# Pydpiper

## What is Pydpiper?

Pydpiper is a development toolkit, written in Python, for creating modular, distributed pipelines. The code is under active development, and can be found at https://github.com/mfriedel/pydpiper. Many of the implementation details are described by comments throughout the code.

This toolkit allows you to replace or modify the parameters of component stages without disrupting the overall pipeline. Furthermore, the software stages interdependencies, manages the execution of the pipeline, and remotely distributes stages to computing clusters for effective simultaneous computation of multiple stages. Modularized, discrete stages can be removed, modified, or replaced with equivalent stages. The software was developed using image registration as a target application, but can be used for any computing pipeline.

## Writing Pydpiper Code

This wiki page is the starting point for users who want to write their own applications and modules with the Pydpiper framework. It includes information for developers who want to write their own atoms and modules from scratch, potentially doing significant coding and non-developers who want to use existing Pydpiper modules to put together new pipelines, without having to get under the hood or do an extensive amount of development.

## Using Pydpiper Applications

This wiki page is for users who want to run existing Pydpiper applications. It contains information that is relevant to all Pydpiper applications as well as additional information relevant to the specific applications linked below.

| Application | Wiki page | program name | Description |
|---|---|---|---|
| MAGeT | Creating Atlases with MAGeT | MAGeT.py | Creating atlases with multiple automatically generated templates. |
| MBM 2.0 | Iterative Model Building (MBM 2.0) (MBM.py) | MBM.py | Newer, more flexible version of MICe-build-model. |
| Registration chain | Longitudinal Registration Tools | registration_chain.py | Code that sequentially registers longitudinal data in instances where all brains cannot be put in a single pipeline. |
| Two-level Model Building | Twolevel model building | twolevel_model_building.py | Intra-subject registration followed by inter-subject registration. |

## Pydpiper testing - MBM and MAGeT

Some of the general functionality of the Pydpiper software can be tested by a combination of a MBM.py and a MAGeT.py pipeline. We have a set of test data containing two groups of brains (10 samples per group). The test bed first registers the data together, then aligns a segmented atlas to the group average and determines the amount of volumetric change found in the striatum. This test will take about a day to run, but touches on many parts of the code, so it's a good test to run.

```
# download the test data:
wget http://repo.mouseimaging.ca/repo/Pydpiper_test_files/test-data.tar.gz
tar xf test-data.tar test-data

# alter the config file in the test-data directory as needed:
{some_editor} test-data/sample.cfg

# run the test:
{pydpiper-repository}/pydpiper_testing/test_MBM_and_MAGeT.py test-data/

# the program will indicate success, which is along these lines (the number might be slightly different):
[1] "In these test data, we've introduced a 10% decrease in the volume of the striatum in the mutants. The
difference we found using the image registration software is: -7.65%."

[1] "Succeeded! We expect the registration procedure to underestimate the true underlying change (see
https://www.ncbi.nlm.nih.gov/pubmed/23756204). The difference found lies within our acceptance boundaries:
-12.5% and -7.5%."
```

## Pydpiper testing - Virtual Machine

A Virtual Machine (VM) can be downloaded that contains test data for the 4 core applications. Information on how to download the VM and run the examples can be found here.

## Workflow Diagrams

We have created a page with example workflow diagrams for the four core Pydpiper applications. It can be found here.

## Disk Clean Up after a registration

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

## Pydpiper development

Check list when working on a new release