

《 화학 1 세특 자유 탐구 보고서 》

하이에듀

주제 파이썬과 RDKit으로 삼차원 분자 구조 그리기

요약

파이썬과 파이썬 라이브러리 RDKit를 이용해 분자 구조를 그릴 수 있습니다. 이번 탐구에는 Google Colab라는 사이트에서, 간단한 분자의 구조를 직접 파이썬으로 그려보고, 전자쌍 반발 이론과 일치하는지 확인해보겠습니다. 코드는 ipynb 파일 확인 부탁드립니다.

내용

I. 탐구 동기

계산화학이 발달해 다양한 화학 실험을 컴퓨터 안에서 할 수 있는 요즘, '전자쌍 반발 이론'에 대해서도 컴퓨터로 알아볼 수 있을지가 궁금해졌다. 분자 구조를 그릴 수 있는 프로그램은 ChemSketch, ChemDraw 등 많지만, 직접 파이썬과 파이썬 라이브러리를 이용해 코딩해보고 싶었다.

파이썬 라이브러리 rdkit을 이용하면 2차원으로 표현해 분자 구조를 그릴 수 있고, kora와 py3Dmol을 이용하면 3차원 분자 구조를 그릴 수 있다. 이번 탐구에서는 Google Colab이라는 일종의 텍스트 에디터를 이용해, 파이썬으로 분자 구조를 그리고, 전자쌍 반발 이론과 일치하는지 확인해보기로 하였다.

II. 탐구 내용 및 결과

1. 파이썬과 RDKit로 2차원, 3차원 분자 구조 그리고 비교하기

구글 콜랩(Google Colab)에 접속해, rdkit을 설치하고 Chem, Draw, AllChem을 임포트한다.

```
[ ] pip install rdkit
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting rdkit
  Downloading rdkit-2022.9.3-cp38-cp38-manylinux_2_17_x86_64_manylinux2014_x86_64.whl (29.3 MB)
-----
29.3/29.3 MB 50.8 MB/s eta 0:00:00
Requirement already satisfied: Pillow in /usr/local/lib/python3.8/dist-packages (from rdkit) (7.1.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.8/dist-packages (from rdkit) (1.21.6)
Installing collected packages: rdkit
Successfully installed rdkit-2022.9.3
```

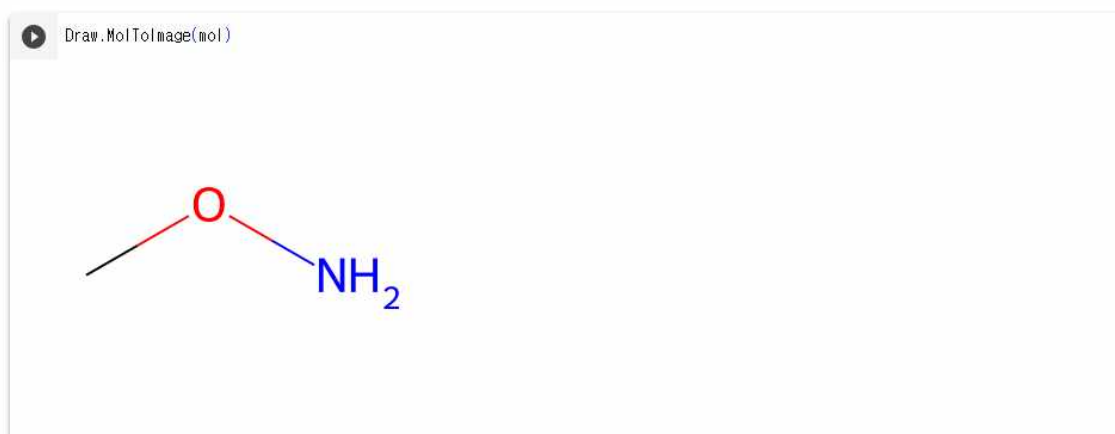
```
[ ] from rdkit import Chem
from rdkit.Chem import Draw
from rdkit.Chem import AllChem
```

그다음으로, mol에 분자 정보를 저장해 Draw로 직접 그릴 수 있게 한다.

여기에서 작은 따옴표 안에 들어가는 건 그릴 분자의 SMILES 코드이다. SMILES 코드는 화학물질의 구조를 선형으로 표현하는 표기법 중 하나이다. 보통 H(수소)는 생략되어 표현되는 경우가 많다. 예시 분자의 경우에도, CH₃ONH₂에서 H가 생략되어 'CON'으로 표기되었다.

