

# PYSPECTRUM: AN OPEN SOURCE SOFTWARE FOR HYPERSPECTRAL ANALYSIS

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

In Partial Fulfillment of the Requirements for the Degree of

Master of Science

by

Danielle Lieber

August 2016

© 2016 Danielle Lieber

## ABSTRACT

Hyperspectral imaging is a technique in which a unique spectrum is obtained at each spatial point on a sample. The collected data is used to construct an image showing the amount and distribution of constituent components of the sample. Datasets such as these may be immense in scale and require computer processing and analysis techniques to adequately study each of its facets. PySpectrum software was created to be a free and open source tool to facilitate analysis of Raman hyperspectral data. The software functionalities include baseline subtraction, generation of scrollable pixel intensity maps, and creation of heat maps through peak selection. These functionalities work on both area and line scans. Spectra can be displayed by choosing the spatial location the user would like to explore. Peaks in each respective spectrum are analyzed using a peak selection interface. The software capabilities were tested on three Raman datasets taken from biological samples. Heat maps showing the concentration of apatite, collagen, and lipids in an area scan of a bovine meniscal enthesis are detailed.

## BIOGRAPHICAL SKETCH

Danielle Lieber attended Stony Brook University in Stony Brook, New York from 2009 to 2014, studying Engineering Science. She received her Bachelor of Engineering degree after graduating *magna cum laude*. While at Stony Brook, she was a teaching assistant for several chemistry and engineering laboratory classes. She also participated in a National Science Foundation Research Experience for Undergraduates site in the summer of 2013 at the Materials Engineering School of Purdue University in West Lafayette, Indiana. At this site, Lieber worked with the research group of Professors John Howarter and Inez Hua on a project involving the use of clay as an environmentally friendly alternative flame retardant for eventual incorporation in printed circuit boards.

Following her graduation, she attended Cornell University in Ithaca, New York in pursuit of a Master of Science degree in Materials Science and Engineering. After joining Professor Lara Estroff's research group, Lieber worked on a project studying calcite crystal growths in the presence of polymeric additives. In the following year from 2015 to 2016, she developed PySpectrum, an open source software for analysis of hyperspectral datasets.

## ACKNOWLEDGMENTS

I would like to thank my thesis advisor Professor Lara Estroff for continuing help and guidance, as well as support in pursuing a project that combined my interests in materials science and programming. In addition, thank you to my committee member Professor Eve Donnelly, for offering excellent feedback and suggestions in regards to the future direction of PySpectrum software.

I would also like to thank the developers of Cornell Spectrum Imager, Robert Hovden and Paul Cueva, who allowed me to collaborate with them on expansions to their software and aided in the development of PySpectrum by providing an example to aspire to.

Thank you to Alex Boys, Jennie Russ Kunitake, and Kiara Chan for contributing valuable data and feedback to help refine the software.

Finally, thank you to my partner and experienced programmer Pedro Rittner, for answering any questions I had about the specifics of coding and for the unending love and encouragement needed to accomplish this thesis.

## TABLE OF CONTENTS

|   |    |
|---|----|
| Abstract .....  | i  |
| Biographical Sketch .....   | ii |
| List of Figures .....   | v  |
| 1 Introduction .....  | 1  |
| 1.1 Raman Spectroscopy.....   | 1  |
| 1.2 Hyperspectral Data .....  | 1  |
| 1.3 Computational Analysis and Review of Non-Commercial Analysis Software ..... | 5  |
| 2 Use of PySpectrum .....   | 7  |
| 2.1 Download and Install .....  | 7  |
| 2.2 Open and View Datasets.....   | 7  |
| 2.3 Navigation Toolbar Functions .....  | 9  |
| 2.4 Baseline Subtraction .....  | 9  |
| 2.5 Data Visualization .....  | 11 |
| 2.5.1 Intensity Maps .....  | 11 |
| 2.5.2 Concentration Heat Maps .....   | 11 |
| 3 Workflow Example .....  | 12 |
| 4 Test Data Analysis .....  | 15 |
| 5 Discussion .....  | 18 |
| 4.1 Python as Programming Language .....  | 18 |
| 4.2 File Formats .....  | 19 |
| 4.3 Open Source Software .....  | 19 |
| 6 Conclusions .....   | 22 |
| 7 References .....  | 23 |

## LIST OF FIGURES

|   |    |
|---|----|
| Figure 1.1: Hyperspectral imaging process diagram .....   | 2  |
| Figure 1.2: Bright field, C-H intensity, and PCA images of HeLa cell.....   | 3  |
| Figure 1.3: Cytochrome c, protein, and lipid distribution images of HeLa cell .....   | 3  |
| Figure 1.4: Time-resolved imaging of cytochrome c, protein, and lipid distributions in HeLa cell<br>during cytokinesis .....                  | 4  |
| Figure 1.5: Intensity maps of muscle tissue for lipid analysis.....   | 5  |
| Figure 2.1: Screenshot of the file browser and main PySpectrum plot window.....   | 8  |
| Figure 2.2: The navigation toolbar shortcut icons and functions.....  | 9  |
| Figure 2.3: Example of baseline subtraction method used .....   | 10 |
| Figure 2.4: Screenshot of an individual spectrum before and after baseline subtraction .....  | 10 |
| Figure 3.1: Screenshots of PySpectrum interface for scrollable intensity maps.....  | 13 |
| Figure 3.2: Heat map example with peak selection window .....   | 14 |
| Figure 4.1: Plot displays of spectra collected from Raman area scan of a medial enthesis.....   | 16 |
| Figure 4.2: Heat maps of the area scanned medial enthesis section showing location and<br>concentration of apatite, lipid, and collagen ..... | 17 |
| Figure 4.3: Intensity map and heat maps showing CH and amide I concentration in line scan<br>from tibial plateau .....                        | 17 |

## **CHAPTER 1 INTRODUCTION**

Hyperspectral imaging is an innovative approach to quantitative analysis with applications ranging from environmental and agricultural monitoring, mineral surveys, and chemical hazard detection. It is also an emerging research tool that can be implemented using a variety of microscopy techniques including energy-dispersive X-ray spectroscopy (EDS), electron energy loss spectroscopy (EELS), infrared spectroscopy (IR), and Raman spectroscopy.

### **1.1 Raman Spectroscopy**

Raman spectroscopy is a non-destructive vibrational spectroscopic technique perfectly suited for microscopy and hyperspectral imaging. It is used to identify molecules in a sample using “fingerprint” wavenumbers. Samples can be analyzed in water with minimal preparation, which makes Raman microscopy an excellent technique for examining biological and soft tissue samples. Several biological samples were analyzed using Raman hyperspectral imaging and are discussed in the subsequent sections of this thesis.

### **1.2 Hyperspectral Data**

Hyperspectral imaging can provide detailed information about the composition of a specimen. As shown in Figure 1.1, a unique spectrum is acquired at each spatial point on the sample. The collected spectra are used to create images showing the location and amount of different components in the sample, combining the advantages of spectroscopic methods and imaging. Hyperspectral imaging techniques are particularly important in biological cell and tissue studies, where fluorescent staining may cause unwanted effects. For example, photobleaching occurs when a fluorescent dye or molecule permanently loses the ability to fluoresce due to chemical interactions during imaging [1]. As detailed in the following findings,



hyperspectral imaging is easily applied in analysis of cell samples to identify individual components.

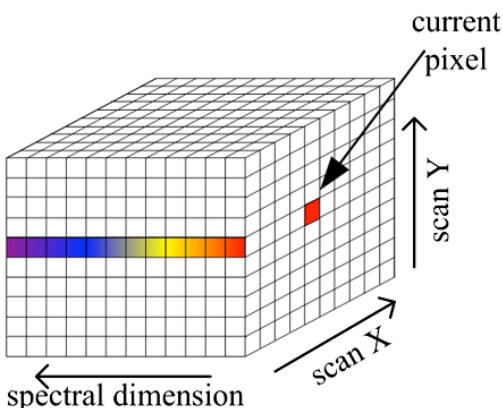


Figure 1.1. A spectrum taken at each pixel located at a spatial coordinate on a sample is used to create a hyperspectral image [2].

