**Preprocess**

Benchmark\_Data - create datasets for all representations

MACCS\_SOMS - RF and SVM classifiers for data with and without SOMS

NN - neural networks for training

pZZS - create representations from PZZS data

**Best**

Contains in depth notebooks for analysis

**Benchmark**

Contains biogrid job submitters and GCNN code

**pZZS**

BestPBT -> for prediction on new dataset

SIDENOTE (new atoms in pZZS data, problems feature vector size..)

Environments

Conda activate GNN (for preprocess)

Conda activate benchmark\_gnn (for running GCNN)

Step 1 - Preprocess data in benchmark data notebook or with scripts

Adjust for own dataset

Works with SMILES + TOX label spreadsheets

Parse into functions

Copy data from data folder to tox folder in benchmark

Step 2 - Adjust the hardcoded parts in the benchmark dataset for new data

Step 3 - Biogrid parser for running jobs

Create hyperparameter settings needed

Step 4 - Parse results with notebook parses and check for best results

Step 5 - Copy best to best folder and analyze results in notebook

**Local**

-Clone repo

-Conda activate GNN

-Start jupyter notebook

-Run Benchmarkdataset.ipynb

For new data → adjust to own dataset

Needed: SMILES and labels in csv/excel

-Copy from preprocess/data the xxx\_rep1 to xxx\_rep4 data

-Paste data in benchmark/data/TOX

-(Adjust hardcoded parts on next page for new data)

-conda deactivate GNN

-conda activate benchmark\_gnn

-Run every representation once for 1 epoch (to create the train/test/val split)

-All splits are equal because of hardwired seed

-Example for local run with python:

python main\_TUs\_graph\_classification.py --dataset CMR\_Rep4 --config ./configs/test/TOXGIN0.json

Configs can be created with the JobSubmitterGin

Run JobSubmitterGIN once with standard settings to create config

Copy paste part from python until end and run for Rep1 to Rep4

Close notebook

Git add --all

Git commit -a -m “added representations/splits”

Git push

**Biogrid**

Git pull

Conda activate benchmark\_gnn

Jupyter notebook

Run JobSubmitterXXX with desired grid search for hyperparameters

Check with “bjobs” in command prompt if jobs are done (4~12 hours)

Copy “CheckResults.ipynb” to benchmark/out/tox/results

Run CheckResults and find best accuracy

Copy best checkpoints and config of accuracy to best and run particular in depth notebook

Copy data folder from benchmark to best or pZZS folder for analysis

HARDCODED IN BENCHMARK (information for adding new dataset)

**Add rep2 to list (hydro/arom rep has extra bond labels)**

1 (TOX.py rule 171)

if self.name in ["PBT\_Rep2", "PBT\_Repn2", "CMR\_Rep2"]:

edges\_dimension = 8

Else:

edges\_dimension = 4

**Add list with representations and if statement with num\_graphs for dataset**

2 (TOX.py rule 239)

pbt = ['PBT\_Rep1', 'PBT\_Rep2', 'PBT\_Rep3', 'PBT\_Rep4']

pbtn = ['PBT\_Repn1', 'PBT\_Repn2', 'PBT\_Repn3', 'PBT\_Repn4']

cmr = ['CMR\_Rep1', 'CMR\_Rep2', 'CMR\_Rep3', 'CMR\_Rep4']

data\_dir = './data/TOX'

if name in pbt:

dataset = TOXLoad(data\_dir, self.name, num\_graphs=494)

elif name in pbtn:

dataset = TOXLoad(data\_dir, self.name, num\_graphs=971)

elif name in cmr:

dataset = TOXLoad(data\_dir, self.name, num\_graphs=652)

**Add representations to list**

3 (DATA.py rule 18)

toxic\_reps = ['PBT\_Repn1', 'PBT\_Repn2', 'PBT\_Repn3', 'PBT\_Repn4', 'PBT\_Rep1', 'PBT\_Rep2', 'PBT\_Rep3', 'PBT\_Rep4', 'CMR\_Rep1', 'CMR\_Rep2', 'CMR\_Rep3', 'CMR\_Rep4']

# handling for (TOX) molecule dataset

if DATASET\_NAME in toxic\_reps:

return TOXDataset(DATASET\_NAME)