Introduction

IGSRegistrationTools provides a no frills graphical user interface to the registration software under MacOS X. Use the RegistrationRunner application to do registrations and generate new images from the calculated registrations. Use the Average application to generate average brains (useful if your images are a bit noisy; may also protect somewhat against using an unusual brain as a template).

Installation

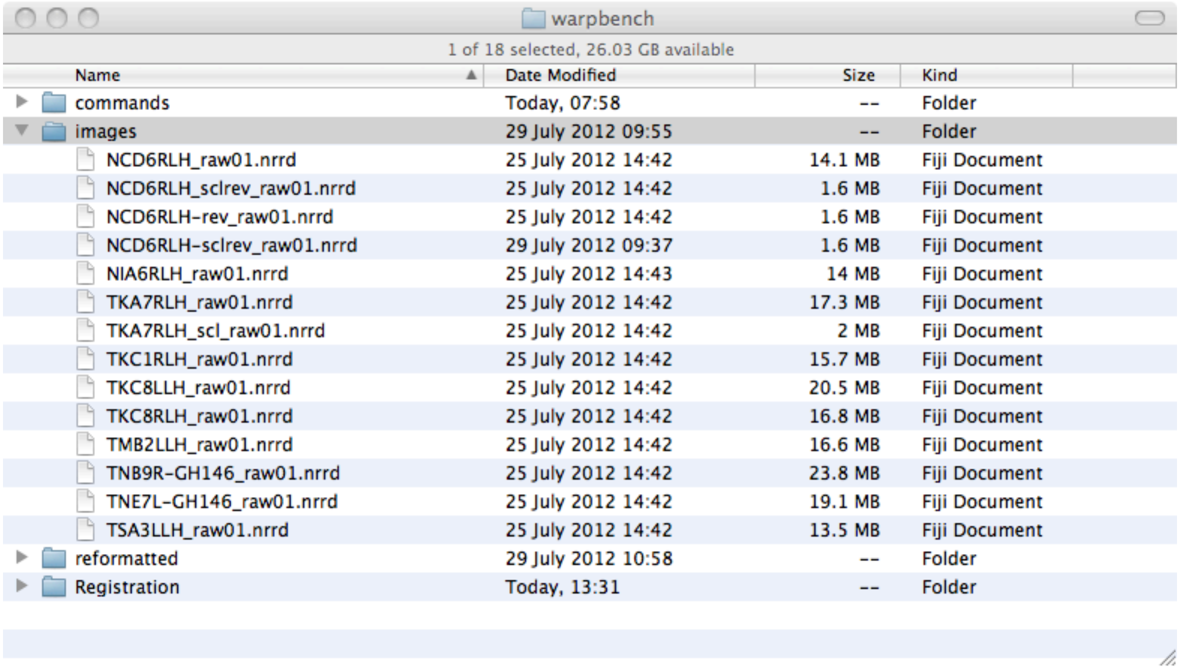
1. To install download the IGSRegistrationTools.zip file and expand by double clicking. I recommend copying the resulting IGSRegistrationTools directory to your Applications directory. All the required programs and scripts are bundled inside the applications (you can right click and choose "Show package contents" to see what's inside).
2. To prepare input images and inspect output images, install the Fiji distribution of ImageJ from http://fiji.sc/.

Preparing for Registration

To use the GUI registration (and substantially simplify your life), your input files need to obey certain conventions.

1. They should be all be saved in nrrd format using the Nrrd_Writer ImageJ plugin that I have written. Make sure that the spatial calibration information is correct in all 3 axes when you save. You can save a lot of space in sparse images by making the Nrrd_Writer plugin save the data in compressed format. To do this press the Alt key *before and while* selecting the "File … Save As … Nrrd …" menu option.
2. Separate files are used for each image channel (red, green, blue ...). The filenames must follow the Biorad convention by ending in 01.nrrd, 02.nrrd, 03.nrrd. Use simple alphanumeric names only - avoid underscores.
3. Make a folder to contain your registration work
4. Make a subsidiary folder names **images** (all lower case) and put your nrrd files inside. Organise them in subfolders if you want; this folder hierarchy will be regenerated in the output directories.