Hyperspectral Raman scans of label-free lab-grown HeLa cells were analyzed in order to create an image reconstruction highlighting specific components in the cell [3]. A bright-field image of the cell in water is seen in Figure 1.2a. CH stretching intensities in the  $2800\text{-}3100\text{ cm}^{-1}$  wavenumber range were integrated and mapped in combination with spatial data. Figure 1.2b shows the resulting image, where dark brown represents the lowest intensities present and yellow corresponds to highest intensities. The researchers were able to show that this technique highlighted the nucleus, nucleoli, and mitochondria regions of the individual cell. Additionally, researchers applied principle component analysis (PCA) techniques to extract the three most abundant components in the datasets. These three principle components were mapped and overlaid onto a single image, which is shown in Figure 1.2c [3].

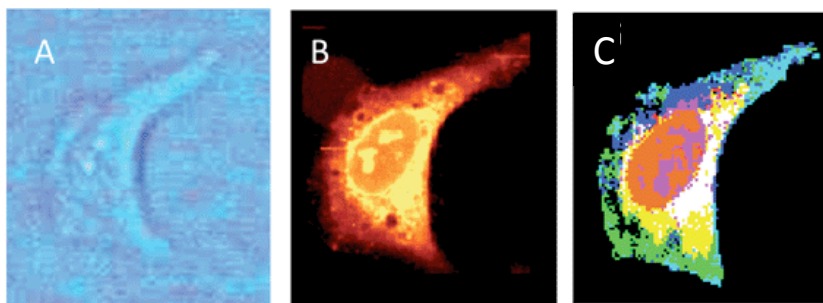


Figure 1.2. (a) Bright field image of HeLa cell in water (b) Image created from integrated Raman CH stretching intensities (dark brown lowest, yellow highest intensities). (c) Image overlay of principle components 1, 2 and 3 from PCA [3].

In a similar study, living HeLa cells were imaged using Raman scattering. Reconstructed hyperspectral images showing the distribution of cytochrome c, protein beta sheet, and lipid components in the cell are shown in Figure 1.3 [4]. These were obtained by integrating over the Raman peaks at  $753\text{ cm}^{-1}$ ,  $1686\text{ cm}^{-1}$  (Amide I), and  $2852\text{ cm}^{-1}$  ( $\text{CH}_2$  stretching) respectively [4].

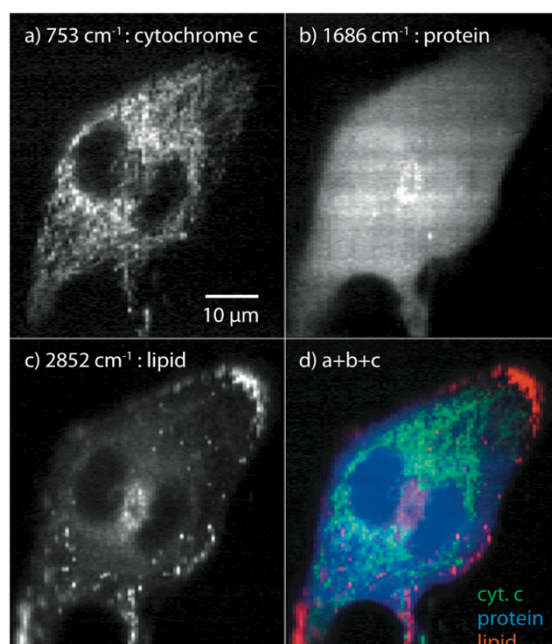


Figure 1.3. Raman scattering images of unstained, unlabeled living HeLa cells reconstructed using Raman signals at (a)  $753\text{ cm}^{-1}$ , (b)  $1686\text{ cm}^{-1}$ , and (c)  $2852\text{ cm}^{-1}$ , showing the distribution of cytochrome c, protein beta sheet, and lipid molecules, respectively. Image (d) was constructed by merging images (a) through (c) [4].

Additionally, the cell was dynamically imaged during cytokinesis. Raman scans were taken at 115-second intervals. The same components that were mapped in Figure 1.3 are mapped in Figure 1.4 at 4 different time intervals. The components were overlaid into a single image at each time interval. Using these hyperspectral images, researchers were able to observe that cytochrome c was highly concentrated near the cleavage furrow during division of the cell [4]. Because the distribution of cytochrome c is thought to play a role in apoptosis, showing cytochrome c concentration *in situ* may elucidate its particular functions.

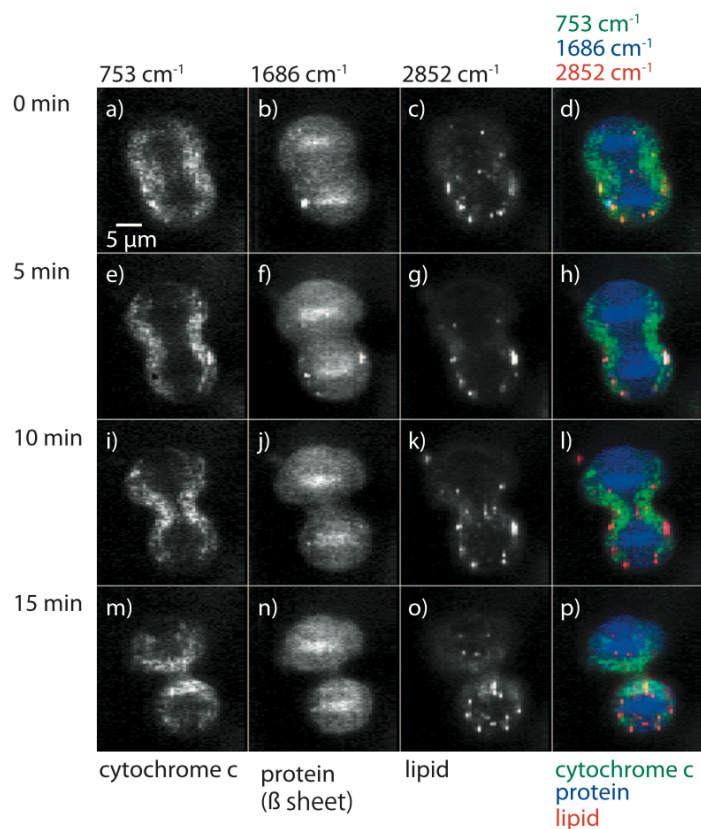


Figure 1.4. Images of cytochrome c, protein, and lipid molecule distributions in an unlabeled HeLa cell during cytokinesis taken at 5 minute intervals [4].

Hyperspectral imaging techniques have also been shown to be useful on larger scale samples. In one study, muscle tissue was analyzed for the purpose of finding lipid droplets [5]. Understanding the chemistry in the tissue surrounding these lipid accumulations would elucidate

their role in development of diseases such as type 2 diabetes. 20 $\mu$ m and 5 $\mu$ m thick sections of transverse abdominal muscles from rats fed a high-fat diet were imaged using Raman microscopy. As shown in Figures 1.5a and 1.5b, intensity maps were generated by integrating over the CH wavenumber region (2800 to 3100  $\text{cm}^{-1}$ ) and  $\text{CH}_2$  stretching (2845  $\text{cm}^{-1}$ ) peak seen in Figure 1.5c. The CH map was effective in correlating higher concentrations of CH groups with higher deposits of lipids. The researchers narrowed the CH region to  $\text{CH}_2$ , generating a map highlighting lipids with long acyl chains. The high intensity regions in this map matched the high intensity regions in the CH map and highlighted lipid droplets 2-3 $\mu$ m in size, consistent with samples that were analyzed using fluorescence imaging [5].

