Wageningen University

Department of Bioinformatics

FERMO Online: A web application to identify promising molecules in mass spectrometry data

Sarah Habicht

Supervisors:

Mitja M. Zdouc, Justin J.J. van der Hooft, Marnix H. Medema

June 30th, 2023

Table of Contents

List of Abbreviations

Cascading Style Sheets	CSS
Gas chromatography	GC
Graphical user interface	GUI
High performance liquid chromatography	HPLC
Hypertext Markup Language	HTML
Hypertext Transfer Protocol	HTTP
JavaScript	JS
Mass spectrometry	MS
Mass-to-charge	m/z
Natural product research	NPR
Natural products	NP
Tandem mass spectrometry	MS/MS
Uniform Resource Locators	URL
Universal Spectrum Identifier	USI
Web Server Gateway Interface	WSGI

1 Abstract

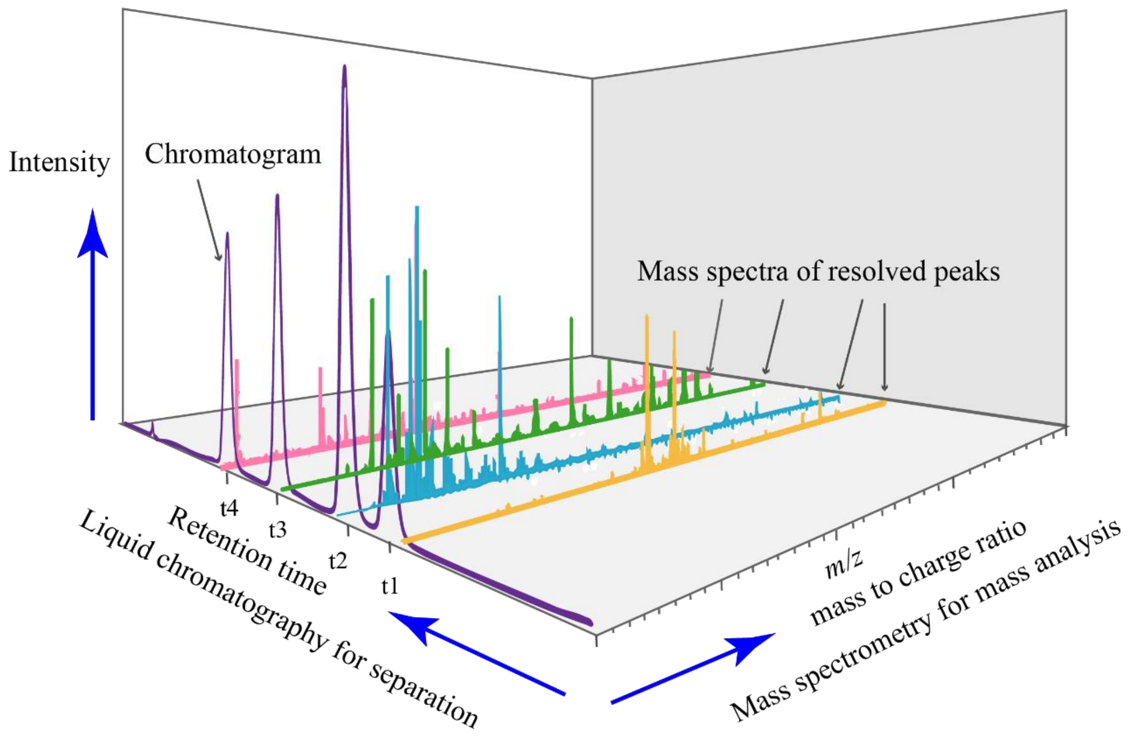
Mass spectrometry is a widely used technology for the analysis of biological samples. In drug discovery, it is often applied to facilitate the identification of thus far unknown molecules in biological extracts. However, each sample may contain thousands of molecular features, making it difficult to determine which ones warrant further examination. To address this issue, FERMO was recently developed. This tool allows researchers to prioritize mass spectrometric features by filtering based on custom scores and to visualize them in an interactive dashboard. However, the current version of FERMO employs a framework that limits the interactivity of the dashboard. Moreover, FERMO is only available as an offline tool and therefore necessitates installation. To eliminate these limitations, we are developing a web application based on a more flexible framework while incorporating additional changes to further improve the user experience. While the development of the complete web application is still ongoing, we have already revised the backend and redesigned the dashboard as well as added previously impossible interactivity. A fully functional demo of the new dashboard is available at: https://fermo.bioinformatics.nl/example. These improvements facilitate the prioritization of molecular features and thus not only contribute to drug discovery but to all research fields where characterization of mass spectrometric features is involved.

2 Introduction

Natural product research (NPR) is the study of secondary metabolites (also referred to as "specialized" metabolites) of living organisms like plants, fungi or bacteria. Identifying their chemical and biological properties aims to find pharmacologically relevant qualities which may be utilized to develop new drugs. In fact, NPR used to be the biggest source of active agents for medical drugs for a long time [1]. However, this research suffers from rediscovering already identified molecules which takes away resources from less futile approaches [2]. Moreover, in the early 2000's synthetic molecules became more promising to discover new active ingredients because the process from a natural extract to an active agent is slow, difficult and consequently expensive [3]. Therefore, many pharmaceutical companies withdrew their support for this research [4]. Nevertheless, natural products (NP) remain important since they can act as a source of inspiration for synthetic drugs [5] to an extent such that from 1984 to 2014, 64 % of all approved, small-molecule drugs were in some way derived from NP's [6].

Analysis of NP's most often involves a step of mass spectrometric analysis, after the individual compounds from a natural extract have been separated by high performance liquid chromatography (HPLC) or gas chromatography (GC). Mass spectrometry (MS) is a powerful technology that allows for identification of molecules by determining their mass-to-charge ratio (m/z). To measure this, individual molecules are ionized and their mass analyzed in one of various ways. One such method is time-of-flight, which utilizes the speed difference between ions with lower mass (faster) and ions with higher mass (slower) to calculate their m/z ratio. The resulting mass spectrum, plotted as a diagram of intensity (number of Ions) against m/z [7], depicts all molecules which were detected at the same retention time. Consequently, multiple mass spectra are generated to represent the entire retention period. By considering all peaks of the same m/z but across the different retention times, an extracted ion chromatogram, plotted as intensity against retention time, can be reconstructed (see Figure 1). Here, one peak is considered one feature and ideally corresponds to a distinct kind of ion. However, since different ions can have the same molecular weight, an additional step is often performed to differentiate between them: In tandem mass spectrometry (MS/MS) the ionized parent molecule is fragmented into smaller ions, for example by collision-induced dissociation or photodissociation. Analysis of the resulting fragments then allows conclusions about the structure of the parent ion. [8]

HPLC-MS can measure thousands of features in one single metabolite extract, producing large amounts of complex data [9]. This makes manual inspection of all features and selection of promising ones a time-consuming and ineffective endeavor. For drug discovery, however, only features that represent potentially new compounds and likely possess pharmaceutically interesting bioactivity are worth further examination. Tools and workflows have been developed to facilitate the identification of such features. For example, an analysis pipeline that was created by Zhou et al. connects different software tools, databases and algorithms to allow for more precise molecule identification [10], Nothias et al. introduced a workflow which links bioactivity scores to molecular networks [11] and Lee, et al. developed a web application to predict bioactive compounds from complex mixtures, called NP Analyst [12]. The recently developed software FERMO [13] is another notable solution. This software aims to facilitate the analysis of MS data by providing functionality to enter preprocessed MS or MS/MS spectra together with metadata like phylogenetic affiliation and phenotypic data such as biological activity. The data can then be filtered according to the user's needs, before custom scores, which



Intensity
Chromatogram
Mass spectra of resolved peaks
t4
t3
t2
t1
Retention time
Liquid chromatography for separation
m/z
mass to charge ratio
Mass spectrometry for mass analysis

Figure 1: Schematic illustration of how the extracted ion chromatograms (purple) and mass spectra (pink, green, blue and yellow) in LC-MS of one sample are interrelated. Created by Daniel Norena-Caro.

quantify aspects like novelty, specificity and diversity of the features, are calculated. After that, the information is displayed in an interactive dashboard which allows the user to easily inspect it and filter for features of interest.

What sets FERMO apart from most of these tools, in particular, is that it offers a graphical user interface (GUI) and therefore allows even researchers without extensive computational experience to visualize and prioritize their data effectively. However, FERMO operates fully locally. This ensures data security because it does not rely on a connection to the internet and no user data is logged. But it also means that the software has to be installed by the user. This can take time and may require troubleshooting if compatibility issues arise. Moreover, the computational performance of FERMO relies on the user's hardware. As a result, individuals with less powerful hardware or those working with particularly large datasets may experience longer processing times. Here, we present FERMO Online: A solution to the previously mentioned concerns by offering FERMO's functionality as a web application. To guide the development, we performed a preliminary user analysis and revised FERMO's underlying technology. Based on the results, we created the web application as an addition to the local version. Simultaneously, we incorporated beta-testing feedback and modified the layout, especially of the dashboard page. While the development of the full web application is still ongoing, a demo version, showcasing the new dashboard, is already available.

3 Technical Background

A web application is a software that can be accessed on the internet. While everything the user can see and interact with, is handled on the frontend, data processing is usually handled on the backend. The following paragraphs provide a brief explanation of the key concepts in web development.

3.1 Frontend

The frontend, also known as the client-side, comprises code which is interpreted and executed by a web browser. Hypertext Markup Language (HTML) is employed to define the basic structure of one webpage of the application by explicitly declaring different components such as headings, paragraphs, images etc. Additionally, HTML templates often contain basic content to be displayed upon first access. The appearance of the different elements is defined with Cascading Style Sheets (CSS) which can be facilitated by using a library with predefined CSS styles such as Bootstrap [14]. Finally, JavaScript (JS) can be applied to make websites interactive. It allows to define functions that handle user input like the click on a button. Frontend frameworks such as React.js [15] or Vue.js [16] can be used to simplify the development of dynamic web applications. They introduce certain ways how to build the software, thus warrant homogeneity and avoid code redundancy [17].

3.2 Backend

The backend, or server-side, encompasses code to handle user data. In a web application a user can often upload files or enter text manually. This input is then sent to the server and processed there. For the GNPS Dashboard [18] for example, a user can enter a Universal Spectrum Identifier (USI) which is transmitted to the server and compared to databases in order to retrieve the corresponding MS data. The results are returned to the client to be displayed in the browser.

A crucial step of backend-development is choosing an appropriate framework since it represents the foundation of a web application. Frameworks determine how the routing of Uniform Resource Locators (URL) is handled and enable the usage of databases. Moreover, they can facilitate the development of client-server-interaction or offer additional features like tools to avoid HTML code redundancy. Since the backend can be developed in any language, many different frameworks exist: examples include Express for JavaScript, and Flask for Python [19, 20]. Another noteworthy example is Dash, the Python-framework used for FERMO [21] because FERMO adopts a browser-based approach, similar to that of a web application, despite operating locally.

3.3 Client-Server Communication

Both front- and backend are essentially a collection of files with code or static data such as images. These files are hosted on a server with a connection to the internet. To access the content of the web application, users need to enter the URL, which specifies the server's location, into the address bar of a web browser. Following that, the browser sends a request to the server to ask for the content [22]. The request is a message that has to follow a set of rules, the Hypertext Transfer Protocol (HTTP), in order to be correctly understood by all involved technologies [23]. As depicted in Figure 2, the request passes through multiple proxies before it reaches the server. The first of these proxies is known as an HTTP server or simply web server.

