International Journal of Psychophysiology xxx (2009) xxx–xxx

Contents lists available at ScienceDirect

International Journal of Psychophysiology

journal homepage: www.elsevier.com/locate/ijpsycho

Database-managed Grid-enabled analysis of neuroimaging data: The CNARI framework

Steven L. Small [a,b,c,*], Michael Wilde [c,d], Sarah Kenny [c], Michael Andric [b], Uri Hasson [e]

[a] *Department of Neurology, The University of Chicago, United States*
[b] *Department of Psychology, The University of Chicago, United States*
[c] *Computation Institute, The University of Chicago, United States*
[d] *Mathematics and Computer Science Division, Argonne National Laboratories, United States*
[e] *Center for Mind/Brain Sciences and Faculty of Cognitive Sciences, The University of Trento, Italy*

A R T I C L E I N F O

Article history:
Received 19 August 2008
Received in revised form 13 January 2009
Accepted 13 January 2009
Available online xxxx

Keywords:
fMRI
Imaging
Brain
Image analysis
Data analysis
Functional MRI
High-performance computing
Grid computing
Grid
Cluster computing
Database
Relational database
Data storage
Cortex
Language
Functional imaging
Infrastructure
Cognition
Cognitive neuroscience
Neuroscience
Neurology

A B S T R A C T

Functional magnetic resonance imaging (fMRI) has led to an enormous growth in the study of cognitive neuroanatomy, and combined with advances in high-field electrophysiology (and other methods), has led to a fast-growing field of human neuroscience. Technological advances in both hardware and software will lead to an ever more promising future for fMRI. We have developed a new computational framework that facilitates fMRI experimentation and analysis, and which has led to some rethinking of the nature of experimental design and analysis. The Computational Neuroscience Applications Research Infrastructure (CNARI) incorporates novel methods for maintaining, serving, and analyzing massive amounts of fMRI data. By using CNARI, it is possible to perform naturalistic, network-based, statistically valid experiments in systems neuroscience on a very large scale, with ease of data manipulation and analysis, within reasonable computational time scales. In this article, we describe this infrastructure and then illustrate its use on a number of actual examples in both cognitive neuroscience and neurological research. We believe that these advanced computational approaches will fundamentally change the future shape of cognitive brain imaging with fMRI.

1. Introduction

Functional magnetic resonance imaging (functional MRI or fMRI) has revolutionized the study of human neuroscience, and its future remains highly promising. Although it has been possible for many years to perform functional imaging of the human brain, using various radiographic and electrophysiological methods, it has only been since the advent of positron emission tomography (PET) that such imaging has had sufficient spatial resolution to contribute significantly to the assessment of brain–behavior relationships in humans. Prior to PET imaging, the primary way to assess the functional organization of the human brain was through the study of people with brain injuries, particularly focal brain injuries, trying to relate their behavior to the site or sites of injury (Broca, 1861; Geschwind, 1965). The underlying assumption of this endeavor is that when a brain region is injured, the resulting behavioral disorder reflects closely on the nature of the function performed by that region. Direct cortical stimulation represents a related "lesion-based" approach that is possible only with neurosurgical patients (Ojemann et al., 1989; Penfield and Boldrey, 1937). Electroencephalography (EEG) also has a history in structure/function mapping (Caton, 1875; Donchin et al., 1963). What

* Corresponding author. Department of Neurology, The University of Chicago, 5841 S. Maryland Ave., MC-2030, Chicago, IL 60637, United States. Tel.: +1 773 702 1600.
E-mail address: small@uchicago.edu (S.L. Small).
URL: http://www.humanneuroscience.org (S.L. Small).

0167-8760/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.ijpsycho.2009.01.010

PET provided for the first time was a source of information at high spatial resolution on brain–behavior relationships from the study of individuals without brain injury, who were performing carefully designed experimental tasks. Although fMRI can achieve greater temporal resolution than PET, its importance has come from its ubiquity and its non-invasiveness. Virtually every large hospital in the US and Europe has an MRI scanner, and MRI scans require no radiation exposure or blood sampling.

This ubiquity of fMRI has led to an enormous growth in the study of cognitive neuroanatomy, and combined with advances in high-field electrophysiology (and other methods), has led to a fast-growing field of human neuroscience. The future of fMRI is just as bright as its recent past, with enhanced scientific productivity coming about from technological advances in hardware and improvements in methods of experimental design and analysis. In this article, we focus on a new computational framework that facilitates fMRI experimentation and analysis, and which has led to some rethinking of the nature of experimental design and analysis. We believe that the advanced computational approaches that we describe here will fundamentally change the future shape of cognitive brain imaging with fMRI.

It might be thought that computational infrastructure *per se* does not impact fundamentally on the scientific enterprise, except by virtue of speeding up tasks that could be performed anyway, but more slowly. We take an entirely different view, and believe that both the science of computation and the existence of high-performance computing can change the course of research, by qualitatively altering the nature of questions asked, the turnaround in getting partial results, and the development of theory. Questions that would not have been asked before due to their computational complexity or need for complex simulations can be answered in realistic timeframes using strong computational resources. We will illustrate with one example from our laboratory. In recent years, it is less common to ask questions about the role of focal brain activations in subserving behavioral functions, and more common to ask questions such as (a) how such focal brain regions interact with other regions to achieve these functions; and (b) how these interactions change with even slight modifications in behavior. Examining connections over individual regions magnifies the computational demands to unimaginable proportions, due to the combinatorial explosion inherent in studying sets of connected regions. This is especially true when researchers enjoy finer and finer resolution, which in the future may very well enable routine sub-millimeter resolution in fMRI. Furthermore, slight changes in the definitions of regions or in the method of combining individual voxels into representative regional time series lead to a computationally intensive iterative approach to such connectivity-based research. Theories based on connectivity are inherently different from those based on individual regional activation, lead to different conclusions about brain/behavior relations and have only been possible due to high-performance computing. Although we do not currently work at very high degrees of resolution, the same issues apply when one considers the relation between number of regions in the brain and their connections (e.g., see Felleman and Van Essen, 1991) or the number of voxels in a typical brain image and their potential interactions (e.g., see Biswal et al., 1995).

Our new approaches are motivated first by some very practical observations about the way that computation is currently performed in brain imaging laboratories and second by several assumptions about the future of human cognitive neuroanatomy research. In both cases, advanced high-performance computation can play an important role in insuring a future enterprise that is effective, reliable, and necessary.

Practically speaking, current computational practices for brain imaging are *ad hoc* and inefficient. The current brain imaging laboratory operates within a computational infrastructure that complicates a number of simple tasks and fails to provide various assurances that could be more easily accomplished within an alternative computational framework. Brain images are stored in hierarchical file structures ("directories"), with raw and processed images, functional images of multiple formats, and different types of structural images all in different parts of the file system. Stimuli are typically stored in completely different parts of the file system or even on different computers altogether (e.g., on stimulus-presentation computers). Behavioral data that accompany the imaging study are found in spreadsheets, again in different parts of the file system or on yet another set of computers. Often, raw data are also stored in backup form on compact disks or DVDs. Thus, the relevant information for a study is distributed across a large intricate set of files and computers. Data processing workflows to prepare and analyze these data are often very complicated, and do not easily permit accountability of data provenance, i.e., it is not always clear what were the intermediate steps and parameters of the analysis nor how to reconstruct these exact steps later for validation or replication, or to test alternative hypotheses.

Speaking speculatively on human cognitive neuroanatomy research, we present a number of assumptions that could enhance the future of brain imaging but can only be achieved with a novel computational infrastructure. The first assumption is that task design for functional imaging studies should be as natural as possible and should not include the types of ancillary tasks that are common parts of experimental psychology experiments (i.e., decision-making and manual responses) but represent confounds for image interpretation. This demand leads to particular computational needs and requires unique solutions for analysis and data representation. Second is that imaging data are most productively stored in relational form, rather than in hierarchical file systems. Third is that the result of a brain imaging study should be a network of interacting activation components, rather than a map of the activations *per se*. This treats the activations *and* their statistical and anatomical relationships as having equal importance for scientific inference. Fourth is that permutation-based statistics (randomization approaches) can replace analytical parametric-based statistics for determination of statistical significance in brain imaging studies, and avoid some of the most difficult confounds in statistical inference from brain images (e.g., quantifying and accounting for voxel independence).

In this report, we describe an ongoing project at the Human Neuroscience Laboratory and Computation Institute at The University of Chicago, which we call the Computational Neuroscience Applications Research Infrastructure (CNARI), which aims to develop novel methods for maintaining, serving, and analyzing massive amounts of fMRI data. By using CNARI, it is possible to work within the four premises described above for brain imaging experimental design and analysis, and perform naturalistic, network-based, statistically valid experiments in systems neuroscience. In the current article, we describe this infrastructure and then illustrate its use on a number of actual examples in both cognitive neuroscience and neurological research.

