

Supplementary File S1: Python filtering algorithm to identify significant metabolites in mass spectrometry results.

```
/*Index:
  - AI was used in the process of code creation and debugging
  - Code Purpose:
    - This code filters a metabolomics volcano-plot dataset
      to identify statistically significant metabolites (raw
      p-value < 0.05 and |log2 fold change| > 1), then
      parses the RT/mass field to extract only the mass
      values. It outputs a clean list of those significant
      metabolite masses for downstream identification or
      database searching.
*/

!pip install bioservices pandas

import pandas as pd
import numpy as np

# Load your table
df = pd.read_csv('volcano-3.csv')

# Add significance logic
df['significant'] = (df['raw.pval'] < 0.05) &
(abs(df['log2(FC)']) > 1)

# Export significant metabolites
significant_metabs = df[df['significant']]
significant_metabs.to_csv('significant_metabolites.csv',
index=False)

import pandas as pd

# Load CSV using comma as delimiter
df = pd.read_csv("significant_metabolites.csv", header=None)

# Split the first column into RT and mass using '/'
df[['RT', 'mass']] = df[0].str.split('/', expand=True)
df['mass'] = df['mass'].astype(float) # Convert to float for
HMDB search

# (Optional) Drop the old mixed column
df = df.drop(columns=[0])

# Preview the parsed data
```

```
print(df['mass'])

import pandas as pd

# Load CSV using comma as delimiter
df = pd.read_csv("significant_metabolites.csv", header=None)

# Split the first column into RT and mass using '/'
df[['RT', 'mass']] = df[0].str.split('/', expand=True)

# Convert mass to float, suppress errors from NaNs
df['mass'] = pd.to_numeric(df['mass'], errors='coerce')

# Drop rows with NaN masses (if any)
df = df.dropna(subset=['mass'])

# Save mass values to a new file, no index, no header
df['mass'].to_csv("mass_values_only.txt", index=False,
header=False)
```