

# MBM.py cookbook

- [Disk Clean up](#)
- [At HPF](#)
  - [Example: Registration of 108 in vivo images on hpf.](#)
  - [Example: 40 micron ex-vivo mouse brains on HPF](#)
  - [Example: Registration of > 1300 in vivo images on hpf using the two level model building](#)
  - [Example: In vivo images on hpf using registration chain](#)
- [At MICe](#)
  - [Example: Basic registration pipeline at MICe on 20 56um brains](#)
  - [Example: registration of embryo data at 27 micron resolution](#)
  - [Example: Registration of data with ambiguous shape and variable rotational position](#)
- [At SciNet](#)
  - [Example: Registration of 149 x 56 um brains on scinet.](#)
  - [Example: Registration of 143 x 40 um brains on scinet using modified nlin protocol and lsq6 alignment](#)

This page provides examples of MBM.py runs on different kinds of data.



Why is the --pipeline-name argument is so important...

The names of files and directories in a pipeline are eventually all based on the pipeline name. All files will end up in the {pipeline\_name}\_processed, {pipeline\_name}\_lsq6, ... etc directories. If you do not supply a pipeline name, the code will generate one based on the time you start your pipeline. For instance: pipeline-21-11-2016-at-16-11-33. This is fine if your pipeline finished without any issues, however... if for some reason you pipeline did not finish, and you need to restart your pipeline, you will most likely rerun the command you ran before. I.e., without the --pipeline-name argument. This means that once again, the pipeline will assign a pipeline name to your pipeline based on the current date/time, which will inevitably be different from the previously generated pipeline name, and as such you code thinks that nothing has been processed yet and will start your pipeline from scratch!

What to do when you forgot to supply a pipeline name and you want to restart your pipeline? Let's assume the example above and the following directories exist:

```
pipeline-21-11-2016-at-16-11-33_init_model
pipeline-21-11-2016-at-16-11-33_lsq12
pipeline-21-11-2016-at-16-11-33_lsq6
pipeline-21-11-2016-at-16-11-33_nlin
pipeline-21-11-2016-at-16-11-33_pipeline_stages.txt
pipeline-21-11-2016-at-16-11-33_processed
...
```

You can use this to deduce the pipeline name that was given to that pipeline by the code (the common prefix to all these file/directory names) and when you rerun it, supply it as follows:

```
MBM.py --pipeline-name=pipeline-21-11-2016-at-16-11-33 [rest of your command]
```



What does --maget-no-mask really mean? Am I not using masks in my pipeline???

No no... MAGEt can be used to generate custom masks of your input files. These masks should fit your input data better than the mask that comes with the initial model, and should improve for instance the estimation of the linear part of the transformation out of the entire transformation. If you specify --maget-no-mask those masks are not created, and instead the mask from the initial model is used. This is the same as what has happened in our pipelines so far (i.e. the last many years)

## Disk Clean up

When you're finished with a registration, have the data analysed and are ready to archive the pipeline, you can remove more than just the tmp directories from the pipeline. See [these notes on how to perform a thorough disk clean up after a MBM run](#)

## At HPF

### Example: Registration of 108 in vivo images on hpf.

At time of writing, I used 8g of vmem because I kept getting kicked off the qlogin node.

1. Login to hpf
  - `ssh -A hpf.ccm.sickkids.ca`
2. Start cluster and load modules

```
qlogin -l vmem=8G,walltime=96:00:00
```

Please refer to the Pyd Piper on the SickKids HPF page for details on which modules to load:

[Pyd Piper on the SickKids HPF](#)

```
MBM.py \  
--pipeline-name=some_pipeline_name \  
--init-model=/hpf/largeprojects/MICe/tools/initial-models/PydPiper-MEMRI-90micron-saddle-july-2015  
/p65_MEMRI_mouse_brain.mnc \  
--registration-method=ANTS \  
--lsq6-centre-estimation \  
--lsq6-protocol=/hpf/largeprojects/MICe/tools/protocols/linear/PydPiper_abbreviated_minctracc_center_estimation.  
csv \  
--num-executors=108 \  
--lsq12-protocol=/hpf/largeprojects/MICe/tools/protocols/linear/PydPiper_testing_default_ls12.csv \  
--no-run-maget \  
--maget-no-mask \  
--files /hpf/largeprojects/MICe/vousdend/CREB_EE/mnc_files_live_dc/*.mnc # NB.: '--files=/*.*.mnc' causes  
the wildcard not to be expanded :|
```

## Example: 40 micron ex-vivo mouse brains on HPF

The final mincANTS stages for a 40 micron pipeline take more than 24 hours to finish. By default each job that is sent to the HPF cluster has a wall time limit of 24 hours, and this means that those stages are unable to actually finish. You'll have to use the --time flag to increase the wall time given to the executors on the cluster.

1. Login to hpf
  - ssh -A [hpf.ccm.sickkids.ca](http://hpf.ccm.sickkids.ca)
2. Start cluster and load modules

```
qlogin -l vmem=4G,walltime=96:00:00
```

Please refer to the Pyd Piper on the SickKids HPF page for details on which modules to load:

[Pyd Piper on the SickKids HPF](#)

```
MBM.py \  
--pipeline-name=some_pipeline_name \  
--init-model=/hpf/largeprojects/MICe/tools/initial-models/PydPiper-40-micron-basket-dec-2014  
/basket_mouse_brain_40micron.mnc \  
--nlin-protocol=/hpf/largeprojects/MICe/tools/protocols/nonlinear/PydPiper_mincANTS_standard_40_micron.pl \  
--lsq6-centre-estimation \  
--lsq6-protocol=/hpf/largeprojects/MICe/tools/protocols/linear/PydPiper_abbreviated_minctracc_center_estimation.  
csv \  
--num-executors=30 [general rule of thumb is to use as many as the number of subjects in your pipeline] \  
--max-failed-executors=31 \  
--time=48:00:00 \  
--lsq12-protocol=/hpf/largeprojects/MICe/tools/protocols/linear/PydPiper_testing_default_ls12.csv \  
--no-run-maget \  
--maget-no-mask \  
--files [*input files*].mnc
```

Run a MAGE T.py call after your MBM.py call on HPF using the 40 micron DSUR atlas ([Mouse Brain Atlases](#))

```
MAGeT.py \
--atlas-library=/hpf/largeprojects/MICe/tools/atlasses
/Dorr_2008_Steadman_2013_Ullmann_2013_Richards_2011_Qiu_2016_Egan_2015_40micron/ex-vivo \
--pipeline-name=full_MAGeT \
--max-templates=5 \
--num-executors=30 [general rule of thumb is to use as many as the number of subjects in your pipeline]\
--lsq12-protocol=/hpf/largeprojects/MICe/tools/protocols/linear/default_linear_MAGeT_prot.csv \
--nlin-protocol=/hpf/largeprojects/MICe/tools/protocols/nonlinear/default_nlin_MAGeT_minctracc_prot.csv \
--registration-method=minctracc \
--no-mask
--files [*lsq6 resampled files from MBM pipeline*].mnc
```

## Example: Registration of > 1300 in vivo images on hpf using the two level model building

At time of writing, I used 4G of vmem, but this could likely be reduced to 2G.

1. Login to hpf
  - `ssh -A hpf.ccm.sickkids.ca`
2. Start cluster and load modules

```
qlogin -l vmem=4G,walltime=96:00:00
```

Please refer to the Pydpiper on the SickKids HPF page for details on which modules to load:

[Pydpiper on the SickKids HPF](#)

