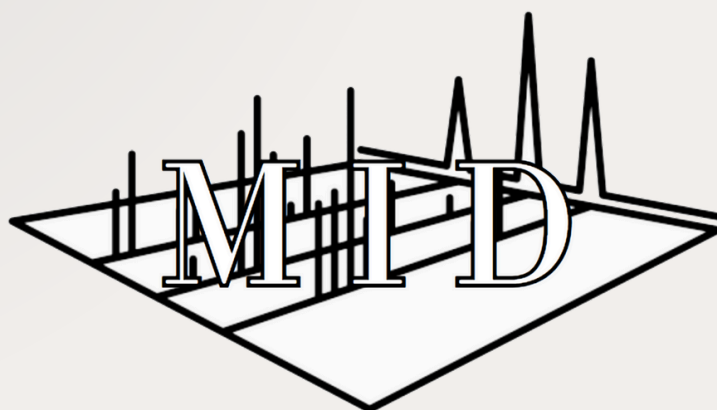


Molecule_ID based MSMS
fingerprint Similarity



MANUEL
D'UTILISATION

Fayçal Hassani

SOMMAIRE

01. OBJECTIF D'UTILISATION

02. PRÉREQUIS

08. UTILISATION DU LOGICIEL

Objectif d'utilisation

Le logiciel Molecule_ID, basé sur la similarité des empreintes MSMS, a pour objectif principal de calculer la similarité entre de nouveaux spectres MS2 et une base de référence préenregistrée.

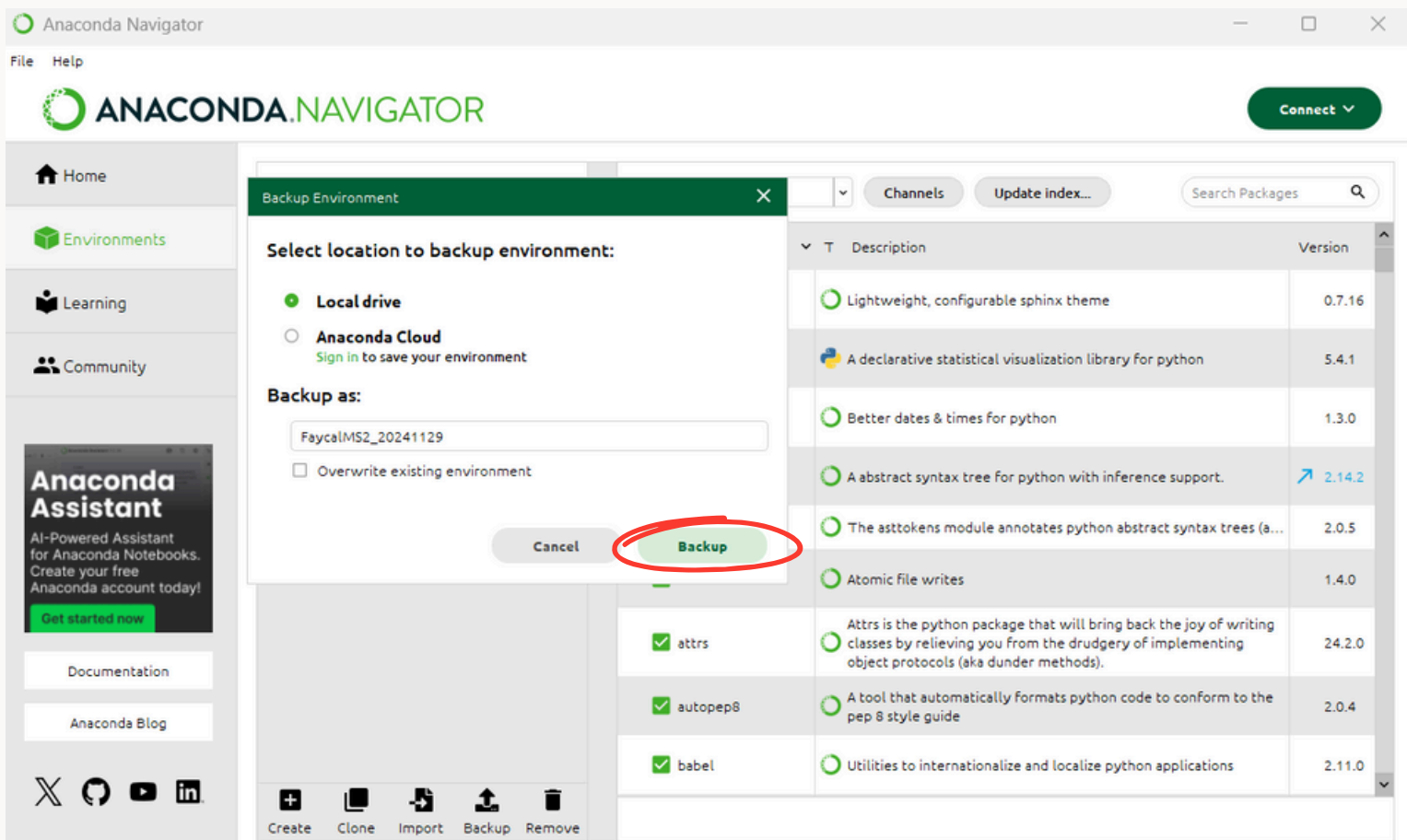
En comparant ces spectres, le logiciel détermine le spectre le plus similaire parmi ceux de la base de données.

En outre, Molecule_ID permet également d'afficher le nouveau spectre importé, offrant ainsi une visualisation claire et précise des données spectrales analysées.

