
DSTAR
사용 설명서

범례

1. Input 저장하기

2. Target 저장하기

3. Fingerprint 변환 및 ML model 훈련하기
   3 – 1. Fingerprint 변환만 하기 or ML model 훈련만 하기
   3 – 2. 기타 Argument Custom

4. Fingerprint 치환
   4 – 1. 치환할 원소 및 치환 방법 선택
   4 – 2. 기타 Argument Custom

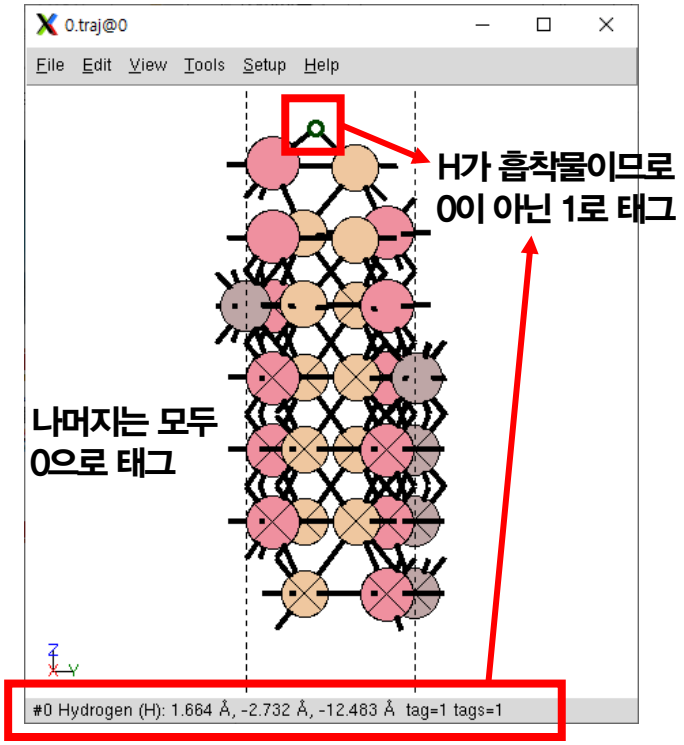
5. 치환된 Fingerprint Target 예측

1. Input 저장하기

1) 학습시키고자 하는 데이터 (흡착물이 붙은 표면)을 .traj 파일 형식으로 변환 후 ~/DSTAR/atoms/ 폴더에 저장

```
/home/ahrehd0506/git/DSTAR/atoms
(base) [ahrehd0506@haber atoms]$ ls
0.traj   11.traj  13.traj  15.traj  17.traj  19.traj  20.traj  2.traj  4.traj  6.traj  8.traj
10.traj  12.traj  14.traj  16.traj  18.traj  1.traj   21.traj  3.traj  5.traj  7.traj  9.traj
```

※ 이때 **흡착물에 해당하는 원자들**은 **0이 아닌 정수값**으로 tag가 되어있어야 하며, 흡착물을 제외한 원자는 모두 0으로 tag가 되어있어야 함!



2. Target 저장하기

1) name 과 target이라는 column을 가지는 Dataframe을 생성. 이 두 column이 있으면 다른 건 있든 말든 상관 없음.

2) name column에는 표면데이터 .traj 파일명에 해당하는 id를 넣어 주고, target column에는 해당 id에 대응되는 target 값 (아마 흡착에너지) 를 넣어 줌.

12.traj 이라는 표면 데이터의 target 값인 4를 넣어준 모습

	name	target
0	1	1
1	2	2
2	3	3
3	4	4
4	5	5
5	6	6
6	7	7
7	8	8
8	9	9
9	0	1
10	10	2
11	11	3
12	12	4
13	13	5
14	14	6
15	15	7
16	16	8
17	17	9
18	18	1
19	19	2
20	20	3
21	21	4

