



POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

RamApp: a modern toolbox for the processing and analysis of hyperspectral imaging data

TESI DI LAUREA MAGISTRALE IN
COMPUTER SCIENCE AND ENGINEERING -
INGEGNERIA INFORMATICA

Author: **Elia Broggio**

Student ID: 927938

Advisor: Prof. Rosario Michael Piro

Co-advisors: Prof. Dario Polli

Academic Year: 2022-23

Abstract

The processing and analysis of hyperspectral imaging data in various fields, such as biomedical research and material science, often necessitate custom software development or the use of specialized commercial tools. However, these options frequently present challenges, including the need for programming expertise, difficulty in use, high costs, or a combination of these factors. Consequently, researchers' ability to efficiently perform various analyses is hindered, limiting research output and collaboration. This problem is further exacerbated by the limited computing resources available.

To address these issues, RamApp was developed as a comprehensive solution. It is a free and web-based application designed to be intuitive, interactive and user-friendly, enabling researchers from diverse backgrounds to effectively process, explore and analyze hyperspectral imaging data. Its web-based nature allows access through any modern browser and operating system without necessitating a local installation or computing resources. Moreover, users can seamlessly benefit from new features and bug fixes.

Supporting both popular open and commercial file formats, RamApp promotes data interoperability and provides a versatile tool for users of commercial and custom-built instruments. Easy export options for raw and processed data, as well as high-quality images, facilitate downstream analysis and publication.

RamApp offers several spectral and spatial preprocessing methods and algorithms, along with various analysis and visualisation features for hyperspectral data. These include cropping, denoising, substrate identification and correction, clustering, spectral unmixing (MCR, N-FINDR) and the creation of masks and intensity maps.

Although the application is primarily tailored for Raman spectroscopy data, its fundamental features also make it compatible with other kinds of hyperspectral data.

Keywords: hyperspectral imaging, data processing, data analysis, Raman spectroscopy, web application, chemometrics

Abstract in lingua italiana

L'elaborazione e l'analisi dei dati di imaging iperspettrale, utilizzati in vari ambiti come la ricerca biomedica e la scienza dei materiali, richiedono lo sviluppo di codice e software personalizzati o alternativamente l'uso di strumenti commerciali. Tuttavia, entrambe le opzioni presentano dei limiti come la necessità di possedere competenze di programmazione, la difficoltà di utilizzo, i costi elevati o una combinazione di questi fattori. Tutto ciò limita la capacità dei ricercatori di analizzare i dati in modo efficiente, impattando direttamente sulla produttività. Inoltre, questi problemi sono aggravati dalle limitate risorse di calcolo disponibili.

Per far fronte a questi ostacoli è stata sviluppata RamApp, un'applicazione web gratuita appositamente progettata per essere intuitiva, interattiva e di facile utilizzo, consentendo agli utenti di elaborare, esplorare e analizzare efficacemente dati di imaging iperspettrale. Essendo un'applicazione web, RamApp può essere utilizzata attraverso qualsiasi browser e sistema operativo senza richiedere un'installazione locale o impegnare risorse di calcolo. Inoltre, gli utenti possono beneficiare dell'implementazione di nuove funzionalità e di correzioni di bug senza dover scaricare alcun aggiornamento software.

Potendo importare immagini iperspettrali provenienti da formati aperti e proprietari, RamApp consente l'interoperabilità dei dati fornendo uno strumento versatile sia per i ricercatori che utilizzano strumenti commerciali sia per chi ha costruito su misura il proprio setup di acquisizione.

L'applicazione offre diversi algoritmi di elaborazione dei dati iperspettrali a livello spaziale e spettrale, oltre a funzionalità di analisi e visualizzazione. Tra queste ci sono il cropping, il denoising, l'identificazione e rimozione del substrato, il clustering, la scomposizione spettrale (MCR, N-FINDR) e la creazione di maschere e mappe di intensità.

Sebbene RamApp sia stata pensata per i dati di spettroscopia Raman, le sue caratteristiche la rendono compatibile anche con altri tipi di dati iperspettrali.

Parole chiave: imaging iperspettrale, elaborazione dati, analisi dati, spettroscopia Raman, applicazione web, chemometria

Contents

Abstract	i
Abstract in lingua italiana	iii
Contents	v
1 Introduction	1
1.1 Project context and actors involved	1
1.1.1 3rdPlace S.r.l.	1
1.1.2 Politecnico di Milano	1
1.1.3 Research projects	2
1.2 Structure of the Thesis	4
2 Preliminaries	5
2.1 Hyperspectral data	5
2.2 Raman spectroscopy	6
2.3 Preprocessing and analysis software	9
2.3.1 State of the art	9
3 System description	13
3.1 Graphical interface	14
3.2 Functionalities	15
3.2.1 Data import and export	15
3.2.2 Preprocessing	15
3.2.3 Analysis	29
3.3 Architecture	36
3.3.1 Frontend	36
3.3.2 Backend	37
3.3.3 Computational module	39

4	Use case example	41
4.1	Data import	41
4.2	Data exploration	42
4.3	Preprocessing	43
4.3.1	Spikes removal	43
4.3.2	Fluorescence baseline removal	44
4.3.3	Substrate background removal	44
4.3.4	Smoothing	45
4.3.5	Truncation	46
4.3.6	Normalisation	46
4.4	Analysis	46
4.4.1	Univariate analysis	47
5	Conclusions and future developments	49
	Bibliography	51
	List of Figures	55
	Acknowledgements	57

1 | Introduction

This thesis presents the development and implementation of RamApp, a web-based application designed to provide an accessible, intuitive and powerful platform for hyperspectral data analysis.

Before going into details, in this chapter some context about the project will be provided.

1.1. Project context and actors involved

1.1.1. 3rdPlace S.r.l.

This project was developed in 2022 during my curricular internship period at **3rdPlace S.r.l.**, an innovative SME part of **Datrix S.p.A.** group.

Founded in 2010, 3rdPlace is a data-driven tech company that specializes in machine learning model serving, data governance and data science. Recently, it merged with ARAMIS S.r.l. to become Aramix S.r.l..



Figure 1.1: 3rdPlace S.r.l. and Datrix S.p.A. logos.

The company is also a technological partner of international consortia for important Research & Development projects funded by European or national institutions and based on Artificial Intelligence.

1.1.2. Politecnico di Milano

The need for a new tool to process and analyse hyperspectral imaging data emerged from interactions between 3rdPlace and research groups at **Politecnico di Milano**, specifically with members of **Nonlinear Optical Microscopy Lab (VIBRA)** from the Physics

Department.

In this lab, Professor Dario Polli and his team, which includes a researcher from the Italian national research council (**Consiglio Nazionale delle Ricerche, CNR**), are developing a next-generation microscopy system based on coherent Raman spectroscopy, which is capable of rapidly visualising the chemical content of a biological sample to identify tumours in human biopsies with greater accuracy and reproducibility than current methods.

Given that the team developed their own custom-built setups for both coherent and spontaneous Raman microscopy, commercial software solutions for managing experiment results were found to be incompatible with these systems. Consequently, the team found themselves in need of a flexible and adaptable tool to process and analyse their data. This necessity led to the conception and development of **RamApp**, a platform designed to accommodate various hyperspectral imaging data, including those obtained from custom-built Raman microscopy systems.

1.1.3. Research projects

The collaboration between 3rdPlace and Politecnico di Milano took place in the context of two research projects: **NEWMED** and **CRIMSON**.

NEWMED

NEWMED is a collaborative research project co-funded by the European Regional Development Fund of the Lombardy Region that aims to develop innovative solutions and technologies to address the challenges faced by the healthcare industry. The project focuses on various aspects of healthcare, such as diagnostics, therapeutics and medical devices, with the ultimate goal of improving patient outcomes and reducing healthcare costs.



Figure 1.2: NEWMED logo.

Through the collaboration of various stakeholders, including academic institutions, research organisations and private companies, NEWMED seeks to foster innovation and accelerate the translation of research findings into practical applications. By working to-

gether, project participants aim to develop new methods and tools that will help advance the state of the art in healthcare and contribute to the well-being of the population.

Some of the key areas of focus for the NEWMED project include:

1. Development of novel diagnostic techniques, such as advanced imaging methods and biomarker identification, to facilitate early detection and more accurate diagnosis of diseases.
2. Exploration of new therapeutic approaches, including targeted drug delivery systems and personalized medicine, to enhance treatment efficacy and minimize side effects.
3. Advancement of medical devices and technologies, such as wearable sensors and telemedicine platforms, to improve patient monitoring and healthcare delivery.

By addressing these and other challenges in healthcare, the NEWMED project aims to contribute to a more sustainable and efficient healthcare system that is better equipped to meet the needs of the growing and aging population.

CRIMSON

CRIMSON, acronym for Coherent Raman Imaging for the Molecular Study of the Origin of Diseases, is a project funded by the European Union that aims to develop an innovative biophotonic system based on vibrational spectroscopy for cell/tissue imaging, which will be used as a research tool to understand the cellular origin of diseases, allowing novel approaches toward personalised therapy.

The main impact of CRIMSON will be the development of a non-invasive, label-free optical microscopy-endoscopy tool, based on **broadband coherent Raman scattering (CRS)** spectroscopy, for fast, quantitative and objective imaging of biological specimens like 2D/3D cells, tissue sections or organs, to determine their morphological and molecular nature with an unprecedented precision.



Figure 1.3: CRIMSON logo.

Coherent Raman microscopy enables real-time observation of a cell and the mapping of various chemical species' concentrations at any given moment. In biology, this technique offers valuable insights into the spatial distribution of proteins, lipids, DNA, water and

other cellular components. It also eliminates the need for sample preparation with contrast agents, which could disrupt or even contaminate the sample, altering its biological function. Moreover, this method allows for remote analysis and minimizes the risk of sample damage under investigation, as its faster acquisition time compared to spontaneous Raman methods reduces the likelihood of photodamage.

CRIMSON brings together a multidisciplinary team of academic organisations, biomedical end users and innovative SMEs with the aim to bridge the gap between research and product development, increasing the Technology Readiness Level (TRL) and making CRS a user-friendly, robust and cost-effective mainstream tool for a vast biological research community.

1.2. Structure of the Thesis

The document is structured in the following Chapters.

- Chapter 2 contains essential preliminaries information, laying the foundation for a better understanding of the concepts and topics discussed in the subsequent chapters and clarifying some aspects already mentioned in this introduction.
- Chapter 3 covers the entire system description, detailing all the functionalities implemented within RamApp and its overall architecture.
- Chapter 4 presents a practical use case example that shows RamApp's capabilities and how it can be applied to real-world scenarios.
- Chapter 5 provides a summary of the conclusions drawn from the project and describes potential future developments.

2 | Preliminaries

In this chapter, essential concepts and foundational knowledge pertinent to the subject matter are presented.

2.1. Hyperspectral data

Hyperspectral imaging (HSI) data, also referred to as imaging spectroscopy, is a type of data that combine spatial and spectral information, resulting in a **three-dimensional data cube**. In this data cube, each pixel corresponds to a spectrum which contains the reflectance, emissivity, radiance or other arbitrary unit values at various wavelengths or wavenumbers.

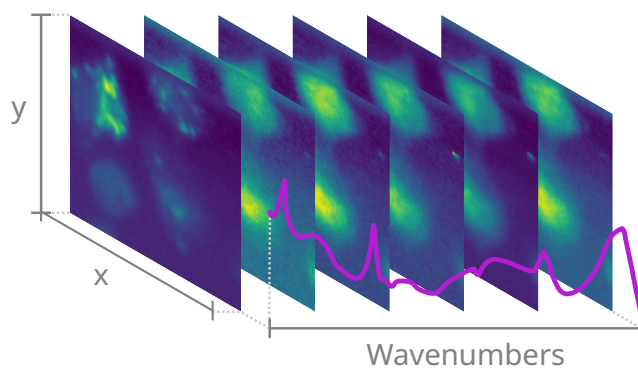


Figure 2.1: A 3D representation of HSI data cube.

These data can provide valuable information about the chemical composition, molecular structure and physical properties of materials. This is because different chemical compounds and materials exhibit unique **spectral signatures**, or patterns of reflectance or absorption across different wavelengths of the electromagnetic spectrum.

By analyzing these spectral signatures, researchers can gain insights into the properties of the materials being studied. For example, the spectral signature of a mineral can be used to identify its chemical composition and distinguish it from other minerals that have different spectral signatures. Similarly, the spectral signature of a plant leaf can be

used to identify the presence of specific pigments or other molecules that are indicative of plant health or stress. For these reasons, imaging spectroscopy has a wide range of applications, including remote sensing, agriculture, geology, environmental monitoring, biomedical imaging and many others.

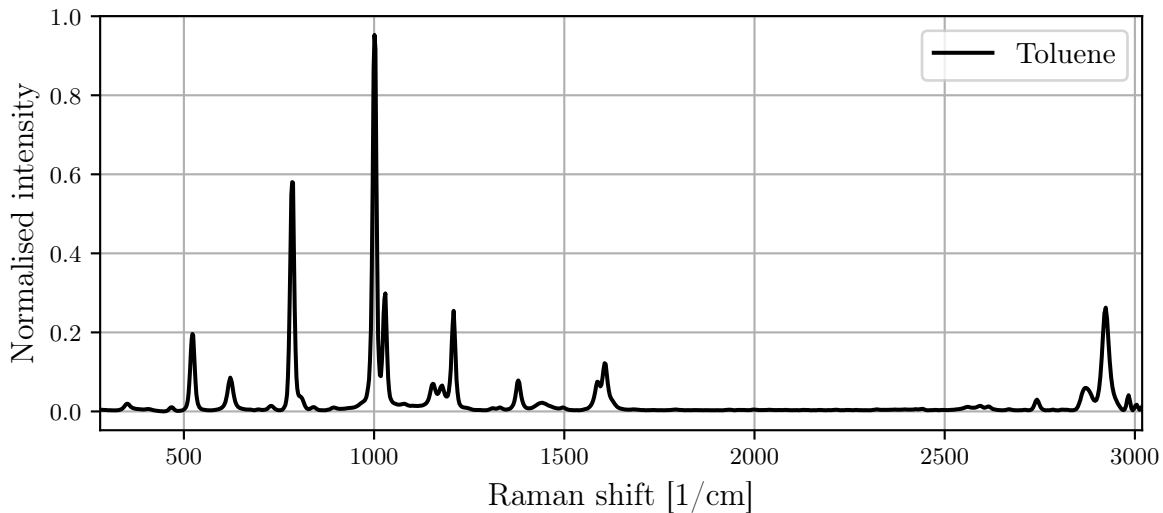


Figure 2.2: Raman spectral signature of toluene.

The collection of hyperspectral data is based on various types of spectroscopy, each of which can employ a different portion of the electromagnetic spectrum. Some examples of spectroscopy techniques include: **infrared spectroscopy**, **Ultraviolet-Visible spectroscopy**, **mass spectrometry** and **Raman spectroscopy**.

Although a discussion of the various spectroscopy techniques is beyond the scope of this thesis, Raman spectroscopy, which is particularly relevant to the presented work, will be explained in greater detail in the following section. In fact, the name «RamApp» is a portmanteau that combines the words «Raman», «Map» and «App», reflecting its primary focus on Raman spectroscopy.

2.2. Raman spectroscopy

Raman spectroscopy is a non-destructive analytical technique based on inelastic scattering of monochromatic light, typically from a laser source. It provides valuable information about the vibrational modes and molecular structure of a sample, making it an essential tool for the study of a wide range of materials, including chemicals, polymers, biological samples and minerals.

When a laser beam interacts with a sample, most of the scattered light has the same frequency as the incident light, which is known as elastic or **Rayleigh scattering**. However, a small portion of the scattered light has a frequency different from the incident light due to the molecular vibrations of the sample. This inelastic scattering is called **Raman scattering** and it can be either Stokes or anti-Stokes, depending on whether the scattered light has a lower or higher frequency than the incident light, respectively.

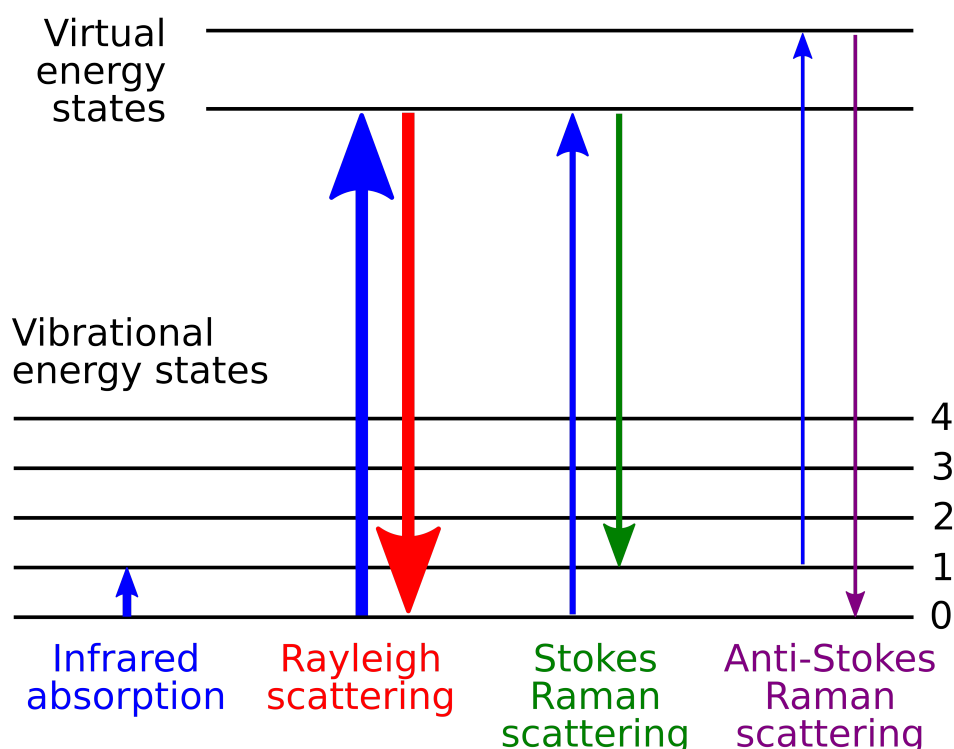


Figure 2.3: Molecular energy levels and Raman effect [1].