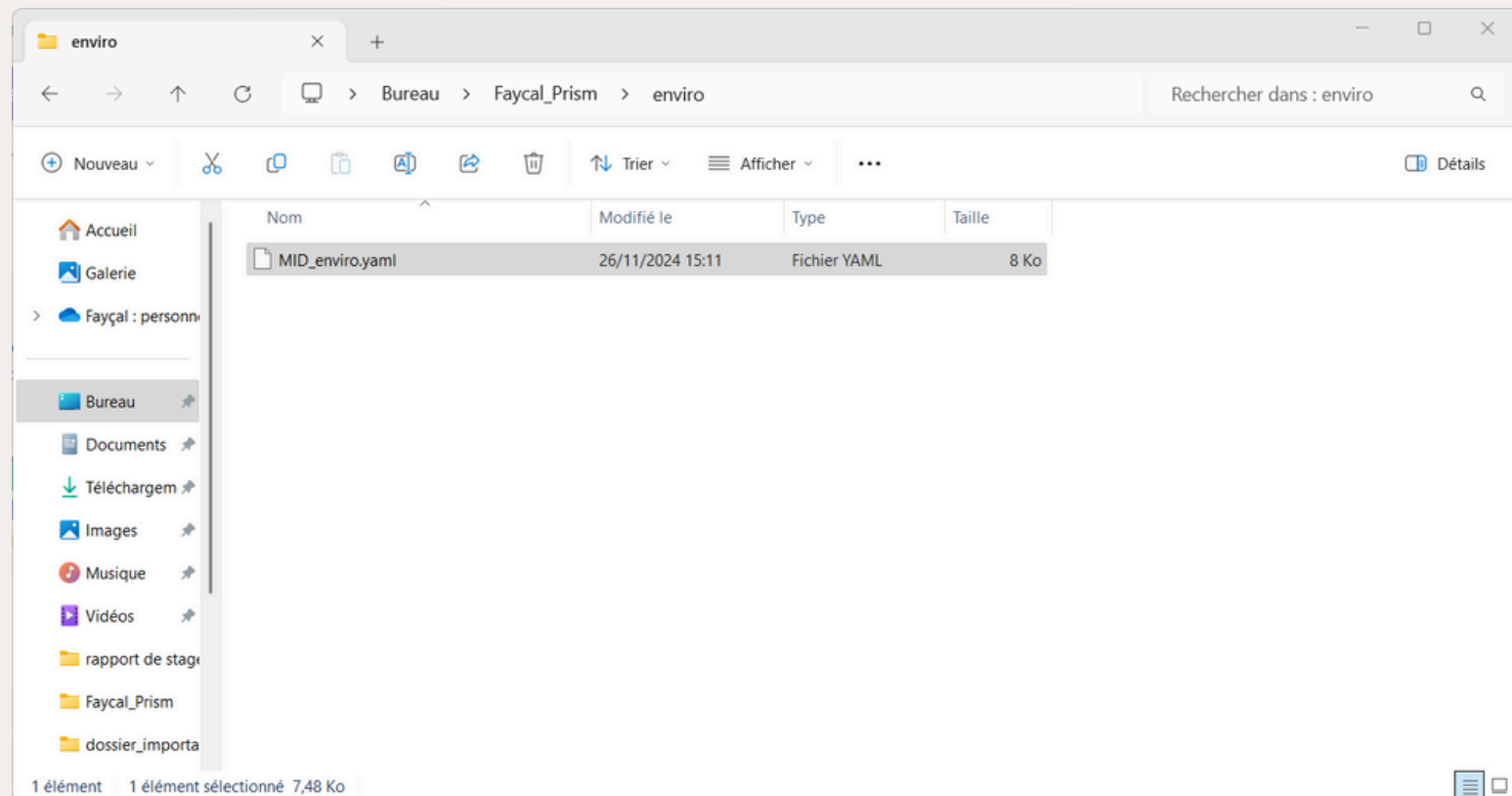
Prérequis

1. Importation de l'environnement Anaconda :

- Ouvrez Anaconda Navigator.
- Dans le menu principal, cliquez sur "Environments" pour accéder à la gestion des environnements.
- Ensuite, allez dans "Backup Local Drive" pour accéder aux sauvegardes locales.



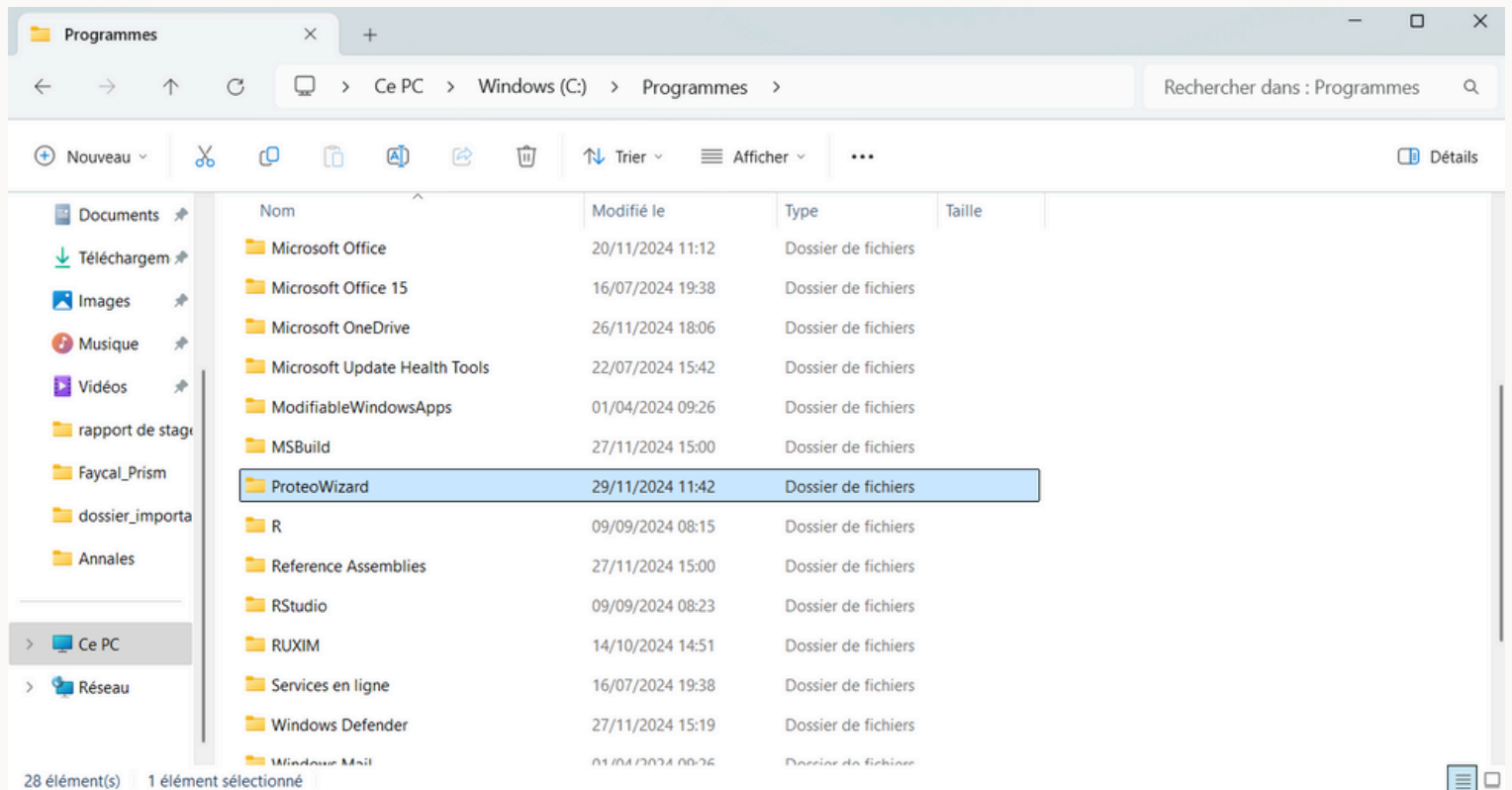
- Recherchez le dossier nommé "enviro" qui contient les environnements sauvegardés.
- Dans ce dossier, trouvez le fichier "MID_enviro.yaml" et importez-le en suivant les instructions à l'écran.