2. The CNARI infrastructure

The CNARI infrastructure consists of two integrated elements: (1) a database system for storing and retrieving brain imaging and behavioral data (Hasson et al., 2008), and (2) a collection of data pre-processing and analysis procedures written in the parallel scripting language SwiftScript (Zhao et al., 2007), which capture the essential "workflows" of our research group (Stef-Praun et al., 2007) and speed up their execution using both local high-performance computers and distributed computing resources (Foster, 2002, 2003; Foster and Tuecke, 2001). Public consortia such as the Open Science Grid (Pordes et al., 2007) and the TeraGrid (Beckman, 2005) pool the processors and storage systems of multiple organizations into large-scale shared computing facilities that are then made available to members of "virtual organizations" (Foster and Tuecke, 2001) based on various allocation schemes. The Swift system enables CNARI SwiftScript programs to execute on these distributed computing systems in a manner that is simpler and more transparent than otherwise possible, and thus to perform larger volumes of data preparation and analytical work at higher speeds and with greater reproducibility. Through this integration, CNARI combines database-centered data modeling and storage with high-throughput parallel computing to facilitate research at the functional MRI laboratory at The University of Chicago.

2.1. Database representation and access

In relational database management systems, data are not stored in separate user-accessible files, but are encoded in a tabular internal representation that reflects relations among data elements or tables of such elements. Because of their flexibility, uniformity, maintainability, and query (essentially "question answering") capabilities, relational databases are being increasingly applied in scientific domains as scientists wrestle with the exponential growth and increased complexity of their data sets (Szalay and Gray, 2006). For scientists, they offer benefits in terms of data sharing, structured representation, searchability, management, leverage of metadata, and scalability. Such advantages of database approaches over file-based approaches are becoming clear in a growing number of disciplines (Gray et al., 2005).

fMRI analyses typically use and generate a vast number of data files. Many types of functional MRI time series (e.g., unregistered, registered, detrended, despiked, error terms) are analyzed to generate various statistical maps, both at an individual subject and group level. Group-level statistical maps might reflect the results of various types of statistical analyses such as analysis of variance (ANOVA), principal components analysis (PCA), *t*-tests, and a multitude of non-parametric tests. Together, the number of flat files generated (i.e., linear unstructured data stored in files and organized in directories) can become large and the entire set is typically complex, difficult to manage, and enormous in size. Databases offer a practical and elegant solution to the management of this enormous amount of related data by storing them in a manner that makes clear the relationships among data elements, and that makes it possible to easily extract highly specific subsets of those data via sophisticated queries, such as "select all voxels with BOLD signal change $\geq 0.3\%$ from the left ventral premotor cortex of all left-handed male subjects with age >65, who also showed mean activation across both supramarginal gyri at a corrected global p value<0.05".

We believe that fMRI analysis tools should – and can – interface with a database management system, and that such integration provides significant advantages to the researcher. Current data analysis tools (e.g., AFNI (Cox, 1996), SPM (Friston et al., 1990, 1991), BrainVoyager (Goebel, 1997), and FSL (Smith et al., 2004)) are integrated packages that use flat files to save data throughout the analysis flow, and allow users to invoke statistical procedures using integrated commands or extensions. Using a database as a storage system for these tools would allow users to access data via database queries (rather than from a file) thus benefiting from database features described above, while still retaining a familiar working environment. Recent additions to AFNI enable this model by supporting the passing of sparse data queried from databases directly as input to AFNI applications.

In addition, many software systems and programming languages (e.g., R (Gentleman and Ihaka, 1997), Matlab (Gilat, 2007), Excel (Billo, 2007), Perl (Wall, 2000), Python (van Rossum and de Boer, 1991), C/C++ (Ritchie, 1993; Stroustrup, 2000) and Java (Gosling et al., 2000)) have powerful and well-supported interfaces to relational databases, which allow for parallelized, concurrent, and access-controlled query and processing by multiple users (beyond those who collected the data).

In CNARI, replication is used to mirror parts of the database to remote sites to offer load balancing when needed. Database "views" can be instantiated to cache specific queries that represent frequently used subsets of the data. Furthermore, views can offer limited access to particular cross sections of the data to remote collaborators. Indeed, remote collaboration and data sharing represent main emphases in CNARI, and these are easily achieved by running analysis routines that access databases via the Internet.

2.2. Grid resources and the SwiftScript language

Achieving some of our goals for brain imaging research requires high-throughput computing and infrastructures focused on supporting such research (Beckman, 2005; Pordes et al., 2007). We use grid computing technologies with existing grid infrastructures. We thus define the "Grid" to refer to any one of a number of large, distributed computing services composed of (possibly overlapping) infrastructures of this type, used in common and "on demand" by a wide group of scientists. From the end-user's point of view, the Grid is a powerful, multi-user computer, with familiar resource sharing and user access mechanisms. Grid resources are shared according to specified policies or may be reserved upon request. The security of each user's applications and data is enforced by public key infrastructure (PKI), which maps Grid users to local resource permissions which often provide enhanced access control list security. Production Grids provide storage services, large-scale execution services, high-performance data movement utilities, and supporting tools that allow users to take advantage of large amounts of computing power. These capabilities can conveniently address specific requirements of medical research, such as access control to patient data, as mandated by HIPAA rules in the USA; high bandwidth to rapidly transfer large DICOM images from the patient's records (Erberich et al., 2007; Hastings et al., 2005); and sophisticated image analysis algorithms to aid in the interpretation of medical conditions.

Core middleware toolkits, such as Globus (Foster, 2005; Foster and Kesselman, 1999) and Condor (Litzkow et al., 1988; Tannenbaum and Litzkow, 1995), on which Grids are based, do not make the use of Grid resources transparent to the scientific user. Swift, a higher-level middleware component, is designed to address this need by allowing specification of a script-like mode of executing applications in a loosely-coupled manner (Ousterhout, 1998) in which many of the details of distributed parallel computing are automated and made transparent to the user. Using Swift as a "workflow" system simplifies parallel distributed Grid computing by automating the selection of distributed resources for execution, the transfer of input data and output results files to and from the execution site, the retry of alternate sites, and the rate throttling of these execution and data management operations.

Swift promotes the abstraction and encapsulation of data processing procedures by allowing users to specify how to invoke scientific applications (e.g., fMRI image processing tools like AFNI) and analytical tools (e.g., R statistical procedures) in a manner that explicitly specifies the input and output data sets and parameters of each procedure, but need not address the cluttering details of how to execute the application in diverse distributed computing environments.

Swift also allows the user to express data parallelism, via "foreach" statements that logically process all members of a data collection in parallel. This makes Swift ideal for massively parallel tasks. The actual physical degree of parallelism of such constructs is not specified by the user and is bounded only by the number of elements in the data collection. Swift decides and dynamically adjusts at runtime, based on the availability and performance of resources, what degree of parallelism it should actually use at any given time. Further, within pipelines that the user specifies in Swift scripts by simply chaining the output data set of one procedure to the input of another, Swift determines automatically at runtime what parts of the pipeline can be run in what order, based on the dynamic availability of dependent data sets.

Swift encapsulates application programs (which can themselves be scripts written in any other scripting language such as Perl, Python or a shell) with an interface definition that makes clear what parameters and data sets are passed into the program, and what results are returned. It is this functional model (Hudak, 1989) of well-defined inputs and outputs that enables both the location transparency and automated parallelization defined above, as well as the ability to reproduce and audit the results.

Swift procedures can be defined in a nested, compound manner, as in a typical programming language, to form hierarchical libraries of increasingly sophisticated scientific functionality. But at every level of such a library, the same functional abstraction, location independence, and parallel execution capabilities are afforded. The same Swift script can be executed without change on a local workstation, a cluster, a grid of clusters, or a highly parallel supercomputer such as the IBM Blue Gene/P (Overview of the IBM Blue Gene/P project, 2008).

Swift represents collections of file system data as abstract data set types with a simplified logical structure and uses mappers to convert between this logical structure and the physical on-disk representations of that structure. This mechanism is particularly well suited to the typical manner in which research data for neuroscience studies are stored, where, for example individual images or time series of images are stored in metadata/voxel file pairs (as in the Analyze format (Robb & Hanson, 1990), and then composed upwards into nested collections for functional runs, subjects, experiments, and entire studies. Simple examples are shown in Appendix A and in http://www.ci.uchicago.edu/swift/guides/tutorial.php.

Because Swift workflows are specified at a logical level, without environment-specific details, they serve better than *ad hoc* scripts written in a lower-level scripting language such as Perl or Python, or various shells, for enabling reproducibility and providing provenance tracking of resulting data sets. This in turn can enable collaboration in the actual research process, not only in sharing the results. This approach is discipline-independent, and has shown benefits in several other domains (e.g., radiology, bioinformatics, biochemistry, economics) in which we also have ongoing collaborations (Raicu et al., 2008; Stef-Praun, 2005).

