_i \beta_c + \epsilon_i, \quad \text{where } \epsilon_i \sim \mathcal{N}(0, \sigma^2).$$

In order to estimate the parameters $\boldsymbol{\beta}$, we use a hard-assignment EM variant, alternating until convergence between the E-step that computes the cluster assignments:

$$z_i = \arg\min_c ||v_i - X_i \boldsymbol{\beta}_c||_2^2 \,, \tag{5.2.3}$$

and the M-step that solves for each $\boldsymbol{\beta}_c$ using standard least-squares linear regression:

$$\boldsymbol{\beta}_c = (\boldsymbol{X}^T Z_c \boldsymbol{X})^{-1} \boldsymbol{X}^T Z_c \boldsymbol{v}, \tag{5.2.4}$$

where Z_c is a diagonal binary matrix; $Z_c(i,i) = 1$ if $z_i = c$. The resulting algorithm is similar to k-means clustering.

Within each cluster, we use Nadaraya-Watson kernel regression (B. Davis et al., 2007; Nadaraya, 1964; Watson, 1964) on the WMH label maps to visualize representa-

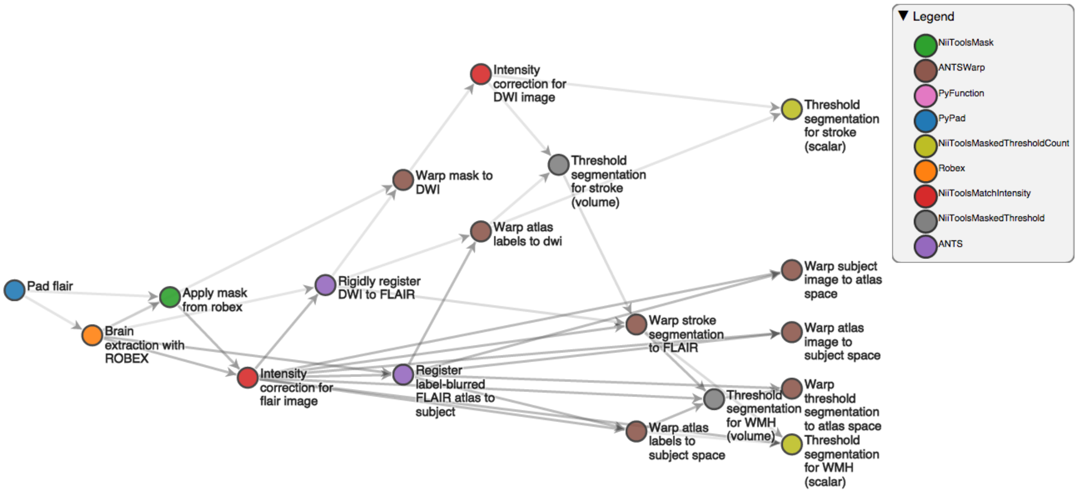

▼ Legend
NiiToolsMask
ANTSWarp
PyFunction
PyPad
NiiToolsMaskedThresholdCount
Robex
NiiToolsMatchIntensity
NiiToolsMaskedThreshold
ANTS
Intensity correction for DWI image
Threshold segmentation for stroke (scalar)
Threshold segmentation for stroke (volume)
Warp mask to DWI
Warp atlas labels to dwi
Pad flair
Apply mask from robex
Rigidly register DWI to FLAIR
Warp subject image to atlas space
Brain extraction with ROBEX
Warp stroke segmentation to FLAIR
Warp atlas image to subject space
Intensity correction for flair image
Register label-blurred FLAIR atlas to subject
Threshold segmentation for WMH (volume)
Warp threshold segmentation to atlas space
Warp atlas labels to subject space
Threshold segmentation for WMH (scalar)

Figure 5.6: Automatic visualization of the pipeline for registration and segmentation of T2-FLAIR scans. This pipeline uses the registration strategy with a T2-FLAIR template as described in Section 5.1.2. Note the DWI segmentation steps (top center) and the stroke segmentation steps (bottom right).

tive images $I_c(t)$ for each cluster c:

$$I_c(t) = \frac{\sum_{i=1}^{N} Z_c(i,i) K_h(t - t_i) I_i}{\sum_{i=1}^{N} Z_c(i,i) K_h(t - t_i)}, \tag{5.2.5}$$

where t is the age of interest, N is the number of patients, I_i is the WMH label map of patient i warped into atlas space, and $K_h(\cdot)$ is a Gaussian kernel function with standard deviation h and mean 0. The standard deviation of the kernel is determined assuming approximate normality of the underlying distribution, with a robust median absolute deviation used instead of the standard deviation (Bowman and Azzalini, 2004).

Intuitively, a representative WMH image is a weighted average of all WMH label maps, with patients close to age t contributing more to the average. Visualizing representative images helps understand the progression of the disease with age across the two clusters and across sites.

■ 5.3 Application to multi-site study and facilitation by PipeBuilder

Figure 5.6 presents the visualization of the pipeline created by PipeBuilder. The visualization shows the parallel structures that register and segment both T2-FLAIR and DWI scans. In addition, the visualization highlights the role of preprocessing, as the output of the intensity correction step (red) is used extensively throughout the remainder of the pipeline.

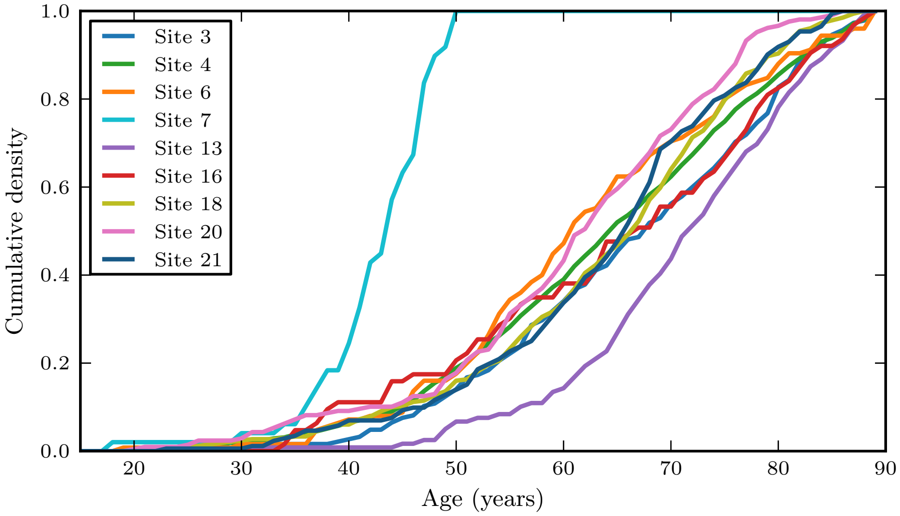


Figure 5.7: Age distributions for each site in the stroke study. Most of the sites follow the same distribution, while site 7 (teal) and site 13 (purple) have younger and older populations respectively.

In the course of our analysis, we examined the histograms of white matter hyperintensity volume for each site separately and detected that site 7 had noticeably lower volumes due to a younger population. To address this variability, we constructed a separate atlas from this population. The modular nature of our proposed pipeline system made this change easy to implement, requiring minor adaptation of the atlas construction step and the segmentation pipeline to use the new template for this specific site.

Additionally, the ability to easily identify outliers using PipeBuilder, particularly during the registration evaluation step, greatly simplified the exclusion of those subjects from further analysis. In particular, identification of outliers was immediately followed by visual inspection of several subjects to ensure the validity of our registration quality measure.

■ 5.4 Stroke data

In this section we outline the acquisition protocol used to collect the data used in this study, and describe in detail the imaging data available across each site for the multi-site study. The data described in this section are provided by the International Stroke Genetics Consortium (ISGC) Stroke Genetics Network (SiGN) study (Meschia et al., 2013).

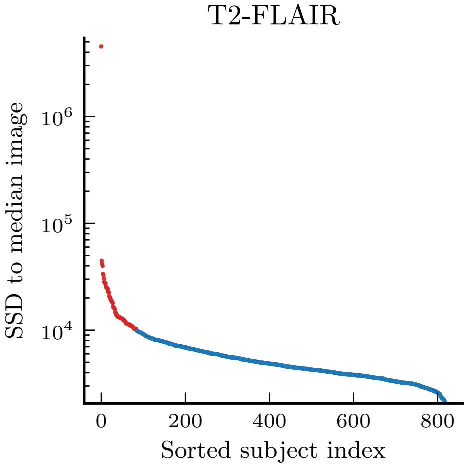


Figure 5.8: Registration quality results for the T1-based pipeline applied to site 4. Outliers, which were removed from further analysis, are shown in red. Here, the difference is measured only within the mask of the brain from the atlas image.

Each patient in the study has one to three images depending on modalities acquired. The available modalities are a T1 localizer scan (0.8594 × 0.8594mm in-plane, slice thickness 5–7mm, sagittal slices), a T2-FLAIR scan (1 × 1mm in-plane, slice thickness 5–7mm, PROPELLER sequence sometimes used if the patient moved, axial slices), and a diffusion-weighted imaging (DWI) scan (at least 6 directions, b-value 1000 s/mm^2, 1 × 1mm in-plane, slice thickness 5–7mm, axial slices).

We analyze the images provided by nine sites. Figure 5.7 illustrates the age distribution for each site. While most of the sites exhibit similar distributions, site 7 has a substantially younger population than the rest of the sites, while site 13 has a slightly older population than the rest of the sites.

For site 4, manual segmentations are available for WMH in T2-FLAIR and for stroke in DWI for 825 and 432 patients respectively.

5.5 Results

5.5.1 Registration evaluation

Figure 5.8 shows the registration quality results for the first registration pipeline applied to data from one site (site 4). Our procedure identifies 86 of 819 manually segmented T2-FLAIR scans as outliers, leading us to exclude them from further analysis.

Figure 5.9 shows the registration quality results on T2-FLAIR images for the second pipeline applied to all sites. We see that the results are consistent across sites. Com-

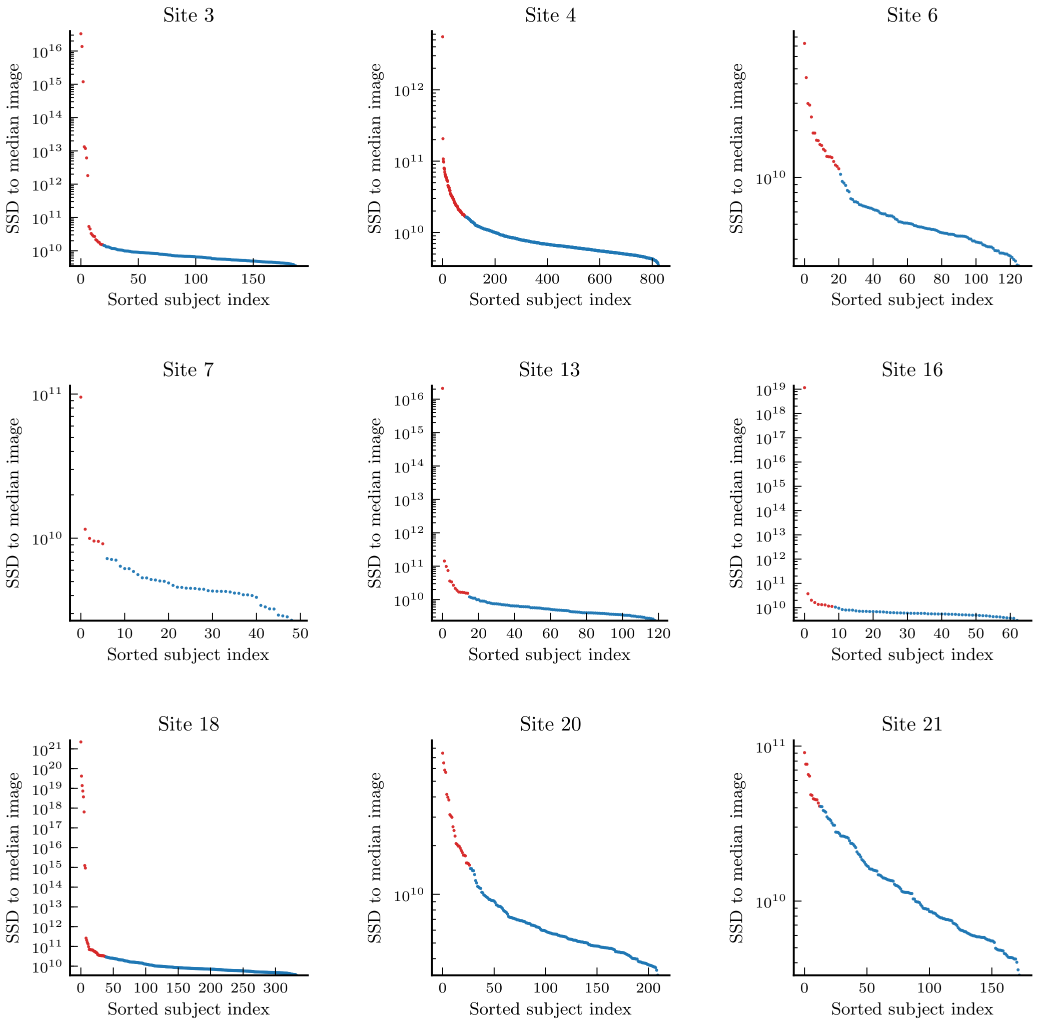


Figure 5.9: Registration quality results for the T2-FLAIR pipeline applied to all sites. Outliers, which were removed from further analysis, are shown in red. The difference is measured across the entire image.

paring the first graph of this figure to Figure 5.8 shows that the second registration strategy, which works directly with the T2-FLAIR images, achieves better performance in registering T2-FLAIR images than the initially proposed method, which is based on alignment of T1 localizer scans. This is primarily due to two factors. First, the slightly higher contrast of the T2-FLAIR images in this dataset compared to the T1 images leads to more accurate registrations when these images are used as the input. Second, this strategy requires only one registration rather than two. This prevents error

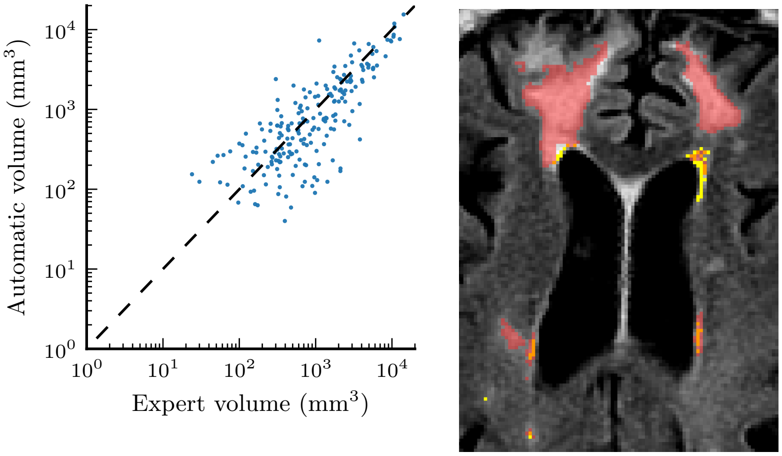


Figure 5.10: (Left) Agreement between manual and automatic segmentations of WMH in site 4. Each point represents one patient, and the dashed line ($y = x$) corresponds to perfect agreement. (Right) An example of strong disagreement between the automatic and the manual segmentations. The large high intensity region in the frontal lobes bilaterally represents chronic territorial strokes that, although hyperintense, are excluded by the expert since they are not of interest in WMH volume calculations.

from compounding across multiple registrations and warps. This is a particularly useful feature of the second registration pipeline, as our goal is segmentation and analysis of pathology visible primarily on T2-FLAIR images.

