2021

A Review and Evaluation of Techniques for Improved Feature Detection in Mass Spectrometry Data

Annika R. Tostengard
University of Montana, Missoula

Rob Smith
University of Montana, Missoula

A REVIEW AND EVALUATION OF TECHNIQUES FOR IMPROVED FEATURE DETECTION IN MASS SPECTROMETRY DATA

By
Annika Tostengard
Bachelor of Science, University of Montana, Missoula, MT, 2016

Thesis

presented in partial fulfillment of the requirements
for the degree of

Master of Science
in Computer Science

The University of Montana
Missoula, MT

January 2021

Approved by:
Scott Whittenburg, Dean of The Graduate School
Graduate School

Rob Smith
Computer Science

Michael Cassens
Media Arts

Doug Brinkerhoff
Computer Science

Table of Contents

Abstract – A Review and Evaluation of Techniques for Improved Feature Detection in Mass Spectrometry Data

Mass spectrometry (MS) is used in analysis of chemical samples to identify the molecules present and their quantities. This analytical technique has applications in many fields, from pharmacology to space exploration. Its impacts on medicine are particularly significant, since MS aids in the identification of molecules associated with disease; for instance, in proteomics, MS allows researchers to identify proteins that are associated with autoimmune disorders, cancers, and other conditions. Since the applications are so wide-ranging and the tool is ubiquitous across so many fields, it is critical that the analytical methods used to collect data are sound.

Data analysis in MS is challenging. Experiments produce massive amounts of raw data that need to be processed algorithmically in order to generate interpretable results in a process known as feature detection, which is tasked with distinguishing signals associated with the chemical sample being analyzed from signals associated with background noise. These experimentally meaningful signals are also known as features or extracted ion chromatograms (XIC) and are the fundamental signal unit in mass spectrometry. There are many algorithms for analyzing raw mass spectrometry data tasked with distinguishing real isotopic signals from noise. While one or more of the available algorithms are typically chained together for end-to-end mass spectrometry analysis, analysis of each algorithm in isolation provides a specific measurement of the strengths and weaknesses of each algorithm without the confounding effects that can occur when multiple algorithmic tasks are chained together. Though qualitative opinions on extraction algorithm performance abound, quantitative performance has never been publicly ascertained. Quantitative evaluation has not occurred partly due to the lack of an available quantitative ground truth MS1 data set.

Because XIC must be distinguished from noise, quality algorithms for this purpose are essential. Background noise is introduced through the mobile phase of the chemical matrix in which the sample of interest is introduced to the MS instrument, and as a result, MS data is full of signals representing low-abundance molecules (i.e. low-intensity signals). Noise generally presents in one of two ways: very low-intensity signals that comprise a majority of the data from an MS experiment, and noise features that are moderately low-intensity and can resemble signals from low-abundance molecules deriving from the actual sample of interest. Like XIC algorithms, noise reduction algorithms have yet to be quantitatively evaluated, to our knowledge; the performance of these algorithms is generally evaluated through consensus with other noise reduction algorithms.

Using a recently published, manually-extracted XIC dataset as ground truth data, we evaluate the quality of popular XIC algorithms, including MaxQuant, MZMine2, and several methods from XCMS. XIC algorithms were applied to the manually extracted data using a grid search of possible parameters. Performance varied greatly between different parameter settings, though nearly all algorithms with parameter settings optimized with respect to the number of true positives recovered over 10,000 XIC. We also examine two popular algorithms for reducing background noise, the COmponent Detection Algorithm (CODA) and adaptive iteratively reweighted Penalized Least Squares (airPLS), and compare their performance to the results of feature detection alone using algorithms that achieved the best performance in a previous evaluation. Due to weaknesses inherent in the implementation of these algorithms, both noise reduction algorithms eliminate data identified by feature detection as significant.

Quantitative Evaluation of Ion Chromatogram Extraction Algorithms

Annika Tostengard and Rob Smith

Abstract

Extracted ion chromatograms (XIC) are the fundamental signal unit in mass spectrometry. There are many algorithms for analyzing raw mass spectrometry data tasked with distinguishing real isotopic signals from noise. While one or more of the available algorithms are typically chained together for end-to-end mass spectrometry analysis, analysis of each algorithm in isolation provides a specific measurement of the strengths and weaknesses of each approach. Though qualitative opinions on extraction algorithm performance abound, quantitative performance has never been publicly ascertained. Quantitative evaluation has not occurred partly due to the lack of an available quantitative ground truth MS1 dataset.

Using a recently published data set of manually-extracted XIC as ground truth data, we evaluate the quality of popular XIC algorithms, including MaxQuant, MZMine2, and several methods from XCMS. The manually-curated dataset comprises 48 human proteins stratified over 6 abundance orders of magnitude. Signals in the sample were manually curated into XIC using a commercial tool for visually identifying XIC and isotopic envelopes. XIC algorithms were applied to the manually extracted data using a grid search of possible parameters. Performance varied greatly between different parameter settings, though nearly all algorithms with parameter settings optimized with respect to the number of true positives recovered over 10,000 XIC.

Introduction

Liquid chromatography mass spectrometry (LC-MS) is a popular modality for the identification and quantification of molecular content within biological samples. It is particularly well-suited to high throughput, label-free experiments. LC-MS experiments result in raw output, typically consisting of MS1 and MS2 data, that must be analyzed with data processing software to yield molecular identities and quantities.

One increasingly critical component of data processing is the extraction of extracted ion chromatograms (XIC) from raw data. Correct XIC extraction is essential to important downstream tasks such as molecular identification, charge state assessment, and run-to-run normalization. There are many algorithms for extracting XIC from raw mass spectrometry data. These algorithms vary in approach. Some rely on MS2 identifications to locate peptide XIC within specific mass-to-charge (m/z) and retention time (RT) ranges in precursor data (possible, for example, in Skyline[1], openSwath[2] and Spectronaut[3]). Other algorithms are capable of MS1 XIC extraction without explicit MS2 identification, such as Dinosaur[4]. In this evaluation, we have included open source, popular algorithms for data-independent XIC extraction, discussed below.

Although the choice and application of data processing software can have as dramatic an effect on experimental results as benchtop protocol[5], very few algorithms and software have been quantitatively evaluated[6]. This is, in part, due to a lack of positive control data sets which can be used to quantify accuracy and precision, also known as ground truth data. To date, published XIC extraction algorithms have primarily been evaluated through qualitative comparison to chemical ground truth (e.g., spiked-in standards), quantitative comparison to simulated data, or qualitative comparison to results from other methods (so-called “consensus results”). Each of these methods has significant drawbacks. Simulated data as ground truth poses a problem because the simulation is only as good as model, and an accurate model requires ground truth to build. Chemical ground truth does not provide the granularity necessary to isolate the experimental effect of XIC extraction results due to the presence of many confounding factors (such as variance in digestion, ionization, and elution). Consensus evaluations, on the other hand, are inherently limited to answering the question, “How well do new methods perform compared to old methods?” and not the real question of interest: “How well do new methods perform compared to the true answer?

Other evaluations of XIC extraction software performance include an evaluation of whether reported variations in protein abundance between samples is driven by software or is naturally-occurring[7], which suggested that the software introduced more variability in protein abundance than was found with biologically introduced protein variability. There is also an R package called LFQbench, which allows software developers to determine precision and accuracy in data-independent acquisition on certain high-complexity data sets[8].

Given the limitations of previously-used surrogates for ground truth and that the “true” answer is not known, we adopt the assumption general to most every computational field—that the best available performance measurement is how well methods compare to the best answer obtainable by manual (human) curation. While hand-curated data is not *perfectly* correct, it does (by definition) represent as close to the correct answer as possible given human limitations.

In this study, we present a benchmark analysis of XIC algorithms in isolation using a new, manually-curated data set. To this end, we use a recently published dataset of over 11,000 hand-curated XIC from a published UPS2 protein standard spiked into *ecoli*[9].