For example, a researcher can express a workflow in the form of a Swift script utilizing multiple analysis tools and packages as a single function and allowing Swift to parallelize independent processes implicitly and send them to remote Grid sites.

A trivial example of a Swift script:

```
type image; // type "image" represents a single image
file.
app (image output) rotate(image input, int angle) {
   convert "-rotate" angle @input @output;
}
image brain <"subject1.jpg">;
int angles[] = [45, 90, 120];
foreach a in angles {
   image output <single_file_mapper; file=@strcat
   ("rotated-",a,".jpeg")>;
   output = rotate(brain, a);
}
```

In this example, the "rotate" function encapsulates a common image processing tool. Each call to rotate requested in the "foreach" loop is executed in parallel.

Appendix A shows a more complex example. For further details of Swift scripting, readers are directed to http://www.ci.uchicago.edu/swift/docs/index.php for a Users Guide and tutorials.

During a typical functional MRI experiment, the scanner collects data from around 70,000 voxels in the brain, and during the transformations carried out during the analyses, these 70,000 time series can increase to ~400 K functional time series per participant via interpolation. This massive amount of data requires computationally intensive operations and calls for unique methods of storage, mining and workflow management, particularly when analysis occurs in a grid environment. The Swift workflow system permits us to analyze these data sets on Grid resources, providing a location-independent way to specify logical computations at a high level suitable for use by non-programming scientists. The Swift system maps these high-level expressions of scientific computing into parallelized workflows that perform automated transportation of data sets to and from remote execution sites. Our analysis routines have been executed at locations across the TeraGrid including Argonne National Laboratory, the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign (NCSA), the Texas Advanced Computing Center (TACC), and Indiana University, as well as our own local test cluster, at which over 72,000 processors are available for shared use. A sample workflow is shown in Fig. 1. All the analysis methods and software routines we discuss below are submitted to TeraGrid sites via Swift.

In our laboratory, analysis relies on open source (or other freely distributed) software, including the "R" statistical language (Gentleman and Ihaka, 1997), Python (van Rossum and de Boer, 1991), AFNI (Cox, 1996), SUMA (Saad et al., 2004), and FreeSurfer (Dale et al., 1999; Fischl et al., 1999), all of which are highly portable and execute without change on numerous TeraGrid sites. Swift automatically tries to run jobs on the most responsive Grid sites. It automatically balances jobs across Grid sites and dynamically favors sites that are completing work more productively. This capability enables us to split large jobs into smaller independent units that run on many CPUs across multiple sites in parallel.

To reduce workflow queuing delays at remote Grid sites, jobs are submitted with an estimated specification of their expected run time. In addition to splitting large workflows over many sites, for efficiency Swift can then automatically batch very short jobs (such as trivial operations on huge collections of images) back together into longer, more efficient jobs via a 'clustering' mechanism whose purpose is to reduce the queue wait time overhead involved in job submission. When a clustered job is dispatched to a Grid node for computation, all jobs in the cluster are executed without having to wait in the queue in between. Thus, Swift reduces manual effort by automatically managing the subtle tradeoff between queue time and parallelization.

3. Examples from the use of CNARI

3.1. Extraction of cortical surfaces

Given that different individuals have different brain anatomy, drawing conclusions about *groups* of participants relies on procedures whose purpose is to register different brain anatomies to a common standard template. One such procedure performs automatic segmentation (separation) of gray matter and white matter areas in the brain, extraction of the gray matter area, and inflation of the gray matter gyri and sulci to a two-dimensional surface. This procedure requires between 40 and 60 h on a single machine, per individual participant.

3.2. Permutation methods for fMRI statistical analysis

Carrying out statistical analysis for the purpose of hypothesis testing is perhaps the most complicated and contentious procedure in fMRI data analysis. The aim of such analyses is to identify the brain regions that show differential activation profiles for different experimental conditions (e.g., areas that differentiate auditory from visual stimuli). We typically identify such regions by finding clusters of voxels that differentiate among these conditions. The likelihood that such clusters are found by chance (a false positive) is quite large since

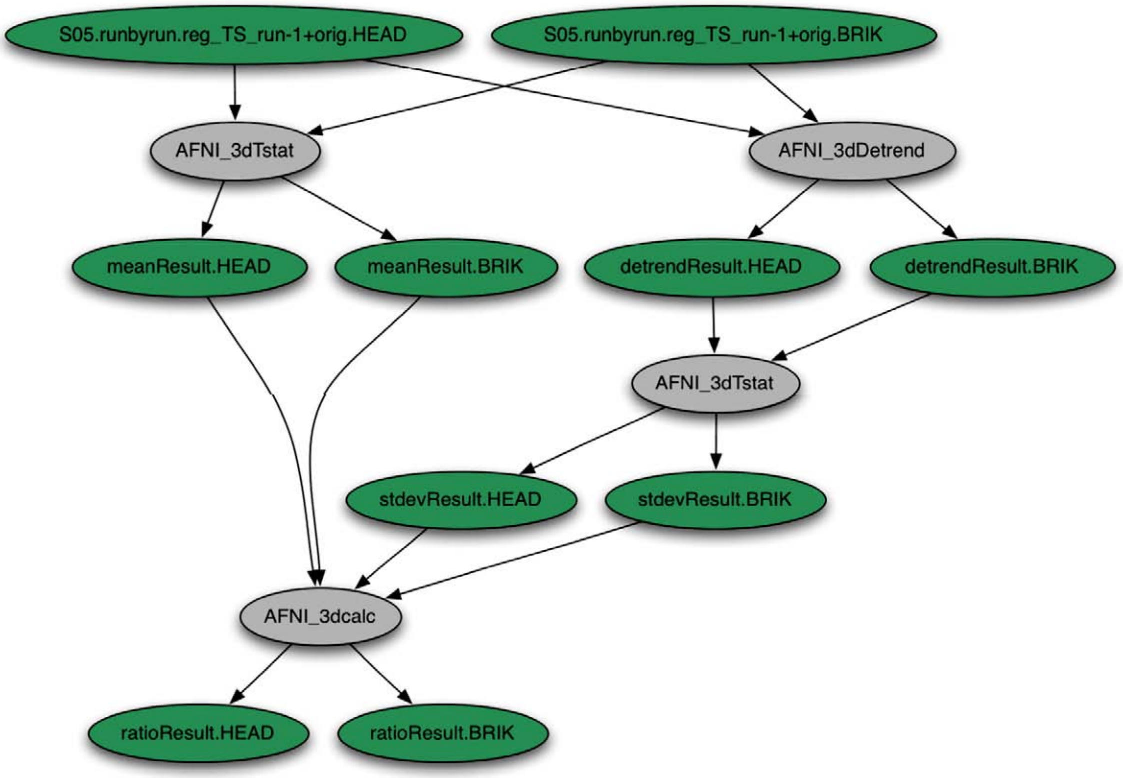

Fig. 1. Sample Grid-optimized workflow in neuroimaging analysis. Swift enables researchers to declare data dependencies in their workflows so that they can be scheduled for parallel computation on Grid resources. The workflow graph in the Figure exemplifies a procedure for calculating a simple measure of signal-to-noise ratio (SNR). The workflow contains two independent branches, each of which is processed on separate computing nodes, possibly on different Grid sites. Nodes in green indicate input files and nodes in gray indicate functions operating on those files. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

many thousands of voxels are tested for the experimental effect leading to a high probability that clusters of certain size will form by chance. This chance probability is also greatly determined by the inherent spatial smoothing of the fMRI image. Thus, the only way to estimate the likelihood of a false positive, given the inherent smoothing in the image, depends on repeatedly generating permuted versions of the original data, and then statistically evaluating the data and clustering the results. The theoretical justification for such procedures has been extensively supported in the domain of fMRI data analysis and their adequacy shown via simulations (Biswal et al., 1995; Nichols and Holmes, 2002). An example of permutation output is given in Fig. 2. We have previously documented the applicability of grid environments to such permutation-based analyses, which naturally lend themselves to distributed and parallel computation, since each computing node can be assigned to generate a small set of permutations and analyze their statistical features (Stef-Praun et al., 2007).

Generating a sufficient number of permutations ($n=3000$) for even a single statistical test can use as many as 250 CPU hours. We have implemented permutation procedures for several of the essential statistical tests used in our field (between and within participant contrasts, using both parametric (t, F tests) and non-parametric tests (Fisher, McNemar). A typical research project entails generating permutations for 6–8 statistical tests of interest. Thus, permutation analyses are only practical within a framework such as CNARI that employs high-performance computing.

