NeuroImage

www.elsevier.com/locate/ynimg
NeuroImage 39 (2008) 693 – 706

Improving the analysis, storage and sharing of neuroimaging data using relational databases and distributed computing

Uri Hasson,[a,b,*] Jeremy I. Skipper,[a,b] Michael J. Wilde,[c,d] Howard C. Nusbaum,[b,e,f] and Steven L. Small[a,b,f]

[a]*Department of Neurology, The University of Chicago, Chicago, IL, USA*
[b]*Department of Psychology, The University of Chicago, Chicago, IL, USA*
[c]*The Computation Institute, The University of Chicago, Chicago, IL, USA*
[d]*Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, USA*
[e]*Centre for Cognitive and Social Neuroscience, The University of Chicago, Chicago, IL, USA*
[f]*The Brain Research Imaging Centre, The University of Chicago, Chicago, IL, USA*

Received 17 November 2006; revised 30 July 2007; accepted 6 September 2007
Available online 21 September 2007

The increasingly complex research questions addressed by neuroimaging research impose substantial demands on computational infrastructures. These infrastructures need to support management of massive amounts of data in a way that affords rapid and precise data analysis, to allow collaborative research, and to achieve these aims securely and with minimum management overhead. Here we present an approach that overcomes many current limitations in data analysis and data sharing. This approach is based on open source database management systems that support complex data queries as an integral part of data analysis, flexible data sharing, and parallel and distributed data processing using cluster computing and Grid computing resources. We assess the strengths of these approaches as compared to current frameworks based on storage of binary or text files. We then describe in detail the implementation of such a system and provide a concrete description of how it was used to enable a complex analysis of fMRI time series data.

Introduction

The development of non-invasive neuroimaging methods, such as positron emission tomography (PET), and functional magnetic resonance imaging (fMRI), has produced an explosion of new findings in human neuroscience. Scientific advancement in this domain has been the direct result of developments both in hardware technology for data acquisition and algorithms for data processing and image analysis. As these analytical approaches have improved in sensitivity and power, they have made it possible to address increasingly complex scientific questions. Yet, while the scientific questions and analysis methods have become more sophisticated, the computational infrastructures to support this work have generally not kept pace. In this article, we discuss a novel computational approach to support analysis of functional imaging data. The importance of this approach is that it allows neuroscientists to address more complex questions while concomitantly speeding up the rate at which these questions can be evaluated.

Early neuroimaging research was based on grouping trials of the same sort into a single presentation sequence in so-called “block designs”. While these designs enabled researchers to address certain a priori questions, they left little room for a posteriori data analysis. More recently, “event-related designs” (both slow and fast variants) have not only enabled researchers to evaluate a priori research questions but, importantly, also enabled a variety of interesting a posteriori analyses that have been of tremendous value. For example, some researchers have partitioned the stimuli according to post hoc classifications after data have been collected, as in a study by Wagner et al. (1998), which analyzed stimuli as a function of whether they were subsequently remembered or forgotten. The use of event-related designs has also opened the way to new statistical analysis methods for estimation of event-linked hemodynamic responses, and for assessing the correlation between neural activity and finer features of stimuli properties.

In light of these advancements, it is noticeable that there has been substantially less progress in the development of computational infrastructures supporting the storage, analysis, and sharing of fMRI data. Although there are significant efforts underway to

* Corresponding author. Human Neuroscience Laboratory, Biological Sciences Division, University of Chicago Hospital Q300, 5841 S. Maryland Avenue MC-2030, Chicago, IL 60637, USA. Fax: +1 773 834 7610.
E-mail address: uhasson@uchicago.edu (U. Hasson).

Available online on ScienceDirect (www.sciencedirect.com).

1053-8119/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.neuroimage.2007.09.021

represent and store imaging data for large multi-center studies (Van Horn et al., 2001), the infrastructures at individual research centers are often not optimally designed to support everyday imaging research tasks. Most importantly, the performance of increasingly complex analyses, such as evaluation of functional connectivity between brain regions, requires certain computational tasks that can be cumbersome and even prohibitively difficult using traditional data representation approaches (i.e., hierarchical file systems and matrix representation of images). Such complex analyses require, for example, repeated averaging of subsections of time series (TS) data and correlating TS data, but currently employed frameworks for data storage are ill equipped for this task. Furthermore, as the complexity of analyses increases, current approaches to data representation generate prohibitively large amounts of intermediate data (e.g., "mask" files) in addition to the final results. This in itself causes serious management overhead. The immediate result of these weaknesses is that the computational infrastructure becomes a bottleneck in the progress of research: it results in slower data analysis, reduces the number of questions that can be asked of the data, and makes it difficult to enable concurrent access to the data (for local and remote users) as is often needed for complex analyses and collaborative research. Thus, the current computational demands for imaging research call for a different approach to storage and analysis of fMRI data. The basic requirements of such systems are that they store data efficiently, enable rapid selection of data, and make data easily accessible for both local and remote users.

In what follows, we present a unified framework for the analysis, storage and sharing of neuroimaging data that addresses these needs, using an approach based on the general data representation and manipulation abilities of database management systems (DBMSs). While this framework is technical in nature, its forte is in extending the researcher's ability to ask more questions about neuroimaging data and obtain rapid responses to these questions while employing advanced statistical tools. These advantages increase the efficiency of a scientific inquiry process that is often based on being able to ask increasingly refined questions about data.

A major advantage of the database-centric framework we present here is that it not only uses DBMSs for storing and sharing of data, but also takes advantage of DMBS capabilities by making the database an integral part of the fMRI data analysis workflow. We review the advantages that this approach offers over the traditional methods of storing and analyzing data using flat files (i.e., binary or text files), and show how these directly bear on the scientific routine and daily research in brain imaging. We demonstrate the scalability of these methods when coupled with modern distributed cluster computing (Pfister, 1998) and Grid computing technologies (Foster, 2005), in which numerous computers (computing nodes) perform tasks in parallel, and discuss issues such as efficient data storage, data sharing, data transparency, and advanced data analysis. Finally, we detail our implementation of such a system.

Our aim is to introduce such systems to researchers who have not considered this approach so that they can become acquainted with both the strengths and limitations of database-oriented analysis of brain images. We therefore first describe our general approach rather than the specific details of our implementation (in section, Relational databases and their application to imaging). We then present the description of the system's actual implementation (in section, System description). The system is based on open source software tools (widely available and supported by large developer communities) and a client–server approach; the data are stored using a database server, and analyzed by remote client computers, which request data over the network and analyze the data using a powerful statistical programming language (R Development Core Team, 2005; http://www.R-project.org). We then provide concrete details of one example analysis to communicate more practical information (in section, Detailed example: reverse correlation analysis). Specifically, we explain how this system was employed to conduct an analysis that exemplifies beneficial aspects of using DBMS in conjunction with distributed computing to conduct fMRI data analysis. This analysis is a "reverse correlation" of fluctuations in hemodynamic responses with specific stimulus properties of naturalistic stimuli. We trust that these descriptions on both abstract and concrete levels will allow researchers to consider more diverse and creative analysis methods and efficient ways for sharing and storing data.

Relational databases and their application to imaging

As scientists wrestle with the exponential growth of their datasets, the power and utility of the relational database is being applied with increasing breadth and frequency across a range of scientific disciplines (Szalay and Gray, 2006). The benefits in terms of indexability, leveraging of metadata, and scalability of database approaches over file-based approaches are becoming clear in a growing number of disciplines (Gray et al., 2005). This trend can be seen clearly in digital astronomy, where the Sloan Digital Sky Survey (http://www.sdss.org/) is making an increasing use of DBMS technology to describe millions of celestial objects, and to enable searches across that data (Nieto-Santisteban et al., 2005). In this effort, improved data organization and relational representation enables database queries, performed in a distributed manner on Grid resources, to run an order of magnitude faster than a file-based implementation of the same algorithm operating over file-based catalogs.

In bioinformatics, the warehousing of file-based data from both curated public data sources and laboratory experiments into integrated relational databases affords new methods for search and analysis. Here, the Genomics Unified Schema (http://www.gusdb.org; cf., Davidson et al., 2001) provides a fabric for creating integrated relational databases for functional genomics data analysis from public data sources and from laboratory experiments in sequence analysis and proteomics (Stoeckert, 2005).

Researchers using imaging data are already facing similar challenges. fMRI analyses typically use and generate a vast number of data files. For example, individual participant data might include structural images optimized for different tissue parameters (e.g., T1, T2, FLAIR), diffusion-weighted images (isotropic and anisotropic), perfusion images, angiograms, surface representations of volumes, regions of interest, numerous TS (e.g., unregistered, registered, detrended, despiked, error terms), various masks, as well as numerous statistical maps. Group-level statistical maps might reflect the results of various types of statistical analyses performed on the individual level data (e.g., analysis of variance (ANOVA), principal components analysis (PCA), *t*-tests, etc.). Together, the number of flat files generated (i.e., linear unstructured data stored in files and organized in directories) can become quite large and the entire set is typically complex, difficult to manage, and enormous in size. This is particularly so when data are kept in the form of text files for purposes of certain advanced analyses. DBMS offers many advantages over flat files in terms of storage, sharing and analysis, and we discuss some of these in what follows. Certainly flat file systems allow more rapid sequential access to data, which under the right circumstances, can result in faster processing. Yet, this advantage is less important when the data in the database are analyzed in parallel utilizing high-performance distributed computing systems.