In Raman spectroscopy, the difference in frequency between the incident light and the Raman scattered light (**Raman shift**) is measured. The Raman shift is typically expressed in wavenumbers (cm^{-1}) and plotted against the intensity of the scattered light to generate a Raman spectrum, which serves as a unique molecular fingerprint for the sample. This allows the identification of molecular species and the investigation of molecular interactions, conformations and crystal structures.

There are several types of Raman spectroscopy techniques, each designed to address specific analytical challenges or to optimize the collection of Raman signals. Some common types include Spontaneous Raman Spectroscopy, Resonance Raman Spectroscopy, Surface-Enhanced Raman Spectroscopy (SERS), Stimulated Raman Spectroscopy and

Coherent Anti-Stokes Raman Spectroscopy (CARS). Each technique offers unique advantages and applications, making Raman spectroscopy a versatile tool for the analysis of diverse samples.

Although Raman spectroscopy has been used in a range of specialised areas, including pharmaceutical analysis, geology, mineralogy and environmental monitoring, its applications in the biomedical field are increasingly gaining prominence.

In the following chapters, hyperspectral imaging data obtained from biomedical samples using Raman spectroscopy will be used to demonstrate the key features of RamApp. In particular, they are as follows:

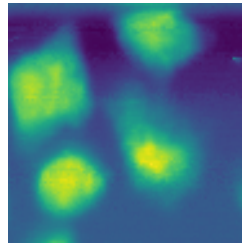


Figure 2.4: Raman imaging of *in vitro* labeled murine microglial cells with nanoformulations of PERFECTA [2].

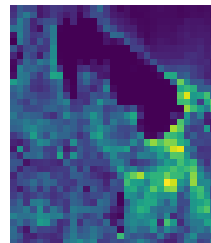


Figure 2.5: Raman imaging of breast tissues [3].

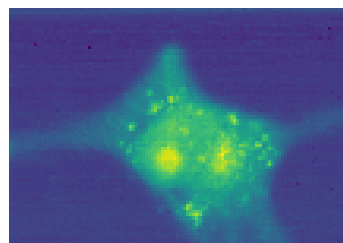


Figure 2.6: Raman imaging of a cultured breast cancer cell (MCF7).

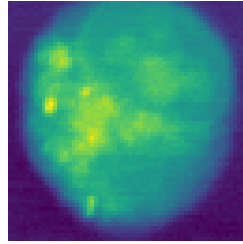


Figure 2.7: Raman imaging of a leukemic cell from a patient suffering from acute myeloid leukemia (AML M5a).

The 2D images presented above depict the HSI cubes by displaying the mean spectrum intensity for each pixel. This offers a visually accessible representation of the hyperspectral data, enabling users to rapidly evaluate and interpret the spatial distribution of the spectral information within the sample.

2.3. Preprocessing and analysis software

Preprocessing and analysis software for hyperspectral data play a crucial role in the efficient handling and interpretation of large and complex datasets generated by hyperspectral imaging systems.

These software packages are designed to facilitate the extraction of meaningful information from a large amount of data by applying various algorithms, statistical methods and visualisation techniques.

A wide range of software tools have been developed over the years, both commercial and open source, each offering unique features and catering to different application domains.

2.3.1. State of the art

Some notable commercial Raman spectroscopy software packages include:

- **WiRE** (Renishaw) [4]: WiRE (Windows-based Raman Environment) is a software package developed by Renishaw for their Raman spectrometers. It offers a range of functionalities for spectral acquisition, processing and analysis, such as baseline correction, peak fitting and multivariate analysis.
- **LabSpec** (HORIBA) [5]: LabSpec is a software suite developed by HORIBA for their Raman spectrometers. It provides a comprehensive set of tools for data acquisition, processing and analysis, including advanced functions like Raman mapping, multivariate curve resolution and particle analysis.

In the open-source domain, several software tools and libraries have been developed for Raman spectroscopy data processing and analysis, often leveraging the power of popular programming languages:

- **hyperSpec** [6]: hyperSpec is an R package designed for handling and processing hyperspectral data, including Raman spectroscopy data. It provides a suite of functions for data import, preprocessing and visualisation.
- **Raman Tool Set**: Raman Tool Set is an open-source software for Raman spectra preprocessing and analysis. It offers a variety of functions, including baseline correction, peak fitting and multivariate analysis.
- **RamanLIGHT** (MATLAB[®] app) [7]: RamanLIGHT is a MATLAB app to preprocess Raman mapping datasets and apply unsupervised unmixing algorithms to find spectra of the pure compounds and create abundance maps.

Commercial tools, while often feature-rich and user-friendly, present certain downsides that can limit their suitability to some researchers. These limitations include:

- **Costs**: Commercial software packages typically come with a significant price tag, which can be prohibitive for researchers with limited budgets, particularly those in academia or smaller research institutions.
- **Limited customization**: Commercial software may not always be easily customizable or adaptable to address unique or specific research requirements. Furthermore, researchers using home-built setups typically face difficulties when attempting to utilize software provided by other vendors for their analyses.
- **Black box algorithms**: Commercial tools may use proprietary algorithms without disclosing the underlying details, making it difficult for researchers to understand and interpret the results fully. This lack of transparency can pose challenges when it comes to validating findings, reproducibility and peer review.
- **Restricted access to updates and support**: Access to software updates and technical support may be limited to paid subscribers or those with active maintenance contracts. This can lead to outdated software versions and hinder researchers from taking advantage of new features or improvements.

On the other hand, the use of open source packages can also present challenges for researchers in certain contexts. Some of the downsides associated with these tools include:

- **Steeper learning curve**: Writing custom code to leverage open tools often necessitates a deeper understanding of programming languages, libraries and algorithms. This

can result in a steeper learning curve for researchers who may not have extensive programming experience.

- Time-consuming development: Developing custom code for data preprocessing and analysis can be time-consuming, particularly for complex research projects. This additional time investment can detract from the core research focus and reduce overall productivity.
- Lack of user-friendly interfaces: Open tools and custom code solutions may not always include intuitive or user-friendly interfaces. This can make the tools less accessible for researchers who are not comfortable with programming or those who prefer a graphical user interface (GUI) for data analysis.
- Code maintenance: Custom code may require ongoing maintenance and updates to remain compatible with new software versions, libraries, or operating systems.

Given the challenges and limitations associated with both commercial tools and open-source solutions that require custom code, the development of a user-friendly, flexible and cloud-based platform for the preprocessing and analysis of hyperspectral data, particularly in the context of Raman spectroscopy, was deemed necessary and consequently undertaken.

Using RamApp, researchers can benefit from a solution that combines the best features of commercial and open-source tools, offering an accessible interface, extensive functionality and adaptability to various research needs.

In the following chapter, a comprehensive and detailed overview of RamApp will be presented, covering its features, functionalities and underlying architecture to offer readers a complete understanding of the application and its potential applications in the field of hyperspectral data analysis.



Figure 2.8: RamApp logo.

3 | System description

The proposed solution is an intuitive, user-friendly and modular web application designed to streamline the processing and analysis of hyperspectral imaging data.

As a web-application, RamApp offers several advantages over traditional desktop applications. One of the key benefits is platform independence, allowing users to access the application from any device with a modern web browser and an internet connection. This ensures compatibility across different operating systems, including Windows, macOS and Linux, providing a consistent user experience.

In addition, web applications facilitate easy deployment and maintenance. Users always access the latest version of the application without needing to install updates manually, as updates and bug fixes can be centrally managed.

Scalability is another advantage of web applications. RamApp can be designed to handle a wide range of users and workloads, making it possible to scale the application as needed. This is particularly useful for managing an increasing number of users or larger hyperspectral datasets over time.

Additionally, web applications generally have lower system requirements for the client-side, as the majority of processing occurs on the server. This enables users to work with large hyperspectral datasets without the need for high-performance local hardware, reducing the barrier to entry for users with limited resources.

Lastly, web applications can easily integrate with other web-based services and APIs, enhancing the functionality of RamApp by incorporating additional features and tools as needed. This integration capability makes it possible to continuously expand and adapt the application to meet the evolving needs of the hyperspectral imaging community.

In the following sections, the various aspects and functionalities of RamApp will be thoroughly detailed and discussed.

3.1. Graphical interface

The web-application interface of RamApp has been designed with an emphasis on user-friendliness and intuitive navigation, aiming to make the platform self-explanatory and easily accessible to researchers. This approach ensures that users can quickly understand the functionalities and efficiently utilize the available tools for preprocessing and analyzing hyperspectral data.

The application's layout consists of three main sections.

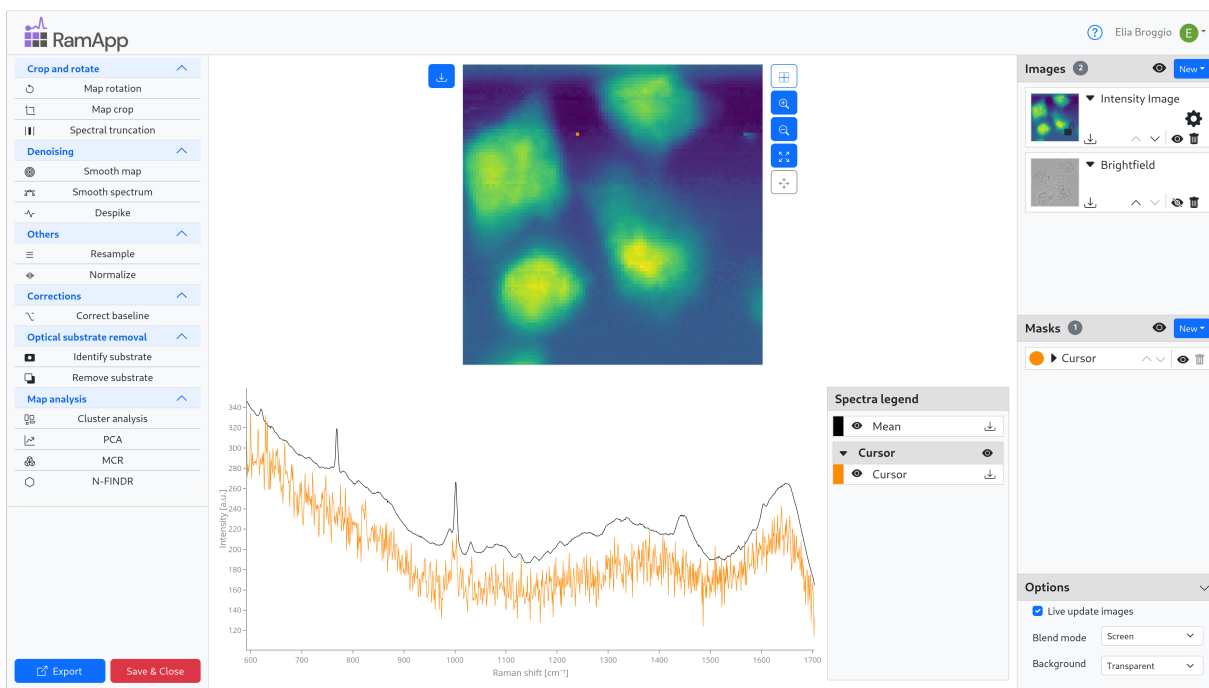


Figure 3.1: Graphical appearance of RamApp for hyperspectral imaging data analysis.

On the left side, a menu provides access to all the preprocessing and analysis functions.

In the centre, the upper part is dedicated to the map visualiser, which displays the images blended together and the masks representing spatial regions of interest. This visualization can be zoomed in and out, offering users a more accurate and detailed view of the map. Immediately below the map visualiser, a plot presents the spectra associated with the data.

On the right side, a menu lists all the generated images and masks, allowing users to easily navigate and select the desired results.

This organization of the interface ensures that users can quickly locate the necessary functions and tools, streamlining their workflow and enhancing their overall experience.

3.2. Functionalities

In this section, a detailed overview of the functionalities incorporated within RamApp is provided, elucidating the various techniques employed to address the complexities of hyperspectral imaging data.

3.2.1. Data import and export

RamApp provides users with the flexibility to work with hyperspectral data in multiple formats, addressing the wide-ranging needs of the research community.

The platform supports importing data from commercial Raman spectroscopy instruments, such as **Renishaw**[™] grid map (.wdf) and **Horiba**[™] LabSpec 5 (.ngc), as well as other formats like **CSV tables** (.csv), **Apache Parquet/Feather** (.parquet/.feather), **MATLAB**[®]/**Octave** (.mat) and custom-defined formats, ensuring compatibility between multiple instruments and software packages.

For exporting the processed hyperspectral cube, RamApp provides options such as CSV tables (.csv), Feather data frames (.feather) and MATLAB[®]/Octave files (.mat), facilitating seamless integration with other data processing and analysis tools and promoting efficient data exchange between researchers. In addition, RamApp enables the export of the entire project as a single file (.zarr [8]), including processed data, generated images and masks, allowing users to easily share complete projects with colleagues or collaborators.

3.2.2. Preprocessing

The necessity of preprocessing hyperspectral imaging data prior to analysis arises from several factors that can significantly affect the quality and reliability of the resulting information.

First, these high-dimensional datasets often contain noise, originating from sensor imperfections or environmental factors, which may obscure the underlying patterns and signals of interest. Preprocessing techniques help reduce this noise and enhance the signal-to-noise ratio, thus improving data quality.

Second, variations in illumination, atmospheric conditions, or sensor calibration can introduce inconsistencies in the spatial and spectral dimensions of the data. When preprocessing methods are applied, these inconsistencies can be mitigated, allowing for more accurate comparisons and interpretations.

Lastly, the sheer volume and complexity of hyperspectral imaging data can present com-

putational challenges, necessitating data reduction techniques to minimise processing time and resource demands. Dimensionality reduction and image compression methods can be employed during preprocessing to achieve this goal, ensuring efficient analysis without compromising data integrity.

In the following paragraphs, the main preprocessing features of RamApp are presented.

Map rotation

This simple preprocessing function enables users to manipulate the spatial orientation of the hyperspectral cube within its spatial axes, allowing for precise alignment and optimal visualisation of specific regions of interest.

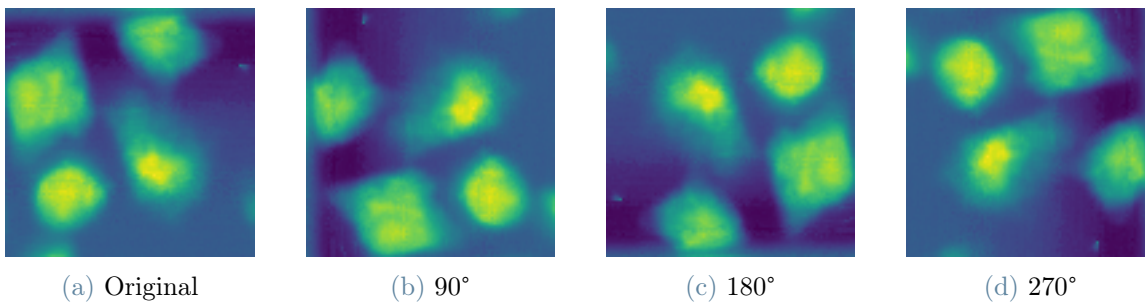


Figure 3.2: Anti-clockwise rotation of the hyperspectral cube over the spatial axes.

Map crop

The *Map crop* function offers users the ability to selectively isolate and extract specific regions of interest from the hyperspectral cube along the spatial axes, significantly enhancing both visualization and computational efficiency.

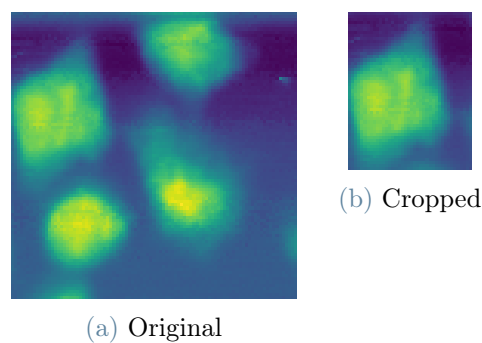


Figure 3.3: Hyperspectral cube cropped over the spatial axes.

Through an intuitive interface, users can define the desired spatial boundaries for cropping,

effectively trimming the cube to retain only the specified region. This tailored approach not only improves visualization by allowing for a closer examination of the selected area but also facilitates more focused and accurate analysis by eliminating extraneous data that may otherwise introduce noise or distractions.

Spectral truncation

The preprocessing function *Spectral truncation* provides users with the capability to reduce the size of the hyperspectral cube by selectively truncating the spectral range. It also supports the selection of non-contiguous spectral regions, enabling users to exclude irrelevant or uninformative sections, such as the silent region in a Raman spectrum. By allowing researchers to focus on specific bands of interest in the spectral axis, this function contributes to a more targeted and efficient analysis process, minimizing processing time and resource demands.

Smooth map

The *Smooth map* function equips users with a spatial denoising tool to enhance the quality of their hyperspectral cube by applying a non-linear digital filter designed to remove noise across the spatial dimensions of the data.

Two distinct filtering methods can be employed for this purpose: the **median filter** and the **Gaussian filter**.

The median filter works by replacing each pixel's value with the median value of the neighboring pixels in a defined window size. This nonlinear filtering method is particularly effective in reducing salt-and-pepper noise while preserving edges, as it considers the local distribution of pixel intensities and maintains sharp transitions.

The Gaussian filter, on the other hand, employs a Gaussian function to calculate the weighted average of neighboring pixel values with a specified scale. The weights decrease with distance from the central pixel, resulting in a smooth transition between adjacent pixels. This filter is adept at reducing Gaussian noise and blurring the image, which can improve the overall quality of the hyperspectral data.

By mitigating the impact of noise on the individual maps corresponding to each wavenumber, this function fosters a clearer representation of the underlying patterns and features.

