How to run a registration tool using MICe-build-model.pl

Kitti Koonsanit

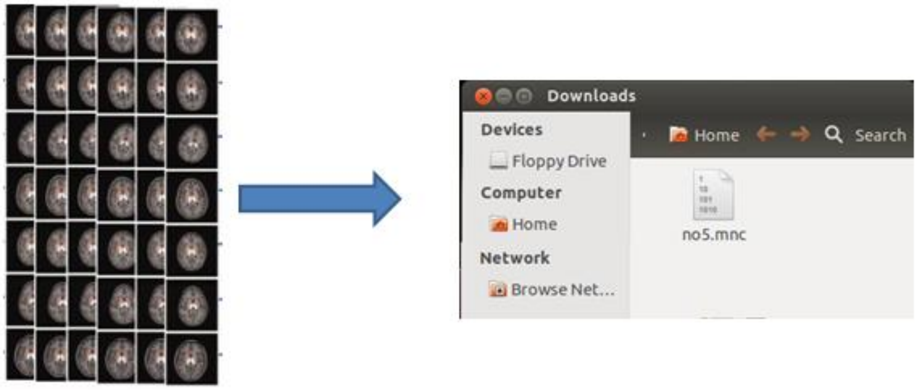
17 January 2013

How to run a registration tool using MICe-build-model.pl

MICe-build-model.pl is a software package optimized to perform image registration on mouse brains. Registration may be defined as the fitting of two or more images to an average image.

3 commands for Mouse Registration (convert registration and display)

1) convert dcm2mnc

NAME	dcm2mnc – The dcm2mnc command is used to convert DICOM format files to MINC format.
SYNOPSIS	**`dcm2mnc`** `[<options>]` `<input-list>` `<output-dir>`
Example	dcm2mnc XY*.dcm /home/sc431137/Downloads/120419/dest/
	 Downloads Devices Floppy Drive Computer Home Network Browse Net... Home Search no5.mnc
	Dataset for Test http://www.mouseimaging.ca/technologies/mouse_embryo_atlas.html

2) MICe-build-model (Registration)

NAME

MICe-build-model
– MICe-build-model.pl is a software package optimized to perform image registration on mouse images.

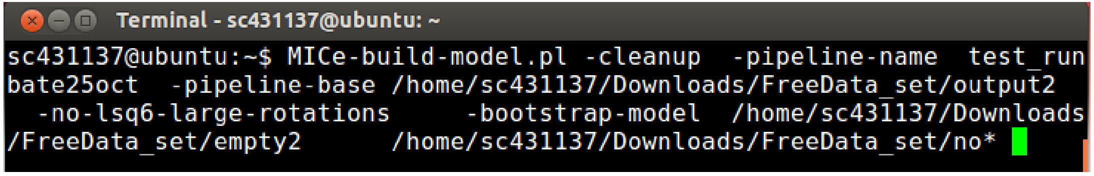
SYNOPSIS

```
MICe-build-model.pl -pipeline-name name_pipeline
-pipeline-base   /path/to/new/output/
-no-lsq6-large-rotations
-bootstrap-model /path/to/new/tempfolder/
mouse1.mnc mouse2.mnc mouse3.mnc mouse4.mnc
```

Dataset for Test
http://www.mouseimaging.ca/technologies/mouse_embryo_atlas.html

Example

MICe-build-model.pl -cleanup -pipeline-name *test_runbate25oct*
-pipeline-base /home/sc431137/Downloads/FreeData_set/output2
-no-lsq6-large-rotations
-bootstrap-model /home/sc431137/Downloads/FreeData_set/empty2
/home/sc431137/Downloads/FreeData_set/no*.mnc

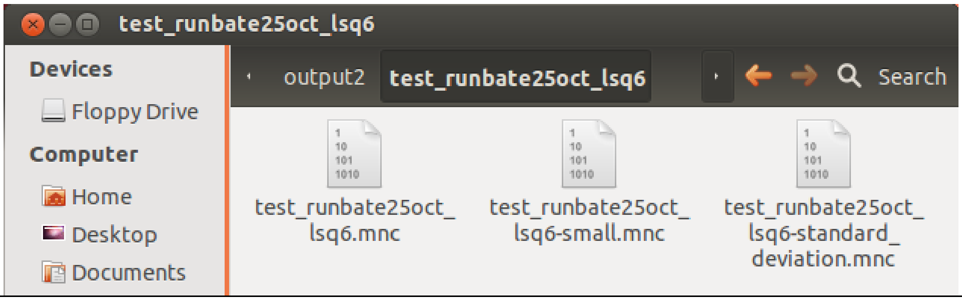


```
sc431137@ubuntu:~$ MICe-build-model.pl -cleanup  -pipeline-name  test_run
bate25oct  -pipeline-base /home/sc431137/Downloads/FreeData_set/output2
  -no-lsq6-large-rotations     -bootstrap-model  /home/sc431137/Downloads
/FreeData_set/empty2      /home/sc431137/Downloads/FreeData_set/no*
```

