A VISUALIZATION TOOL FOR FMRI DATA MINING

by

NICU DANIEL CORNEA

A thesis submitted to the

Graduate School - New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Electrical and Computer Engineering

written under the direction of

Professor DEBORAH SILVER

and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

October, 2005

ABSTRACT OF THE THESIS

A Visualization Tool for fMRI Data Mining

by NICU DANIEL CORNEA

Thesis Director:

Professor Deborah Silver

fMRI is an imaging technique that is used to understand brain functionality. Scans are taken at intervals as a patient performs some mental tasks, resulting in hundreds of datasets. It is an increasingly popular technique in fields ranging from medicine, psychology or even marketing and economics. However, these images tend to be noisy and new packages are constantly being developed to analyze and filter these large datasets. Because of the large data size and many analysis parameters, comparisons between results or between experiments are difficult. We present a visualization tool that allows interactive comparison of different analyzed datasets. Such analyzed datasets can be results of different analytic methods used in fMRI analysis, on data from one or more

subjects and/or one or more experiments. We treat every analysis result as a functional clustering of voxels mapped into brain space and employ visualization techniques to allow the user to interactively explore the similarity and differences between the different datasets. This can provide valuable insight into the data or the analysis methodologies being studied. Thus, the tool can be used as a visualization interface of a data mining engine and could also support a "query-by-example" approach to fMRI data retrieval.

Acknowledgements

I would like to thank Dr. Deborah Silver and Dr. Paul Kantor for their guidance and support throughout the entire course of this work. I would also like to thank all the members of the New Brunswick-Drexel-Toronto group of the Novel Indexing and Retrieval of Dynamic Brain Images Project for their valuable discussions and suggestions. Special thanks to Narges Kasiri, for providing the first version of the user manual and conducting the first case study.

The oddball and event perception data were provided by our collaborators Dr. Stephen Hanson and Dr. Catherine Hanson from Rutgers University in Newark. The recall data sets were provided by Dr. Sean Polyn from the Neuroscience of Cognitive Control Laboratory at Princeton University directed by Dr. Jonathan Cohen.

We gratefully acknowledge the support of NSF through grant number ITR EIA-0205178.

Dedication

To my beloved wife, Amalia, for her invaluable support and understanding,

My parents, Grigore and Nastasia, and my brother, Alin, for always encouraging me.

Table of Contents

List of Figures

Chapter 1

Introduction and Motivation

Functional Magnetic Resonance Imaging (fMRI) is a medical imaging technique that captures the brain's dynamic behavior. In the last few years, it has been used extensively for both clinical applications and psychological research to map functions to specific regions of the brain. Unlike PET (Positron Emission Tomography), it does not require injecting the patient with radioactive isotopes, the contrast being provided by the varying concentration and magnetic properties of hydrogen atoms in the human body.

An fMRI *dataset* consists of a sequence of 3D images (volumes) of the subject's brain taken over a short period of time (a few minutes) while the subject is performing a given task. The experiment is designed so that certain brain functions are used more than others, allowing one to create a direct mapping of functionality to brain regions that are highlighted in the collected images.

With the increased use of functional imaging techniques, a number of different strategies of analyzing this type of data were developed. The most widely used method is known as the General Linear Model (GLM) where one assumes a pattern of activation and then checks each voxel in the image to see whether its behavior is consistent with the assumed activation model. The result of this analysis is a probability map showing the

chance that the brain region corresponding to each voxel is participating in a task-related activity. Another set of analysis methods use no assumption about activation patterns. By looking at the intensity measured in each of the collected 3D images, they try to group voxels with similar behavior into distinct classes and then label them as task-related or not. Clustering, independent component analysis (ICA), principal component analysis (PCA) and neural networks are some of the methods from this category.

With all the different analysis methods available and others constantly developing, it is increasingly difficult to compare these methodologies and asses their strengths and weaknesses. What analysis method should be used for a given set of experimental data ? Why is one method better than another ? For the same experimental data, will two analysis methods provide the same results ? These are some of the questions we are trying to address by developing a visualization tool that allows for comparisons between the outcomes of different methodologies.

Large databases containing thousands of fMRI scans are already accessible to the research community: the Brain Image Database (BRAID) database by Letovsky et al.[25], the fMRI Data Center (fMRIDC) by Van Horn et al.[37], the BrainMap database by Fox et al.[13] , etc. Data retrieval applications retrieve datasets based on search criteria specified by the user, such as text descriptions of the datasets or of the final analysis results. However, one would like to browse the results of such a query and subjectively asses the relevance of the presented results. The visualization tool developed here allows one to investigate the results of a query in the same way one would inspect the results of a web search by clicking on the links presented as relevant to the search criteria. In the case of a brain-image search, one would like to “see” the relevant results

and decide whether they present the similarity that one is looking for. Although finding an fMRI experiment in a database based on similarity to a given exemplar is still an active area of research, a visualization tool that allows for visual assessment of similarity could provide a front-end for any such database engine.

The goal of this work is to create a comparison tool that is completely independent of the present and future analysis methods. The ultimate result of any analysis methodology for fMRI data consists of a labeling of voxels in the brain as relevant or not relevant to the experimental task, similar to clustering the voxels in the brain into only two clusters: "interesting" and "not interesting". Thus, we treat any analytic result as a labeling or clustering of voxels in brain space, but not limited to only two clusters. The tool provides an easy to use interface to query the similarity between clusters in two or more analyzed datasets and a fast visualization module that allows easy mapping of the relevant clusters to spatial locations in brain. The datasets being compared can represent results of different analysis methods, query and result datasets from a data retrieval environment or data from a patient during a clinical trial that involved multiple scans.

A comparison tool can help clinicians and psychologists evaluate different analysis methods without in-depth knowledge of the underlying processing techniques. The end-results can be easily compared or cross-referenced with functional regions in the brain. Furthermore, patient response to therapy can be evaluated by comparing scans taken at regular time intervals during a recovery process.

Comparing results of different analysis methods is also an important component in the development of new analysis methodologies for fMRI datasets. Developers of new analysis algorithms can compare their results with other well-established methodologies

that are widely accepted in the field, allowing one to evaluate the new algorithm in terms of the final results it produces.

In the next chapter we present some background on fMRI data acquisition and the analysis methods most frequently used in the scientific community. Then we describe the approach and the implementation of the tool we have developed to enable comparison between different analysis results.

1.1 Contributions

In this work we have developed a real-time visualization tool that allows interactive exploration and comparison of the results of fMRI analysis. This is useful in comparing different methodologies or comparing different subjects. The tool allows quantitative and qualitative exploration of the similarities and differences present in the compared datasets providing insight into the underlying data and analysis methodology. It is completely decoupled from any existing analysis tool and can accommodate various data formats. As we will show in the discussion section, the proposed tool can be successfully used in a query-by-example approach to data retrieval from a database of fMRI studies.

Chapter 2

Background on fMRI and fMRI Analysis Methods

Magnetic Resonance Imaging (MRI) has allowed scientists and physicians to study the structure of the living human body in a safe and non-invasive manner. A large proportion of the human body (about 70% of its weight) is composed of water, and different tissue types contain different amounts of water. Each of the hydrogen atoms in the water molecule is a tiny magnetic dipole; when placed in a very strong magnetic field (about 50,000 times stronger than the Earth's magnetic field) each of these tiny dipoles will align themselves with the external magnetic field. A short pulse of energy perturbs these tiny magnets from their preferred orientation and as they return to their initial orientation they give off small amounts of energy which are detected and amplified by a *receiver coil* placed around the head. As different tissue types contain different amounts of water, the intensity of the MRI signal varies from one region to another, providing a detailed image of the internal structure of the body (the brain in our case). The resolution of the MRI images is very good, with about 0.5mm per image element (pixel or voxel).

Although MRI images provide valuable information about the internal structure of the brain, they do not say anything about how the brain functions. Functional MRI (fMRI) is a new technique aimed at mapping brain functionality.

2.1 Functional Magnetic Resonance Imaging (fMRI)

Functional Magnetic Resonance Imaging (fMRI) is a medical imaging technique that captures brain functionality by measuring the change in oxygen concentration in the blood that flows into different parts of the brain as the subjects performs some task. The tasks performed by subjects are carefully designed to emphasize one or few functions or brain areas.

The fMRI methodology provides information about brain activity by measuring and recording the side effects of the electrical brain activation. Instead of directly measuring the brain activity, it reveals the local changes in blood oxygenation in the brain tissue, which is the direct result of brain activity. It is known that some brain functions are localized in different regions of the human brain. As brain areas become active during a specific task there is a local increase in the volume of fresh oxygenated blood that flows in that area to provide the harder working neurons with the energy they need. However, the increase in local oxygen concentration is not accompanied by an increase of the same magnitude in oxygen consumption by the underlying brain tissue. As a result, an overall increase in blood oxygenation can be observed in the activated area. The magnetic properties of the hydrogen atoms differ slightly between the areas near oxygenated blood and areas near de-oxygenated blood and the fMRI signal reflects this increase in oxygen levels by an increase in signal intensity, which is why the collected signal is also called BOLD (Blood Oxygenation Level Dependent) signal. Figure 1 shows schematically the causal relations between brain activation and the fMRI signal.

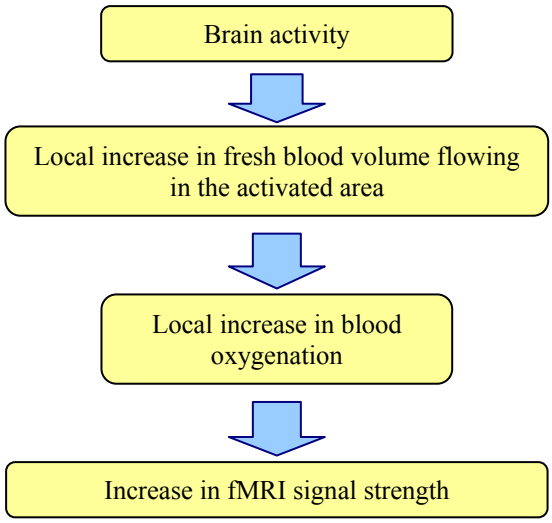
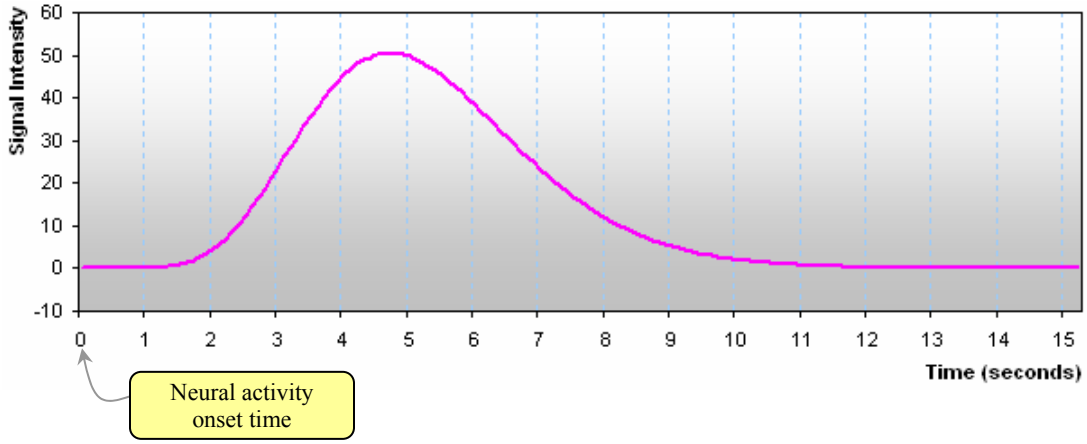


Figure 1 Causal relations between brain activity and fMRI BOLD signal.

The fMRI machine consists of a specially modified MRI scanner (*the magnet*) that can produce *functional* images by detecting these small signal intensity variations due to changing oxygenation levels.

At this stage in the evolution of the fMRI technology, the spatial resolution of the functional images is quite low, typical values for the spatial dimensions of a voxel are 3mm x 3mm x 5mm. A complete functional image of the brain is taken every 2 to 4 seconds. While it is possible to reduce the time required for a scan by focusing on a narrower region of the brain, the scan time is clearly an order of magnitude bigger than the neuron firing time (activation) in a normal brain which is in the order of milliseconds. However, fMRI does not probe the brain activation directly, but one of its aftereffects. The increase in fresh blood flow in an activated area, also known as the hemodynamic response, takes place gradually, reaching a maximum at 4 to 6 seconds after the start of the neural activity and completely disappearing only 10 to 15 seconds after the activation [3]. This slow hemodynamic process is what makes the activation visible even on such

low temporal resolution images. Although there is some debate in the scientific community regarding the exact shape of the hemodynamic response function, Figure 2 shows the gamma function model proposed by Cohen in [3].



$Si = 0.452 \cdot t^{8.60} \cdot e^{\frac{-t}{0.547}}$, where Si is the signal intensity.

Figure 2 Hemodynamic response function modeled as a gamma function proposed by Cohen in [3].

The subject lies inside this machine and is given a task to perform. First, a high resolution static MRI image (a whole 3D volume) of the subject's brain is taken; this is also known as the anatomical image and it is used for normalization. Then, the subject is asked to perform a series of tasks, during which the magnet takes 3D functional images of the patient's head at regular intervals, usually once every 2 seconds. It is common for such a session to last for a few minutes (five to ten minutes). Depending on the experiment, the task may involve a number of stimuli that the patient is supposed to react to. For example, one common task is known as the *oddball task*: the subject is presented with a set of pictures at regular time intervals and asked to react to the "oddball" picture

by pressing a button. The “oddball” picture may be a face among a series of circles. The time of appearance of the “oddballs” is recorded and provides the *stimulus onset times*. The onset times are used in most analysis packages to help process the data.

2.1.1 The Data

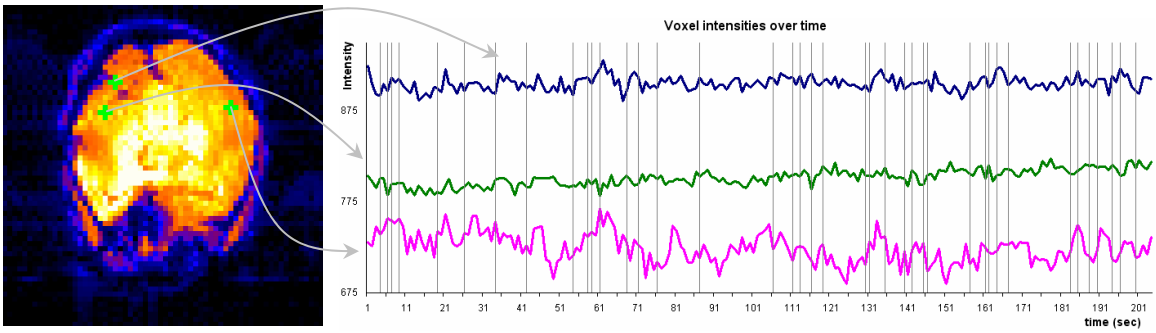


Figure 3 A slice from an fMRI dataset (oddball). The intensities of three voxels over time are shown on the right. The times when the “oddballs” were shown are represented with vertical lines on the graph.

An fMRI *dataset* contains a number of 3D images (volumes) of the subject’s brain, taken over the entire period of the experiment. The size of each volume is machine dependent but a typical value is 64x64x32. Anywhere from tenths to hundreds of volumes can be generated in a single session. For example, our “oddball” test described above contains 205 volumes taken over a period of 410 seconds, each having 64x64x16 voxels. Clearly, each experiment is different and will have a different number of volumes. Every volume is a scalar field representing the signal intensity at each voxel at the moment the image was taken. The series of intensity values corresponding to a single voxel position throughout the experiment forms the time series of that voxel. Thus an fMRI dataset can be thought of as a series of hundreds of volumes or a set of hundreds of thousands of

individual voxel time series. Figure 3 shows a slice from one of the images in the "oddball" dataset and the corresponding time series of three randomly selected voxels. The times when the "oddballs" were shown are represented with vertical lines on the graph.

2.2 fMRI Analysis Methods

Since fMRI measures and records the hemodynamic response to brain activation rather than the brain activity itself, extensive pre-processing and analysis methods have been employed to recover the original brain activation pattern from the acquired BOLD signal. fMRI analysis methods are usually grouped into two broad categories: hypothesis-driven and data-driven.

In hypothesis-driven methods, an a-priori model of activation is assumed and the data is checked to see how closely it corresponds to this model. The stimulus onset times are recorded and used for this kind of analysis. These onset times represent the initial points at which a change in the acquired signal is expected. In Figure 3, the vertical bars on the graph represent the stimulus onset times for the "oddball" experiment. In addition, a pre-defined hemodynamic response function is used to predict the effects of the brain activity. One example is the hemodynamic function showed in Figure 2. The convolution of the hemodynamic response function and the stimuli onset function provides the expected shape and duration of the BOLD signal for a voxel that shows activation in relation to the experiment that was performed. The next step uses statistical methods to decide whether the time series of a particular voxel behaves as expected for a task-related

activated voxel or not.

The most widely used package for analyzing fMRI sequences is Statistical Parametric Mapping (SPM) [33]. SPM uses hypothesis testing in the form of a generalized linear model (GLM) [16]. In this approach, a parametric statistical model of activation is assumed at each voxel location. The analysis is totally dependent on a hypothesized hemodynamic response function, which relates the dynamics of blood flow in brain capillaries (and the observed oxygen surplus or depletion) to the stimulus function built from the onset-times recorded during the experiment. The parameter values of a linear regression are estimated at each voxel location by analyzing the behavior of the voxel across the entire time sequence of the experiment. A statistical test (F- or t-test) is performed for each voxel to produce the measure of fit between the postulated activation model and the actual behavior of the voxel over time. Results are compiled into a probability map containing a value at each voxel. The 4D data is thus reduced to a 3D scalar field containing probabilities for each voxel. A high value indicates a high probability that the voxel's behavior over time is linked with the activation model and the performed task. Thresholding the probability map at high values selects those voxels that most likely show task-related activation during the experiment. SPM99 [33] is a self contained and lock-step program that reads in the data and produces the probability map.

There are many other hypothesis-driven functional MRI analysis tools based on the same GLM approach. These include: AFNI (Analysis of Functional NeuroImages) [4], FSL (FMRIB Software Library) [14], VoxBo [39], RUMBA (Rutgers University Mind Brain Analysis) [31], etc.

The second category of methods is data-driven or exploratory methods [35] where no

prior hypothesis about the activation pattern of a voxel is assumed. Using only the intensity signal over time, the data-driven methods attempt to classify voxels into distinct classes based solely on signal characteristics. The data driven methods include Independent Component Analysis (ICA), Principal Component Analysis (PCA), Canonical Correlation (CCA) and clustering.

ICA, PCA and CCA attempt to recover independent mixed source signals from recordings at several locations (this is also called the blind separation problem). In the case of brain images, the independent sources could be different functional areas of the brain and the recordings are the intensity of the signal over time as measured at different locations (at every voxel). The methods attempt to identify a number of uncorrelated signals that could have produced the observed data. Additional constraints differentiate between these methods: ICA tries to maximize the statistical independence of the components; PCA imposes the orthogonality constraint on the source, while CCA tries to identify components that have maximum autocorrelation [15].