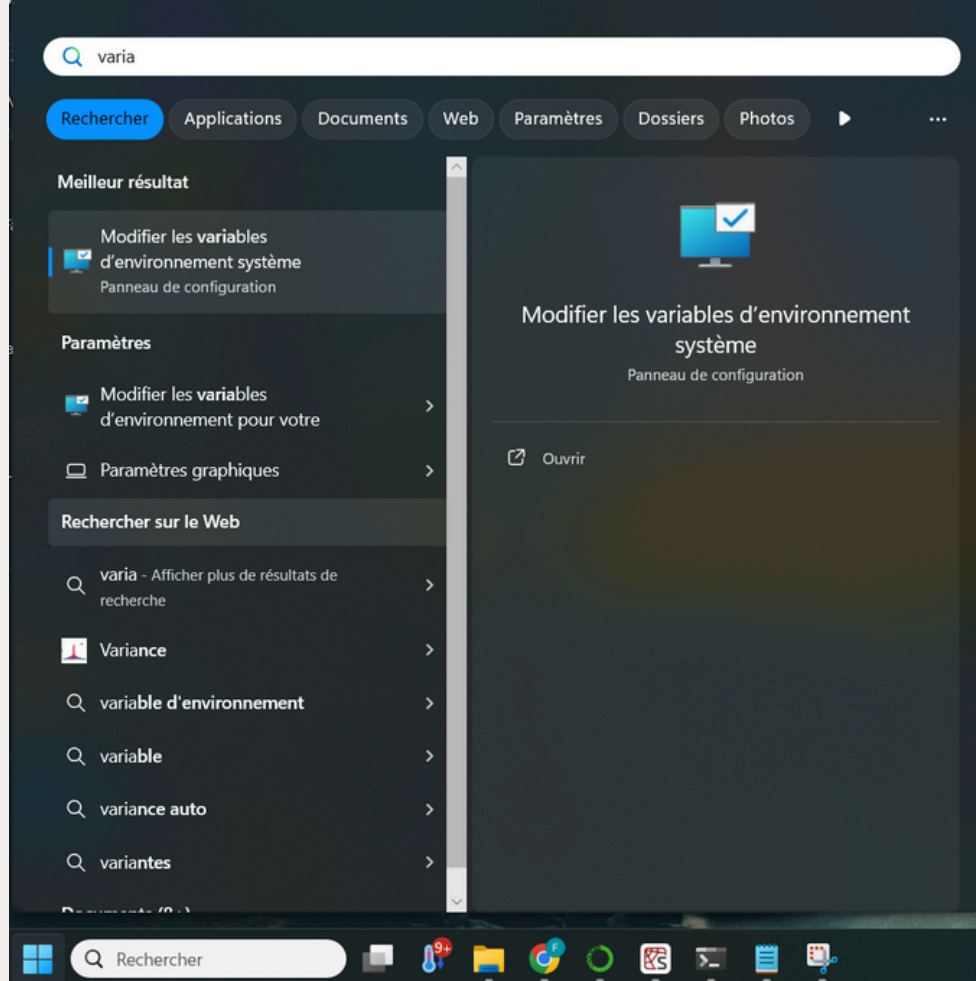
- Une fois l'importation terminée, l'environnement sera prêt à être utilisé et exécuté.

2.Installation de MSconvert :

