KLTPicker 1.0 – user guide (Python version)

Amitay Eldar*, Boris Landa, Yoel Shkolnisky†

September 2020

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

For optional GPU support, the package requires:

- NVIDIA CUDA GPU with the Compute Capability 3.0 or larger
- CUDA Toolkit: v9.0 / v9.2 / v10.0 / v10.1 / v10.2 / v11.0

2 Installing KLTPicker

2.1 Install KLTPicker via pip:

We recommend installing KLTPicker via pip.

For installation **without** GPU support run:

```
$ pip install kltpicker
```

For installation **with** GPU support (provided that your system satisfies the above requirements) run:

```
$ pip install kltpicker[gpu]
```

2.2 Install KLTPicker from source

The tarball of the source tree is available via pip download kltpicker. You can install KLTPicker from the tarball:

* amitayeldar@mail.tau.ac.il
† yoelsh@tauex.tau.ac.il

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

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

```
$ git clone https://github.com/dalitco54/kltpick.git
$ cd kltpicker
$ pip install.
```

2.3 Uninstall KLTPicker

Use pip to uninstall KLTPicker:

```
$ pip uninstall kltpicker
```

2.4 Upgrade KLTPicker

Just use pip with -U option:

```
$ pip install -U kltpicker
```

3 Using the KLT picker

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

3.1 Interactive mode

To run the script in interactive mode simply run:

```
$ kltpicker
```

The following questions will appear, one by one

1. **Enter full path of micrographs MRC files**:
   Type the full path to the directory which contains the micrograph MRC files.

2. **Enter full path of output directory**:
   Type the full path where the particles coordinate files will be saved.

3. **Enter the particle size in pixels**:
   Type the particle diameter in pixels (more precisely, the diameter of the extracted box).

4. **Pick all particles? (Y/N):**
   [Y] Type Y to pick all particles using the optimal threshold derived on the paper.
   [N] If you type N, then the following question will appear:
   a. **How many particles to pick:**
      Type the number of particles to pick in each micrograph.

5. **Pick noise images? (Y/N):**
   [N] Type N if you don't want to pick noise images.
   [Y] If you type Y then the following question will appear:
   a. **How many noise images to pick**?
      Type the number of noise images to pick in each micrograph.

6. **Display detailed progress? (Y/N):**
   [N] Type N to display a simple progress-bar.
   [Y] Type Y to display during runtime the number of particles and noise images picked from each micrograph.

7. **Maximum number of parallel processes to run:**
   Specify the maximum number of processes you want to allow the program to run in parallel.

If you have installed the version that supports GPU the following question will appear:

8. **Do you want to use the GPU? (Y/N)?**
   [N] Type N to not use GPUs found on your system (CPU only).
   [Y] Type Y to use the GPUs found on your system. If you type Y the following question will appear:
   a. **Which GPUs would you like to use?**
      Specify the indices of the GPUs on your system which you want to use, separated by whitespaces or commas. Type -1 to use all available GPUs.

3.2 Using flags to pass arguments

You can also run the program by passing arguments to the program using flags. To view a help message and the available flags run:

```
$ kltpicker -h
```

Usage:

```
$ kltpicker -i INPUT_DIR -o OUTPUT_DIR -s PARTICLE_SIZE -p
NUM_PARTICLES -n NUM_NOISE --max-processes MAX_PROCESSES
--gpu-indices GPU_INDICES -v --no-gpu
```

In red – flags that are only available in the version with GPU support.

Explanation of the arguments and flags:

Flag	Argument	Default	Optional
`-i, --input-dir`	Full path of input directory.	(N/A)	No
`-o, --output-dir`	Full path of output directory.	(N/A)	No
`-s, --particle-size`	Expected size of particles in pixels.	(N/A)	No
`-p, --num-particles`	Number of particles to pick per micrograph. Enter -1 to pick all particles.	-1	Yes
`-n, --num-noise`	Number of noise images to pick per micrograph.	0	Yes
`-v, --verbose`	Verbose. Choose this to display number of particles and noise images picked from each micrograph	False	Yes

	during runtime. Otherwise, you get a simple progress bar.		
`--max-processes`	Limit the number of concurrent processes to run. Enter -1 to let the program choose.	-1	Yes
`--no-gpu`	Don't use GPUs.	False	Yes
`--gpu-indices`	Indices of GPUs to be used, **separated by whitespaces.** Valid indices: zero through one less than the number of GPUs on the system. Enter -1 to use all available GPUS.	-1	Yes

The KLT picker will then start running, while displaying progress notifications. The outputs are the coordinate files (box and star) and a text file summarizing the picking process, written to the output directory.

4 Examples

In this section we demonstrate the execution of the program on 5 micrographs of the EMPIAR-10028 data set (Plasmodium Falciparum 80S ribosome) [2] from the EMPIAR repository [1].