Output Folders

PROJECT_NAME_lsq6: Contains the average result following 6 parameter alignment.

test_runbate25oct_lsq6/
test_runbate25oct_lsq6.mnc = average-result.mnc
test_runbate25oct_lsq6-small.mnc = small-result.mnc
test_runbate25oct_lsq6-deviation.mnc = standard deviations away from the average result



How to run MICe-build-model perl script

MICe-build-model.pl is a software package optimized to perform image registration on mouse brains. Registration may be defined as the fitting of two or more images to an average image. A difference between the average image and the individual images allows us to calculate mean differences between 2 or more groups of mice. Some examples of groups of mice which may be compared are
(1) Mice homozygous for mutation A, heterozygous for mutation A, and wild type mice
(2) Mice homozygous for mutation A and mice homozygous for mutation B (where it remains to be determined whether gene A and B perform the same function)
(3) Mice that are given a specific treatment versus those that are not. Differences between groups may also be examined by determining the precise location of differences and the amount of difference.

An assumption in registration is that every point in a brain can be mapped to a point in another brain. Without this, registration fails. This is why we cannot perform registration on brain images that are drastically different from each other. There are a number of problems that can creep up which may result in a failure to register. These are detailed in the 'Troubleshooting' section.

Special Note: Although the parameters of the software are specifically set for mouse brains, MICe-build-model.pl may also be used for embryos (tail may cause problems) , whole mouse body, human brain, etc.. It is not suitable for organs for which homology is indeterminable (e.g. vasculature).

Setting up

At its most basic MICe-build-model needs just two or more input MINC files, the name of the pipeline, and it's ready to go. In most situations, however, an initial model used for rigid body alignment and masking of areas of interest is highly recommended.

MICe-build-model is run by executing a single script called, not surprisingly, MICe-build-model. There's a large set of options which control how MICe-build-model runs; anyone wanting all the gory choices should take a look at:

```
MICe-build-model.pl -help
```

Described in the rest of this page are the most common options and general usage patterns.

General settings

During a registration process there are hundreds of processes being executed, many of which can be run in parallel. For this reason, the MICe-build-model script gives you the option to run the pipeline on one machine, but also to use a sun grid engine (sge) in order to exploit the parallel nature of the registration process.

Option	Argument	Description
-spawn	(none)	run the processes (jobs) sequentially on your machine (default).
-sge	(none)	use the sun grid engine (SGE) to run the processes.

6 Basic Steps

There are 6 steps carried out in MICe-build-model.pl. A basic overview is provided below:

1. Rigid alignment, referred to as 6 parameter alignment, consists of rotating and sliding an image to fit a model. The model may be a pre-existing model or a model created through bootstrapping. Using a pre-existing model carries the advantages of saving computation time, saving disk space, and also results in data that is in the desired orientation. In bootstrapping, a model is created out of your dataset. Once 6 parameter alignment is complete, all brain images are now in the same space.

2. The first level of improvement is now carried out. Affine alignment, also known as 12 parameter alignment, consists not only of rotating and translating an image, but also scaling and shearing an image to produce an average. It is carried out in a pairwise manner.

3. Non-linear alignment is then carried out in which the remaining differences are dealt with. Initially, the atlas model from the 12 parameter alignment is used as a target. Then, progressively (from non-linear atlas model 1 to non-linear atlas model 6), improvements are carried out until there is a final average model. The transformation begins with a blurry image that is gradually clarified as the linear transformation proceeds. Settings for this portion of the script are configurable for images with higher or lower resolution.

4. Add segmentation: A segmented atlas brain (in which individual brain structures have been defined) is used to add segmentation to the average brain.

5. Backpropagate segmentation: Segmentation added to the average brain is now transferred to the individual brains.

6. The intensities are stored in an output folder along with the following variables (which are calculated for each brain image): displacement, magnitude, scaled jacobians, and jacobians. These variables are useful in analyzing the data.