In DBMSs, 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 (how or where this information is stored is irrelevant to users, and so we will not address this further). All a user needs to know in order to access the data is the name of the table storing the data and what data attributes it holds. For example, a user can request to see all the information in the *subject04* table by issuing a command (equivalent to): *show all information in table subject04*. Or, if more specific information is needed: *show all information in table subject04 where the condition is 'tone-presented'*. DBMSs are therefore indispensable for querying (i.e., asking subset and relational questions of) large amounts of data, and in the System description section we demonstrate how such capabilities can be utilized for rapid development and execution of sophisticated fMRI analyses. A number of research projects have utilized databases for archiving and making available large numbers of imaging datasets (Kotter, 2001; Van Horn et al., 2001), or the results of statistical analyses (Fox and Lancaster, 2002). Such large-scale projects, however, use DBMS to manage large amounts of file data, rather than to maintain data in a form that facilitates use in outside analysis routines. They are not aimed at affecting the daily practices of researchers working on fMRI projects in those stages of the work where data are still being analyzed (or in some cases, mined) for certain patterns. Rather, they are intended for archiving, reanalysis and meta-analysis.

For the individual researcher or a research laboratory, storing data in a database implies that given proper permissions, the data could be accessed from any remote computer (whether on the local network, or over the Internet) obviating the need to save multiple copies of data at different locations. As a result, sharing data with remote collaborators is greatly simplified, because servers can accept requests for data (queries) over computer networks. For example, two research groups can analyze the same dataset using different methods of analysis (e.g., ICA vs. contrast analysis). DBMSs also allow for data filtering on the server side, thus eliminating unnecessary network traffic. In practice, an analysis script written at one location can be sent to remote collaborators and executed from their computers without any modification whatsoever, since the remote center will access the original data, and the output of the analysis would be identical across sites independent of the complexity of the analysis or its subtleties (see Appendix for example). Furthermore, databases offer a single point-of-update: updating data on the server will immediately affect all analyses conducted on those data without the need to send newer versions of the data to other individuals involved in its analysis. Given proper coordination (updates should not occur during data analysis proper), this feature assures that all relevant parties access the exact same dataset.

Because database systems allow simultaneous access to data from multiple sources, they lend themselves to distributed computing of various types. One distributed approach involves cluster-computing frameworks in which multiple computers (computing nodes) work in parallel to distribute the processing of a single computing job (Pfister, 1998). Another approach, termed Grid computing (Buyya et al., 2005; Foster, 2005; Foster et al., 2001), is based on more loosely associated computing groups with intelligent 'middleware' software that makes those computers appear as a single computing resource from the user's perspective. In both types of solutions, dozens or even hundreds of computers perform analysis in parallel, simultaneously accessing the same dataset (the approach described here was implemented on a computing cluster that supports Grid computing; functionality that necessitates Grid computing is highlighted in the text).

While offering the possibility of storing data at a single location, if needed, DBMSs offer integral replication features that can speed up analyses and serve as a backup mechanism. For instance, data stored on a database in a neuroimaging laboratory can be replicated to a "mirror" database (technically known as a 'slave') at a different laboratory, allowing a remote collaborator to work on a local copy of the data if needed. This scenario is particularly useful if the dataset is very large. A large raw TS dataset can consist of dozens of gigabytes that would otherwise have to be transferred over the network during each analysis. In another scenario, the slave database might be set up on the same network as a computing cluster. In this configuration, during data analysis the cluster nodes access the data on the slave database, which is located on the same local area network as the cluster and is accessible via fast (e.g., fiber or gigabit) connections (see Fig. 1). This configuration offers more efficient data access than connecting to the original database over relatively slower wide-area network connections (e.g., Internet connections). Replication can also be used to reduce the workload on a server when multiple machines need to access the database in parallel, such as when multiple nodes are processing data simultaneously. For example, 20 nodes can be configured to query the master database, and 20 others can be configured to query the slave thus offering the required scalability for parallel environments. (More sophisticated implementations, such as 'rolling out' partial copies of a database to database engines running on the computing nodes are also possible.) Finally, slave databases serve as immediately accessible backup systems if the main system becomes inaccessible.

Existing fMRI analysis tools could potentially interface with DBMS. Current data analysis systems (e.g., AFNI, SPM, BrainVoyager, FSL) 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 'backend' in these systems would allow users to access data via database queries (rather than from a file) thus benefiting from DBMS features described above, while still retaining a familiar working environment. In addition, many software systems and programming language (e.g., Matlab, Excel, Perl, Python, C) can currently interface with relational databases, which allows for parallelized data processing by users others than those who had collected the data.

Effective and easy documentation of data structures is a natural byproduct of data representation in DBMS. Relational databases can easily be used to serve metadata such as the names of the tables in the database, the columns (attributes) that exist in each table, and the type of data stored in each column. This feature makes it easy to document the structure of the database and facilitates more effective sharing of information with others. We now turn to

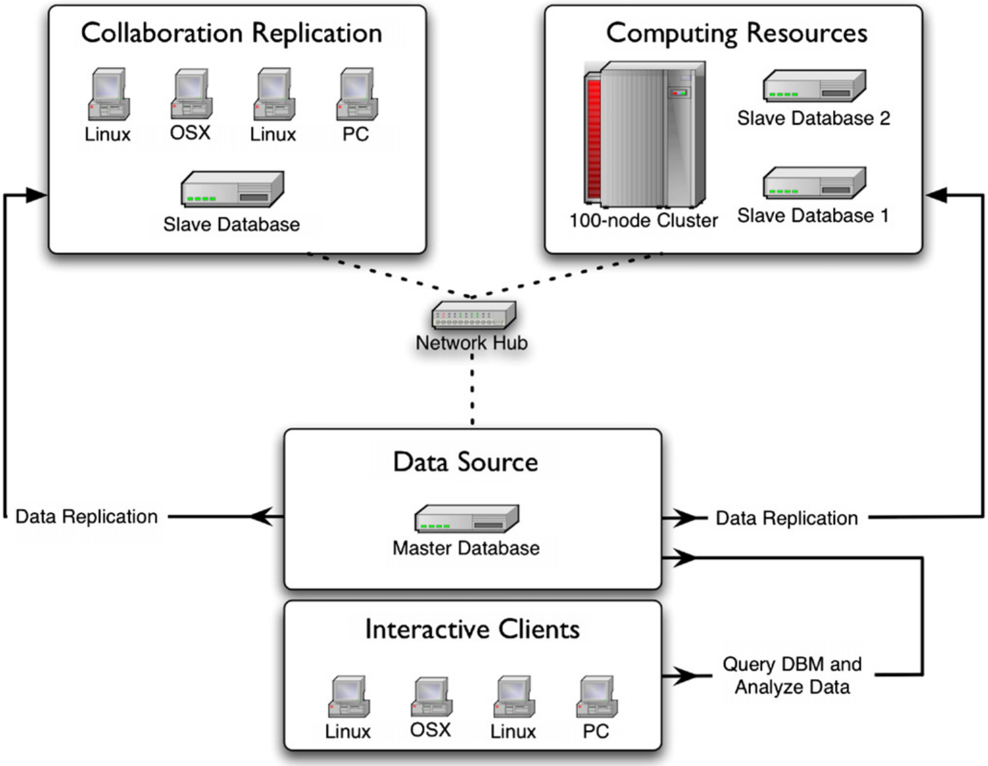


Fig. 1. Sharing and analyzing data using databases. fMRI data collected at one center (the Data Source) are stored on a Master database, and are replicated to a collaborator, as well as to a 100-node computing cluster. Collaborators can either analyze the data locally, or query data from the master database. The computing cluster holds two copies of the data using two separate DBMS servers, to serve 100 clients simultaneously.

describe the specific details of the neuroimaging data analysis system we have implemented.

System description

General

The system we have implemented is based on an architecture similar to that in the framework described above, in which distributed clients pull data from a central server, and work independently and simultaneously to conduct a voxel-based analysis (volume domain), a node or vertex-based analysis (surface-mapping domain; e.g., Argall et al., 2005), or a region-based analysis. In what follows we refer to voxels as a default, unless specifically referring to analyses conducted in the surface domain. The server maintains a relational database that stores the data that are to be analyzed as well as information about that data, e.g., the assignment of nodes to anatomical regions of interest (henceforth ROIs). The clients that conduct the data analysis are compatible with all major operating systems (e.g., Microsoft Windows, UNIX variants or Apple Mac OSX).

