Set up the environment

- Change runtime to GPU (in the menu select `Runtime -> Change runtime type -> GPU`)
- Connect Google Drive to be able to save the analysis output:

```
# Run this cell to connect to Google Drive. To run this cell click on the Play butt
from google.colab import drive

drive.mount("/content/drive")
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
```

- Note: To view connected Google Drive filesystem click on the Folder icon on the left.

- Run the cell below to install `tapqir` (takes about a minute; green checkmark means you are done):

```
!pip install --quiet tapqir > install.log
```

- **Restart the runtime** after installing Tapqir (in the menu click `Runtime -> Restart runtime`)

Download input data

In this notebook you can either analyze the example data from the tutorial or your own data. Do A) or B) below.

A) Download tutorial data

These data were acquired with Glimpse and pre-processed with the imscroll program (Friedman et al., 2015). Change directory to MyDrive:

```
%cd drive/MyDrive/
```

```
/content/drive/MyDrive
```

Download data files using wget and then unzip files:

```
!wget https://zenodo.org/record/5659927/files/DatasetA_glimpse.zip
!unzip DatasetA_glimpse.zip && rm DatasetA_glimpse.zip
```

The raw input data are:

- `garosen00267` - folder containing image data in glimpse format and header files
- `green_DNA_locations.dat` - aoiinfo file designating target molecule (DNA) locations in the binder channel
- `green_nonDNA_locations.dat` - aoiinfo file designating off-target (nonDNA) locations in the binder channel
- `green_driftlist.dat` - driftlist file recording the stage movement that took place during the experiment

B) Upload your own data to Google Drive

You will need:

- folder containing image data in glimpse format and header files
- aoiinfo file designating target molecule (DNA) locations in the binder channel
- aoiinfo file designating off-target (nonDNA) locations in the binder channel
- driftlist file recording the stage movement that took place during the experiment

Create a new analysis folder

To start the analysis create an empty folder (here named `tutorial`) which will be the working directory:

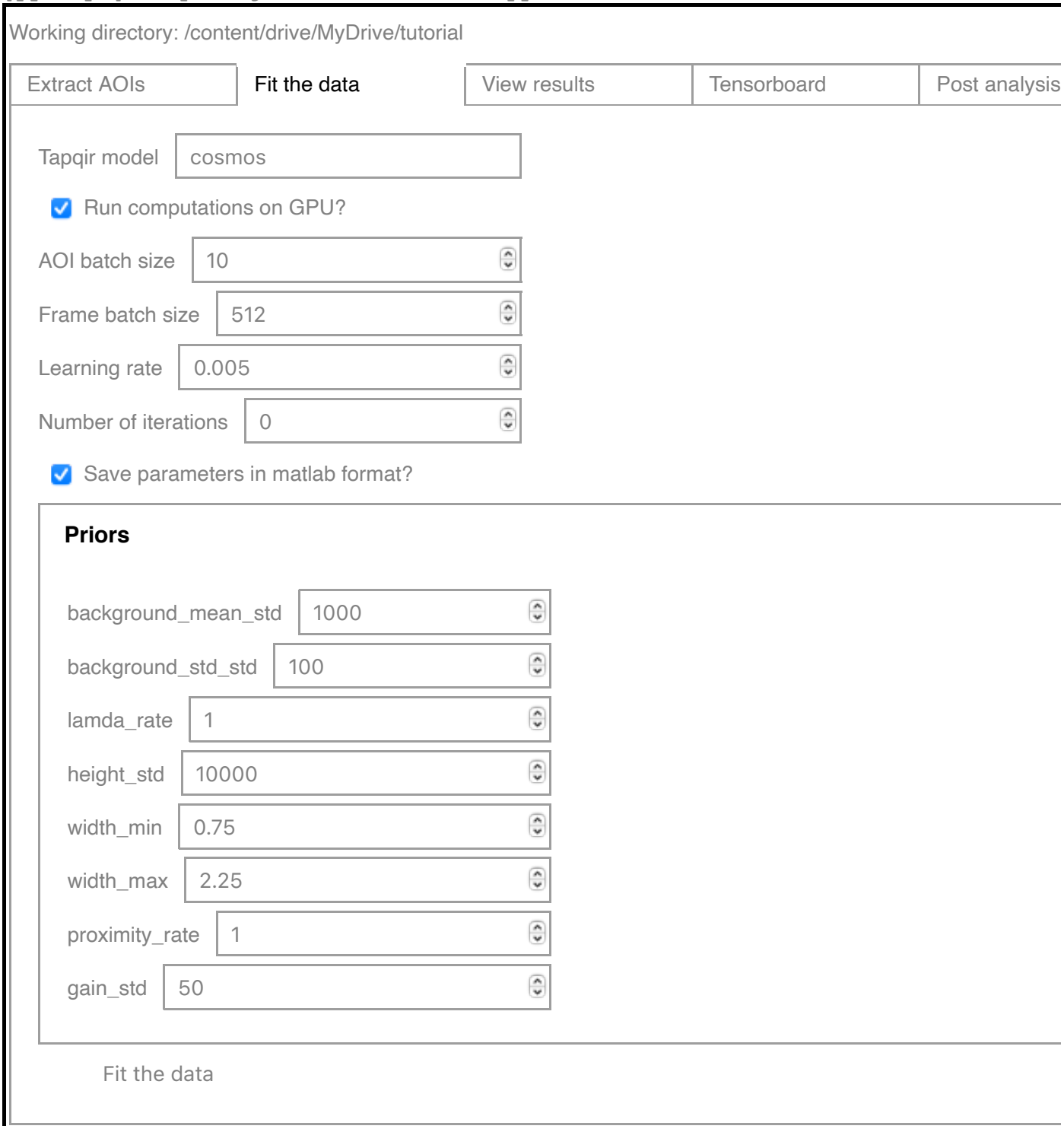
```
%mkdir /content/drive/MyDrive/tutorial
```

