

# User Manual

## **HAMMER: Automated operation of Mass Frontier to construct *in-silico* mass spectral fragmentation libraries**

Jiarui Zhou<sup>1,†</sup>, Ralf J. M. Weber<sup>2,†</sup>, J. William Allwood<sup>2</sup>, Robert Mistrik<sup>4</sup>, Zexuan Zhu<sup>5</sup>,  
Zhen Ji<sup>5</sup>, Siping Chen<sup>6</sup>, Warwick B. Dunn<sup>2</sup>, Shan He<sup>3</sup> and Mark R. Viant<sup>2</sup>

<sup>1</sup>College of Biomedical Engineering and Instrument Science, Zhejiang University, Hangzhou 310027, China; <sup>2</sup>School of Biosciences and <sup>3</sup>School of Computer Science, University of Birmingham, Birmingham, B15 2TT, United Kingdom; <sup>4</sup>HighChem, Ltd., Leskova 11, 81104 Bratislava, Slovakia; <sup>5</sup>Shenzhen City Key Laboratory of Embedded System Design, College of Computer Science and Software Engineering, <sup>6</sup>School of Medicine, Shenzhen University, Shenzhen 518060, China;

Oct, 2013. Version 1.5

## Contents

Chapter 1 - Overview.....	4
Chapter 2 - Installation .....	5
2.1 Dependent Software .....	5
2.2 Development Environment .....	6
2.3 Components .....	7
2.4 Using Sikuli for GUI Automation .....	8
2.5 Install & Configure .....	9
2.5.1 Install Dependent Software.....	9
2.5.2 Install HAMMER .....	11
2.5.3 Patterns .....	11
2.5.4 Regions.....	23
2.5.5 Determine Patterns and Regions.....	24
Chapter 3 - Using HAMMER .....	28
3.1 Using the RetrieveStructures.....	28
3.1.1 Target File.....	28
3.1.2 Console Mode.....	28
3.1.3 Interactive Mode .....	29
3.1.4 Results .....	29
3.2 Using the InSilicoFragmentation.....	31
3.2.1 Before Start.....	31
3.2.2 Parameter Settings for Mass Frontier.....	31
3.2.3 Perform Fragmentation .....	32
3.2.4 Results .....	33
3.3 Using the OrganiseFragments.....	35
3.3.1 Input Data .....	35
3.3.2 Console Mode.....	35

3.3.3 Interactive Mode .....	35
3.3.4 Results .....	36
3.4 Using the SpectralMatching.....	43
3.4.1 In-silico Library & Input Real Spectral.....	43
3.4.2 Console Mode.....	43
3.4.3 Interactive Mode .....	44
3.4.4 Results .....	44
Chapter 4 - Update History and Troubleshooting .....	46
4.1 Update History .....	46
4.2 Troubleshooting .....	46

## Chapter 1 - Overview

Experimental MS<sup>n</sup> mass spectral libraries currently do not adequately cover chemical space. This limits the robust annotation of metabolites in metabolomics studies of complex biological samples. *In-silico* fragmentation libraries would improve the identification of compounds from experimental multi stage fragmentation data when experimental reference data is unavailable.

Here we present HAMMER, a freely available software package to automatically control Mass Frontier software to construct *in-silico* fragmentation libraries.

## Chapter 2 - Installation

### 2.1 Dependent Software

1. **System requirements:** HAMMER supports Windows 7, Windows Vista, and Windows XP (both 32 and 64 bit versions). Windows 8 is not fully supported. The software is developed and tested on 32-bit Windows 7.
2. **Java:** official 32-bit version Java development kit (JDK) / Java runtime environment (JRE) can be downloaded from <http://www.oracle.com/technetwork/java/javase/downloads/index.html>. HAMMER works on Java 6 and / or Java 7. The 64-bit version is not supported.
3. **Mass Frontier 7:** Mass Frontier is a scientific software package for the management, evaluation and interpretation of mass spectra. HAMMER has been developed for Mass Frontier 7. Previous versions will be supported in the future.
4. **Open Babel:** Open Babel is an open source toolbox for chemical data formats conversion. The latest version of Open Babel can be downloaded from [http://openbabel.org/wiki/Get\\_Open\\_Babel](http://openbabel.org/wiki/Get_Open_Babel).
5. **Python 2.7.5 (Optional):** the official Python package can be downloaded from <http://www.python.org/download/>. The latest version of 32-bit Python 2.7.5 is recommended.

The binary version of HAMMER already contains a python runtime. This step is optional.

6. **Sikuli X (Optional):** Sikuli is a popular visual scripting software package. The latest version of Sikuli can be downloaded from <http://www.sikuli.org/download.html>. Before installation, make sure that all previous versions are uninstalled.

The latest HAMMER already contains a standalone version of Sikuli X. This step is optional. If Sikuli does not work properly, online support can be found here: <https://answers.launchpad.net/sikuli/+faq/2005>.

## 2.2 Development Environment

HAMMER is developed and tested under the following environment:

- Intel Core i5 2.5GHz CPU, 4GB memory, screen resolution 1680×1050
- Windows 7 (32-bit professional edition)
- JDK 6 update 43 (32-bit x86)
- Mass Frontier 7
- Open Babel 2.3.1
- Python 2.7.3 (32-bit x86)
- Sikuli X 1.0rc3 (with r930 patch)

## 2.3 Components

HAMMER includes four major components:

1. **RetrieveStructures** is a batch processing based automation tool to search and download compound structures.
2. **InSilicoFragmentation** is a visual scripting tool to manipulate Mass Frontier and perform high-throughput multi stage *in-silico* fragmentation on multiple compounds.
3. **OrganiseFragments** is a tool to organize the Mass Frontier results and export them as fragmentation data files.
4. **SpectralMatching** is a tool to perform batch matching of real spectral against *in-silico* fragmentation libraries.

## 2.4 Using Sikuli for GUI Automation

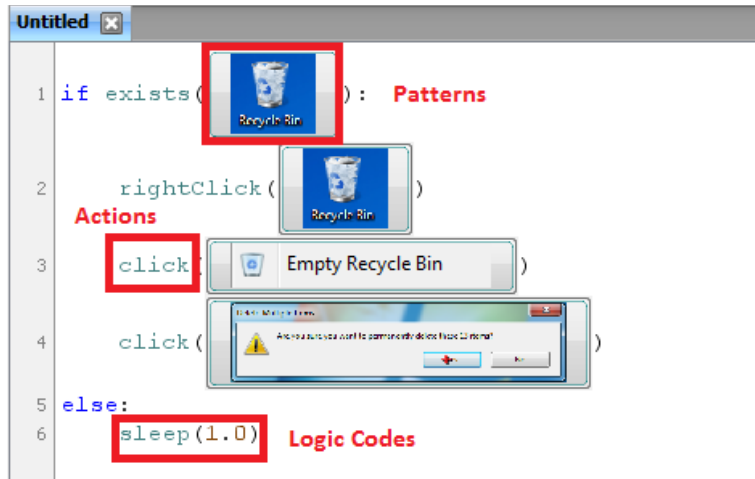
Sikuli is a visual scripting software package to automate GUI operations using image recognition and keyboard / mouse events simulation. It contains a script runtime and an integrated development environment (IDE). The latest version, Sikuli X, is available for Windows, Mac OS X, and Linux.

Sikuli can be installed easily by using the official installer, or simply by copying the program folder from one computer to another. For general users, the Sikuli runtime is already included as part of the HAMMER package, and therefore does not need to be installed separately. Advanced users may want to install the full Sikuli package in order to conduct further developments.

Written in the Python language, a Sikuli script usually consists of four major components:

1. **Patterns:** GUI images that should be searched and recognized (called matched) on the screen. The patterns are normally status identifiers (e.g. status bars) or GUI controllers (e.g. command buttons).
2. **Regions:** rectangular areas defined on the screen to search the patterns, in order to reduce search time and improve recognition accuracy.
3. **Logic Codes:** source codes to detect the pattern status, process accordingly, and decide the proper actions in response.
4. **Actions:** simulated keyboard events (e.g. press Enter) and mouse events (e.g. double-click a pattern) that should be sent to the automated software.

The Sikuli IDE is a very useful tool to develop Sikuli scripts. It displays the patterns and regions intuitively along with the logic source codes. The IDE also provides built-in tools to capture GUI patterns and select search regions (see section 2.5.5 for more information). A demo script opened in the Sikuli IDE is illustrated as follows:



For more information and examples, visit the Sikuli documentation website: <http://doc.sikuli.org>.



## 2.5 Install & Configure

### 2.5.1 Install Dependent Software

The dependent software can easily be installed using the official installers. Users should check the software status using following commands (in cmd console):

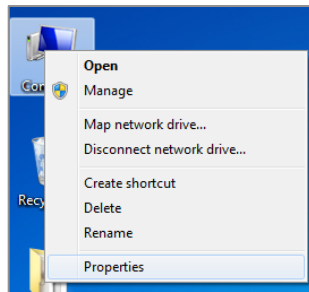
```
> java -version
java version "1.6.0_43"
Java(TM) SE Runtime Environment (build 1.6.0_43-b01-447-11M4203)
Java HotSpot(TM) 64-Bit Server VM (build 20.14-b01-447, mixed mode)
> babel -V
Open Babel 2.3.1 -- Nov 22 2012 -- 22:14:42
> python -V
Python 2.7.3
```

Note that the version numbers may be different depending on which packages are installed, and Python is optional. If a Command-Not-Found error occurs, add the following path to the system environment variable "PATH":

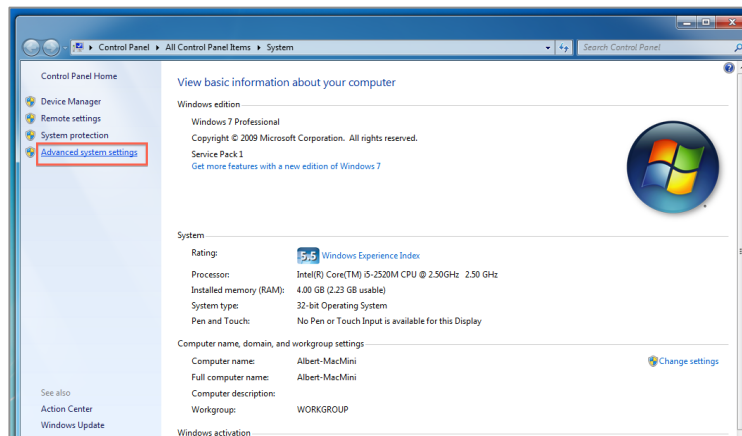
1. **Java:** %JAVA\_PATH%\bin\ (%JAVA\_PATH% is the path where JDK/JRE is installed)
2. **Open Babel:** %BABEL\_PATH%\ (%BABEL\_PATH% is the path where Open Babel is installed)
3. **Python:** %PYTHON\_PATH%\ (%PYTHON\_PATH% is the path where Python is installed)

To add the paths into system environment variables:

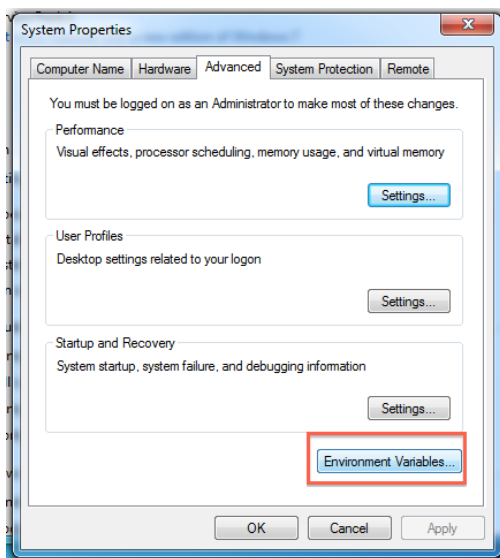
1. Right-click "My Computer" on the desktop, and then click "Properties".



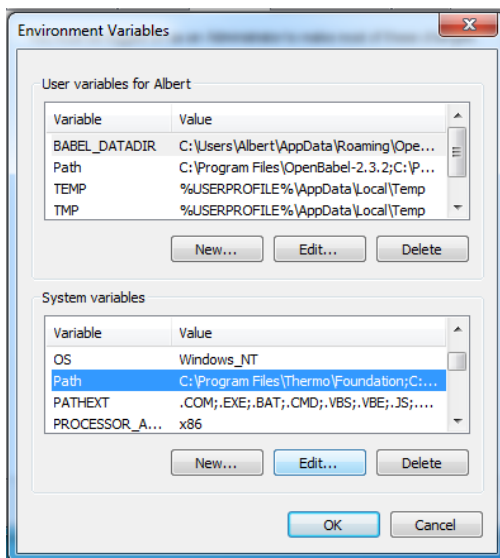
2. In Windows 7 and Vista, click "Advanced system settings" on the left pane. Ignore this step in Windows XP.



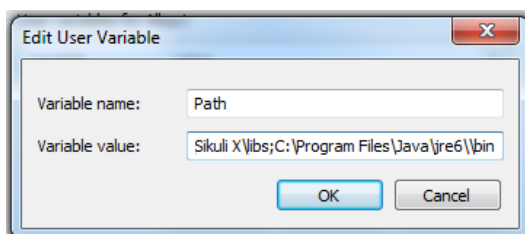
3. Click the "Advanced" tab, and then click "Environment variables".