We evaluate several popular algorithms for XIC extraction, including Massifquant[10], the unnamed XIC extraction algorithm from MaxQuant[11] (referred to by its parent program hereafter), CentWave[12], the “centroid” algorithm from MZMine2[13] (also referred to by its parent program hereafter), and MatchedFilter[14].

OpenMS[15], a popular tool that includes feature detection, was excluded from the evaluation due to the fact that it uses a single-stage envelope level detection algorithm.

General description, performance and specific measurement of the strengths and weaknesses of each approach are provided.

The intent of this study is to provide insight on the real-world performance of common XIC extraction approaches, highlight persisting weaknesses, and provide direction for novel approaches. This benchmark can be used to ensure that any novel XIC extraction algorithm performs at least as well as existing algorithms, helping to mitigate the proliferation of publications that make it difficult for practitioners to keep track of the state of the art.

Massifquant

Massifquant uses 2D Kalman filters (KFs) to identify XIC in XCMS data, where a single KF tracks an XIC’s m/z and intensity over the chromatogram. Each instance of a KF, called a track, starts with the detection of centroids in a single scan, seen in Figure 1a. The existence of a centroid in the next scan is then predicted. If a real centroid is detected in the next scan, and that centroid is close enough to the prediction made by the KF (where closeness is determined by quasi-confidence intervals centered about the prediction), that centroid will be added to the current track, seen in Figure 1b and 1c. The track is terminated once the signal disappears, and unclaimed centroids spawn new tracks, so there are several tracks being maintained at once. Tracks are also discarded if they do not meet criteria for minimum length, intensity, expected m/z deviance or consecutive missed predictions.

Massifquant also uses the concept of a Kalman gain, a weight given to the estimate of the location of the current centroid; a smaller Kalman gain indicates that a centroid's location as perceived by the algorithm is more likely to be close to the true location of the centroid.

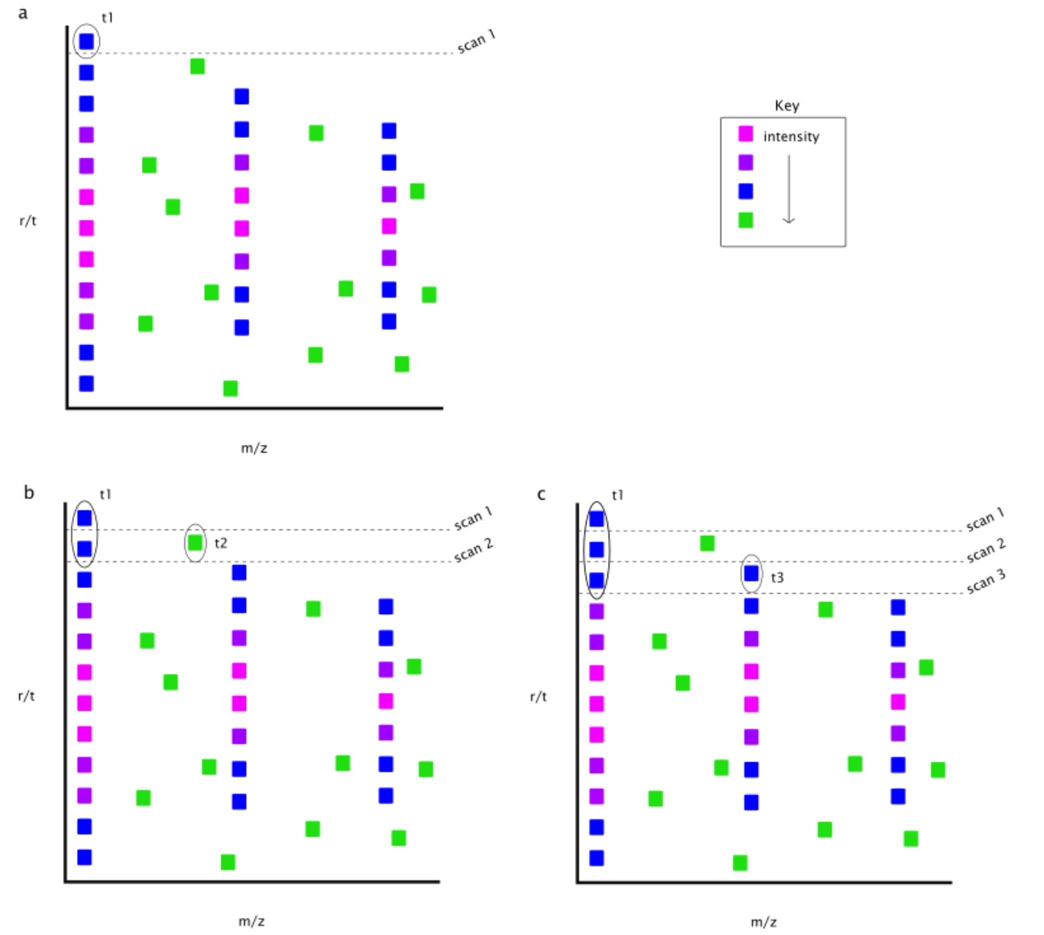

Figure 1: Detection of XIC by Massifquant. a) an XIC *t1* is created after detection of a centroid in the first scan. b) a centroid is added to t1 as its location is close enough to the predicted location of a centroid made after scan 1 and a new XIC (t2) is created. c) another centroid is added to t1, a new XIC (t3) is started, and t2 is

MaxQuant

XIC extraction is the first part of the two-phase MaxQuant isotopic envelope extraction algorithm, where a Gaussian peak shape is fitted over the high density regions of data points in each scan, seen in Figure 2b. On the retention time axis, XIC are split at significant local minima, seen in Figure 1a. The masses of the peaks are then estimated from the centroid masses, weighting intensity where precision is calculated by bootstrap replication. Because MaxQuant is not open-source, the algorithm was reimplemented based on the original publication in order to access results after the XIC extraction portion. Although MaxQuant is not suggested for use with centroided data, we have included it here because it is so widely used.

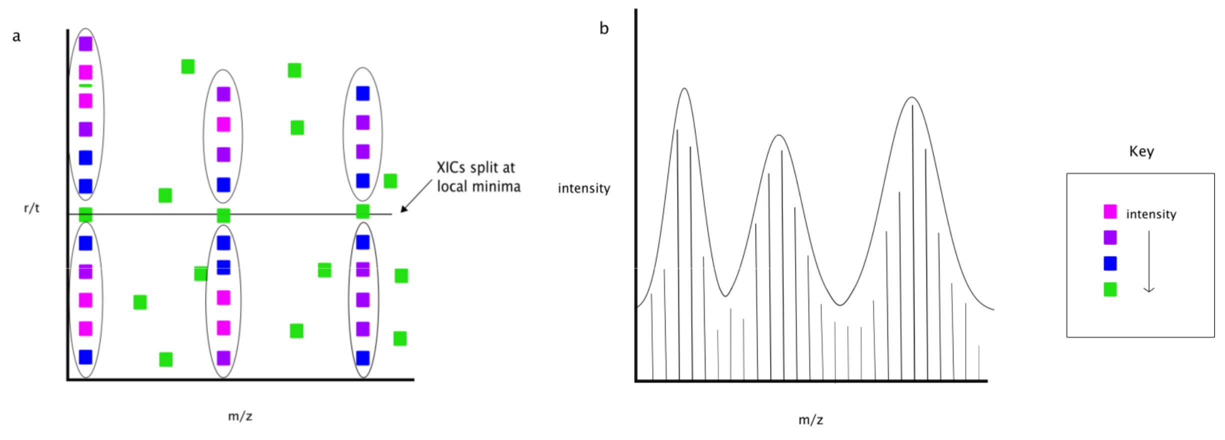

Figure 2: Detection of XIC by MaxQuant. a) MaxQuant splits XIC at local minima. b) MaxQuant fits Gaussian peak shapes over a single scan.

Matched Filter

MatchedFilter extracts ion chromatograms by first splitting the run into bins 0.1 m/z wide (shown in Figure 3b) and isolating the maximum intensity at each time point in the bin. The lists of mass/intensity pairs (one for each scan) are converted into a matrix. The matrix rows represent equally spaced masses, and the columns represent a single scan.