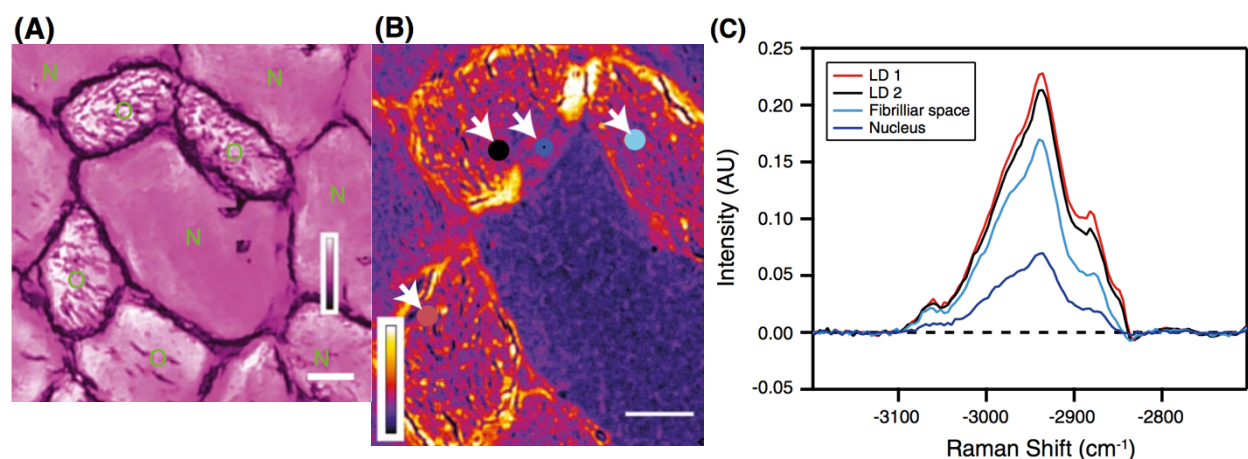


Figure 1.5. (a) Intensity map of integrated Raman spectra over the CH region of the 20 $\mu$ m thick sample. (b) Intensity map of integrated Raman spectra over the  $\text{CH}_2$  peak of the 5 $\mu$ m thick sample (c) Raman spectra corresponding to lipid droplets, myofibrillar space, and nucleus. Arrows in (b) show where spectra in (c) were collected. Scale bars = 20 $\mu$ m in (a) and (b) [5].

### 1.3 Computational Analysis and Review of Non-Commercial Analysis Software

Although hyperspectral imaging is a valuable tool for identifying distributions of components in a sample, it results in an extremely large quantity of data. Computational tools are necessary in order to manage the data and produce meaningful results in the form of hyperspectral images. Therefore, PySpectrum was developed as a free and open source software

created to assist researchers in processing and analyzing hyperspectral data primarily obtained from Raman spectroscopy.

The inspiration for PySpectrum software came from working with Cornell Spectrum Imager (CSI) [6]. CSI is a plugin for ImageJ developed for analyzing hyperspectral data, specifically atomic resolution electron energy loss spectra. While ImageJ is a widely used software within academia, there were several problems encountered when trying to integrate the desired functionality that eventually became part of PySpectrum. The integration problems and their solutions are detailed in the discussion section (Chapter 5).

Several other well-known non-commercial hyperspectral analysis programs include hyperSpec [7], Hyperspectral Image Analysis Toolbox (HIAT) [8], and HyperSpy [9]. These programs provide similar features to PySpectrum, but are extensions to existing programs such as MATLAB. Software like hyperSpec and HyperSpy are text-based programming language modules that function by being written into an existing piece of code. Therefore, there is a limited graphical user interface and the learning curve for these tools can be steeper than expected. PySpectrum was developed with the aim to be more accessible for researchers from a variety of backgrounds. As such, it is a self-contained software with a simple graphical user interface that can be run on Windows, Mac OS X, and Linux operating systems.

## CHAPTER 2

### USE OF PYSPECTRUM

#### 2.1 Download and Install

There are several options available to download the program. The source code can be downloaded from GitHub, where files are in .py format [10]. The files can be run on any machine with a Python interpreter. Additionally, an executable with bundled dependencies including all libraries used is available for Linux and Windows. For all operating systems including Mac OSX, PySpectrum will be available as a one-line installable using the Python pip package manager.

#### 2.2 Open and View Datasets

Upon opening PySpectrum, select the *Line Scan* option if uploading a one-dimensional dataset or *Area Scan* option if uploading a two-dimensional dataset. When the directory dialog browser appears, select the directory in which the text-file data is contained. This becomes the working directory where output files are saved. The main PySpectrum application window as shown in Figure 2.1 will appear after directory selection. Two drop down menus labeled *Select X* and *Select Y* are used to select the spectral data specific to a certain spatial area of the sample. Once an X and Y are selected, the *Plot Spectrum* button will display the plotted data as an intensity versus wavenumber graph.

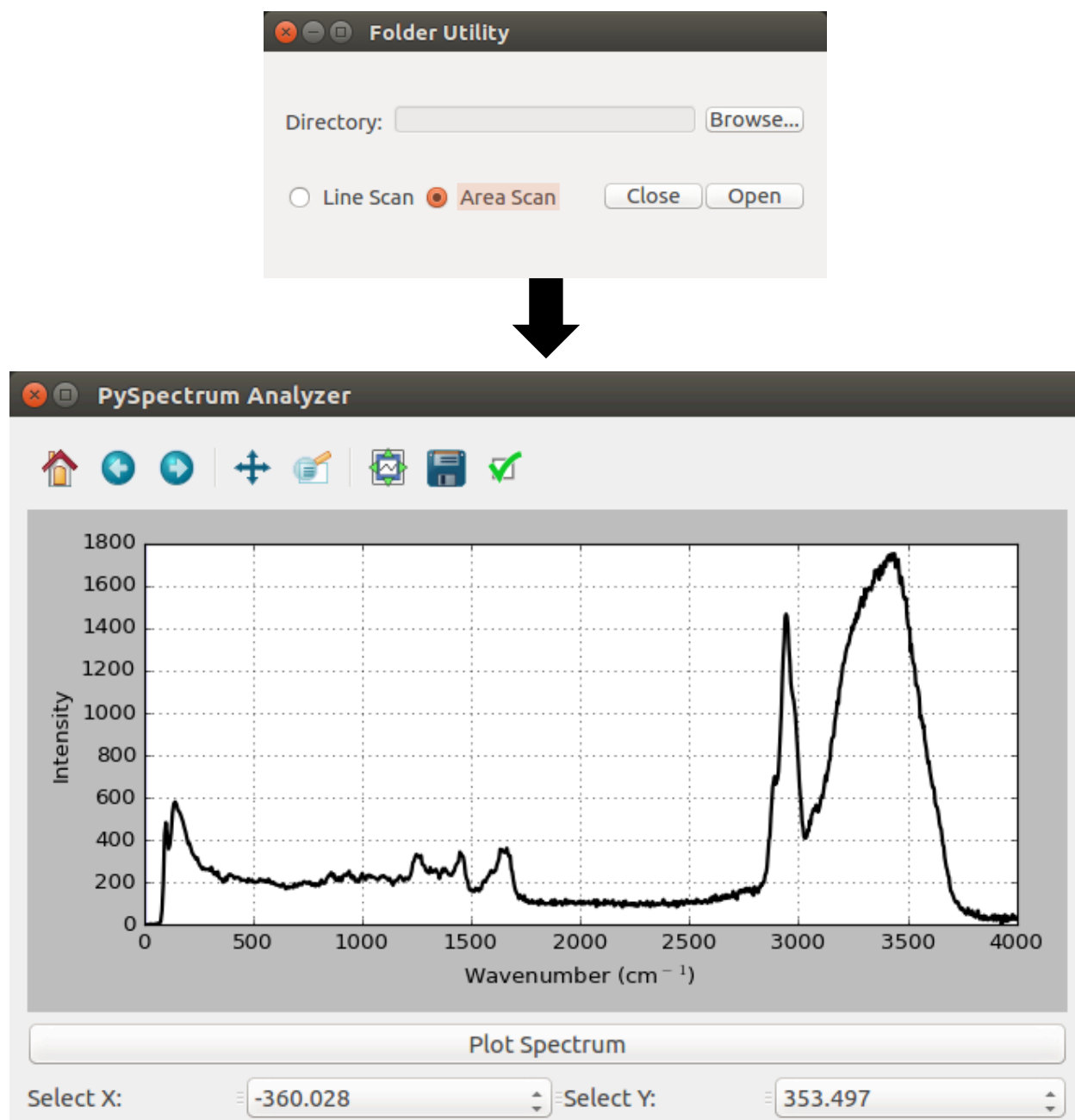


Figure 2.1. The option to upload a line or area scan is selected upon opening the folder utility. An individual spectrum is viewed in the plot window as shown by selecting a spatial location from the drop down menus and clicking *Plot Spectrum*.