3.3. Connectivity analyses

3.3.1. Functional connectivity: correlational analysis

One of our main research tenets is that brain imaging studies need to report not only the areas of activation but the nature of the interactions among these areas. Both the activations *and* their statistical and anatomical relationships are valuable information in understanding brain function for cognitive process. Two different types of network analysis have been contrasted, functional connectivity and effective connectivity. The former relates to the statistical dependencies among remote neurophysiological measurements, whereas the latter describes the statistical influence that one neuronal system exerts over another (Marrelec et al., 2008).

The simplest approach to network analysis is a functional connectivity approach that involves cross-correlation of the time course of activation of a voxel of interest (or an average of the time courses of a cluster of such voxels) with the time course of some or all other voxels in the brain (Biswal et al., 1995). This can be performed in the regular geometric space defined by the activation pattern, or in the space defined by the eigenvectors. These methods have an established history in functional brain imaging (Friston et al., 2000, 1993). Although dependent on the resolution of the images, functional images can contain as many as 400,000 voxels. Thus the computation of functional connectivity is typically constrained to one or two "seed" regions of maximal interest. The computation for a single seed region typically takes about 3 computing hours per participant per seed region (60 computing hours per group analysis). If the goal were to perform such computations on the entire brain, using many anatomical or functional seeds, it would be critical to employ a system like CNARI (Hasson et al., 2008).

3.3.2. Effective connectivity: structural equation modeling

Structural equation modeling (SEM) is an extension of correlational methods that uses known anatomy to augment the functional information with structural connectivity information, to create a

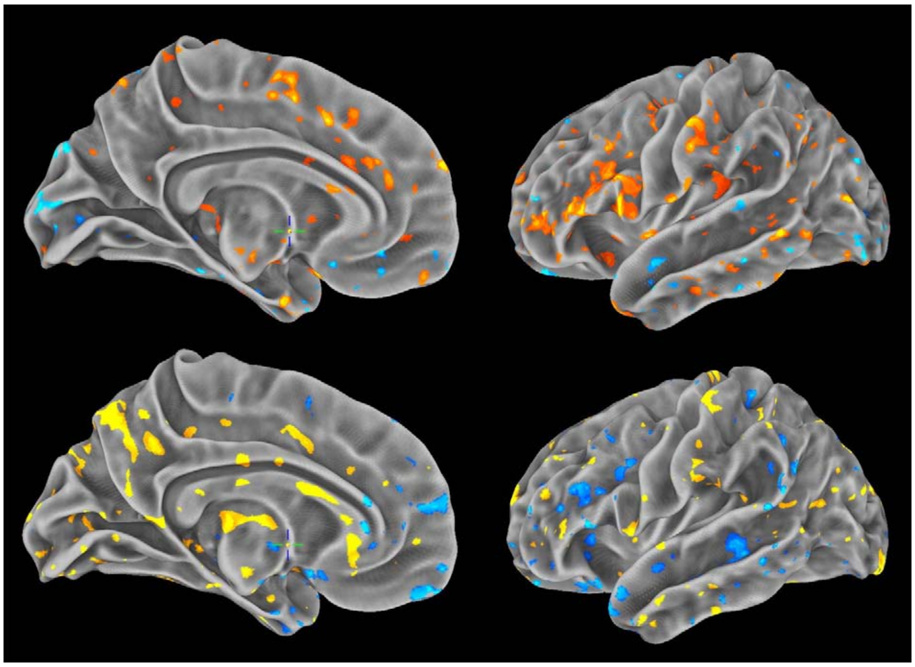

Fig. 2. Graphical depiction of permutations of original data. Brain activity data were collected from 24 participants that were exposed to two experimental conditions. A single permutation consisted of (a) switching the condition labels of the experimental conditions, for 1 or more participants, (b) re-calculating the group-level effects, and (c) spatially clustering the results to identify what is the largest cluster found for the current permutation. This procedure results in a sampling distribution for the largest cluster in each permutation, which is then used to threshold the original data.

model of both static and dynamic relationships (Horwitz et al., 1999; McIntosh et al., 1994). We have developed a connection matrix for the areas involved in audiovisual story comprehension, based on a combination of primate and human data (Ban et al., 1984; Ban et al., 1991; Barbas, 2000; Hackett et al., 1999; Petrides and Pandya, 1984, 1988, 1999; Rizzolatti et al., 1997, 1998; Rosa et al., 1993; Seltzer and Pandya, 1994). Using this network, a structural equation (path analysis) model was constructed based on these anatomical areas, using the Amos™ software (Arbuckle, 1989).

SEM has certain limitations, one of which is that it does not lead to a unique solution. There are many possible models that are a good fit to the anatomy and to the data, but it has generally been impossible to explore the large space of models. We have developed an unbiased approach to the construction of such network models (Skipper et al., 2007). Although the approach requires massive amounts of computational resources, requiring use of cluster and grid computing methods (Foster and Tuecke, 2001; Hasson et al., 2008; van Horn et al., 2005), it has the advantage that the resulting model can be assured to be the "best" model to describe the system in a formal mathematical sense. We have now implemented this approach in Swift and it now forms part of the CNARI infrastructure.

We used the system to explore the neural circuitry underlying gesture, building a model of the brain regions involved in audiovisual discourse comprehension. Following preliminary analysis, time series data from each of the conditions from active areas ($p<0.05$ corrected) were analyzed by exhaustive search of all possible structural equation models (SEMs) for five regions of interest: anterior and posterior portions of the superior temporal gyrus (aST and pST), ventral and dorsal segments of premotor cortex (vPM and dPM), and the supramarginal gyrus (SMG). SEMs with a "good fit" were combined by Bayesian Model averaging (Hoeting et al., 1999; Kass et al., 2005), culminating in a single SEM characterizing the independent influence of each area on the others in the model. This demonstrated two distinguishable motor networks associated with facial vs. manual gestures: Fig. 3 shows that when the hands are resting, there are strong connection weights among pST, vPM, and aST, but when they are gesturing, the strongest connections are among SMG, dPM, and aST. Furthermore, the mean weight of aST connections was strongest during the gesture condition.

In an exhaustive SEM approach used in our previous study (Skipper et al., 2007), all possible models are considered via complete search, with the Bayesian information measure used to rank models that explain most of the variance in the statistical interactions among brain regions (Hoeting et al., 1999). The Bayesian information criterion adjusts the χ^2 of each model for the number of parameters in the model, the number of observed variables, and the sample size. Individual connection weights are compared for the different models using *t*-tests correcting for heterogeneity of variance and unequal sample sizes by the Games–Howell method, with degrees of freedom calculated using Welch's method (Kirk, 1995). See (Skipper et al., 2007) for a more detailed description of this method.

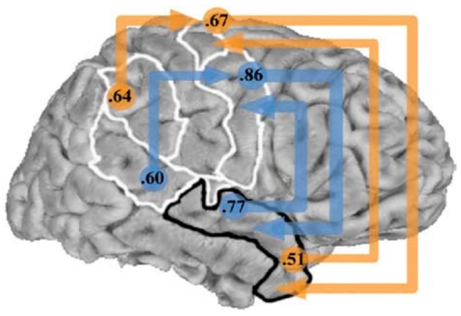

Fig. 3. Structural equation model for audiovisual language comprehension, comparing a condition in which the speaker is using normal gestures to one in which she keeps her hands on her lap to a third in which she is not visible at all (audio-only). Orange connections distinguish the gesture condition from the others. Blue connections distinguish the hands resting condition from the others. Connection weights are shown at the beginning of the arrowed lines. The model was not lateralized (shown here on the right.)

3.4. Simulations

Simulation methods provide a useful method for understanding statistical properties of fMRI data and are used in a myriad of applications, e.g., determining significance for cluster-level effects, permutation-based procedures and others. Complex simulations take a considerable period to complete and distributed computing offers a useful method for returning results in a reasonable time frame. We describe below how distributed computing may be used for such purposes by describing a signal-to-noise ratio (SNR) simulation method that is typically conducted prior to data analysis.

As pointed out by Parrish et al. (2000), when interpreting the results of a fMRI analysis it is important to establish what is the minimum SNR under which experimental effects could be reliably identified. Once the minimum SNR is established, the researcher can examine their own data set and observe whether the activity patterns are sensible or not, as "activity" in an area with low SNR would be highly suspect. Determining what SNR is needed for finding an experimental effect in a particular study is achieved via a simulation procedure. This procedure determines, for each SNR level, what is the probability of identifying a signal change of a given magnitude, e.g., 1%.