The matrix is constructed in one of four ways according to a user-selected algorithm. The "bin" algorithm places an intensity into the matrix cell that is nearest to it in mass and is suggested for use with centroided data[16]. As UPS2 is a centroided dataset, only runs using the "bin" algorithm are included; originally, several runs were performed with all the binning algorithms available,

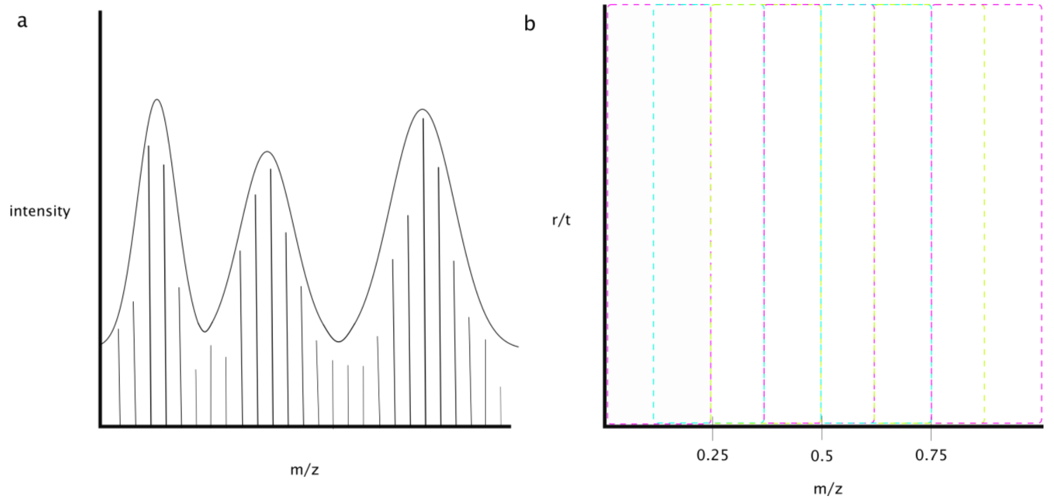

Figure 3: Detection of XIC by MatchedFilter. a) Gaussian peak fitting. b) MatchedFilter divides the chromatogram into bins.

and the preliminary results clearly showed that the "bin" algorithm outperformed all others on this dataset.

After the matrix is constructed, it is then filtered by matched filtration using a second-derivative Gaussian as the model peak shape, seen in Figure 3a. A signal-to-noise ratio cutoff, calculated by taking the mean of the unfiltered data, is then used to discard some of the peaks. To find peak mass, a two-step strategy is used: using the full-resolution data, the mass is calculated in each spectrum containing the peak, then the overall peak mass is calculated as a weighted mean of all the full-resolution masses, using intensities as weights.

CentWave

CentWave avoids binning by making Regions of Interest (ROI) using the m/z values from the first scan, seen in Figure 4a, then calculates the mean m/z value for each ROI detected. Then, in consecutive scans, the algorithm checks whether the absolute difference between the current scan's m/z value and the mean m/z value for the detected ROI is less than the mass accuracy. If it is, additional centroids are added to ROI, as seen in Figure 4b and 4c. The baseline intensity is calculated by discarding both the 5% least intense signals and the 5% most intense signals, then finding the mean intensity of the remaining 90% of the data; the standard deviation is used as the noise level. The Continuous Wavelet Theorem is then applied to the intensity values of each ROI and chromatographic peaks are located by descent on the filtered peak similar to MatchedFilter. A Gaussian curve can also be fitted to each feature according to user specification.

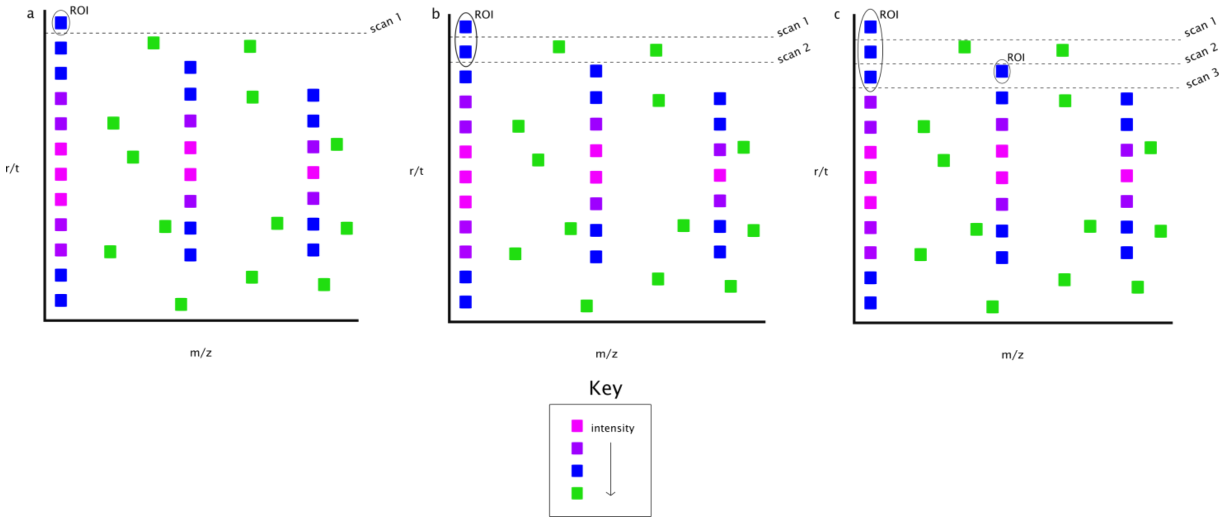

Figure 4: Detection of XIC by CentWave. a) a new ROI (region of interest) is detected. In scan 2 and 3, centroids are added to XIC (see b and c). In scan 3, a new ROI is also detected (c).

MZMine2

In MZMine2, there are many different modules available depending on the type of data to be processed, and only the specific workflow and modules used here are described. The initial step in the MZMine2 workflow is to run one of the mass detection modules. Five are available, but only one, "centroid", is suitable for centroided data. This mass detector detects all data points above the specified noise level as peaks, seen in Figure 5. The next step in MZMine2's workflow is to connect consecutive m/z values spanning over multiple scans into chromatogram objects, which it does by constructing a set of potential chromatograms spanning over consecutive scans. In each scan, the mass list of that scan is iterated through so that each ion is connected to its corresponding chromatogram as specified by the m/z tolerance parameter. If no corresponding chromatogram can be found, a new one is created. When there are no longer any m/z peaks being connected to a particular chromatogram, it is terminated and checked for RT span and intensity to make sure they fall within the user-specified parameters.

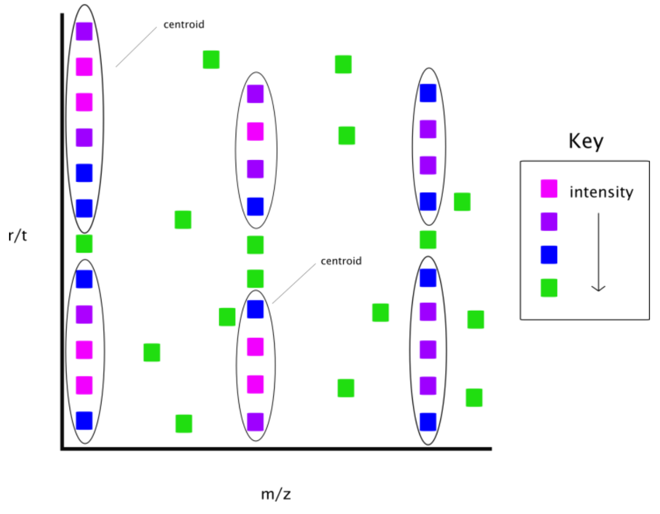

Figure 5: Detection of XIC by MZMine2. The centroid algorithm from MZMine2 only adds signals to a centroid if the signal is above the signal-to-noise threshold.

