Longitudinal Registration Tools

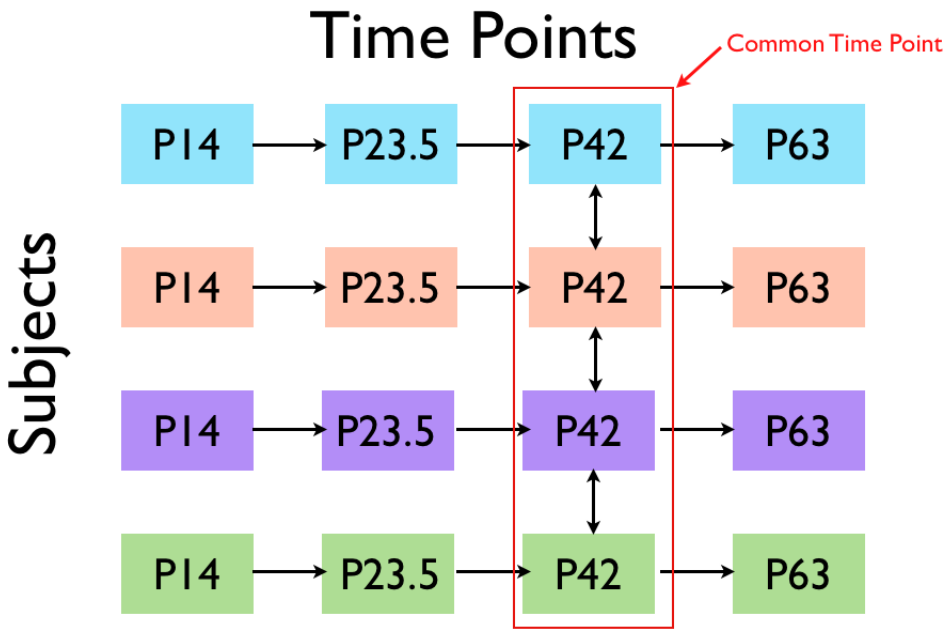
This wiki page describes code that provides alternate registration and analysis approaches for longitudinal data. All of this code has been written using the pydpiper framework.

Registration Chain

Introduction

Registration_chain.py is code that sequentially registers longitudinal data in instances where not all brains can be put into a single pipeline. As an example, consider a tumour-prone mouse. At early timepoints in a longitudinal study, a brain tumour might be small or non-existent, but by later time points, quite large. It is unlikely that a brain with a large tumour can be successfully registered to a brain with a small one. However, scans at adjacent timepoints (e.g. time point 1 and time point 2, time point 2 and time point 3, etc) have a much higher chance at being successfully registered together. Additionally, the registration chain method could be applied to any set of longitudinal data, potentially providing more accurate registrations and robust statistics.

A schematic of how the registration chain code works is shown below.



In the above diagram, four subjects are depicted (coloured in blue, pink, purple and green) and each subject is scanned at four different time points, P14, P23.5, P42 and P63. The way the registration chain method works is as follows:

1. For each subject, the scan at P14 is registered to the scan at P23.5, P23.5 to P42 and P42 to P63. The transformation for each of these registrations, as well as its inverse, is saved.
2. Displacement fields, absolute jacobians and relative jacobians are calculated for each of the above transformations. This means that we will have statistics from P14 to P23.5, P23.5 to P42 and P42 to P63 for each subject.
3. All subjects at a specified time point are registered together and a consensus average is created. In the above diagram, the selected time point is P42, but it could be any of the other time points as well. The appropriate time point for this will depend on the study design. This consensus average provides a common space for comparing registrations.

> ⚠ If your data is longitudinal and there are significant overall size differences between the early and the later time points, you should use one of the later (if not the last) time point as the common time point. The reason for this, is that concatenating the transformations from the smaller subject towards the larger subject behaves well. However, we often see that concatenating transformation from the later (larger subject) time points towards the earlier ones results in very ill defined transformations. This probably is due to misregistrations, whose effect is very apparent from large -> small, but not from small -> large.

