

Représentation de molécules

Page à actualiser...

Certaines fonctions de ce programme nécessite des fichiers de données : [base.csv](#) et [bdd.csv](#) <sxh python; title : representation_molecules.py> `#!/usr/bin/env python # -*- coding: UTF-8 -*- # travail de RL, ba2 chimie 2012-2013 'Fonctions utilisées ' def stripTags(s):`

```
''' Strips HTML tags.
    Taken from
    http://aspn.activestate.com/ASPN/Cookbook/Python/Recipe/440481
    Sert à nettoyer une page HTML de son code pour ne garder que le texte
    ...
    intag = [False]

    def chk(c):
        if intag[0]:
            intag[0] = (c != '>')
            return False
        elif c == '<':
            intag[0] = True
            return False
        return True

    return ''.join(c for c in s if chk(c))
```

`def find_words(text, search):`

```
''' Sert à vérifier la présence d'un mot dans une chaîne.
    Trouvé sur :
    http://stackoverflow.com/questions/5319922/python-check-if-word-is-in-a-string
    ...
    dText = {}
    dSearch = {}
```

```
dText = text.split()
dSearch = search.split()
```

```
lenText = len(dText)
lenSearch = len(dSearch)
#print dText, lenText
#print dSearch, lenSearch
```

```
found_word = 0
```

```
for text_word in dText:
    for search_word in dSearch:
        if hash(search_word) == hash(text_word):
```

```
found_word += 1
```

```
if found_word == lenSearch:
    return lenSearch
else:
    return False
```

```
def DLink(SMILES):
```

```
'''Crée un lien pour vers une visualisation en 3D de la molécule
...
link="http://chemapps.stolaf.edu/jmol/jmol.php?model="+SMILES
return link
```

```
def SMILES(name):
```

```
'''Donne le code SMILES d'une molécule à partir de la page Wikipédia
...
import urllib, urllib2
namemod=name.replace(' ', '_') #Edition du nom au cas où il contient un
espace
link='http://fr.wikipedia.org/wiki/'+namemod #Création du lien de la
page de la molécule
opener=urllib2.build_opener() #Obtention de la page Wikipedia de la
molécule
opener.addheaders=[('User-agent', 'Mozilla/5.0')] #On se fait passer pour
un vrai navigateur
buffer=opener.open(link) #On charge la page
htmlSource=buffer.read() #Et on la stock
buffer.close()
htmlClean=stripTags(htmlSource) #On nettoie le code HTML de ses balises
#SMILES
verif=find_words(htmlClean, 'SMILES') #On Vérifie si la page Wikipédia
possède un code SMILES
if verif is False: #Si non, message d'erreur
    return 'La page Wikipédia de cette molécule ne fournit pas de code
SMILES.'
else: #Si oui, on le localise et on le renvoie
    htmlList=htmlClean.split()
    SMILESValue=htmlList.index('SMILES')
    return htmlList[SMILESValue+1]
```

```
def Draw2D(name, smiles):
```

```
''' Trouvé sur http://ctr.wikia.com/wiki/Depict\_a\_compound\_as\_an\_image
sert à dessiner une molécule en 2D à partir de son code SMILE
...
file_name=name+'.png'
from rdkit.Chem import AllChem
```

```
from rdkit.Chem import Draw
mol = AllChem.MolFromSmiles(smiles)
AllChem.Compute2DCoords(mol)
Draw.MolToFile(mol, file_name, size=(200,250))
import Image
im = Image.open(name+'.png')
im.save(name+'.gif')
import os
os.remove(name+'.png')
```

```
def AddToClipboard(text):
```

```
    '''Copier un texte dans le presse-papiers
    ...
    text = str(text)
    r = Tk()
    r.withdraw()
    r.clipboard_clear()
    r.clipboard_append(text)
    r.destroy()
```

```
def FindName(smiles):
```

```
    '''Trouve le nom correspondant à un code SMILES dans une base de données
    CSV
    ...
    import csv
    c=open('bdd.csv', 'rb')
    reader=csv.reader(c)
    for row in reader:
        if row[1] == smiles: #On vérifie la seconde rangée de chaque ligne de
la base de données jusqu'à trouver un SMILES identique
            return row[0] #Et on retourne le nom correspondant si on le trouve
    else:
        return "Nom introuvable dans la base de données."
    c.close()
```

```
'Interface ' from Tkinter import * root=Tk() #On crée la fenêtre root.title('SMILES')
root.geometry("400x500")
```

```
#Boutons onglets def ActionOnglet1():
```

```
    FrameOnglet2.grid_forget()
    FrameOnglet3.grid_forget()
    FrameOnglet1.grid(row=1, columnspan=3)
```

```
Onglet1=Button(root, text="A partir du nom", command=ActionOnglet1) Onglet1.grid(row=0,
column=0) def ActionOnglet2():
```

```
    FrameOnglet1.grid_forget()
    FrameOnglet3.grid_forget()
```

```
FrameOnglet2.grid(row=1, colspan=3)
```

```
Onglet2=Button(root, text="A partir du SMILES", command=ActionOnglet2) Onglet2.grid(row=0, column=1) def ActionOnglet3():
```

```
FrameOnglet1.grid_forget()
FrameOnglet2.grid_forget()
FrameOnglet3.grid(row=1, colspan=3)
```

```
Onglet3=Button(root, text="A partir d'un CSV", command=ActionOnglet3) Onglet3.grid(row=0, column=2) #Frames FrameOnglet1=Frame(root) FrameOnglet1.grid(row=1, colspan=3)
FrameOnglet2=Frame(root) FrameOnglet3=Frame(root)
```

```
##Onglet 1 (nom) #Nom NomLabel=Label(FrameOnglet1, text="Quel est le nom de la molécule ?").pack() NomEntry=Entry(FrameOnglet1) NomEntry.pack() #Menubuttons ClipboardCase=IntVar() Clipboard=Checkbutton(FrameOnglet1, text="Copier le code SMILES dans le presse-papiers", variable=ClipboardCase).pack() NavigateurCase=IntVar() Navigateur=Checkbutton(FrameOnglet1, text="Afficher une vue 3D dans le navigateur", variable=NavigateurCase).pack() ImageCase=IntVar() Image=Checkbutton(FrameOnglet1, text="Créer une représentation 2D de la molécule", variable=ImageCase).pack() #Bouton valider nom def action():
```

```
name=str(NomEntry.get())
if ClipboardCase.get() == 1:
    AddToClipboard(SMILES(name))
if NavigateurCase.get() == 1:
    import webbrowser
    new=2
    webbrowser.open(DLink(SMILES(name)), new=new)
if ImageCase.get() == 1:
    Draw2D(name, SMILES(name))
    global photo
    photo = PhotoImage(file =name+".gif")
    DisplayImageZone.create_image(100, 100, image=photo)
DisplaySmiles.delete(1.0, END)
DisplaySmiles.insert(END, SMILES(name))
```

```
confirm=Button(FrameOnglet1, text="Valider", command=action).pack() #Affichage SMILES
AffichageLabel=Label(FrameOnglet1, text="Code SMILES :").pack()
DisplaySmiles=Text(FrameOnglet1, height=2, width=30) DisplaySmiles.pack() #Canvas
ImageLabel=Label(FrameOnglet1, text="Représentation 2D :").pack()
DisplayImageZone=Canvas(FrameOnglet1, width =200, height =250, bg ='white')
DisplayImageZone.pack()
```

```
##Onglet2 (SMILES) #SMILES SmilesLabel=Label(FrameOnglet2, text="Quel est le code SMILES de la molécule ?").pack() SmilesText=Entry(FrameOnglet2) SmilesText.pack() #Menubuttons
ClipboardCase2=IntVar() Clipboard2=Checkbutton(FrameOnglet2, text="Copier le nom dans le presse-papiers", variable=ClipboardCase2).pack() NavigateurCase2=IntVar()
Navigateur2=Checkbutton(FrameOnglet2, text="Afficher une vue 3D dans le navigateur", variable=NavigateurCase2).pack() ImageCase2=IntVar() Image2=Checkbutton(FrameOnglet2,
```

