a. Extraction of ADMET data from RD Kit.

```
DEFINE FUNCTION calculate_properties(smiles):
    SET mol TO Chem.MolFromSmiles(smiles)
    IF mol is None:
        RETURN None
    # calculate ADME/T properties
    SET mw TO Descriptors.MolWt(mol)
    SET logp TO Descriptors.MolLogP(mol)
    SET tpsa TO Descriptors.TPSA(mol)
    SET hbd TO Descriptors.NumHDonors(mol)
    SET hba TO Descriptors.NumHAcceptors(mol)
    SET rotb TO Descriptors.NumRotatableBonds(mol)
    SET arom TO Descriptors.NumAromaticRings(mol)
    SET # ring TO Descriptors.NumRings(mol)
    # additional ADME/T properties
    SET gi_absorption TO 'Unknown'
    IF logp >= 0.4 and logp <= 5.6 and tpsa <= 140:
        SET gi_absorption TO 'High'
    ELSEIF logp >= -2 and logp <= 6 and tpsa <= 180:
        SET gi_absorption TO 'Moderate'
    ELSE:
        SET gi_absorption TO 'Low'
    SET bbb_permeant TO 'Unknown'
    IF logp >= -0.4 and logp <= 2.5 and hbd <= 2:
        SET bbb_permeant TO 'Yes'
    ELSE:
        SET bbb_permeant TO 'No'
    SET p_gp_substrate TO 'Unknown'
    IF logp <= 3 and hbd <= 5 and hba <= 10:
```

```
    SET p_gp_substrate TO 'Yes'
  ELSE:
    SET p_gp_substrate TO 'No'
  SET cyp1a2_inhibitor TO 'No'
  SET cyp2c19_inhibitor TO 'No'
  SET cyp2c9_inhibitor TO 'No'
  SET cyp2d6_inhibitor TO 'No'
  SET cyp3a4_inhibitor TO 'No'
  IF logp <= 5 and tpsa <= 120:
    SET cyp1a2_inhibitor TO 'Yes'
    SET cyp2c19_inhibitor TO 'Yes'
    SET cyp2c9_inhibitor TO 'Yes'
    SET cyp2d6_inhibitor TO 'Yes'
    SET cyp3a4_inhibitor TO 'Yes'
  SET lipinski TO 0
  IF mw <= 500 and logp <= 5 and hba <= 10 and hbd <= 5:
    SET lipinski TO float(1)
  SET ghose TO 0
 IF mw >= 160 and mw <= 480 and logp >= -0.4 and logp <= 5.6 and hba >= 0 and hba <= 10 and hbd >= 0 and
hbd <= 5:
    SET ghose TO float(1)
  SET veber TO 0
  IF rotb <= 10 and tpsa <= 140 and logp <= 5:
    SET veber TO float(1)
  SET result TO {
    'Molecular weight': mw,
    'LogP': logp,
    'TPSA': tpsa,
    'H-bond donors': hbd,
    'H-bond acceptors': hba,
    'Rotatable bonds': rotb,
```

'Aromatic rings': arom,

'Rings': ring,

'GI absorption': gi_absorption,

'BBB permeant': bbb_permeant,

'P-gp substrate': p_gp_substrate,

'CYP1A2 inhibitor': cyp1a2_inhibitor,

'CYP2C19 inhibitor': cyp2c19_inhibitor,

'CYP2C9 inhibitor': cyp2c9_inhibitor,

'CYP2D6 inhibitor': cyp2d6_inhibitor,

'CYP3A4 inhibitor': cyp3a4_inhibitor,

'Lipinski': lipinski,

'Ghose': ghose,

'Veber': veber

}

RETURN result

Appending the properties to a dataframe

SET admet_prop TO []

SET nan_smiles TO []

FOR smile IN df['SMILES']:

SET res TO calculate_properties(smile)

IF res :

SET res['SMILES'] TO smile

admet_prop.append(res)

ELSE:

nan_smiles.append(smile)

b. Extraction of compound information by Pubchem.Py

DEFINE FUNCTION smile_to_iupac(smile):

TRY:

compounds =pubchempy.get_compounds(smile, namespace='smiles')

SET match TO compounds[0].to_dict(properties=['cid','canonical_smiles','exact_mass',

```
                'h_bond_acceptor_count','h_bond_donor_count','iupac_name','molecular_formula','inchi'])
        RETURN match
    except PubChemHTTPError as e:
        RETURN None
```

Appending the properties to a dataframe

```
SET pub_prop TO {}
FOR val IN df['SMILES']:
    IF val NOT IN pub_prop:
        SET prop TO smile_to_iupac(val)
        IF prop:
            pub_prop[val]= prop
with open('result.json', 'w') as fp:
    json.dump(pub_prop, fp)
```

Save the new dataframe with all properties as a .csv file

Python code for extraction of compound information by RDKit and Pubchempy