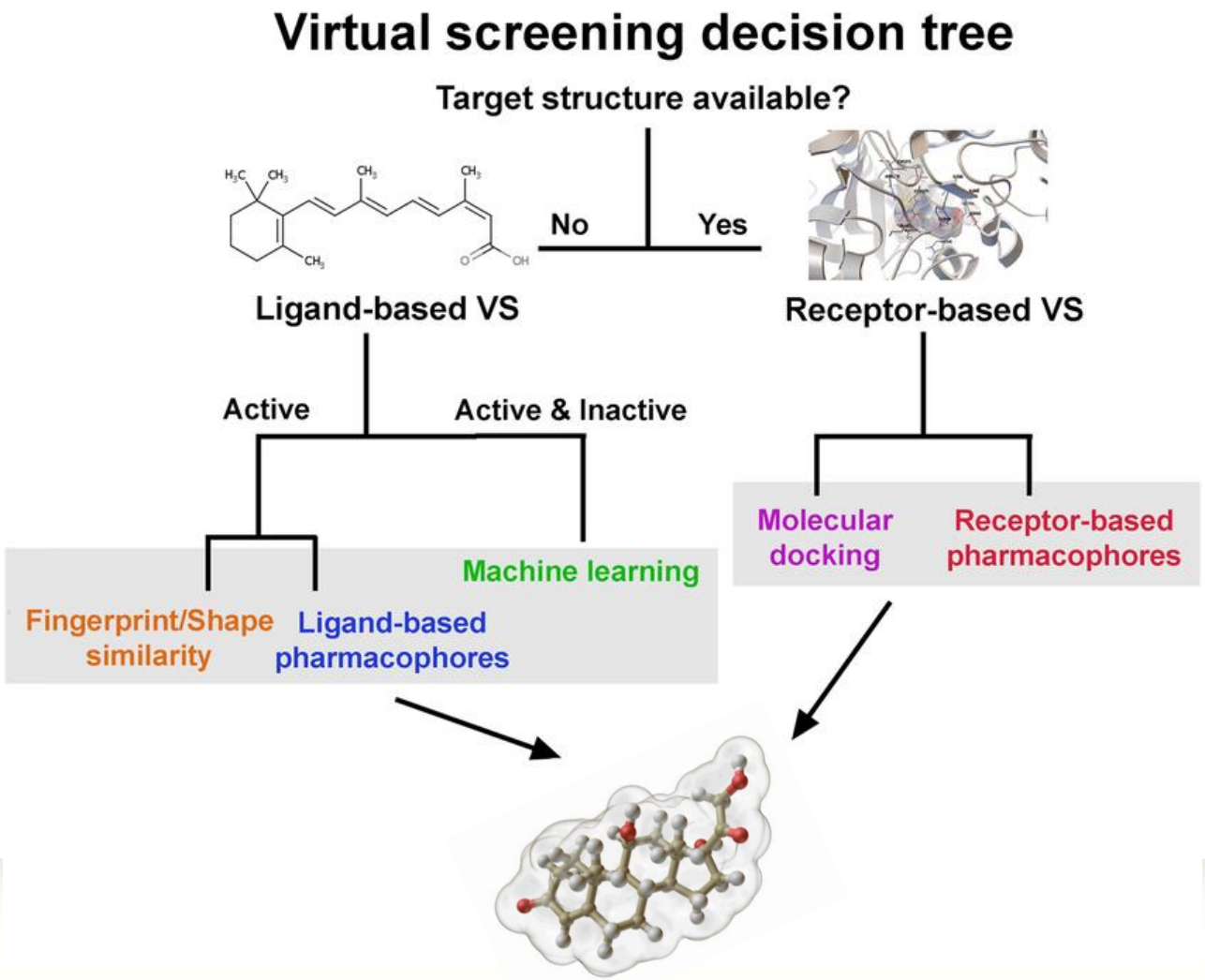
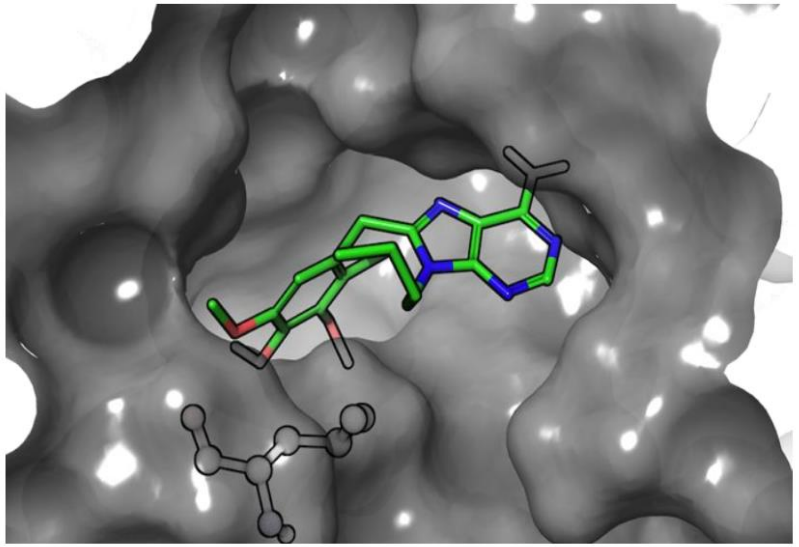
- 리간드 기반 가상 스크리닝

- 리간드 기반 가상 스크리닝



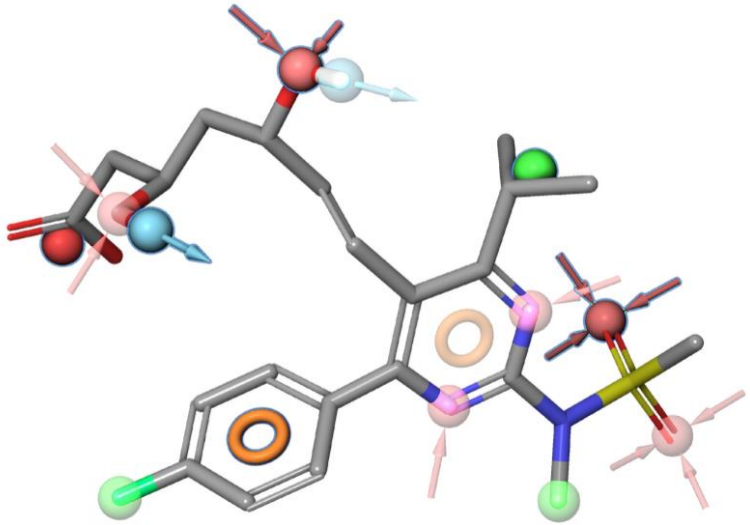
Parasitology, 147(6):1-68

- 구조 기반 vs 리간드 기반 가상 스크리닝



Structure-Based Virtual Screening (SBVS)

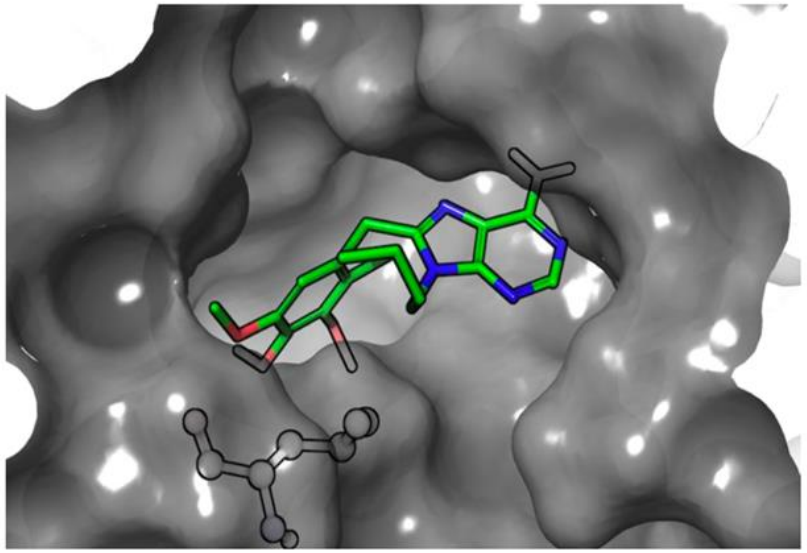
- Structure of Target
- Ligand-binding site known
- (optional) ligand/hit bound

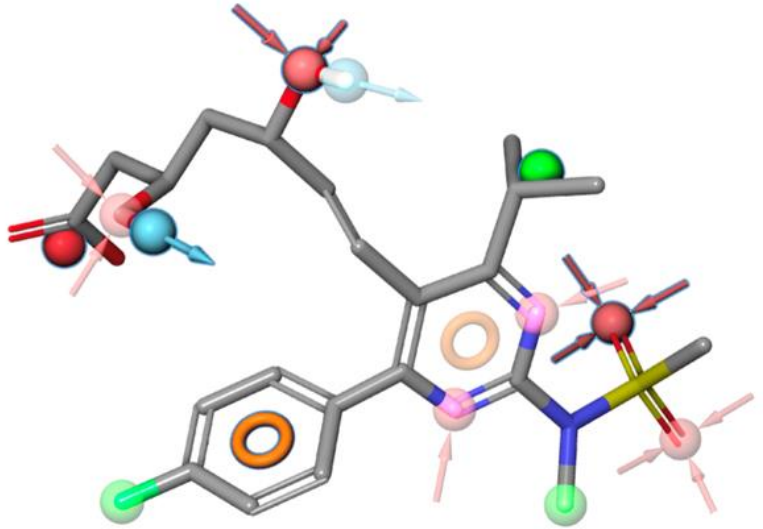


Ligand-Based Virtual Screening (LBVS)

- A single known hit
- multiple known hits
- (optional) active conformation

- 구조 기반 vs 리간드 기반 가상 스크리닝





Structure-Based Virtual Screening (SBVS)

3D

- Docking
- Pharmacophore screening

Ligand-Based Virtual Screening (LBVS)

2D

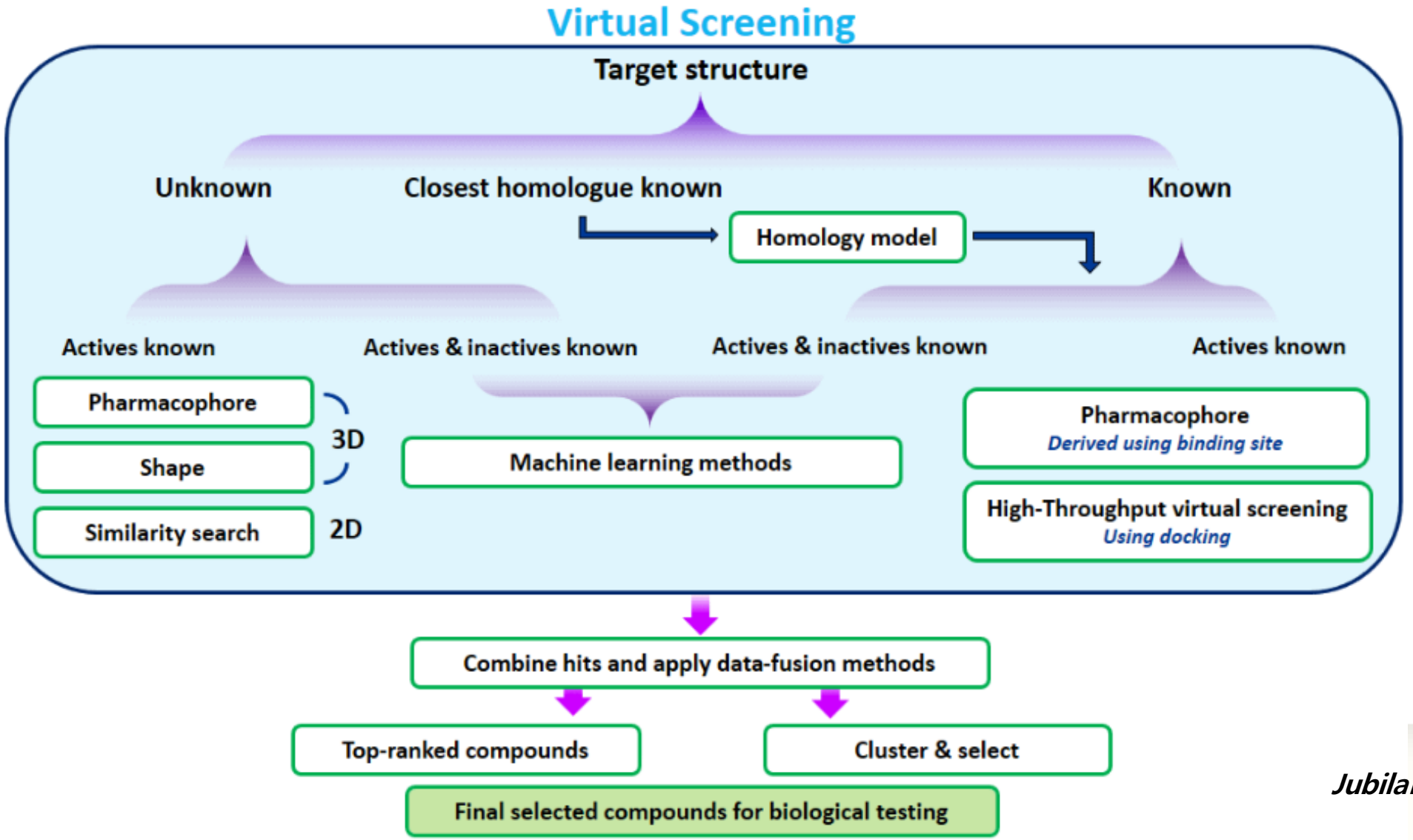
Fingerprint searching
2D pharmacophore
2D QSAR

3D

Shape-based
3D pharmacophore
3D QSAR

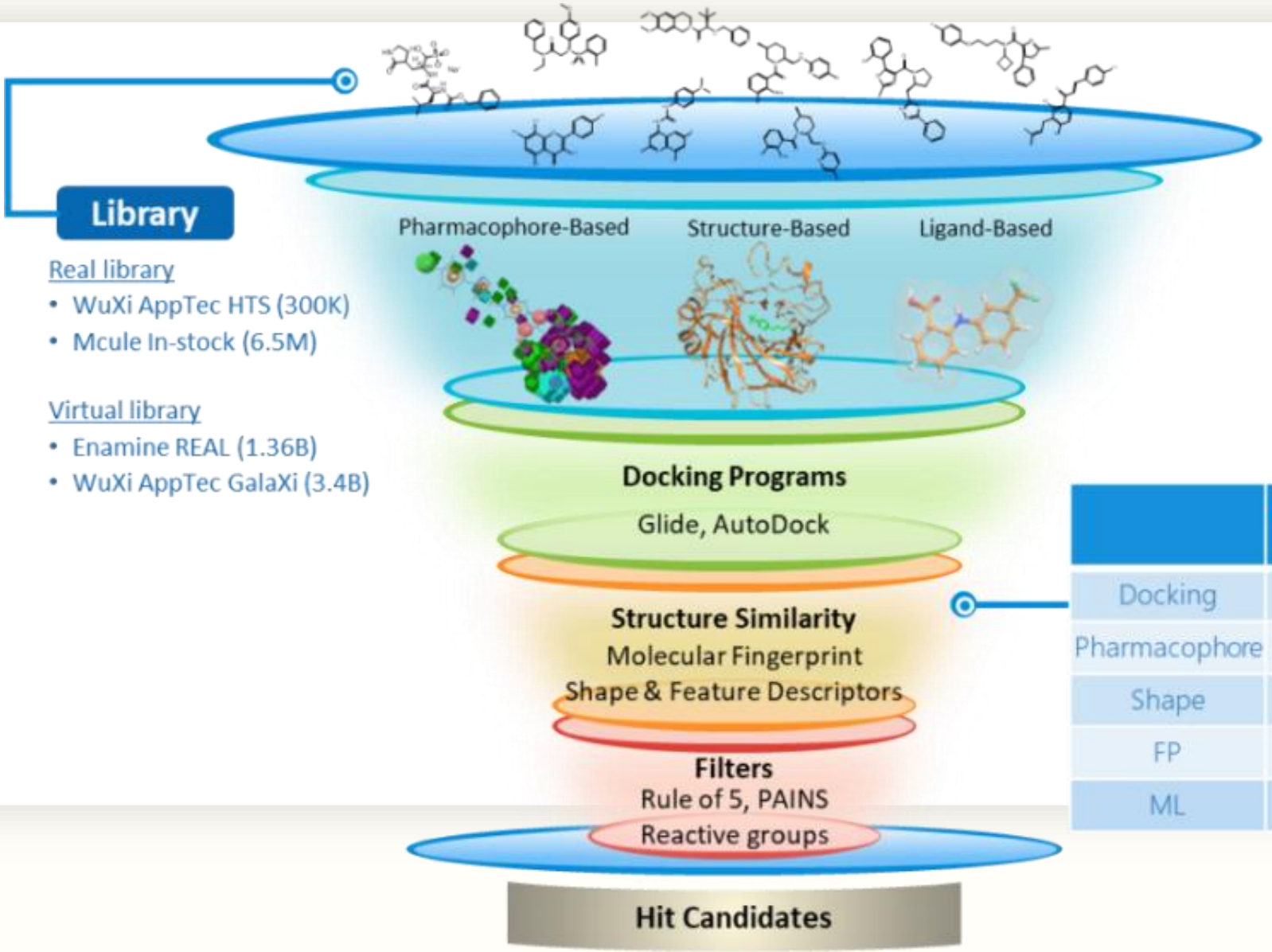
Schrodinger

- **Virtual Screening Process**



Jubilant Biosys

- **Virtual Screening Process**

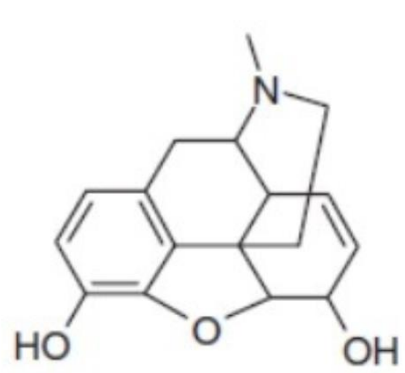


	Chemical Space	Time consuming
Docking	Millions	Days
Pharmacophore	Millions	Days
Shape	Billions	Hours
FP	Billions	Hours
ML	Billions	Hours

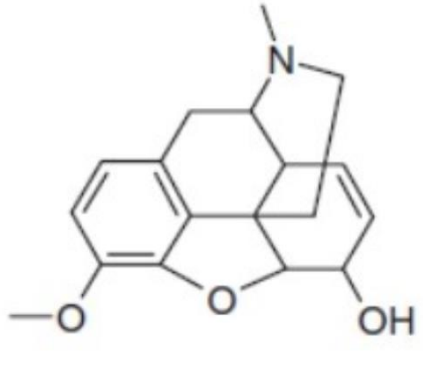
WuXi Biology

- 리간드 유사도

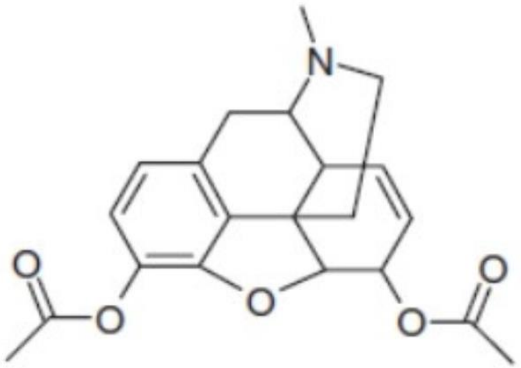
Structurally similar molecules are assumed to have similar biological properties



Morphine

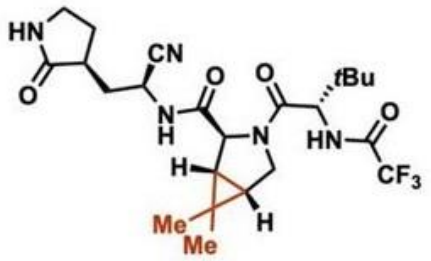


Codeine



Heroin

- 리간드 기반 가상 스크리닝



Nirmatrelvir (Covid-19)

Query molecule

Ligand library

Screening Database

Encoding

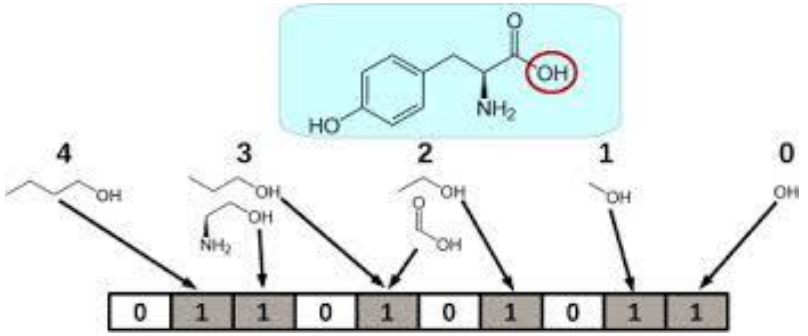
Scoring

Post-processing and selection

Rank Index	Prediction Score	QED Score	SA Score	Docking Score (by SMINA)
1	−6.68	0.24	0.55	−12.44
2	−6.59	0.29	0.76	−12.68
3	−6.20	0.50	0.61	−12.38
4	−6.06	0.34	0.67	−12.68
5	−5.97	0.40	0.68	−12.42
6	−5.89	0.45	0.56	−12.62
7	−5.86	0.50	0.66	−12.43
8	−5.59	0.24	0.84	−12.32
9	−5.44	0.30	0.73	−12.31
10	−5.30	0.57	0.61	−12.34

- **Fingerprint 유사도 기반 가상탐색**

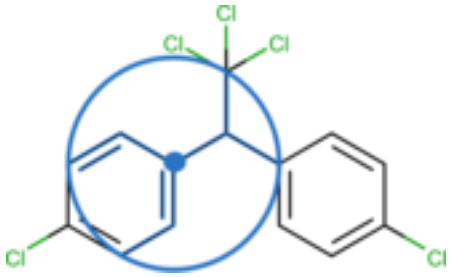
- 정의: 화합물의 구조적 특징을 이진 벡터로 표현하여 유사성을 비교하는 방법
- 방법
  - Fingerprint 생성: 화합물의 2D 구조를 바탕으로 이진 벡터(예: MACCS, ECFP)를 생성
  - 유사성 계산: Tanimoto 계수를 사용하여 화합물 간 유사성을 계산
- 장점
  - 속도: 빠른 연산이 가능
  - 효율성: 대규모 라이브러리에서도 효과적
- 단점
  - 정보 손실
  - 중복 발생 가능 : 서로 다른 화학구조가 동일한 Bit에 있을 수 있다.

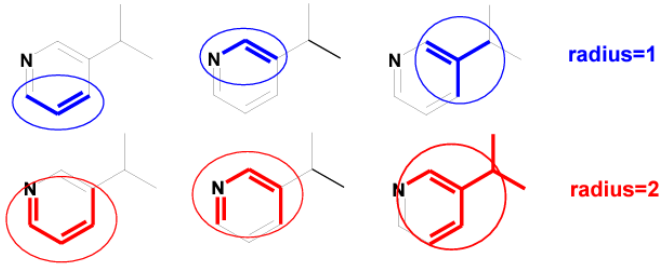


- **Fingerprint**

Morgan/ECFP

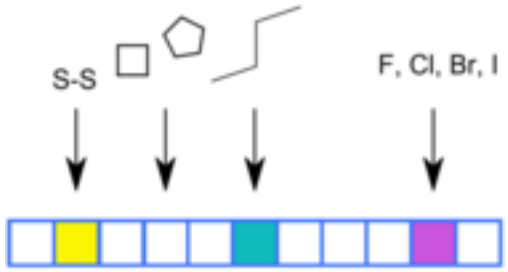
- Hash Atom types and bond from atom within given **radius.**
- Fingerprint **bit number** can be modified : 1024, 2048