HTTP servers, such as Nginx [24] and Apache [25], represent the connection between the client and the content the client tries to access. While these servers can store static content directly, most information has to be requested from an application which generates the response data dynamically and is sent via the web server back to the client [26]. Another proxy is required when the application is based on a language the HTTP server does not understand natively (e.g. Python or Ruby). For Python applications these proxies are Web Server Gateway Interface (WSGI) servers such as Gunicorn [27] and mod_wsgi [28]. They both convert incoming HTTP requests to the standard WSGI "environ" (a Python dictionary containing the environment, i.e. the HTTP request variables) and convert outgoing WSGI responses to HTTP responses [29]. WSGI servers often already include web server functionality. In those cases, dedicated web servers like Nginx are considered reverse proxies. These are nonetheless useful as they improve performance by distributing load among several servers and add a layer of security. [30]

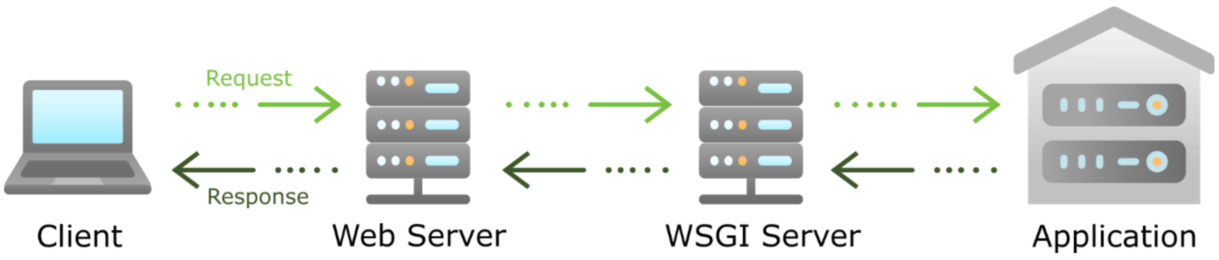


Figure 2: Model of the client-server communication.

4 Methods and Implementation

4.1 User Analysis

As a first step, we conducted a user analysis to set a well-reasoned foundation for the project. This is a means to anticipate users as well as situations those users might encounter while working with a given software. Here, the goals of the user analysis were to identify potential users of FERMO besides the initial target group of natural product researchers, what type of data they might use, which problems they could encounter and how those could be resolved. To answer these questions, we conducted a literature search, looking for papers which describe the analysis of MS data in different fields of research and with various workflows. Hypothetical problem-scenarios were developed based on general knowledge as well as drawing on input from colleagues of the department.

4.2 Framework Selection

As a second step, we established the foundation for the application itself by selecting a framework. For this, we created a list of potential frameworks by searching online for some of the most common web frameworks. Then, we began to compare these frameworks to each other based on five criteria.

Flexibility and maintainability were considered to ensure long-term viability of the application, allowing for potential future additions to the functionality as well as low effort for small code changes, such as bugfixes. Performance was a factor as users may upload large datasets, making a slower framework more noticeable. Furthermore, ease-of-use was taken into account, because a complex framework would be impractical to learn and implement within the limited timeframe of this project. To assess these four criteria, each framework was evaluated based on official documentation, testimonials of other developers, and personal experience. As the final factor, we considered popularity, judged by Github stars and forks, as an indication for available support and resources. Evaluation of each criterion was summarized in ratings of 'low', 'medium', or 'high'.

4.3 Code Design

4.3.1 Main Application

While developing the web application, we placed special emphasis on traceability as well as maintainability. Therefore, the web application was developed in an open source environment with version control on GitHub. Furthermore, the online version was conceptualized to extend, rather than replace the offline version. Consequently, new features that were developed in the context of the web application should also be transferable to the local application. In fact, there should only be one code foundation for both the online and offline version. The code should only differ between the two versions where absolutely necessary to keep the maintenance simple. For example, a database to store data of online users is not applicable to the offline version.

To test Python functions which were not already present in, or derived from the original application, we developed unit tests. These tests were invoked automatically to ensure continuous integration. For this automation we utilized GitHub Actions, a feature which allows to define workflows to be executed when a certain event occurs or on a defined schedule [31]. For the development of this web application, we configured the workflows to be triggered whenever changes were pushed to the repository.

However, unit testing was often unfeasible due to the dependence of many functions on the application context. For instance, functions that are executed in response to a user request require a request object which depends on the user's input e.g. whether the user clicked on a button or hovered over a tooltip.

In such cases, we performed manual integration testing instead. This involved manually accessing each route and triggering all expected events such as button clicks and hovering over specific elements.

Furthermore, we made efforts to adhere to best coding practices: All new functions were provided with extensive docstrings. Additionally, we added type-hinting to both new functions as well as those derived from the original application to ensure comprehensibility of the functions. Moreover, we maintained consistent styling according to the official style guide PEP 8 [32] by employing the linter Flake 8 [33]. Since frontend code, HTML and CSS, does not inherently offer ways to follow the DRY-principle ("Don't repeat yourself") [34], we utilized 'macros'. Macros are a tool that allows to reuse HTML code for recurring elements, provided by Flask's underlying templating engine Jinja2.

4.3.2 Demo

To deploy a demo-version of the application, we created a new branch of the GitHub repository (https://github.com/mmzdouc/FERMO/tree/fermo_flask_demo) which only contains the minimal necessary files. We removed code for pages other than the example page and changed the main route "/" to redirect to the example page "/example" automatically. Subsequently, we containerized the application using Docker and Docker-Compose [35] in order to avoid compatibility issues as well as introduce another layer of safety. Since Flask's own development server was not designed to handle a lot of traffic or take care of security concerns [29], Gunicorn and Nginx were chosen as the WSGI and webserver/reverse proxy, respectively.

We created two Dockerfiles: One containing instructions for the Docker container that executes the actual application with Gunicorn, and the other for the Nginx server container. The two containers are connected by a corresponding Docker-Compose file. For our demo, the relevant code is stored and the Docker-Compose file executed, on the 'Thornton' server of Wageningen University. After successfully launching the containers, the ports leading to the executing application were exposed to the public.

5 Results

5.1 User Analysis

To identify potential users and their use cases, we conducted a user analysis. This served to confirm whether a web application would be the most beneficial advancement of FERMO. The analysis showed that many individuals working with MS or MS/MS data of various small molecules from diverse backgrounds could benefit from FERMO (see Supplementary Table 1 and Supplementary Table 2). This includes natural product researchers, aiming to identify new compounds, but also environmental scientists seeking to monitor pollution. Also in the context of doping prevention FERMO could be applied. There, FERMO could support the identification of unknown metabolites from illicit drugs. These metabolites could henceforth be used to detect fraudulent athletes. In terms of computational experience, users may be bioinformatics experts as well as wet lab biologists or chemists with little experience regarding the different available software tools.

The data may derive from various sources, ranging from biological and environmental to synthetic and can encompass different types of molecules such as amino acids, sugars etc. Additionally, the data can be formatted differently depending on what preprocessing software was used. Also, the size of the dataset may vary in number of samples and number of features per sample. All these different aspects give rise to several possible scenarios, where the original version of FERMO reaches limitations. All corresponding examples are shown in Supplementary Table 3. For instance, users without access to powerful computers may encounter difficulties analyzing their data as FERMO gets slow when processing many samples and/or features. Another user may encounter issues during the installation of FERMO due to a lack of troubleshooting experience. These limitations are eliminated with a web application, therefore FERMO Online is a justified further development.

5.2 Framework Selection

After we established the justification for a web application for FERMO, we had to choose an appropriate framework which would dictate how to develop the online platform. As a first step, we identified React.js, Angular, Vue, Svelte, Django, Flask and Dash as the most popular/relevant frameworks [17, 29, 36-40]. However, due to time constraints of this project, conducting a comprehensive comparison of all these frameworks was deemed unfeasible (see Discussion for details). Therefore, we narrowed them down to the two most relevant ones: Dash, as it is the framework already used for the original FERMO and Flask because it was recommended by colleagues.

Dash is a very good solution to build dashboard applications for Python-developers who have no experience with HTML, CSS and JS, as it allows to work entirely in Python. To do so, Dash was built on top of and depends on Flask, React.js and Plotly.js. It offers many elements such as graphs or dropdown menus 'out of the box'. While creating custom elements is possible, it can be time consuming and more complex than creating them in HTML directly. Although Dash is popular among some scientists, the number of GitHub forks and stars (about 1900 and 18900, respectively on June, 29th, 2023 [41]) reveals, that it is far less used than Flask (15700 forks and 63400 stars on June 29th, 2023 [42]). Furthermore, the dependency on other technologies may negatively affect Dash's performance because code has to be processed on multiple levels (from Dash to Flask, to React, to JS).

Flask, in comparison, is a Python-based microframework. This implies, that it has minimal dependencies by default, allowing for flexible extensions as needed. Consequently, the entire frontend repertoire (HTML, CSS, JS) is available even though it requires manual integration. To facilitate frontend development, Flask could be utilized in combination with a frontend framework. Moreover, as mentioned before, Dash itself depends on Flask. Therefore, a significant portion of the functionality can be seamlessly transferred to the Flask-only application. Furthermore, the wide popularity among scientific applications, including some within our institute, ensures the availability of prompt support. Considering all these factors, as summarized in Figure 3, Flask emerges as the right choice for this project.

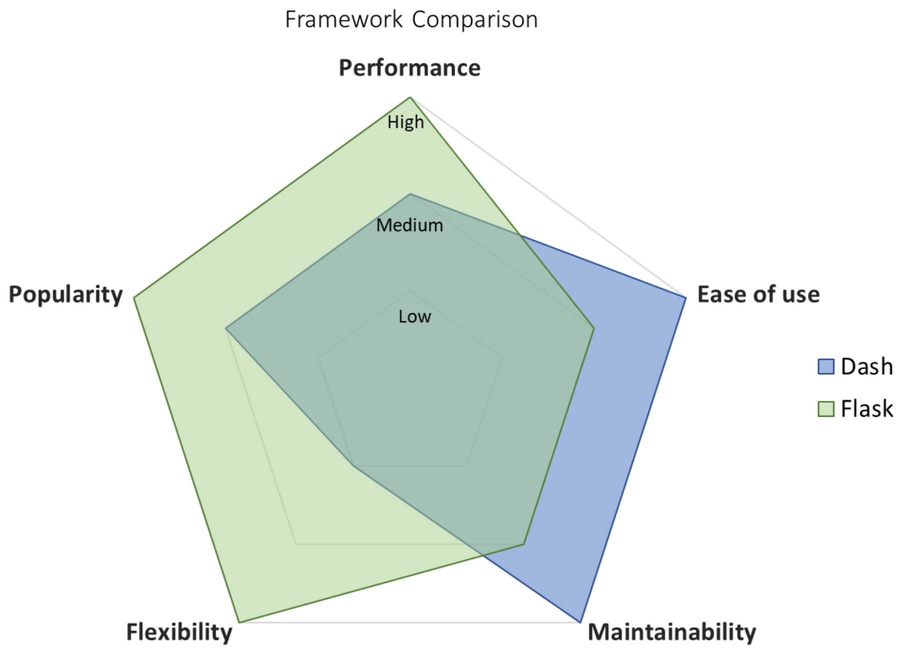


Figure 3: Framework comparison between Dash (blue) and Flask (green). Criteria are judged on a scale from low to high.

5.3 The Application

5.3.1 Backend

With Flask, we rebuilt FERMO's underlying code responsible for routing and client-server interaction from the ground up. This was necessary because the approach to building a web application differs significantly between Dash and Flask. The first difference occurs when implementing the routing. Dash required accessing and changing the URL explicitly, whenever a different page was requested. In contrast, a Flask-application is set up by defining 'routes' for the various pages from the start. Each route is assigned to a function which contains instructions on what should happen when that route is accessed. This includes declaring different actions for different kinds of requests (e.g. GET or POST), consequently handling user input as appropriate.