Clustering methods are extensively used for fMRI analysis both for post-processing the results of the hypothesis-driven analysis (where spatial clusters are found after thresholding the t-test from SPM processing for example) or as a data-driven exploratory method. In the case of exploratory methods, after identifying all the different clusters based on signal patterns, the correlation with the stimulus function (or the activation function) is used to rule out clusters that do not seem to capture task-related activity.

There are several possible bases for clustering. In one, each voxel is taken to be a point in the n-dimensional space where n is the number of scans in the fMRI sequence. A point in n-dimensional space is identified by its n coordinates taken to be the intensities

of the BOLD signal at that voxel at each of the *n* scans in the experiment. The resulting clusters have a meaning in terms of brain functionality, since voxels behaving in a similar manner throughout the scan will map to neighboring points in this space. It was shown in [19] that clustering on the cross-correlation between stimulus and voxel behavior is more robust than clustering on raw fMRI time series. The output is the set of clusters such that all voxels in a set have similar correlations with the stimuli pattern. New clustering techniques are continually being developed, as are improvements to existing methods e.g. [8][18].

2.2.1 Normalization

The two data analysis categories do not differ in the extensive pre or post processing that is required in order to formulate a conclusion from such an experiment. Since brain size and shape varies considerably among different subjects it is imperative to have a standardized coordinate system to identify regions of the brain that are considered to be active during a given experiment. The Talairach brain was published by Talairach and Tournoux [34] and introduced three important innovations: (a) a coordinate system fixed to anatomical landmarks in the brain; (b) spatial transformations to map one brain to another; and (c) an atlas of the standard brain with anatomical labels [1].

The Montreal Neurological Institute (MNI) created the MNI template brain based on the average of several hundred MRI scans of normal brains. The International Consortium of Brain Mapping (ICBM) adopted the MNI template as an international standard. Locations in Talairach and MNI brains can also be mapped to the well-known (to neurologists) Brodmann areas, which are a classification of brain areas based on tissue

structure [1].

Mapping the data to the standard space is a two-step process: first, the raw data has to be aligned with the subject's high-resolution anatomical image (MRI) taken at the beginning of the experiment, a phase called co-registration; then, in the normalization phase, a transformation that maps the high-resolution image into the standardized space is computed. The transformations resulting from each of the two steps ($T_{\text{fMRI to anatomical}}$ and $T_{\text{anatomical to standard}}$ are concatenated into a single transformation that transforms the fMRI data into the standardized space: $T_{\text{fMRI to standard}}$. Figure 4 illustrates this process.

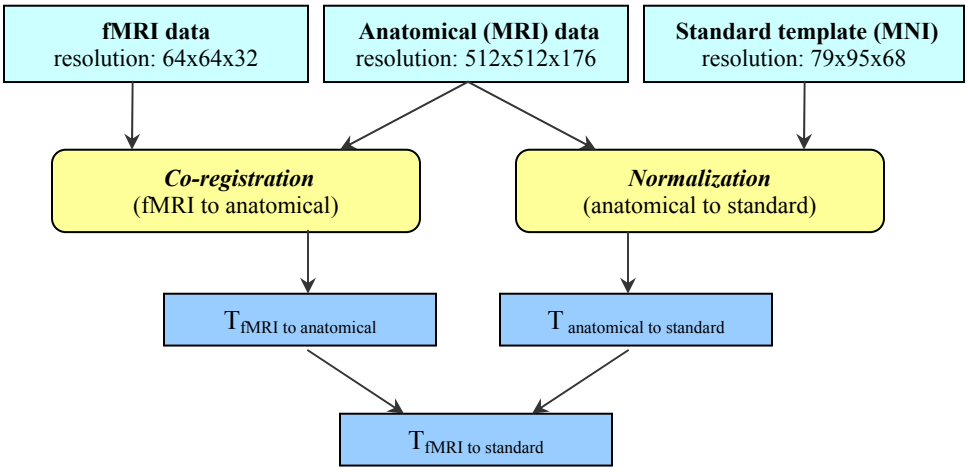


Figure 4 Registration of fMRI data to the standardized space.

This process can be done either as a pre-processing step or as a post-processing step; in the later case only the resulting activated regions are mapped to the normalized space.

Whatever the methodology, the ultimate goal of any type of fMRI analysis is to attempt to partition the brain itself into regions that are somehow related to the experiment and regions that are not. As discussed above, a number of different methods have been devised to accomplish this partition and comparing the efficacy of these

methods is non-trivial, particularly because the “underlying truth” is typically unknown. Researchers thus face many tools to use, and no good criteria on which to make a selection. Being able to compare the results of different algorithms on real datasets would be useful in understanding different techniques and determining their efficacies. The approach taken here is general enough to allow one to use the results of any kind of analysis methodology in this process.

Chapter 3

Related Work

In the fMRI community, research is focused on developing new analysis methods or improving the existing methodologies. The majority of researchers use the hypothesis-driven approach in the form of a generalized linear model (GLM). All these tools like SPM, AFNI, FSL and RUMBA have limited visualization capabilities consisting of axial views of projections of the entire dataset or individual slices only. Maybe the most important aspect of the analysis of an fMRI dataset is the mapping of the activated clusters to a set of labels identifying anatomical and/or functional regions of the human brain like the Brodmann regions or Talairach labels. This mapping is performed by a table look-up using the normalized coordinates of the activated regions and the results are presented as a table with a cluster identifier, the size of the cluster and the associated coordinates. While this is a rigorous way of presenting the information, a visual representation of this mapping would be more intuitive and easy to parse.

Several software packages were developed for visualization of MRI and fMRI data. Among these, BrainVISA/Anatomist provides pipeline building facilities for processing and visualization of brain imaging data; MRIcro is a visualization package that supports a variety of medical image formats.

Since in our approach, all analysis results are considered to be clusters in the three-dimensional space of the brain, a related topic is visualization of data clustering in three or more dimensions, a task useful in data mining. In [36], Tzeng and Ma presented an information visualization technique for classification of multi-dimensional data. The application is a front-end for a multi-dimensional classification (clustering) engine. An easy-to-use user interface provides not only visualization of the classification results, but also allows the user to interact with the classification algorithm for parameter tuning using operations like splitting a class into finer components, or merging two related classes. In [5], Davidson presents a particle-based framework for visualization of multi-dimensional clustering results. The original data points and the identified cluster centers are represented as particles in the three-dimensional space using Multi Dimensional Scaling (MDS) to reduce the dimensionality. Cluster centers are mapped to points in a 3-dimensional cube such that similar clusters are close to each other, and dissimilar clusters are as far apart as possible. The data points are shown as particles attracted by the cluster centers with forces that are proportional to the degree of membership to each cluster, such that the framework is useful for fuzzy clustering as well, where data points are assigned to more than one cluster.

Visual exploration of multi-dimensional datasets is another related field. In particular in the fMRI field, the data is 4-dimensional and the result of the analysis is a 3-dimensional map of the activation regions. XmdvTool [38][43] is a tool for visual exploration of multivariate datasets. It employs several multi-dimensional data visualization techniques such as: scatter plots, glyphs, parallel coordinates or dimensionality reduction. It provides tools to highlight data of interest like the n-

dimensional brush. A Visual Hierarchical Dimension Reduction (VHDR) methodology described in [44], also incorporated in XmdvTool can be used to reduce the dimensionality of the data, allowing the user to interact with the dimensionality reduction algorithm in order to create meaningful visualizations in a reduced dimensional space. In IBM's TableView, multi-dimensional data are represented in a dynamic table with dimensions represented as columns and data points shown as rows. The user can collapse rows/columns to focus on data of interest.

Also related is the data mining approach, where large databases of fMRI images [25][37][13] are mined to find relationships between brain lesions and functional deficits (e.g. vision or speech deficits) [25][27], or to classify patients based on fMRI activation patterns [9]. At present information retrieval is usually based on text queries and metadata stored with each dataset. With the proposed tool it may be possible to use clustering and visual representation in a query by example mode that will retrieve similar datasets based on actual data similarity not on similar textual description. Furthermore, the user is allowed to interactively explore the similarities and differences between the retrieved datasets. More recently, Forsberg, in [12], described a visualization and query tool designed for the NeuroGenerator database system [28]. NeuroGenerator is a repository of fMRI data which collects the raw data and analyses it in a homogenous way, using a single analysis package. The tool described in [12] allows visual exploration and comparison of several analysis results and can be used to formulate queries in terms of both activated clusters and keywords associated with each experiment.

Chapter 4

The Problem

The large number of existing analysis methodologies for fMRI, each with their own advantages and disadvantages generated a mixture of data formats and difficulties in comparing the results of these different analyses in an intuitive way. Our goal is to provide scientists and algorithm developers with a visualization tool that will allow intuitive and rapid comparison between analysis results of different fMRI datasets obtained using various methodologies. The tool should have an intuitive and easy-to-use interface and should allow the user to formulate simple queries in order to asses the similarities of differences present in the examined datasets.

For scientists, such a tool may be useful in determining which analysis method is best suited for their specific needs. It is known that different analysis tools may give very different results partly because different algorithms are used and partly because of the different parameter settings. One issue that was only addressed in the later versions of the different analysis packages was logging which can provide reproducibility of results. A tool for comparing results of different analysis methods and even results obtained using the same tool under different parameter settings provides an invaluable help in deciding which analysis tool to use for certain experiments. Algorithm developers looking to

create new analysis methodologies could also benefit from our tool by being able to quickly compare the results of their new algorithms with the results of recognized analysis tools. Examples are shown in the discussion section.

Additionally, a tool that can highlight various similarities between two datasets can be used effectively as a visualization interface for an fMRI database environment. Assume we have a database of classified fMRI datasets, including both raw data and analyzed versions of each dataset. In a query-by-example approach to searching, a user presents the system with a new dataset and asks it to scan the database and retrieve similar datasets. A more general problem is the data mining problem, where the data mining engine is asked to find similarities in the database without a specific key to search for. As soon as the system produces the results of this query based on some built-in similarity metric, the user would like to investigate these results and subjectively asses how similar they are with the query dataset or among themselves. It should be no surprise that the user may have a different notion of similarity than the automated system. Our tool allows one to quickly find the similarities in the analyzed versions of these datasets, provided the analysis results are stored together with the raw time series. The user can now conclude whether the search was successful or not and maybe tune the search parameters for another try. Figure 5 illustrates the fMRI data mining and data retrieval scenario. This functionality is similar to that of a web browser: following a web search, the user is presented with a set of links that he or she can click and investigate whether the retrieved documents are relevant or not.

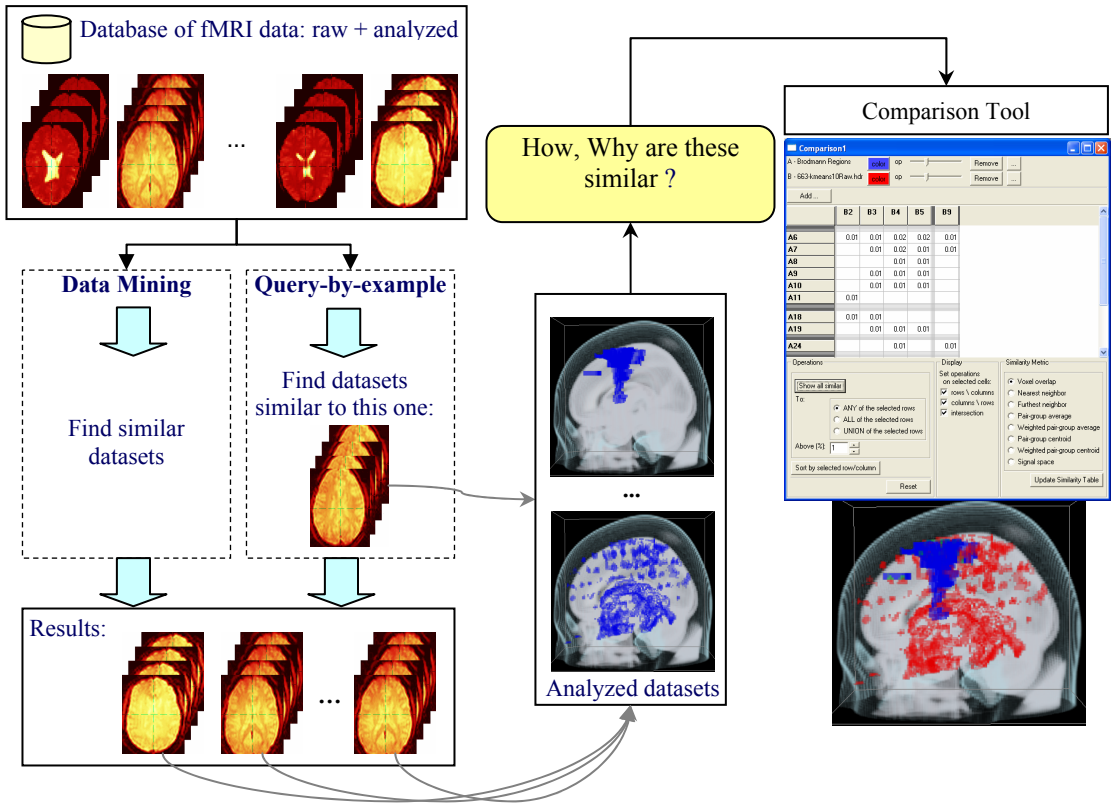


Figure 5 fMRI data mining and data retrieval scenario.

The same tool could potentially be used for the actual retrieval operation provided the analyzed versions of the datasets are also stored in the database and the similarities between different datasets can be quantified and ranked.

The original idea for the development of this tool came from the difficulty of comparing the results of various clustering algorithms we employed on the raw fMRI time series. Later, the extension of the tool to include comparison of any kind of analysis was straight-forward.

Chapter 5

Approach

We focus our attention on providing a visualization tool that can facilitate the interactive exploration of analyzed fMRI datasets as well as a query-by-example approach to fMRI data retrieval. We use the fMRI analysis results to provide the necessary information needed for comparison of two datasets and we provide a spreadsheet-like interface to query the data. The queries are mapped back to the brain space to allow for visual exploration of the relationships present in the datasets.

The notion underlying this work can be thought of as a complex extension of a Venn diagram, in which the overlap between two sets is shaded to call attention to it. Our system displays groupings or labelings of voxels, which result from various analytical programs. The user is allowed to visualize the groupings and determine which "overlap" or are similar using any one of a set of similarity metrics. The spreadsheet-like interface facilitates questions in an intuitive and graphical manner.

When processing an fMRI time series, regardless of the employed methodology, one is interested in the set of brain voxels that are activated as a result of the experiment. Thus, the result of any analytical processing of an fMRI dataset can be viewed as a labeling of the voxels in the brain space as "interesting" (meaning that those voxels show

an activation pattern that is correlated with the experimental conditions) or "not interesting". In consequence, we treat any analytic result as a labeling of voxels in the brain, and we will use the term **cluster** to identify a set of voxels with the same label. Note that the term cluster does not imply spatial connectivity or proximity among voxels with the same label. The term cluster is used in a functional way, to mean that voxels in the same cluster behave similarly, as determined by the analysis tool. Although, labeling voxels as "interesting" or "not interesting" seems to be the final goal of the analysis, we do not limit our tool to only two labels, since this distinction may not be known in certain phases of the processing. For example, a clustering based approach to fMRI analysis that works in the signal space will directly produce a label for each voxel representing the cluster it belongs to but the distinction "interesting" vs. "not interesting" is not known at this stage. A representative from each cluster will later be compared with the signal representing the experimental conditions and clusters that show good correlation are labeled as "interesting". We believe our visualization tool can also be useful in the early stages of the processing especially for algorithm developers as we will show in the discussion section, thus our tool will support an arbitrary number of labels in each dataset. Additionally, we would like to support special datasets such as Brodmann regions or a brain atlas, which are just another labeling of voxels in the brain based on anatomical structures or functionality and which clearly have more than two labels. We will show in the discussion section how such special datasets can be used with our tool.

Voxel labels can be assigned either by the analysis method or by our tool at the time the dataset is loaded. The analysis method can indicate that a set of voxels exhibits similar behavior by assigning them the same label. This is the case of clustering methods

where voxels with the same label belong to the same cluster, which usually groups voxels with similar signal patterns. Depending on the clustering scope, voxels in the same cluster may or may not be spatially connected. For example, applying the canonical sets algorithm [7][6] to an fMRI sequence results in a set of representative voxels for the entire sequence. Then, using a minimum distance classifier that works in the n-dimensional signal space (where n is the number of scans in the experiment), every voxel is associated with one of these representatives. The result is a set of clusters that group voxels with similar behavior in time, each voxel being labeled with the number of the cluster it belongs to (Figure 6).

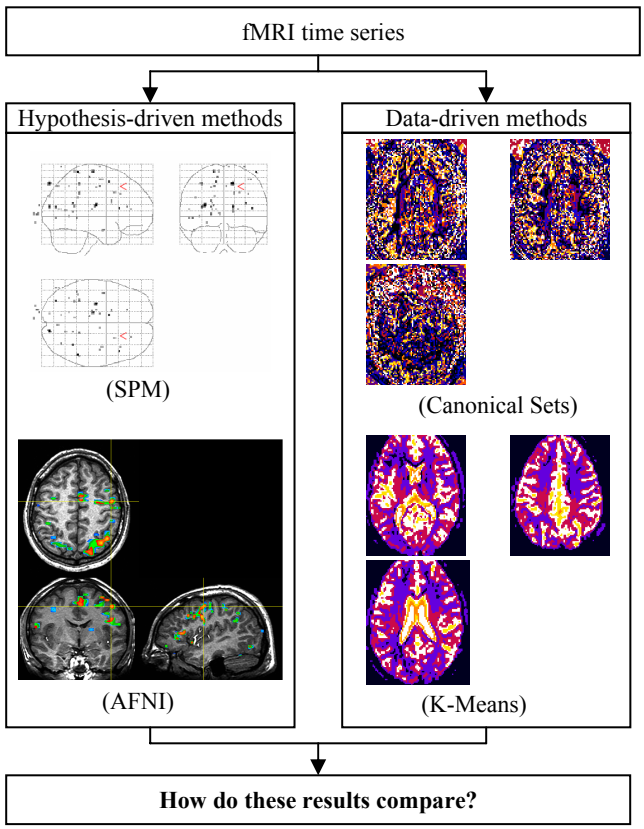


Figure 6 Different analysis methodologies and the kind of results they produce.

Another widely used clustering algorithm is k-means; here voxels are directly clustered in the signal space and the result is again a voxel labeling to the k clusters (Figure 6). Clustering algorithms can be applied to the raw time series (signal space) or cross-correlation data [19], representing cross-correlation between the voxel time series and the stimulus function.

GLM analysis methods such as SPM, can also produce clusters of activated voxels, but the general output of such a method is a probability map. For example, an SPM analysis produces a parametric map describing the probability of each voxel being significantly correlated with the experimental conditions. Voxels that have a probability value above a certain threshold are considered active. It is important to support this kind of results since the GLM methods are the most widely used in the field. We have two ways of converting this type of datasets to our "labeling" requirement when loaded into our application: we can consider that all voxels with probability value above a specified threshold form one functional cluster and we can assign them the same label. At the same time, the voxels that fall under the threshold are discarded or assigned a different label. Figure 7(a) shows the result presented by SPM after thresholding the probability map.

