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

Objectives

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). Simultaneously, it should allow developers and researchers with programming skills the flexibility of modifying the processes that operate on their data.

CIVET is designed as an interface linking ever-growing and improving imaging software with the researcher's demands for reproducibility and consistency of results. It will enable researchers to perform vertex-based corticometric tasks (VBC), voxel-based morphometric tasks (VBM), volumetry and symmetry analysis.

The CIVET project will attempt to provide extensive documentation that would meet the needs of users of varying degrees of computational expertise.

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 code-base to minimize inter-component dependency so that each component (set of processing activities, see Fig.1) can be independently developed, and then combined into a customized analysis pipeline.

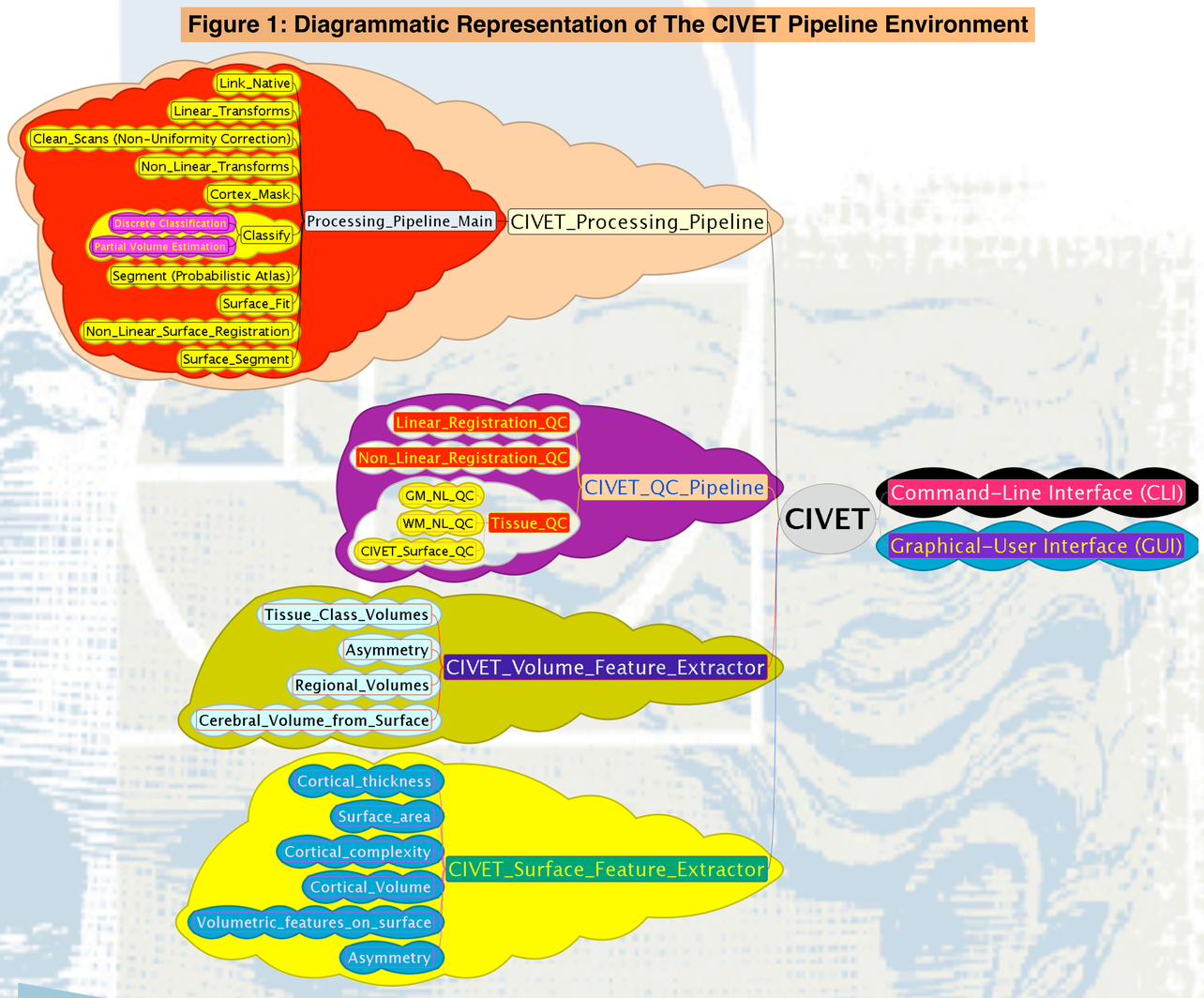
5- Implementation of Automated Quality-Control (AQC) for registration, classification, and surface-fitting using reference distributions to identify outlier voxels and vertices. (see Fig.1)

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.2) 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.