Server

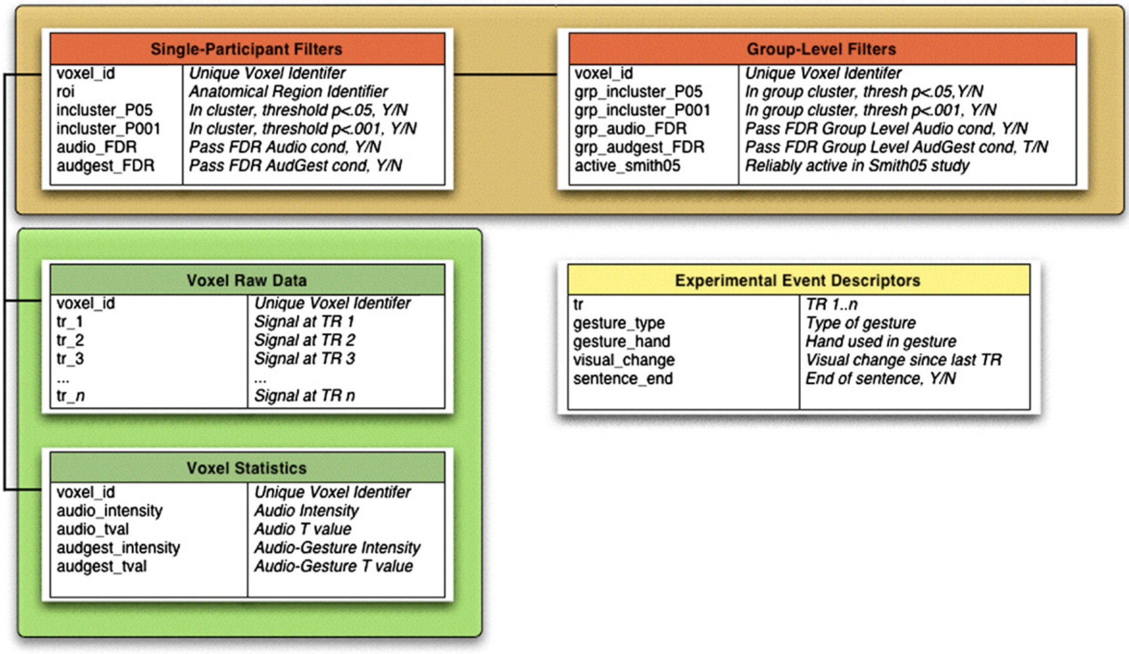
Data representation

In our implementation, each experiment is assigned a single database, and each database can contain a varying number of data tables. The guiding principle in designing such databases is to separate the fMRI data tables that store functional data for volume voxels or surface vertices (e.g., BOLD data) from the tables that hold descriptive information about these voxels or vertices.

The fMRI data for each individual participant are stored in a table (or tables) that holds all data for that participant, i.e., for all voxels (in the volume domain) or nodes/vertices (in the surface domain), for all conditions.[1] If the data are signal estimates from a statistical analysis, such tables will have [*N*(voxels) * *M*(conditions)] cells. If the data are the raw TS, the table will have [*N* (voxels) * *M*(time points)] cells. For example, in an experiment with two conditions, where each hemisphere is represented as a flat surface map consisting of 196,000 vertices, data would be stored in a table with 196,000 rows, and two columns.

Theoretical descriptions (classifications) of the data that are used for filtering and selection purposes during analysis are stored in different tables in the database (see Fig. 2). These tables are used to classify voxels or surface nodes according to criteria that are of theoretical interest. For example, one such table could associate each voxel with an anatomical brain region. Such a table would contain two columns: one for the voxel number and one for the brain region descriptor (label or number). In this case, the classification can record as many values as needed in the researcher's anatomical parcellation system. Tables can also record whether a voxel is part of

[1] We use the term "fMRI data" to refer to two types of data. One is the actual TS data, i.e., the sequences of signals from a single voxel that are measured over the entire course of an experimental run. These data are typically mean normalized and analyzed by regression models. The second type of data is the signal estimates that are the result of statistical analyses (e.g., beta values estimated from regression or deconvolution analyses).



Single-Participant Filters	
voxel_id	Unique Voxel Identifer
roi	Anatomical Region Identifier
incluster_P05	In cluster, threshold p<.05, Y/N
incluster_P001	In cluster, threshold p<.001, Y/N
audio_FDR	Pass FDR Audio cond, Y/N
audgest_FDR	Pass FDR AudGest cond, Y/N

Group-Level Filters	
voxel_id	Unique Voxel Identifer
grp_incluster_P05	In group cluster, thresh p<.05,Y/N
grp_incluster_P001	In group cluster, thresh p<.001, Y/N
grp_audio_FDR	Pass FDR Group Level Audio cond, Y/N
grp_audgest_FDR	Pass FDR Group Level AudGest cond, T/N
active_smith05	Reliably active in Smith05 study

Voxel Raw Data	
voxel_id	Unique Voxel Identifer
tr_1	Signal at TR 1
tr_2	Signal at TR 2
tr_3	Signal at TR 3
...	...
tr_n	Signal at TR n

Experimental Event Descriptors	
tr	TR 1..n
gesture_type	Type of gesture
gesture_hand	Hand used in gesture
visual_change	Visual change since last TR
sentence_end	End of sentence, Y/N

Voxel Statistics	
voxel_id	Unique Voxel Identifer
audio_intensity	Audio Intensity
audio_tval	Audio T value
audgest_intensity	Audio-Gesture Intensity
audgest_tval	Audio-Gesture T value

Fig. 2. Example of database scheme for storing data from an fMRI experiment. Each titled table reflects a table in the database and the information it maintains. Separate tables store the time series data and signal estimates (green). The database returns the data of voxels satisfying a certain criteria. If no criteria are specified, the data for all voxels are returned. Criteria are specified as constraints based on the filter tables (orange). Some filters are linked to individual participants (single-participant filters) whereas others are linked to the entire group of participants in the experiment (group-level filters).

a region that has certain functional properties, e.g., whether it is implicated in emotional processing as determined by an independent “localizer” task, whether its intensity passes a certain reliability criterion, whether it was found active in a certain previous study, or any other classification that is of interest to the researcher (Fig. 2).

Note that some of these filters may be linked to specific participants whereas others are not. For example, due to differences in brain structure, assignment of voxels to their anatomical regions will often be performed on an individual basis so that the relation between voxels and anatomical labels would be unique for each participant (e.g., as established via automatic parcellation: Desikan et al., 2006; Fischl et al., 1999). By contrast, classifying voxels according to whether or not they were active in a previous experiment on a group level would be represented in one table that would be applicable to all participants in the study. Finally, some classifications, such as whether a voxel demonstrated reliable intensity in a given condition, could be described on the group or individual participant level. This decision depends on whether a researcher wants to select voxels active at the group level, or those active for each participant on an individual basis (even though these are likely different voxels). In the latter case it would be necessary to identify separately for each participant which voxels were active in each experimental condition.

Once the descriptor tables have been constructed, researchers can rapidly select data according to highly specific criteria that implement one or more constraints in any logical combination. For the database in Fig. 2, it is trivial to select voxels that meet criteria such as being in the left inferior frontal gyrus, having a *t*-value that is greater than a certain criteria in one or more experimental conditions, or having been classified as active in a prior study. Because relational databases are designed to resolve such complex queries, it is straightforward to combine any such criteria in a query. Consider the following query that can be constructed using a single statement to extract voxel data for a focused analysis: for each participant, extract data of voxels in the left superior temporal gyrus that are part of an active cluster at group level in the audio condition, or had a reliable *t*-value in that condition at the individual-participant level. This sort of query may be particularly useful when trying to establish regularities at the group level while at the same time accounting for inter-individual differences that exist in the location of activation peaks (cf. Patterson et al., 2002).

Server implementation

The database server software that we use is the MySQL database engine, which is freely available on the Web (http://www.mysql.com/), and can be installed on UNIX variants, Apple’s Mac OSX platform or Microsoft Windows. This database system has extensive documentation, use publications, and graphical interface management tools that allow it to be rapidly mastered by non-specialists. Tables can be created via graphical interfaces or command line tools, and loaded from text files. The database supports the Structured Query Language (SQL; Eisenberg et al., 2004) that is used to specify what information is to be pulled from the database. Access to the database is typically achieved via Internet protocols, so that remote data can be accessed given proper security permissions, but for development purposes, a command line mode is also available. In our work, each experiment is assigned a unique database — a collection of data tables that contain both functional data and theoretical classifications of those data as described above. The database can also function as a job dispatch manager and manage the parallelization of jobs to the computing nodes (see Appendix). This makes it possible to run one script repeatedly, while assuring that each instance of the script is initialized with different runtime operation parameters.

Database security and controlled collaboration

Security policies on DBMS control what operations each user can perform on the data. Because the database is accessible over the network, a user's account consists of both a user name (to which a password is assigned) and a collection of hosts (i.e., terminals) from which that user can access the database. This combination assures that certain users will be able to access the database from any host, given that a password is provided, whereas others will be able to access it only from certain hosts.

Different users or groups of users can be given different rights to the data, and this is the typical approach for an fMRI study. The researcher who collected the data will likely receive all permissions to the database and remote colleagues will likely be granted more limited privileges. For example, such users should not be able to delete tables from the database, or to change their structure.

Because databases are designed with data sharing as a design principle, DBMS offers a powerful and flexible permission scheme. In MySQL, the privileges granted to an account can apply to an entire database, specific tables in the database, or even specific columns in a table. Certain users could view data in all tables in the database, whereas others could be limited to a few tables. The most basic procedures for which security would be implemented include rights to select (i.e., access) data, update data, or delete data. In many research laboratories, such security is mandated to protect the identity of subjects or patients.

