Journal of Systemics, Cybernetics and Informatics (2022) 20(5), 138-147
https://doi.org/10.54808/JSCI.20.05.138

Metabolite Fragmentation Visualization

Myungjae Kwak[1], Matthew Molina[1], Spencer Arnold[2], Andrew Woodward[1],
Jin-Young An[1], Estelle Nuckels[3], and * Yingfeng Wang[2]
[1]Department of Information Technology, Middle Georgia State University, Macon, GA 31206, USA
[2]Department of Computer Science and Engineering, University of Tennessee at Chattanooga, Chattanooga, TN 37403, USA
[3]Department of Natural Sciences, Middle Georgia State University, Macon, GA 31206, USA

Abstract[1]

Tandem mass spectrometry (MS/MS) is a popular technology for identifying small molecules involved in metabolism, better known as metabolites. Coupled with liquid chromatography (LC), LC-MS/MS instruments first separate, ionize, and fragment metabolites, then measure mass-to-charge ratios (m/z) and intensities of metabolite fragments. Understanding metabolite fragmentation is crucial to develop computational tools for identifying metabolites based on this spectroscopic data. Metabolite fragmentation patterns have large variations making it especially difficult for computer scientists to design and implement metabolite identification approaches. To address this interdisciplinary challenge, this article presents FragView, a web-based application providing the web service for visualizing metabolite fragmentation. Users can break chemical bonds to produce metabolite fragments and export 3D fragment structures for 3D printing. Developing FragView is an opportunity for exposing student participants to this interdisciplinary bioinformatics project. This paper summarizes the experience of training student participants in bootcamps and designing the implementation plan based on student backgrounds. Students were exposed to project meeting discussions on coding and raw data visualization and visited a lab with an LC-MS/MS instrument. FragView is an open source, freely accessible tool, released under the GPLv3 license. We will continue to improve and update FragView in the future based on feedback.

Keywords: *Tandem mass spectrometry, Metabolite Fragmentation, Web Server, Visualization and 3D Printing.*

* Corresponding Author. Email address: yingfeng-wang@utc.edu.
[1] The authors acknowledge Dr. Hong Qin who made the peer-editor of the final proofreading of the article.

1. Introduction

The rapidly expanding field of metabolomics has far-reaching implications from biochemistry to medicine, and areas in between. Accurately detecting, identifying, and quantifying the small molecules, better known as metabolites, that are found in biological samples such as cells, tissues, and body fluids, has provided significant supports to plant, microbial, and mammalian studies over the past two decades (Alseekh et al., 2021; Alseekh & Fernie, 2018; Chaleckis et al., 2019). Detecting metabolites using liquid chromatography (LC) coupled to tandem mass spectrometry (MS/MS) has become a common analytical technology for metabolite identification (Cao et al., 2021; Li et al., 2020; Stancliffe et al., 2021; F. Wang et al., 2021). LC-MS/MS instruments separate, ionize, and fragment metabolites. The resulting mass-to-charge ratio and intensity data of metabolite fragments are used for metabolite identification.

Large variations in MS/MS fragmentation make identifying metabolites challenging (Dührkop et al., 2015). After ionization and potential rearrangement, nearly any chemical bond may break in LC-MS/MS instruments. Although most metabolites are small, the number of possible fragments of a metabolite is usually large. Understanding metabolite fragmentation with the LC-MS/MS technology is necessary for developing metabolite identification tools. Connecting biology, chemistry, and computer science, this interdisciplinary work looks at predicting molecular fragments and building the relationship between fragments.

In this paper, we present FragView for visualizing metabolite fragmentation. FragView is a web server, which is a web-based application providing the web service. The FragView project is an undergraduate project for exposing undergraduate developers to this interdisciplinary bioinformatics area. Faculty mentors guide undergraduate participants to study biological, chemical, and computer science topics related to this project. Students with different backgrounds contribute, developing FragView based on their expertise and interdisciplinary collaboration. Furthermore, FragView is a useful tool for researchers in identification of metabolites. As an educational web server, FragView visualizes 2D and 3D structures of the original metabolite. Users can generate and analyze fragments by using FragView's fragmentation tree feature. FragView can also export individual STL format files of the metabolite and subsequent fragments for 3D printing.

The rest of the article introduces MolView and fragmentation trees in section 2. Next, we summarize our experience of developing FragView in section 3, followed by a description of FragView features in section 4. Then, how to access FragView and its source code is detailed in section 5, finishing with a conclusion and future work in

section 6.