Step #1 - 6 Parameter Registration

The first part of running MICe-build-model is the rigid body alignment. There are a few choices to be made here: whether or not to use an initial model (and if so, which one), and which strategy to employ for bringing the images into the same space. The choice of initial model is specified by:

Option	Argument	Description
-bootstrap-model	empty directory	Use this option when you do not have an initial model. The directory specified will be created if it does not exist already. Make sure to specify an absolute (not relative) path.
-init-model	basename of initial model	Use this option when you have an initial model. See examples below for how to specify the initial model.
-pipeline-name	prefix for pipeline folders	A registration pipeline will create a number of directories. This option is used to give a distinguishing name to that set of folders. For instance if you have a set of MRI brains images from a project called sonic-hedgehog, you could use mri-brains-sonic-hedgehog for the -pipeline-name option.

For example, to run without an initial model using four mice you'd use the following command:

```
MICe-build-model.pl -pipeline-name simple_pipeline -no-lsq6-large-rotations -
bootstrap-model /path/to/new/directory/ mouse1.mnc mouse2.mnc mouse3.mnc
mouse4.mnc
```

The mouse1.mnc ... mouse4.mnc would, of course, be replaced by the actual filenames of your images. The -pipeline-name option takes a string giving the name of the pipeline, which will be used when creating the necessary output directories and files. It'll create the pipeline in the directory from which you launched the command; if you want the data to be written elsewhere, an optional -pipeline-base /path/to/output can be specified.

Using an initial model would result in the equivalent command being:

```
MICe-build-model.pl -pipeline-name simple_pipeline -init-model
/path/to/init/model/basename mouse1.mnc mouse2.mnc mouse3.mnc mouse4.mnc
```

After the choice of whether to use an initial model you are now faced with three options for how to perform the linear registration:

Option	Argument	Description
-lsq6-large-rotations	(none)	This is the default - it uses a fairly time consuming process, but allows for the largest variance in rotations between the input images. Recommended for the scanning of fixed samples in particular, where ensuring equivalent positioning across scans is difficult or impossible.
-no-lsq6-large-rotations	(none)	Much faster, but more likely to incorrectly handle images with large rotations. Recommended for in-vivo brain scans or similar acquisitions where the initial positioning of the animals is fairly consistent.
-lsq6-identity	(none)	An add-on option for -no-lsq6-large-rotations, it assumes that the position in world space between all the scans is similar. If it is not similar (i.e. data from different scanners, etc.) then a centre of gravity estimation needs to be performed first, which is done with ...
-no-lsq6-identity	(none)	... does not assume equivalent position in world space, and performs centre of gravity estimation first. This is the default. Remember that these two options are ignored when -lsq6-large-rotations is in use
-lsq6-kernels	e.g. 1.0_blur, 0.5_blur, 0.3_blur	Comma separated list of kernels (default depends on other -lsq6 choices). Altering the choice of kernels can sometimes improve registration.
-registration-accuracy	(none)	Performs a registration accuracy test on the pipeline. Locates brain images that are 2, 3, and 4 standard deviations away from the average brain in terms of intensities and number of deformations.This information is useful in determining the cause of failure when registration cannot be completed.

Step #2 - 12 Parameter Pairwise Registration

After the rigid body alignment, a full affine alignment stage can be run, where each file is aligned to each of the other file using 12 parameters (3 translations, 3 rotations, 3 scales, and 3 shears). In the end, an average affine transformation is created from the individual pairwise registrations. This stage is run by adding the -lsq12 flag to the parameters to MICe-build.model.pl.

In the case of the bootstrap mode, this is how to continue:

```
MICe-build-model.pl -pipeline-name simple_pipeline -no-lsq6-large-rotations -
bootstrap-model /path/to/new/directory/ mouse1.mnc mouse2.mnc mouse3.mnc
mouse4.mnc -lsq12
```

and when using the initial mode:

```
MICe-build-model.pl -pipeline-name simple_pipeline -init-model
/path/to/init/model/basename mouse1.mnc mouse2.mnc mouse3.mnc mouse4.mnc -
lsq12
```

Step #3 - Non-linear Registration

Option	Argument	Description
-nlin-partial	none	Run only the first two non-linear fitting steps. This is much faster than running all of the steps. If errors arise while running the first two, corrections may be made before proceeding, thereby saving the user a significant amount of time.
-nlin-stats	none	Create volumes for analysis of deformation fields. Jacobians, scaled jacobians, and magnitudes are deposited into folders.
-stiffness	number	Less stiffness allows greater flexibility when performing deformations of brain images. Specify stiffness as a number; the default is 0.98.
-similarity	number	Specify similarity as a number; the default is 0.8.
-weight	number	This is the amount by which the deformation is underestimated. Specify weight as a number; the default is 0.8.
-nlin-protocol	file	File containing the nonlinear fitting protocol.