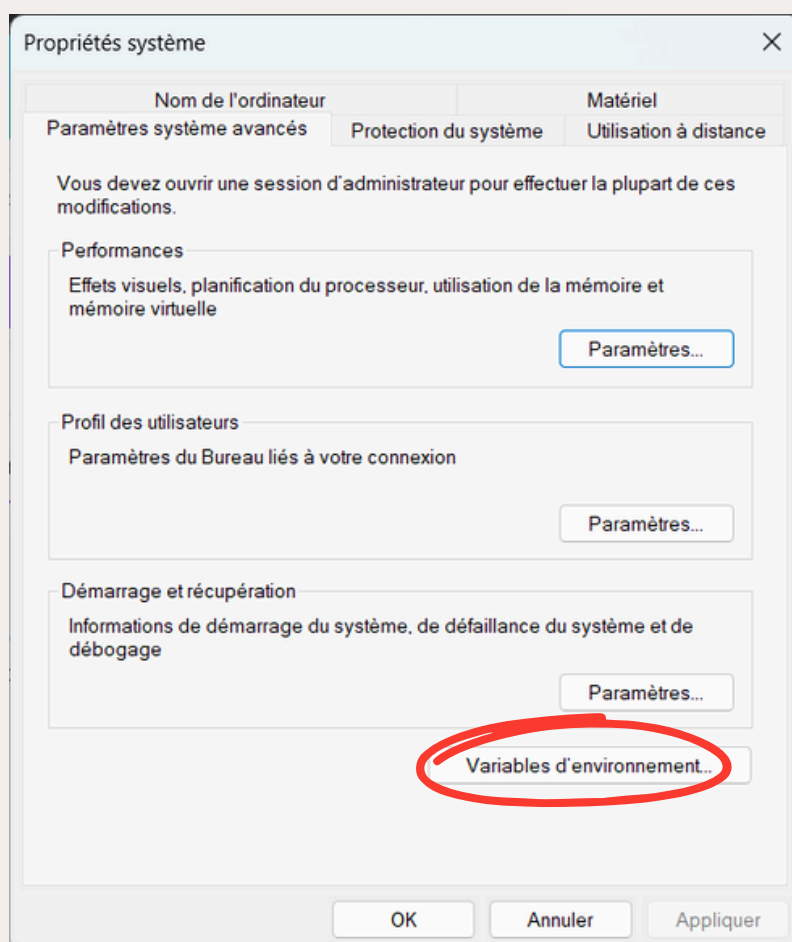
- Dans le dossier fourni, localisez le dossier "ProteoWizard".
- Copiez ce dossier et collez-le dans le répertoire "Programmes" de votre disque dur (généralement situé sous "C:\Program Files" selon votre système).



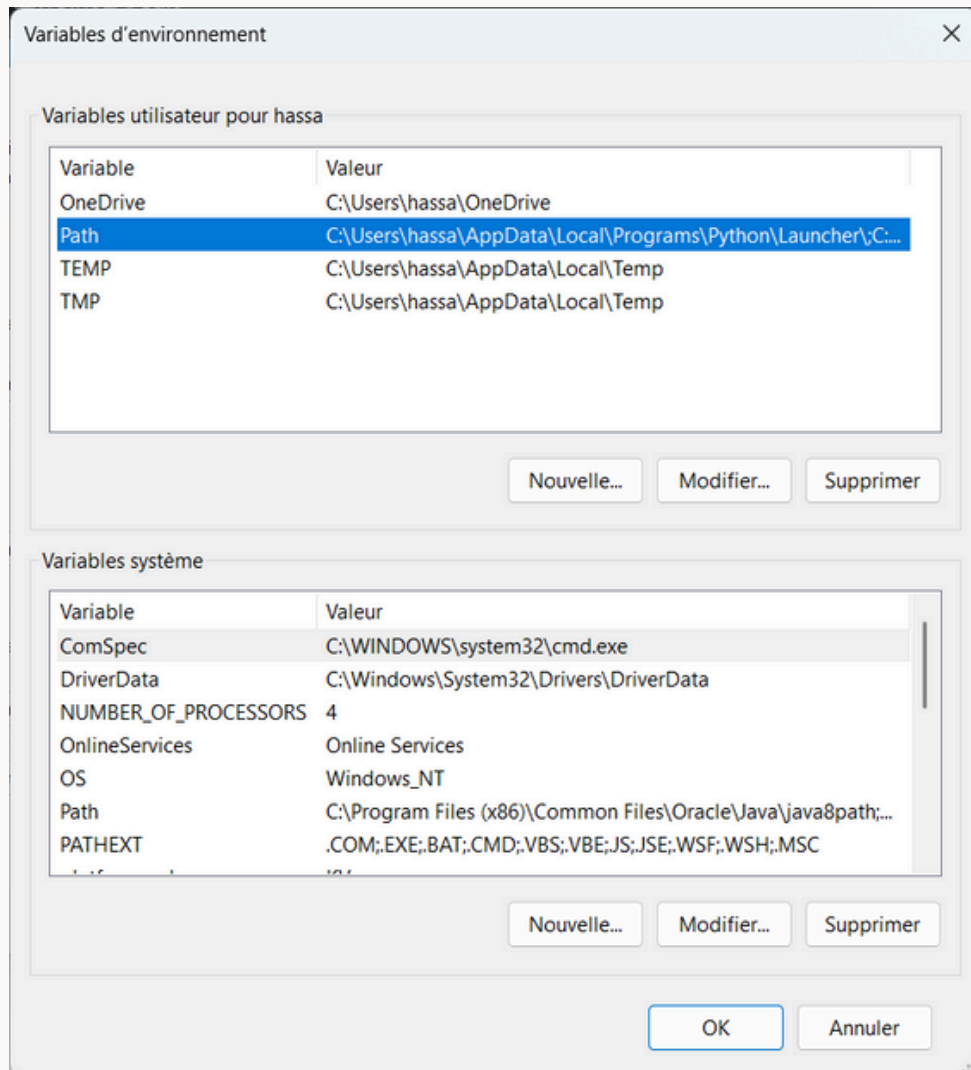
- Ensuite, ouvrez le menu Démarrer et recherchez "Modifier les variables d'environnement système".