각 분자에 해당하는 SMILES 코드는 PubChem 홈페이지에서 찾을 수 있다.

```
[ ] def mol_with_atom_index(mol):  
    for atom in mol.GetAtoms():  
        atom.SetAtomMapNum(atom.GetIdx())  
    return mol  
  
[ ] mol = Chem.MolFromSmiles('CON')
```



이렇게 되면 굽은 모양의 분자가 만들어진다. 중심 분자인 O의 경우 빈 오비탈이 2개, 결합이 2개이기 때문에 굽은형으로 표현된다. 따라서 분자 모형이 잘 그려졌음을 알 수 있다.

다음으로 분자의 3D 구조가 잘 그려지는지 확인해보았다. pip으로 kora와 py3Dmol을 설치하고, 모두 코드 안에서 활용하기 위해 임포트하였다.

```
!pip install kora py3Dmol  
import kora.install.rdkit  
import py3Dmol
```

show를 이용해, SMILES 코드를 받아 3D 이미지를 출력하는 함수를 만든다. 그렇게 되면 분자의 3D 이미지가 출력된다.

아까 입력했던 'CON'을 그대로 입력하면 다음 그림이 나온다.

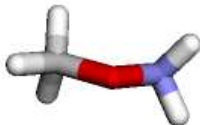
```

def show(smi, style='stick'):
    mol = Chem.MolFromSmiles(smi)
    mol = Chem.AddHs(mol)
    AllChem.EmbedMolecule(mol)
    AllChem.MMFFOptimizeMolecule(mol, maxIters=200)
    mblock = Chem.MolToMolBlock(mol)

    view = py3Dmol.view(width=300, height=300)
    view.addModel(mblock, 'mol')
    view.setStyle({'style':{}})
    view.zoomTo()
    view.show()

show('CON')

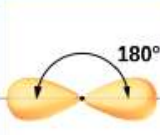
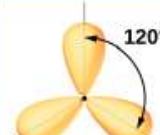

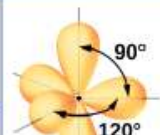
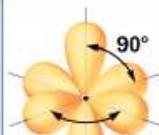
```



아까와 마찬가지로, 가운데 산소의 두 결합이 굽은 모양이 나왔다. 따라서 전자쌍 반발 모형이 적용되고 있다고 생각할 수 있다.

3. 전자쌍 반발 이론과 3차원 분자 구조의 일치 확인하기

아래와 같은 분자들을 통해, 코드의 3차원 구조와 전자쌍 반발 이론의 3차원 구조를 비교해보았다. 분자는 오른쪽 4개만 사용하였다.

Spatial arrangement					
Line-dash-wedge notation	<chem>H-Be-H</chem>	<chem>H H-B-H</chem>	<chem>H H-C-H / \</chem>	<chem>F F-P-F / \</chem>	<chem>F F-S-F / \</chem>

```

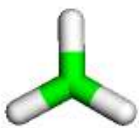


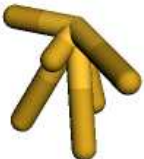
def show(smi, style='stick'):
    mol = Chem.MolFromSmiles(smi)
    mol = Chem.AddHs(mol)
    AllChem.EmbedMolecule(mol)
    AllChem.MMFFOptimizeMolecule(mol, maxIters=200)
    mblock = Chem.MolToMolBlock(mol)

    view = py3Dmol.view(width=300, height=300)
    view.addModel(mblock, 'mol')
    view.setStyle({'style':{}})
    view.zoomTo()
    view.show()

show('B') #BH3
show('C') #CH4
show('FP(F)(F)(F)F') #PF5
show('FS(F)(F)(F)(F)F') #SF6

```

위는 사용했던 코드이다.

			
BH3	CH4	PF5 (error)	SF6 (error)

아래는 결과이다.

BH3, CH4의 경우, 전자쌍 반발 모형과 일치하는 제대로 된 모형이 나왔다. 그러나 PF5와 SF6의 경우에는 비공유 전자쌍이 하나도 없음에도 한쪽으로 쏠려 있는, 비정상적인 모형이 나왔다. 그 이유를 찾아본 결과, 결합이 5개가 넘어갈 때 사용하는 오비탈이 다른데, 이 과정에서 오류가 발생한 것 같다..

III. 결론 및 느낀 점

파이썬으로 분자 구조를 알아볼 수 있는 코드를 짜면서, 라이브러리를 이용하는 것인데도 어렵다는 것을 느꼈다. 그렇지만, 그만큼 화학에도 컴퓨터가 필요하다는 것을 알게 된 것 같다.

찾아보니 분자 모양을 결정하는 이론은 전자쌍 반발 이론 이외에도 정말 많다는 것을 알게 되었다. 분자 모양을 결정하는 다른 이론에 대해서도 더 알아보고 싶었다.