e.g. here I have made a directory called warpbench in which I will do some registration work.



warpbench

1 of 18 selected, 26.03 GB available

Name	Date Modified	Size	Kind
▶ commands	Today, 07:58	--	Folder
▼ images	29 July 2012 09:55	--	Folder
NCD6RLH_raw01.nrrd	25 July 2012 14:42	14.1 MB	Fiji Document
NCD6RLH_sclrev_raw01.nrrd	25 July 2012 14:42	1.6 MB	Fiji Document
NCD6RLH-rev_raw01.nrrd	25 July 2012 14:42	1.6 MB	Fiji Document
NCD6RLH-sclrev_raw01.nrrd	29 July 2012 09:37	1.6 MB	Fiji Document
NIA6RLH_raw01.nrrd	25 July 2012 14:43	14 MB	Fiji Document
TKA7RLH_raw01.nrrd	25 July 2012 14:42	17.3 MB	Fiji Document
TKA7RLH_scl_raw01.nrrd	25 July 2012 14:42	2 MB	Fiji Document
TKC1RLH_raw01.nrrd	25 July 2012 14:42	15.7 MB	Fiji Document
TKC8LLH_raw01.nrrd	25 July 2012 14:42	20.5 MB	Fiji Document
TKC8RLH_raw01.nrrd	25 July 2012 14:42	16.8 MB	Fiji Document
TMB2LLH_raw01.nrrd	25 July 2012 14:42	16.6 MB	Fiji Document
TNB9R-GH146_raw01.nrrd	25 July 2012 14:42	23.8 MB	Fiji Document
TNE7L-GH146_raw01.nrrd	25 July 2012 14:42	19.1 MB	Fiji Document
TSA3LLH_raw01.nrrd	25 July 2012 14:42	13.5 MB	Fiji Document
▶ reformatted	29 July 2012 10:58	--	Folder
▶ Registration	Today, 13:31	--	Folder

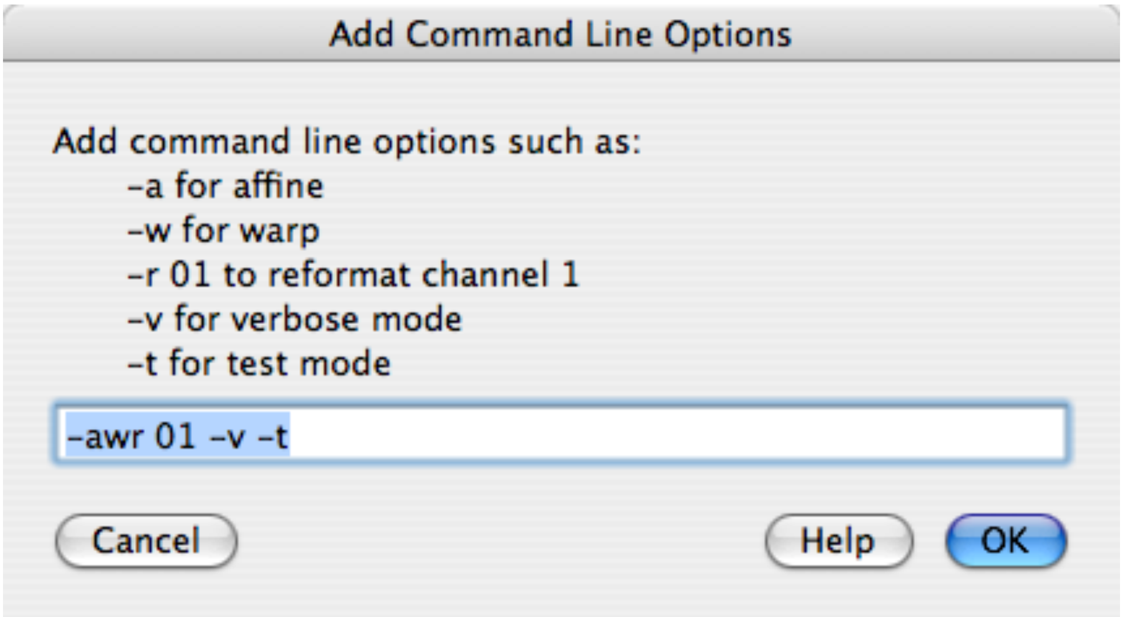
Running Registrations

The registration process involves

1. Choosing a set of input images (sample brains)
2. Choosing a reference brain
3. Selecting what you want to do with these brains (see **Registration Options** below)
4. Running the registration
5. *OR* inspecting the command line for later use in the Terminal application, in your own scripts etc.