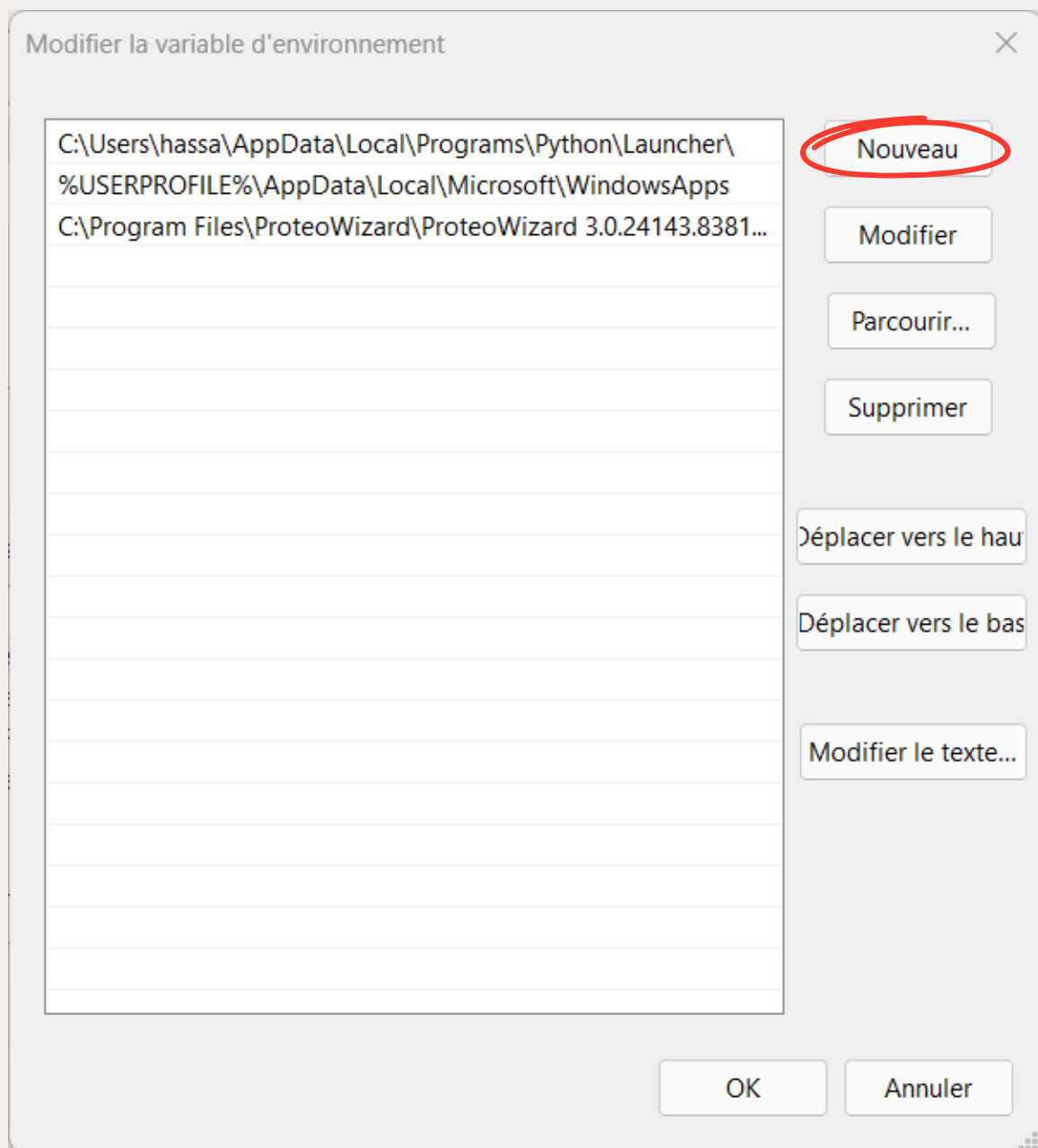
- Cliquez sur "Variables d'environnement" pour ouvrir la fenêtre de configuration des variables d'environnement.



- Dans la section "Variables système", trouvez la variable "Path" et sélectionnez-la, puis cliquez sur "Modifier".



- Dans la nouvelle fenêtre, cliquez sur "Nouveau" et collez le chemin complet du dossier "ProteoWizard 3.0.24143.8381ed5 64-bit" (par exemple, "C:\Program Files\ProteoWizard 3.0.24143.8381ed5 64-bit").



- Validez les modifications et fermez les fenêtres de configuration.

- Ou vous pouvez aller dans le script et donner le chemin du fichier \msconvert. (c'est a vous de choisir ce que vous préfère réaliser mais la 1er méthode évite de redonner le chemin a chaque fois)

```

4 import pandas as pd
5 import plotly.graph_objects as go
6 from scipy.signal import find_peaks
7 from sklearn.metrics.pairwise import cosine_similarity
8 from scipy.stats import pearsonr, spearmanr
9 from scipy.sparse import csr_matrix
10 import streamlit as st
11 import subprocess
12 from joblib import Parallel, delayed
13 import time
14
15 # Function to generate output mzML path from RAW file path
16 def generate_mzml_path(raw_file_path):
17     base, _ = os.path.splitext(raw_file_path)
18     return base + ".mzML"
19
20 # Function to convert RAW to mzML
21 def convert_raw_to_mzml(raw_file):
22     mzml_file = generate_mzml_path(raw_file)
23     subprocess.run(["msconvert", raw_file, "-o", os.path.dirname(mzml_file), "-mzML"])
24     return mzml_file
25
26 # Function to analyze an mzML file
27 def analyze_mzml_file(mzml_file):
28     exp = pyopenms.MSEExperiment()
29     pyopenms.MzMLFile().load(mzml_file, exp)
30
31     chromatogram = exp.getChromatograms()[0]
32     chrom_times, chrom_intensities = chromatogram.get_peaks()
33
34     peaks, _ = find_peaks(chrom_intensities, height=1)
35     max_peak_index = np.argmax(chrom_intensities[peaks])
36     max_peak_time = chrom_times[peaks[max_peak_index]]
37     max_peak_intensity = chrom_intensities[peaks[max_peak_index]]
38
39     # Plot chromatogram and spectrum side by side using Plotly
40     fig = go.Figure()
41
42     # Add chromatogram trace
43     fig.add_trace(go.Scatter(x=chrom_times, y=chrom_intensities, mode='lines', name='Chromatogram'))

```

- Une fois ces étapes réalisées, MSconvert sera correctement installé et votre logiciel sera fonctionnel.