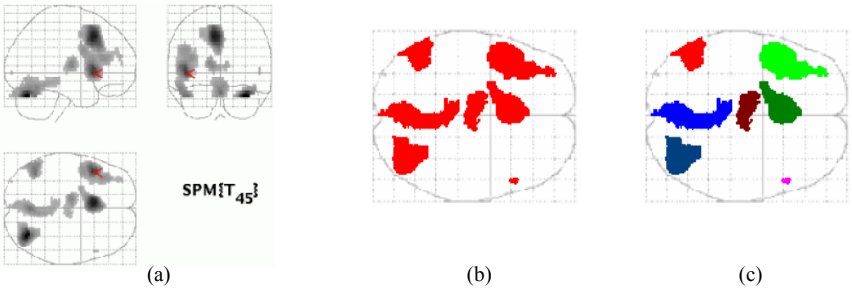


Figure 7 Treating SPM results (a) as one functional cluster (b) or many functional-and-spatial clusters (c).

Note that the selected voxels showed in black, form several spatially coherent clusters but we are assigning them the same label producing one "functional cluster".

Since now we have control over the labeling process, we can use a simple clustering algorithm based on spatial location and assign different labels to the selected voxels, creating several "spatial-and-functional clusters", one for each connected component that can be identified in the image: Figure 7(b and c). Thus the user may view an SPM-like result as defining a single functional cluster, or apply a subsequent clustering algorithm to transform it into a set of spatial and functional clusters, an approach consistent with the belief that well-defined regions of the brain are responsible for certain functions. To generalize this approach, we could apply the same clustering step to the results presented directly in labeled form (from clustering algorithms) and split the given functional clusters into several spatial-and-functional clusters.

A formal specification of the problem dealt with in this work is as follows: given a set of n different fMRI datasets $a_1 \ldots a_n$ (where each a_i is a 4D dataset resulting from an experiment), and a set of analysis routines $f_1 \ldots f_m$, where $f_x(a_i)$ represents the result of the analysis routine f_x applied on dataset a_i (*the processed dataset*), we would like to understand the relationship between:

- $f_x(a_i)$ and $f_y(a_i)$: comparing results of different analysis algorithms applied to the same dataset.
- $f_x(a_i)$ and $f_y(a_j)$: comparing results of different algorithms on different datasets.
- $f_x(a_i)$ and $f_y(a_i)$: comparing results of one algorithm on different datasets.

- $f_x(a_q)$ and $f_y(a_1)$, $f_z(a_2)$, … $f_w(a_k)$: query-by-example data retrieval: assessing the similarity between the clusters in a query dataset $f_x(a_q)$ and the clusters in a number of classified datasets $f_y(a_1)$, $f_z(a_2)$, … $f_w(a_k)$ returned by the search algorithm.
- $f_x(a_q)$, $f_y(a_1)$, $f_z(a_2)$, … $f_w(a_k)$: in the more general framework of data mining, these could be datasets returned by the data mining engine.

Our visualization tool will provide an intuitive interface that will facilitate a set of queries targeted at discovering similarities among a number of processed datasets. Given a number of processed datasets with voxels labeled according to a number of clusters (the voxels may also have associated signals), the basic queries we would like to provide are described below:

Cluster-wise similarities:

- Show all pairs of similar clusters in two or more datasets.
- For any specific cluster in one of the datasets, show all similar clusters in all other datasets.

Group-wise similarities:

- Given a set of clusters in one dataset, consider the cluster formed by the UNION of these clusters (union cluster) and show all clusters in other datasets that are similar to the union cluster.
- Given a set of clusters in one dataset, show all clusters in other datasets that are similar to ANY of the clusters in the selected set.
- Given a set of clusters from one dataset, show all clusters in other datasets that

are similar to ALL the clusters in the selected subset (similar to every one of the selected clusters).

We will now define what we mean by similar in the context of voxel clusters resulting from the analysis of an fMRI dataset.

5.1 Cluster Similarity Measures

There are a number of methods to compute the similarity between two clusters that were developed for the use of clustering algorithms, where one would like to merge two clusters if they are similar, or split a cluster if its components are not similar enough [20][21]. In our case, we want to assign a similarity score for every pair of clusters taken from different datasets. This is meaningful since all clusters in all the different datasets are ultimately mapped into the same space: the brain space. The various distance metrics used to quantify the similarity of two clusters are described below [32][2]:

- **Voxel overlap**: because all the clusters in our application map to the same spatial domain (i.e. the brain) the number of voxels in the two clusters that overlap seems the obvious choice for measuring similarity. A numeric value for the similarity of two clusters a and b based on voxel overlap, can be obtained using the following formula (also known as the Jaccard coefficient):

$$S = \frac{a \cap b}{a \cup b}$$

In other words, the similarity score of a and b, is the ratio of the number of cluster components (voxels) located at the same position in the common space (common voxels),

to the total number of voxel in union of the two clusters. This coefficient ranges between 0 and 1, 0 indicating total dissimilarity and 1 indicating that the two clusters are identical.

Voxel overlap is a simple similarity measure that unfortunately does not work very well for small clusters in the presence of noise. It is well known that the normalization process that maps the experimental results into the standardized space is prone to a number of different sources of errors. As a result, the clusters (activation regions or simply clusters from a clustering algorithm) may be offset from their true position by a few voxels. While this is not a problem for big clusters with hundreds of voxels, it becomes a serious drawback when working with small clusters with just a few voxels, because the voxel overlap based similarity of two identical small clusters may even become 0 if one of them is offset by just a few voxels such that they have no common voxels anymore. Figure 8 shows such an example.

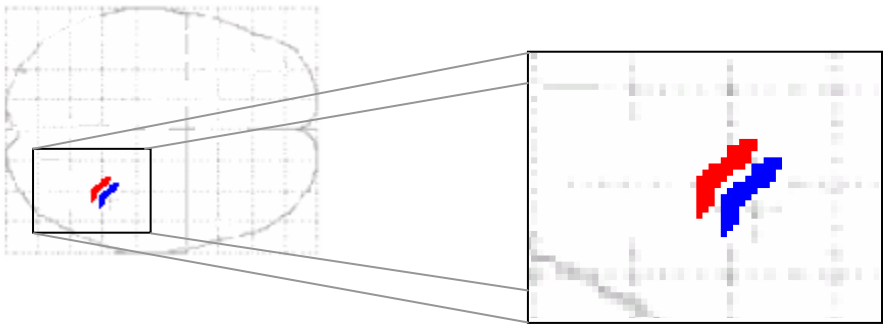


Figure 8 Two small identical clusters that have no overlap at all as a result of registration errors.

Small clusters are not at all uncommon among the results of fMRI analysis, given the poor spatial resolution of functional data and that some functional regions of the brain are actually quite small. The following similarity metrics provide the means to deal with this

problem:

- **Single linkage (nearest neighbor)**: the distance between two clusters is determined by the distance between the two closest voxels in the two clusters.
- **Complete linkage (furthest neighbor):** the distance between two clusters is determined by the greatest distance between any two voxels in the two clusters.
- **Un-weighted pair-group average**, where the distance between two clusters is the average distance between all pairs of voxels in the two clusters.
- **Weighted pair-group average** is similar to un-weighted pair-group average method, but uses the size of each cluster as a weight in the computation.
- **Un-weighted pair-group centroid**, where the distance between two clusters is defined as the distance between the centroids of the respective clusters.
- **Weighted pair-group centroid**, which is similar to the un-weighted pair-group centroid method above but weights each centroid by the size of the respective clusters.

The distance between two points in two different clusters can be taken to be the simple Euclidean distance in 3-dimensional space, or any other metric in this space. However, for the problem at hand, the original raw time sequence may be available. In that case, each voxel has an associated BOLD signal (intensity value over the entire experiment), which could be used to compute similarity between clusters based on distance between two points in the signal space. Alternatively, other statistical cluster properties derived in signal space like mean, variance, etc could be used for the purpose of obtaining a similarity score.

Note that while computing the similarity score of two clusters, voxels are not being regrouped using these metrics (we do not re-cluster the data).

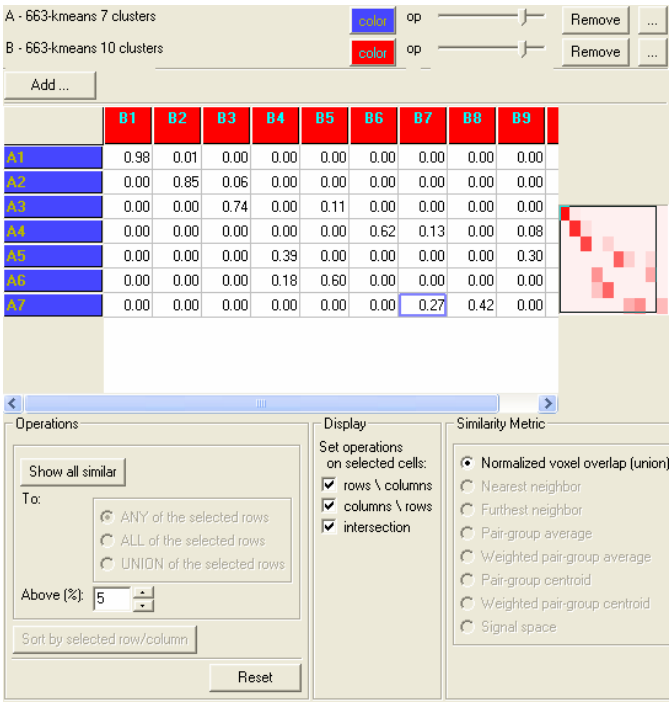
The similarity scores between each pair of clusters, with the two clusters belonging to different datasets, will be presented to the user in an intuitive interface that can naturally support all types of queries described in this section and provides easy access to meaningful visualizations. For this purpose we have selected a familiar spreadsheet like interface that we will describe in the following section.

Our tool is completely independent of the analysis method applied to the fMRI time series. However, there is one requirement that has to be met by the final results of the analysis in order to allow comparison between different experiments: the results have to be presented in the same standardized space i.e. they have to be normalized. The normalization process can either be conducted prior to the analysis or as a post-processing step.

Chapter 6

Implementation

Our implementation of the visualization tool is centered on the concept of comparison between different analyzed datasets. As described in Chapter 5, we treat every analyzed dataset (result of some type of analysis on the raw fMRI data) as a labeling of voxels in the brain, using the term cluster to refer to groups of voxels with the same label. Our tool provides an intuitive user interface, organized in two independent panels: the data operations panel (Figure 9a) and the visualization panel (Figure 9b).



(a)

(b)

Figure 9 The user interface of our tool. (a) Data Operations Panel; (b) Visualization Panel.

The two components of the user interface have complementary roles. The data operations panel displays quantitative information about the datasets being compared and provides the means to formulate the types of queries described in Chapter 5. It also allows the user to quickly identify the clusters that present high mutual similarity and are worth investigating. On the other hand, the visualization panel provides a visual display of the selected clusters in a common space overlaid onto a rendering of the anatomy of the brain.

We have chosen such a layout so that the user can focus on the visual aspect of the data being compared, shown in the visualization panel, or on the quantitative aspects of similarity presented in the data operations panel. This design provides the greatest flexibility, allowing one to completely ignore the quantitative similarity scores and focus solely on the appearance of similarity that results from the visualization, or the other way around.

We will provide a more detailed description of the two components in the following sections.

6.1 The Data Operations Panel

The data operations panel, shown in detail in Figure 10 is the host for the quantitative aspects associated with the basic functionality of the application: the comparison between several datasets. This is why we will also refer to it as the comparison panel. It is divided into several sections:

- Dataset properties area (DP) – provides controls for loading, removing and

setting the properties of the datasets involved in a comparison.

- Similarity table (ST) – shows similarity scores both in numerical and graphical form (GV).
- Operations area (OA) – provides tools for query formulation.

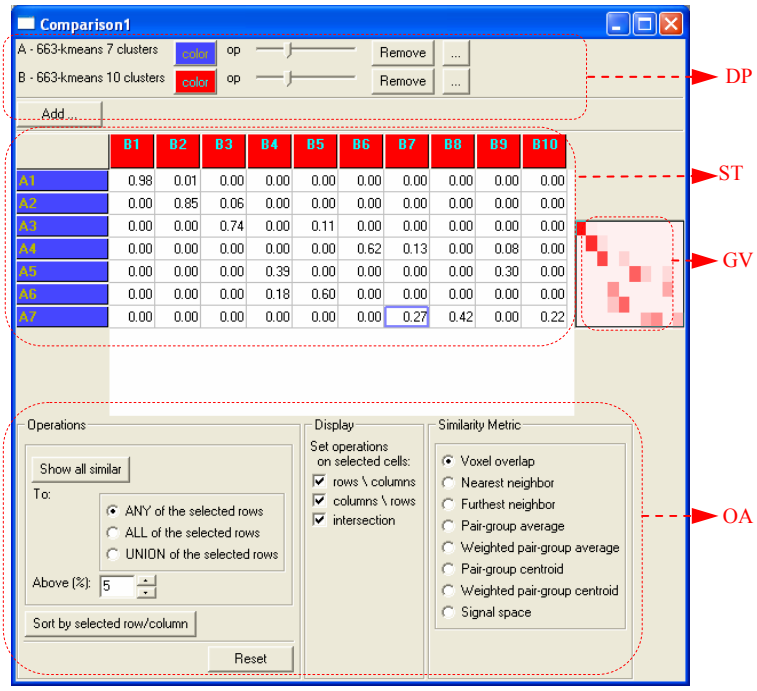


Figure 10 Data operations panel

The dataset properties area (DP) provides the means to load and remove datasets from the comparison table. It also provides quick access to the most commonly used properties of the datasets: color and opacity. Every dataset that is loaded into the comparison panel is assigned a color and an opacity that will be used by the rendering engine that manages the data visualization panel. Since two or more clusters can be shown in the visualization panel at the same time, different colors and opacities for clusters belonging to different datasets can facilitate a better understanding of their relationships. A dataset can be added to the comparison panel via the *Add…* button and removed via the corresponding *Remove*

button.

The central part of the data operations panel is the similarity table, based on a spreadsheet format, which displays similarity scores between pairs of clusters. The rows and columns of the similarity table represent the different clusters in the loaded datasets and each individual cell shows the similarity score of the clusters corresponding to that cell's row and column. In the simple case of two datasets A and B being compared, each cluster in dataset A is represented by a row in the spreadsheet and the columns represent the clusters in dataset B. In Figure 10 we show such a case where we compare two results of the k-means algorithm applied to the same dataset ("oddball" experiment subject 663) but with different parameters (the number of clusters: 7 and 10 respectively). The rows represent clusters in the 7-cluster result (dataset A) and the columns represent clusters in the 10-cluster result (dataset B). The color assigned to each dataset in the dataset properties area also colors the row/column headings.

Formally, if we consider each dataset to be a set of clusters, then the similarity table can be thought of as a mapping from the cartesian product of the two datasets into a set of real similarity scores between 0 and 1 according to some similarity metric:

$$s : A \times B \rightarrow [0,1]$$

where s is the similarity function and A and B are two datasets having n and m clusters respectively:

$$A = \{a_1, a_2 \ldots a_n\}, B = \{b_1, b_2 \ldots b_m\}.$$

An intuitive representation of this similarity function is in a table format, where the rows correspond to the clusters in dataset A and the columns represent the clusters in

dataset B. The cell at row i and column j in the table will show the value of s for the pair of clusters (a_i, b_j). We will refer to a similarity function defined as above as a pair-wise similarity function.

This formalism can be easily extended to accommodate any number of sets of clusters (datasets), so that s is defined on the Cartesian product of more than two sets. Given k sets of clusters (datasets) $A_1, A_2, \ldots A_k$, the similarity function s in this case is defined as:

$$s : A_1 \times A_2 \times \ldots \times A_k \rightarrow [0,1].$$

Analogous to the two sets case, such a function could be represented in a k-dimensional table but it becomes more difficult to visualize such a table in an efficient way as k increases. In the case of three sets one could imagine a three dimensional table containing a similarity score for each tuple of three clusters.

Since it is difficult to visualize a higher dimensional table in an intuitive manner, we (linearly) map the higher dimensions to the columns of our two-dimensional spreadsheet. Therefore if 3 or more datasets A, B, C, D, … are to be compared, the first one (A) will reside on the rows of the table and the second one (B) on the columns as in the two-sets case; the third set (C) and all the subsequent ones (D, …) will also be mapped to the columns of the table after the columns occupied by B, as shown in Figure 11. There is only one dataset mapped to the rows of the similarity table and this is the first dataset that is loaded into the comparison panel. One limitation of this approach is that in fact we can only display similarity scores between the dataset mapped to the rows and all the other

datasets mapped on columns, without any possibility of showing similarities between two datasets loaded on the columns of the table. Formally, if we denote the dataset loaded into the rows of the table by R (with clusters $R_1, R_2, \ldots R_m$) and the k datasets loaded into the columns as C_1 (with clusters $C_{1,1}, C_{1,2}, \ldots C_{1,n1}$), C_2 (with clusters $C_{2,1}, C_{2,2}, \ldots C_{2,n2}$), $\ldots$ C_k (with clusters $C_{k,1}, C_{k,2}, \ldots C_{k,nk}$) then our similarity table is now a collection of pair-wise similarity functions $s_1, s_2, \ldots s_k$ where each s_i gives the similarity between the clusters in R and the clusters in C_i, as shown in Figure 11.

$$s_i : R \times C_i \rightarrow [0,1], i = 1,...k$$

		C_1				C_2				...	C_k			
		$C_{1,1}$	$C_{1,2}$	...	$C_{1,n1}$	$C_{2,1}$	$C_{2,2}$	...	$C_{2,n2}$		$C_{k,1}$	$C_{k,2}$	...	$C_{k,nk}$
R	R_1	s_1				s_2				...	s_k			
	R_2													
	...													
	R_m													

Figure 11 Similarity table showing k pair-wise similarity functions.

The choice of mapping a single dataset to the rows of the table was based on the observation that our most common usage scenario for the tool involved comparing one analysis result against the result obtained using other methodologies or different parameters: a one-to-many comparison style. Additionally, the user can dynamically change which dataset is mapped to the rows, causing a complete update of the similarity table. An extension of our current approach could involve the possibility of mapping more than one dataset to the rows of the table, thus increasing the number of pair-wise similarity functions that can be hosted by it. For example, when comparing $k+1$ datasets,

if only one dataset is allowed to map to the rows, the table will display k pair-wise similarity functions as described above. However, if two datasets could be mapped on rows, the number will increase to $2*(k-1)$ since now we can display $k-1$ similarity functions for each row-mapped dataset, as shown in Figure 12.