## 2.3 Navigation Toolbar Functions

The interactive toolbar at the top of the application window is used to navigate between plots, edit plot and window appearance, and save work. It contains the following buttons shown in Figure 2.2: *home*, *forward*, *backward*, *pan/zoom*, *zoom to rectangle*, *configure plot*, *save*, and *figure options*. The *configure plot* button configures the borders of the plot and workspace using sliders. The *figure options* button is used to title the plot, set the axes labels and scales, and set the color, size and shape of data points. Once configured, the displayed spectrum can be saved in the following file formats: .png, .ps, .eps, .svg, and .pdf.

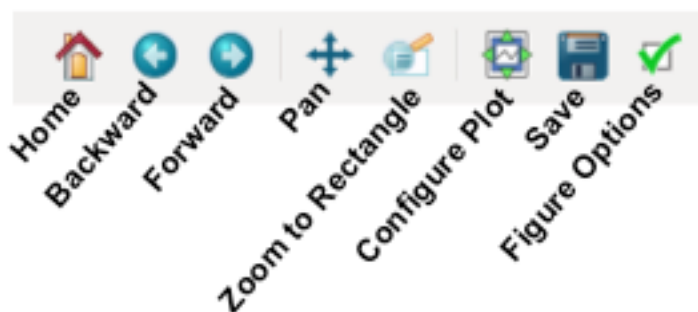


Figure 2.2. The navigation toolbar shortcut icons and functions.

## 2.4 Baseline Subtraction

The *Baseline Correction* button performs a baseline subtraction on the current spectrum. Baseline correction was implemented by dividing the spectrum into smaller sections, finding the minimum vertex points in each section [7]. Linear interpolation was used to construct the baseline these vertices, similar to the red line in Figure 2.3. The resulting line is then subtracted from the original spectrum. The new spectrum will be displayed in place of the old spectrum. A spectrum before and after baseline subtraction is seen in Figure 2.4. Selecting a wavenumber range to create a heat map can be done on either baseline-unsubtracted and baseline-subtracted spectra.



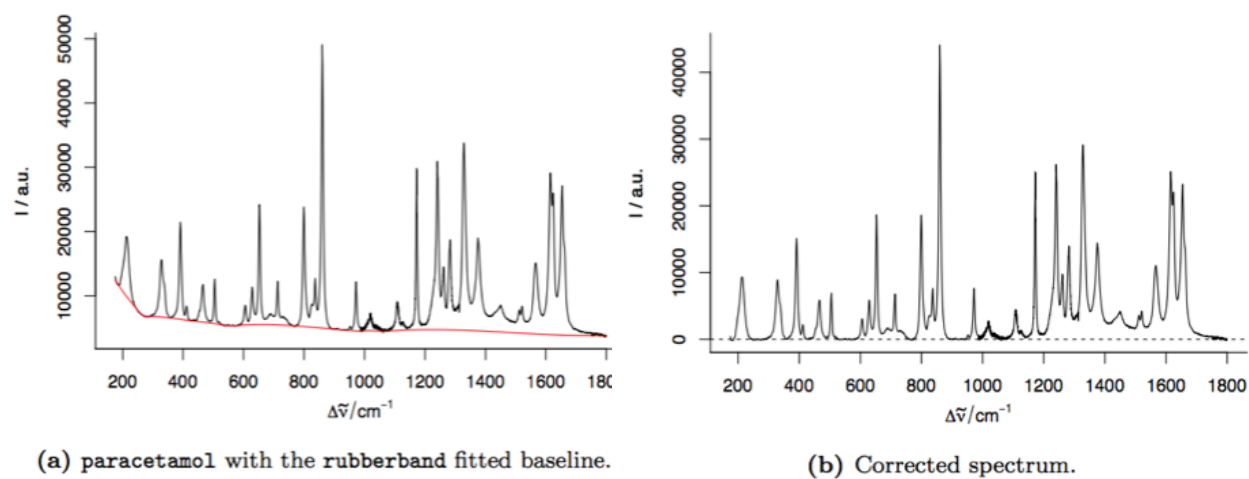


Figure 2.3. Example of baseline subtraction method used [7].

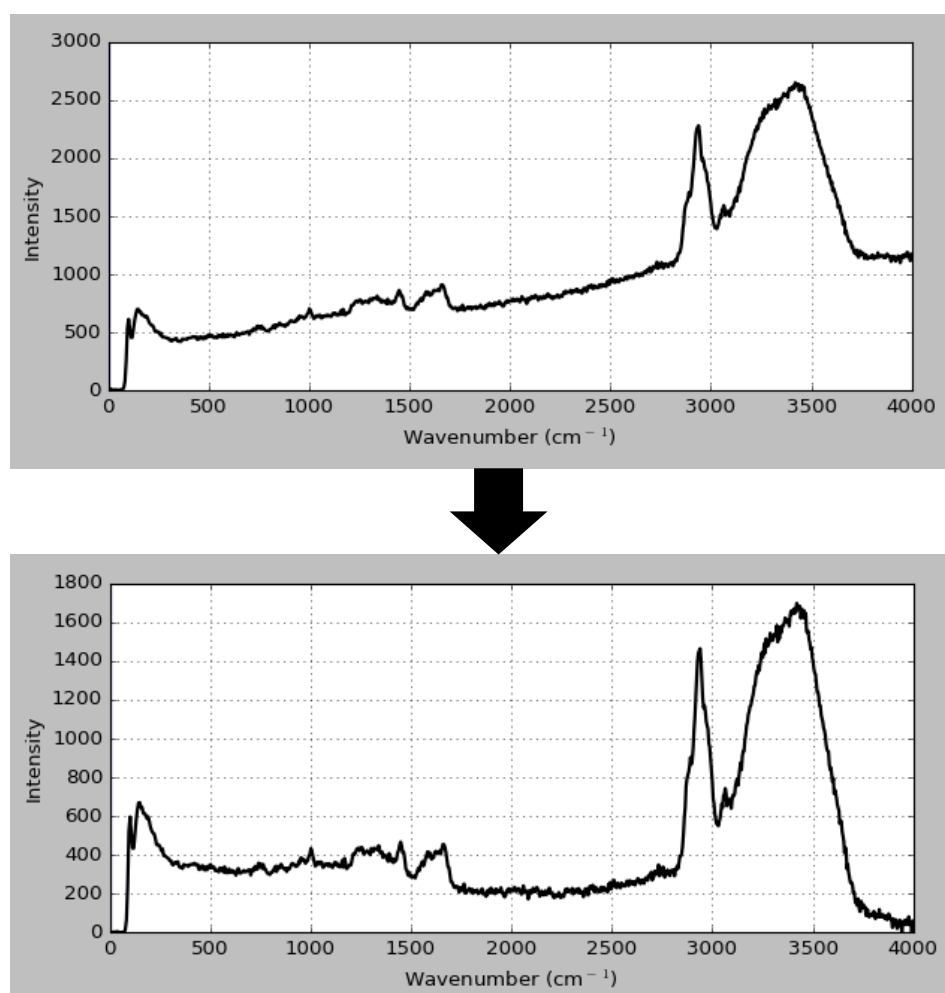


Figure 2.4. Screenshot of an individual spectrum in the main plot window before (top) and after (bottom) baseline subtraction.

## 2.5 Data Visualization

Data can be visualized in two separate ways, intensity maps or concentration heat maps.

### 2.5.1 Intensity Maps

Select the *View Image Stack* button to generate intensity maps showing the intensity in grayscale at each pixel of the image. The images named by wavenumber will be created from the data, saving them in the previously selected directory. Depending on the amount of data, the image generation may take up to a few minutes. The images are viewed on the software as a scrollable stack from lowest to highest wavenumber. A screenshot of the scrollable interface is shown in Figure 3.1.

