Tutorial for installing and running MSIonization package

Installing the MSIonization package:

1) Download the MSIonization package from the following link:
   Link:
2) Download and install the latest version of python for macOS or windows operating system from this link: https://www.python.org/
3) Check the python version by typing this command in terminal:
   macOS/Linux: python3 –version
   windows: py –version
4) For installing the MSIonization package, pip package must be installed on your computer. The user can check it by the following commands:
   Windows: py -m pip –version
   Linux/macOS: python3 -m pip –version
5) If pip has not been installed, the user can install it by using:
   Windows: py -m ensurepip --default-pip
   Linux/macOS: python3 -m ensurepip –default-pip
6) If it still has not been installed, the user can download get-pip.py script from this link: https://bootstrap.pypa.io/get-pip.py
   Go to the downloaded folder (directory) in cmd (in windows) or terminal (in linux or macOS) and then run this command: python get-pip.py
7) The user can navigate to the folder (or directory) containing the MSIonization package in cmd (in windows) or terminal (in linux or macOS) and install the package using this command:
   macOS/Linux: pip install msionization-1.0-py3-none-any.whl
   Windows: py -m pip install msionization-1.0-py3-none-any.whl
8) The computer must have an internet connection to install the required packages (numpy, scikit-learn, rdkit, joblib and pandas) for using the MSIonization package. These required packages will be installed automatically.

Checking MSIonization installation:

1) Open the terminal in macOS/Linux or cmd in windows and run the appropriate command:
   macOS/Linux: pip show MSIonization
   Windows: py -m pip show MSIonization

Running the MSIonization package using spyder IDE

1) Install spyder IDE using:
   macOS/Linux: pip install spyder (internet connection is required).
   Windows: py -m pip install spyder (internet connection is required).
2) Load spyder IDE by running this command in terminal in macOS/Linux or cmd in windows: spyder
3) For prediction of a single molecule, e.g., one with the SMILES string = "CC(C(C1=CC=CC=C1)Cl)NC", run the following code in spyder:

```
################################################################
from MSIonization import predict_ionization
```

```
smiles = "CC(C(C1=CC=CC=C1)Cl)NC"  # Example SMILES notation
result = predict_ionization(smiles)
print(result)
##############################################################
```

In this code, the user can change the SMILES notation to whatever is needed.

4) To predict the class for several molecules at once, the user can use the following code, in which an example list of molecules' SMILES strings is shown:

```
##############################################################
from MSIonization import predict_ionization
# List of multiple SMILES strings
smiles_list = ["CC(C(C1=CC=CC=C1)Cl)NC",
"CC12CCC3C(C1CCC2OC4C(C(C(C(O4)C(=O)O)O)O)O)CCC5=CC(=O)CCC35",
"C1CN1P(=O)(N2CC2)N3CC3",
"CC(COCC(C)OC(=O)C1=CC=CC=C1)OC(=O)C2=CC=CC=C2"]
# Iterate through each SMILES string
for smiles in smiles_list:
    result = predict_ionization(smiles)
    print(f"SMILES: {smiles} -> Ionization: {result}")
##############################################################
```

5) If the user wants to read SMILES strings from a CSV file and import them to spyder IDE, the user can use this code:

```
##############################################################
import pandas as pd
from MSIonization import predict_ionization

# Read SMILES strings from a CSV file
data = pd.read_csv("SMILES.csv")

# Ensure column name consistency
if 'SMILES' not in data.columns:
    raise ValueError("CSV file must contain a 'SMILES' column.")

# Iterate through each SMILES string and predict ionization
for smiles in data['SMILES'].dropna():
    try:
        result = predict_ionization(smiles)
        print(f"SMILES: {smiles} -> Ionization: {result}")
    except Exception as e:
        print(f"Error processing SMILES '{smiles}': {e}")
##############################################################
```

The user must place the CSV file in the working directory of spyder IDE. All three of these codes can be found in the 'MSIonization_commands.py' file.

Running the MSIonization package using MSIonization_script.py file

1) To make predictions for a list of SMILES strings in a CSV file and save the results in another CSV file, we have provided a separate script called 'MSIonization_script.py'. The user can put both the script and the CSV file in the same directory. Then, open the terminal (macOS/Linux) or cmd (in windows) in the same directory and run the following command:

   ```
   python3  Msionization_script.py
   ```

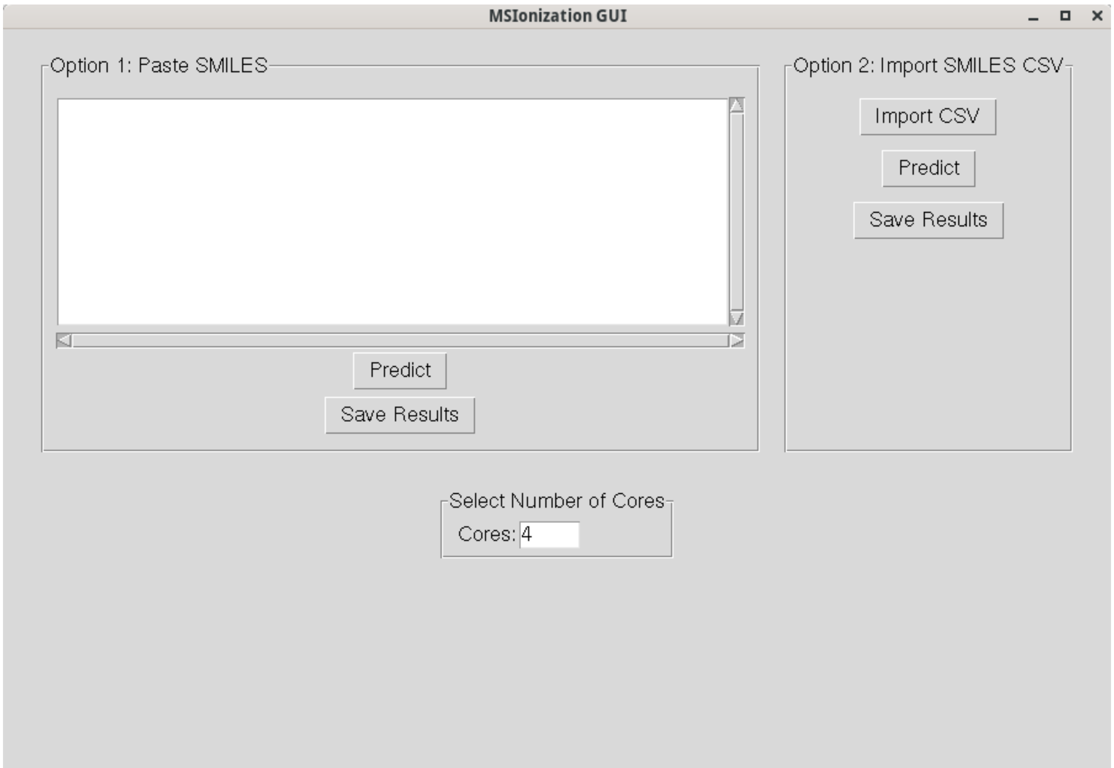
   A message will appear that ask the user to select the number of cores for running the calculations.

Running the MSIonization package using GUI

1) For loading the designed GUI, the user needs to install the tkinter and setuptools packages:
   To install setuptools:
   macOS/Linux: `pip install setuptools`
   windows: `py -m pip install setuptools`
   To install tkinter in different operating systems:
   Rocky Linux (RHEL-based): `sudo dnf install python3-tkinter`
   Ubuntu Linux (Debian-based): `sudo apt install python3-tk`
   macOS: `brew install python-tk`
   Windows: Tkinter is usually included with the standard Python installation.
2) Install the MSIonization package as 'Installing the MSIonization package' section.
3) Load GUI on macOS/Linux by typing this command and pressing the enter key: `MSIonization-GUI`
4) To load GUI in windows, navigate to the installation directory in cmd by this command: cd Installation directory
   Example: `cd C:\Users\lab\AppData\Local\Programs\Python\Python313\Scripts\`
   Then load GUI by this command in cmd: `MSIonization-GUI.exe`
5) This GUI consists of two panels for importing molecules, running predictions and finally saving results. In the first panel (Option 1: Paste SMILES), there is a place to paste a list of SMILES notations for any number of molecules. After providing the SMILES notation of molecules, the user needs to click 'Predict' button and wait until all calculations are completed. During this step, a message in blue will appear that shows calculations are running. After completing the prediction step, a message in green will appear showing that the prediction step has been completed. Then the results could be saved in a CSV file using the 'Save Results ' button.