3) 값을 넣은 Dataframe을 target.csv 이라는 파일명으로 ~/DSTAR/atoms/ 폴더에 저장

```
(base) [ahrehd0506@haber atoms]$ pwd
/home/ahrehd0506/git/DSTAR/atoms
(base) [ahrehd0506@haber atoms]$ ls
0.traj   11.traj  13.traj  15.traj  17.traj  19.traj  20.traj  2.traj  4.traj  6.traj  8.traj  target.csv
10.traj  12.traj  14.traj  16.traj  18.traj  1.traj   21.traj  3.traj  5.traj  7.traj  9.traj
```

3. Fingerprint 변환 및 ML model 훈련하기

1) ~/DSTAR/ 폴더에서 python train.py 입력

2) 아래와 같이 Fingerprint 생성 –› 모델 훈련 –› 성능 평가 진행되면 성공

```
Namespace(algo='total', atom_path='./atoms/', convert_only=False, data_path='./data/fp.csv', desire_range=0.1, desire_target=-0.67, load_data=False, model_path='./model/', subs_p
ath='./subs/fp/', test=False, test_ratio=0.2)

Initiate Conversion Atoms to Fingerprint...
100%|| 23/23 [00:03<00:00,  6.27it/s]
Successfully Generate Fingerprint!

2022-07-13 11:33:29,004 - INFO - Start Gradient Boosting Regression
2022-07-13 11:33:30,608 - INFO - Start Kernel Ridge Regression
2022-07-13 11:33:30,836 - INFO - Start ElasticNet Regression
/home/ahrehd0506/miniconda3/lib/python3.8/site-packages/sklearn/linear_model/_coordinate_descent.py:645: ConvergenceWarning: Objective did not converge. You might want to increas
e the number of iterations, check the scale of the features or consider increasing regularisation. Duality gap: 3.875e+00, tolerance: 4.991e-02
  model = cd_fast.enet_coordinate_descent(
2022-07-13 11:33:30,924 - INFO - Start Support Vector Regression
2022-07-13 11:33:30,962 - INFO - Start Gaussian Process Regression
/home/ahrehd0506/miniconda3/lib/python3.8/site-packages/sklearn/gaussian_process/kernels.py:430: ConvergenceWarning: The optimal value found for dimension 0 of parameter k1__cons
tant_value is close to the specified upper bound 100000.0. Increasing the bound and calling fit again may find a better value.
  warnings.warn(
/home/ahrehd0506/miniconda3/lib/python3.8/site-packages/sklearn/gaussian_process/kernels.py:420: ConvergenceWarning: The optimal value found for dimension 0 of parameter k2__leng
th_scale is close to the specified lower bound 1e-05. Decreasing the bound and calling fit again may find a better value.
  warnings.warn(
==================================================

Best performance model : Kernel Ridge Regressor
MAE : 5.066529515330492
RMSE : 6.360076086100101

==================================================
```

3) ~/DSTAR/ 폴더에 Train_results.csv 와 Test_results.csv 파일이 생성.

name, target, pred column 을 가지며 각각 표면 데이터 파일 id,
실제 target 값, 모델에 의해 예측된 target 값을 표현

4) 훈련에 사용된 Fingerprint가 ~/DSTAR/data/ 폴더에 fp.csv 파일로 생성. Active motif-based Fingerprint로 변환된 표면 정보를 저장

	name	FNN	Same	Sub	target
2	0	{'Co': 1, 'Si': 1}	{'Co': 5, 'Si': 3}	{'Al': 2}	1

0.traj이라는 표면 구조가 FNN / SNN_same / SNN_sub 에 포함된 원자 개수로 변환된 것을 확인

※ DSTAR 실행 시 --data-path {your_data_path} argument 로 원하는 경로 및 이름으로 저장 가능

```
python main.py --data-path /data/test.csv
```

5) 훈련에 사용된 ML model과 Scaler가 ~/DSTAR/model/{train-date}/model.pkl & scaler.pkl 로 저장

```
(base) [ahrehd0506@haber model]$ ls
2022-06-01/  2022-06-02/  2022-06-03/  2022-07-13/
(base) [ahrehd0506@haber model]$ cd 2022-07-13/
(base) [ahrehd0506@haber 2022-07-13]$ ls
model.pkl  scaler.pkl
```