Registration Options

This GUI is still pretty basic, so at one point you will need to provide an option string to control what the registration will actually do.



Add Command Line Options

Add command line options such as:
-a for affine
-w for warp
-r 01 to reformat channel 1
-v for verbose mode
-t for test mode

-awr 01 -v -t

The basic decisions that you have to make are:

1. Whether to do both affine (linear) and warp (nonrigid) registration
2. Whether to reformat the sample images into the reference coordinate system
3. ... and if so which image channel(s) should be reformatted
4. Whether to run a test to see what would happen

There are many additional switches that control the registration that can be added. For a full list see **Full Registration Options** below.

Registration Option Examples

A typical set might be

```
-a -w -r 01
```

this would run the **a**ffine **w**arp and **r**eformat commands on channel 01. Channel 01 will also be reformatted.

If channel 02 contains the images that will be registered, then add -c 02 eg:

```
-aw -r 01 -c 02
```

The order of options is largely unimportant; multiple options can be combined (ie -aw = -a -w)

If you want to reformat channel 02 and 03 images after previously running registrations on channel 01 then

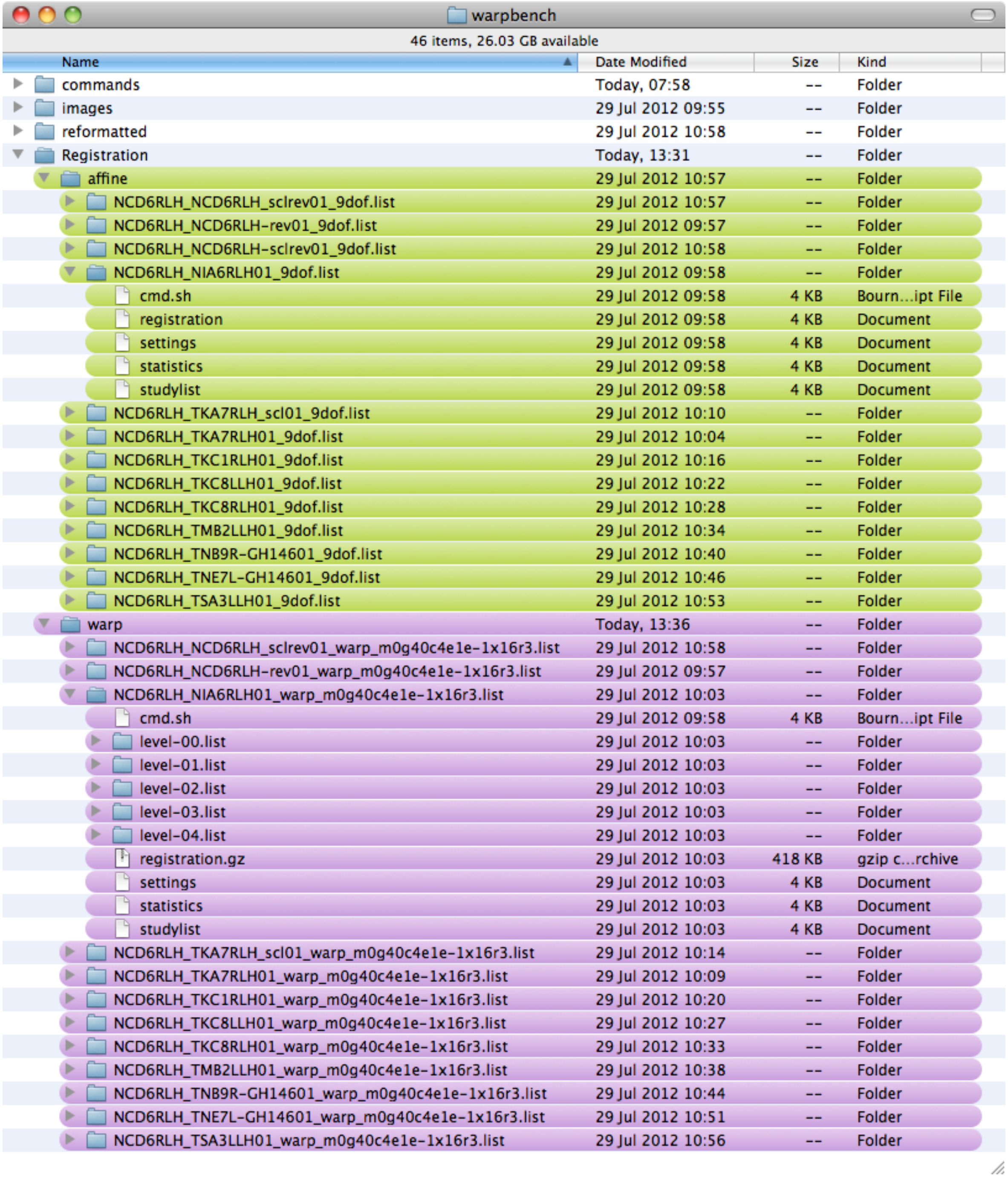
```
-r 0203 -f 01
```

If you want to check to see what registrations have happened:

```
-u
```

Output - Registration

The output of the registration/reformatting will go into 2 directories at the same level as the images folder, **Registration** and **reformatted**. Each registration is written out into a folder with a unique name reflecting the reference and sample brains and some of the registration parameters and ending in the suffix "list". Here is an example:



warpbench

46 items, 26.03 GB available

Name	Date Modified	Size	Kind
commands	Today, 07:58	--	Folder
images	29 Jul 2012 09:55	--	Folder
reformatted	29 Jul 2012 10:58	--	Folder
Registration	Today, 13:31	--	Folder
affine	29 Jul 2012 10:57	--	Folder
NCD6RLH_NCD6RLH_sclrev01_9dof.list	29 Jul 2012 10:57	--	Folder
NCD6RLH_NCD6RLH-rev01_9dof.list	29 Jul 2012 09:57	--	Folder
NCD6RLH_NCD6RLH-sclrev01_9dof.list	29 Jul 2012 10:58	--	Folder
NCD6RLH_NIA6RLH01_9dof.list	29 Jul 2012 09:58	--	Folder
cmd.sh	29 Jul 2012 09:58	4 KB	Bourn...ipt File
registration	29 Jul 2012 09:58	4 KB	Document
settings	29 Jul 2012 09:58	4 KB	Document
statistics	29 Jul 2012 09:58	4 KB	Document
studylist	29 Jul 2012 09:58	4 KB	Document
NCD6RLH_TKA7RLH_scl01_9dof.list	29 Jul 2012 10:10	--	Folder
NCD6RLH_TKA7RLH01_9dof.list	29 Jul 2012 10:04	--	Folder
NCD6RLH_TKC1RLH01_9dof.list	29 Jul 2012 10:16	--	Folder
NCD6RLH_TKC8LLH01_9dof.list	29 Jul 2012 10:22	--	Folder
NCD6RLH_TKC8RLH01_9dof.list	29 Jul 2012 10:28	--	Folder
NCD6RLH_TMB2LLH01_9dof.list	29 Jul 2012 10:34	--	Folder
NCD6RLH_TNB9R-GH14601_9dof.list	29 Jul 2012 10:40	--	Folder
NCD6RLH_TNE7L-GH14601_9dof.list	29 Jul 2012 10:46	--	Folder
NCD6RLH_TSA3LLH01_9dof.list	29 Jul 2012 10:53	--	Folder
warp	Today, 13:36	--	Folder
NCD6RLH_NCD6RLH_sclrev01_warp_m0g40c4e1e-1x16r3.list	29 Jul 2012 10:58	--	Folder
NCD6RLH_NCD6RLH-rev01_warp_m0g40c4e1e-1x16r3.list	29 Jul 2012 09:57	--	Folder
NCD6RLH_NIA6RLH01_warp_m0g40c4e1e-1x16r3.list	29 Jul 2012 10:03	--	Folder
cmd.sh	29 Jul 2012 09:58	4 KB	Bourn...ipt File
level-00.list	29 Jul 2012 10:03	--	Folder
level-01.list	29 Jul 2012 10:03	--	Folder
level-02.list	29 Jul 2012 10:03	--	Folder
level-03.list	29 Jul 2012 10:03	--	Folder
level-04.list	29 Jul 2012 10:03	--	Folder
registration.gz	29 Jul 2012 10:03	418 KB	gzip c...rchive
settings	29 Jul 2012 10:03	4 KB	Document
statistics	29 Jul 2012 10:03	4 KB	Document
studylist	29 Jul 2012 10:03	4 KB	Document
NCD6RLH_TKA7RLH_scl01_warp_m0g40c4e1e-1x16r3.list	29 Jul 2012 10:14	--	Folder
NCD6RLH_TKA7RLH01_warp_m0g40c4e1e-1x16r3.list	29 Jul 2012 10:09	--	Folder
NCD6RLH_TKC1RLH01_warp_m0g40c4e1e-1x16r3.list	29 Jul 2012 10:20	--	Folder
NCD6RLH_TKC8LLH01_warp_m0g40c4e1e-1x16r3.list	29 Jul 2012 10:27	--	Folder
NCD6RLH_TKC8RLH01_warp_m0g40c4e1e-1x16r3.list	29 Jul 2012 10:33	--	Folder
NCD6RLH_TMB2LLH01_warp_m0g40c4e1e-1x16r3.list	29 Jul 2012 10:38	--	Folder
NCD6RLH_TNB9R-GH14601_warp_m0g40c4e1e-1x16r3.list	29 Jul 2012 10:44	--	Folder
NCD6RLH_TNE7L-GH14601_warp_m0g40c4e1e-1x16r3.list	29 Jul 2012 10:51	--	Folder
NCD6RLH_TSA3LLH01_warp_m0g40c4e1e-1x16r3.list	29 Jul 2012 10:56	--	Folder

Notice a few things.
1) There is an affine and warp registration folder for each brain. Normally you would just use the warp registration folder for subsequent work. I have opened up the folders for the first brain, NCD6RHLH. The critical files in the affine and warp folders are called **registration** and **registration.gz**, respectively. The warp registration has been gzipped because it is quite large. Notice also that the warp registration contains several subfolders because it proceeds through several levels of warping operating initially on a coarsely resampled image and eventually on the fine resolution image.

