

raptcouple_test

Code of RaptCouple, a unsupervised machine learning of SELEX data. Raptcouple learns structure and fitness information from SELEX data.

Environment

```
mamba env create -f environment.yaml
```

install plmc

After installing [plmc](#) as the instruction, please edit `PLMC_TO_PATH` variable in `src/coupling.py`.

Description

[example/Ishida2020/Ishida2020.ipynb](#) contains the whole workflow described below.

Data preparation

```
.
├── DRR201870.fa
├── DRR201871.fa
├── DRR201872.fa
├── DRR201873.fa
└── config.yaml
```

Preprocessing

`python scripts/merge_and_cutadapt_all_rounds.py` performs preprocessing based on `config.yaml`.

This script performs:

1. cutadapt
2. sequence merging
3. remove seqs of small count

Preprocessing part of `config.yaml` should follow this format:

```
Preprocess_parameters:
  N_random: 40
  adapter_3: TATGTGCGCATACATGGATCCTC
  adapter_5: TAATACGACTCACTATAGGGAGAACTTCGACCAGAAG
```

```
data_dir: ./example/Ishida2020/data
fasta_annotation:
  DRR201870.fa: Ishida2020-3R
  DRR201871.fa: Ishida2020-4R
  DRR201872.fa: Ishida2020-5R
  DRR201873.fa: Ishida2020-6R
# remove_lowcount: # remove sequences which count is smaller than
# mincount.
#   DRR201870.fa: 1
#   DRR201871.fa: 1
merged_fasta: Ishida2020.count.ann.all_selex.fa
```

This is an example of preprocessing.

```
python scripts/merge_and_cutadapt_all_rounds.py --config
./example/data/Ishida2020/config.yaml
```

MSA constraction by jackhmmer

Set the MSA parameters in config.yaml as follows:

```
MSA_parameters:
  all_fasta: ./example/Ishida2020/data/Ishida2020.count.ann.all_selex.fa
  target_id: Ishida2020-6R-1-2626-55264.43
  save_dir: ./example/Ishida2020/outputs
  prefix: ""
  iters: 10
  F1: 0.02
  F2: 0.001 # 1e-3
  F3: 0.0001 # 1e-4
  T: 5
  domT: 5
  incT: 5
  incdomT: 5
  print_result: true
```

Generate a multiple sequence alignment (MSA) using jackhmmer:

```
python ./scripts/run_jackhmmer.py --config
./example/data/Ishida2020/config.yaml
```

Note: We found these parameters work for most SELEX data. But, if the MSA depth is insufficient, consider relaxing the jackhmmer parameters (iters, F1, F2, F3, T, domT, incT, incdomT). For further details, please refer to the HMMER user guide.

Potts model training

Potts model part of `config.yaml` should follow this format:

```
Potts_parameters:
  input_fasta: ./example/Ishida2020/outputs/Ishida2020-6R-1-2626-
55264.43.msa
  sim_threshold: 0.05
  vocab: AUGC.
  iters: 200
  suffix: ""
  print_result: true
```

Train the Potts model by running:

```
python scripts/train_potts.py --config
./example/data/Ishida2020/config.yaml
```

Folding with coupling scores

Once you have obtained coupling scores from the Potts model training, predict the 2D structure by using the coupling information. For example:

Prediction of mutation effects

Evaluate the impact of mutations on sequence fitness and structure with:

`mutations.txt` should list mutations in a standard format (e.g., A15G). The script outputs predicted effects for each mutation, facilitating the analysis of mutation impact.

Citation

If you use this code, please cite the following paper:

```
@article{sumi2025raptcouple
  title={Learning structure and fitness of RNA discovered by SELEX},
  author={Sumi, Shunsuke and Kawahara, Daiki and Hada, Yuki and Yoshii,
Tatsuyuki and Adachi, Tatsuo and Saito, Hirohide and Hamada, Michiaki},
```

```
journal={Journal Name},      % 論文掲載ジャーナル名に置き換えてください
volume={XX},                 % 巻番号に置き換えてください
number={YY},                 % 号番号に置き換えてください
pages={ZZ-ZZ},               % ページ番号に置き換えてください
year={2025},                 % 発行年に置き換えてください
note={Correspondence should be addressed to: mhamada@waseda.jp,
hirosaito@iqb.u-tokyo.ac.jp}
}
```