brain-view

brain-view

- brain-view
  - Overview
  - What's new in version 2?
  - Obtaining the code
  - Prerequisites
  - Building brain-view
  - Using brain-view
  - Discrete Template Colormap for CIVET AAL Atlas
  - Installation notes for Ubuntu 16.04 (September 2016)
  - Installation

Overview

Brain-view is designed to visualize bits of geometry - such as meshes representing the cortical surface, line objects representing the centre line of vessels, or connected cylinder objects representing segmented vasculature - along with associated information, such as cortical thickness at every vertex, statistical maps, labels of vessel segments on every cylinder, etc. It is based on Coin (Inventor), Qt, and the various MINC file formats.

What's new in version 2?

Version 2 is a complete rewrite of brain-view. The main motivation were to be able to deal with multiple objects, such as left and right surfaces, etc., and in the process clean up some of the cruft that had accumulated in the previous code-base. Not all features of brain-view 1 are included - there's no interaction with R, no painting ability, etc. All that is planned, however. For the moment only use brain-view2 if you want to experiment with it and help with the testing.

Obtaining the code

Brain-view is hosted at github: https://github.com/sghanavati/brain-view2 Go to the Downloads section of that website to get the latest release, or, to stay most up to date, get the source code directly from github (the version control system):

```
git clone https://github.com/Mouse-Imaging-Centre/brain-view2.git
```

Prerequisites

To build brain-view you will need:

- Qt version 4.x (development was done against 4.5, but seems to work on 4.4 and 4.3 as well).
- Coin 3.0. (http://coin3d.org - Coin 2.5 might also work with brain-view, but this has not been tested yet).
- Quarter (http://coin3d.org/lib/quarter/releases/1.0.0)
- MINC version 2.x, bicpl, and oobicpl (http://packages.bic.mni.mcgill.ca/tgz/)
- boost_1.48.0 or higher (http://www.boost.org/users/download/)
- hdf5-1.8.6 or higher (http://www.hdfgroup.org/HDF5/release/obtain5.html)

Building brain-view

brain-view is built using Qt's qmake. The basic steps are the following:

```
qmake MINCDIR=/path/to/minc INVENTORDIR=/path/to/bicInventor QUARTERDIR=/path/to/quarter HDF5DIR=/path/to/HDF5
brain-view2.pro
make
```

On Linux you will then have a binary called brain-view2 which can be copied where other such binaries live. On OS X you then have an application bundle that can be moved to the Applications folder.