### 2.5.2 Concentration Heat Maps

Concentrations of material of interest can be visually displayed as a heat map. The heat map shows the intensity of the chosen component in the dataset. It is displayed by selecting a peak of interest in the currently viewed spectrum. The peak is selected using the sliders labeled *left* and *right*. These move two vertical lines on the plot, one to the immediate left of the peak and one to the immediate right. Once the peak is selected, PySpectrum will use the range of wavenumbers across the peak to find peaks in the same range in the entire data set. These are then plotted as an image with a color bar to spatially show intensity across the sample specific to that range of wavenumbers. Axes are scaled according to the image array dimensions, as number of rows by number of columns. The color bar axis is scaled according to peak area.

### CHAPTER 3

#### WORKFLOW EXAMPLE

A workflow example showing the main features of PySpectrum is shown here. The data used in this workflow was taken on the Renishaw InVia Confocal Raman microscope on a sample of breast cancer tissue [11]. The aim of the study from which this data was taken is to explore the correlation between microcalcifications like apatite and malignant cell growth, specifically in cases of ductal carcinoma.

Figure 3.1 shows a sequence of several intensity map images of the sample at various wavenumbers. The intensity stack is scrollable within the PySpectrum application to highlight features that would otherwise be obscured. For example, the duct is extremely visible in the center of the image taken at wavenumbers  $114\text{ cm}^{-1}$  and  $1007\text{ cm}^{-1}$ , but loses contrast wavenumber  $3765\text{ cm}^{-1}$ .

Information from the intensity maps can direct the user to select certain wavenumber ranges for creating a concentration heat map. The intensity map showed the central duct feature noticeably visible at wavenumbers  $900\text{-}1000\text{ cm}^{-1}$ . Combined with the information that apatite has a Raman fingerprint wavenumber at  $960\text{ cm}^{-1}$ , selecting a peak in the range of  $930\text{-}965\text{ cm}^{-1}$  generated a heat map of the sample that can be seen in Figure 3.2. A high intensity cluster is shown in the center of the heat map within the ductal region.

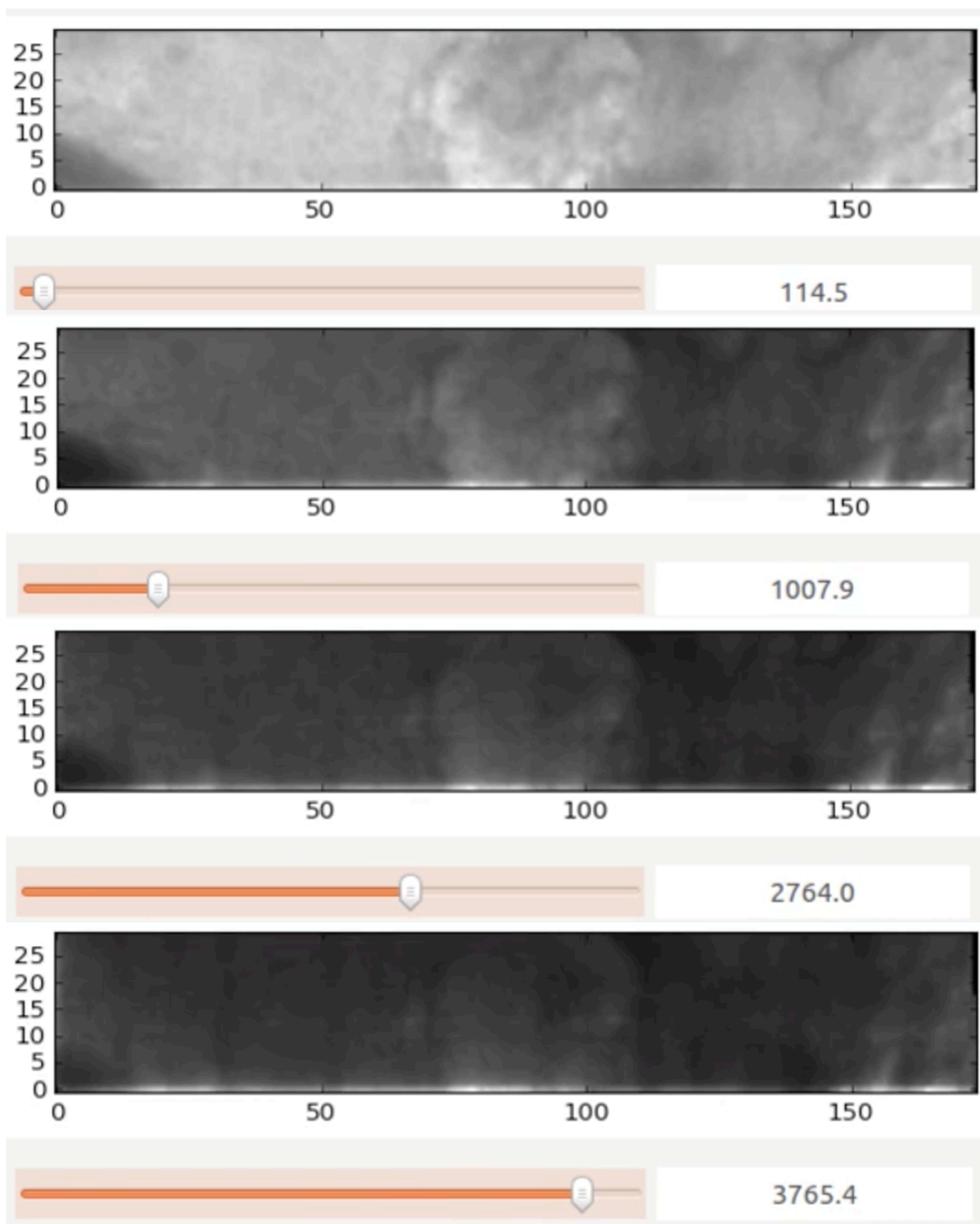


Figure 3.1. After clicking the *View Image Stack* button, PySpectrum will generate TIFF images mapping the data at each wavenumber. These will be saved in the working directory and can be viewed as a scrollable stack. Images at  $114\text{ cm}^{-1}$ ,  $1007\text{ cm}^{-1}$ ,  $2764\text{ cm}^{-1}$ , and  $3765\text{ cm}^{-1}$  wavenumbers are shown in the scrollable application window.