Methods

The curated data set is an untargeted protein identification sample consisting of 48 Universal Proteomics Standard (UPS) proteins. UPS2 has been used in many publications as a known set of molecules and abundances that approaches the large dynamic range of abundances present in naturally-occurring biological samples. The proteins are organized into six groups of abundances with eight protein types per group. The abundances per group vary from 0.5 fmol to 50,000 fmol with each group differing by an order of magnitude. The raw data consists of a trypsin-digested run of UPS2 produced and recently published by the Nesvizhskii group as part of comparison of state-of-the-art data-dependent and data-independent acquisition methods[17]. It provides an independent representative example of a run using modern instrumentation, wet lab, and instrumental protocol. The file in question was created using a data independent acquisition protocol on an AB Sciex TripleTOF 5600, using a 250-ms ion accumulation time for MS1 survey scans. The raw data file is publicly available in PRIDE repository PXD001587 under filename 18185_REP2_4pmol_UPS2_IDA_1.mzXML and consists of data centroided by ProteinPilot (Sciex) software. The mzXML file was converted to mzML using msconvert.

Using a newly available software platform for computational mass spectrometry (Prometheus by Prime Labs), we conducted an extensive manual annotation process comprised of more than 1,000 human hours. The data set consists of more than 62 million points, with 1,294,008 points grouped into 57,518 extracted ion chromatograms.

The abundance of parameter settings required to run most published algorithms can significantly affect the performance of the algorithm on particular datasets. For all algorithms except MaxQuant, in order to ensure that an algorithm is not undervalued because of poor parameter choices, runs were performed over parameter ranges. Published parameter settings[6,10,12] were used for all algorithms, while for CentWave and MatchedFilter, these published parameter suggestions were also combined with the optimized settings suggested by the Isotopologue

Parameter Optimization (IPO) tool[18], created in part by one of the authors of the CentWave XCMS algorithm.

Parameters

Massifquant

The parameters tested for Massifquant were parts per million, signal-to-noise threshold, peakwidth and critical value[12,18]. Caution should be exercised when choosing Massifquant parameters, as the selection of certain values will result in unreasonable runtimes. For example, setting the peak width to anything below 4 resulted in the run taking more than several days and up to a week. Critical value was another parameter that drastically affects the runtime; a large peak width with a small critical value would run within a few hours while a large peak width with large critical values would take ~8 hours. The parameters tested are shown below; the algorithm was run with a grid search so that every combination of the parameters was evaluated (see Table 1).

TABLE 1: Parameters tested for Massifquant.

criticalValue	0.1	1	2	3	
ppm	10	20	30	40	50
Peakwidth (sec)	4	8	16		
snthresh	1	10	100		

Matched Filter

The parameters tested for Matched Filter were step, full width half maximum (fwhm), signal-to-noise threshold (snthresh), and the m/z difference (mzdiff) (see Table 2). The initial parameters were suggested by IPO, the Isotopologue Parameter Optimization tool for the MatchedFilter and CentWave methods in the XCMS package; the final values were chosen in a range around these values. Again, because the "bin" algorithm from MatchedFilter is recommended for centroided data, it was the only one included in this evaluation.

TABLE 2: Parameters tested for Matched Filter.

step	0.01	0.02	0.04	0.06	0.08	
fwhm	10	20	30	40	50	
mzdiff	0	0.2	0.4	0.6	0.8	1
algorithm	bin					
snthresh	1,2,…20					

CentWave

We tested parts per million (ppm), peakwidth, and signal-to-noise threshold (snthresh), as these are considered to be the most pertinent parameters[12]. CentWave is the second algorithm that is available for use with the IPO tool, so the range of parameters were chosen considering these suggestions as well as the default settings. For both peakwidth and snthresh, values between 1 and 20 were tested, incrementing by 1. For ppm, 10, 20, 30, 40, and 50 were tested. As with all the other XCMS algorithms, a grid search was run on all combinations of parameter settings.

MZMine2

We originally wanted to do three runs of MZMine2, using noise levels of 1, 10 and 100. However, as with Massifquant, choosing a noise level of 1 prevented the algorithm from being able to build a chromatogram within a reasonable time.

Even choosing 5, it still took several hours to complete the chromatogram builder for each parameter permutation. The final values chosen for the noise level were 5, 10, and 100.

Metrics

The metrics compare the manually-curated XIC assignments to each algorithm's results. The data from the manually-curated CSV file is stored as a point object consisting of an m/z, RT and intensity. Points that have been identified as belonging to the same XIC are all given the same manually-curated ID number, seen to the right of each point in Figure 6. All points, whether they have been assigned a feature ID or not, are stored in a master list that is then sorted by m/z once all manually-curated data points have been read in; this master list is used to find matching points from the algorithmic files later.

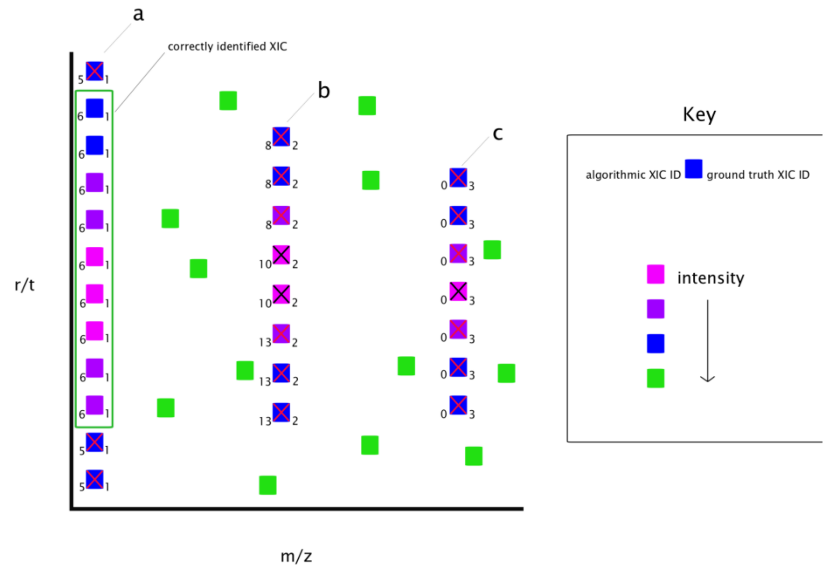

Figure 6: Evaluation of XIC extraction. There are three manually-extracted XIC: a, b, and c. XIC a has been correctly identified because the algorithm grouped the same points under the same algorithmically-curated feature ID, 6. XIC b was not correctly extracted because none of the algorithmically-curated XIC IDs (8, 10, 13) contained at least 50% of the intensity of the total intensity contained in the manually-extracted XIC (manually-curated feature ID 2). In c, the points remained unassigned by the algorithm.

After reading in the manually-curated points, the file of algorithmically-curated XIC is read in. For each of these points, if they have been detected as belonging to an XIC, a search through the manually-curated points is conducted in order to find the point from the manually-curated file that matches in m/z and RT. When a match is found, the original manually-curated point is also assigned an algorithmically-curated XIC ID, seen to the left of the points in Figure 6.

Once all the algorithmically-curated XIC have been read in, all XIC which have been identified in the manually-curated file are examined to see whether the same points that were manually curated into an XIC were also grouped by the algorithm. This is accomplished by checking that the algorithmically-curated XIC's total intensity is greater than 50% of the manually-curated XIC's intensity, as suggested by the Massifquant paper. The XIC IDs are used to do this; all the points in each XIC will have the same manually-curated ID and a majority of the points in a correctly-extracted XIC will also share the same algorithmically-curated XIC ID. This is demonstrated by the leftmost XIC in Figure 6, XIC a. The total intensity for a given algorithmically-curated XIC is the sum of the intensity of the points that are assigned the algorithmically-curated XIC's ID. Any XIC in which the algorithmically-curated XIC's total intensity was not greater than 50% of the manually-curated XIC's total intensity are considered false negatives, demonstrated by the center XIC in Figure 6, XIC b. This also ensures that there is only a single match for each manually-curated XIC. Points which were not curated by the algorithm are also considered incorrect, seen in the rightmost XIC in Figure 6, XIC c.