Figure 1: Diagrammatic Representation of The CIVET Pipeline Environment

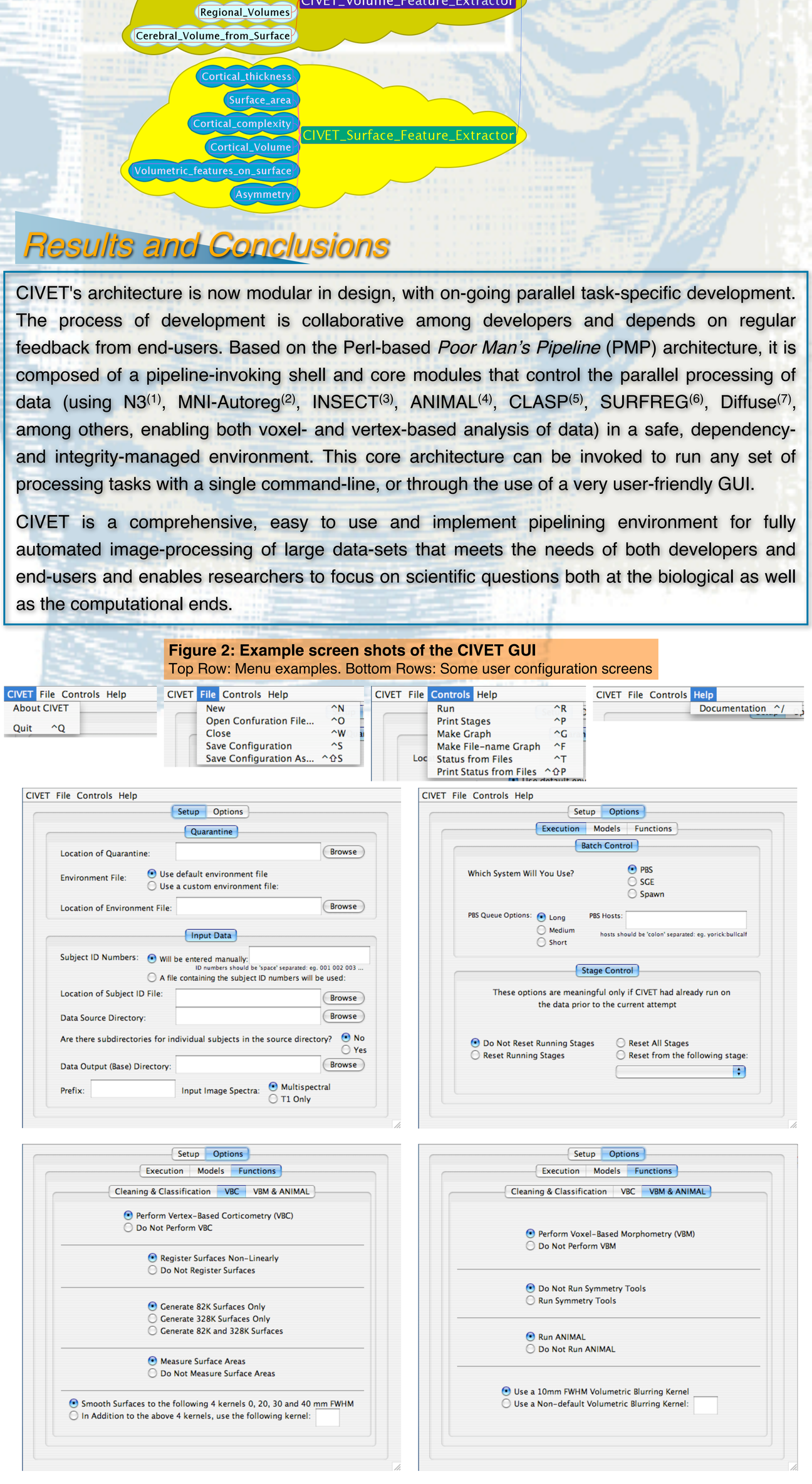


Results and Conclusions

CIVET's architecture is now modular in design, with on-going parallel task-specific development. The process of development is collaborative among developers and depends on regular feedback from end-users. Based on the Perl-based *Poor Man's Pipeline* (PMP) architecture, it is composed of a pipeline-invoking shell and core modules that control the parallel processing of data (using N3[(1)], MNI-Autoreg[(2)], INSECT[(3)], ANIMAL[(4)], CLASP[(5)], SURFREG[(6)], Diffuse[(7)], 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.

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.

Figure 2: Example screen shots of the CIVET GUI
Top Row: Menu examples. Bottom Rows: Some user configuration screens



References & Credits

A- CIVET was developed collaboratively by:

- *Yasser Ad-Dab'bagh*: Developed pre-modular versions of CIVET; Project coordination; Assessment of output quality; Interface design for GUI wrapper; Documentation…
- *Dale Einarson:* Major role in developing the early quarantine-building scripts…
- *Oliver Lyttelton:* Modularized CIVET; Added AQC pipeline; Substantially Improved surface extraction & registration; Added capability to extract surface area & complexity among other features…
- *J-Sebastian Muehlboeck:* Debugging and standardization of CIVET modules; Further developed PMP; Developed methods for connecting data-based sets with AQC output…
- *Kelvin Mok:* Developed independent debugging tool…
- *Oleg Ivanov:* Developed the scripts connecting the GUI wrapper to underlying architecture…
- *Robert D. Vincent:* Extensively debugged minc tools that CIVET uses…
- *Claude Lapage:* Developed the 3rd edition of CLASP; Further developed and improved the quarantine-building script; Further developed PMP; Contributed tools such as Diffuse and mincdefrag; Debugging minc and CIVET tools …
- *Jason Lerch*: Original developer of PMP and pipelines designed to run the gamut of BIC tools; Developed statistical framework for analysis of corticometric results.

B- References for tools called upon by CIVET:

1. Sled JG, et al.: *IEEE Trans Med Imaging* 1998; 17:87–97
2. Collins DL, et al.: *J Comput Assist Tomogr* 1994; 18:192–205
3. Zijdenbos A, et al: *MICCAI'98: First International Conference*, Cambridge, MA, USA, October 1998 Proceedings.
4. Collins DL, et al.: *Human Brain Mapping* 1995; 3:190–208
5. Kim JS, et al. *NeuroImage*, 27:210–21, 2005.
6. Robbins SM: *PhD Thesis*, School of Computer Science, McGill University. 2003.
7. Chung MK, et al.: *Neuroimage* 2001; 13:S95.