4. For each subject at each time point, a transform is created from the consensus average back to that individual time point. This is done via transform concatenation. For example, to find the transform from P14 to the consensus average (nlin_avg) for a single subject, we would use the following concatenated transform: P14_to_P23.5.xfm + P23.5_to_P42.xfm + P42_to_nlin_avg.xfm. We can then invert this concatenated transform to get the transform from nlin_avg back to P14 for that subject. This procedure is repeated for all subjects at all time points.
5. Absolute and relative jacobians are calculated from the consensus average back to each individual time point, using the transforms created in the previous step.
6. The absolute and relative jacobians calculated in step #2 are resampled into the space of the consensus average created in step #3. Example: consider the absolute jacobian from P14 to P23.5 for a single mouse. This jacobian can be resampled by using the concatenated transform described in step #4 (P14_to_P23.5.xfm + P23.5_to_P42.xfm + P42_to_nlin_avg.xfm).

Pre-requisites

In order to run this software, you must have the following:

1. A set of brains in a longitudinal study, in either native (scanner) space or after a standard lsq6 alignment. Currently, we recommend lsq6. (See note below.)
2. A csv file containing all brains in the study. Table must have atleast three columns: subject_id, timepoint, and filename. 'filename' is the MR images and can be either relative or full paths.
3. A csv file with init models. It has two columns: time_point and modelfile. This way you can have different init models at different timepoints in the registration.

> ⚠ **Important Pre-requisite Caveats**
>
> Pre-requisite #1: If input scans are in native space, they will be aligned first using the LSQ6 module, prior to starting the chain alignment.

Usage

The latest tagged version of this code (available on github) is installed at MICe. To run at MICe, simply type registration_chain.py with the appropriate options, or --help for a complete list. To install and run this elsewhere, see the pydpiper wiki page and github repository for more information.

Options unique to the registration_chain.py code are:

```
Registration-chain options:
 Options for registering consecutive timepoints of longitudinal data.

  --chain-csv-file CSV_FILE
                        The spreadsheet with information about your input
                        data. For the registration chain you are required to
                        have the following columns in your csv file: "
                        subject_id", "timepoint", and "filename". Optionally
                        you can have a column called "is_common" that
                        indicates that a scan is to be used for the common
                        time point registration using a 1, and 0 otherwise.
  --chain-common-time-point COMMON_TIME_POINT
                        The time point at which the inter-subject registration
                        will be performed. I.e., the time point that will link
                        the subjects together. If you want to use the last
                        time point from each of your input files, (they might
                        differ per input file) specify -1. If the common time
                        is not specified, the assumption is that the
                        spreadsheet contains the mapping using the "is_common"
                        column. [Default = None]
  --chain-common-time-point-name COMMON_TIME_POINT_NAME
                        Option to specify a name for the common time point.
                        This is useful for the creation of more readable
                        output file names. Default is "common". Note that the
                        common time point is the one created by an iterative
                        group-wise registration (inter-subject).
  --pride-of-models PRIDE_OF_MODELS
                        (selected longitudinal pipelines only!) Specify a csv
                        file that contains the mapping of all your initial
                        models at different time points. The idea is that you
                        might want to use different initial models for the
                        time points in your data. The csv file should have one
                        column called "model_file", and one column called
                        "time_point". The time points can be given in either
                        integer values or float values. Each model file should
                        point to the file in standard space for that
                        particular model. [Default = None]
```

In addition to the above options, there are a number of other options that may be specified, but are not unique to the registration chain code. These are described elsewhere in the wiki.