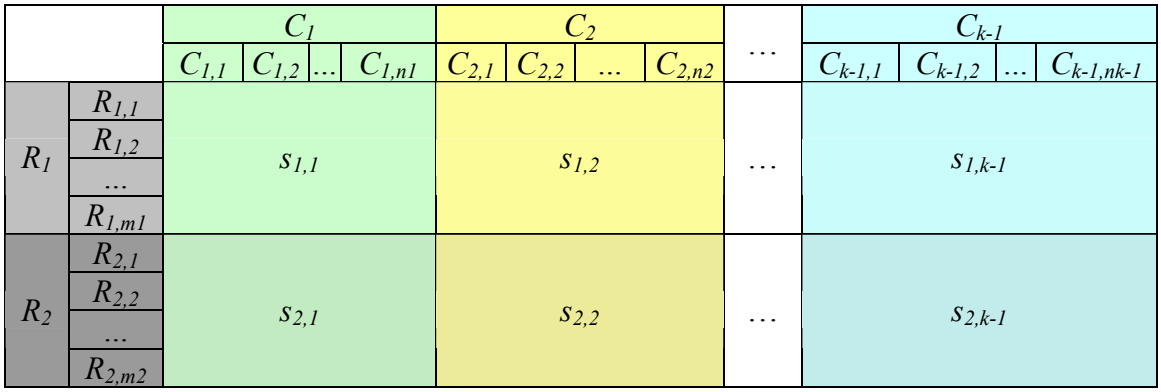


Figure 12 Extension of similarity table by mapping two datasets onto rows. The table can now show 2*(k-1) pair-wise similarity functions.

The number of columns in the similarity table can easily become quite large if the loaded datasets contain large numbers of clusters. It is not uncommon for a clustering algorithm working in n-dimensional signal space to return hundreds of clusters. Obviously such a large number of columns or rows cannot fit in one screen and locating large similarity values in the table can turn into a tedious task. We have provided a few tools to make this task easier. One of them is the table navigation widget (GV in Figure 10), which displays a color-coded image of the similarity table in a reduced-size bitmap, right next to the similarity table. The similarity scores showed in the cells of the similarity table are transformed into colored pixels on this icon, with high values mapping to intense red and small values mapping to light red. This bitmap provides a

global graphical view of the similarity scores present in the table and it also allows the user to navigate to a specific part of the table by simply clicking on the bitmap. A rectangle drawn on the bitmap indicates the part of the table that is currently visible in the spreadsheet.

The similarity scores displayed in the table can be computed using any of the cluster similarity metrics discussed in Chapter 5. The metric of choice can be selected from the right section of the data operations panel, right under the similarity table.

The third region of the data operations panel is the operations area, which provides the query tools (Figure 10). The queries described in Chapter 5 can be easily answered using the few controls here and selections of rows/columns in the similarity table. We will review the types of queries we want to support and then show how they can be formulated using our interface. We would like to be able to answer the following type of queries:

- Cluster-wise similarities:
    - **Q1**: Show all pairs of similar clusters in two or more datasets.
    - **Q2**: For any specific cluster in one of the datasets, show all similar clusters in all other datasets.
- Group-wise similarities
    - **Q3**: Given a set of clusters in one dataset, show all clusters in other datasets that are similar to ANY of the clusters in the selected set.
    - **Q4**: Given a set of clusters from one dataset, show all clusters in other datasets that are similar to ALL the clusters in the selected subset (similar to every one of the selected clusters).

- **Q5**: Given a set of clusters in one dataset, consider the cluster formed by the UNION of these clusters (union cluster) and show all clusters in other datasets that are similar to the union cluster.

The *Show all similar* button displays all similar pairs of clusters with one cluster from the rows and the other from the columns (Q1). The user can specify the minimum acceptable similarity score in the *Above* box. Rows and columns of the table that do not contain at least one significant similarity score will collapse to make room for the "interesting" ones (Figure 13). This is again useful when dealing with a large number of rows and/or columns.

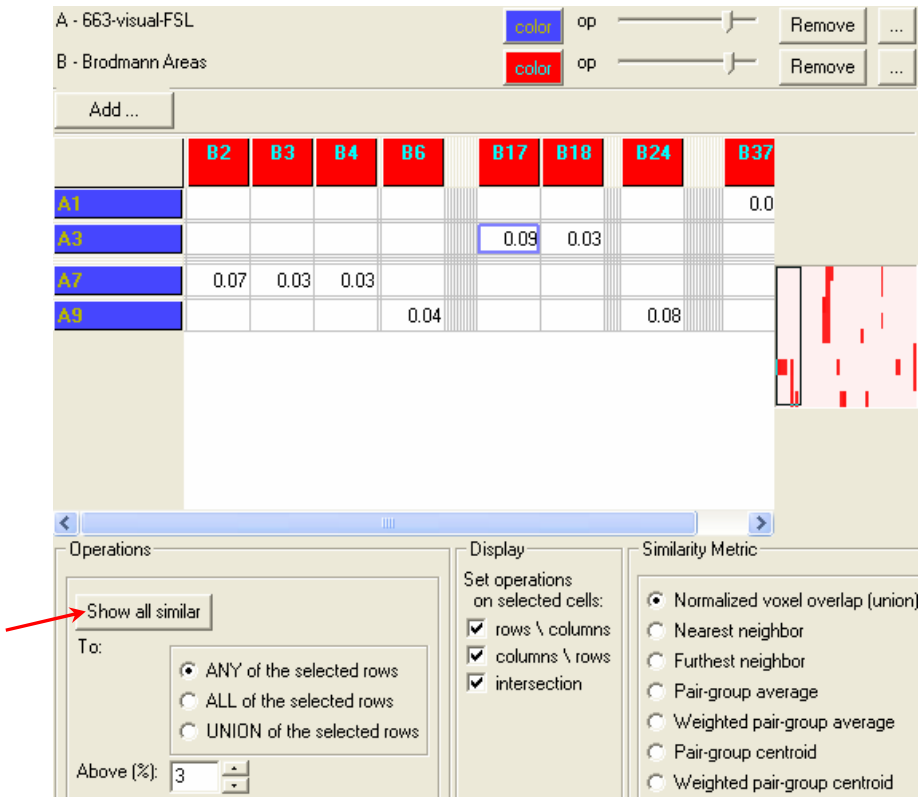


Figure 13 The *Show all similar* button displays only clusters that meet the similarity criteria.

A Q2 type query can be formulated by simply selecting one of the row clusters and then pressing the *Show all similar* button. This will show only the column clusters that are significantly similar to the selected row. The user can select more than one row cluster, choose how the selection should be interpreted and then press the same *Show all similar* button to display column clusters similar to the selection. A multiple row selection can be interpreted in one of three ways: ANY, ALL and UNION, corresponding to the last three queries Q3, Q4 and Q5. For a column cluster to be considered similar to the selection, it must be significantly similar to:

- at least one of the selected row clusters (the *ANY* option)
- every one of the selected clusters (the *ALL* option), or
- the union of the selected clusters (the *UNION* option)

Another helpful tool in a spreadsheet is sorting (the *Sort by selected row/column* button). The user can sort the similarity table by any row or column, making it easier to locate similar clusters when the number of clusters is large.

6.1.1 Loading a dataset

When loading a new dataset into a comparison table, the user is presented with a number of options. Our tool allows one to directly load a un-thresholded statistical map produced by analysis tools like SPM. These datasets contain a floating point value at each voxel in the dataset, both inside and outside the brain. The user is allowed to choose a threshold while interactively viewing the supra-threshold voxels (Figure 14). This functionality is desirable especially because the choice of a suitable threshold is not a simple matter and is usually done in a blind fashion at run time in the analysis tool. While a high enough

threshold guarantees a high confidence that the selected voxels are really active, it also selects only a few usually spatially disconnected voxels. On the other hand, a lower threshold will include more voxels, maybe more spatially coherent but there is a bigger chance that some of them are not really active. We believe this choice should be left to the user, as conditions vary from experiment to experiment.

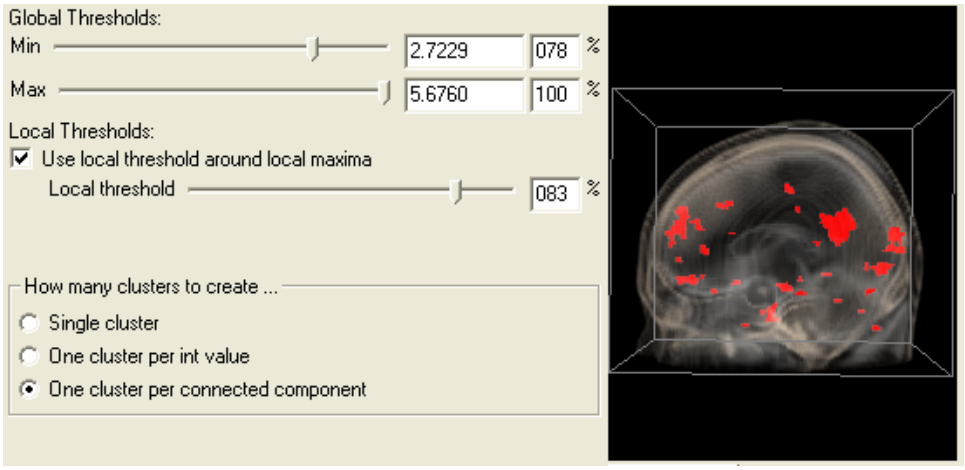


Figure 14 Loading a dataset.

Apart from the global thresholding described above, one can also use a local threshold to increase the size of the individual spatial clusters (Figure 14). This local threshold is effective in the vicinity of the voxels selected by the global threshold and allows voxels that are under the global threshold but within a certain percentage of the value of the local maximum to be included in the cluster. The effect of this local threshold is to increase the number of voxels in each spatial cluster.

After thresholding the voxels, the user can choose a clustering method for the supra-threshold voxels. The choices are (Figure 14):

- *One cluster* – all selected voxels will form a single functional cluster.

- *One cluster per connected component* – each connected set of voxels will be treated as a separate cluster (26-connectivity is used).
- *One cluster per int value* – the actual value of the voxel in the dataset determines which cluster it belongs to. Each different value will be treated as a different cluster. (Note that when loading a probability map containing floating point values, the application will convert them to integers by truncating all the non-zero values and adding one).

6.2 The Visualization Panel

The visualization panel graphically displays the selections made in the similarity table, overlaid on a volume-rendered image of the brain anatomy. This image provides visual information about the location of different clusters in the brain space. The three buttons on the panel are shortcuts to axis-aligned visualizations of the data: *Axial* views the data from above the head, *Sagittal* views the data from the left side of the head and *Coronal* shows the data as viewed from the front of the head.

The graphical display in the visualization panel is linked to the similarity table. As the user browses through the similarity table, the two clusters corresponding to the rows and columns of the current selection of spreadsheet cells are rendered together with the anatomical image in the same space. Each cluster is rendered using the color and opacity selected for its dataset in the dataset properties region of the data operations panel.

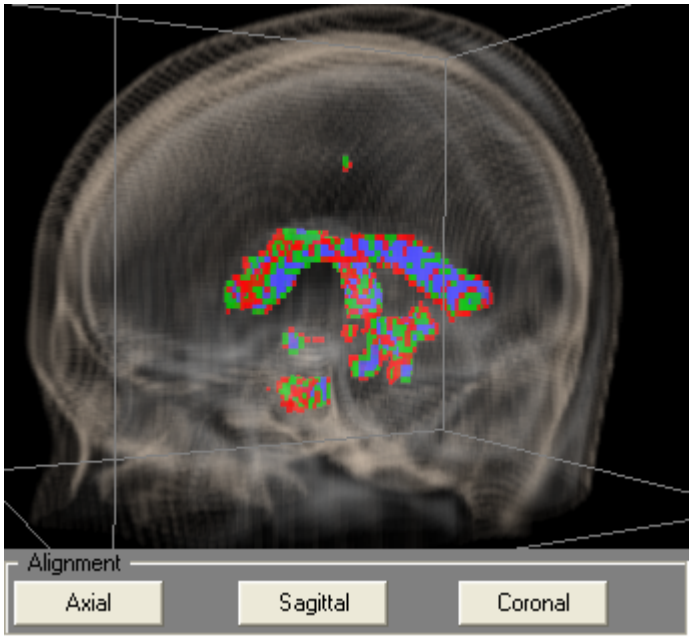


Figure 15 The visualization Panel.

When looking for similarity between two clusters, their geometric intersection (common voxels) can provide valuable information (as in the Venn diagram) and it is rendered in a different color. In Figure 9(b) (enlarged in Figure 15) the visualization panel shows cluster 7 from dataset A in blue, cluster 7 from dataset B in red and their common voxels (intersection) in light green.

A few controls in the middle part of the data operations section of the data operations panel, labeled *Display* control the rendering of the selected clusters. As a set of clusters is selected by selecting a cell in the spreadsheet, the voxels in the selected clusters form three distinct classes:

- voxels that belong only to the row cluster (*row* \ *column*),
- voxels that belong only to the column cluster (*column* \ *row*) and
- voxels that belong to both the row and the column cluster (*intersection*).

The user can choose to display one, two or all of these classes by checking the corresponding options. For example, by checking the *rows \ columns* option, and un-checking the other two, the visualization panel will show those voxels that belong to the row cluster and do not also belong to the column cluster. This is equivalent to a set difference operation on the two sets of voxels (Figure 16).

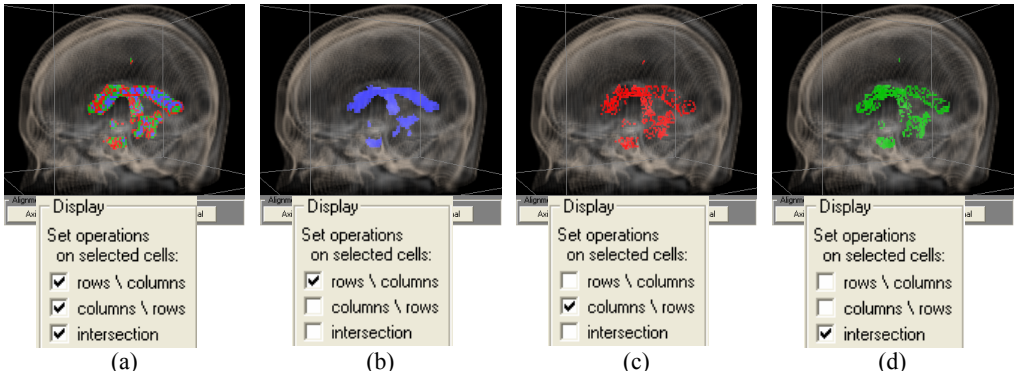


Figure 16 Set operations on two clusters. Showing all voxels in both clusters (a), voxels belonging only to the row cluster (b), voxels belonging only to the column cluster (c) or only the common voxels (d).

The tool is implemented in C++ with most of the development process done under MS Windows. However, for the user interface we used the wxWidgets library [41] and the Visualization Toolkit (VTK) [40], cross-platform libraries that can be compiled on other platforms like Linux and Mac without any changes in the code.

Over fifteen classes were developed during the implementation. Figure 17 illustrates the collaboration between the most important components of the system.

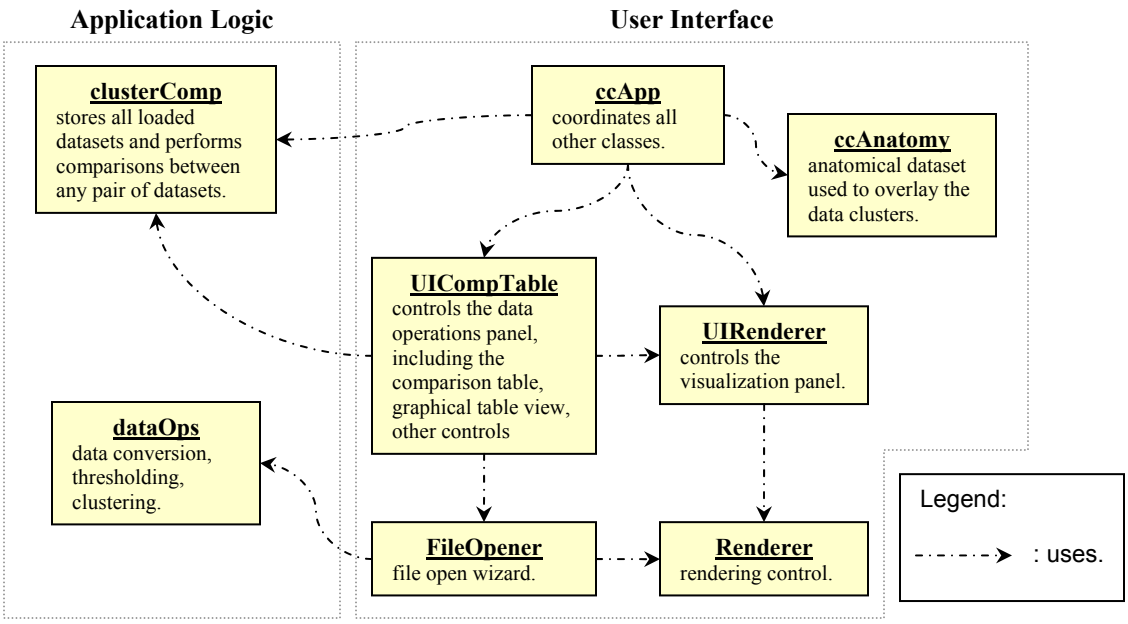


Figure 17 Collaboration diagram for the most important components of our implementation.

The *ccApp* class coordinates the activity of all other classes, representing the high-level logic of the interactive application. At application startup, it constructs instances of *UICompTable* and *UIRenderer* and introduces them to each other so that further interaction is done directly, without the mediation of *ccApp*. *UICompTable* encapsulates the functionality in the data operations panel, including the spread-sheet based comparison table, the table navigation widget and the rest of the user interface controls in the panel. The *UIRenderer* component implements the visualization panel. It included an instance of *Renderer* which is the actual rendering control and several push buttons. The *FileOpener* component is a wizard-like interface for loading a dataset into our tool, accessed through the data operations panel. The interface allows file selection, various threshold setting and clustering type selection. Data conversion from various types to the internal representation, thresholding and clustering are done via the *dataOps* component.

ccAnatomy is a special component that manages the underlying anatomical image on top of which the data clusters are shown. The heart of the application is the *clusterComp* class which manages all datasets loaded at a given time and performs comparisons between any pair of datasets; the result of each comparison is a similarity table.

We tried to maintain a separation between user interface components and application logic as seen in Figure 17. This separation allowed us to create a non-interactive command-line version of the tool that can be used in batch processing of a large number of datasets. This command-line version was developed under Linux.

Chapter 7

Examples and Discussion

We demonstrate our tool by describing a number of usage scenarios and examples. Voxel overlap was used as similarity metric in all of the following examples. First, we will review the types of scenarios where our tool may be useful:

- For clinicians
    - Identifying similarities and differences between different analysis methods.
    - Explore similarities and/or differences between different subjects performing the same experiment.
    - Identify similarities and/or differences between brain regions activated in different experiments.
    - Assessing patient progress during recovery by examining successive scans acquired during the recovery process.
- For algorithm developers
    - Establishing similarities and differences between their methodology and well-established methods.
- For users of fMRI database search engine

- Perform the actual data retrieval based on one of the available similarity measures.
- Inspect the degree of similarity between the retrieved datasets the query dataset.