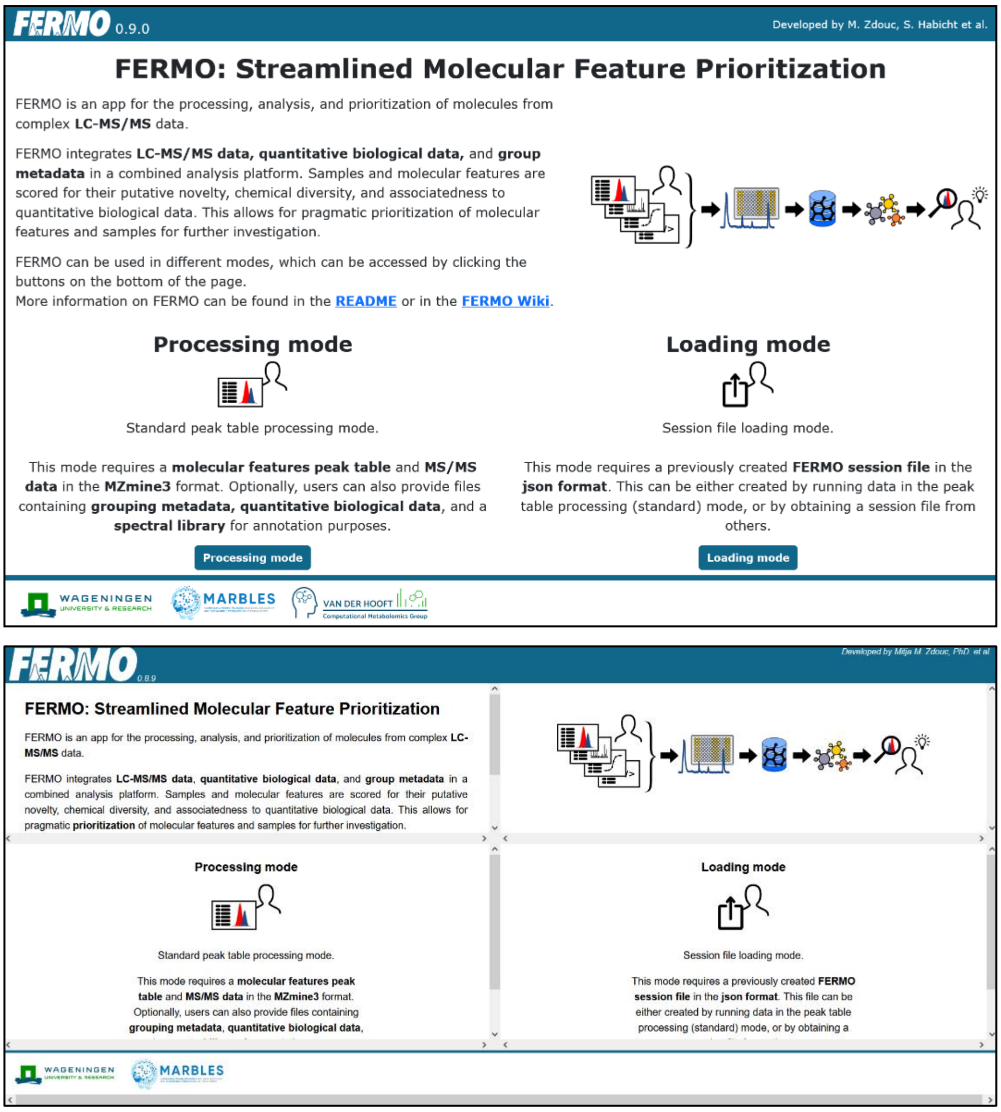
Functional code responsible for data processing was reused as much as possible. Nevertheless, the output format of almost all functions had to be adjusted, due to the new framework and changes in routing. Initially, many functions returned output in a Dash-specific format, or objects that were not JSON-compatible. However, JSON-compatibility was necessary for the Flask-application because the browser cannot handle python formats. The interaction between server and client, i.e. the handling of requests and responses, had to be implemented explicitly since Flask cannot automatically convert all data.

5.3.2 Frontend

On the frontend the general structure of the application (i.e. the user first accesses the landing page, then chooses processing or loading mode before being directed to the dashboard page) and overall design remained unchanged. Nevertheless, the increased flexibility of using HTML and Bootstrap/custom CSS directly allowed for some modifications to the layout.

General changes, affecting all the pages, include increased spacing around elements in the header and footer. Additionally, we removed unnecessary scrollbars and buttons now exhibit a visual effect when the mouse hovers over them: The button darkens slightly in shade, and the mouse pointer transforms into a pointing-hand icon.

The landing, processing and loading page underwent only few additional modifications. In the original application, each of these pages comprised separate scrollable sections. However, in the updated version, we merged these sections into a single cohesive section to create a more spacious and visually appealing layout. Nevertheless, we made efforts to maintain a clear distinction between the different sections. A comparison between the layouts is depicted in Figure 4, using the landing page as an example.



FERMO 0.9.0

Developed by M. Zdouc, S. Habicht et al.

FERMO: Streamlined Molecular Feature Prioritization

FERMO is an app for the processing, analysis, and prioritization of molecules from complex **LC-MS/MS** data.

FERMO integrates **LC-MS/MS data, quantitative biological data,** and **group metadata** in a combined analysis platform. Samples and molecular features are scored for their putative novelty, chemical diversity, and associatedness to quantitative biological data. This allows for pragmatic prioritization of molecular features and samples for further investigation.

FERMO can be used in different modes, which can be accessed by clicking the buttons on the bottom of the page.
More information on FERMO can be found in the **README** or in the **FERMO Wiki**.

Processing mode

Standard peak table processing mode.

This mode requires a **molecular features peak table** and **MS/MS data** in the **MZmine3** format. Optionally, users can also provide files containing **grouping metadata, quantitative biological data,** and a **spectral library** for annotation purposes.

Processing mode

Loading mode

Session file loading mode.

This mode requires a previously created **FERMO session file** in the **json format**. This can be either created by running data in the peak table processing (standard) mode, or by obtaining a session file from others.

Loading mode

FERMO 0.8.9

Developed by Mitja M. Zdouc, PhD. et al.

FERMO: Streamlined Molecular Feature Prioritization

FERMO is an app for the processing, analysis, and prioritization of molecules from complex **LC-MS/MS** data.

FERMO integrates **LC-MS/MS data**, **quantitative biological data**, and **group metadata** in a combined analysis platform. Samples and molecular features are scored for their putative novelty, chemical diversity, and associatedness to quantitative biological data. This allows for pragmatic **prioritization** of molecular features and samples for further investigation.

Processing mode

Standard peak table processing mode.

This mode requires a **molecular features peak table** and **MS/MS data** in the **MZmine3** format. Optionally, users can also provide files containing **grouping metadata**, **quantitative biological data**,

Loading mode

Session file loading mode.

This mode requires a previously created **FERMO session file** in the **json format**. This file can be either created by running data in the peak table processing (standard) mode, or by obtaining a

Figure 4: Layout of the new loading page (top) compared to the old loading page (bottom).

Noteworthy elements are the multi range sliders such as the relative intensity filter on the processing page. While Dash offers convenient Python functions to create such sliders, with Flask we had to develop them using the frontend tools directly. Even though various input elements are available in HTML, range sliders with more than one handle are not provided. Consequently, a custom solution was necessary, involving the placement of two range sliders on top of each other and the utilization of JavaScript to modify their functionality. The corresponding code was based on an article published by Davidovic [43]. Images of all the different pages are displayed in comparison to the old layout in Supplementary Figure 1, Supplementary Figure 2 and Supplementary Figure 3.

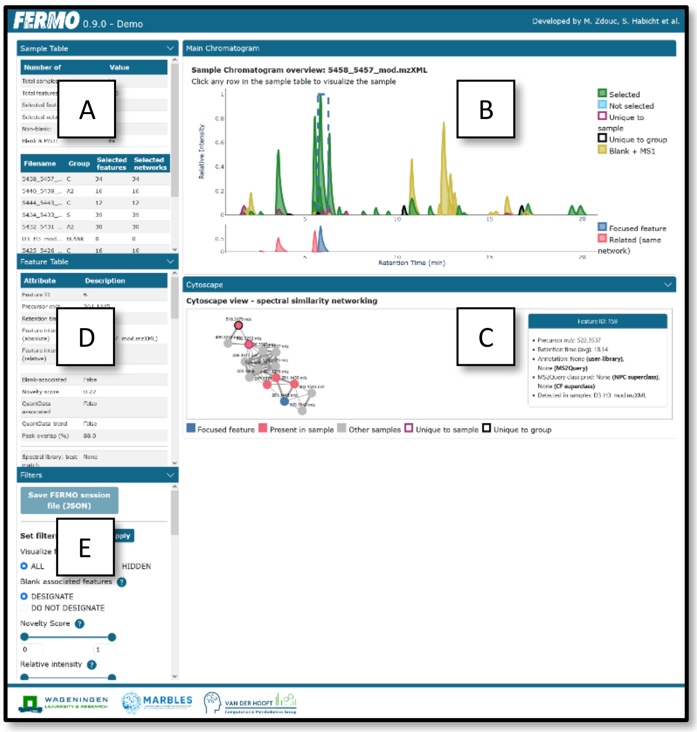
More significant modifications were made to the dashboard page since it represents the core of the application and thus allowed for the most substantial enhancement of the user experience. Here, we also incorporated feedback of the beta testing, conducted independently of this project. The most notable change lies in the new panels (see Figure 5). Originally, they had a fixed size but now they can be extended and collapsed which allows users to focus on the panel they are interested in. The two layouts are depicted next to each other for a better comparison in Supplementary Figure 3. Furthermore, we moved panel D, the feature table, to the left side. Removing the panel with chromatograms of samples containing the selected feature (see Supplementary Figure 3) altogether allowed more room for panel C, the Cytoscape graph. This decision was based on user feedback which suggested this panel to be more confusing than helpful. As shown in Figure 5A, horizontal scrolling in the sample table is prevented by arranging the table at the top in a vertical style and truncating the lower table after four columns. However, the truncated information is not omitted, but displayed when hovering over the respective row with the mouse. Similarly, we replaced the table below the Cytoscape graph, which presented node and edge information when clicking them, by a popup that appears in the top right corner of panel C when hovering over the nodes and edges (see Figure 5C). This is the result of a workaround, because the initial plan was to make use of the cytoscape.js-popper extension [44]. This extension allows to display popups right next to the elements where they belong to. However, unknown errors occurred, so that we were not able to utilize popper. Therefore, we manually created elements containing the corresponding information. Instead of appearing nest to the element, they are always displayed in the top right corner of the panel.

Next to these visual changes, the dashboard also gained functionality. Due to the way Dash processes interactions, updating the chromatogram and feature table after selecting a feature in the Cytoscape graph was difficult to implement. With Flask, such interactions between the panels are handled with JavaScript directly, therefore, all elements are now updated after clicking a node that is present in the active sample.

5.3.3 Further Necessary Steps

The above-described modifications to front- and backend resulted in a prototype with a fully functional example dashboard. Nevertheless some critical functionality for the application is still missing.

First and foremost, it is crucial to address the processing of user data on the processing page. To accomplish this, first the filter and settings data need to be collected and transmitted to the server. Additionally, input sanitation has to be employed to constrain file size and prevent malicious activity. Subsequently, the actual data handling has to be implemented. For this, a majority of the code from the original application can be reused because the processing functionality remains the same. Some adaptations however are required to incorporate the filter settings due to the fundamental differences in how Dash and Flask handle user input.



FERMO 0.9.0 - Demo

Developed by M. Zdouc, S. Habicht et al.

Sample Table

Number of	Value
Total samples:	
Total features:	
Selected features:	
Selected networks:	
Non-blank:	
Blank & MS1:	49

Filename	Group	Selected features	Selected networks
5458_5457_...	C	34	34
5440_5439_...	A2	16	16
5444_5443_...	C	12	12
5434_5433_...	S	39	39
5432_5431_...	A2	30	30
D3_H3_mod....	BLANK	0	0
5425_5426_...	C	16	16

Feature Table

Attribute	Description
Feature ID	6
Precursor m/z	301.1445
Retention time	
Feature intensity (absolute)	mod.mzXML)
Feature intensity (relative)	
Blank-associated	False
Novelty score	0.22
QuantData associated	False
QuantData trend	False
Peak overlap (%)	88.0
Spectral library: best match	None

Filters

Save FERMO session file (JSON)

Set filter Apply

Visualize f

ALL HIDDEN

Blank associated features ?

DESIGNATE

DO NOT DESIGNATE

Novelty Score ?

0 1

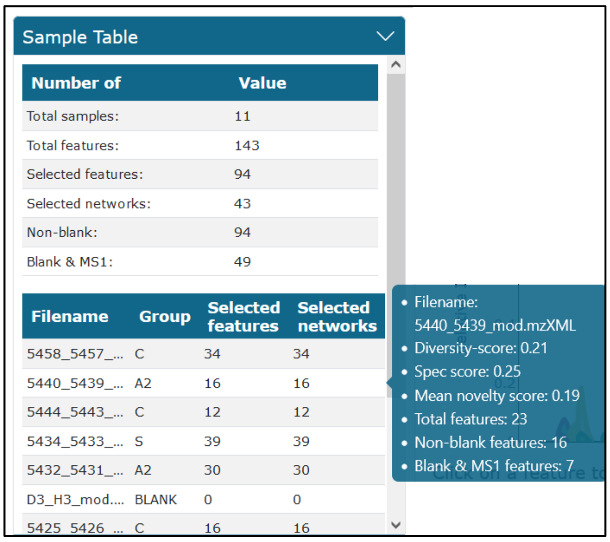
Relative intensity ?