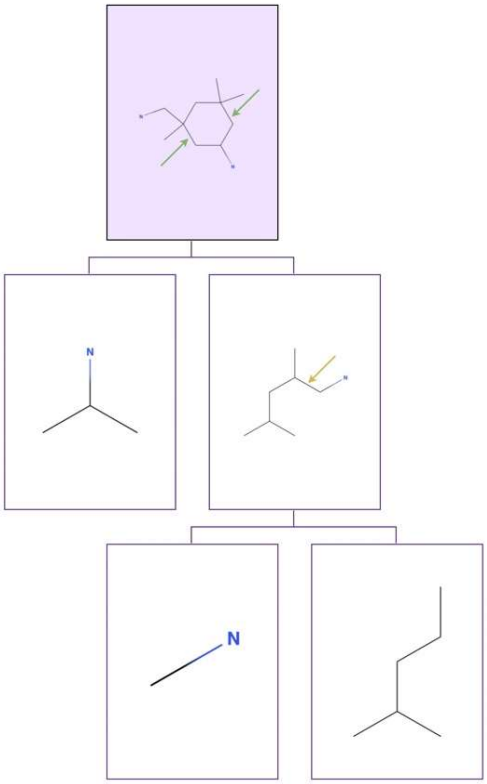

Figure 1: A sample fragmentation tree generated by FragView. The node with the purple background is the root. Breaking the root's two ring bonds, indicated by two green arrows, generates two fragments represented by the nodes of the middle layer. Breaking the bond indicated by the yellow arrow generates two fragments represented by the nodes of the bottom layer.

2. Related Works

2.1. MolView

FragView evolved from the code base of MolView (Bergwerf, 2015). MolView is an open-source web server for visualizing chemical compound structures using the SMILES input. MolView generates 2D and 3D structures of compounds and supports structure editing. However, MolView was not designed for visualizing metabolite fragmentation and did not organize fragments. Therefore, we adopted some MolView features and implemented new features in FragView for metabolite fragmentation visualization.

2.2. Fragmentation Tree

A fragmentation tree is a tree-like graph organizing fragments of a metabolite. Böcker and Rasche proposed the fragmentation tree for representing the relationships between fragments of a metabolite (Böcker & Rasche, 2008). It has been applied in multiple metabolite identification tools such as MetFrag (Wolf et al., 2010), MIDAS (Y. Wang et al., 2014), and CSI:FingerID (Dührkop et al., 2015). The root of a fragmentation tree is the original metabolite. Fragments are formed by breaking one linear bond or two ring bonds. These fragments are represented by the child nodes of the root node containing the original metabolite. More fragments can be generated from the child node fragments in a similar fashion and are represented by another set of child nodes beneath the previous fragment. A sample fragmentation tree is given in Figure 1. The root node, whose background is purple, contains the 2D structure of the original metabolite. When the two bonds denoted by green arrows are removed, two new fragments have been created and the child nodes in the middle layer are generated. Similarly, deleting the bond indicated by the yellow arrow in the right fragment, two more fragments form, as shown by the nodes in the bottom layer.

3. Project Development

3.1. Student Training

Most undergraduate student participants in this interdisciplinary project have a computer science background, while some also have biology and/or chemistry backgrounds. However, none of these students had the experience of processing MS/MS data. The students with biology and chemistry backgrounds had basic knowledge with chemical compounds but had limited to no experience with LC-MS/MS, computer generated chemical visualization methods and the programming the visualization tools were based upon. The students with computer science background had experience with programming tools but had limited to no experience with any chemical spectroscopy.

To address this issue, we trained students by two approaches. One was to host bootcamps for introducing fundamental knowledge of chemical compound representation, LC-MS/MS technology, and MS/MS data. The other was to go to Mercer University to visit a wet lab equipped with an LC-MS/MS instrument. These activities helped the students have the basic understanding of the LC-MS/MS technology and its data.

3.2. Research Design

Student backgrounds had a strong influence on the FragView project. Most computer science students involved in the project were familiar with Python and JavaScript, so FragView is based on these two computer languages. MolView's code base is primarily implemented in JavaScript and already incorporates features needed for FragView, so it was an ideal starting place. Improvements were made to the existing code to include additional features.