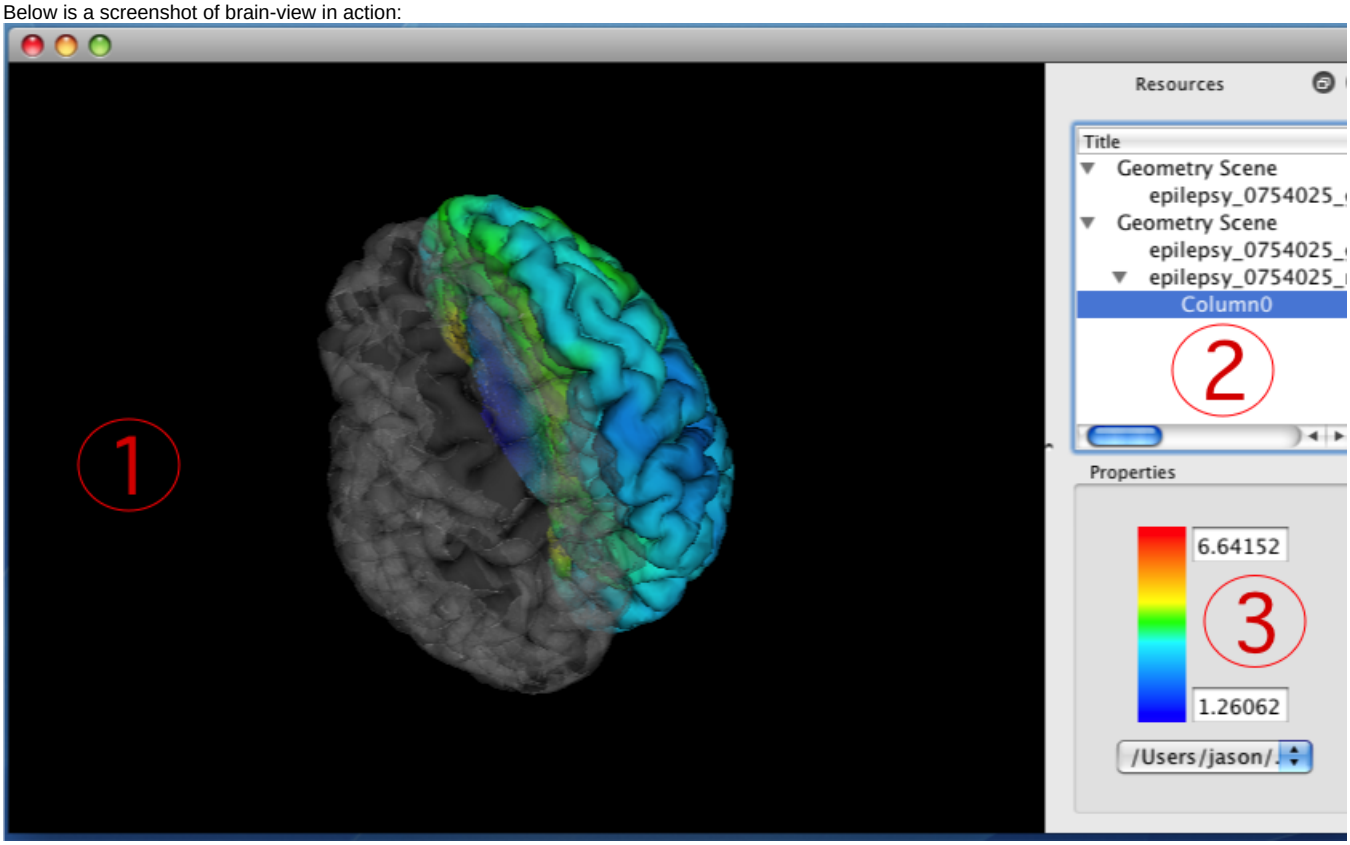
Note that on many Linux distributions both qt3 and qt4 will be present. In that case make sure you are using the qmake associated with qt4 - on Ubuntu, for example, this can be done by using 'qmake-qt4' in place of the plain 'qmake'.

(See also the end of this page for more details).

Using brain-view

Start brain-view by double clicking on the brain-view2 icon or entering it on the command line. If started from the command line, it can take an arbitrary number of filenames corresponding to obj files (polyhedra or line at the moment), files associated with the obj files (.txt or .vertstats files), tag files, hdf5 files (.h5 files are created by vessel tracking code and post-processed into hdf5 database format. In order to successfully read these files into brain-view, required groups and datasets should exist in the file. more detail will be provided below), and configuration files (.config providing the numerical and anatomical labels and their corresponding RGB colors) . The order of the filenames matter, since .txt/.vertstats files have to follow the geometry file with which they are associated. Files can also be opened through the file menu once the application has started.

Below is a screenshot of brain-view in action:



There are three main areas to the application: area (1) is the rendering area, area (2) a hierarchical list detailing all of the surfaces and associated files, and area (3) containing properties associated with each item in the hierarchical list. In the example above two surfaces (left and right hemisphere for the same subject in this case) have been loaded, resulting in two geometry scenes visible in area (2). Below each geometry-scene is an entry for the surface itself - clicking on it will allow one to change the colour and opacity in the property window (the left surface having been set to an opacity of 0.75 in the above example). The right surface also has a vertstats file opened, which in this case contains a single column (Column0) describing the cortical thickness. Clicking on each column will cause that column to be mapped onto the surface using the colourmap shown in the property area. Different ranges for the colour mapping can be chosen by entering the desired values in the upper and lower range text boxes, and different colourmaps are selectable from the drop-down box.