4. Find "PATH" in "System variables" or "User Variables", double-click it to edit.



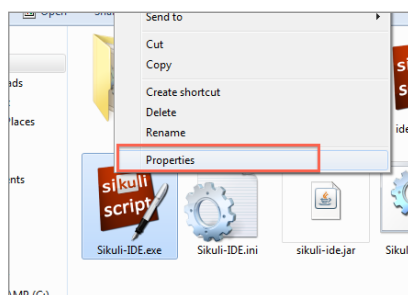
5. Append the paths to the "Variable value". Note that the different paths must be separated using a semicolon ";". For instance, assume that the original value of the variable "PATH" is "C:\Windows\system32; C:\Windows", and the %JAVA\_PATH% is "C:\Program Files\Java\jdk1.6.0\_43". The modified "PATH" variable value should be: "C:\Windows\system32; C:\Windows; C:\Program Files\Java\jdk1.6.0\_43\bin".



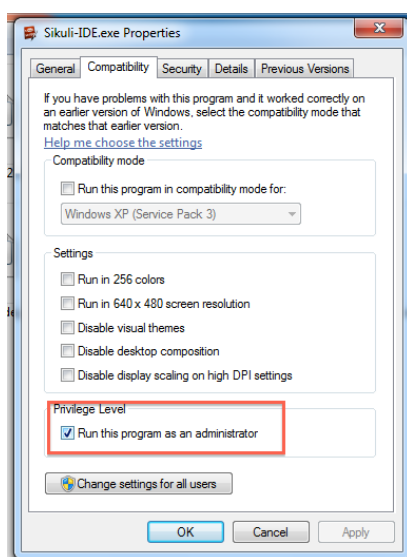
6. Click "OK" to save the changes.

In Windows 7 and Vista, Mass Frontier needs to be run as administrator. Therefore the HAMMER should also be run as administrator to function correctly:

1. Find the “Sikuli-IDE.exe” in folder “InSilicoFragmentation\Sikuli”, or in the path where Sikuli X is installed.
2. Right-click “Sikuli-IDE.exe”, and then click “Properties”.



3. Check the “Run this program as an administrator” option.



4. Click “OK” to save the changes.

## 2.5.2 Install HAMMER

The RetrieveStructures, OrganiseFragments, and SpectralMatching work out of the box as long as the dependent software is correctly installed. However the InSilicoFragmentation requires configuration before it can be used.

The InSilicoFragmentation is developed based on Sikuli, which uses image recognition to search and operate GUI elements. HAMMER provides two versions of InSilicoFragmentation for Windows 7 and XP users separately. They should be able to work directly without any setup. However, the automation tool may sometimes need to be configured for different screen sizes, resolutions, or GUI styles (themes). In that case users should modify the contents in “res.index” and “Resources” before using the tool.

Plain text file “res.index” includes the index of pattern images and search regions, and the folder “Resources” contains the corresponding pattern image files. The index file “res.index” consists of two sections: patterns and regions (see section 2.4).

## 2.5.3 Patterns

Image File Name	Similarity
<code>mf_openfile_dialog = OpenFilename.png;</code>	<code>[-90, -20]; 0.7</code>
Pattern ID	Target Offset

1. **Pattern ID:** identity of the GUI image pattern. Should not be changed by users.

2. **Image File Name:** name of the pattern image file stored in the “Resources” folder.
3. **Target Offset:** focus offset of the target GUI element. Users can specify a different click point other than the center of the pattern. Negative offset values indicate that the click point is on the top/left side of the center, and positive values indicate it is on the bottom/right side.

More details of Target Offset can be found here: <http://doc.sikuli.org/pattern.html#Pattern.targetOffset>

4. **Similarity:** minimum similarity to match the pattern image. Its value should be between 0 and 1. The smaller the similarity value, the easier an image can be matched to the pattern.

More details of Similarity can be found here: <http://doc.sikuli.org/pattern.html#Pattern.similar>

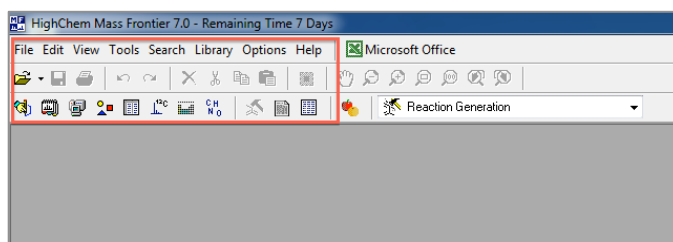
The Image File Name, Target Offset, and Similarity are separated using semicolons “;”. If users want to specify the Similarity but not the Target Offset, the second part of the pattern value should be set to “None”. For instance, “mf\_openfile\_dialog = OpenFileDialog.png; None; 0.7”.

The patterns used in HAMMER are:

1. **mf\_check\_opened:**



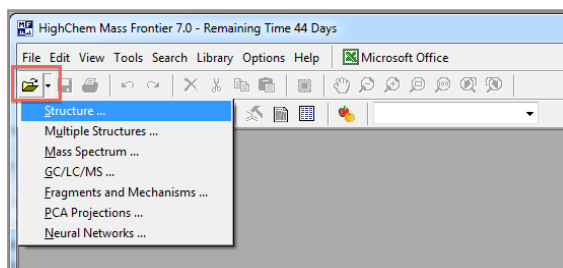
Check whether Mass Frontier has launched and is sitting on top of all other software. The automation tool will search the whole screen for this pattern.



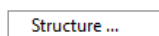
2. **mf\_openfile\_icon:**



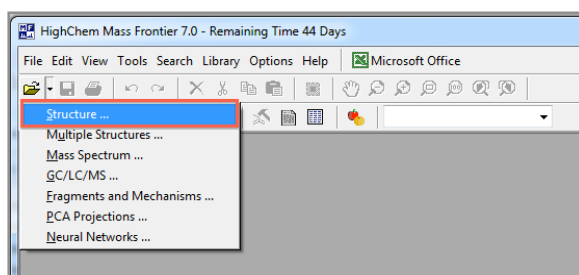
Click to open a file. This pattern is searched within region “mf\_openfile\_reg” (see the “regions” section).



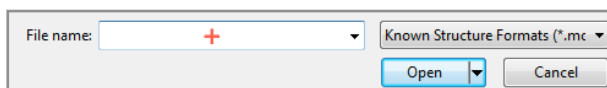
3. **mf\_openfile\_struct:**



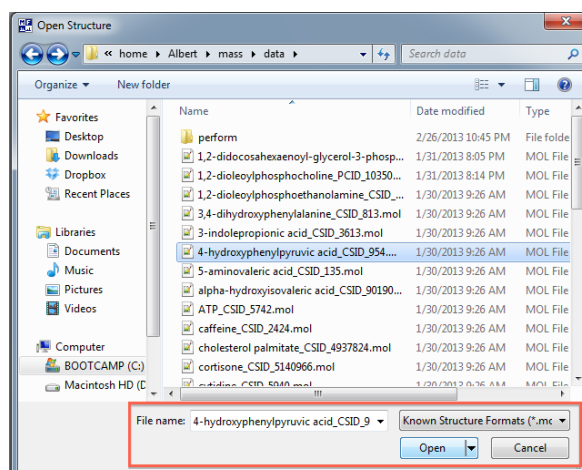
Click to open a structure file (.mol). This pattern is searched within region “mf\_openfile\_reg”.



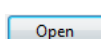
#### 4. mf\_openfile\_dialog:



Check whether the open file dialog is launched. This pattern is searched within region "mf\_operate\_reg". The target offset (marked with "+") should be placed inside the "File name" input box.

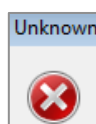


#### 5. mf\_openfile\_dialog\_open:

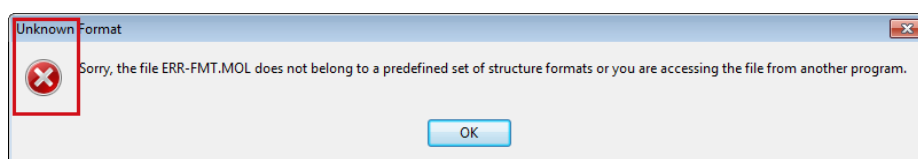


Click to open the selected .mol file. This pattern is searched within the region of pattern "mf\_openfile\_dialog".

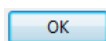
#### 6. mf\_openfile\_dialog\_error\_format:



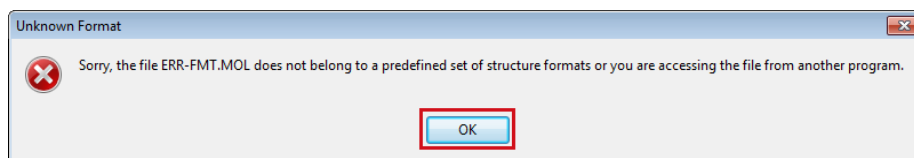
This error message appears when the .mol file cannot be opened. The pattern is searched within region "mf\_operate\_reg". File "err-fmt.mol" is provided to produce this situation.



#### 7. mf\_openfile\_dialog\_error\_format\_ok:



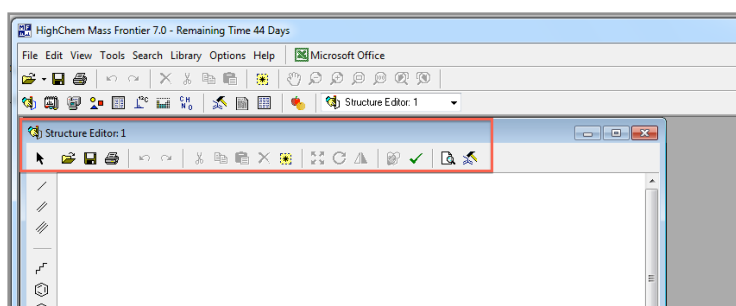
Click to close the error message box. This pattern is searched within region “mf\_operate\_reg”.



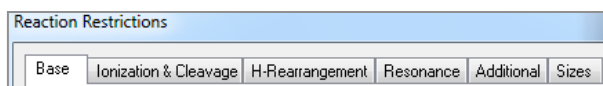
### 8. mf\_frag\_icon:



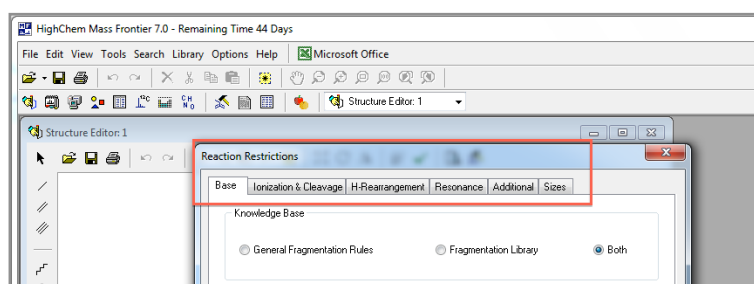
Check whether the .mol file is opened in structure editor. The target offset should be placed on top of the “Start fragmentation” button (on the right-hand side of the image).



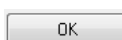
### 9. mf\_frag\_setting:



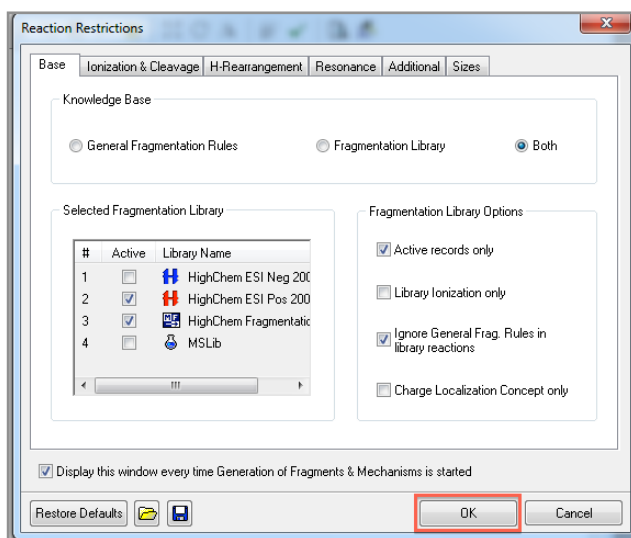
Check whether the structure can be fragmented. This pattern is searched within region “mf\_operate\_reg”.



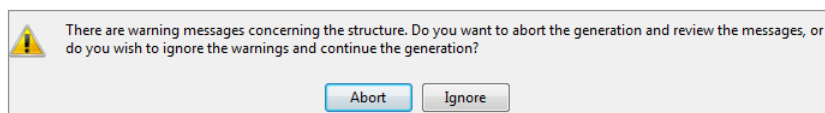
### 10. mf\_frag\_setting\_ok:



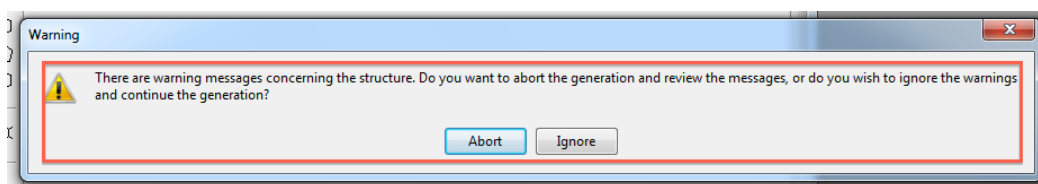
Click to start fragmentation. This pattern is searched within region “mf\_operate\_reg”.



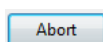
### 11. mf\_frag\_warning:



This warning message appears when the structure cannot be fragmented. The pattern is searched within region "mf\_operate\_reg". File "err-frag.mol" is provided to produce this situation.