Output - Reformat

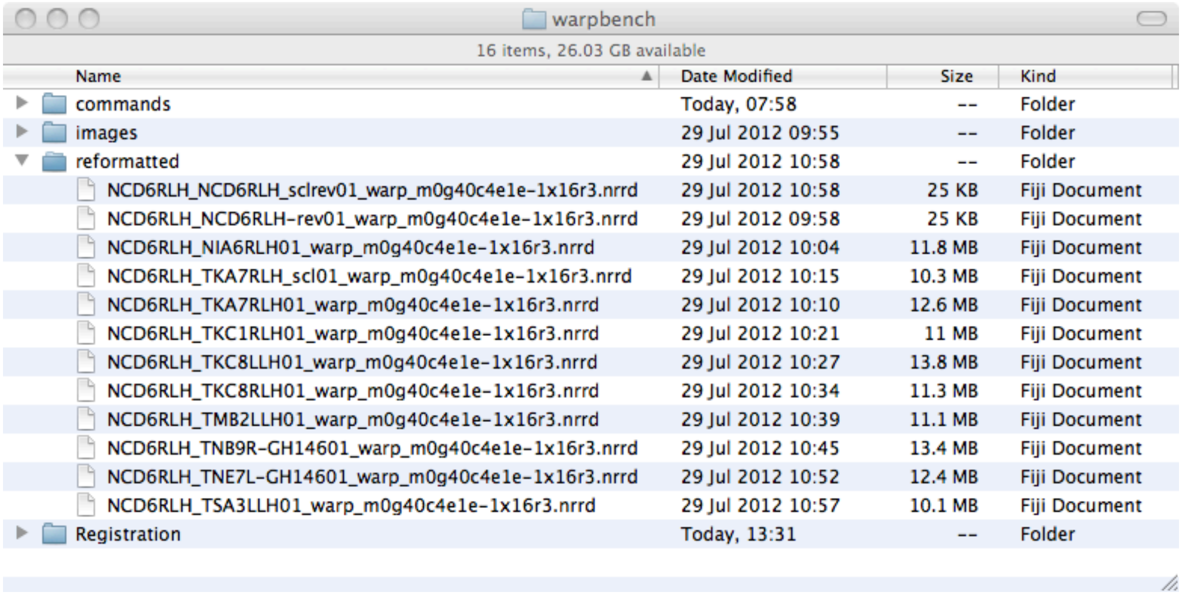
The default format for reformatted images is nrrd (as for input). For example the registration:

average-goodbrains_NCD6RLH01_warp_m0g40c4e1e-1x16r3.list

has resulted in the image:

average-goodbrains_NCD6RLH01_warp_m0g40c4e1e-1x16r3.nrrd

The nrrd files contains spatial calibration information etc in a text header followed by a data chunk. You can open the file in ImageJ/Fiji.



warpbench

16 items, 26.03 GB available

Name	Date Modified	Size	Kind
commands	Today, 07:58	--	Folder
images	29 Jul 2012 09:55	--	Folder
reformatted	29 Jul 2012 10:58	--	Folder
NCD6RLH_NCD6RLH_sclrev01_warp_m0g40c4e1e-1x16r3.nrrd	29 Jul 2012 10:58	25 KB	Fiji Document
NCD6RLH_NCD6RLH-rev01_warp_m0g40c4e1e-1x16r3.nrrd	29 Jul 2012 09:58	25 KB	Fiji Document
NCD6RLH_NIA6RLH01_warp_m0g40c4e1e-1x16r3.nrrd	29 Jul 2012 10:04	11.8 MB	Fiji Document
NCD6RLH_TKA7RLH_scl01_warp_m0g40c4e1e-1x16r3.nrrd	29 Jul 2012 10:15	10.3 MB	Fiji Document
NCD6RLH_TKA7RLH01_warp_m0g40c4e1e-1x16r3.nrrd	29 Jul 2012 10:10	12.6 MB	Fiji Document
NCD6RLH_TKC1RLH01_warp_m0g40c4e1e-1x16r3.nrrd	29 Jul 2012 10:21	11 MB	Fiji Document
NCD6RLH_TKC8LLH01_warp_m0g40c4e1e-1x16r3.nrrd	29 Jul 2012 10:27	13.8 MB	Fiji Document
NCD6RLH_TKC8RLH01_warp_m0g40c4e1e-1x16r3.nrrd	29 Jul 2012 10:34	11.3 MB	Fiji Document
NCD6RLH_TMB2LLH01_warp_m0g40c4e1e-1x16r3.nrrd	29 Jul 2012 10:39	11.1 MB	Fiji Document
NCD6RLH_TNB9R-GH14601_warp_m0g40c4e1e-1x16r3.nrrd	29 Jul 2012 10:45	13.4 MB	Fiji Document
NCD6RLH_TNE7L-GH14601_warp_m0g40c4e1e-1x16r3.nrrd	29 Jul 2012 10:52	12.4 MB	Fiji Document
NCD6RLH_TSA3LLH01_warp_m0g40c4e1e-1x16r3.nrrd	29 Jul 2012 10:57	10.1 MB	Fiji Document
Registration	Today, 13:31	--	Folder