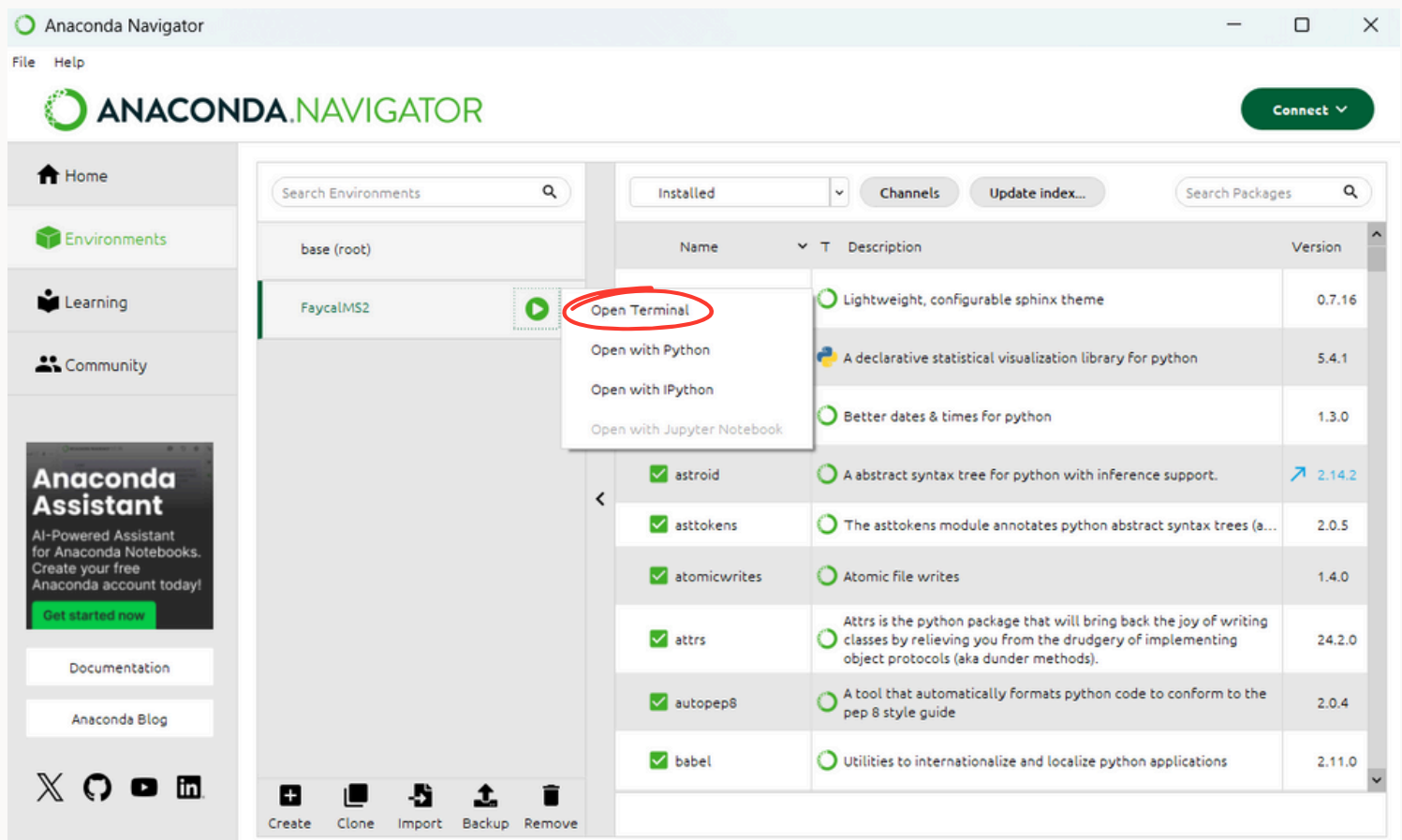
```

1 import os
2 import pyopenms
3 import numpy as np
4 import pandas as pd
5 import plotly.graph_objects as go
6 from scipy.signal import find_peaks
7 from sklearn.metrics.pairwise import cosine_similarity
8 from scipy.stats import pearsonr, spearmanr
9 from scipy.sparse import csr_matrix
10 import streamlit as st
11 import subprocess
12 from joblib import Parallel, delayed
13 import time
14
15 # Function to generate output mzML path from RAW file path
16 def generate_mzml_path(raw_file_path):
17     base, _ = os.path.splitext(raw_file_path)
18     return base + ".mzML"
19
20 # Function to convert RAW to mzML
21 def convert_raw_to_mzml(raw_file):
22     mzml_file = generate_mzml_path(raw_file)
23     subprocess.run(["C:\\Program Files\\ProteoWizard\\ProteoWizard 3.0.24143.838\\msconvert", raw_file, "-o", os.path.dirname(mzml_file)
24     return mzml_file
25
26 # Function to analyze an mzML file
27 def analyze_mzml_file(mzml_file):
28     exp = pyopenms.MSEExperiment()
29     pyopenms.MzMLFile().load(mzml_file, exp)
30
31     chromatogram = exp.getChromatograms()[0]
32     chrom_times, chrom_intensities = chromatogram.get_peaks()
33
34     peaks, _ = find_peaks(chrom_intensities, height=1)
35     max_peak_index = np.argmax(chrom_intensities[peaks])
36     max_peak_time = chrom_times[peaks[max_peak_index]]
37     max_peak_intensity = chrom_intensities[peaks[max_peak_index]]
38
39     # Plot chromatogram and spectrum side by side using Plotly
40     fig = go.Figure()
41
42     # Add chromatogram trace
43     fig.add_trace(go.Scatter(x=chrom_times, y=chrom_intensities, mode='lines', name='Chromatogram'))
44     fig.add_trace(go.Scatter(x=[max_peak_time], y=[max_peak_intensity], mode='markers', name='Most Intense Peak', marker=dict(color='red', size=100)))

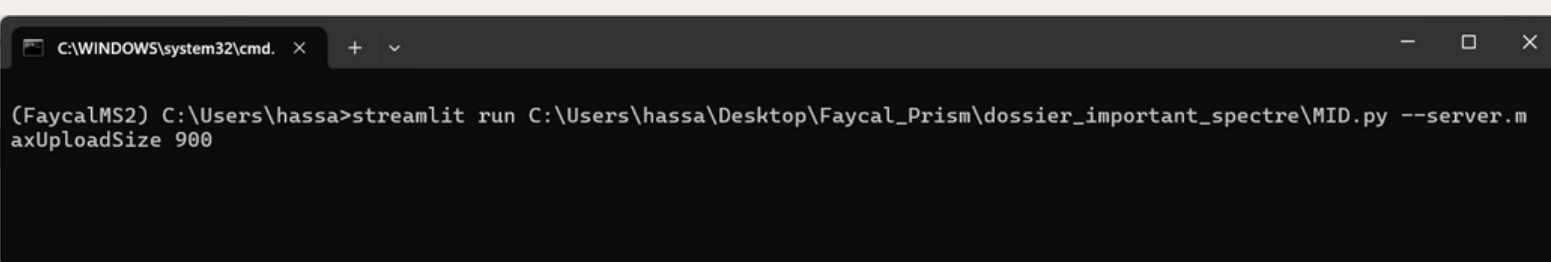
```