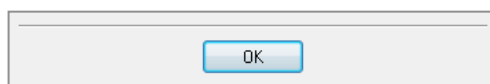


### 12. mf\_frag\_warning\_abort:

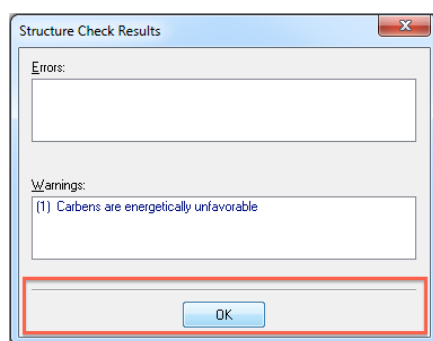
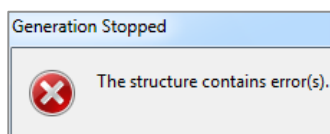


Click to abort the fragmentation process. This pattern is searched within the region of pattern "mf\_frag\_warning".

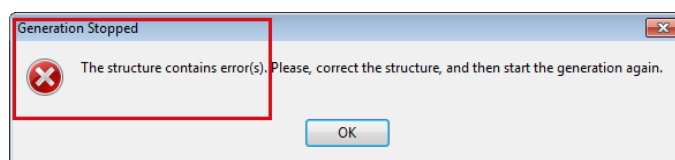
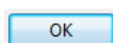
### 13. mf\_frag\_warning\_ok:



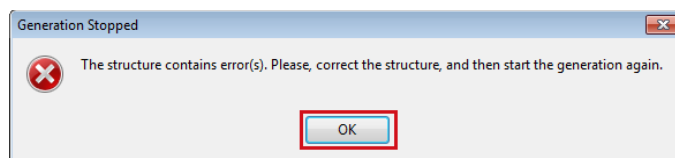
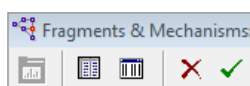
Click to confirm the process abortion. This pattern is searched within region "mf\_operate\_reg".

**14.mf\_frag\_error:**

This message appears when errors occurred during the fragmentation. The pattern is searched within region "mf\_operate\_reg". File "gen-fail.mol" is provided to produce this situation.

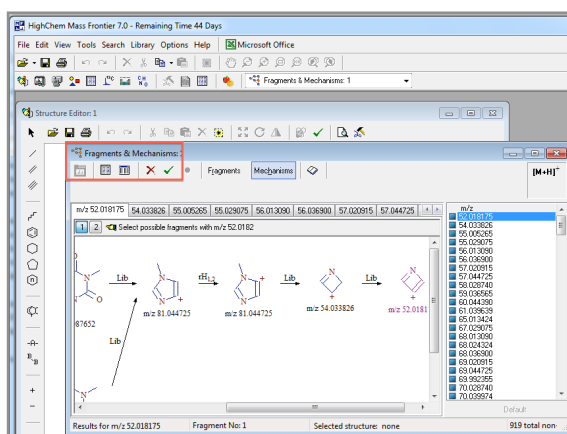
**15.mf\_frag\_error\_ok:**

Click to close the error message box. This pattern is searched within region "mf\_operate\_reg".

**16.mf\_fragres\_fm:**

Check whether the fragmentation is finished. This pattern is searched within region "mf\_operate\_reg".

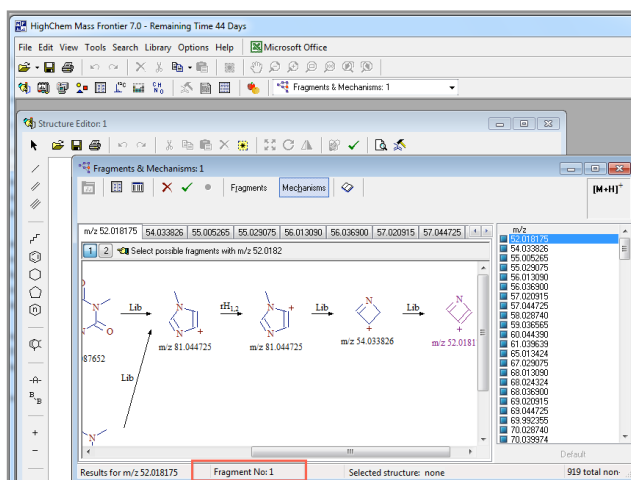




### 17. mf\_fragres\_fmcheck:

Fragment No: 1

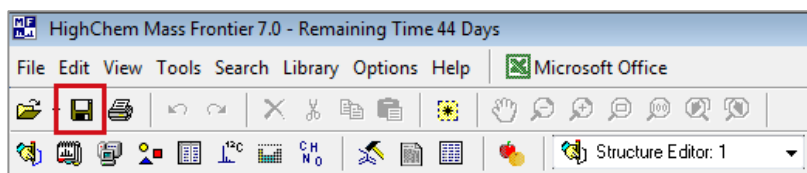
Check whether all the fragments have been listed in the “Fragments & Mechanisms” dialog. This process may take a few seconds or even minutes to finish for large compounds. The InSilicoFragmentation needs to confirm it before moving to the next step. This pattern is searched within region “mf\_operate\_reg”.



### 18. mf\_save\_mech:



Click to save the fragmentation mechanisms. This pattern is searched within region “mf\_toolbar\_reg”.

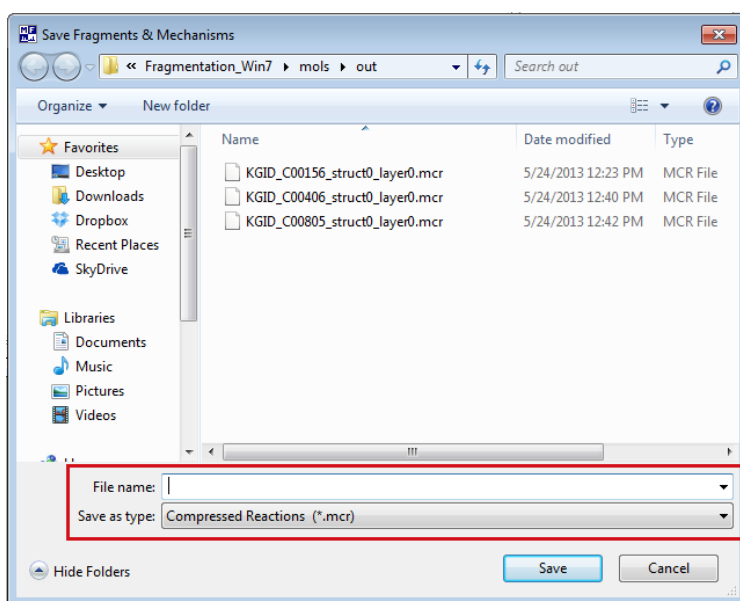


### 19. mf\_save\_mech\_dialog:

File name:  +

Save as type: Compressed Reactions (\*.mcr)

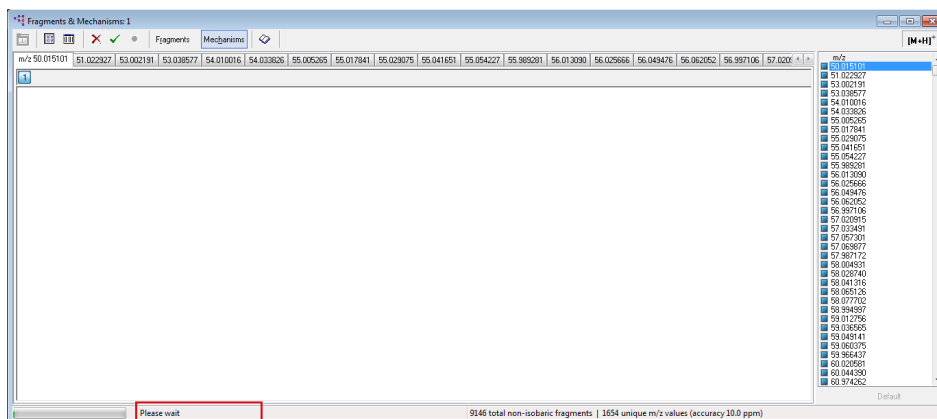
Double-click to enter the export filename. This pattern is searched within region “mf\_operate\_reg”. The target offset should be placed inside the “File name” input box.



## 20. mf\_save\_mech\_wait:

Please wait

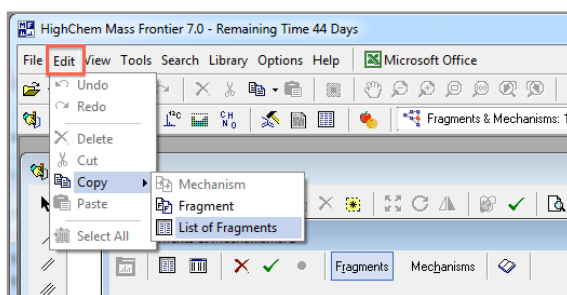
Check whether the saving process is finished. This pattern is searched within region “mf\_operate\_reg”.



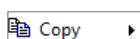
## 21. mf\_copymenu\_edit:

Edit

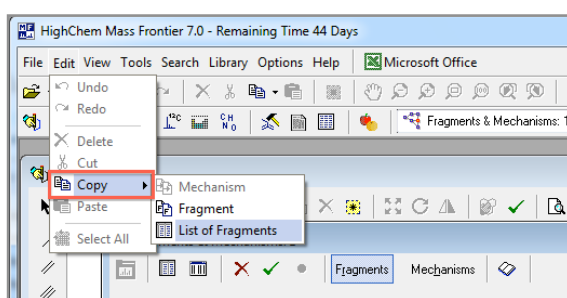
Click to open the copy menu. This pattern is searched within region “mf\_copymenu\_reg”.



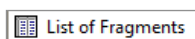
22. mf\_copymenu\_copy:



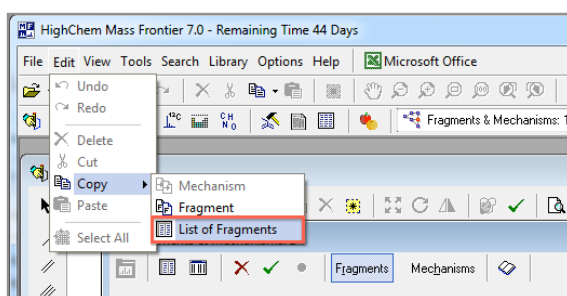
Click to copy fragments. This pattern is searched within region "mf\_copymenu\_reg".



23. mf\_copymenu\_listoffrags:



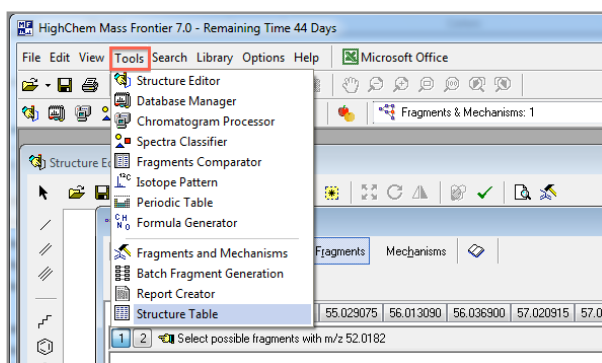
Click to copy list of fragments. This pattern is searched within region "mf\_copymenu\_reg".



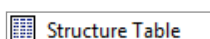
24. mf\_tablemenu\_tools:



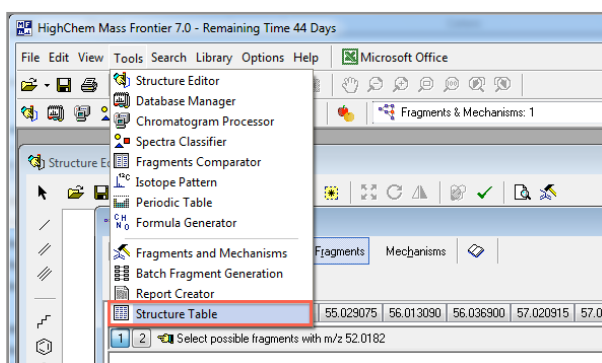
Click to open structure table menu. This pattern is searched within region "mf\_tablemenu\_reg".



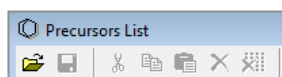
25. mf\_tablemenu\_structs:



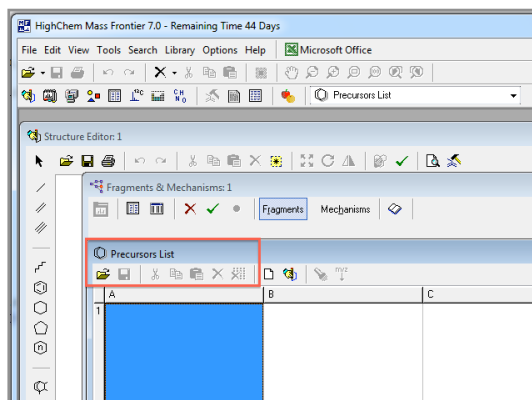
Click to open precursors list dialog. This pattern is searched within region "mf\_tablemenu\_reg".



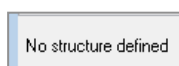
26. mf\_stable\_precursor:



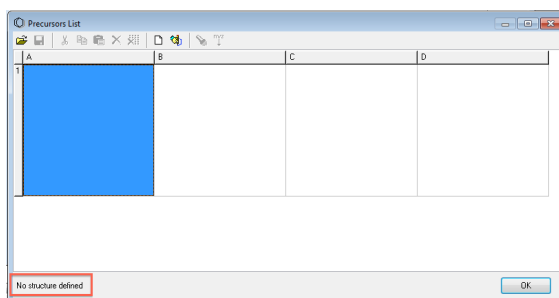
Check whether the precursors list dialog is opened. This pattern is searched within region "mf\_operate\_reg".