■ 5.5.2 Segmentation evaluation

Figure 5.10 illustrates the results of the agreement of the manual and automatic WMH segmentation for site 4. We observe that the automatic segmentation is close to the manual one. In some cases, our algorithm oversegments relative to the manual segmentation. Investigating these patients reveals cases like the one shown in Figure 5.10, where during manual segmentation experts excluded large regions determined to be attributable to chronic stroke lesions, as described in Section 4.1.2. The registration methods described in this thesis have enabled new lines of work that address this problem (Dalca et al., 2014).

■ 5.5.3 Multi-site segmentation

Figure 5.11 shows the distribution of WMH for all of the sites in the study. As expected from the age distributions in Figure 5.7 and the positive relationship between WMH burden and age, the distribution for site 7 is shifted noticeably to the left. As discussed in Section 5.3, this motivated our use of a separate atlas due to increased registration

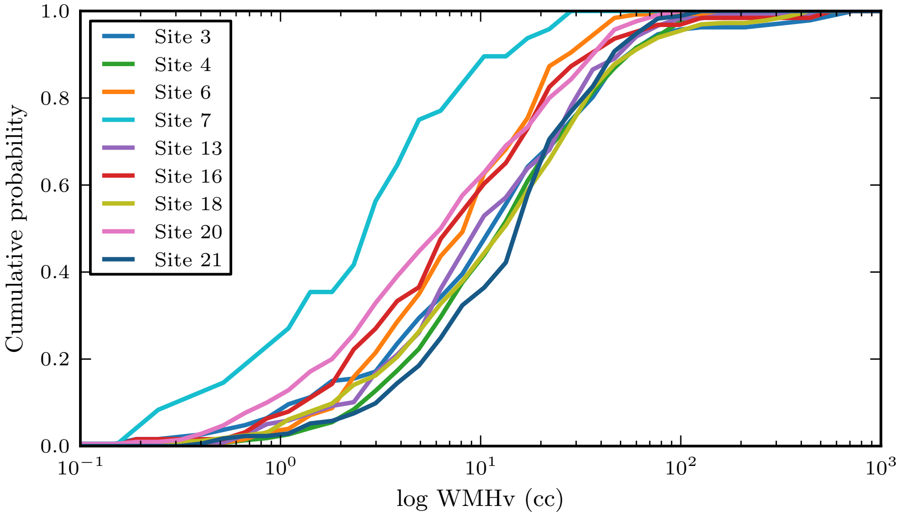


Figure 5.11: Computed volumes of WMH from the automatic segmentation algorithm. While most of the sites follow a similar distribution of WMH burden, patients from site 7 tend to have less. Volumes shown for site 7 are based on the site-specific atlas.

error relative to the atlas.

This confounding effect of age on WMH volume also motivates our analysis of the relationship between the two across the sites. Figure 5.12 shows the distributions for each site within a particular set of age ranges. We see that within each age group, site 7 is no longer an outlier. While its distribution of WMH in the 15–40-year-old group and the 40–50-year-old group are slightly lower than the other distributions, the degree of difference is not as pronounced as in the overall distribution. Indeed, there are no patients over the age of 45 within this site. Within each age group, the WMH distributions across the different sites are fairly consistent.

Figure 5.13 shows the spatial distributions of WMH across the different sites. While the overall pattern remains similar across sites, the shape varies considerably. As expected from their heavier-tailed distributions above, sites 3 and 18 present a noticeably greater spread of average distribution across the brain despite having a similar age distribution to the rest of the sites. This is particularly true in the posterior regions behind the ventricles (center slice, bottom of image), where these two sites show noticeably more pathology compared to the other sites. In contrast, the other sites show a slightly greater concentration of WMH near the front of the ventricles (center slice, top of image). Finally, as expected, the distribution of WMH in patients from site 7 is very small due to the younger age of patients from this site.

We can more directly analyze the relationship between WMH volume and age by using the kernel regression procedure described above. Figure 5.14 illustrates the overall relationship between these two. The lines are the result of applying kernel regression as described in Section 5.2.2 to each site's full data. As before, we see that the sites are largely similar. Site 7 has a slightly lower amount of WMH among younger patients, and has no older patients.

Figure 5.15 shows the result of applying the two-component regression mixture model to the data from all sites. The method identifies a large group of patients for whom there is little dependence of WMH on age, and a larger group for which it starts higher and grows substantially. This larger group exhibits many of the spatial patterns seen in Figure 5.13, with larger growth through the posterior regions of the brain. Registration enables direct comparison of the segmentations across patients and sites, confirming the spatial distribution seen in the average maps. While these groups have distinct spatial patterns, the boundary between them is not well-defined. We believe detailed investigation into clinical and genetic factors can provide further insight into these patterns.

■ 5.6 Summary

We presented an application of PipeBuilder to a study of WMH in stroke patients. PipeBuilder facilitated the experimentation and refinement necessary to successfully process this large clinical image collection. In addition, the adaptability of our PipeBuilder pipeline facilitated the extension of our method to a multi-site study of thousands of patients. We adapted the pipeline to successfully register images from all sites, including one substantially different from the rest. We also demonstrated a visualization of the pipeline as well as its results, highlighting the structure of the computational process. This pipeline enabled us to study the relationship between WMH and age in a large cohort, where we found broadly similar trends across the subpopulations.

In the next chapter, we take the analysis of WMH and age one step further by applying PipeBuilder to study longitudinal changes in a group of Alzheimer's patients and healthy controls.

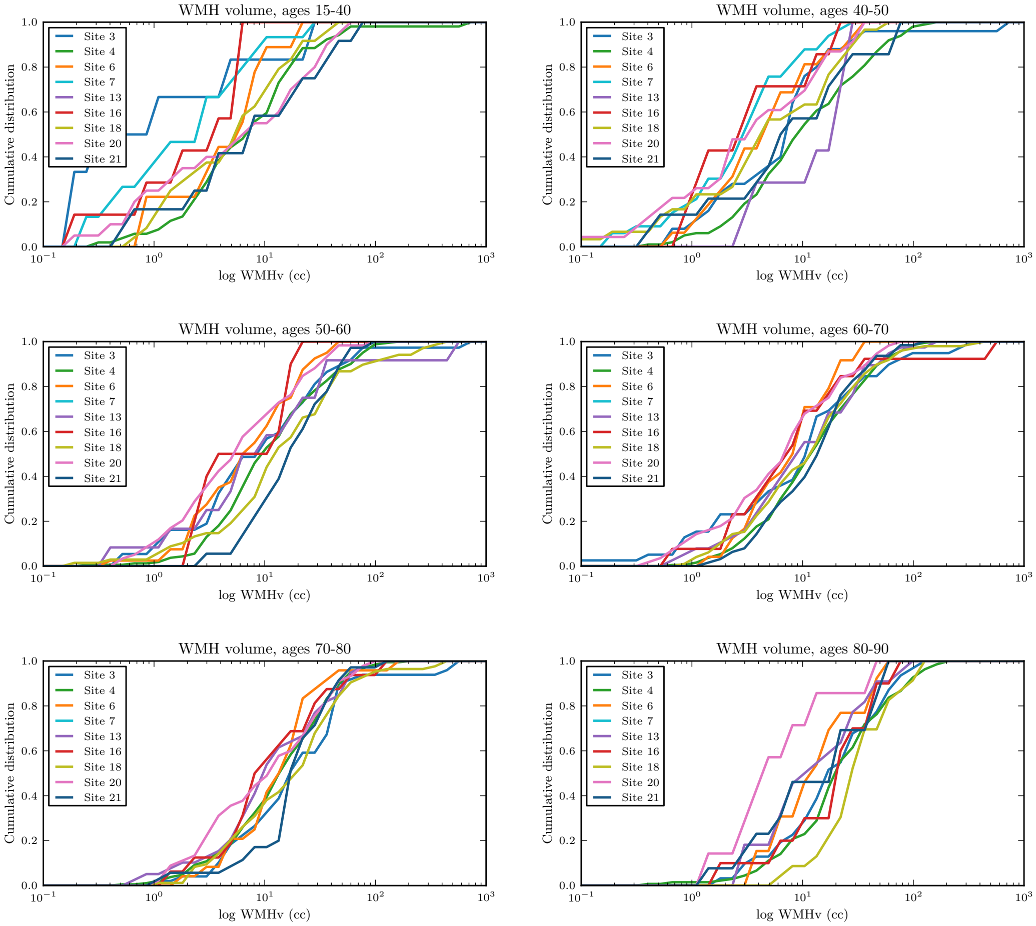


Figure 5.12: WMH burden distributions for each site within different age groups. The overall distributions shift to the right consistently with age, and within a particular age group, the sites are fairly consistent. Sites 3 and 18 (blue and red, respectively) tend to have heavier-tailed distributions than the other sites.

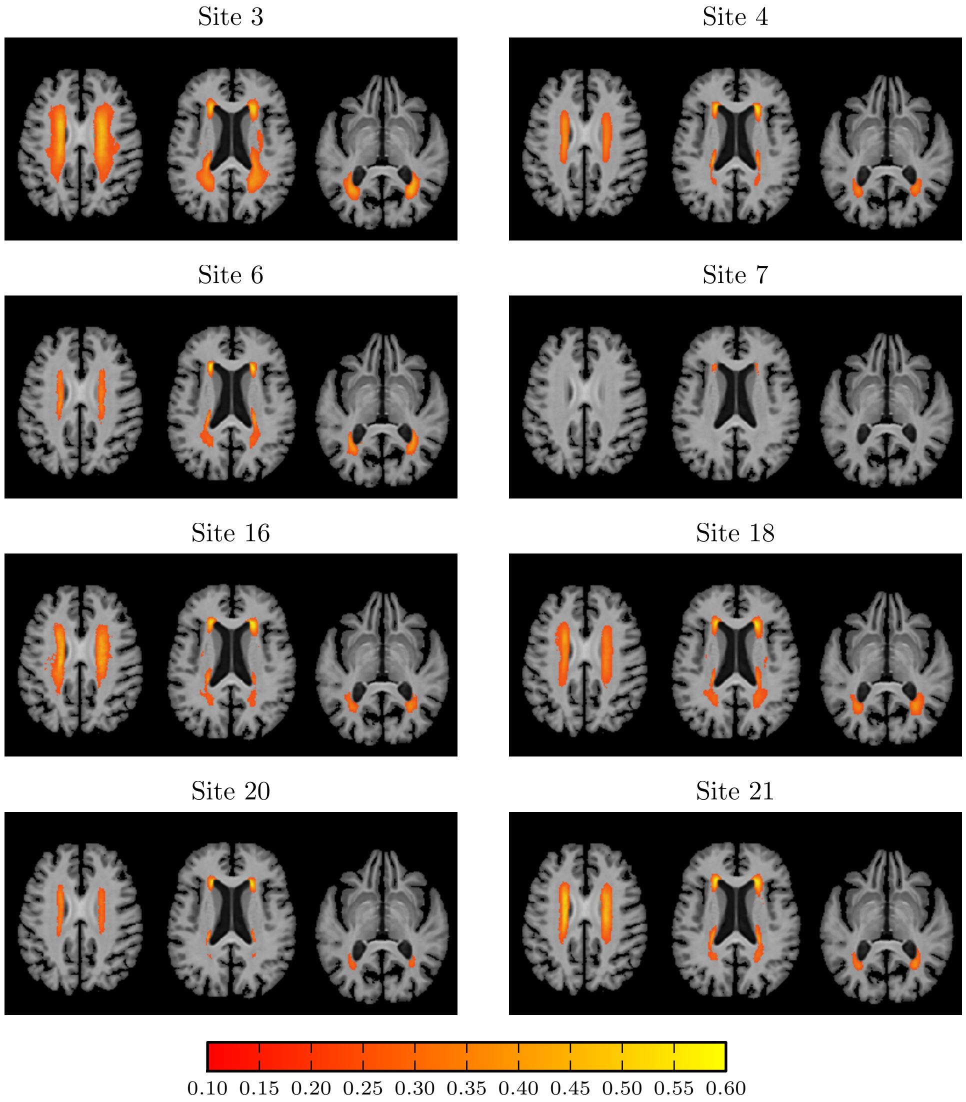


Figure 5.13: Average maps of WMH for each site. Regions where 10% (red) to at least 60% (yellow) are shown in color overlaid on the atlas. For site 7, where a different atlas was used, the resulting segmentations are then mapped back into the common atlas using the atlas-atlas registration for comparison across sites.

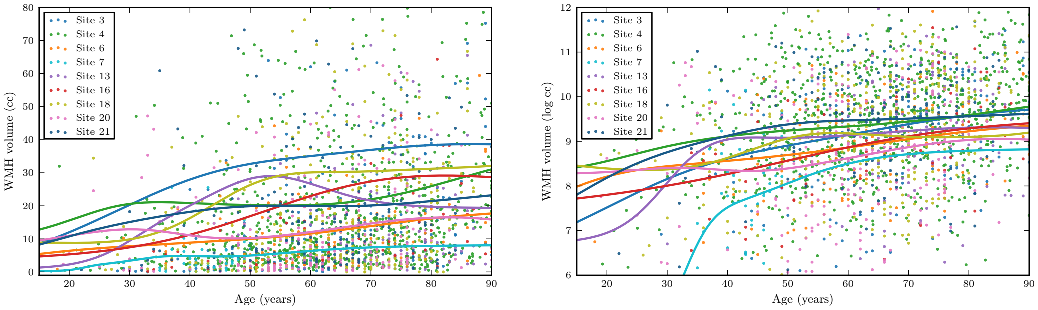


Figure 5.14: Relationship between age and WMH burden on a linear scale (left) and on a log scale (right) for all sites. The overall distributions are fairly similar, and generally show an increasing amount of WMH and increasing variability with age. The solid lines correspond to kernel regression curves for each site.

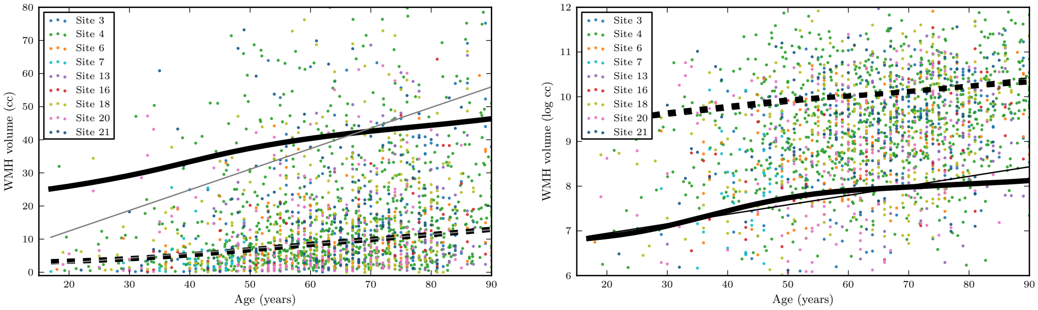


Figure 5.15: Two-component regression mixture model applied to age and WMH burden for both original data (left) and log-transformed data (right). The linear regression lines for each component are shown in gray, while the kernel regression lines are shown in black.