Running, Interrupting and Rerunning RegistrationRunner

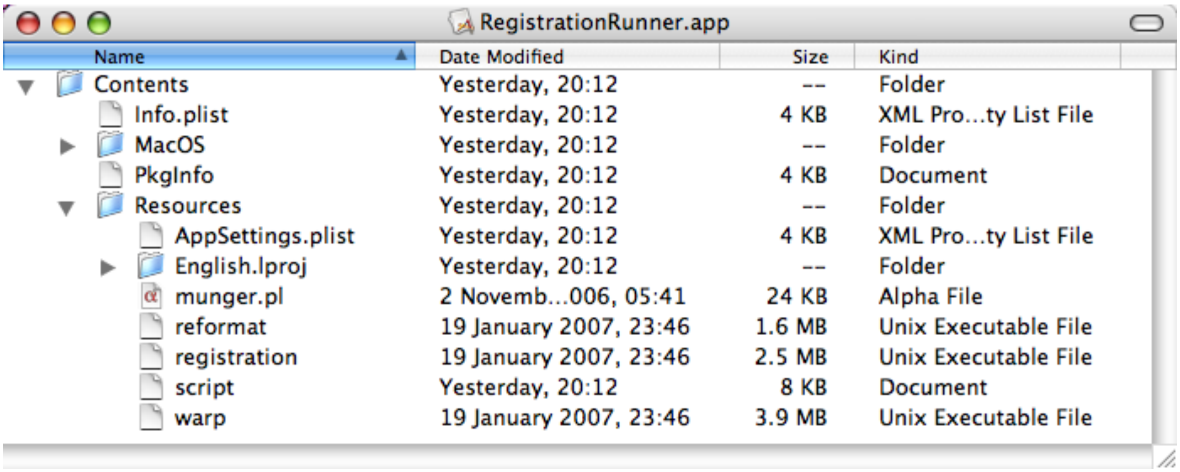
You can cancel a run at any time however the software is not smart enough to clean up afterwards. In particular, look out for **.lock** files. These are placeholder files that are created to indicate that a brain is in progress. If you interrupt a run you will have to delete all these .lock files before you try and rerun **RegistrationRunner**.

At the moment you cannot run more than one copy of **RegistrationRunner**. However, the underlying applications are multi-threaded. This means that if you have an 8 or 12 processor Mac (recommended) then the CPU intensive programs (the main affine and warp registrations) will run across all four CPUs.

Instead of just double clicking on **RegistrationRunner**, you can also run it by dragging and dropping a registration directory onto the application icon. However since this bypasses all the dialog boxes, this is only useful if you have correctly set the

Customising RegistrationRunner

You can do a basic customisation of **RegistrationRunner** with minimal knowledge of perl. Right click on the Application icon and choose "Show Package Contents". You'll get something like this:



RegistrationRunner.app

Name	Date Modified	Size	Kind
▼ Contents	Yesterday, 20:12	--	Folder
Info.plist	Yesterday, 20:12	4 KB	XML Pro...ty List File
▶ MacOS	Yesterday, 20:12	--	Folder
PkgInfo	Yesterday, 20:12	4 KB	Document
▼ Resources	Yesterday, 20:12	--	Folder
AppSettings.plist	Yesterday, 20:12	4 KB	XML Pro...ty List File
▶ English.lproj	Yesterday, 20:12	--	Folder
munger.pl	2 Novemb...006, 05:41	24 KB	Alpha File
reformat	19 January 2007, 23:46	1.6 MB	Unix Executable File
registration	19 January 2007, 23:46	2.5 MB	Unix Executable File
script	Yesterday, 20:12	8 KB	Document
warp	19 January 2007, 23:46	3.9 MB	Unix Executable File

The relevant files are called **script** and **munger.pl**. script shows the dialog boxes and sends the results to munger which coordinates the registration process. If you want to edit the default registration options (as discussed in **Registration Options** section above**).**

script is a fairly simple perl script and you should be able to edit it at a few locations to alter the default behaviour of RegistrationRunner. In particular modify the line:

    $MAINARGS="-awr 01 -v ";

to specify your preferred default set of registration options. e.g.