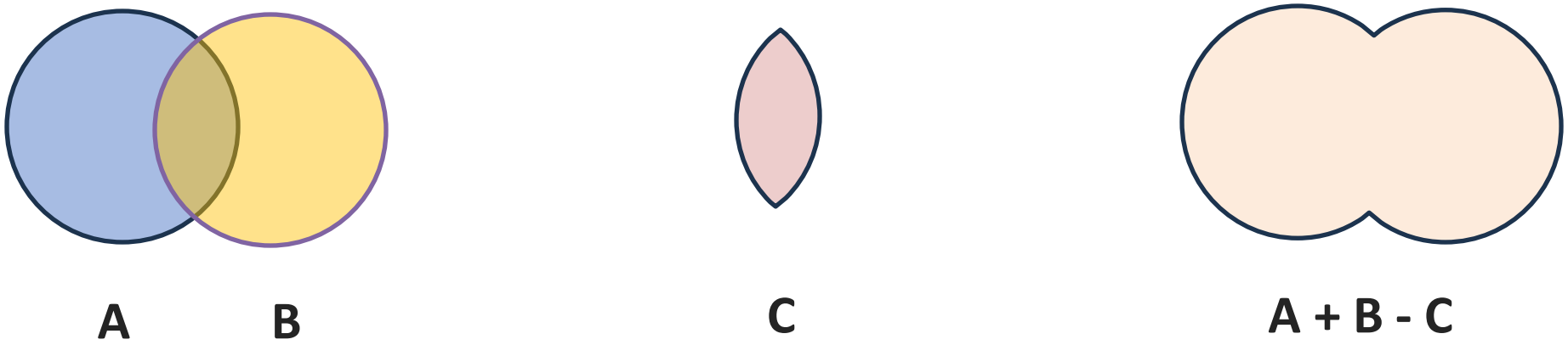


MACCS

- consist of 166 predefined structural fragments



- **유사도 : similarity coefficient**

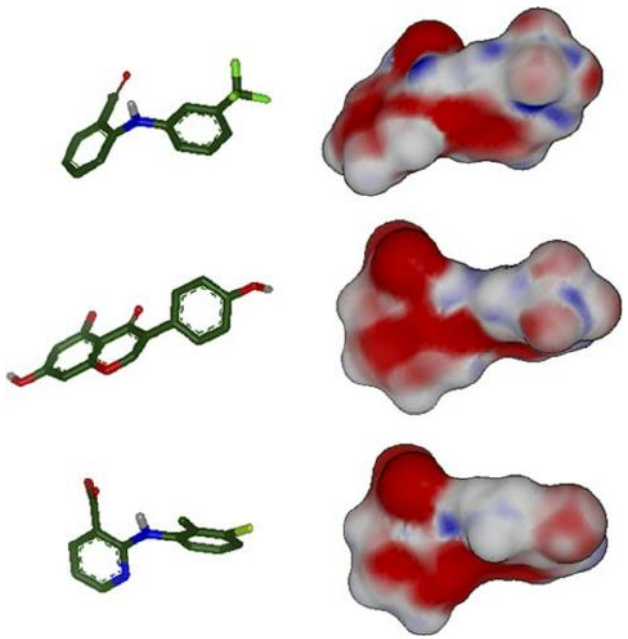


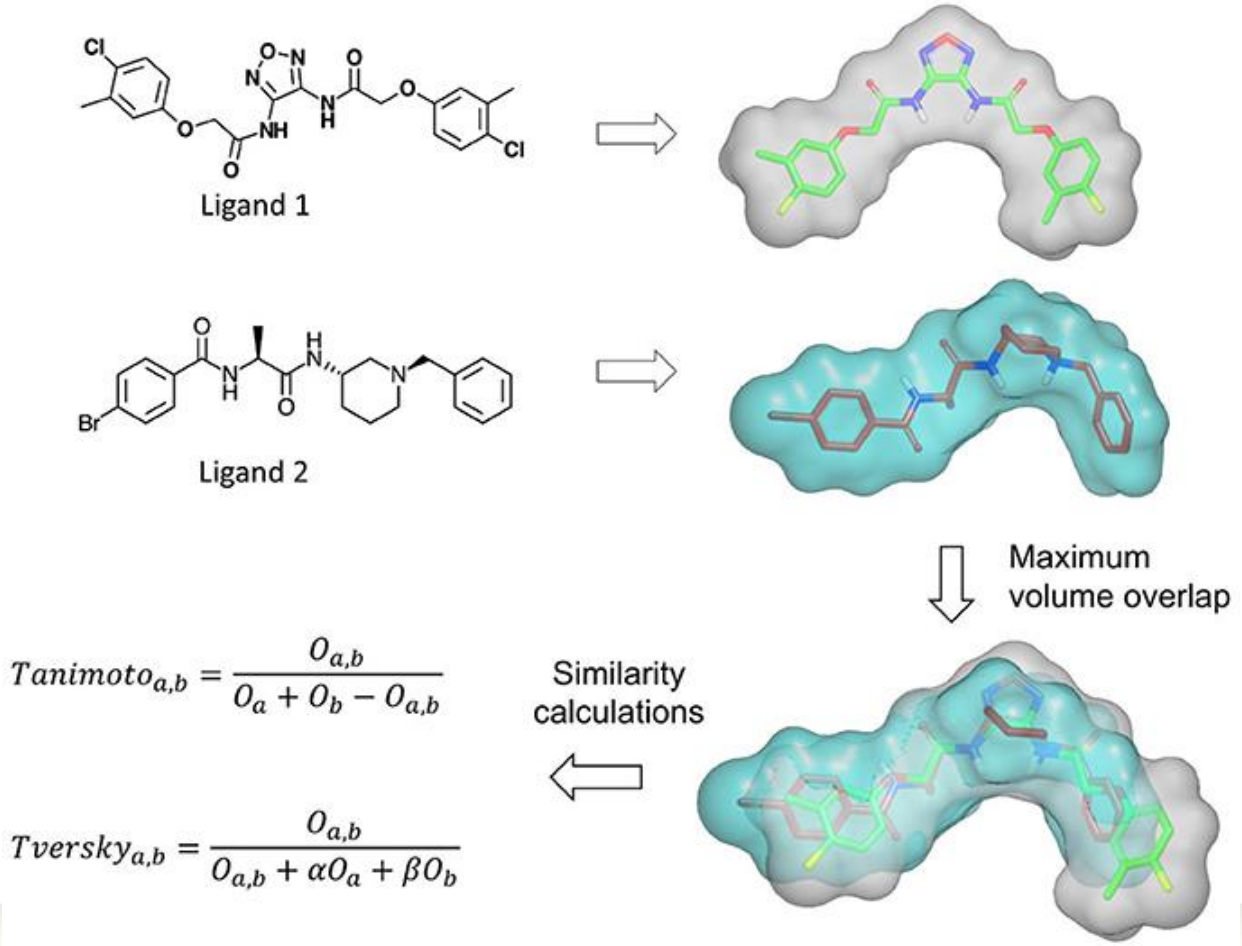
$$\text{Tanimoto Similarity} = \frac{C}{A+B-C}$$

$$\text{Dice Similarity} = \frac{2C}{A+B}$$

$$\text{Cosine Similarity} = \frac{C}{\sqrt{A+B}}$$

- **3D 구조 유사도 기반 탐색**



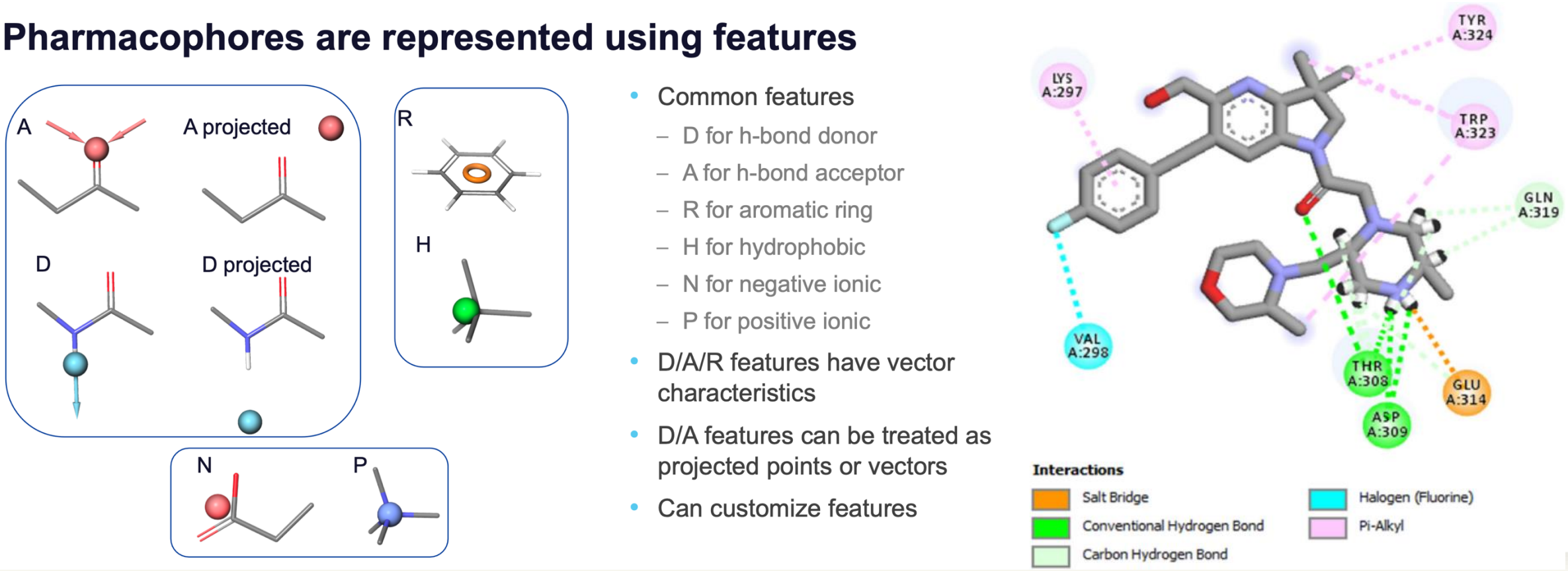


- **3D 구조 유사도 기반 탐색**

- 정의 : 타겟 단백질에 결합하는 것으로 알려진 물질의 3D 구조와 유사한 구조를 가지는 물질을 탐색
- 방법
  - 화합물 라이브러리의 3D 구조를 생성 → reference 물질과 alignment → 3D shape 비교
- 장점
  - **3D 구조 정보를 이용**하여 타겟 단백질 pocket 과의 입체적 특성을 고려하여 2D 구조에 비해 더 잘 결합할 가능성이 높은 물질 탐색이 가능
- 단점
  - 3D complex 구조가 필요
  - High computing cost : 3D conformation을 생성해야 하기 때문에 많은 계산이 필요
  - 3D 구조 유사도는 높아도 활성에는 차이가 있을 수 있음
- Software
  - Openeye, Schrodinger

- **Pharmacophore 기반 가상탐색**

- Pharmacophore는 receptor와의 binding mode를 고려한 가상 탐색이 가능



Scientific Reports volume 11, 4049 (2021)

- **Pharmacophore 기반 가상탐색**

I. 정의: 생물학적 활성에 중요한 화학적 특성을 추출하여 모델링한 3D 구조를 사용

II. 방법

- Pharmacophore 모델링: 활성 리간드에서 핵심 기능 요소(예: 수소 결합 수용체/공여체, 양성자화 중심 등)를 추출하여 3D 모델 생성

1. 유사성 검색: 후보 화합물들이 Pharmacophore 모델과 얼마나 일치하는지 평가

III. 장점

- 정밀성: 3D 정보와 기능 그룹을 고려

1. 높은 예측력: 생물학적 활성에 대한 정확한 예측 가능

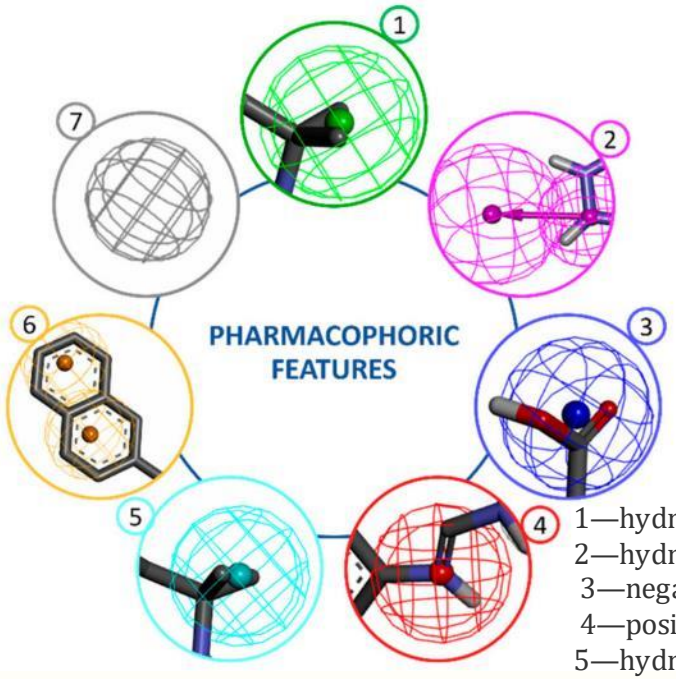
IV. 단점

- 복잡성: 모델 생성 및 유사성 평가가 복잡

1. 계산 비용: 높은 계산 자원 요구

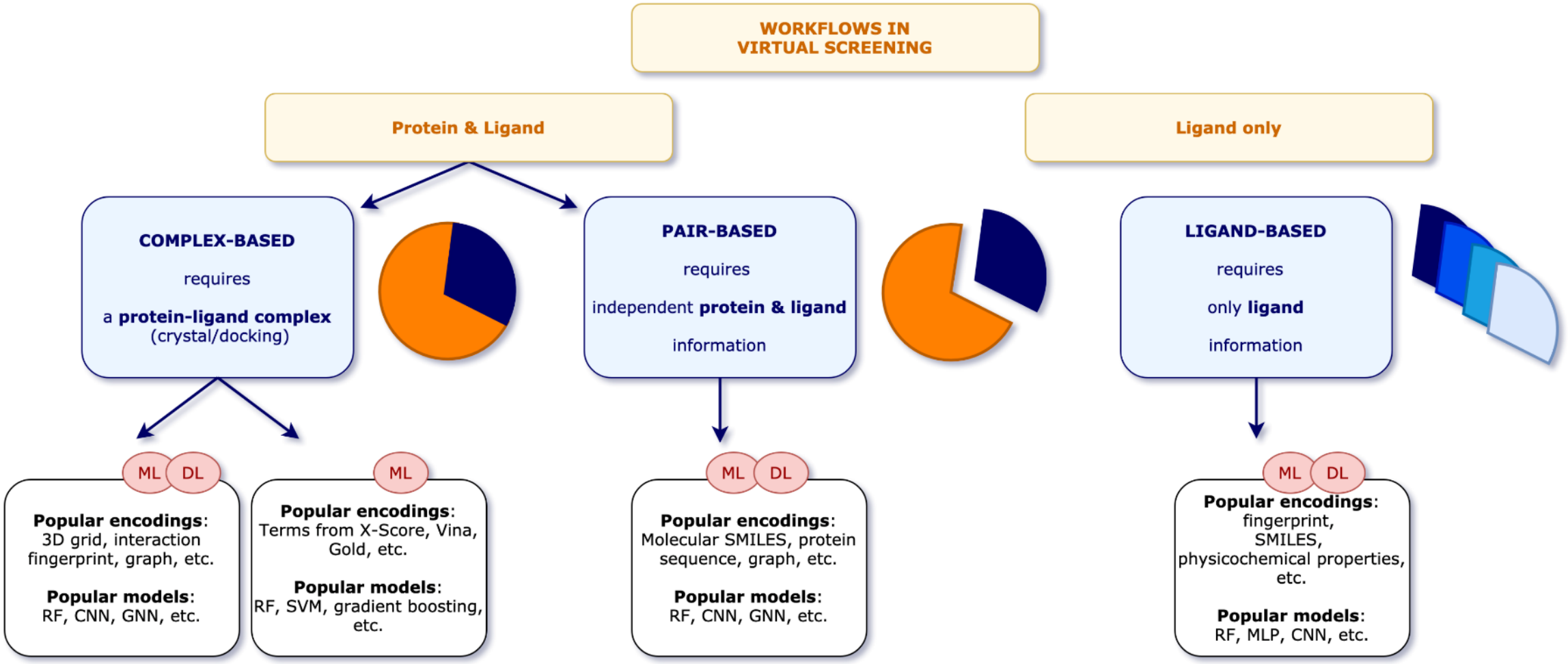
V. Software

I. Schrodinger, Openeye 외 다수



1—hydrogen bond acceptor (HBA),
2—hydrogen bond donor (HBD),
3—negative ionizable (NI),
4—positive ionizable (PI),
5—hydrophobic (H),
6—aromatic (AR),
7—exclusion volume (XVOL).

- **ML/DL 기반 가상탐색**

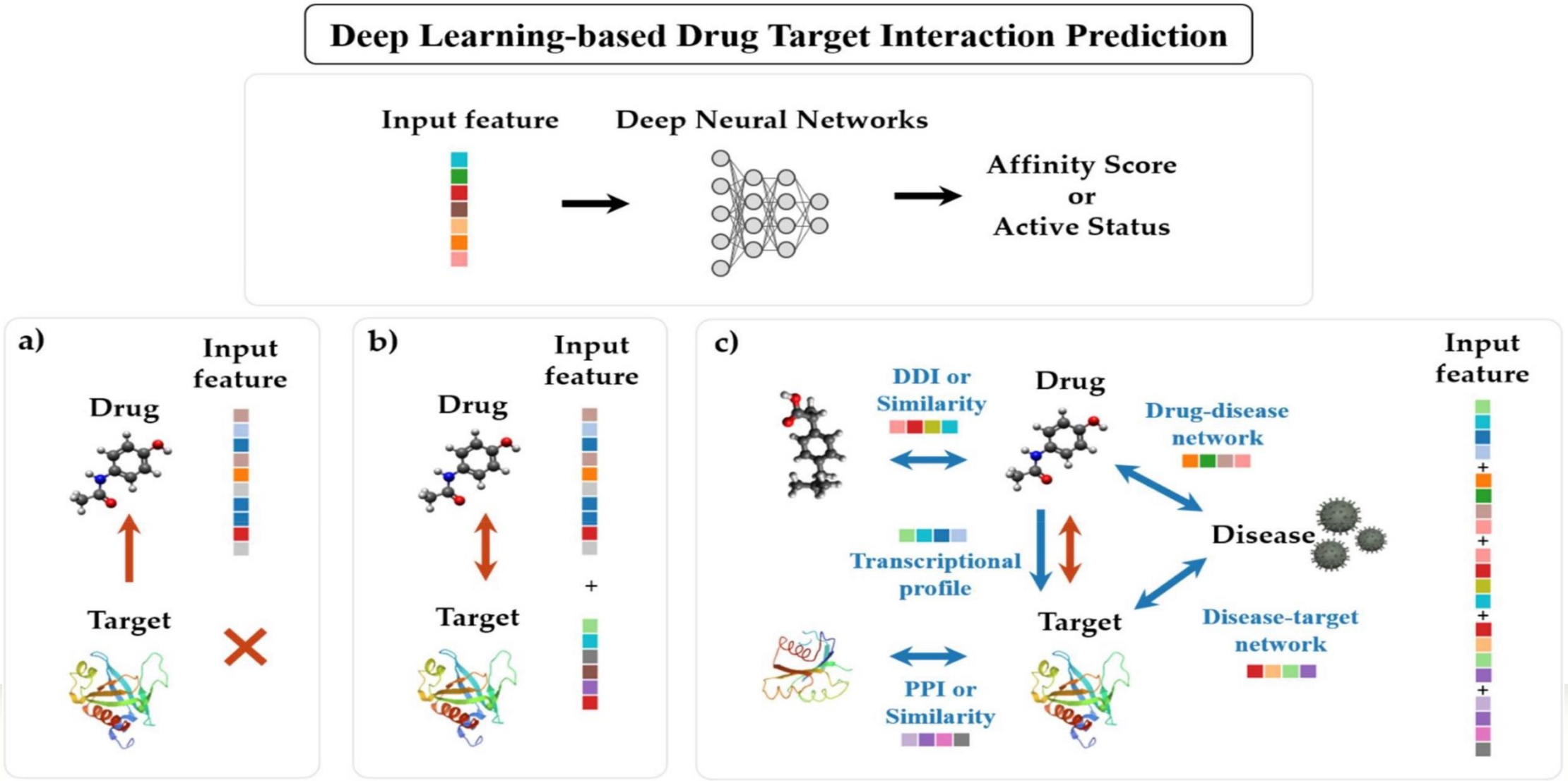


Int. J. Mol. Sci. **2021**, *22*(9), 4435

- **ML/DL 기반 가상탐색**

- 정의 : 화합물의 생물학적 활성/Interaction을 예측하는 모델을 학습
  - 분류 (classification) : 활성/비활성 화합물을 예측
  - 회귀 (regression) : 활성 값(ex: IC50)을 예측
- 방법 :
  - Machine Learning : Random Forest, SVM, Xgboost 등
  - Deep Learning : Convolution Neural Network, Graph Neural Network, AutoEncoder, Transformer
- 장점:
  - 빠른 “예측” 속도로 대규모 화합물을 빠르게 예측 가능
  - 구조/결합정도 대신 실험값 자체에 대한 예측이 가능
- 단점:
  - Lack of high quality and big data
  - Generalization of chemical space
  - Low interpretability

- **DL 기반 Drug-Target Interaction 예측**



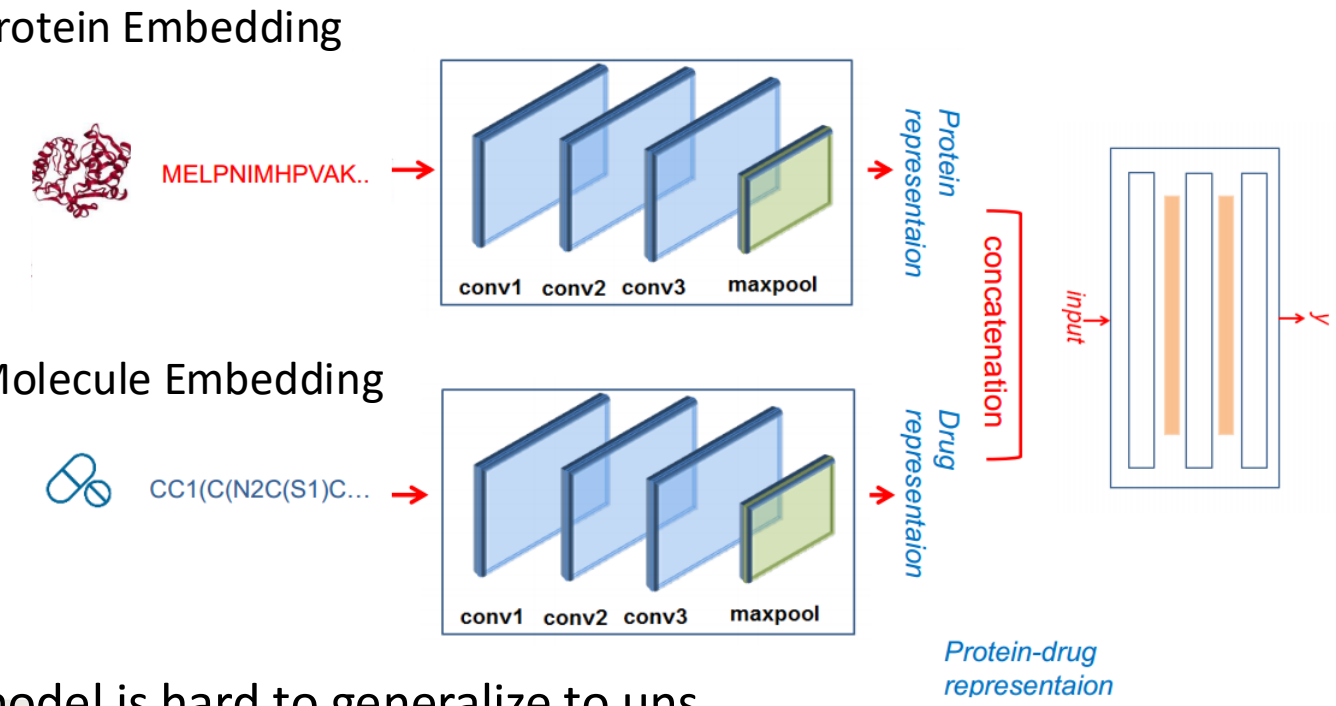
Kim, J.; Park, S.; Min, D.; Kim, W. Int. J. Mol. Sci. 22, 9983 (2021). DOI: 10.3390/ijms22189983.

- **Structure-free DTI**

- Structure dependent binding prediction is still a challenging task
  - MD or Docking for Protein – ligand binding 3D structure need binding site information
  - Protein dynamically fluctuate with binding of small molecules

- Amino acid sequence
- Protein motifs
- Protein sequence composition
- …

Structure free protein information



- Structure free binding prediction model is hard to generalize to uns een protein.

Ozturk et al., *Bioinformatics*, 34(17) i821–i829, 2018

Ligand Data

- Dataset

- Structure and bioactivity data sets

Name	Size and Contents	Availability
PDBbind v2021	structures + activities general: 27,408	http://www.pdbbind.org.cn https://www.pdbbind-plus.org.cn/
BindingDB	2,918,219 activities	https://www.bindingdb.org
BindingMOAD		https://bindingmoad.org
ChEMBL	15,598 target; 2,431,025 cpds; 20,772,701 activities	https://www.ebi.ac.uk/chembl

- **Dataset**

II. Benchmark data set w/ complex structure

Name	Size	Data Source	Label	Availability
CASF-2016	57 targets 285 complexes	PDBbind	Affinity	http://www.pdbbind.org.cn/casf.php
DUD-E	102 targets 22,886 actives 50 decoys/active	PubChem, ZINC	Active / decoy	http://dude.docking.org
MUV	17 targets ~90,000 compounds	PubChem, ZINC	Active / decoy	https://www.tu-braunschweig.de/pharmchem/forschung/baumann/muv

- **Dataset**

III.Benchmark Kinase Inhibitor data set w/o complex structure

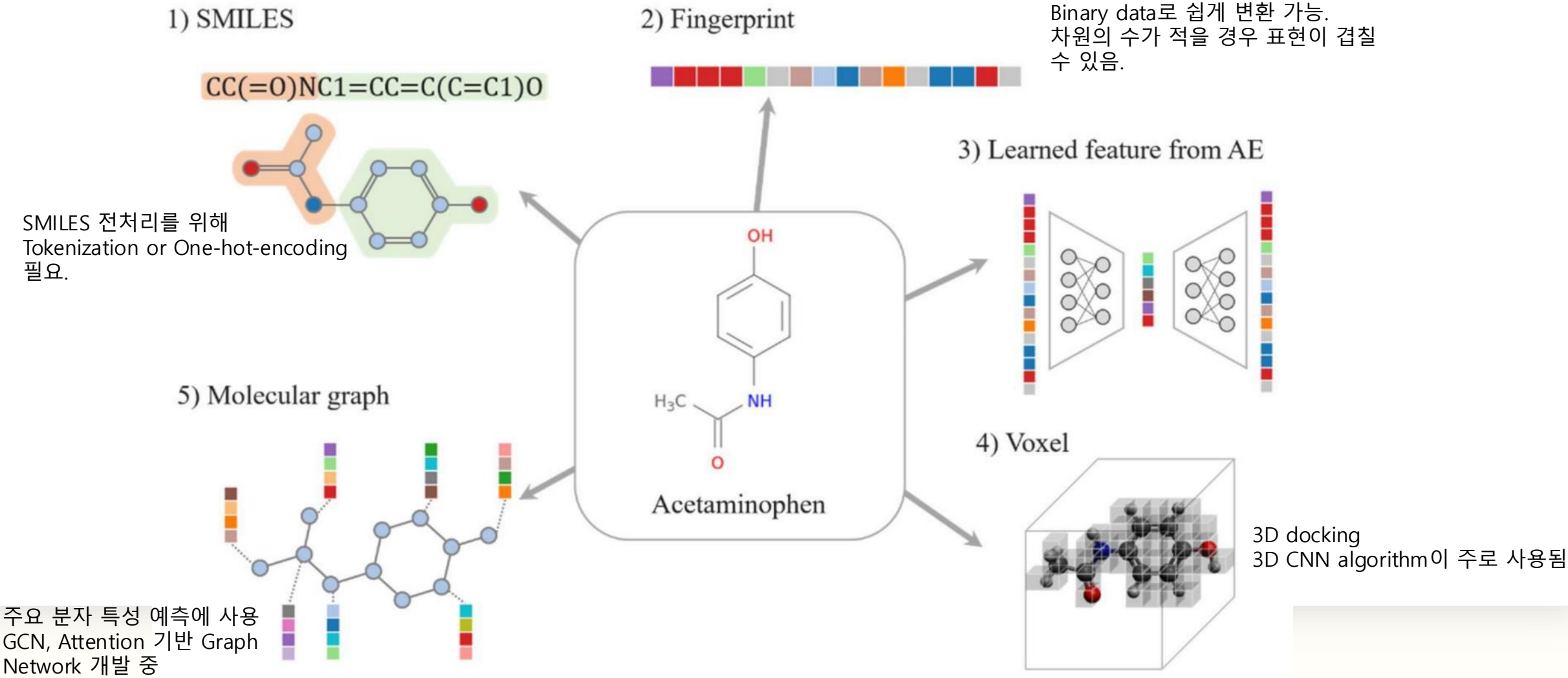
Name	Size	coverage	Label	Availability
DAVIS	25,772 DTI pairs 68 drugs 379 proteins	80%	Affinity	https://tdcommons.ai/multi_pred_tasks/dti/
KIBA	117,657 DTI pairs 2,068 drugs 229 proteins	24%	Affinity	https://tdcommons.ai/multi_pred_tasks/dti/

```
[2]: from tdc.multi_pred import DTI
data = DTI(name = 'DAVIS')
split = data.get_split()
split
```

```
Downloading...
100%|████████████████████████████████████████████████████████████████████████████| 21.4M/21.4M [00:10<00:00, 1.99MiB/s]
Loading...
Done!
```

```
[2]: {'train':         Drug_ID                                               Drug Target_ID  \
 0      11314340  Cc1[nH]nc2ccc(-c3cncc(OCC(N)Cc4ccccc4)c3)cc12      AAK1
 1      11314340  Cc1[nH]nc2ccc(-c3cncc(OCC(N)Cc4ccccc4)c3)cc12     ABL1p
 2      11314340  Cc1[nH]nc2ccc(-c3cncc(OCC(N)Cc4ccccc4)c3)cc12      ABL2
 3      11314340  Cc1[nH]nc2ccc(-c3cncc(OCC(N)Cc4ccccc4)c3)cc12     ACVR1
 4      11314340  Cc1[nH]nc2ccc(-c3cncc(OCC(N)Cc4ccccc4)c3)cc12    ACVR2A
```

- **분자 Data**



Binary data로 쉽게 변환 가능.
차원의 수가 적을 경우 표현이 겹칠 수 있음.

SMILES 전처리를 위해 Tokenization or One-hot-encoding 필요.