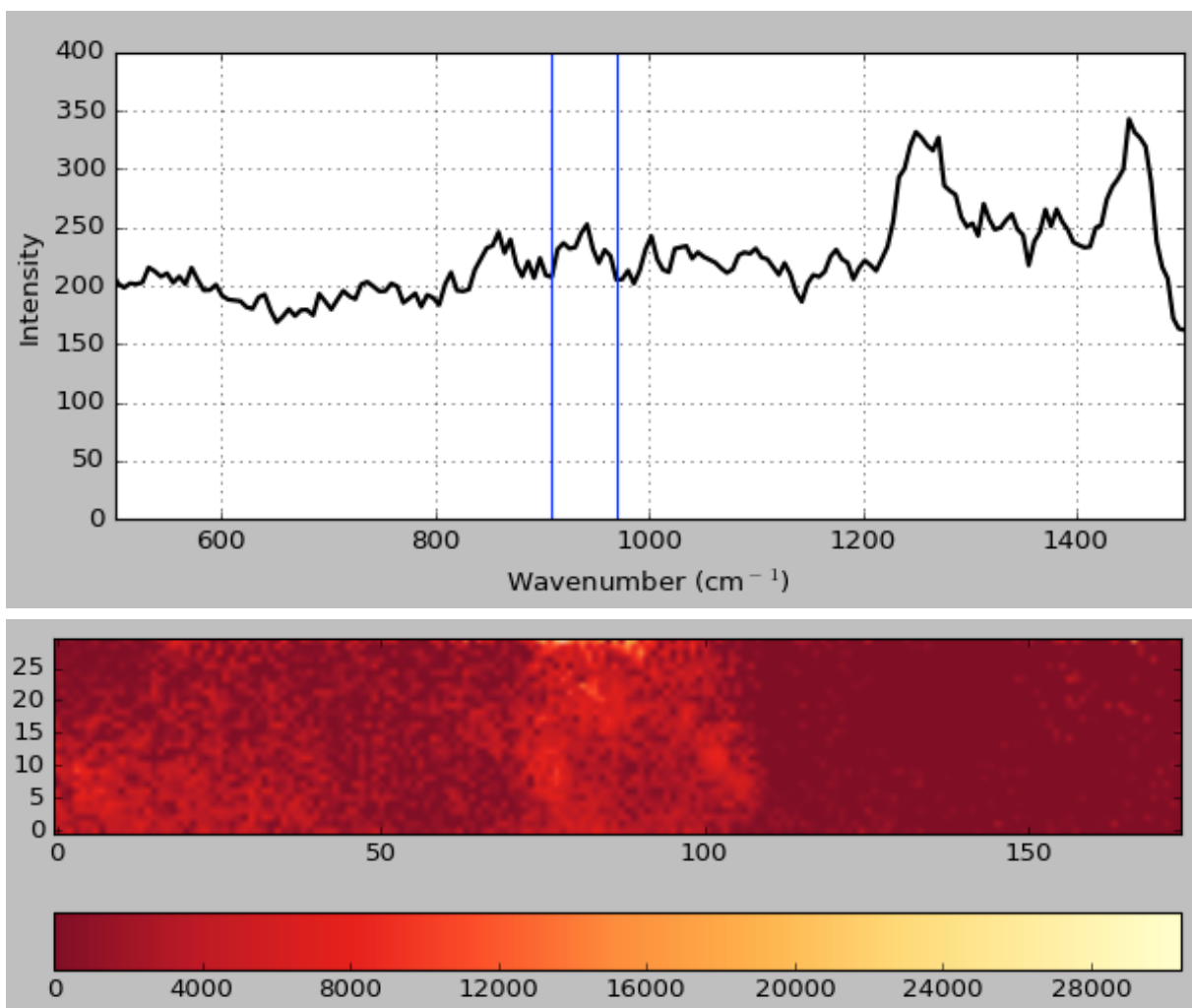


Figure 3.2. Screenshot of spectrum with peak selection lines at 931 and 965 cm<sup>-1</sup> and resulting heat map. The spectrum was zoomed in to be displayed between 500 and 1500 cm<sup>-1</sup>. Apatite is found at 960<sup>-1</sup>, contributing to the high intensity found in the center of the duct shown in the heat map.

## CHAPTER 4 TEST DATA ANALYSIS

The software was tested using Raman scans on sections taken from a bovine meniscal enthesis [12, 13]. The meniscal enthesis is a small piece of tissue that attaches the menisci to bone in the knee joint. The Raman scans were used to characterize the enthesis, which consists of ligamentous tissue, uncalcified fibrocartilage, calcified fibrocartilage, and bone.

Due to the varied and complex tissue make-up, characterization and in-depth study of the enthesis will aid in the effort to produce a similar synthetic structure. Meniscal implants can be used in cases of knee osteoarthritis and degenerative tears, but long-term viability of these implants depend on the quality of their underlying framework.

Detailed in this section are the results of software analysis on an area scan of a section from the medial caudal meniscal enthesis [12] and a line scan of a section of the tibial plateau region [13] including only the ligamentous uncalcified cartilage components.

Screenshots of displayed spectra from the medial meniscal enthesis sample are shown in Figure 4.1. Peak selection was performed on each of the three spectra using wavenumber ranges  $930\text{-}970\text{ cm}^{-1}$ ,  $2823\text{-}2879\text{ cm}^{-1}$ , and  $2907\text{-}3024\text{ cm}^{-1}$ . The wavenumber ranges correspond to Raman spectral fingerprints of apatite, lipid, and collagen, respectively. The heat maps displaying the concentrations of these components are shown in Figure 4.2.

Line scan data from the lateral meniscal enthesis can be seen in Figure 4.3. Heat maps from this sample were generated from the C-H stretching intensity at the  $1452\text{ cm}^{-1}$  wavenumber and Amide I intensity at the  $1667\text{ cm}^{-1}$  wavenumber.

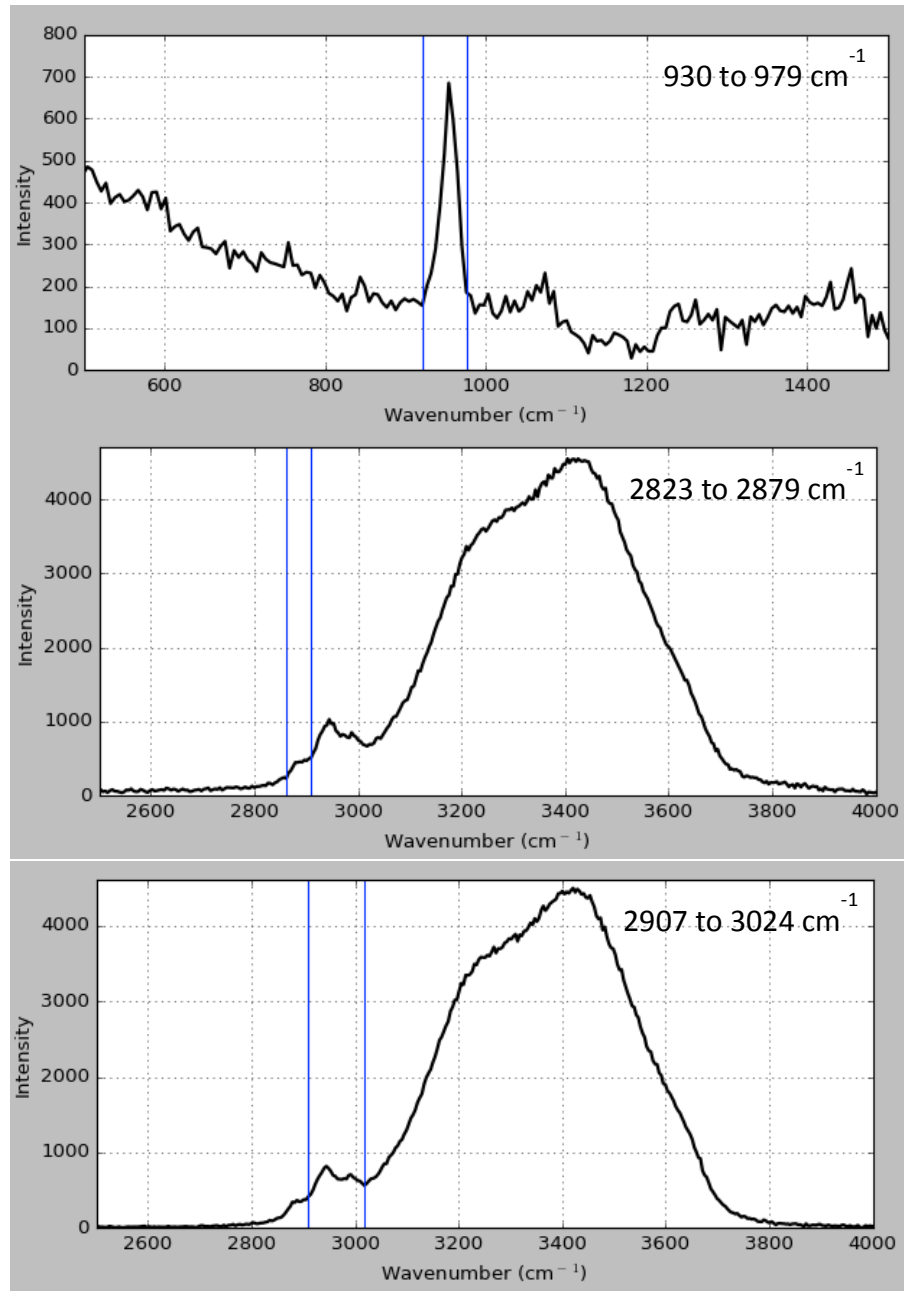


Figure 4.1. Plot displays of spectra collected from Raman area scan of a medial enthesis section. Peaks are highlighted at 930-970  $\text{cm}^{-1}$ , 2823-2879  $\text{cm}^{-1}$ , and 2907-3024  $\text{cm}^{-1}$ . These wavenumber ranges were used to create heat maps showing concentrations of apatite, lipid, and collagen, respectively.