**Note: Right now you also need to specify an lsq12 protocol, in addition to an nlin-protocol. Additionally, your csv file needs to have at least the following columns: group (aka how you want scans to be grouped, which is typically by mouse ID) and file).**

```
twolevel_model_building.py --num-executors=600
--init-model=/hpf/largeprojects/MICe/tools/initial-models/Pydpiper-MEMRI-90micron-saddle-july-2015
/p65_MEMRI_mouse_brain.mnc
--pipeline-name=09feb17
--lsq12-protocol=/hpf/largeprojects/MICe/tools/protocols/linear/Pydpiper_testing_default_lsq12.csv
--nlin-protocol=/hpf/largeprojects/MICe/tools/protocols/nonlinear/Pydpiper_minctracc_standard_90_micron.pl
--registration-method=minctracc
--no-run-maget
--maget-no-mask
--csv-file=/hpf/largeprojects/MICe/vousdend/CREB_EE/documents/creb_ee_images_to_reg_09feb17.csv
--output-dir=/hpf/largeprojects/MICe/vousdend/CREB_EE/registration/09feb17/
```

## Example: In vivo images on hpf using registration chain

At time of writing, I used 4G of vmem, but this could likely be reduced to 2G.

1. Login to hpf
  - `ssh -A hpf.ccm.sickkids.ca`
2. Start cluster and load modules

```
qlogin -l vmem=4G,walltime=96:00:00
```

Please refer to the Pydpiper on the SickKids HPF page for details on which modules to load:

[Pydpiper on the SickKids HPF](#)

**Note: Right now you also need to specify an lsq12 protocol, in addition to an nlin-protocol. Additionally, your csv file needs to have at least the following columns: group (aka how you want scans to be grouped, which is typically by mouse ID) and file).**

```
registration_chain.py --pipeline-name=arghef_group2 --chain-csv-file
Group2_LongitudinalFiles_cleaned_with_id_new.csv --num-executors 250 --lsq12-protocol /hpf/largeprojects/MICe
/tools/protocols/linear/Pydpiper_testing_default_lsq12.csv --chain-common-time-point 12 --pride-of-models /hpf
/largeprojects/MICe/tools/initial-models/pride_of_models_mapping.csv --latency-tolerance 1800
```

--chain-csv-file tells the pipeline how to group data. You need to have at least three columns: timepoint, subject\_id, filename (relative of full paths of the images being registered).

--chain-common-time-point is the timepoint to be registered together

--pride-of-models csvfile with at least two columns: time\_point, and model\_file. Each row is an init model so you can have a unique init model for each age.

## At MICE

### Example: Basic registration pipeline at MICE on 20 56um brains



Please ensure you are using the latest quarantine. For a list of all quarantines see this page: [Registration Quarantines \(deprecated\)](#)

```
MBM.py \  
--num-executors 20 \  
--queue-type sge \  
--init-model /axiom2/projects/software/initial-models/Pydpiper-init-model-basket-may-2014/basket_mouse_brain.mnc \  
mnc \  
--pipeline-name test_registration \  
input_files_*.mnc
```