Our database of fMRI datasets consists of 4 “oddball” experiments, over 20 “event perception” experiments [45] and 9 “recall” experiments [29][30]. Every experiment includes several conditions and runs, so the number of datasets exceeds 500. In the “oddball” experiment, the subject is presented with a sequence of images and is asked to react by pressing a button when an “oddball” image is displayed. The “oddball” image was the image of a face while the “non-oddball” images may be geometrical figures. In the “event perception” experiment, the subject is watching a short video sequence and is asked to press a button when he/she sees an event happening. The definition of an event is left to the subject. Two video sequences were used for this experiment: one is a movie of a student preparing for study in a library, also known as the “study” video; the other is a cartoon of a small circle moving around a set of obstacles that represent the layout of a room in a house, also known as the “house” video. In the “recall” experiment, the subjects studied a set of famous faces, locations or objects (the study condition). Later the subject was asked to verbally recall the studied items (the “recall” condition).

Several analysis methods were used SPM and FSL from the hypothesis-driven category and three clustering methods from the data-driven category: k-means, mean shift and canonical sets.

The results of the hypothesis-driven methods were converted to a cluster format by thresholding the resulting statistical map and spatially clustering the resulting voxels.

From the usage scenarios described at the beginning of the chapter, we identify several types of comparisons that would be useful:

- Comparing the results of different analysis methods applied to the same data – cross-method comparison.
- Comparing results of one analysis methodology applied to different data – cross-subject comparison.
- Mapping activation regions to a standardized brain atlas.
- Comparing two unrelated datasets to explore the similarities and differences between their activated regions.

In the remaining of this chapter we will present examples for each type of comparison and highlight the kind of insight one could get from such a comparison.

7.1 Comparing results different analysis methods on the same data

This particular kind of comparison might be used by scientists in order to analyze the results of different analysis methods and make inferences about the similarity or differences in the results. Alternatively, algorithm developers might be interested in this type of comparison in order to compare their results with those obtained using one of the standard packages. While it is natural to see differences in the results obtained with different algorithms, one would expect a certain degree of similarity since all the methods try to extract the same “underlying truth” from the same data.

We exemplify this approach by comparing the results of four methods applied to the same data: the "oddball" task, performed by subject 663. All the result datasets are loaded into a single comparison table. The first loaded dataset defines the rows of the spreadsheet; all others map to columns. The fastest way to identify similar clusters is to set the similarity threshold (the *Above* box in the Operations panel) to the desired minimum acceptable similarity, and then press the Show all similar button. Figure 18 illustrates this case. The geometric intersection of the two clusters corresponding to the selected spreadsheet cell is rendered in light green in the rendering window. The similarity table shows clusters in the row dataset (results of the canonical sets algorithm) that overlap with clusters obtained using other algorithms (k-means (B), mean-shift (C) and SPM (D)).

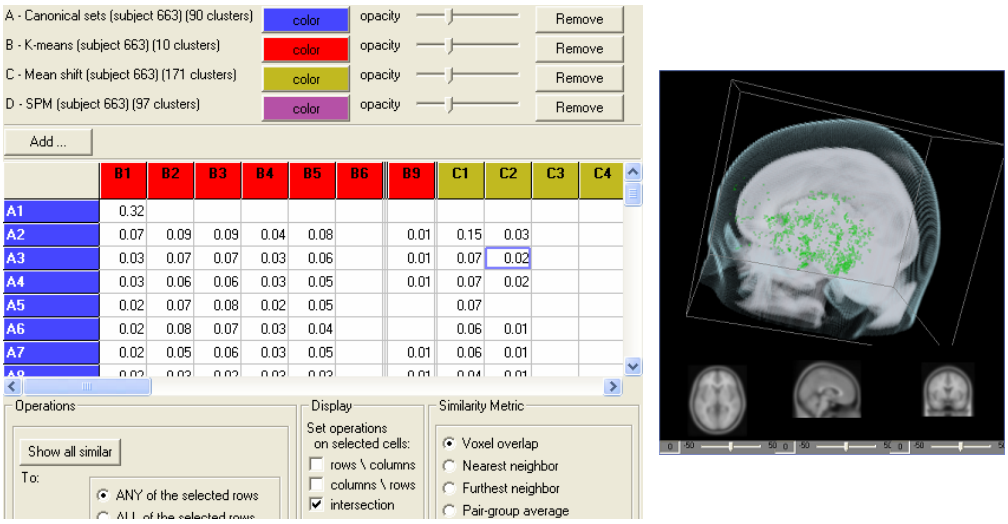


Figure 18 Comparing the results of four different analysis methods applied to the same data.

One of the common problems in using all these analysis tools is choosing the right parameters. The ideal results should show reasonably large clusters in the expected areas

of activation. However, choosing the right parameters is not a trivial task, especially when the inner mechanisms of the analysis algorithm are not well understood by the user. In such cases, one could simply make a few choices for the necessary parameters, run the analysis tool and then compare the results. In the following example we applied the k-means clustering algorithm to "oddball" subject 663. Clustering was done in signal space. We have a choice for the number of clusters we want to obtain. On one hand we want to work with as few clusters as possible in order to be able to quickly analyze each one of them. On the other hand, too few clusters may force the algorithm to merge groups of voxels that are different, creating less interesting clusters. In Figure 19 we show a comparison of the results obtained using 7 (the A set) and 10 (the B set) as the parameter value (number of clusters). We want to identify clusters located inside the brain which are also spatially coherent.

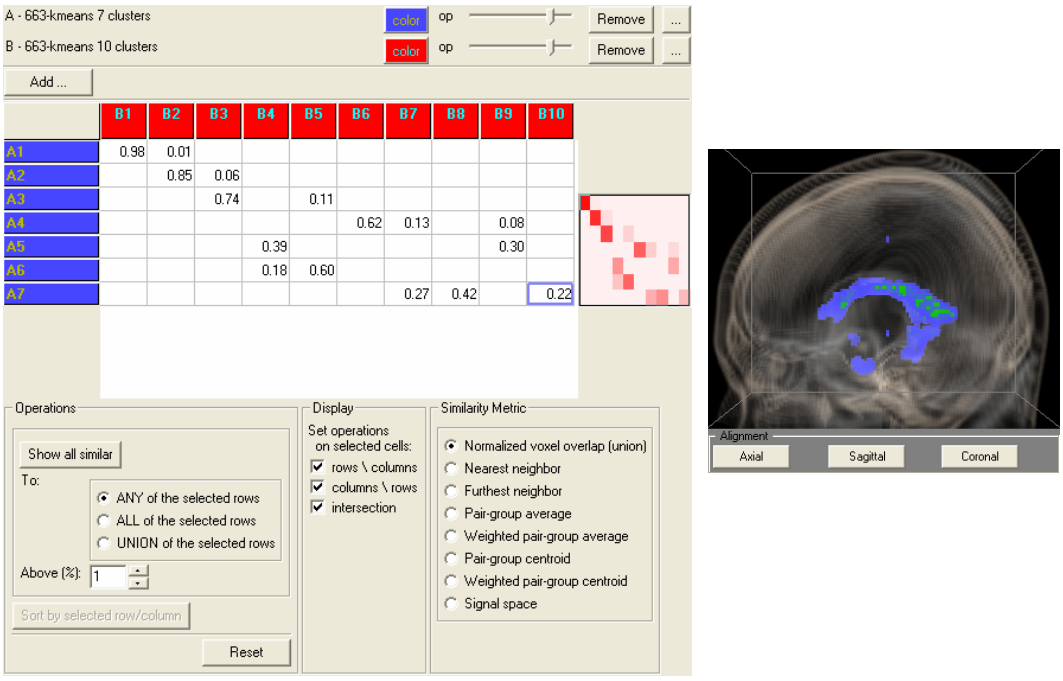


Figure 19 Comparing the results of applying k-means clustering with different parameters on the same dataset.

For this particular example, we can see that cluster B10 is completely included in A7 since in the visualization panel we can only see two colors: blue corresponding to voxels that are part of A7 only and green which are voxels that are included in both A7 and B10 (red would indicate voxels in B10 which are not in A7). This suggests that cluster B10 was obtained by breaking cluster A7 into several smaller pieces. The exact number of pieces that A7 was broken into can also be seen quickly by inspecting the row corresponding to A7 where we see only 3 non-zero values. Thus, when changing the parameter value from 7 to 10, the 7^{th} cluster was broken into 3 pieces that were assigned to different clusters in the 10 clusters result.

7.2 Comparing results of same analysis method on different subjects (cross-subject comparison)

In the same way we could compare the results of the same analysis method applied to different subjects of the same experiment. Pairs of clusters with high similarity score will represent common activation regions across subjects.

In Figure 20 we show a comparison between results of the same analysis methodology (using FSL) applied to several subjects in the "event perception" experiment. We filtered the similarity table to show only similarity scores above 4% and we can immediately see that activation cluster A8 from subject 16 is similar to one cluster from each of the other subjects (B2 from subject 17, C40 from subject 19 and D28 from subject 20), indicating a common activation area present in all subjects. Cluster A8 from subject 16 and its intersection with cluster D28 from subject 20 are shown in the

visualization panel in Figure 20.

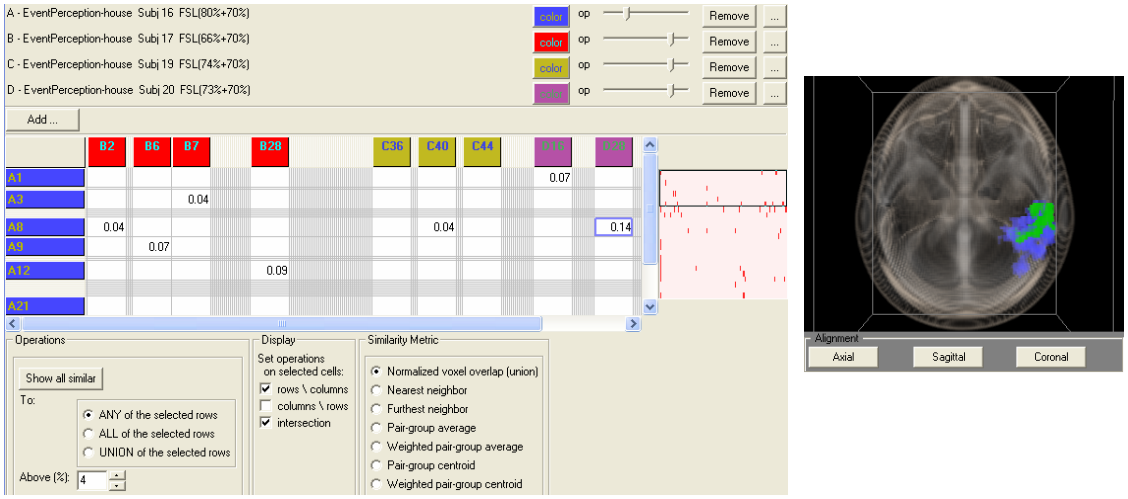


Figure 20 Comparing the results of one analysis methodology (FLS) applied to several different subjects of the same "event perception" experiment. Cluster A8 and its intersection with D28 are shown in the visualization panel.

7.3 Mapping analysis results to a brain atlas

In a different scenario, we want to map the results of some analysis method (k-means clustering for example) to the well-known (to brain scientists) Brodmann regions or some other standard brain atlas. Brodmann regions represent a classification of brain voxels based on anatomical tissue properties, and each region is believed to be responsible for a number of functions. The Brodmann map can be regarded as another clustering of the brain-space voxels and it is treated in the same way as any other clustering. This task is commonly performed after the analysis step, to map the active voxels to the Brodmann areas.

Brain-specific questions can also be posed about a specific labeling such as the

following:

- Select all clusters that overlap with any Brodmann region.
- Given an interesting Brodmann region, select all clusters that overlap with it.
- Given a few Brodmann regions that the researcher considers relevant for the given experiment, select all clusters that overlap with any or all of them.

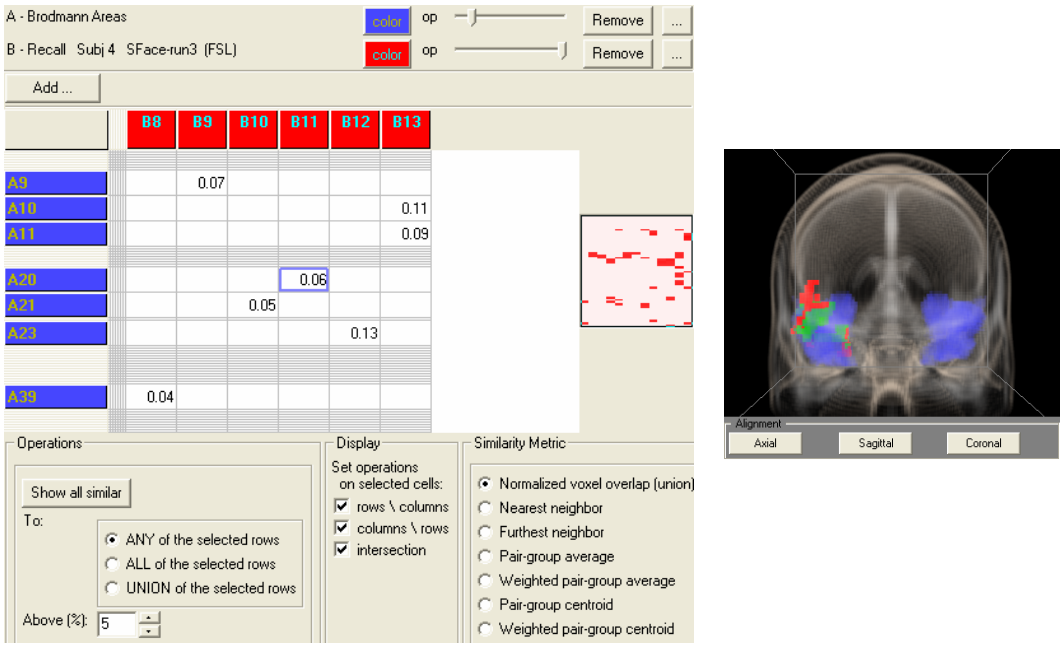


Figure 21 Brodmann region mapping. Brodmann areas shown on the rows in blue and the clusters of subject 4 of the "recall" experiment are shown in red on the columns.

In Figure 21 we show a mapping of the results of FSL analysis applied to "recall" subject 4 ("study face" condition, run number 3) to the Brodmann regions. We load the 47 Brodmann clusters on the rows of the comparison table (in blue); FSL results are mapped to the columns. The similarity table shows how each activation cluster of subject 4 overlaps with every one of the Brodmann regions. Based on the similarity table, we can

say that the subject shows activation in Brodmann regions 9, 10, 11, 20, 21, 23 and 39.

7.4 Query-by-example Data Retrieval

One of the valuable usage scenarios of our tool is in a data retrieval environment. Here the user of an fMRI database would like to retrieve datasets which present similarities with a given query dataset, presented by the user: a query-by-example approach to fMRI database searching. Suppose we have acquired a new dataset ("recall" subject 7, study face condition for example). We analyze this dataset using any method at hand and then we would like to find a similar dataset by querying a repository of analyzed fMRI data. We will use overlap as our similarity requirement. Figure 22 shows a comparison between our query dataset (set A) and eight datasets from our fMRI database. These are: three "study face" conditions: two for the same subject as our query dataset (subject 7), but different runs: run 2 (set D) and run 3 (set I), while the third one is for subject 4 (set F); one "study object" condition (set B); two "study location" conditions (sets C and G); one "event perception" experiment (set E) and one "oddball" experiment (set H). All datasets were analyzed using FSL. The "study face", "study location" and "study object" conditions are from the same experiment: "recall". "oddball" and "event perception" are different experiments. The similarity table displays pairs of clusters with a similarity score above 6%.

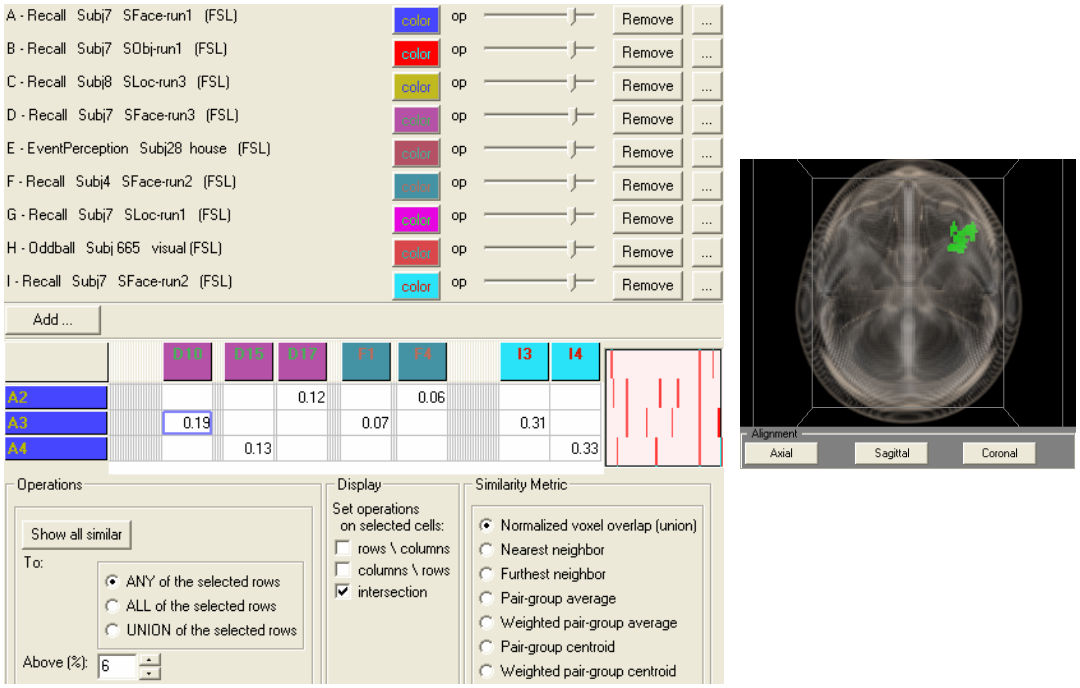


Figure 22 Query-by-example data retrieval.

The collapsed similarity table shows that our query dataset is more similar to the other study face conditions (sets D, F and I). The visualization panel shows one of the common regions activated in sets A and D (same subject: 7, same condition: study face, different runs). Also, note that the similarity scores between rows and the columns labeled F (F1 and F4) are lower than those corresponding to the other columns that were not collapsed. Set F corresponds to the same condition, study face, but performed by a different subject (subject 4). Thus, in this example, we can distinguish between datasets corresponding to different conditions, and among those, we can differentiate between different subjects, all based on the scores presented in the similarity table.

The example showed in Figure 22 and discussed above uses a very small number of datasets. This is because the interactive version of our tool requires the user to manually load each of the datasets involved in the comparison. When large databases need to be

searched for similar datasets, a scripting capability would be desirable.

7.5 Investigating similarity reported by other methods – visualization of data mining results

In addition to its use as a retrieval tool, our application provides valuable insight into the data retrieved using other systems. Let us assume that a database of fMRI studies has its own query-by-example search engine. When the user presents a query dataset, the system will identify a number of similar datasets based on some internal similarity concept. More generally, assume the database is equipped with a data mining engine which can identify similarities in the catalogued datasets and report them to the user. Again, the similarity concept is a built-in internal function of the data mining algorithm and the user may have little or no control over specific parameters. However, the end user may wish to subjectively assess the degree of similarity between the reported datasets, focusing on the aspects he/she believes to be relevant for the given case. In this usage scenario, our tool performs a function similar to that of a web browser that allows the user to click and investigate each of the results of a web query and asses their relevance.