If any bit of the scene is clicked on in the rendering area the information in the hierarchical list will update with the values at the selected location. You might have to scroll to the right in the hierarchical list to see it. Also note that the hierarchical list can be detached from the main window by selecting and dragging it (where it says "Resources") away from the main window. It can be redocked later through the same process.

Discrete Template Colormap for CIVET AAL Atlas

If you are using the `CIVET-1.1.12/models/AAL_atlas_left.txt` and `CIVET-1.1.12/models/AAL_atlas_right.txt` files, there is a discrete unique-color bar here. This allows for nice clear labeling of a CIVET object in brain-view2. Intensity range must be set from 0-91 to work properly with the surface atlas. (Thanks Gabriel Devenyi for providing the color bar. It was generated using http://tools.medialab.sciences-po.fr/iwanthue/ + matplotlib + gimp.)

Installation notes for Ubuntu 16.04 (September 2016)

1) install the minc-toolkit (https://github.com/BIC-MNI/minc-toolkit) or the minc-toolkit-v2 (https://github.com/BIC-MNI/minc-toolkit-v2) (more information about the minc-toolkit (both version 1 and version 2). In this example version 2 is used:

```
git clone --recursive https://github.com/BIC-MNI/minc-toolkit-v2.git
mkdir minc-toolkit-v2-build
cd minc-toolkit-v2-build/
# see https://github.com/BIC-MNI/minc-toolkit-v2 for complete cmake/ccmake build instructions
# cmake -DCMAKE_INSTALL_PREFIX=/path/to/install/all-minc-tools/ ../minc-toolkit-v2
make
make test
make install
cd ..
```

2) install Coin3D:

```
hg clone https://bitbucket.org/Coin3D/coin
cd coin
hg update CMake
cd ../
mkdir coin-build
cd coin-build
cmake -DCMAKE_INSTALL_PREFIX=/path/to/install/all-minc-tools/ -DCMAKE_BUILD_TYPE=Release ../coin
make
make install
cd ..
```

3) Qt4 - get those through the system libraries

```
aptitude install qt4-default libqt4-dev libqt4-dbg qt4-qmake libqt4-opengl-dev
```

4) Quarter

```
hg clone https://bitbucket.org/Coin3D/quarter
cd quarter
```

Quarter wants to use Qt5 if it's installed, but I got a fair bit of errors when trying that. So, I turned it off as follows. Edit the CMakeLists.txt in the quarter directory. Replace these lines:

```
FIND_PACKAGE(Qt5 QUIET COMPONENTS UiTools OpenGL)
IF(Qt5_FOUND)
  SET(QT_LIBRARIES Qt5::UiTools Qt5::OpenGL)
ELSE(Qt5_FOUND)
  FIND_PACKAGE(Qt4 REQUIRED)
  INCLUDE(${QT_USE_FILE})
ENDIF(Qt5_FOUND)
```

with:

```
FIND_PACKAGE(Qt4 REQUIRED)
INCLUDE(${QT_USE_FILE})
```

```
cd ..
mkdir quarter-build
cd quarter-build
export PATH=/path/to/install/all-minc-tools/bin:$PATH
export LD_LIBRARY_PATH=/path/to/install/all-minc-tools/lib
cmake -DCMAKE_BUILD_TYPE=Release -DCMAKE_INSTALL_PREFIX=/path/to/install/all-minc-tools/ ../quarter
make
make install
cd ..
```

5) Boost - system packages

```
aptitude install libboost-all-dev libboost-chrono1.58-dev libboost-thread1.58-dev libboost-filesystem1.58-dev
libboost-iostreams1.58-dev libboost-program-options1.58-dev libboost-regex1.58-dev libboost-signals1.58-dev
libboost-system1.58-dev
```