Start the program

To start the program run:

```
from tapqir import gui

gui.run()
```

```
[KeOps] Compiling cuda jit compiler engine ... OK
[pyKeOps] Compiling nvrtc binder for python ... OK
```



```
Fitting the data ...
INFO:tapqir:Fitting the data ...
DEBUG:tapqir.models.model:Loaded data from /content/drive/MyDrive/tutorial/dat
Iteration #74200. Loaded a model checkpoint from
/content/drive/MyDrive/tutorial/.tapqir/cosmos_model.tpqr
INFO:tapqir.models.model:Iteration #74200. Loaded a model checkpoint from
/content/drive/MyDrive/tutorial/.tapqir/cosmos_model.tpqr
DEBUG:tapqir.models.model:Tapqir version - v1.1.12
DEBUG:tapqir.models.model:Model - cosmos
DEBUG:tapqir.models.model:Device - cuda
DEBUG:tapqir.models.model:Floating precision - torch.float64
DEBUG:tapqir.models.model:Optimizer - Adam
DEBUG:tapqir.models.model:Learning rate - 0.005
DEBUG:tapqir.models.model:AOI batch size - 10
DEBUG:tapqir.models.model:Frame batch size - 512
1%                                          1000/100000 [08:05<12:44:29, 2.16it/s]
DEBUG:tapqir.models.model:Iteration #74200: Successful.
DEBUG:tapqir.models.model:Iteration #74400: Successful.
DEBUG:tapqir.models.model:Iteration #74600: Successful.
DEBUG:tapqir.models.model:Iteration #74800: Successful.
DEBUG:tapqir.models.model:Iteration #75000: Successful.
DEBUG:tapqir.models.model:Iteration #75200: Successful.
Iteration #75200 model converged.
INFO:tapqir.models.model:Iteration #75200 model converged.
Fitting the data: Done
INFO:tapqir:Fitting the data: Done
Computing stats ...
INFO:tapqir:Computing stats ...
- credible intervals
INFO:tapqir.utils.stats:- credible intervals
- spot probabilities
INFO:tapqir.utils.stats:- spot probabilities
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
/usr/local/lib/python3.7/dist-packages/tapqir/gui.py in fitCmd(b, layout, out,
    532             pykeops=True,
    533             no_input=True,
--> 534             progress_bar=tqdm_notebook,
    535         )
    536

                                6 frames
/usr/local/lib/python3.7/dist-packages/matplotlib/backends/backend_agg.py in p
filename_or_obj, metadata, pil_kwargs, *args, **kwargs)
    503             Metadata in the PNG file as key-value pairs of bytes or la
    504             encodable strings.
--> 505             According to the PNG specification, keys must be shorter t
    506             chars.
    507

ImportError: cannot import name '_png' from 'matplotlib' (/usr/local/lib/pytho
packages/matplotlib/__init__.py)
```