Main Chromatogram

Cytoscape

A
B
C
D
E
WAGENINGEN UNIVERSITY & RESEARCH
MARBLES
VAN DER HOOFT Computational Metabolomics Group

A



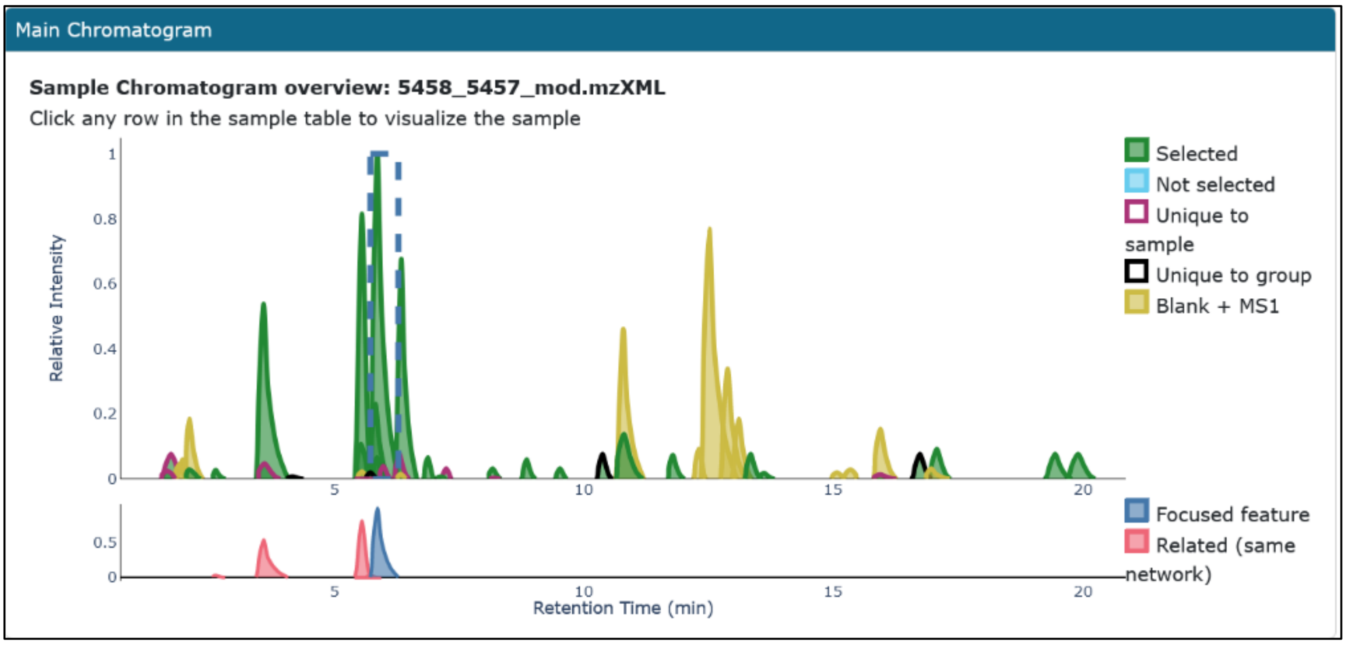
Sample Table

Number of	Value
Total samples:	11
Total features:	143
Selected features:	94
Selected networks:	43
Non-blank:	94
Blank & MS1:	49

Filename	Group	Selected features	Selected networks
5458_5457_...	C	34	34
5440_5439_...	A2	16	16
5444_5443_...	C	12	12
5434_5433_...	S	39	39
5432_5431_...	A2	30	30
D3_H3_mod....	BLANK	0	0
5425_5426_...	C	16	16

- Filename: 5440_5439_mod.mzXML
- Diversity-score: 0.21
- Spec score: 0.25
- Mean novelty score: 0.19
- Total features: 23
- Non-blank features: 16
- Blank & MS1 features: 7

B

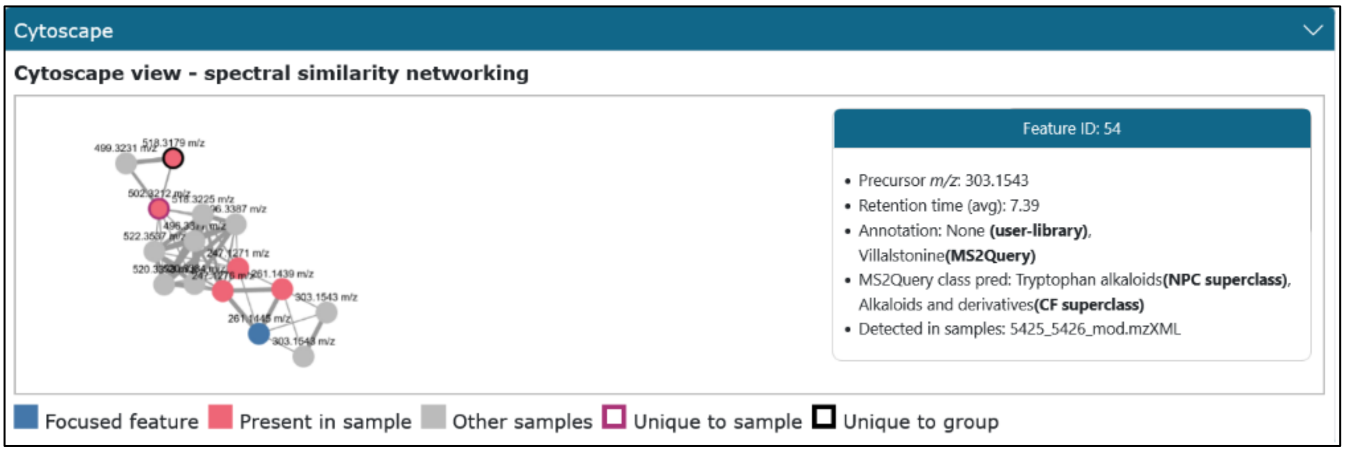


Main Chromatogram

Sample Chromatogram overview: 5458_5457_mod.mzXML

Click any row in the sample table to visualize the sample

C



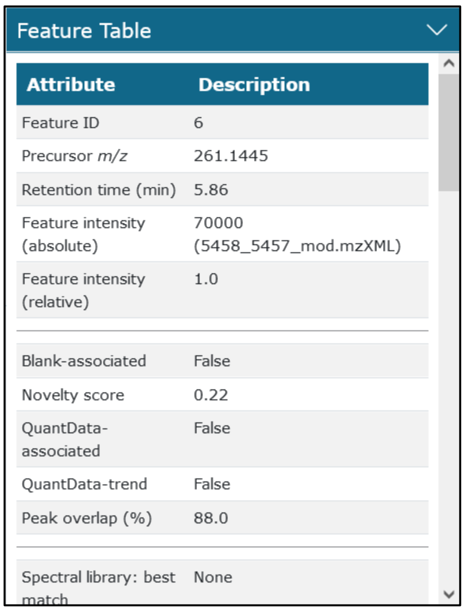
Cytoscape

Cytoscape view - spectral similarity networking

Feature ID: 54

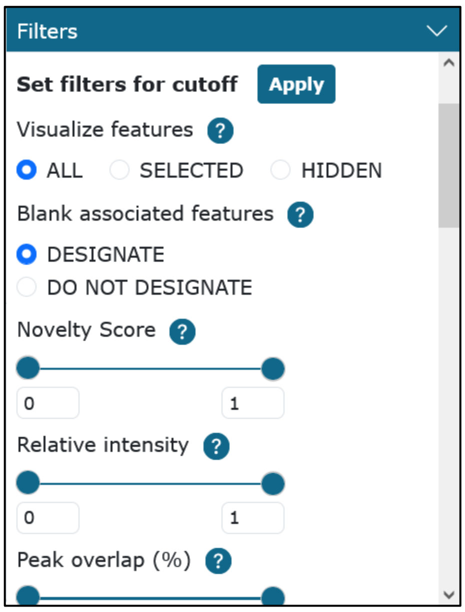
- Precursor *m/z*: 303.1543
- Retention time (avg): 7.39
- Annotation: None **(user-library)**, Villalstonine**(MS2Query)**
- MS2Query class pred: Tryptophan alkaloids**(NPC superclass)**, Alkaloids and derivatives**(CF superclass)**
- Detected in samples: 5425_5426_mod.mzXML

D


Feature Table

Attribute	Description
Feature ID	6
Precursor *m/z*	261.1445
Retention time (min)	5.86
Feature intensity (absolute)	70000 (5458_5457_mod.mzXML)
Feature intensity (relative)	1.0
Blank-associated	False
Novelty score	0.22
QuantData-associated	False
QuantData-trend	False
Peak overlap (%)	88.0
Spectral library: best match	None

E


Filters

Set filters for cutoff **Apply**

Visualize features: [✓ ALL] SELECTED HIDDEN

Blank associated features: [✓ DESIGNATE] DO NOT DESIGNATE

Novelty Score: 0 – 1

Relative intensity: 0 – 1

Peak overlap (%)

Figure 5: Overview over the entire dashboard (top left), zoomed in to each panel for a more detailed view. A: Sample table panel, while hovering over the second row in the lower table to display the additional information; B: Main chromatogram panel, after clicking one of the peaks. The selected feature is marked by the dashed frame; C: Cytoscape graph, corresponding to the selected feature and while hovering over one of the nodes to display the additional node information; D: Feature table panel, displaying information of the selected feature; E: Filters panel with the default settings.

The processed user data should then be stored in an appropriate session-file, because the dashboard page was developed based on such a file. For the web application, however, this file must be assigned a unique name to prevent it from being overwritten by the file of another user. Furthermore, limited memory capacity of the server has to be taken into account. Session files should not be stored forever but automatically removed once the session is terminated.

Finally, the completed application has to be deployed to the public. Since the additions to the code do not affect how the application can be containerized, the Dockerfile that was created for the demo version (see chapter 4.3.2) can be reused unchanged. That is, if the demo version is stopped because each port (i.e. the access point for an application on a server) can be used only once. Otherwise, the file would have to be adjusted to point to a different port. However, since the full application still contains a route for an example dashboard with preloaded data, the demo can be taken down without any downsides. Consequently, the finished application only has to be installed on the server, before the Docker-Compose file can be executed (see Methods, for explanation).

These steps outline the minimal necessary requirements to deploy an online version of FERMO. However, we have also planned additional functionality to enhance FERMO further. These optional improvements are discussed in detail in chapter 6.3 of the discussion.

5.3.4 Demo

While the web application is still being developed, we made a demo version available to already showcase the new dashboard at https://fermo.bioinformatics.nl/example. This, fully functional, demo contains all improvements to the dashboard described above. It allows interested individuals to explore the dashboard and interact with an example dataset as described by Zdouc et al. (Preprint) [13]. Due to an upcoming update to the format of the session file and peak tables, these files are currently unavailable. Therefore we disabled the respective download buttons. To inform the user of this circumstance, we implemented a popup that appears when hovering over the corresponding elements.

6 Discussion

6.1 Framework Comparison

Over the course of this project, we encountered several challenges that required adjustments to our plans. One example occurred in context of the framework comparison: The initial plan was to conduct an in-depth evaluation of all seven frameworks that were identified as relevant. However, the amount of time required to complete such an analysis was underestimated. The comparison was further complicated by the difficulty to objectively judge the chosen criteria. Consequently, we focused on two frameworks based on which seemed most relevant for the project. To evaluate the criteria, we summarized our research using ordinal values. This is however not guaranteed to result in the same conclusion for other researchers, as an assignment of these values is a subjective decision. Nevertheless, we argue that the framework selection was sufficiently reasoned because Flask indeed provided the desired flexibility, easy development and no reason to expect issues regarding the maintenance in the future.

Furthermore, we identified several problematic situations a FERMO-user might encounter (see Supplementary Table 3) but it was unfeasible to address all of them during this project. We decided to create a web application because this provides the most benefits. Nevertheless, the other situations were not simply disregarded but can be addressed in future updates as described below.

6.2 Other Tools

Integration with Inventa

An example for a remaining limitation is that FERMO was not designed to handle large numbers of samples; the current limit is reached at about a few dozen samples [13]. Some studies, however, may consider bacterial extracts obtained from many different growth conditions or water and soil samples from different locations, combined with different times of the day and different sampling methods, leading to very large numbers of samples. In these cases, FERMO is best used in combination with a tool that prioritizes samples first, such as Inventa [45]. Similar to FERMO, Inventa's goal is to accelerate the mining of MS data. However, unlike FERMO, Inventa does not indicate the most promising features but identifies the most promising samples, i.e., those with a high probability of containing new compounds. It accomplishes this by calculating a priority score for samples based on structural novelty. The two tools could be combined in a single workflow, where Inventa would be employed as a preliminary method to filter out samples with presumably mostly known molecules. Then, FERMO could be used to prioritize the individual features of the remaining samples.