Databases also offer flexible mechanisms for separating between data that are to be shared and those that are not. For various reasons, researchers are very careful with the portions of the data they share with others (cf. Ascoli, 2006), and managing the sharing of neuroimaging data is a nontrivial problem (e.g., Smith et al., 2004). To illustrate, a researcher might want to store the data of 50 participants in a database table for purposes of his or her own analyses but share only those data belonging to the subset of participants (e.g., 20 participants) whose data have been published. In a database, this is easily enabled by creating a "virtual table" (technically called a "view") that is in itself a result of a query, but that appears as a table when querying the database. In this case, the view named "limited.20ss.table" would be the result of a query selecting all data belonging to the relevant 20 participants. Other users will interact with this view as if it were a table and analyze it according to their interest (e.g., 'select all data from limited.20ss.table where condition1.tvalue > 4'). Views make it possible to share data without needing to make additional custom-tailored copies of the data to suit different types of sharing. Also, when data in the primary tables are updated, these changes are immediately seen in the views (see, Gray et al. (2005), for advantages of views in the context of scientific research).

Standards, conventions, and local practices

Given that the type of system described here is aimed at individual researchers or research laboratories, local practices will ultimately determine the structure of databases and table-naming conventions, and the nature of the metadata maintained. Though adopting a common standard aids in data sharing, in a system of the order we are describing, sharing is carried out on a peer-to-peer level (i.e., by having research centers establish direct contact), rather than via a central data warehouse that holds numerous datasets.

The development of general representation schemes that can accommodate different types of fMRI analyses and their associated data types is a matter of ongoing research (e.g., *OntoNeuroBase*, Temal et al., 2006*)*. Intensive work has also been conducted by the BIRN project (http://www.nbirn.net) to develop a logical model for documenting results of statistical analyses using XML (Keator et al., 2006). This model provides a framework for storing metadata about functional scans, functional data, and various annotations. However, it is a non-trivial task to establish a domain ontology for neuroimaging that would be readily adopted by a large number of research laboratories and aid data interoperability. On the theoretical level, one would need to establish a set of data types and characterize how these types relate to each other. Even then, it is unclear whether in practice such a general scheme would be adopted by researchers; e.g., different research centers would need to agree on a common nomenclature for naming cortical regions, possibly within a larger context of a hierarchy of brain structures (e.g., *NeuroNames*, Bowden and Martin, 1995). In the absence of such agreements, any such implementation would need to incorporate flexibility, such as accommodating multiple anatomical labelings for the same data (cf. Keator et al., 2006, for such an implementation).

In reality, the description of the data in many centers is likely to be quite idiosyncratic and even project-specific. What is important is that the database structure be accurately described, and that this description be publicly available. Once the analysis is completed and the data submitted to a central repository (e.g., fMRIDC), standard metadata conventions could be applied to the data (see, e.g., Gardner et al. (2003), for standards in central and peer-to-peer repositories).

Rather than developing a general storage scheme, during our 2.5-year experience with DBMS-driven analysis of fMRI data we instead opted to construct database schemes for different usage cases. Some schemes, as the one described in the Data representation section and Fig. 2, are quite detailed. Other schemes, supporting relatively simple analyses, contain only two tables. For example, a database set up to support analysis of a block-design experiment with three conditions (analyzed in the surface domain) would have the following fields in each table, where each field corresponds to a column in the table:

table 1 (individual participant data): hemisphere, participant_id [1 … *n*], node_id [1 … 196,000], cond1_beta [signal estimate in condition1], cond2_beta, cond3_beta.

table 2 (group level descriptor): node_id, roi_id [anatomical region in common space], reliable_cond1 [reliable by FDR on group level, y/n], reliable_cond2, reliable_cond3.

A conceptually similar study using an event-related design would have a similar table structure, except that instead of one signal estimate per each condition, the table would store the data for the estimated impulse response function (IRF) in each condition; e.g., if the IRF is estimated by 7 data points, these would be stored as cond1_tr1beta … cond1_tr7beta, and so on.

If the study were extended to include two groups of participants presented with the same stimuli under different task instructions (that is, in two separate experiments), a between-participant factor (task) would be coded in an additional column in tables 1 and 2, as follows:

table 1 (individual participant data): task, hemisphere, participant_id, node_id, cond1_beta, cond2_beta, cond3_beta.

table 2 (group level descriptor): task, node_id, roi_id, reliable_cond1, reliable_cond2, reliable_cond3.

This last example illustrates how data from multiple experiments can be stored in the same table or database when such a scheme is useful for answering the theoretical question at hand. Schemes for TS analyses can also be developed, and we detail a few in the section, Detailed example: reverse correlation analysis.

Data from separate databases can be cross-referenced or joined in a single query, if those separate databases reside on the same server. This makes it possible to extract data from one study on the basis of results derived in another study. To illustrate, signal estimates could be selected only for voxels that were reliable in a certain condition in a prior study (certain commercial DBMS, e.g., MS SQL Server, also enable queries that access databases residing on different servers). This also makes it possible to create on the fly (via SQL queries) newly 'joined' tables from data collected in two different experiments.

The example cases we have discussed above were rapidly implemented by individuals at the graduate- and undergraduate-student level, with minimal oversight by more experienced users. These use cases show that while each study may dictate its own table organization, some general principles are emerging, such as the separation of data themselves from the descriptors of the data, which allows filtering of data from one experiment on the basis of constraints from another. Implementing similar systems in research centers would likely involve a similar process, in which experience with the system will lead to commonalities in schema design and the emergence of 'prototypical' schemes.

Data storage requirements

The data storage requirements associated with storing fMRI data in DBMS depend on a number of factors, including the number of participants, and the types of data being stored (statistical estimates such as beta coefficients, and/or entire TSs). Here we report the storage requirements for two types of example datasets when stored in a database vs. when stored in imaging file formats. The first dataset consists of signal estimates in three experimental conditions for each voxel in the volume domain (73,000 voxels per each participant). The second consists of TS data (1620 acquisitions) for each vertex in the surface domain (196,000 surface vertices per hemisphere, per participant, making for more than 3×10^8 data points per participant).

The first dataset required ~37 MB when stored in the database (the database included indexes on two columns for faster data selection, which slightly increase its size). On the traditional hierarchical file system, it required ~3.5 MB when stored in a compressed binary format (BRIK.gz), or ~10 MB when stored in a non-compressed binary format (BRIK). In both the database and the BRIK files, the data were stored as a floating-point numeric type with precision of five decimal places. The command line utilities we routinely use are part of the AFNI suite and can perform voxel-based analysis on compressed BRIK.gz files thus benefiting from the smaller storage requirements.

The second dataset contained surface vertex data and consisted of several large TS files, one per each participant's hemisphere, each stored as a separate table. Each file required ~3600 MB when stored in its typical form, which is as a text file (in AFNI, surface-based analyses take text files as input rather than binary files). Each corresponding database table was ~1250 MB in size (with an index on one of the columns) when stored in the database.

When considering storage requirements, it is important to note the following: first, databases offer compression options, and in MySQL, such compression achieves between 40 and 70% reduction in data size, but entails making a table read only. The data sizes we report above are for uncompressed data. Second, storing data in compressed formats can be associated with increased processing time during data access because of the requisite decompression and recompression. Working with compressed files (e.g., BRIK.gz) via a graphical interface (e.g., the AFNI interface) can also be associated with reduced responsiveness of the interface (see http://afni.nimh.nih.gov/pub/dist/src/README.compression). Thus, implementing compression in either file-based or database environments should be carefully considered depending on the particular demands of each project. For instance, projects whose analysis has ended are good candidates for compression.

Interfaces with imaging workflow

The workflow of a typical imaging analysis consists of a large number of processing stages, often beginning from reconstructing data from *k*-space files, and culminating in thresholding. Our work to date has mainly utilized DBMS capabilities for one part of this workflow; namely, group analyses of the sort described in the prior sections. Here we consider other potential interfaces between DBMS and typical stages in imaging analysis (we follow a typical processing workflow as outlined by Smith, 2002).

The initial stages of image analysis typically involve reconstruction of *k*-space data into functional TS runs. These TS data often undergo a number of transformations before they are analyzed statistically (e.g., alignment, temporal and spatial smoothing, mean adjustment etc.). Because the TS is only analyzed statistically after these steps are completed, there is no strong reason to keep the intermediate data representations in a database as these are rarely needed following preprocessing. They can be stored offline (e.g., on backup tape), or in so-called 'near-line' solutions such as relatively slow network-mounted storage repositories.

Whether or not the final TS will be stored in a database depends on the research question. Storing the TS in the database affords convenient execution of sophisticated analyses of TS data such as structural equation modeling (cf., Skipper et al., 2007a, for an example use), and flexible selection of TS subsets on the basis of categorizations of those data (as discussed in the section, An alternative data representation scheme). Yet, oftentimes TS data are not the domain of inquiry *per se*, but are only used for establishing the relative sensitivity of each voxel/vertex to each experimental condition, using standard regression based approaches. Here, there is no strong rationale for storing the entire TS data in a database, but there is good reason to store the signal estimates in each voxel for each experimental condition, as these are the basis for the subsequent second-level group analysis. In any case, the voxels' coordinates can be stored alongside the statistical values (in the future, this could potentially allow existing command line utilities to interface with database-stored data in the same way they currently operate on flat files).