### Example: registration of embryo data at 27 micron resolution

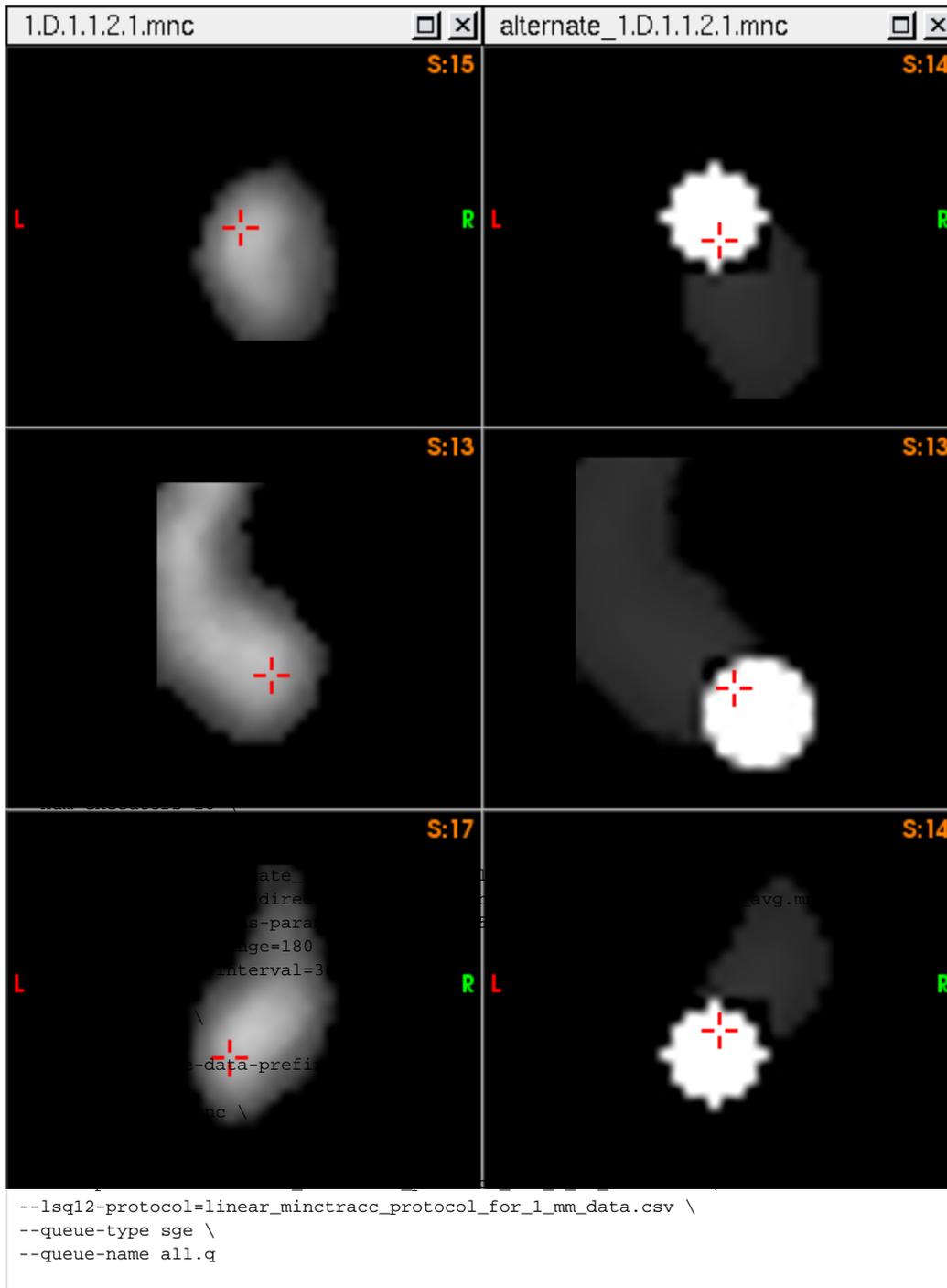
For the registration of embryo files, we use the following protocols: 6 parameter protocol, 12 parameter protocol and non linear protocol

```
MBM.py \  
--mem 14 \  
--num-executors 10 \  
--proc 1 \  
--queue-type sge \  
--pipeline-name embro_pipe \  
--init-model /axiom2/projects/software/initial-models/Pydpiper-Embryo-E15.5/E15.5_mouse.mnc \  
--lsq6-simple \  
--no-nuc \  
--no-inormalize \  
--lsq6-protocol MBM_lsq6_embryo_protocol.csv \  
--lsq12-protocol MBM_lsq12_embryo_protocol.csv \  
--registration-method minctracc \  
--nlin-protocol MBM_nlin_embryo_protocol_w_stiffness_etc.csv \  
--calc-stats \  
{input/embryo_files*.mnc}
```

### Example: Registration of data with ambiguous shape and variable rotational position

The input files to this registration are somewhat cylindrical in shape. In order to align the proper ends of each of the input data to one another, a second set of input data was generated such that a high intensity blob was formed on each object's endpoint. On the left you see an example of what was done. The far left shows the original input image, on the right is the same image with a high intensity blob ("image registration anchor"). The registration procedure starts by aligning all input images using the modified alternate images. Then, the transformation from these files is applied to the original input images, and the registration pipeline continues with the original images.