Intensity and m/z error metrics were also calculated. To measure intensity error, we report the ratio between the total intensity of all the points clustered into true positive XIC and the total intensity of all manually curated points.

intensity ratio = ∑(algorithm_true_positive_intensity) / ∑(manually_curated_intensity)

To calculate the m/z error, we first calculated the average m/z weighted by intensity by multiplying each point's intensity and m/z, then summing those values over each XIC, then normalizing by the total intensity.

weighted average m/z = ∑(intensity * m/z) / ∑ intensity

The total error was found by then taking the absolute value of the difference between the weighted average manually-curated m/z and the weighted average algorithmically-curated m/z.

m/z error = ∑(|ground_truth_weighted_average_mz - experimental_ weighted_average_mz |)

Results

Each algorithms' highest-performing parameter set, with regard to true positives, is documented in Tables 3-5.

TABLE 3: Best performing parameters for MatchedFilter.

parameter	*step*	*fwhm*	*snthresh*	*mzdiff*
value	0.02	10	1	0

TABLE 4: Best performing parameters for Massifquant.

parameter	*ppm*	*peakwidth*	*snthresh*	*critical value*
value	10	4	1	1

TABLE 5: Best performing parameters for CentWave.

parameter	*ppm*	*peakwidth*	*snthresh*
value	50	1	20

For MZMine2, the lowest *snthresh* setting, 5, resulted in the greatest number of matches to the manually curated data. A listing of how each algorithms' best parameter set performed across all five metrics is shown in Table 6. A full listing of every parameter set tested can be found in Supplement tables 1-4. The total XIC reported from both the XIC from the manual curation and other XIC not included in the manual curation is shown in the first column. The number of true positives, shown in the second column, describes the number of XIC from the algorithmically-curated file which matched at least 50% of the total intensity of a manually-curated XIC. The third column, % true positives, is the ratio of the number of true positives discovered by the algorithm over the total number of manually-curated XIC. The false negatives column describes XIC which were manually-curated but not identified by the algorithm, or where the algorithm's XIC did not include at least 50% of the total intensity of the manual XIC (see Figure 7). The fifth column, m/z error, shows the average weighted m/z error, described above. The fifth column shows the intensity ratio, also described above. In cases where the intensity ratio is greater than one, it indicates that the algorithm included more points in true positive XIC than were included in manual curation.

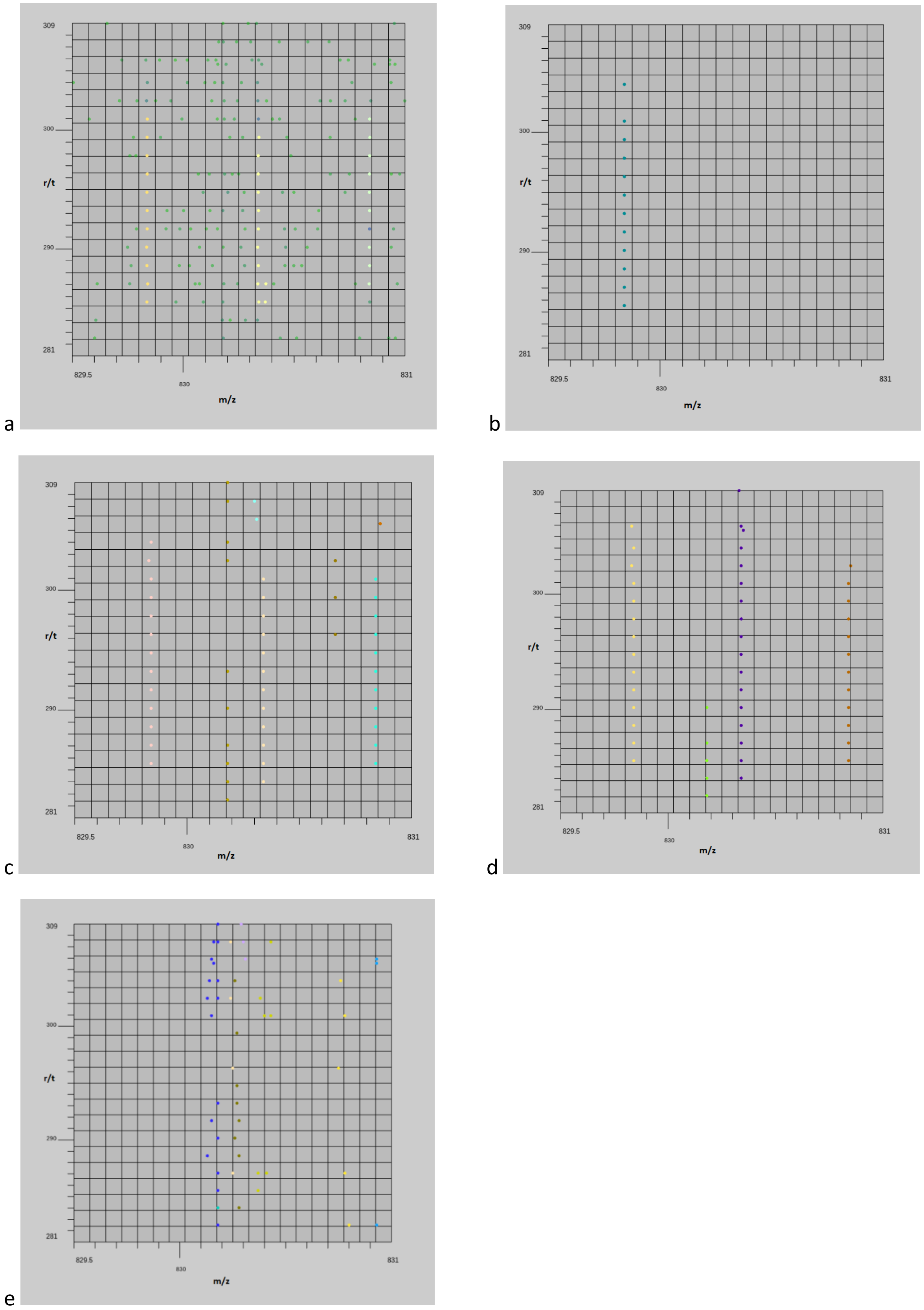

Figure 7: False negative XIC are XIC where < 50% of the hand curated points were captured. Each algorithm differs in the number of and quality of false negatives. Panels show a) manual XIC extraction b) XIC extraction by CentWave c) XIC extraction by Massifquant d) XIC extraction by MatchedFilter e) XIC extraction by MZMine2. Each unique color represents a single XIC.

TABLE 6: Collated results from each algorithm.

	total XIC	true positives	% true positives	false negatives	avg m/z error	intensity ratio
MZMine2	68,664	43,177	76.6%	13,963	0.03	13.0
Massifquant	462,679	38,099	66.7%	19,041	0.02	2.25
MatchedFilter	98,684	27,632	48.4%	29,508	0.02	0.68
CentWave	36,694	17,670	30.9%	39,470	0.02	0.74
MaxQuant	6,532	1,078	1.9%	56,062	13.9	0.15

Below are screenshots of how the algorithms performed on a particular window as viewed in JS-MS[19], an open-source software used to manually extract XIC. Each panel shows a different algorithm, where each XIC is given a unique color. Figure 8 shows the window before any curation has taken place. In Figure 8, color indicates intensity, with pink and purple representing areas of high intensity and blue and green representing areas of low intensity. Figure 9a shows how the window is manually curated. One thing that should be noted in Figure 9a is that it demonstrates the imperfections of manual curation. While low m/z variability is a criterion for choosing which points to include in each XIC, here the operator made a mistake. In Figure 8, there are low-intensity (green) points very close in m/z to the heaviest and lightest XIC in this envelope. In Figure 9a, it can be seen that those low-intensity points which deviated from the m/z of the more intense points were included in the XIC at manual curation, which is indicated by the fact that all of those points are now the same color. This is one reason that adopting the rule that an algorithmically-curated XIC must extract at least 50% of the intensity of the manually-curated XIC is useful, as it mitigates some of the mistakes made during manual curation.