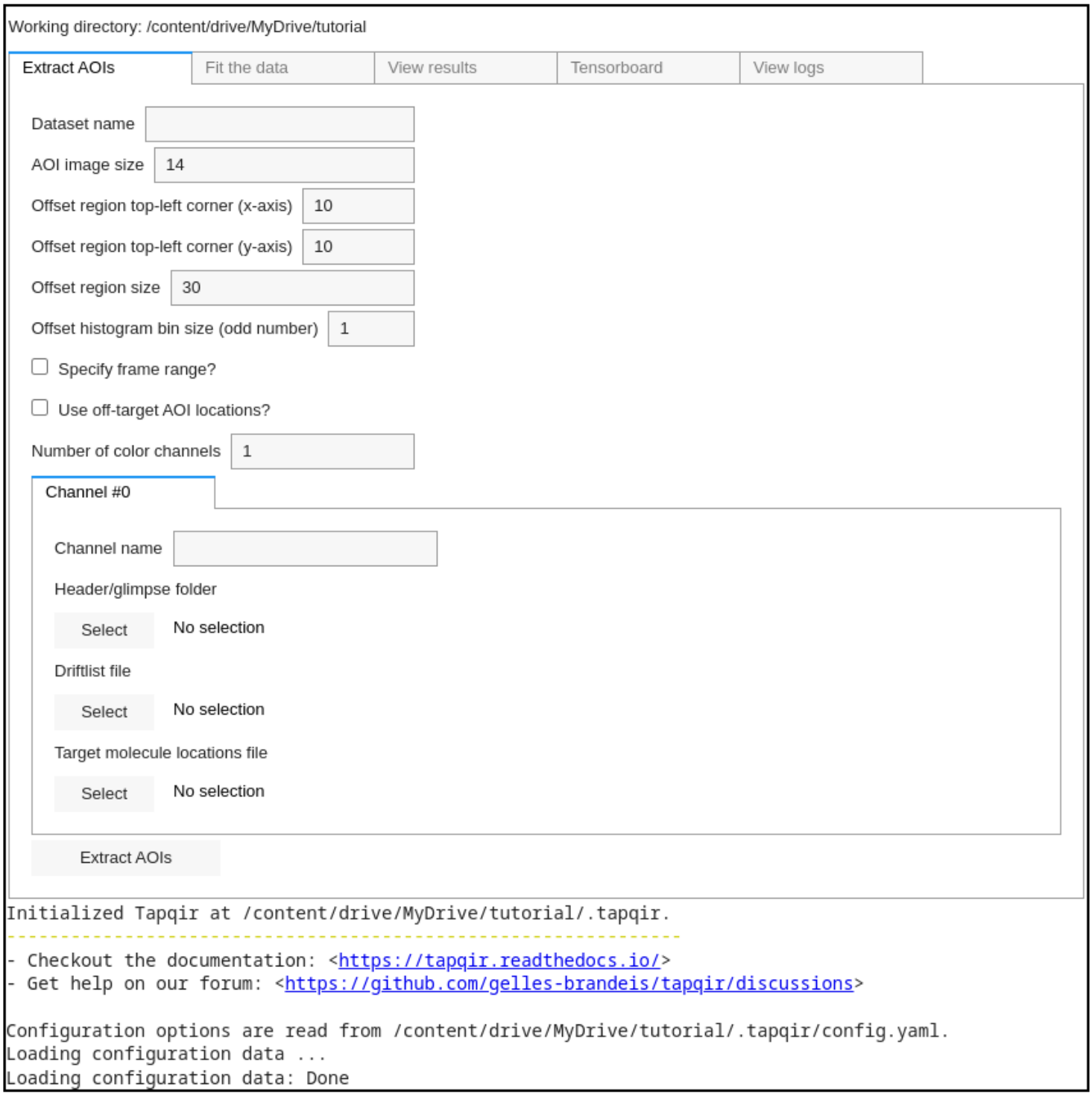
SEARCH STACK OVERFLOW

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
/usr/local/lib/python3.7/dist-packages/IPython/core/formatters.py in
__call__(self, obj)
    339                 pass
    340             else:
--> 341                 return printer(obj)
    342             # Finally look for special method names
    343             method = get_real_method(obj, self.print_method)

                                4 frames
/usr/local/lib/python3.7/dist-packages/matplotlib/backends/backend_agg.py in
print_png(self, filename_or_obj, metadata, pil_kwargs, *args, **kwargs)
    503             Metadata in the PNG file as key-value pairs of bytes or
latin-1
    504             encodable strings.
--> 505             According to the PNG specification, keys must be shorter
than 79
    506             chars.
    507

