doi:10.2533/chimia.2025.66 Chimia 79 (2025) 66–69

Unlocking the Potential of Browser-Based Scientific Data Analysis: A 20-Year Journey of Expertise

Luc Patiny*

Abstract: The browser has become an indispensable tool for a variety of everyday tasks, yet its potential in scientific data processing remains underexplored and is often perceived as slow. This paper presents four examples of advanced web applications that we have developed during the last 20 years and demonstrates the browser's ability to compete with traditionally installed software. These applications were made possible through the development of over 100 open-source libraries, extending the FAIR (Findable, Accessible, Interoperable, and Reusable) principles to include not only data but also software.

Keywords: FAIR · Mass spectrometry · NMR · Structure search · Web application

With a PhD in organic chemistry, ***Luc Patiny*** has always been interested in computer programming and has made significant contributions to the field of cheminformatics and chemical education. He has authored over 50 peer-reviewed publications and contributed to over 200 open-source projects. He is currently teaching and managing chemical information at EPFL and is the Chief Scientific Officer at Zakodium. His pioneering work in using web technology to provide intuitive and useful tools for chemists has been invaluable in both education and research.

1. Introduction

In the modern digital age, a significant portion of our daily activities and tasks have migrated online. From sending emails to creating and storing documents on platforms like Google Docs and OneDrive, the browser has become a central hub for a wide array of functions. However, in the context of scientific research, the transition to browser-based operations is not yet complete. Daily analyses, which often require complex processing, still necessitate the installation of specific software for tasks such as image processing, mass spectra analysis, and NMR spectra analysis, although this trend is evolving rapidly.[1]

My interest in the potential of the browser as a platform for scientific data processing and analysis dates back to the previous century. My colleagues and I have long recognized the browser's capacity as the most efficient way to share and process scientific data. Over 10 years ago, we developed various web applications, such as ecoscale.cheminfo.org,[2] www.chemcalc.org,[3] wikipedia.cheminfo.org,[4] and www.nmrdb.org,[5] which have proven that complex scientific data processing can be effectively performed online, directly within a browser.

In this paper, I briefly explain how we tackled various challenges in creating web applications for scientific data manipulation and present examples that are available for free online.

*Correspondence: Dr. L. Patiny, E-mail: luc.patiny@zakodium.com
Zakodium Sàrl, CH-1027 Lonay

2. Building the Foundation for Online Data Analysis Software

Historically, the scarcity of online data analysis software can likely be attributed to the absence of the libraries required to construct chemistry-oriented web applications in pure java-script. These building blocks include format conversion (such as JCAMP-DX, mzML, NetCDF, TIFF, PNG), mathematical analysis (including matrix manipulation, vector similarity, Principal Component Analysis (PCA), Hierarchical Clustering, *etc.*), and more specific libraries dealing for example with mass spectrometry, nuclear magnetic resonance, and chemical structures.

Recognizing this need, I have worked with more than 100 programmers over the last 15 years to develop numerous open-source libraries, ensuring they are easily maintainable and reusable across various projects. This effort has resulted in the creation of over 200 projects, distributed across the GitHub organizations cheminfo,[6] mljs,[7] image-js,[8] and zakodium-oss.[9] These libraries were initially developed to meet our specific requirements for creating scientific web applications, but over the years, they have proven useful to many developers worldwide. For instance, npm libraries like ml-matrix[10] and ml-distance[11] are currently downloaded over 1 million times per month.

This observation underscores that developing small, reusable modules, rather than a large monolithic application, offers significant benefits to the scientific community. It extends the FAIR principles beyond data to include libraries that are Findable, Accessible, Interoperable, and Reusable. Thanks to the efforts of numerous contributors, the development of online data analysis software has become significantly more accessible and efficient.

3. Why the Browser?

The primary reason for using the browser is its simplicity. It is already installed on most devices, so users only need to navigate to a specific URL to access the application. This makes it cross-platform, supporting operating systems including Windows, Linux, macOS, Android, and iOS. Additionally, users can always access the latest version of the application by simply refreshing the webpage.

An often-overlooked advantage is the ease with which the browser enables the integration of databases and other systems. For example, it becomes feasible to use an electronic laboratory

notebook online and process NMR spectra in another web application, all without the need to install any software.

From a development perspective, the browser offers significant advantages in terms of graphical performance and user experience. Modern web technologies and frameworks allow developers to build sophisticated, high-performance applications with rich user interfaces.