The existing code base only allowed users to input compound structures with the SMILES format. To make FragView more convenient, the team decided to add the InChI format as an additional structure input format. MolView can export 3D structures as images, but these images cannot be used on 3D printers. The team added the feature of exporting STL format files for 3D printing. Since the final FragView product would include multiple fragments of the same molecule, the team also added the feature to save the file, including any generated fragments, to resume visualization in the future. To streamline FragView, MolView's features that were not closely related to metabolite fragmentation were made invisible.

The second part of the plan was to add fragmentation visualization. Besides the existing panels for visualizing 2D and 3D structures, FragView has a third panel for visualizing the fragmentation tree, reflecting the relationship between fragments. FragView assumes the full valence shell of a fragment by adding new hydrogen bonds with the yellow color.

3.3. Trouble Shooting

FragView's implementation was not smooth. While unexpected problems slowed the web server development, it allowed undergraduate students to practice problem-solving skills with real-world problems. Initially, students were encouraged to go through the details to identify the exact reason for an issue, achieved by code review and testing. After locating the code segment triggering the problem, students searched for possible solutions in the literature or Internet and implemented the best option.

A good example was the integration of the Jmol library with FragView. Unfortunately, some of Jmol's features did not work as expected. Students were guided to search whether another library or another Jmol version could resolve the issue, or if they could devise their own solution. We led students to estimate the workload of implementing any devised solutions. If the workload is relatively low, the team should implement the solutions. Otherwise, the team should first figure out a workaround, then add the implementation plans for decent solutions to the future

work with suitable priorities.

When designing problem-solving strategies, we suggested the students to prepare alternative strategies. For instance, FragView used the RDKitJS library, the JavaScript version of RDKit (Landrum, 2013). Although RDKitJS provided most of the features FragView needed, we still planned to use RDKit Python if necessary.

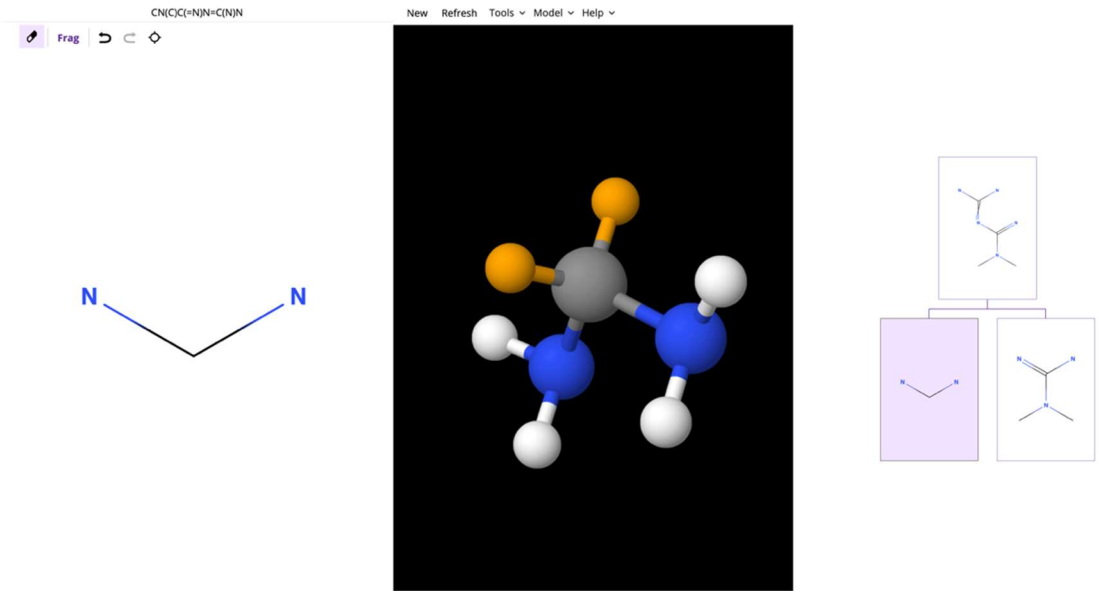

Figure 2: The FragView interface for a sample compound. Users can input a compound structure with the SMILES or InChI format in the upper-left corner of the interface. The compound is displayed on three panels. The left and middle panels draw the 2D and 3D structures, respectively. The right panel contains the fragmentation tree.