Verifying your pipeline

Common time point registration: in the {pipeline_name}_nlin folder you will find a bunch of averages. The highest number is the final non linear average. The subjects that make up this average are in the {pipeline_name}_processed directory. This directory also contains files from subjects that were not used in creating the common time point average, but it's easy to find the files you need: they all end in -resampled-final-nlin.mnc.

```
# note that this command only works at MICe. If you do not have OCCIviewer installed, you can check the final
non linear average against the individual files using the program "register"
# also note that your last common time point could have a different name, and have more or less generation.
Which means that your last time point is not necessarily time point 3.

OCCIviewer {pipeline-name}_nlin_common/common-nlin-3.mnc {pipeline-name}_processed/*/resampled
/*_N_I_lsq6_lsq12_and_nlin-resampled.mnc
```

The above registration most often works well. The following is really the crux of the registration chain pipeline. We now need to verify that the concatenation of all the chaining transformation get all time points in the final average space in a *reasonable* way. Remember that these are registrations between potentially a very small and less developed brain into the space of a fully developed brain. This can never be perfect, since many features from the old brain don't exist in the young brain. Each of the subjects that was not part of the common time point average is resampled into that space and will reside in the "resampled" directory. The name will have the form {file_basename}_to_{common_time_point_name}.mnc. By default the common time point is called "common". Given those defaults, look at the following files:

```
OCCIviewer {pipeline-name}_nlin_common/common-nlin-3.mnc {pipeline-name}_processed/*/resampled/*_to_common_avg-
resampled.mnc
```

Files created for each input

For each input file, stats volumes are created. These contain absolute and relative Jacobian determinants. Absolute Jacobians are calculated from transformations that include linear scaling/shearing and the relative Jacobians are calculated from the non linear deformation fields only.

Stats volumes

Kind of stats volume	name for non-common time point inputs	name for time points that were used for the common time point
Absolute Jacobians from the transformation common_time_point -> input file	/stats-volume/{filebase}_to_common_inverted_absolute_log_determinant.mnc	/stats-volume/{filebase}-final-nlin_with_additional_inverted_absolute_log_determinant.mnc
Relative Jacobians from the transformation common_time_point -> input file	/stats-volume/{filebase}_to_common_inverted_pure_nlin_relative_log_determinant.mnc	/stats-volume/{filebase}-final-nlin_with_additional_inverted_pure_nlin_relative_log_determinant.mnc
Absolute Jacobians from time point to next time point (resampled into common space) *	/stats-volume/{filebase}_to_{next_time_point}_absolute_log_determinant_common.mnc	(same)
Relative Jacobians from time point to next time point (resampled into common space) *	/stats-volume/{filebase}_to_{next_time_point}_pure_nlin_relative_log_determinant_common.mnc	(same)

* These do not exist for the last time point. When you have the 4 time points from the example, these stats files only exist for P14, P23.5 and P42

Transforms

Kind of transform	name for non-common time point inputs	name for time points that were used for the common time point
Transform from file (lsq6 space) to common time point	/transforms/{filebase}_to_common.xfm	/transforms/{filebase}-final-nlin_with_additional.xfm
Transform from common time point to file (lsq6 space)	/transforms/{filebase}_to_common_inverted.xfm	/transforms/{filebase}-final-nlin_with_additional_inverted.xfm
Transform from time point (lsq6 space) to next time point (lsq6) *	/transforms/{filebase}_to_{next_time_point}_chain_4.xfm	(same)

Example

To run this code for a 12 subjects that have been scanned at 4 time points each, using pydpiper at hpf, your command would look something like:

```
registration_chain.py \
--pipeline-name=05April2013_12subjects \
--num-executors 12 \
--latency-tolerance 1800 \
--lsq12-protocol /hpf/largeprojects/MICe/tools/protocols/linear/Pydpiper_testing_default_lsq12.csv \
--chain-common-time-point 3 \
--pride-of-models /hpf/largeprojects/MICe/tools/initial-models/pride_of_models_mapping.csv \
--chain-csv-file list_of_files.csv
```

As usual, the amount of memory, executors, directory names, etc will all be dependent on the particular data you are registering.