The simulation procedure is conducted as follows. First, the experimenter creates a schematic model of the expected activity, typically by convolving a boxcar design with a canonical hemodynamic response function (HRF); this is the predictor model. To this model, smoothed noise of a certain magnitude (defined by the variance of the noise distribution) is added to obtain a particular SNR ratio (the higher the variance of the noise, the lower the SNR). Ten thousand such "noisy" time series are created per SNR level. Once these 10,000 series are created, it is possible to determine the likelihood of finding a reliable correlation for the given SNR level, by fitting each of the "noisy boxcar" time series to the original predictor. If a high proportion of the 10,000 simulations are found to correlate, this would indicate that for the given SNR level, the chances of finding a "true positive" are quite high (see Fig. 4 for example result).

The simulation process (using R) is quite slow, and takes about 90 s for a single SNR value, since each simulation involves generating 10,000 time series and 10,000 regressions. Given that 70 SNR values are needed to sample the noise space, the process is quite time consuming (about 3 h per study, when simulating for a given signal change, e.g., 1%). Using distributed computation on multiple Grid sites, this task can be split into multiple compute nodes, each assigned with one or more jobs.

It should be noted, however, that splitting the simulation into multiple nodes does not mean that the analysis time is reduced in direct proportion to the number of nodes used, since the workflow system, which among other things tracks the job submission, sends and retrieves files from the Grid clusters, and chooses new sites for submission, imposes additional overhead on the process. Fig. 5 demonstrates how the analysis time for a 70-job SNR simulation changes as a function of number of nodes. The figure describes processing times for configurations ranging from 2 potentially available nodes (each running 35 jobs) to 70 potential nodes (each running a single job). When operating in what might be considered "Forced Mode", the workflow system is configured to trust the cluster (s) to which it is submitting jobs. This increases the trust rank of the cluster so that the workflow system allows itself to submit a large number of parallel jobs to the cluster. In this case, increasing the number of nodes quickly achieves rapid time saving: For example, increasing the number of nodes from 2 to 8 shortens processing time from 91 to 28 min. However, this mode of operation is unreasonable when working against multiple Grid sites, whose reliability can change at any given moment. In that case, the workflow system is typically configured to operate so that it updates its knowledge of Grid sites by initially submitting a small number of jobs to multiple sites, assessing responsiveness, and recalibrating its submission policy. This ensures, e.g., that 70 jobs would not be submitted to an unreliable cluster that might start processing them but then 'hang' without delivering results. This mode of operation results in slower processing if not only for the fact that not all jobs are submitted initially. The timings for such a mode of operation are shown in Fig. 5 ("Learning Mode"), which demonstrates that this mode of operation is associated with overall longer response times, and it is only with a relatively larger number of available nodes that a marked reduction in processing time is found. In this mode, even when 70 nodes are available, processing time is quite long (12 min, vs. 3 min in "Forced Mode").

Developing methods for optimizing the learning process for such neuroimaging analyses is one of the goals of the CNARI project and both applied and theoretical work needs to be done to better optimize these modes of operation.

3.5. Grid-enabled database queries

Another central aspect of our work has been to use database management systems to increase the flexibility and parallelization potential of fMRI analyses. We have successfully used databases for

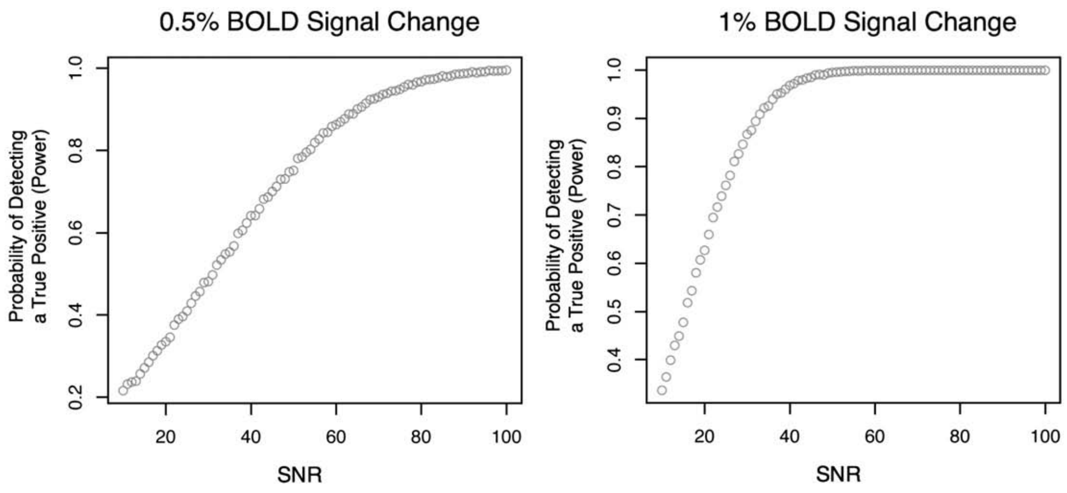

Fig. 4. Sample results of SNR simulation. The plot represents the probability of detecting a true positive (power) for different SNR levels, assuming an effect size of 0.5% or 1% signal change.

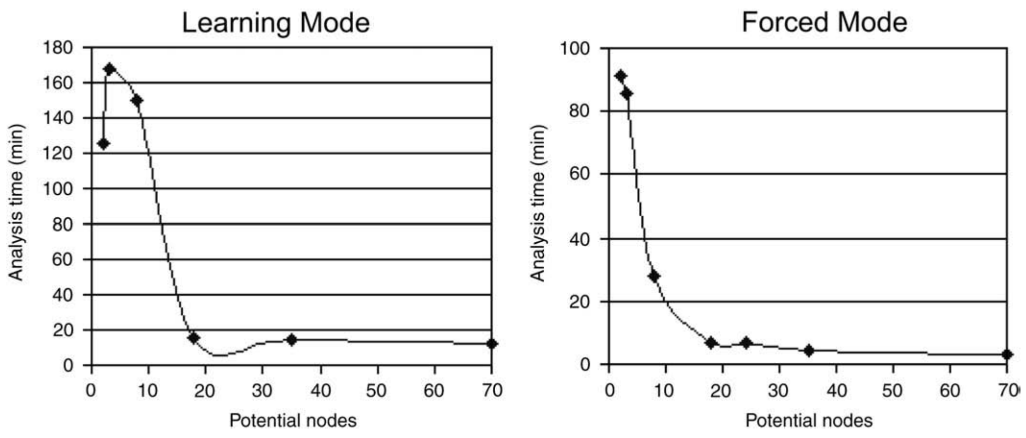

Fig. 5. Running time for a 70-job process, as a function of the number of available potential nodes and workflow configuration.

distributed analysis of large data sets and for enabling data mining via highly specific database queries using the Structured Query Language (SQL) (Astrahan and Chamberlin, 1975; Codd, 1970). CNARI now incorporates a "mediator" mechanism, based on the popular information integration paradigm (Wiederhold, 1992), by which compute nodes on TeraGrid sites query large data sets stored in CNARI relational databases using a Python database interface (DBI). The CNARI mediator enables Swift workflows to process data sets stored in relational databases, in addition to file-resident data sets.

The CNARI mediator splits a complex voxel- or vertex-wise analysis across multiple nodes by automatically assigning each compute node a range of voxels for which it will be responsible. On the Swift side, which is run on the client, the mediator is configured to take an SQL query, a processing script and the parameters and environment variables required by the script and pass them onto a compute node on a Grid site. On the Grid compute node, the mediator then runs the query and passes the results to statistical software (e.g., R, Matlab) for analysis. Typically, the mediator is given a voxel range combined with a query specification and it iterates through the voxel range generating a batch of queries. This produces a clustering effect on the remote Grid site, which helps to speed up processing by minimizing queue time for individual jobs. The main advantage of utilizing database queries via this mechanism is that it enables analysis of large data sets on Grid sites while avoiding the need to transfer the complete data set (potentially quite large) to each compute node. Instead each node extracts only a subset of the data for processing thereby significantly reducing network traffic. Full specification of the mediator as well as an example of a Swift script that calls it can be found on the CNARI collaboration website (http://www.ci.uchicago.edu/wiki/bin/view/CNARI/SampleScripts).

Prior to developing this interface, we extensively used databases for storing and analyzing fMRI data on a local cluster computer at The University of Chicago. The mediator enables us to now perform a large proportion of our database-driven analyses under CNARI, using Swift workflows and Grid resources. Several of these analysis methods are particularly computationally intensive, including analyses of functional connectivity and analyses employing complex SQL queries.

We have also developed workflows within CNARI for extracting specific subsets of time series data sets with highly specific selection criteria. By storing neuroimaging data in relational schemas, it is possible to code the data so that it can be mined for particular patterns of activity. An SQL query can then be used to extract only those parts of the time series that were acquired, for example, when a particular stimulus was presented to the participant, or to extract only those points in the time series associated with a particular feature (after the time series has been coded for some such auditory or visual features).