6) pcre++ - system libraries

```
sudo aptitude install libpcre++-dev
```

7) bicInventor

```
git clone https://github.com/BIC-MNI/bicInventor.git
cd bicInventor/
./autogen.sh
./configure --prefix=/path/to/install/all-minc-tools/ --with-build-path=/path/to/install/all-minc-tools/ --with-
minc2
make
make install
cd ..
```

8) brain-view2

```
git clone https://github.com/Mouse-Imaging-Centre/brain-view2.git
cd brain-view2/
qmake MINCDIR=/path/to/install/all-minc-tools/ INVENTORDIR=/path/to/install/all-minc-tools/ QUARTERDIR=/path/to
/install/all-minc-tools/ HDF5DIR=/path/to/install/all-minc-tools/ brain-view2.pro
make
cp brain-view2 /path/to/install/all-minc-tools/bin/
cd ..
```

Installation

In order to install brain-view on Mac OSX mountain lion you will need to take the following steps. The installation on a Linux-based OS such as ubuntu will follow similar steps but probably with less headache!

0. Install X11 and Xcode (This step is only required for Mac OSX which doesn't have a C/C++ compiler pre-installed on it)

install X11 from XQuartz2.7.3 project: http://xquartz.macosforge.org/landing/

Install Xcode 4.4.1 from apple store: https://developer.apple.com/xcode/ . This contains gcc and llvm compilers + the IDE.
Install command tools from inside Xcode session.

1. Install HDF5

The latest supported public release of HDF5 is available from ftp://ftp.hdfgroup.org/HDF5/current/src.

$ tar zxf hdf5-X.Y.Z.tar.gz

$ ./configure --prefix=/usr/local/hdf5

$ sudo make?

$ sudo make check # run test suite.?

$ sudo make install?

$ sudo make check-install # verify installation.

2. Install Coin3D

wget http://ftp.coin3d.org/coin/src/all/Coin-3.1.3.tar.gz
$ tar xvzf Coin-3.1.3.tar.gz
$./configure
$ make
$ sudo make install

ftp.coin3d.org is no longer reachable (Nov 26, 2015). Home page redirects to a Bitbucket mercurial project.

```
hg clone https://bitbucket.org/Coin3D/coin
mkdir coin-build
cd coin
hg update CMake #switch to CMake branch, main no longer builds
cd ../coin-build
cmake -DCMAKE_INSTALL_PREFIX:PATH=$HOME/local/ ../coin #or ccmake for easier customization
make
make install
```

Then you can clear out the build directory. Make sure $HOME/local/bin is on your path.

#TODO: ensure this works at MICe, I didn't do a local install on my machine

3. Install Qt

download from http://qt-project.org/downloads

if you choose not to go with binary, and want to install from the source code then:

$ tar xvzf qt-everywhere-opensource-src-4.8.3.tar.gz
$ ./configure -prefix /usr/local/Qt-4.8.3

$ make

$ sudo make install

Go for a coffee break. Qt takes a while to be installed.

4. Install Quarter

download from http://ftp.coin3d.org/coin/src/all/Quarter-1.0.0.tar.gz
$ tar xvzf Quarter-1.0.0.tar.gz

I had some problems with usage of Quarter and Qt by brain-view which caused the program to crash. The problem was that I had installed X11 beforehand which installed some of the Qt libraries in /opt/local and for some reason brain-view was looking in /opt/local for Qt libraries instead of /usr/local /Qt-4.8.3. I solved the problem by removing those Qt libraries from /opt/local and setting LDFLAG and CPPFLAG in Quarter installation:

$export QTDIR=/usr/local/Qt-4.8.3/
$./configure --prefix=/usr/local --with-qt=/usr/local/Qt-4.8.3/
$sudo make LDFLAGS=-L/usr/local/Qt-4.8.3/lib/ CPPFLAGS=-I/usr/local/Qt-4.8.3/include/

$sudo make LDFLAGS=-L/usr/local/Qt-4.8.3/lib/ CPPFLAGS=-I/usr/local/Qt-4.8.3/include/ install

