2019-03-15

Computational Regiospecific Analysis of Brain Lipidomic Profiles

Austin Ahlstrom
Brigham Young University - Provo

COMPUTATIONAL REGIOSPECIFIC ANALYSIS OF BRAIN LIPIDOMIC PROFILES

by

Austin Ahlstrom

Submitted to Brigham Young University in partial fulfillment

of graduation requirements for University Honors

Department of Chemistry and Biochemistry

Brigham Young University

April 2019

Advisor: John C. Price

Faculty Reader: Samuel H. Payne

Honors Coordinator: Paul Jenkins

ABSTRACT

COMPUTATIONAL REGIOSPECIFIC ANALYSIS OF BRAIN LIPIDOMIC PROFILES

Austin Ahlstrom

Mathematics Department

Bachelor of Science

Mass spectrometry provides an extensive data set that can prove unwieldy for practical analytical purposes. Applying programming and machine learning methods to automate region analysis in DESI mass spectrometry of mouse brain tissue can help direct and refine such an otherwise unusable data set. The results carry promise of faster, more reliable analysis of this type, and yield interesting insights into molecular characteristics of regions of interest within these brain samples. These results have significant implications in continued investigation of molecular processes in the brain, along with other aspects of mass spectrometry, collective analysis of biological molecules (i.e. omics), and biology in general.

ACKNOWLEDGMENTS

Special thanks to Dr. Price for patiently helping me understand scientific concepts that would have otherwise been well beyond my comprehension, making this thesis possible. Thanks to Dr. Payne and Dr. Jenkins for diligent and insightful comments helping me refine and improve my writing, and to my father especially—along with other family and friends—for helping clarify my writing and improve flow. Thanks to Dr. Tyler Jarvis for advice in the computational elements of my research.

TABLE OF CONTENTS

LIST OF FIGURES

Introduction

Technological advancements have been crucial for scientific understanding of human biology, and have potential to do so much more. Consider the study of neurodegeneration (including Alzheimer's and Parkinson's disease, etc.), a condition that afflicts more than twenty million people worldwide, and is only increasing in prevalence.[1] Despite significant scientific study over the past several years, the molecular processes that underlie these conditions are still not completely understood.

This is not to say that progress has not been made. Scientific advancements in the study of neurodegeneration are numerous, like research performed by Dawson and Dawson[2], Rubinsztein[3], and Glass, et al.[4] Discoveries fueled by improvements in scientific technology have dramatically expanded the knowledge of the scientific community regarding neurodegenerative conditions.

However, the results of investigations such as these have been limited in terms of substantive treatment developments for neurodegeneration. This is true not only of neurodegeneration, but also of many similar areas wherein scientific comprehension of base-level processes is incomplete. Further expansion and refinement of techniques and technologies must be explored to help accelerate such developments. This paper will introduce such new techniques in the area of mass spectrometry toward the aim of distilling massive data sets into a usable form, adding to modern scientific understanding.

Mass Spectrometry

Mass spectrometry has received considerable attention within the scientific community. The general idea of mass spectrometry has been in use for more than a century, and methods applying these principles have been developed continuously. Through the relatively recent efforts of researchers including Fenn, et al.[5] and Takáts, et al.[6], effective capabilities to apply mass spectrometry to biological macromolecules now exist. The result of these achievements is a previously unprecedented capacity to examine organs such as the brain on a molecular level.

Among the advantages of mass spectrometry is the amount of data it yields. Given a thin slice of tissue, a mass spectrometer is able to divide the specimen into tiny stripes, then to sample each of these stripes at regular intervals in order to divide the specimen into a two-dimensional grid. Each subdivision can then be sampled, using ionization and

---

[1] Mayeux, R. Epidemiology of Neurodegeneration. *Annual Review of Neuroscience* **26,** 801-104 (2003).

[2] Dawson, T.M. & Dawson,V.L. Molecular Pathways of Neurodegeneration in Parkinson's Disease. *Science* **31,** 819-822 (2003).

[3] Rubinsztein, D.C. The roles of intracellular protein-degradation pathways in neurodegeneration. *Nature* **443,** 780-786 (2006).

[4] Glass, C.K., Saijo, K., Winner, B., Marchetto, M.C. & Gage, F.H. Mechanisms Underlying Inflammation in Neurodegeneration. *Cell* **140,** 918-934 (2010).

[5] Fenn, J.B., Mann, M., Meng, C.K., Wong, S.F. & Whitehouse, C.M. Electrospray ionization for mass spectrometry of large biomolecules. *Science* **246,** 64-71 (1989).

[6] Takáts, Z., Wiseman, J.M., Gologan, B. & Cooks, R.G. Mass Spectrometry Sampling Under Ambient Conditions with Desorption Electrospray Ionization. *Science* **306,** 471-473 (2004).

rate of movement through a vacuum to determine individual abundances of molecules separated by their mass and charge. Within a relatively short period of time, this generates millions of numeric observations from the single specimen, corresponding to different tissue regions as well as different inferred molecule types, sorted by mass-to-charge ratio (m/z).

In addition, mass spectrometry permits a crucial spatial component for tissue analysis. Whereas methods antedating these breakthroughs would often blend, for example, entire organs together, and thus be incapable of analyzing molecular composition of anything more specific than that individual organ, modern mass spectrometric methods allow for analysis of highly specific regions. In organs such as the brain, for example, there is scientific consensus that different regions of the brain perform different—albeit integrally interconnected—functions, and accordingly are characterized by completely different molecular compositions (see Figure 1). Thus, mass spectrometry has played a key role in contemporary studies of the full molecular level of biological systems, like in proteomics, lipidomics, and other types of omics. This grants comprehension of an ever-increasing extent of regiospecific molecular phenomena in the brain.

The Problem

The abundance and spatial significance of these data, however, are accompanied by inherent challenges. The sheer quantity of data produced from mass spectrometry of an individual tissue sample can be intractable. In essence, the result of a mass spectrometry scan is a three-dimensional tensor composed of millions upon millions of numeric entries, a scale that quickly prohibits any practical holistic manual analysis.

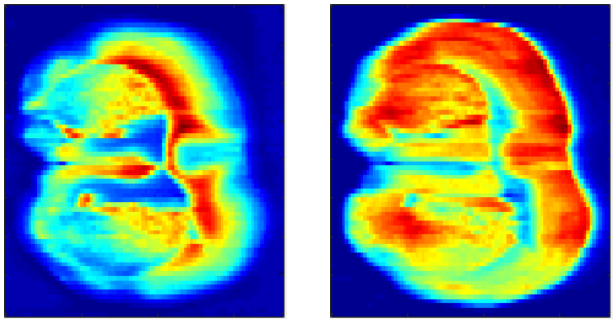


Figure 1: Two images produced from mass spectrometry data of the same brain sample, each showing the abundance (by ion count intensity) of a different molecular mass-to-charge ratio value. Each image, then, shows abundance of a different type of molecule; note the differences in abundance between sections of the brain for each type.

Of course, processes still exist for deriving pertinent insights from this large data set. In many cases, however, such methods are difficult, imprecise, and/or time-consuming. Furthermore, these methods often run the risk of overlooking salient patterns in the data that may prove important in furthering scientific comprehension of molecular processes.

As a relevant aside, large-scale data sets like this are nothing new to the modern world, where data analysis of even larger data sets than this has become a key aspect of operations for many organizations. Developments in the fields of mathematics and technology, and in particular of machine learning, have been implemented as a method to draw relevant conclusions from massive troves of data. However, applications of these modern methods to fields like mass spectrometry are not yet as ubiquitous. The

procedure and methodology discuss in this paper merges these mathematically driven techniques with the mass spectrometry field.

Addressing the Need

The core purpose of the activities documented in this paper has been to apply technological methods to mass spectrometric data in order to produce a faster and easier process of making sense of and managing these sizable data sets. The goal in doing so is to eliminate potential inefficiencies and oversights, while adding statistical and mathematical rigor to current analytic methods. The intent of this is to stimulate new discoveries with far-reaching repercussions in relevant scientific fields. Updated technological methods such as these increase the rate at which scientific study can be performed.

Turning to the specifics of my work, by identifying specific shortcomings and difficulties in the mass spectrometry analysis process of my lab, I significantly sped up our existing workflow. To do so, I programmed a script that performed a file conversion in a fraction of the time that had previously been required, using inputs that we had previously lacked the capability to analyze. This programming-driven methodology paves the way for a more meaningful analysis of the massive spectrometry data.

Furthermore, I used various methods arising from contemporary applications of mathematics and machine learning to perform in-depth analysis on our mass spectrometry data. Region of interest-based analysis of brain samples is a key area in the studies conducted in our biochemistry lab, and the results of my research, in relation to these studies, are intriguing. In particular, these results have clear implications in aspects of: 1. automated detection of selected regions of interest within brain sample data, 2. determining density of specific molecules as a trait of specific regions in the brain, and 3. assessment of different brain regionification schemes and resultant increased understanding of spatial properties within brain samples. Taken together, these results all contribute to an improved understanding of the molecular processes of brain samples, adding important observations to the existing body of scientific discovery.