3 – 1. Fingerprint 변환만 하기 or ML model 훈련만 하기

1) 표면 데이터를 Fingerprint 변환만 하고 싶을 경우
`--convert-only` argument 사용 (Default : False)

```
(base) [ahrehd0506@haber DSTAR]$ python train.py --convert-only
Namespace(algo='total', atom_path='./atoms/', convert_only=True,
th='./subs/fp/', test=False, test_ratio=0.2)

Initiate Conversion Atoms to Fingerprint...
100%|
Successfully Generate Fingerprint!
```

2) 이미 존재하는 Fingerprint로 model 학습만 시키고 싶을 경우
`--load-data --data-path {your-data-path}` argument 사용
(data–path Default : ./data/fp.csv)

```
(base) [ahrehd0506@haber DSTAR]$ python train.py --load-data --data-path ./data/test.csv
Namespace(algo='total', atom_path='./atoms/', convert_only=False, data_path='./data/test.
path='./subs/fp/', test=False, test_ratio=0.2)

Load Motif From test.csv...
Initiate Conversion Motifs to Fingerprint...
22it [00:00, 148.06it/s]
Successfully Generate Fingerprint!

2022-07-13 13:44:51,304 - INFO - Start Gradient Boosting Regression
2022-07-13 13:44:53,923 - INFO - Start Kernel Ridge Regression
2022-07-13 13:44:53,943 - INFO - Start ElasticNet Regression
```

3 – 2. 기타 Argument Custom

1) 훈련시킬 표면 데이터가 있는 폴더 직접 지정
`--atom-path {your_atom_path}` argument 사용 (Default : ./atoms/)

2) 훈련 시 사용할 Train / Test ratio 조정
`--test-ratio {test_ratio}` argument 사용 (Default : 0.2)

3) 훈련 시 사용할 ML algorithm 선택
`--algo {algorithm}` argument 사용 (Default : 모두 훈련 후 제일 좋은 모델 사용)

사용가능한 알고리즘 : GBR / KRR / ELN / SVR / GRP

4. Fingerprint 치환

1) 앞 과정에서 생성된 Fingerprint 파일을 이용하여 원소 치환

2) ~/DSTAR/ 폴더에서 python subs.py 입력. 아래와 같이 Successfully ~ 라고 나오면 성공

```
(base) [ahrehd0506@haber DSTAR]$ python subs.py
Namespace(bi_only=False, data_path='./data/fp.csv', get_bi=False, subs_path='./subs/fp/', subs_type='comb')

Initiate Active Motif Subsititution
100%|
Successfully Generate New Active Motifs!!
```

※ --data-path {your_data_path} argument로 치환 시킬 Fingerprint 파일 선택 가능 (Default : ./data/fp.csv)

3) 치환된 Fingerprint 파일들이 ~/DSTAR/subs/{your_subs_path}/ 폴더에 원소 조합별로 생성 (Default : ~/DSTAR/subs/fp/)

```
(base) [ahrehd0506@haber fp]$ ls
Ag_Ag.csv  Al_Al.csv  As_Au.csv  Au_Cu.csv  Co_In.csv  Cr_Os.csv  Cu_Ru.csv  Fe_W.csv   Ge_Os.csv
Ag_Al.csv  Al_As.csv  As_Co.csv  Au_Fe.csv  Co_Ir.csv  Cr_Pb.csv  Cu_Sb.csv  Fe_Zn.csv  Ge_Pb.csv
```

4 – 1. 치환할 원소 및 치환 방법 선택

1) ~/DSTAR/sub.py 열어 Line 31~34 에서 정의되는
el_set_A / el_set_B 리스트 수정하는 것으로 치환할 원소 선택 가능

```
def subs(args):
    el_set_A =['Ag','Al','As','Au','Co','Cr','Cu','Fe','Ga','Ge','In','Ir','Mn','Mo',
               'Ni','Os','Pb','Pd','Pt','Re','Rh','Ru','Sb','Se','Si','Sn','Ti','V',
               'W','Zn']
    el_set_B = []
```

el_set_A 리스트의 default 원소들 (전이금속 30개)
el_set_B의 역할은 후술

2) `--subs-type {subs_type}` argument 사용하여 치환 방법 선택 (Default : comb)