Currently (June 26, 2015) the feature to use alternate input files for the LSQ6 registration is not part of the master pydpiper branch (currently version v1.14-beta). If you want to use this feature, check out the develop branch and get the following commit: cf4e881608.



The input data has (an artificial) resolution of 1mm isotropic. Prerequisite: in the directory with the input data, a second file exists for each input file with the prefix "alternate\_" (--lsq6-alternate-data-prefix). E.g., an "ls" in the inputfiles directory will show:

```

1.D.1.mnc
1.D.2.mnc
...
alternate_1.D.1.mnc
alternate_1.D.2.mnc
...

```

## At SciNet

### Example: Registration of 149 x 56 um brains on scinet.

1. Transfer my data to scinet (after logging into scinet)
    - ssh datamover2
    - rsync -cav -e 'ssh -p 10248' --progress vousednd@sshgw.phenogenomics.ca:/projects/mice/stsatski/Palmer\_Sex\_differences/distortion\_corrected\_fixed\_MRI/\*.mnc /scratch/j/jlerch/dulcie/palmer\_compensation/imgs/
  2. Go to login node and load modules
    - /scinet/gpc/bin/gpcdev
    - module load gcc intel/14.0.1 python/2.7.8 gotoblas hdf5 gnuplot Xlibraries octave extras ImageMagick
- module use -a /project/j/jlerch/matthijs/privatemodules

module load quarantine\_tigger pydpiper/master-v1.13.1

3. Run PydPiper using 56 um pydpiper atlas and default protocols

```
• MBM.py \  
  --pipeline-name=31mar15 \  
  --init-model=/project/j/jlerch/matthijs/init-models/Pydpiper-init-model-basket-may-2014  
  /basket_mouse_brain.mnc \  
  --nuc \  
  --inormalize \  
  --registration-method=mincANTS \  
  --num-executors=39 \  
  --time=36:00:00 \  
  /scratch/j/jlerch/dulcie/palmert_compensation/imgs_study1/*.mnc \  
  /scratch/j/jlerch/dulcie/palmert_compensation/imgs_study2/*.mnc
```

4. Time to completion: 8 days.

- Submitted March 31st, Started running April 3rd
- Completed April 8th
- Had to resubmit registration on April 4th as the executors timed-out

## Example: Registration of 143 x 40 um brains on scinet using modified nlin protocol and lsq6 alignment

In this dataset, I was having problems with the initial alignment, likely due to variations in the amount of tissue surrounding the brain. I also had problems with the final nlin average having a distorted cortex - it was being "pulled out" from the brain. We got around these issues by using a different lsq6 alignment protocol and the more 'conservative' mincANTS nonlinear protocol. Specifying a time of 186 hours was probably excessive and I'm not sure how I got on.

1. PydPiper command using different lsq6 and nlin protocols

```
• MBM.py \  
  --pipeline-name 13feb15_est_conserv \  
  --init-model=/scratch/j/jlerch/matthijs/2014-12-Dulcie-40micron/Pydpiper-40-micron-basket-dec-  
  2014_crop/basket_mouse_brain_40micron_crop.mnc \  
  --registration-method=mincANTS \  
  --nuc \  
  --inormalize \  
  --nlin-protocol=/scratch/j/jlerch/matthijs/2014-12-Dulcie-40micron/Pydpiper_minicANTS_SyN_0.  
  1_40_micron.pl \  
  --lsq6-centre-estimation \  
  --lsq6-protocol=/scratch/j/jlerch/matthijs/2014-12-Dulcie-40micron  
  /Pydpiper_abbreviated_minctracc_center_estimation.csv \  
  inputs/*.mnc \  
  --time=186:00:00
```

2. Time to completion: 3 days