

Following <https://github.com/HannesStark/FlowSite/tree/main>, with slight modifications

11/4/23 this version is updated from earlier readme which is [here](#)

1. Conda env setup

#From <https://github.com/HannesStark/FlowSite/tree/main>, adapt for CPU and activate environment also

```
conda create -c conda-forge -n flowsite rdkit python
```

#Dina added a new line to activate:

```
conda activate flowsite
```

```
pip install torch torchvision torchaudio
```

```
pip install torch_geometric
```

#Dina had to run this one below line twice because had an issue with prody wheel building the 1st time, not sure if this helped/hurt/was neutral

```
pip install pyyaml wandb biopython spyrmsd einops biopandas  
plotly prody tqdm lightning imageio
```

```
pip install e3nn
```

#Dina revises for cpus - ****This is different for GPU, GitHub has GPU command

```
pip install torch_scatter torch_sparse torch_cluster -f  
https://data.pyg.org/whl/torch-2.1.0+cpu.html
```

2. Pulling weights

Create a `pocket_gen` dir in the `Flowsite` dir

Take weights from Google drive link in Hannes' repo (under "Running on the test set with trained models") and place in `pocket_gen`

*Note difference from earlier readme where weights from Slack were used

3. Pulling PDBBind

Follow steps in “To obtain the pdbbind data” from the GitHub section “Retrain Flowsite/HarmonicFlow or run trained model on test set”/”Dataset”
(<https://github.com/HannesStark/FlowSite/tree/main#retrain-flowsiteharmonicflow-or-run-trained-model-on-test-set>)

4. Code updates

A. To avoid a logging error I had to edit pytorch code (see 10/24 Slack messages for more details)- maybe CPU-dependent?:

```
cp
/Users/dsharon/anaconda3/envs/flowsite/lib/python3.10/site-packa
ges/lightning/pytorch/loops/evaluation_loop.py
/Users/dsharon/anaconda3/envs/flowsite/lib/python3.10/site-packa
ges/lightning/pytorch/loops/evaluation_loop_old.py
```

Here is how my new code looks (older part commented out):

```
#Dina comment out 10/24/23
    #try:
    #    # some terminals do not support this
character
    #    if sys.stdout.encoding is not None:
    #        "-".encode(sys.stdout.encoding)
    #except UnicodeEncodeError:
    #    bar_character = "-"
    #else:
    #    bar_character = "-"
bar_character = "-"
```

B. Note- to save all PDB files earlier a code edit was needed (more details in earlier readme), but now Hannes added in a command line option, so no code edits are needed- but a new argument needs to be added

5. Run

In the flowsite dir, run the below command. This is the GitHub command adding `--save_all_batches` (blue highlight) so all output poses are saved and removing a couple of arguments which I had to take out to run on CPU

Also edit `--checkpoint` (blue highlight) if using different weights and edit `--run_name` (blue highlight) to what would be helpful for you

Note for CPU: I removed `--wandb` and `--num_workers` due to errors, more information on the sequences of commands I tried and errors I received is in my 12:21 PM 10/25 reply to my 11:50 AM 10/25 message

****Note for GPU I am not certain whether `--wandb` and `--num_workers` should be kept- you may want to try first running with these args

```
python -m train --run_name test_HarmonicFlow_timesplit_DistPock --run_test --checkpoint
pocket_gen/duw71q7p/checkpoints/best.ckpt --lr 1e-3 --batch_size 4 --train_split_path
index/timesplit_no_lig_overlap_train --val_split_path index/timesplit_no_lig_overlap_val
--predict_split_path index/timesplit_test --clamp_loss 10 --epochs 150 --num_inference 10
--gradient_clip_val 1 --save_inference --check_nan_grads --num_all_res_train_epochs 100000
--fake_constant_dur 0 --fake_decay_dur 0 --fake_ratio_start 0 --fake_ratio_end 0
--residue_loss_weight 0 --use_tfn --time_condition_tfn --correct_time_condition
--time_condition_inv --time_condition_repeat --flow_matching --flow_matching_sigma 0.5
--prior_scale 1 --layer_norm --tfn_detach --max_lig_size 200 --check_val_every_n_epoch 1
--cross_radius 50 --protein_radius 30 --lig_radius 50 --ns 32 --nv 8 --tfn_use_aa_identities
--self_condition_x --pocket_residue_cutoff_sigma 0.5 --pocket_center_sigma 0.2
--pocket_residue_cutoff_sigma 0.5 --pocket_center_sigma 0.2 --pocket_type ca_distance
--pocket_residue_cutoff 14 --save_all_batches
```

This is a command which would be a possible GPU starting point, with the only edit from the GitHub being adding `--save_all_batches`:

```
CUDA_VISIBLE_DEVICES="2" python -m train --run_name
test_HarmonicFlow_timesplit_DistPock --wandb --run_test --checkpoint
pocket_gen/duw71q7p/checkpoints/best.ckpt --lr 1e-3 --batch_size 4
--train_split_path index/timesplit_no_lig_overlap_train --val_split_path
index/timesplit_no_lig_overlap_val --predict_split_path index/timesplit_test
--clamp_loss 10 --epochs 150 --num_inference 10 --gradient_clip_val 1
--save_inference --check_nan_grads --num_all_res_train_epochs 100000
--fake_constant_dur 0 --fake_decay_dur 0 --fake_ratio_start 0 --fake_ratio_end
0 --residue_loss_weight 0 --use_tfn --time_condition_tfn
--correct_time_condition --time_condition_inv --time_condition_repeat
--flow_matching --flow_matching_sigma 0.5 --prior_scale 1 --layer_norm
--tfn_detach --max_lig_size 200 --num_workers 4 --check_val_every_n_epoch 1
--cross_radius 50 --protein_radius 30 --lig_radius 50 --ns 32 --nv 8
--tfn_use_aa_identities --self_condition_x --pocket_residue_cutoff_sigma 0.5
--pocket_center_sigma 0.2 --pocket_residue_cutoff_sigma 0.5
--pocket_center_sigma 0.2 --pocket_type ca_distance --pocket_residue_cutoff 14
--save_all_batches
```