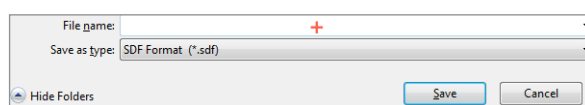
27. mf\_stable\_notexist:



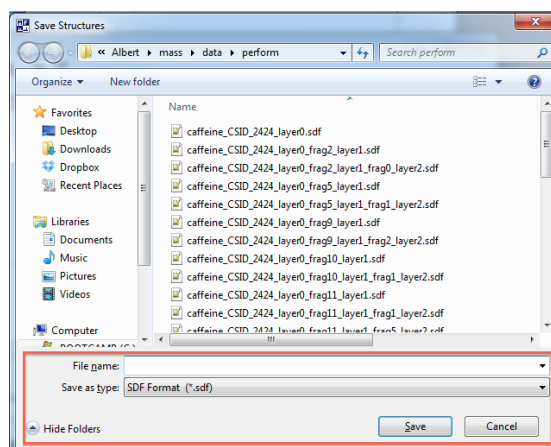
Check whether the list of fragments have been pasted to the “Precursors List” dialog. This pattern is searched within region “mf\_operate\_reg” and the default matching similarity is set to 0.9.



### 28.mf\_stable\_save\_dialog:



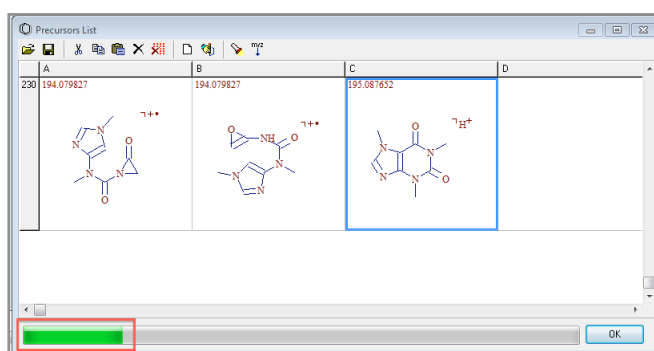
Check whether the “Save Structures” dialog is opened. This pattern is searched within region “mf\_operate\_reg”. The target offset should be placed inside the “File name” input box.



### 29.mf\_stable\_save\_procbar:



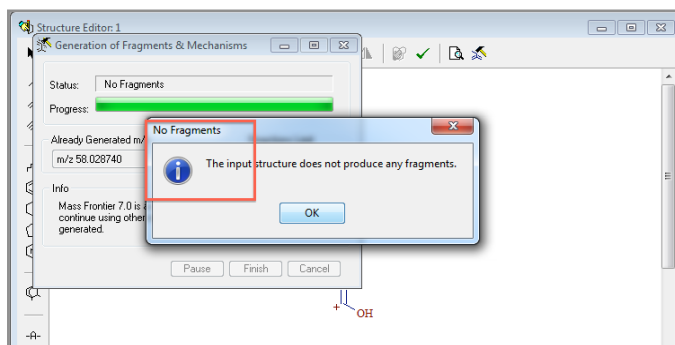
Check whether the fragments’ saving process is finished. This pattern is searched within region “mf\_operate\_reg”.



### 30.mf\_fragres\_nofrag:



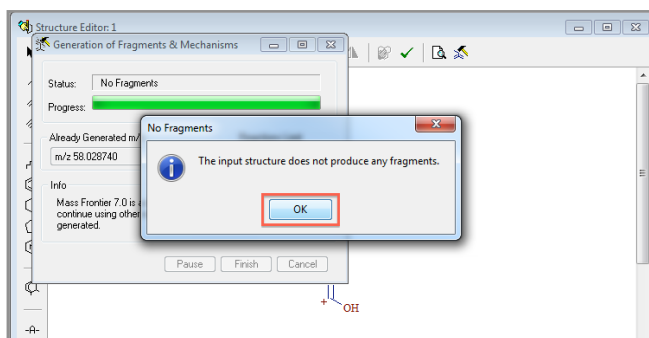
This message appears when no fragment can be generated from the original structure. This pattern is searched within region "mf\_operate\_reg". File "no-frag.mol" is provided to produce this situation.



**31.mf\_fragres\_nofrag\_ok:**



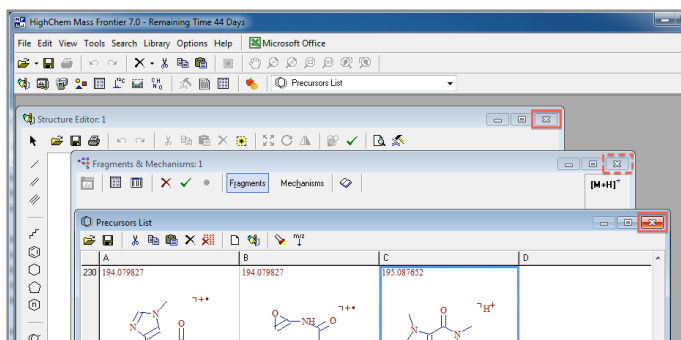
Click to close the "No Fragments" message box. This pattern is searched within region "mf\_operate\_reg".



**32.mf\_close\_cross:**



Click to close all sub windows in the Mass Frontier. This pattern is searched within region "mf\_operate\_reg" repeatedly until no such image pattern can be found.



## 2.5.4 Regions

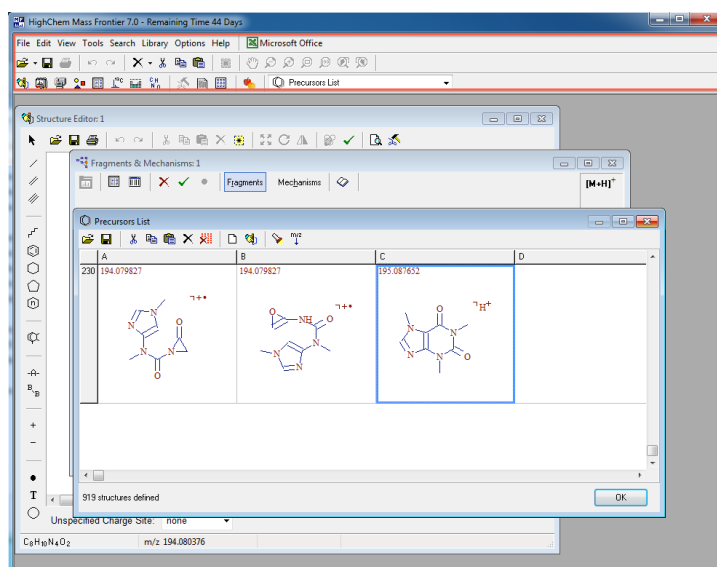
`mf_toolbar_reg = 0, 0, -1, 115`

Region ID      Region Offset

1. **Region ID:** identity of the search region. Should not be changed by users.
2. **Region Offset:** x position, y position, width and height of the region rectangle. Negative value (usually -1) indicates the screen's width / height is used as that of the region rectangle.

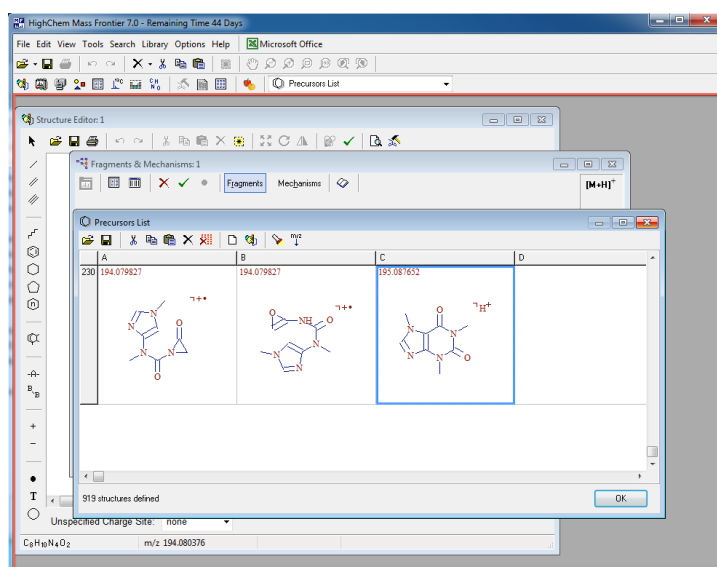
The regions used in HAMMER are:

### 1. `mf_toolbar_reg`:



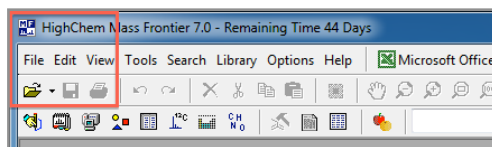
This region contains the menu and toolbar of the Mass Frontier.

### 2. `mf_operate_reg`:



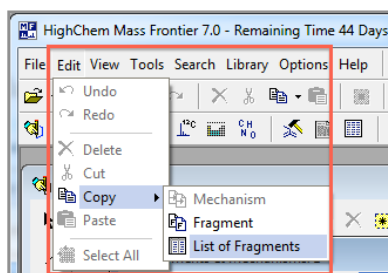
This is the region wherein all the sub windows of the Mass Frontier are shown. Most of the patterns listed above are searched within this region. Therefore its values must be defined carefully.

### 3. mf\_openfile\_reg:



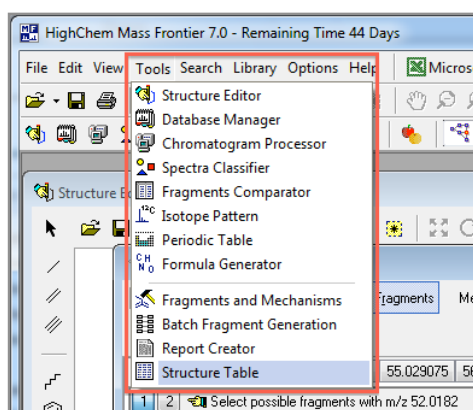
This is the region to search pattern “mf\_openfile\_icon”. It must contain this pattern image.

### 4. mf\_copymenu\_reg:



This is the region to search patterns “mf\_copymenu\_edit”, “mf\_copymenu\_copy”, and “mf\_copymenu\_listoffrags”.

### 5. mf\_tablemenu\_reg:



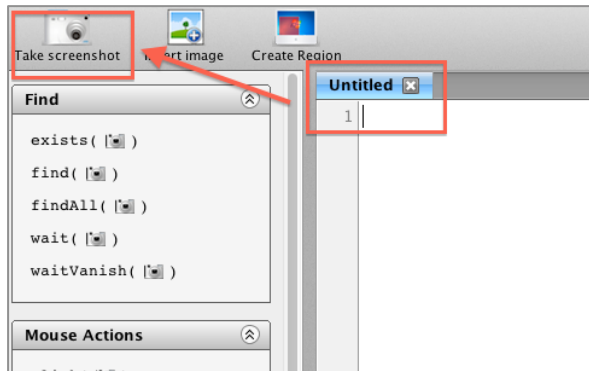
This is the region to search patterns “mf\_tablemenu\_tools” and “mf\_tablemenu\_structs”.

## 2.5.5 Determine Patterns and Regions

**Using Sikuli IDE:** Sikuli IDE (“Sikuli-IDE.exe”) provides built-in methods to capture pattern images and determine search regions. To capture new image patterns, users should:

1. Launch the Sikuli IDE and open a new script file.
2. Click the “Take screenshot” icon, and the software will enter the screen capture mode.



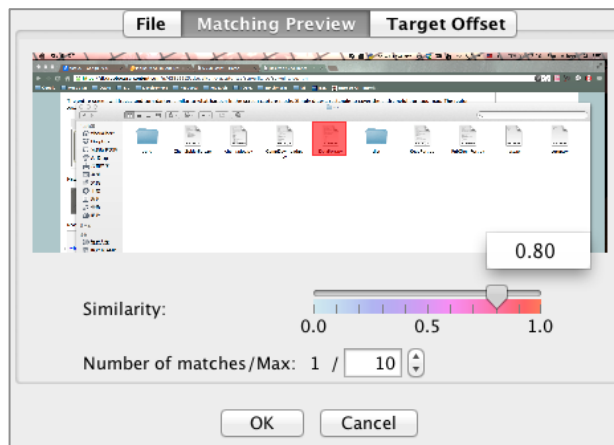


3. Drag a rectangle to include the target image. The center of the red cross indicates the focus offset of the pattern.

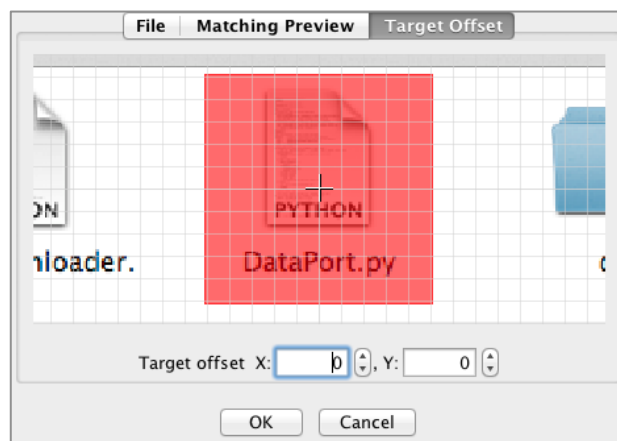


4. If users want to change the focus offset or similarity, click the pattern to open the “Pattern Settings” window.

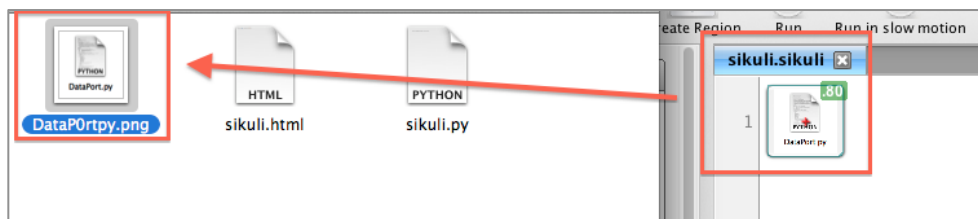
5. Click the “Matching Preview” tab to change the similarity value.