As with Coin3d above, this project has moved and uses mercurial and cmake

```
hg clone https://bitbucket.org/Coin3D/quarter
mkdir quarter/quarter-build
cd quarter/quarter-build
ccmake .. #Change CMAKE_INSTALL_PREFIX to $HOME/local, press c, if it configures press g
make
make install
```

#TODO: make sure this works at MICe, I didn't do a local install on my machine

5. Install boost

download from http://www.boost.org/users/download/

tar -xvf boost_1_47
cd boost_1_47
./bootstrap.sh --prefix=/usr/local/
./b2 install

6. Install netCDF3

http://www.unidata.ucar.edu/downloads/netcdf/netcdf-3_6_3/index.jsp

or http://packages.bic.mni.mcgill.ca/tgz/

7.Install minc-2

http://www.bic.mni.mcgill.ca/ServicesSoftware/MINC

$./configure --with-build-path=/usr/local/hdf5

$ make

$ sudo make install

8. Install bicInventor (require GL libraries /opt/X11/include/GL/)

wget http://packages.bic.mni.mcgill.ca/tgz/bicInventor-0.3.1.tar.gz

$./configure --prefix=/usr/local/ --with-build-path=/usr/local/:/usr/local/hdf5/ --includedir=/usr/local/include --libdir=/usr/local/lib --with-minc2 LDFLAGS="-L /Library/Frameworks/Inventor.framework/Versions/C/Libraries" --x-includes=/usr/X11/include --x-libraries=/usr/X11/lib

$make

$sudo make install

bicInventor requires pcre++, so build that first. To avoid build difficulties, use the development version of bicInventor off github:

```
git clone https://github.com/BIC-MNI/bicInventor
cd bicInventor
./autogen.sh
./configure --prefix=$HOME/local --with-minc2
make
make install
```

9. Install oobicpl and bicpl

http://packages.bic.mni.mcgill.ca/tgz/

10. Install pcre++

http://www.daemon.de/PCRE

Get and compile version 0.9.5:

```
wget http://www.daemon.de/idisk/Apps/pcre++/pcre++-0.9.5.tar.gz
tar xvzf pcre++-0.9.5.tar.gz
cd pcre++-0.9.5
./configure --prefix=$HOME/local
make
make install
```

Finally, the actual installation of brain-view as mentioned above.

Many thanks to Mishkin Derakhshan for sending his travails in building brain-view2 to the minc-users list - his message is reproduced below:

> *I had a hell of time installing brain-view2 so I documented my errors*
> *and solutions/hacks in case others come across the same problems*
> *(read: when I have to do this again I can look back at this post).*
>
> *This was on a ubuntu 12.04 64 bit machine.*
>
> *I followed the instruction here (thank you jason and sghanavati):*
> *https://wiki.phenogenomics.ca/display/MICePub/brain-view*
>
> *Pre-requisites*
> *1. MINC. I used the minc-toolkit binaries and only ran into one minor problem.*
> *opus[~]$ sudo apt-get install libc6 libstdc++6 imagemagick perl*
> *freeglut3 libgl1 libxcb1 libxdmcp6 libx11-6 libxext6 libxau6 libuuid1*
> *libjpeg62 libexpat1 libtiff4*
> *Reading package lists... Done*
> *Building dependency tree*
> *Reading state information... Done*
> *Package libgl1 is a virtual package provided by:*
> *libgl1-mesa-swx11 8.0.2-0ubuntu3.1*
> *libgl1-mesa-glx 8.0.2-0ubuntu3.1*
> *You should explicitly select one to install.*
>
> *E: Package 'libgl1' has no installation candidate*
>
> *I chose libgl1-mesa-glx at random, and then everything else worked.*
>
> *2. Qt4: sudo apt-get install qt4-qmake libqt4-dev libqt4-core*
> *3. Boost: sudo apt-get install libboost1.48*
> *4.Coin:*
> *wget http://ftp.coin3d.org/coin/src/all/Coin-3.1.3.tar.gz*
> *tar xvzf Coin-3.1.3.tar.gz*
> *mkdir coin-build*
> *cd coin-build/*
> *../Coin-3.1.3/configure*
> *make*
> *sudo make install*
>
> *5. Quarter:*
> *wget http://ftp.coin3d.org/coin/src/all/Quarter-1.0.0.tar.gz*
> *tar xvzf Quarter-1.0.0.tar.gz*
> *mkdir quarter-build*
> *cd quarter-build/*
> *../Quarter-1.0.0/configure*
> *make*
>
> *ERROR:../../../Quarter-1.0.0/src/Quarter/Quarter.cpp:148:13: error:*
> *'stderr' was not declared in this scope*
> *SOLUTION:Added*
> *#include "stdio.h"*
> *to this file: ../Quarter-1.0.0/src/Quarter/Quarter.cpp*
>
> *make*
> *sudo make install*
>
> *6. HDF5 1.8.9:*