Our similarity computation experiments included a similarity measure based on overlap with Brodmann areas discussed previously. Every analysis result was converted into a *Brodmann vector*, a vector with b components, where b is the number of Brodmann regions. Each vector component represents the percent of the corresponding Brodmann area covered by the given dataset. The similarity measure between two

datasets was computed as the cosine of the angle between the two corresponding *Brodmann vectors*. One of the high-ranking pairs among the datasets in the "recall" experiment was subject 4, study location condition (run 3) and subject 8, study location condition (run3). The Brodmann vector similarity score for this pair was 0.89, where 1.00 indicates perfect similarity. Let us assume this is the result of a data mining operation on a database of fMRI data. To further investigate this pair, using overlap between the two datasets as our similarity criterion, we loaded the two datasets into our interactive comparison tool. The similarity table in Figure 23 shows all cluster pairs with overlap similarity score above 1%.

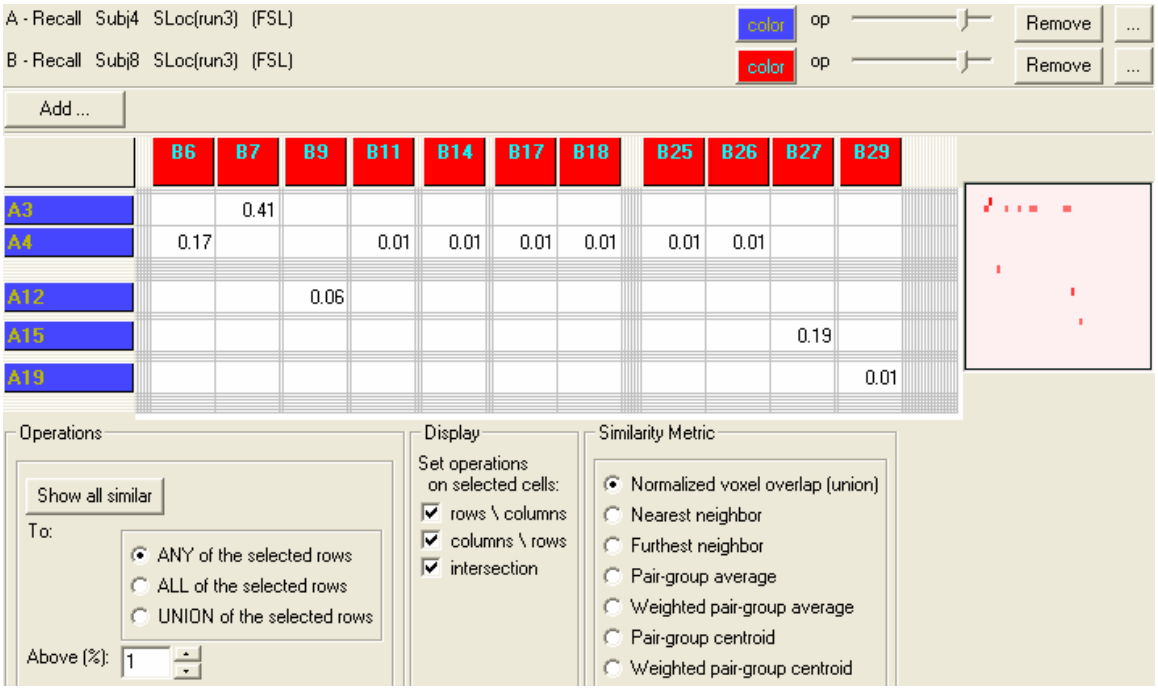


Figure 23 Investigating high similarity reported by other methods.

Dataset A has 25 connected clusters, while dataset B consists of 46 connected clusters. A quick inspection of the similarity table presented in Figure 23 reveals that a

large number of clusters from the first dataset (A1, A2, A5 to A11, A13, A14, A16 to A18 and A20 to A25) do not overlap with any of the clusters in the second dataset. We can see that only 11 pairs of clusters out of the total of 1150 (25 x 46) show any similarity at all; this constitutes less than 1% of the total number of pairs. In fact, the total overlap similarity of the two datasets is only 17% (this score was obtained by loading each of the two datasets as a single functional cluster – not shown). This demonstrates that the high *Brodmann vector* similarity score is only partially due to actual voxel overlap between the two datasets. The rest could be accounted for by the different clusters that do not overlap with each other but are in the same Brodmann areas. If overlap was the required similarity, then we would reject this first result of the data mining operation.

The above example illustrates the case where overlap similarity between two datasets is low while another similarity measure could produce a high score. The opposite case (high overlap similarity and small "other similarity" score) is less likely simply because a high overlap similarity implies that the two datasets are virtually identical not only with respect to shape but also with respect to position and orientation. Thus any reasonable similarity metric would recognize them as similar and assign a high score to the pair.

Chapter 8

Conclusions and Future Work

The tool presented here can provide useful information to the brain studies research community in several ways:

- It enables one to visually compare the results and assess the similarity of different analysis methods that are available and, hopefully, identify the best method to use for the study at hand.
- It allows developers to compare the results of their new analysis algorithm with other standard analysis techniques already available.
- It can serve as a query-by-example data retrieval tool on a small database.
- As the visualization interface of a database engine, the tool can be used to browse through the results of a data-mining operation or a query-by-example search and decide whether the required similarity is present in the retrieved datasets; much as a browser serves this purpose in the case of a web search.
- It can provide valuable insight into the data being examined in an intuitive and graphical manner as well as quantitative measurements of similarity.

8.1 Observation of Real Users

The current interface was designed to help us understand the type of queries that are applicable to this specific problem. As part of the development of this tool, we have begun to investigate the usability of the prototype by asking several experienced researchers in the fMRI field to work with the tool. The investigation included a short introduction to the tool and its basic interface features and a guided test, where the researchers were asked to perform some basic tasks like comparing several datasets and mapping them to Brodmann areas.

Several conclusions were drawn from this case study. One conclusion was that setting the threshold as an absolute value of the t statistic when loading a t-map is not meaningful, and the expert users would like to be able to set the threshold based on probability (the p-value which can be derived from the t distribution). Regarding the comparison paradigm, the current version of the tool only allows one-to-many comparisons (one dataset is compared to several other datasets). The user evaluation revealed that many-to-many comparisons would be desirable in cases where more than two datasets need to be compared.

To summarize:

- We have built a new real-time visualization tool for comparing analyzed fMRI datasets which can provide valuable insight into the studied data or analysis methodology.
- We have provided two linked views of the similarity between two datasets: a quantitative view based on a spreadsheet paradigm and a qualitative view

provided by the visualization component.

- We have demonstrated how the tool can be used for various tasks such as: comparing the results of different methodologies, comparing activations of different subjects, mapping activation regions to brain atlases, query-by-example data retrieval and exploration of data mining results.

8.2 Future Work

Future developments will include the recommendations resulting from the user study: setting thresholds based on probability rather than absolute t values and implementing the many-to-many comparison.

When a large number of datasets needs to be compared, the time required for the user to load all the datasets could be prohibitive. In such cases, adding scripting capabilities to the application could prove very useful.

In this design, we only implemented one of the similarity computations (voxel overlap). We plan to finish implementing the other metrics described here and possibly implement new ones, such as a metric based on the earth-movers distance. In addition, we will investigate the appropriate visualization methodologies for these new similarity metrics.

References

[1] M. Brett, I. Johnsrude, A. Owen. The problem of functional localization in the human brain. Neuroscience, 3:243-249, 2002.

[2] Cluster Analysis. Online at: http://www.statsoft.com/textbook/stcluan.html.

[3] M.S. Cohen. Parametric Analysis of fMRI Data Using Linear Systems Methods. Neuroimage 6, 93–103, 1997.

[4] R.W. Cox. AFNI: Software for analysis and visualization of functional magnetic resonance neuroimages. Computers and Biomedical Research, 29:162-173, 1996.

[5] I. Davidson. "Visualizing Clustering Results", SIAM International Conference on Data Mining, 2002

[6] T. Denton, J. Abrahamson, A. Shokoufandeh. Approximation of Canonical Sets and Their Applications to 2D View Simplification. IEEE Computer Society Conference on Computer Vision and Pattern Recognition – CVPR 2004. Washington, DC, June 27th - July 2nd, 2004.

[7] T. Denton, M.F. Demirci, J. Abrahamson, A. Shokoufandeh, S. Dickinson. Bounded Canonical Sets and their Applications to View Indexing. 17th IAPR International Conference on Pattern Recognition (ICPR 2004). Cambridge, United Kingdom, August 23-26, 2004

[8] E. Dimitriadou, M. Barth, C. Windischberger, K. Hornik, E. Moser. A Quantitative Comparison of functional MRI Cluster Analysis, Preprint submitted to Elsevier Science, July 2003

[9] J. Ford, H. Farid, F. Makedon, L.A. Flashman, T.W. McAllister, V. Megalooikonomou, and A.J. Saykin. Patient Classification of fMRI Activation Maps, 6th Annual International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI'03), 2003.

[10] J. Ford, H. Farid, L. Flashman, A. Saykin. Patient Classification of fMRI Activation Maps. Medical Image Computing and Computer-Assisted Intervention, 2003.

[11] S.D. Forman, J.D. Cohen, M. Fitzgerald, W.F. Eddy, M.A. Mintun, D.C. Noll. Improved assessment of significant activation in functional magnetic resonance imaging (fMRI): Use of a cluster-size threshold. Magnetic Resonance in Medicine, 33:636–647, 1995

[12] L. Forsberg. Visualization and Query Tool for a Human Brain Database, Master's Thesis, Royal Institute of Technology (KTH), Numerical Analysis and Computer Science Department, TRITA-NA-E04049.

[13] P. Fox, J. Lancaster. Neuroscience on the net. Science, 266:994-996, 1994.

[14] FMRIB Software Library: http://www.fmrib.ox.ac.uk/fsl

[15] O. Friman, M. Borga, P. Lundberg, H. Knutsson. Exploratory fMRI Analysis by Autocorrelation Maximization, NeuroImage 16, p. 454–464, 2002.

[16] K.J. Friston, A.P. Holmes, K.J. Worsley, J.P. Poline, C.D. Frith, R.S.J. Frackowiak. Statistical Parametric Maps in Functional Imaging: A General Linear Approach (1995) Human Brain Mapping 2:189-210

[17] K.J. Friston, K.J. Worsley, R.S.J. Frackowiak, J.C. Mazziotta, A.C. Evans. Assessing the significance of focal activations using their spatial extent. Human Brain Mapping, 1:213–220, 1994.

[18] C. Goutte, L.K. Hansen, M.G. Liptrot, E. Rostrup. Feature-space clustering for fMRI meta-analysis, Hum Brain Mapping, 13(3):165-83, 2001.

[19] C. Goutte, P. Toft, E. Rostrup, F. Nielsen, K. Hansen. On clustering fMRI time series. Neuroimaging, 9(3):298-310, 1999.

[20] M. Halkidi, Y. Batistakis, M. Vazirgiannis. Cluster Validity Methods: Part II. SIGMOD Record, 2002.

[21] M. Halkidi, Y. Batistakis, M. Vazirgiannis. Cluster Validity Methods: Part I. SIGMOD Record, 2002.

[22] L.K. Hansen, F.Å. Nielsen, P. Toft, M.G. Liptrot, C. Goutte, S.C. Strother, N. Lange, A. Gade, D.A. Rottenberg and O.B. Paulson . "lyngby" - a modeler's Matlab toolbox for spatio-temporal analysis of functional neuroimages, NeuroImage, 9(6):S241, June 1999.

[23] Introduction to fMRI. Online at: http://www.fmrib.ox.ac.uk/fmri_intro.

[24] A. Lazarevic, D. Pokrajac, V. Megalooikonomou and Z. Obradovic. Distinguishing Among 3-D Distributions for Brain Image Data Classification, In Proceedings of the 4th International Conference on Neural Networks and Expert Systems in Medicine and Healthcare, pp. 389-396, 2001.

[25] S. Letovsky, S. Whitehead, C. Paik, G. Miller, J. Gerber, E. Herskovits, T. Fulton, R. Bryan. A brain image database for structure/function analysis. American Journal of Neuroradiology, 19:1869-1877, 1998.

[26] Lyngby Toolbox web site: http://hendrix.imm.dtu.dk/software/lyngby/lyngby.html

[27] V. Megalooikonomou, J. Ford, L. Shen, F. Makedon. Data mining in brain imaging. Statistical Methods in Medical Research, 9:359-394, 2000.

[28] NeuroGenerator – a database project for the neuroimaging community. Online at: http://www.neurogenerator.org.

[29] S.M. Polyn, J.D. Cohen, K.A. Norman. Detecting distributed patterns in an fMRI study of free recall. Society for Neuroscience conference, San Diego, CA, 2004.

[30] S.M. Polyn. Neuroimaging, behavioral, and computational investigations of memory targeting. Doctoral dissertation, Princeton University, Princeton, NJ, USA, 2005.

[31] RUMBA: http://psychology.rutgers.edu/RUMBA

[32] P.H.A. Sneath, R.R. Sokal, Numerical Taxonomy. Freeman, San Francisco, CA, 1973.

[33] SPM web site: http://www.fil.ion.ucl.ac.uk/spm/

[34] J. Talairach, P. Tournoux. Co-planar Stereotaxic Atlas of the Human Brain. Thieme Medical, New York, 1988.

[35] B. Thirion. fMRI data analysis: statistics, information and dynamics. Ph.D. dissertation, INRIA Sophia-Antipolis, 2003.

[36] F.Y. Tzeng, K.L. Ma. A Cluster-Space Visual Interface for Arbitrary Dimensional Classification of Volume Data, In Eurographics - IEEE TCVG Symposium on Visualization, 2004

[37] J. Van Horn, M. Gazzaniga. Databasing fMRI studies – towards a ‘discovery science’ of brain function. Nature, 3:314-318, 2002.

[38] M.O. Ward. XmdvTool: Integrating multiple methods for visualizing multivariate data, Proc. of Visualization' 94, pp. 326-333 (Octomber 1994).

[39] VoxBo: http://www.voxbo.org/index.html

[40] VTK web site: http://www.vtk.org/

[41] wxWidgets home page: http://www.wxwidgets.org/

[42] J. Xiong, J.-H. Gao, J.L. Lancaster, P.T. Fox. Assessment and optimization of functional MRI analyses. Human Brain Mapping, 4:153–167, 1996.

[43] XmdvTool: http://davis.wpi.edu/~xmdv/index.html

[44] J. Yang, M.O. Ward, E.A. Rundensteiner, S. Huang. Visual Hierarchical Dimension Reduction for Exploration of High Dimensional Datasets, VisSym 2003.

[45] A. Zaimi, C. Hanson, S.J. Hanson. Event Perception of Schema-Rich and Schema-Poor Video Sequences During fMRI Scanning: Top Down Versus Bottom Up Processing. In Proceedings of the Annual Meeting of the Cognitive Neuroscience Society, San Francisco, CA, 2004.

Appendix 1

User Manual

Cluster Comparison and Visualization Tool (CCVT)

User Manual (v2.1.2)

(software version: 0.35)

by

Narges Kasiri and Nicu Cornea

1. Starting CCVT

To start the program, double click on CCVT icon. 

Three main windows will pop up (Figure 1):

- The *Comparison Panel* displays quantitative information about the datasets being compared
- The *Visualization Panel* provides a visual display of the selected clusters in a common space overlaid onto a rendering of the anatomy of the brain.
- *Loading Dataset window* to load some datasets into CCVT.

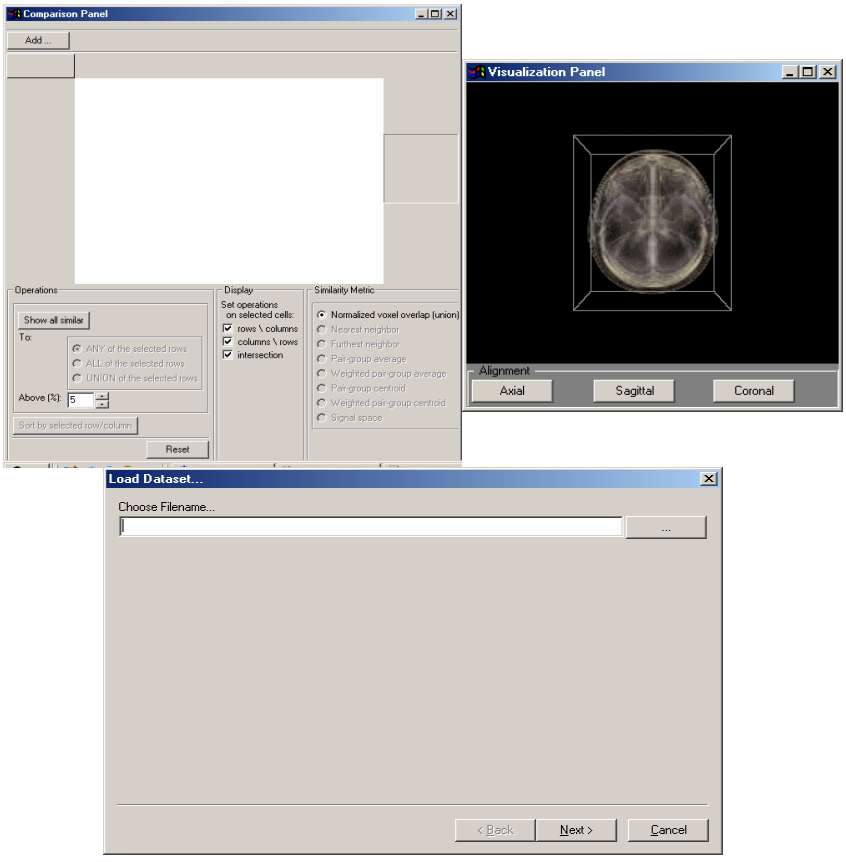


Figure 1. CCVT user interface (comparison panel, visualization panel, loading dataset).

2. Loading a dataset

In the Load Dataset window (Figure 2), click on '…' to open a browsing window (Figure 3). You can always access the Load Dataset window from the Comparison Panel by clicking on the '*Add*...' button.

You must load at least two datasets to be able to do a comparison. The only supported file format is the ANALYZE format (a pair of .hdr and .img files).