4. Program Features

FragView is a web server for visualizing metabolite fragmentation. Its interface for a sample compound is given in Figure 2. Users can input a compound structure with the SMILES or InChI format in the upper-left corner of the interface. After the "New" button is pushed, the compound is displayed on three panels. The left and middle panels draw the 2D and 3D structures, respectively. The right panel contains the fragmentation tree. When the initial structure is drawn in the left and middle panels, only the root node is shown in the right panel. When bonds are removed in the left panel and the "Frag" button is pushed, the child nodes are created in the right panel. Each fragment will have its own node under the root node. If they are fragmented

further, more child nodes will appear, like in Figure 1.

The default 2D mode draws molecular structures in line-angle formula. Line-angle formula is a form of chemistry shorthand where all bonds are drawn using lines connecting atoms together, where a single line represents a single bond, two lines represent a double bond, and three for a triple bond. As carbon and hydrogen are extremely common, especially in metabolites, they are not explicitly written. Any end or bend in a line that has an atomic symbol is that element, otherwise it is assumed to be a carbon atom. The molecule is also assumed to have a full valence shell unless denoted otherwise, so any bonds that are not explicitly written are bonds to hydrogen. Users can generate new fragments by erasing bonds on the panel and pushing the "Frag" button.

The 3D panel draws the ball and stick model of the molecule or fragment. FragView uses different colored balls to distinguish elements, and sticks to denote the bonds. When a fragment is generated, its atoms associated with the breaking bonds are either radicals, charged, or both. The Jmol library automatically assumes a full valence shell and adds hydrogen bonds to all atoms associated with the breaking bonds. For example, there are two yellow atoms on the middle panel of Figure 2, indicating the two new hydrogen bonds. We developed an algorithm for identifying these hydrogen atoms as not part of the original molecule. This algorithm considers each molecular fragment as a simple graph whose vertices and edges refer to atoms and bonds, respectively. When new fragments are generated, hydrogen atoms are added to fill the valence shell of the molecule. This algorithm compares the graphs representing the parent fragment and the new fragments for identifying the broken bonds and the newly added hydrogen atoms associated with these bonds. These new fragments may not be neutral, thus appearing in LC-MS/MS spectra. This algorithm traces the change of hydrogen atoms in the structure, coloring all newly added hydrogen atoms yellow. The yellow atoms denote the location of broken bonds.

The root of the fragmentation tree refers to the original metabolite. When users generate fragments by erasing bonds, the nodes representing new fragments will be added as child nodes under the original fragment or metabolite. When users select the node on the right panel, the 2D and 3D structures represented by the selected node are drawn on the left and middle panels, respectively.

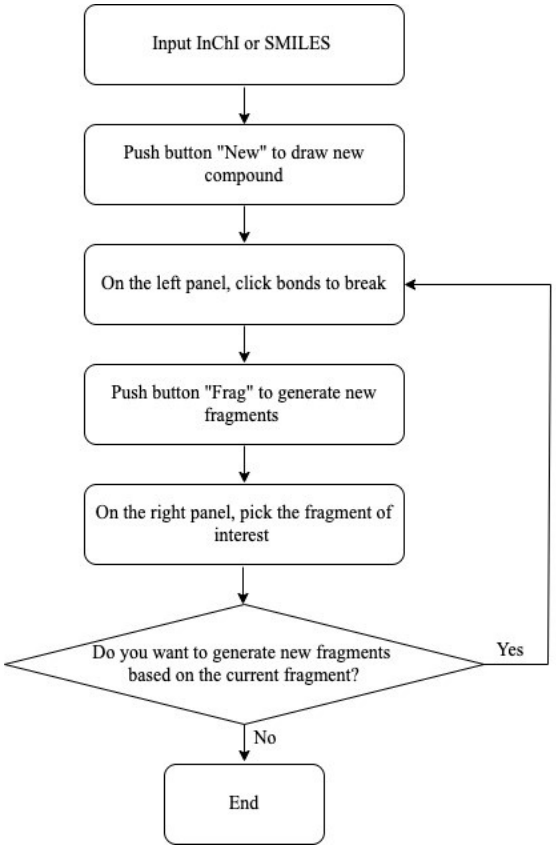

Figure 3: The flowchart for fragmenting and visualizing a compound on FragView.

The above FragView features make compound fragmentation and visualization convenient. Users can quickly fragment a compound and visualize fragments within several steps. Figure 3 gives the flowchart for fragmentation and visualization.

5. Web Server and Source Code Access

Users can watch the tutorial video and read the online manual provided in FragView. FragView is an open-source, freely accessible tool released under the GPLv3 license. The links to FragView and its source code are available at https://www.yingfengwang.org/fragview/.