Current Data

The processes and paradigms described here will be specifically described in application to desorption electrospray ionization (DESI) mass spectrometry, though they will have generalized application beyond this narrow area. This type of spectrometry is effective in determining abundance of lipids within tissue samples. The tissue on which these samples have been performed comes from brains of mice. Our biochemistry lab has focused on DESI mass spectrometry of mouse brains for years, and studies and publications from these efforts to date have yielded fascinating information. Some of the research has been focused on

Brain Scan with Five Regions Labeled

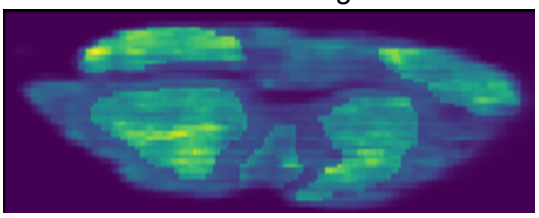


Figure 2: A brain sample scan, shown with five regions labeled: the left and right cortices, left and right caudoputamens, and mesencephalon. These labels were drawn by hand by a researcher in our lab.

identifying and comparing specific regions of interest within brains; Figure 2 shows a brain scan with such regions highlighted. Some of these investigations involved the use of brains from mice treated with deuterated water, resulting in slight differences in the m/z values of high-intensity molecules in these scans because of the greater abundance of heavy molecular isotopes.

These investigations were generally performed using a Bruker mass spectrometer. Based on this, Brigham Young University (BYU) faculty produced a tool in the MATLAB programming language capable of visualizing the in-sample intensity of user-specified m/z values.[7] This tool provided a graphical user interface for researchers to perform computer-assisted analysis of molecular abundances within regions of the brain, and effectively facilitated enlightening research into various aspects of sample brains. A screenshot of this tool in action is shown in Figure 3.

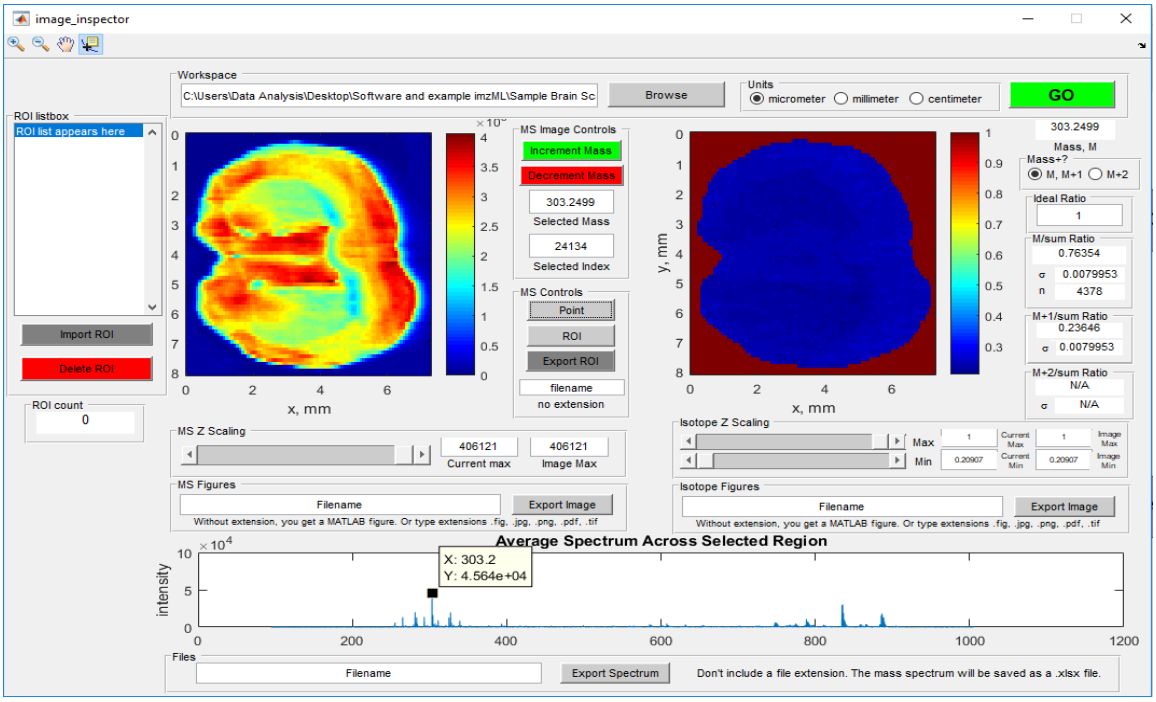


Figure 3: A screenshot of Image Inspector, a MATLAB tool developed by BYU faculty to visualize mass spectrometric data.

Recently, however, our lab began to use an Agilent mass spectrometer more often than using the Bruker machine. The file conversion process that had been used to prepare the Bruker output binary files for use in the BYU MATLAB tool, however, involved the use of a couple of intermediary third-party file conversion programs, and these were incompatible with the output data of the Agilent spectrometer. Thus, in spite of having

[7] Carson, R.H., Lewis, C.R., Erickson, M.N., Zagieboylo, A.B., Naylor, B.C., Li, K.W., Farnsworth, P.B., & Price, J.C. Imaging regiospecific lipid turnover in mouse brain with desorption electrospray ionization mass spectrometry. *Journal of Lipid Research*, **58,** 1884-1892 (2017).

ready-to-use Agilent spectrometric data, researchers in the lab were unable to use this tool for the desired analysis.

Analysis Methods

This was an area where it was clear that new computational strategies would go a long way toward increasing overall effectiveness. We had access to an API for using the C++ programming language to access data from the Agilent output files, and I used this API to program a script for the mass spectrometry file conversion. The result was a program that allowed us to use the MATLAB tool on data that had previously been inaccessible. Also, by accessing the data directly through the Agilent API rather than using third-party conversion programs, the time required to perform the necessary file conversion was drastically decreased.

Though the programming of this script did not constitute analysis in and of itself, it did provide a larger set of brain mass spectrometry scans on which to perform research. Having example files from two spectrometers helped achieve analysis that was not restricted to a single spectrometer's formatting. With this increased availability of data, I began investigating computational analysis methods on the data in question.

I used C++ for the conversion part of the process because it was one of the few languages compatible with the Agilent tool; for other parts of the analysis process, I worked almost exclusively in the Python programming language. This choice was made not only because of Python's increasing popularity and open-source development principles, but also because of a broad range of machine learning tools already available from the language. Python also facilitated significant speed in programming and running the type of algorithms being implemented.

In general, the analysis I have performed involves a pixel-by-pixel approach to the mass spectrometric data. The values returned from the mass spectrometer were in the form of a two-dimensional grid subdividing the tissue sample provided. In the case of the Agilent machine, a single pixel in this grid was associated with just over 100,000 numeric values corresponding to different inferred molecular m/z values. In terms of the mathematical analysis, this could be seen as each pixel being a point of a dimensionality exceeding 100,000.

For many algorithms, this sort of dimensionality would practically inhibit efficient analysis. Aside from that, many of the individual values did not seem to contribute much additional information. Intensity values generally followed a peak pattern, with neighboring values symmetrically distributed around local maxima. By paring the data set down to these centroids, I was able to reduce the dimensionality of the problem without losing significant degrees of molecular information. Figure 4 shows one such peak, with average intensities for each of the five labeled regions. Figure 6 shows the selected peaks from this method.

Although this process allowed simplified analysis of complex data, there were several challenges encountered. Figure 5 shows peaks averaged over the same five regions at a different m/z value, and it can be seen that the local maximum is attained at two different m/z values in different regions, though the actual difference between the average intensity at either m/z value across the regions was minimal. Because of instances like this, it seemed appropriate to select m/z values based on an average of all in-brain pixels, reducing some of the variability while not impacting the intensity values too significantly.

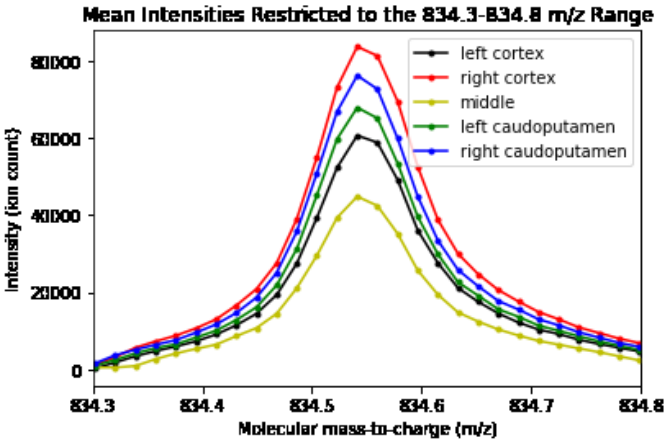


Figure 4: The primary phosphatidylserine peak—the peak that achieved the maximum mean intensity within the sample—in this given brain sample, averaged within each of the labeled regions. Note how within all regions, the intensity values are distributed in a Gaussian-like curve around the main peak.

