Human Brain Mapping 2006 Print

Abstract Number: 2266 Submitted By: Yasser Ad-Dab'bagh
Last Modified: January 8 2006

The CIVET Image-Processing Environment: A Fully Automated Comprehensive Pipeline for Anatomical Neuroimaging Research

Yasser Ad-Dab'bagh[1,2,4], Dale Einarson[4], Oliver Lyttelton[3,4], J-Sebastian Muehlboeck[4], Kelvin Mok[3,4], Oleg Ivanov[3,4], Robert D. Vincent[4], Claude Lepage[4], Jason Lerch[2,4], Eric Fombonne[1], Alan C. Evans[2,3,4]
[1]Department of Psychiatry, Division of Child Psychiatry, McGill University
[2]Department of Neurology and Neurosurgery, McGill University
[3]Department of Biomedical Engineering, McGill University
[4]McConnell Brain Imaging Centre of the Montreal Neurological Institute, McGill University

Objective: The CIVET project intends to consistently provide researchers with no programming background with the means to conduct automated structural research at the McConnell Brain-Imaging Centre (BIC). It is designed as an interface linking ever-growing and improving imaging software with the researcher's demands for reproducibility and consistency of results.

Methods: Development of CIVET has been pursued in parallel along eight main directions:

1- Quarantine-Building: Rapid ongoing development of BIC software has implications on reproducibility of results. By quarantining installed software, a particular data-set can be associated with a "fixed" set of "versioned" software for any future work on that data-set, ensuring reproducibility and comparability of results. This simultaneously allows for quantification of changes introduced by new versions of software, thereby facilitating software validation. We have developed a script that automates the process of installing the latest stable versions for hundreds of software packages available at the BIC in a controlled manner on most platforms.

2- Centralization of maintenance and debugging of quarantined software.

3- Integration of functionality into CIVET to permit all expedient combinations of processing operations.

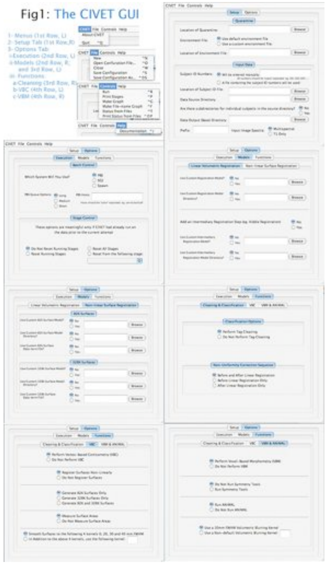
4- Modularization of CIVET to minimize inter-component dependency so that each component (set of processing activities) can be independently developed, and then combined into a customized analysis pipeline.

5- Implementation of Automated Quality-Control (AQC) for registration, classification, and surface-fitting using reference distributions to identify outlier voxels and vertices.

Thresholds can then be set specifying acceptable limits for the number of outliers per scan, and the AQC images produced will highlight regions of divergence.

6- Development of an independent debugging tool that assures integrity of CIVET, assesses the functionality of compiled modules, and reports any dependency conflicts within the entire CIVET framework to the developer. Moreover, it will guide the end-user towards selecting appropriate and compatible combinations of options.

7- Development of a JAVA-based graphical-user interface (GUI) wrapper for CIVET (Fig.1) that activates a Perl-script which in turn sources the desired quarantine-environment, then produces a command-line and passes it to CIVET, while generating a configuration file that contains the choices made by the user.



8- Detailed ongoing documentation by the community of developers working on any component of the project.

Results: CIVET's architecture is now modular in design, with on-going parallel task-specific development. It is composed of a pipeline-invoking shell and core-modules that control the parallel processing of data (using N3, MNI-Autoreg, INSECT, ANIMAL, CLASP, SURFREG, Diffuse, SEAL, among others, enabling both voxel- and vertex-based analysis of data) in a safe, dependency- and integrity-managed environment. This core architecture can be invoked to run any set of processing tasks with a single command-line, or through the use of a very user-friendly GUI.

Conclusions: CIVET is a comprehensive, easy to use and implement pipelining environment for fully automated image-processing of large data-sets that meets the needs of both developers and end-users and enables researchers to focus on scientific questions both at the biological as well as the computational ends.

References: Documentation of the CIVET project can be found at wiki.bic.mni.mcgill.ca/index.php/CIVET

Close this window