3D docking
3D CNN algorithm이 주로 사용됨

주요 분자 특성 예측에 사용
GCN, Attention 기반 Graph Network 개발 중

Int. J. Mol. Sci. **2021**, *22*(18), 9983

- **Data Format**

- Database에서 주로 제공하는 데이터 file format

Protein Data Bank (.pdb)
: Protein 구조 파일
Protein-Ligand complex 구조에도 사용됨

```
                                  Chain name
      Amino Acid                 /   Sequence Number
                \               /   /
       Element   \             /   /      -----Coordinates-----
                  \           /   /       X       Y       Z
ATOM      1  N   ASP L   1       4.060   7.307   5.186
ATOM      2  CA  ASP L   1       4.042   7.776   6.553
ATOM      3  C   ASP L   1       2.668   8.426   6.644
ATOM      4  O   ASP L   1       1.987   8.438   5.606
ATOM      5  CB  ASP L   1       5.090   8.827   6.797
ATOM      6  CG  ASP L   1       6.338   8.761   5.929
ATOM      7  OD1 ASP L   1       6.576   9.758   5.241
ATOM      8  OD2 ASP L   1       7.065   7.759   5.948
            \\
             Element position within amino acid
```

SMILES (.smi)
: Ligand SMILES를 text로 적은 방식
CSV, TSV 형식으로도 제공됨

```
CCCC                    butane
C(C)(CC1)C              isobutane
c(c(ccc1)ccc2)(c1)c2    anthracene
c1ccccc1                benzene
c1ccc2ccccc2c1          naphthalene
CC1CCCCC1               methylcyclohexane
C(CL)(CL)CL             trichloromethane
CN(=O)=O                nitromethane
C1=CC=CCC1              1,3-cyclohexadiene
```

Structure-Data File (.sdf)
: 물질의 2D/3D 구조 정보 저장 가능
물질의 property 정보도 반영할 수 있음

(a) 2D SDF of L-alanine

```
5950
  -OEChem-04121503402D

 13 12  0     1  0  0  0  0  0999 V2000
    5.1350   -0.2500    0.0000 O   0  0
    4.2690    1.2500    0.0000 O   0  0
    2.5369    0.2500    0.0000 N   0  0
    3.4030   -0.2500    0.0000 C   0  0
    3.4030   -1.2500    0.0000 C   0  0
    4.2690    0.2500    0.0000 C   0  0
    3.4030    0.3700    0.0000 H   0  0
    2.7830   -1.2500    0.0000 H   0  0
    3.4030   -1.8700    0.0000 H   0  0
    4.0230   -1.2500    0.0000 H   0  0
    2.0000   -0.0600    0.0000 H   0  0
    2.5369    0.8700    0.0000 H   0  0
    5.6720    0.0600    0.0000 H   0  0
  1  6  1  0  0  0  0
  1 13  1  0  0  0  0
  2  6  2  0  0  0  0
  4  3  1  6  0  0  0
  3 11  1  0  0  0  0
  3 12  1  0  0  0  0
  4  5  1  0  0  0  0
  4  6  1  0  0  0  0
  4  7  1  0  0  0  0
  5  8  1  0  0  0  0
  5  9  1  0  0  0  0
  5 10  1  0  0  0  0
M  END
> <PUBCHEM_COMPOUND_CID>
5950

> <PUBCHEM_COMPOUND_CANONICALIZED>
1
```

(b) 3D SDF of L-alanine

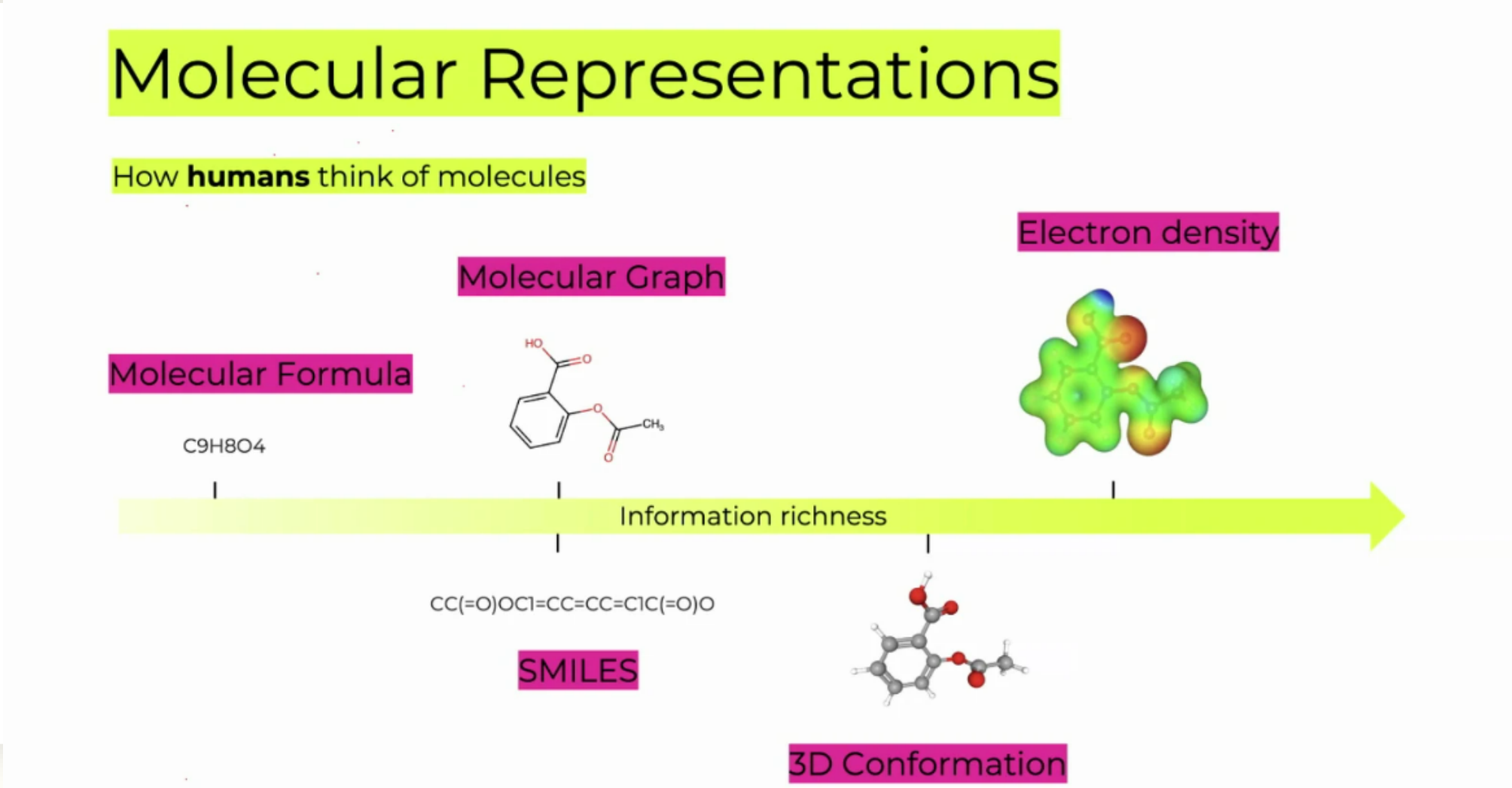
```
5950
  -OEChem-04121503413D

 13 12  0     1  0  0  0  0  0999 V2000
    1.4573   -1.0438    0.2682 O   0  0
    1.2492    1.1165   -0.4047 O   0  0
   -1.4105    1.1507    0.1821 N   0  0
   -0.7085   -0.1136    0.3937 C   0  0
   -1.3345   -1.2000   -0.4702 C   0  0
    0.7470    0.0903    0.0308 C   0  0
   -0.7666   -0.3737    1.4558 H   0  0
   -0.8580   -2.1695   -0.2878 H   0  0
   -2.4023   -1.3127   -0.2521 H   0  0
   -1.2248   -0.9797   -1.5384 H   0  0
   -2.3916    1.0420    0.4376 H   0  0
   -1.4071    1.3875   -0.8099 H   0  0
    2.4062   -0.9341    0.0447 H   0  0
  1  6  1  0  0  0  0
  1 13  1  0  0  0  0
  2  6  2  0  0  0  0
  3  4  1  0  0  0  0
  3 11  1  0  0  0  0
  3 12  1  0  0  0  0
  4  5  1  0  0  0  0
  4  6  1  0  0  0  0
  4  7  1  0  0  0  0
  5  8  1  0  0  0  0
  5  9  1  0  0  0  0
  5 10  1  0  0  0  0
M  END
> <PUBCHEM_COMPOUND_CID>
5950

> <PUBCHEM_CONFORMER_RMSD>
0.4
```

- 그 외 다양한 화학구조 데이터 file format

  : mol, gjf, pdbqt, cif, cdx, chm …

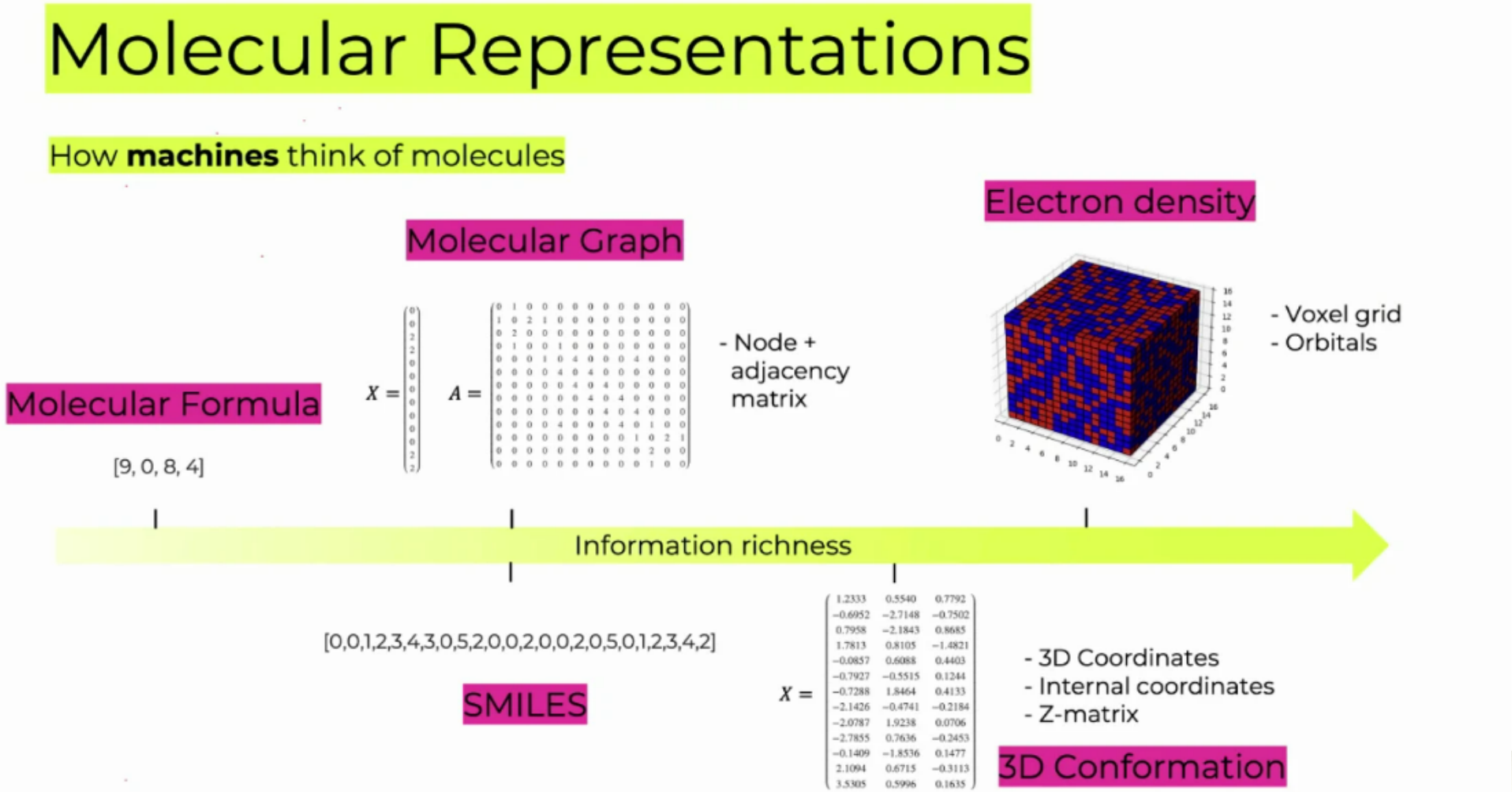
  Openbabel : 다양한 format 간의 변환에 사용된다.

- **분자 Data Encoding**



MLDD 2023 workshop, Djork-Arne Clevert

- 분자 Data Encoding



MLDD 2023 workshop, Djork-Arne Clevert

- **One-Hot Encoding**

- 표현하고 싶은 단어의 인덱스에 1의 값을 부여하고, 다른 인덱스에는 0을 부여하는 단어의 벡터 표현 방식

- SMILES의 문자 Character 수만큼의 벡터에 해당 character 값을 1로 정의

Original Data

Team	Points
A	25
A	12
B	15
B	14
B	19
B	23
C	25
C	29

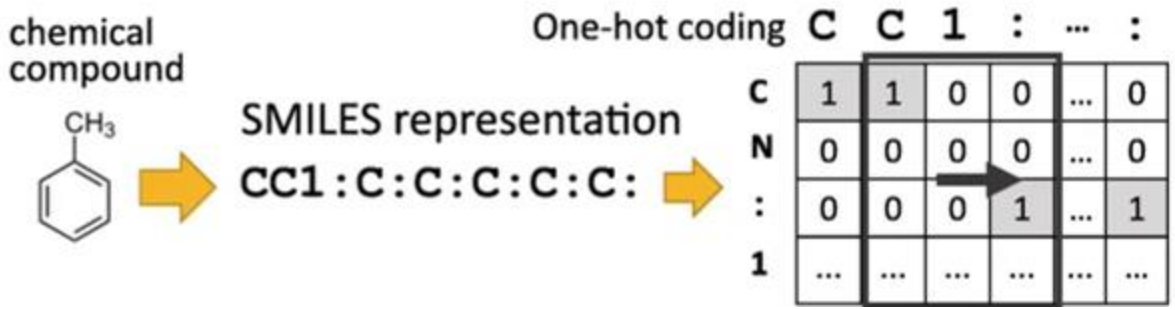
One-Hot Encoded Data

Team_A	Team_B	Team_C	Points
1	0	0	25
1	0	0	12
0	1	0	15
0	1	0	14
0	1	0	19
0	1	0	23
0	0	1	25
0	0	1	29

id	color
1	red
2	blue
3	green
4	blue

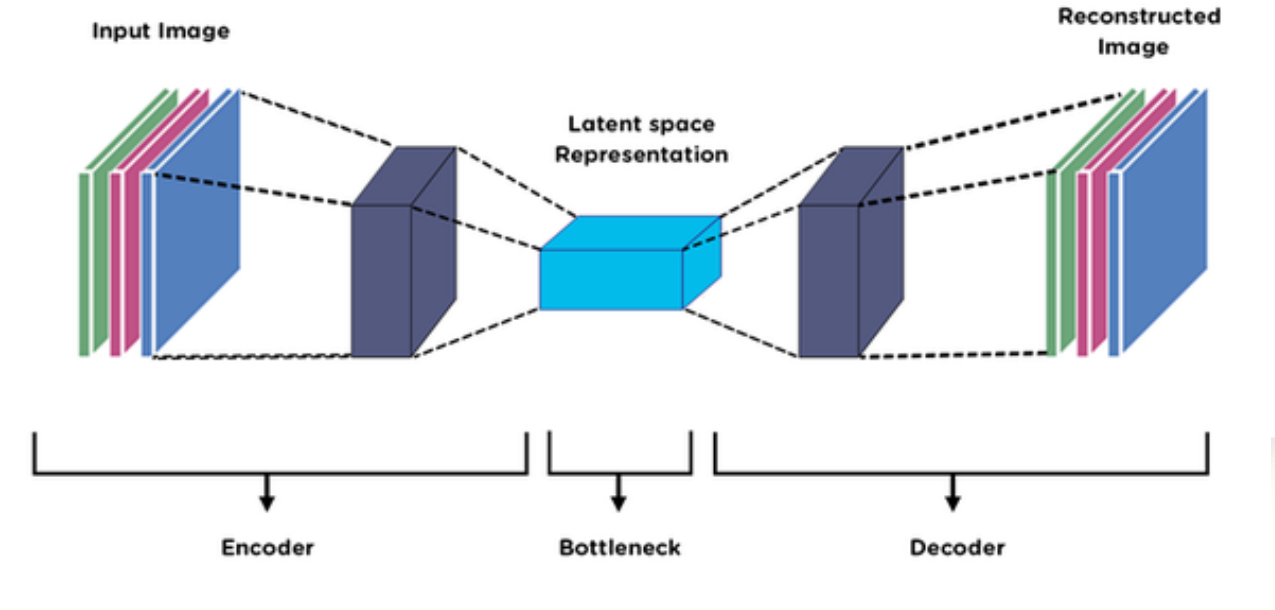
One Hot Encoding

id	color_red	color_blue	color_green
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0



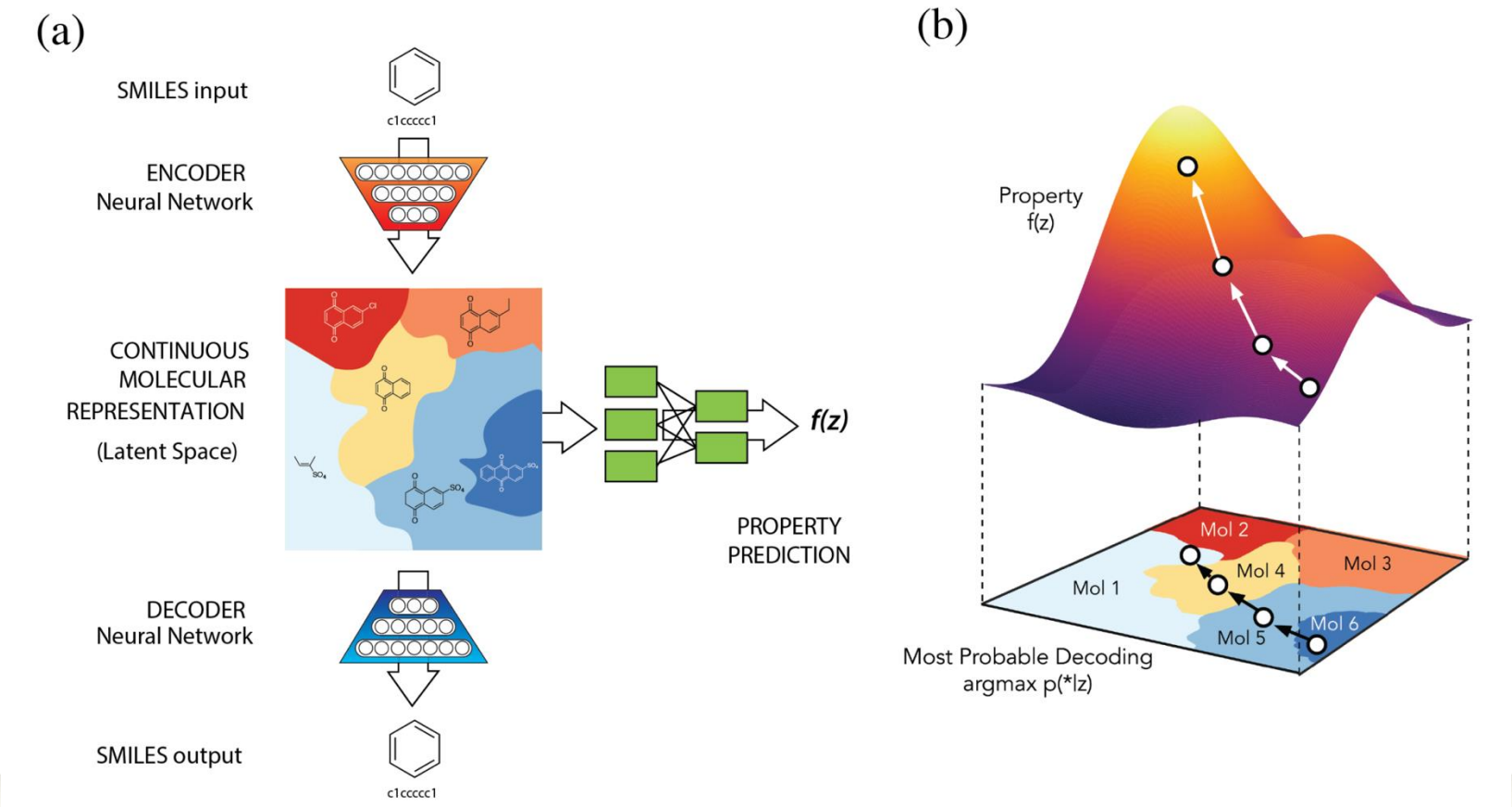
- **Auto Encoder**

- 입력을 저차원 잠재공간으로 인코딩한 후 디코딩하여 복원하는 네트워크. 고차원의 데이터를 압축하여 특징을 파악하는데 주로 사용.
- **구성**
  - **Encoder :** 데이터를 압축하는 부분
  - **Latent space** : 압축 데이터 공간 Z (**latent vector)**
  - **Decoder : 데이터 복원**



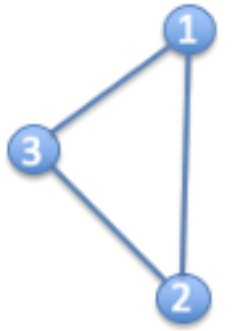
- **Auto-encoder**

SMILES auto-encoder : SMILES의 특징을 latent space에서 추출

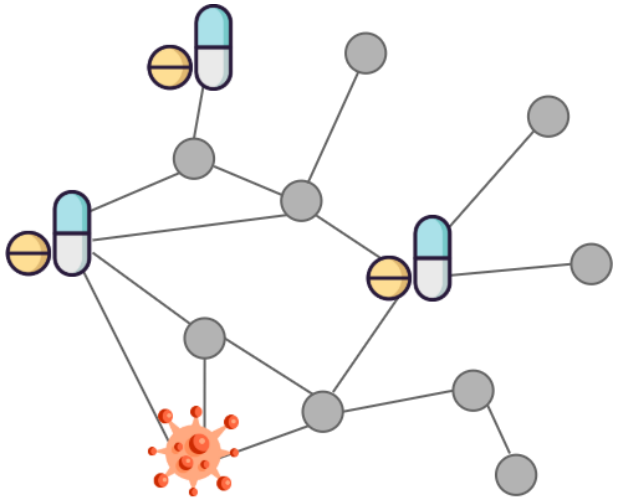


- **Graph**

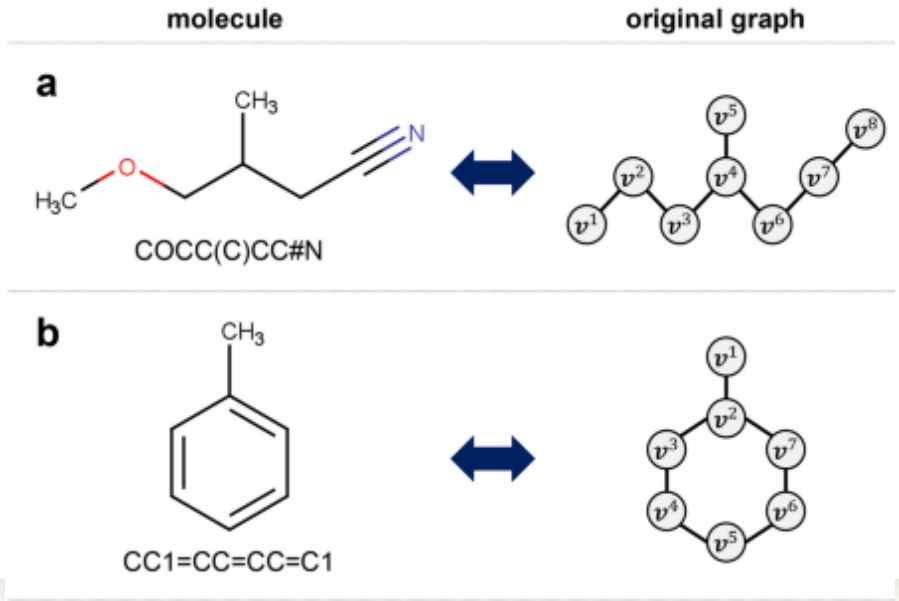
- **분자 그래프**
  - 원자 -> node, 공유결합 -> edge
  - 원자와 공유결합 정보를 node와 edge에 vector로 표현.
  - 모델 학습에 필요한 다양한 정보를 표현 가능.



$G = (V, E)$



Drug-repurposing network

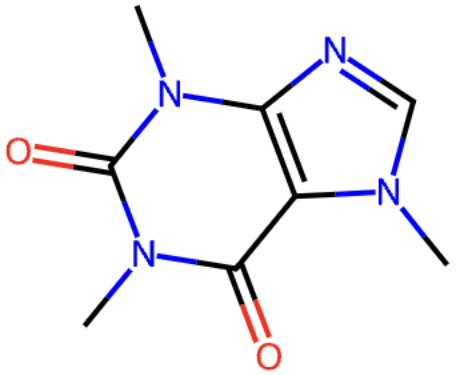


https://doi.org/10.1186/s13321-020-00463-2

- **Graph**

```
import rdkit
from rdkit import Chem
from rdkit.Chem.Draw import IPythonConsole
from rdkit.Chem import Draw

import networkx as nx

import pandas as pd
import numpy as np
```

```
smiles = 'CN1C=NC2=C1C(=O)N(C(=O)N2C)C'
mol = Chem.MolFromSmiles(smiles)
mol
```

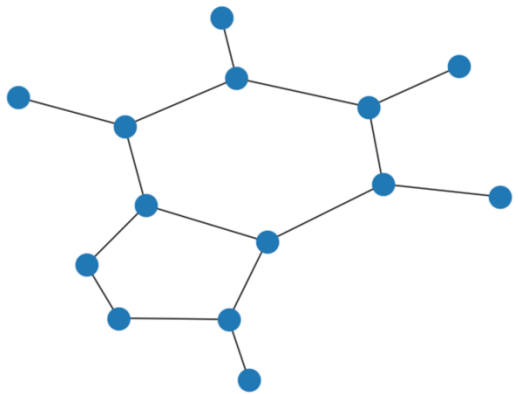


https://colab.research.google.com/github/aifactory-team/AFCompetition/blob/main/2106/molecule_visauli zation.ipynb

- **Graph**

```
def mol_to_graph(mol):
        G = nx.Graph()

    for atom in mol.GetAtoms():
        G.add_node(atom.GetIdx(),
                   atomic_num=atom.GetAtomicNum(),
                   formal_charge=atom.GetFormalCharge(),
                   chiral_tag=atom.GetChiralTag(),
                   hybridization=atom.GetHybridization(),
                   num_explicit_hs=atom.GetNumExplicitHs(),
                   is_aromatic=atom.GetIsAromatic())
    for bond in mol.GetBonds():
        G.add_edge(bond.GetBeginAtomIdx(),
                   bond.GetEndAtomIdx(),
                   bond_type=bond.GetBondType())
    return G
```

```
graph = mol_to_graph(mol)
nx.draw(graph)
```



- **Graph**

```
[13] graph.nodes
```

```
NodeView((0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13))
```

```
[14] graph.edges
```

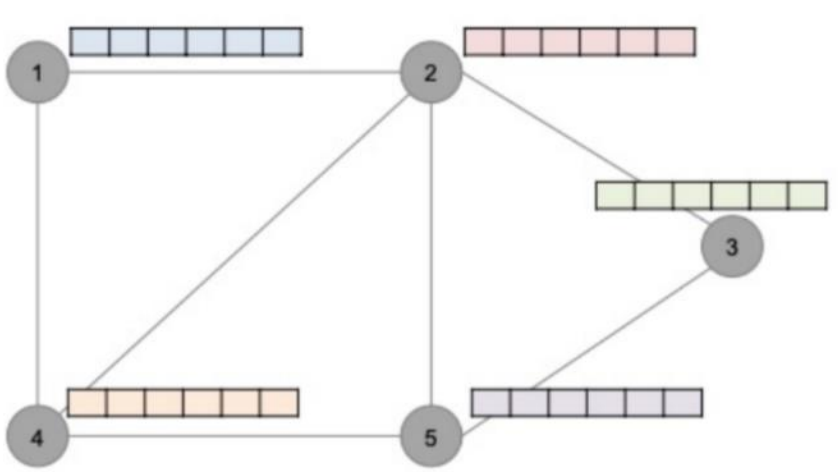
```
EdgeView([(0, 1), (1, 2), (1, 5), (2, 3), (3, 4), (4, 5), (4, 11), (5, 6), (6, 7), (6, 8), (8, 9), (8, 13), (9, 10), (9, 11), (11, 12)])
```

```
[17] nx.adjacency_matrix(graph).todense()
```

```
array([[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
       [1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0],
       [0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1],
       [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
       [0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]])
```

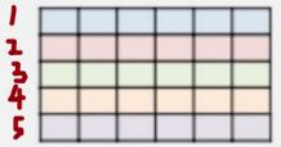
```
[29] graph.nodes[0]
```

```
{'atomic_num': 6,
 'formal_charge': 0,
 'chiral_tag': rdkit.Chem.rdchem.ChiralType.CHI_UNSPECIFIED,
 'hybridization': rdkit.Chem.rdchem.HybridizationType.SP3,
 'num_explicit_hs': 0,
 'is_aromatic': False}
```



$Node - feature\ matrix$

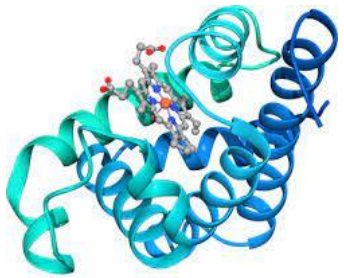
$X \in R^{n \times F}$



- **Protein data**

- **3D structure information**

  Crystal 구조가 밝혀진 targe에 대해 각 amino acid의 3차원 위치정보가 있는 구조 데이터. 모든 target에 대해 존재하지는 않는다. (Binding Pocket 단위@PDB)



Atomic Coordinates: PDB Format

Chain name
Amino Acid / Sequence Number
Element / -----Coordinates-----

						X	Y	Z	(etc.)
ATOM	1	N	ASP	L	1	4.060	7.307	5.186	...
ATOM	2	CA	ASP	L	1	4.042	7.776	6.553	...
ATOM	3	C	ASP	L	1	2.668	8.426	6.644	...
ATOM	4	O	ASP	L	1	1.987	8.438	5.606	...
ATOM	5	CB	ASP	L	1	5.090	8.827	6.797	...
ATOM	6	CG	ASP	L	1	6.338	8.761	5.929	...
ATOM	7	OD1	ASP	L	1	6.576	9.758	5.241	...
ATOM	8	OD2	ASP	L	1	7.065	7.759	5.948	...