To restrict the data as needed, it was necessary to determine which pixels were contained within the actual brain tissue, as opposed to pixels on the edges of the scan. That challenge proved relatively easily manageable. Compared to the regions of pixels within the brain, the molecular disparities between pixels inside and outside the brain was stark. A single computationally identified m/z aspect was generally sufficient to differentiate, given a specific cutoff, between brain and non-brain pixels.

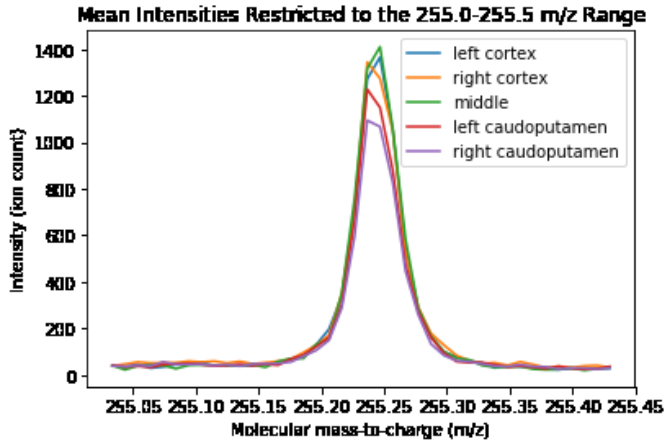


Figure 5: A lower intensity peak, showing local maxima attained at two different points in different regions. This occurred frequently with lower intensity peaks, where statistical variation resulted in this type of discrepancies. This informed my decision to use the mean of all in-brain pixels to select m/z values for centroids.

Once the data had been preprocessed in this way, I was able to apply numerous machine learning methods to the data. Each pixel became a distinct observation, a data point usable in training a machine learning algorithm. In general, machine learning falls into two classes: supervised learning, where the input data are given with labels and the program is trained to apply these labels to unlabeled data, and unsupervised label, where the data are given without any labels and the program is tasked with finding useful categorizations of the data without user-provided labeling.

Over the course of this project, I used both types of methods to analyze the DESI mass spectrometric data more closely and observe what conclusions could be drawn.

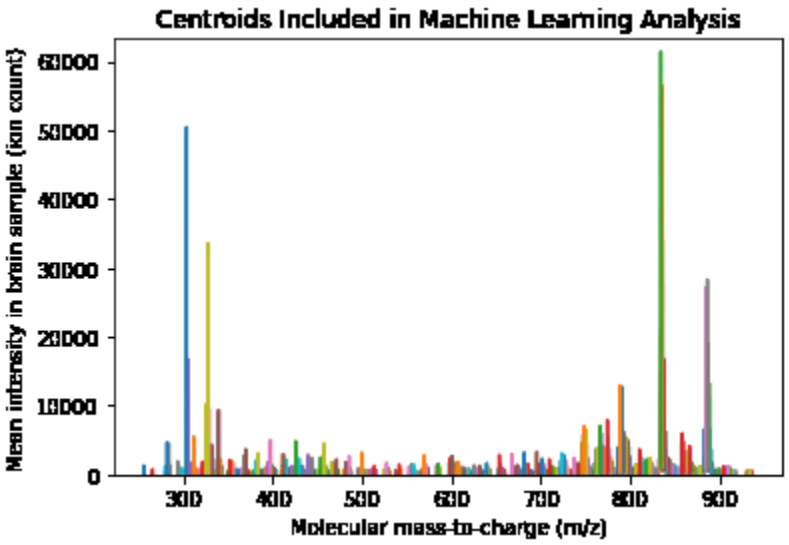


Figure 6: The centroids used in machine learning investigation of the data. The spikes shown are local maxima in intensity, graphed at the corresponding m/z value.

Supervised learning methods were useful for analyzing existing labeled brain scans. These scans had certain regions of interests demarcated (see Figure 2), providing labels to be used for supervised algorithms. These algorithms, in turn, often had associated methods for determining feature importance, producing quantitative assessments of which molecules might play key roles in different regions, since each feature corresponded to a single m/z value. This work also generated interesting observations regarding automated identification of the same regions within unlabeled scans.

A different utility was observed with unsupervised algorithms. These permitted a comparison of the region division scheme we used to a scheme generated based solely on spectral data properties. These observations allowed us to further expand our understanding and intuition regarding the brain samples in question.

Outcomes—C++ File Conversion

The C++ script for converting Agilent binary files to formats interpretable by the BYU MATLAB tool became a useful contribution to the lab (see Figure 7). The original process involved converting the files in question to a .mat MATLAB matrix file, which I was able to imitate in a more direct process with the C++ API. I also created a similar script with the ability to convert files into a standard .bin format, anticipating our lab's efforts to move away from MATLAB in

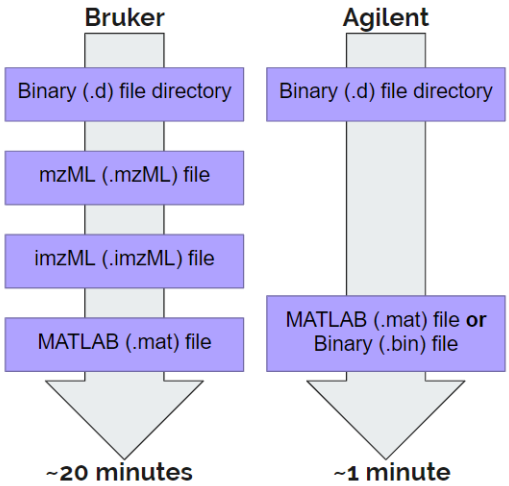


Figure 7: The conversion processes in place for Agilent and Bruker mass spectrometry scans, showing the file conversions involved and the amount of time generally required.

favor of more open-source, less expensive alternatives, such as Python, with the intention of increasing accessibility for other users. The script is given in Appendix A.

Because of the Component Object Model (COM) framework in which the Agilent data access functions were written, it made the coding process significantly simpler to use Microsoft Visual Studio's ATL capabilities in this script. This dependency runs counter to our lab's effort to make code more open source and accessible, but has so far not caused any significant issues. Since using these tools helps with code simplicity and readability, we are currently accepting the tradeoff.

Unsupervised Machine Learning

There are a variety of unsupervised machine learning algorithms developed with the express purpose of "clustering" similar data points into categories. I attempted using a variety of such algorithms, in order to see what these algorithms might reveal regarding "natural" regions within the brain based solely on mass spectrometry data. Thus, these algorithms were given intensities of hundreds of centroids, but not the location of the pixel on the brain. The results of this analysis are given in Figure 8, showing a variety of results for the problem of identifying regions.

These clustering algorithms use various paradigms to perform group categorization. Affinity propagation works by identifying several data points, labeled as exemplars, and grouping according to similarity to these.[8] Spectral clustering uses the eigensystem of the data points to effectively perform dimension reduction and find similar data points in the projected data;[9] this method is related to the partitioning algorithm used in density-based spatial clustering of applications with noise (DBSCAN), which adds a criterion analyzing the density of data in groups in order to assign categories and reduce sensitivity to noise.[10] Balanced iterative reducing and clustering using hierarchies (BIRCH) also uses a subdivision scheme to reduce sensitivity to noise, but does so while implementing a hierarchical algorithm, as is used in agglomerative clustering, to label points close to each other as such using a distance metric.[11] The mean shift algorithm works by finding the maxima of a density function of data points and using it to categorize them.[12]

---

[8] Dueck, D. & Frey, B. Non-metric affinity propagation for unsupervised image categorization. *Proceedings: IEEE International Conference on Computer Vision* **11,** 1-8 (2007).

[9] Pothen, A., Simon, H.D & Liou, K. Partitioning Sparse Matrices with Eigenvectors of Graphs. *SIAM J. Matrix Anal. Appl.*, **11:3,** 430–452 (1989).

[10] Ester, M., Kriegel, H., Sander, J. & Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Proceedings: Int'l Conf. on Knowledge Discovery, Data Mining* **2,** 226-231 (1996).

[11] Zhang, T., Ramakrishnan, R. & Livny, M. BIRCH: an efficient data clustering method for very large databases. *Proceedings: ACM SIGMOD Int'l Conference on Management of Data* **22,** 103-114 (1996).

[12] Fukunaga, K. & Hostetler, L. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory* **21,** 32-40 (1975).