6. Click the “Target Offset” tab to change the offset value.



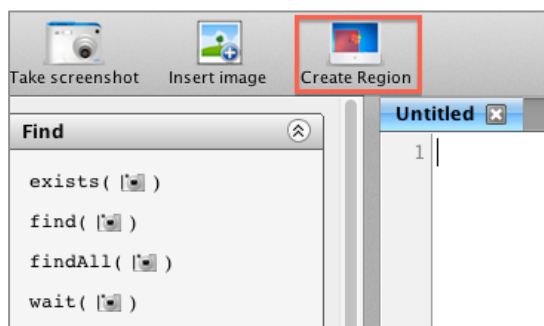
- Save the script file and then open it in Windows explorer. The captured image is in the script folder.



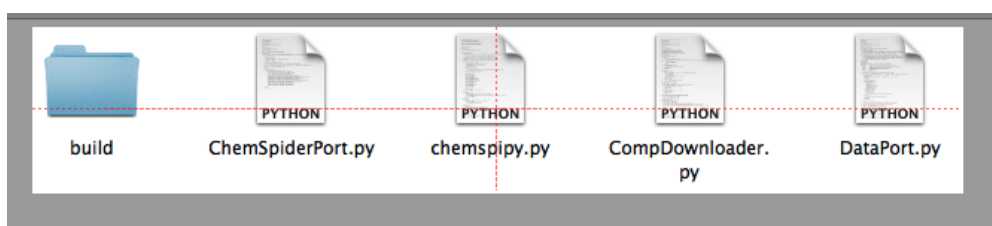
- Copy the image file into the "Resources" folder, and then change the content in "res.index" to use it in the InSilicoFragmentation. If the focus offset or the similarity is changed, add this information into the pattern value as well (see the "Patterns" section).

To determine a new search region, users should:

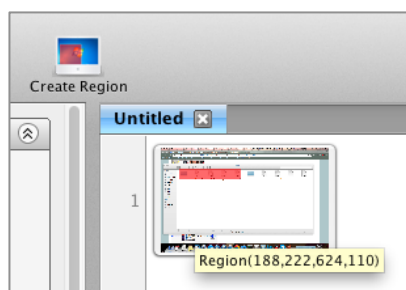
- Launch the Sikuli IDE and open a new script file.
- Click the "Create region" icon, and the software will enter the screen capture mode.



- Drag a rectangle to include the target region.



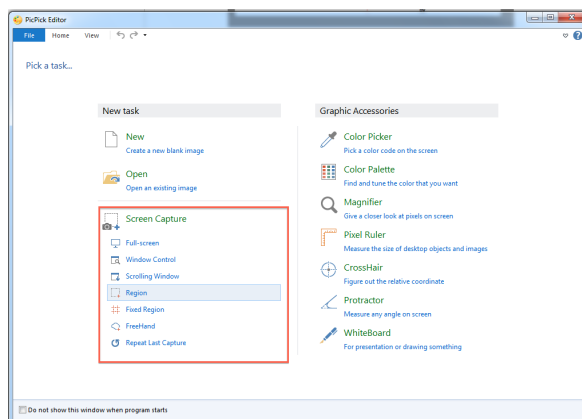
- Move mouse arrow on top of the captured region. The x position, y position, width, and height of the region rectangle will be displayed in a tooltip.



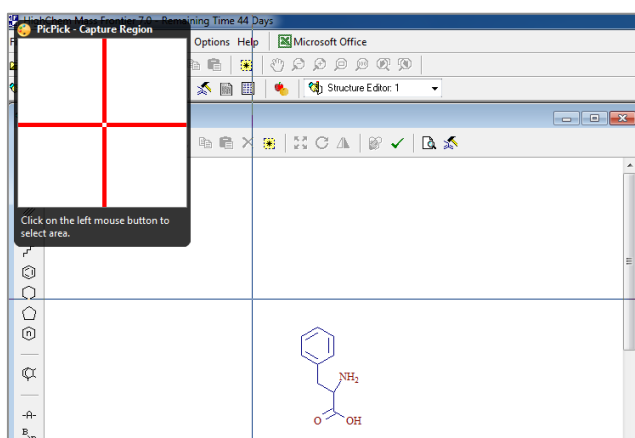
- Change the region values in "res.index" to use them in the InSilicoFragmentation (see the "Regions" section).

**Using Third Party Software:** users can utilize other software to capture the pattern images. One of the most popular free screen capture software is the PicPick (website: <http://www.picpick.org/en/>). To obtain new image patterns using PicPick, users should:

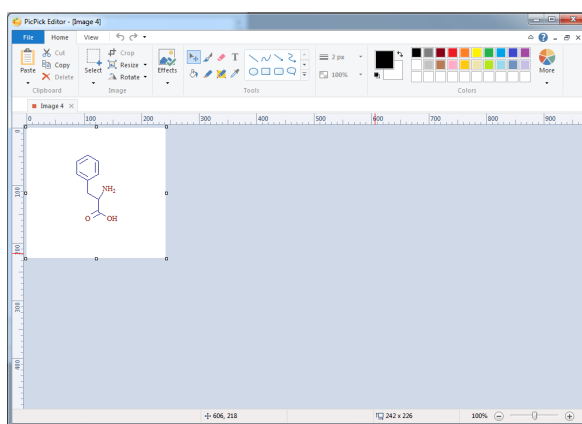
1. Launch the PicPick. Select one of the methods in the “Screen Capture” pane (the “Region” capture is recommended).



2. The software will enter the screen capture mode. Drag a rectangle to include the target image.



3. The captured image is shown in the PicPick Editor. Save it into a file.



4. Move the image file into the “Resources” folder, and then change the content in “res.index” to use it in the InSilicoFragmentation.

## Chapter 3 - Using HAMMER

### 3.1 Using the RetrieveStructures

#### 3.1.1 Target File

The RetrieveStructures utilizes a plain text file, namely the target file, to define compounds that need to be downloaded. The first line of the target file is the header, which will be ignored during the download:

```
TypeSearch Database SearchTerm
```

Other lines are used to define the target compounds:

```
Database(s) Comments
name CHEMSPIDER, PUBCHEM Tri-Sprintec # not found in ChemSpider, but found in PubChem
```

Search Type

Search Term

- **Search Type:** the type of search term. RetrieveStructures can search a compound according to the database ID (search type = "id"), the compound name (search type = "name"), or the empirical formula (search type = "formula"). It can also obtain all of the compounds within specific pathways (search type = "pathway"). The search type, databases, and search term are separated using one or more tabs "\t". Note that the spaces should not be used for separation.
- **Database(s):** databases to search the compounds within. RetrieveStructures supports searching in ChemSpider, PubChem, and KEGG. Only one database should be provided to the "id" search type, and the "pathway" search type supports the KEGG database only.
- **Search Term:** the key words to search a compound in online databases. Its value varies for different search types:
  - **id:** the search term is the entry ID of the target compound in the corresponding database.
  - **name:** the search term is the name of the target compound.
  - **formula:** the search term is the empirical formula of the target compound.
  - **pathway:** the search term is the name of the target pathway. RetrieveStructures will search the KEGG database for candidate pathways (ChemSpider and PubChem do not support pathway searching). All the compounds within each candidate pathway will be downloaded separately.
- **Comments (Optional):** comments starts with the hash character "#". All the contents in a line behind "#" will be ignored.

An example of the target file is shown as follows:

```
TypeSearch Database SearchTerm
id KEGG C07481
id PUBCHEM 2519
id CHEMSPIDER 2424
name KEGG, CHEMSPIDER caffeine
formula KEGG, PUBCHEM C8H10N4O2
pathway KEGG Phenylalanine metabolism # obtain pathway compounds in KEGG
```

#### 3.1.2 Console Mode

In console mode, the RetrieveStructures can be used as a single line command:

```
> RetrieveStructures.exe [--options] targets export
```

where “targets” is the path to the target file described in the above section, and “export” is the path to the folder wherein the results are going to be stored. Options of the RetrieveStructures are:

1. **no-split**: do not split multi-molecule entries into separate files.
2. **xls**: generate an excel report to summarize the search results. Otherwise a plain text report will be generated instead.
3. **outfmt format**: export data format, feasible formats include “mol” (default), “inchi”, “smiles”, and “formula”.
4. **max number**: maximum candidates number of a target compound. It should be an integer value between 0 and 100 (0 means to download all the candidate compounds). Default value is 10.
5. **help**: display the software help.

Here is an example to use the RetrieveStructures in console mode:

```
> RetrieveStructures.exe --xls --outfmt inchi targets.txt results\
```

It functions to search the compounds listed in target file “target.txt”, convert them into “inchi” format, export to the “results” folder, and generate an excel report.

### 3.1.3 Interactive Mode

Besides the console mode, RetrieveStructures can also be run in the interactive mode. Double-click “RetrieveStructures.exe” in windows explorer to enter this mode, wherein the parameters are set in an interactive manner.

```
> RetrieveStructures.exe
```

Enter the path of the target file:

```
Targets file >>
```

Enter the path of the export folder:

```
Export path >>
```

Decide whether the multi-molecule entries should be split into separate files (“y” for yes, “n” for no):

```
Split multi-molecule entries into separate files? (y/n, enter = y) >>
```

Decide whether the excel report should be generated (“y” for yes, “n” for no):

```
Generate excel report? (y/n, enter = n) >>
```

Enter the export data format (“mol”, “inchi”, “smiles”, or “formula”):

```
Export data format (enter = mol) >>
```


Enter the maximum candidates number (between 0 to 100):

```
Maximum candidates number (enter = 10) >>
```

Press enter and the RetrieveStructures will start the search and download process.

### 3.1.4 Results

Downloaded compounds are stored under the following format:

Candidate ID    Data Format  
 KGID\_C07481\_struct0.mol  
 Database ID    Structure ID

- Database ID:** ID of the database. “CSID” indicates the compound is downloaded from ChemSpider, “PCID” indicates PubChem, and “KGID” indicates KEGG.
- Candidate ID:** entry ID of the candidate compound in the corresponding database.
- Structure ID:** ID of the substructure. For multi-molecule entries, “Struct*N*” indicates the *N*th substructure (molecule) separated from the original compound. If the candidate compound contains only one structure, its value is “Struct0”.

If the RetrieveStructures is set to “no-split”, this part is absent.

- Data Format:** export data format set to the RetrieveStructures.

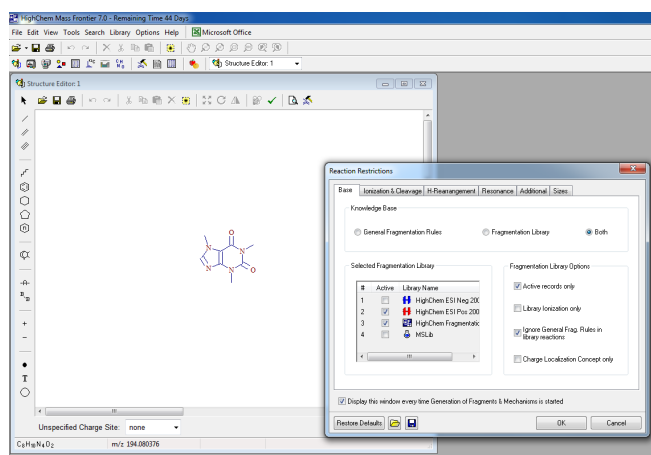
A report file is generated in the name of “compounds.txt” or “compounds.xls” to summarize the results. The search type, search term, candidates number, candidates IDs, website links, and remarks of each target compound are included in the report. An example of the excel report is shown as follows:

	A	B	C	D	E	F
1	Type	Search Term	Hits	Candidates	Link	Remark
2	formula	C8H10N4O2	13	KGID_C04087	<a href="http://www.genome.jp/dbget-bin/www_bget?cpd:C04087">http://www.genome.jp/dbget-bin/www_bget?cpd:C04087</a>	
3				KGID_C13403	<a href="http://www.genome.jp/dbget-bin/www_bget?cpd:C13403">http://www.genome.jp/dbget-bin/www_bget?cpd:C13403</a>	3 structures splitted from the original compound, remove [[Na]]
4				PCID_2519	<a href="http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?cid=2519">http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?cid=2519</a>	
5				PCID_67294	<a href="http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?cid=67294">http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?cid=67294</a>	
6				PCID_31212	<a href="http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?cid=31212">http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?cid=31212</a>	
7				PCID_72700	<a href="http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?cid=72700">http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?cid=72700</a>	
8				KGID_C07481	<a href="http://www.genome.jp/dbget-bin/www_bget?cpd:C07481">http://www.genome.jp/dbget-bin/www_bget?cpd:C07481</a>	
9				PCID_15061	<a href="http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?cid=15061">http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?cid=15061</a>	
10				PCID_76682	<a href="http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?cid=76682">http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?cid=76682</a>	
11				PCID_1326	<a href="http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?cid=1326">http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?cid=1326</a>	
12				PCID_1676	<a href="http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?cid=1676">http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?cid=1676</a>	
13				PCID_64136	<a href="http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?cid=64136">http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?cid=64136</a>	
14				PCID_79438	<a href="http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?cid=79438">http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?cid=79438</a>	
15	id	2424	1	CSID_2424	<a href="http://www.chemspider.com/Chemical-Structure.2424.html">http://www.chemspider.com/Chemical-Structure.2424.html</a>	
16	id	2519	1	PCID_2519	<a href="http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?cid=2519">http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?cid=2519</a>	
17	id	C07481	1	KGID_C07481	<a href="http://www.genome.jp/dbget-bin/www_bget?cpd:C07481">http://www.genome.jp/dbget-bin/www_bget?cpd:C07481</a>	
18	name	caffeine	3	KGID_C07481	<a href="http://www.genome.jp/dbget-bin/www_bget?cpd:C07481">http://www.genome.jp/dbget-bin/www_bget?cpd:C07481</a>	
19				KGID_C13403	<a href="http://www.genome.jp/dbget-bin/www_bget?cpd:C13403">http://www.genome.jp/dbget-bin/www_bget?cpd:C13403</a>	3 structures splitted from the original compound, remove [[Na]]
20				CSID_2424	<a href="http://www.chemspider.com/Chemical-Structure.2424.html">http://www.chemspider.com/Chemical-Structure.2424.html</a>	
21	pathway	Phenylalanine_map00360	72	KGID_C07086	<a href="http://www.genome.jp/dbget-bin/www_bget?cpd:C07086">http://www.genome.jp/dbget-bin/www_bget?cpd:C07086</a>	
22				KGID_C00156	<a href="http://www.genome.jp/dbget-bin/www_bget?cpd:C00156">http://www.genome.jp/dbget-bin/www_bget?cpd:C00156</a>	
23				KGID_C02137	<a href="http://www.genome.jp/dbget-bin/www_bget?cpd:C02137">http://www.genome.jp/dbget-bin/www_bget?cpd:C02137</a>	
24				KGID_C19975	<a href="http://www.genome.jp/dbget-bin/www_bget?cpd:C19975">http://www.genome.jp/dbget-bin/www_bget?cpd:C19975</a>	
25				KGID_C06866	<a href="http://www.genome.jp/dbget-bin/www_bget?cpd:C06866">http://www.genome.jp/dbget-bin/www_bget?cpd:C06866</a>	
26				KGID_C00596	<a href="http://www.genome.jp/dbget-bin/www_bget?cpd:C00596">http://www.genome.jp/dbget-bin/www_bget?cpd:C00596</a>	
27				KGID_C00079	<a href="http://www.genome.jp/dbget-bin/www_bget?cpd:C00079">http://www.genome.jp/dbget-bin/www_bget?cpd:C00079</a>	
28				KGID_C15524	<a href="http://www.genome.jp/dbget-bin/www_bget?cpd:C15524">http://www.genome.jp/dbget-bin/www_bget?cpd:C15524</a>	
29				KGID_C00084	<a href="http://www.genome.jp/dbget-bin/www_bget?cpd:C00084">http://www.genome.jp/dbget-bin/www_bget?cpd:C00084</a>	
30				KGID_C04479	<a href="http://www.genome.jp/dbget-bin/www_bget?cpd:C04479">http://www.genome.jp/dbget-bin/www_bget?cpd:C04479</a>	
31				KGID_C11457	<a href="http://www.genome.jp/dbget-bin/www_bget?cpd:C11457">http://www.genome.jp/dbget-bin/www_bget?cpd:C11457</a>	

## 3.2 Using the InSilicoFragmentation

### 3.2.1 Before Starting

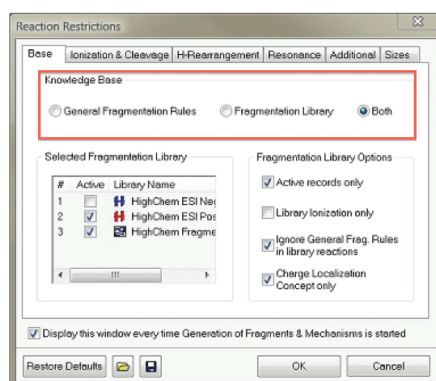
- Make sure no popup dialogs or notifications show up (e.g. Dropbox notification messages). It will interfere with the process and block the screen capturing.
- It is recommended to quit all other software besides Sikuli and Mass Frontier to avoid interference.
- Place all the target .mol files (MDL MOL format) in the same folder. Note that the Unicode paths are not supported.
- Open Mass Frontier. Change parameter settings if needed (e.g. change the *in-silico* fragmentation parameters). The InSilicoFragmentation will not modify the parameters for users.



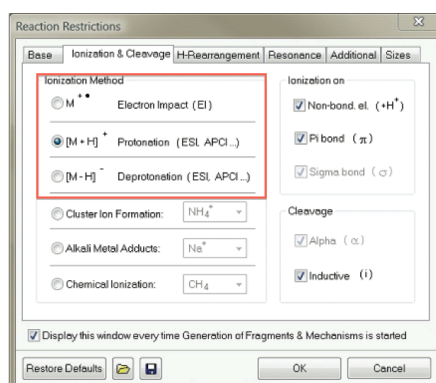
### 3.2.2 Parameter Settings for Mass Frontier

The fragmentation results of Mass Frontier highly depends on the reaction restrictions settings. Crucial settings include:

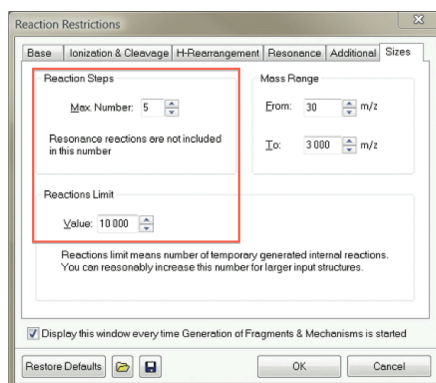
1. **Knowledge Base:** decide whether the fragmentation rules and / or fragmentation libraries should be used. The first option predicts reaction pathways based on general fragmentation and rearrangement rules. The second option accesses an intelligent fragmentation mechanism knowledge base collected from scientific publications and real experiments to predict the molecule decomposition reactions. Mass Frontier produces much more fragments when using the fragmentation libraries. Meanwhile the computational time is significantly increased.



2. **Ionization Method:** decide whether the target molecule should be ionized in electron impact (EI), positive or negative Electrospray Ionization (ESI)/Atmospheric Pressure Chemical Ionization (APCI) mode. Fragments produced in EI mode are not yet supported in HAMMER (may cause unexpected error). Although the Mass Frontier supports both ESI/APCI positive and negative ionization, much fewer fragments will be generated in the negative mode (i.e. based largely upon fragmentation rules since there is a lack of information in the literature on negative mode fragmentation).



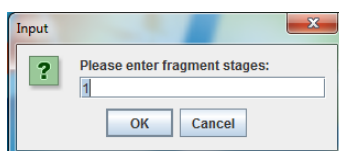
3. **Reaction Steps & Reactions Limit:** the reaction steps number gives the maximum size of cascaded fragment reactions, i.e. the maximum reaction generations in each fragmentation rules / libraries prediction. The reactions limit defines the maximum total reaction steps in the entire fragmentation process. The number of fragments produced in a given reaction path increases exponentially as the number of reaction steps increases. To avoid excessive computational resources usage, the most important fragmentation reactions are conducted first, and are followed by lesser important fragmentation rules until the reactions limit is reached. In HAMMER we suggest the users to set the reactions limit to its maximum value, i.e. 20,000.



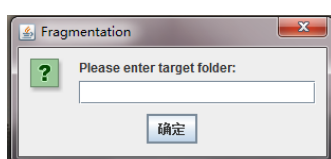
Other settings may cause minor changes to the fragmentation results. See Mass Frontier User Guide, Chapter 6, the Reaction Restrictions section for more details.

### 3.2.3 Perform Fragmentation

1. Double-click "InSilicoFragmentation.cmd".
2. Enter fragmentation stages. A positive integer value is expected here.

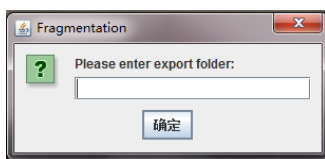


3. Enter the path of the target folder.

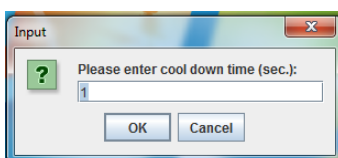


4. Enter the path of the export folder. If the export folder does not exist, the InSilicoFragmentation will create it.





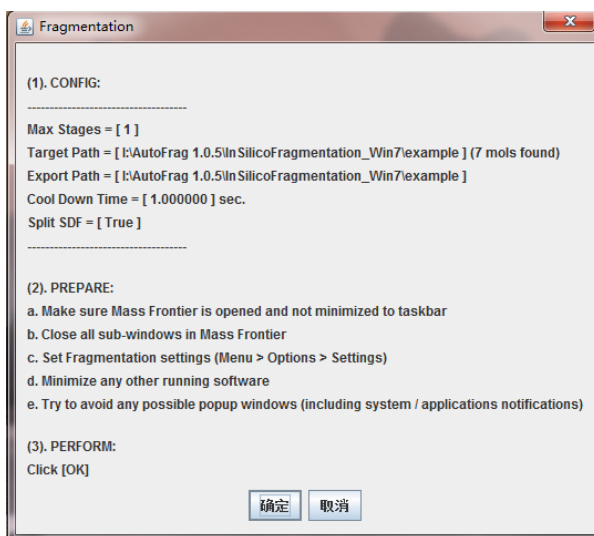
5. Enter cool down time. This is the time interval between two operations in Mass Frontier. More cool down time is needed for large compounds to ensure stability of the software. A positive float value is expected here.



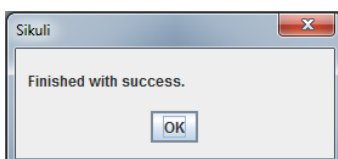
6. Select whether the exported .sdf file should be split into separate .mol files.



7. Before starting, a message box pops up to show current settings and requirements. Click "OK" to continue.



8. The InSilicoFragmentation will wait 5 seconds to allow users to bring the Mass Frontier window on top of all other open software windows. The *in-silico* fragmentation will start soon after that. Do not operate the computer when the InSilicoFragmentation is still running, otherwise the process will be interrupted.
9. A message box popups when the fragmentation process is finished. Click "OK" to quit InSilicoFragmentation.



### 3.2.4 Results

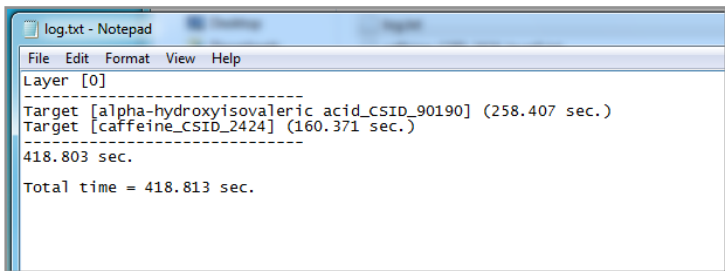
Result files are placed in the previously defined export folder. These files are exported for each compound:

- **%COMPOUND\_NAME%\_{layer%LAYER\_ID%.txt**: unique  $m/z$  values exported from the list of fragments. The %COMPOUND\_NAME% is the filename of the target compound; %LAYER\_ID% is the number of fragmentation stages. For multi-stage fragmentation, the filename contains more than one %LAYER\_ID% parts separated by %FRAGMENT\_ID% (see the 3rd point).

For instance, file “Caffeine\_layer0.txt” is the  $m/z$  values exported from “Caffeine.mol” on the first stage of fragmentation, and file “Caffeine\_layer0\_frag0\_layer1.txt” is the  $m/z$  values from the MS/MS fragmentation (“layer1”) whereas its precursor is the first fragment (“frag0”) generated by the MS fragmentation (“layer0”) of “Caffeine.mol”.

- **%COMPOUND\_NAME%\_{layer%LAYER\_ID%.sdf**: *in-silico* fragments results generated by Mass Frontier. The fragment structures are stored in .sdf (structure-data file) format.
- **%COMPOUND\_NAME%\_{layer%LAYER\_ID%.mcr**: *in-silico* fragmentation mechanisms and results generated by Mass Frontier.
- **%COMPOUND\_NAME%\_{layer%LAYER\_ID%\_frag%FRAGMENT\_ID%.mol (Optional)**: structure of each fragment split from the .sdf file. These files are generated only when the “Is-split” option (step 7 in the above section) is set to “True”.

An overall log file “log.txt” that contains the fragmentation time of each compound and the total process time is also provided. Following is an example of the log file.



```
log.txt - Notepad
File Edit Format View Help
Layer [0]
-----
Target [alpha-hydroxyisovaleric acid_CSID_90190] (258.407 sec.)
Target [caffeine_CSID_2424] (160.371 sec.)
-----
418.803 sec.
Total time = 418.813 sec.
```

## 3.3 Using the OrganiseFragments

### 3.3.1 Input Data

The OrganiseFragments utilizes the InSilicoFragmentation results as its input data, and processes the .mol files according to the %COMPOUND\_NAME%, %LAYER\_ID%, and %FRAGMENT\_ID% in their filenames. To perform OrganiseFragments, the “Split SDF” option (step 6 in section 3.2.2) must be set to “Yes”.

All the fragments must be stored in the same folder. The folder can contain fragments exported from more than one compound. For the results generated by the InSilicoFragmentation, the export folder should be provided.

### 3.3.2 Console Mode

In console mode, the OrganiseFragments is used as a single line command:

```
> OrganiseFragments.exe [--options] targets [export]
```

where “targets” is the path to the target folder that contains all the .mol fragments, and “export” is the path to the folder wherein the results are going to be stored. If an export folder is not provided, the results will be saved in the same folder as “targets”. Options of the OrganiseFragments are:

1. **image**: generate 2D image for each fragment.
2. **mzxml**: generate mzXML file for each target compound.
3. **cml**: generate chemical markup language (CML) file for each target compound.
4. **misp**: generate NIST MSP file for each target compound.
5. **mzlist**: generate plain text peak list for each target compound.
6. **plist**: generate property list structure for each target compound.
7. **tree**: generate tree visualization for each target compound.
8. **help**: display the software help.

Here is an example to use OrganiseFragments in console mode:

```
> OrganiseFragments.exe --image --mzxml --msp results\ results_org\
```

It functions to search all the .mol fragments in “results\” folder, organize them according to the precursor, generate 2D images, generate mzXML and MSP files, and store the results in “results\_org\” folder.