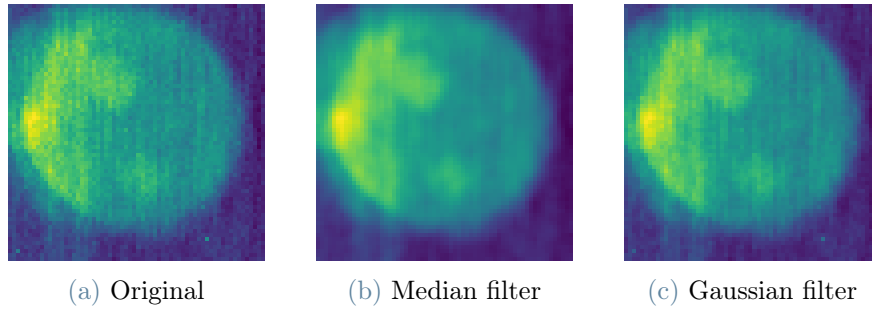


Figure 3.4: Hyperspectral cube smoothed with a median filter (window size 3x3 pixels) and with a Gaussian filter (filter scale $\sigma=0.5$).

Smooth spectrum

The function *Spectral smoothing* offers users the ability to enhance the quality of their hyperspectral cube by applying a filter designed to smooth the spectrum associated with each pixel. This function provides two versatile filter options: the **Savitzky-Golay filter** [9] and the **Whittaker filter** [10].

The Savitzky-Golay filter operates by fitting a low-degree polynomial to a set of adjacent data points using a least-squares approach. The fitted polynomial is then used to estimate the central point of the window. This method is effective in preserving the shape of the original spectrum while reducing high-frequency noise. The filter can be fine-tuned by adjusting the window size and polynomial degree, allowing the user to balance between noise reduction and spectral resolution.

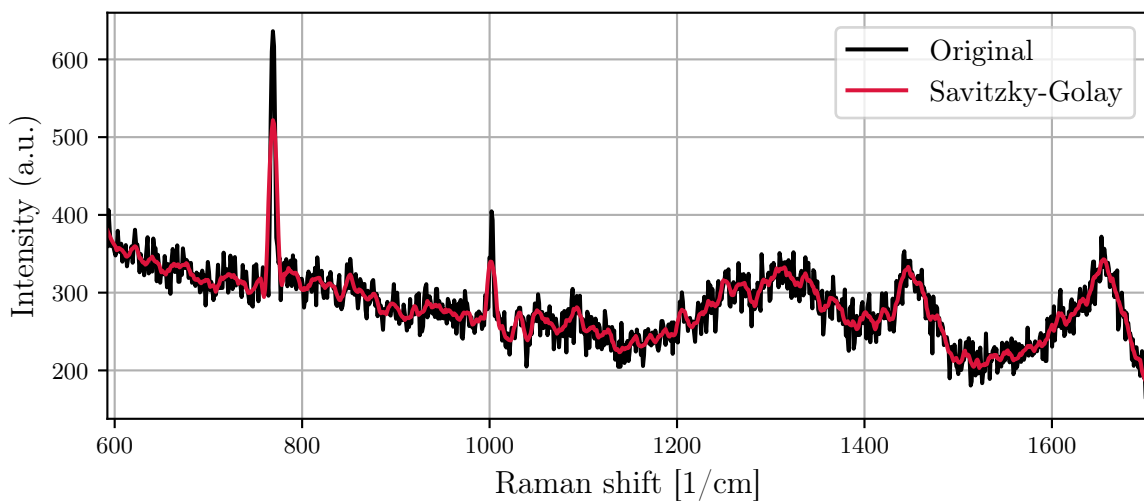


Figure 3.5: A selected pixel of the hyperspectral cube smoothed with Savitzky-Golay filter.

The Whittaker filter, on the other hand, employs a roughness-penalized least squares approach to control variations in the second derivative of the spectrum. This method aims to find a smooth curve that best fits the original data while minimising the sum of squared residuals and the roughness penalty, controlled by a regularisation parameter. The Whittaker filter is particularly useful for removing baseline drifts and preserving sharp peaks in the spectrum.

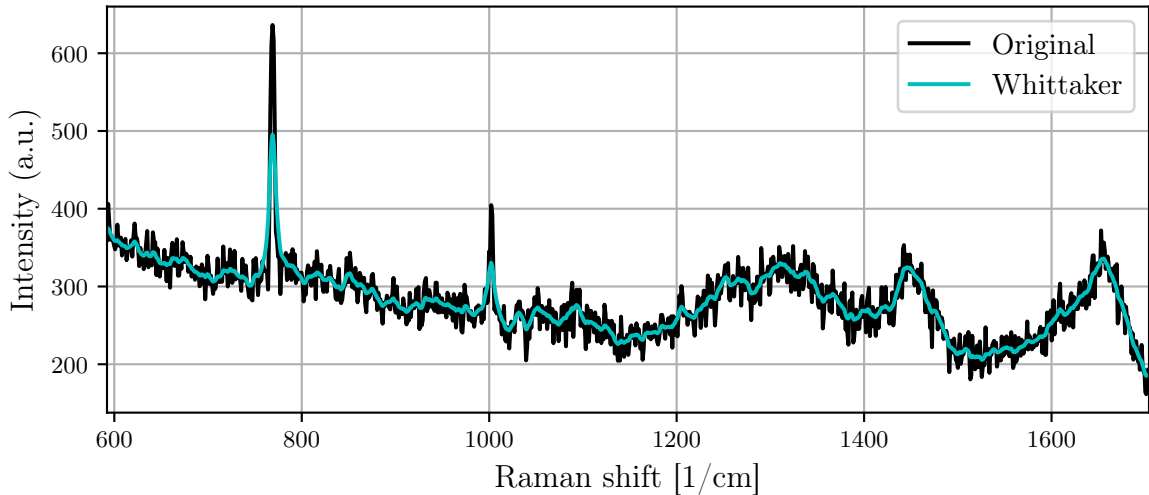


Figure 3.6: A selected pixel of the hyperspectral cube smoothed with Whittaker filter.

Despike

The *Despike* preprocessing function helps to address the issue of bad pixels containing spikes in their signal, which may result from various factors such as cosmic rays. By detecting and correcting these anomalous pixels, the *Despike* function contributes to a more accurate and reliable representation of the hyperspectral data.

Users are provided with a choice of two outlier detection algorithms to suit their specific needs.

1. **Z-score:** The Z-score method identifies intensities in the spectrum that deviate significantly from the mean, flagging them as potential bad pixels. To utilize this method, users must specify a threshold value, which determines the degree of deviation from the mean required for a pixel to be classified as an outlier.
2. **Modified Z-score:** The modified Z-score method offers a more robust approach by employing the median absolute deviation (MAD) as a measure of variability. This method is less sensitive to extreme values in the data and can more accurately identify bad pixels in the presence of such values. Similarly to the Z-score method, users

need to define a threshold value for the modified Z-score method, which dictates the deviation required from the median to classify a pixel as an outlier.

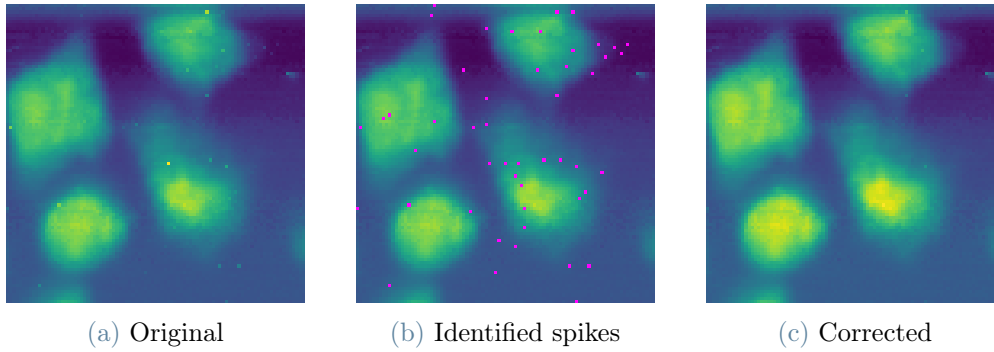


Figure 3.7: Hyperspectral cube presenting spikes detected using the Z-score algorithm and subsequently corrected using a linear interpolation.

In addition to detecting spikes within spectra, the `Despike` preprocessing function also offers the capability to correct these anomalies using linear interpolation. In fact, once the spikes are identified within a spectrum, they can be replaced with interpolated data derived from their neighbouring data points within a specified window.

Resample

The *Resample* method gives users the ability to restructure their hyperspectral data using a new spectral grid featuring equally-spaced nodes. This function is particularly beneficial when comparing datasets with different calibration axes, as it standardizes the data and facilitates more accurate comparisons and analyses.

Five distinct interpolation methods are available to cater to varying requirements: four of them employ spline interpolation (constant, linear, quadratic and cubic), while the fifth utilizes Whittaker's interpolation. Spline interpolation methods, by using lower-degree polynomials over smaller intervals, can help mitigate the issue of Runge's phenomenon, ensuring a smoother and more accurate representation of the data. Whittaker's interpolation, on the other hand, offers a robust alternative.

To utilize this resampling function, users must specify a step size for the new grid. This parameter determines the spacing between the nodes in the resampled spectral grid and directly influences the interpolation process.

Normalize

The preprocessing function for normalisation and scaling offers users a versatile solution to optimise their hyperspectral data by transforming the values into a more standardised and comparable format. With eight distinct normalisation and scaling methods available, users can select the most appropriate approach for their specific requirements.

- **L1 Norm:** By employing the L1 norm method, each spectrum is normalized based on the sum of the absolute values of its elements.
- **L2 Norm:** The L2 norm method normalises each spectrum using the square root of the sum of the squared values of its elements.
- **Max:** This method scales each spectrum by dividing its elements by the maximum value found within the spectrum. For non-negative values, this results in a range between 0 and 1.
- **Scale:** This approach standardises each spectrum by centering the data around the mean and scaling it according to the standard deviation, resulting in a mean of 0 and a standard deviation of 1.
- **Min-Max Scale:** The Min-Max scaling method normalises each spectrum by linearly transforming its values into a range between 0 and 1, using the minimum and maximum values found within the spectrum.
- **Frobenius Norm:** This method normalises the entire data matrix by calculating the Frobenius norm, a measure of the overall magnitude of the data.
- **Wavenumber Normalisation:** This approach normalizes each spectrum relative to a provided wavenumber, ensuring that the values are directly comparable across different wavenumbers.
- **Area:** This method normalises each spectrum according to the area under the curve, ensuring that the integrated values of all spectra are equal.

Correct baseline

The *Correct baseline* preprocessing function offers users a powerful tool to remove the fluorescence baseline.

The fluorescence baseline is an unwanted low-frequency background signal that can be present in hyperspectral data, particularly when dealing with fluorescent materials or samples. It can interfere with the accurate representation and interpretation of the data

as it can obscure or distort the true spectral features and peaks that are of interest to researchers. Removing the fluorescence baseline is a crucial preprocessing step in the analysis of hyperspectral data, as it helps to minimise the impact of the unwanted background signal and enhance the clarity of the underlying spectral features.

With several distinct methods provided by the *pybaselines* library [11], users can select the most appropriate approach for their specific needs. These functions were divided into three categories:

1. Polynomial methods

- (a) *Polynomial*: This method uses a straightforward least-squares fitting approach to model the fluorescence baseline with a polynomial function. The polynomial method is suitable for cases with smooth and relatively predictable baseline shapes.

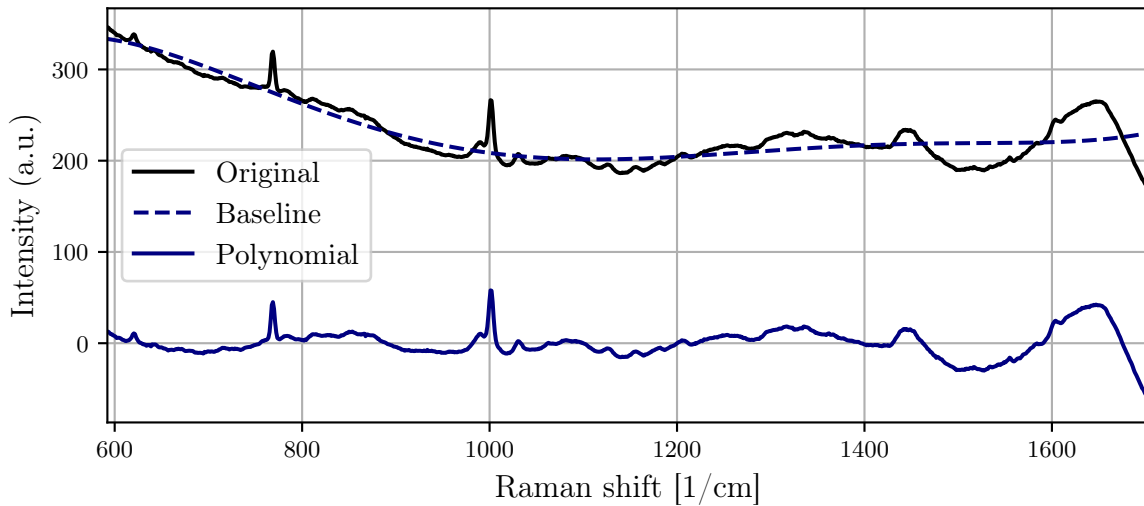


Figure 3.8: Mean spectrum of an hyperspectral cube before and after applying a Polynomial baseline correction.

- (b) *IModPoly (Improved Modified Polynomial)* [12]: The IModPoly method improves on simple polynomial fitting by incorporating an iterative reweighted least-squares fitting process. This approach provides a more robust fit by assigning lower weights to points that are likely to be part of peaks, reducing their influence on the baseline estimate.

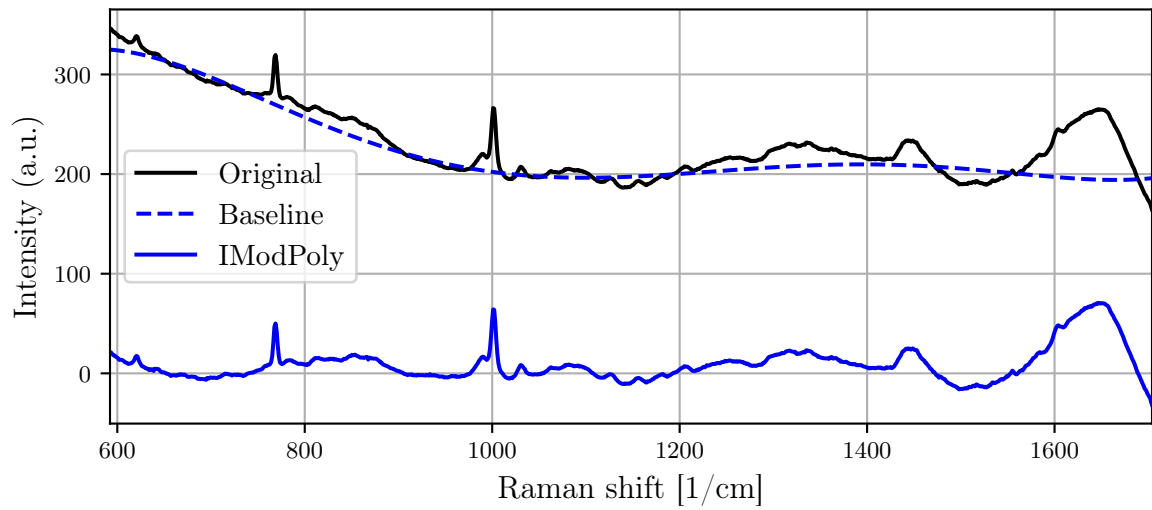


Figure 3.9: Mean spectrum of an hyperspectral cube before and after applying IModPoly baseline correction.

- (c) *Goldindec* [13]: The Goldindec is a highly effective method based on polynomial fitting approach that uses a linear combination of a polynomial function and a discrete Gaussian function to model the baseline. By incorporating the Gaussian component, this method effectively captures the influence of overlapping peaks on the baseline.

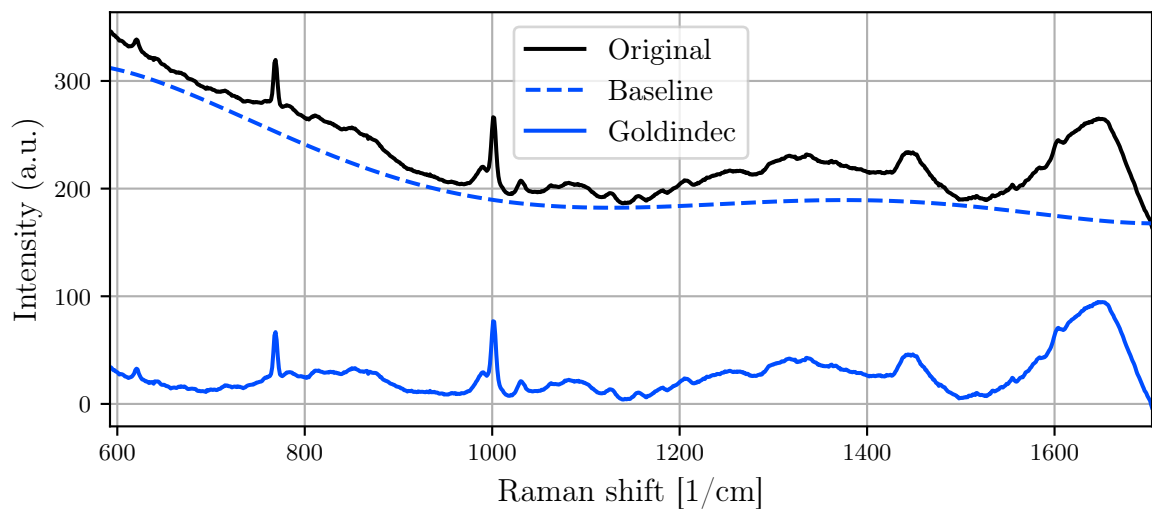


Figure 3.10: Mean spectrum of an hyperspectral cube before and after applying Goldindec baseline correction.

2. PLS-based (Penalised Least Square) methods

- (a) *arPLS* (*Asymmetrically reweighted PLS*) [14]: This method introduces an asymmetric weighting scheme to the PLS-based approach.