- comb : el_set_A의 원소들 N 개 내에서 Combination ($_{N}\Pi_{2}$)
- prod : el_set_A 와 el_set_B 원소들의 Product
  ex) Metal + Chalcogen 조합으로 치환하고 싶을 경우 아래와 같이 수정 후 prod 사용

```
el_set_A =['Ag','Al','As','Au','Co','Cr','Cu','Fe','Ga','Ge','In','Ir','Mn','Mo',
           'Ni','Os','Pb','Pd','Pt','Re','Rh','Ru','Sb','Se','Si','Sn','Ti','V',
           'W','Zn']
el_set_B = ['O','S','Se','Te']
```

4 – 2. 기타 Argument Custom

1) 치환된 Fingerprint 들 저장할 폴더 선택
`--subs-path {your_subs_path}` argument 사용 (Default : ./subs/fp/)

2) 치환할 원소 조합 생성 시 단일 원소 물질 제외하여 생성 ($_NC_2$)
`--bi-only` argument 사용 (Default : False)

3) 훈련할 표면 데이터로 생성한 Fingerprint 중 이원소 물질 추출하여 저장
`--get-bi` argument 사용 (Default : False)

(이 arg. 없어도 자동으로 이원소 물질만 치환)

추출된 이원소 물질 Fingerprint 파일은
{your_data_path}/{data_name}_binary.csv 로 생성
(Default : ./data/fp_binary.csv)

5. 치환된 Fingerprint Target 예측

1) ~/DSTAR/ 폴더에서 `python train.py --test --model-path {your_model_path}` 입력. 아래와 같이 나오면 성공

```
(base) [ahrehd0506@haber DSTAR]$ python train.py --test --model-path ./model/2022-07-13/
Namespace(algo='total', atom_path='./atoms/', convert_only=False, data_path='./data/fp.cs
3/', subs_path='./subs/fp/', test=True, test_ratio=0.2)

Load Model 2022-07-13
Initiate Screening...
100%|
Successfully Predict Ideal Surface Density!!
```

※ model path는 반드시 입력해줘야 함

※ `--subs-path {your_subs_path}` argument로 예측할 Substituted Fingerprint 경로 설정 가능 (Default : ./subs/fp/)

2) {your_subs_path} 의 Fingerprint 파일들에 pred column (예측 값) 업데이트

	name	FNN	Same	Sub	target	pred
0	5	{'Ag': 1}	{'Ag': 2}	{'Ag': 1}	0	4
1	5	{'Ag': 1}	{'Ag': 2}	{'Ag': 1}	0	4

동일한 id가 2개 있는 이유는 한 물질에서 A_xB_y / B_xA_y 둘 다 생성되기 때문

4) ~/DSTAR/ 폴더에 dens.csv 파일 생성. elements 와 density 라는 column을 가짐
desnity는 해당 elements 조합으로 치환된 표면들 중 예측 값이 desire target range를 만족하는 표면의 비율 –〉 높을수록 보고자 하는 반응에 활성이 좋을 것으로 예측된다.

A	B
elements	density
Ga_Ge	0
Fe_Pd	0
As_Ru	0
Re_Zn	0
Fe_Ti	0
Fe_Se	0
Ga_Pb	0
Fe_Fe	0
Au_Re	0
Cr_Mn	0
Ag_Ga	0

※ `--desire-target {target_value}` argument로 이상적인 표면으로 선택되기 위한 target 값을 설정
(Default : – 0.67 , CO2RR의 이상적인 CO 흡착에너지)

※ `--desire-range {range_value}` argument로 이상적인 표면으로 선택되기 위한 target 오차 범위를 설정
(Default : ± 0.1)