Chapter 6

Case study 2: Longitudinal changes in WMH in an Alzheimer's disease cohort

In this chapter, we describe our work on analysis of multimodal data in a cohort of Alzheimer's patients and healthy controls. We present an analysis pipeline for the study of longitudinal changes in anatomy and pathology. Our pipeline enables analysis of multimodal longitudinal brain MRI data, including high resolution T1-weighted scans and FLAIR scans similar to the ones explored in the stroke study in Chapter 5. For each subject in the study, several scans of varying modalities are acquired over time.

The work in this chapter was motivated by the goal of predicting anatomical changes in longitudinal data; Appendix A includes a paper under review that employs the registration framework described in this chapter to produce anatomical predictions of image change in longitudinal cohorts.

We first describe our pipeline for registration of the longitudinal cohort to an atlas, aligning all images from all subjects to a common coordinate frame. We then demonstrate an analysis of white matter hyperintensity and discuss further research directions.[1]

■ 6.1 Methods

We describe our method for registration of longitudinal scans in the study into a common coordinate frame.

■ 6.1.1 Longitudinal registration

Registration of a longitudinal cohort presents a set of challenges and promises significant advantages. Accurate data fusion in our application requires alignment of all scans from

[1] The work presented in this chapter has been conducted in collaboration with Adrian Dalca from the Computer Science and Artificial Intelligence Laboratory at MIT.

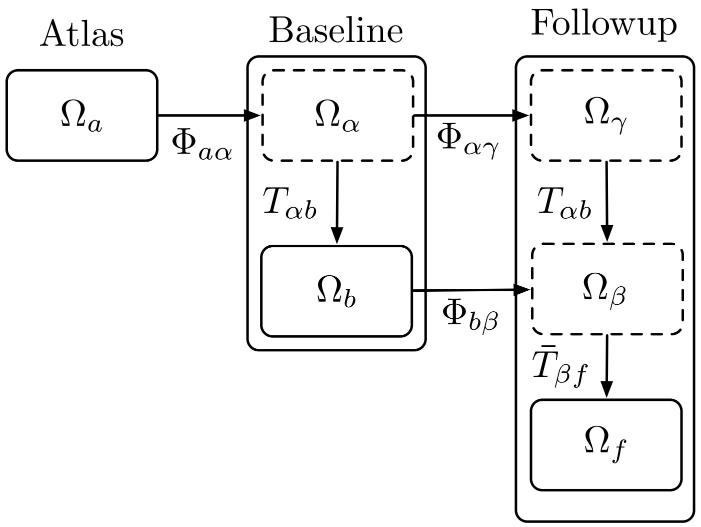


Figure 6.1: An illustration of the different coordinate systems and the relationships between them. Solid boxes and roman subscripts (Ω_a, Ω_b, and Ω_f) denote coordinates frames in which input images exist, while dotted boxes and Greek subscripts (Ω_α, Ω_β, and Ω_γ) denote coordinate frames which we define for convenience. These transformations are replicated for each followup for a subject, and again for each subject.

all subjects to a common template. However, as described in Section 4.2.3, nonlinear transformations between scans from different subjects are larger and more difficult to estimate than nonlinear transformations between different scans from the same subject. Motivated by this, we align the first longitudinal scan, which we refer to as the baseline scan, to the atlas and align each subsequent, or followup, scan to the first one. Such alignment enables comparison of the segmented regions of interest and the estimated nonlinear transformations measuring change in volume of these structures. For the transformations to serve as relevant measures of anatomical differences, we must constrain the algorithm to compute all transformations from a common coordinate system. In particular, transformations from this coordinate system should capture anatomical information rather than irrelevant differences in orientation or scale (Rohlfing et al., 2009a; Rueckert et al., 2003). We construct a registration pipeline that aligns all images affinely to the atlas coordinate system and provides computed transformations between images in this coordinate space.

Let I_a, I_b, and I_f denote the atlas image and baseline image and followup images for a particular subject respectively. Similarly, let Ω_a, Ω_b, and Ω_f denote the respective coordinate systems of these images. We define the coordinate system Ω_α such that the affine transformation between Ω_a and Ω_α is negligibly small, and the nonlinear transformation between these two coordinate systems captures the relevant anatomical

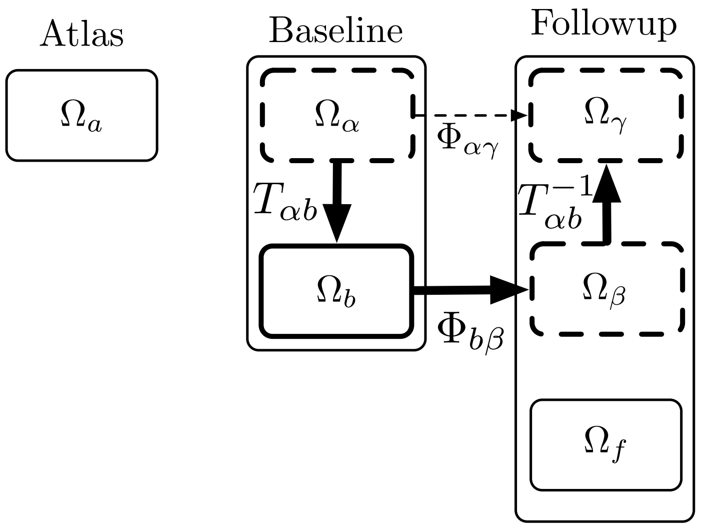


Figure 6.2: An illustration of the transformation $\Phi_{\alpha\gamma}$ expressed as the composition of the affine transformation $T_{\alpha b}$, the nonlinear transformation $\Phi_{b\beta}$, and the inverse affine transformation $T_{\alpha b}^{-1}$.

information for this subject. Let Φ denote nonlinear transformations, T denote affine transformations, and $\bar{T}$ denote rigid transformations.

Figure 6.1 illustrates the different coordinate systems described above and the relationships between them. The overall transformation from the atlas image I_a to the baseline image I_b is described by the composition $\Phi_{a\alpha} \circ T_{\alpha b}$. Similarly, the overall transformation from the baseline image I_b to the followup image I_f is described by the composition $\Phi_{b\beta} \circ \bar{T}_{\beta f}$. These transformations can be computed using any registration algorithm.

The transformation of interest is the nonlinear transformation between the images in the coordinates Ω_α and Ω_γ, or $\Phi_{\alpha\gamma}$. The affine transformation between coordinate systems Ω_γ and Ω_β is identical to the affine transformation $T_{\alpha b}$, assuming that this transformation captures all the change in orientation, scale, and skew between coordinate systems Ω_α and Ω_b, and that the nonlinear transformation $\Phi_{b\beta}$ is small. However, the same is not true of the nonlinear transformations $\Phi_{b\beta}$ and $\Phi_{\alpha\gamma}$, since not only are they applied to different coordinate spaces, but the degree of deformation caused by the transformation must scale with the affine transformation as well. Unfortunately, direct computation of the transformation of interest $\Phi_{\alpha\gamma}$ would involve instantiation of images in coordinate systems Ω_α and Ω_γ, requiring interpolation of the input images into these coordinates. As the input data can be of very low resolution, such interpolation would introduce significant error compared to the equivalent registration between the native image coordinates Ω_b and Ω_f. Thus, we instead compute the transformation $\Phi_{\alpha\gamma}$ using a composition of transformations, illustrated in Figure 6.2. That is,

$$\Phi_{\alpha\gamma} = T_{\alpha b} \circ \Phi_{b\beta} \circ T_{\alpha b}^{-1}.$$

Given the input images I_a, I_b, and I_f, we can directly compute the transformations $T_{\alpha b}$ and $\Phi_{b\beta}$. As the former is directly invertible, we can compute $\Phi_{\alpha\gamma}$ without using interpolated images in registration, leading to more accurate results.

In order to enable accurate registration, we remove the skull from the input images using ROBEX (Iglesias et al., 2011), and correct for field inhomogeneities using bias field correction (Tustison and Gee, 2009). Finally, we adjust the intensities using the same global strategy described in Section 5.1.2.

■ 6.1.2 Segmentation

As in Chapter 5, accurate alignment of all images enables segmentation and analysis in a common coordinate frame. We segment white matter hyperintensity using the same approach as described in Section 5.2, using the same threshold learned from the manually segmented images of stroke patients.

■ 6.1.3 Growth of WMH

Our registration framework enables detailed spatial analysis of each set of longitudinal images. In particular, we take advantage of the computed warps $\Phi_{\alpha\gamma}$ that are defined in a common coordinate frame and describe the transformation between a baseline image and a followup image. In particular, the determinant of the Jacobian of this transformation at every voxel is a measure of the growth or shrinkage of that voxel under the warp. A voxelwise variant of this approach is referred to as tensor-based morphometry (Ashburner and Friston, 2000). Formally, let $J(\cdot)$ be the Jacobian operator, and $\Phi^i_{\alpha\gamma}(x)$, $x \in \Omega_\gamma$, indicate the warp for subject i at voxel x. We construct a map

$$\psi^i(x) = \log \left| J(\Phi^i_{\alpha\gamma})(x) \right|$$

such that negative values of the map $\psi^i(x)$ indicate shrinkage and positive values indicate growth across time in subject i. We emphasize that due to the low resolution of the underlying images and the increased error in registration, the methods described here are exploratory in nature (Ashburner and Friston, 2001; Bookstein, 2001).

Maps of deformation of these regions, specifically within regions of white matter hyperintensity, can provide information about expansion of those regions. We construct two maps within the region of possible white matter hyperintensity, Ψ_H and $\Psi_{\bar{H}}$, where H and $\bar{H}$ correspond to hyperintense and healthy regions respectively. As in

Section 5.2.1, let $L^i(x) \in \{H, \bar{H}\}$ represent the tissue label at each voxel for subject i. Then the affine alignment facilitates the construction of the maps as follows:

$$\Psi_H(x) = \frac{\sum_i \psi^i(x)\mathbb{1}(L(x) = H)}{\mathbb{1}(L(x) = H)}$$

$$\Psi_{\bar{H}}(x) = \frac{\sum_i \psi^i(x)\mathbb{1}(L(x) = \bar{H})}{\mathbb{1}(L(x) = \bar{H})}$$

For each voxel, the maps Ψ_H and $\Psi_{\bar{H}}$ provide the average change across subjects in healthy tissue and WMH respectively in that voxel. Comparing the two can provide insight into the spatial pattern of WMH growth.

■ 6.2 Facilitation by PipeBuilder

Figure 6.3 demonstrates the automatic visualization of the pipeline described above. As with the pipeline of Chapter 5, the result of preprocessing is used extensively for the remainder of the pipeline. The visualization highlights the parallelizability of the pipeline, as the top portion which processes the baseline scan is roughly parallel to the bottom portion which processes one followup scan.

The development of the registration pipeline described above required substantial iteration. In particular, the correct composition of warps as shown in Figure 6.2 was preceded by several incorrect attempts. Applying these involved the use of several auxiliary tools provided by ANTS, as well as a number of custom scripts. PipeBuilder's ability to provide this allowed rapid iteration through unsuccessful attempts before achieving the method described in Section 6.1.1.

■ 6.3 Data

The data used in this section were provided by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (Weiner et al, 2012). In particular, we use images from the MR imaging core as described in Jack et al., 2008 and Jack et al., 2010. The data include T2-FLAIR (0.8594×0.8594mm in-plane resolution, slice thickness 5 mm), and T1-weighted ($1 \times 1 \times 1$mm resolution) images, acquired at varying time points. In particular, for each patient, the acquisition of images of different modalities does not occur at the same time. Each subject has been diagnosed as either a healthy control subject, an Alzheimer's patient (AD), or as being afflicted with mild cognitive impairment (MCI), an intermediate level of impairment that is not severe enough to be classified as Alzheimer's or other dementia. Patients ranged in age from 55 to 95, and each patient had between zero and five followup scans.

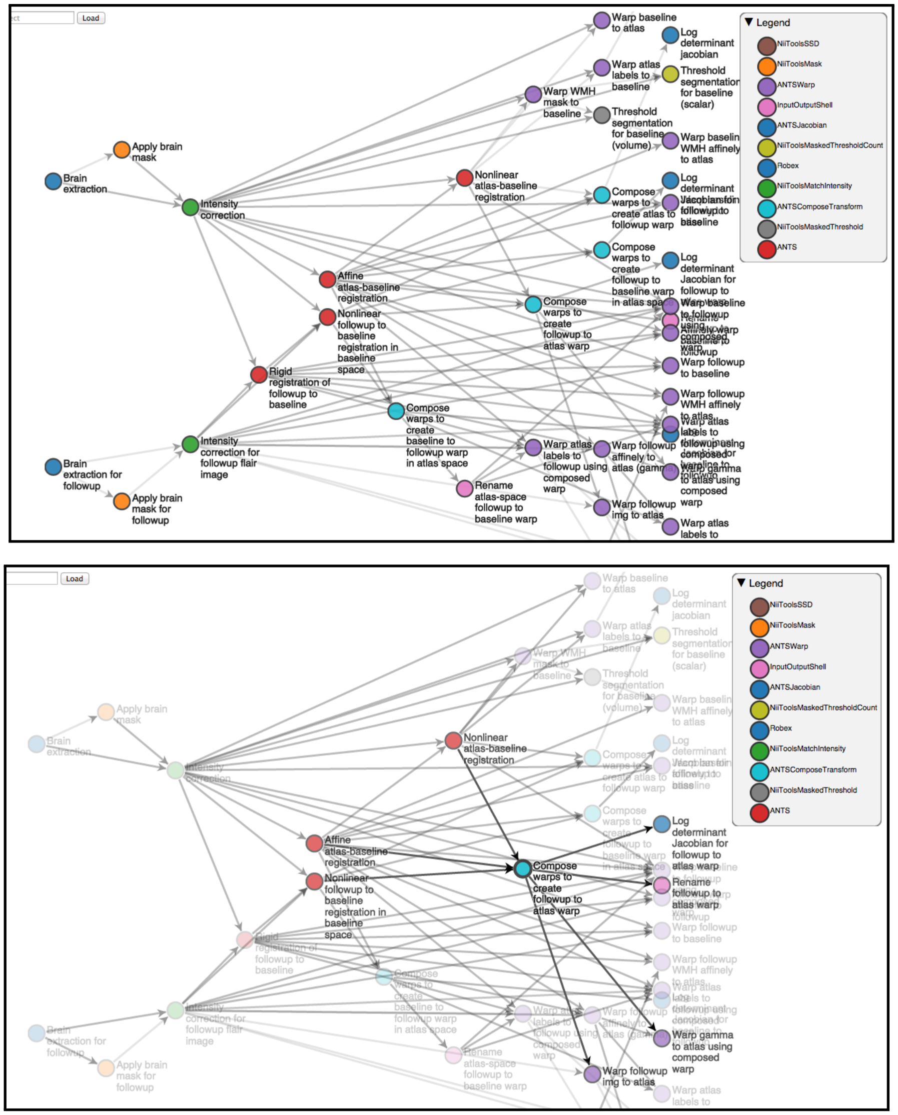

Load
Legend
NiiToolsSSD
NiiToolsMask
ANTSWarp
InputOutputShell
ANTSJacobian
NiiToolsMaskedThresholdCount
Robex
NiiToolsMatchIntensity
ANTSComposeTransform
NiiToolsMaskedThreshold
ANTS
Brain extraction
Apply brain mask
Intensity correction
Affine atlas-baseline registration
Nonlinear followup to baseline registration in baseline space
Rigid registration of followup to baseline
Intensity correction for followup flair image
Brain extraction for followup
Apply brain mask for followup
Nonlinear atlas-baseline registration
Compose warps to create baseline to followup warp in atlas space
Rename atlas-space followup to baseline warp
Compose warps to create followup to atlas warp
Warp WMH mask to baseline
Warp baseline to atlas
Warp atlas labels to baseline
Threshold segmentation for baseline (volume)
Compose warps to create atlas to followup warp
Compose warps to create followup to baseline warp in atlas space
Warp atlas labels to followup using composed warp
Warp followup affinely to atlas
Warp followup img to atlas
Log determinant jacobian
Threshold segmentation for baseline (scalar)
Warp baseline WMH affinely to atlas
Log determinant Jacobian for followup to atlas warp
Warp baseline to followup
Rename followup to atlas warp
Warp followup to baseline
Warp followup WMH affinely to atlas
Warp atlas labels to followup using composed warp
Warp gamma to atlas using composed warp
Warp atlas labels to

Figure 6.3: Top: automatic visualization of the pipeline for registration of baseline and one followup scan. The upper area of the graph mostly contains steps for registration and segmentation of the baseline scan, while the lower area of the graph mostly contains nodes for registration and segmentation of the followup scan. While most of the pipeline is clearly visualized in this graph, the many warp operations (purple) make it difficult to understand the last stages at a glance. Bottom: an alternate view of the pipeline visualized in Figure 6.3. Mousing over one node (teal) near the end of the pipeline highlights it and its inputs and outputs. While the overall structure at the right of the visualization is complex, this enables us to focus on a particular portion of it.