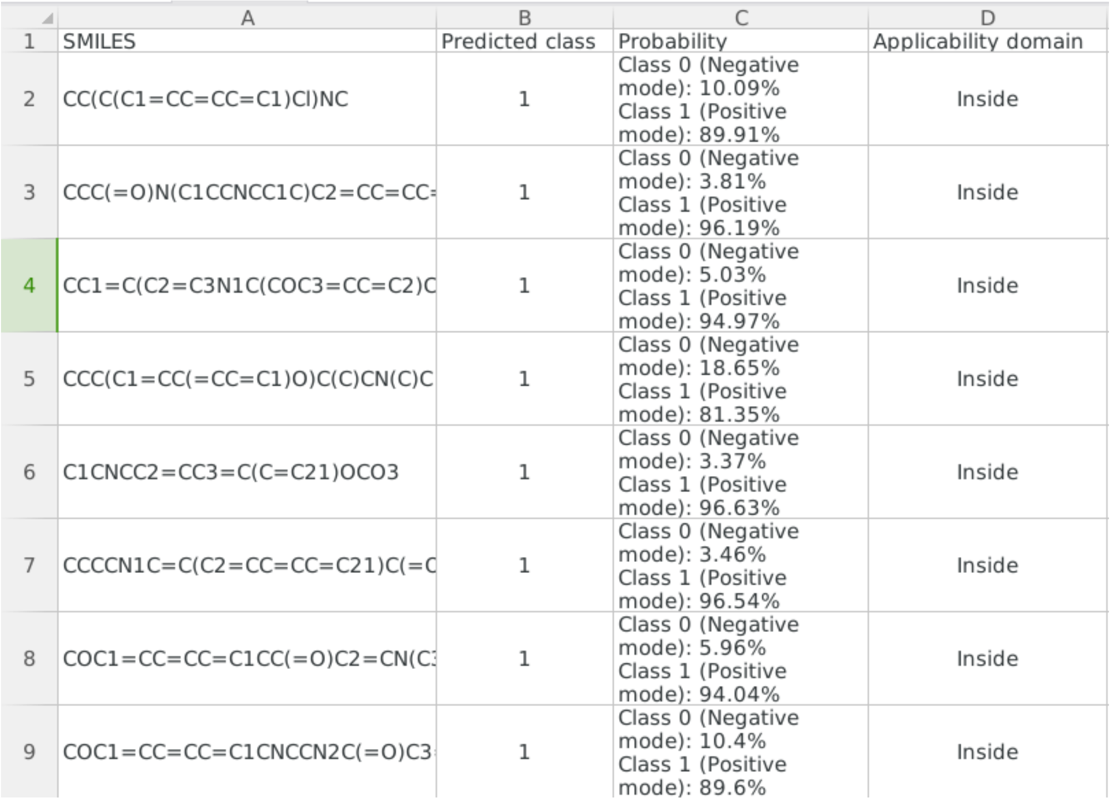
   The second panel is the 'Option 2: Import SMILES CSV' panel. In this panel, the user can import a list of SMILES notations in a CSV file named 'SMILES.csv' that contains one column with header 'SMILES'. To do this, the user can click on 'Import CSV' to select and import the file. The user can then run the prediction using the 'Predict' button and save the results using the 'Save Results' button.

   The following figure shows the designed GUI for MSIonization package.



Cores: 4

The CSV output file (Predicted_Ionization_efficiencies.csv) contains four columns including SMILES, Predicted class, Probability, and Applicability domain. The following figure shows its content:



	A	B	C	D
1	SMILES	Predicted class	Probability	Applicability domain
2	CC(C(C1=CC=CC=C1)Cl)NC	1	Class 0 (Negative mode): 10.09% Class 1 (Positive mode): 89.91%	Inside
3	CCC(=O)N(C1CCNCC1C)C2=CC=CC	1	Class 0 (Negative mode): 3.81% Class 1 (Positive mode): 96.19%	Inside
4	CC1=C(C2=C3N1C(COC3=CC=C2)C	1	Class 0 (Negative mode): 5.03% Class 1 (Positive mode): 94.97%	Inside
5	CCC(C1=CC(=CC=C1)O)C(C)CN(C)C	1	Class 0 (Negative mode): 18.65% Class 1 (Positive mode): 81.35%	Inside
6	C1CNCC2=CC3=C(C=C21)OCO3	1	Class 0 (Negative mode): 3.37% Class 1 (Positive mode): 96.63%	Inside
7	CCCCN1C=C(C2=CC=CC=C21)C(=C	1	Class 0 (Negative mode): 3.46% Class 1 (Positive mode): 96.54%	Inside
8	COC1=CC=CC=C1CC(=O)C2=CN(C	1	Class 0 (Negative mode): 5.96% Class 1 (Positive mode): 94.04%	Inside
9	COC1=CC=CC=C1CNCCN2C(=O)C3	1	Class 0 (Negative mode): 10.4% Class 1 (Positive mode): 89.6%	Inside

Installing the MSIonization package using Anaconda

1) Anaconda is a popular distribution of Python (and R) that's designed specifically for data science, machine learning, scientific computing, and large-scale data processing. It can be downloaded from: https://www.anaconda.com/download/success
2) This is a very good option when you have the older version of python and are not able to update it. Anaconda let you to have access to a newer versions of python.
3) Anaconda installation guide can be found here: https://www.anaconda.com/docs/getting-started/anaconda/install#macos-linux-installation
4) If the user install anaconda, he/she have access to the pip package (windows: anaconda prompt; macOS/Linux: Terminal) and spyder IDE (in Anaconda Navigator). On macOS or Linux, you can launch the Spyder IDE by typing spyder in the terminal and pressing the Enter key.
5) To install the MSIonization package, in macOS/Linux, open a terminal, and in windows open anaconda prompt in the folder or the directory containing the MSIonization package and install it using this command: pip install msionization-1.0-py3-none-any.whl

   Internet connection is needed to install the required packages.
6) To load the MSIonization package, in macOS/Linux terminal or in anaconda prompt (in windows), type this command and then press enter key: MSIonization-GUI

Key note: To open cmd in a directory or a folder in windows, follow this instruction:

1) Navigate to the desired folder in File Explorer.
2) Click on the address bar (or press Alt + D) to highlight it.
3) Type cmd and press Enter.