An efficient global and local mixed Cryo-EM density map retrieval tool based on parallel-accelerated CryoAlign

Contents

1 Introduction

Cryogenic electron microscopy (Cryo-EM) has enabled the determination and storage of an increasing number of biomolecular density maps in the public database EMDB. To fully leverage this valuable resource, conducting searches based on the structural similarity of these density maps is crucial. To address this need, we have developed a tool based on CryoAlign that efficiently performs both global and local searches. This user-friendly software allows users to compute point cloud features of all density maps in a custom dataset and carry out comparative retrieval with a single command. We have optimized CryoAlign's local alignment algorithm, designed a method for the automated construction of a point cloud retrieval library, and devised a multi-scale scoring function, enabling users to conduct efficient and accurate similarity searches. Additionally, our retrieval application provides overlays for the retrieved maps, allowing researchers to easily visualize the results. Our application demonstrates precise and robust capabilities in structural similarity retrieval, offering a convenient tool for scientists engaged in Cryo-EM data research. our retrieval application is an open-source toolkit, with the source code accessible at https://github.com/JokerL2/CryoSearch

2 Installation

The sections below explain how to download and install CryoSearch on your computer.

2.1 Prerequisites

Note that CryoSearch depends on and uses several external programs and libraries.If you have a Docker environment with GPU, we strongly recommend you to generate image using the dockerfile.

- Ubuntu 18.04 or later
- CMake 3.20+
- Open3D 0.18.0
- Libtorch 2.2.0 CUDA(Version above 12.2)
- FLANN
- PCL(point cloud libraries)
- EIGEN3
- MLPACK 3.2.2

- CNPY
- TEASER++
- FFTW

2.2 Installtion of CryoSearch

We store the public release versions of CryoSearch on GitHub, a site that provides code-development with version control and issue tracking through the use of git. To clone the repository, run the following command

```
git clone https://github.com/JokerL2/CryoSearch_cpp.git
```

For the convenience of environment configuration, we have provided Docker images that include the necessary environments and external libraries. Users must ensure that the external libraries have been properly installed on their servers.

- Using Docker

```
docker build -f dockerfile -t [image name] .
docker run -it --name [container name] --gpus all [image name]
```

- Some external libraries and tools

```
Download libtorch
cd ~/
wget https://download.pytorch.org/libtorch/nightly/cu121/libtorch-cxx11-
    abi-shared-with-deps-2.2.0.dev20231213%2Bcu121.zip
unzip libtorch-cxx11-abi-shared-with-deps-2.2.0.dev20231213+cu121.zip
```

- Install CryoSearch

```
cd /CryoSearch && mkdir build && cd build
cmake ..
make
```

3 Executable file description

After installation, three executable files will be generated, corresponding to the three execution needs of CryoSearch.

Usage:

```
CryoAlign [data dir] [source.map] [source contour level] [target.map]
[target contour level] [source.pdb] [source sup.pdb] [voxel_size] [feature_radius] [alg_type]
```

CryoAlign Usage

Options:

- `--data_dir`: Map file path.
- `--source_map`: Source emdb num.
- `--source_contour_level`: Author recommend contour level.
- `--target_map`: Target emdb num.
- `--target_contour_level`: Author recommend contour level.
- `--source_pdb` (optional): Source pdb name.
- `--source_sup_pdb` (optional): Transformed source pdb name (ground truth).
- `--voxel_size`: Sampling interval (defaults 5.0).
- `--feature_radius`: Radius for feature construction (defaults 7.0).
- `--alg_type`: `Global_alignment` or `Mask_alignment`.

Example:

For `Global_alignment`:

```
CryoAlign --data_dir ../../example_dataset/emd_3695_emd_3696/ --source_map EMD-3695.map --source_cont
```

For `Mask_alignment`:

```
CryoAlign --data_dir ../../example_dataset/emd_3695_emd_3696/ --source_map EMD-3695.map --source_cont
```

CryoAlign_extract_keypoints Usage

Usage:

```
CryoAlign_extract_keypoints [data dir] [source.map] [source contour level] [target.map] [target conto
```

Options:

- `--data_dir`: Map file path.
- `--map_name`: Source emdb num.
- `--contour_level`: Author recommend contour level.
- `--voxel_size`: Sampling interval. (defaults 5.0)

Example:

```
CryoAlign_extract_keypoints --data_dir ../../example_dataset/emd_3695_emd_3696/ --map_name EMD-3695.m
```

CryoAlign_alignment Usage

```
CryoAlign_alignment [data dir] [source_xyz] [target_xyz] [source_sample]
[target_sample] [source.pdb] [source sup.pdb] [voxel_size] [feature_radius]
[alg_type]


Options:
  --data_dir: Map file path.
  --source_xyz: Source map keypoints file.
  --target_xyz: Target map keypoints file
  --source_sample: Source map sample file.
  --target_sample: Target map sample file.
  --source_pdb(optional): Source pdb name.
  --source_sup_pdb(optional): Transformed source pdb name (ground truth).
  --voxel_size: Sampling interval. (defaults 5.0)
  --feature_radius: Radius for feature construction. (defaults 7.0)
  --alg_type: Global_alignment or Mask_alignment.
```

Examples

For Global Alignment

```
CryoAlign_alignment --data_dir ../../example_dataset/emd_3695_emd_3696/ \
--source_xyz Points_3695_5.00_Key.xyz --target_xyz Points_3696_5.00_Key.xyz \
--source_sample EMD-3695_5.00.txt --target_sample EMD-3696_5.00.txt \
--source_pdb 5nsr.pdb --source_sup_pdb 5nsr_sup.pdb --voxel_size 5.0 \
--feature_radius 7.0 --alg_type global
```

For Mask Alignment

```
CryoAlign_alignment --data_dir ../../example_dataset/emd_3695_emd_3696/ \
--source_xyz Points_3695_5.00_Key.xyz --target_xyz Points_3696_5.00_Key.xyz \
--source_sample EMD-3695_5.00.txt --target_sample EMD-3696_5.00.txt \
--source_pdb 5nsr.pdb --source_sup_pdb 5nsr_sup.pdb --voxel_size 5.0 \
--feature_radius 7.0 --alg_type mask
```