There are also indirect benefits. The browser is arguably the most scrutinized application from a security standpoint, ensuring that users stay up to date with the latest security patches. Furthermore, competition among major browsers, such as Edge, Chrome, Firefox, and Safari, drives continuous improvements in speed and performance, benefiting both developers and users. Finally, from an IT perspective, using the browser reduces maintenance costs, by eliminating the need for complex software installations or updates.

4. Four Examples Highlighting Challenges

4.1 *Structure Search Engine in the Browser*

The wikipedia.cheminfo.org[4] project demonstrates some of the current capabilities of modern web browsers. It allows users to perform substructure searches across all molecules described in Wikipedia (Fig. 1). To ensure the sustainability and simplicity of this project, it was designed as a self-contained application that does not rely on dedicated infrastructure. By utilizing GitHub Actions, the system automates its workflow: every night, it crawls Wikipedia to extract SMILES (Simplified Molecular Input Line Entry System) notations from articles about molecules. These notations are compiled into a JSON-encoded text file, creating a lightweight yet robust database. This file, along with a user-friendly graphical interface, is then deployed *via* Cloudflare and made publicly accessible at https://wikipedia.cheminfo.org. When visiting the website, the entire database (over 20,000 structures) is loaded into memory. Visitors can then perform substructure searches, executed directly in the browser, thanks to the JavaScript version of OpenChemLib.[12]

4.2 *Scientific Image Analysis*

The analysis of scientific images presents unique challenges. These images often have 16 bits per channel and typically require the identification and analysis of regions of interest. Processing them in a browser is not straightforward, as formats like PNG are converted to 8 bits by default. Additionally, the TIFF format, commonly used in scientific applications, is not natively supported by browsers. For these reasons, we developed several libraries to decode TIFF[13] and PNG[14] images while preserving full precision, and we created a new library, image-js,[15] designed specifically for scientific image processing.

The image-js library is a powerful tool for processing scientific images in the browser. It is designed to handle complex images with 16 bits per channel and supports working with regions of interest, which is crucial in various scientific and research contexts, such as counting and determining the size of particles. To enhance accessibility, an interactive tool has been developed that allows users to analyse images and measure particle sizes directly within the browser. This tool is available for testing at https://tem.cheminfo.org (Fig. 2).

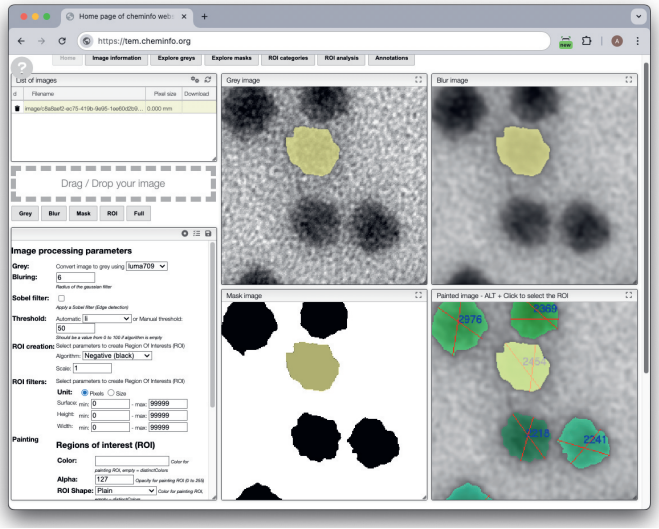

Fig. 2. https://tem.cheminfo.org process images locally in the browser.

4.3 *High Resolution Mass Spectrometry*

High-resolution mass spectra in profile mode may contain over 700,000 data points, requiring peak picking to identify the centroids. This first challenge can be addressed on-the-fly during spectrum loading using an algorithm called Global Spectra Deconvolution, for which we developed the library ml-gsd.[16] Another problem involves calculating isotopic distributions, which may include thousands of centroids in complex molecules like antibodies, requiring precise calculations and optimised algorithms. Other challenges include the analysis of complex mixtures containing hundreds of molecules, such as in the case of automatic identification of chlorinated paraffins (CPs) or polymers. Over the years, we have developed numerous algorithms to solve the problems that emerge and they have been combined into a single repository called mass-tools.[17]

As an example, the goal of the project CP-Hunter is to automatically identify by mass spectrometry CPs and their transformation products. Based on a list of possible molecular formula the system will calculate the corresponding isotopic distribution and determine the contribution of each molecular formula using non-negative matrix factorisation.[18]

CP-Hunter, like all the other examples, is a web-based platform where no data leaves your browser. This not only allows for the processing of confidential information but also provides a better user experience. The platform is accessible at https://cphunter.cheminfo.org[1] (Fig. 3).