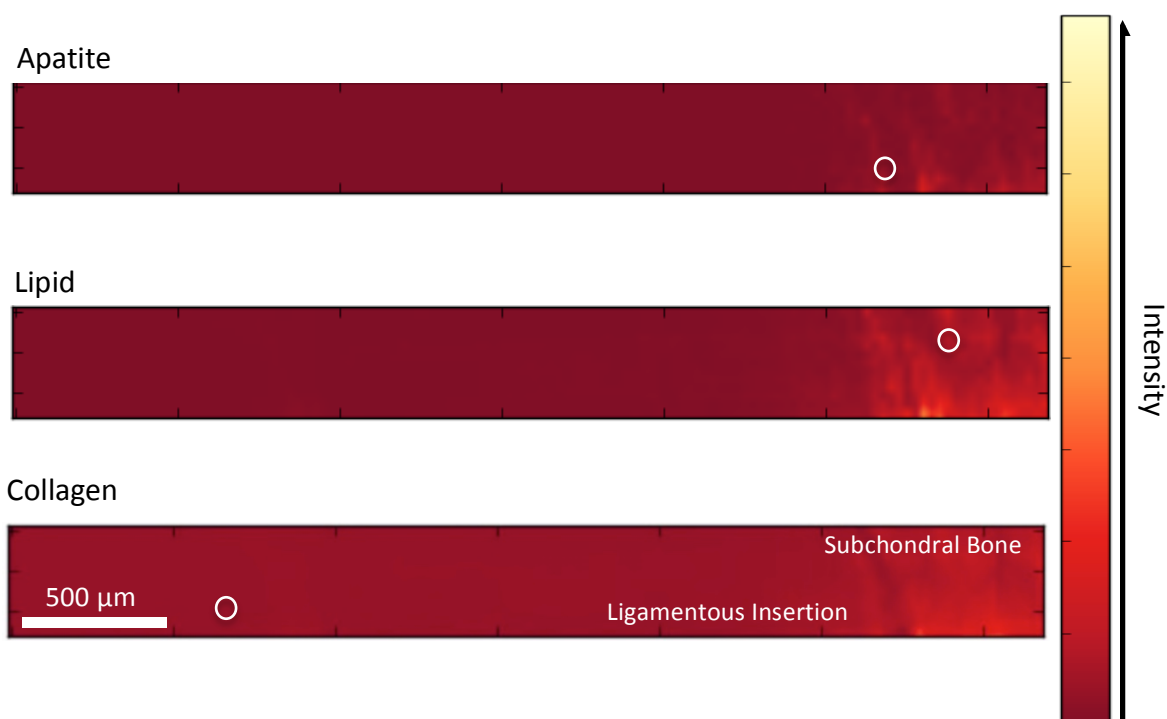


Figure 4.2. Heat maps of the area scanned medial enthesis section showing location and concentration of apatite, lipid, and collagen components. Circles represent locations at which corresponding spectra shown in Figure 4.1 were taken.

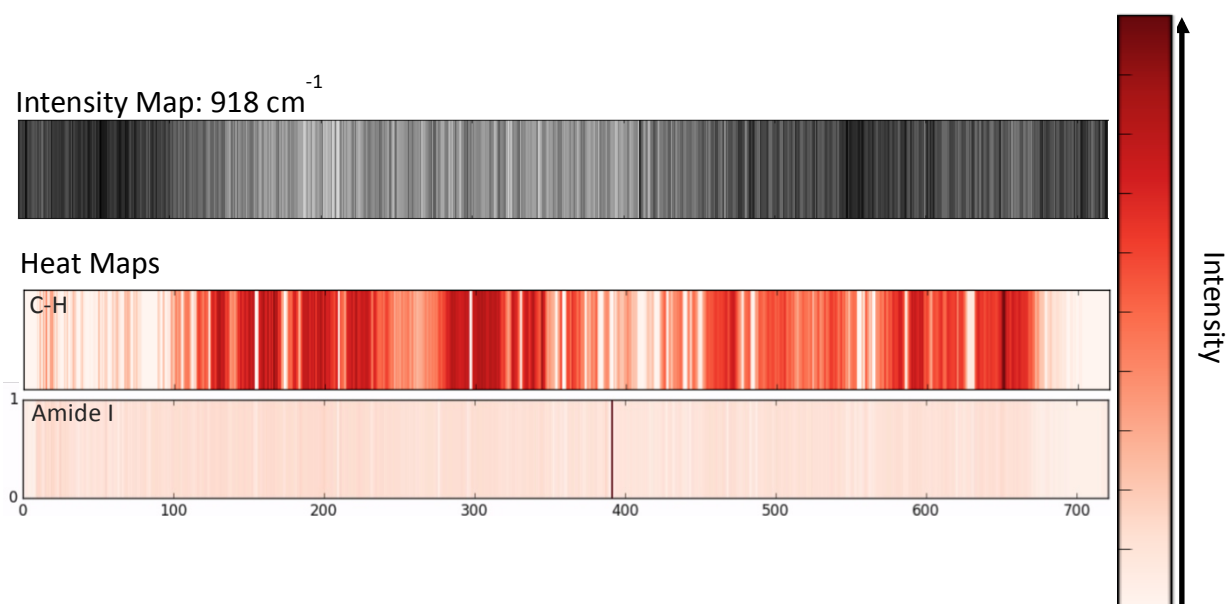


Figure 4.3. Example of line scan intensity map at 918  $\text{cm}^{-1}$  wavenumber taken from the tibial plateau sample. Heat maps showing concentrations of C-H and Amide I were generated by integrating over the 1452  $\text{cm}^{-1}$  and 1667  $\text{cm}^{-1}$  peaks respectively.



## **CHAPTER 5**

### **DISCUSSION**

The following discussion details the usage of the Python programming language, output file formats, software licensing, and continuing development of PySpectrum as a hyperspectral analysis software.

#### **4.1 Python as Programming Language**

Though the project initially started as a plugin for ImageJ, Java was found to be an unsuitable language for the targeted functionalities. Java is losing popularity for scientific computing because it is difficult to package for distribution. Additionally, Linux and Mac OS X operating systems both include Python, but not Java, installed by default. With respect to specifics for this project, the Java type system was found to be extremely inflexible when working with a multivariate dataset. It is also less efficient than Python for workflows including image processing, such as those used when creating TIFF images. Python libraries significantly increase runtime speed in contrast to Java. For example, NumPy is about as fast as using C, which is faster than Java [14, 15].

Therefore, PySpectrum is a stand-alone executable written entirely in the Python programming language. Python was chosen due to its conciseness and readability, useful data structures such as ordered dictionaries, ease of packaging and distribution, and extensive free and open source scientific libraries. Libraries used in PySpectrum include matplotlib, PIL, SciPy, and NumPy [16, 17, 18, 19]. These libraries allow for straightforward manipulation and processing of data and increase the efficiency of the program. They are used in the workflow of PySpectrum to convert inputted columns of numerical text to two-dimensional arrays to images.

Python also supports the use of Qt, a toolkit for graphical user interface development [20]. Toolbars, sliders, and other control elements used to interact with the data were generated using Qt.

## **4.2 File Formats**

The Tagged Image File Format (TIFF) was chosen for the intensity map stack. It is a widely used, universally supported, and extremely flexible image format that uses lossless compression. TIFF supports 32-bit floating points per pixel when displaying grayscale value, which is ideal for preserving the resolution of data taken from the Renishaw Raman microscope. Generally with other image formats, support is limited to an 8-bit grayscale image (values 0 to 255), 24-bit color image (8-bit red, green, and blue), or 48-bit color image (16-bits per channel). In the case of TIFF, the 32-bit floating point values can be directly inputted from the original text data into Python Image Library's Image module in array format [17].

Heat maps are generated in Portable Network Graphics (PNG) format for compatibility with the matplotlib Python library [16], which allows for a customizable layout, axes, color map, and pixel interpolation. The heat maps are in color to better highlight differences in intensity. Bilinear interpolation is used to smooth out the map and allow for easier visual interpretation. A color bar is created alongside the heat map to show how each color corresponds to a numerical value.