Element position within amino acid

- **Amino Sequence**

  구조 정보가 없어도 대부분의 target sequence 정보는 밝혀져 있음

  → Text processing을 통해 vector 생성 후 이용 (factorize, embedding)

```
        10         20         30         40         50         60         70         80         90        100
MRPSGTAGAA LLALLAALCP ASRALEEKKV CQGTSNKLTQ LGTFEDHFLS LQRMFNNCEV VLGNLEITYV QRNYDLSFLK TIQEVAGYVL IALNTVERIP
       110        120        130        140        150        160        170        180        190        200
LENLQIIRGN MYYENSYALA VLSNYDANKT GLKELPMRNL QEILHGAVRF SNNPALCNVE SIQWRDIVSS DFLSNMSMDF QNHLGSCQKC DPSCPNGSCW
       210        220        230        240        250        260        270        280        290        300
GAGEENCQKL TKIICAQQCS GRCRGKSPSD CCHNQCAAGC TGPRESDCLV CRKFRDEATC KDTCPPLMLY NPTTYQMDVN PEGKYSFGAT CVKKCPRNYV
       310        320        330        340        350        360        370        380        390        400
VTDHGSCVRA CGADSYEMEE DGVRKCKKCE GPCRKVCNGI GIGEFKDSLS INATNIKHFK NCTSISGDLH ILPVAFRGDS FTHTPPLDPQ ELDILKTVKE
       410        420        430        440        450        460        470        480        490        500
ITGFLLIQAW PENRTDLHAF ENLEIIRGRT KQHGQFSLAV VSLNITSLGL RSLKEISDGD VIISGNKNLC YANTINWKKL FGTSGQKTKI ISNRGENSCK
       510        520        530        540        550        560        570        580        590        600
ATGQVCHALC SPEGCWGPEP RDCVSCRNVS RGRECVDKCN LLEGEPREFV ENSECIQCHP ECLPQAMNIT CTGRGPDNCI QCAHYIDGPH CVKTCPAGVM
       610        620        630        640        650        660        670        680        690        700
GENNTLVWKY ADAGHVCHLC HPNCTYGCTG PGLEGCPTNG PKIPSIATGM VGALLLLLVV ALGIGLFMRR RHIVRKRTLR RLLQERELVE PLTPSGEAPN
       710        720        730        740        750        760        770        780        790        800
QALLRILKET EFKKIKVLGS GAFGTVYKGL WIPEGEKVKI PVAIKELREA TSPKANKEIL DEAYVMASVD NPHVCRLLGI CLTSTVQLIT QLMPFGCLLD
       810        820        830        840        850        860        870        880        890        900
YVREHKDNIG SQYLLNWCVQ IAKGMNYLED RRLVHRDLAA RNVLVKTPQH VKITDFGLAK LLGAEEKEYH AEGGKVPIKW MALESILHRI YTHQSDVWSY
       910        920        930        940        950        960        970        980        990       1000
GVTVWELMTF GSKPYDGIPA SEISSILEKG ERLPQPPICT IDVYMIMVKC WMIDADSRPK FRELIIEFSK MARDPQRYLV IQGDERMHLP SPTDSNFYRA
```

Algorithm

ML/DL model training / optimization

- **Rule-based Vs Data-driven**

Rule-based approach

Rule 1) 귀가 뾰족한가? → Yes

Rule 2) 수염이 있는가? → Yes

Rule 3) 코가 뭉툭한가? → Yes

Model

Prediction

고양이

Data-driven approach

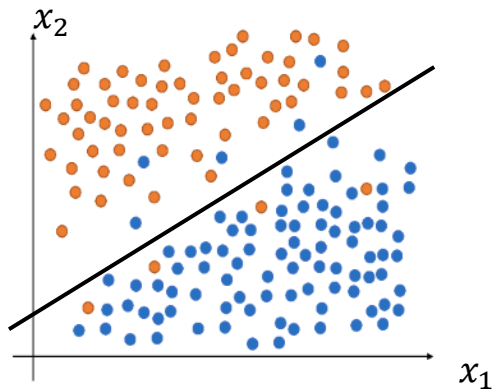
Machine-learning
Deep Learning

X  + 레이블 Y

Data-driven Model

Prediction

고양이

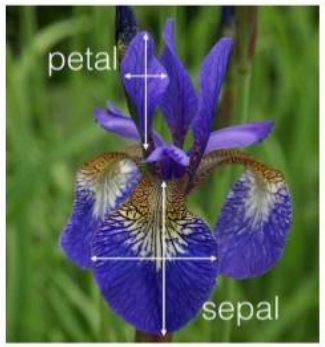


딥러닝 모델에서는 Raw data를 사용한다

https://cs231n.github.io/classification/

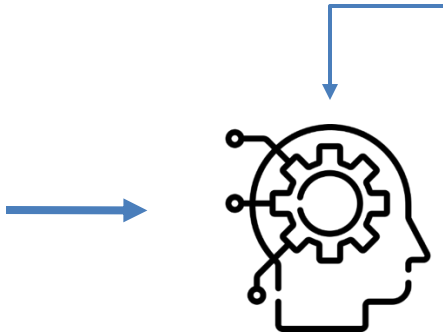
- **ML/DL 모델 학습 과정**

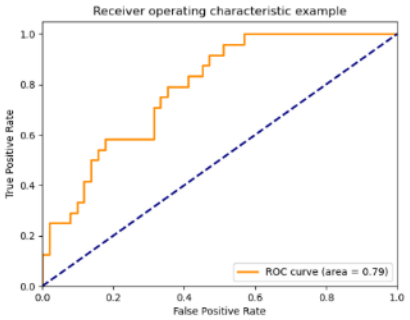
Supervised learning *classification* problem
(using the Iris flower data set)



Training / test data

Features				Labels
Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	Iris setosa
4.9	3.0	1.4	0.2	Iris setosa
7.0	3.2	4.7	1.4	Iris versicolor
6.4	3.2	4.5	1.5	Iris versicolor
6.3	3.3	6.0	2.5	Iris virginica
5.8	3.3	6.0	2.5	Iris virginica





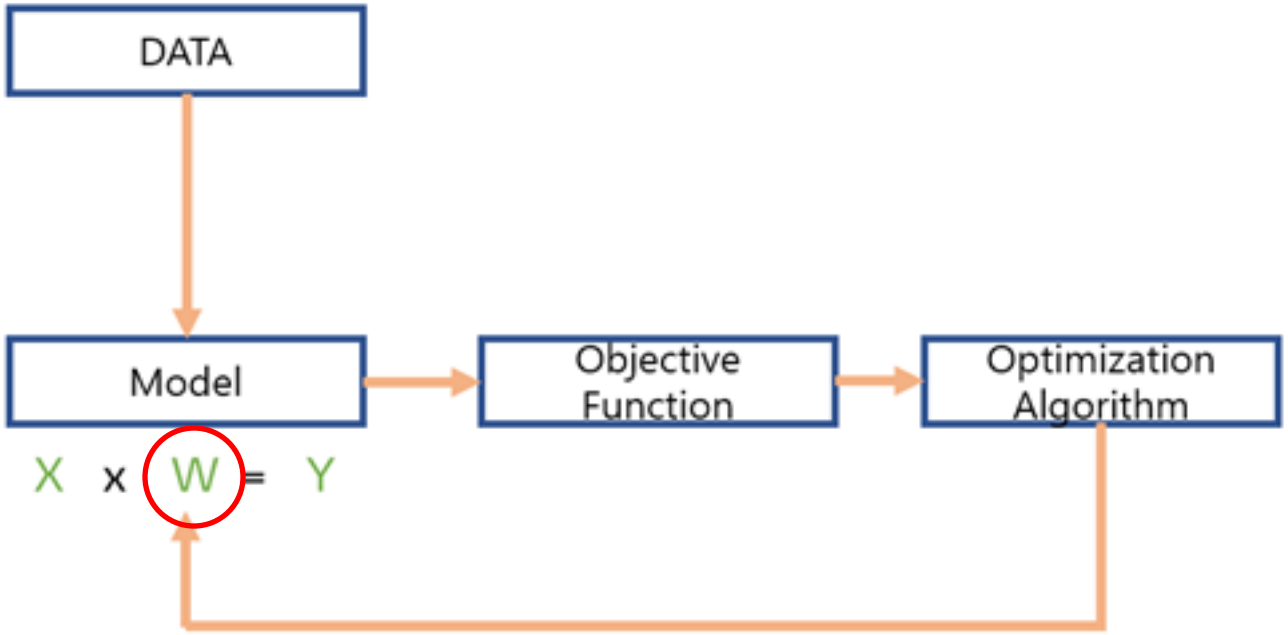
② 데이터 수집 및 전처리

① 모델 설계 및 학습

③ 모델 성능 평가

- **ML/DL 모델**

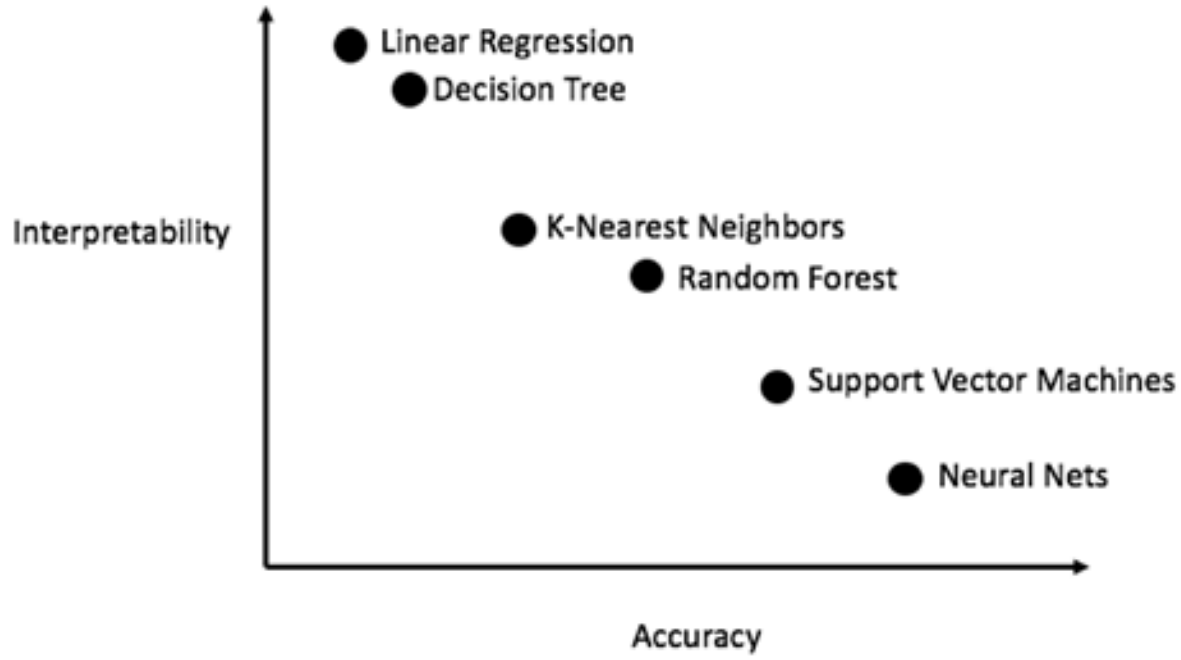
딥러닝/머신러닝 **모델**을 학습한다는 것은 주어진 **데이터**를 이용하여 **목적 함수**(Objective function)를 최소화하는 방향으로 **최적화**(Optimization)를 수행하여 모델의 **가중치 파라미터**(Weight Pa



- **ML techniques**

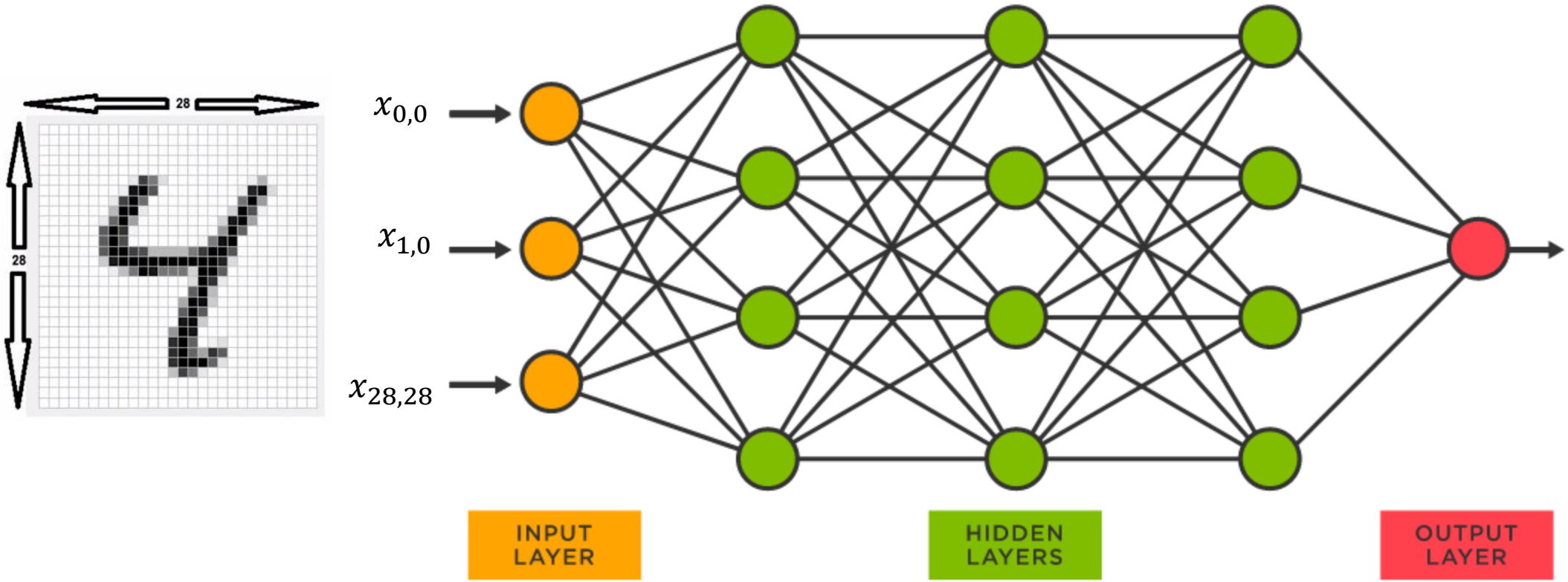
- Naïve Bayes classifier
- Decision tree (RandomForest, XGBoost)
- Random forest
- Support vector machine
- Linear regression
- Hidden Markov model
- Gaussian mixture model

…



- **Deep Neural Network**

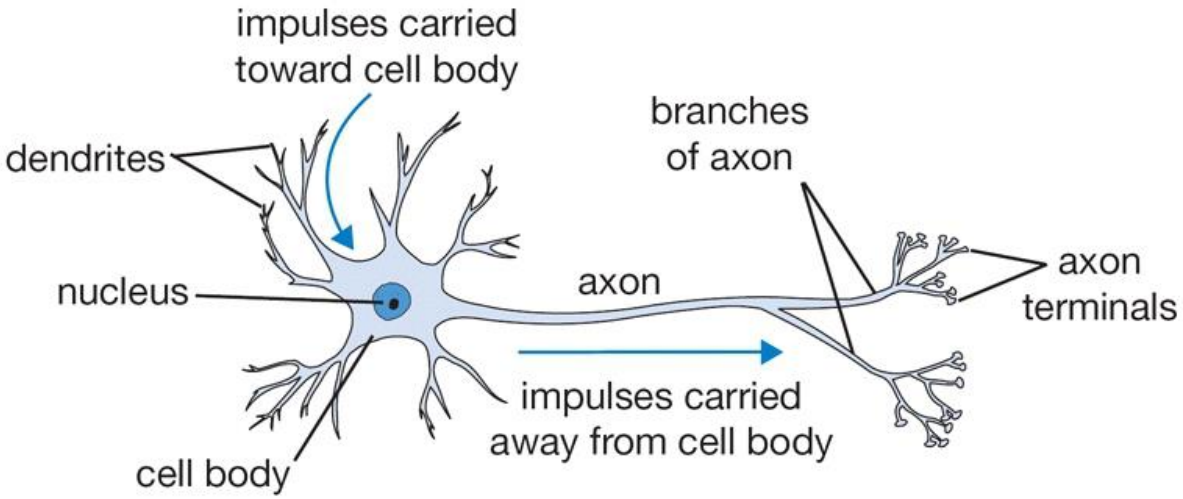
- Fully-connected Layer (FC) : 딥러닝의 가장 기본 레이어



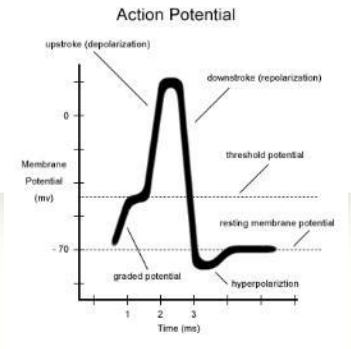
Raw data에서 Output과의 상관관계 혹은 중요도를 모델이 스스로 학습하도록 하자

- **Deep Learning : artificial neural network**

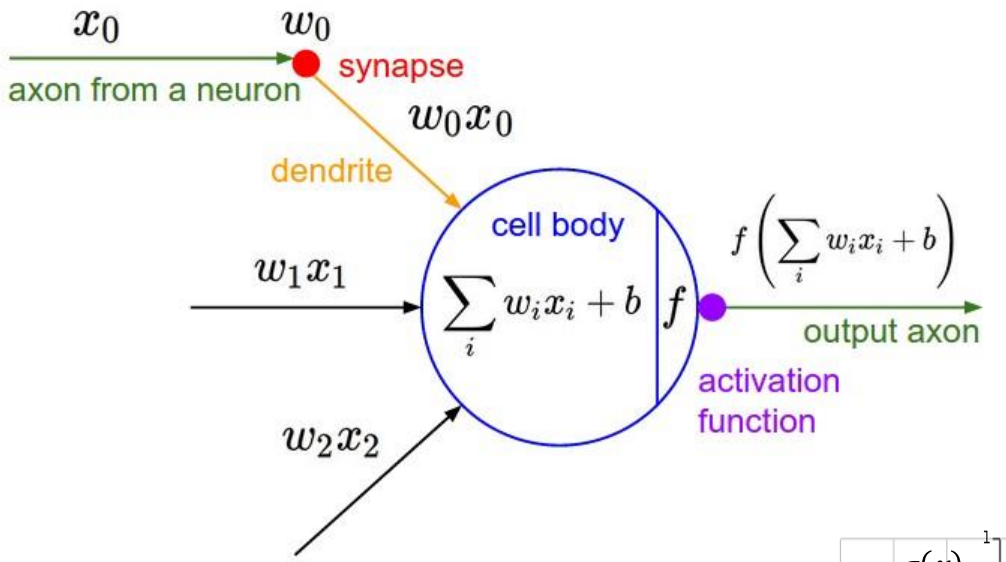
- Perceptron : 딥러닝 구조의 빌딩 블록. 선형(linear)연산과 비선형 연산의 조합으로 구성



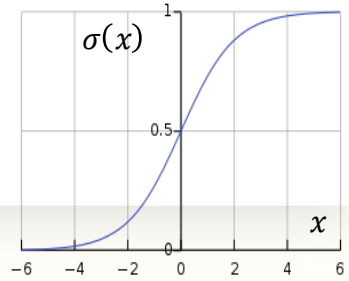
biological neuron



Inputs weights sum Non-linearity Outputs



mathematical model

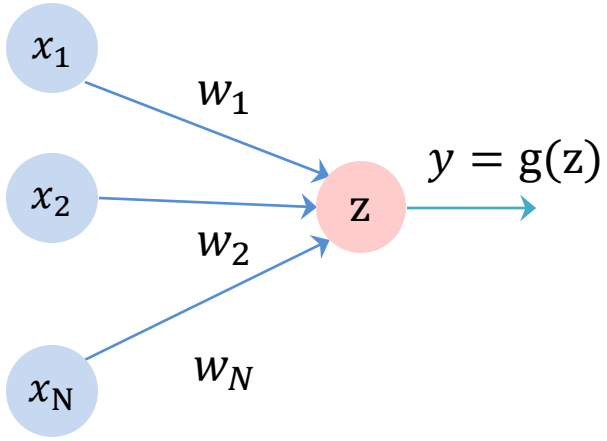


activation function

- **Neural Network**

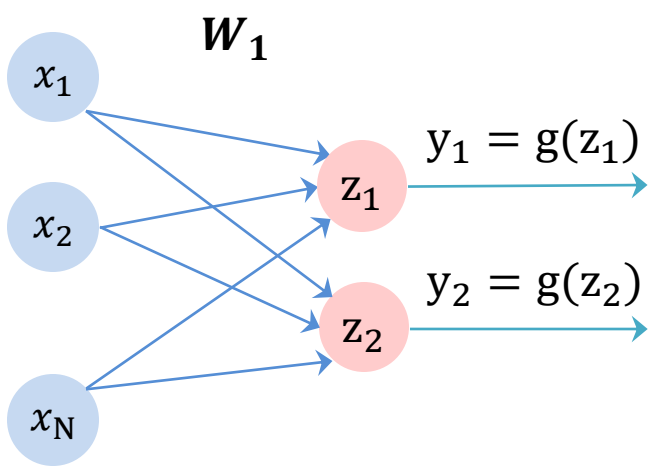
- Perceptron을 네트웍으로 확장

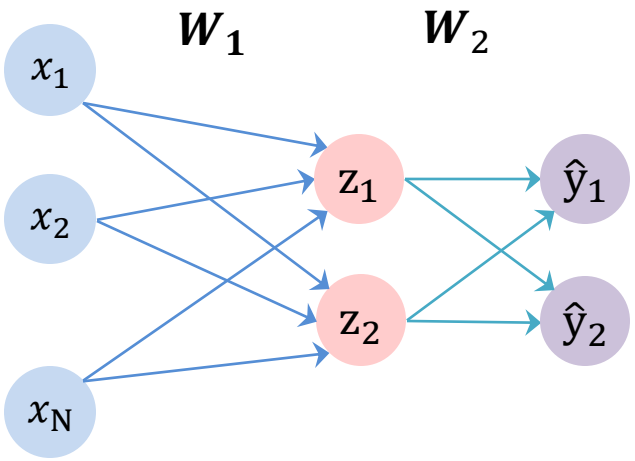
Single Perceptron



$$y = g(w_0 + \boldsymbol{X}^T\boldsymbol{W})$$

Multi-Output Perceptron





$$\boldsymbol{Z} = g(w_{1\_0} + \boldsymbol{X}^T\boldsymbol{W_1})$$

$$\boldsymbol{Y} = g(w_{2\_0} + Z^T\boldsymbol{W_2})$$

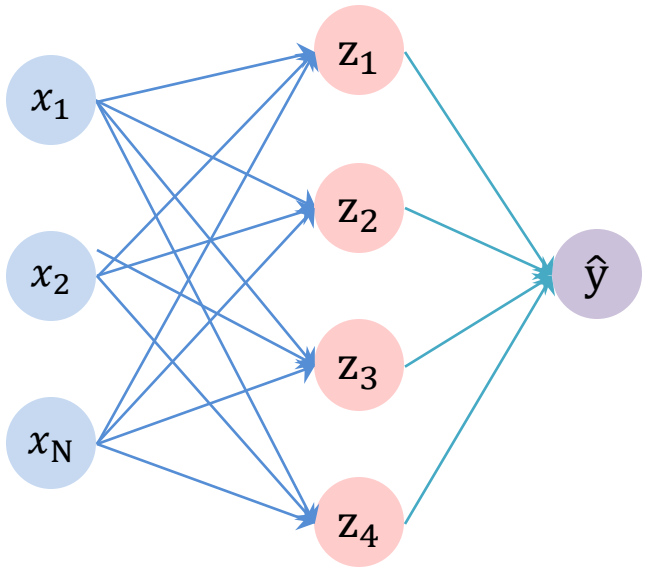
- **Deep learning training**

- Loss minimization = Weight optimization

$$\mathcal{L}\left(f\left(x^{(i)};\boldsymbol{W}\right),y^{(i)}\right)=\mathcal{L}\left(\hat{y}^{(i)},y^{(i)}\right)$$

Predicted Actual

$$X=\begin{bmatrix}0.0 & \dots & 0.7\\ 0.1 & \dots & 0.3\\ 0.0 & \dots & 0.1\end{bmatrix}$$



y $f(x)$

① Binary Classification task

$$\begin{bmatrix}1\end{bmatrix} \qquad \begin{bmatrix}0.1\end{bmatrix}$$

② Classification task

$$\begin{bmatrix}1\\0\\0\end{bmatrix} \qquad \begin{bmatrix}0.1\\0.8\\0.6\end{bmatrix}$$