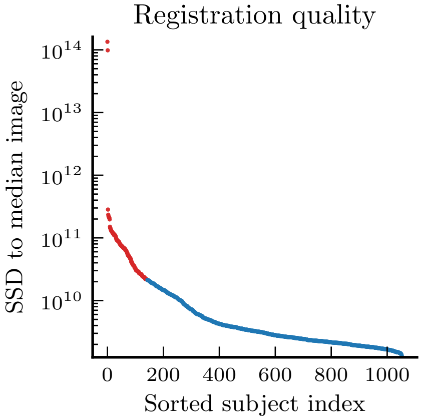


Figure 6.4: Registration quality results for baseline scans. Outliers, which were removed from further analysis, are shown in red.

■ 6.4 Results

■ 6.4.1 Registration

Figure 6.4 reports the registration quality evaluation metric from Section 5.1.3. As both the registration process for the baseline scans and the quality of the scans themselves are very similar to those described in Section 5.1.2, it is unsurprising that these results closely resemble the results from the multi-site evaluation shown in Figure 5.9. As the registrations between scans from the same subject are not as subject to error, we exclude images based only the quality of registration of the baseline image. As before, we use a similar threshold, since visual inspection of images near the threshold indicates that this separates well registered images from poorly registered ones.

■ 6.4.2 Segmentation and analysis

Figure 6.5 reports WMH volumes for each baseline scan as a function of age, derived from the automatic segmentations using T2-FLAIR images. Figure 6.6 shows the distributions of subpopulations grouped by age. We observe that this population exhibits notably less volume of pathology than the stroke patient cohort described in the previous chapter, but shows a similar progression with age. This is consistent with prior literature indicating that white matter hyperintensity is a risk factor for stroke (Kuller et al., 2004; R. Schmidt et al., 1992). this dataset and our longitudinal registration framework enable longitudinal analysis to better explore the change in white matter over time. Figure 6.7 illustrates the progression of each patient's WMH volume over time. We see that while a large fraction of the cohort maintains a relatively constant

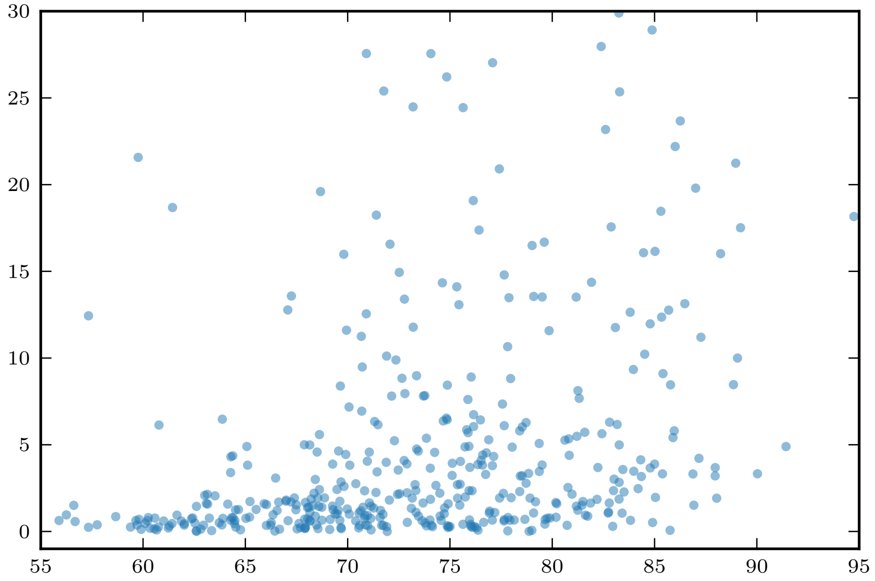


Figure 6.5: Automatic WMH segmentation results as a function of age. Only baseline values are shown in this plot.

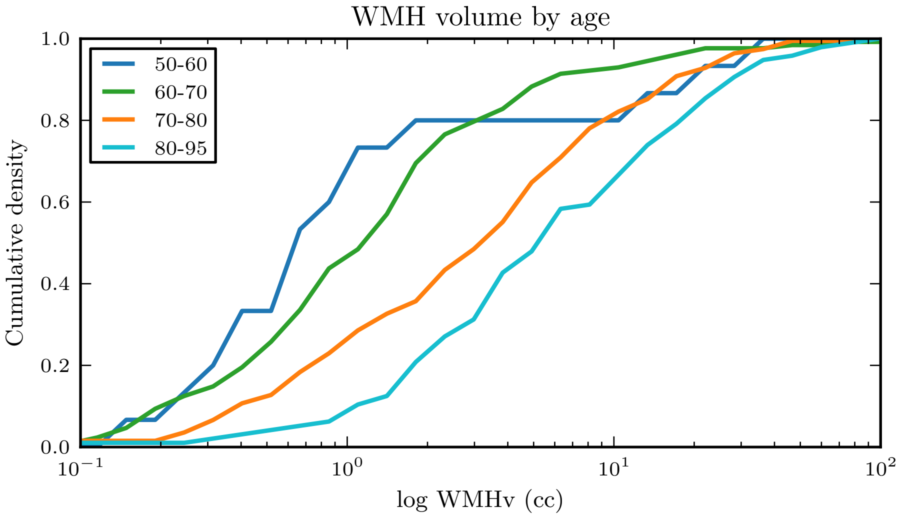


Figure 6.6: Cumulative density of white matter hyperintensity for patients in four different age groups. The distribution shifts to the right as age increases.

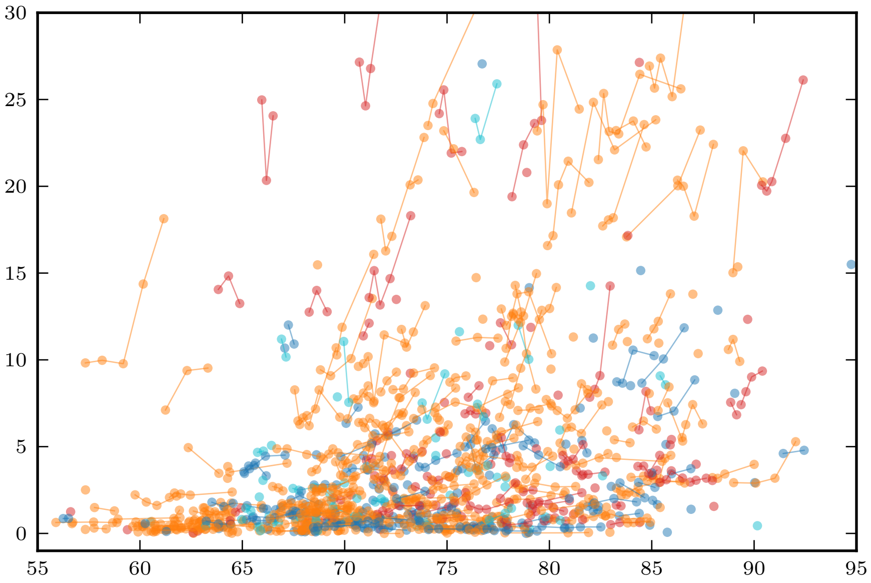


Figure 6.7: Longitudinal data for WMH progression. Each trajectory represents a single subject, and colors denote diagnosis: blue indicates healthy controls, red indicates Alzheimer's, and orange indicates MCI.

WMH burden over time, some patients exhibit substantial growth. However, there appears to be no strong link between WMH burden and clinical diagnosis.

Motivated by the noticeable difference between the majority of patients who exhibit little to no change and those who exhibit substantial growth over the period of observation, we apply a two-component analysis to the patient cohort. Each group is characterized by a different fitted slope within a patient's trajectory; the threshold of 1.1cc/year was chosen by applying k-means clustering to the scalar slopes computed using linear regression for each subject. The average baseline images from each cluster in this analysis are shown in Figure 6.8. While the correlation between initial volume and average slope is relatively weak ($r^2 = 0.16$), we see that images with substantial growth still often start out with a higher burden of WMH.

Figure 6.9 shows the two maps Ψ_H and $\Psi_{\bar{H}}$, restricted to the mask in which WMH is found. We see that throughout the regions of white matter hyperintensity, the values are positive, indicating average growth of WMH from baseline to followup. In particular, the growth seems greatest in the occipital lobe (near the back of the brain). While the difference across the brain is relatively small, this finding, along with the results from Chapter 5 suggests that more severe WMH may develop not only in the back of the

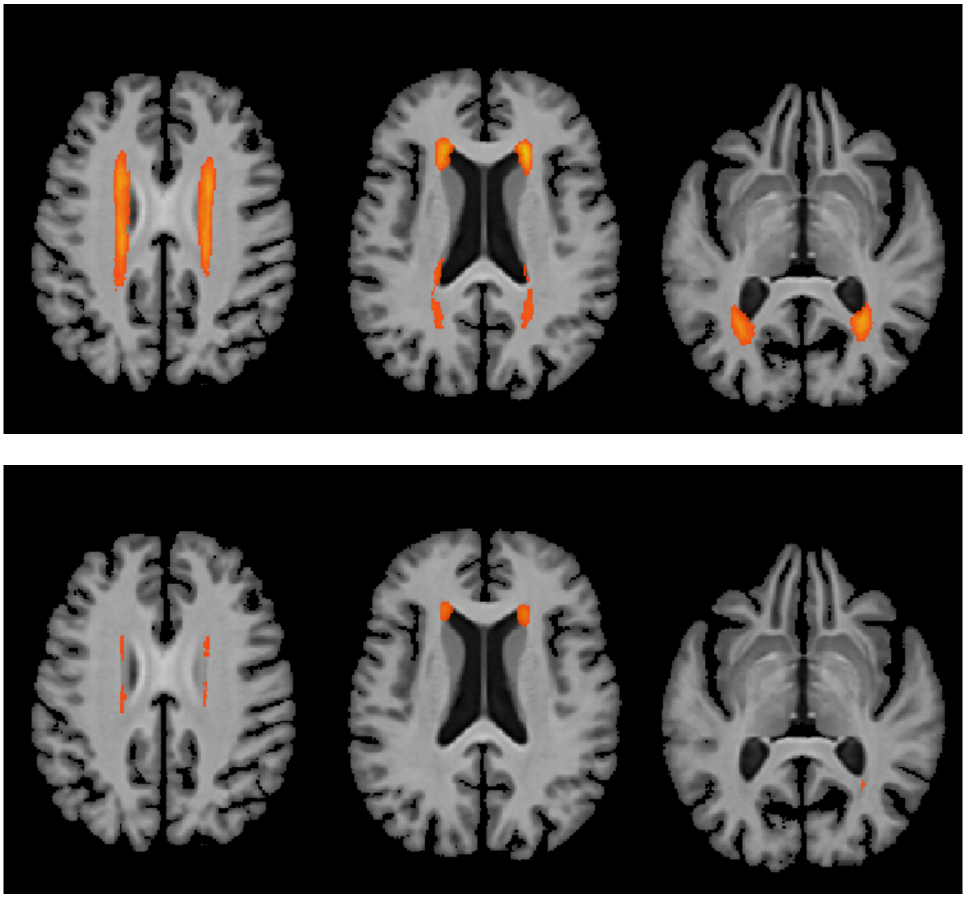


Figure 6.8: Average WMH at baseline for patients in the high slope cluster (top) and low slope cluster (bottom). Subjects were chosen for inclusion based on fitting a slope for their trajectory, while the image shows the average value in the baseline scans.

brain, but also near the lower portions of the occipital lobe.

■ 6.5 Summary

We presented a pipeline for analysis of longitudinal data. Enabled by PipeBuilder, our pipeline was constructed through experimentation and iterative development. We adapted portions of the registration and segmentation framework from Chapter 5 to analyze WMH and change in WMH in this cohort.

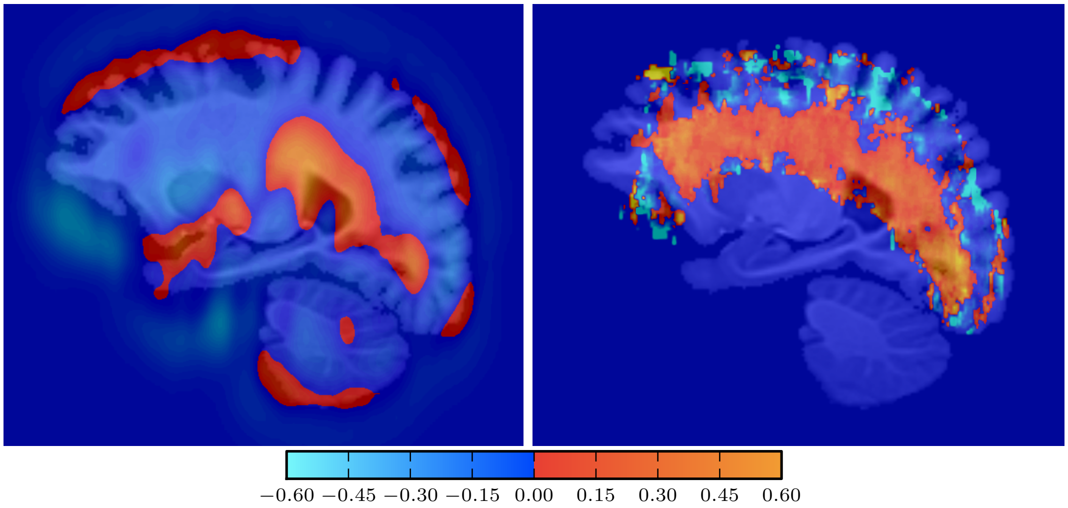


Figure 6.9: Average shrinkage (blue) or growth (red) of each voxel over time for subjects with voxels labeled WMH (left) and healthy (right). Light blue values indicate greater shrinkage, while yellow values indicate greater growth.