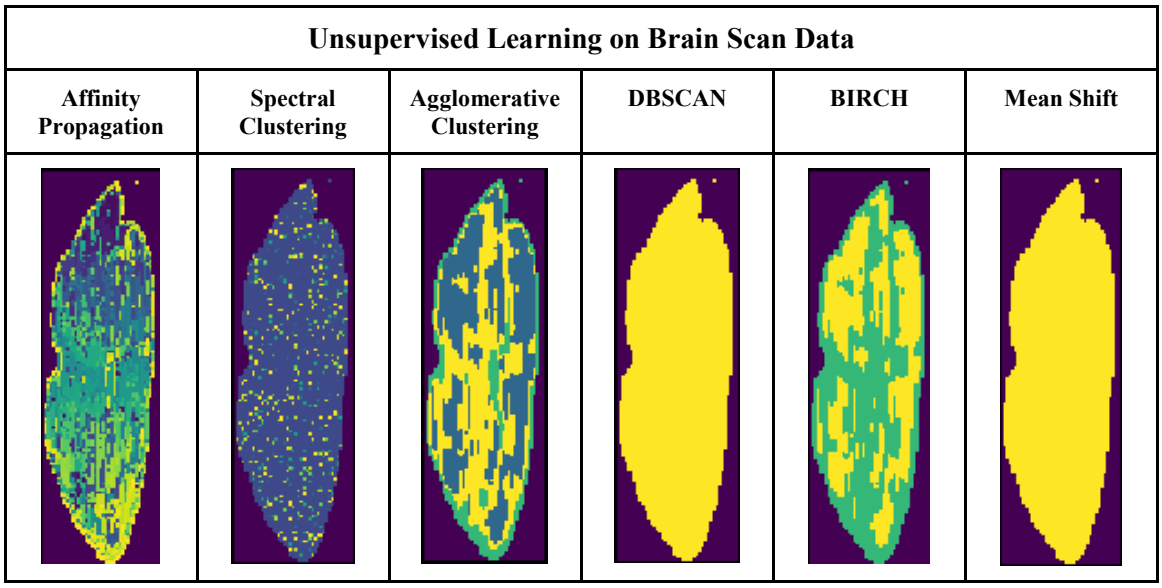


Figure 8: The results of a variety of unsupervised clustering algorithms applied to the same mass spectrometry sample.

It is interesting to note the variety of results derived from applying different algorithms to the data. In the case of the affinity propagation result, for example, the brain was split into over one hundred categories, too many to be reasonably used. On the other hand, the mean shift and DBSCAN algorithms determined that all in-brain pixels were too similar to split into any different categories. Out of these, the BIRCH and agglomerative clustering results are of particular interest. The similar patterns shown in these are relatively simple and show some bilateral symmetry, which should be expected from the brain. This inspires some confidence that the methods might be capturing real and interesting trends.

For the most part, nevertheless, these results failed to yield particularly salient features that seem to meaningfully reflect molecular compositions in the brain. In order to further investigate these procedures, some feature engineering of the data was needed. In particular, after running the same algorithms with mean-adjusted pixel values, wherein each intensity value was divided by its mean intensity within the brain, some different results were obtained. These results are displayed in Figure 9. A result of particular interest in this rendition is the one yielded by the BIRCH algorithm. Note the symmetry in this outcome, as well as the relative visibility of what appears to be the corpus callosum, along with other brain features.

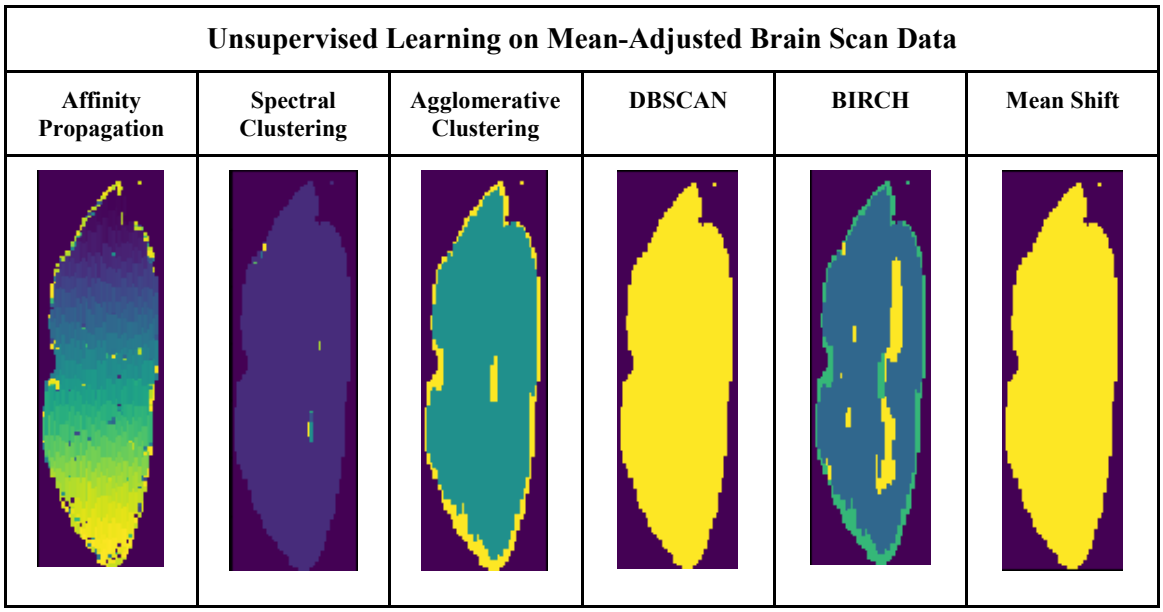


Figure 9: The results of the same unsupervised clustering algorithms applied to the mass spectrometry sample, but using mean-adjusted intensities.

Supervised Machine Learning

It was also enlightening to see the results of a variety of supervised machine learning algorithms that were applied to the same data set. For the purposes of this experimentation, the following four labels were applied: “cortex”, “caudoputamen”, “middle” (mesencephalon), and “brain - other”, based on the labels that had been manually applied to the mass spectrometry scans. This labeling permitted investigation into different assessments of feature importance for their labeling, with implications in learning defining molecular characteristics of the regions of interest in question, as well as the potential to perform future labelings computationally and automatically.

One common algorithm for such categorization problems is multinomial logistic regression, also known as softmax regression. This type of regression works by determining log-likelihood of a linear predictor function with some weighted combination of the features to be categorized, and is a generalization of logistic regression, a fitting scheme that uses a derivation of the logistic function to categorize data into

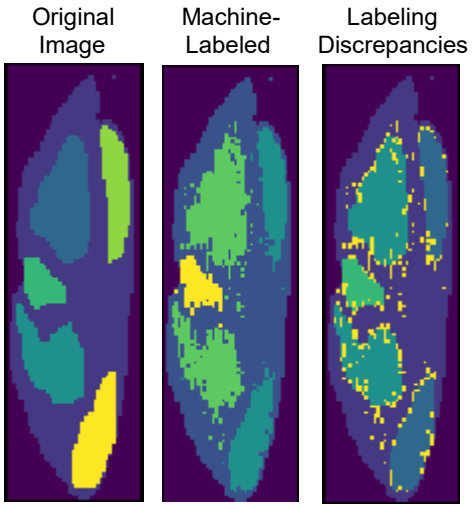


Figure 10: From left to right: 1) The original labeling provided, 2) the labeling generated by the softmax algorithm, and 3) comparing the labelings, with discrepancies highlighted in yellow.

two classes at a given cutoff.[13] Figure 10 shows the results of this algorithm, run on a 30-70 train-test split of the original data. The resulting labeling is just under 90% accuracy, appearing as shown in the figure.

The softmax algorithm also provides a framework for extrapolating feature importance, in terms of identification of each of the categories assigned to it. In the context of this problem, this indicates the m/z values corresponding to the intensities that the algorithm determined were most useful in identifying each type of region. The results of this are shown in Appendix B, along with comparisons of results from other methods. In most of the images shown, some of the regions of interest from the labeling are fairly visible. This provides an interesting starting point for investigations of lipids abundant in these regions.

This algorithm, however, is not the only algorithm used in modern classification problems, and in fact, many modern classification methods rely on random forest algorithms, instead. In order to gain a broader perspective on this task with regard to machine learning, I applied both a standard random forest and an XGBoost forest method to the relevant data. These algorithms are ensembles of decision trees, which iteratively take a single feature, determine a cutoff for that feature, and assign a category based on that. Random forests create a large random group of such decision trees, which tends to converge to a desirable categorization scheme, whereas XGBoost selects such trees based on defining an objective function and choosing trees that optimize it.[14] Each of these algorithms has an out-of-bag feature importance determination method, which was used to determine which m/z values were the most predictive in identifying regions. The graphical results of this can be seen in Appendix B. The XGBoost algorithm resulted in a higher accuracy 30-70 train-test result than the random forest algorithm, a visual representation of which is given in Figure 11.

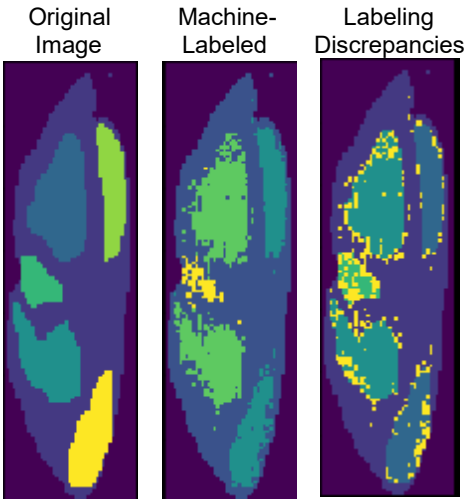


Figure 11: From left to right: 1) The original labeling provided, 2) the labeling generated by the XGBoost algorithm, and 3) comparing the labelings, with discrepancies highlighted in yellow.