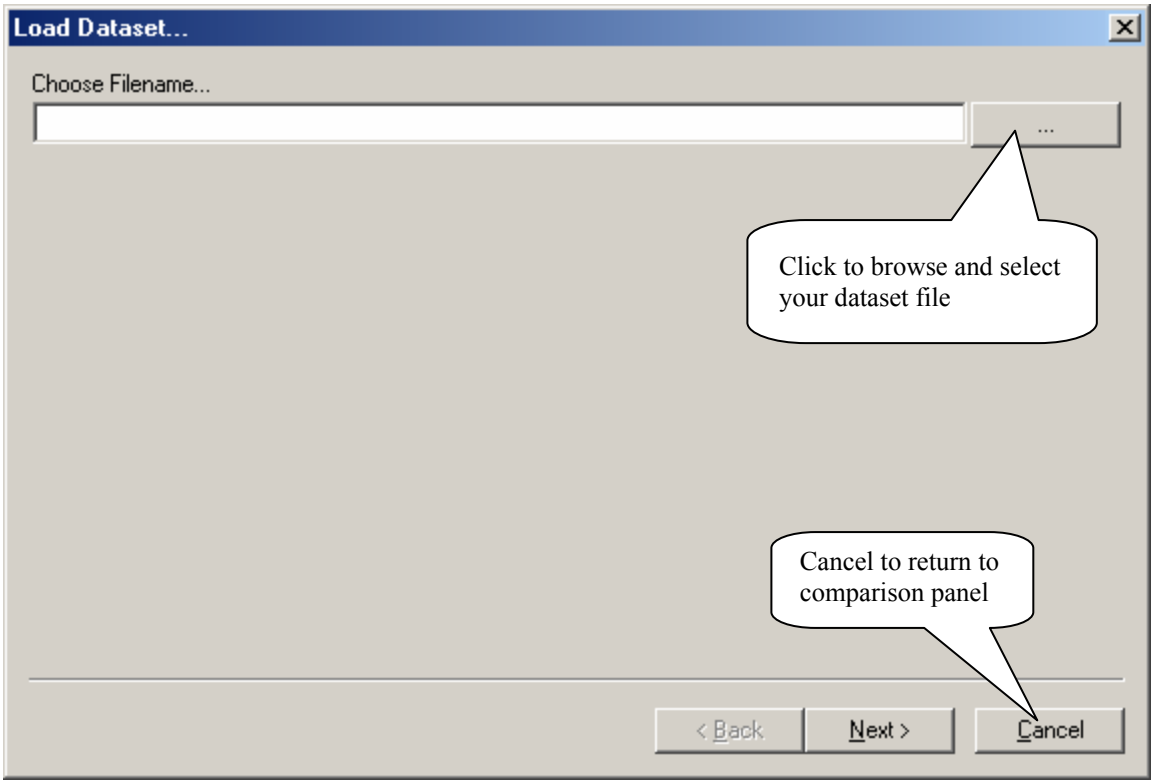


Figure 2. Load dataset (step 1).

Several sample datasets are installed with the tool in the 'Example Data' directory under the installation directory (default: C:\Program Files\Rutgers University\Cluster Comparison and Visualization Tool). In the browse window, click on the file you want to load and then click on the '*Open*' button (Figure 3) to go to the next step.

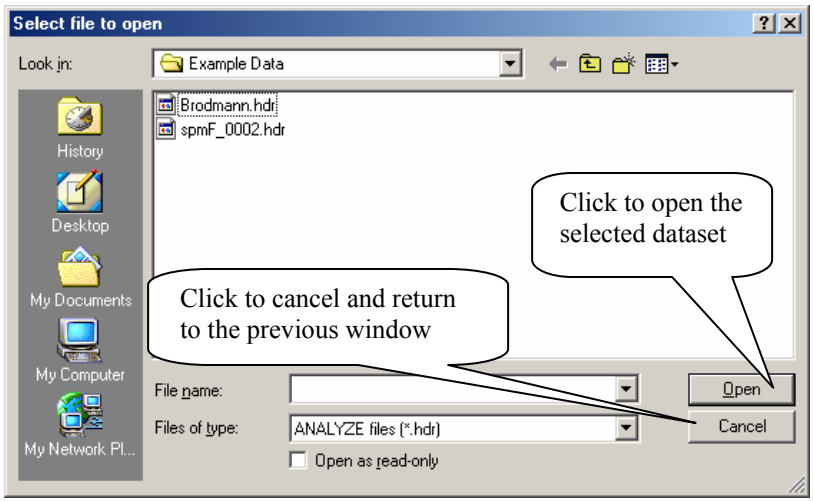


Figure 3. Choose a file to load.

Click on the 'Next' button (Figure 4) to go to the next step.

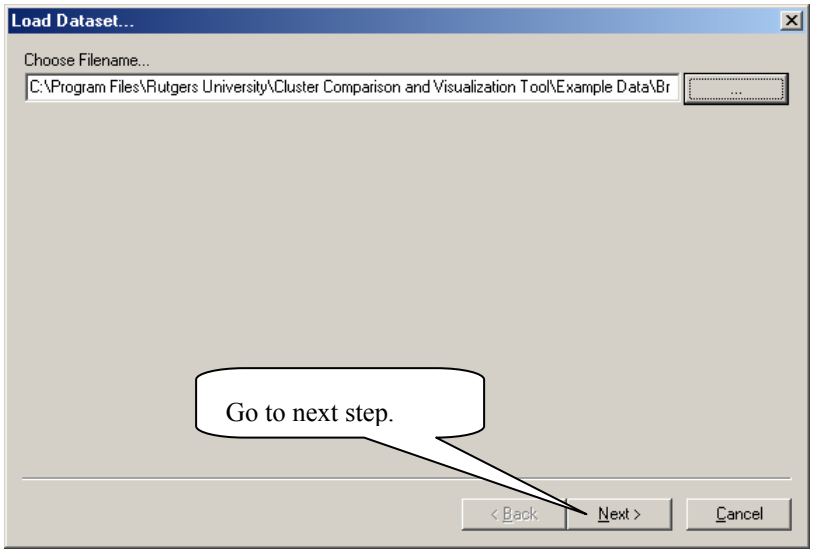


Figure 4. Load selected dataset.

The tool will allow you to directly load a t- or f- map produced by analysis tools such as SPM, AFNI, or FSL. Any type of statistical map can be loaded. The maps can have any dimensions but they must be normalized.

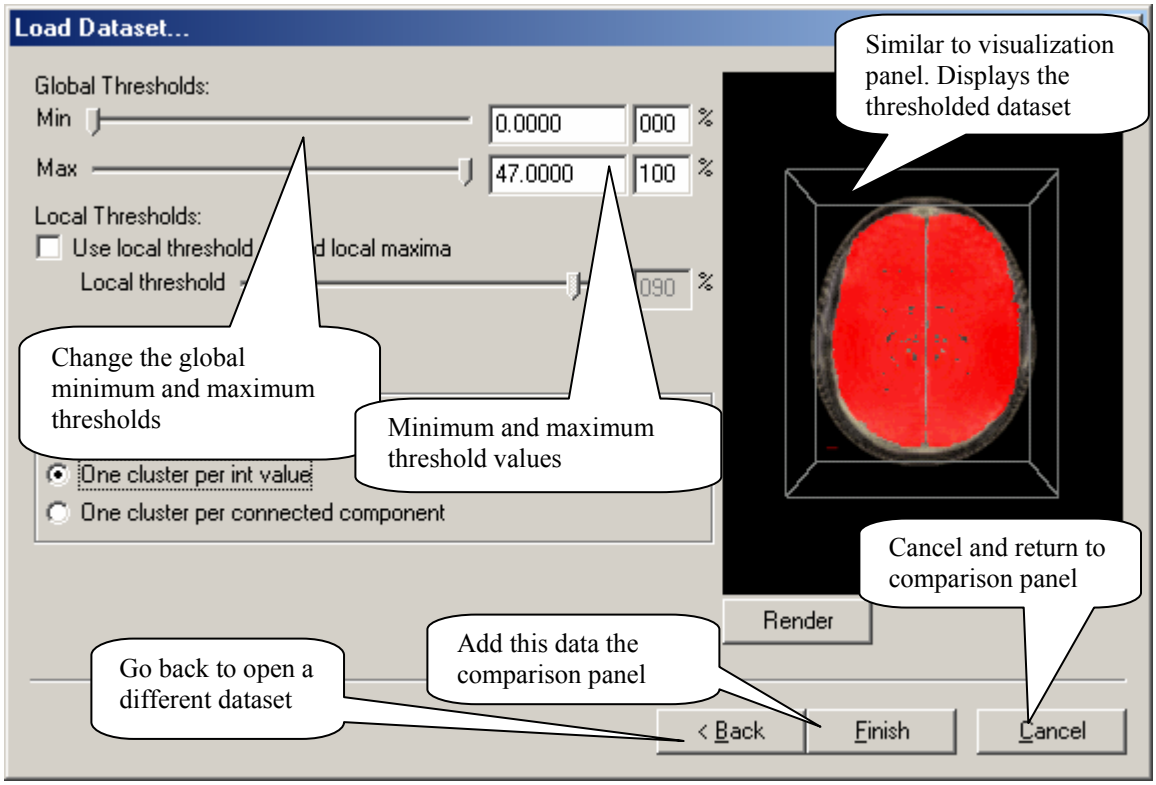


Figure 5. Loading Dataset (step 2).

When loading a statistical map, the dataset contains a floating point value at each voxel, both inside and outside the brain. You are allowed to choose a threshold while interactively viewing the supra-threshold voxels (Figure 5). This functionality is desirable especially because the choice of a suitable threshold is not a simple matter and is usually done in a blind fashion at run time in the analysis tool.

You can also load other types of datasets besides statistical maps, such as cluster masks, where each voxel contains an integer value specifying the cluster it belongs to. Changing the threshold values is not meaningful for this type of already clustered datasets.

2.1 Changing the global threshold

Use global threshold sliders (Figure 5) to increase or decrease the global threshold level. While a big enough threshold guarantees a high probability that the selected voxels are really active, it also selects only a few usually spatially disconnected voxels. On the other hand, a smaller threshold will include more voxels, maybe more spatially coherent but there is a bigger chance that some of them are not really active. CCVT gives you this choice, as conditions vary from experiment to experiment.

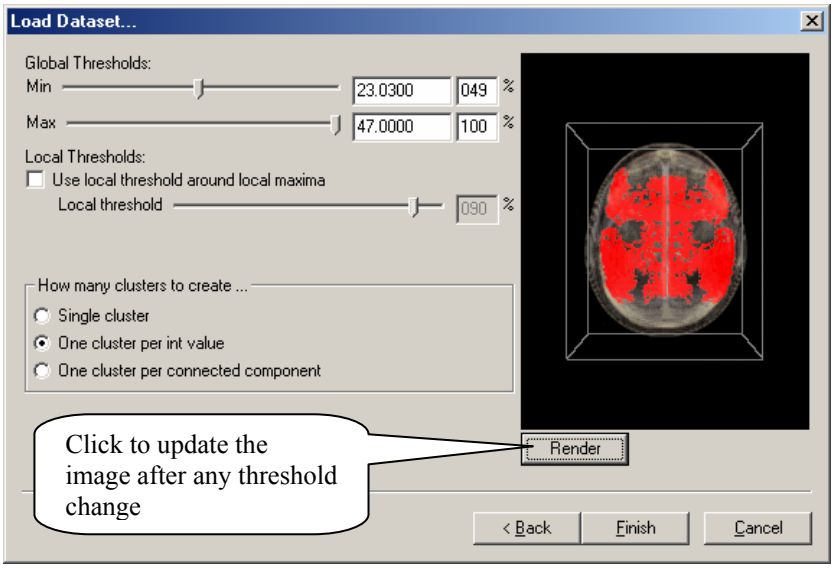


Figure 6. Modified Min Threshold.

The Min/Max sliders specify a range of threshold values. The dataset is thresholded so that all values below the Min threshold and above the Max threshold are set to zero. While moving the sliders to change the threshold values, in the textboxes next to each slider, the application displays the current threshold both as an absolute value and as a percentage of the maximum value in the dataset. For positive correlations, one generally needs to set the max threshold at the highest value and change the 'Min' slider (Figure 6).

For the negative correlations, the ‘Min’ slider should be set at the minimum value and the ‘Max’ slider should be varied (Figure 7).

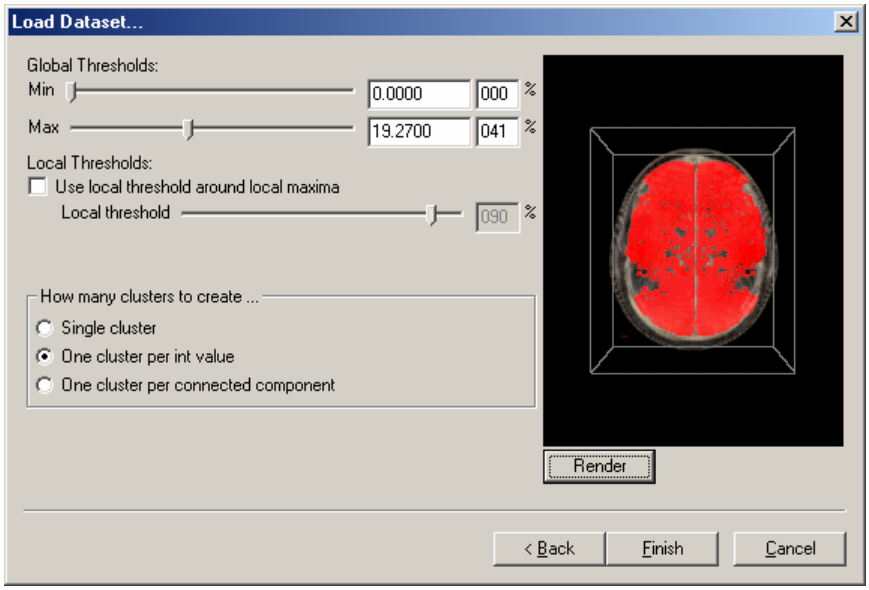


Figure 7. Modified Max Threshold.

2.2 Changing the local threshold

Use a local threshold (Figure 8) to increase the size of the individual spatial clusters. This local threshold is effective in the vicinity of local maxima that obey the global thresholds and allows voxels that are outside the global threshold but within a certain percentage of the value of the local maximum to be included in the cluster. The effect of this local threshold is to increase the number of voxels in each spatial cluster.

To visualize the effect of any threshold change, one must click on the ‘Render’ button. The visualization part in this window is similar to what will be displayed in the Visualization Panel (Figure 13). To rotate and see the effect of changing thresholds on the image click and drag on the image using the left mouse button.

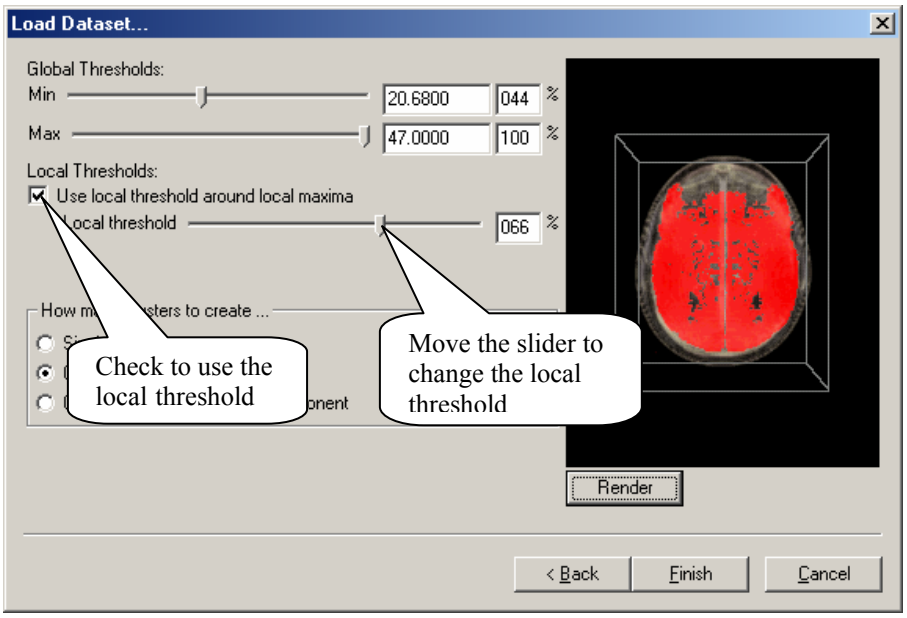


Figure 8. Setting the Local Threshold.

2.3 Select Cluster-type

This choice (Figure 9) allows one to treat the supra-threshold voxels as a single cluster (a functional cluster) or as multiple clusters. It is used for visualization purposes. Different choices are as follows:

- Choose '*single cluster*' to have all supra-threshold voxels as a single functional cluster. By selecting this choice, there will be only one column or row in the comparison table for this dataset.
- Choose '*One cluster per int value*' so that the actual value of the voxel in the dataset determines the cluster it belongs to. Each different value will be treated as a different cluster. When using this option with datasets containing floating point values, the application will truncate the non-zero absolute

values to an integer and add one. This option is meant to be used with data that is already clustered (cluster masks) by the analysis tool.

- Choose '*One cluster per connected component*' to have each connected set of voxels treated as a separate cluster (26-connectivity is used). This option will actually cluster the data.

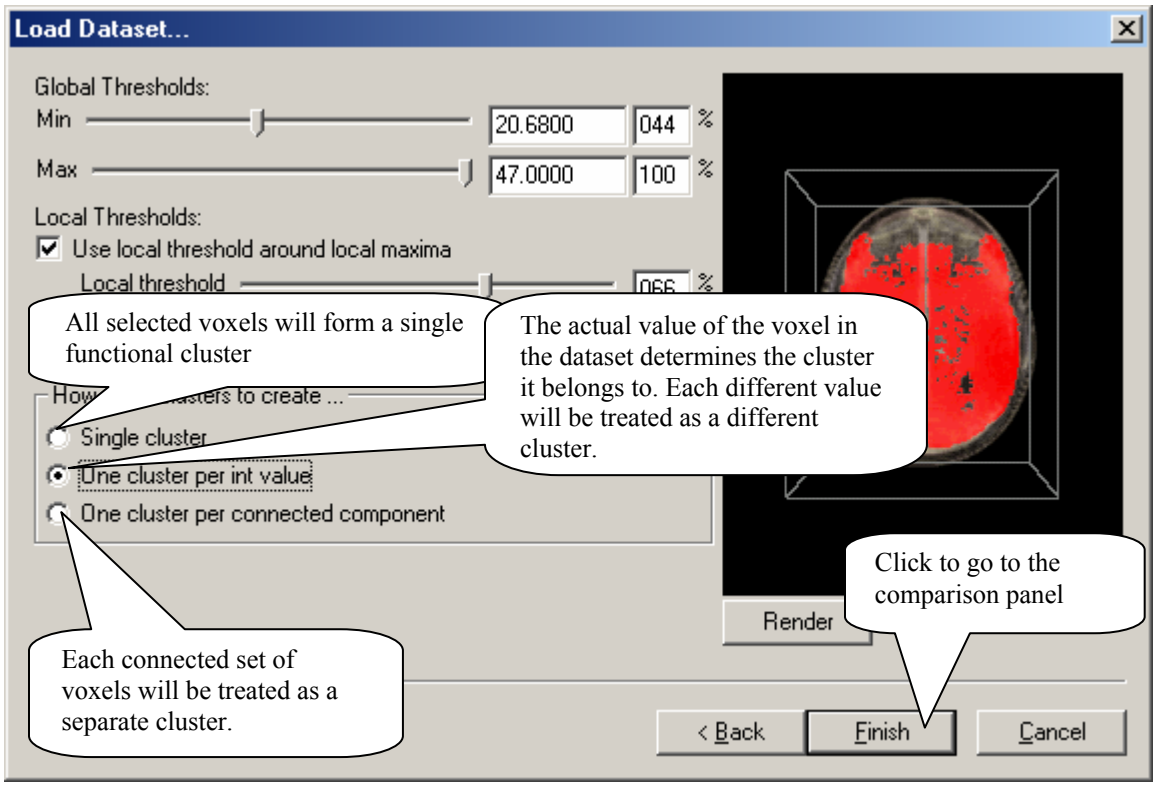


Figure 9. Change the cluster type.