First, download the micrographs manually from https://www.ebi.ac.uk/pdbe/emdb/empiar/entry/10028/

For the sake of the example, assume you placed the downloaded files in a directory named `/home/user/example/micrographs`

Then proceed to run the program in interactive mode or by passing arguments using flags.

4.1 Interactive mode

In this section we demonstrate the execution of the program in interactive mode.

```
$ kltpicker

Enter full path of micrographs MRC files:
/home/user/example/micrographs

Found 5 MRC files.

Enter full path of output directory:
/home/user/example/results

Enter the particle size in pixels: 300

Pick all particles? (Y/N): y

Pick noise images? (Y/N): y

How many noise images to pick: 30

Display detailed progress? (Y/N): n

Enter maximum number of concurrent processes (-1 to let the
program decide): -1

Use GPU? (Y/N): y
```

```
Which GPUs would you like to use?
(Valid indices: 0,...,7. Enter -1 to use all): 0 1 2 3

Running on 5 files.
Using GPUs 0, 1, 2, 3.
Preprocessing (usually takes up to 1 minute)...
Preprocess finished. Picking particles...
[Elapsed Time: 0:00:53] |##############################| (100%)
Finished successfully!

Writing picking summary at the output path.
Picked *** particles and 0 noise images out of 5 micrographs.
```

4.2 Using flags to pass arguments

In this section we demonstrate the execution of the program using flags to pass arguments.

The most basic way to run the program:

```
$ kltpicker -i /home/user/example/micrographs -o
/home/user/example/results -s 300
```

If you want detailed progress output, run:

```
$ kltpicker -i /home/user/example/micrographs -o
/home/user/example/results -s 300 -v
```

If you don’t want to use the GPU, run:

```
$ kltpicker -i /home/user/example/micrographs -o
/home/user/example/results -s 300 --no-gpu
```

To specify additional optional parameters such as number of noise images to pick, the maximum number of processes to run, or the GPUs you want to use:

```
$ kltpicker -i /home/user/example/micrographs -o
/home/user/example/results -s 300 -n 30 --max-processes 16 --
gpu-indices 0 1 2 5 7
```

etc.

4.3 Displaying the results in EMAN and RELION

In order to display the picking results in EMAN [3], open a terminal in the output directory `/home/user/example/results` and create a new directory named eman, using the command:

```
$ mkdir eman
```

Change directory to eman and enter the following commands one by one:

```
$ e2rawdata.py /home/user/example/micrographs/*.mrc --invert --edgenorm --xraypixel --ctfest --apix=1.0 --voltage=200.0 --cs=2.0 --ac=10.0 --threads=4 --defocusmin=0.6 --defocusmax=4.0
```

```
$ e2import.py
```

```
/home/user/example/results/pickedParticlesParticleSize300/box/*.box --import_boxes --box_type=boxes
```

```
$ e2boxer.py --allmicrographs --boxsize=300 --ptclsize=300 --apix=1.0 --no_ctf --gui --threads=4
```

In order to display the picking results of the first micrograph in RELION [4], open the terminal in the output directory `/home/user/example/results` and create a new directory named relion, using the command:

```
$ mkdir relion
```

Change directory to relion and enter the following command:

```
$ relion_display --i /home/user/example/micrographs/001.mrc --cords /home/user/example/results/pickedParticlesParticleSize300/star/001.star --scale 0.15 --particle_radius 150 --angpix 1 --lowpass 20 --pick
```

5 Citation

If you use the KLT picker, please cite KLT picker: Particle picking using datadriven optimal templates, Journal of Structural Biology, Accepted for publication.. A preprint is available at https://arxiv.org/abs/1912.06500.

6 References

[1] Iudin, A., Korir, P., Salavert-Torres, J., Kleywegt, G., and Patwardhan, A. (2016). EMPIAR: A public archive for raw electron microscopy image data. Nature Methods, 13.

[2] Wong, Wilson and Bai, Xiao-chen and Brown, Alan and Fernandez, Israel S and Hanssen, Eric and Condron, Melanie and Tan, Yan Hong and Baum, Jake and Scheres, Sjors H W. (2014). Cryo-EM structure of the Plasmodium falciparum 80S ribosome bound to the anti-protozoan drug emetine. Wong et al. eLife ,3,e03080.

[3] G. Tang, L. Peng, P.R. Baldwin, D.S. Mann, W. Jiang, I. Rees & S.J. Ludtke. (2007). EMAN2: an extensible image processing suite for electron microscopy. Journal of structural biology, 157, 38-46.

[4] Scheres, Sjors HW. (2015). Semi-automated selection of cryo-EM particles in RELION 1.3. Journal of structural biology, 128, 114-122.