Another algorithm often used for categorization problems like this is principal component analysis (PCA). This method selects principal axes in high-dimensional space to separate categories on, rather than retaining the mindset that each feature is an axis. The PCA algorithm is a dimension reduction algorithm used to project high-dimensional data into lower-

[13] Böhning, D. Multinomial logistic regression algorithm. *Ann. Inst. Statist. Math.* **44:1,** 197-200 (1992).
[14] XGBoost documentation: Introduction to Boosted Trees (2016). https://xgboost.readthedocs.io/en/latest/tutorials/model.html

dimension spaces[15], and even a single dimension of this algorithm achieved a 78% rate of variance prediction, with rapid decline in accuracy gains from added dimensions. The m/z values identified as most predictive with this method are also given in Appendix B.

As a final note, I also performed experimentation into using these algorithms to train an algorithm on multiple labeled spectrometry files, then using the trained models to identify labels on an input file without any same-file labels given, to see the potential for automatic region identification. Initial efforts into this yielded essentially no predictive power for identifying regions of interest; spectral differences between individual files appeared to be too great to use for training, after only simplistic data engineering. It seems likely that this type of supervised labeling will eventually be practicable, but may require further standardizations in the spectral data.

Conclusions

The results of the C++ script using Agilent's API to expedite the file conversion process were successful and promising. Being able to perform the same conversion and adjustment process in a shorter period of time with less manual effort is exciting. It serves as an indication of the extent that technology can be helpful in performing useful scientific tasks. Using methods that are as up-to-date as possible can pay long-term dividends in terms of time required for conducting research.

In investigating the unsupervised learning algorithms on the sample data, it may be said that the initial hypothesis was that methods like this can be used to effectively identify regions of interest like the ones we were investigating. The actual results, in many cases, differed from the base expectation; many identified non-contiguous regions or no regions at all. This seems to speak to the molecular complexity of the brain; though there are certainly noticeable divisions between sections, some of these may not be extremely well defined in terms of their component molecules, or at least in terms of their lipids.

Still, some promise was shown, particularly in the BIRCH analysis of mean-adjusted intensities. This may mean that clustering algorithms like BIRCH have the potential to yield regional divisions like those shown in the labeled data, perhaps given the right initial feature engineering. This would be significant, since the initial labeling of the data is still prone to human error; a more mathematically rigorous system of region identification would be useful.

Nevertheless, assuming that the human-labeled data are generally accurate with a few overall mistakes, the supervised approach seems to be a useful direction. The most interesting takeaways from these methods, however, did not come from automated identification of brain sample regions. It was more interesting to investigate the feature importance assigned by machine learning algorithms to the input data.

[15] Hotelling, H. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* **24,** 417–441 (1933).

By using the feature importance methods associated with each supervised algorithm, I was able to gain a significant list of m/z values that may be predictive in identifying the regions in which our lab has been interested. These preliminary results have given us a set of information to look into to investigate different molecular species, painting a more detailed picture of the molecular abundances and processes that occur in these brain regions. Our lab has begun investigating some of these results.

It may accurately be noted that the centroid analysis I used for this investigation was rather simplistic, simply identifying local maxima to identify peaks of spectra. The third, fourth, and fifth entries in the XGBoost column of Appendix B demonstrate the implications of this type of analysis; it seems that all three of these are likely the same molecule, but with different amounts of heavy isotopes of constituent elements. It may yield even more intriguing information to further hone this analysis by identifying particular molecular species, as well as the multiple peaks associated with different isotopic configurations of the same molecule. I did not include this level of preprocessing in my analysis, in favor of using a simpler method, as well as approaching the initial data with a more minimal amount of assumptions for initial experimentation. In the future, I may identify molecule packages across multiple associated peaks, and perhaps analyze ratios between peaks as well as overall intensity.

This paper has explored the idea that application of modern computational principles would make large spectrometry brain data sets more manageable and useful. This included both fast processing of mass spectrometer data and automated analysis of that data after preprocessing. The programming principles applied did, in fact, reduce and simplify data, though the predicted detectable patterns were not always as anticipated. Whereas many algorithmic models did not immediately reveal patterns from the simplified data, some did show some promise for using spectral data to identify and analyze regions of interest in a fast, automatic way. This should serve as a new and powerful tool to continue to identify and interpret molecular patterns in the brain.

REFERENCES

Böhning, D. Multinomial logistic regression algorithm. *Annals of the Institute of Statistical Mathematics* **44:1,** 197-200 (1992).

Carson, R.H., Lewis, C.R., Erickson, M.N., Zagieboylo, A.B., Naylor, B.C., Li, K.W., Farnsworth, P.B., & Price, J.C. Imaging regiospecific lipid turnover in mouse brain with desorption electrospray ionization mass spectrometry. *Journal of Lipid Research* **58,** 1884-1892 (2017).

Dueck, D. & Frey, B. Non-metric affinity propagation for unsupervised image categorization. *Proceedings: IEEE International Conference on Computer Vision* **11,** 1-8 (2007).

Dawson, T.M. & Dawson,V.L. Molecular Pathways of Neurodegeneration in Parkinson's Disease. *Science* **31,** 819-822 (2003).

Ester, M., Kriegel, H., Sander, J. & Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Proceedings: International Conference on Knowledge Discovery, Data Mining* **2,** 226-231 (1996).

Fenn, J.B., Mann, M., Meng, C.K., Wong, S.F. & Whitehouse, C.M. Electrospray ionization for mass spectrometry of large biomolecules. *Science* **246,** 64-71 (1989).

Fukunaga, K. & Hostetler, L. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory* **21,** 32-40 (1975).

Glass, C.K., Saijo, K., Winner, B., Marchetto, M.C. & Gage, F.H. Mechanisms Underlying Inflammation in Neurodegeneration. *Cell* **140,** 918-934 (2010).

Hotelling, H. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* **24,** 417–441 (1933).

Mayeux, R. Epidemiology of Neurodegeneration. *Annual Review of Neuroscience* **26,** 801-104 (2003).

Pothen, A., Simon, H.D & Liou, K. Partitioning Sparse Matrices with Eigenvectors of Graphs. *SIAM Journal on Matrix Analysis and Applications*, **11:3,** 430–452 (1989).

Rubinsztein, D.C. The roles of intracellular protein-degradation pathways in neurodegeneration. *Nature* **443,** 780-786 (2006).

Takáts, Z., Wiseman, J.M., Gologan, B. & Cooks, R.G. Mass Spectrometry Sampling Under Ambient Conditions with Desorption Electrospray Ionization. *Science* **306,** 471-473 (2004).

XGBoost documentation: Introduction to Boosted Trees (2016). https://xgboost.readthedocs.io/en/latest/tutorials/model.html

Zhang, T., Ramakrishnan, R. & Livny, M. BIRCH: an efficient data clustering method for very large databases. *Proceedings: ACM SIGMOD Int'l Conference on Management of Data* **22,** 103-114 (1996).

Appendix A: Agilent Data Conversion Code

The following is the script used to leverage Agilent's API and convert the data directly to the desired format. This particular iteration of the program formats the output as a generic .bin file, which was not originally supported in the Image Inspector MATLAB tool, though with a couple of tweaks, the MATLAB tool now reads these correctly. Other researchers are working on a Python port to support potential users who benefit from the use of an open-source language, and it is anticipated that this .bin output format will be more helpful for that.

This is a single sample of the code used for this project. Further code samples may be found in the GitHub repository at the following link: https://github.com/aahlstrom1.

Agilent .d to .bin converter:

```
#include <assert.h>
#include <atlbase.h>
#include <filesystem>
#include <iostream>
#include <shobjidl.h>
#include <string>
#include <vector>
#include <windows.h>

// These files contain the Agilent mass spectrometry data access functions.
#import "BaseCommon.tlb" raw_interfaces_only, no_namespace, named_guids
#import "BaseDataAccess.tlb" raw_interfaces_only, rename_namespace("BDA"), named_guids
#import "MassSpecDataReader.tlb" raw_interfaces_only, no_namespace, named_guids

namespace fs = std::experimental::filesystem;

std::vector<std::wstring> getFoldersFromUserSpecifiedDirectory();
CComBSTR* sortFoldersByNumberInPath(std::vector<std::wstring> list, int size);
float* generateAxisArray(int length, int pixelSize);

int main()
{
    CoInitialize(NULL);
    HRESULT hr = S_OK;

    int current = 0, pixelSizeX = 0, pixelSizeY = 0, xSize = 0, ySize = 0, zSize = 0;
    LONG lBound, uBound, count;
    float *matrix;
    double *zVals;
    const int FLOAT_SIZE = 4, MAX_FILE_NAME_SIZE = 200;
    bool matrixDefined = false;
    std::vector<float> v;
    char* filename = new char[MAX_FILE_NAME_SIZE];

    std::cout << "Enter desired output file name (without file type extension, " <<
            "e.g. 'file' will output to 'file.bin')" << std::endl;
    std::cin >> filename;
    std::strcat(filename, ".bin");
    std::cout << "Enter integer lengths for pixel width and pixel height." << std::endl;
    std::cout << "Then, select the directory to read spectral data from." << std::endl;
    std::cout << "Enter pixel width: ";
    std::cin >> pixelSizeX;
```