Chapter 7

Conclusions and Discussion

In this thesis we introduced PipeBuilder, a powerful system for building computational pipelines to analyze medical imaging data. PipeBuilder is particularly well suited for analysis of large collections of potentially low resolution or noisy data, where a lot of methodological experimentation. We presented analyses in two such applications that were enabled by PipeBuilder. In a neuroimaging study of stroke patients, we studied white matter hyperintensity burden in a large cohort. In a large longitudinal cohort of Alzheimer's patients, we studied longitudinal change in these patients.

The studies were facilitated by PipeBuilder, which was created with three primary design goals in mind: enabling iterative method refinement, supporting adaptability and robustness, and providing visualization of the analysis pipeline as well as the data itself. Iterative refinement is a critical component of research, especially when analyzing low resolution data for which many traditional analysis methods fail. Large collections of such data also require the design of robust and adaptable processing methods: variations in quality and in the populations under study necessitate adaptable methods that can work despite such variability. Finally, visualization of a pipeline's structure facilitates a better understanding of complex computational structures, and easily accessible visualization of the intermediate results enables not only rapid iteration and refinement, but also quality control. Motivated by these goals, PipeBuilder provides a framework for building and visualizing robust, adaptable pipelines that can be easily refined and iterated on.

We demonstrated the strengths of this framework in two different investigations of WMH in large neuroimaging studies. In a study of stroke patients, we used a three-step approach that included registration to a template for data fusion, automatic segmentation of WMH using multimodal MRI, and analysis of WMH volumes and spatial distributions. We exploited the adaptability of our PipeBuilder workflow to expand our method to a multi-site study, where we demonstrated consistency in the amount of WMH in several patient populations. PipeBuilder enabled straightforward adaptation of the pipeline to these sites, including one site where a site-specific template was required due to younger patients in that site. This paves the way for future work examining the relationship between templates and quality of the resulting registrations: age-specific (rather than site-specific) templates have shown promise (Fonov et al., 2009; Fonov et al., 2011) in previous work, and the combination of a large patient cohort and the adaptability provided by PipeBuilder holds promise in applying similar analyses to this data.

We also demonstrated PipeBuilder on a longitudinal study, where we examined subject-specific changes in members of a population over time. We constructed a registration pipeline that accounted for longitudinal change within patients and for variability across patients, and adapted our analysis of WMH to this pipeline. We explored the change in volume and spatial distribution of white matter hyperintensity over time per patient, pooling not only the data about each patient's WMH but also the change in WMH as well.

Beyond several interesting extensions to the work in this thesis, including a more robust method of detecting outliers, visualization of large image collections, and access to a broader range of tools, PipeBuilder enables a range of further research into analysis of large collections of low quality images that previously would have been difficult. Another important future direction of work for PipeBuilder is more robust user evaluation and feedback. While a full user study would be difficult as discussed in Section 3.4, some recent work in workflow systems for other domains (Patel et al., 2010) has involved smaller scale user studies; similar studies for PipeBuilder, however informal, could provide invaluable feedback and guide future directions of research and development.

One important direction of work that PipeBuilder enables is a better understanding of the performance of various widely used algorithms for medical image computing tasks. A broad range of work exists quantifying and benchmarking the performance of such algorithms (Klein et al, 2009; Iglesias et al., 2011). However, with its focus on iterative refinement, PipeBuilder is well suited to facilitating automatic evaluation of this nature, as algorithms can be measured and evaluated as part of the iterative design process. This could facilitate a deeper understanding of the strengths and weaknesses of different algorithms with respect to different properties of the underlying data. Aggregation of such information across multiple users of PipeBuilder using different algorithms across a range of image collections could enable such evaluation on a broader scale than previously conducted.

Within the application domain of the study of WMH, the work in this thesis motivates further analysis. Beyond important extensions and improvements such as investigating ways to improve quality of registration and detection of outliers, the large scale registration and segmentation pipelines in this thesis make richer analyses possible. Combining the information extracted in this manner with clinical and genetic factors has potential to unlock exciting new directions in understanding patient outcomes and risk profiles. For example, combining the information available in both Chapter 5 and Chapter 6 could enable case-control studies that might provide insights into stroke risk profiles given WMH level.

In summary, PipeBuilder is a novel framework for building and refining robust and adaptable pipelines for analysis of large scale medical imaging data. The applications provided in the thesis are just the start demonstrate its capabilities in two real clinical studies. We believe that PipeBuilder has the potential to empower researchers to tackle new problems and seek new insights from medical imaging data.

Appendix A

Predictive Modeling of Anatomy with Genetic and Clinical Data

Here we present our work under review that motivates the longitudinal registration pipeline described in Chapter 6. In particular, longitudinal registration of a large cohort enables prediction of images using existing images for training and incorporating external information such as genetic and clinical factors.

■ A.1 Introduction

We present a method for predicting anatomy based on other available information, including genetic and clinical indicators. Specifically, given the baseline scan of a subject in a longitudinal study, our model predicts anatomical changes and generates a subsequent image using subject-specific genetic and clinical information. Such voxel-wise prediction opens up several new areas of analysis, enabling novel investigations both at the voxel-level and at the level of derivative biomarker measures. For example, we show that voxel-level differences are observed between the true progression of a patient with dementia and their predicted anatomy under a healthy model, highlighting spatial patterns of disease. We validate our method by comparing measurements of volumes of anatomical structures based on predicted images to those extracted from the acquired scans.

Our model describes the change from a baseline image explained by population trends and by subject-specific information. The first scan of a subject in a longitudinal study serves as the baseline for the purposes of prediction. First, we model how anatomical appearance changes with age across a population. We then model deviations from the population average using a person's *health profile*, which we characterize non-parametrically based on the genotype, clinical information, and the baseline image. Subject-specific change is learned from the similarity of health profiles in the cohort, which are incorporated into our predictions using a Gaussian process parametrized by a population health covariance. Given the predicted change, we synthesize a new image through an appearance model.

Statistical population analysis is one of the central topics in medical image computing. The classical correlation-based analysis has yielded important characterization of relationships within imaging data and with independent clinical variables (B. C. Davis et al., 2010; Misra et al., 2009; Pfefferbaum et al., 2013; Risacher et al., 2010; Rohlfing et al., 2009b). Regression models of object appearance have been previously used for atlas construction and population analysis (B. C. Davis et al., 2010; Rohlfing et al., 2009b). These methods characterize population trends with respect to external variables, such as age or gender, and construct clinically relevant population averages. Longitudinal analyses also characterize subject-specific temporal effects, usually in terms of changes in the biomarkers of interest. Longitudinal cohorts and studies promise to provide crucial insights into aging and disease (Misra et al., 2009; Pfefferbaum et al., 2013; Risacher et al., 2010). Mixed effects models have been shown to improve estimation of subject-specific longitudinal trends by using inter-population similarity of a biomarker of interest (Datar et al., 2012; Sadeghi et al., 2013). While these approaches offer a powerful basis for analysis of biomarkers or images across populations, they do not aim to provide subject-specific predictions. The parameters of the models are examined for potential scientific insight, but they are not tested for predictive power. In contrast, we define the problem of population analysis as predicting anatomical changes for individual subjects. Our generative model incorporates a population trend and uses subject-specific genetic and clinical information, along with baseline images, to generate subsequent anatomical images. This approach provides avenues for novel analysis, as illustrated by our experimental results.

■ A.2 Prediction Model

Given a dataset of patients with longitudinal data and a baseline image for a new patient, we predict a follow-up anatomical state for a patient. We model anatomy as phenotype y that captures the underlying structure of interest. For example, y can be a low-dimensional descriptor of the anatomy at each voxel. We assume we have a measurement of our phenotype at baseline y_b for a new subject. Our goal is to predict the phenotype y_t at a later time t. We let x_t be the subject age at time t, and define $\Delta x_t = x_t - x_b$ and $\Delta y_t = y_t - y_b$. We model the change in phenotype y_t using linear regression:

$$\Delta y_t = \Delta x_t \beta + \epsilon, \tag{A.1}$$

where β is the subject-specific regression coefficient, and $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is sampled from zero-mean Gaussian distribution with variance σ^2.

In the remainder of this section, we first describe the core of our model that captures the changes in anatomical appearance using the population trend and patient specific information, and then outline the resulting prediction algorithm. In the next section, we describe our particular choice of appearance model for phenotype y to enable prediction of follow-up scans, and describe the subject-specific health profiles.

■ A.2.1 Subject-Specific Longitudinal Change

To model subject-specific effects, we define $\beta = \bar{\beta} + H(g, c, f_b)$, where $\bar{\beta}$ is a global regression coefficient shared across the entire population and H captures a deviation from this coefficient based on the subject's genetic variants g, clinical information c, and baseline image features f_b.

We assume that patients' genetic variants and clinical indicators affect the anatomical appearance, and that subjects with similar health profiles exhibit similar patterns of anatomical change. We define $h_G(\cdot)$, $h_C(\cdot)$, $h_I(\cdot)$ to be functions that capture genetic, clinical and imaging effects on the regression coefficients:

$$H(g, c, I_b) = h_G(g) + h_C(c) + h_I(f_b). \tag{A.2}$$

Combining with (A.1), we arrive at the full model

$$\Delta y_t = \Delta x_t \bar{\beta} + \Delta x_t \left(h_G(g) + h_C(c) + h_I(f_b)\right) + \epsilon, \tag{A.3}$$

which captures the population trend through $\bar{\beta}$, and the subject-specific deviations from that trend through the subject-specific health $[h_G(\cdot), h_C(\cdot), h_I(\cdot)]$.

For a longitudinal cohort of N subjects, we group all observations Δy_t for subject i to form $\Delta \boldsymbol{y}_i = [y_{i_1}, y_{i_2}, ... y_{i_{T_i}}]$, where T_i is the number of scans for patient i. We then form the global vector $\Delta \boldsymbol{y} = [\Delta \boldsymbol{y}_1, \Delta \boldsymbol{y}_2, ..., \Delta \boldsymbol{y}_N]$. We similarly form vectors $\Delta \boldsymbol{x}$, $\boldsymbol{h}_G$, $\boldsymbol{h}_C$, $\boldsymbol{h}_I$, $\boldsymbol{g}$, $\boldsymbol{c}$, $\boldsymbol{f_b}$ and $\boldsymbol{\epsilon}$, and obtain:

$$\Delta \boldsymbol{y} = \Delta \boldsymbol{x} \bar{\beta} + \Delta \boldsymbol{x} \odot \left(\boldsymbol{h}_G(\boldsymbol{g}) + \boldsymbol{h}_C(\boldsymbol{c}) + \boldsymbol{h}_I(\boldsymbol{f}_b)\right) + \boldsymbol{\epsilon}. \tag{A.4}$$

where $\odot$ is the Hadamard, or element-wise product. This formulation is mathematically equivalent to a General Linear Model (GLM) (McCullagh, 1984) in terms of the health profile predictors $[\boldsymbol{h}_G, \boldsymbol{h}_C, \boldsymbol{h}_I]$.

We employ Gaussian process priors to model the predictor functions:

$$h_D(\cdot) \sim GP(\boldsymbol{0}, \tau_D^2 K_D(\cdot, \cdot)), \tag{A.5}$$

where covariance kernel function $\tau_D^2 K_D(z_i, z_j)$ captures the similarity between subjects i and j using feature vectors $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ for $D \in \{G, C, I\}$. We discuss the particular forms of $K(\cdot, \cdot)$ used in the experiments later in the paper.

A.2.2 Learning

The Bayesian formulation (A.4) can be interpreted as a linear mixed effects model (LMM) (McCulloch and Neuhaus, 2001) or a least squares kernel machine (LSKM) regression model (Ge et al., 2015; D. Liu et al., 2007). Linear mixed models have been recently used in population genetics to account for confounding structure. Here, we use the linear mixed effects model interpretation to learn the parameters of our model, and the LSKM interpretation to perform final predictions of phenotypes $\boldsymbol{y}$.

Under the linear mixed effects model interpretation, $\bar{\beta}$ is the coefficient vector of fixed effects and $\boldsymbol{h}_G, \boldsymbol{h}_C$, and $\boldsymbol{h}_I$ are independent random effects. We seek the maximum likelihood estimates of parameters $\bar{\beta}$ and $\boldsymbol{\theta} = (\tau_G^2, \tau_C^2, \tau_I^2, \sigma^2)$, by adapting standard procedures for LMMs (Ge et al., 2015; D. Liu et al., 2007). As standard LMM solutions become computationally expensive for thousands of observations, we take advantage of the fact that while the age and phenotype data is large, the kernels only operate on baseline data. We obtain intuitive iterative updates that project the residuals at each step onto the expected rate of change in likelihood, and update $\bar{\beta}$ using the best linear unbiased predictor.

A.2.3 Prediction

Under the LSKM interpretation, the terms $h(\cdot)$ are estimated by minimizing a penalized squared-error loss function, which leads to the following solution (Ge et al., 2015; Kimeldorf and Wahba, 1971; D. Liu et al., 2007; Wahba, 1990):

$$h(z_i) = \sum_{j=1}^{N} \alpha_j K(z_i, z_j) \quad \text{or} \quad \boldsymbol{h} = \boldsymbol{\alpha}^T \boldsymbol{K} \tag{A.6}$$

for some vector $\boldsymbol{\alpha}$. Combining with the definitions of the LMM, we estimate coefficients vectors $\boldsymbol{\alpha}_G, \boldsymbol{\alpha}_C$ and $\boldsymbol{\alpha}_I$ from the following linear system of equations:

$$\begin{bmatrix} \mathbb{1}^T\mathbb{1} & \mathbb{1}^T\boldsymbol{K}_G & \mathbb{1}^T\boldsymbol{K}_C & \mathbb{1}^T\boldsymbol{K}_I \\ \boldsymbol{K}_G^T\mathbb{1} & \boldsymbol{K}_G^T\boldsymbol{K}_G + \frac{\tau_G^2}{\sigma^2}\boldsymbol{K}_G^T\mathbb{1} & \boldsymbol{K}_G^T\boldsymbol{K}_C & \boldsymbol{K}_G^T\boldsymbol{K}_I \\ \boldsymbol{K}_C^T\mathbb{1} & \boldsymbol{K}_G^T\mathbb{1} & \boldsymbol{K}_C^T\boldsymbol{K}_C + \frac{\tau_C^2}{\sigma^2}\boldsymbol{K}_C^T\mathbb{1} & \boldsymbol{K}_C^T\boldsymbol{K}_I \\ \boldsymbol{K}_I^T\mathbb{1} & \boldsymbol{K}_I^T\boldsymbol{K}_G & \boldsymbol{K}_I^T\mathbb{1} & \boldsymbol{K}_I^T\boldsymbol{K}_I + \frac{\tau_I^2}{\sigma^2}\boldsymbol{K}_I^T\mathbb{1} \end{bmatrix} \begin{bmatrix} \bar{\beta} \\ \boldsymbol{\alpha}_G \\ \boldsymbol{\alpha}_C \\ \boldsymbol{\alpha}_I \end{bmatrix} = \begin{bmatrix} \mathbb{1}^T\boldsymbol{y}' \\ \boldsymbol{K}_G^T\boldsymbol{y}' \\ \boldsymbol{K}_C^T\boldsymbol{y}' \\ \boldsymbol{K}_I^T\boldsymbol{y}' \end{bmatrix}, \tag{A.7}$$

where $\boldsymbol{y}' = \Delta \boldsymbol{y} \odot \left[\frac{1}{\Delta \boldsymbol{x}_1} \cdots \frac{1}{\Delta \boldsymbol{x}_T}\right]$. We can then re-write (A.4) as

$$\Delta \boldsymbol{y} = \Delta \boldsymbol{x} \bar{\beta} + \Delta \boldsymbol{x} \left(\boldsymbol{\alpha}_G^T \boldsymbol{K}_G + \boldsymbol{\alpha}_C^T \boldsymbol{K}_C + \boldsymbol{\alpha}_I^T \boldsymbol{K}_I \right) \qquad \text{(A.8)}$$

and predict a phenotype at time t for a new subject:

$$y_t = y_b + \Delta x_t \left[\bar{\beta} + \sum_{j=1}^{N} \alpha_{G,j} K_G(g_i, g_j) + \alpha_{C,j} K_C(c_i, c_j) + \alpha_{I,j} K_I(f_i, f_j) \right]. \qquad \text{(A.9)}$$

A.3 Model Instantiation for Anatomical Predictions

The full model (A.3) can be used with many reasonable definitions of phenotype y. In this section, we describe the phenotype model we use for anatomical predictions and specify the similarity kernels for each component of the health profile.