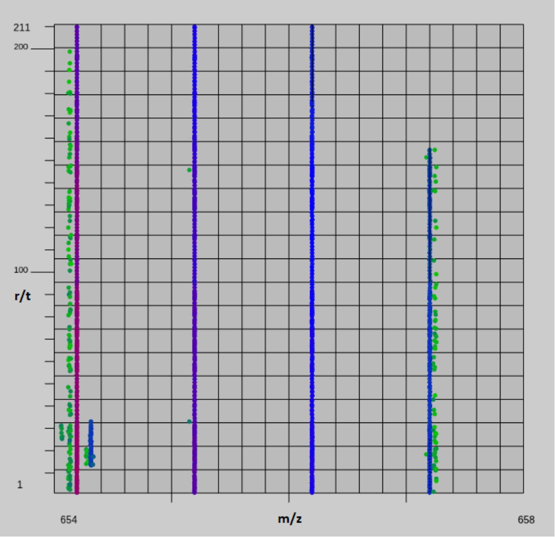

Figure 8: The original, uncurated tile as seen in JS-MS, open source software used for XIC

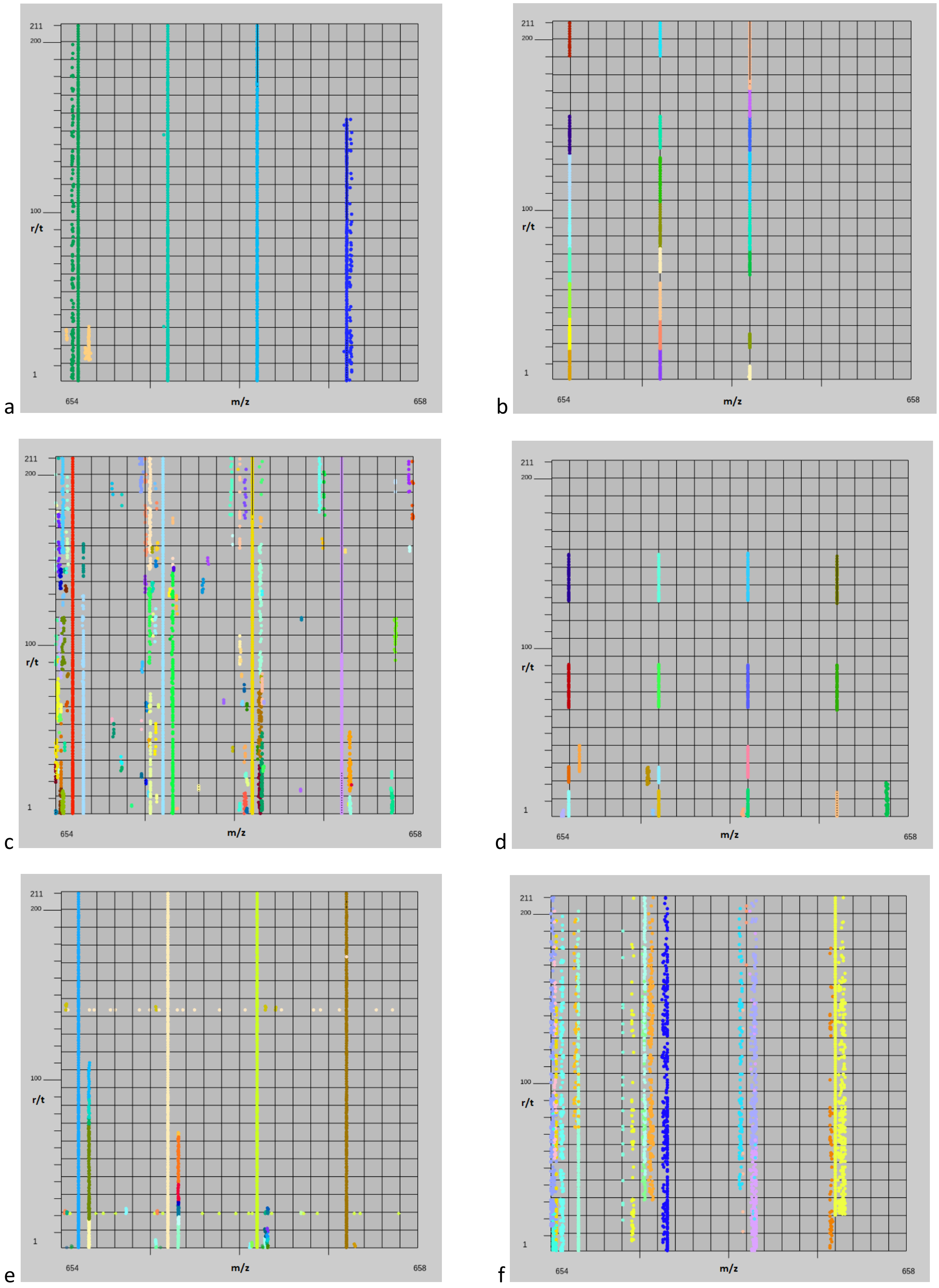

Figure 9: The same window as Figure 8 but curated by a) manual XIC extraction b) XIC extraction by CentWave c) XIC extraction by Massifquant d) XIC extraction by MatchedFilter e) XIC extraction by MaxQuant f) XIC extraction by MZMine2. Each unique color represents a single XIC.

Figures 9b-f are screenshots of how the other algorithms performed XIC extraction on the same window. Each XIC has been assigned a unique color, and each point belonging to that XIC is given that color. 8b and 8d show extraction by CentWave and Matched Filter; both algorithms had a tendency to split a single XIC into multiple XIC. 8c and 8f show extraction by Massifquant and MZMine2. These two algorithms were also qualititatively similar in that they both recovered more XIC than were found during manual curation, and both were found to recover over half the XIC found in the manually-curated file. Figure 9e shows extraction by MaxQuant, which looks very similar to manual curation with some extra signals recovered in the lower bottom half of the window.

Figures 10-12 demonstrate how widely performance varies depending on parameter settings; each figure is a histogram where each bar represents a single run using a single parameter set. The number of true positives recovered by each parameter set is shown on the y-axis. CentWave's lowest-performing settings yielded less than 1,000 correctly-extracted XIC, while Massifquant's lowest correctly extracted less than 4,000 and MatchedFilter's lowest correctly extracted less than 6,000.

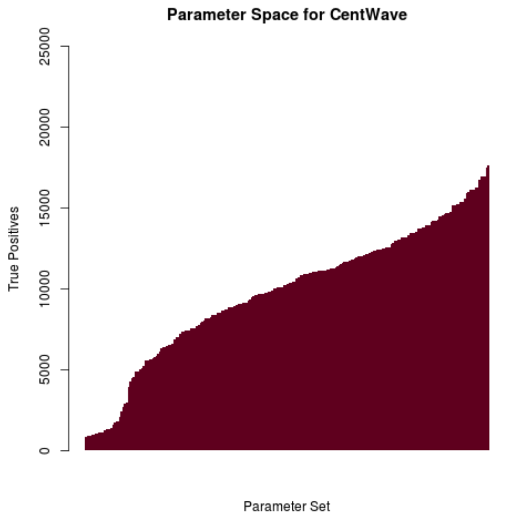

Figure 10: Difference in performance among different parameter sets for CentWave. Each bar represents one parameter combination.

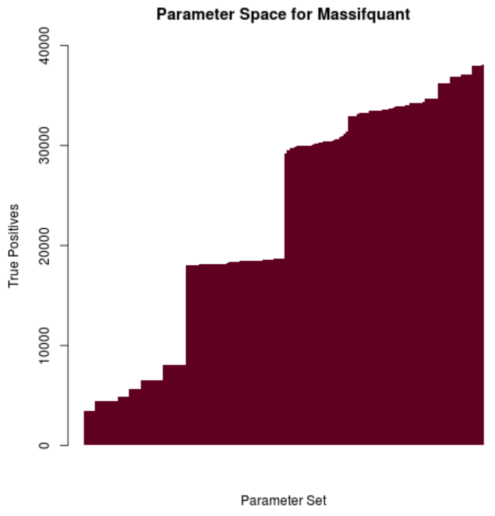

Figure 11: Difference in performance

among different parameter sets for Massifquant. Each bar represents one parameter combination.