③ Regression task

$$\begin{bmatrix}0.3\end{bmatrix} \qquad \begin{bmatrix}0.1\end{bmatrix}$$

- Model Training

→ **find best weights parameter sets to minimize loss for given dataset**

- **최적화**

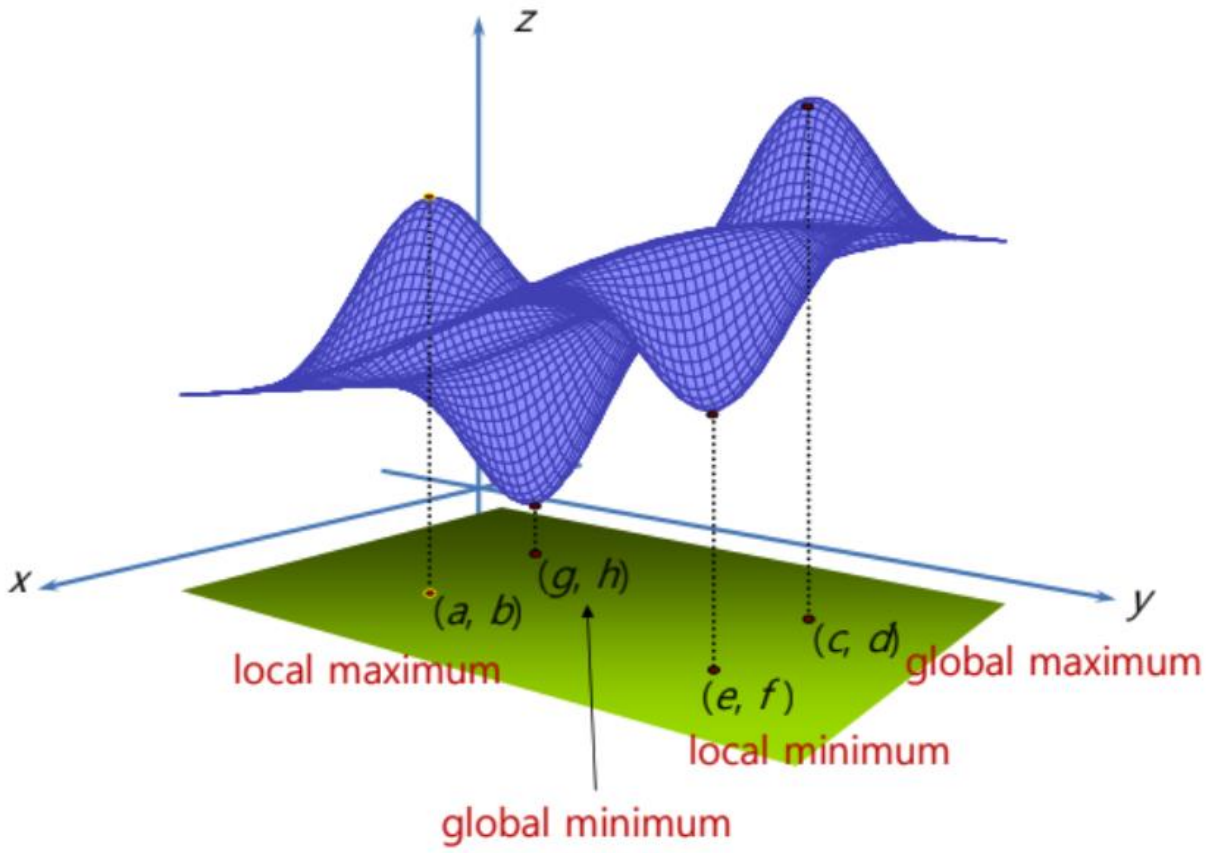
- ML/DL model 학습 : 손실 함수(loss function)의 global minimum을 탐색하여, 손실이 최소가 되는 지점의 모델 파라미터를 찾는 과정

손실함수

실제값 과 예측값의 차이

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y - \hat{f}(x_i))^2$$

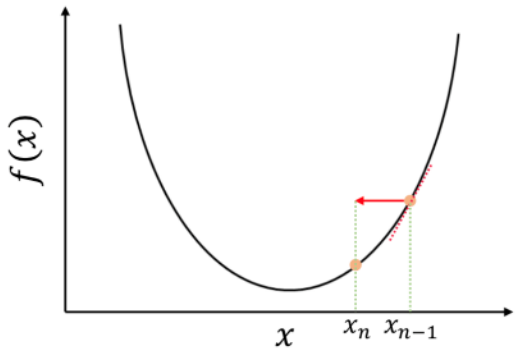
$$\ell(p) = -\sum_{i=1}^{n}[y_i log(p_i) + (1 - y_i) log(1 - p_i)]$$



- Gradient Descent

Learning Rate : 모델의 매개변수를 업데이트할 때 한 번의 경사하강 단계에서 얼마만큼 이동할지를 결정 [속도]
Batch 설정 : 사용하는 데이터셋을 변경하여 이동하는 방향을 변경 [방향]

경사 하강법



1-D의 경우

$$x_n = x_{n-1} - \alpha \frac{df(x_{n-1})}{dx}$$

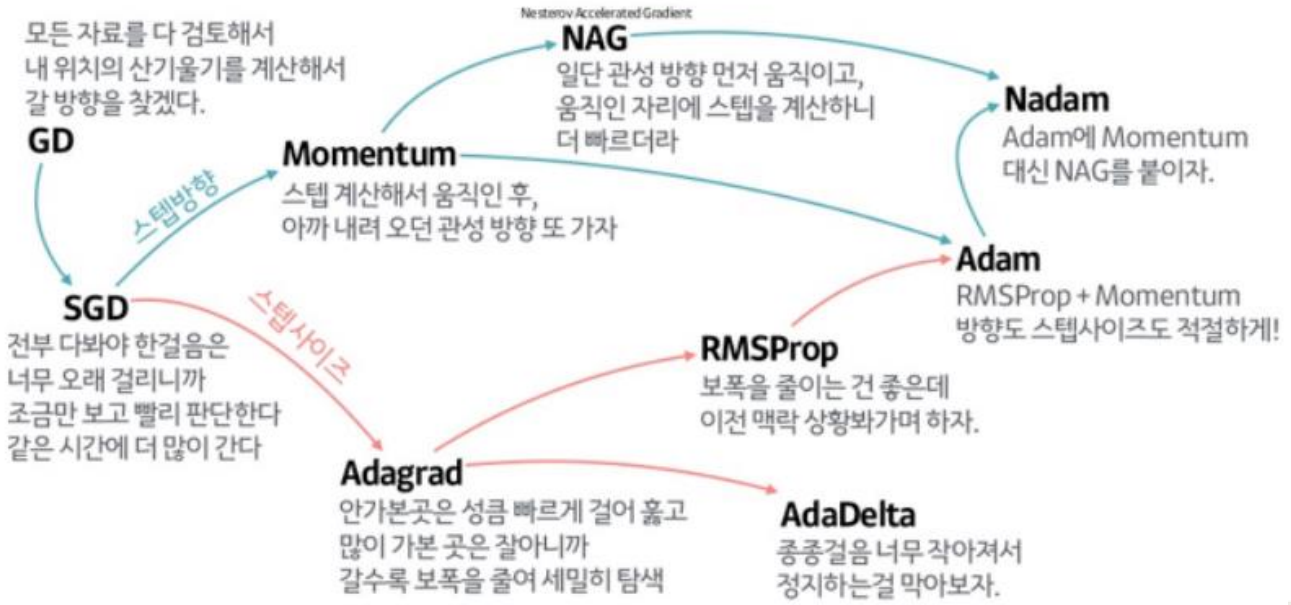
N-D 의 경우

$$\boldsymbol{x}_n = \boldsymbol{x}_{n-1} - \alpha \nabla f(\boldsymbol{x}_{n-1})$$

α: 학습률 (Learning rate)

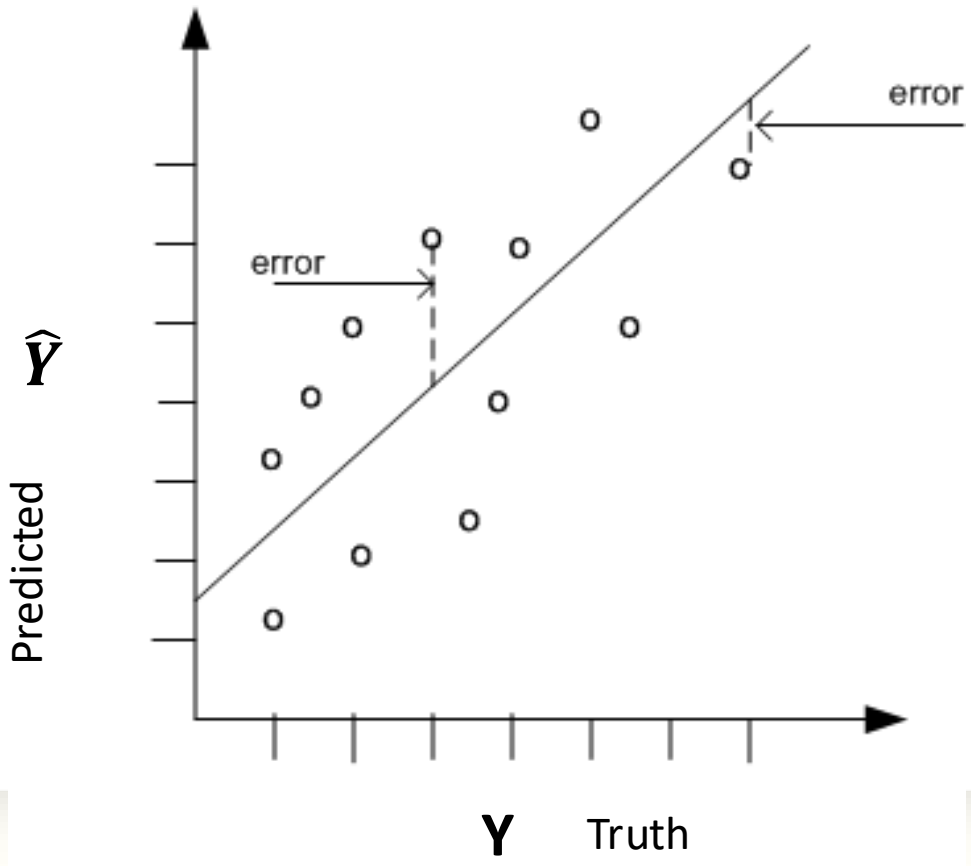
경사 하강법 (Gradient descent)의 한 스텝

경사 하강법은 $f(x)$의 값이 변하지 않을 때 까지 스텝을 반복한다.



Performance Metrics

- **Regression Performance 평가**



작을수록 좋음 (망소)

- Mean Absolute Error (**MAE**)

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}|$$

- Mean Squared Error (**MSE**)

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2$$

- Root Mean Squared Error (**RMSE**)

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2}$$

클수록 좋음 (<= 1)

- **R2**

$$R^2 = \frac{\text{RegSS}}{\text{TSS}} = \frac{\sum_i (\hat{y}_i - \bar{y})^2}{\sum_i (y_i - \bar{y})^2}$$

- **Classification Performance 평가**

		Predicted	
		Negative **(N)** -	Positive **(P)** +
Actual	Negative -	True Negative **(TN)**	**False Positive (FP) Type I Error**
	Positive +	**False Negative (FN) Type II Error**	True Positive **(TP)**

Confusion matrix

a. Accuracy

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{\text{Correct Predictions}}{\text{Total Predictions}}$$

b. Precision

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{\text{Predictions Actually Positive}}{\text{Total Predicted positive}}$$

c. Recall (TPR, Sensitivity)

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{\text{Predictions Actually Positive}}{\text{Total Actual positive}}$$

d. F1-Score

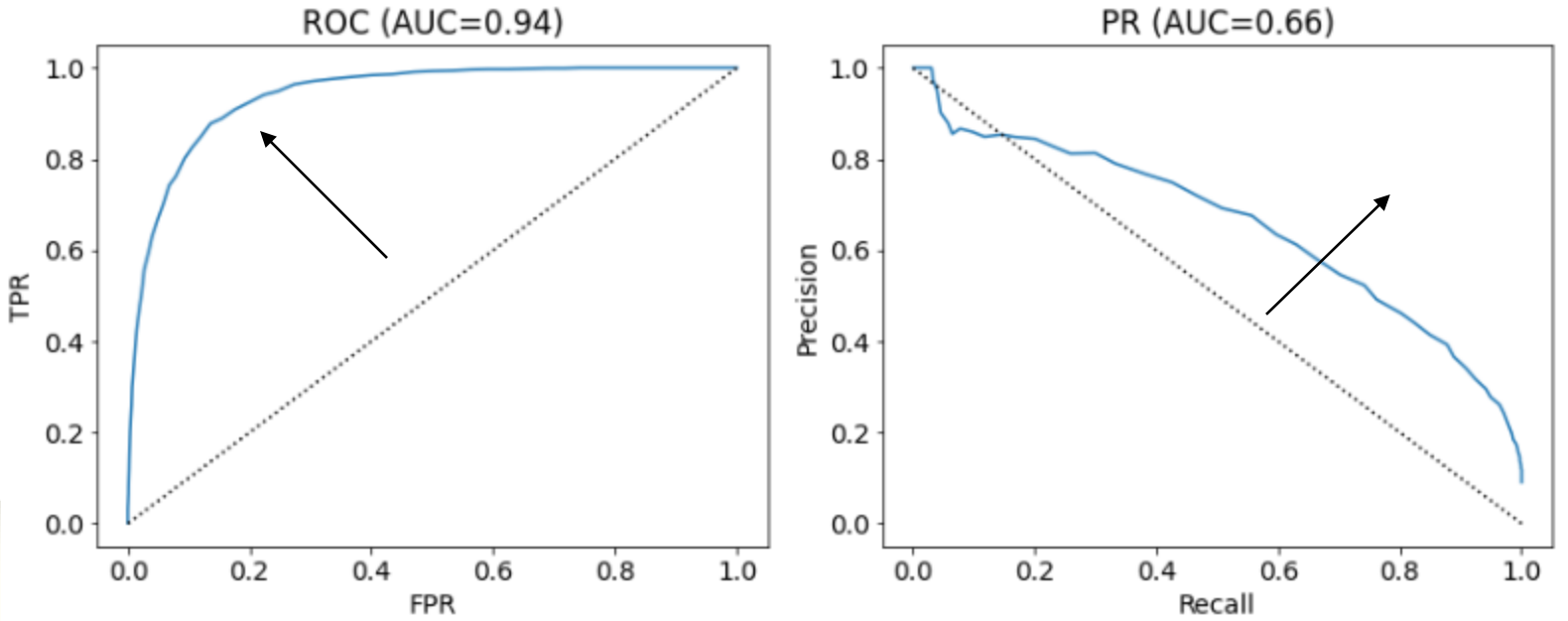
$$\text{F1-Score} = 2 * \frac{(\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})}$$

e. FPR (Type I Error) FPR = FP / (FP+TN)

f. FNR (Type II Error)

- **Performance 평가**

	ROC-AUC	PR-AUC
x 축	**FPR**	**Recall (TPR)**
y 축	**TPR**	**Precision**
좋은 모델	FPR 낮을수록, TPR 높을수록 (곡선이 **왼쪽, 위**로 향할수록)	Recall 높을수록, Precision 높을수록 (곡선이 **오른쪽, 위**로 향할수록)



→ **AUC는 클수록 좋다**

AUC : Area Under Curve

- **Concordance index**

- CI index 정의 : 생존 분석에서 두 개체 간의 생존 시간 순서를 비교하여, **모델이 올바르게 순서를 예측했는지를 측정하는 지표**
- C-index : 0과 1 사이의 값을 가지며, 1은 완벽한 예측, 0.5는 무작위 예측, 0에 가까울수록 예측력이 떨어짐을 의미함.

→ DTI regression task에서 binding affinity이 좋은 순서를 잘 매기는지 평가하는 지표

$$c = \frac{\sum_{i \in U} \left\{ \sum_{T_j > T_i} 1_{f_j > f_i} \right\}}{\sum_{i \in U} \left\{ \sum_{T_j > T_i} 1 \right\}}$$

where U: a set of uncensored data
T_i: an observed survival time of sample i
f_i: a predicted survival time of sample i
$1_{a>b}$: 1 if $a > b$, and 0 otherwise.

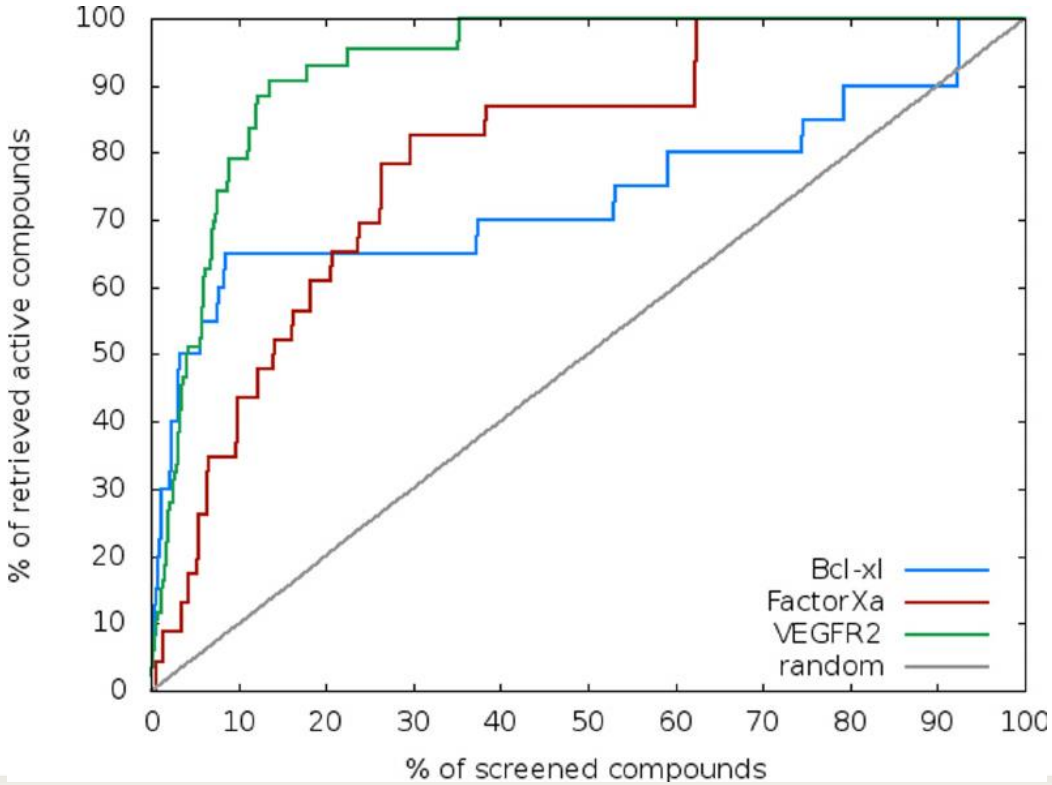
- **Enrichment Factor**

  - Enrichment factor : 전체 분자 중에서 active의 비율에 대한 선별된 분자들 중에서 active의 비율

$$Enrichment\ factor = \frac{\frac{real\_active}{predicted\_active}}{\frac{real\_active}{total\ number}}$$

Enrichment factor	Sediment quality
EF < 2	Deficiency to minimal enrichment
2 < EF < 5	Moderate enrichment
5 < EF < 20	Significant enrichment
20 < EF < 40	Very high enrichment
EF > 40	Extremely high enrichment

Barbieri (2016), Abdulqaderismaeel and Kusag (2015), Mei et al. (2011), Salah et al. (2012)



- Performance 비교

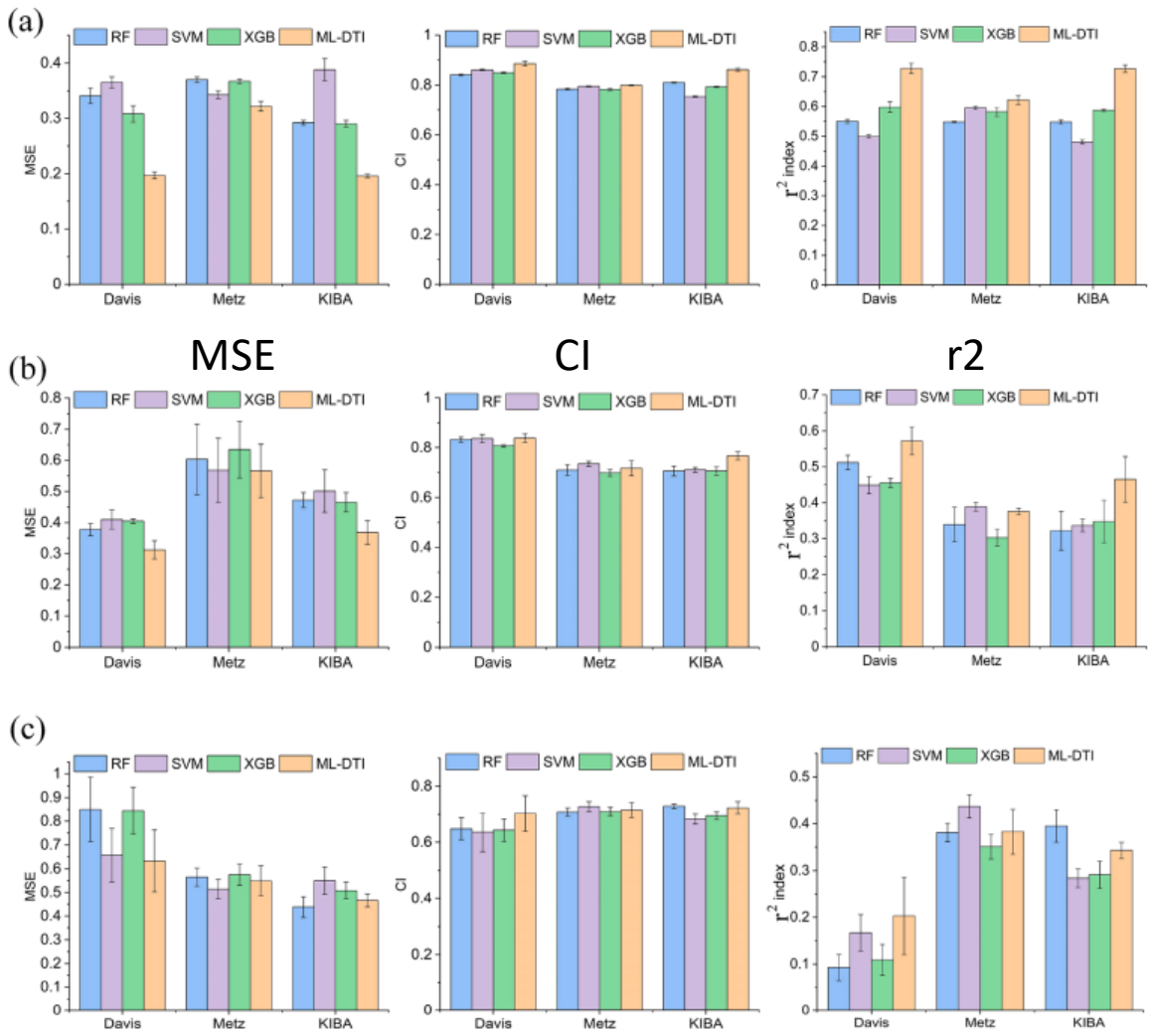
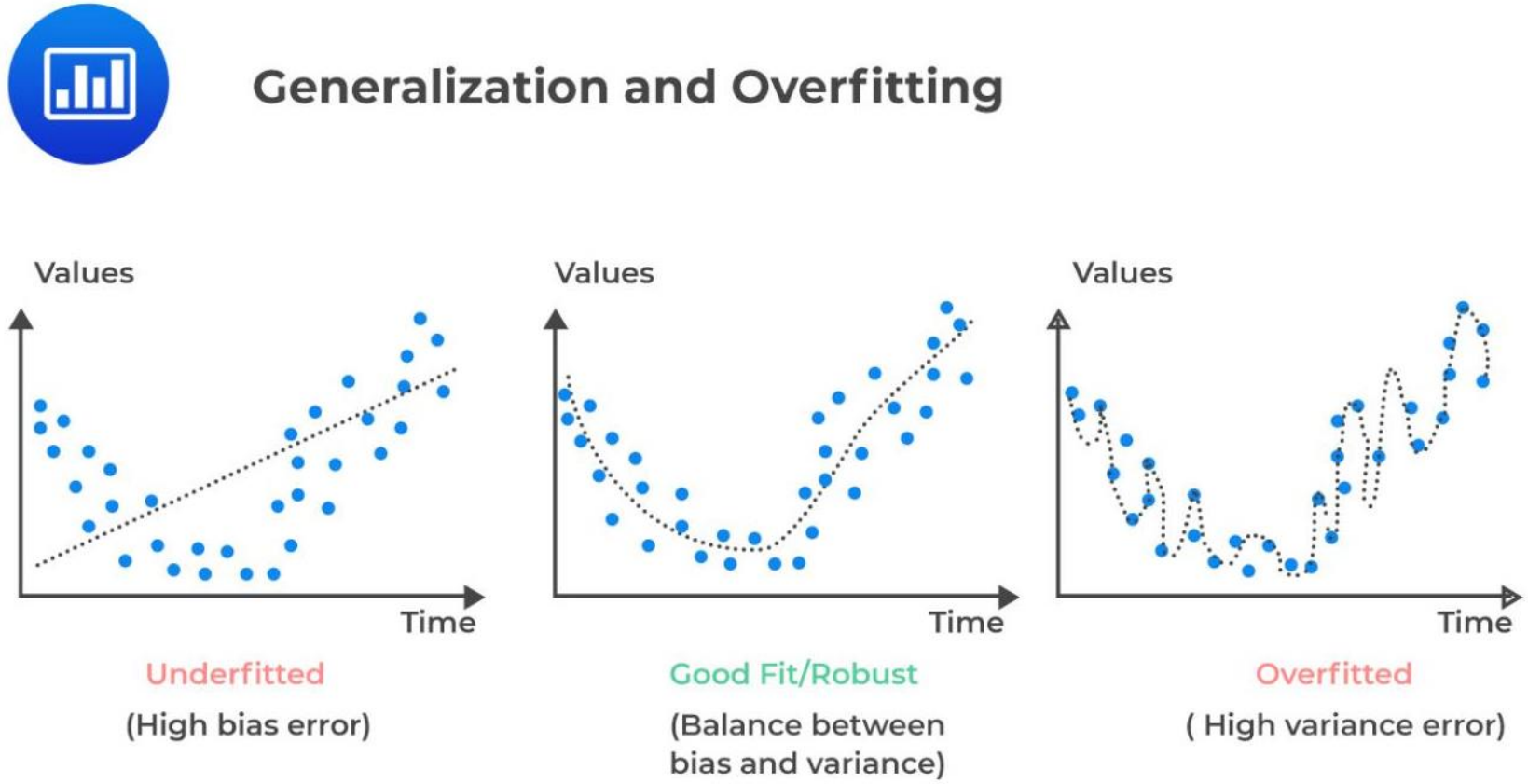


Figure 9. Comparison of ML-DTI to RF, SVM, and XGB in three data sets in terms of MSE, CI, and r^2 index with the (a) random split, (b) orphan-target, and (c) orphan-drug settings.

- **모델 선택**

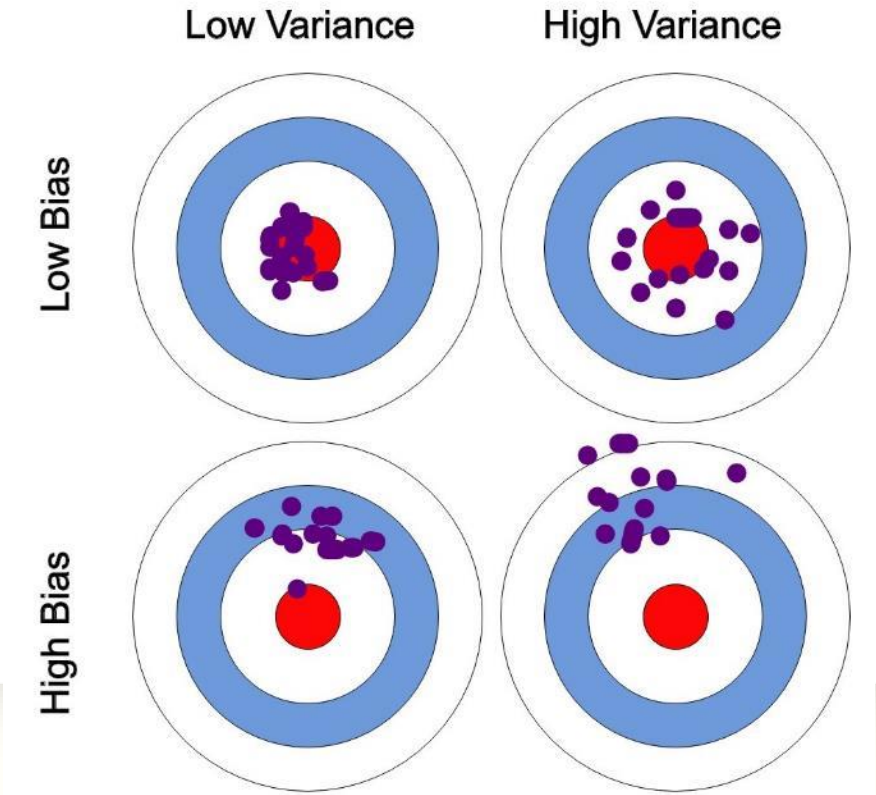
- 오차가 작은 모델이 best일까?