ImportError: cannot import name '_png' from 'matplotlib'
```

which will display the Tapqir GUI:

Select working directory

Click the `Select` button to set the working directory to `/content/drive/MyDrive/tutorial`:



Setting working directory creates a `.tapqir` sub-folder that will store internal files such as `config.yaml` configuration file, `loginfo` logging file, and model checkpoints.

Extract AOIs

To extract AOIs specify the following options in the `Extract AOIs` tab:

- A dataset name: `Rpb1SNAP549` (an arbitrary name)
- Size of AOI images: we recommend to use `14` pixels
- Starting and ending frame numbers to be included in the analysis (`1` and `790`). If starting and ending frames are not specified then the full range of frames from the driftlist file will be analyzed.
- The number of color channels `1` (this data set has only one color channel available)
- Use off-target AOI locations?: `True` (we recommended including off-target AOI locations in the analysis).

And specify the locations of input files for each color channel (only one color channel in this example):

- Channel name: `SNAP549` (an arbitrary name)
- Header/glimpse folder: `/content/drive/MyDrive/tutorial/DatasetA_glimpse/garosen00267`
- Driftlist file: `/content/drive/MyDrive/tutorial/DatasetA_glimpse/green_driftlist.dat`
- Target molecule locations file: `/content/drive/MyDrive/tutorial/DatasetA_glimpse/green_DNA_locations.dat`
- Off-target control locations file: `/content/drive/MyDrive/tutorial/DatasetA_glimpse/green_nonDNA_locations.dat`

See Advanced settings below for details on adjusting prior parameters.

> **About indexing**. In Python indexing starts with 0. We stick to this convention and index AOIs, frames, color channels, and pixels starting with 0. Note, however, that for starting and ending frame numbers we used `1` and `790` which are according to Matlab indexing convention (in Matlab indexing starts with 1) since driftlist file was produced using a Matlab script.

Next, click `Extract AOIs` button:

Working directory: /content/drive/MyDrive/tutorial

Extract AOIs | Fit the data | View results | Tensorboard | View logs

Dataset name: Rpb1SNAP549

AOI image size: 14

Offset region top-left corner (x-axis): 10

Offset region top-left corner (y-axis): 10

Offset region size: 30

Offset histogram bin size (odd number): 1

☑ Specify frame range?

Starting frame: 1

Ending frame: 790

☑ Use off-target AOI locations?

Number of color channels: 1

Channel #0

Channel name: SNAP549

Header/glimpse folder

Change /content/drive/MyDrive/DatasetA_glimpse/garosen00267/

Driftlist file

Change /content/drive/MyDrive/DatasetA_glimpse/green_driftlist.dat

Target molecule locations file

Change /content/drive/MyDrive/DatasetA_glimpse/green_DNA_locations.dat

Off-target control locations file

Change /content/drive/MyDrive/DatasetA_glimpse/green_nonDNA_locations.da

Extract AOIs