Comparison to the GNPS Dashboard

Another tool, that can be used in combination with FERMO is the GNPS Dashboard [18]. Like FERMO, it offers visualization of MS data in a dashboard view and is available online. However, it uses raw data as input and presents a lot of detailed information at once which necessitates scrolling to view all details. This can make it challenging to focus on specific aspects of interest. In contrast, we designed FERMO to present the most important information at a glance which helps to keep an overview while providing custom scores to prioritize features. However, as discussed in the article on the original version of FERMO [13], it is important to note that the chromatogram in the FERMO dashboard is a pseudo-chromatogram which may not accurately represent peaks with abnormal shapes. We therefore recommend features of interest to be analyzed in more detail using a tool like the GNPS dashboard. Hence, the most effective analysis can be achieved by leveraging the strengths of both tools in combination, where FERMO aids in identification of features of interest and the GNPS dashboard allows for a more in-depth exploration of those features.

6.3 Further Improvements

Database and Processing Settings

Next to an integration with Inventa, many other useful additions and improvements can be imagined. Among other things, we are considering the implementation of a database. This would facilitate data storage on the server and enable us to save session files for a longer period of time. An inspiration for this was antiSMASH [46], an online platform for the identification of biosynthesis gene clusters. There, a job is stored on the application server for a month. During this time, it can be accessed by the user with an individual URL. Similarly, we envision access to previous FERMO-sessions by entering an identifier in the URL or in a designated input field on the loading page. This would ensure that users can access their processed data even if they did not download the session file initially.

Furthermore, we have made plans to incorporate a way for users to save the processing settings in order to improve the user experience. When a user has multiple files to be processed with the same non-default settings, so far, each option has to be set manually for each peaktable file. This may take a long time when all settings are changed. Generating a file where the settings are stored, offering it for download and implementing an input field for this file, would allow users to reupload it thus ensuring exact restoring of settings in very little time.

Cytoscape Popups

Furthermore, the popups on the Cytoscape graph should be revised. As described in the results, we used a workaround to display them, but the solution comes with a drawback: Moving the cursor too quickly over and off of nodes or edges can cause popups to get stuck on top of each other (see supplementary information, chapter 11.2 for an explanation on why this happens). The current 'fix' is to stay long enough on an element for the first request to finish before moving the cursor away to trigger the removal of all popups. Although this does not lessen FERMO's functionality, it may give the impression that the application is not error-free and impair the user experience. Therefore, efforts should be made to prevent the popups from getting stuck.

Draggable Panels

While we already made the panels of the dashboard collapsible, it is possible to give the user even more control over the appearance of the dashboard by making the panels also draggable. Users could then decide exactly where each element is placed. However, this is likely rather complex to implement, thus the benefits must be carefully weighed against the costs. As long as the dashboard is relatively compact, the enhanced user experience is not enough to justify such a complex feature. Nevertheless, if more panels are added in the future, the benefits may take over and we will reconsider the possibility.

Technical Aspect

On a technical level, FERMO could be improved by developing automatic integration testing. For the application as it is presented in this study, integration testing was done manually. This holds the risk of missing potential issues and is consequently prone to introduce errors. A solution would be to make more use of GitHub Actions [47]. Even though this functionality is already utilized to test the installation and run unit tests of the Python code, it could be extended to include more advanced integration tests. Furthermore, we did not introduce any unit tests to the frontend yet, as it necessitates the incorporation of a JavaScript testing framework which requires a time investment that was not feasible in this project. This step is, however, strongly advised to keep the JavaScript code easily maintainable, especially when FERMO's functionality is extended in the future and consequently becomes more complex.

Beta Testing

Once the web application is completed, a second round of beta testing could be conducted. This would serve to identify additional possibilities to make FERMO more user friendly, and to evaluate whether the implemented changes had the intended effect. Ideally both individuals, who already participated in the first round or already used the original version for their research and new users without prior experience with FERMO will be asked to give their feedback. By obtaining input from both groups, we can evaluate how the impression has changed compared to the original version, as well as how intuitive the new application is. Specifically asking about opinions on the above discussed suggestions as well as explicitly inviting testers to express their own ideas and wishes would ensure that FERMO is tailored to researchers needs.

6.4 Conclusion

In conclusion, the improvements described in this report detail more than just the development of a web application; they encompass an extensive update of the original software. While the web application increases FERMO's accessibility by being platform-independent, we also incorporated feedback from the beta testing in order to improve the user experience. By transitioning to a new underlying framework, we facilitated maintenance and development while increasing the application's flexibility. This is especially beneficial for the many potential future modifications. Additionally, all changes have been implemented in a manner that not only enhances the web application, but also carries over to the local version. Even though there are many regards in which FERMO Online can still be optimized, we accomplished crucial steps towards an online platform. This platform harnesses the large amount of features in MS data by facilitating the identification of the most promising ones.

FERMO adds valuable functionality to existing tools and data ecosystems as it offers a clear overview over relevant features. The most effective analysis can, in fact, be achieved, by using FERMO in addition to other tools such as Inventa and the GNPS Dashboard but also NP Analyst [12]. These tools complement each other by focusing on different aspects. Together, they can be utilized to avoid rediscovery of known molecules and reduce the waste of resources. This could convince pharmaceutical companies to reconsider their withdrawal of support for NPR. Moreover, we found in the user analysis that also researchers from other disciplines could apply FERMO. Therefore the completed update of the web application will undoubtedly bring substantial benefits to research endeavors across various disciplines.

7 Data availability

Data for the example page was taken from **MSV000085376** (https://doi.org/doi:10.25345/C5412V) [48] and processed with the original FERMO application (version 0.8.1). The resulting session-file can be found on the GitHub page of the FERMO project in the branch "FERMO_Flask" (https://github.com/mmzdouc/FERMO/tree/FERMO_Flask/example_data). Additional files used to test the behavior of the application were created by manually modifying this session-file. They can be found here: https://github.com/mmzdouc/FERMO/tree/FERMO_Flask/example_data/dev_files

8 Code availability

All code was developed open source under version control via GitHub and is available at https://github.com/mmzdouc/FERMO/tree/FERMO_Flask for the main application and at

https://github.com/mmzdouc/FERMO/tree/fermo_flask_demo for the demo version.

9 References

[1] A. L. Harvey, "Natural products in drug discovery," *Drug Discov Today,* vol. 13, no. 19-20, pp. 894-901, Oct 2008, doi: 10.1016/j.drudis.2008.07.004.

[2] J. Hubert, J.-M. Nuzillard, and J.-H. Renault, "Dereplication strategies in natural product research: How many tools and methodologies behind the same concept?," *Phytochemistry Reviews,* vol. 16, no. 1, pp. 55-95, 2015, doi: 10.1007/s11101-015-9448-7.

[3] J. L. Wolfender, J. M. Nuzillard, J. J. J. van der Hooft, J. H. Renault, and S. Bertrand, "Accelerating Metabolite Identification in Natural Product Research: Toward an Ideal Combination of Liquid Chromatography-High-Resolution Tandem Mass Spectrometry and NMR Profiling, in Silico Databases, and Chemometrics," *Anal Chem,* vol. 91, no. 1, pp. 704-742, Jan 2 2019, doi: 10.1021/acs.analchem.8b05112.

[4] J. Y. Ortholand and A. Ganesan, "Natural products and combinatorial chemistry: back to the future," *Curr Opin Chem Biol,* vol. 8, no. 3, pp. 271-80, Jun 2004, doi: 10.1016/j.cbpa.2004.04.011.

[5] T. Ito and M. Masubuchi, "Dereplication of microbial extracts and related analytical technologies," *J Antibiot (Tokyo),* vol. 67, no. 5, pp. 353-60, May 2014, doi: 10.1038/ja.2014.12.

[6] Y. Chen, M. Garcia de Lomana, N. O. Friedrich, and J. Kirchmair, "Characterization of the Chemical Space of Known and Readily Obtainable Natural Products," *J Chem Inf Model,* vol. 58, no. 8, pp. 1518-1532, Aug 27 2018, doi: 10.1021/acs.jcim.8b00302.

[7] J. S. Morris, K. R. Coombes, J. Koomen, K. A. Baggerly, and R. Kobayashi, "Feature extraction and quantification for mass spectrometry in biomedical applications using the mean spectrum," *Bioinformatics,* vol. 21, no. 9, pp. 1764-75, May 1 2005, doi: 10.1093/bioinformatics/bti254.

[8] G. L. Glish and R. W. Vachet, "The basics of mass spectrometry in the twenty-first century," *Nat Rev Drug Discov,* vol. 2, no. 2, pp. 140-50, Feb 2003, doi: 10.1038/nrd1011.

[9] R. Tautenhahn, G. J. Patti, D. Rinehart, and G. Siuzdak, "XCMS Online: a web-based platform to process untargeted metabolomic data," *Anal Chem,* vol. 84, no. 11, pp. 5035-9, Jun 5 2012, doi: 10.1021/ac300698c.

[10] B. Zhou, J. F. Xiao, and H. W. Ressom, "Prioritization of putative metabolite identifications in LC-MS/MS experiments using a computational pipeline," *Proteomics,* vol. 13, no. 2, pp. 248-60, Jan 2013, doi: 10.1002/pmic.201200306.

[11] L.-F. Nothias *et al.*, "Bioactivity-Based Molecular Networking for the Discovery of Drug Leads in Natural Product Bioassay-Guided Fractionation," *Journal of Natural Products,* vol. 81, no. 4, pp. 758-767, 2018/04/27 2018, doi: 10.1021/acs.jnatprod.7b00737.

[12] S. Lee *et al.*, "NP Analyst: An Open Online Platform for Compound Activity Mapping," *ACS Central Science,* vol. 8, no. 2, pp. 223-234, 2022/02/23 2022, doi: 10.1021/acscentsci.1c01108.

[13] M. M. Zdouc *et al.*, "[Preprint] FERMO: a Dashboard for Streamlined Rationalized Prioritization of Molecular Features from Mass Spectrometry Data," Dec 22 2022, doi: 10.1101/2022.12.21.521422.

[14] *Bootstrap*. (2023). [Online]. Available: https://getbootstrap.com/

[15] *React*. (2023). [Online]. Available: https://react.dev/

[16] *Vue.js*. (2023). [Online]. Available: https://vuejs.org/

[17] Mozilla. "Introduction to client-side frameworks." https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/Introduction (accessed Jun 18, 2023).

[18] D. Petras *et al.*, "GNPS Dashboard: collaborative exploration of mass spectrometry data in the web browser," *Nature Methods,* vol. 19, no. 2, pp. 134-136, 2022/02/01 2022, doi: 10.1038/s41592-021-01339-5.

[19] Mozilla. "Server-side web frameworks." https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Web_frameworks (accessed Jun 18, 2023).

[20] *Flask*. (2022). [Online]. Available: https://github.com/pallets/flask/releases/tag/2.2.2

[21] *Dash*. (2022). [Online]. Available: https://github.com/plotly/dash/releases/tag/v2.7.1

[22] J. Pederson. "What happens when you type a URL into your browser?" https://aws.amazon.com/blogs/mobile/what-happens-when-you-type-a-url-into-your-browser/ (accessed Jun 18, 2023).

[23] R. Mustapha. "What is HTTP? Protocol Overview for Beginners." https://www.freecodecamp.org/news/what-is-http/ (accessed Jun 18, 2023).

[24] W. Reese, "Nginx: the high-performance web server and reverse proxy," *Linux J.,* vol. 2008, no. 173, p. Article 2, 2008.

[25] *Apache HTTP Server Project*. (2023). [Online]. Available: https://httpd.apache.org/

[26] F5. "What Is a Web Server?" https://www.nginx.com/resources/glossary/web-server/ (accessed Jun 12, 2023).

[27] *Gunicorn*. (2021). [Online]. Available: https://docs.gunicorn.org/en/stable/

[28] *mod_wsgi*. (2022). [Online]. Available: https://github.com/GrahamDumpleton/mod_wsgi

[29] Pallets. "Deploying to Production." https://flask.palletsprojects.com/en/2.3.x/deploying/ (accessed Jun 12, 2023).