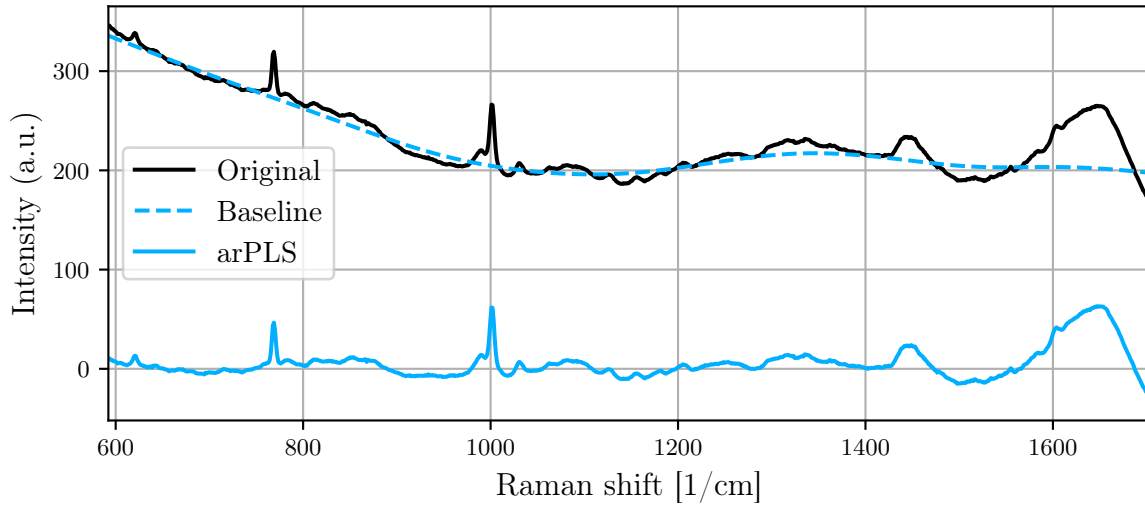


Figure 3.11: Mean spectrum of an hyperspectral cube before and after applying arPLS baseline correction.

- (b) *IarPLS* (*Improved asymmetrically reweighted PLS*) [15]: Building upon the arPLS method, IarPLS further enhances the baseline estimation by incorporating an iterative reweighted least-squares fitting process.

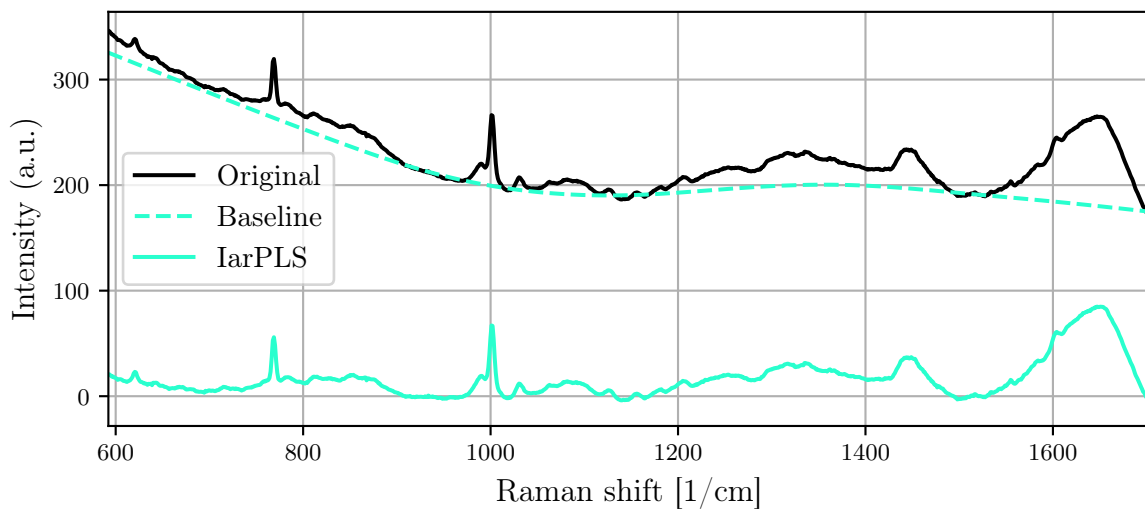


Figure 3.12: Mean spectrum of an hyperspectral cube before and after applying IarPLS baseline correction.

- (c) *drPLS (Doubly reweighted PLS)* [16]: The drPLS method uses a doubly reweighted least squares fitting approach, which combines the benefits of both symmetric and asymmetric weighting schemes.

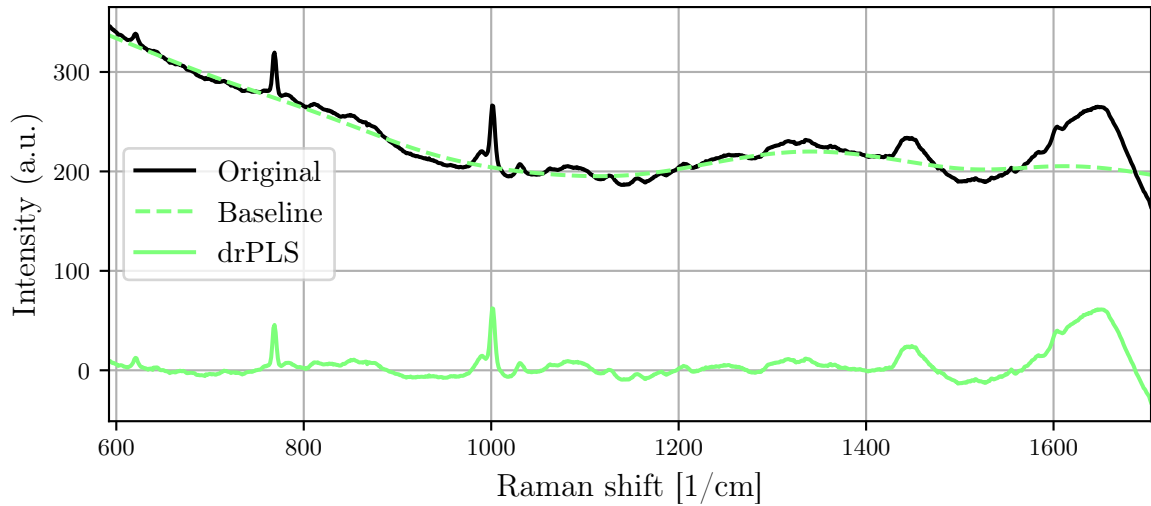


Figure 3.13: Mean spectrum of an hyperspectral cube before and after applying drPLS baseline correction.

- (d) *asPLS (Adaptive smoothness PLS)* [17]: The asPLS method focuses on adaptively adjusting the smoothness of the baseline estimation based on the local characteristics of the spectrum by considering the local variability in the data.

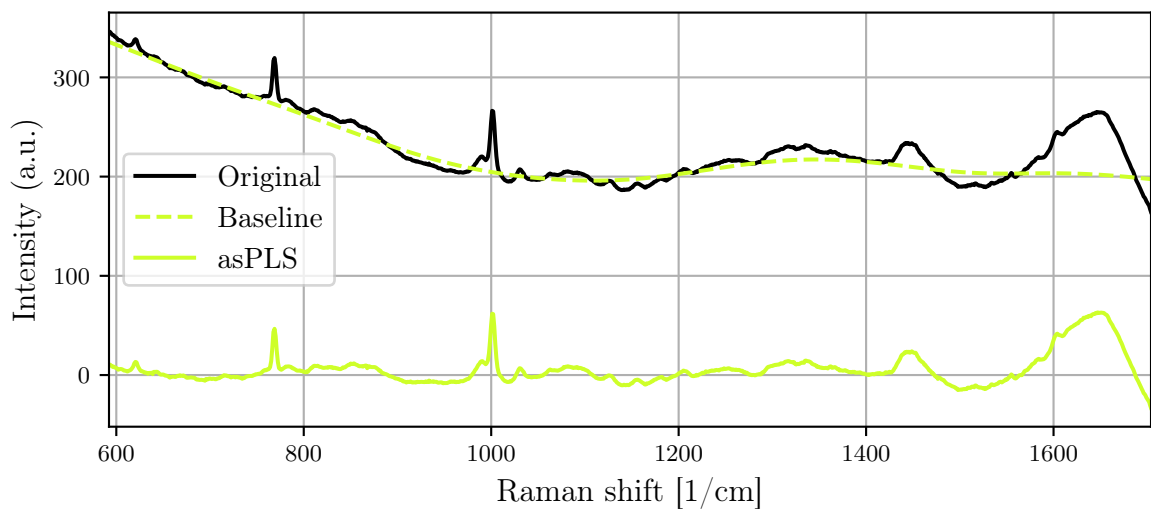


Figure 3.14: Mean spectrum of an hyperspectral cube before and after applying asPLS baseline correction.

3. Miscellaneous

- (a) *Rubberband*: The Rubberband method is a parameter-free approach that estimates the baseline by connecting local minima in the spectrum. This method effectively creates a convex hull around the spectrum, which is then subtracted to remove the baseline. The Rubberband method is particularly useful for cases where the baseline has a simple, convex shape and requires minimal user input.

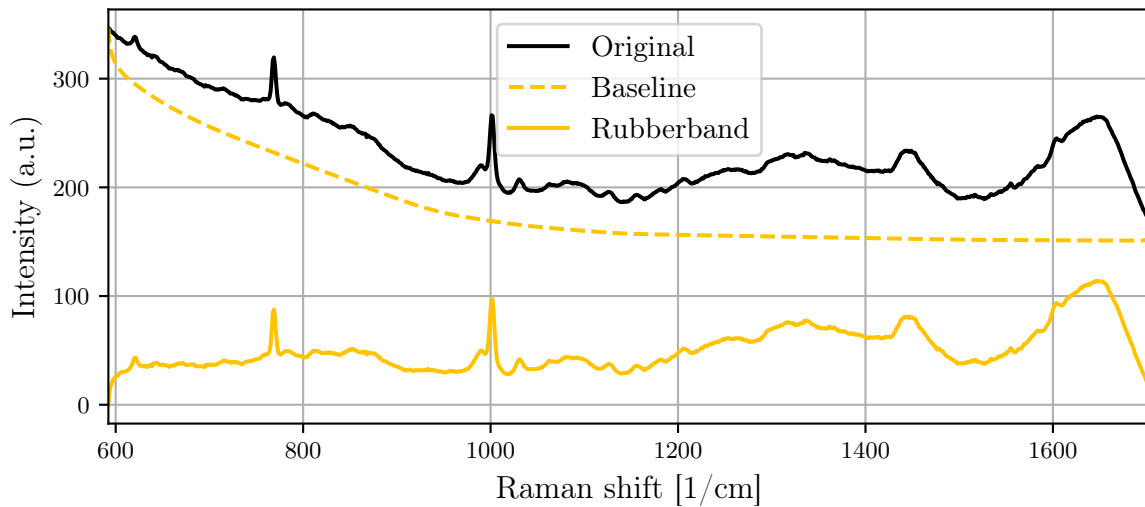


Figure 3.15: Mean spectrum of an hyperspectral cube before and after applying Rubberband baseline correction.

- (b) *BEADS (Baseline estimation and denoising with sparsity)* [18]: The BEADS method is a more advanced approach that uses the concept of sparsity to simultaneously estimate the baseline and denoise the spectrum. This method models the baseline as a smooth, low-frequency component and the noise as a sparse, high-frequency component.

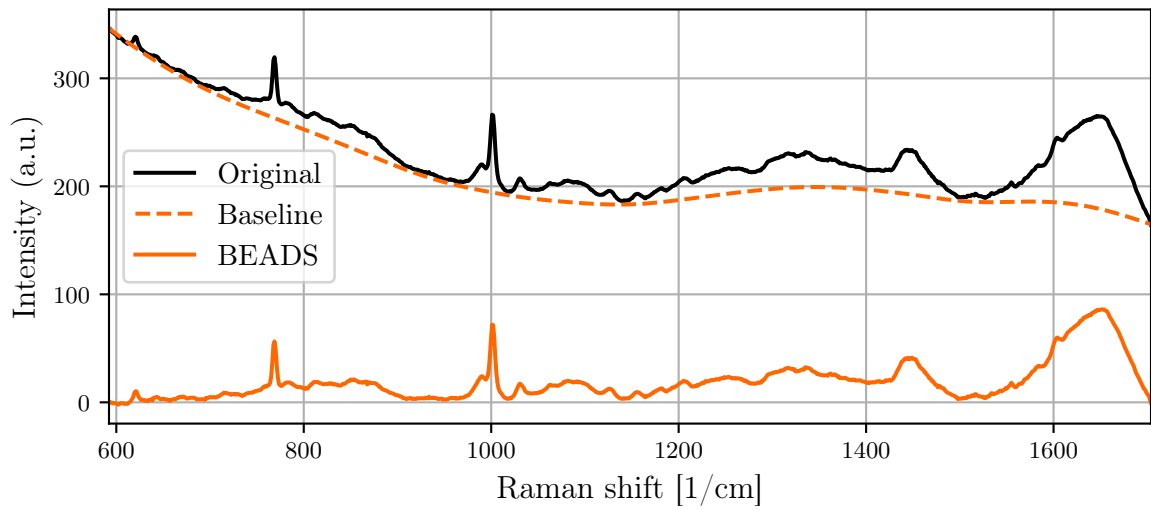


Figure 3.16: Mean spectrum of an hyperspectral cube before and after applying BEADS baseline correction.

- (c) *SNIP (Statistics-sensitive Non-linear Iterative Peak-clipping)* [19]: The SNIP method is a non-linear, iterative approach that progressively clips peaks from the spectrum to estimate the baseline. By iteratively clipping and averaging the data points, SNIP considers the local statistical properties of the spectrum.

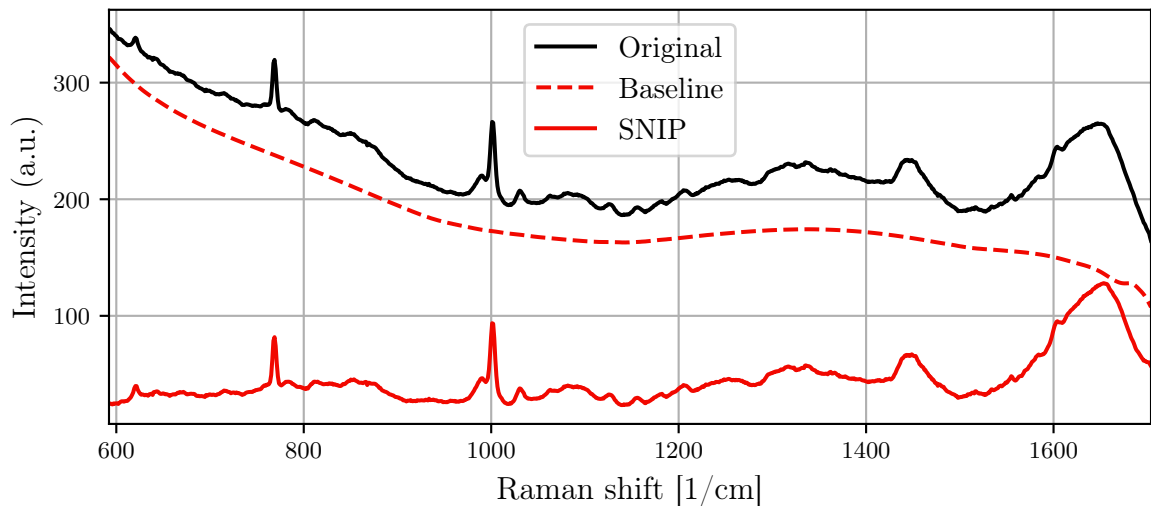


Figure 3.17: Mean spectrum of an hyperspectral cube before and after applying SNIP baseline correction.

For all baseline correction methods with the potential to produce negative spectra values, an option to enforce non-negativity is provided, which applies a rubberband correction

after the specified method.

Identify substrate

The *Identify substrate* function is designed to segment cells (the «foreground») from their substrate (the «substrate») using a clustering approach, such as k-means, mini-batch k-means or hierarchical clustering. These algorithms group the data into distinct clusters based on their similarity. The cluster with the lowest average signal is then considered as the substrate. By applying this segmentation method, users can remove the average background signal from the foreground or restrict certain processing steps to operate exclusively on foreground pixels. This enhances the clarity and accuracy of subsequent analyses, making it easier to focus on the relevant information within the hyperspectral data.

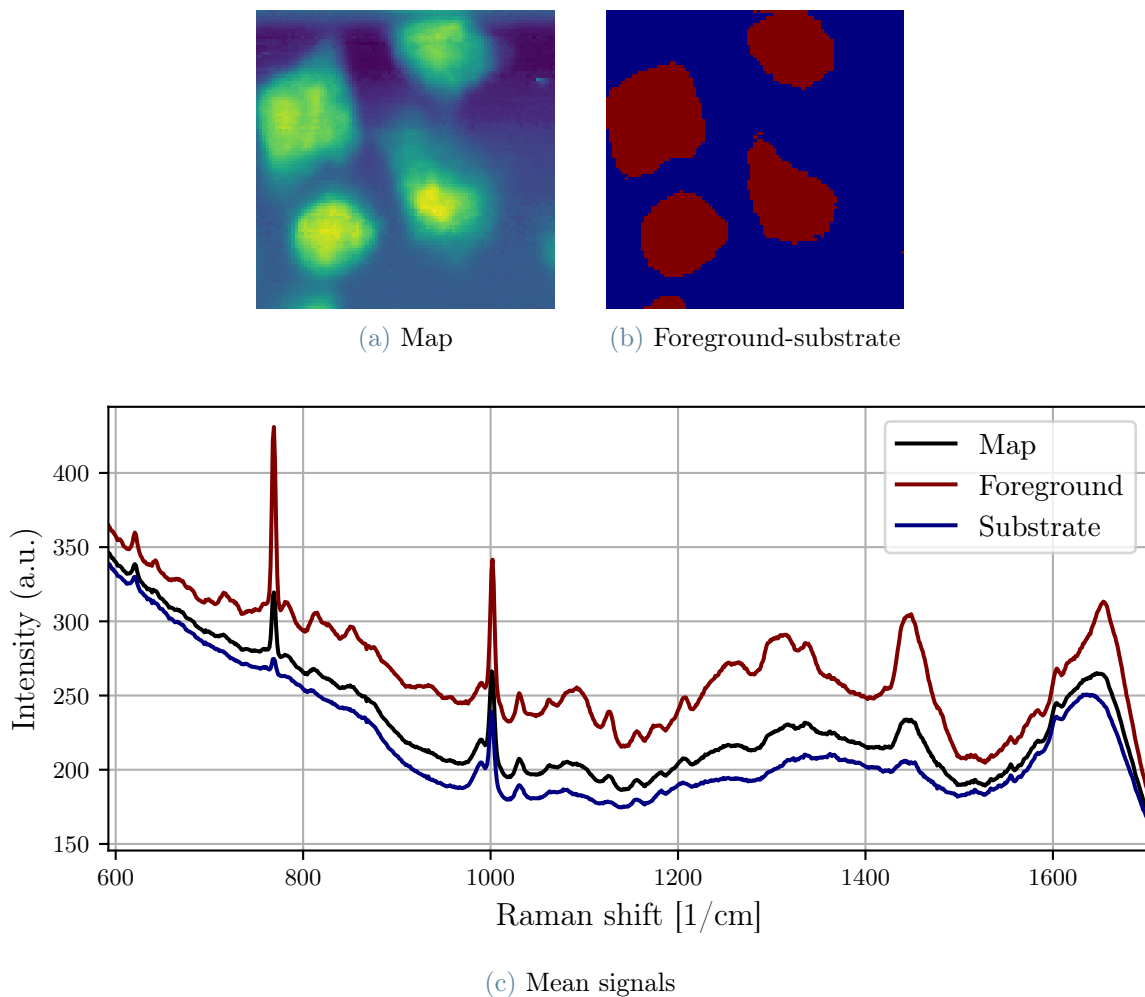


Figure 3.18: Identification of substrate pixels from the hyperspectral cube's signal.