4.4 *Nuclear Magnetic Resonance*

This last example presents another challenge we faced in scientific web applications: the presence of complex user interac-

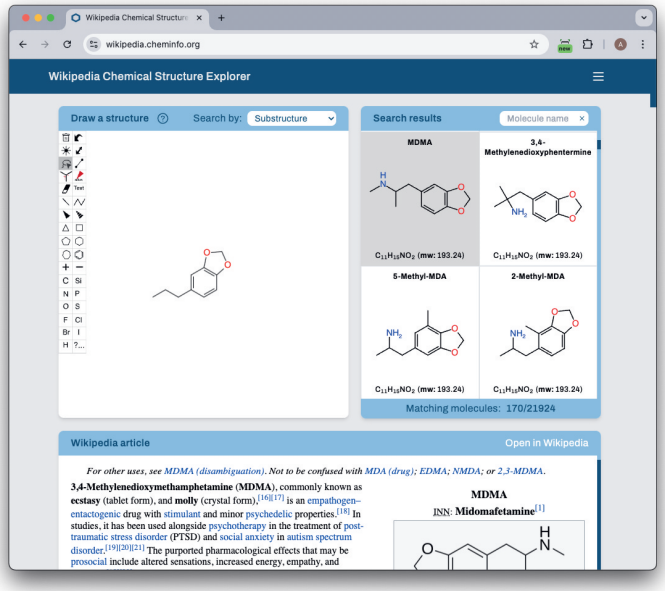

Fig. 1. Substructure and similarity searches of molecules present in Wikipedia.

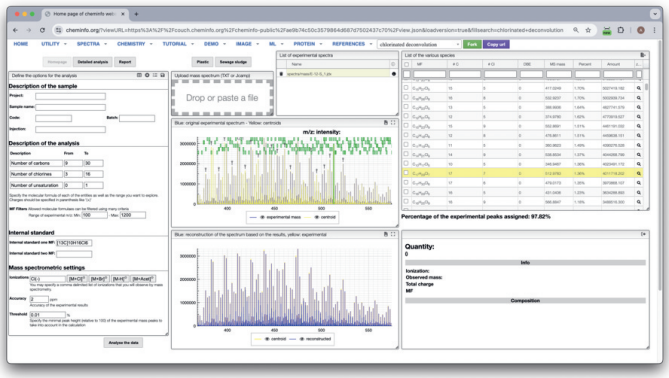

Fig. 3. CP-Hunter: automatic identification of chlorinated paraffins (CPs) from high-resolution mass spectra.

tions. This is especially true for NMR processing, where we often deal with large datasets (*e.g.* 100 metabolomics 1D spectra) or complex datasets involving both 1D and 2D spectra. Moreover, many processing algorithms need to be supported, such as apodization, Fourier transform, baseline correction, peak picking, multiplet analysis, *etc.*

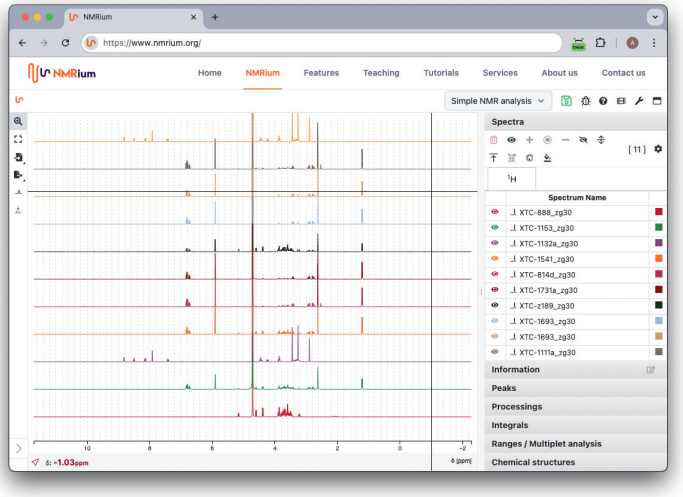

Fig. 5. Process and superimpose NMR spectra in the browser on www.nmrium.org.