[30] F5. "What Is a Reverse Proxy Server?" https://www.nginx.com/resources/glossary/reverse-proxy-server/ (accessed Jun 18, 2023).

[31] I. GitHub. "About continuous integration." https://docs.github.com/en/actions/automating-builds-and-tests/about-continuous-integration (accessed Jun 20, 2023).

[32] G. van Rossum, B. Warsaw, and N. Coghlan. "PEP 8 – Style Guide for Python Code." https://peps.python.org/pep-0008/ (accessed Jun 14, 2023).

[33] *Flake 8*. (2022). [Online]. Available: https://pypi.org/project/flake8/

[34] A. Hunt and D. Thomas, *The Pragmatic Programmer: From Journeyman to Master* Reading, Mass. : Addison-Wesley, 2000.

[35] D. Merkel, "Docker: lightweight linux containers for consistent development and deployment," *Linux journal,* no. 239, p. 2, 2014.

[36] L. S. Vailshery. "Most loved, dreaded and wanted web-frameworks and technologies." https://survey.stackoverflow.co/2022/#section-most-loved-dreaded-and-wanted-web-frameworks-and-technologies (accessed Jun 8, 2023).

[37] L. S. Vailshery. "Web framework and technologies." https://survey.stackoverflow.co/2022/#most-popular-technologies-webframe (accessed Jun 8, 2023).

[38] I. Deed. "Highest Performing Web Framework Benchmarks." https://www.pangea.ai/resources/web/ (accessed Jun 7, 2023).

[39] A. Jones. "Plotly with Streamlit, Dash or Flask." https://towardsdatascience.com/plotly-with-streamlit-dash-or-flask-4d78fa025ea2 (accessed Feb 4, 2023).

[40] M. Schmitt. "Streamlit vs. Dash vs. Shiny vs. Voila vs. Flask vs. Jupyter." https://www.datarevenue.com/en-blog/data-dashboarding-streamlit-vs-dash-vs-shiny-vs-voila (accessed Feb 4, 2023).

[41] Plotly. "plotly/dash." https://github.com/plotly/dash (accessed Jun 29, 2023).

[42] pallets. "pallets/flask." https://github.com/pallets/flask (accessed Jun 29, 2023).

[43] P. Davidovic. "Native dual range slider — HTML, CSS & JavaScript." https://medium.com/@predragdavidovic10/native-dual-range-slider-html-css-javascript-91e778134816 (accessed Feb 24, 2023).

[44] *Cytoscape.js-popper*. (2021). [Online]. Available: https://github.com/cytoscape/cytoscape.js-popper

[45] L. M. Quiros-Guerrero *et al.*, "Inventa: A computational tool to discover structural novelty in natural extracts libraries," *Front Mol Biosci,* vol. 9, p. 1028334, 2022, doi: 10.3389/fmolb.2022.1028334.

[46] K. Blin *et al.*, "antiSMASH 7.0: new and improved predictions for detection, regulation, chemical structures and visualisation," *Nucleic Acids Research,* 2023, doi: 10.1093/nar/gkad344.

[47] I. GitHub. "Understanding GitHub Actions." https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions (accessed Jun 23, 2023).

[48] M. M. Zdouc, M. Iorio, S. I. Maffioli, M. Crusemann, S. Donadio, and M. Sosio, "Planomonospora: A Metabolomics Perspective on an Underexplored Actinobacteria Genus," *J Nat Prod,* vol. 84, no. 2, pp. 204-219, Feb 26 2021, doi: 10.1021/acs.jnatprod.0c00807.

10 Acknowledgements

Without the help of other people this project would not have been possible. I would like to use the following lines to acknowledge this.

First and foremost, Mitja, I thank you for your time, constant support, feedback and guidance through weekly meetings. Special thanks to you for letting me work on your software; I can only imagine how hard it is to give something out of your hands you built from scratch in so little time and let someone else 'butcher' it. It was an amazing opportunity to learn about web development and your openness to letting me figure things out along the way taught me a great deal about developing new skills.

Thank you Marnix and Justin for your kind supervision. You helped me to stay on track and to adjust the planning when necessary. When working on the same project for six months it is easy to get caught up in the details, so thank you for helping me to not lose sight of the bigger picture. The regular meetings with you added another level of highly appreciated guidance.

Moreover, I would like to thank my Buddy Group, the Pancakes, for your feedback on my proposal, and thesis report as well as preparation for the presentations. Our meetings helped to clear my head sometimes and knowing that things do not go according to plan all the time also for other people can be soothing.

My appreciation also goes out to Hannah. Thank you for your advice on the design of the software as well as making time for some meetings to help me with Flask and Docker.

Furthermore, I would like to thank the other staff members and students for helping me troubleshoot and making the lunch break something to look forward to, not just because it is a break.

Almost last but far from least, I would like to thank my family and friends for giving me so much moral support. Thanks Mama, Papa, Simona, Benjamin and Niklas for being there for me when I needed you, for listening and giving me advice on what to do.

Finally, thank you Matteo. You helped me through the most devastating loss in my life so far, allowing me to push through my thesis even when everything felt like a secondary concern. Without you I would not have been able to keep going like I did.

11 Supplementary materials

11.1 User analysis

Supplementary Table 1: Overview over the backgrounds potential FERMO users might have. This list contains examples to demonstrate how versatile MS/MS and consequently FERMO can be utilized but makes no claim to completeness. Applications for Environmental sciences and Forensics are given more extensively since these fields may be less obvious to bring out potential users.

Field	Application	Source
Natural product research	Drug discovery	[1]
Microbiome research	Understanding the role of microbiota	[2]
Clinical research	Personalized drug therapy	[3]
Plant research	Plant metabolism	[4]
Food science	Detection of toxic food contaminants	[5]
Environmental science	Detection, monitoring of pollution in water/soil,	[6]
	Identification of chemicals of emerging concern	[7]
Forensics	Identification of unknown metabolites from drugs,	[8]
	New illicit drug identification,	[9, 10]
	Doping prevention	[11]

Supplementary Table 2: Overview over the aspects in which a potential user's data might differ. The list is however not exhaustive, other aspects and examples can be imagined.

Data Aspect	Examples
Preprocessing software	Mzmine3 [12],
	OpenMS [13],
	XCMS [14],
	MAVEN [15],
	MetaboAnalyst [16] etc.
Size	Few MB up to many GB
Number of samples	Tens up to thousands
Type of molecules	Lipids,
	Sugars,
	Amino acids,
	Fatty acids etc.

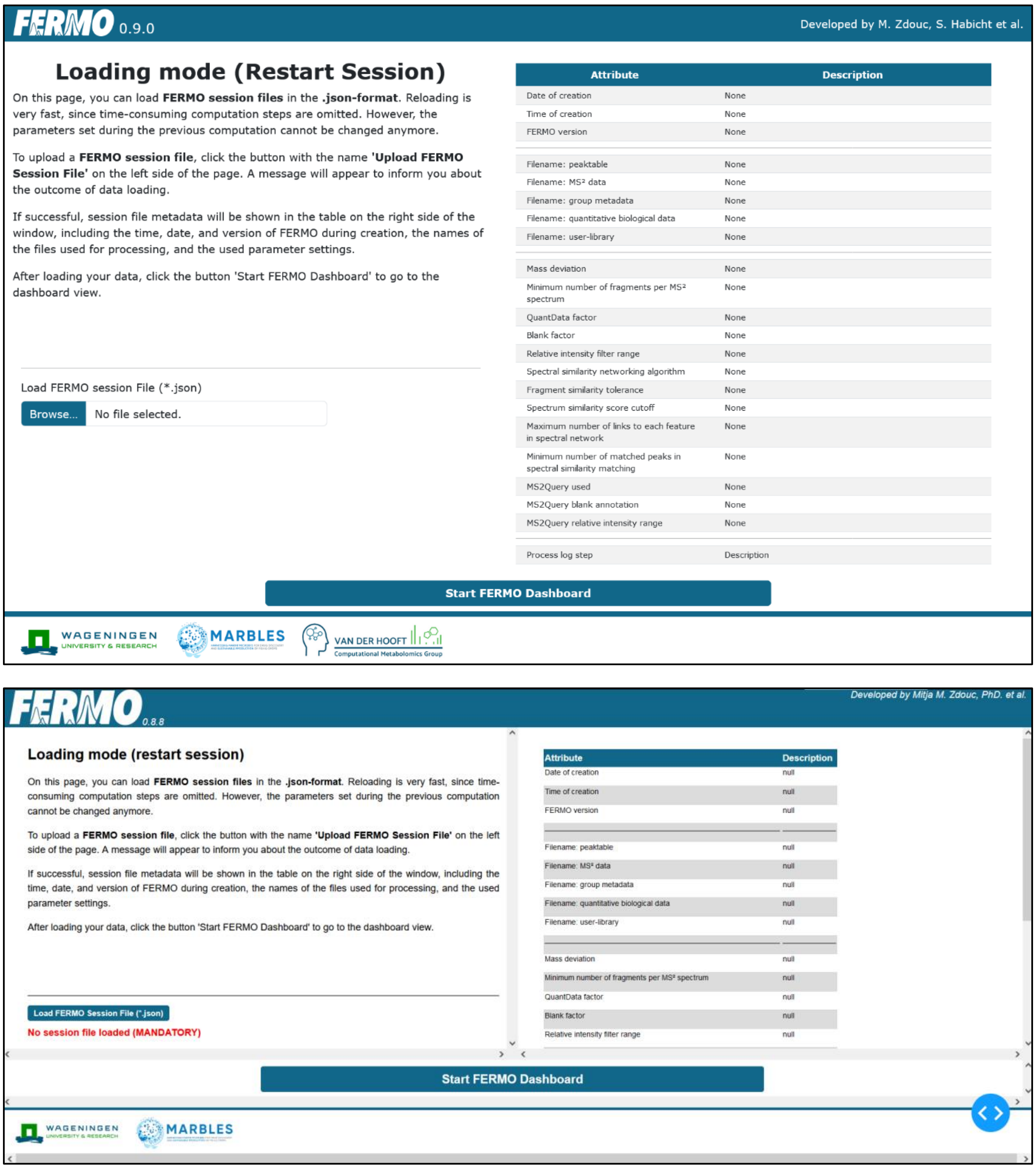
Supplementary Table 3: Possible "problem" situations a user might encounter when using FERMO and their ideal solutions. This list contains examples and makes no claim to completeness.

Situation	Solution
User has access to their institution's server but no rights to install software and the private computer is too old/slow or just not powerful enough to handle huge datasets	FERMO online would not require installation and would allow remote computation
NP researcher measured 3000 samples: extracts from different strains of a bacterium cultivated under different conditions, extracted in different ways etc. → too many for FERMO to handle	Integration of sample preselection tools such as Inventa would help to choose a feasible number of samples
Employee of a pharma company wants to analyze their sensitive data but it should NOT in any way become publicly accessible	The local version of FERMO on the company's computer does not expose the data to the public
FERMO processing was performed in the browser but now the user wants to look at their results while they have to travel, however the internet connection is unreliable	Loading the processed data into the local application makes evaluation of the results possible without an internet connection
Junior researcher has never worked with a command-line, does not even know how to navigate to folders → installation may become difficult	FERMO online would not require installation or knowledge of command-line usage
User has changing workplaces such as laboratory, office and home office with each a different computer but their institution does not have servers to work on	FERMO online would allow access to the software without installation on any device
User normally preprocesses the data with XCMS and has never worked with Mzmine before	Accepting output of different preprocessing tools would allow more flexibility for the user to work with their preferred preprocessing software

11.2 Implementation of Popovers

To display information about nodes and edges in the Cytoscape-graph when the user hovers over them, we intended to utilize the cytoscape.js-popper extension. Unfortunately, the implementation was not successful due to unknown errors or simply not appearing popups. The workaround we used was however also not ideal: In the current implementation, a "mouseover event" is triggered when the user hovers over a node or an edge and initiates the creation of the popup, whereas a "mouseout event" is triggered when the cursor is moved away again and removes the popup. These events are meant to occur successively, i.e. first the mouseover event, then the mouseout event. However, when the user quickly moves the cursor over and off of a node/edge, both events are triggered almost simultaneously. Since we used the asynchronous fetch API to implement communication between front- and backend, functions following the fetch-call can execute without waiting for the fetch to finish. But because retrieving the node and edge information from the server takes longer than removing popups does, the mouseout event finishes before the mouseover event. Consequently, moving the cursor too quickly over and off of an element can cause popups to get seemingly stuck.