### 3.3.3 Interactive Mode

Besides the console mode, OrganiseFragments can also be run in the interactive mode. Double-click “OrganiseFragments.exe” in windows explorer to enter this mode, where the parameters are set in an interactive manner.

```
> OrganiseFragments.exe
```

Enter the path of the target folder:

```
Target path >>
```

Enter the path of the export folder. If users want to store the results in the original target folder, leave it empty and press enter:

```
Export path >>
```

Decide whether the 2D images should be generated:

Generate 2D images? (y/n, enter = n) >>

Decide whether the mzXML files should be generated:

Generate mzXML files? (y/n, enter = n) >>

Decide whether the chemical markup language (CML) files should be generated:

Generate CML files? (y/n, enter = n) >>

Decide whether the NIST MSP files should be generated:

Generate MSP files? (y/n, enter = n) >>

Decide whether the property list structures should be generated:

Generate plist files? (y/n, enter = n) >>

Decide whether the tree views should be generated:

Generate tree views? (y/n, enter = n) >>

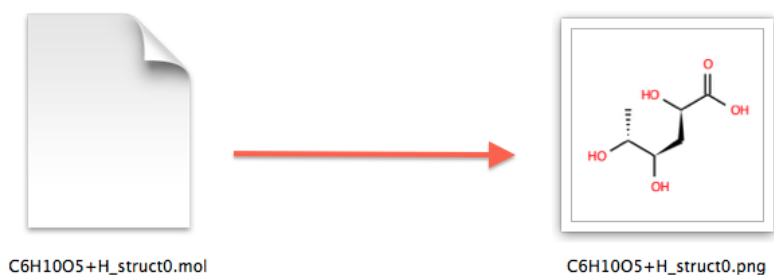
Decide whether the plain text peak list (mzList) should be generated:

Generate mzList files? (y/n, enter = n) >>

Press enter and the OrganiseFragments will start processing.

### 3.3.4 Results

The 2D image (in Portable Network Graphics format) can be exported for each *in-silico* fragment under the same filename as the original .mol file and with the extension .png.



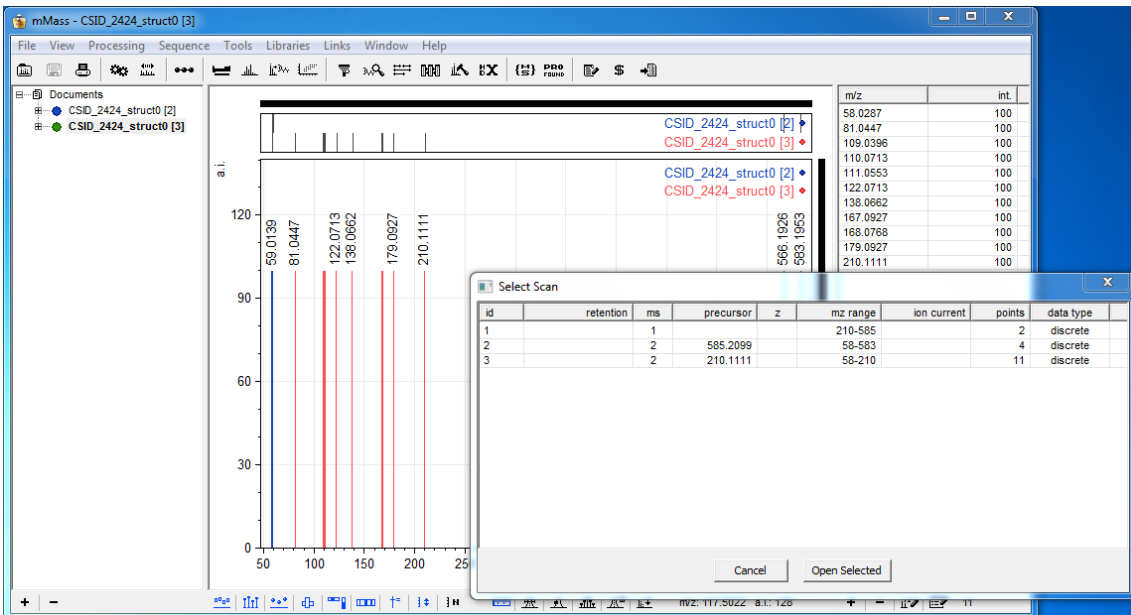
The mzXML file can be exported for each target compound containing the encoded spectral information. These files are compatible with common mass spectrometry software e.g. mMass, Mass++, and MetiTree.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<msRun>
  <parentFile fileName="example/out_org/CSID_2424_struct0/CSID_2424_struct0.mol" fileType="processedData" fileSha1="8c666bc9fa4f95966027ee4d946424199b8eb042"/>
  <dataProcessing centroided="1">
    <software type="processing" name="HAMMER" version="v1.0.0"/>
  </dataProcessing>
  <scan num="1" msLevel="1" peaksCount="1" polarity="+" lowMz="195.087652" highMz="195.087652">
    <peaks precision="32" byteOrder="network" contentType="m/z-int" compressionType="none" compressedLen="0">Q0MWcELIAAA=</peaks>
  </scan>
  <scan num="2" msLevel="2" peaksCount="11" polarity="+" lowMz="58.028740" highMz="195.087652">
    <precursorMz precursorScanNum="1">195.087652</precursorMz>
    <peaks precision="32" byteOrder="network" contentType="m/z-int" compressionType="none" compressedLen="0">QmgdbkLIAABCOhbmQsgAAELaFExCvAAAQtwkfkLIAABC3hxPQsgAAEL0JH5CyAAAQwoQ8kLIAABDJxe+QsgAAEMoE6ZCyAAAQzM=</peaks>
  </scan>
  <scan num="3" msLevel="3" peaksCount="4" polarity="+" lowMz="83.060375" highMz="138.066188">
    <precursorMz precursorScanNum="2">138.066188</precursorMz>
    <peaks precision="32" byteOrder="network" contentType="m/z-int" compressionType="none" compressedLen="0">QqYe6ULIAABC3CR+QsgAAELeHE9CyAAAQwoQ8kLIAAA=</peaks>
  </scan>
  <scan num="4" msLevel="3" peaksCount="1" polarity="+" lowMz="83.060375" highMz="83.060375">
    <precursorMz precursorScanNum="2">111.055289</precursorMz>
    <peaks precision="32" byteOrder="network" contentType="m/z-int" compressionType="none" compressedLen="0">QqYe6ULIAAA=</peaks>
  </scan>
  <scan num="5" msLevel="3" peaksCount="2" polarity="+" lowMz="122.071274" highMz="179.092738">
    <precursorMz precursorScanNum="2">179.092738</precursorMz>
    <peaks precision="32" byteOrder="network" contentType="m/z-int" compressionType="none" compressedLen="0">QvQkfkLIAABDMxe+QsgAAA=</peaks>
  </scan>
  <scan num="6" msLevel="3" peaksCount="1" polarity="+" lowMz="81.044725" highMz="81.044725">
    <precursorMz precursorScanNum="2">81.044725</precursorMz>
    <peaks precision="32" byteOrder="network" contentType="m/z-int" compressionType="none" compressedLen="0">QvQkfkLIAABDMxe+QsgAAA=</peaks>
  </scan>
</msRun>

```



The CML file can be exported for each target compound containing the spectral, fragment formulas, and fragmentation mechanisms. These files are compatible with common chemical software e.g. MetiTree.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<cml xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  ▼<spectrumList xmlns="http://www.xml-cml.org/schema">
    ▼<metadataList>
      <metadata dictRef="nmc:countScans" content="11"/>
      <metadata dictRef="nmc:parentFile" content="example/out_org/CSID_2424_struct0/CSID_2424_struct0.mzxml" convention="mzXMLData"/>
      <metadata dictRef="nmc:elements" content="C8...8,H10...10,N4...4,O2...2,H1...1"/>
      <metadata dictRef="nmc:parentCompounds" content="InChI=1S/C8H10N4O2/c1-10-4-9-6-5(10)7(13)12(3)8(14)11(6)2/h4H,1-3H3"/>
    </metadataList>
    ▼<spectrum type="MS" id="1">
      ▼<conditionList>
        <scalar dictRef="nmc>window" dataType="xsd:string">1</scalar>
        <scalar dictRef="nmc:polarity" dataType="xsd:string">positive</scalar>
      </conditionList>
      ▼<metadataList>
        <metadata dictRef="nmc:scanNum" content="1"/>
        <metadata dictRef="nmc:msLevel" content="1"/>
      </metadataList>
      ▼<peakList>
        <peak id="1" xValue="195.087652" yValue="100.0"/>
      </peakList>
    </spectrum>
    ▶<spectrum type="MS" id="2">...</spectrum>
    ▶<spectrum type="MS" id="3">...</spectrum>
    ▶<spectrum type="MS" id="4">...</spectrum>
    ▶<spectrum type="MS" id="5">...</spectrum>
    ▶<spectrum type="MS" id="6">...</spectrum>
    ▶<spectrum type="MS" id="7">...</spectrum>
    ▶<spectrum type="MS" id="8">...</spectrum>
    ▶<spectrum type="MS" id="9">...</spectrum>
    ▶<spectrum type="MS" id="10">...</spectrum>
    ▶<spectrum type="MS" id="11">...</spectrum>
  </spectrumList>
  ▼<moleculeList xmlns="http://www.xml-cml.org/schema" convention="cdk:moleculeSet">
    ▼<molecule id="1">
      ▼<formula concise="C 8 H 10 N 4 O 2 H 1">
        <atomArray elementType="C H N O H" count="8 10 4 2 1"/>
      </formula>
    </molecule>
  </moleculeList>
</cml>
```

The MSP file can be exported for each target compound containing the spectra and formulas of the precursor and its *in-silico* fragments. These files are compatible with common MS software and web services e.g. LipidBlast MSSearch and MassBank.

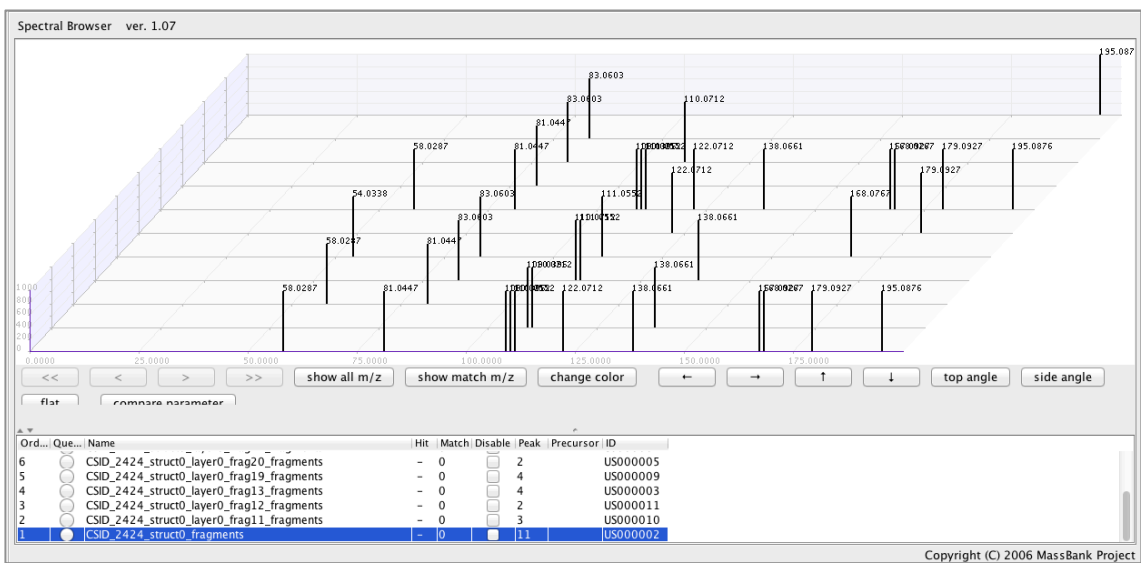
```

Name: Precursor
Comment: MS1
Num Peaks: 1
195.087652 100.0 "C8H10N4O2+H"

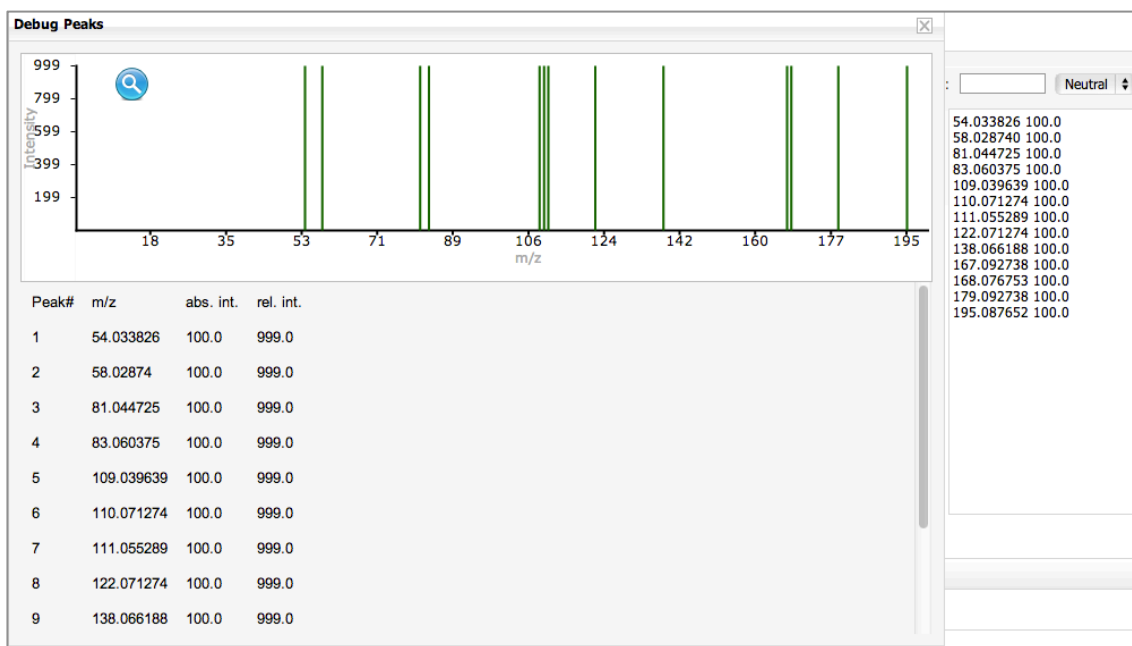
Name: CSID_2424_struct0_fragments
PRECURSORMZ: 195.087652
Comment: MS2
Num Peaks: 11
58.028740 100.0 "C2H4N0+"
81.044725 100.0 "C4H5N2+"
109.039639 100.0 "C5H5N2O+"
110.071274 100.0 "C5H8N3+"
111.055289 100.0 "C5H7N2O+"
122.071274 100.0 "C6H8N3+"
138.066188 100.0 "C6H8N3O+"
167.092738 100.0 "C7H11N4O+"
168.076753 100.0 "C7H10N3O2+"
179.092738 100.0 "C8H11N4O+"
195.087652 100.0 "C8H11N4O2+"

Name: CSID_2424_struct0_layer0_frag13_fragments
PRECURSORMZ: 138.066188
Comment: MS3
Num Peaks: 4
83.060375 100.0 "C4H7N2+"
110.071274 100.0 "C5H8N3+"
111.055289 100.0 "C5H7N2O+"
138.066188 100.0 "C6H8N3O+"

```



The plain text peak list file (mzList) can be exported for each target compound containing the *m/z*-intensity fingerprints. These files are compatible with common MS software and web services e.g. mMass, Mass++, MassBank, and MetFrag.

