PetroPalette: The Petro-Informatics Chemical Structure Database

Sydney Mazat and Benjamin Janesko*

Department of Chemistry & Biochemistry, Texas Christian University

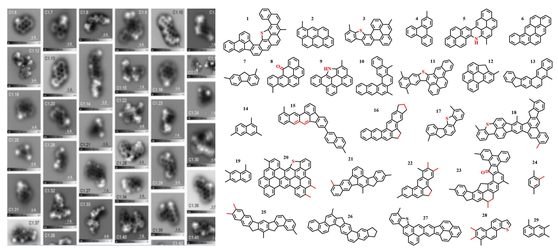
THIS PROJECT

Petroleum crude oil, unconventional crudes, and renewable bio-crudes are essential materials in our everyday lives. Crudes are highly complex chemical mixtures, estimated to contain between 100,000 and 100,000,000,000,000,000 unique molecules. Since 2015, single-molecule imaging has visualized hundreds of chemical structures, and historical literature has published thousands of proposed structures. This project builds an open database populated with published crude structures enabling data-driven analysis of these structures, and detailed workflows, allowing for easy future insertion of new molecules into the database.

LONG TERM GOAL

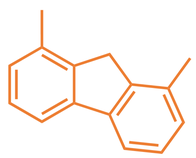
This database can be used to make calculations and predict characteristics of molecules, such as viscosity, density, and reactivity, which are all critical in refinery plants, transportation, and usage of these fuels. Machine learning can be used to determine important characteristics of crudes molecules, leading to more refined and successful hypotheses. PetroPalette will eventually predict possible structures of petroleum crudes and their ensemble property distributions. The next step in this project is to add more real crudes into PetroPalette using the successful workflows, followed by machine learning to build predicted structures.

BUILDING PETROPALETTE:



1. Single Molecule Imaging produces chemical structures

2. Convert published structures into SMILES string



Cc1ccc2c(c1)Cc3c(C)cccc23

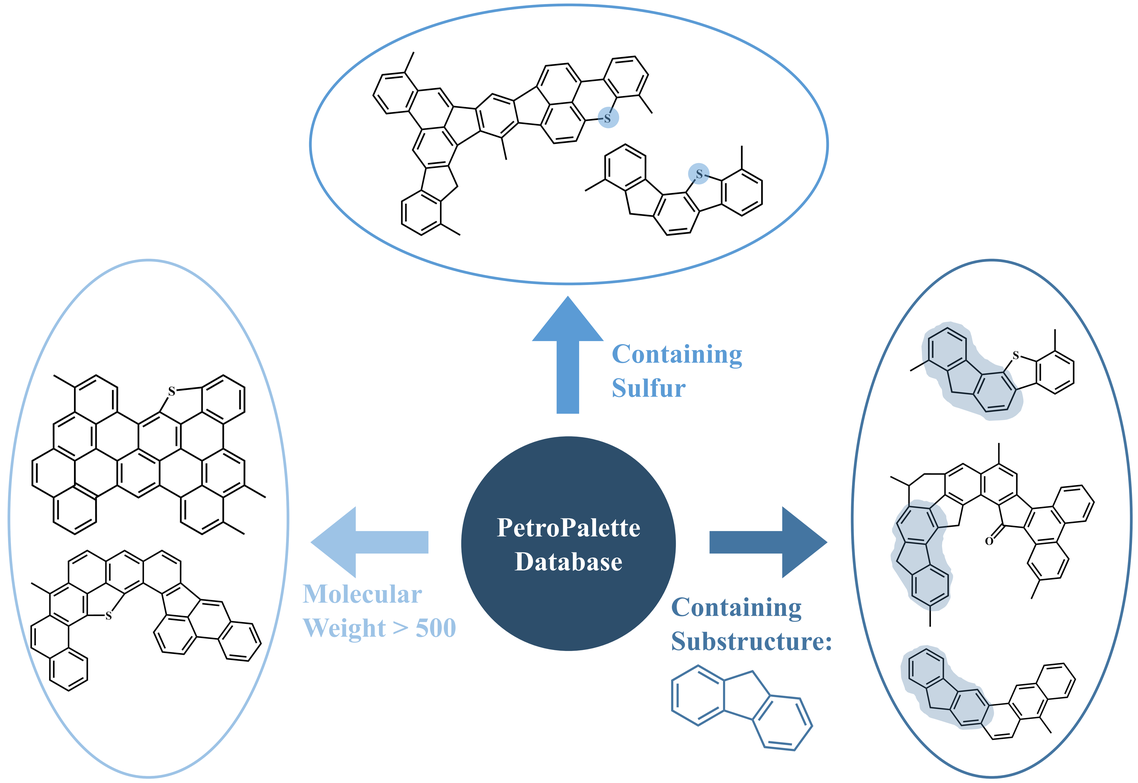
```
def descriptor_table (smiles, classification, doi):
#create iupac name, molecular weight, and molecular formula list from smiles
    iupac=[]
    mw = []
    formula = []
    logP = []
    for sml in smiles:
        #iupac list
        compounds = pub.get_compounds(sml, namespace = 'smiles')
        match = compounds[0]
        iupac.append(match.iupac_name)
        #molecular weight list
        mol_noH = Chem.MolFromSmiles(sml)
        molwithH = Chem.AddHs(mol_noH)
        mass = Descriptors.MolWt(molwithH)
        mw.append(mass)
        #molecular formula list
        form = CalcMolFormula(molwithH)
        formula.append(form)
        #log P list
        logP_value = crip.MolLogP(molwithH)
        logP.append(logP_value)
    #create dataframe and add classication & doi source columns
    d = {'smiles_string': smiles,
         'iupac_name': iupac,
         'molecular_weight': mw,
         'molecular_formula': formula,
         'logP': logP}
    description = pd.DataFrame(data = d)
    description['Classification'] = classification
    description['DOI_Source'] = doi
    return description
```