11.3 The application


FERMO 0.9.0

Developed by M. Zdouc, S. Habicht et al.

Loading mode (Restart Session)

On this page, you can load **FERMO session files** in the **.json-format**. Reloading is very fast, since time-consuming computation steps are omitted. However, the parameters set during the previous computation cannot be changed anymore.

To upload a **FERMO session file**, click the button with the name **'Upload FERMO Session File'** on the left side of the page. A message will appear to inform you about the outcome of data loading.

If successful, session file metadata will be shown in the table on the right side of the window, including the time, date, and version of FERMO during creation, the names of the files used for processing, and the used parameter settings.

After loading your data, click the button 'Start FERMO Dashboard' to go to the dashboard view.

Load FERMO session File (*.json)

Browse... No file selected.

Attribute	Description
Date of creation	None
Time of creation	None
FERMO version	None
Filename: peaktable	None
Filename: MS² data	None
Filename: group metadata	None
Filename: quantitative biological data	None
Filename: user-library	None
Mass deviation	None
Minimum number of fragments per MS² spectrum	None
QuantData factor	None
Blank factor	None
Relative intensity filter range	None
Spectral similarity networking algorithm	None
Fragment similarity tolerance	None
Spectrum similarity score cutoff	None
Maximum number of links to each feature in spectral network	None
Minimum number of matched peaks in spectral similarity matching	None
MS2Query used	None
MS2Query blank annotation	None
MS2Query relative intensity range	None
Process log step	Description

Start FERMO Dashboard

WAGENINGEN UNIVERSITY & RESEARCH   MARBLES   VAN DER HOOFT Computational Metabolomics Group

FERMO 0.8.8

Developed by Mitja M. Zdouc, PhD. et al.

Loading mode (restart session)

On this page, you can load **FERMO session files** in the **.json-format**. Reloading is very fast, since time-consuming computation steps are omitted. However, the parameters set during the previous computation cannot be changed anymore.

To upload a **FERMO session file**, click the button with the name **'Upload FERMO Session File'** on the left side of the page. A message will appear to inform you about the outcome of data loading.

If successful, session file metadata will be shown in the table on the right side of the window, including the time, date, and version of FERMO during creation, the names of the files used for processing, and the used parameter settings.

After loading your data, click the button 'Start FERMO Dashboard' to go to the dashboard view.

Load FERMO Session File (*.json)

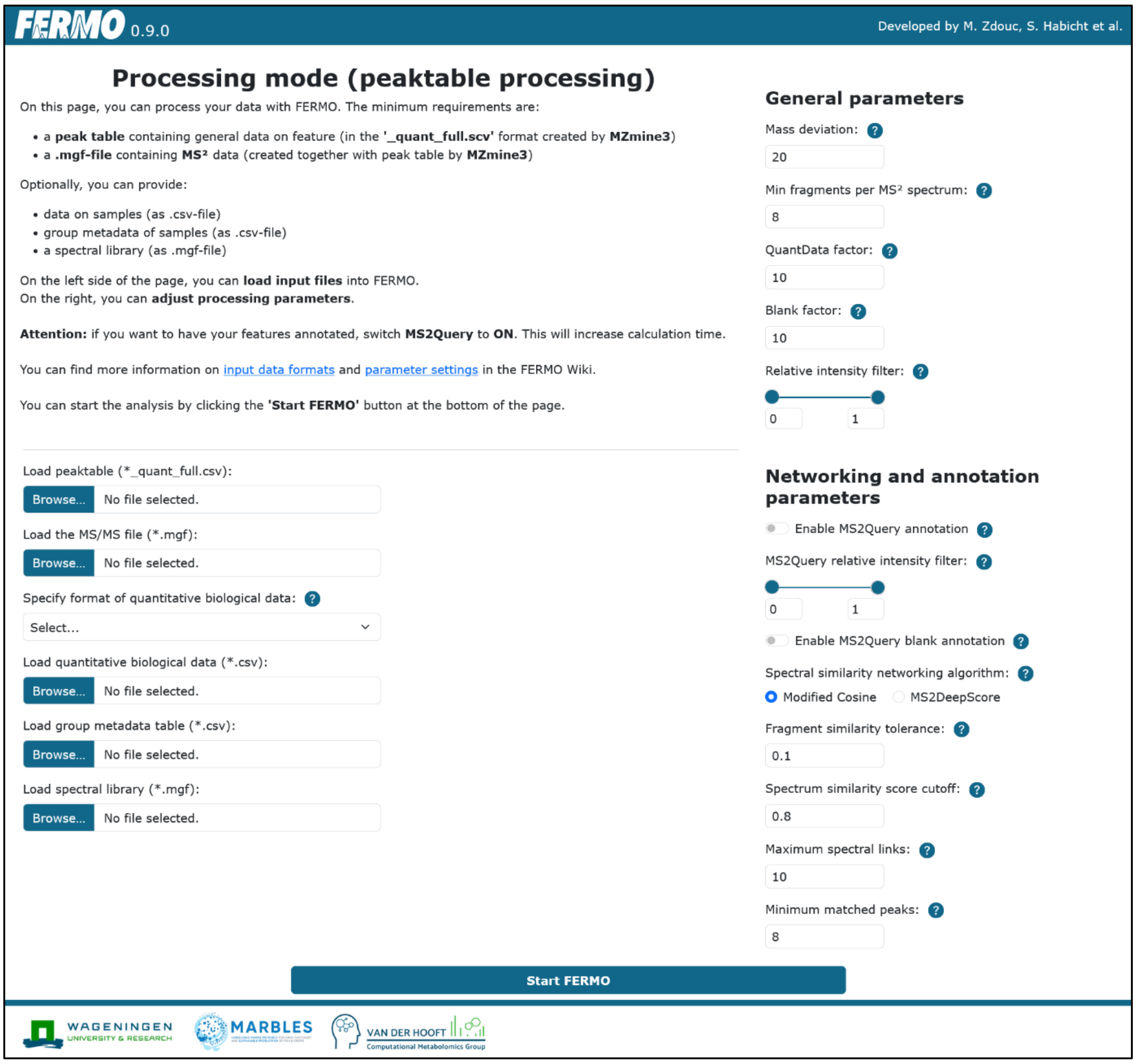
No session file loaded (MANDATORY)

Attribute	Description
Date of creation	null
Time of creation	null
FERMO version	null
Filename: peaktable	null
Filename: MS² data	null
Filename: group metadata	null
Filename: quantitative biological data	null
Filename: user-library	null
Mass deviation	null
Minimum number of fragments per MS² spectrum	null
QuantData factor	null
Blank factor	null
Relative intensity filter range	null

Start FERMO Dashboard

WAGENINGEN UNIVERSITY & RESEARCH   MARBLES

Supplementary Figure 1: New layout of the loading page when no file is loaded (top) compared to the equivalent old loading page (bottom).


FERMO 0.9.0

Developed by M. Zdouc, S. Habicht et al.

Processing mode (peaktable processing)

On this page, you can process your data with FERMO. The minimum requirements are:

- a **peak table** containing general data on feature (in the **'_quant_full.scv'** format created by **MZmine3**)
- a **.mgf-file** containing **MS²** data (created together with peak table by **MZmine3**)

Optionally, you can provide:

- data on samples (as .csv-file)
- group metadata of samples (as .csv-file)
- a spectral library (as .mgf-file)

On the left side of the page, you can **load input files** into FERMO.
On the right, you can **adjust processing parameters**.

Attention: if you want to have your features annotated, switch **MS2Query** to **ON**. This will increase calculation time.

You can find more information on input data formats and parameter settings in the FERMO Wiki.

You can start the analysis by clicking the **'Start FERMO'** button at the bottom of the page.

Load peaktable (*_quant_full.csv): Browse... No file selected.

Load the MS/MS file (*.mgf): Browse... No file selected.

Specify format of quantitative biological data: Select...

Load quantitative biological data (*.csv): Browse... No file selected.

Load group metadata table (*.csv): Browse... No file selected.

Load spectral library (*.mgf): Browse... No file selected.

General parameters

Mass deviation: 20

Min fragments per MS² spectrum: 8

QuantData factor: 10

Blank factor: 10

Relative intensity filter: 0 1

Networking and annotation parameters

Enable MS2Query annotation

MS2Query relative intensity filter: 0 1

Enable MS2Query blank annotation

Spectral similarity networking algorithm: [✓ Modified Cosine] MS2DeepScore

Fragment similarity tolerance: 0.1

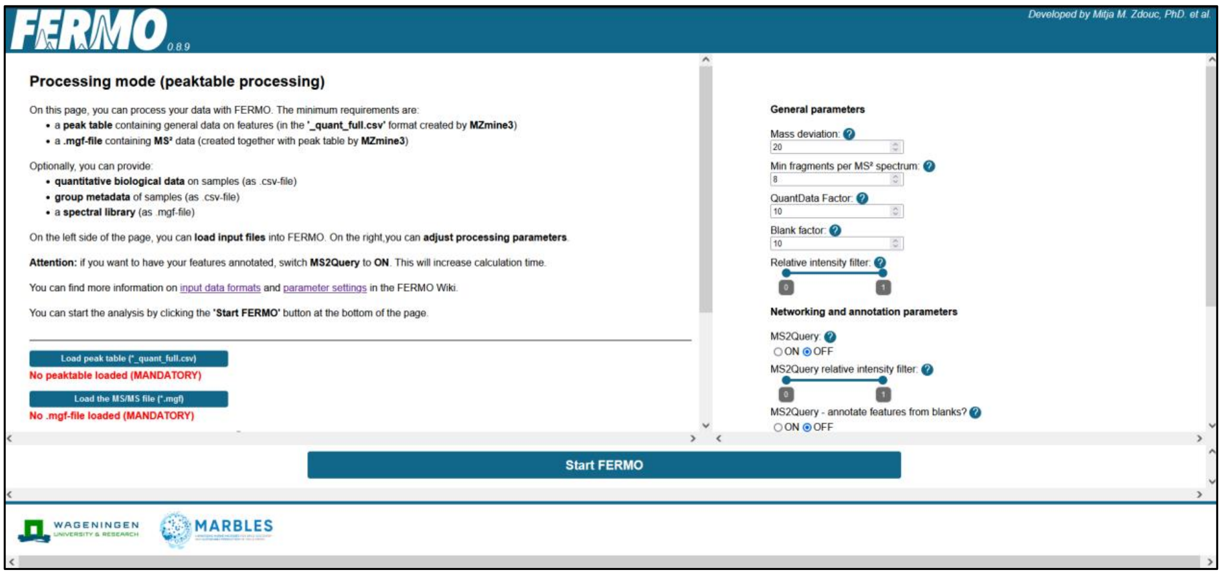
Spectrum similarity score cutoff: 0.8

Maximum spectral links: 10

Minimum matched peaks: 8

Start FERMO

WAGENINGEN UNIVERSITY & RESEARCH
MARBLES
VAN DER HOOFT Computational Metabolomics Group


FERMO 0.8.9

Developed by Mitja M. Zdouc, PhD. et al.

Processing mode (peaktable processing)

On this page, you can process your data with FERMO. The minimum requirements are:

- a **peak table** containing general data on features (in the **'_quant_full.csv'** format created by **MZmine3**)
- a **.mgf-file** containing **MS²** data (created together with peak table by **MZmine3**)

Optionally, you can provide:

- **quantitative biological data** on samples (as .csv-file)
- **group metadata** of samples (as .csv-file)
- a **spectral library** (as .mgf-file)

On the left side of the page, you can **load input files** into FERMO. On the right,you can **adjust processing parameters**.

Attention: if you want to have your features annotated, switch **MS2Query** to **ON**. This will increase calculation time.

You can find more information on input data formats and parameter settings in the FERMO Wiki.

You can start the analysis by clicking the **'Start FERMO'** button at the bottom of the page.