4. CNARI and related work

A number of ongoing development efforts share partially common goals with those of the CNARI infrastructure. The Extensible Neuroimaging Archive Toolkit (XNAT; Marcus et al., 2007) is aimed at offering researchers an integrated environment for archival, search and sharing of neuroimaging data sets. It is aimed at managing large amounts of data via a three-tier design infrastructure consisting of a client front end, the XNAT middleware and a data store consisting of both a relational database and a file system on which images are stored. In XNAT, relational databases are used to store pointers to data files, and researchers can pull data, analyze it on a local computer, and store summary results in the database if needed. CNARI development complements this effort in making available an environment that is directly geared towards distributed analysis of fMRI data. Thus, functional data (and if needed, anatomical data) are stored directly in the relational database and are extracted from remote and local client nodes via SQL queries. Storing time series data in the database offers the ability to ask highly precise questions of the data, which would be very difficult to analyze without sophisticated SQL queries (Hasson et al., 2008). While XNAT does not use a dedicated workflow and provenance engine, CNARI employs SWIFT as a workflow engine, which makes it possible to reconstruct intermediate data sets by re-executing the workflow that generated them.

GridPACS (Hastings et al., 2005; Kumar et al., 2008) and Globus MEDICUS (Erberich et al., 2007) are mature infrastructures geared towards analysis and archival of medical images using grid infrastructures. They share with CNARI the concern that large data sources should be pushed to nearby Grid sites so as to minimize what can amount to terabytes of data transfer in fields such as confocal microscopy where single slides can amount to multiple gigabytes of data. GridPACS offers unique features for biomedical researchers, e.g., the ability to extract data based on spatially bound boxes. GridPACS offers researchers a common schema that also supports multiple, researcher-specific vocabulary. In contrast, CNARI development is based on the notion that different researchers will gravitate towards different models in storing functional data of different types and therefore does not utilize a common schema. GridPACS offers a "declustering" component whose function is to distribute the analysis of a data set between as many computing nodes as possible. Similar functions are achieved in CNARI via its mediator mechanism, which is intended to offer researchers the ability to break up analyses on voxel ranges, ROIs, or any other grouping element that is coded in the database and that can be queried on. GridPACS also offers a workflow description mechanism with similar functionalities offered by Swift. The main difference between the two systems is that CNARI is oriented towards implementing advanced analyses of fMRI data (3D +Time dimension) and so is geared towards allowing users create complex relational databases and interface analyses scripts against these databases using the general purpose Swift scripting language with its far more powerful functional and data abstraction capabilities, and its broader range of target execution environments.

The LONI system (Dinov et al., 2006; Rex et al., 2003; Toga, 2002) enables both atlas creation and analyses of neuroimaging data using a graphical environment that allows the user to describe the required workflow (see also Fissell, 2007 for a review of workflow specification environments). The LONI pipeline environment allows specification and execution of complex workflows, and it allows for execution of jobs in a Sun Grid Engine processing environment. Our particular efforts are complementary to these developments as they focus on the facilitation of neuroimaging analyses using grid and parallel computing, and particularly on using distributed resources to process data residing in relational databases. It is reasonable to expect that in the future, the Swift workflow system could benefit from the existence of graphical editors such as those developed by LONI, and initial experiments indicate that the function descriptions generated by the LONI graphical editor can be readily mapped into Swift scripts.

5. Future considerations

The functional brain imaging laboratory of the future can look fundamentally different computationally from that of the present. We have argued that several fundamental infrastructure changes can have dramatic effects, first by improving the organization and efficiency of data representation and processing, and second by permitting a rethinking of many assumptions of experimental design and analysis, leading to valuable advances in how imaging experiments are conceived and executed. We present the CNARI architecture, currently in place at the Human Neuroscience Laboratory at The University of Chicago, which combines database management systems for storing and manipulating data (Hasson et al., 2008), and formal workflow specifications (using the Swift language (Zhao et al., 2007)) that facilitate high-performance computing and provenance tracking (Stef-Praun et al., 2007).

The use of a database management system to store and manipulate data has a number of significant advantages that will simplify several aspects of functional imaging research for the future. A relational database codes information in such a way that it is routine to access highly selective subsets of the data, limited only by the ability of the user to specify the desired subset in a database query (a formal but easily articulated specification of the desired data). Furthermore, particular users can be given permission to access only portions, or “views”, of an entire data set, such that they can build queries to probe subsets within their portion, but not outside. Collaboration is greatly simplified, as users from any part of the world can be given secure access to just those portions of a data set that are part of the particular joint research activity. Data sharing, an important goal of the National Institutes of Health in the USA, and of other sponsoring agencies worldwide, becomes greatly simplified.

We have tested these notions using a data set containing four longitudinal fMRI scans of twelve people performing finger and wrist movements (Small et al., 2002), from which we provided collaborative access to the first and fourth scans (and not the second and third), to the images in which the subjects performed finger movements (and not wrist movements), and from only those subjects who had the best hand function (six scans representing a split half of the twelve subjects). We additionally tested our methods by querying this data set to focus on just those voxels in the ventral premotor cortex and inferior parietal lobule, and correlated the mean time series from these regions. We believe that the future of brain imaging will take tremendous advantage of database management systems to uncover relationships in experimental data that are otherwise difficult or impossible to probe and test.

The use of Grid-enabled workflows and high-performance computing allows us to approach functional brain imaging in some important new ways. First and foremost is that advanced computing enables a rethinking of experimental design and analysis approaches such that large combinatorially intensive network searches, high volume iterative procedures, and large geometric operations are no longer off limits. These three examples relate directly to the types of functional imaging approaches that we envision for the future. We use combinatorial search to find ideal structural equation models of regional fMRI data by using a distributed processing algorithm (formalized in a Swift workflow) that searches an entire space of possible models containing millions of regional interactions. By parallelizing iterative procedures, it is possible to use randomization methods (Manly, 2007) (e.g., permutation analyses (Nichols and Holmes, 2002)) to determine significance levels for fMRI statistical inference, thereby minimizing a number of confounds (e.g., voxel independence) that are present in more traditional parametric (Fisherian) statistical approaches. By performing geometric operations of different parts of a brain in parallel, many new visualization possibilities arise for the first time. In addition to these benefits, high-performance computing leads to new possibilities for integrating multiple types of time series data (e.g., fMRI and EEG) in complex analyses (e.g., correlations of large time series). Finally, the use of workflows makes it possible to use high-performance computing without actually performing by hand the distribution of the work onto multiple processors or integrating the results. The integration of databases and Grid-enabled workflows together permit the facile performance of such tasks as meta-analyses of large collections of already processed data sets.

6. Conclusions

The future functional imaging laboratory will benefit from advances in computer science that will put high-performance relational databases, advanced scripting languages, and multiprocessor computer clusters and grids in the hands of psychological and neurobiological scientists. We are building the CNARI architecture to implement these methods in the present, incorporating open source software, including the MySQL database management system and Swift workflows, and open collaborative communities for grid computing, including Open Science Grid and TeraGrid. Our computational infrastructure is already permitting new types of collaboration and novel experimental approaches that could not be considered otherwise. We are hopeful that this will lead to new advances in neuroscience and psychology that are only possible because of an expansion in the types of questions we are able and willing to ask.

Acknowledgments

This work was supported by the National Institute of Deafness and Other Communication Disorders (NIDCD) of the National Institutes of Health (NIH) under Grants R21/R33 DC008638 and R01 DC07488, by the James S. McDonnell Foundation award to the Brain Network Recovery Group (BrainNRG), and by the Computation Institute of The University of Chicago. The Swift System used in CNARI is also supported by the National Science Foundation under Grant OCI-0721939. The support of these sponsors is gratefully acknowledged. We also appreciate the help of Dr. E Elinor Chen with the Peak Analysis, Drs. Jeremy Skipper and Ana Solodkin with the structural equation modeling, and Ben Clifford, Mihael Hategan, and Dr. Tiberiu Stef-Praun in the use and support of Swift. We also appreciate the comments of Dr. Ian Foster on the penultimate draft of this manuscript.