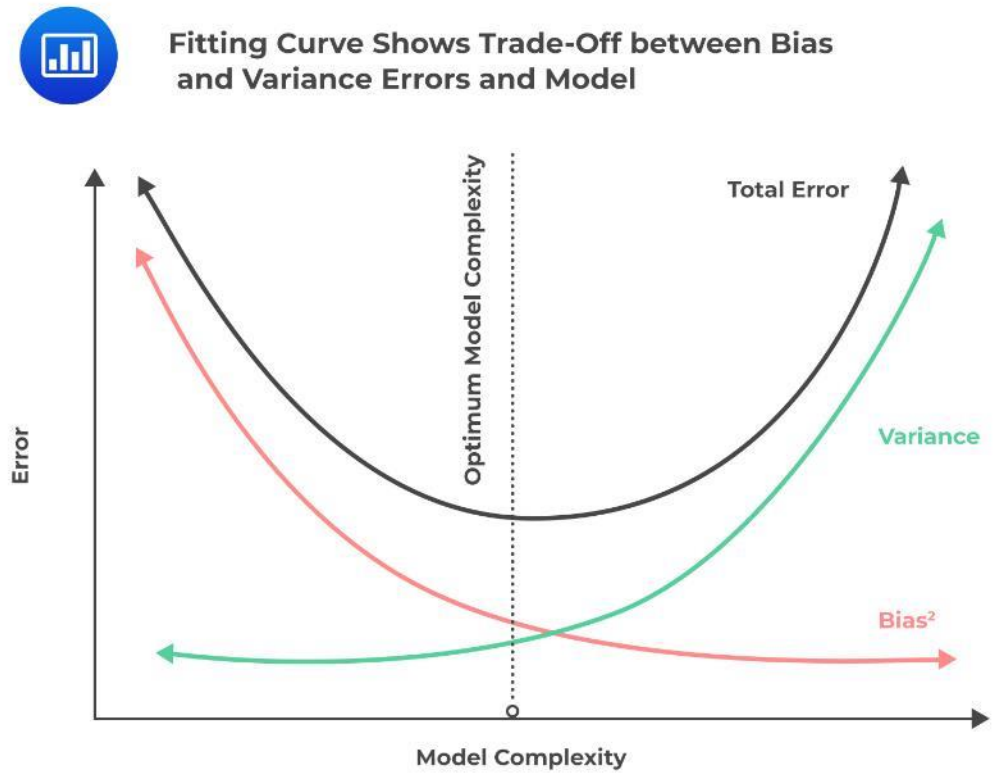


Training error를 무조건 작게 만드는 모델은 overfitting일 가능성이 높다

- **Model bias and variance**

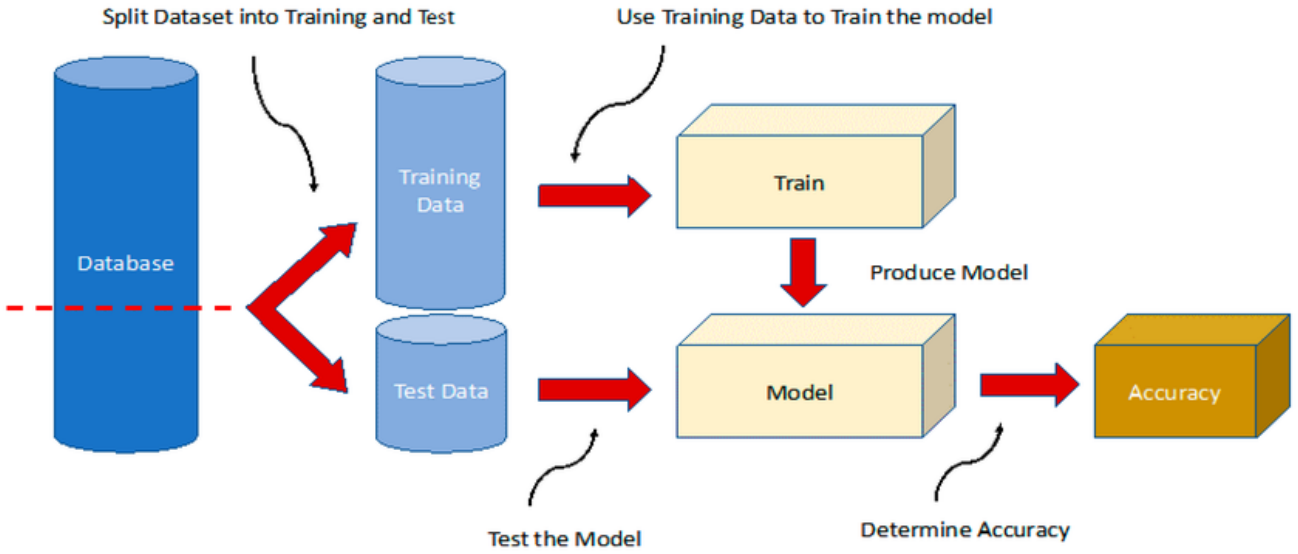
- Bias : 모델이 예측한 값과 실제 정답 간에 일정하게 차이가 나는 정도
- Variance : 다양하게 주어지는 데이터에 대한 모델 예측의 가변성 “flexibility” 의 지표





- **모델 검증 (Validation)**

Generalization performance : 학습하지 않은 데이터에서의 성능 확인 필요
→ Test 오차가 작은 모델이 좋은 모델이다



- **Cross Validation**

- 일반적 모델 학습

  Train vs **Test** set으로 dataset 분류

  : 고정된 Test set을 사용할 경우 모델 성능이 test set에서 overfitting 될 수 있다.

- K-Fold Cross Validation (CV) : 별도의 validation set을 설정

  Train/validation/test dataset 분류

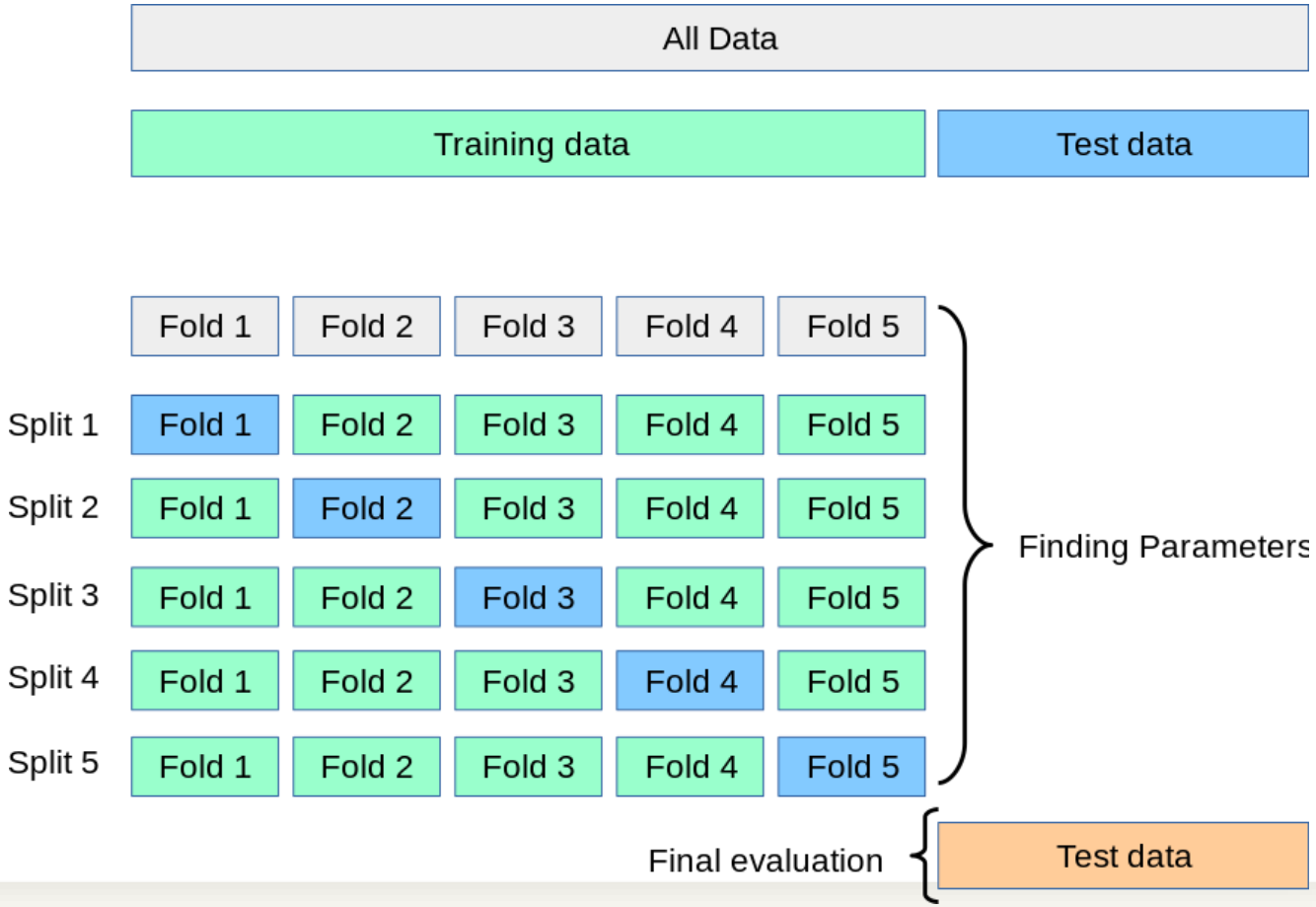
  : Train set으로 model training

  Validation set 으로 최적 model 선택

  Final Test set으로 성능 검증

→ 특정 dataset에 편중되는 것을 방지

모든 dataset을 훈련에 활용할 수 있음



DTI / DTA models

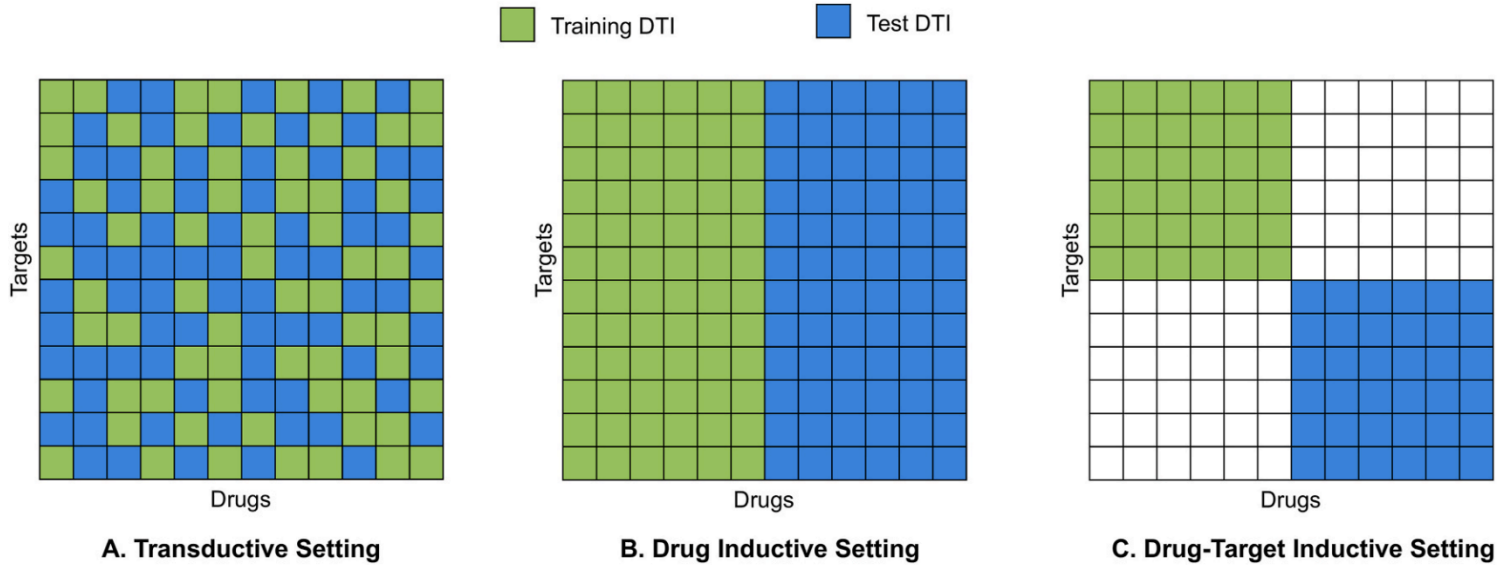
- DL Models

Year	Name	Ligand Encoding [1]	Protein Encoding [1]	ML/DL Model	Framework
2018	DeepDTA [60]	SMILES	full seq.	CNN	reg.
2019	WideDTA [152]	SMILES & MCS	full seq. & domains/motifs	CNN	reg.
2019	DeepAffinity [74]	SMILES	struct. property seq.	RNN+CNN	reg.
2016	DL-CPI [153]	substructure FP	domains	MLP	class.
2018	Kundu et al. [65]	div. feat. FP	div. feat. FP	RF & SVM & MLP	reg.
2018	Sorgenfrei et al. [70]	Morgan FP	z-scales	RF	class.
2019	DeepConv-DTI [154]	Morgan FP	full seq.	CNN	class.
2019	Torng and Altman [57]	graph	graph	GCNN	class.
2020	DGraphDTA [155]	graph	graph	GCNN	reg.
2018	PADME [156]	graph (or Morgan FP)	seq. comp.	GCNN (or MLP)	reg.

[1] Abbreviations: FP: fingerprint, MCS: maximum common substructure, div. feat.: diverse feature count, physicochemical and structural properties, seq.: sequence, comp.: composition.

- **DTI Prediction data split**

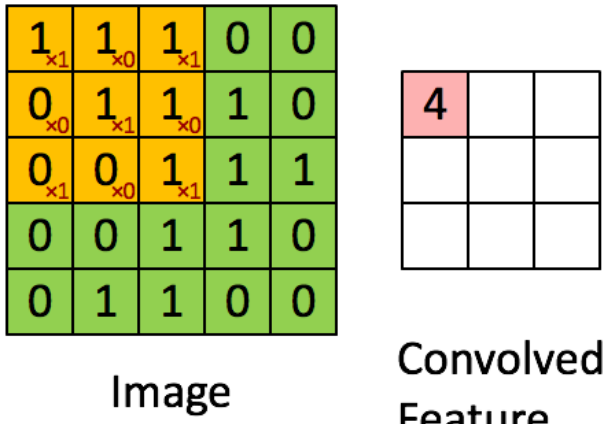
- Unseen drug, Unseen target 에 대한 평가를 위해 data split 이 중요

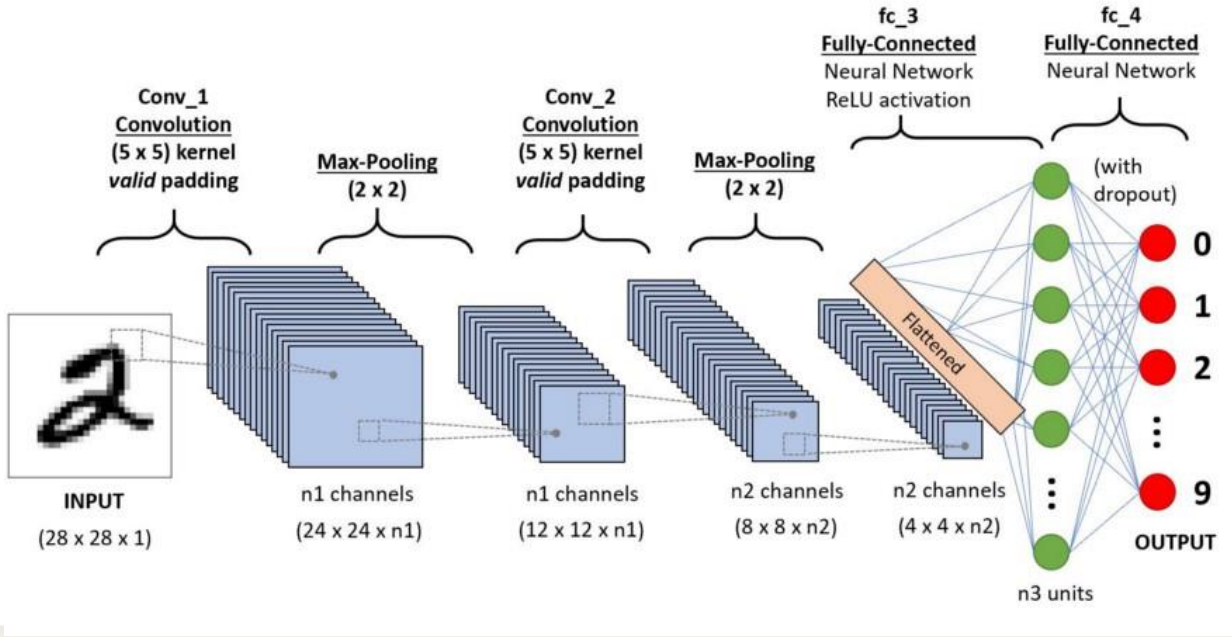


- Target 구조를 이용한 cluster 를 통해 unseen target을 분류하기도 함

- **Deep Learning : Convolution Neural Network**

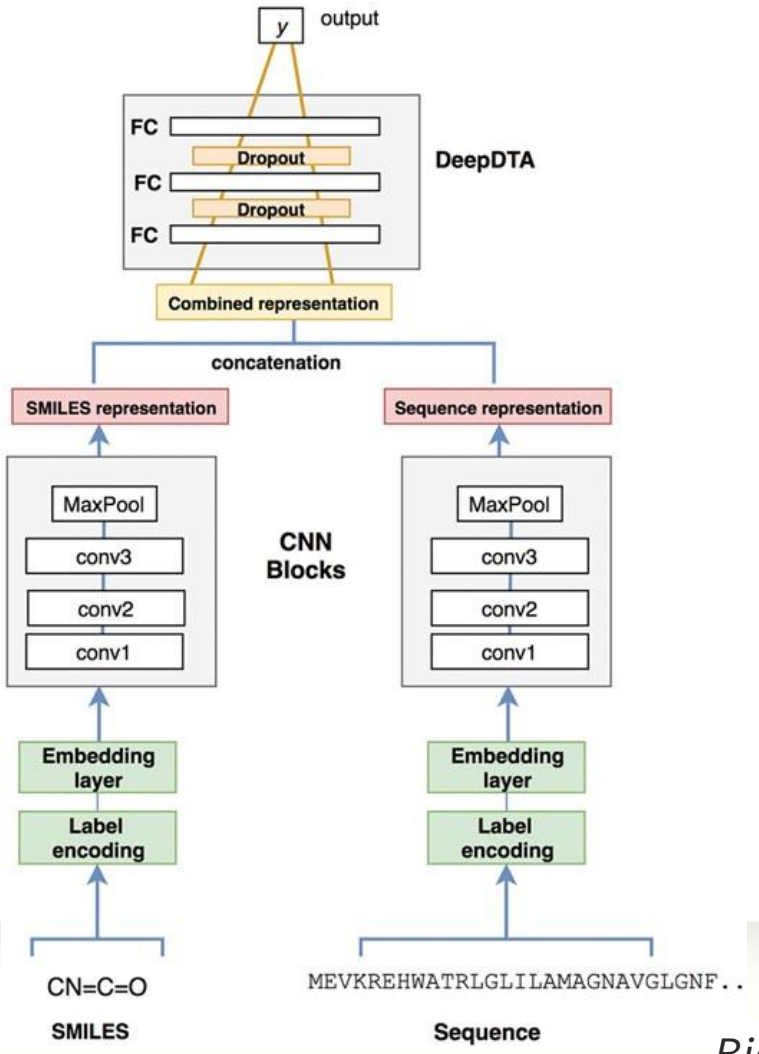
- Convolution : 2D or 3D image에서 주변 데이터와의 연계성을 고려한 feature를 추출하기 위해 matrix 연산곱(kernel filter)을 적용
- 2D image에서 feature 를 추출하는 용도로 주로 사용





Application : Vision 분야. Image classification, object detection, image translation

- DeepDTA



Davis set

$$r_m^2 = r^{2*}(1 - \sqrt{r^2 - r_0^2})$$

	Proteins	Compounds	r_m^2 (std)	AUPR (std)
KronRLS (Pahikkala *et al.*, 2014)	S–W	Pubchem Sim	0.407 (0.005)	0.661 (0.010)
SimBoost (He *et al.*, 2017)	S–W	Pubchem Sim	0.644 (0.006)	0.709 (0.008)
DeepDTA	CNN	CNN	0.630 (0.017)	0.714 (0.010)

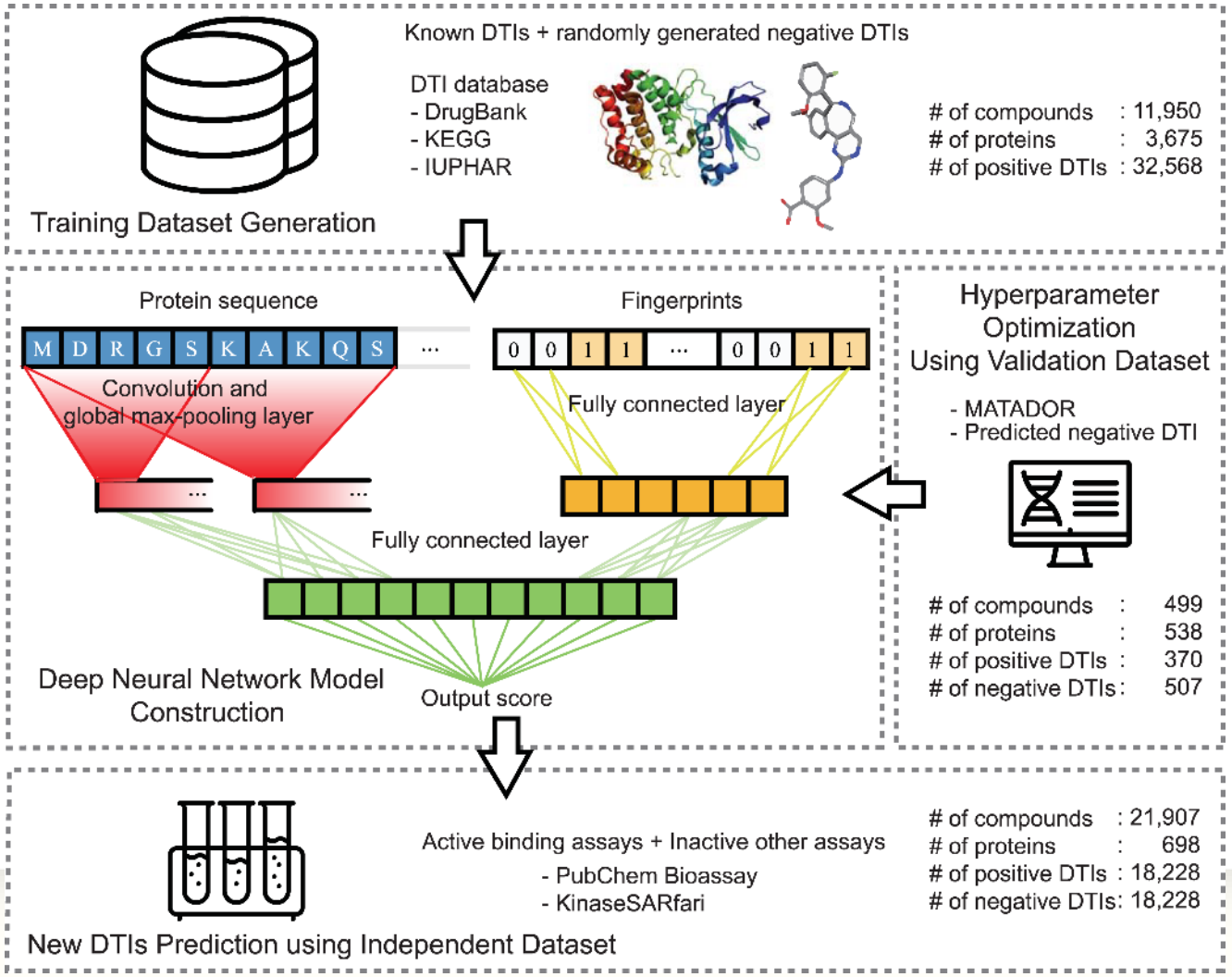
KIBA set

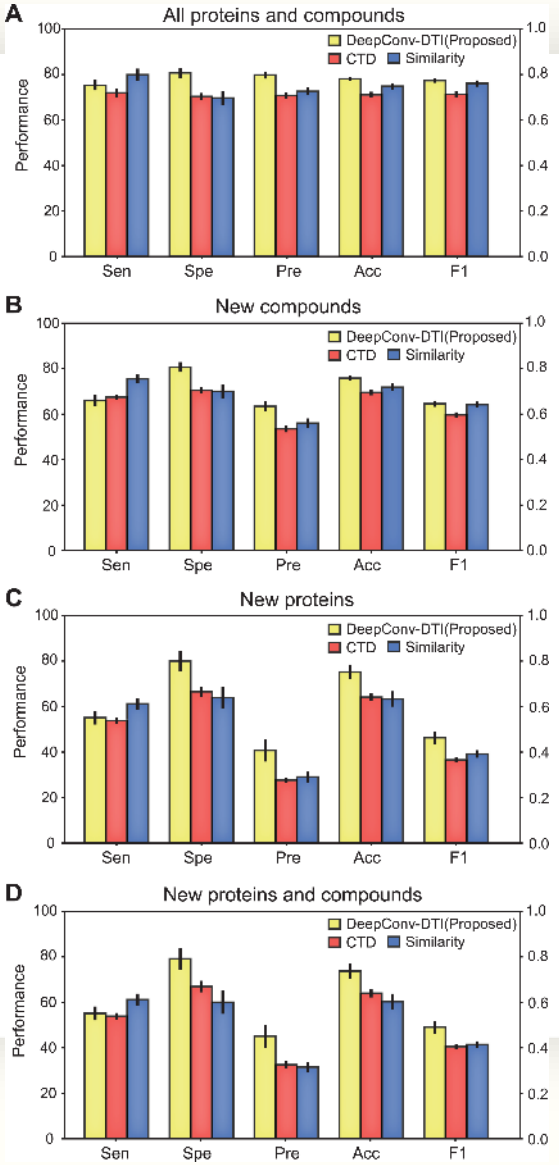
	Proteins	Compounds	r_m^2 (std)	AUPR (std)
KronRLS (Pahikkala *et al.*, 2014)	S–W	Pubchem Sim	0.342 (0.001)	0.635 (0.004)
SimBoost (He *et al.*, 2017)	S–W	Pubchem Sim	0.629 (0.007)	0.760 (0.003)
DeepDTA	CNN	CNN	0.673 (0.009)	0.788 (0.004)

S-W : Smith–Waterman similarity

Bioinformatics, Volume 34, Issue 17, Pages i821–i829 (2018)

- DeepConv-DTI : 1D CNN 기반 DTI



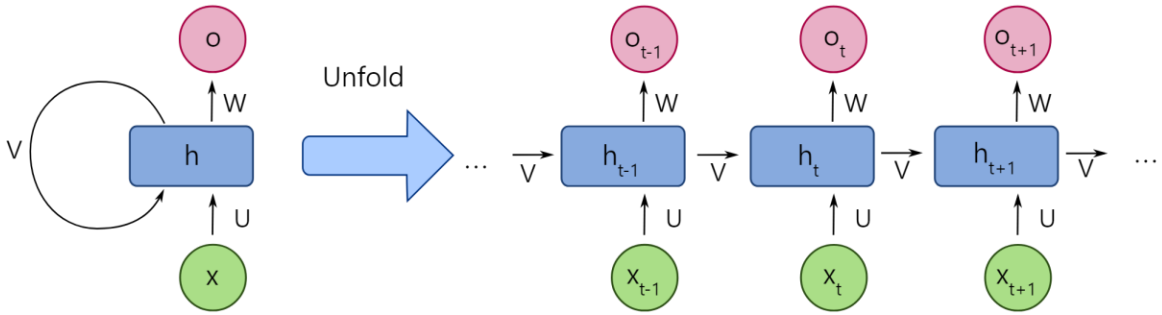


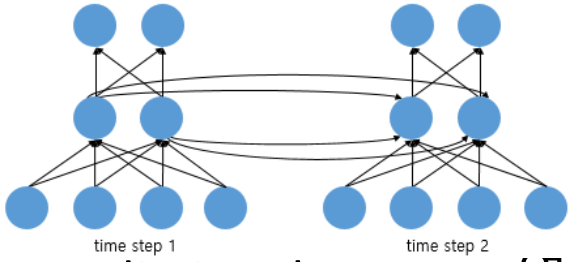
Seensitivity (Sen.), specificity (Spe.), precision (Pre.), accuracy (Acc.), F1 score (F1)

Lee I, Keum J, Nam H (2019) DeepConv-DTI: Prediction of drug-target interactions via deep learning with convolution on protein sequences. PLoS Comput Biol 15(6): e1007129

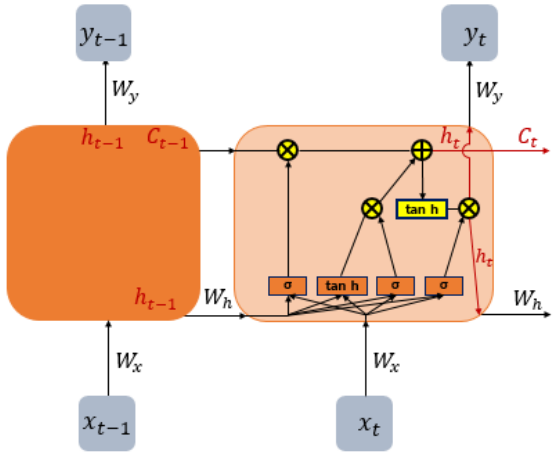
- **Deep Learning : Recurrent Neural Network**

- 입력과 출력을 시퀀스 단위로 처리하는 시퀀스(Sequence) 모델. 앞의 status를 입력받아 다음 status인 output을 내놓는 과정이 연속적으로 발생.