3.2.3. Analysis

The analysis of hyperspectral data serves multiple purposes across various fields and contexts, as it enables the extraction of detailed information about the composition, structure and properties of materials and objects within the imaged scenes.

After performing the analysis, users may want to export their results. These exports can be of high quality and resolution, suitable for use in scientific publications, presentations, or other forms of communication.

In the following sections, a detailed explanation of the various functions for hyperspectral data analysis, as implemented within RamApp, will be provided.

Univariate analysis

Univariate analysis is a vital aspect of hyperspectral data analysis, focusing on the study and interpretation of individual spectral bands, or wavelengths, within the hyperspectral data cube.

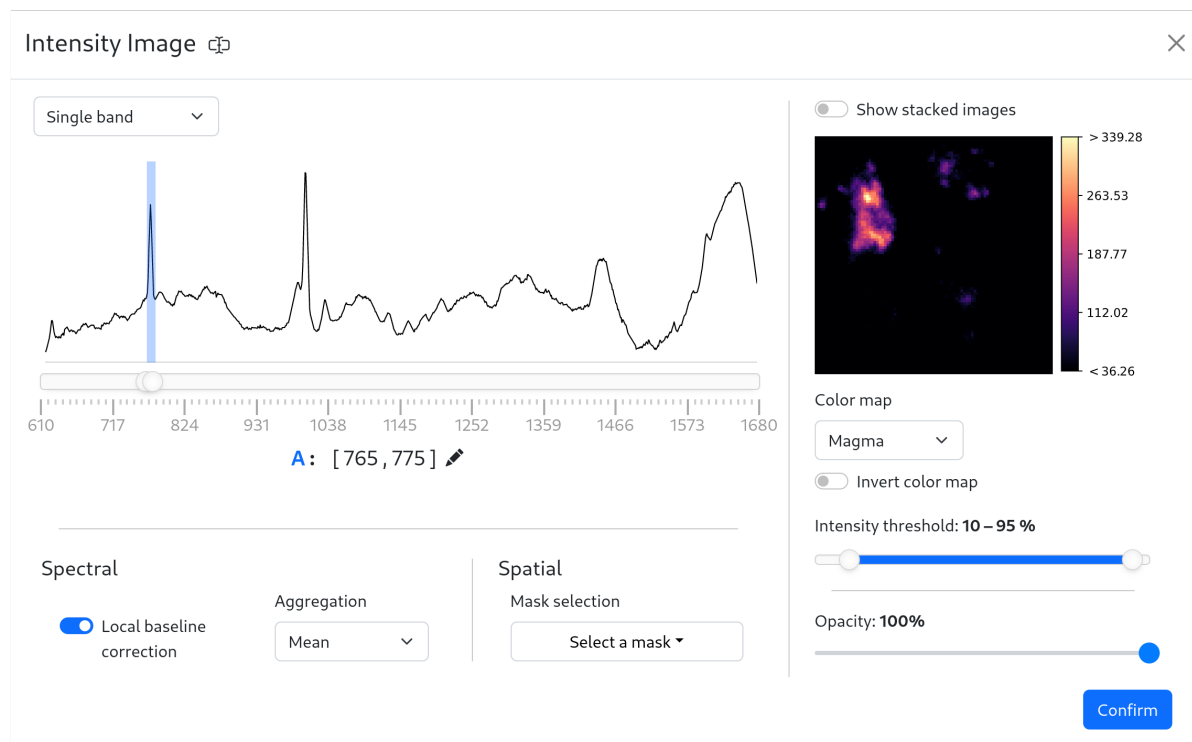


Figure 3.19: Menu appearance for customizing intensity images.

Unlike multivariate techniques that consider the relationships between multiple spectral bands, univariate analysis is concerned with extracting information from each band inde-

pendently. This approach can reveal specific characteristics of the materials or chemical species present in the scene based on their spectral signatures at individual wavelengths.

RamApp provides an intuitive interface to perform univariate analysis, starting by the creation of individual intensity images.

Univariate analysis can provide insights into the following:

- **Spectral features:** Examine individual spectral bands to identify specific features, such as peaks, valleys, or inflection points, that may be indicative of the presence of certain materials or chemical species. This information can be used to detect, identify and characterize the substances present in the scene based on their characteristic spectral peaks.
- **Band selection:** Select a subset of spectral bands that provide the most relevant information for a particular application or analysis objective. This may involve selecting bands that correspond to specific features, highlight particular materials or chemical species, or maximise the signal-to-noise ratio.
- **Visualization and interpretation:** Generate false-colour images or other visual representations using individual or combinations of spectral bands. This can help visualise and interpret the spatial distribution of materials or chemical species within the scene, as well as reveal patterns or structures that may not be apparent in multivariate analyses.

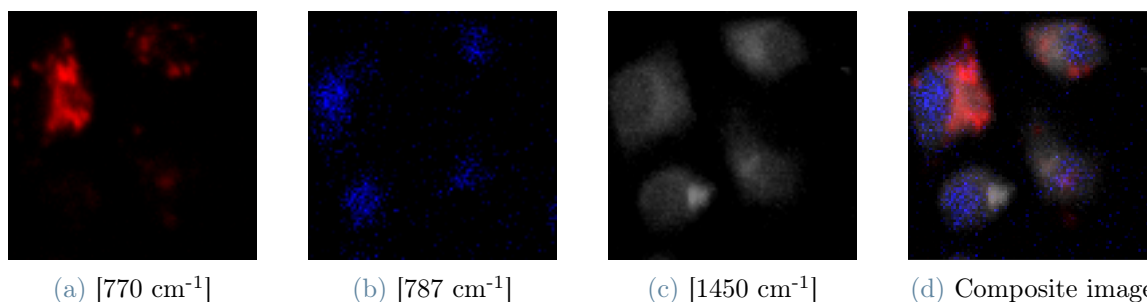


Figure 3.20: Example of univariate analysis performed on individual spectral bands. The intensity images are then blended to produce the final result.

Cluster analysis

This analysis function performs a cluster analysis on the hyperspectral map, allowing users to identify patterns and groupings within the data. Users, after specifying the number of clusters, can choose from three clustering algorithms:

1. **K-means**: A popular and efficient clustering algorithm that minimizes within-cluster sum of squares by iteratively assigning data points to their nearest cluster centroid.
2. **Mini-batch k-means** [20]: A variation of the k-means algorithm that uses a random subset (or "mini-batch") of data points in each iteration, reducing the computational complexity and processing time.
3. **Hierarchical agglomerative clustering (with Ward linkage)**: A bottom-up clustering method that successively merges clusters based on their similarity, with the Ward linkage criterion minimizing the total within-cluster variance.

The *Cluster Analysis* function and the subsequent analyses techniques presented in the following sections enable users to narrow down the analysis to specific regions of interest in both spatial and spectral domains. This targeted approach facilitates more focused and in-depth analyses of relevant areas within the hyperspectral data.

Additionally, users have the option of performing a reduction in principal components (PCs) prior to clustering. This step can help reduce dimensionality and computational complexity, while retaining the majority of the information in the data.

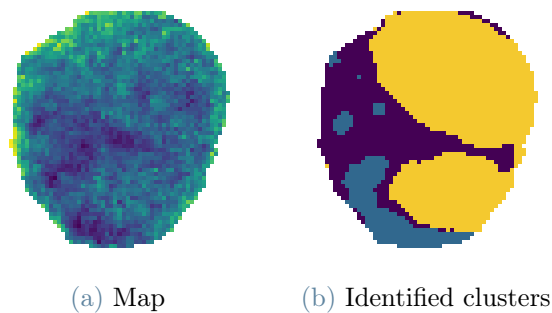


Figure 3.21: Cluster analysis performed on a leukemia cell.

The results of the cluster analysis will be displayed as cluster masks. A **mask**, in the context of the RamApp, refers to a spatial region of interest within the hyperspectral data. Specifically, it is a binary representation of the pixels in the data, with each pixel assigned a value of either 1 (included in the region of interest) or 0 (excluded from the region of interest).

When cluster analysis is performed, the resulting masks separate the pixels into distinct groups based on their cluster membership. By visually overlaying these masks on the hyperspectral map, users can better understand the spatial distribution and relationships between the identified clusters.

PCA

The Principal Component Analysis (*PCA*) function in RamApp is an essential analytical tool that performs dimensionality reduction on hyperspectral data, allowing users to gain insight into the underlying structure and variations in the data. PCA identifies the directions in the data with the most significant variance, transforming the original data into a new set of linearly uncorrelated components.

Algorithm 3.1 Principal Component Analysis (PCA)

Require: Data matrix $X \in \mathbb{R}^{n \times p}$ (n pixels, p spectral points)

- 1: Standardize X if necessary: $X_{\text{standardized}} \leftarrow \frac{X - \text{mean}(X)}{\text{std}(X)}$
- 2: Compute covariance matrix: $\text{Cov}(X) \leftarrow \frac{1}{n-1} X_{\text{standardized}}^T X_{\text{standardized}}$
- 3: Calculate eigenvectors (V) and eigenvalues (D) of $\text{Cov}(X)$: $\text{Cov}(X) \times V = V \times D$
- 4: Sort eigenvectors and eigenvalues in descending order based on eigenvalues
- 5: Compute PCA scores (T): $T \leftarrow X_{\text{standardized}} \times V$

Ensure: PCA Loadings (V) and PCA Scores (T)

Principal Component Analysis generates two main sets of results:

- **Scores:** The scores are the transformed data points in the new coordinate system defined by the principal components. Each score represents the projection of the original data point onto the respective principal component's axis. Scores can be visualised as intensity images, where each image corresponds to a principal component. These images reveal the spatial distribution of the variance explained by each principal component, highlighting patterns or regions of interest within the hyperspectral data.
- **Loadings:** The loadings represent the coefficients of the linear combination of the original variables used to create the principal components. Each loading vector corresponds to a principal component and indicates the contribution of each original variable to that component. In the context of hyperspectral data, loadings can be interpreted as the spectral signature associated with each principal component.

By examining PCA loadings and scores, researchers can identify patterns, relationships and sources of variance within the hyperspectral data. This information can be utilised for various applications, including dimensionality reduction, noise reduction, feature extraction and data visualisation. Furthermore, PCA results can be employed to guide subsequent analyses and enhance the understanding of the underlying processes responsible for the observed spectral features.

MCR

Multivariate Curve Resolution (*MCR*) is a powerful chemometric technique used for the analysis of hyperspectral data. It aims to decompose the data into a set of pure component spectra (also called endmember spectra) and their corresponding spatial distribution profiles (abundances or concentrations).

One of the most common implementations of MCR is the alternating least squares (ALS) approach, called MCR-ALS [21]. Implementing the algorithm was made possible through the *pyMCR* library [22].

Algorithm 3.2 Multivariate Curve Resolution - Alternating Least Squares (MCR-ALS)

Require: Data matrix $X \in \mathbb{R}^{n \times p}$ (n pixels, p spectral points), number of components (r)

- 1: Initialize pure component spectra matrix $C \in \mathbb{R}^{p \times r}$ or spatial profiles matrix $ST \in \mathbb{R}^{n \times r}$
- 2: **repeat**
- 3: Update spatial profiles matrix: $ST \leftarrow \operatorname{argmin}_{ST} \|X - CST\|^2$
- 4: Apply constraints on ST (e.g., non-negativity, unimodality, normalization)
- 5: Update pure component spectra matrix: $C \leftarrow \operatorname{argmin}_C \|X - CST\|^2$
- 6: Apply constraints on C (e.g., non-negativity, unimodality, normalization)
- 7: **until** convergence or stopping criterion is met

Ensure: Pure component spectra matrix (C) and spatial profiles matrix (ST)

The primary outputs of MCR-ALS are the estimated pure component spectra (C) and their corresponding spatial profiles (ST). These results can be utilized for various applications, including the identification and quantification of chemical species, the extraction of features related to the underlying physical or chemical processes and the visualization of spatial distribution patterns.

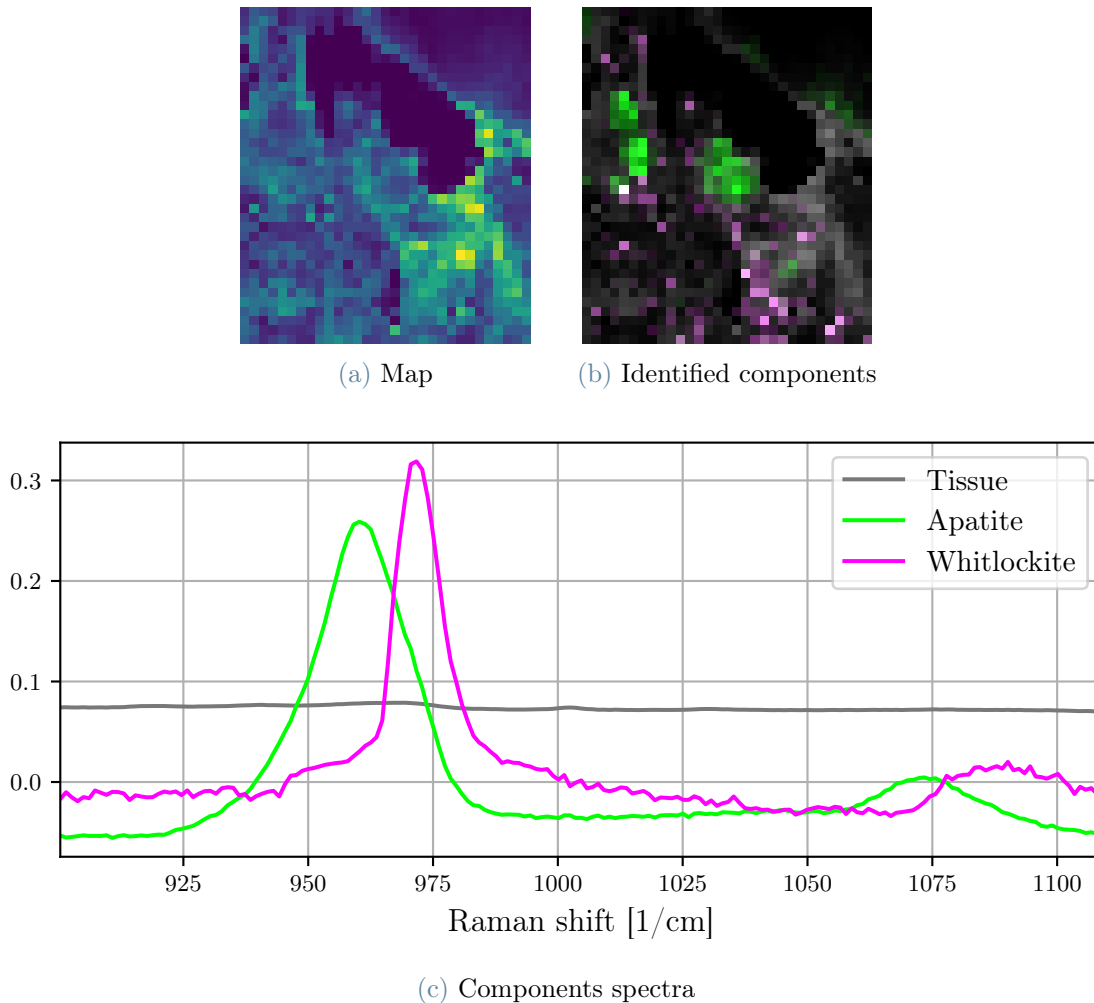


Figure 3.22: Example of MCR analysis on a breast tissue.

N-FINDR

The *N-FINDR* [23] algorithm is a widely used method for endmember extraction in hyperspectral data analysis. The goal of endmember extraction is to identify the purest spectral signatures (endmembers) present in the data, which represent the underlying materials or chemical species. These endmembers can be used for various purposes, such as unmixing analysis, classification and anomaly detection.