Appendix A. Swift example of fMRI workflow

The following is an AFNI workflow in Swift for calculating the signal-to-noise ratio (SNR) of a data set. After one determines via simulation the minimum SNR a voxel needs to hold such that it could reasonably demonstrate activity, one calculates the SNR of the data set using this script to identify those voxels.

```
type file{}

type AFNI_obj{
file HEAD;
file BRIK;
};

(AFNI_obj meanResult) AFNI_mean(AFNI_obj meanInput,
string baseName){
   app {
      AFNI_3dTstat @strcat("-mean -prefix ./mean.",
      baseName) @meanInput.BRIK;
      }
}

(AFNI_obj stdevResult) AFNI_stdev(AFNI_obj stdevInput,
string baseName){
   app {
      AFNI_3dTstat @strcat("-stdev -prefix ./stdev.",
      baseName) @stdevInput.BRIK;
      }
}

(AFNI_obj detrendResult) AFNI_detrend(AFNI_obj
detrendInput, string baseName){
   app {
      AFNI_3dDetrend "-polort 3 -prefix" @strcat
("detrend.",baseName) @detrendInput.BRIK;
   }
}

(AFNI_obj ratioResult) AFNI_doratio(AFNI_obj stdev
Result, AFNI_obj meanResult , string baseName){
   app {
      AFNI_3dcalc "-verbose" "-a" @meanResult.BRIK "-b"
      @stdevResult.BRIK
      "-expr" "(a/b)" @strcat("-prefix snr.",baseName);
   }
}

(AFNI_obj detrendResult, AFNI_obj stdevResult,
AFNI_obj meanResult, AFNI_obj ratioResult) AFNI_snr
(string baseName, AFNI_obj inputTS)
{
   (meanResult) = AFNI_mean(inputTS, baseName);
   (detrendResult) = AFNI_detrend(inputTS, baseName);
   (stdevResult) = AFNI_stdev(detrendResult,
   baseName);
   (ratioResult) = AFNI_doratio(stdevResult, mean
   Result, baseName);
}

doAfni()
{
string declarelist = ["03","04","05"];
foreach subject in declarelist {
  int runs[] = [1:1];
    foreach run in runs {
      AFNI_obj srun[]<ext; exec="afnimapper">;
      string baseName = @strcat("S",subject,".run",run);
      AFNI_obj meanResult<simple_mapper;prefix=
      @strcat("mean.",baseName,"+orig.")>;
      AFNI_obj stdevResult<simple_mapper;prefix=
      @strcat("stdev.",baseName,"+orig.")>;
      AFNI_obj detrendResult<simple_mapper;prefix=
      @strcat("detrend.",baseName,"+orig.")>;
      AFNI_obj ratioResult<simple_mapper;prefix=
      @strcat("snr.",baseName,"+orig.")>;
      (detrendResult, stdevResult, meanResult,
ratioResult) = AFNI_snr (baseName, srun[run]);

      }

   }
}

doAfni();
```

References

Arbuckle, J.L., 1989. AMOS: analysis of moment structures. The American Statistician 43, 66–67.

Astrahan, M.M., Chamberlin, D.D., 1975. Implementation of a structured English query language. Communications of the ACM 18 (10), 580–588.

Ban, T., Naito, J., Kawamura, K., 1984. Commissural afferents to the cortex surrounding the posterior part of the superior temporal sulcus in the monkey. Neuroscience Letters 49 (1–2), 57–61.

Ban, T., Shiwa, T., Kawamura, K., 1991. Cortico-cortical projections from the prefrontal cortex to the superior temporal sulcal area (STs) in the monkey studied by means of HRP method. Archives Italiennes de Biologie 129 (4), 259–272.

Barbas, H., 2000. Connections underlying the synthesis of cognition, memory, and emotion in primate prefrontal cortices. Brain Research Bulletin 52 (5), 319–330.

Beckman, P.H., 2005. Building the TeraGrid. Philosophical Transactions of the Royal Society of London, Series A: Mathematical and Physical Sciences 363 (1833), 1715–1728.

Billo, E.J., 2007. Excel for Scientists and Engineers: Numerical Methods. Wiley-Interscience, Hoboken, N.J.

Biswal, B., Yetkin, F.Z., Haughton, V.M., Hyde, J.S., 1995. Functional connectivity in the motor cortex of resting human brain using echo-planar MRI. Magnetic Resonance in Medicine 34 (4), 537–541.

Broca, P.P., 1861. Nouvelle Observation d'Aphémie produite par une Lesion de la Partie Postérieure des Deuxième et Troisième Circonvolutions Frontales. Bulletins de la Société anatomique de Paris 6, 398–407.

Caton, R., 1875. The electric currents of the brain. British Medical Journal 2, 278.

Codd, E.F., 1970. A relational model of data for large shared data banks. Communications of the ACM 13 (6), 377–387.

Cox, R.W., 1996. AFNI: software for analysis and visualization of functional magnetic resonance neuroimages. Computers in Biomedical Research 29 (3), 162–173.

Dale, A.M., Fischl, B., Sereno, M.I., 1999. Cortical surface-based analysis. I. Segmentation and surface reconstruction. NeuroImage 9 (2), 179–194.

Dinov, I.D., Valentino, D., Shin, B.C., Konstantinidis, F., Hu, G., MacKenzie-Graham, A., Lee, E.F., Shattuck, D., Ma, J., Schwartz, C., 2006. LONI visualization environment. Journal of Digital Imaging 19 (2), 148–158.

Donchin, E., Wicke, J.D., Lindsley, D.B., 1963. Cortical evoked potentials and perception of paired flashes. Science 141, 1285–1286.

Erberich, S.G., Silverstein, J.C., Chervenak, A., Schuler, R., Nelson, M.D., Kesselman, C., 2007. Globus MEDICUS — federation of DICOM medical imaging devices into healthcare Grids. Studies in Health Technology and Informatics 126, 269–278.

Felleman, D.J., Van Essen, D.C., 1991. Distributed hierarchical processing in the primate cerebral cortex. Cerebral Cortex 1 (1), 1–a-47.

Fischl, B., Sereno, M.I., Dale, A.M., 1999. Cortical surface-based analysis. II: inflation, flattening, and a surface-based coordinate system. NeuroImage 9 (2), 195–207.

Fissell, K., 2007. Workflow-based approaches to neuroimaging analysis. Methods in Molecular Biology 401, 235–266.

Foster, I., 2002. The grid: a new infrastructure for 21st century science. Physics Today 55 (2), 42–47.

Foster, I., 2003. The grid: computing without bounds. Scientific American 288 (4), 78–85.

Foster, I., 2005. Globus toolkit version 4: software for service-oriented systems. Paper Presented at the IFIP International Conference on Network and Parallel Computing.

Foster, I., Kesselman, C., 1999. The Globus project: a status report. Future Generation Computer Systems 15 (5–6), 607–621.

Foster, I.A.K.C., Tuecke, S., 2001. The anatomy of the grid: enabling scalable virtual organisations. International Journal of Supercomputer Applications 15 (3).

Friston, K.J., Frith, C.D., Liddle, P.F., Dolan, R.J., Lammertsma, A.A., Frackowiak, R.S., 1990. The relationship between global and local changes in PET scans. Journal of Cerebral Blood Flow and Metabolism 10 (4), 458–466.

Friston, K.J., Frith, C.D., Liddle, P.F., Frackowiak, R.S., 1991. Comparing functional (PET) images: the assessment of significant change. Journal of Cerebral Blood Flow and Metabolism 11 (4), 690–699.

Friston, K.J., Frith, C.D., Liddle, P.F., Frackowiak, R.S., 1993. Functional connectivity: the principal-component analysis of large (PET) data sets. Journal of Cerebral Blood Flow and Metabolism 13 (1), 5–14.

Friston, K., Phillips, J., Chawla, D., Buchel, C., 2000. Nonlinear PCA: characterizing interactions between modes of brain activity. Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences 355 (1393), 135–146.

Gentleman, R., Ihaka, R., 1997. The R language, Fairfax Station, VA, USA.

Geschwind, N., 1965. Disconnection syndromes in animals and man. Brain 88 (237–294), 585–644.

Gilat, A., 2007. MATLAB: an Introduction with Applications. John Wiley & Sons, Inc., New York, NY.

Goebel, R., 1997. Brain voyager 2.0: from 2D to 3D fMRI analysis and visualization. NeuroImage 5 (4 PART II).

Gosling, J., Joy, B., Steele, G., Bracha, G., 2000. Java language specification, The Java Series, Second Edition. Addison-Wesley Longman Publishing Co., Inc., Boston, MA.

Gray, J., Liu, D.T., Nieto-Santisteban, M., Szalay, A., DeWitt, D.J., Heber, G., 2005. Scientific data management in the coming decade. SIGMOD Record 34 (4), 34–41.

Hackett, T.A., Stepniewska, I., Kaas, J.H., 1999. Callosal connections of the parabelt auditory cortex in macaque monkeys. European Journal of Neuroscience 11 (3), 856–866.

Hasson, U., Skipper, J.I., Wilde, M.J., Nusbaum, H.C., Small, S.L., 2008. Improving the analysis, storage and sharing of neuroimaging data using relational databases and distributed computing. NeuroImage 39 (2), 693–706.

Hastings, S., Oster, S., Langella, S., Kurc, T.M., Pan, T., Catalyurek, U.V., Saltz, J.H., 2005. A grid-based image archival and analysis system. Journal of the American Medical Informatics Association 12 (3), 286–295.