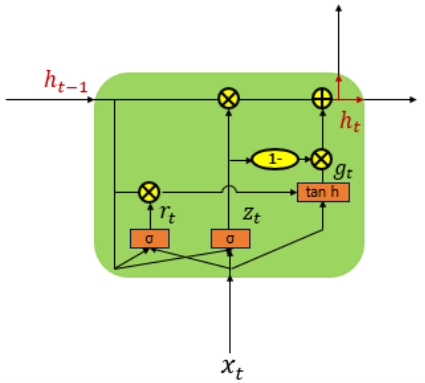




LSTM
(Long Short-term Memory)



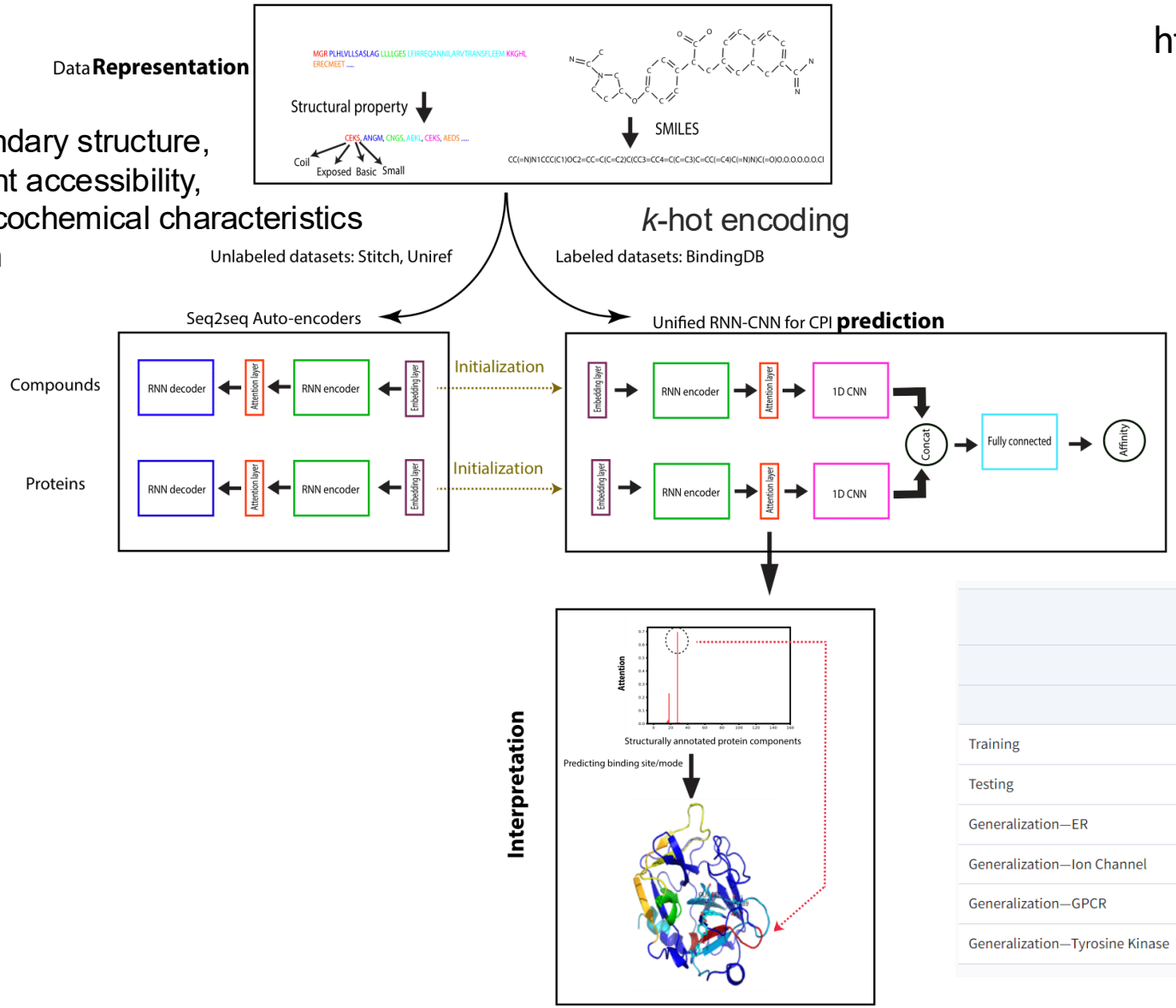
GRU
(Gated Recurrent Unit)



Application : time series prediction, language 영역

- DeepAffinity

https://github.com/Shen-Lab/DeepAffinity



RMSE (and Pearson's r) for pIC_{50} prediction

		Separate RNN-CNN models			Unified RNN-CNN models		
	RF	Single	Parameter ensemble	Parameter+NN ensemble	Single	Parameter ensemble	Parameter+NN ensemble
Training	0.63 (0.91)	0.68 (0.88)	0.67 (0.90)	0.68 (0.89)	0.47 (0.94)	0.45 (0.95)	**0.44** (0.95)
Testing	0.91 (0.78)	0.94 (0.76)	0.92 (0.77)	0.90 (0.79)	0.78 (0.84)	0.77 (0.84)	**0.73** (0.86)
Generalization—ER	**1.41** (0.26)	1.45 (0.24)	1.44 (0.26)	1.43 (0.28)	1.53 (0.16)	1.52 (0.19)	1.46 (0.30)
Generalization—Ion Channel	**1.24** (0.30)	1.36 (0.18)	1.33 (0.18)	1.29 (0.25)	1.34 (0.17)	1.33 (0.18)	1.30 (0.18)
Generalization—GPCR	1.40 (0.25)	1.44 (0.19)	1.41 (0.20)	1.37 (0.23)	1.40 (0.24)	1.40 (0.24)	**1.36** (0.30)
Generalization—Tyrosine Kinase	1.58 (0.11)	1.66 (0.09)	1.62 (0.10)	1.54 (0.12)	1.24 (0.39)	1.25 (0.38)	**1.23** (0.42)

- **Graph Neural Network**

Task : 이웃 노드들 간의 정보를 이용해서 특정 node/edge/graph를 잘표현하는 feature를 찾아내는 것이 목표

1	1	1	0
1	1	1	0
1	1	1	1
0	0	1	1

Adjacency matrix (A)

node

1	0	0
0	1	0
0	0	1
0	1	1

Feature matrix (X)

$$A \times X = H$$

1	1	1	0
1	1	1	0
1	1	1	1
0	0	1	1

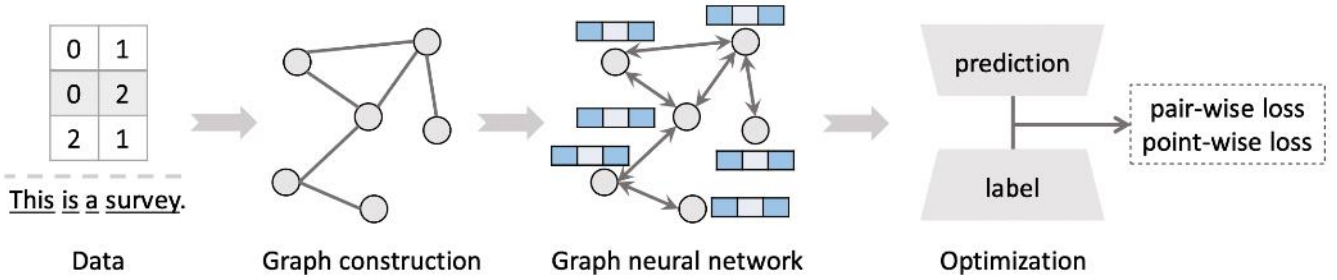
×

1	0	0
0	1	0
0	0	1
0	1	1

=

1	**1**	**1**
1	1	1
1	2	2
0	1	2

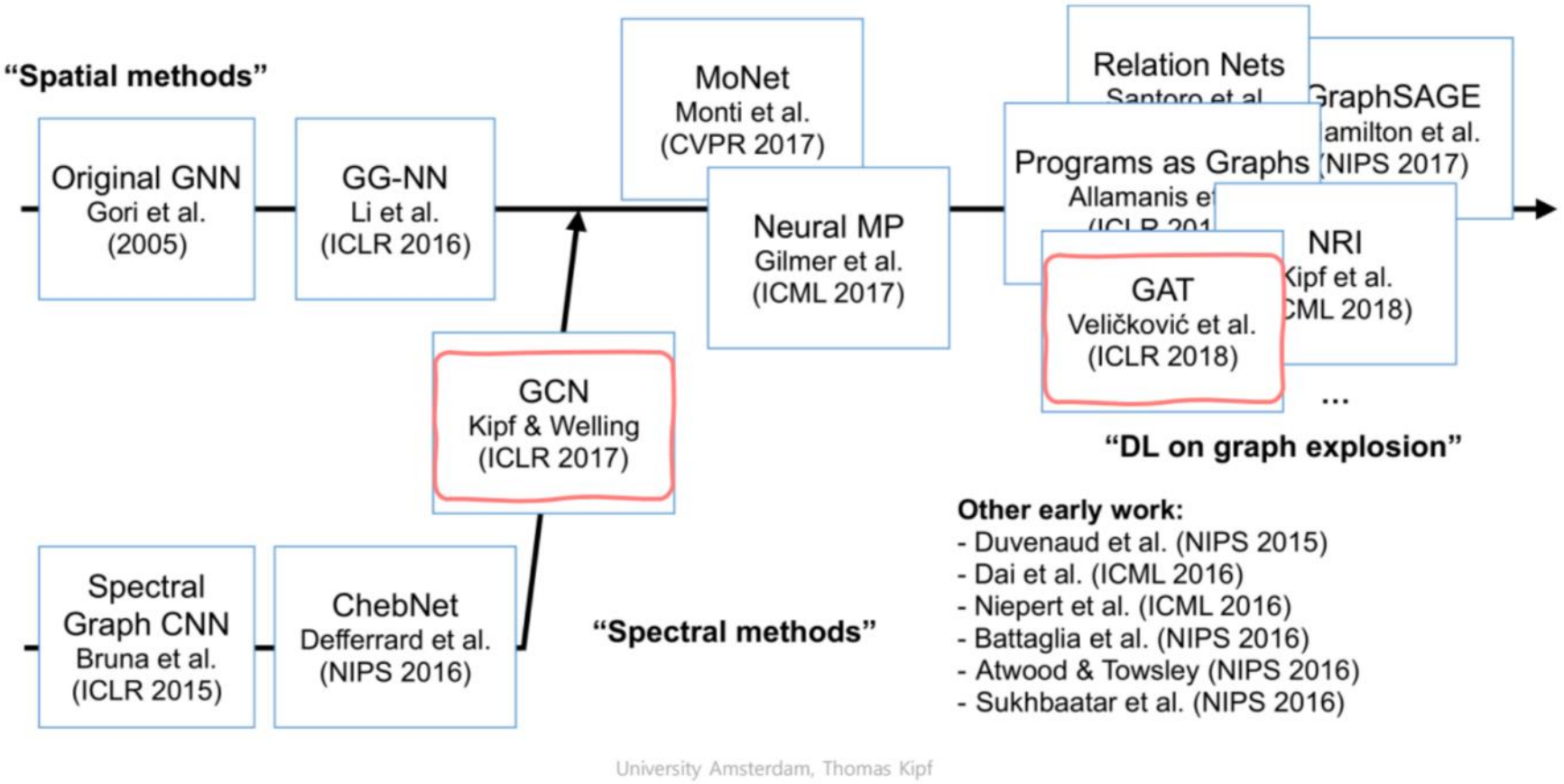
$$h_{1,1} = A_{1,1}X_{1,1} + A_{1,2}X_{2,1} + A_{1,3}X_{3,1} + A_{1,4}X_{4,1}$$
$$h_{1,2} = A_{1,1}X_{1,2} + A_{1,2}X_{2,2} + A_{1,3}X_{3,2} + A_{1,4}X_{4,2}$$
$$h_{1,3} = A_{1,1}X_{1,3} + A_{1,2}X_{2,3} + A_{1,3}X_{3,3} + A_{1,4}X_{4,3}$$



https://arxiv.org/pdf/2109.12843.pdf, Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions

Node 1 feature : 주변 node와의 adjancy를 고려한 주변 feature를 합한 값

- Graph Network



- DGraphDTA

- Graph network

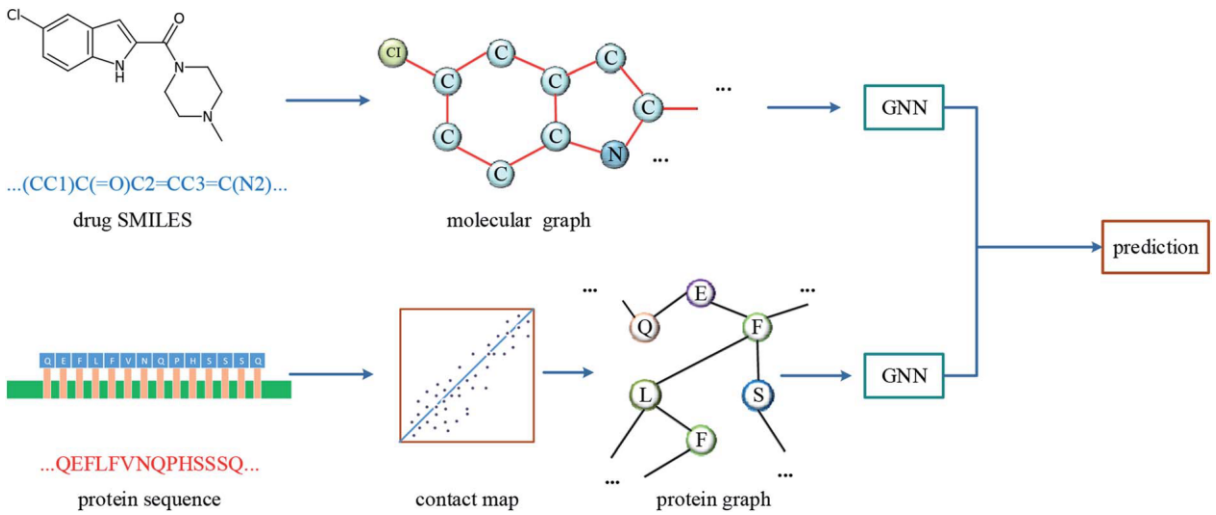


Fig. 1 The architecture of DGraphDTA. Drug molecule SMILES is used for molecule construction and the graph is built up based on it. For the protein, the contact map is constructed based on the protein sequence, and then the graph is built up. After getting two graphs, they enter two GNNs to extract the representations. Finally the representations are concatenated for affinity prediction.

Table 8 Performances of various methods on KIBA dataset

Method	Proteins and compounds	CI	MSE	Pearson
KronRLS	S–W & Pubchem Sim	0.782	0.411	—
SimBoost	S–W & Pubchem Sim	0.836	0.222	—
DeepDTA	S–W & Pubchem Sim	0.710	0.502	—
DeepDTA	CNN & Pubchem Sim	0.718	0.571	—
DeepDTA	S–W & CNN	0.854	0.204	—
DeepDTA	CNN & CNN	0.863	0.194	—
WideDTA	PS + PDM & LS + LMCS	0.875	0.179	0.856
GraphDTA	GAT + GCN & 1D	0.891	0.139	—
DGraphDTA	GCN & GCN	**0.904**	**0.126**	**0.903**

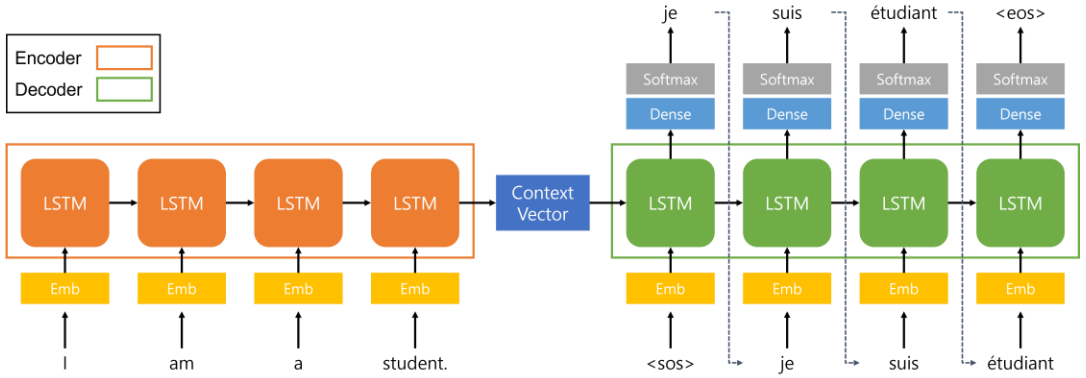
Table 10 r_m^2 scores of various methods on KIBA dataset

Method	Proteins and compounds	r_m^2
KronRLS	S–W & Pubchem Sim	0.342
SimBoost	S–W & Pubchem Sim	0.629
DeepDTA	CNN & CNN	0.673
DGraphDTA	GCN & GCN	**0.786**

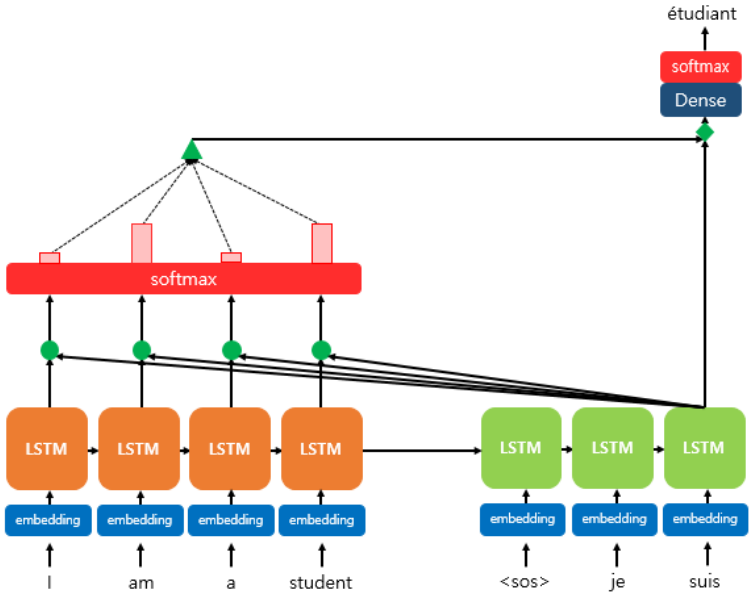
RSC Adv., 2020, 10, 20701–20712

- **Deep Learning : Transformer (Attention)**

- RNN 기반의 seq2seq model 의 한계점 : sequential process의 context embedding이 정보의 손실을 가져옴 → 주어진 set(keys)과 찾고자하는 query와의 상관성을 모두 고려한 attention 을 통해 정보의 손실이 없도록 구성
- 2017년 Google 에서 발표 이후 다양한 분야에서 사용 중 : Chat-GPT



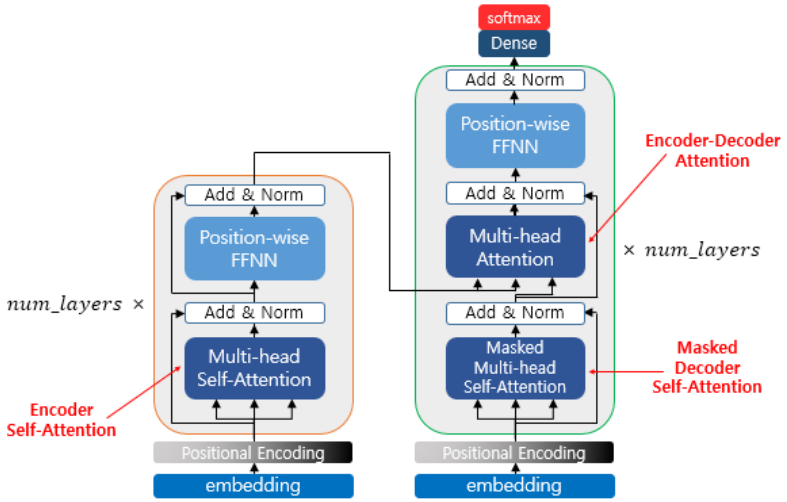
Sequence to sequence RNN model
: input 정보를 압축하는 encoder와 출력을 생성하는 decoder로 구성



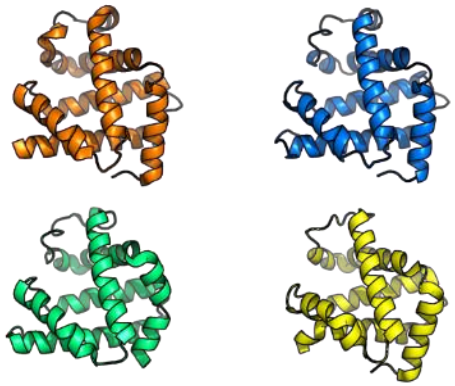
Dot-product Attention
: query 항목의 input keys들과의 관계성을 attention 으로 계산

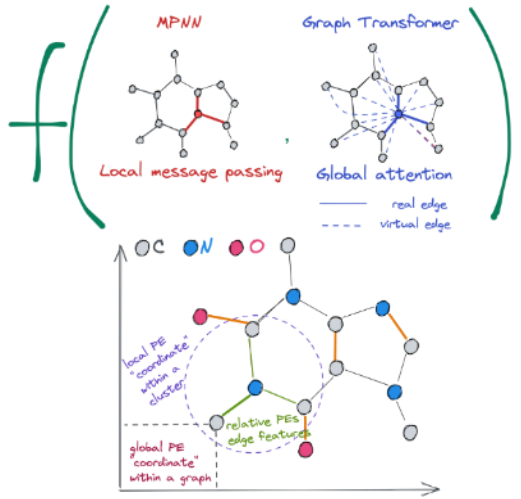
- **Transformer**

- 2년간 아카이브(arXiv)에 게재된 AI 관련 논문의 70%에 트랜스포머가 등장
- Drug Discovery 분야에도 다양한 transformer 기반의 연구가 다수 진행 중



Alphafold3 : Protein-molecule pose





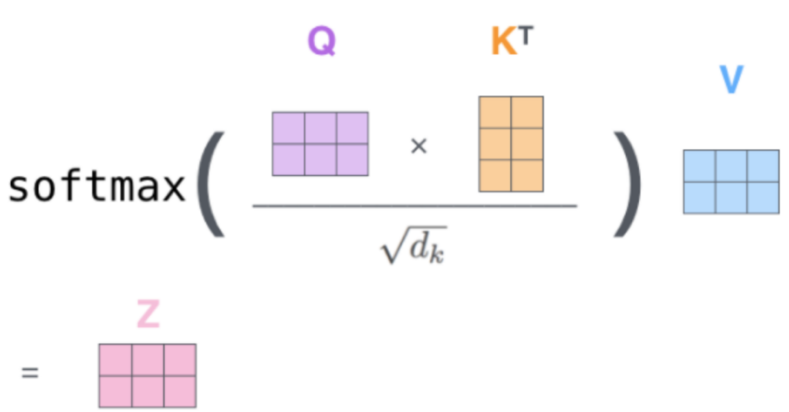
Molecule graph transformer : molecule property prediction

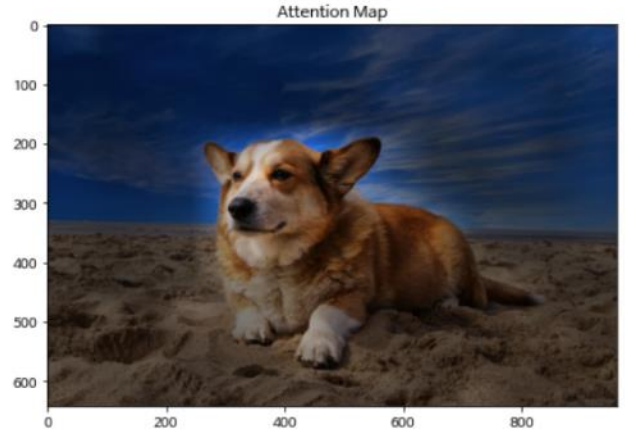
System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$BERT_{BASE}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$BERT_{LARGE}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

2018년 BERT model이 기존의 NLP SOTA model들보다 우수한 성능을 보임

- **Attention Map**

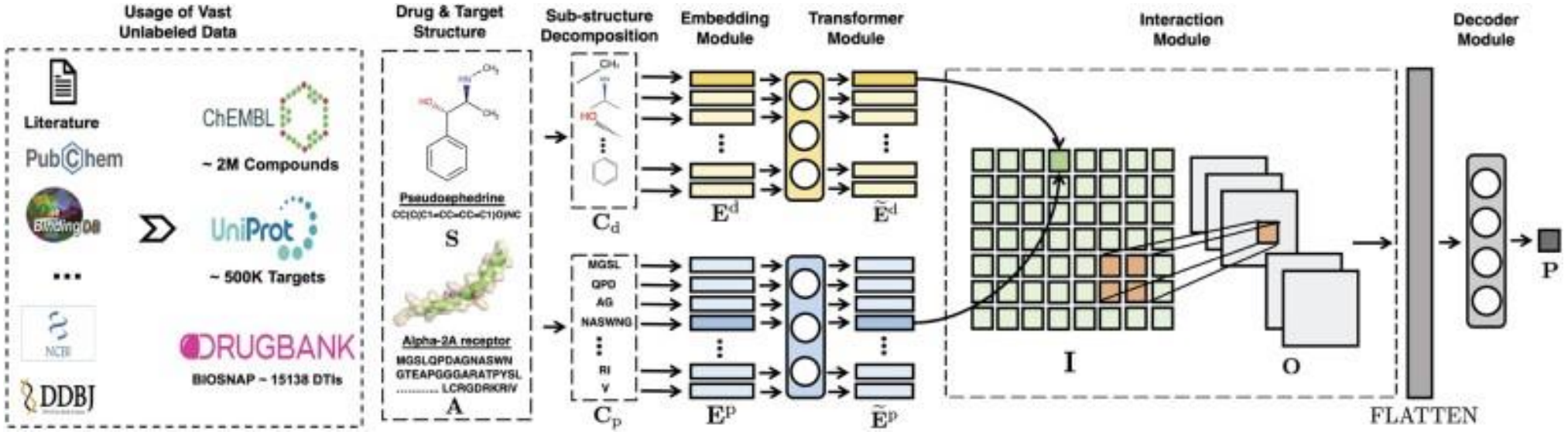
- Attention Matrix 시각화 : 각 입력 간의 관계(weight)를 계산하는 과정에서 계산되는 Attention 가중치를 시각화하여 중요한 weight를 가지는 항목을 시각화할 수 있다.





