Optical property of possible candidates from CEPDB database

Shuo Dai

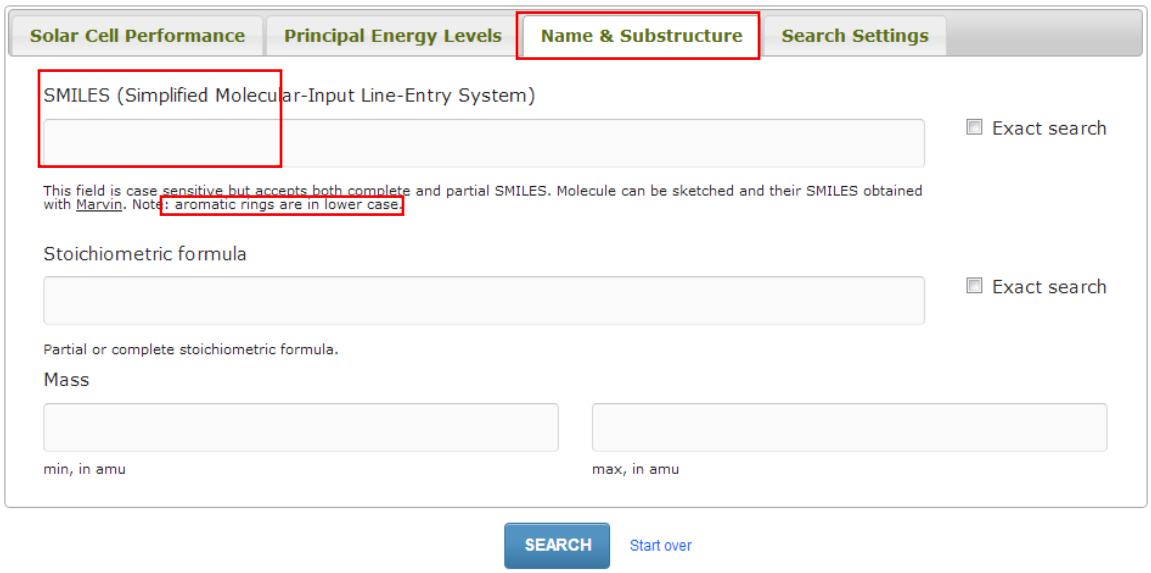
20131124

1. Database

Registration is needed to search and see the top candidates.

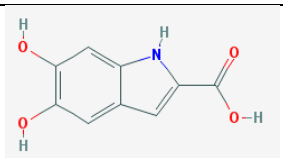
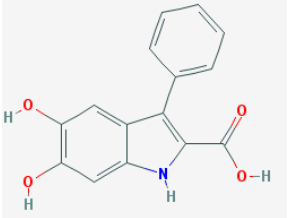
For our purpose, search can be done using the SMILE formula

In this input here, **the atoms in the rings should be in lower case**. Otherwise, the engine couldn't find it.

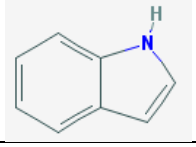


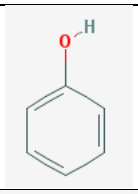
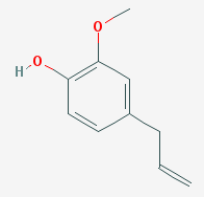
2. Search in the database

a. Search the whole molecule

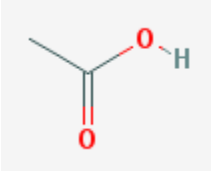
DHICA	c1=c2c=c(nc2=cc(=c1O)O)c(=O)O		None
3-phenyl-DHICA	c1=cc=c(c=c1)c2=c(nC3=cc(=c(c=C32)O)O)c(=O)O		None

b. Search the fragment

Indole	c1=cc=c2c(=c1)c=cN2		None

Phenol	c1=cc=c(c=c1)O		None
Eugenol	cOc1=c(c=cc(=c1)~~CC=C~~)O		None

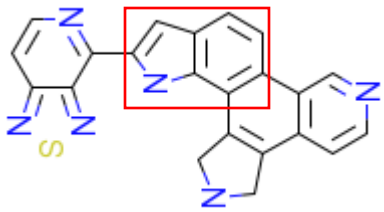
c. Search the function group

Carboxylic acid	CC(=O)O		None

d. Observation of top 1000 compound
   i. All of them have one Sulfur atom or more. Most of them have Silicon atom. Many of them have Se atom.
   ii. Nearly all of them form ring structure.

e. Identify the Indole group from the top 1000 compound. (doing it)
   i. Feature: 5-atom ring with one N, connect to 6-atom ring.
   ii. Problem:
      1. The SMILES for Indole changes in different compounds? (Counting start point different?)

f. The Abs spectra of top compounds
   i. Calculate the "c1[nH]cc2c1c1ccncc1c1ccc3cc([nH]c3c21)-c1nccc2nsnc12" (https://cepdb.molecularspace.org/single/c1%5BnH%5Dcc2c1c1ccncc1c1ccc3cc%28%5BnH%5Dc3c21%29-c1nccc2nsnc12)



Computational Results & Predictions

Principal Energy Levels

ε_{HOMO} [eV]	ε_{LUMO} [eV]	ε_{gap} [eV]
-5.41	-4.00	1.41

Photovoltaic Performance Parameters

PCE (Power Conversion efficiency [%])	V_{oc} (Open-Circuit Voltage [V])	J_{sc}(Short-Circuit Current Density [A/m^2])
11.12	0.81	210.89

3. Workflow:

Step1. Geometry opt using Gaussian09. (B3LYP/6-31++G(d,p))

Python script for this step: generate_G09_input_from_html.py:

Generate structure from the html from the website;
Initial semi-classical optimization using cclib;
Using steepest Descent to opt the structure
Save as mol_name.pdb file

Shell script: submit09:

Use the template to generate the input file

Step2. gs calculation using octopus

Python script for this step: extract_G09_output_to_xyz.py

Gaussian09 output extraction and save as *.xyz file.