Importing data from the file representation into the database entails creating a table, and populating it with data from a text file. The following two MySQL commands create a table with three columns, reflecting the assignment of anatomical regions of interest (ROIs) to voxels for each subject, and load data into that table from a text file (vox2roi.txt):

```
create table vox2roi (subject int, voxels int, roi int);
load data local infile 'vox2roi.txt' into table vox2roi fields
terminated by ' ';
```

Database queries can be performed more quickly if the fields (columns) by which data are typically selected have associated 'indexes'. In this example, it is expected that users would want to select nodes on the basis of some *a priori* ROI classification; in this case, faster data selection could be achieved if the table is created with an index on the ROI column:

```
create table vox2roi (subject int, voxels int, roi int, index (roi)).
```

Once individual data have been registered to common space and stored in the database, group-level analyses of various types can be performed, and the results of such analyses can be stored in the DBMS in the form of information about each voxel.

After group-level statistics have been established for each voxel or surface vertex, they are typically followed by mathematically motivated thresholding procedures. Thresholding controls for the family-wise error (FWE) associated with the multiple statistical tests performed on the data, and with the fact that the data are not independent due to spatial filtering. Spatial filtering is often explicitly introduced in the workflow to increase signal to noise but is also introduced implicitly during any number of spatial transformations of the data, e.g., motion correction, alignment to common space, or volume-to-surface mappings. Some thresholding methods such as random field theory (Worsley et al., 1996) or Monte Carlo simulations of active cluster extent (Forman et al., 1995) estimate the smoothing in the dataset in each axis (i.e., the smoothing kernel specified in terms of full-width half maximum, FWHM), and use this estimate in simulations that establish voxel- or cluster-level thresholds. Currently, these utilities do not operate on database-stored data, and so the estimation of the smoothing kernel and the subsequent clustering could only be performed once the group level results have been converted to a compatible file format. Other thresholding methods, such as those based on permutations (e.g., Nichols and Holmes, 2002) or on false-discovery rate (e.g., Genovese et al., 2002) do not rely on pre-assessment of FWHM. Assessment of FDR is currently available as an "R" package, and permutation methods are easily implemented, and benefit from the capabilities of distributed computing (see Stef-Praun et al., 2007).[2]

Given the importance of being able to visually assess and report the results of imaging analyses (whether in 3D space or cortical surfaces) it is important to know how the results of analyses such as the ones reported here can be graphically displayed. While "R" has graphical output functions, these are quite generic and not customized for the complex display of brain imaging data that often involves visualization of anatomical data and functional overlays. It is also reasonable to assume that researchers would want to display the results of their group- or individual-level analyses in the same space (and interface) from which the input data originated. In some circumstances, the analysis results can be saved and immediately loaded into the graphical interface (e.g., the SUMA software can load single column text files representing whole-brain activity and display this information directly on a cortical surface image). In other cases, the results of the analyses must be imported to a native file format (e.g., using AFNI's 3dUndump). There are also two "R" packages specifically aimed at fMRI analysis that can be used to load, save and graphically display anatomical and functional data stored in ANALYZE and AFNI file formats (Marchini, 2002; Polzehl and Tabelow, in press). While we have not used these packages in our data analysis workflow, they offer the future prospect of being able to analyze data in a distributed manner and plot the results from within "R".

Clients

In the simplest implementation, both the client and the server can be installed and run on the same machine, whether for purposes of testing or actual data analysis. However, to make full use of the distributed processing capabilities, client software is usually run on a number of computers separate from the host running the database. The client sends a query to the database and receives in return a table (i.e., the set of rows) that satisfies the query (see Appendix for instructions on how to download and invoke an example "R" script that demonstrates this functionality).

Client implementation

In our approach, clients are implemented in the statistical language "R" (http://www.r-project.org), a free, publicly licensed statistical environment similar to the commercial software S/S+ (http://www.insightful.com). "R" is compatible with Microsoft Windows and various UNIX based platforms such as Linux or Mac OSX. Similar to other mathematical programming languages, scripts written in the "R" language can access and query relational databases via standard database protocols using SQL.

A simple data analysis script for a cross-participant contrast between two conditions might consist of a small number of steps, e.g.:

(1) Retrieve data from the database for a certain range of voxels (e.g., voxels numbered 1–100) [SQL Query].
(2) From the returned data, select the data for the voxel #1 [Internal R array].
(3) Conduct a statistical test on the data in that voxel (ANOVA, paired sample *t*-test) [Internal R procedure].
(4) Store the result in a temporary array; select the next voxel (step 2) [Internal R procedure].
(5) Upon finishing, write the result array to a file [Internal R procedure] or to the database [SQL Query].

The ability to analyze a large number of spatial units also makes DBMS-based approaches applicable to domains such as voxel-based morphometry (Ashburner and Friston, 2000). In such methods, where data are sampled at a high spatial resolution, the number of analysis units can exceed 1.5 million (given in-plane resolutions of 1 × 1 mm or better).

One advantage of using "R" for data analysis is that the retrieved data are directly accessible for examination and manipulation. "R" provides over 600 distinct packages for analyzing and plotting statistical data, covering domains such as Bayesian, multivariate and TS analysis, PCA, ICA, and nonparametric methods. (See the "R" reference manual: http://cran.r-project.org/doc/manuals/fullrefman.pdf.) Using these packages we have implemented analyses of fMRI data including (a) standard analysis of variance (ANOVA), (b)

[2] All the thresholding methods mentioned account for spatial smoothing (blurring) in the data. In certain cases, it could be important to spatially filter the data with different smoothing kernels and apply the same analysis to the resulting datasets. In such cases, DBMS offers a convenient way to store multiple versions of individual-level data smoothed with different kernels. These sorts of analyses could be important when it is known that a large smoothing kernel reduces sensitivity to finding activity in certain anatomical regions (Buchsbaum et al., 2005).

clustering of voxels on the basis of Beta values, (c) tests of whether the hemodynamic response peaks at different time points under different experimental conditions, (d) correlations between hemodynamic response functions in different experimental conditions, (e) post hoc contrasts, (f) analyses of functional connectivity, (g) generation of data for permutation tests, (h) voxel-wise correlations between voxel intensities and behavioral data, and (i) reverse correlation methods (see section, Detailed example: reverse correlation analysis).

Client's suitability for distributed computing environments

The availability of multiple computing nodes holds the promise of speeding up fMRI data analysis by distributing the computational load. For some analytical procedures, such a speed-up is virtually a necessity due to their intensive computational demands. Randomization methods in statistics represent a classic example of combinatorial explosion, and in fMRI analysis, such a procedure is the basis of statistical analyses using permutation tests (e.g., Bullmore et al., 1999; Nichols and Holmes, 2002), in which new datasets are created to assess whether an experimental dataset has characteristics that differ from those found by chance. In such cases the bulk of the analysis is in generating the permutations and performing clustering on each permutation, rather than in running the statistical test itself, making this task optimal for distributed computing. We have shown (Stef-Praun et al., 2007) how permutation-based statistical analysis of fMRI data can be sped up using Grid computing technologies in which multiple computing clusters parallelize both the generation and clustering of permutated datasets.

Client–server based systems are particularly well suited for parallel computing, as the clients are independent of each other, and exploit the availability of computing cycles by breaking up large analysis jobs into smaller jobs and running those jobs simultaneously (parallelizing a single job onto multiple processors can also be implemented, but this issue is outside the current scope; see Li and Rossini, 2001, for more discussion). However, achieving distributed analysis using multiple clients does not necessitate having access to a computing cluster or Grid facilities. At small scales, it is feasible to launch a number of "R" processes on computers in a local laboratory to attain similar functionality.

Detailed example: reverse correlation analysis

Here we present a detailed implementation (by JIS) of a reverse correlation analysis using the system described above. Reverse correlation is an objective method for associating properties of a stimulus with fluctuations in a TS, in this case with regional fluctuations in the blood oxygenation level-dependent (BOLD) response. In this specific implementation, the database is queried for nodes in given anatomical regions in which activity exceeds a set threshold, and the TS of these nodes is returned from the database for further analysis. The analysis requires that the parcellation of each individual's cortical anatomy into regions has been completed such that each node's data in the database is associated with a symbol (a number) that uniquely identifies an anatomical region. The number of values comprising the TS corresponds to the number of functional brain acquisitions in the study. Within each region, the TSs from the returned nodes are averaged into a single "mean" TS for that region. The fluctuations in the TS are then examined with respect to the timeline of the stimuli presented in the experiment to evaluate which properties of the stimuli correlate with the signal fluctuations.

Background

We have shown that the ventral premotor cortex (PMv) plays a role in using observable mouth movements to aid speech perception (Skipper et al., 2007b). The analysis described here examined the impact of observable hand movements on comprehension (detailed results will be reported elsewhere). Participants listened to stories (Aesop's Fables) when a storyteller was either not visible, visible but made no gestures, visible and made meaningful gestures associated with the stories, or visible but made non-meaningful self-adaptive hand movements (e.g., scratching or adjusting clothing). The present analyses tested hypotheses about the effect of hand movements on PMv activity (Skipper et al., 2006). Peaks in the BOLD TS from PMv were predicted to correspond to meaningful gestures when the gestures were visually related to the story content. In contrast, peaks in the TS from PMv were not predicted to correspond to non-meaningful hand movements in these stories. Finally, it was predicted that hand movements would not correspond to peaks in primary auditory and visual cortices.