Load peak table (*_quant_full.csv)

No peaktable loaded (MANDATORY)

Load the MS/MS file (*.mgf)

No .mgf-file loaded (MANDATORY)

General parameters

Mass deviation: 20

Min fragments per MS² spectrum: 8

QuantData Factor: 10

Blank factor: 10

Relative intensity filter: 0 1

Networking and annotation parameters

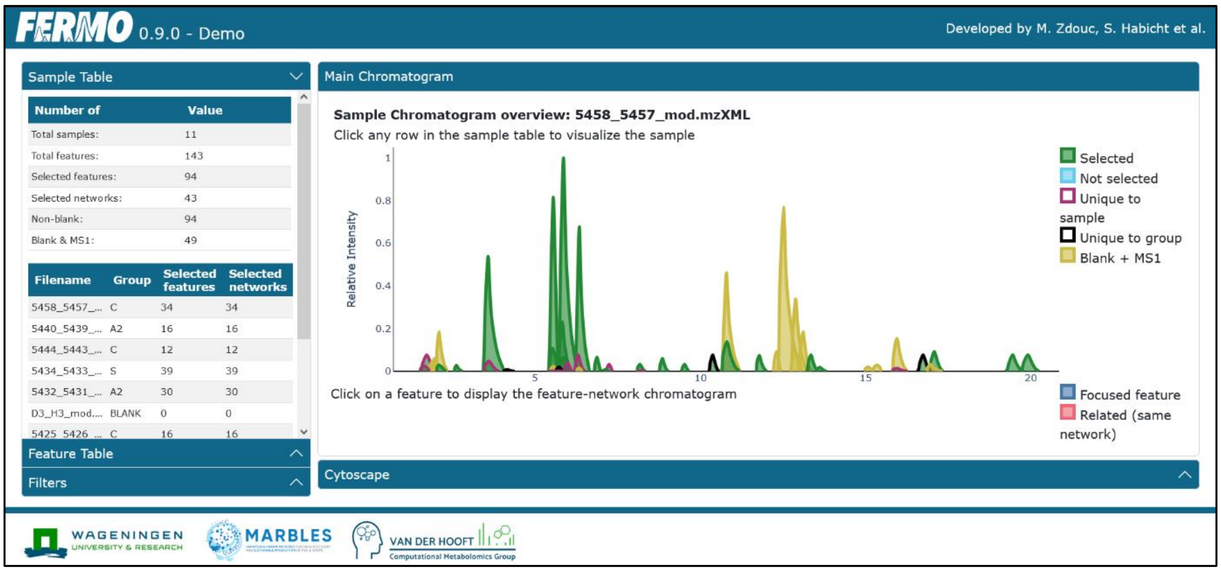
MS2Query: ON [✓ OFF]

MS2Query relative intensity filter: 0 1

MS2Query - annotate features from blanks? ON [✓ OFF]

Start FERMO

Supplementary Figure 2: New layout of the processing page (top) compared to old layout (bottom).


FERMO 0.9.0 - Demo

Developed by M. Zdouc, S. Habicht et al.

Sample Table

Number of	Value
Total samples:	11
Total features:	143
Selected features:	94
Selected networks:	43
Non-blank:	94
Blank & MS1:	49

Filename	Group	Selected features	Selected networks
5458_5457_...	C	34	34
5440_5439_...	A2	16	16
5444_5443_...	C	12	12
5434_5433_...	S	39	39
5432_5431_...	A2	30	30
D3_H3_mod....	BLANK	0	0
5425_5426_...	C	16	16

Feature Table

Filters

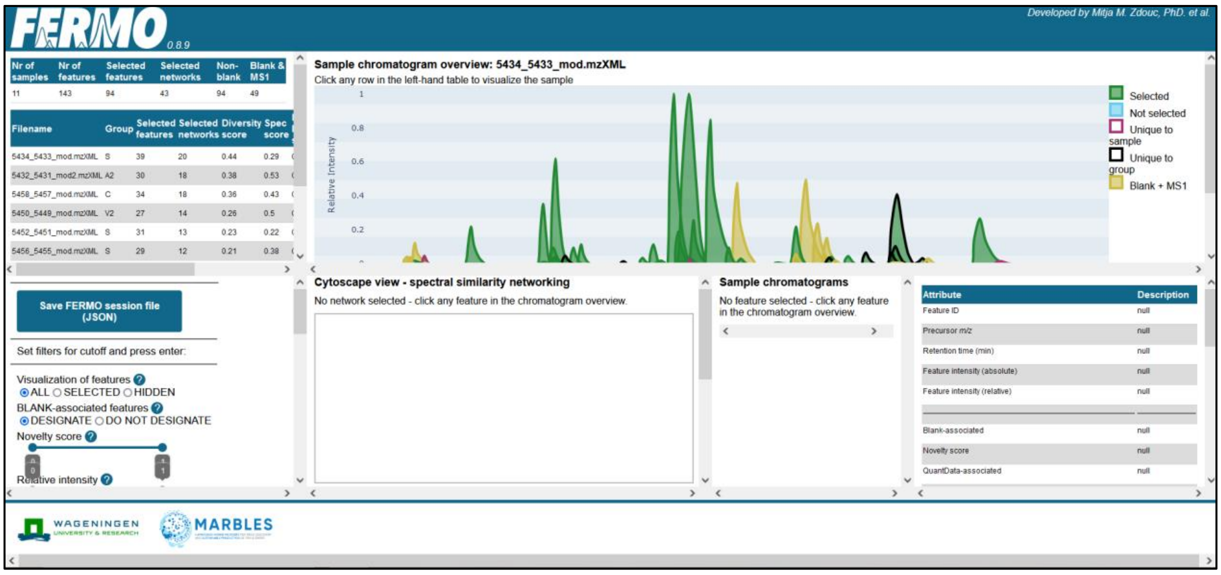
Main Chromatogram

Sample Chromatogram overview: 5458_5457_mod.mzXML

Click any row in the sample table to visualize the sample

Click on a feature to display the feature-network chromatogram

Cytoscape


FERMO 0.8.9

Developed by Mitja M. Zdouc, PhD. et al.

Nr of samples	Nr of features	Selected features	Selected networks	Non-blank	Blank & MS1
11	143	94	43	94	49

Filename	Group	Selected features	Selected networks	Diversity score	Spec score
5434_5433_mod.mzXML	S	39	20	0.44	0.29
5432_5431_mod2.mzXML	A2	30	18	0.38	0.53
5458_5457_mod.mzXML	C	34	18	0.36	0.43
5450_5449_mod.mzXML	V2	27	14	0.26	0.5
5452_5451_mod.mzXML	S	31	13	0.23	0.22
5456_5455_mod.mzXML	S	29	12	0.21	0.38

Save FERMO session file (JSON)

Set filters for cutoff and press enter:

Visualization of features [✓ ALL] SELECTED HIDDEN

BLANK-associated features [✓ DESIGNATE] DO NOT DESIGNATE

Novelty score

Relative intensity

Sample chromatogram overview: 5434_5433_mod.mzXML

Click any row in the left-hand table to visualize the sample

Cytoscape view - spectral similarity networking

No network selected - click any feature in the chromatogram overview.

Sample chromatograms

No feature selected - click any feature in the chromatogram overview.

Attribute	Description
Feature ID	null
Precursor *m/z*	null
Retention time (min)	null
Feature intensity (absolute)	null
Feature intensity (relative)	null
Blank-associated	null
Novelty score	null
QuantData-associated	null

Supplementary Figure 3: New layout of the dashboard (top) compared to old layout (bottom). Both versions are depicted, as displayed when the dashboard is first accessed.

12 Supplementary References

[1] A. Bouslimani, L. M. Sanchez, N. Garg, and P. C. Dorrestein, "Mass spectrometry of natural products: current, emerging and future technologies," *Nat Prod Rep,* vol. 31, no. 6, pp. 718-29, Jun 2014, doi: 10.1039/c4np00044g.

[2] A. Bauermeister, H. Mannochio-Russo, L. V. Costa-Lotufo, A. K. Jarmusch, and P. C. Dorrestein, "Mass spectrometry-based metabolomics in microbiome investigations," *Nature Reviews Microbiology,* vol. 20, no. 3, pp. 143-160, 2022/03/01 2022, doi: 10.1038/s41579-021-00621-9.

[3] J. J. Cui, L. Y. Wang, Z. R. Tan, H. H. Zhou, X. Zhan, and J. Y. Yin, "Mass Spectrometry-Based Personalized Drug Therapy," *Mass Spectrom Rev,* vol. 39, no. 5-6, pp. 523-552, Sep 2020, doi: 10.1002/mas.21620.

[4] T. F. Jorge, A. T. Mata, and C. Antonio, "Mass spectrometry as a quantitative tool in plant metabolomics," *Philos Trans A Math Phys Eng Sci,* vol. 374, no. 2079, Oct 28 2016, doi: 10.1098/rsta.2015.0370.

[5] A. K. Malik, C. Blasco, and Y. Pico, "Liquid chromatography-mass spectrometry in food safety," *J Chromatogr A,* vol. 1217, no. 25, pp. 4018-40, Jun 18 2010, doi: 10.1016/j.chroma.2010.03.015.

[6] F. Hernandez, J. V. Sancho, M. Ibanez, E. Abad, T. Portoles, and L. Mattioli, "Current use of high-resolution mass spectrometry in the environmental sciences," *Anal Bioanal Chem,* vol. 403, no. 5, pp. 1251-64, May 2012, doi: 10.1007/s00216-012-5844-7.

[7] T. J. H. Jonkers *et al.*, "High-Performance Data Processing Workflow Incorporating Effect-Directed Analysis for Feature Prioritization in Suspect and Nontarget Screening," *Environ Sci Technol,* vol. 56, no. 3, pp. 1639-1651, Feb 1 2022, doi: 10.1021/acs.est.1c04168.

[8] F. L. Sauvage, N. Picard, F. Saint-Marcoux, J. M. Gaulier, G. Lachatre, and P. Marquet, "General unknown screening procedure for the characterization of human drug metabolites in forensic toxicology: applications and constraints," *J Sep Sci,* vol. 32, no. 18, pp. 3074-83, Sep 2009, doi: 10.1002/jssc.200900092.

[9] S. Y. Lee, S. T. Lee, S. Suh, B. J. Ko, and H. B. Oh, "Revealing Unknown Controlled Substances and New Psychoactive Substances Using High-Resolution LC-MS-MS Machine Learning Models and the Hybrid Similarity Search Algorithm," *J Anal Toxicol,* vol. 46, no. 7, pp. 732-742, Aug 13 2022, doi: 10.1093/jat/bkab098.

[10] B. Le Dare, S. Allard, A. Couette, P. M. Allard, I. Morel, and T. Gicquel, "Comparison of Illicit Drug Seizures Products of Natural Origin Using a Molecular Networking Approach," *Int J Toxicol,* vol. 41, no. 2, pp. 108-114, Mar-Apr 2022, doi: 10.1177/10915818211065161.

[11] J. S. Yu, H. Seo, G. B. Kim, J. Hong, and H. H. Yoo, "MS-Based Molecular Networking of Designer Drugs as an Approach for the Detection of Unknown Derivatives for Forensic and Doping Applications: A Case of NBOMe Derivatives," *Anal Chem,* vol. 91, no. 9, pp. 5483-5488, May 7 2019, doi: 10.1021/acs.analchem.9b00294.

[12] T. Pluskal, S. Castillo, A. Villar-Briones, and M. Oresic, "MZmine 2: modular framework for processing, visualizing, and analyzing mass spectrometry-based molecular profile data," *BMC Bioinformatics,* vol. 11, p. 395, Jul 23 2010, doi: 10.1186/1471-2105-11-395.

[13] H. L. Rost *et al.*, "OpenMS: a flexible open-source software platform for mass spectrometry data analysis," *Nat Methods,* vol. 13, no. 9, pp. 741-8, Aug 30 2016, doi: 10.1038/nmeth.3959.

[14] C. A. Smith, E. J. Want, G. O'Maille, R. Abagyan, and G. Siuzdak, "XCMS: Processing Mass Spectrometry Data for Metabolite Profiling Using Nonlinear Peak Alignment, Matching, and Identification," *Analytical Chemistry,* vol. 78, no. 3, pp. 779-787, Jan 7 2006, doi: https://doi.org/10.1021/ac051437y.

[15] M. F. Clasquin, E. Melamud, and J. D. Rabinowitz, "LC-MS Data Processing with MAVEN: A Metabolomic Analysis and Visualization Engine," in *Current Protocols in Bioinformatics*, 2012.

[16] Z. Pang *et al.*, "Using MetaboAnalyst 5.0 for LC-HRMS spectra processing, multi-omics integration and covariate adjustment of global metabolomics data," *Nat Protoc,* vol. 17, no. 8, pp. 1735-1761, Aug 2022, doi: 10.1038/s41596-022-00710-w.