Utilisation du logiciel

- Pour utiliser le logiciel, commencez par ouvrir Anaconda Navigator.
- Dans Anaconda Navigator, sélectionnez l'environnement que vous avez importé précédemment.
- Ouvrez le terminal de commande en cliquant sur "Open Terminal" dans l'environnement sélectionné.



- Dans le terminal, exécutez la commande suivante pour lancer Streamlit :
- `streamlit run chemin_jusqu_à_MID.py --server.maxUploadSize 900`
- Remplacez "chemin_jusqu_à_MID.py" par le chemin complet vers le fichier **MID.py**.

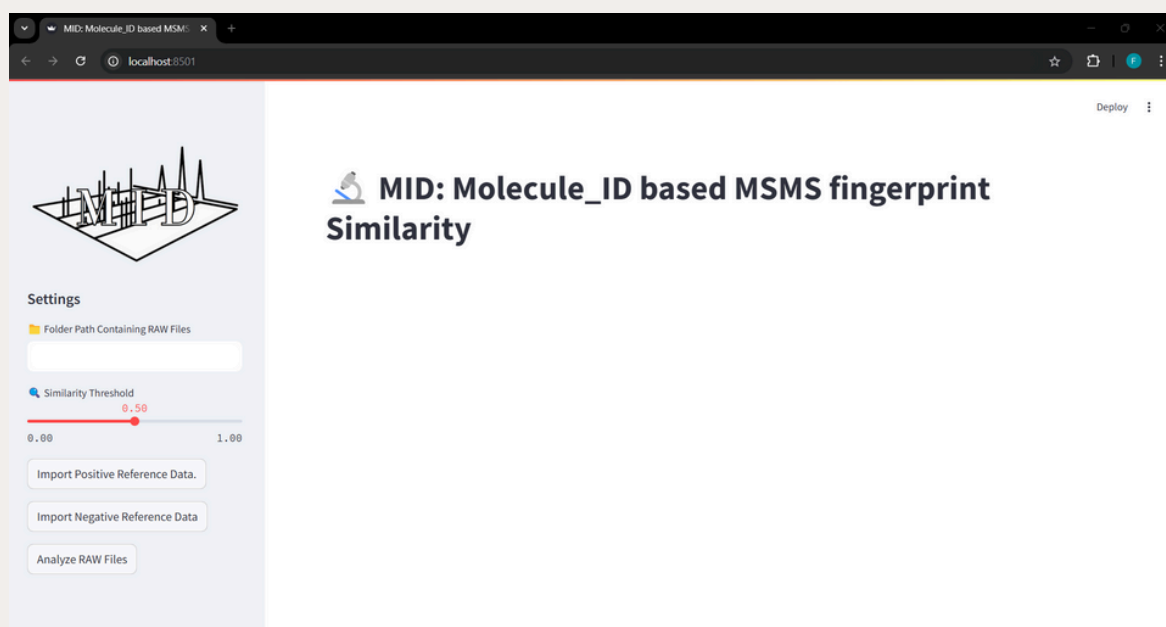


```

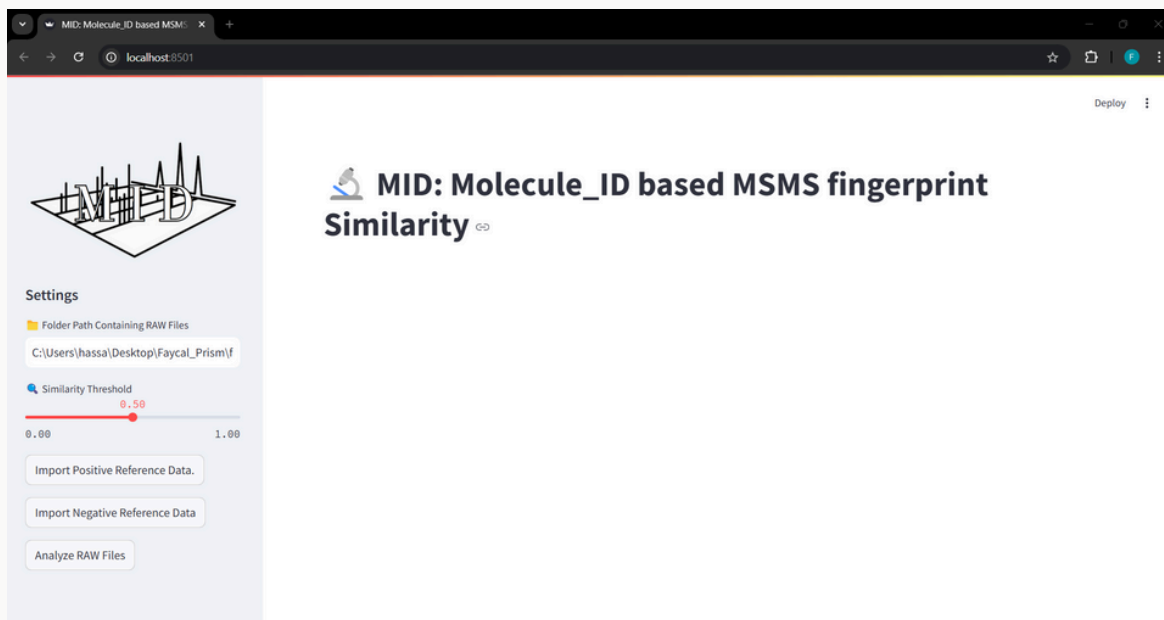
C:\WINDOWS\system32\cmd. x + v

(FaycalMS2) C:\Users\hasa>streamlit run C:\Users\hasa\Desktop\Faycal_Prism\dossier_important_spectre\MID.py --server.maxUploadSize 900
  