Downloaded a build of HDF5 1.8.9:
http://www.hdfgroup.org/HDF5/release/obtain5.html
tar xvzf hdf5-1.8.9-linux-x86_64-static.tar.gz

7. BicInventor:
wget http://packages.bic.mni.mcgill.ca/tgz/bicInventor-0.3.1.tar.gz
cd bicInventor-0.3.1
./configure --prefix=/opt/minc --with-build-path=/opt/minc
--includedir=/opt/minc/include --libdir=/opt/minc/lib --with-minc2
ERROR: (lots of erros when I tried to configure)
SOLUTION: sudo apt-get install build-essential realpath zlib1g-dev
bison flex libx11-dev libxmu-dev byacc libsoqt4-dev automake libtool
libdbi- perl libblas-dev liblapack-dev libblitz0-dev libreadline-dev
bzr bzrtools octave3.2 imagemagick

./configure --prefix=/opt/minc --with-build-path=/opt/minc
--includedir=/opt/minc/include --libdir=/opt/minc/lib --with-minc2
make
sudo make install

ACTUALLY INSTALLING BRAIN-VIEW2
mkdir bv2
cd bv2
git init
git remote add origin git@github.com:sghanavati/brain-view2.git
git pull -u origin master

qmake-qt4 MINCDIR=/opt/minc/
HDF5DIR=/home/mishkin/hdf5-1.8.9-linux-x86_64-static/ brain-view2.pro
make
ERROR: GeometryNode.h:25:30: fatal error: H5Cpp.h: No such file or directory
SOLUTION: sudo apt-get install libhdf5-serial-dev (probably a bad hack
because it installed 1.8.4 but it worked)

ERROR: H5AtomType.cpp:(.text+0x257): undefined reference to `H5Tget_pad'
SOLUTION: put -lhdf5_cpp before -lhdf5 in the g++ command
g++ -m64 -o brain-view2 MainWindow.o BrainQuarter.o TreeItem.o
TreeModel.o ResourceForm.o GeometryScene.o GeometryNode.o main.o
textureColumn.o textureFileItem.o tagFileItem.o tagPointItem.o
moc_MainWindow.o moc_BrainQuarter.o moc_TreeItem.o moc_TreeModel.o
moc_ResourceForm.o moc_GeometryNode.o moc_textureColumn.o
qrc_colourbars.o -L/usr/lib/x86_64-linux-gnu -L/usr/X11R6/lib64
-L/usr/local/lib -lCoin -L/lib/ -lQuarter -L/opt/minc//lib -loobicpl
-lpcre++ -lbicpl -lvolume_io2 -lminc2 -L/lib -lbicInventor
-L/home/mishkin/hdf5-1.8.9-linux-x86_64-static//lib/ -lhdf5_cpp -lhdf5
-lQtOpenGL -lQtGui -lQtCore -lGL -lpthread

And that gave me the brain-view2 binary which i copied into /opt/minc/bin.