- **MolTrans**

Token embedding



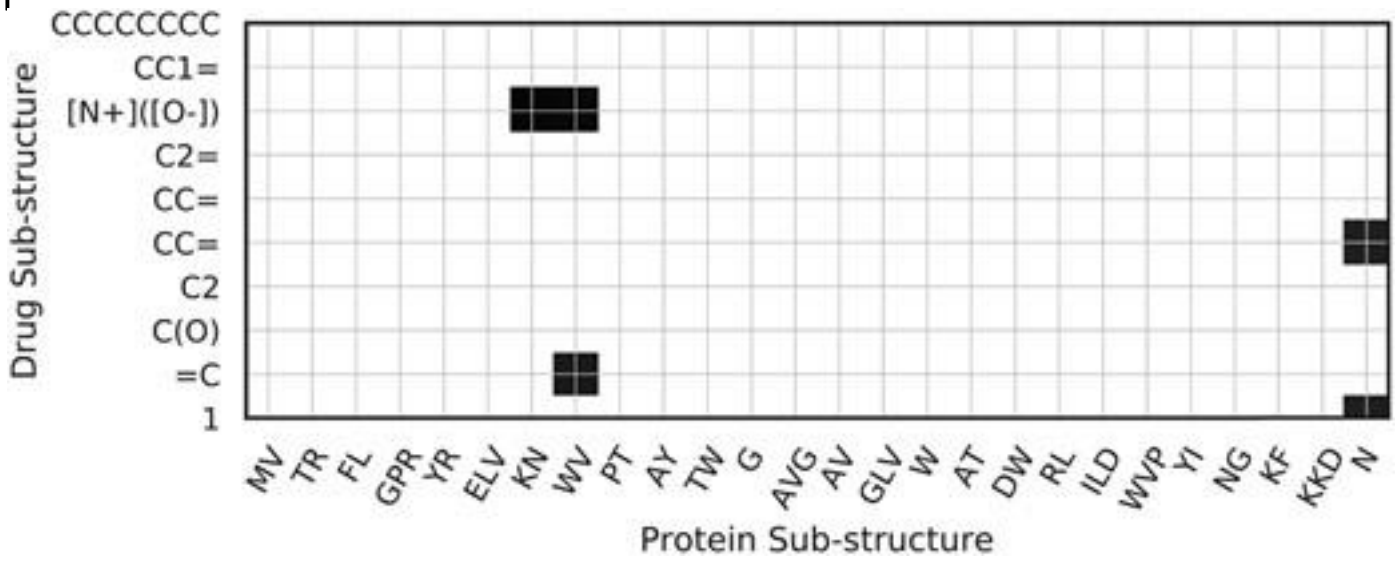
ROC-AUC of five random runs

Settings	DeepDTI	DeepDTA	DeepConv-DTI	MolTrans
Unseen drugs	0.843 ± 0.003	0.849 ± 0.007	0.847 ± 0.009	0.853 ± 0.011
Unseen proteins	0.759 ± 0.029	0.767 ± 0.022	0.766 ± 0.022	0.770 ± 0.029

Bioinformatics 37.6 (2021): 830-836

- **MolTrans**

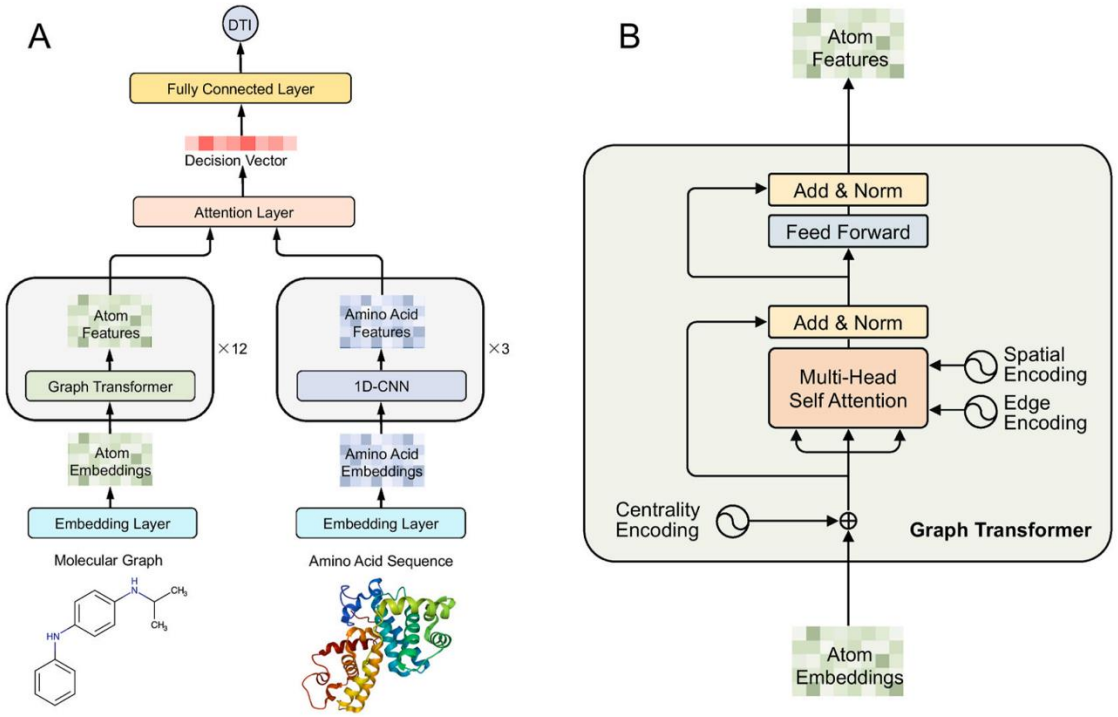
- Ligand-protein sub-structure의 attention map으로 interaction이 있을만한 위치를 시각화



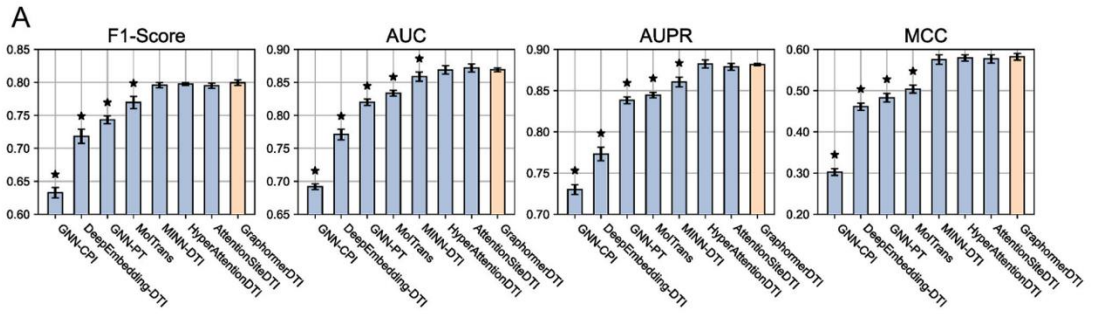
“ nitrogen oxide group [N+]([O-]) and KNWV has the highest interaction coefficient, matching with the previous study (Lightbown and Jackson, 1956) who showed that nitrogen oxide group is essential for cytochrome inhibition activity”

- Graph-Transformer

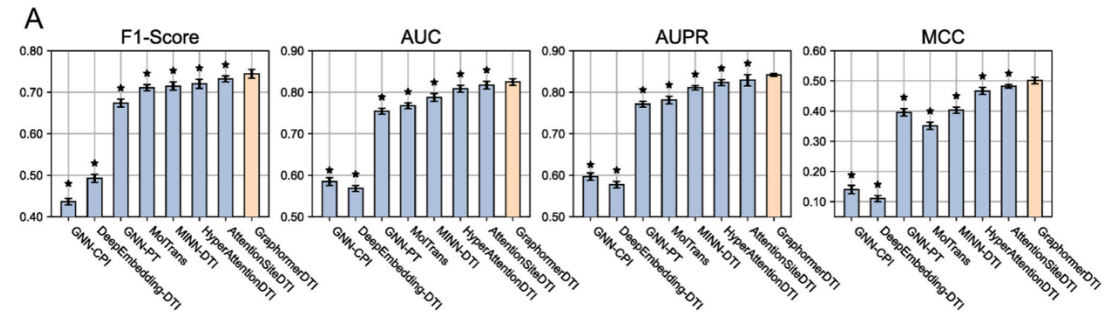
- GraphomerDTI



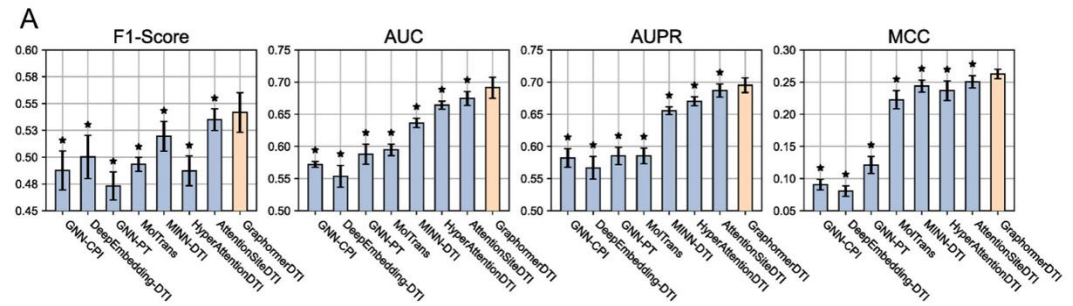
Transductive



Drug inductive



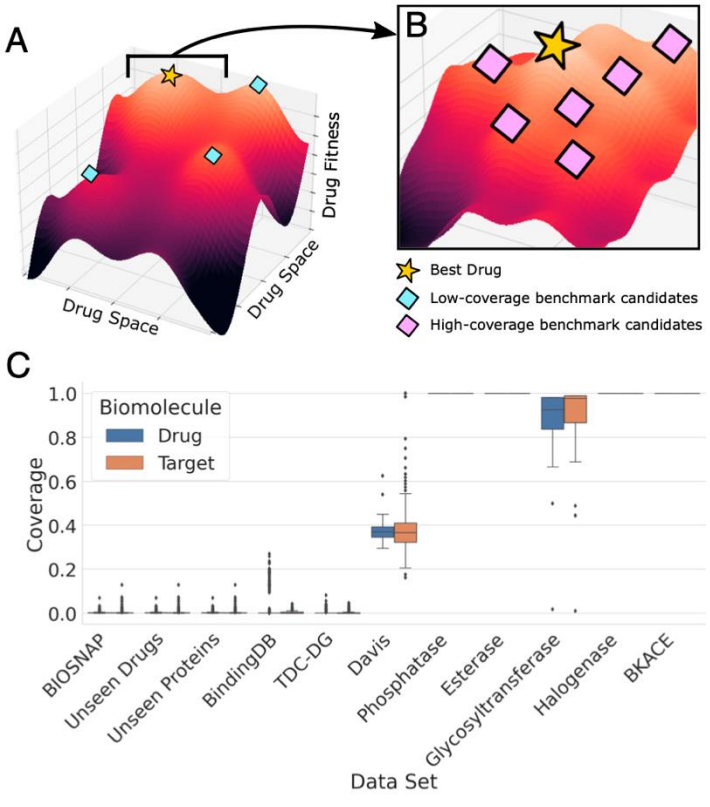
Drug-target inductive

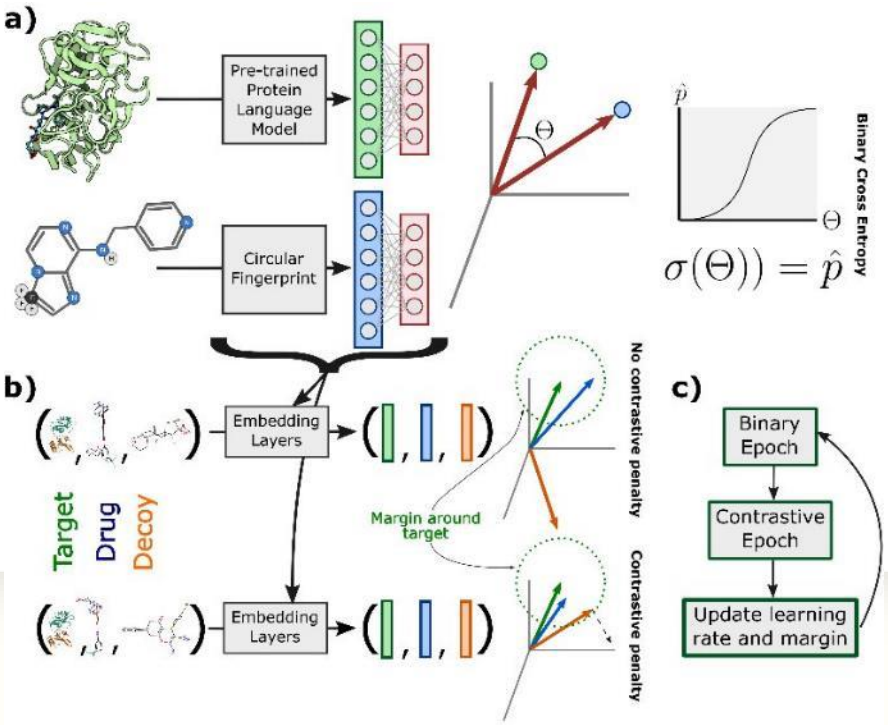


Computers in Biology and Medicine, 173, 108339 (2024)

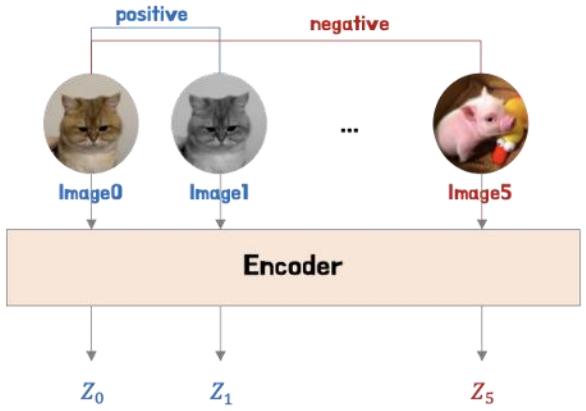
- **ConPLex**

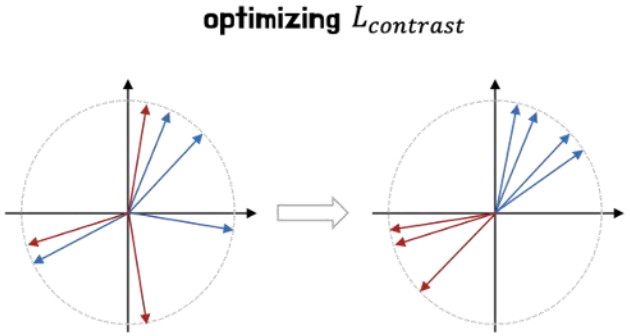
- **특징**
  - BindingDB+pre-trained bert model을 이용하여 general embedding 특성 학습
  - DUDe Decoy set을 이용하여 specificity를 높임
  - 다양성이 적은 데이터에서 높은 specificity를 얻을 수 있음





- **ConPlex**



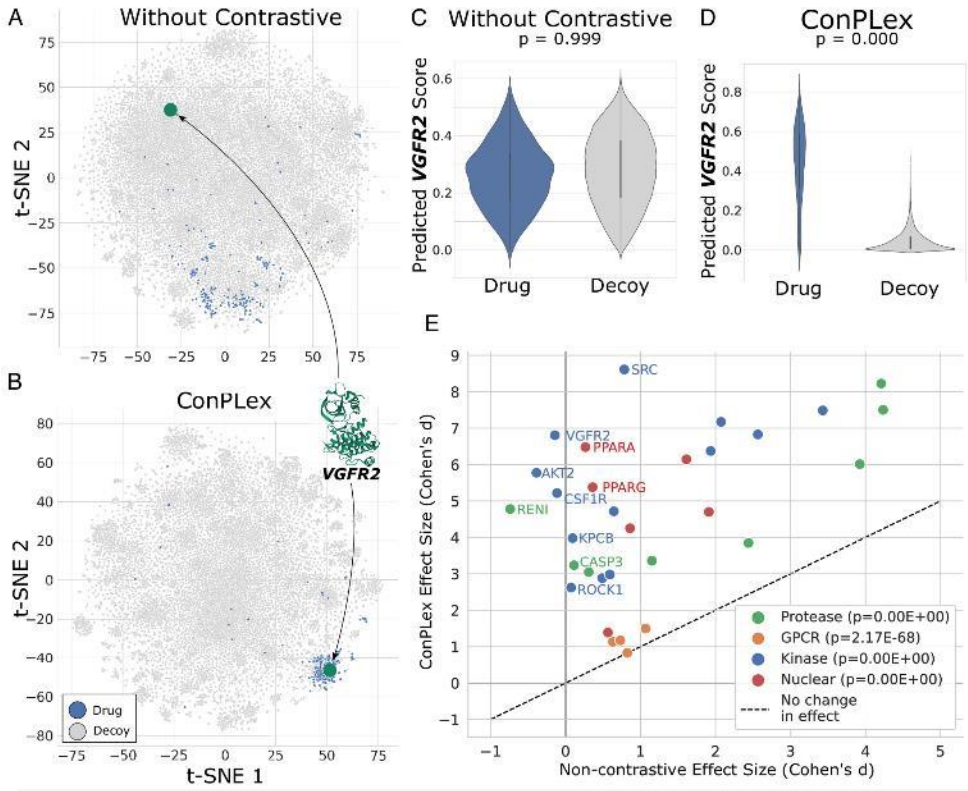


$$L_{contrast} = -log \frac{\exp(z_0 \cdot z_1)}{\sum_{i=1}^{K} \exp(z_0 \cdot z_i)}$$

Table 1. ConPLex is highly accurate and generalizes broadly in low coverage settings

Dataset	ConPLex	EnzPred-CPI	MolTrans	GNN-CPI†	DeepConv-DTI†	Ridge
BIOSNAP	0.897 ± 0.001	0.866 ± 0.003	0.885 ± 0.005	0.890 ± 0.004	0.889 ± 0.005	0.641 ± 0.000
BindingDB	0.628 ± 0.012	0.602 ± 0.006	0.598 ± 0.013	0.578 ± 0.015	0.611 ± 0.015	0.516 ± 0.000
DAVIS	0.458 ± 0.016	0.277 ± 0.009	0.335 ± 0.017	0.269 ± 0.020	0.299 ± 0.039	0.320 ± 0.000
Unseen Drugs	0.874 ± 0.002	0.844 ± 0.005	0.863 ± 0.005	–	0.847 ± 0.009	N/A
Unseen Targets	0.842 ± 0.006	0.795 ± 0.004	0.668 ± 0.045	–	0.766 ± 0.022	0.617 ± 0.000

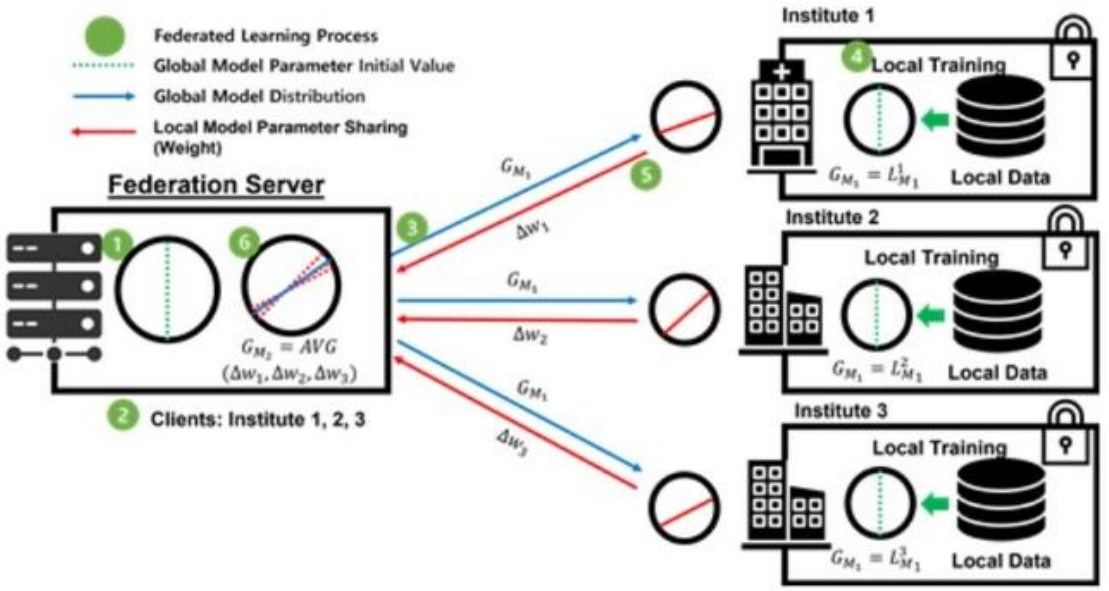
ConPLex outperforms several state-of-the-art methods, including EnzPred-CPI (25), MolTrans (13), GNN-CPI (34), and DeepConv-DTI (12), as well as a simple single-target Ridge regression model, on several low- and zero- coverage benchmark datasets. We report the average and SD of the area under the precision–recall curve (AUPR) for 5 random initializations of each model. Metrics for models with † are taken from ref. 13. Ridge regression cannot be applied for the **Unseen Drugs** dataset since a separate model is trained for each drug in the training set.

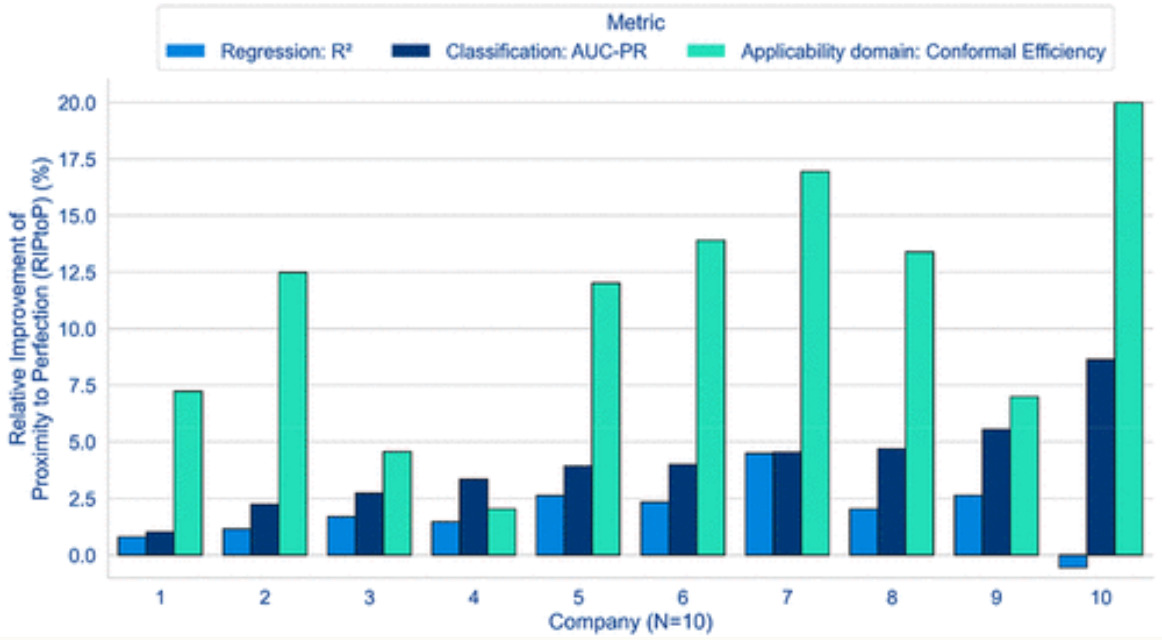


- [참고] 연합학습 : K-MELLODDY

- Federated learning : joint modeling of tasks from multiple companies without explicitly sharing the underlying raw training data.

  Data 공유 없이 각 dataset에서 학습한 모델의 parameter를 합쳐 하나의 모델을 만드는 방법





- **Post-Processing**

- Data-fusion
  - Rank-based : 각 기법의 순위를 결합
  - Score-based : 각 기법의 Score/Z-score를 가중 평균
  - Independent selection : 각 기법에서 상위 선택

compound	2D similarity	Pharmacophore	ML model	Average
cpd0001	7500	**60**	2000	3186
cpd0002	5000	9800	7600	7466
cpd0003	**370**	1050	9000	**3473**
…				
cpd9999	9500	4010	**120**	4543

compound	2D similarity	Pharmacophore	ML model	Average
cpd0001	0.41	**0.87**	0.65	**0.64**
cpd0002	0.56	0.16	0.23	0.32
cpd0003	**0.79**	0.67	0.34	0.6
…				
cpd9999	0.24	0.55	**0.81**	0.53

- **Post-Processing**

II. Molecular Docking
- Docking score/Energy calculation 이용한 2차 filter
- Interaction 분석

III. Physiochemical Filtering
- Physiochemical Property filter : logP, logS, Druggability, Permeability, PAINS filter
- Toxicity : 특정 function group 제거, 예측모델 사용

IV. Diversity analysis
- 구조 다양성을 위해 상위 ranking 에서 similarity cutoff 를 주고 상위 N개를 취득
- Clustering 후 상위 cluster 내에서 물질을 선택

V. Visual selection (option)

- **Summary**

- 리간드 기반 가상탐색
  - fingerprint, shape, pharmacophore screening
  - ML/DL 기반의 drug-target interaction model
- 데이터
  - Database : affinity/structure를 가지는 public database로 PDBbind, BindingDB, Chembl 등이 있음
  - Encoding : SMILES One-hot encoding, Graph, Fingerprint 등
- ML/DL algorithm training, optimization
  - 모델학습은 최적화를 통해 손실함수를 최소화하는 모델의 파라미터를 찾는 것
- Virtual Screening metrics
  - Model Task (회귀, 분류)에 따른 기본적인 metric들이 존재
  - Virtual screening 성능 평가를 위한 metric : CI, Enrichment Factor
- Post-processing
  - 최종 물질 선택에 앞서 Data fusion, Physiochemical filter, Clustering 등의 후처리 과정이 중요
- Research trend and examples