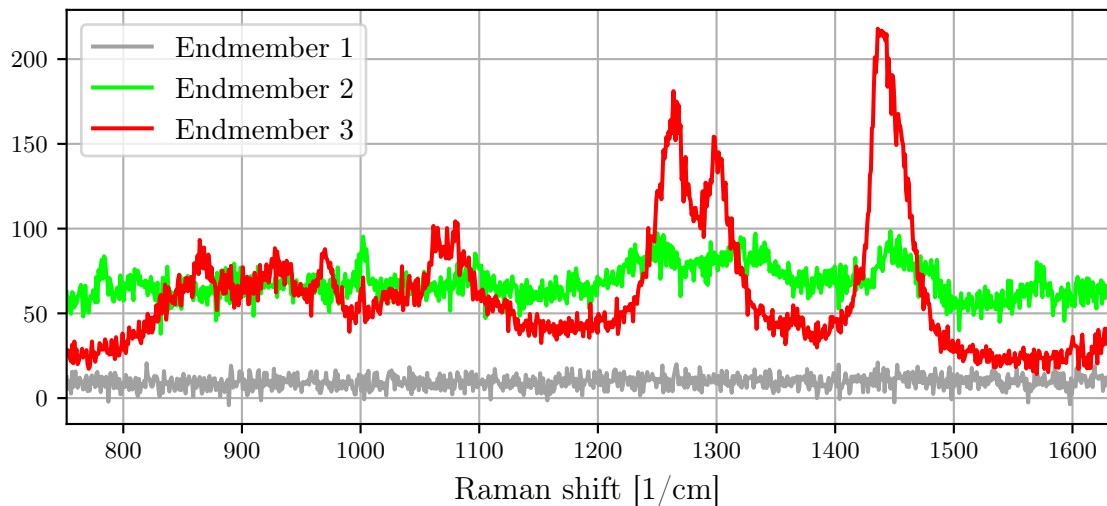
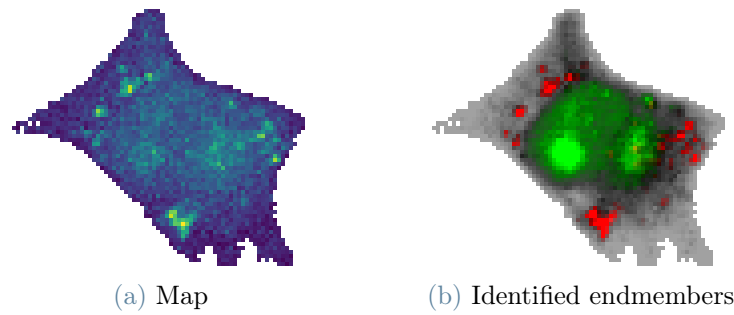
Implemented using the *pyspc-unmix* library [24], N-FINDR is a geometric approach that aims to identify the endmembers by searching for the simplex with the maximum volume in high-dimensional spectral space. The vertices of this simplex are assumed to correspond to the endmember spectra.

Algorithm 3.3 N-FINDR

Require: Hyperspectral data X , number of endmembers p

Ensure: Endmember matrix E

- 1: Initialize E by randomly selecting p spectra from X
 - 2: **while** not converged **do**
 - 3: Compute the projection matrix $P = E(E^T E)^{-1} E^T$
 - 4: Compute the residual matrix $R = X - PX$
 - 5: Compute the norms of the columns of R , r_1, r_2, \dots, r_n
 - 6: Select the column of X with the maximum norm as the next endmember
 - 7: Orthogonalize the new endmember w.r.t. E using Gram-Schmidt process
 - 8: Replace the endmember with the lowest loading in P with the new endmember
 - 9: **end while**
 - 10: **return** Endmember matrix E
-



(c) Endmembers spectra

Figure 3.23: Example of N-FINDR analysis on a neuronal cell.

The primary result provided by the N-FINDR algorithm is a set of endmember spectra, which represent the purest spectral signatures in the hyperspectral data. These end-

members correspond to the vertices of the simplex with the maximum volume in the high-dimensional spectral space. In the context of hyperspectral data, each endmember spectrum is associated with a distinct material or chemical species present in the scene.

3.3. Architecture

The architecture of the web application is designed to provide a user-friendly and efficient platform. RamApp is built upon a modular structure consisting of three distinct components that work together to deliver a comprehensive and interactive experience to users.

These components are:

- the **frontend**, the module responsible for the graphical interface of the application;
- the **backend**, the module that serves as the backbone of the application, handling user requests and managing their interaction with the computation module;
- a **computational module**, called *ramappy*, that performs all the operations on the hyperspectral data cube.

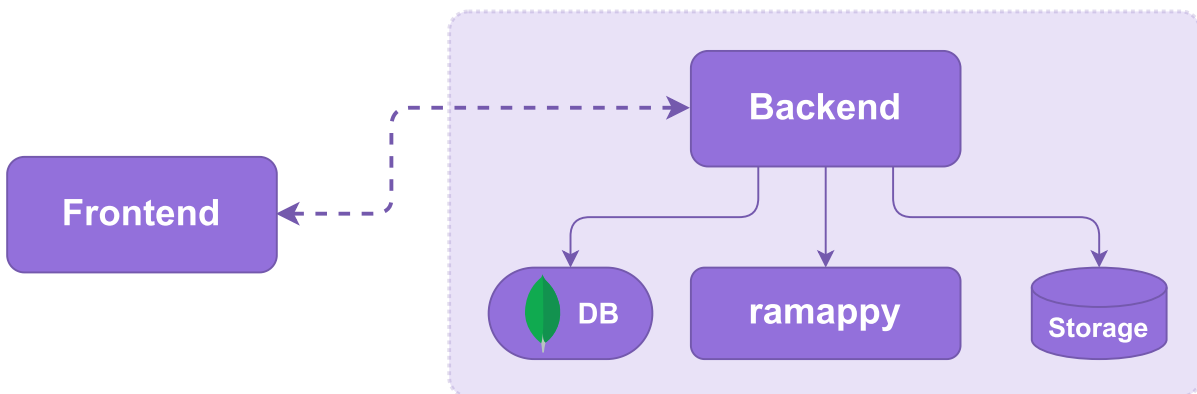


Figure 3.24: Diagram of the architecture of RamApp.

Each module will be detailed in the following sections.

3.3.1. Frontend

The frontend of RamApp is designed to provide users with an intuitive interface for interacting with the application, ensuring an enjoyable and efficient experience. Built using the popular web development library **React** and **TypeScript**, the frontend allows users to easily navigate through the application and input the necessary data for processing

and analysis.

To enhance the application's visual aesthetics and functionality, the frontend employs the react-bootstrap library. This library provides a collection of reusable components that adhere to the **Bootstrap** framework, ensuring a clean and modern design. By leveraging react-bootstrap, the application ensures a consistent and professional appearance across various platforms, while also providing users with familiar and easy-to-use interface elements.

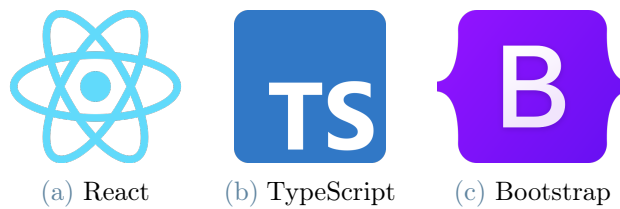


Figure 3.25: Frontend technology stack.

A key aspect of the frontend design is its ability to communicate effectively with the backend, exchanging messages and processing requests in real time. To achieve this, the application utilises the axios library, which enables the frontend to send asynchronous HTTP requests to the backend, ensuring a smooth and uninterrupted user experience. This approach allows efficient data exchange between the frontend and backend, facilitating quick processing and presentation of results.

3.3.2. Backend

The backend of RamApp serves as the central hub for managing user requests and orchestrating the interactions between the frontend and the computation module. Developed using **Python**, the backend is designed to be both efficient and robust, ensuring a seamless user experience as the application processes hyperspectral imaging data.

FastAPI, a modern and high-performance web framework for Python, is employed as the foundation of the backend. With its simplicity and ease of use, FastAPI enables the development of a highly scalable and reliable backend architecture. This framework not only ensures that the application can handle a large number of user requests simultaneously, but also allows for rapid development and deployment of new features and improvements.

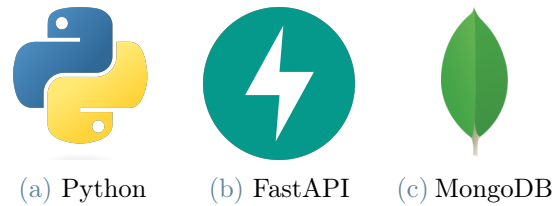


Figure 3.26: Backend technology stack.

User data management, including login and registration, is handled by the **FastAPI-users** library. This library provides an extensive set of tools to manage user authentication, authorisation and data storage.

In addition, FastAPI-users also supports various login and registration options. RamApp offers users the flexibility to register and log in using their email and password, or opt for a more convenient approach by leveraging Google as a social login option. The integration of Google's social login simplifies the registration and authentication process by allowing users to quickly access the platform using their existing Google account credentials.

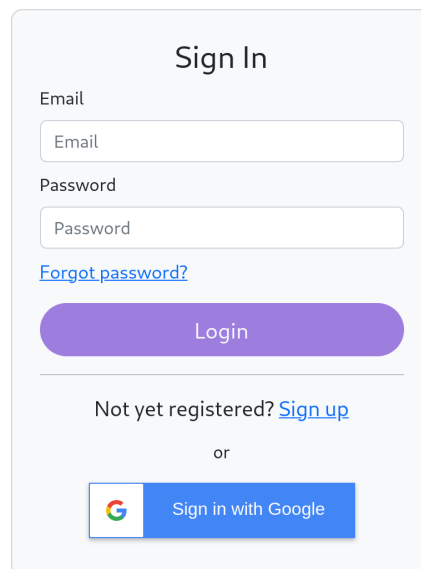


Figure 3.27: RamApp login interface.

MongoDB, a popular NoSQL database, is used for user data and project metadata. Its flexible schema design and high scalability make it a suitable choice for handling user data.

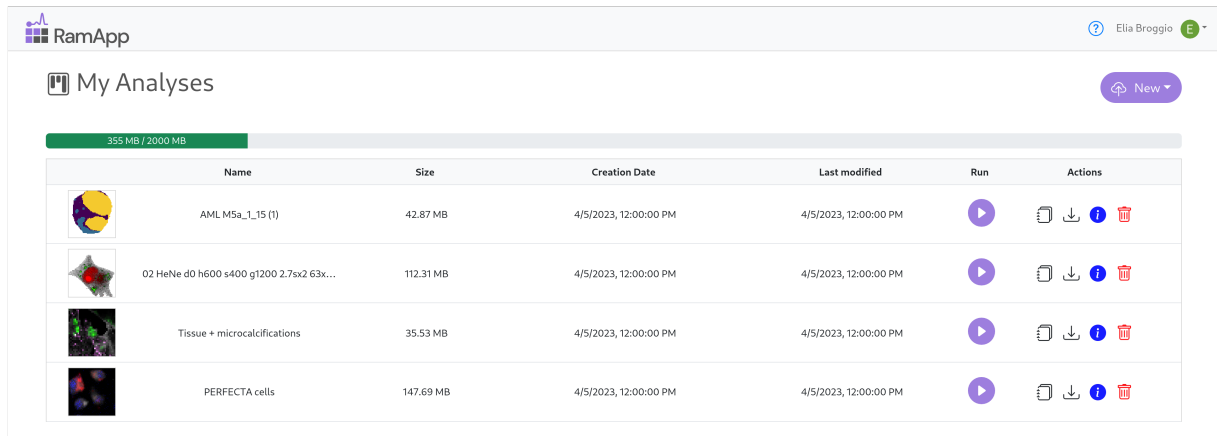


Figure 3.28: Interface for RamApp user projects.

3.3.3. Computational module

The computational module, *ramappy*, serves as the core engine responsible for processing and analysing data within RamApp. Written in **Python**, *ramappy* is structured with a collection of classes designed to represent and manipulate hyperspectral data.

Several renowned Python libraries play a vital role in enabling *ramappy* to deliver powerful and efficient functionality to users. Among these libraries are **NumPy** [25], **Scikit-learn** [26], **SciPy** [27][28] and **Pandas** [29] to perform complex mathematical operations, data manipulation, pre-processing and analysis on hyperspectral data arrays. **Numba** [30] is particularly valuable for optimising performance, as it employs just-in-time (JIT) compilation to accelerate the execution of computationally intensive tasks, significantly reducing processing time. **Python Imaging Library (PIL)** [31] and **Matplotlib** [32] are used to process and manipulate images generated from the hyperspectral data.

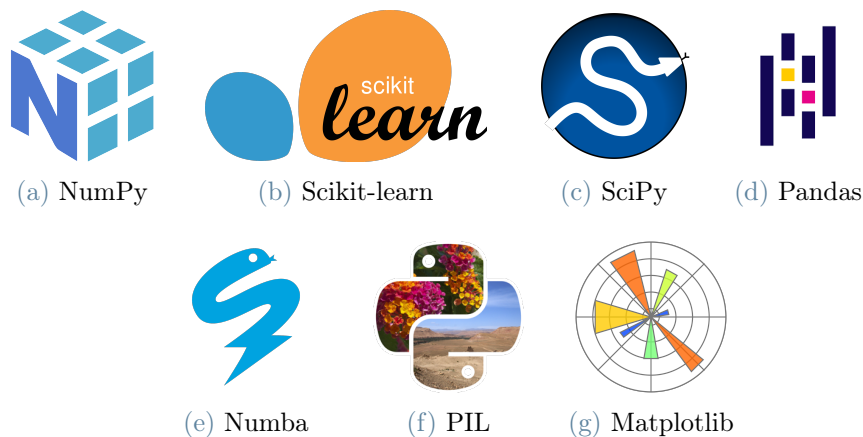


Figure 3.29: Ramappy technology stack.

4 | Use case example

In this chapter, a use case example is presented, showcasing the capabilities and potential of RamApp in processing and analyzing HSI data.

The selected imaging spectroscopy data for this example is a Raman image of in vitro labeled murine microglial cells with nanoformulations of PERFECTA, a superfluorinated molecular probe for highly sensitive Functional Magnetic Resonance Imaging (F-MRI) [2]. These data were acquired with spatial size of 93x93 pixels and spectral resolution of 1015 spectral points.

4.1. Data import

The initial step involves properly uploading the data to RamApp. Depending on the file format, the application may necessitate additional information prior to processing the data (name of the variables, width and height of the image, scan pattern, etc.).

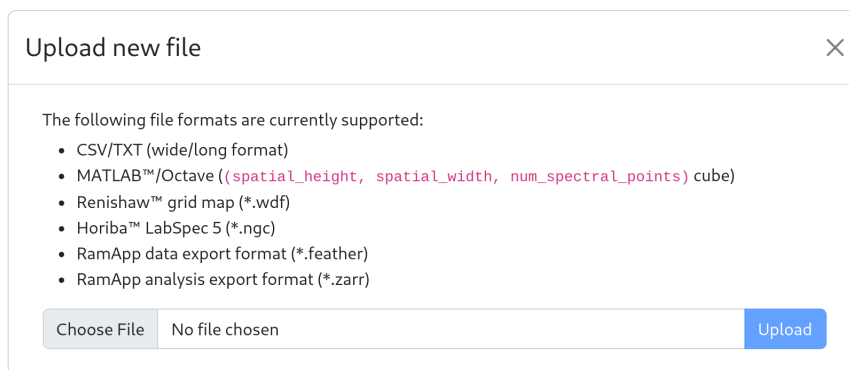


Figure 4.1: File uploader interface.

In this case, the hyperspectral data were acquired using a commercial setup (Renishaw), ensuring that all the parameters required to accurately read the cube were included within the proprietary file format (*.wdf). Once the data file has been successfully uploaded, it will appear in the user's personal workspace, ready to be opened and analysed.

4.2. Data exploration

After opening a new analysis, the main focus is on exploring the data. The first intensity image displayed represents the mean intensity for each pixel. This initial visualisation provides a useful starting point for users to get a general overview of the hyperspectral data. This image offers a basic understanding of the spatial distribution of the signal across the map, allowing users to identify any patterns or anomalies before diving deeper into the analysis.

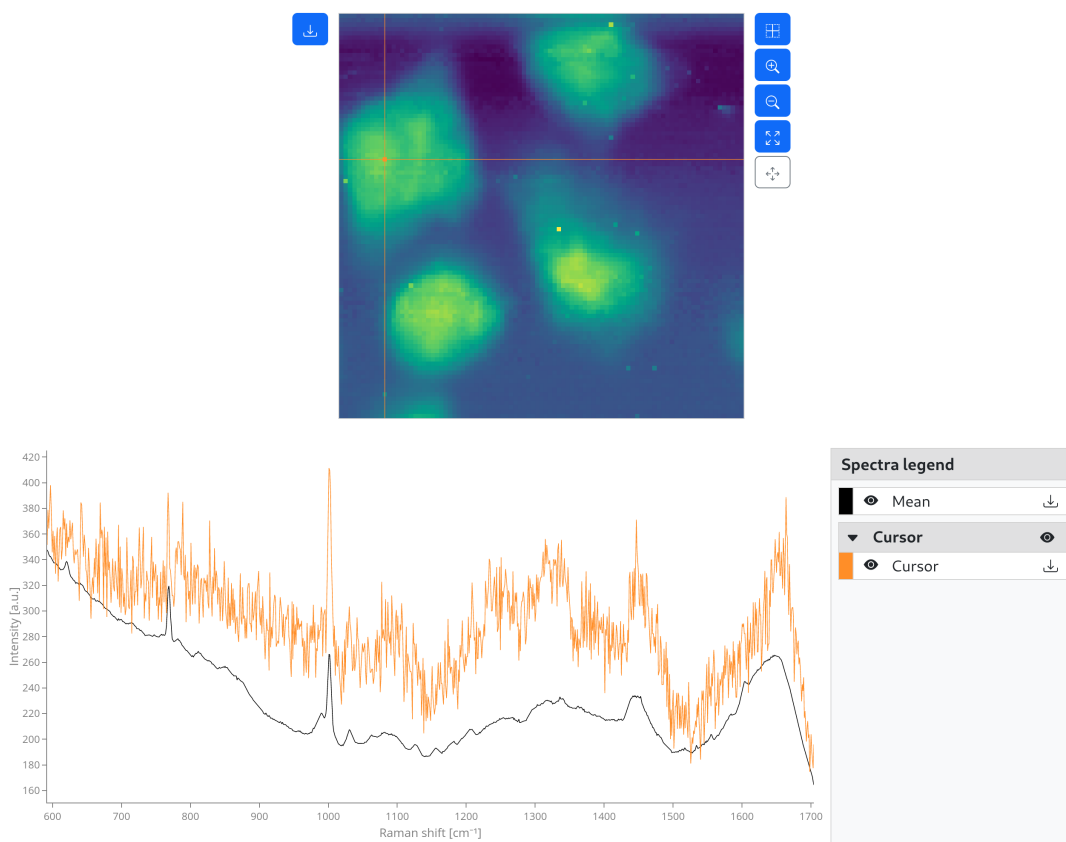


Figure 4.2: Map visualiser and spectra plot of RamApp. In orange, the spectrum relative to the orange pixel.