Output folders of mouse registration processes

Below is a summary of the data contained in the output folders. 'PROJECT_NAME' will differ depending upon the name provided by the user when running MICe-build-model.

1. PROJECT_NAME_lsq6: Contains the average brain following 6 parameter alignment.

2. PROJECT_NAME_lsq12: Contains the average brain following 12 parameter alignment.

3. PROJECT_NAME_nlin: Contains the final average brain following non-linear alignment.

4. PROJECT_NAME_resampled_atlas: Contains the resampled atlas.

5. PROJECT_NAME_registration_accuracy:

6. PROJECT_NAME_processed: Contains all of the transformations and deformations for each individual brain.

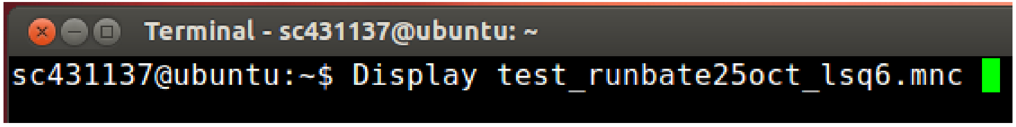
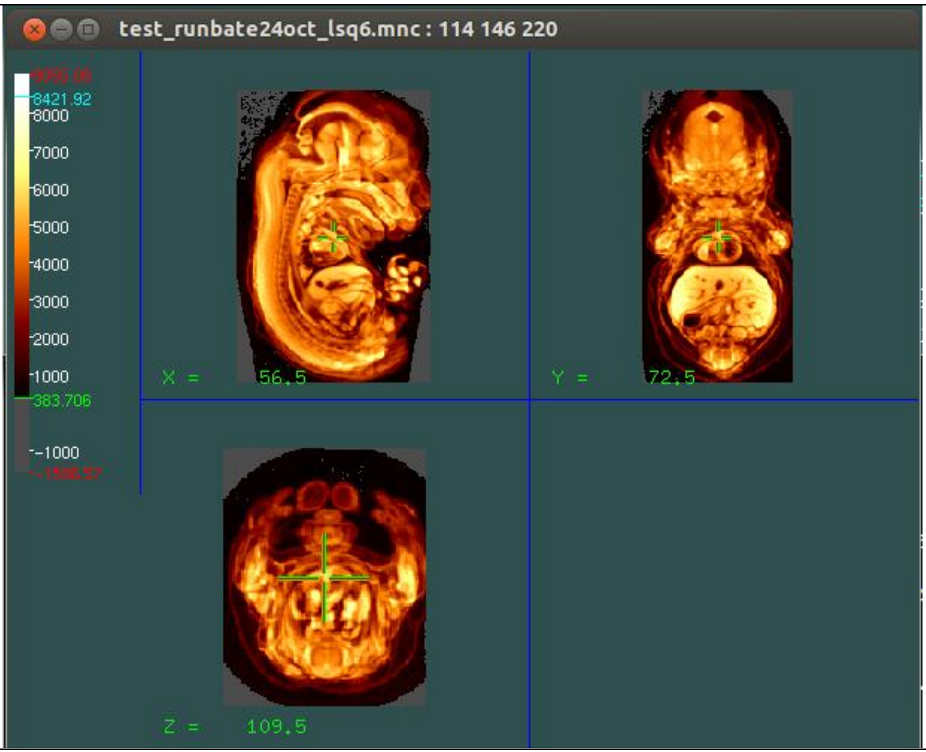
native.mnc The full size file in the same orientation as the native files. This file exists for reference purposes only and is never actually used in the registration process.

native-small.mnc The downsampled version of native.mnc. Used in the lsq6 stage of the registration process. Multiple blurs of this file are created during registration.

References

https://wiki.phenogenomics.ca/display/MICePub/MiceBuildModelUsage
https://wiki.phenogenomics.ca/plugins/viewsource/viewpagesrc.action?pageId=1867935

3. Display Tool

NAME	Display • Display is a hugely visualization program. Its main purposes are segmenting, visualizing/overlaying several MINC files and visualization of 3D objects.
SYNOPSIS	• Display volume1.mnc
Example	• Display volume1.mnc  Terminal - sc431137@ubuntu: ~ `sc431137@ubuntu:~$ Display test_runbate25oct_lsq6.mnc`
	 test_runbate24oct_lsq6.mnc : 114 146 220 8421.92 8000 7000 6000 5000 4000 3000 2000 1000 383.706 -1000 X = 56.5 Y = 72.5 Z = 109.5