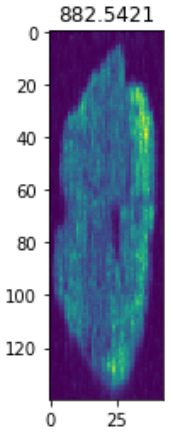
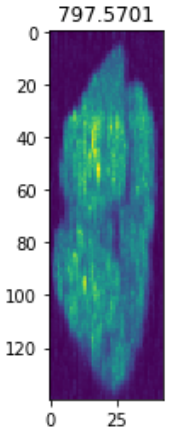
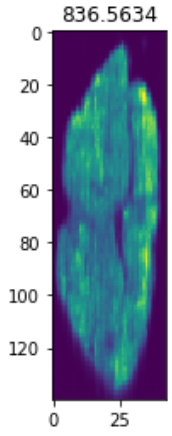
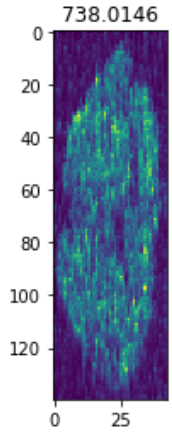
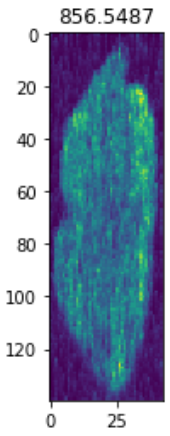
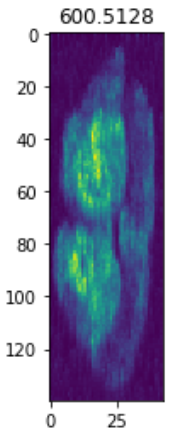
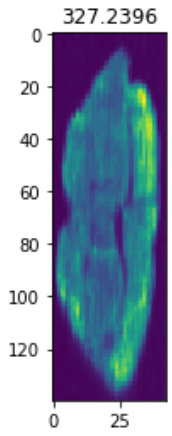
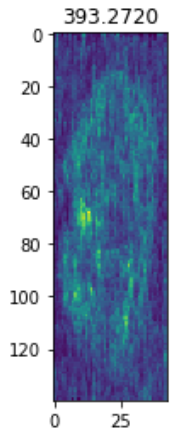
```
std::cout << "Enter pixel height: ";
std::cin >> pixelSizeY;

std::vector<std::wstring> spectrumFiles = getFoldersFromUserSpecifiedDirectory();
ySize = spectrumFiles.size();
CComBSTR *filePaths = sortFoldersByNumberInPath(spectrumFiles, ySize);

FILE* pFile;
pFile = fopen(filename, "wb");

for (int path = 0; path < ySize; path++) {
    std::cout << "Reading path " << path + 1 << " of " << ySize << "\r";
    CComPtr<IMsdrDataReader> pMSDataReader;
    hr = CoCreateInstance(CLSID_MassSpecDataReader, NULL, CLSCTX_INPROC_SERVER,
        IID_IMsdrDataReader, (void**)&pMSDataReader);
    assert(hr == S_OK);

    VARIANT_BOOL pRetVal = VARIANT_TRUE;
    hr = pMSDataReader->OpenDataFile(filePaths[path], &pRetVal);
    assert(hr == S_OK);

    CComPtr<BDA::IBDAChromData> pChromData;
    hr = pMSDataReader->GetTIC(&pChromData);
    assert(hr == S_OK);

    long dataPoints = 0;
    hr = pChromData->get_TotalDataPoints(&dataPoints);
    if (xSize == 0)
        xSize = dataPoints;
    assert(hr == S_OK);

    for (int scan = 0; scan < xSize; scan++) {

        v.clear();

        if (scan < dataPoints) {
            CComPtr<BDA::IBDASpecFilter> specFilter;

            hr = CoCreateInstance(BDA::CLSID_BDAChromFilter, NULL,
                CLSCTX_INPROC_SERVER,
                BDA::IID_IBDAChromFilter,
                (void**)&specFilter);
            assert(hr == S_OK);

            CComPtr<BDA::IBDASpecData> spectrum;
            hr = pMSDataReader->GetSpectrum_6(scan, NULL, NULL, &spectrum);

            if (hr == S_OK) {
                float* yArray = NULL;
                SAFEARRAY *safeYArray = NULL;
                hr = spectrum->get_YArray(&safeYArray);
                assert(hr == S_OK);
                SafeArrayGetLBound(safeYArray, 1, &lBound);
                SafeArrayGetUBound(safeYArray, 1, &uBound);
                SafeArrayAccessData(safeYArray, reinterpret_cast<void**>(&yArray));

                v.assign(yArray, yArray + uBound - lBound + 1);
                SafeArrayUnaccessData(safeYArray);

                if (!matrixDefined) {
                    assert(scan == 0);
                    zSize = v.size();
                    ySize -= path;
                    matrix = new float[ySize * xSize * zSize];
                    matrixDefined = true;
```

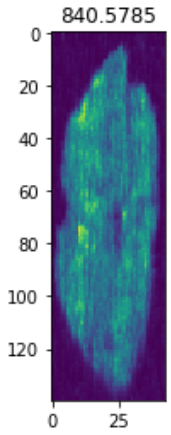
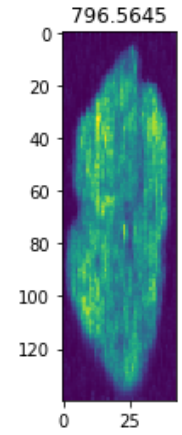
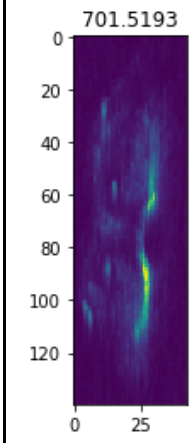
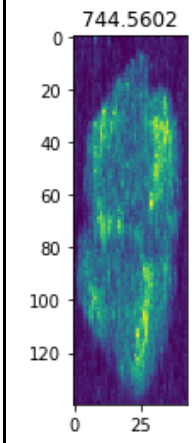
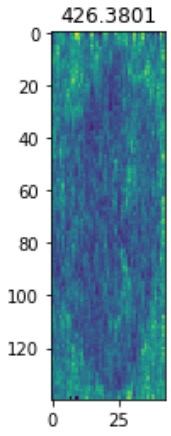
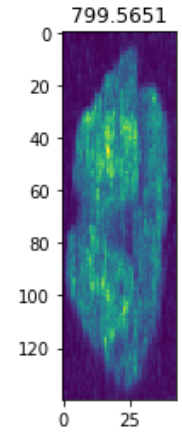
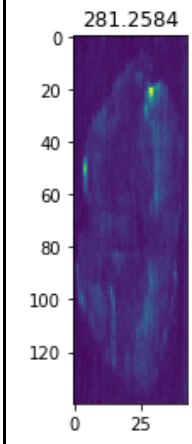
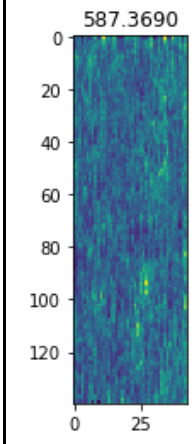
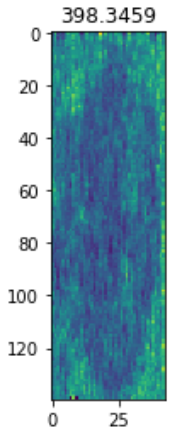
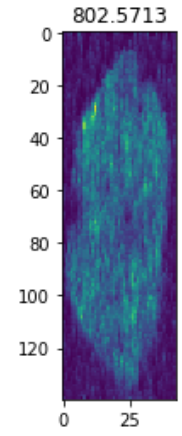
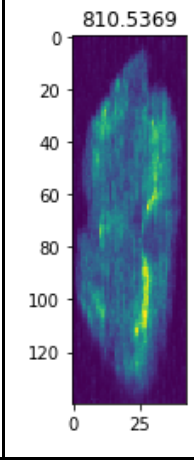
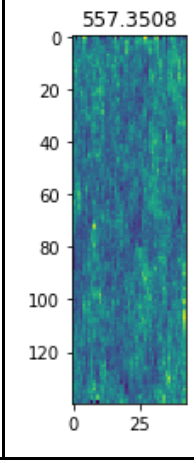
```
                        double* xArray = NULL;
                        SAFEARRAY *safeXArray = NULL;
                        hr = spectrum->get_XArray(&safeXArray);
                        assert(hr == S_OK);
                        SafeArrayGetLBound(safeXArray, 1, &lBound);
                        SafeArrayGetUBound(safeXArray, 1, &uBound);
                        SafeArrayAccessData(safeXArray, reinterpret_cast<void**>(&xArray));
                        zVals = new double[zSize];
                        std::copy(xArray, xArray + zSize, zVals);
                        float x_ptr = (float)xSize;
                        float y_ptr = (float)ySize;
                        float z_ptr = (float)zSize;

                        fwrite(&x_ptr, FLOAT_SIZE, 1, pFile);
                        fwrite(&y_ptr, FLOAT_SIZE, 1, pFile);
                        fwrite(&z_ptr, FLOAT_SIZE, 1, pFile);
                    }

                }
            }

            if (matrixDefined) {
                v.resize(zSize, 0);
                copy(v.begin(), v.end(), matrix + current);
                fwrite(&v[0], FLOAT_SIZE, v.size(), pFile);
                current += zSize;
            }
        }
    }

    float *xVals, *yVals, *start_of_pr;
    v.clear();

    xVals = generateAxisArray(xSize, pixelSizeX);
    yVals = generateAxisArray(ySize, pixelSizeY);
    for (int i = 0; i < zSize; i++) {
        v.push_back((float)zVals[i]);
    }
    fwrite(xVals, FLOAT_SIZE, xSize, pFile);
    fwrite(yVals, FLOAT_SIZE, ySize, pFile);
    fwrite(&v[0], FLOAT_SIZE, v.size(), pFile);

    fclose(pFile);

    std::cout << "Done. Results have been written to file: " << filename << std::endl;

    system("Pause");
    return 0;
}

std::vector<std::wstring> getFoldersFromUserSpecifiedDirectory() {
    std::vector<std::wstring> folders;
    IFileDialog *pfd = NULL;
    DWORD dwOptions;
    IShellItem *psiResult;
    PWSTR pszFilePath = NULL;
    HRESULT hr = CoCreateInstance(CLSID_FileOpenDialog,
        NULL,
        CLSCTX_INPROC_SERVER,
        IID_PPV_ARGS(&pfd));
    assert(SUCCEEDED(hr));

    if (SUCCEEDED(pfd->GetOptions(&dwOptions)))
        pfd->SetOptions(dwOptions | FOS_PICKFOLDERS);
```