## **4.3 Open Source Software**

PySpectrum is an open source software provided under the MIT license which allows for free of charge commercial and private use, distribution, and modification. The source code is available in full on the project-hosting site GitHub [10], one of the largest source code repositories. Users are free to download and alter the code to customize it to their individual

needs, add features, or improve existing features. Therefore, PySpectrum can continue to develop and evolve collaboratively.

In contrast, commercial closed source software can inhibit progress in research because of its high cost and lack of tailorability. Proprietary software is developed with a specific set of tools that cannot be modified because the source code cannot be inspected. Oftentimes, the software is specific to one operating system. If the license limits redistribution or is extremely costly, users may only have access to such software on campus computers. All of these factors markedly limit the ability of students and researchers to use the software. Accessibility and wide distribution of essential analytical tools to researchers is more easily achieved with free of charge, open source, collaborative software.

#### **4.4 Continuing Development**

Features that will be added in the future include improvements to background subtraction, principle component analysis, and component overlay images. As discussed in Section 2.4 of Chapter 2, background subtraction is currently supported only for the displayed spectrum, calculating a baseline based on the individual plot. In the future, baseline subtraction will be applied in batch, meaning that it will occur across all spectra in the dataset. Depending on the length of computation time, a baseline may be discretely calculated for each spectrum when the *Baseline Subtraction* button is clicked. Otherwise, it will be based on a particular input set by the user. Additionally, as a result of collecting data on hydrated biological tissue, the water peak (Seen in Figure 4.1 at  $3400\text{ cm}^{-1}$ ) is extremely prevalent and negatively influences the quality of peak analysis. Because the water peak is so large, it dominates the spectrum and makes it difficult to view smaller, more relevant peaks present in lower wavenumber ranges without significantly zooming in on the plot. An approach proposed to solve this issue involves a user

interface similar to the existing peak selection tool where the user selects a point on either side of the water peak to subtract it from the dataset.

In many studies using hyperspectral datasets, principle component analysis is performed to discover which components account for the most variability in the data or most influence the dataset. Figure 1.2c displays the advantages of using such a method by visually identifying three main components of the HeLa cell. PySpectrum will integrate the capability to carry out PCA on the user inputted data, as well as composing intensity maps to highlight these components within the sample.

Once PCA is integrated, a feature will be added to allow the user to overlay components onto the same image. The overlaying feature will be applicable to both PCA components (Figure 1.2c) and components generated using user-selected wavenumber ranges (Figure 1.3d). Mapping several components onto the same image will allow for easier visual determination of their relative locations and distributions within the sample.

## **CHAPTER 6**

### **CONCLUSIONS**

PySpectrum is an application made primarily as a tool to facilitate Raman hyperspectral analysis. It is free and open source, which lends itself to collaborative development amongst users who can improve or add useful features to the existing program by downloading the source code from the hosting site GitHub. Researchers often develop in-house tools such as PySpectrum to aid their own individual research, and these can be shared in the same manner. This is especially important in academia where there may be constraints on resources. In other cases, researchers have differing approaches to studying data that may not be feasible with closed source, proprietary software tools.

With PySpectrum software, researchers are able to process large quantities of data and transform the data into visual interpretations. This was exemplified using Raman spectroscopy data of a section of meniscal enthesis, showing how intensities of individual components in the data are extracted and displayed in heat maps. This type of characterization and further comprehensive analysis of the enthesis will be beneficial in producing successful and long-lasting meniscal implants. The development of PySpectrum was driven in an effort to support research endeavors such as these.

## REFERENCES

- [1] A. Diaspro, G. Chirico, C. Usai, P. Ramoino, and J. Dorucki. "Photobleaching." In *Handbook of Biological Confocal Microscopy*, 690-99. 3rd ed. Springer Science & Business Media, (2010).
- [2] L. Qingli, X. He, Y. Wang, H. Liu, D. Xu, and F. Guo. "Review of Spectral Imaging Technology in Biomedical Engineering: Achievements and Challenges." *J. Biomed. Opt. Journal of Biomedical Optics* 18, no. 10 (2013): 100901. doi:10.1117/1.jbo.18.10.100901.
- [3] M. Miloš, T. Chernenko, M. J. Romeo, B. Bird, C. Matthäus, and M. Diem. "Label-free Imaging of Human Cells: Algorithms for Image Reconstruction of Raman Hyperspectral Datasets." *Analyst* 135, no. 8 (2010): 2002-13. doi:10.1039/c0an00042f.
- [4] H. Keisaku, K. Fujita, N. I. Smith, M. Kobayashi, Y. Inouye, and S. Kawata. "Raman Microscopy for Dynamic Molecular Imaging of Living Cells." *J. Biomed. Opt. Journal of Biomedical Optics* 13, no. 4 (2008): 044027. doi:10.1117/1.2952192.
- [5] B. Nils, G. Rago, M. Bosma, G. Eijkel, A. Gemmink, P. Leproux, G. Huss, P. Schrauwen, et al. "Chemical Imaging of Lipid Droplets in Muscle Tissues Using Hyperspectral Coherent Raman Microscopy." *Histochemistry and Cell Biology* 141, no. 3 (2013): 263-73. doi:10.1007/s00418-013-1161-2.
- [6] P. Cueva, R. Hovden, J. A. Mundy, H. I. Xin, D. A. Muller. "New Approaches to Data Processing for Atomic Resolution EELS." *Microscopy and Microanalysis* 18, no. S2 (2012): 970-71. doi:10.1017/s1431927612006708.
- [7] C. Beleites and V. Sergo. "hyperSpec: a package to handle hyperspectral data sets in R." <http://hyperspec.r-forge.r-project.org> (2015).
- [8] L. O. Jimenez, M. Velez-Reyes, S. Hunt, D. Kaeli, E. Arzuaga-Cruz, S. Rossario-Torres, A. Castrodad. "Hyperspectral Image Analysis Toolbox." <https://www.censsis.neu.edu/software/hyperspectral/hyperspectral.html> (2016).
- [9] F. de la Peña, P. Burdet, V. T. Fauske, T. Ostasevicius, M. Sarahan, M. Nord, J. Taillon, et al. "Hyperspy: HyperSpy 0.8.4." (2016). Zenodo. doi:10.5281/zenodo.46897.
- [10] D. Lieber. "PySpectrum." GitHub repository. <https://github.com/danushkana/pyspectrum> (2016).
- [11] J. R. Kunitake. [Breast Tissue Area Scan], Unpublished raw data. (2015).
- [12] A. J. Boys. [Medial meniscal enthesis Raman area scan], Unpublished raw data. (2016).
- [13] K. Chan. [Tibial plateau enthesis Raman line scan], Unpublished raw data. (2016).

- [14] F. Perez, B. E. Granger, and J. D. Hunter. "Python: An Ecosystem for Scientific Computing." *Computing in Science & Engineering* 13, no. 2 (2011): 13-21. doi:10.1109/MCSE.2010.119
- [15] J. M. Bull, L. A. Smith, L. Pottage, and R. Freeman. "Benchmarking Java against C and Fortran for scientific applications." *Proceedings of the 2001 joint ACM-ISCOPE conference on Java Grande - JGI '01*. (2001). doi:10.1145/376656.376823.
- [16] M. Droettboom, J. Hunter, T. A. Caswell, E. Firing, J. H. Nielsen, P. Elson, B. Root, et al. 2016. "Matplotlib: Matplotlib v1.5.1." Zenodo. doi:10.5281/zenodo.44579.
- [17] E. Soroos, A. Clark, A. Murray, A. Karpinsky, C. Gohlke, B. Crowell, et al. (2016). "Pillow: 3.1.0." Zenodo. doi:10.5281/zenodo.44297.
- [18] E. Jones, E. Oliphant, P. Peterson, et al. "SciPy: Open Source Scientific Tools for Python." <http://www.scipy.org> (2001).
- [19] S. van der Walt, C. Colbert, and G. Varoquaux. "The NumPy Array: A Structure for Efficient Numerical Computation." *Computing in Science & Engineering* 13, (2011): 22-30. doi:10.1109/MCSE.2011.37.
- [20] "PySide: Python for Qt." <https://pyside.org> (2009).