```

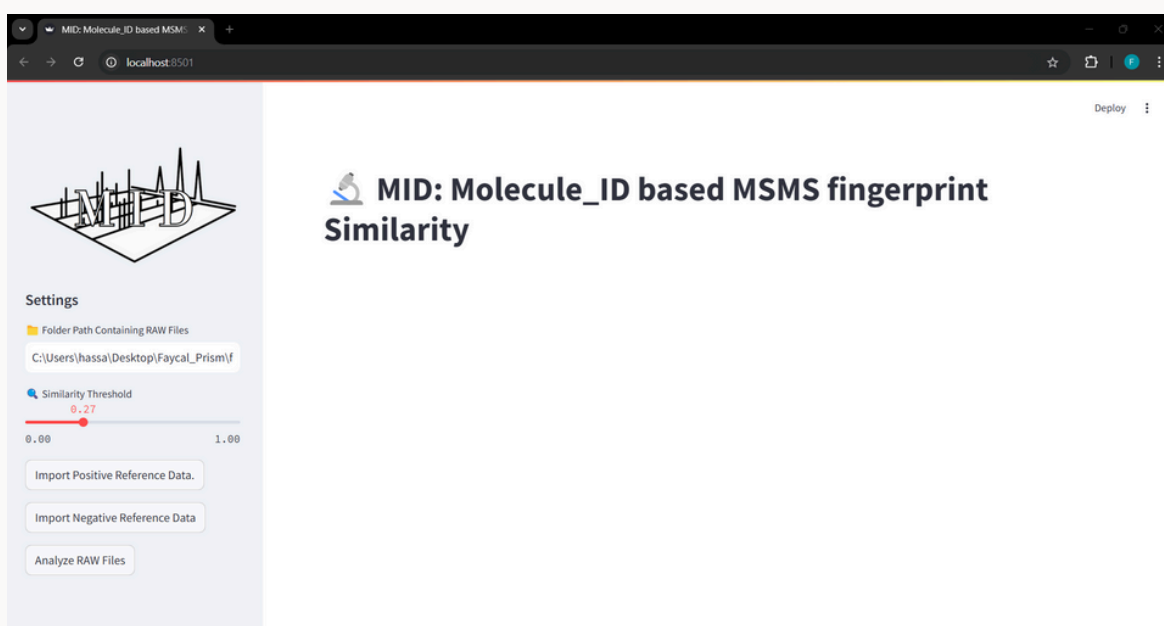
- Une fois la commande exécutée, une nouvelle page Google s'ouvrira avec l'interface du logiciel.



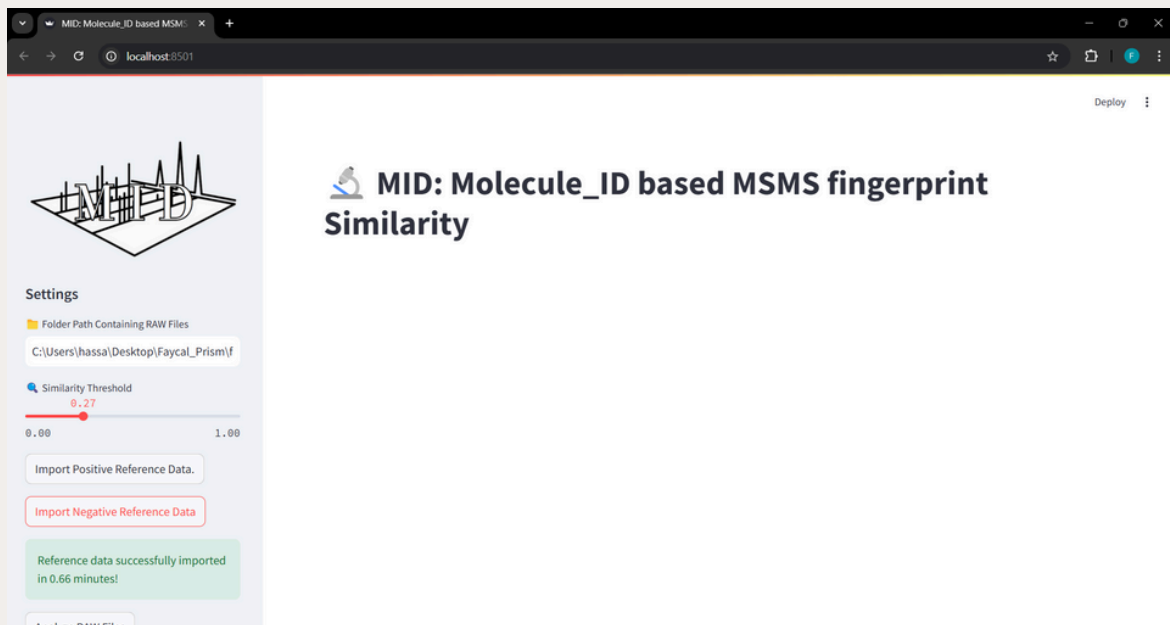
- Dans l'interface, indiquez le chemin du fichier qui contient le dossier **raw** à importer.



- Ensuite, spécifiez le **Similarity Threshold** que vous souhaitez utiliser. Ce seuil doit être compris entre 0 et 1.



- Importez les fichiers positifs ou négatifs en fonction du fichier que vous avez importé.



- Enfin, cliquez sur "Analyze Raw files" pour lancer l'analyse. Le logiciel commencera alors à traiter les fichiers et à afficher les résultats.