Hoeting, J.A., Madigan, D., Raftery, A.E., Volinsky, C.T., 1999. Bayesian model averaging: a tutorial. Statistical Science 14 (4), 382–417.

Horwitz, B., Tagamets, M.A., McIntosh, A.R., 1999. Neural modeling, functional brain imaging, and cognition. Trends in Cognitive Science 3 (3), 91–98.

Hudak, P., 1989. Conception, evolution, and application of functional programming languages. Computing Surveys 21 (3), 359–411.

Kass, R.E., Ventura, V., Brown, E.N., 2005. Statistical issues in the analysis of neuronal data. Journal of Neurophysiology 94 (1), 8–25.

Kirk, R.E., 1995. Experimental Design: Procedures for the Behavioral Sciences. Brooks/Cole Publishing Company, Pacific Grove, California.

Kumar, V.S., Rutt, B., Kurc, T., Catalyurek, U.V., Pan, T.C., Chow, S., Lamont, S., Martone, M., Saltz, J.H., 2008. Large-scale biomedical image analysis in grid environments. Computers in Biomedical Research 12 (2), 154–161.

Litzkow, M.J., Livny, M., Mutka, M.W., 1988. Condor—a hunter of idle workstations. Paper Presented at the 8th International Conference on Distributed Computing Systems Washington, DC, USA.

Manly, B.F.J., 2007. Randomization, bootstrap and Monte Carlo methods, Biology, Third ed. Chapman & Hall/CRC, Boca Raton, Florida.

Marcus, D.S., Olsen, T.R., Ramaratnam, M., Buckner, R.L., 2007. The extensible neuroimaging archive toolkit. Neuroinformatics 5 (1), 11–33.

Marrelec, G., Kim, J., Doyon, J., Horwitz, B., 2008. Large-scale neural model validation of partial correlation analysis for effective connectivity investigation in functional MRI. Human Brain Mapping published online March 14.

McIntosh, A.R., Grady, C.L., Ungerleider, L.G., Haxby, J.V., Rapoport, S.I., Horwitz, B., 1994. Network analysis of cortical visual pathways mapped with PET. Journal of Neuroscience 14 (2), 655–666.

Nichols, T.E., Holmes, A.P., 2002. Nonparametric permutation tests for functional neuroimaging: a primer with examples. Human Brain Mapping 15 (1), 1–25.

Ojemann, G., Ojemann, J., Lettich, E., Berger, M., 1989. Cortical language localization in left, dominant hemisphere: an electrical stimulation mapping investigation in 117 patients. Journal of Neurosurgery 71, 316–326.

Ousterhout, J.K., 1998. Scripting: higher level programming for the 21st Century. Computer 31 (3), 23–30.

Overview of the IBM Blue Gene/P project, 2008. IBM Journal of Research and Development 52 (1–2), 199–219.

Parrish, T.B., Gitelman, D.R., LaBar, K.S., Mesulam, M.M., 2000. Impact of signal-to-noise on functional MRI. Magnetic Resonance in Medicine 44 (6), 925–932.

Penfield, W., Boldrey, E., 1937. Somatic motor and sensory representation in the cerebral cortex of man as studied by electrical stimulation. Brain 60, 389–443.

Petrides, M., Pandya, D.N., 1984. Projections to the frontal cortex from the posterior parietal region in the rhesus monkey. Journal of Comparative Neurology 228 (1), 105–116.

Petrides, M., Pandya, D.N., 1988. Association fiber pathways to the frontal cortex from the superior temporal region in the rhesus monkey. Journal of Comparative Neurology 273 (1), 52–66.

Petrides, M., Pandya, D.N., 1999. Dorsolateral prefrontal cortex: comparative cytoarchitectonic analysis in the human and the macaque brain and corticocortical connection patterns. European Journal of Neuroscience 11 (3), 1011–1036.

Pordes, R., Petravick, D., Kramer, B., Olson, D., Livny, M., Roy, A., Avery, P., Blackburn, K., Wenaus, T., Wuerthwein, F., Foster, I., Gardner, R., Wilde, M., Blatecky, A., McGee, J., Quick, R., 2007. The open science grid. Journal of Physics 012057 Conference Series.

Raicu, I., Zhao, Y., Foster, I.T., Szalay, A., 2008. Accelerating large-scale data exploration through data diffusion. Paper Presented at the Proceedings of the 2008 International Workshop on Data-aware Distributed Computing.

Rex, D.E., Ma, J.Q., Toga, A.W., 2003. The LONI pipeline processing environment. NeuroImage 19 (3), 1033–1048.

Ritchie, D.M., 1993. The development of the C language. Paper Presented at the Second ACM SIGPLAN Conference on History of Programming Languages.

Rizzolatti, G., Fogassi, L., Gallese, V., 1997. Parietal cortex: from sight to action. Current Opinion in Neurobiology 7 (4), 562–567.

Rizzolatti, G., Luppino, G., Matelli, M., 1998. The organization of the cortical motor system: new concepts. Electroencephalography and Clinical Neurophysiology 106 (4), 283–296.

Robb, R.A., Hanson, D.P., 1990. ANALYZE: a software system for biomedical image analysis. Paper Presented at the First Conference on Visualization in Biomedical Computing, Los Alamitos, CA, USA.

Rosa, M.G., Soares, J.G., Fiorani Jr., M., Gattass, R., 1993. Cortical afferents of visual area MT in the *Cebus* monkey: possible homologies between New and Old World monkeys. Visual Neuroscience 10 (5), 827–855.

Saad, Z.S., Reynolds, R.C., Cox, R.W., Argall, B., Japee, S., 2004. SUMA: an interface for surface-based intra- and inter-subject analysis with AFNI. Paper Presented at the IEEE International Symposium on Biomedical Imaging: Macro to Nano.

Seltzer, B., Pandya, D.N., 1994. Parietal, temporal, and occipital projections to cortex of the superior temporal sulcus in the rhesus monkey: a retrograde tracer study. Journal of Comparative Neurology 343 (3), 445–463.

Skipper, J.I., Goldin-Meadow, S., Nusbaum, H.C., Small, S.L., 2007. Speech-associated gestures, Broca's area, and the human mirror system. Brain and Language 101 (3), 260–277.

Small, S.L., Hlustik, P., Noll, D.C., Genovese, C., Solodkin, A., 2002. Cerebellar hemispheric activation ipsilateral to the paretic hand correlates with functional recovery after stroke. Brain 125 (Pt 7), 1544–1557.

Smith, S.M., Jenkinson, M., Woolrich, M.W., Beckmann, C.F., Behrens, T.E., Johansen-Berg, H., Bannister, P.R., De Luca, M., Drobnjak, I., Flitney, D.E., Niazy, R.K., Saunders, J., Vickers, J., Zhang, Y., De Stefano, N., Brady, J.M., Matthews, P.M., 2004. Advances in functional and structural MR image analysis and implementation as FSL. NeuroImage 23 (Suppl 1), S208–S219.

Stef-Praun, T., 2005. An extended service oriented architecture for Web services adoption through economic incentives. Paper Presented at the First International Workshop on Advanced Architectures and Algorithms for Internet Delivery and Applications, Los Alamitos, CA, USA.

Stef-Praun, T., Clifford, B., Foster, I., Hasson, U., Hategan, M., Small, S.L., Wilde, M., Zhao, Y., 2007. Accelerating medical research using the Swift workflow system. Studies in Health Technology and Informatics 126, 207–216.

Stroustrup, B., 2000. The C++ Programming Language. Addison-Wesley Longman Publishing Co., Inc., Boston, MA.

Szalay, A., Gray, J., 2006. Science in an exponential world. Nature 440 (7083), 413–414.

Tannenbaum, T., Litzkow, M., 1995. The Condor distributed processing system. Dr. Dobb's Journal 20 (2), 40–42.

Toga, A.W., 2002. Imaging databases and neuroscience. The Neuroscientist 8 (5), 423–436.

van Horn, J., Dobson, J., Woodward, J., Wilde, M., Zhao, Y., Voeckler, J., Foster, I., 2005. Grid-based computing and the future of neuroscience computation. Methods in Mind. MIT Press.

van Rossum, G., de Boer, J., 1991. Linking a Stub Generator (AIL) to a Prototyping Language (Python), Buntingford, UK.

Wall, L., 2000. Programming Perl. O'Reilly & Associates, Inc., Sebastopol, CA.

Wiederhold, G., 1992. Mediators in the architecture of future information systems. Computer 25 (3), 38–49.

Zhao, Y., Hategan, M., Clifford, B., Foster, I., von Laszewski, G., Nefedova, V., Raicu, I., Stef-Praun, T., Wilde, M., 2007. Swift: fast, reliable, loosely coupled parallel computation. 2007 IEEE Congress on Services, pp. 199–206.