A.3.1 Anatomical Phenotype

We define a voxel-wise phenotype that enables us to predict entire anatomical images. Let Ω be the set of all spatial locations v (voxels) in an image, and $I_b = \{I_b(v)\}_{v \in \Omega}$ be the acquired baseline image. We similarly define $A = \{A(v)\}_{v \in \Omega}$, the population atlas template. We assume each image I is generated through a deformation field $\Phi_{AI}^{-1} = \{\boldsymbol{u}(v)\}_{v \in \Omega}$ from the common atlas to the subject-specific coordinate frame (Rohlfing et al., 2009b), such that $I(v) = A(v + \boldsymbol{u}(v))$. We further define a follow-up image I_t as a deformation Φ_{Bt} from the baseline image I_b, which can be composed to yield an overall deformation from the atlas to the follow-up scan via $\Phi_{At}^{-1} = \Phi_{AB}^{-1} \circ \Phi_{Bt}^{-1} = \{\boldsymbol{u}'(v)\}_{v \in \Omega}$:

$$I_t(v) = A(v + \boldsymbol{u}'(v)). \qquad \text{(A.1)}$$

Using deformation $\boldsymbol{u}'(v)$ as the phenotype of interest in (A.1) captures the necessary information for predicting new images, but leads to very high dimensional descriptors. To regularize the transformation and to improve efficiency, we define a low-dimensional embedding of $\boldsymbol{u}'(v)$. Specifically, we assume that the atlas provides a parcellation of the space into L anatomical labels $\mathcal{L} = \{\Psi\}_{l=1}^{L}$. We build a low-dimensional embedding of the transformation vectors $\boldsymbol{u}(v)$ within each label using PCA. We define the relevant phenotypes $y_{l,c}$ as the coefficients associated with the first C principal components of the model that capture 95% of the variance, for $l = 1 \ldots L, c = 1 \ldots C$.

We predict the phenotypes using (A.9). To construct a follow-up image I_t given the phenotype y_t, we first form a transformation field $\widehat{\Phi}_{At}^{-1}$ by reconstruction from the estimated phenotype y_t, and use $\widehat{\Phi}_{At}$ assuming an invertible transformation. Using the baseline image, we predict a subsequent image via $\Phi_{Bt} = \widehat{\Phi}_{At} \circ \Phi_{AB}^{-1}$. Note that we do not directly model changes in image intensity. While population models necessitate capturing such changes, we predict changes from a baseline image. We also assume that affine transformations are not part of deformations of interest, and thus all baseline images are affinely registered to the atlas.

■ A.3.2 Health Similarities

To fully define the health similarity term $H(\cdot,\cdot,\cdot)$, we need to specify the forms of the kernel functions $K_G(\cdot,\cdot)$, $K_C(\cdot,\cdot)$, and $K_I(\cdot,\cdot)$.

For genetic data, we employ the identical by state (IBS) kernel often used in genetic literature, which captures the similarity between two individuals in terms of their genotypes (Queller and Goodnight, 1989). Given a vector of genetic variants g of length S, each genetic locus is encoded as $g(s) \in [0, 2]$, and

$$K_G(g_i, g_j) = \frac{1}{2S} \sum_{s=1}^{S} (2 - |g_i(s) - g_j(s)|). \tag{A.2}$$

To capture similarity of clinical indicators c, we define the diagonal weight matrix $\boldsymbol{W}$ that captures the effect size of each clinical indicator on the phenotype and form the kernel function

$$K_C(c_i, c_j) = \exp\left(-\frac{1}{\sigma_C^2}(c_i - c_j)^T \boldsymbol{W} (c_i - c_j)\right), \tag{A.3}$$

where σ_C^2 is the variance of the clinical factors.

We define the image feature vectors f_b as the set of all PCA coefficients defined above for the baseline image. We define the image kernel matrix as

$$K_I(f_{b,i}, f_{b,j}) = \exp\left(-\frac{1}{\sigma_I^2}||f_{b,i} - f_{b,j}||_2^2\right), \tag{A.4}$$

where σ_I^2 is the variance of the image features.

■ A.4 Experiments

We illustrate our approach by predicting image-based phenotypes based on genetic, clinical and imaging data in the ADNI longitudinal dataset (Jack et al., 2008)[1] that includes two to ten follow-up scans acquired $0.5 - -7$ years after the baseline scan. We

[1] This data was released as part of the Alzheimer's Disease Big Data DREAM Challenge at https://www.synapse.org/#!Synapse:syn2290704

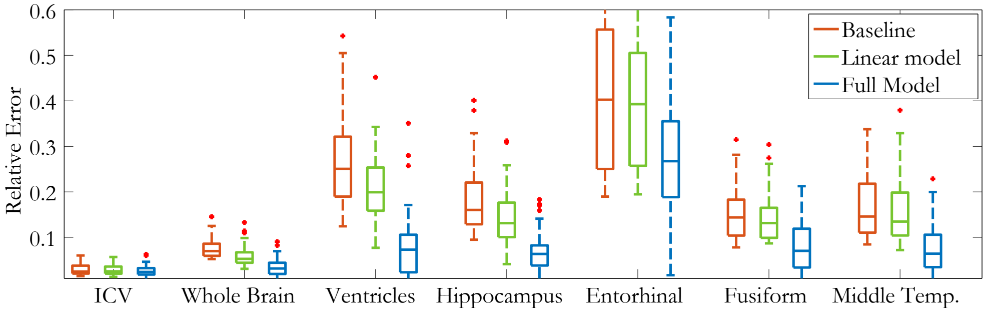


Figure A.1: Relative error (lower is better) of volume prediction for seven structures for high-change subjects (top decile of volume change). For the intracranial volume, whole brain volume, ventricles, hippocampus, entorhinal cortex, fusiform gyrus, and middle temporal gyrus, we report relative change between the baseline and the follow-up measurement (red), relative error in prediction using a linear model (green), and relative error in prediction using our complete model (blue).

use affine registration to align all subjects to a template constructed from 145 randomly chosen subjects. We then compute non-linear registration warps Φ_{AI} for each image using ANTs (Avants et al., 2011b). We include a list of 21 genetic loci associated with Alzheimer's disease (AD) as the genetic vector g, and standard clinical factors including age, gender, marital status, education, disease diagnostic, and cognitive tests, as the clinical indicator vector c. We learn the model parameters from 341 randomly chosen subjects and predict follow-up volumes on a separate set of 100 subjects. To evaluate the advantages of the proposed predictive model, we compare its performance to a population-wide linear regression model that ignores subject-specific health profiles (i.e., $H = 0$).

■ A.4.1 Volumetric Predictions

In the first simplified experiment, we define phenotype y to be a vector of several scalar volume measurements obtained using FreeSurfer (Fischl, 2012). These include intracranial volume, whole brain volume, ventricles, hippocampus, entorhinal cortex, fusiform gyrus, and middle temporal gyrus. In addition to the population-wide linear regression model, we include a simple approach of using the volume measurements at baseline as a predictor of the phenotype at the follow-up time, effectively assuming no volume change between the two time points. Since in most subjects, the volume differences are small, all three methods perform comparably when evaluated on the whole test set. To evaluate the differences between the methods, we focus on the subset of subjects with substantial volume changes, reported in Fig. A.1. We observe that our method consistently achieves smaller relative errors than the two baseline approaches.

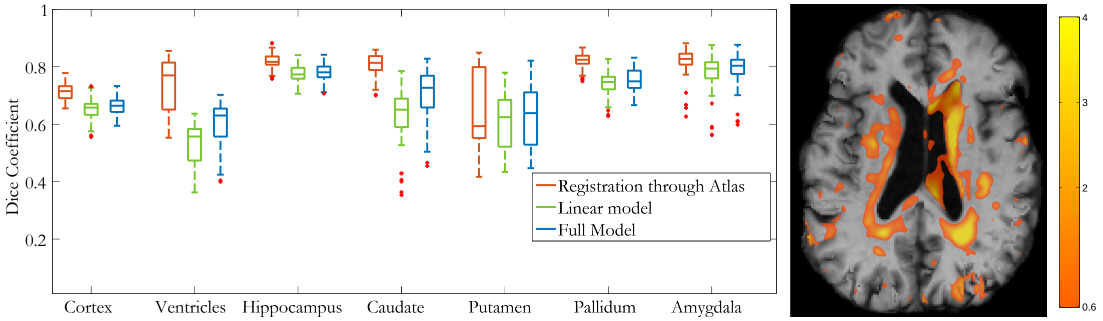


Figure A.2: Prediction results. Left: The Dice scores of labels propagated through three methods for several AD-relevant structures in subjects with the most volume change for each structure. We report the prediction based on the registration of the actual follow-up scan to the atlas as an upper bound for warp-based prediction accuracy (red), predictions based on the population-wide linear regression model (green), and the full model (blue). Right: A predicted anatomical image for a patient diagnosed with AD using a healthy model. The color overlay shows the squared magnitude of the difference in predicted versus observed deformations, indicating a significantly different expansion trajectory of the ventricles.

■ A.4.2 Anatomical Prediction

We also evaluate the model for full anatomical scan prediction. To quantify prediction accuracy, we propagate segmentation labels of relevant anatomical structures from the baseline scan to the predicted scan using the predicted warps. We then compare the predicted segmentation label maps with the actual segmentations of the follow-up scans. In this case, the warps computed based on the actual follow-up scans through the atlas provide an indication of the best accuracy the predictive model could achieve when using warps to represent images. Similar to the volumetric predictions, the full model offers modest improvements when evaluated on the entire test set, and substantial improvements in the segmentation accuracy when evaluated in the subjects who exhibit large volume changes between the baseline scan and the follow-up scan, as reported in Fig. A.2. In both experiments, we found that all components h_g, h_c and h_I contributed significantly to the improved predictions.

Our experimental results suggest that the anatomical model depends on registration accuracy. In particular, we observe that directly registering the follow-up scan to the baseline scan leads to better alignment of segmentation labels than when transferring the labels through a composition of the transformations from the scans to the atlas space. This suggests that a different choice of appearance model may improve prediction accuracy, a promising direction for future work.

To demonstrate the potential of the anatomical prediction, we predict the follow-up scan of a patient diagnosed with dementia as if the patient were healthy. Specifically, we train our model using healthy subjects, and predict follow-up scans for AD patients. In Fig. A.2 we illustrate an example result, comparing the areas of brain anatomy that differ from the observed follow-up in the predicted *healthy* brain of this AD patient. Our prediction indicates that ventricle expansion would be different if this patient had a healthy trajectory.

■ A.5 Conclusions

We present a model to predict the anatomy in patient follow-up images using population trends and subject-specific genetic and clinical information. We validate our prediction method on scalar volumes and anatomical images, and show that it can be used as a powerful tool to illustrate how a subject-specific brain might differ if it were healthy. Through this and other new applications, our prediction method presents a novel opportunity for the study of disease and anatomical development.

Bibliography

Admiraal-Behloul et al, F. (2005). “Fully automatic segmentation of white matter hyperintensities in MR images of the elderly”. In: *Neuroimage* 28.3, pp. 607–617.

Aljabar, P, KK Bhatia, M Murgasova, JV Hajnal, JP Boardman, L Srinivasan, MA Rutherford, LE Dyet, AD Edwards, and D Rueckert (2008). “Assessment of brain growth in early childhood using deformation-based morphometry”. In: *Neuroimage* 39.1, pp. 348–358.

Ashburner, John and K.J. Friston (2000). “Voxel-based morphometry-the methods”. In: *Neuroimage* 11.6, pp. 805–821.

— (2001). “Why voxel-based morphometry should be used”. In: *Neuroimage* 14.6, pp. 1238–1243.

Ashburner, John, Gareth Barnes, C Chen, Jean Daunizeau, Guillaume Flandin, K.J. Friston, D Gitelman, S Kiebel, J Kilner, V Litvak, et al. (2008). “SPM8 manual”. In: *Functional Imaging Laboratory, Institute of Neurology.*

Avants, B.B., N.J. Tustison, G. Song, P.A. Cook, A. Klein, and J.C. Gee (2011a). “A reproducible evaluation of ANTs similarity metric performance in brain image registration”. In: *Neuroimage* 54.3, pp. 2033–2044.

— (2011b). “A reproducible evaluation of ANTs similarity metric performance in brain image registration”. In: *Neuroimage* 54.3, pp. 2033–2044.

Barker, Adam and Jano Van Hemert (2008). “Scientific workflow: a survey and research directions”. In: *Parallel Processing and Applied Mathematics.* Springer, pp. 746–753.

Bellec, P., S. Lavoie-Courchesne, P. Dickinson, J.P. Lerch, A.P. Zijdenbos, and A.C. Evans (2012). “The pipeline system for Octave and Matlab (PSOM): a lightweight scripting framework and execution engine for scientific workflows”. In: *Frontiers in neuroinformatics* 6.

Berriman, G Bruce, Ewa Deelman, John C Good, Joseph C Jacob, Daniel S Katz, Carl Kesselman, Anastasia C Laity, Thomas A Prince, Gurmeet Singh, and Mei-Hu Su (2004). “Montage: a grid-enabled engine for delivering custom science-grade mosaics on demand”. In: *Astronomical Telescopes and Instrumentation.* International Society for Optics and Photonics, pp. 221–232.

Berthold, Michael R, Nicolas Cebron, Fabian Dill, Thomas R Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Christoph Sieb, Kilian Thiel, and Bernd Wiswedel (2008). *KNIME: The Konstanz information miner.* Springer.

Bhatia, Kanwal K, Joseph V Hajnal, Basant K Puri, A David Edwards, and D. Rueckert (2004). "Consistent groupwise non-rigid registration for atlas construction". In: *Biomedical Imaging: Nano to Macro, 2004. IEEE International Symposium on.* IEEE, pp. 908–911.