3. Produce descriptors from SMILES string

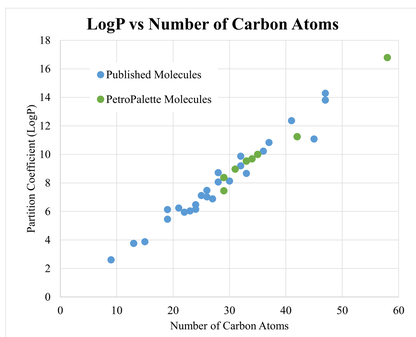
4. Insert structure and its descriptors into PetroPalette

smiles_string	IUPAC_name	molar_mass	molecular_formula	log_p	DBE	classification	doi_source
Cc8c2ccc1cccccc1c2c9sc%11c7c(cccc6c5cc3ccccc...	18-methyl-41-thiaund...	547	C41H22S	12.3673	31	steam-cracked tar	https://doi.org/10.1021/acs.iecr.8b03962
Cc1ccc2CCC3Cc5ccccc6ccc4cc1c2c3c4c56	NULL	294	C23H18	6.04052	15	steam-cracked tar	https://doi.org/10.1021/acs.iecr.8b03962
Cc6ccc7c5ccc4c3ccccc2cc(C)c1ccccc1c23)c4c5...	NULL	387	C28H18S	8.72194	20	steam-cracked tar	https://doi.org/10.1021/acs.iecr.8b03962
Cc4ccc3c1ccccc1c2ccccc2c3c4	2-methyltriphenylene	242	C19H14	5.45462	13	steam-cracked tar	https://doi.org/10.1021/acs.iecr.8b03962
c2cc3ccc4cc1c6cc9ccccBccc7ccc[nH]c1c5ccc(c2...	NULL	415	C32H17N	9.2054	25	steam-cracked tar	https://doi.org/10.1021/acs.iecr.8b03962
c1ccc2c(c1)cc3ccc4cc7ccccBccc5cc2c3c4c5c67	heptacyclo[11.11.1.1...	326	C26H14	7.4814	20	steam-cracked tar	https://doi.org/10.1021/acs.iecr.8b03962
Cc1ccc2c(c1)Cc3c(C)cccc23	1,7-dimethyl-9H-fluor...	194	C15H14	3.87464	9	steam-cracked tar	https://doi.org/10.1021/acs.iecr.8b03962
Cc1ccc2c1cc3ccc4ccc5ccccc6c(=O)c2c3c4c56	NULL	318	C24H14O	6.15222	18	steam-cracked tar	https://doi.org/10.1021/acs.iecr.8b03962
Cc2cc3ccc4c7ccccBc1ccccc1c5[nH]c2c(c34)c5c67	NULL	329	C23H15N	7.11792	19	steam-cracked tar	https://doi.org/10.1021/acs.iecr.8b03962
Cc3cc2cc1cccc1cc2c6c3ccc5c(C)c4ccccc4cc56	NULL	356	C28H20	8.06944	19	steam-cracked tar	https://doi.org/10.1021/acs.iecr.8b03962
Cc1ccc(C)c2cc7c(cc12)Bc8c6ccc3ccccc3c5ccc4cc...	NULL	437	C32H20S	9.87514	23	steam-cracked tar	https://doi.org/10.1021/acs.iecr.8b03962
Cc1ccc2c6cccc5ccc4ccc3Cc1c2c3c4c56	NULL	278	C22H14	5.94982	16	steam-cracked tar	https://doi.org/10.1021/acs.iecr.8b03962

PETROPALETTE IN ACTION:



This figure compares the Partition Coefficient value of published crudes[1] (blue) to that of a fully automated analysis of other crudes[2] (green) using the workflows and PetroPalette. These values following the linear trend help to validate the workflows.



Molecules Follow Trend
↓
Valid Workflows

Acknowledgement: This work is supported by the American Chemical Society Petroleum Research Fund

1. *Ind. Eng. Chem. Res.* 2018, 57, 46, 15935–15941
2. *Energy & Fuels* 2022, 36, 16, 8714–8724

Contact: s.mazat@tcu.edu