```
Extracting AOIs ...
Channel #0 (SNAP549)
100% 790/790 [02:04<00:00, 5.40it/s]
Processing extracted AOIs ...
Dataset: N=331 on-target AOIs, Nc=526 off-target AOIs, F=790 frames, C=1 channels, Px=14 pixels, Py=1
Data is saved in /content/drive/MyDrive/tutorial/data.tpqr
- saving images
Extracting AOIs: Done
```

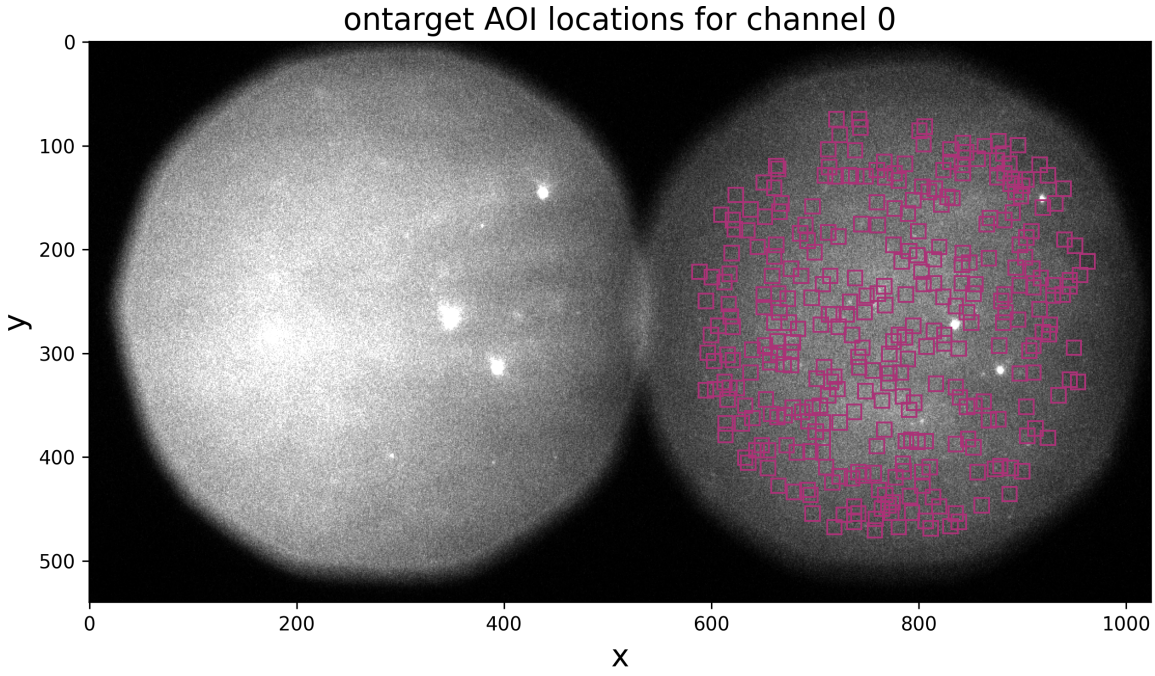
Great! The program has outputted a `data.tpqr` file containing extracted AOI images (N=331 target and Nc=526 off-target control locations):

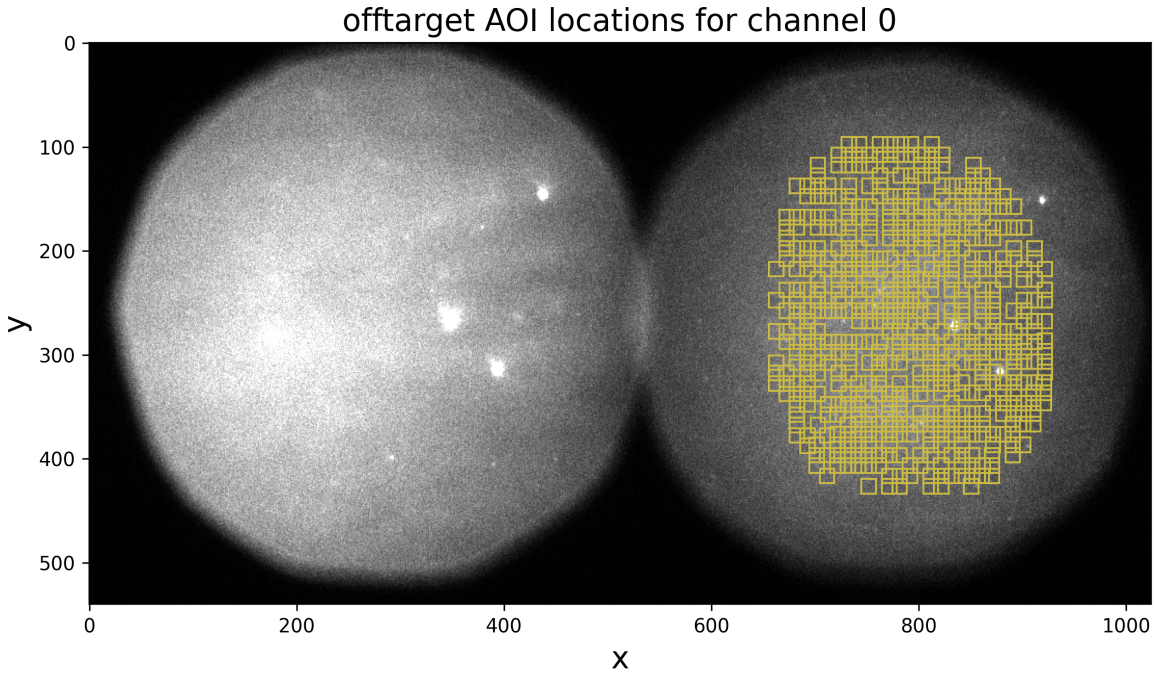
```
%ls /content/drive/MyDrive/tutorial/
```