Data processing steps prior to database import

Preprocessing stages were conducted prior to loading the data into the database and included: (a) inflating anatomical volumes to a surface representation and aligning them to a template of average curvature using FreeSurfer (Dale et al., 1999; Fischl et al., 1999); (b) automatically parcellating the surface of each participant into anatomical regions using FreeSurfer (Fischl et al., 2004); (c) importing the resulting parcellation into the SUMA software package (Saad et al., 2004); and (d) warping the resulting data to a standard mesh (Argall et al., 2005). Following these steps, all subsequent data analyses were performed on the nodes in the surface domain rather than voxels in the volume domain.

We mapped two types of data from the volume domain to the surface representation (cf. Saad et al., 2004, for details of mapping procedure). These data were: (1) each participant's TS for each voxel (i.e., the signal intensity in a single voxel over time, sampled at each functional image acquisition) and (2) the statistics derived from regression analyses performed on the individual participant data. These latter statistics were obtained by regressing waveforms of the predicted hemodynamic response in each experimental condition against the TS data, thereby establishing the sensitivity of each voxel to each experimental condition. Preprocessing of the raw TS consisted of removing artifactual spikes, removing linear and quadratic trends, and mean normalization. After interpolation to the surface, these two types of data for each hemisphere, for each participant, were imported into separate tables in the database as described in the following section.

Structure of the database

Given the typical way neuroimaging data are organized within file systems, it is simple to organize data tables in a DBMS so that their structure corresponds to this organization. While creating such analogous structures may not be the optimal configuration for a database schema (as we will discuss subsequently) it is functional and transparent, facilitating use by researchers with relatively little database experience. This was the approach taken here, in the first database schema design effort by one of the authors (JIS) with extensive experience in file-based fMRI analysis. The database *Gesture* was created in MySQL. This database contained 77 tables (five for each of the 15 participants and two global tables). Specifically, for each of the 15 participants, two tables were associated with each hemisphere: one for the analyzed functional data (beta coefficients) and one for the raw TS data, and the fifth table identified the region associated with each node. Two additional tables contained information relevant to entire group, and stored the baseline values for each participant and which experimental condition was associated with each functional volume acquisition.

Analysis procedures

"R" was used to carry out the reverse correlation analysis on a computing cluster, utilizing up to 80 computing nodes at a time. The first part of the procedure established representative TSs for the regions of interest (for each condition) and the second part of the procedure performed the reversed correlation analysis. Each computing node was assigned a group of ROIs for analysis (for exploratory purposes, 84 anatomical ROIs were examined in total). The core parallel computation process consisted of repeated database queries that selected, for each participant, the TS of voxels that were reliably active in at least one of the four experimental conditions ($T=3.32$, $p<0.001$). This query was performed for each participant, for each anatomical ROI, in both the left and right hemispheres (given that there were 15 participants and 84 ROIs in each of the two hemispheres, the query was run 2520 times). A specific instantiation of a query (in pseudocode) would be:

```
select all_timeseries_data from particpant1_leftHemisphere
Data for surface nodes that are (a) part of ROI_82 and (b)
have a t-value greater than 3.321 in at least one of the four
experimental conditions.
```

Note that this query returns information from the table containing participant 1's left hemisphere TS data (particpant1_leftHemisphereData) on the basis of constraints from two different tables: the table assigning nodes to ROIs, and the table storing for each node the *T*-valued for the four experimental conditions.

The returned TS data were partitioned (binned) by condition, generating a TS for each of the four conditions. For each such TS, time points with extreme values (signal change > 10%) were replaced with the median signal value. For each participant the TS was normalized against the baseline estimation for that participant. Then a mean TS was established for the entire group by averaging over participants. The resulting TSs reflected activity in an ROI during each condition.

Finally, for each TS we automatically identified local maxima and minima in the fluctuating signal and correlated them against the properties of the stimuli presented on the screen (Fig. 3). The TSs were first decomposed by placing gamma functions of variable heights and widths with similarity to the shape of the hemodynamic response at maxima in the TS (grey curves in Fig. 3) as determined by the second derivative of the TS (Rundell, 1990). Half of the full width half maximum (FWHM/2) of the gamma functions determined which of the aligned stimulus attributes were associated with maxima in the hemodynamic response. The distance between the FWHM/2 of two temporally adjacent gamma functions determined which stimulus attributes were associated with minima in the response.

Results and discussion

We found that in PMv, meaningful gestures resulted in peaks in the TS when those gestures described the content of the stories, and valleys in the TS when the hands were still (Fig. 3A). But, this relationship did not hold for non-meaningful gestures (Fig. 3B). Furthermore, gestures were not associated with peaks in primary auditory or early visual cortex, indicating that the PMv responded to the linguistic meaning and the semantic content in the gestures rather than to lower level acoustic or visual properties of the stimuli. The analysis above was implemented in a distributed manner on a local cluster of 128 nodes (256 processors), in which

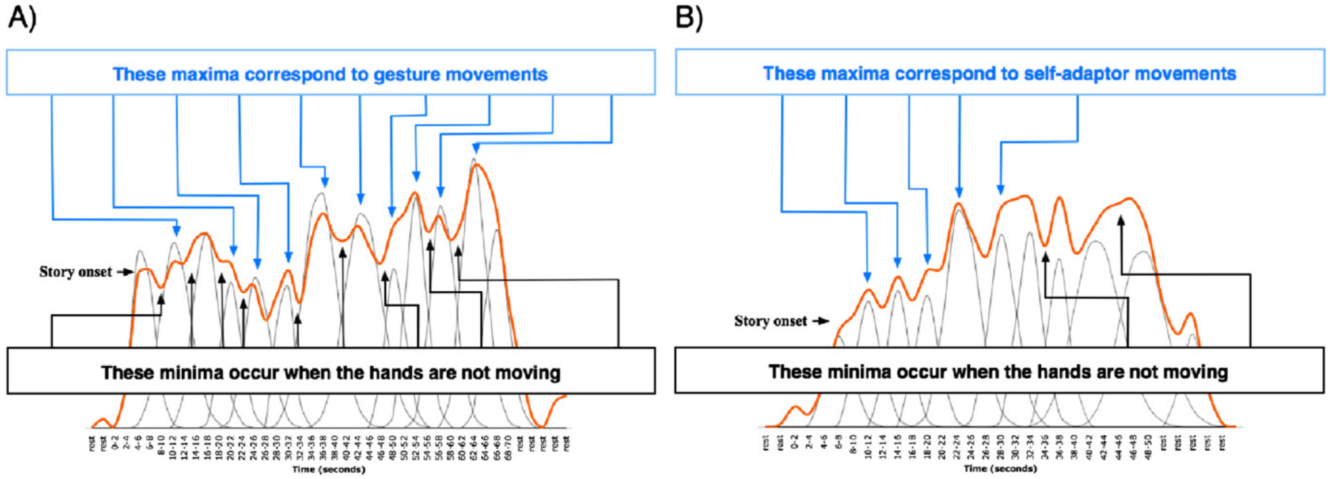
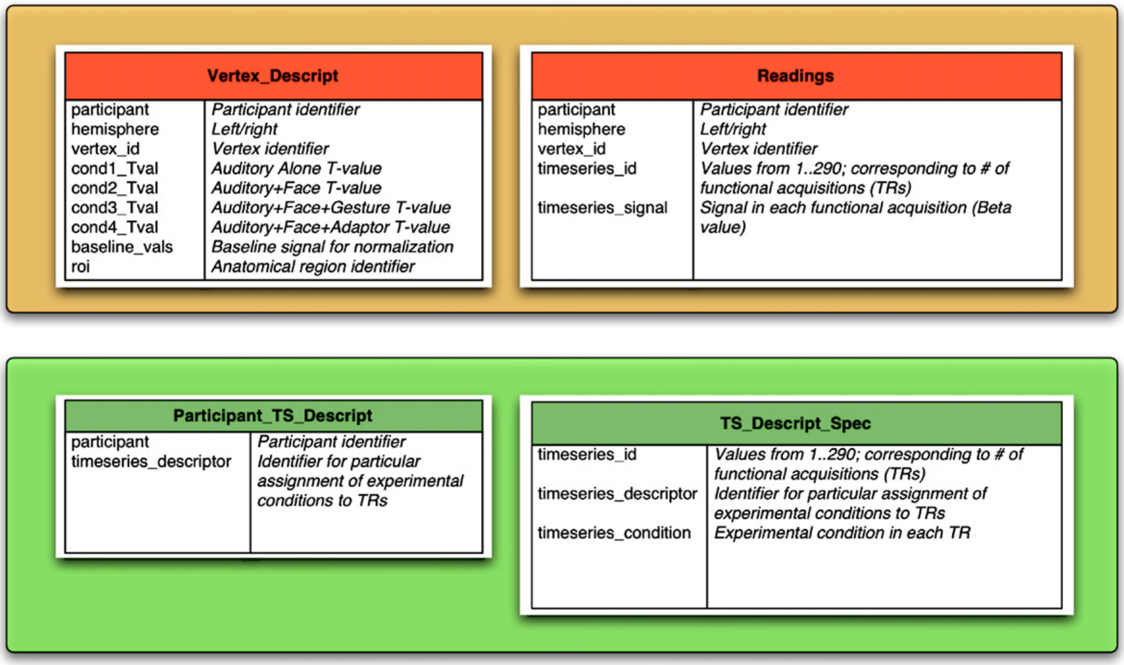


Fig. 3. Results of a reverse correlation analysis performed using a database and Grid computing. Orange lines are the hemodynamic response in the ventral premotor cortex during (A) the gesture condition and (B) the self-adaptor gesture condition, in which gestures were uninformative with respect to story content. Grey lines are the gamma functions fit to each maxima in the response. These were used to objectively determine which stimulus aspects produce maxima and minima (see text). Blue arrowed lines point to maxima while black arrowed lines point to minima. Meaningful gestures were far more likely to occur at maxima in the response than in minima, whereas non-meaningful self-adapting hand movements are as likely to occur at maxima as minima.