text="Créer une représentation 2D de la molécule", variable=ImageCase2).pack() #Bouton valider
code def action2():

```
CodeSMILES=str(SmilesText.get())
if ClipboardCase2.get() == 1:
    AddToClipboard(FindName(CodeSMILES))
if NavigateurCase2.get() == 1:
    import webbrowser
    new=2
    webbrowser.open(DLink(CodeSMILES), new=new)
if ImageCase2.get() == 1:
    Draw2D("Molécule", CodeSMILES)
    global photo
    photo = PhotoImage(file ="Molécule.gif")
    DisplayImageZone2.create_image(100, 100, image=photo)
DisplayName.delete(1.0, END)
DisplayName.insert(END, FindName(CodeSMILES))
```

confirm2=Button(FrameOnglet2, text="Valider", command=action2).pack() #Affichage nom
AffichageLabel2=Label(FrameOnglet2, text="Nom :").pack() DisplayName=Text(FrameOnglet2,
height=2, width=30) DisplayName.pack() #Canvas ImageLabel2=Label(FrameOnglet2,
text="Représentation 2D :").pack() DisplayImageZone2=Canvas(FrameOnglet2, width =200, height
=250, bg ='white') DisplayImageZone2.pack()

Onglet3 (CSV) Consignes=Label(FrameOnglet3, text="Veuillez placer le fichier CSV dans le même
répertoire \net le nommer base.csv. Les noms/SMILES doivent \nse trouver dans la première
colonne.\n\nLe fichier complété se trouvera dans le même répertoire\n et portera le nom
base_done.csv.\n").pack() DeleteCase=IntVar() Delete=Checkbutton(FrameOnglet3, text="Supprimer
le fichier original une fois la tâche accomplie", variable=DeleteCase).pack() def CSVName():

```
import csv
FileRead=open("base.csv", "rb")
cr=csv.reader(FileRead)
FileWrite=open("base_done.csv", "wb")
cw=csv.writer(FileWrite, delimiter=',', quotechar='')
for row in cr:
    cw.writerow([row[0], SMILES(row[0])])
FileRead.close()
FileWrite.close()
if DeleteCase.get()==1:
    import os
    os.remove("base.csv")
```

FromName=Button(FrameOnglet3, text="Trouver les SMILES à partir des noms",
command=CSVName).pack() def CSVSmiles():

```
import csv
FileRead=open("base.csv", "rb")
cr=csv.reader(FileRead)
FileWrite=open("base_done.csv", "wb")
cw=csv.writer(FileWrite, delimiter=',', quotechar='')
```

```
for row in cr:
    cw.writerow([FindName(row[0]), row[0]])
FileRead.close()
FileWrite.close()
if DeleteCase.get()==1:
    import os
    os.remove("base.csv")
```

```
FromSmiles=Button(FrameOnglet3, text="Trouver les noms à partir des SMILES",
command=CSVSmiles).pack()
```

```
root.mainloop() </sxh>
```

From:
<https://dvillers.umons.ac.be/wiki/> - **Didier Villers, UMONS - wiki**

Permanent link:
https://dvillers.umons.ac.be/wiki/teaching:progappchim:representation_molecules_2013

Last update: **2020/03/09 14:42**