Conclusions

The evaluation of popular XIC algorithms on a quantitative manually curated data set showed that performance among the available algorithms for mass spectrometry is variable.

One principal observation from this study include that there is no one clearly superior algorithm for ion chromatogram extraction, and that the state-of-the-art still has room for improvement. Surprisingly, only two algorithms (Massifquant and MZmine) identified more than half of the XIC present in the manually curated dataset, while the others did not identify at least half of the manually curated XIC. In addition, the absolute intensity and m/z error across all algorithms was quite high.

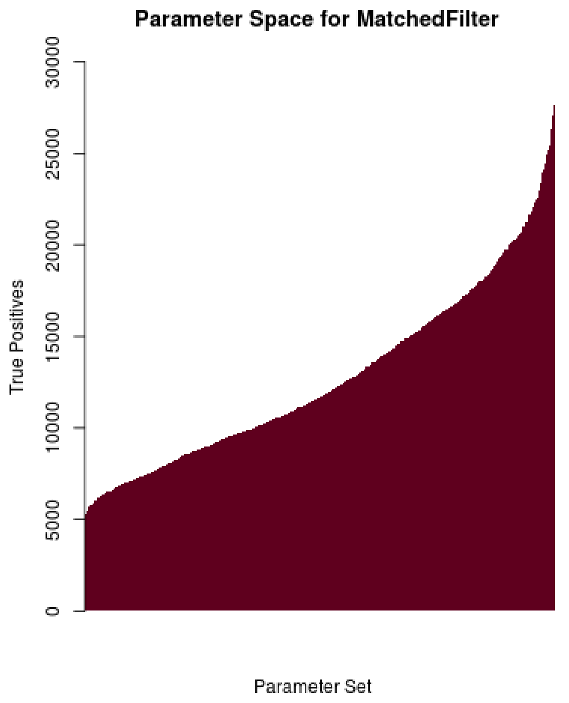

Figure 12: Difference in performance among different parameter sets for MatchedFilter. Each bar represents one parameter combination.

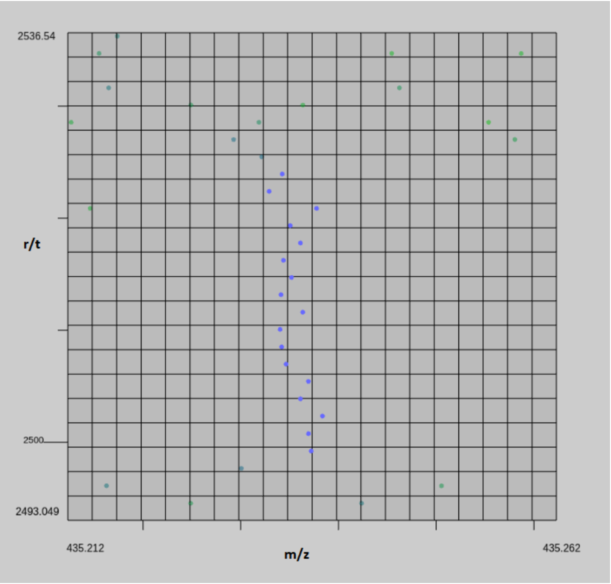

Figure 13: An XIC that was manually curated but not discovered by any algorithm. Lilac/light purple areas are the points included in this signal.

Parameter selection significantly affects results. Each algorithm had drastically differing performance across different parameter settings. In other words, no algorithm was always or even mostly better than any other across parameter settings, though MatchedFilter's performance varied the least. The impact of parameter selection suggests that without a way of optimizing parameters that does not require ground truth, the utility of any XIC extraction algorithm is still unknown. In addition, it is impractical for algorithm developers to suggest a grid search of parameters for algorithm optimization. This experiment took months of runtime, just for a single dataset.

One observation in this regard is that for all algorithms except CentWave, lower signal-to-noise thresholds have a significant impact on the number of XIC found, and so focusing on this parameter as opposed to others that were not found to have as significant an impact could be beneficial to researchers. For Massifquant, for example, any experiments with a critical value of 0.1 performed worse than any other experiment with a critical value between 1-3, though it is unclear if either of these patterns generalize to other data sets.

One of the most interesting results was that the algorithms often split XIC prematurely, or did not split XIC at a local intensity minima. For example, a single XIC that had been extracted manually may

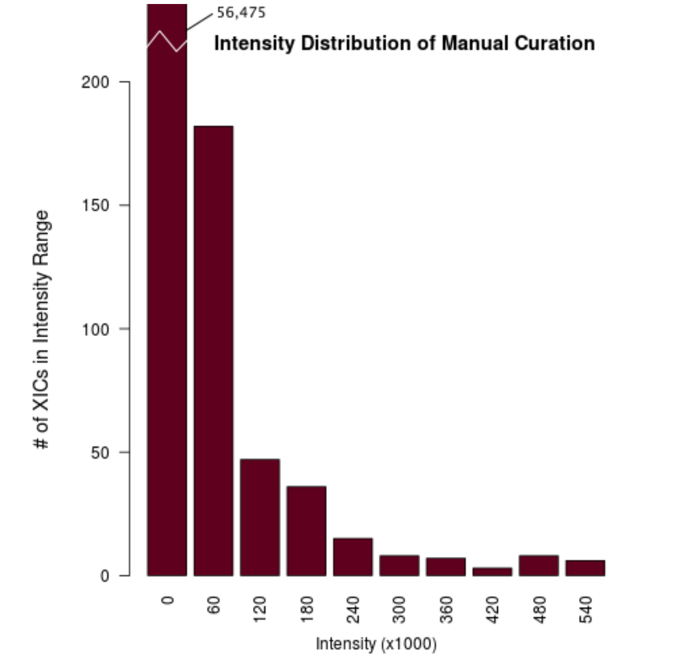

Figure 14: Each bin covers a range of intensity and the height represents the number of XIC that were curated, at that intensity.

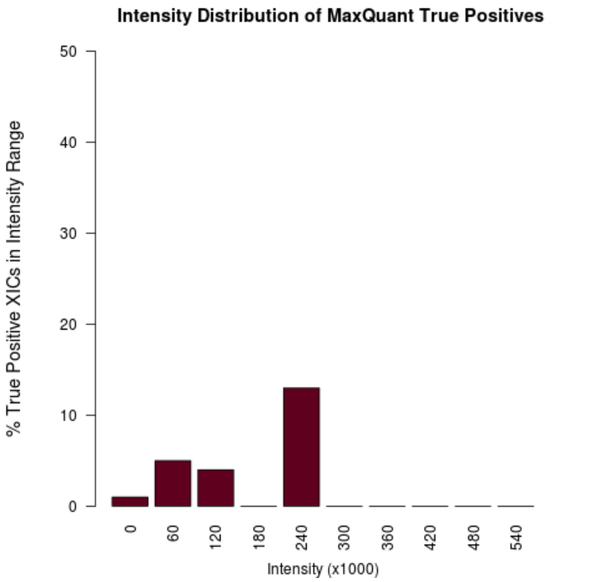

Figure 15: Intensity of MaxQuant true positives. Each bin covers a range of intensity and the height represents the percentage of true positive XIC, at that intensity.

