

emalign 1.0 – user guide (Python version)

Yael Harpaz, Yoel Shkolnisky*

May 2022

1 System requirements

Recommended Environments:

The package has been tested on Ubuntu 16.04 and Windows 10. It should probably work on other versions of Windows and Linux but has not been tested on them yet. Similarly, for macOS.

Python 3.6.0+ is required.

The package makes use of the pyfftw package, which in turn uses the FFTW library. Before installing KLTPicker make sure you have the FFTW library installed on your system:

http://www.fftw.org/fftw3_doc/Installation-and-Customization.html#Installation-and-Customization

2 Installing emalign

2.1 Install emalign via pip:

We recommend installing emalign via pip.

For installation run:

```
$ pip install emalign
```

2.2 Install emalign from source

The tarball of the source tree is available via:

```
$ pip download emalign
```

You can install emalign from the tarball:

```
$ pip install emalign-x.x.x.tar.gz
```

You can also install the development version of emalign from a cloned Git repository:

```
$ git clone https://github.com/ShkolniskyLab/emalign.git
```

```
$ cd EAlign
```

```
$ pip install .
```

* yoelsh@tauex.tau.ac.il

2.3 Uninstall emalign

Use pip to uninstall emalign:

```
$ pip uninstall emalign
```

2.4 Upgrade emalign

Just use pip with -U option:

```
$ pip install -U emalign
```

3 Using emalign

Note: after installing emalign, you need to open a new terminal and run emalign from there.

3.1 Getting help

Type

```
$ emalign -h
```

to print help

3.2 Generate test data

The call

```
$ emalign --make-test-data -v
```

will generate two files (in the direction from which the command was called) named `map_ref_2660.mrc` and `map_transformed_2660.mrc`, which are two density maps of size $129 \times 129 \times 129$ pixels that are related by rotations and translation. These maps can be used to test the functionality of emalign, and will be used in the examples below.

3.3 Interactive mode

To run emalign in interactive mode simply run:

```
$ emalign
```

The following questions will appear, one by one

- 1. Enter full path of reference volume MRC file:**
Type the full path to the map file of one of the two maps to align (the next map will be aligned to this map). For example, type `map_ref_2660.mrc`.
- 2. Enter full path of query volume MRC file:**
The fill path to the map to align to the reference volume. For example, type `map_transformed_2660.mrc`.

3. **Enter full path of output aligned volume MRC file:**
Full path of the MRC file containing the query map aligned to the reference map.
Type `map_aligned_2660.mrc`.
4. **Enter the downsampled size in pixels (default 64):**
To speed up computations, the maps to align are downsampled before processing.
The default downsampling is to $64 \times 64 \times 64$ pixels. Just press enter to use default value.
5. **Enter the number of projections (default 30):**
The alignment algorithm aligns the two volume by generating their projections and detecting common lines in the projections. The value indicated the number of projections to use. Noisier maps may require more projections. Just press enter.
6. **Do you want to save output parameters [y/n]:**
Do you want to write the estimated alignment parameters into a file. If YES, you will have to provide next the filename to which to save the parameters.
7. **Enter full path of output parameters txt file:**
Type `params.txt`
8. **Display detailed progress? [y/n]:**
Type `y`

Once `emalign` return, make sure that the file has been created and that this map file is aligned with `map_transformed_2660.mrc`. You can inspect `params.txt` for the alignment parameters.

3.4 Using command-line arguments

You can also run the program by passing command-line arguments to the program. To view help message and the available command-line arguments run:

```
$ emalign -h
```

Usage:

```
$ emalign [-h] [-v1 VOL1] [-v2 VOL2] [-o OUTPUT_VOL]
[--downsample DOWNSAMPLE] [--n-projs N_PROJS]
[--output-parameters OUTPUT_PARAMETERS] [-v]
[--make-test-data] [--version]
```

Explanation of the arguments and flags:

Flag	Argument	Default	Optional
-v1 VOL1, --vol1 VOL1	Full path of first mrc input file	(N/A)	No
-v2 VOL2, --vol2 VOL2	Full path of second mrc input file VOL2 is aligned to VOL1, that is, <code>emalign</code> find the transformation that needs to be applied to VOL2 to be aligned to VOL1	(N/A)	No
-o OUTPUT_VOL, --output-vol OUTPUT_VOL	Full path of output mrc file containing VOL2 aligned to VOL1	(N/A)	No
--downsample DOWNSAMPLE	Dimension to downsample input volumes to speed up computations	64	Yes
--n-projs N_PROJS	Number of projections to use for alignment	30	Yes

--output-parameters OUTPUT_PARAMETERS	Full path of output txt file containing the alignment parameters	(N/A)	Yes
-v, --verbose	Print detailed progress and info messages	False	Yes
--make-test-data	Generate test volumes to test emalign. This options overrides other options	False	Yes
--version	print program version and exit	False	Yes

4 Citation

If you use the emalign, please cite