Boesen, Kristi, Kelly Rehm, Kirt Schaper, Sarah Stoltzner, Roger Woods, Eileen Lüders, and David Rottenberg (2004). "Quantitative comparison of four brain extraction algorithms". In: *NeuroImage* 22.3, pp. 1255–1261.

Bookstein, Fred L (2001). ""Voxel-based morphometry" should not be used with imperfectly registered images". In: *Neuroimage* 14.6, pp. 1454–1462.

Bostock, Michael, Vadim Ogievetsky, and Jeffrey Heer (2011). "D^3 data-driven documents". In: *IEEE Transactions on Visualization & Computer Graphics* 12, pp. 2301–2309. URL: https://github.com/mbostock/d3/.

Bots, Michiel L, Arno W Hoes, Peter J Koudstaal, Albert Hofman, and Diederick E Grobbee (1997). "Common carotid intima-media thickness and risk of stroke and myocardial infarction the Rotterdam Study". In: *Circulation* 96.5, pp. 1432–1437.

Bowman, Adrian W and Adelchi Azzalini (2004). *Applied smoothing techniques for data analysis.* Clarendon Press.

Callahan, Steven P, Juliana Freire, Emanuele Santos, Carlos E Scheidegger, Cláudio T Silva, and Huy T Vo (2006). "VisTrails: visualization meets data management". In: pp. 745–747.

Chen, YW, ME Gurol, J Rosand, A Viswanathan, SM Rakich, TR Groover, SM Greenberg, and EE Smith (2006). "Progression of white matter lesions and hemorrhages in cerebral amyloid angiopathy". In: *Neurology* 67.1, pp. 83–87.

Curcin, Vasa and Moustafa Ghanem (2008). "Scientific workflow systems-can one size fit all?" In: pp. 1–9.

Dalca, Adrian V et al. (2014). "Segmentation of Cerebrovascular Pathologies in Stroke Patients with Spatial and Shape Priors." In: *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2014*. Lecture Notes in Computer Science.

Daly et al, E. (2000). "Predicting conversion to Alzheimer disease using standardized clinical information". In: *Archives of Neurology* 57.5, p. 675.

Das, Samir, Alex P Zijdenbos, Jonathan Harlap, Dario Vins, and Alan C Evans (2011). "LORIS: a web-based data management system for multi-center studies". In: *Frontiers in neuroinformatics* 5.

Datar, Manasi, Prasanna Muralidharan, Abhishek Kumar, Sylvain Gouttard, Joseph Piven, Guido Gerig, et al. (2012). "Mixed-Effects shape models for estimating longitudinal changes in anatomy". In: *Spatio-temporal Image Analysis for Longitudinal and Time-Series Image Data*. Springer, pp. 76–87.

Davis, B.C., P.T. Fletcher, E. Bullitt, and S. Joshi (2007). "Population shape regression from random design data". In: *IEEE International Conference on Computer Vision*. IEEE, pp. 1–7.

Davis, Brad C, P Thomas Fletcher, Elizabeth Bullitt, and Sarang Joshi (2010). "Population shape regression from random design data". In: *Int J Comp. Vis.* 90.2, pp. 255–266.

Debette, Stéphanie, HS Markus, et al. (2010). "The clinical importance of white matter hyperintensities on brain magnetic resonance imaging: systematic review and meta-analysis". In: *Bmj* 341.

Delon, Remi (2014). *CherryPy-A Minimalist Python Web Framework*.

Durrleman, Stanley, Xavier Pennec, Alain Trouvé, José Braga, Guido Gerig, and Nicholas Ayache (2013). "Toward a comprehensive framework for the spatiotemporal statistical analysis of longitudinal shape data". In: *International journal of computer vision* 103.1, pp. 22–59.

Ehrhardt, Jan, René Werner, Alexander Schmidt-Richberg, Benny Schulz, and Heinz Handels (2008). "Generation of a Mean Motion Model of the Lung Using 4D-CT Image Data." In: *VCBM*, pp. 69–76.

Ellson, John, Emden Gansner, Lefteris Koutsofios, Stephen C North, and Gordon Woodhull (2002). "Graphviz-open source graph drawing tools". In: *Graph Drawing*. Springer, pp. 483–484.

Fazekas, F, R Kleinert, H Offenbacher, Rea Schmidt, G Kleinert, F Payer, H Radner, and H Lechner (1993). "Pathologic correlates of incidental MRI white matter signal hyperintensities". In: *Neurology* 43.9, pp. 1683–1683.

Fischl, B. (2012). "FreeSurfer". In: *NeuroImage* 62.2, pp. 774–781.

Fisher, M, Gregg Rothermel, Tyler Creelan, and Margaret Burnett (2006). "Scaling a Dataflow Testing Methodology to the Multiparadigm World of Commercial Spreadsheets". In: *Software Reliability Engineering, 2006. ISSRE'06. 17th International Symposium on*. IEEE, pp. 13–22.

Fonov, V.S., AC Evans, RC McKinstry, CR Almli, and DL Collins (2009). "Unbiased nonlinear average age-appropriate brain templates from birth to adulthood". In: *NeuroImage* 47.

Fonov, V.S., Alan C Evans, Kelly Botteron, C Robert Almli, Robert C McKinstry, D Louis Collins, Brain Development Cooperative Group, et al. (2011). "Unbiased average age-appropriate atlases for pediatric studies". In: *NeuroImage* 54.1, pp. 313–327.

Fox, G.C. and Dennis Gannon (2006). “Special issue: Workflow in grid systems”. In: *Concurrency and Computation: Practice and Experience* 18.10, pp. 1009–1019.

Ge, Tian, Thomas E Nichols, Debashis Ghosh, Elizabeth C Mormino, Jordan W Smoller, Mert R Sabuncu, et al. (2015). “A Kernel Machine Method for Detecting Effects of Interaction Between Multidimensional Variable Sets: An Imaging Genetics Application”. In: *NeuroImage.*

Gee, J.C., Martin Reivich, and Ruzena Bajcsy (1993). “Elastically deforming 3D atlas to match anatomical brain images.” In: *Journal of computer assisted tomography* 17.2, pp. 225–236.

Gerig, Guido, Brad Davis, Peter Lorenzen, Shun Xu, Matthieu Jomier, Joseph Piven, and S Joshi (2006). “Computational anatomy to assess longitudinal trajectory of brain growth”. In: *3D data processing, visualization, and transmission, Third International Symposium on.* IEEE, pp. 1041–1047.

Gibson, Erin, Fuqiang Gao, Sandra E Black, and Nancy J Lobaugh (2010). “Automatic segmentation of white matter hyperintensities in the elderly using FLAIR images at 3T”. In: *Journal of Magnetic Resonance Imaging* 31.6, pp. 1311–1322.

Gootjes, L, SJ Teipel, Y Zebuhr, R Schwarz, G Leinsinger, Ph Scheltens, Hans-Jürgen Möller, and H Hampel (2004). “Regional distribution of white matter hyperintensities in vascular dementia, Alzheimer’s disease and healthy aging”. In: *Dementia and geriatric cognitive disorders* 18.2, pp. 180–188.

Gorgolewski, K.J., C.D. Burns, C. Madison, C. Clark, Y.O. Halchenko, M.L. Waskom, and S.S. Ghosh (Aug. 2011). “Nipype: a flexible, lightweight and extensible neuroimaging data processing framework in python.” In: *Front Neuroinform* 5.

Gorgolewski, K.J., Tal Yarkoni, Satrajit S Ghosh, Russell A Poldrack, Jean-Baptiste Poline, Yannick Schwarz, and Daniel S Margulies (2013). “NeuroVault. org: A web database for sharing statistical parametric maps”. In: *19th Annual Meeting of the Organization for Human Brain Mapping.*

Grenander, Ulf and Michael I Miller (1998). “Computational anatomy: An emerging discipline”. In: *Quarterly of applied mathematics* 56.4, pp. 617–694.

Guimond, Alexandre, Jean Meunier, and Jean-Philippe Thirion (2000). “Average brain models: A convergence study”. In: *Computer vision and image understanding* 77.2, pp. 192–210.

Gurol, ME, MC Irizarry, EE Smith, S Raju, R Diaz-Arrastia, T Bottiglieri, J Rosand, JH Growdon, and SM Greenberg (2006). “Plasma β-amyloid and white matter lesions in AD, MCI, and cerebral amyloid angiopathy”. In: *Neurology* 66.1, pp. 23–29.

Haehn, D. (2013). "Slice:Drop: Collaborative Medical Imaging in the Browser". In: SIGGRAPH '13, pp. 1–1. DOI: 10.1145/2503541.2503645. URL: http://doi.acm.org/10.1145/2503541.2503645.

Hartkens, T., K. Rohr, and H.S. Stiehl (2002). "Evaluation of 3D operators for the detection of anatomical point landmarks in MR and CT images". In: *Computer Vision and Image Understanding* 86.2, pp. 118–136.

Huang, HK (2011). *PACS and imaging informatics: basic principles and applications.* John Wiley & Sons.

Hubert, H.B., M. Feinleib, P.M. McNamara, and W.P. Castelli (1983). "Obesity as an independent risk factor for cardiovascular disease: a 26-year follow-up of participants in the Framingham Heart Study". In: *Circulation* 67.5, pp. 968–977.

Hummel, Robert (1977). "Image enhancement by histogram transformation". In: *Computer graphics and image processing* 6.2, pp. 184–195.

Iglesias, Juan Eugenio, Cheng-Yi Liu, Paul M Thompson, and Zhuowen Tu (2011). "Robust brain extraction across datasets and comparison with publicly available methods". In: *Medical Imaging, IEEE Transactions on* 30.9, pp. 1617–1634.

Jack, C.R., Matt A Bernstein, Nick C Fox, Paul Thompson, Gene Alexander, Danielle Harvey, Bret Borowski, Paula J Britson, Jennifer L Whitwell, Chadwick Ward, et al. (2008). "The Alzheimer's disease neuroimaging initiative (ADNI): MRI methods". In: *Journal of Magnetic Resonance Imaging* 27.4, pp. 685–691.

Jack, C.R., Matt A Bernstein, Bret J Borowski, Jeffrey L Gunter, Nick C Fox, Paul M Thompson, Norbert Schuff, Gunnar Krueger, Ronald J Killiany, Charles S DeCarli, et al. (2010). "Update on the magnetic resonance imaging core of the Alzheimer's disease neuroimaging initiative". In: *Alzheimer's & Dementia* 6.3, pp. 212–220.

Jacobs, R.A., M.I. Jordan, S.J. Nowlan, and G.E. Hinton (1991). "Adaptive mixtures of local experts". In: *Neural computation* 3.1, pp. 79–87.

Jeerakathil, Tom, Philip A Wolf, Alexa Beiser, Joseph Massaro, Sudha Seshadri, Ralph B D'Agostino, and Charles DeCarli (2004). "Stroke Risk Profile Predicts White Matter Hyperintensity Volume The Framingham Study". In: *Stroke* 35.8, pp. 1857–1861.

Jenkinson, M., C.F. Beckmann, T.E.J. Behrens, M.W. Woolrich, and S.M. Smith (2012). "Fsl". In: *Neuroimage* 62.2, pp. 782–790.

Johnson et al, K.A. (1998). "Preclinical prediction of Alzheimer's disease using SPECT". In: *Neurology* 50.6, pp. 1563–1571.

Joshi, Sarang, Brad Davis, Matthieu Jomier, and Guido Gerig (2004). "Unbiased diffeomorphic atlas construction for computational anatomy". In: *NeuroImage* 23, S151–S160.

Killiany et al, R.J. (2000). “Use of structural magnetic resonance imaging to predict who will get Alzheimer’s disease”. In: *Annals of neurology* 47.4, pp. 430–439.

Kimeldorf, George and Grace Wahba (1971). “Some results on Tchebycheffian spline functions”. In: *Journal of mathematical analysis and applications* 33.1, pp. 82–95.

Klein et al, A. (2009). “Evaluation of 14 nonlinear deformation algorithms applied to human brain MRI registration”. In: *Neuroimage* 46.3, p. 786.

Koenig, Matthias, Wolf Spindler, Jan Rexilius, Julien Jomier, Florian Link, and Heinz-Otto Peitgen (2006). “Embedding VTK and ITK into a visual programming and rapid prototyping platform”. In: *Medical Imaging.* International Society for Optics and Photonics, 61412O–61412O.

Konstantinides, Konstantinos and John Robert Rasure (1994). “The Khoros software development environment for image and signal processing”. In: *Image Processing, IEEE Transactions on* 3.3, pp. 243–252.

Kuller, Lewis H, WT Longstreth, Alice M Arnold, Charles Bernick, R Nick Bryan, Norman J Beauchamp, et al. (2004). “White matter hyperintensity on cranial magnetic resonance imaging a predictor of stroke”. In: *Stroke* 35.8, pp. 1821–1825.

Laird, Angela R, Jack J Lancaster, and Peter T Fox (2005). “Brainmap”. In: *Neuroinformatics* 3.1, pp. 65–77.

Lancaster, J.L. and M.J. Martinez (n.d.). *Papaya.* URL: https://github.com/rii-mango/papaya.

Larrabide, Ignacio, Pedro Omedas, Yves Martelli, Xavier Planes, Maarten Nieber, Juan A Moya, Constantine Butakoff, Rafael Sebastián, Oscar Camara, Mathieu De Craene, et al. (2009). “GIMIAS: an open source framework for efficient development of research tools and clinical prototypes”. In: *Functional imaging and modeling of the heart.* Springer, pp. 417–426.

Liu, Dawei, Xihong Lin, and Debashis Ghosh (2007). “Semiparametric Regression of Multidimensional Genetic Pathway Data: Least-Squares Kernel Machines and Linear Mixed Models”. In: *Biometrics* 63.4, pp. 1079–1088.

Liu, Jingyu and Vince D Calhoun (2014). “A review of multivariate analyses in imaging genetics”. In: *Frontiers in neuroinformatics* 8.

Lloyd, Catherine M, Matt DB Halstead, and Poul F Nielsen (2004). “CellML: its future, present and past”. In: *Progress in biophysics and molecular biology* 85.2, pp. 433–450.

Lu, Hanzhang, Lidia M Nagae-Poetscher, Xavier Golay, Doris Lin, Martin Pomper, and Peter van Zijl (2005). “Routine clinical brain MRI sequences for use at 3.0 Tesla”. In: *Journal of Magnetic Resonance Imaging* 22.1, pp. 13–22.

Lucas, Bruce, Gregory D Abram, Nancy S Collins, David A Epstein, Donna L Gresh, and Kevin P McAuliffe (1992). “An architecture for a scientific visualization system”. In: *Proceedings of the 3rd conference on Visualization’92*. IEEE Computer Society Press, pp. 107–114.

Ludäscher, Bertram, Ilkay Altintas, Chad Berkley, Dan Higgins, Efrat Jaeger, Matthew Jones, Edward A Lee, Jing Tao, and Yang Zhao (2006). “Scientific workflow management and the Kepler system”. In: *Concurrency and Computation: Practice and Experience* 18.10, pp. 1039–1065.

