ACTIVELEARNING_FRAMEWORK

- **isotherm-config.bat**

  환경변수 등록 후 사용

  ```
  isotherm-config startproject 프로젝트이름
  ```

- **프로젝트 디렉터리 구조**
  - [생성된 Project]
    - config.json

      RASPA_DIR 위치 정보

      isotherm-config.bat파일 위치 정보
    - kinetic-diameter.json

      kinetic-diameter 정보들
    - mof-database
      - coremof2019.csv ...

        > MOF이름들(RASPA share폴더에 있는 그대로 적혀있어야 함)

        > PLD등 구조적 파라미터들 또는 추가로 사용할 인풋 피쳐들
    - manage.py

      ```
      python manage.py create
      > 298
      > 1
      > MOF DB csv 이름 선택
      > 가스선택 (RASPA_DIR share molecule에 있는 것들 이름)
      > kinetic-diameter 가스 선택 (kinetic-diameter.json 안에 있는 것들 이름)
      # 작동 세부사항
      1) 선택한 MOF DB csv의 MOF들을 PLD기준으로 kinetic diameter와 비교하여 스크리닝
      2) MOF DB csv 이름의 폴더 작성
      3) MOF DB csv 이름 폴더 내부에 가스_온도_압력_screened_by_kinetic_diameter가스이름 폴더 생성
      4) 해당 폴더에 screened 된 MOF DB csv를 low_pressure_gcmc.csv / active_learning_gcmc.csv로 만듦
      >> 이때 low_pressure_gcmc.csv에는 completed/uptake/calculation_time이 존재해야함
      >> active_learning_gcmc.csv에는 iteration/uptake/calculation_time 열이 존재해야함
      5) 해당 폴더에 isotherm-config.bat파일이 있는 폴더에서 low_pressure_gcmc.py와 active_learning_gcmc.py를 가져
      6) 해당 폴더에 isotherm-config.bat파일이 있는 폴더에서 gcmcconfig.json과 base.input을 가져옴
      ```

    - [MOF Database csv이름]
      - [가스_온도_압력_kinetic_diameter분자이름]
        - base.input

          ```
          SimulationType              MonteCarlo
          NumberOfCycles              {NumberOfCycles}
          NumberOfInitializationCycles  {NumberOfInitializationCycles}
          PrintEvery                  {PrintEvery}
          RestartFile                 no
          ```

```
Forcefield                  {Forcefield}
UseChargesFromCIFFile {UseChargesFromCIFFile}

Framework 0
FrameworkName {MOF}
UnitCells {UNITCELL}
ExternalTemperature    {TEMP}
ExternalPressure    {PRESSURE}

Component 0 MoleculeName            {GAS}
          MoleculeDefinition       {MoleculeDefinition}
          FugacityCoefficient      1.0
          TranslationProbability   0.5
          RotationProbability      0.5
          ReinsertionProbability   0.5
          SwapProbability          1.0
          CreateNumberOfMolecules  0
```

- gcmcconfig.json

```
{
    "NumberOfCycles" : 20000,
    "NumberOfInitializationCycles" : 10000,
    "PrintEvery" : 1000,

    "UseChargesFromCIFFile" : null,
    "ExternalTemperature" : null ,
    "ExternalPressure" : null ,
    "Forcefield" : "GarciaPerez2006ForceField",

    "GAS" : null,
    "MoleculeDefinition" : "ExampleDefinitions",

    "CUTOFFVDW" : 14,
    "CUTOFFCHARGECHARGE" : 14,
    "CUTOFFCHARGEBONDDIPOLE" : 14,
    "CUTOFFBONDDIPOLEBONDDIPOLE" : 14
 }
```

- low_pressure_gcmc.csv

```
>> MOF_NAME / PLD / 구조파라미터들 / Uptake / Calculation Time / Completed
>> 아직 GCMC가 수행되지 않았으면 completed / uptake / calculation time 열이 비어있어야함
```

- low_pressure_gcmc.py

```
## create
1) low_pressure_gcmc라는 하위폴더 생성
2) 해당 폴더에 low_pressure_gcmc.csv에 있는 MOF이름들로 하위폴더 생성 (이미 디렉터리 존재하면 패스)
3) MOF 이름 하위 폴더 생성함과 동시에 simulation.input파일 생성
## run --ncpus
1) low_pressure_gcmc.csv에서 completed가 기재되지 않은 MOF들 리스트 추출
2) MOF 리스트 shuffle(비슷한 이름일 경우에 연속으로 오래걸리는 경우가 있어서)
3) MOF에 대해 병렬적으로 GCMC 수행
4) GCMC가 끝날때 마다 Output/.data에서 uptake값과 calculation time값 추출
5) low_pressure_gcmc.csv에 기재 후, completed True로 바꾸기
```

```
## low_pressure_gcmc.csv작성 규칙
GCMC가 완전히 수행이되고 uptake값을 crop했으면 uptake값을 기재하고, calculation time값을 기재하고, com
```