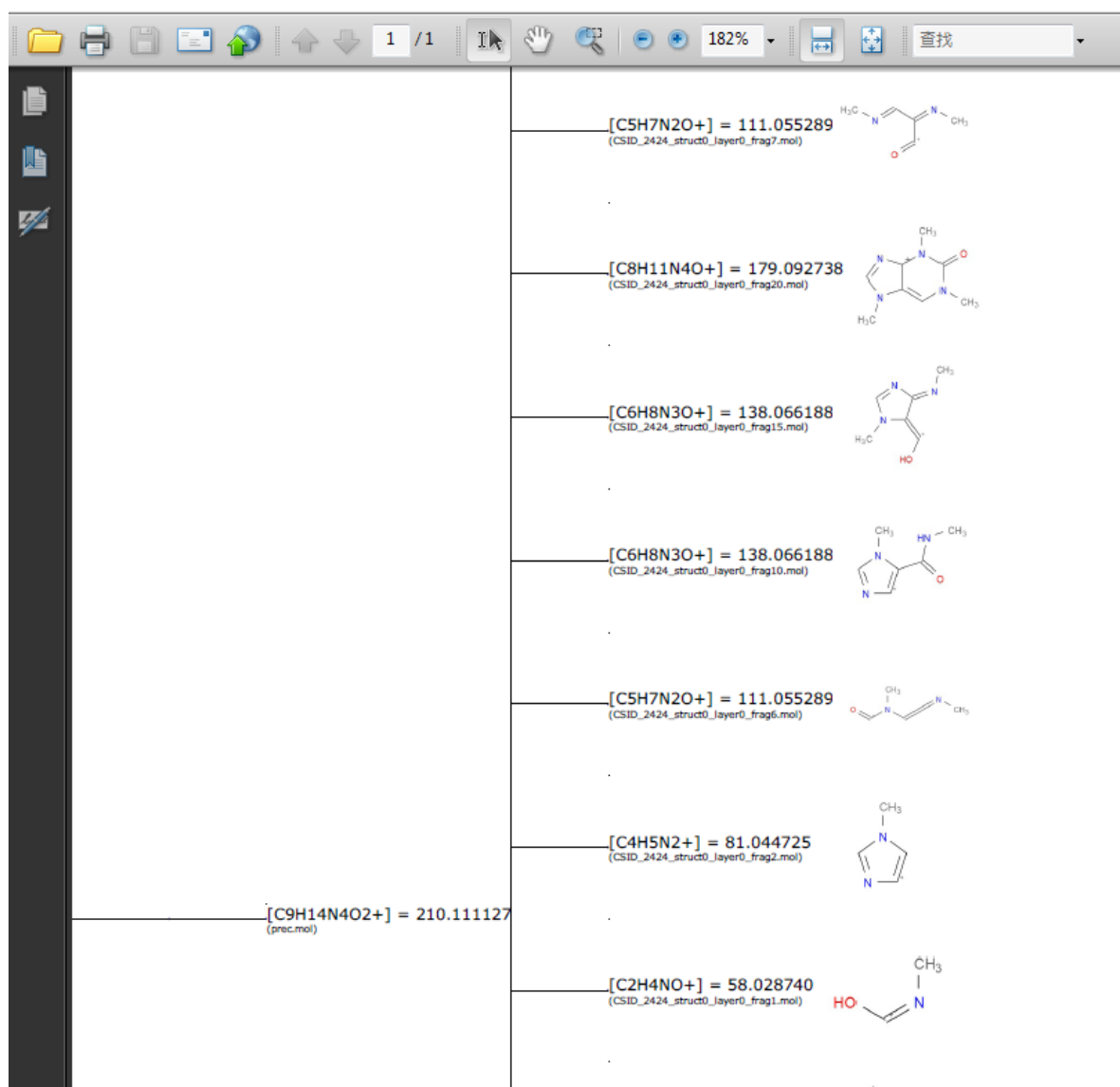




The property list structure can be exported for each target compound containing the spectral, mechanisms, and metadata of the fragments including the exact mass, formula, InChi code, SMILES code, and corresponding .mol file name. Plist is a special xml data type that can be accessed directly in Python. Users can also open it in any xml editor.

```
This XML file does not appear to have any style information associated with it. The document tree is shown below.
▼<plist version="1.0">
  ▼<dict>
    <key>CSID_2424_struct0.mol</key>
    ▼<dict>
      <key>fragments</key>
      ▼<dict>
        <key>CSID_2424_struct0_layer0_frag0.mol</key>
        ▼<dict>
          <key>fragments</key>
          <dict></dict>
          <key>metadata</key>
          ▼<dict>
            <key>exact_mass</key>
            <real>58.02874021809075</real>
            <key>filename</key>
            ▼<string>
              /Users/030bug/Repository/HAMMER/example/out_org/CSID_2424_struct0/CSID_2424_struct0_layer0_frag0.mol
            </string>
            <key>formula</key>
            <string>C2H4NO</string>
            <key>inchi</key>
            <string>InChI=1S/C2H4NO/c1-3-2-4/h1H3,(H,3,4)/q+1</string>
            <key>smiles</key>
            <string>CN[C+] =O</string>
          </dict>
        </dict>
        <key>CSID_2424_struct0_layer0_frag1.mol</key>
        ▶<dict>...</dict>
        <key>CSID_2424_struct0_layer0_frag10.mol</key>
        ▶<dict>...</dict>
        <key>CSID_2424_struct0_layer0_frag11.mol</key>
        ▶<dict>...</dict>
        <key>CSID_2424_struct0_layer0_frag12.mol</key>
        ▶<dict>...</dict>
        <key>CSID_2424_struct0_layer0_frag13.mol</key>
        ▶<dict>...</dict>
        <key>CSID_2424_struct0_layer0_frag14.mol</key>
        ▼<dict>
```

The tree visualization can also be exported for each compound. It is stored in .pdf format and provides a more intuitive way to display the fragmentation mechanisms.



## 3.4 Using the SpectralMatching

### 3.4.1 *In-silico* Library & Input Real Spectral

The SpectralMatching utilizes MSP files to constitute *in-silico* libraries. MSP files contain spectral fingerprints and formulas for both precursor and *in-silico* fragments. They can also be utilized in other spectral matching software e.g. LipidBlast MSSearch and NIST MSPepSearch.

In SpectralMatching each MSP file is exported from a unique target compound, and the *in-silico* library is a folder containing all the MSP files. Users can add or remove compounds easily by moving the corresponding MSP file inside or outside the library folder.

The SpectralMatching accepts real spectra in MSP or plain text peak list (using extension .peaks) format. The target real spectra files should be put into the same folder in order to conduct batch matching. If a real spectra does not contain *m/z* values for the precursor (first peak in MSP files and last peak in plain text files as default), SpectralMatching will ask the users to assign one:

```
Please provide precursor exact mass for [Caffeine HCD 80_pos.peaks] >>
```

MatchSpectral utilize a modified version of the pMatch algorithm for spectral matching and scoring. Precursor *m/z* values are crucial for proper matching in pMatch. For more details, please read the supplementary information and visit the pMatch algorithm website: <http://pfind.ict.ac.cn/pmatch/>.

### 3.4.2 Console Mode

In console mode, the SpectralMatching is used as a single line command:

```
> SpectralMatching.exe [--options] database targets
```

where “database” is the path to the *in-silico* library, and “targets” is the path to the target folder that contains all of the real spectra. Options of the SpectralMatching are:

1. **abs**: measure distance in absolute value.
2. **tp**=: peaks matching range, default value = 100 (ppm).
3. **ift**=: intensity filter threshold, default = none (no limit).
4. **ifn**=: intensity filter maximum peaks number, default = all (no limit).
5. **pth**=: preprocessing peaks per 100 *m/z*, default = all (no limit).
6. **cpt**=: capital peaks intensity threshold, default = 0.05.
7. **cpn**=: maximum capital peaks number, default = all (no limit).
8. **max**=: maximum match candidates number, default = all (no limit).
9. **help**: display the software help.

Here is an example to use SpectralMatching in console mode:

```
> SpectralMatching.exe --tp 10 --ifn 100 --max 5 mzdb\ spectral\
```

It functions to perform batch matching on real spectral files in “spectral\” folder against the *in-silico* library “mzdb\”, using maximum *m/z* matching distance = 10 ppm, using the top 100 real peaks with the highest intensity for matching, and export the top 5 matching results with the highest scores for each real spectra.

### 3.4.3 Interactive Mode

Besides the console mode, SpectralMatching can also be run in the interactive mode. Double-click "SpectralMatching.exe" in the windows explorer to enter this mode, where the parameters are set in an interactive manner.

```
> SpectralMatching.exe
```

Enter the path to the *in-silico* library:

```
Database path >>
```

Enter the path to the real spectral target folder:

```
Target path >>
```

Decide the matching range:

```
Matching range (enter = 100.0) >>
```

Decide the intensity filter threshold:

```
Intensity filter threshold (enter = none) >>
```

Decide the intensity filter maximum peaks number:

```
Intensity filter peaks number (enter = all) >>
```

Decide the maximum preprocess peaks per 100Th:

```
Preprocessing peaks (enter = all) >>
```

Decide the capital peaks intensity threshold:

```
Capital peaks intensity threshold (enter = 0.05) >>
```

Decide the maximum capital peaks number:

```
Maximum capital peaks number (enter = all) >>
```

Decide the maximum match candidates number:

```
Maximum candidates number (enter = all) >>
```

Decide whether the *m/z* distance should be measured as an absolute value:

```
Measure distance in absolute value? (y/n, enter = n) >>
```

Press enter and the SpectralMatching will start processing.

### 3.4.4 Results

Overall scores and the best matching *in-silico* compound will be displayed:

```
Results:
```

```
Input spectrum: real/Caffeine HCD 80_pos.msp (101 peaks)
```

```
Results:
```

Score	Candidates	Formula (Peaks)	Explained	Distance
0.45712	CSID_2424_struct0	[C8H10N4O2+H] <sup>+</sup> (13)	8 (8.1%)	19.984

```
Best Match:
```

Entry: CSID\_2424\_struct0 ([C8H10N4O2+H]+ ... 13 peaks)

Score: 0.45712

Rel. Intensity	Experimental (m/z)	In-silico (m/z)	Formula	Distance (in ppm)
100.000	138.06664	138.06619	[C6H7N3O+H]+	3.274e+00
29.818	110.07142	110.07127	[C5H7N3+H]+	1.326e+00
9.548	83.06017	83.06037	[C4H6N2+H]+	2.468e+00
9.028	195.08854	195.08765	[C8H10N4O2+H]+	4.552e+00
4.463	109.03980	109.03964	[C5H4N2O+H]+	1.477e+00
2.180	111.05544	111.05529	[C5H6N2O+H]+	1.360e+00
0.824	81.04452	81.04472	[C4H4N2+H]+	2.529e+00
0.042	122.07164	122.07127	[C6H7N3+H]+	2.998e+00

report saved to file [real/Caffeine HCD 80\_pos\_match\_results.txt]

The number of *in-silico* compounds displayed in the “Overall Results” section is determined by the “max=” parameter. The explained precursor formula, explained peaks number and percentage, and the total *m/z* distance are also provided. In the “Best Match” section, the explained peaks are listed in the order of relative intensity, as well as the explained fragment formulas and match distances.

A report file containing all the matching results will be exported for each target real spectra under the filename **%SPEATRAL\_NAME%\_match\_results.txt**, wherein **%SPEATRAL\_NAME%** is the filename of the original spectra. Contents in the report file are plain text separated with tabs “\t”. The report can be opened in Excel or any text editors.

## **Chapter 4 - Update History and Troubleshooting**

### **4.1 Update History**

### **4.2 Troubleshooting**