```
data.tpqr             offset-distribution.png  offtarget-channel0.png
offset-channel0.png   offset-medians.png       ontarget-channel0.png
```

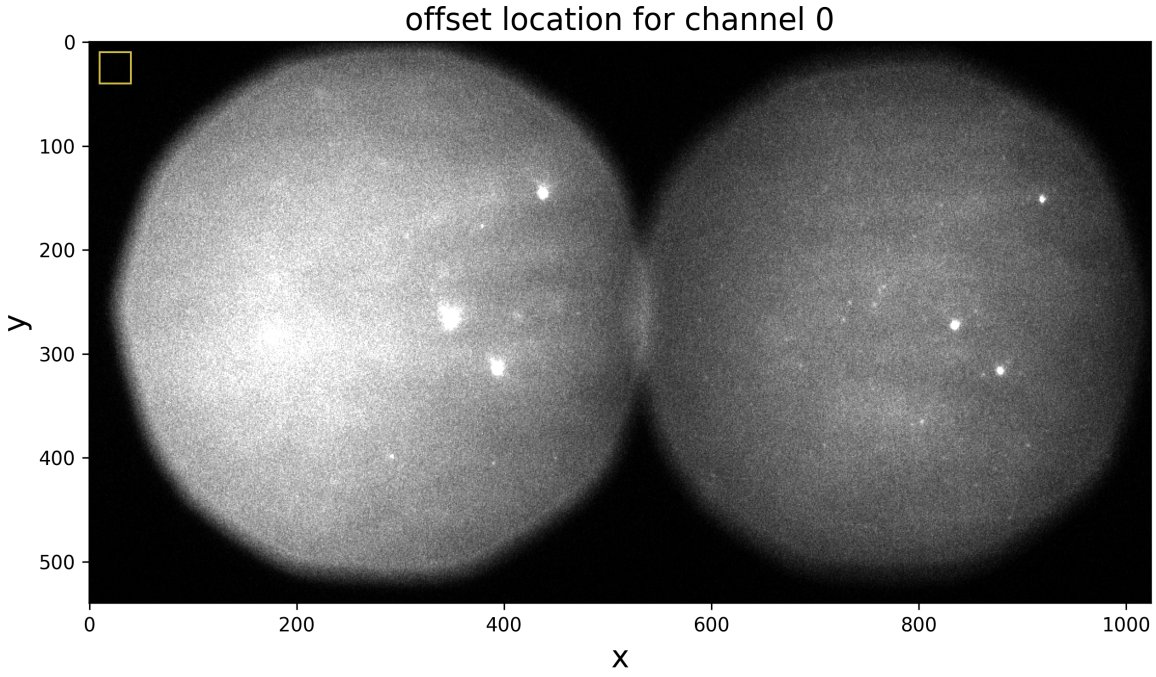
Additionally, the program has saved

- Image files (`ontarget-channel0.png` and `offtarget-channel0.png`) displaying locations of on-target and off-target AOIs in the first frame. You should inspect these images to make sure that AOIs are inside the field of view:

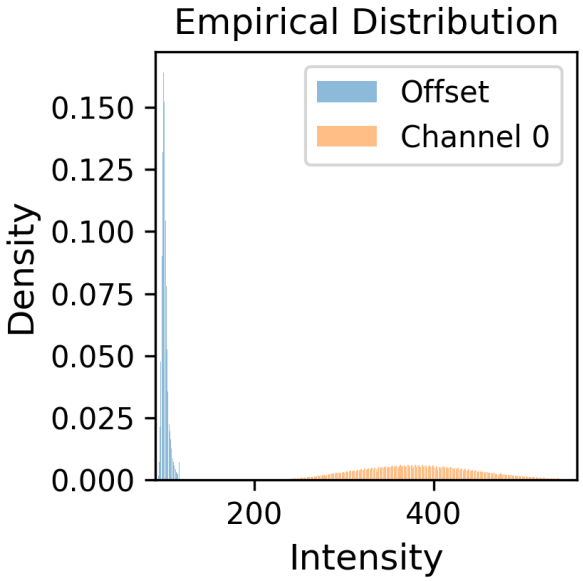


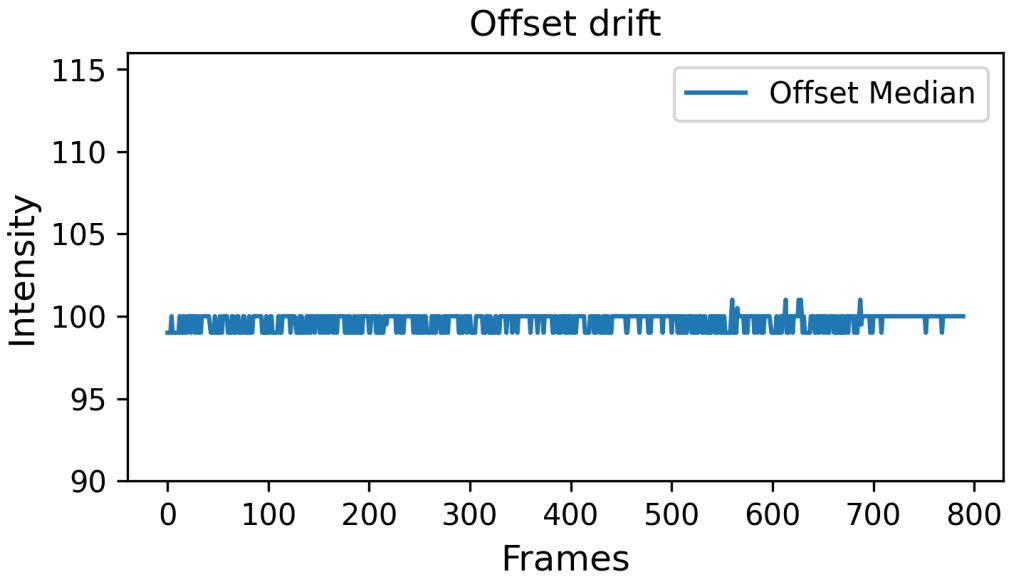


- You should also look at `offset-channel0.png` to check that offset data is taken from a region outside the field of view:



- The other two files show the intensity histograms (`offset-distribution.png`) and the offset median time record (`offset-medians.png`) (offset distribution shouldn't drift over time):


Empirical Distribution
Offset
Channel 0
Density
0.150
0.125
0.100
0.075
0.050
0.025
0.000
200
400
Intensity


Offset drift
Offset Median
Intensity
115
110
105
100
95
90
0
100
200
300
400
500
600
700
800
Frames

Fit the data

Now the data is ready for fitting. Options that we will select:

- Model - the default single-color time-independent `cosmos` model (Ordabayev et al., 2022).
- Color channel number - first chanel (`0`) (there is only one color channel in this data)
- Run computations on GPU: yes (`True`).
- AOI batch size - use default (`10`).
- Frame batch size - use default (`512`).
- Learning rate - use default (`0.005`).
- Number of iterations - use default (`0`)

See Advanced settings below for details on adjusting prior parameters.

> **About batch size**. Batch sizes should impact *training time* and *memory consumption*. Ideally, it should not affect the final result. Batch sizes can be optimized for a particular GPU hardware by trying different batch size values and comparing training time/memory usage (`nvidia-smi` shell command shows Memory-Usage and GPU-Util values).

Next, press `Fit the data` button:

Working directory: /content/drive/MyDrive/tutorial

Extract AOIs | Fit the data | View results | Tensorboard | View logs

Tapqir model: cosmos

Channel numbers: 0

Run computations on GPU?

AOI batch size: 10

Frame batch size: 512

Learning rate: 0.005

Number of iterations: 0

Save parameters in matlab format?

Priors

background_mean_std: 1000

background_std_std: 100

lamda_rate: 1

height_std: 10000

width_min: 0.75

width_max: 2.25

proximity_rate: 1

gain_std: 50

Fit the data