- active_learning_gcmc.csv

```
>> MOF_NAME / PLD / 구조파라미터들 / Uptake / Calculation Time / Iteration / initial_sample
>> 아직 GCMC 계산이 완벽히 끝나지 않았으면 Iteration / uptake / calculation time 열이 비어있어야함
```

- active_learning_config.json

```
{
                "initial_fraction" : 0.01,
        "target_fraction" : 0.1,
        "n_samples" : 10,
        "neural_network":{
           "model_spec" : {
              "hidden_layers" : [
                 {
                    "hidden_dim" : 64,
                    "dropout" : 0.1,
                    "activation_func" : "ReLU"
                 },
                 {
                    "hidden_dim" : 64,
                    "dropout" : 0.1,
                    "activation_func" : "ReLU"
                 }
              ]
        },
        "dataset" : {
           "BATCH_SIZE" : 64
        },
        "training": {
           "max_epoch" : 500,
           "patience" : 30,
           "learning_rate" : 1e-3
        },
        "prediction" : {
           "mcd_numbers" : 20
        }
     }
```

- active_learning_gcmc.py

```
## inital_gcmc
#### create --initial_mode (quantile, random..)
1) active_learning_config.json에서 initial_fraction값을 가져오기
2) 모프 샘플링된 리스트 확보
   > quantile일때
      <low_uptake> <pld> 등 csv에 있는 구조적파라미터들 선택할수 있게
      이후 해당 지표로 quantile 기반 균일 샘플링
   > random일때
      랜덤 샘플링 진행
3) initial_gcmc 폴더 생성
4) 하위 폴더에 샘플링된 모프들의 이름으로 폴더 생성
5) 폴더를 생성할 때, simulation.input도 생성
```

6) active_learning_gcmc.csv의 initial_sample열에 True로 기재

run --ncpus
1) active_learning_gcmc.csv에서 initial_sample가 기재되었는데 iteration에 0이 기재되지 않은 것 리스트들 추출
2) MOF 리스트 shuffle(비슷한 이름일 경우에 연속으로 오래걸리는 경우가 있어서)
3) MOF에 대해 병렬적으로 GCMC 수행
4) GCMC가 끝날때 마다 Output/.data에서 uptake값과 calculation time값 추출
5) active_learning_gcmc.csv에 기재 후, iteration 0으로 바꾸기

init_model
1) active_learning_gcmc.csv에서 iteration이 0인 것들만 추출해서 데이터셋으로 구축
2) Model build
3) Model Training
4) model_weight폴더 생성 > Model 가중치 저장 model_weight_00000.pth

active_gcmc
1) active_learning_gcmc.csv에서 iteration이 null이 아닌 것들만 추출
> 개수확인 / fraction 계산 / config에서 target_fraction보다 큰지 확인
2) iteration값중 가장 높은 값을 식별후 5자리 정수 문자열로 바꿈
3) Model build
4) model_weight_{문자열}.pth을 불러와서 Model에 입히기
5) active_learning_gcmc.csv에서 iteration이 null인것들 추출 (unlabeled pool)
6) unlabeled pool을 MCD로 예측하게 수행
7) 불확실성 측정 및 기록 (uncertainty_{문자열}.csv : MOF이름 / uncertainty)
8) 불확실성 기반 sort및 상위 n개 (n은 config에서 n_samples로 지정) 지정
9) active_learning_gcmc폴더에 들어감
10) 하위 폴더에 iteration{문자열+1} 폴더 생성
11) 해당 폴더에 상위 n개 MOF이름으로 폴더생성
12) 생성과 동시에 simulation.input 생성

13) 모두 생성시 gcmc 병렬로 진행

>> 이를 반복

- prediction_gcmc.py

1) active_learning_gcmc.csv를 열기
2) iteration중 가장 높은 iteration 식별
3) 해당 iteration 으로 다섯자리 정수 문자열 생성
4) 해당 문자열로 model가중치 불러오기
5) 모델 빌드 및 로드
6) active_learning_gcmc에서 iteration이 null인 데이터들 확보
7) 예측 MCD
8) MOF_NAME / uptake / type (predicted / gcmc)으로 predicted.csv 만들기
9) active_learning_gcmc.csv에서 iteration이 기재된 것들은 모두 predicted.csv에서 gcmc로 처리 및 uptake값 기재
나머지는 predicted값기재