Sequence design project legacy info

January 2021

Generating data

Data consists of canonicalized coordinates centered on a particular residue (masked) with all backbone atom information and side-chain information for the non-baseline (main) model. Backbone (BB) only data has atom type indicator only. Main model has atom type indicator, indicator of BB atom, and residue type indicator for non center residue.

These coordinates are then voxelized in the loop during training

Code → See lab repository protein_seq_des_training

Commands to generate data:

For backbone (BB) only dataset:

```
python load_and_save_bb_coords.py --save_dir PATH_TO_SAVE_DATA --pdb_dir PATH_TO_PDB_FILES --workers NUM_WORKERS
--log_dir PATH_TO_LOG_DIR --txt PATH_TO_DOMAIN_TXT_FILE
```

For main dataset:

```
python load_and_save_coords.py --save_dir PATH_TO_SAVE_DATA --pdb_dir PATH_TO_PDB_FILES --workers NUM_WORKERS  --log_dir
PATH_TO_LOG_DIR --txt PATH_TO_DOMAIN_TXT_FILE
```

Inputs are .txt files with domains for train/test set. If you don't have pdb files already downloaded, script will download those for you.

Importantly -- the models retain all 'context'. Even if it is training on a single domain, it retains context from the full biological assembly to train the residue prediction network.

NOTE -- all data is available here on GCP, publicly available with requester pays.

Training baseline model

Description → Model to predict residue type and rotamer angles in autoregressive way without any neighboring side chain atoms

Code → See lab repository protein_seq_des_training

Data → Train data here. Test data here. chunk size is 10K

Training → Across 8 V100 GPUs, DataParallel mode

> python train_autoreg_chi_baseline.py --batchSize 4096 --workers 12 --lr 1.5e-4 --validation_frequency 100 --save_frequency 1000 --log_dir PATH_TO_LOG_DIR --data_dir PATH_TO_DATA
>
> Data_dir should contain folders train_s95_chi_bb and test_s95_chi_bb with train/test data respectively.

Training main model

Description → Model to predict residue type and rotamer angles in autoregressive way conditioned on backbone and neighboring side chain atoms

Code → See lab repository protein_seq_des_training

Data → Train data here. Test data here. chunk size is 10K

Training → Across 8 V100 GPUs, DataParallel mode

python train_autoreg_chi.py --batchSize 2048 --workers 12 --lr 7.5e-5 --validation_frequency 200 --save_frequency 2000 --log_dir PATH_TO_LOG_DIR --data_dir PATH_TO_DATA

Data_dir should contain folders train_s95_chi and test_s95_chi with train/test data respectively.

Running design

All info on how to run the design script is on github here (only design) along with link to pretrained models.

Init model → epoch 11 (zero-indexed, true epoch 12), timestep 1000

Baseline models (ensemble) → Epoch 14 steps 2000, 3977 (last step), epoch 15 steps 2000, 3977