How to Run EMReady

EMReady - Huang Laboratory

Load the program :

On your GPU node

```
module load sbgrid/emready
```

Usage:

```
EMReady.sh in_map.mrc out_map.mrc [Options]
```

Required arguments:

- in_map.mrc: File name of input EM density map in MRC2014 format. (not the sharp map !)
- out_map.mrc: File name of the output EMReady-processed density map.

Options:

- -g GPU_ID: ID(s) of GPU devices to use. e.g. '0' for GPU #0, and '2,3,6' for GPUs #2, #3, and #6. (default: '0')
- -s STRIDE: The step of the sliding window for cutting the input map into overlapping boxes. Its value should be an integer within [12,48]. (default: 12)
- -b BATCH_SIZE: Number of boxes input into EMReady in one batch. (default: 10)
- -m MASK_MAP: Input mask map in MRC2014 format. (default: None)
- -c MASK_MAP_CONTOUR: Set the contour level of the mask. (default: 0.0)
- -p MASK_STRUCTURE: Input structure mask files in PDB or CIF format (default: None)
- -r MASK_STRUCTURE_RADIUS: Zone radius in angstroms (default: 4.0)
- -mo MASK_OUT_PATH: File path of the output binary mask map. (default: None)
- –use_cpu: Run EMReady on CPU instead of GPU.

Notes:

1. Users can specify a larger STRIDE of sliding window (default=12) to reduce the number of overlapping boxes to calculate. If users run out of memory, they may set it to a larger value. Howerver, since the size of the overlapping boxes is 48×48×48, the value of STRIDE should not exceed 48.
2. By default, EMReady will run on GPU(s). Users can adjust the BATCH_SIZE according to the VRAM of their GPU. Empirically, an NVIDIA A100 with 40 GB VRAM can afford a BATCH_SIZE of 30. Users can run EMReady on CPUs by setting –use_cpu. But this may take very long time for large density maps.