Shell script for this step: submit_gs

Set up a folder called “gs” in the HPC and copy the *.inp file there.
Modify the coordinate’s part by adding:

```
XYZCoordinates = "mol_name.xyz"
```

Submit the input file

Step3. td calculation using octopus

Shell script for this step: submit_td1:

Set up 3 folders called td_x, td_y and td_z. Copy all the content from the gs folder to td_x, td_y and td_z, respectively. Then in their td inputs, change the direction to be x, y and z, respectively. This can be done by using:

```
TDPolarizationDirection = 1
```

Step4. Calculation of cross section/absorption

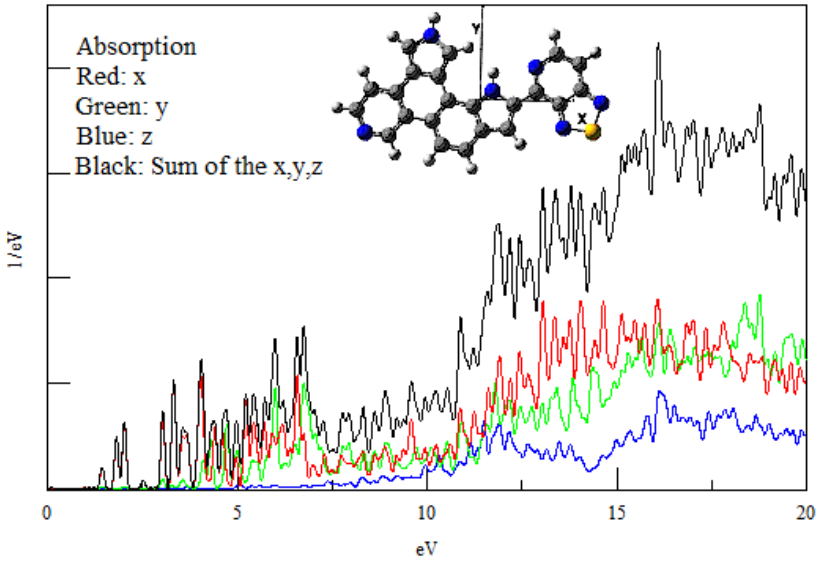
Shell script for this step:

Copy the multiple in the 3 folder and rename them into multiple.1, multiple.2 and multiple.3. Upload them to the ps057-vaquero. Use utility :

```
/state/partition1/apps/bin/oct-propagation_spectrum
```

Step5. Plot the absorption spectra:

X: first column in the cross_section_vector;
Y: 5-th column in the cross_section_vector;
Sum of 3 direction: add x, y and z together.



41 atoms	G09 optimization	Octopus gs	Octopus td_x	Octopus td_y	Octopus td_z
Time used	186m	54m	3201m	2465m	2577m

4. Template for the input file:

    a. for step 3, octopus (ground state part)

```
Units = eV_angstrom
CalculationMode = gs
Theorylevel = DFT

Spacing = 0.20
BoxShape = minimum
# The radius can be set based on the recommendation in the pseudo potential  setting it
at 5 overrides that
# Radius = 5.00

XCFunctional = hyb_gga_xc_b3lyp

fromScratch = yes
# ------------------
# TD RUN Parameters
# ------------------
TDDeltaStrength = 0.01
TDPolarizationDirection = 1

# 1 fs = 1.519 time unit
# 1 time unit = 0.658 fs

# The time needed might not be as large. Especially if you are to use the compressed
sensing

# 25fs
stime = 38.0

dt = 0.0025
TDPropagator = aetrs
TDExponentialMethod = taylor
TDExponentialOrder = 4
TDLanczosTol = 5.0e-5
TDMaximumIter = stime/dt
TDTimeStep = dt
OutputEvery = 500

# --------------------
# Spectrum Parameters
# --------------------

# ExperimentalFeatures = yes
# SpectrumMethod = compressed_sensing
# SpectrumSignalNoise = 0.0001

# Change this number, if we want smaller perturbation time
# PropagationSpectrumEndTime = 3.7975
```

b. for step 3, octopus (TD-DFT part: x-direction)

x-direction: add `TDPolarizationDirection = 1`

y-direction: add `TDPolarizationDirection = 2`

z-direction: add `TDPolarizationDirection = 3`

```
Units = eV_angstrom
CalculationMode = td
Theorylevel = DFT

Spacing = 0.20
BoxShape = minimum
# The radius can be set based on the recommendation in the pseudo potential  setting it at 5 overrides that
# Radius = 5.00


XCFunctional = hyb_gga_xc_b3lyp

fromScratch = yes
# ------------------
# TD RUN Parameters
# ------------------
TDDeltaStrength = 0.01


# 1 fs = 1.519 time unit
# 1 time unit = 0.658 fs

# The time needed might not be as large. Especially if you are to use the compressed sensing

# 25fs
stime = 38.0

dt = 0.0025
TDPropagator = aetrs
TDExponentialMethod = taylor
TDExponentialOrder = 4
TDLanczosTol = 5.0e-5
TDMaximumIter = stime/dt
TDTimeStep = dt
OutputEvery = 500

# --------------------
# Spectrum Parameters
# --------------------

# ExperimentalFeatures = yes
# SpectrumMethod = compressed_sensing
# SpectrumSignalNoise = 0.0001

# Change this number, if we want smaller perturbation time
# PropagationSpectrumEndTime = 3.7975
```