Click on the '*Finish*' button. This will bring up the Comparison Panel (Figure 10). Please note that at least two datasets need to be loaded; the first is put on y axis (the rows of the comparison table) and the second on the x axis (the columns of the comparison table).

3. Data Comparison Panel

The data comparison panel displays quantitative information about the datasets being compared. It also allows the user to quickly identify the clusters that present high mutual similarity and are worth investigating.

3.1 Dataset properties

Use the upper part of the comparison panel (Figure 10) to add or remove a dataset, or change its color and opacity.

Every dataset that is loaded into the comparison panel is assigned a color and an opacity that will be used by the rendering engine that manages the data visualization panel. Since two or more clusters can be shown in the visualization panel at the same time, different colors and opacities for clusters belonging to different datasets can facilitate a better understanding of their relationships. Opacity lets the user adjust the transparency of all the clusters in a dataset.

A dataset can be added to the comparison panel via the '*Add…*' button and removed via the corresponding '*Remove*' button (Figure 10).

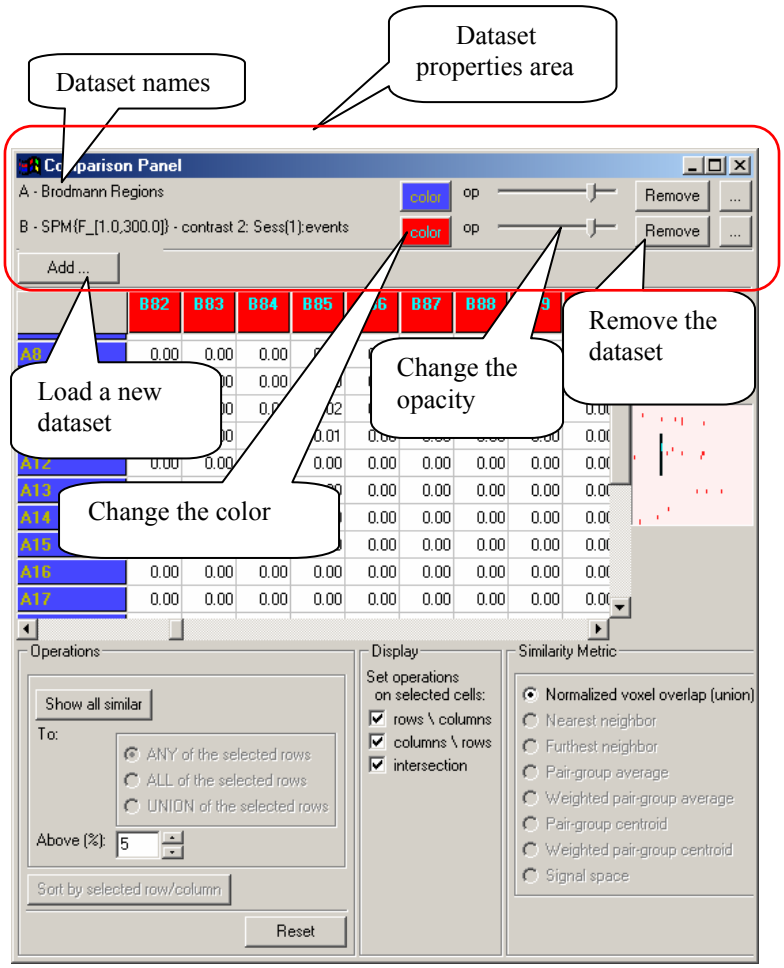


Figure 10. Dataset properties area in the comparison panel.

3.2 The Similarity Table

The central part of the data Comparison Panel is the similarity table (Figure 11), based on a spreadsheet format, which displays similarity scores between pairs of clusters. The similarity scores are computed by the chosen *similarity metric*. The rows and columns of the similarity table represent the different clusters in the loaded datasets and each individual cell shows the similarity score of the clusters corresponding that cell's row and column.

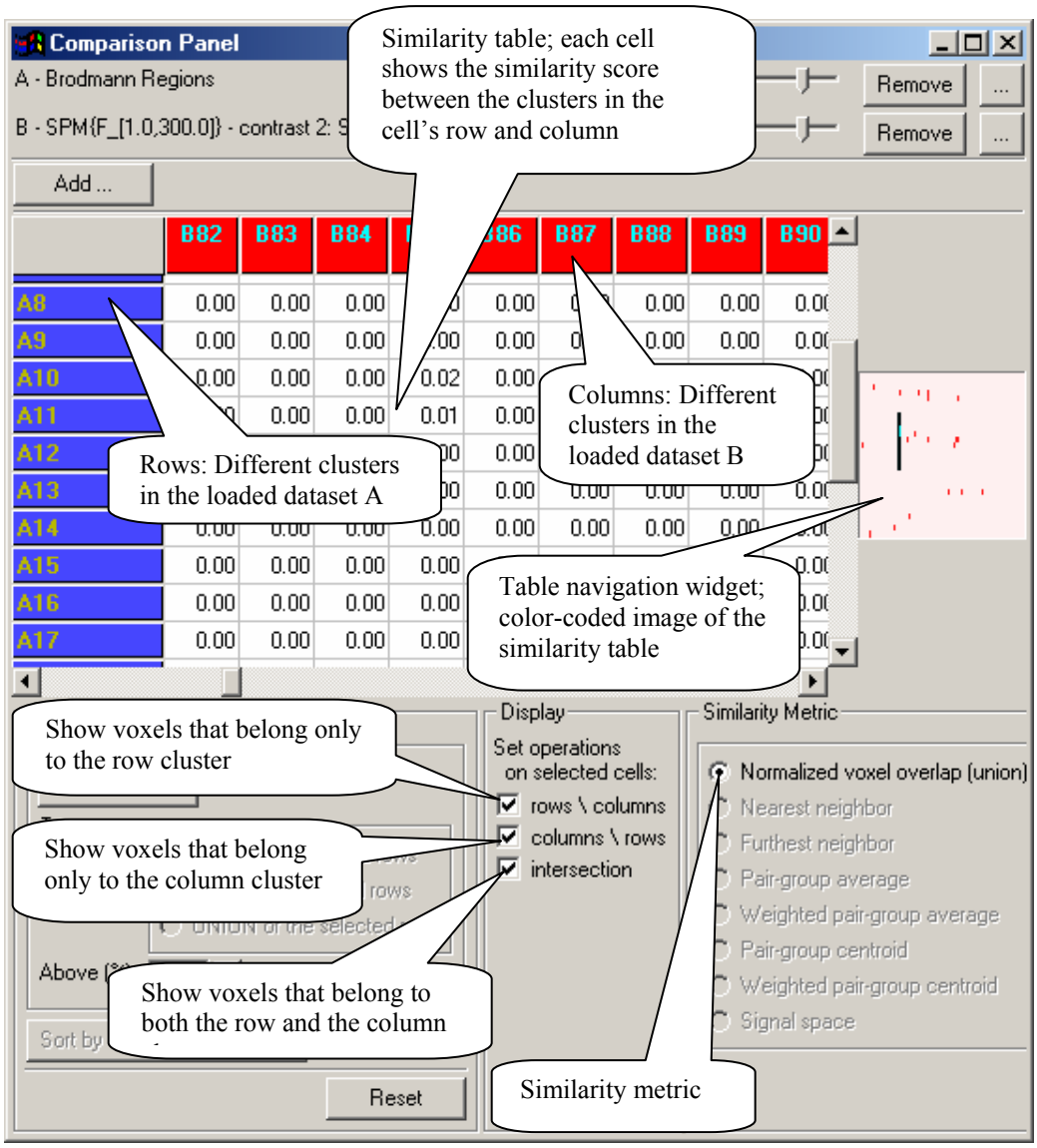


Figure 11. Similarity table.

3.3 Operations area

Because the table can be very large and unwieldy there are a number of "operations" that can be performed to enable the user to focus on the 'important' parts of the table. These are:

- *Table navigation widget*; displays a condensed view of table allowing the user to concentrate on areas of similarities. One can slide the black window (Figure 11) in the navigation widget to see the corresponding cells in the comparison table, either before collapsing the table or after that.
- *Display*; (Figure 11) this area controls the visualization. You can select any cell in the comparison table by clicking on it. Then if you select the '*rows\columns*' box, the cluster from the row dataset corresponding to the current cell will be shown on the Visualization Panel with the color assigned to that dataset. If you select the '*columns\rows*' box, then the cluster from the column dataset corresponding to the selected cell will be shown on the Visualization Panel with the color assigned to that dataset. If you need only to look at the intersection of the two clusters corresponding to the current cell, then you have to click on the '*intersection*' box.
- *Operations*; displays all similar pairs of clusters with one cluster from the rows and the other from the columns. You can specify the minimum acceptable similarity score in the '*Above*' box. Rows and columns of the table that do not contain at least one significant similarity score will collapse to make room for the "interesting" ones. After determining your minimum acceptable similarity score, click on '*Show all similar*' button (Figure 12) to view the collapsed table. This is again useful when dealing with a large number of rows and/or columns. Click on '*Reset*' button to return to the detailed comparison table (Figure 12).

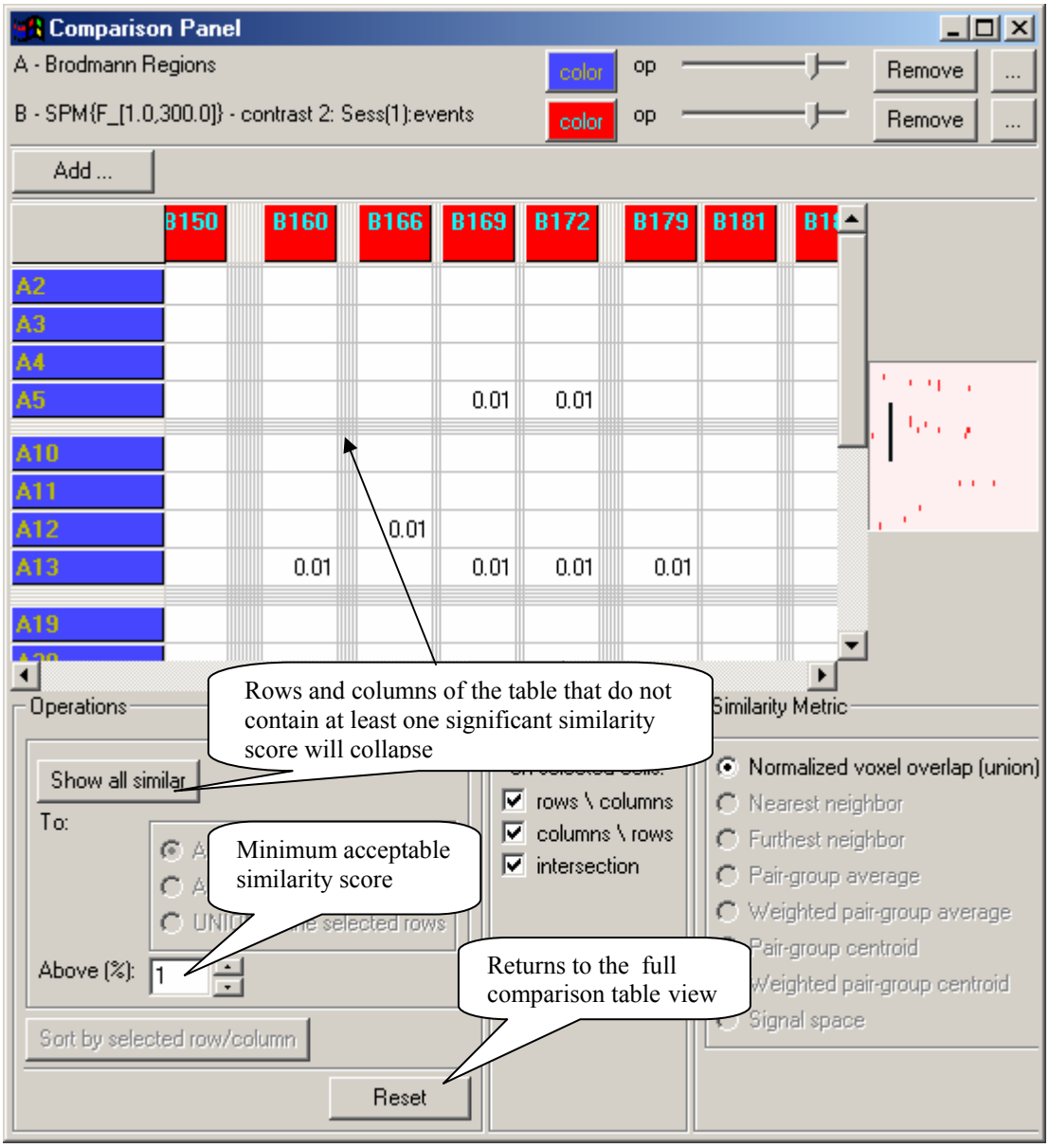


Figure 12. Operations area in the comparison panel.

4. The Visualization Panel

Once a dataset is loaded, the Visualization Panel becomes active. The visualization panel provides a visual display of the selected clusters in a common space overlaid onto a rendering of the brain anatomy. CCVT has such a layout so that the user can either focus

on the visual aspect of the data being compared, shown in the visualization panel or on the qualitative aspect displayed in the similarity table. This design provides the greatest flexibility, allowing one to completely ignore the quantitative similarity scores and focus only on the appearance of similarity which results from the visualization. This image provides visual information about the location of different clusters in the brain space. The graphical display in the visualization panel is linked to the similarity table in the comparison panel. As you browse through the similarity table, the two clusters corresponding to the rows and columns of the current selection in the spreadsheet are rendered together with the anatomical image in the same space. Each cluster is rendered using the color and opacity selected for its dataset in the dataset properties region of the comparison panel.

The visualization Panel maps the clusters into ‘brain’ space. The standard template brain (MNI template) is shown. The view can be rotated by clicking and dragging the left mouse button, zoomed by clicking and dragging the right mouse button or panned using the middle mouse button. There are Four standard views accessible from the push buttons at the bottom of the visualization panel: Axial (from above the head), Sagittal (from the left side of the head), and Coronal (from the front of the head).

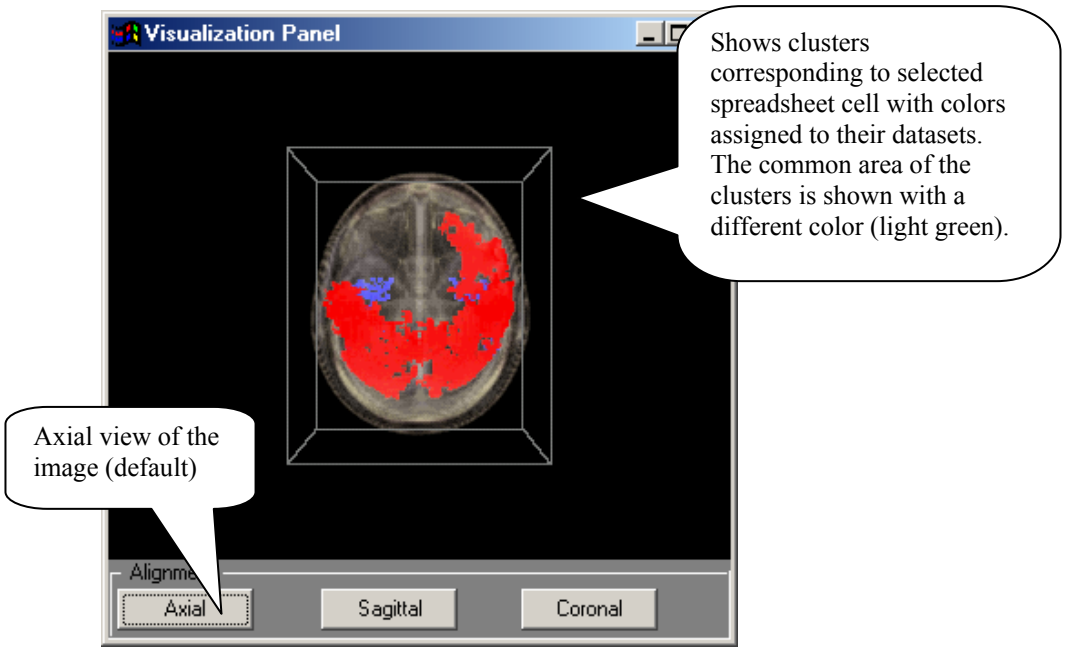


Figure 13. The Visualization Panel (axial view).

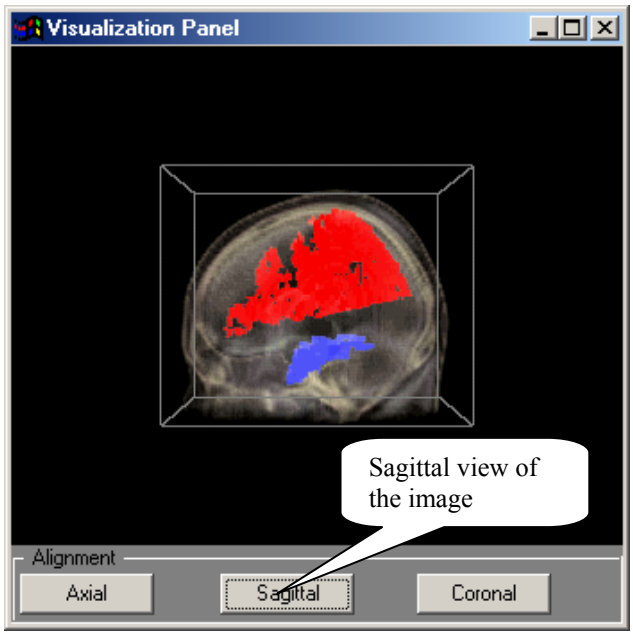


Figure 14. Visualization Panel (sagittal view).

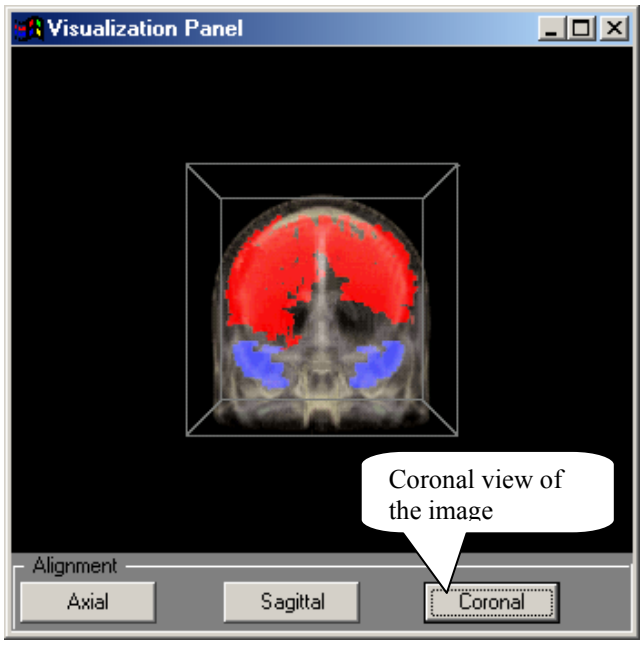


Figure 15. Visualization Panel (coronal view)