Vertex_Descript	
participant	*Participant identifier*
hemisphere	*Left/right*
vertex_id	*Vertex identifier*
cond1_Tval	*Auditory Alone T-value*
cond2_Tval	*Auditory+Face T-value*
cond3_Tval	*Auditory+Face+Gesture T-value*
cond4_Tval	*Auditory+Face+Adaptor T-value*
baseline_vals	*Baseline signal for normalization*
roi	*Anatomical region identifier*

Readings	
participant	*Participant identifier*
hemisphere	*Left/right*
vertex_id	*Vertex identifier*
timeseries_id	*Values from 1..290; corresponding to # of functional acquisitions (TRs)*
timeseries_signal	*Signal in each functional acquisition (Beta value)*

Participant_TS_Descript	
participant	*Participant identifier*
timeseries_descriptor	*Identifier for particular assignment of experimental conditions to TRs*

TS_Descript_Spec	
timeseries_id	*Values from 1..290; corresponding to # of functional acquisitions (TRs)*
timeseries_descriptor	*Identifier for particular assignment of experimental conditions to TRs*
timeseries_condition	*Experimental condition in each TR*

Fig. 4. Example of database schema for storing time series data from an fMRI experiment. The database schema affords selecting the time series of any given set of voxels on the basis of the voxel's estimated signal intensity or anatomical location. In addition, for each voxel it is possible to select either the entire time series, or just those time points in the series where specific experimental condition or conditions occurred.

each cluster node was assigned ROIs for analysis, and the analysis utilized up to 80 computing nodes simultaneously.[3] The speed up afforded by partitioning the job enabled allowed us to understand the results more quickly and consider and devise new questions and hypotheses.

An alternative data representation scheme

The database structure outlined in the section Structure of the database includes many tables because each participant's data were assigned a set of 5 tables. This scheme may therefore be impractical for studies involving a large number of subjects. A more efficient scheme can capture the same data in four tables, independent of the number of subjects, and affords queries that have greater or equivalent power (see Fig. 4).

Simple data queries can be performed by using just the upper two tables in the figure (*Vertex_Descript* and *Readings*). These two tables are sufficient to extract the entire TS of vertices that satisfy functional or anatomical criteria or both (e.g., select the timeseries_signal of vertices whose cond1_Tval>5 and whose ROI=5). The lower two tables in the figure allow more sophisticated queries. The first table, *Participant_TS_Descript*, marks the trial-order sequence received by each participant (e.g., some participants would be presented with the auditory-alone condition prior to the auditory-face condition and for others the order was reversed). Each trial-order assignment sequence is marked by a unique identifier in the timeseries_descriptor field. The second table, *TS_Descript_Spec*, specifies the condition presented at each functional acquisition for each trial-order sequence, e.g., whether the first acquired image was associated with the presentation of a meaningful gesture or with a less related adaptor movement. Using the information in these two latter tables, it is possible to extract only those points in the TS associated with a given condition for each vertex or region (e.g., for vertices in ROI=5, select the data acquired when gestures were presented).

This type of scheme is particularly useful for analyses of natural stimuli: for example, we were interested in the types of words or gestures presented during each acquisition. However, descriptors of a TS can also include details such as whether a sentence or phrase has started or ended at a TR or any other descriptor of interest for which a TS subset should be extracted. Indeed, any dimension of interest in the stimuli could be coded. For example, in the domain of vision, one might code the properties of the video frames, such as the amount of visual change between frames. Each stimulus dimension could be resampled to the time scale of the imaging procedure (i.e., the TR was 2 s) and entered into the database as tables. Alternatively, the TS could be resampled to match the stimuli. The fMRI TS could then be mined on a voxel or ROI basis for a relevant stimulus or combination of stimulus dimensions and their correspondences to local maxima or minima in the TS.

Discussion

The framework we have described here is one that allows both individual users and larger research centers to store data in a way that

[3] The use of multiple nodes could introduce overhead due to the load on the DBMS. We examined this issue using a representative 10-min group-level analysis job in which each computing node issued two database queries per minute (jobs were executed on a computing cluster at the Argonne National Laboratory, and queried a database at the University of Chicago). The measurements indicated that the time per job remained constant whether 5, 10, 20, 30, 40 or 50 jobs were conducted in parallel. The mean job time per computing node (in seconds+*SE*) were: 5 jobs: 496 (14); 10 jobs: 460 (12); 20 jobs: 470 (9); 30 jobs: 465 (7); 40 jobs: 479 (8); 50 jobs: 472 (6). Thus, for this representative analysis, use of even 50 computing nodes did not substantially increase overhead.

can be queried efficiently, from both local and remote sites, and that affords distributed statistical analysis of those data. Flexible sharing via ‘view’ mechanisms and flexible security are also inherent features of the system. We have provided details of the server and client implementation, and explained potential interfaces between such DBMS-based systems and other stages of a typical imaging analysis workflow.

Merging distributed computing resources with DBMS for imaging analysis

The two technologies at the core of this framework are relational database management systems that store data, making them available for remote access, and a distributed computing architecture (cluster or Grid computing) that is used for parallel distributed data analysis. Each technology offers its own distinct advantages, but the strength of the system is in the synergy between the two. DBMS oriented systems do not necessitate large scale distributed computing to aid in imaging analysis. Even when used in a non-distributed setting, the ability to access selected aspects of data from remote locations that is offered by DBMS (e.g., reanalyzing a certain ROI data from abroad) is beneficial to everyday work. Similarly, distributed computing does not ipso facto necessitate DBMS to enable faster statistical analysis. One could construct a framework in which large files are analyzed via distributed computing nodes, with ultimate collation of completed results. One implementation of such a file-based solution would be to propagate the entire dataset to each computing node and implement the types of analyses we have described above using the data selection mechanisms currently offered by command line procedures in imaging analysis packages, and efficient use of ‘mask’ files when necessary (e.g., when using data from other experiments as filtering constraints). Another file-based implementation would be to select just the data needed for any given statistical analysis and propagate those data to the computing nodes in a way that allocates a different part of the dataset to each computing node. This implementation entails a ‘pre-filtering’ step, during which a ‘mask’ of the required data is created by applying a certain filter. In this approach, the requested subset of data is constructed de novo from various flat files in order to optimize each analysis (some imaging analysis software contain functions for optimizing access to large datasets and selecting subsets of the data, e.g., RUMBA’s *librumba*, http://www.rumba.rutgers.edu/projects.php). In contrast to such implementations, a DBMS make it possible to set up a single arrangement of the data (i.e., a database scheme) which affords numerous types of queries, while at the same time serving the client with just the subset of the data that is of interest in the specific analysis, and does so without touching the rest of the data. Furthermore, as we have shown, DBMS naturally allows for data selection over networks (e.g., when conducting concurrent analysis of the same data by more than one research center). While specialized file systems can also allow such access, the implementation of network file systems specifically designed for distributed computing is non-trivial. Thus, using DBMS in the context of distributed computing for image analysis affords a relatively easy way for distributed data analysis. As we have outlined here, file-based solutions could potentially afford similar features, but to the best of our knowledge, such schemes have yet to be developed.

Target population

Who would benefit from storing imaging data using DBMS? On the basis of our experience, two distinct populations could benefit from such representations. The first are individual researchers for whom DBMS-based storage enables the execution of multiple complex analyses on the same dataset and direct and convenient access to the data. The ability to select highly specific cross sections of data from remote computers over the Internet is also an advantage for this target population, and greatly aids in collaboration and replication. Our experience shows that undergraduates, graduate students and post-doctoral students (without background in computer science), as well as technical staff, can rapidly master the basic syntax of SQL and “R” programming.

The other target population comprises the larger research centers that would likely use the DBMS-based system in the context of a distributed computing environment (whether computing clusters or distributed Grid sites). The framework offers this population a convenient method for storing and sharing data, as well as conducting advanced statistical analyses in a distributed manner. While we have emphasized benefits for analysis of imaging (fMRI, PET) data, the approach described can be extended to researchers interested in other types of data. As we have described, the bulk of database use takes place once those processing tasks more tightly linked to image analysis *per se* have been completed (e.g., filtering, registration, removal of volume acquisitions associated with artifacts). Thus, the analysis of database-stored data could potentially be extended to other types of structural data such as VBM or DTI, once those have been processed with tools specifically dedicated to those types of data.

Finally, we consider the role of new technologies in generating new methods of scientific inquiry in the community, and the likelihood that new target populations would emerge because of the availability of such systems. For example, the ability to analyze easily the same dataset and to share analysis code seamlessly across individuals could foster cooperation between small groups of individuals that transcends the traditional cooperation methods that exist today, and that are based on cooperation between research groups. Thus, one individual could store the data in a DBMS, and 3–4 colleagues would analyze the dataset in parallel pursuing specific and diverging theoretical questions.