Fig. 4. Hierarchy of dependencies in NMRium.

Based on over 80 libraries (Fig. 4) that we previously developed, we created the project NMRium[19] (Fig. 5). NMRium is an innovative web application that allows the processing of Nuclear Magnetic Resonance (NMR) spectra without the need for software installation and interaction with the server. Additionally, the application offers assignment capabilities and multiple spectra analysis, making it a comprehensive tool for NMR data analysis.

NMRium also provides a range of educational resources, including prediction, simulation, and quizzes[19] designed to assist in the understanding and assignment of NMR spectra.

5. Conclusions

In conclusion, the potential of browsers as platforms for scientific data processing and analysis has been demonstrated through the development and application of FAIR libraries. These libraries have enabled the creation of complex web applications that can handle intricate scientific data and require complex interactions.

Our experience suggests that there are no significant technical limitations preventing the browser from becoming the only application installed on a computer. This marks a return to the concept of the computer as a 'simple' terminal, similar to its role 60 years ago.

Furthermore, the utilization of web applications facilitates access to FAIR data stored in repositories like Zenodo. Users can readily access live data, such as 1D and 2D NMR spectra, enabling tasks like phase correction without the need for additional software.[20] This underscores the 'A'ccessibility aspect of FAIR, as data can now be reached with just a simple click.

Acknowledgements

I would like to thank the more than 100 contributors across our GitHub organizations for their dedication in making these complex applications possible. Special thanks go to Norman Pellet, Michaël Zasso, and Daniel Kostro for their exceptional contributions.

Received: November 27, 2024

[1] L. A. Abriata, *Informatics* **2017**, *4*, 28, https://doi.org/10.3390/informatics4030028.
[2] K. Van Aken, L. Strekowski, L. Patiny, *Beilstein J. Org. Chem.* **2006**, *2*, 3, https://doi.org/10.1186/1860-5397-2-3.
[3] L. Patiny, A. Borel, *J. Chem. Inf. Model.* **2013**, *53*, 1223, https://doi.org/10.1021/ci300563h.
[4] P. Ertl, L. Patiny, T. Sander, C. Rufener, M. Zasso, *J. Cheminf.* **2015**, *7*, 10, https://doi.org/10.1186/s13321-015-0061-y.
[5] D. Banfi, L. Patiny, *CHIMIA* **2008**, *62*, 280, https://doi.org/10.2533/chimia.2008.280.
[6] Cheminfo, https://github.com/cheminfo, accessed November 22, 2024.
[7] ml.js, https://github.com/mljs, accessed November 22, 2024.
[8] image-js, https://github.com/image-js, accessed November 22, 2024.
[9] Zakodium Open source, https://github.com/zakodium-oss, accessed November 22, 2024.
[10] ml-matrix, https://www.npmjs.com/package/ml-matrix, accessed November 22, 2024.
[11] ml-distance, https://www.npmjs.com/package/ml-distance, accessed November 22, 2024.
[12] M. Zasso, L. Patiny, T. Sander, C. Rufener, *Zenodo* **2023**, https://doi.org/10.5281/zenodo.8245520.
[13] image-js/tiff: TIFF image decoder written entirely in JavaScript, https://github.com/image-js/tiff, accessed November 25, 2024.
[14] image-js/fast-png: PNG image decoder and encoder written entirely in JavaScript, https://github.com/image-js/fast-png, accessed November 25, 2024.
[15] image-js/image-js: Image processing and manipulation in JavaScript, https://github.com/image-js/image-js, accessed November 25, 2024.
[16] mljs/global-spectral-deconvolution: Global Spectra Deconvolution + Peak optimizer, https://github.com/mljs/global-spectral-deconvolution, accessed November 26, 2024.
[17] L. Patiny, M. Zasso, R. Silvestre, *Zenodo* **2024**, https://doi.org/10.5281/zenodo.14185379.
[18] M. Zasso, L. Patiny, S. Miranda, *Zenodo* **2024**, https://doi.org/10.5281/zenodo.8189402.
[19] A. Davies, L. Patiny, *Spectrosc. Europe* **2021**, *21*, https://doi.org/10.1255/sew.2021.a18.
[20] L. Patiny, *Zenodo* **2024**, https://doi.org/10.5281/zenodo.13307304.