RamApp is designed to facilitate the exploration of the data cube, allowing users to quickly gain insights into their hyperspectral data. One way this is achieved is through the map visualiser: by simply clicking on a pixel within the displayed image, users can instantly view the spectrum corresponding to that specific pixel. This interactive feature provides a convenient way to examine the spectral information across the data, enabling a better understanding of the dataset as a whole.

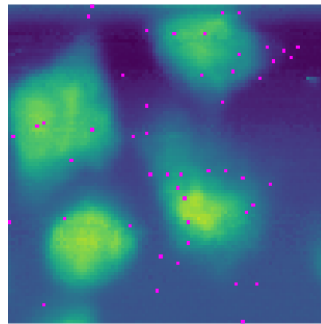
4.3. Preprocessing

Although users can opt for their preferred workflow, this example follows a Raman spectroscopy protocol that outlines a complete preprocessing pipeline [33].

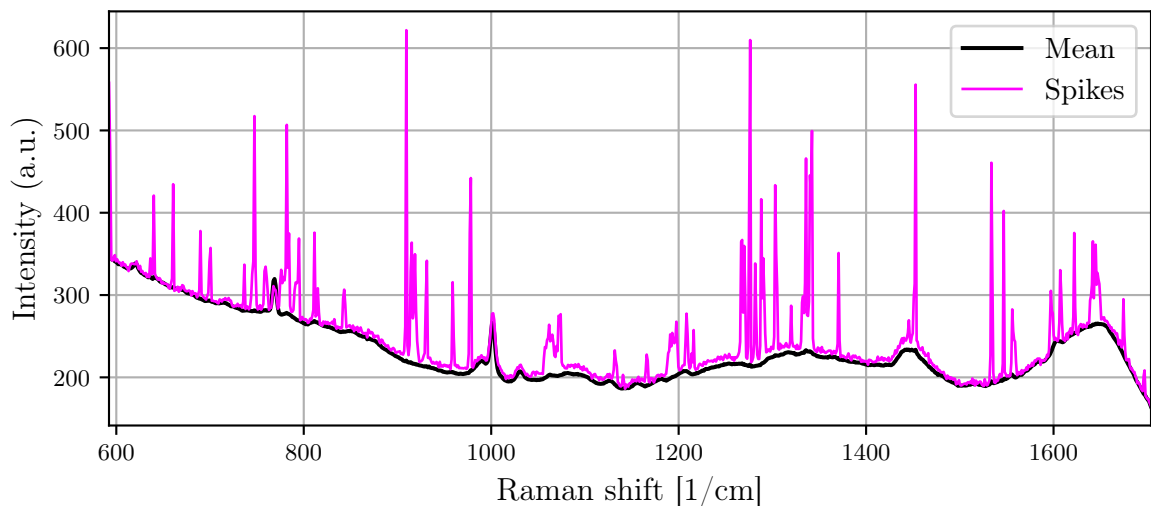
4.3.1. Spikes removal

After exploring the data, it becomes evident that some pixels of the hyperspectral dataset contain spikes in their signal, which can hinder the analysis process. To address this issue and improve data quality, the *Despike* preprocessing function should be used.

In this case, bad pixels were identified using the Z-score method with a threshold of 13, since selecting lower values led to incorrect classification of some pixels as spikes. The signals were subsequently corrected using a linear interpolation function.



(a) Identified spikes



(b) Mean signal of *Spikes* pixels

Figure 4.3: In magenta, the pixels identified as spikes and their mean signal.

4.3.2. Fluorescence baseline removal

The following step in the preprocessing workflow involves removing the fluorescence baseline from the signal by using the *Correct baseline* function. This essential step helps to isolate the Raman signal from the background fluorescence, ensuring that the subsequent analysis focuses on the relevant spectral features.

In this case, the baseline was identified using the Goldindex method with a polynomial order of 5 and subsequently subtracted from the signal.

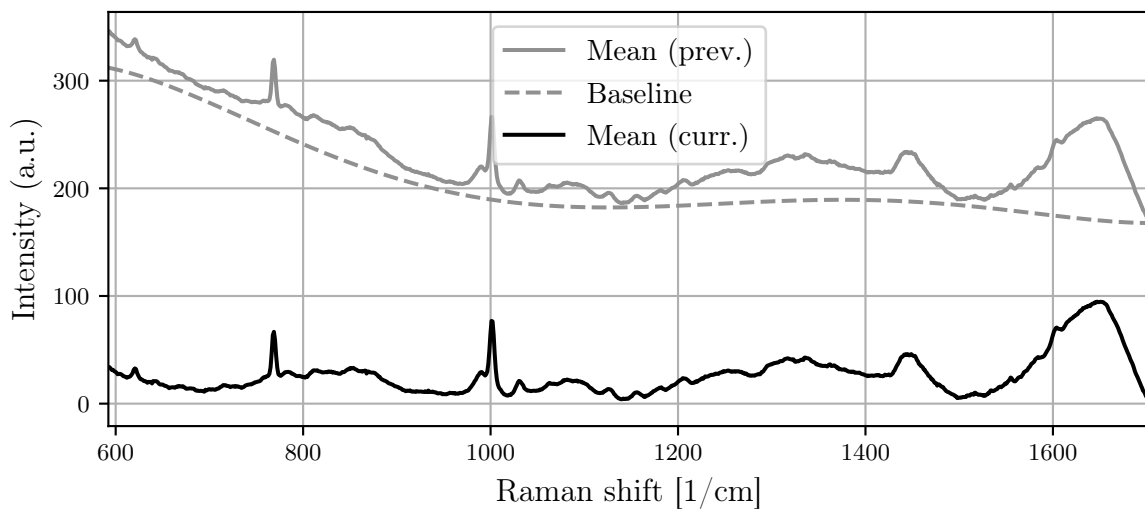
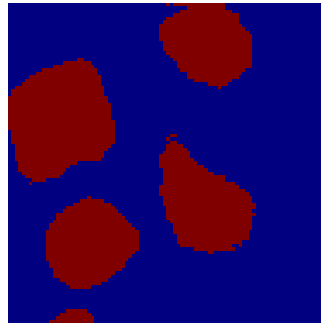


Figure 4.4: The mean signal of the hyperspectral cube before and after removing the fluorescence baseline.

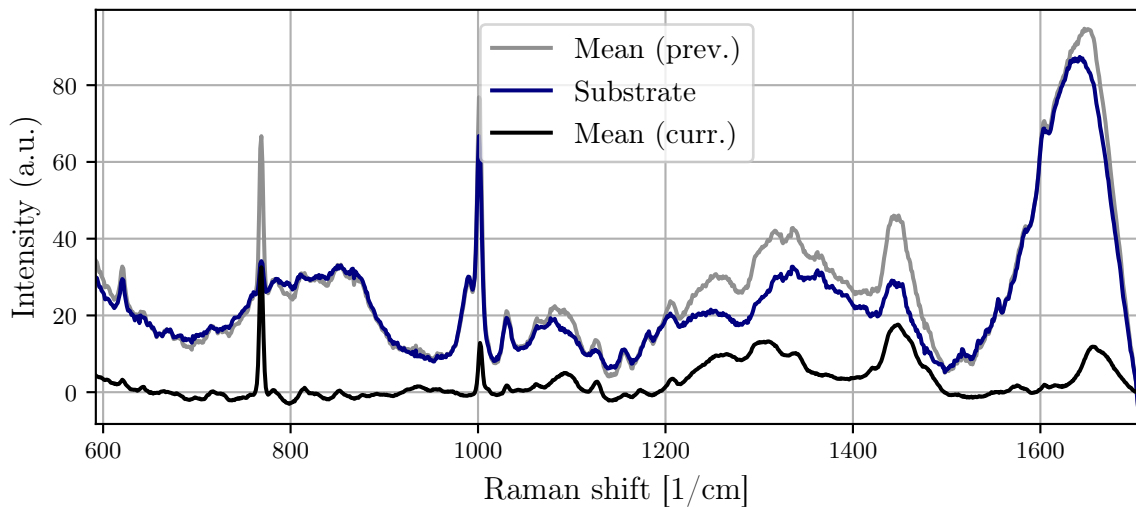
4.3.3. Substrate background removal

Following the removal of the fluorescence baseline, it is necessary to address the presence of substrate-related Raman signals in the measured spectra. To accomplish this, the *Identify substrate* function can be employed. This step helps to isolate and remove the contribution of the substrate from the acquired Raman spectra, ensuring that the remaining signals correspond solely to the sample of interest.

The substrate-related cluster was identified using a k-means approach and subsequently, the mean signal of all pixels in that cluster, was subtracted from the overall mean signal.



(a) Identified substrate



(b) The mean signal of the hyperspectral cube before and after removing the substrate.

Figure 4.5: In dark blue, the identified substrate and its corresponding mean signal.

4.3.4. Smoothing

Smoothing, which is optional in Raman spectra analysis, can be performed through spectral and/or spatial filtering. In this case, spatial smoothing will be performed using the *Smooth map* function, employing a 3x3-pixel square median filter.

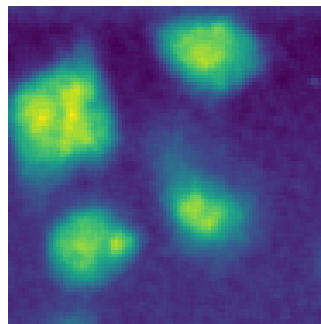


Figure 4.6: The resulting intensity image after applying a spatial smoothing.

4.3.5. Truncation

Spectral truncation helps eliminate wavenumber regions lacking significant Raman signals (silent regions) or those with strong contributions from the substrate, water, or artefacts.

In the example case, the Raman signal displays minor artefacts at the beginning and the end of the spectrum, which were generated during the fluorescence baseline removal phase. With the *Spectral truncation* function, the spectrum was refined from the 592-1704 cm^{-1} range to the 610-1695 cm^{-1} region.

4.3.6. Normalisation

Following spectral truncation, normalisation is applied as the final preprocessing step, with the aim of mitigating the impact of fluctuations in excitation intensity or focusing changes. In this specific case, the L2 norm was used as the normalisation method.

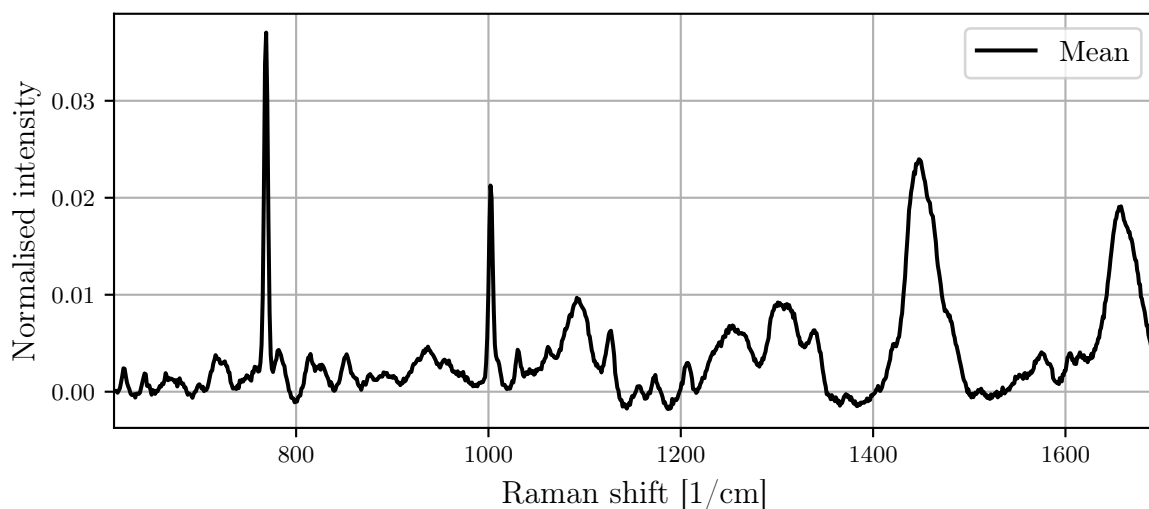


Figure 4.7: Mean signal of the hyperspectral cube after applying L2 norm.

4.4. Analysis

The primary objective of the analysis is to determine the localisation of PERFECTA within cells to gain valuable insight into its behaviour and interactions within the cellular environment.

4.4.1. Univariate analysis

Given that PERFECTA exhibits a well-defined spectral signature, with a particularly intense peak centred around 770 [cm^{-1}], it becomes feasible to employ univariate analysis techniques for its detection.

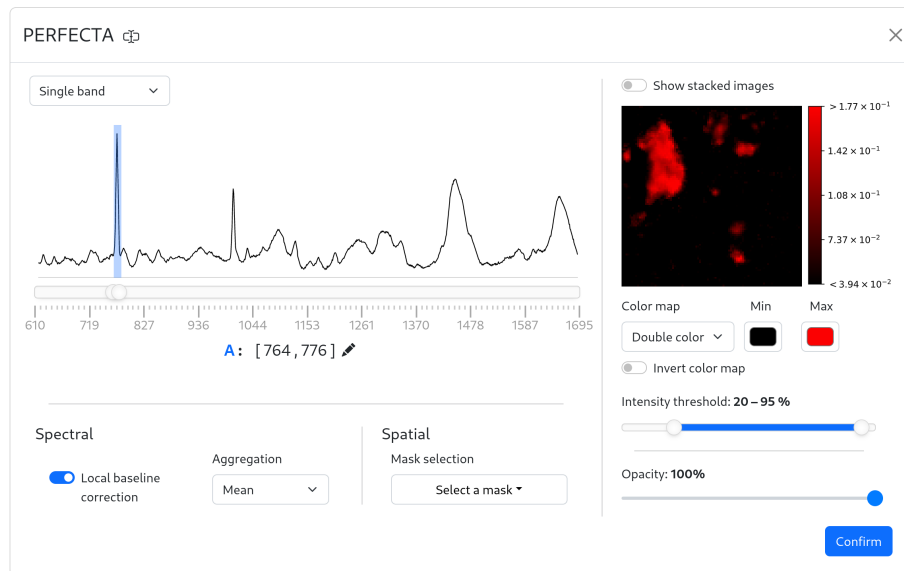


Figure 4.8: Intensity image creation to highlight the localisation of PERFECTA.

Another substance worth locating is DNA. With its low intensity peak near 787 [cm^{-1}], it becomes advantageous to limit the spatial domain of the intensity image to the previously identified *Foreground* mask, ensuring a more focused and accurate analysis.

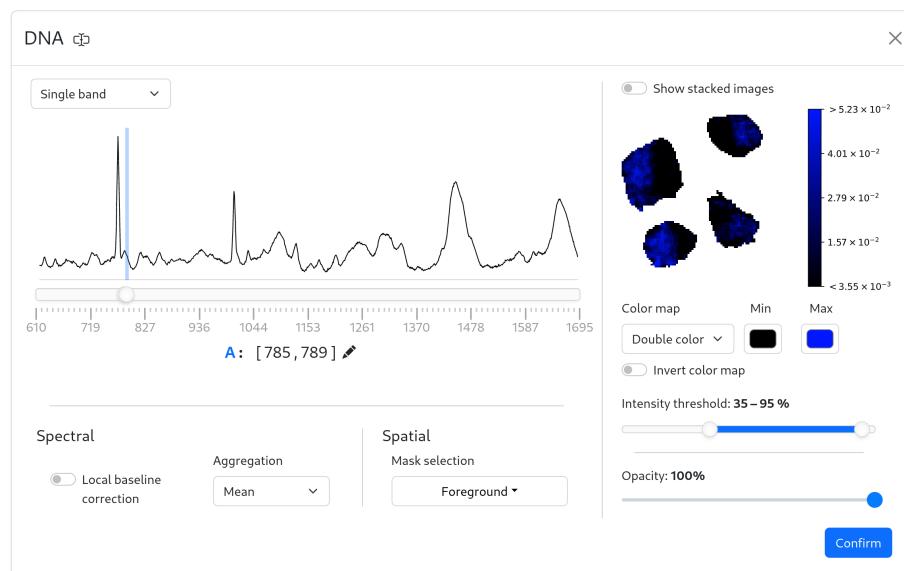


Figure 4.9: Intensity image creation to highlight the localisation of DNA.

Lastly, it is interesting to highlight the cellular organic matrix. This component can be easily identified due to its spectral signature, which features a broad peak around 1450 $[\text{cm}^{-1}]$.

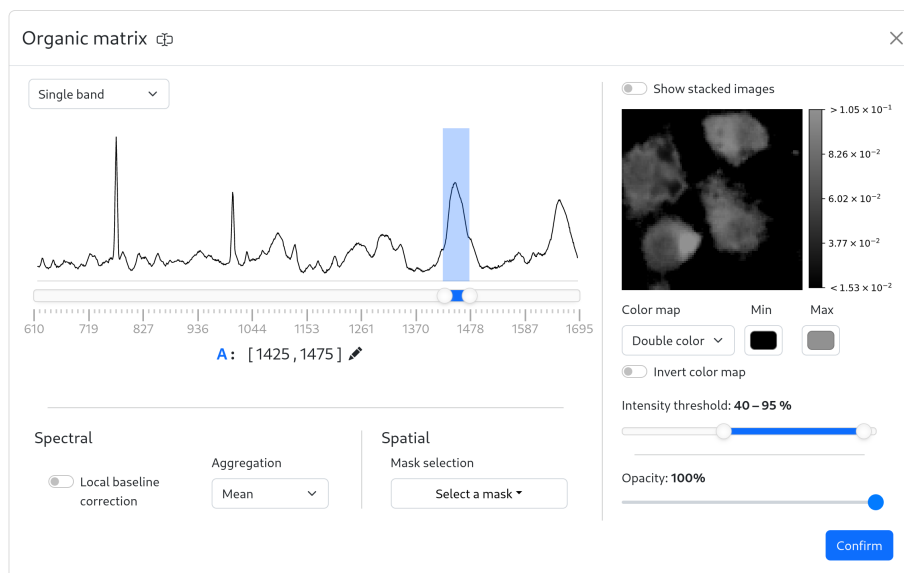


Figure 4.10: Intensity image creation to highlight the localisation of the organic matrix.