Summary

The increasingly complex research questions addressed by fMRI research impose non-trivial demands on computational infrastructures. Already, these infrastructures need to support management of massive amounts of data in a way that affords rapid and precise data selection, to allow collaborative research, and to do so securely and with minimum management overhead. Here we have presented one approach to overcoming current limitations, which is based on freely available (open source) database management systems that support distributed data analysis using cluster or Grid computing resources. We have described how such a system is practically implemented and have shown via a concrete example the advantages offered by such systems during the analysis of imaging data. Implementing such systems in research centers is likely to facilitate cooperation between research centers and aid researchers in gaining a better understanding of their data.

Acknowledgments

We thank T. Stef-Praun for his assistance and advice, and two anonymous reviewers for their comments and suggestions. This work was supported by a grant from the National Institutes of Health: NIH R21-DC008638.

Appendix A

We have made available an "R" script that can be downloaded and executed locally by individuals interested in evaluating a system of the sort described in the manuscript. When executed, the script will connect to an example database we have set up and conduct some simple queries and statistical analyses. Individuals considering implementing MySQL and "R" may want to download the script and make changes to it. The "R" script and instructions can be found at http://www.fmri.uchicago.edu/db/db.instructs.html. Running the script requires installing "R" on the local machine with two packages that enable database access. The website also contains documentation of the job dispatch mechanism described in the Server implementation section.

Users with some experience with Mac OS X or UNIX variants should be able to install R and initialize the script without much problem, following the instructions included on the web address above. However, we do not recommend installing the R client with the database access modules on Microsoft Windows for testing purposes, because installation of the database access package on Microsoft Windows may demand compilation of software on a windows computer (in case the binary package does not install properly), which is somewhat of a lengthy process and requires specialized knowledge.

References

Argall, B.D., Saad, Z.S., Beauchamp, M.S., 2005. Simplified intersubject averaging on the cortical surface using SUMA. Hum. Brain Mapp. 27, 14–27.

Ascoli, G.A., 2006. Mobilizing the base of neuroscience data: the case of neuronal morphologies. Nat. Rev., Neurosci. 7, 318–324.

Ashburner, J., Friston, K.J., 2000. Voxel-based morphometry—The methods. NeuroImage 11, 805–821.

Bowden, D.M., Martin, R.F., 1995. NeuroNames brain hierarchy. NeuroImage 2 (1), 63–83.

Buchsbaum, B.R., Olsen, R.K., Koch, P.F., Kohn, P., Kippenhan, J.S., Berman, K.F., 2005. Reading, hearing, and the planum temporale. NeuroImage 24, 444–454.

Bullmore, E.T., Suckling, J., Overmeyer, S., Rabe-Hesketh, S., Taylor, E., Brammer, M.J., 1999. Global, voxel, and cluster tests, by theory and permutation, for a difference between two groups of structural MR images of the brain. IEEE Trans. Med. Imag. 18, 32–42.

Buyya, R., Date, S., Mizuno-Matsumoto, Y., Venugopal, S., Abramson, D., 2005. Neuroscience instrumentation and distributed analysis of brain activity data: a case for eScience on global Grids. Concurr. Comput.: Pract. Exper. 17, 1783–1798.

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

Davidson, S.B., Crabtree, J., Brunk, B.P., Schug, J., Tannen, V., Overton, G.C., et al., 2001. K2/Kleisli and GUS: experiments in integrated access to genomic data sources. IBM Syst. J. 40 (2), 512–531.

Desikan, R.S., Segonne, F., Fischl, B., Quinn, B.T., Dickerson, B.C., Blacker, D., et al., 2006. An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. NeuroImage. 31, 968–980.

Eisenberg, A., Kulkarni, K., Melton, J., Michels, J-E., Zemke, F., 2004. SQL:2003 has been published. SIGMOD Rec. 33.

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

Fischl, B., Kouwe, A.v.d., Destrieux, C., Halgren, E., Ségonne, F., Salat, D.H., et al., 2004. Automatically parcellating the human cerebral cortex. Cereb. Cortex 14, 11–22.

Forman, S.D., Cohen, J.D., Fitzgerald, M., Eddy, W.F., Mintun, M.A., Noll, D.C., 1995. Improved assessment of significant activation in functional magnetic resonance imaging (fMRI): use of a cluster-size threshold. Magn. Reson. Med. 33, 636–647.

Foster, I., 2005. Service-oriented science. Science 308 (5723), 814–817.

Foster, I., Kesselman, C., Tuecke, S., 2001. The anatomy of the Grid: enabling scalable virtual organizations. Int. J. High Perform. Comput. Appl. 15, 200–222.

Fox, P.T., Lancaster, J.L., 2002. Mapping context and content: the BrainMap model. Nat. Rev., Neurosci. 3 (4), 319–321.

Gardner, D., et al., 2003. Towards effective and rewarding data sharing. Neuroinform. J. 1, 289–295.

Genovese, C.R., Lazar, N.A., Nichols, T., 2002. Thresholding of statistical maps in functional neuroimaging using the false discovery rate. NeuroImage 15 (4), 870–878.

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

Keator, D.B., Gadde, S., Grethe, J.S., Taylor, D.V., Potkin, S.G., FIRST BIRN, 2006. General XML schema and associated SPM toolbox for storage and retrieval of neuro-imaging results and anatomical labels. Neuroinformatics. 4, 199–212.

Kotter, R., 2001. Neuroscience databases: tools for exploring brain structure-function relationships. Philos. Trans. R. Soc. Lond., B Biol. Sci. 356, 1111–1120.

Li, N., Rossini, A., 2001. RPVM: cluster statistical computing in R [Electronic Version]. R. News 13, 4–7 (URL: http://CRAN.R-project.org/doc/Rnews/).

Marchini, J., 2002. AnalyzeFMRI: an R package for the exploration and analysis of MRI and fMRI datasets. R. News 2, 17–24 (URL: http://CRAN.R-project.org/doc/Rnews/).

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

Nieto-Santisteban, M.A., Szalay, A.S., Thakar, A.R., O'Mullane, W.J., Gray, J., Annis, J., 2005. When database systems meet the grid. Paper Presented at the CIDR 2005.

Patterson, R.D., Uppenkamp, S., Johnsrude, I.S., Griffiths, T.D., 2002. The processing of temporal pitch and melody information in auditory cortex. Neuron 36, 767–776.

Pfister, G.F., 1998. In Search of Clusters: Prentice-Hall, Inc. Upper Saddle River, NJ.

Polzehl, J., Tabelow, K., in press. fMRI: a package for analyzing fmri data. *R News*.

R Development Core Team, 2005. R: A Language and Environment for Statistical Computing, Vienna, Austria.

Rundell, G., 1990. Peakfit. Non-linear Curve Fitting Software. Jandel Scientific, San Rafael, USA.

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

Skipper, J.I., Nusbaum, H.C., Josse, G., Goldin-Meadow, S., Small, S.L., 2006. The cortical motor system simulates action descriptions conveyed by words and gestures. Paper Presented at The Annual Meeting of the Cognitive Neuroscience Society, San Francisco, CA.

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

Skipper, J.I., van Wassenhove, V., Nusbaum, H.C., Small, S.L., 2007b. Hearing lips and seeing voices: how cortical areas supporting speech production mediate audiovisual speech perception. Cereb. Cortex, doi:10.1093/cercor/bhl147 (Advance Access published on January 19, 2007).

Smith, S.M., 2002. Preparing fMRI data for statistical analysis. In: Jezzard, P., Matthews, P.M., Smith, S.M. (Eds.), Functional MRI: An Introduction to Methods. Oxford University Press, New York.

Smith, K., Jajodia, S., Swarup, V., Hoyt, J., Hamilton, G., Faatz, D., et al., 2004. Enabling the sharing of neuroimaging data through well-defined intermediate levels of visibility. NeuroImage 22, 1646–1656.

Stef-Praun, T., Foster, I., Hasson, U., Hategan, M., Small, S.L., Wilde, M., 2007. Accelerating medical research using the Swift Workflow System. Paper Presented at the HealthGrid 2007, Geneva.

Stoeckert, C.J., 2005. Functional genomics databases on the web. Cell. Microbiol. 7, 1053–1059.

Szalay, A., Gray, J., 2006. 2020 Computing: science in an exponential world. Nature 440 (7083), 413–414.

Temal, L., Lando, P., Gibaud, B., Dojat, M., Kassel, G., Lapujade, A., 2006. OntoNeuroBase: a multi-layered application ontology in neuroimaging. Paper Presented at the Formal Ontology meets Industry (FOMI).

Van Horn, J.D., Grethe, J.S., Kostelec, P., Woodward, J.B., Aslam, J.A., Rus, D., et al., 2001. The Functional Magnetic Resonance Imaging Data Center (fMRIDC): the challenges and rewards of large-scale databasing of neuroimaging studies. Philos. Trans. R. Soc. Lond., B Biol. Sci. 356, 1323–1339.

Wagner, A.D., Schacter, D.L., Rotte, M., Koutstaal, W., Maril, A., Dale, A.M., et al., 1998. Building memories: remembering and forgetting of verbal experiences as predicted by brain activity. Science 281, 1188–1191.

Worsley, K.J., Marrett, S., Neelin, P., Vandal, A.C., Friston, K.J., Evans, A.C., 1996. A unified statistical approach for determining significant signals in images of cerebral activation. Hum. Brain Mapp. 4 (1), 58–73.