```
Fitting the data ...
Iteration  #6800. Loaded a model checkpoint from /content/drive/MyDrive/tutorial/.tapqir/cosmos-cha
88% 88001/100000 [6:27:09<52:20, 3.82it/s]
Iteration #87800 model converged.
Fitting the data: Done
Computing stats ...
- credible intervals
- spot probabilities
- SNR and Chi2-test
Parameters were saved in /content/drive/MyDrive/tutorial/cosmos-channel0-params.tpqr
Summary statistics were saved in /content/drive/MyDrive/tutorial/cosmos-channel0-summary.csv
Computing stats: Done
```

The program will automatically save a checkpoint every 200 iterations (checkpoint is saved at `.tapqir/cosmos_model.tpqr`). The program can be stopped at any time by clicking in the terminal window and pressing `Ctrl-C`. To restart the program again re-run `tapqir-gui` command and the program will resume from the last saved checkpoint.

After fitting is finished, the program computes 95% credible intervals (CI) of model parameters and saves the parameters and CIs in `cosmos_params.tpqr`, `cosmos_params.mat` (if Matlab format is selected), and `cosmos_summary.csv` files.

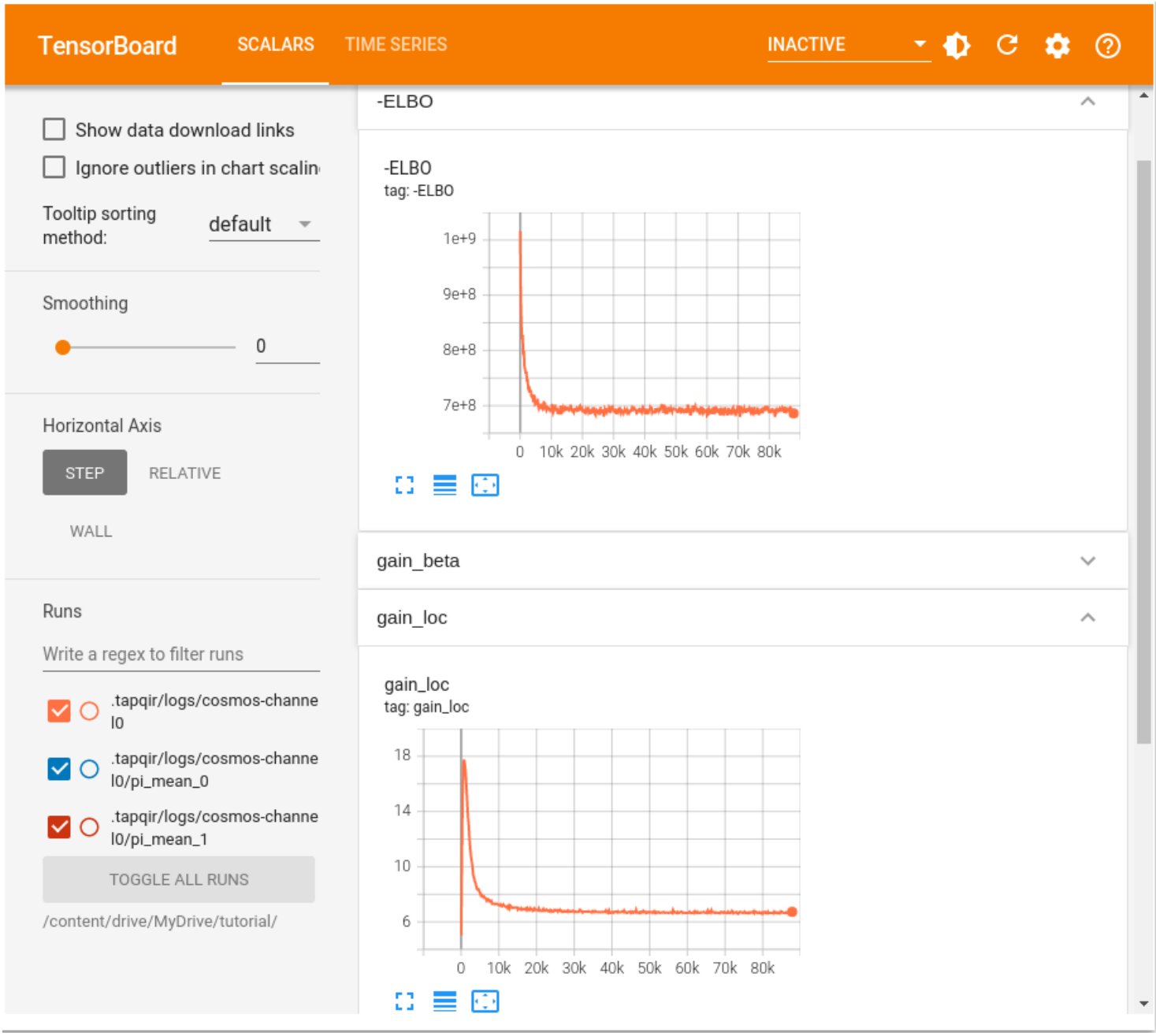
If you get an error message saying that there is a memory overflow you can decrease either frame batch size (e.g., to `128` or `256`) or AOI batch size (e.g., to `5`).

Tensorboard

At every checkpoint the values of global variational parameters (`-ELBO`, `gain_loc`, `proximity_loc`, `pi_mean`, `lamda_loc`) are recorded. Fitting progress can be inspected while fitting is taking place or afterwards with the tensorboard program which shows the parameters values as a function of iteration number:

```
%load_ext tensorboard
```

```
%tensorboard --logdir /content/drive/MyDrive/tutorial/
```



Set smoothing to 0 (in the left panel) and use refresh button at the top right to refresh plots.

Plateaued plots signify convergence.

> **About number of iterations**. Fitting the data requires many iterations (about 50,000-100,000) until parameters converge. Setting the number of iterations to 0 will run the program till Tapqir's custom convergence criteria is satisfied. We recommend to set it to 0 (default) and then run for additional number of iterations if required.

View results

This is not supported in the Colab version of Tapqir yet.

Advanced settings

Tapqir settings can be directly accessed and modified through the configuration file `config.yaml` under `.tapqir` sub-folder of the working directory. It also contains additional options that are not available through the GUI.

Offset

Offset data region (square) can be edited using three variables:

- `offset_x`: left corner of the square (default is 10 pixels)
- `offset_y`: top corner of the square (default is 10 pixels)
- `offset_P`: size of the square (default is 30 pixels)

Bin size for the offset intensity histogram by default is 1. The bin size can be increased (try 3 or 5; odd number) to make the histogram sparser which will speed up fitting.

- `bin_size`: offset intensity histogram bin size (default is 1)

Prior distributions

Parameters of prior distirbutions (Eqs. 6a, 6b, 11, 12, 13, 15, and 16 in Ordabayev et al., 2022):

- `background_mean_std` (default 1000): standard deviation of the HalfNormal distribution in Eq. 6a
- `background_std_std` (default 100): standard deviation of the HalfNormal distribution in Eq. 6b
- `lamda_rate` (default 1): rate parameter of the Exponential distribution in Eq. 11
- `heiht_std` (default 10,000): standard deviation of the HalfNormal distribution in Eq. 12
- `width_min` (default 0.75): minimum value of Uniform distribution in Eq. 13
- `width_max` (default 2.25): maximum value of Uniform distribution in Eq. 13
- `proximity_rate` (default 1): rate parameter of the Exponential distribution in Eq. 15
- `gain_std` (default 50): standard deviation of the HalfNormal distribution in Eq. 16

Colab paid products - Cancel contracts here

11s completed at 7:20 PM