Maechling, Philip, Hans Chalupsky, Maureen Dougherty, Ewa Deelman, Yolanda Gil, Sridhar Gullapalli, Vipin Gupta, Carl Kesselman, Jihic Kim, Gaurang Mehta, et al. (2005). “Simplifying construction of complex workflows for non-expert users of the Southern California Earthquake Center Community Modeling Environment”. In: *ACM SIGMOD Record* 34.3, pp. 24–30.

Marcus, Daniel S, Timothy R Olsen, Mohana Ramaratnam, and Randy L Buckner (2007). “The extensible neuroimaging archive toolkit”. In: *Neuroinformatics* 5.1, pp. 11–33.

Mayer, A and H Greenspan (2009). “An adaptive mean-shift framework for MRI brain segmentation”. In: *Medical Imaging, IEEE Transactions on* 28.8, pp. 1238–1250.

Mazziotta, John C, Arthur W Toga, Alan Evans, Peter Fox, and Jack Lancaster (1995). “A probabilistic atlas of the human brain: theory and rationale for its development the international consortium for brain mapping (ICBM)”. In: *Neuroimage* 2.2PA, pp. 89–101.

McCullagh, Peter (1984). “Generalized linear models”. In: *European Journal of Operational Research* 16.3, pp. 285–292.

McCulloch, Charles E and John M Neuhaus (2001). *Generalized linear mixed models.* Wiley Online Library.

Meschia, James F, Donna K Arnett, Hakan Ay, Robert D Brown, Oscar R Benavente, John W Cole, Paul IW de Bakker, Martin Dichgans, Kimberly F Doheny, Myriam Fornage, et al. (2013). “Stroke Genetics Network (SiGN) Study Design and Rationale for a Genome-Wide Association Study of Ischemic Stroke Subtypes”. In: *Stroke* 44.10, pp. 2694–2702.

Mierswa, Ingo, Michael Wurst, Ralf Klinkenberg, Martin Scholz, and Timm Euler (2006). “Yale: Rapid prototyping for complex data mining tasks”. In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, pp. 935–940.

Misra, Chandan, Yong Fan, and Christos Davatzikos (2009). “Baseline and longitudinal patterns of brain atrophy in MCI patients, and their use in prediction of short-term conversion to AD: results from ADNI”. In: *Neuroimage* 44.4, pp. 1415–1422.

Nadaraya, E.A. (1964). “On estimating regression”. In: *Theory of Probability & Its Applications* 9.1, pp. 141–142.

Nandigam, RN, YW Chen, Mahmut E Gurol, Jonathan Rosand, Steven M Greenberg, and Eric E Smith (2007). "Validation of intracranial area as a surrogate measure of intracranial volume when using clinical MRI". In: *Journal of Neuroimaging* 17.1, pp. 74–77.

Oinn, Tom, Matthew Addis, Justin Ferris, Darren Marvin, Martin Senger, Mark Greenwood, Tim Carver, Kevin Glover, Matthew R Pocock, Anil Wipat, et al. (2004). "Taverna: a tool for the composition and enactment of bioinformatics workflows". In: *Bioinformatics* 20.17, pp. 3045–3054.

Parker, S.G. and C.R. Johnson (1995). "SCIRun: a scientific programming environment for computational steering". In: p. 52.

Patel, Kayur, Naomi Bancroft, Steven M Drucker, James Fogarty, Andrew J Ko, and James Landay (2010). "Gestalt: integrated support for implementation and analysis in machine learning". In: *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. ACM, pp. 37–46.

Paulsen et al, J.S. (2008). "Detection of Huntington's disease decades before diagnosis: the Predict-HD study". In: *Journal of Neurology, Neurosurgery & Psychiatry* 79.8, pp. 874–880.

Pfefferbaum, Adolf, Torsten Rohlfing, Margaret J Rosenbloom, Weiwei Chu, Ian M Colrain, and Edith V Sullivan (2013). "Variation in longitudinal trajectories of regional brain volumes of healthy men and women (ages 10 to 85years) measured with atlas-based parcellation of MRI". In: *Neuroimage* 65, pp. 176–193.

Pieper, S. (2014). private communication. URL: https://github.com/pieper/chronicle.

Pieper, S., M. Halle, and R. Kikinis (2004). "3D Slicer". In: pp. 632–635.

Qiu, Anqi, Marilyn Albert, Laurent Younes, and Michael I Miller (2009). "Time sequence diffeomorphic metric mapping and parallel transport track time-dependent shape changes". In: *NeuroImage* 45.1, S51–S60.

Quandt, R.E. and J.B. Ramsey (1978). "Estimating mixtures of normal distributions and switching regressions". In: *Journal of the American Statistical Association* 73.364, pp. 730–738.

Queller, David C and Keith F Goodnight (1989). "Estimating relatedness using genetic markers". In: *Evolution*, pp. 258–275.

RapidMiner (Apr. 2012). Online. URL: http://rapid-i.com/content/view/181/190/.

Raz, Naftali, Yiqin Yang, Cheryl L Dahle, and Susan Land (2012). "Volume of white matter hyperintensities in healthy adults: contribution of age, vascular risk factors, and inflammation-related genetic variants". In: *Biochimica et Biophysica Acta (BBA)-Molecular Basis of Disease* 1822.3, pp. 361–369.

Regan et al, E.A. (2011). “Genetic epidemiology of COPD (COPDGene) study design”. In: *COPD: Journal of Chronic Obstructive Pulmonary Disease* 7.1, pp. 32–43.

Rex, D.E., J.Q. Ma, and A.W. Toga (2003). “The LONI pipeline processing environment”. In: *Neuroimage* 19.3, pp. 1033–1048.

Risacher, Shannon L, Li Shen, John D West, Sungeun Kim, Brenna C McDonald, Laurel A Beckett, et al. (2010). “Longitudinal MRI atrophy biomarkers: relationship to conversion in the ADNI cohort”. In: *Neurobiology of aging* 31.8, pp. 1401–1418.

Rohlfing, Torsten, Edith V Sullivan, and Adolf Pfefferbaum (2009a). “Regression models of atlas appearance”. In: *Information processing in medical imaging*. Springer, pp. 151–162.

— (2009b). “Regression models of atlas appearance”. In: *Info. Proc. Med. Img.* Springer, pp. 151–162.

Rost et al, N.S. (2010a). “White matter hyperintensity burden and susceptibility to cerebral ischemia”. In: *Stroke* 41.12, pp. 2807–2811.

— (2010b). “White matter hyperintensity volume is increased in small vessel stroke subtypes”. In: *Neurology* 75.19, pp. 1670–1677.

Rost, Natalia S, Rosanna Rahman, Shruti Sonni, Allison Kanakis, Christi Butler, Efi Massasa, Lisa Cloonan, Aaron Gilson, Pilar Delgado, Yuchiao Chang, et al. (2010). “Determinants of white matter hyperintensity volume in patients with acute ischemic stroke”. In: *Journal of Stroke and Cerebrovascular Diseases* 19.3, pp. 230–235.

Rueckert, D., Alejandro F Frangi, and Julia A Schnabel (2003). “Automatic construction of 3-D statistical deformation models of the brain using nonrigid registration”. In: *Medical Imaging, IEEE Transactions on* 22.8, pp. 1014–1025.

Rueckert, D. and J.A. Schnabel (2011). “Medical Image Registration”. In: *Biomedical Image Processing*. Biological and Medical Physics, Biomedical Engineering. Ed. by Thomas Martin Deserno, pp. 131–154.

Sadeghi, Neda, Marcel Prastawa, P Thomas Fletcher, Clement Vachet, Bo Wang, et al. (2013). “Multivariate modeling of longitudinal MRI in early brain development with confidence measures”. In: *2013 IEEE Inter. Symp. Biomed Imag.* IEEE, pp. 1400–1403.

Schmidt, P, C Gaser, M Arsic, et al. (2012). “An automated tool for detection of FLAIR-hyperintense white-matter lesions in multiple sclerosis”. In: *Neuroimage* 59.4, pp. 3774–3783.

Schmidt, Reinhold, Franz Fazekas, Gertrude Kleinert, Hans Offenbacher, Kurt Gindl, Franz Payer, Wolfgang Freidl, Kurt Niederkorn, and Helmut Lechner (1992). “Magnetic resonance imaging signal hyperintensities in the deep and subcortical white matter: a comparative study between stroke patients and normal volunteers". In: *Archives of neurology* 49.8, pp. 825–827.

Schroeder, William J, Lisa Sobierajski Avila, and William Hoffman (2000). "Visualizing with VTK: a tutorial". In: *Computer Graphics and Applications, IEEE* 20.5, pp. 20–27.

Smith, EE, J Rosand, KA Knudsen, EM Hylek, and SM Greenberg (2002). "Leukoaraiosis is associated with warfarin-related hemorrhage following ischemic stroke". In: *Neurology* 59.2, pp. 193–197.

Smith, Stephen M (2002). "Fast robust automated brain extraction". In: *Human brain mapping* 17.3, pp. 143–155.

Sridharan, R., A. V Dalca, and P. Golland (n.d.). "An Interactive Visualization Tool for Nipype Medical Image Computing Pipelines". In:

Sridharan, R., A.V. Dalca, K.M. Fitzpatrick, L. Cloonan, A. Kanakis, O. Wu, K.L. Furie, J. Rosand, N.S. Rost, and P. Golland (2013). "Quantification and Analysis of Large Multimodal Clinical Image Studies: Application to Stroke". In: *Multimodal Brain Image Analysis*. Springer, pp. 18–30.

Star, Susan Leigh (1989). "The Structure of Ill-Structured Solutions: Boundary Objects and Heterogeneous Distributed Problem Solving". In: *Distributed artificial intelligence* 2, pp. 37–54.

— (2010). "This is not a boundary object: Reflections on the origin of a concept". In: *Science, Technology & Human Values* 35.5, pp. 601–617.

Sutherland, William Robert (1966). "The on-line graphical specification of computer procedures". PhD thesis. Massachusetts Institute of Technology.

Talairach, J. and P. Tournoux (1988). "Co-planar stereotaxic atlas of the human brain. 3-Dimensional proportional system: an approach to cerebral imaging". In:

Tanabe, Jody L, Diane Amend, Norbert Schuff, Victoria DiSclafani, Frank Ezekiel, David Norman, George Fein, and Michael W Weiner (1997). "Tissue segmentation of the brain in Alzheimer disease." In: *American journal of neuroradiology* 18.1, pp. 115–123.

Taylor, Ian, Matthew Shields, Ian Wang, and Andrew Harrison (2007). "The triana workflow environment: Architecture and applications". In: *Workflows for e-Science*. Springer, pp. 320–339.

Thanprasertsuk, S, S Martinez-Ramirez, OM Pontes-Neto, J Ni, et al. (2014). "Posterior white matter disease distribution as a predictor of amyloid angiopathy". In: *Neurology* 83.9, pp. 794–800.

Tsai, JZ, SJ Peng, YW Chen, et al. (2014). "Automated Segmentation and Quantification of White Matter Hyperintensities in Acute Ischemic Stroke Patients with Cerebral Infarction". In: *PLoS one* 9.8, e104011.

Tukey, J.W. (1977). *Exploratory data analysis.* Pearson.

Tustison, N.J. and J.C. Gee (2009). "N4ITK: Nick's N3 ITK implementation for MRI bias field correction". In: *Insight Journal.*

Upson, Craig, Thomas A Faulhaber Jr, David Kamins, David Laidlaw, David Schlegel, Jefrey Vroom, Robert Gurwitz, and Andries Van Dam (1989). "The application visualization system: A computational environment for scientific visualization". In: *Computer Graphics and Applications, IEEE* 9.4, pp. 30–42.

Van Essen, David C, Kamil Ugurbil, E Auerbach, D Barch, TEJ Behrens, R Bucholz, A Chang, Liyong Chen, Maurizio Corbetta, Sandra W Curtiss, et al. (2012). "The Human Connectome Project: a data acquisition perspective". In: *Neuroimage* 62.4, pp. 2222–2231.

Van Horn, John Darrell and Arthur W Toga (2009). "Is it time to re-prioritize neuroimaging databases and digital repositories?" In: *Neuroimage* 47.4, pp. 1720–1734.

Van Leemput, Koen, Frederik Maes, Dirk Vandermeulen, and Paul Suetens (1999). "Automated model-based bias field correction of MR images of the brain". In: *Medical Imaging, IEEE Transactions on* 18.10, pp. 885–896.

Vercauteren, T., X. Pennec, A. Perchant, and N. Ayache (2008). "Symmetric log-domain diffeomorphic registration: A demons-based approach". In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2008*, pp. 754–761.

Viola, P. and W.M. Wells III (1997). "Alignment by maximization of mutual information". In: *International Journal of Computer Vision* 24.2, pp. 137–154.

Wahba, Grace (1990). *Spline models for observational data.* Vol. 59. Siam.

Walenstein, Andrew (2003). "Finding Boundary Objects in SE and HCI: An Approach Through Engineering-oriented Design Theories." In: *ICSE Workshop on SE-HCI.* Citeseer, pp. 92–99.

Wang, Lei, Jeffrey S Swank, Irena E Glick, Mokhtar H Gado, Michael I Miller, John C Morris, and John G Csernansky (2003). "Changes in hippocampal volume and shape across time distinguish dementia of the Alzheimer type from healthy aging". In: *Neuroimage* 20.2, pp. 667–682.

Warfield, Simon K, Kelly H Zou, and William M Wells (2002). "Validation of image segmentation and expert quality with an expectation-maximization algorithm". In: *Medical Image Computing and Computer-Assisted Intervention-MICCAI 2002.* Springer, pp. 298–306.

Watson, G.S. (1964). "Smooth regression analysis". In: *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 359–372.

Weiner et al, M.W (2012). "The Alzheimer's Disease Neuroimaging Initiative: a review of papers published since its inception". In: *Alzheimer's & Dementia* 8.1, S1–S68.

Yeo, BT Thomas, Mert R Sabuncu, Rahul Desikan, Bruce Fischl, and Polina Golland (2008). “Effects of registration regularization and atlas sharpness on segmentation accuracy”. In: *Medical image analysis* 12.5, pp. 603–615.

Yoo, B.I., JJ Lee, JW Han, et al. (2014). “Application of variable threshold intensity to segmentation for white matter hyperintensities in fluid attenuated inversion recovery magnetic resonance images”. In: *Neuroradiology* 56.4, pp. 265–281.

Yoo, T.S., M.J. Ackerman, W.E. Lorensen, W. Schroeder, V. Chalana, S. Aylward, D. Metaxas, and R. Whitaker (2002). “Engineering and algorithm design for an image processing API: A technical report on ITK-the Insight Toolkit”. In: *Studies in health technology and informatics*, pp. 586–592.

Yu, Jia and Rajkumar Buyya (2005). “A taxonomy of scientific workflow systems for grid computing”. In: *ACM Sigmod Record* 34.3, pp. 44–49.

Zijdenbos, Alex P, Benoit M Dawant, Richard A Margolin, and Andrew C Palmer (1994). “Morphometric analysis of white matter lesions in MR images: method and validation”. In: *Medical Imaging, IEEE Transactions on* 13.4, pp. 716–724.

Zijdenbos, Alex P, Reza Forghani, and Alan C Evans (2002). “Automatic” pipeline” analysis of 3-D MRI data for clinical trials: application to multiple sclerosis”. In: *Medical Imaging, IEEE Transactions on* 21.10, pp. 1280–1291.