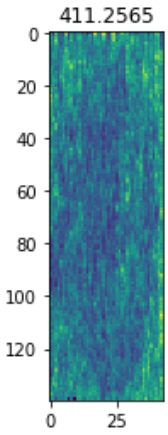
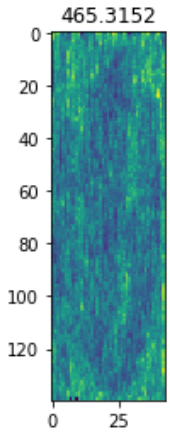
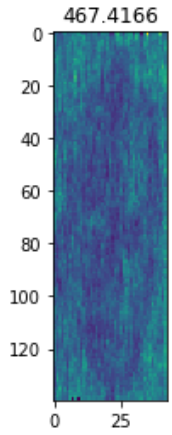
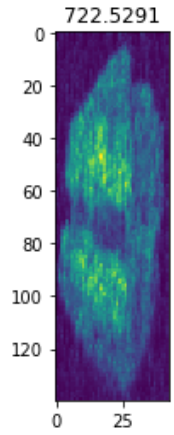
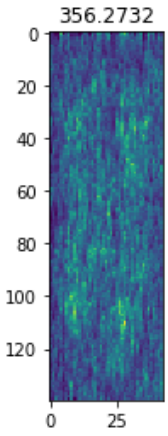
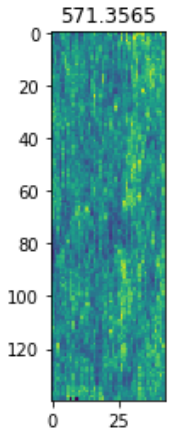
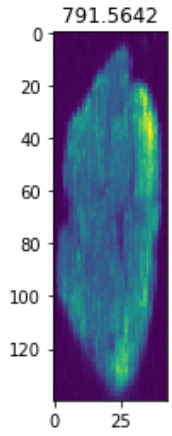
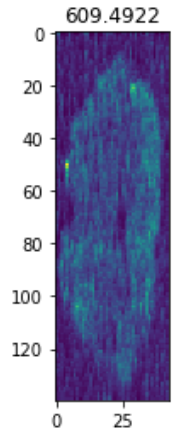
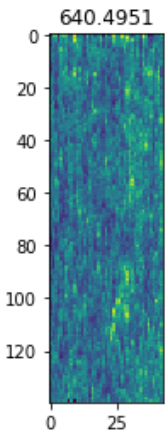
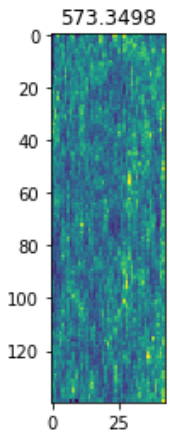
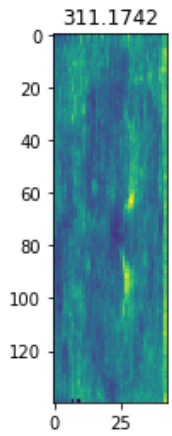
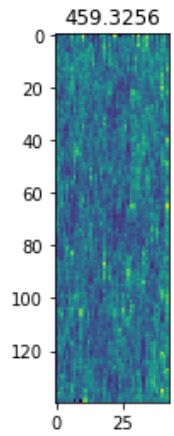
```
    hr = pfd->Show(NULL);
    assert(SUCCEEDED(hr));

    hr = pfd->GetResult(&psiResult);
    assert(SUCCEEDED(hr));
    hr = psiResult->GetDisplayName(SIGDN_FILESYSPATH, &pszFilePath);
    assert(SUCCEEDED(hr));

    std::wstring ws(pszFilePath);
    std::wstring directoryPath(ws.begin(), ws.end()), directoryContent;

    for (const auto & p : fs::directory_iterator(directoryPath)) {
        directoryContent = p.path().wstring();
        if (directoryContent.substr(directoryContent.length() - 2) == L".d") {
            folders.push_back(directoryContent);
        }
    }
    return folders;
}

CComBSTR* sortFoldersByNumberInPath(std::vector<std::wstring> list, int size) {
    CComBSTR* folders = new CComBSTR[size];
    int position, index;
    for (std::wstring p : list) {
        position = p.length() - 3;
        while (p[position] >= '0' && p[position] <= '9') {
            position--;
        }
        index = stoi(p.substr(position + 1, p.length() - 3 - position));
        folders[index - 1] = SysAllocStringLen(p.data(), p.size());
    }

    return folders;
}

float* generateAxisArray(int length, int pixelSize) {
    float* vals = new float[length];
    float currentVal = pixelSize / 2;
    for (int i = 0; i < length; i++) {
        vals[i] = currentVal;
        currentVal += pixelSize;
    }
    return vals;
}
```

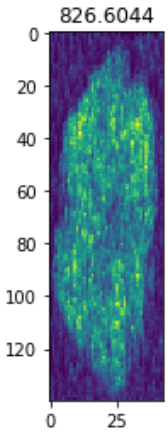
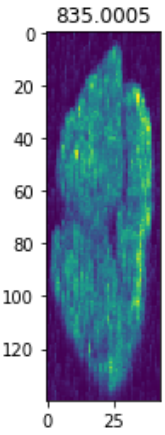
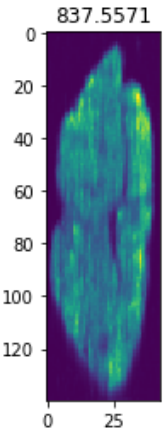
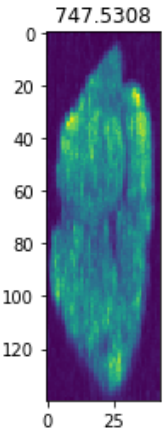
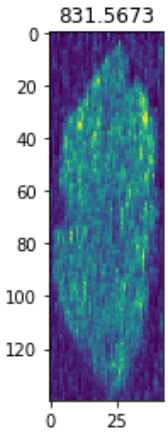
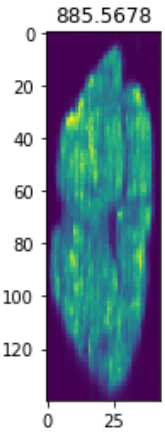
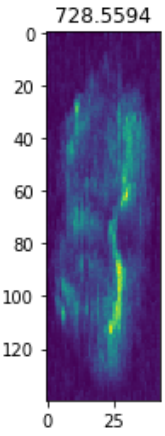
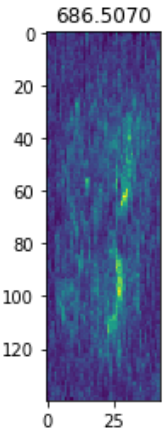
Appendix B: Supervised Learning Selected Features

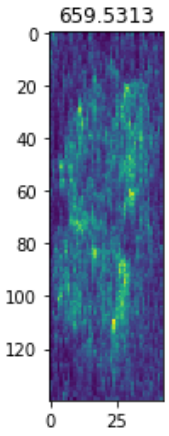
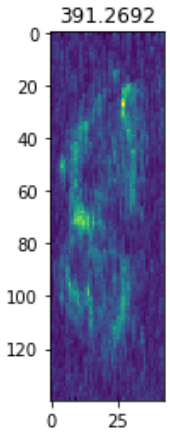
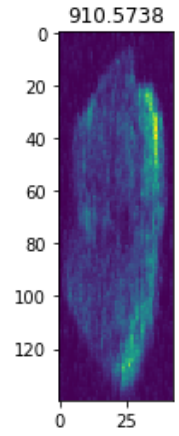
This appendix consists of the results of a variety of machine learning algorithms. Using each algorithm's feature importance measure, the following m/z values were extracted as particularly predictive. The first table shows the results of softmax regression, and the second compares the results of a standard random forest, XGBoost, and the first dimension of a principal component analysis.