    $MAINARGS="-awr 02 "; # trailing space is significant

munger is pretty complicated, but there are a few defaults early on in the script that you could alter without too much effort. In particular feel free to edit the line:

    my $referenceImage=$ENV{"HOME"}."/projects/ReferenceImages/average-goodbrains-warp40-5_e1e-2.study";

so that it is set to your most commonly used reference brain. $ENV{"HOME"} is replaced by the location of the currently active user's home directory e.g. "/Users/joe".

Full Registration Options

You can also see all these command line options by dropping an empty folder on the RegistrationRunner icon.

```
Usage: /Users/jefferis/bin/munger.pl [OPTIONS] <PICFILE/DIR> [<PICFILE2/
DIR2> ...]
Version: 1.19

    A wrapper script for CMTK registration software. For more on CMTK see:
      http://www.nitrc.org/projects/cmtk/

    -h print this help
    -v verbose (provide extra feed back at runtime)
```

```
     -t test run (no commands are actually run)
     -g debug: prints every command run by myexec and the return value
     -p make a scriPt from commands that would be run
        (nb cannot produce commands that depend on earlier commands)
     -u statUs - display number of images, registrations etc
     -z turn gzip off (on by default)
     -k lock message ie contents of lock file (defaults to hostname:process
id)
     -m maximum time to keep starting registrations (in hours, default
8760=1y)
        nb this will not stop any running registrations
     -x [never|only|any]:[truncate|delete] Clear input images when done.
        default is never, eg only:truncate, any:delete
        only => only the job that runs reformat can clear input
        any  => any job can delete; in fact a job started with any will not
                reformat at all. This is useful because you can run a
cleanup
               job if it becomes clear that you are running short of space.
        truncate => leave a 0 length file with same mtime as original image

     -a run affine transform
     -w run warp transform
     -c [01|02|..] channels for registration (default 01 or "")
     -r [01|02|..] run reformat on these channels
     -l [p|a|0..9|f] run reformat on these levels
        (default f=final warp, p=principal axis, a=affine, 0..9=warp
intermediates)
     -f [01|02|..] channel of the images used for registration - default is
01

     [nb use -f to specify the channel of the images that were previously
used to
      generate a registration if you now want to reformat a different
channel using
      that registration information]

     -i register brain to itself if possible (default is to skip)
     -0 Don't output intermediate warp registration levels

     -s [file|fileStem] Reference brain (average e-2 by default)
     -b [path] bin directory
     -d [stem] registration subdirectory (default ./Registration)
        [nb if this begins in a dot then the value will be appended to both
        reformatted and Registration directories]
     -F Read path to input files, one per line, from <PICFILE1> ...
        NB path should be absolute or relative to registration directory
e.g.
          images/imagea-01.nrrd
          images/imageb-01.nrrd

     -e File ending of input images (pic, nrrd, nhdr)
```

```
    -o File ending of output images (bin, nrrd, nhdr) - defaults to nrrd

    -H use Amira's Generalised Hough Transform to initialise affine
registration when available
       (nb the registration folder should be called Registration/affine/
XXX_ght.list)
    -P find initial affine transform using image principal axes
    -L find initial affine transform using landmarks
    -I inverse consistent warp weight (--ic-weight) default 0, try 1e-5
    -E [energy] energy of warp transform (default e-1)
    -X [exploration] (default 16)
    -M [metric] (Supported values: nmi, mi, cr, msd, ncc, default is nmi)
      See warp --help for details
    -C [coarsest] (default 4)
    -G [grid-spacing] (default 40)
    -R [refine] (default 3)
    -J [0 to 1] jacobian-weight volume constraining param (default 0)
    -T [threads] (default auto)

    -A [option] additional options for affine transformation
    -W [option] additional options for warp transformation
```

Acknowledgements

These simple applications were built using the **Platypus** script wrapper tool and **CocoaDialog**. Many thanks to the authors of those programs.

Author

(c) January 2012
Gregory Jefferis, Division of Neurobiology, MRC Laboratory of Molecular Biology, Cambridge, UK.
www.jefferislab.org