pseudocode

January 21, 2024

Extraction of ADMET Data from RD Kit

Calculate ADMET Properties

```
function CALCULATE_PROPERTIES(smiles)
    mol ← Chem.MolFromSmiles(smiles)
    if mol is None then
        return None
    end if
    mw ← Descriptors.MolWt(mol)
    logp ← Descriptors.MolLogP(mol)
    tpsa ← Descriptors.TPSA(mol)
    hbd ← Descriptors.NumHDonors(mol)
    hba ← Descriptors.NumHAcceptors(mol)
    rotb ← Descriptors.NumRotatableBonds(mol)
    arom ← Descriptors.NumAromaticRings(mol)
    #ring ← Descriptors.NumRings(mol)
    gi_absorption ← 'Unknown'
    if logp ≥ 0.4 and logp ≤ 5.6 and tpsa ≤ 140 then
        gi_absorption ← 'High'
    else if logp ≥ −2 and logp ≤ 6 and tpsa ≤ 180 then
        gi_absorption ← 'Moderate'
    else
        gi_absorption ← 'Low'
    end if
    bbb_permeant ← 'Unknown'
    if logp ≥ −0.4 and logp ≤ 2.5 and hbd ≤ 2 then
        bbb_permeant ← 'Yes'
    else
        bbb_permeant ← 'No'
    end if
    p_gp_substrate ← 'Unknown'
    if logp ≤ 3 and hbd ≤ 5 and hba ≤ 10 then
        p_gp_substrate ← 'Yes'
```

```
    else
        p_gp_substrate ← 'No'
    end if
    cyp1a2_inhibitor ← 'No'
    cyp2c19_inhibitor ← 'No'
    cyp2c9_inhibitor ← 'No'
    cyp2d6_inhibitor ← 'No'
    cyp3a4_inhibitor ← 'No'
    if logp ≤ 5 and tpsa ≤ 120 then
        cyp1a2_inhibitor ← 'Yes'
        cyp2c19_inhibitor ← 'Yes'
        cyp2c9_inhibitor ← 'Yes'
        cyp2d6_inhibitor ← 'Yes'
        cyp3a4_inhibitor ← 'Yes'
    end if
    lipinski ← 0
    if mw ≤ 500 and logp ≤ 5 and hba ≤ 10 and hbd ≤ 5 then
        lipinski ← 1.0
    end if
    ghose ← 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 then
        ghose ← 1.0
    end if
    veber ← 0
    if rotb ≤ 10 and tpsa ≤ 140 and logp ≤ 5 then
        veber ← 1.0
    end if
    result ← {
      '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
end function
```

Appending Properties to a DataFrame

Algorithm 1 Appending Properties to a DataFrame

```
admet_prop ← []
nan_smiles ← []
for smile in df['SMILES'] do
    res ← calculate_properties(smile)
    if res then
        res['SMILES'] ← smile
        admet_prop.append(res)
    else
        nan_smiles.append(smile)
    end if
end for
```

Extraction of Compound Information by Pubchem.Py

Algorithm 2 Extraction of Compound Information by Pubchem.Py

```
function SMILE_TO_IUPAC(smile)
    compounds ← pubchempy.get_compounds(smile, namespace='smiles')
    match ← 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 PubChemHTTPError as e
    return None
end function
Appending the properties to a DataFrame
pub_prop ← {}
for val in df['SMILES'] do
    if val ∉ pub_prop then
        prop ← smile_to_iupac(val)
        if prop then
            pub_prop[val] ← prop
        end if
    end if
end for
with open('result.json', 'w') as fp:
  json.dump(pub_prop, fp)
```