6. Conclusion and Future Work

We describe FragView, an educational web server for metabolite visualization. This platform can help students understand metabolite fragmentation in LC-

MS/MS instruments. We also summarize our experience of engaging undergraduate students during the development of FragView. We plan to collect users' feedback and update FragView in the future.

11. Acknowledgments:

This work was partially supported by the National Science Foundation under Grant Number 2053286. The authors thank Dr. Kevin Murnane for hosting the visit to his lab's LC-MS/MS instrument at Mercer University. The authors thank Dr. Yu Liang for reviewing this paper as the non-anonymous peer reviewer and Dr. Lani Gao for proofreading the paper as the beta-editor. The authors also thank Mr. Michael Koohang and Mr. JohnMichael Nichols for participating in this study.

References

Alseekh, S., Aharoni, A., Brotman, Y., Contrepois, K., Auria, J. D., Ewald, J., Ewald, J. C., Fraser, P. D., Giavalisco, P., Hall, R. D., Souza, L. P. De, Saito, K., Sauer, U., Schroeder, F. C., Schuster, S., Siuzdak, G., Skirycz, A., Sumner, L. W., Snyder, M. P., … Fernie, A. R. (2021). Mass spectrometry-based metabolomics: A guide for annotation, quantification and best reporting practices. *Nature Methods*, *18*(7), 747–756. https://doi.org/10.1038/s41592-021-01197-1

Alseekh, S., & Fernie, A. R. (2018). Metabolomics 20 years on: what have we learned and what hurdles remain? *Plant Journal*, *94*(6), 933–942. https://doi.org/10.1111/tpj.13950

Bergwerf, H. (2015). MolView: An attempt to get the cloud into chemistry classrooms. *DivCHED CCCE: Committee on Computers in Chemical Education*, *9*, 1–9.

Böcker, S., & Rasche, F. (2008). Towards de novo identification of metabolites by analyzing tandem mass spectra. *Bioinformatics*, *24*(16), 49–55. https://doi.org/10.1093/bioinformatics/btn270

Cao, L., Guler, M., Tagirdzhanov, A., Lee, Y. Y., Gurevich, A., & Mohimani, H. (2021). MolDiscovery: Learning mass spectrometry fragmentation of small molecules. *Nature Communications*, *12*(1). https://doi.org/10.1038/s41467-021-23986-0

Chaleckis, R., Meister, I., Zhang, P., & Wheelock, C. E. (2019). Challenges, progress and promises of metabolite annotation for LC–MS-based metabolomics. *Current Opinion in Biotechnology*, *55*, 44–50. https://doi.org/10.1016/j.copbio.2018.07.010

Dührkop, K., Shen, H., Meusel, M., Rousu, J., & Böcker, S. (2015). Searching molecular structure databases with tandem mass spectra using CSI:FingerID. *Proceedings of the National Academy of Sciences*, *112*(41), 12580–12585. https://doi.org/10.1073/pnas.1509788112

Landrum, G. (2013). *RDKit: Open-source cheminformatics*. http://www.rdkit.org

Li, Y., Kuhn, M., Gavin, A. C., & Bork, P. (2020). Identification of metabolites from tandem mass spectra with a machine learning approach utilizing structural features. *Bioinformatics*, *36*(4), 1213–1218. https://doi.org/10.1093/bioinformatics/btz736

Stancliffe, E., Schwaiger-haber, M., Sindelar, M., & Patti, G. J. (2021). DecoID improves identification rates in metabolomics through database-assisted MS/MS deconvolution. *Nature Methods*, *18*(7), 779–787. https://doi.org/10.1038/s41592-021-01195-3

Wang, F., Liigand, J., Tian, S., Arndt, D., Greiner, R., & Wishart, D. S. (2021). CFM-ID 4.0: More accurate ESI-MS/MS spectral prediction and compound identification. *Analytical Chemistry*, *93*(34), 11692–

11700. https://doi.org/10.1021/acs.analchem.1c01465
Wang, Y., Kora, G., Bowen, B. P., & Pan, C. (2014). MIDAS: A database-searching algorithm for metabolite identification in metabolomics. *Analytical Chemistry*, *86*(19), 9496–9503.
Wolf, S., Schmidt, S., Müller-Hannemann, M., & Neumann, S. (2010). In silico fragmentation for computer assisted identification of metabolite mass spectra. *BMC Bioinformatics*, *11*, 148. https://doi.org/10.1186/1471-2105-11-148