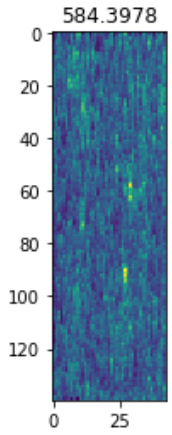
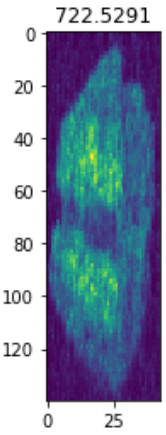
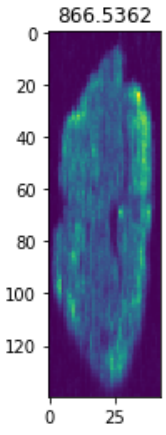
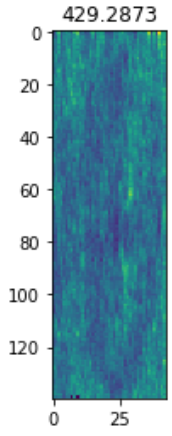
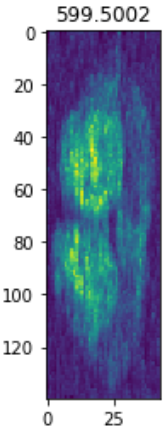
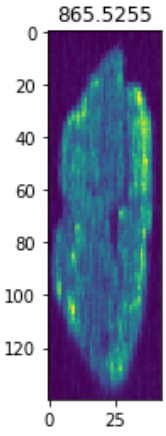
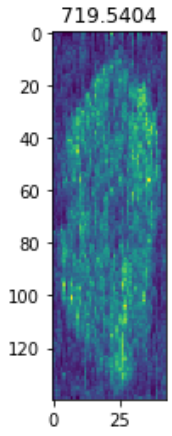
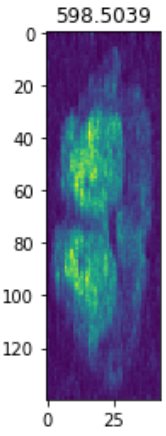
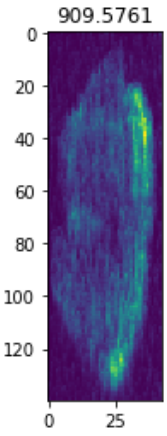
Ostensible Importances	**Softmax Analysis**			
	Caudoputamen	**Cortex**	**Middle**	**Brain - other**
Most important				
2nd-most important				

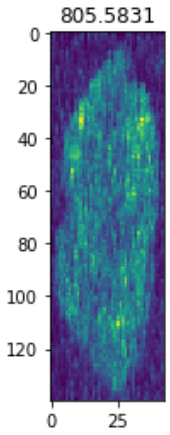
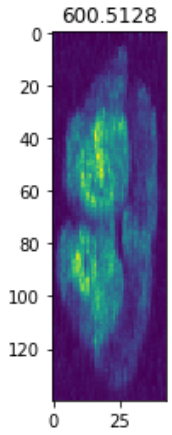
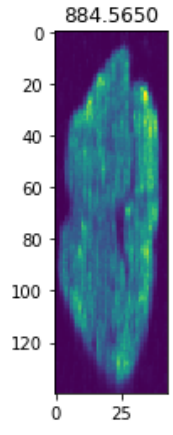
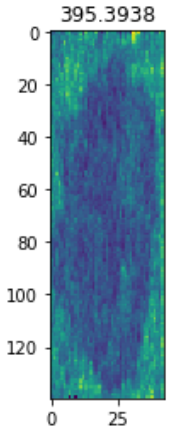
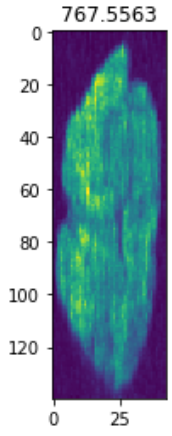
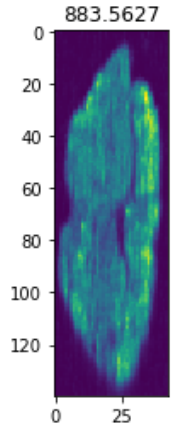
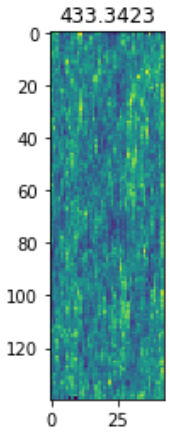
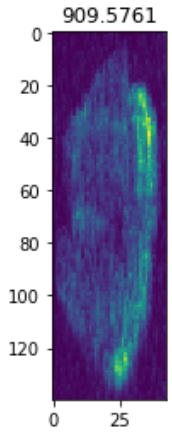
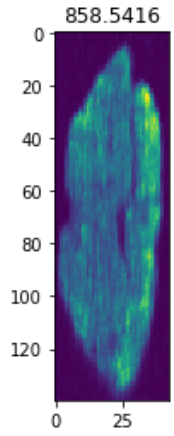
3rd, etc.	 840.5785	 796.5645	 701.5193	 744.5602
4	 426.3801	 799.5651	 281.2584	 587.3690
5	 398.3459	 802.5713	 810.5369	 557.3508

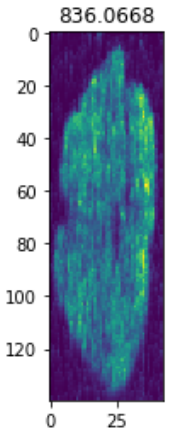
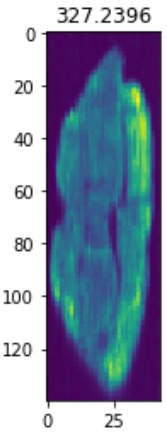
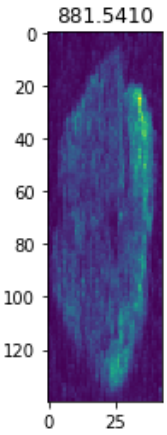
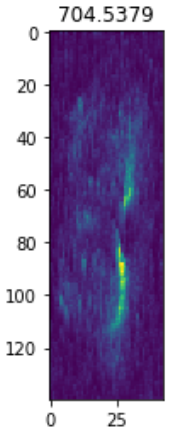
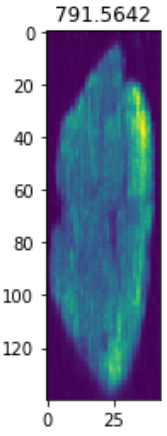
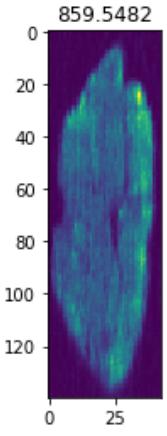
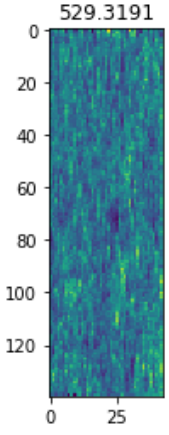
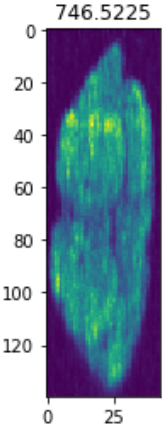
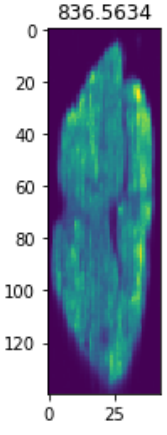
6	 411.2565	 465.3152	 467.4166	 722.5291
7	 356.2732	 571.3565	 791.5642	 609.4922
8	 640.4951	 573.3498	 311.1742	 459.3256

9				
10				

	Random Forest	**XGBoost**	**PCA** (first dimension)
Most important			

2nd-most important			
3rd, etc.			
4			

5	 805.5831	 600.5128	 884.5650
6	 395.3938	 767.5563	 883.5627
7	 433.3423	 909.5761	 858.5416

8	 836.0668	 327.2396	 881.5410
9	 704.5379	 791.5642	 859.5482
10	 529.3191	 746.5225	 836.5634