In conclusion, the univariate analysis and the resulting composite image provide valuable insights into the distribution and localisation of PERFECTA, DNA and the organic matrix within the cells, enhancing the understanding of their spatial relationships and interactions.

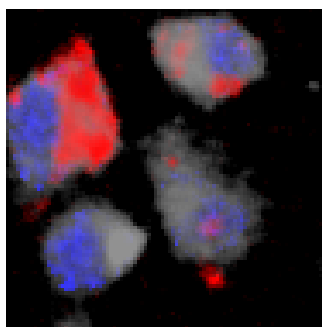


Figure 4.11: Composite image showing the combined intensity maps of PERFECTA, DNA and organic matrix distributions.

5 | Conclusions and future developments

In conclusion, this thesis has successfully documented the development and implementation of RamApp, a versatile and adaptable tool for processing and analysing hyperspectral imaging data.

Since its closed-beta release in September 2022, followed by the official launch in March 2023, RamApp has demonstrated its value to researchers across various fields and countries, offering the flexibility and adaptability often lacking in commercial software solutions. The user-friendly graphical interface, complemented by a comprehensive range of functionalities, facilitates efficient data processing and analysis, empowering researchers to extract significant insights from their hyperspectral data.

Moreover, RamApp has garnered numerous positive testimonials from users, who report that it has substantially accelerated their analysis tasks, streamlined their workflows and allowed them to focus on deriving meaningful insights from their data more quickly and efficiently.

To promote the application, a poster has been presented at some conferences, including the 12th International Conference on Clinical Spectroscopy (SPEC) held in Dublin in June 2022. RamApp is also scheduled to be presented at future events, maintaining engagement with the scientific community and highlighting its capabilities.

Moving forward, there are several areas for future development of RamApp that have great potential to improve its capabilities and user experience.

First, a key objective involves expanding the range of supported import formats, allowing users to easily incorporate data from additional proprietary formats. By increasing RamApp's compatibility with a variety of file types, the platform can better serve a diverse user base and meet the requirements of researchers working with different instruments and software.

Moreover, efforts will be made to simplify the upload process for custom-defined files.

This enhancement will further simplify data import, enabling researchers to seamlessly integrate RamApp into their existing workflows, irrespective of their particular file formats.

Another promising avenue for future development of RamApp involves integrating a task developed within the CRIMSON project, which centres on implementing a neural network to eliminate the non-resonant background from Broadband Coherent Anti-Stokes Raman Scattering (B-CARS) spectra. The non-resonant background refers to the part of the B-CARS signal that does not result from molecular vibrations but arises due to the interaction of the probing light with the sample's electronic environment.

Although various numerical techniques exist for eliminating this undesired component and isolating the resonant vibrational signal of interest, they all necessitate user intervention and are highly dependent on the spectral shape of the non-resonant background. This background, in turn, must be independently measured, adding complexity to the process.

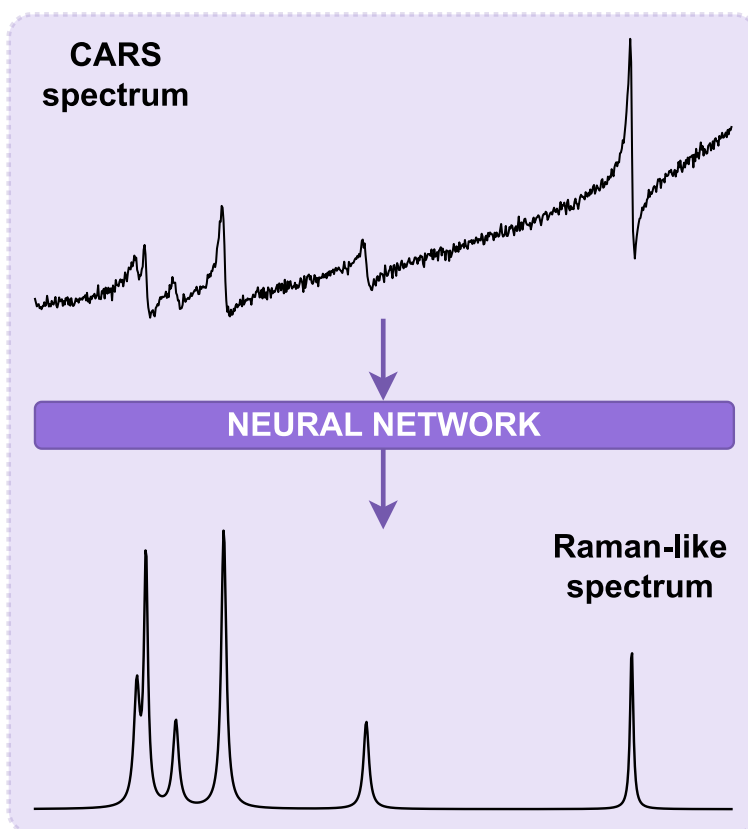


Figure 5.1: CARS to Raman-like spectrum translation scheme.

By incorporating this neural network into RamApp, researchers will be able to more effectively remove the non-resonant background, enhancing the clarity and interpretability of their B-CARS data.

Bibliography

- [1] Moxfyre Pavlina2.0. *Molecular energy levels and Raman effect*. Sept. 18, 2009. URL: https://commons.wikimedia.org/wiki/File:Raman_energy_levels.svg.
- [2] Cristina Chirizzi et al. “A Bioorthogonal Probe for Multiscale Imaging by 19F-MRI and Raman Microscopy: From Whole Body to Single Cells”. In: *Journal of the American Chemical Society* 143.31 (Aug. 11, 2021). Publisher: American Chemical Society, pp. 12253–12260. ISSN: 0002-7863. DOI: 10.1021/jacs.1c05250. URL: <https://doi.org/10.1021/jacs.1c05250>.
- [3] Renzo Vanna et al. “Raman Spectroscopy Reveals That Biochemical Composition of Breast Microcalcifications Correlates with Histopathologic Features”. In: *Cancer Research* 80.8 (Apr. 15, 2020), pp. 1762–1772. ISSN: 0008-5472. DOI: 10.1158/0008-5472.CAN-19-3204. URL: <https://doi.org/10.1158/0008-5472.CAN-19-3204>.
- [4] Renishaw plc. *Renishaw: Raman software*. Renishaw. URL: <http://www.renishaw.com/en/raman-software--9450>.
- [5] Horiba, Ltd. *LabSpec 6 Spectroscopy Suite Software*. URL: <https://www.horiba.com/int/scientific/products/detail/action/show/Product/labspec-6-spectroscopy-suite-software-1843/>.
- [6] Claudia Beleites et al. *hyperSpec: Work with Hyperspectral Data, i.e. Spectra + Meta Information (Spatial, Time, Concentration, ...)* Version 0.100.0. Sept. 13, 2021. URL: <https://CRAN.R-project.org/package=hyperSpec>.
- [7] Robert W. Schmidt, Sander Woutersen, and Freek Ariese. “RamanLIGHT - A graphical user-friendly tool for pre-processing and unmixing hyperspectral Raman spectroscopy images”. In: *Journal of Optics* (Apr. 2022). Publisher: IOP Publishing. DOI: 10.1088/2040-8986/ac6883. URL: <https://iopscience.iop.org/article/10.1088/2040-8986/ac6883>.
- [8] *Zarr storage specification v 2*. URL: <https://zarr.readthedocs.io/en/stable/spec/v2.html>.
- [9] Abraham Savitzky and M. J. E. Golay. “Smoothing and Differentiation of Data by Simplified Least Squares Procedures.” In: *Analytical Chemistry* 36.8 (July 1,

- 1964). Publisher: American Chemical Society, pp. 1627–1639. ISSN: 0003-2700. DOI: 10.1021/ac60214a047. URL: <https://doi.org/10.1021/ac60214a047>.
- [10] Paul H. C. Eilers. “A Perfect Smoother”. In: *Analytical Chemistry* 75.14 (July 1, 2003). Publisher: American Chemical Society, pp. 3631–3636. ISSN: 0003-2700. DOI: 10.1021/ac034173t. URL: <https://doi.org/10.1021/ac034173t>.
- [11] Donald Erb. *pybaselines: A Python library of algorithms for the baseline correction of experimental data*. Version 1.0.0. Oct. 27, 2022. DOI: 10.5281/ZENODO.5608581. URL: <https://zenodo.org/record/5608581>.
- [12] Jianhua Zhao et al. “Automated Autofluorescence Background Subtraction Algorithm for Biomedical Raman Spectroscopy”. In: *Appl. Spectrosc.* 61.11 (Nov. 2007). Publisher: Optica Publishing Group, pp. 1225–1232. URL: <https://opg.optica.org/as/abstract.cfm?URI=as-61-11-1225>.
- [13] Juntao Liu et al. “Goldindec: A Novel Algorithm for Raman Spectrum Baseline Correction”. In: *Applied spectroscopy* 69.7 (July 2015), pp. 834–842. ISSN: 0003-7028. DOI: 10.1366/14-07798. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5030208/>.
- [14] Sung-June Baek et al. “Baseline correction using asymmetrically reweighted penalized least squares smoothing”. In: *Analyst* 140.1 (2015). Publisher: Royal Society of Chemistry, pp. 250–257. DOI: 10.1039/C4AN01061B. URL: <https://pubs.rsc.org/en/content/articlelanding/2015/an/c4an01061b>.
- [15] Jianfeng Ye et al. “Baseline correction method based on improved asymmetrically reweighted penalized least squares for the Raman spectrum”. In: *Applied Optics* 59.34 (Dec. 1, 2020), pp. 10933–10943. ISSN: 1539-4522. DOI: 10.1364/AO.404863.
- [16] Degang Xu et al. “Baseline correction method based on doubly reweighted penalized least squares”. In: *Applied Optics* 58.14 (May 10, 2019). Publisher: Optica Publishing Group, pp. 3913–3920. ISSN: 2155-3165. DOI: 10.1364/AO.58.003913. URL: <https://opg.optica.org/ao/abstract.cfm?uri=ao-58-14-3913>.
- [17] Feng Zhang et al. “Baseline correction for infrared spectra using adaptive smoothness parameter penalized least squares method”. In: *Spectroscopy Letters* 53.3 (Mar. 15, 2020). Publisher: Taylor & Francis, pp. 222–233. ISSN: 0038-7010. DOI: 10.1080/00387010.2020.1730908. URL: <https://doi.org/10.1080/00387010.2020.1730908>.
- [18] Xiaoran Ning, Ivan W. Selesnick, and Laurent Duval. “Chromatogram baseline estimation and denoising using sparsity (BEADS)”. In: *Chemometrics and Intelligent Laboratory Systems* 139 (Dec. 15, 2014), pp. 156–167. ISSN: 0169-7439. DOI: 10.1016/j.chemolab.2014.09.014. URL: <https://www.sciencedirect.com/science/article/pii/S0169743914002032>.

- [19] C. G. Ryan et al. “SNIP, a statistics-sensitive background treatment for the quantitative analysis of PIXE spectra in geoscience applications”. In: *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms* 34.3 (Sept. 1, 1988), pp. 396–402. ISSN: 0168-583X. DOI: 10.1016/0168-583X(88)90063-8. URL: <https://www.sciencedirect.com/science/article/pii/0168583X88900638>.
- [20] D. Sculley. “Web-scale k-means clustering”. In: *Proceedings of the 19th international conference on World wide web*. WWW ’10. New York, NY, USA: Association for Computing Machinery, Apr. 26, 2010, pp. 1177–1178. ISBN: 978-1-60558-799-8. DOI: 10.1145/1772690.1772862. URL: <https://doi.org/10.1145/1772690.1772862>.
- [21] A. de Juan and R. Tauler. “Chapter 2 - Multivariate Curve Resolution-Alternating Least Squares for Spectroscopic Data”. In: *Resolving Spectral Mixtures*. Ed. by Cyril Ruckebusch. Vol. 30. Data Handling in Science and Technology. ISSN: 0922-3487. Elsevier, 2016, pp. 5–51. DOI: <https://doi.org/10.1016/B978-0-444-63638-6.00002-4>. URL: <https://www.sciencedirect.com/science/article/pii/B9780444636386000024>.
- [22] Charles H. Camp. “pyMCR: A Python Library for Multivariate Curve Resolution Analysis with Alternating Regression (MCR-AR)”. In: *Journal of Research of the National Institute of Standards and Technology* 124 (June 24, 2019), p. 124018. ISSN: 2165-7254. DOI: 10.6028/jres.124.018. URL: <https://nvlpubs.nist.gov/nistpubs/jres/124/jres.124.018.pdf>.
- [23] Michael E. Winter. “N-FINDR: an algorithm for fast autonomous spectral end-member determination in hyperspectral data”. In: *Imaging Spectrometry V*. Imaging Spectrometry V. Vol. 3753. SPIE, Oct. 27, 1999, pp. 266–275. DOI: 10.1117/12.366289. URL: <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/3753/0000/N-FINDR--an-algorithm-for-fast-autonomous-spectral-end/10.1117/12.366289.full>.
- [24] Guliev, Rustam. *pyspc-unmix: Python package for unmixing hyperspectral data*. original-date: 2022-08-26T15:11:30Z. Aug. 26, 2022. URL: <https://github.com/r-hyperspec/pyspc-unmix>.
- [25] Charles R. Harris et al. “Array programming with NumPy”. In: *Nature* 585.7825 (Sept. 2020). Publisher: Springer Science and Business Media LLC, pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: <https://doi.org/10.1038/s41586-020-2649-2>.
- [26] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

- [27] Pauli Virtanen et al. “SciPy 1.0: fundamental algorithms for scientific computing in Python”. In: *Nature Methods* 17.3 (Mar. 2, 2020), pp. 261–272. ISSN: 1548-7091, 1548-7105. DOI: 10.1038/s41592-019-0686-2. URL: <http://www.nature.com/articles/s41592-019-0686-2>.
- [28] Wes McKinney. “Data Structures for Statistical Computing in Python”. In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.
- [29] The pandas development team. *pandas-dev/pandas: Pandas*. Version latest. Feb. 2020. DOI: 10.5281/zenodo.3509134. URL: <https://doi.org/10.5281/zenodo.3509134>.
- [30] Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. “Numba: a LLVM-based Python JIT compiler”. In: *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. LLVM ’15. New York, NY, USA: Association for Computing Machinery, Nov. 15, 2015, pp. 1–6. ISBN: 978-1-4503-4005-2. DOI: 10.1145/2833157.2833162. URL: <https://dl.acm.org/doi/10.1145/2833157.2833162>.
- [31] P Umesh. “Image Processing in Python”. In: *CSI Communications* 23 (2012). Publisher: Citeseer.
- [32] J. D. Hunter. “Matplotlib: A 2D graphics environment”. In: *Computing in Science & Engineering* 9.3 (2007). Publisher: IEEE COMPUTER SOC, pp. 90–95. DOI: 10.1109/MCSE.2007.55.
- [33] Shuxia Guo, Jürgen Popp, and Thomas Bocklitz. “Chemometric analysis in Raman spectroscopy from experimental design to machine learning-based modeling”. In: *Nature Protocols* 16.12 (Dec. 2021). Number: 12 Publisher: Nature Publishing Group, pp. 5426–5459. ISSN: 1750-2799. DOI: 10.1038/s41596-021-00620-3. URL: <https://www.nature.com/articles/s41596-021-00620-3>.

List of Figures

1.1	3rdPlace and Datrix logos	1
1.2	NEWMED logo	2
1.3	CRIMSON logo	3
2.1	HSI data cube	5
2.2	Toluene spectral signature	6
2.3	Raman energy levels	7
2.4	HSI data sample: microglial cells	8
2.5	HSI data sample: breast tissue	8
2.6	HSI data sample: neuronal cell	8
2.7	HSI data sample: leukemic cell	9
2.8	RamApp logo	11
3.1	RamApp interface	14
3.2	Map rotation example	16
3.3	Map crop example	16
3.4	Smooth map example	18
3.5	Savitzky-Golay filter example	18
3.6	Whittaker filter example	19
3.7	Despike example	20
3.8	Polynomial baseline correction example	22
3.9	IModPoly baseline correction example	23
3.10	Goldindex baseline correction example	23
3.11	arPLS baseline correction example	24
3.12	IarPLS baseline correction example	24
3.13	drPLS baseline correction example	25
3.14	asPLS baseline correction example	25
3.15	Rubberband baseline correction example	26
3.16	BEADS baseline correction example	27
3.17	SNIP baseline correction example	27

3.18	Identify substrate example	28
3.19	Univariate analysis interface	29
3.20	Univariate analysis example	30
3.21	Clustering example	31
3.22	MCR-ALS example	34
3.23	N-FINDR example	35
3.24	RamApp architecture schema	36
3.25	Frontend technology stack	37
3.26	Backend technology stack	38
3.27	RamApp login interface	38
3.28	RamApp user storage interface	39
3.29	Ramappy technology stack	39
4.1	Upload interface	41
4.2	Data exploration interface	42
4.3	Use case example: Despiking	43
4.4	Use case example: Correct baseline	44
4.5	Use case example: Identify substrate	45
4.6	Use case example: Smooth map	45
4.7	Use case example: Normalisation	46
4.8	Use case example: PERFECTA	47
4.9	Use case example: DNA	47
4.10	Use case example: Organic matrix	48
4.11	Use case example: Result	48
5.1	Future development: CARS to Raman neural network	50

Acknowledgements

RamApp has received funding from the following projects:

- *NEWMED*, Lombardy Region's 2014-2020 Regional Operational Programme.
- *CRIMSON*, European Union's Horizon 2020 research and innovation programme under grant agreement N°101016923.

Infine, vorrei aggiungere alcune considerazioni personali.

Desidero innanzitutto ringraziare i miei colleghi Andrea, Giulia, Manuela e Matteo per avermi supportato e accompagnato durante tutte le fasi di sviluppo del progetto, che è stato per me un'importante opportunità di crescita personale e professionale. Insieme a loro, l'enorme contributo e l'entusiasmo portati da Renzo hanno reso questo percorso molto stimolante.

Non posso inoltre non includere in questi ringraziamenti i miei amici e «colleghi» Andrea e Simone, insieme ai quali ho condiviso gran parte dei miei studi.

Ultima ma non per importanza, ringrazio la mia famiglia, senza la quale non avrei potuto raggiungere questo prestigioso traguardo.