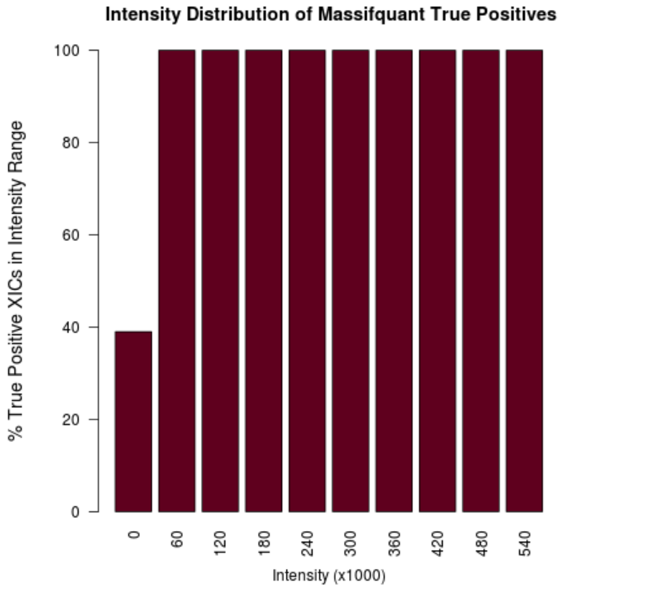

Figure 16: Intensity of Massifquant true positives. Each bin covers a range of intensity and the height represents the percentage of true positive XIC, at that intensity.

have been split into two or more XIC by the algorithm. This phenomenon can be seen in both Figures 8b and 8d, in the segmentation of CentWave and MatchedFilter, respectively. Sometimes, however, the reverse was seen, and the algorithm merged two XIC that had originally been split in the manually-curated data. These “duplicates” appear to be more common in areas where the centroiding did not perform as well and the data was more difficult to interpret.

Centroiding also has a tendency to combine two or more more XIC into a single XIC. These combinations are very difficult for algorithms to recover (see Figure 13). Whether a signal like the one seen in Figure 13 should have been included in the evaluation to begin with is debatable.

Another interesting thing to note is the difference in performance between MatchedFilter and CentWave, as CentWave was developed as a high mass accuracy alternative to MatchedFilter. MatchedFilter uses Gaussian shape fitting to filter potential peaks, while CentWave uses wavelets to model peak shapes, and so the data in UPS2 is likely more amenable to peak fitting via Gaussians rather than wavelets.

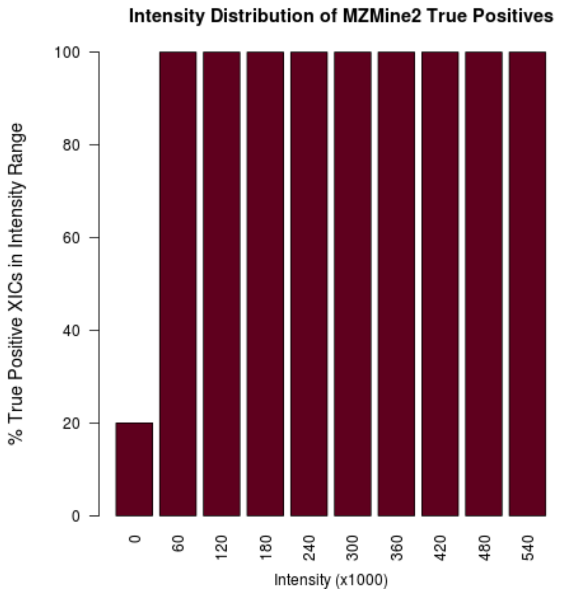

Figure 17: Intensity of MZMine2 true positives. Each bin covers a range of intensity and the height represents the percentage of true positive XIC, at that intensity.

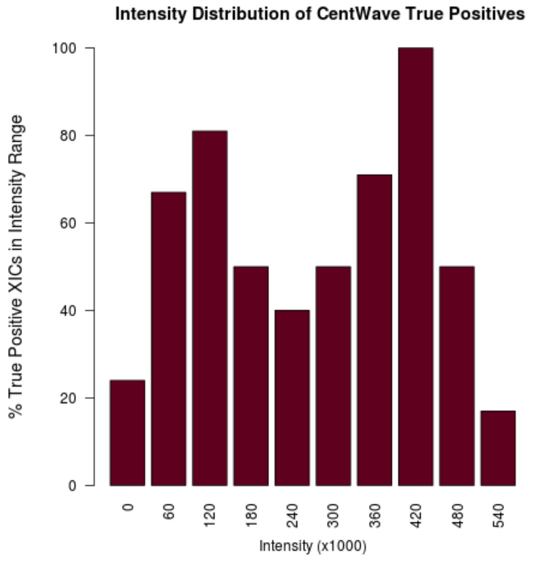

Figure 18. Intensity of CentWave true positives. Each bin covers a range of intensity and the height represents the percentage of true positive XIC, at that intensity.

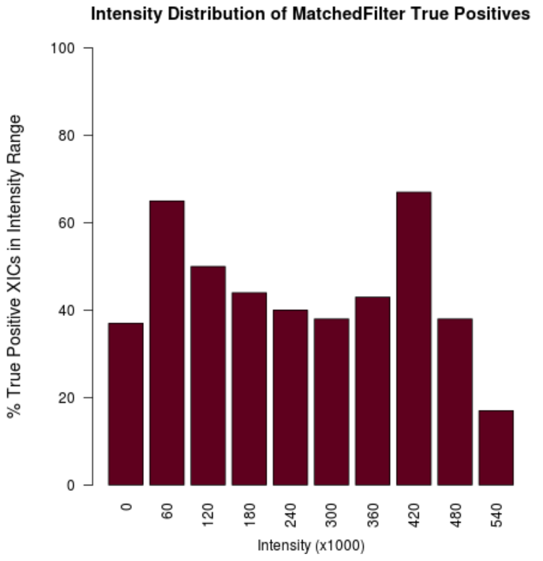

Figure 19. Intensity of MatchedFilter true positives. Each bin covers a range of intensity and the height represents the percentage of true positive XIC, at that intensity.

The performance of the algorithms across ascending ranges of intensity for all XIC found to be true positives is shown in Figures 15-19. Figure 14 shows the stratification of XIC in the hand curated data by intensity as a baseline for comparison.

Figure 13 shows the intensity ranges of manually-curated XIC. Figure 15 shows the intensity ranges of true positives for MaxQuant; because MaxQuant is not for use on centroided data, Figure 15 shows that MaxQuant did not recover many of the XIC with the highest intensities and its performance on centroided data is unpredictable and not similar to the other algorithms' performance.

Figures 15 and 16, however, show that MZMine2 and Massifquant had fairly similar performance. They both recovered a high number of XIC and also, the total number of points in each intensity range is higher than for the manually-curated XIC. This means that they both had a tendency to cluster more points into XIC than were chosen to be clustered during manual curation.

CentWave and MatchedFilter also have similar intensity distributions, as seen in Figures 18 and 19. Both algorithms have a tendency to cluster less points in each XIC than were clustered during manual segmentation, so the total number of points in each intensity range is generally less than is seen in the intensity distribution of the manually-curated XIC.

There are several limitations to our approach. The primary limitation is that this experiment was performed with a single dataset—this being the only quantitative, manually-curated data set available to date. As other data sets are made available, this analysis can be extended and the results generalized. Future work will include performing this evaluation on additional files in order to get a representation of performance on profile data, data of higher/lower resolution, rate of MS/MS, and varying degrees of data complexity.

Associated Content Available

Table S1 – all results from parameter grid search for MatchedFilter.
Table S2 – all results from parameter grid search for CentWave.
Table S3 – all results from parameter grid search for Massifquant.
Table S4 – all results from parameter grid search for MZMine2.
evaluationMetrics.rb – code used to measure performance of each result.

Author contributions

A.T. conducted experiments, and interpreted results. R.S. conceived of, designed, supervised the study. Both authors wrote code and contributed to the manuscript.

Conflict of interest

Authors declare no conflict of interest.

Funding

This research was supported by NSF grant number 366208 to R.S.

References

1. MacLean, B., Tomazela, D. M., Shulman, N., Chambers, M., Finney, G. L., Frewen, B., … MacCoss, M. J. (2010). Skyline: an open source document editor for creating and analyzing targeted proteomics experiments. *Bioinformatics (Oxford, England)*, *26*(7), 966–968. doi:10.1093/bioinformatics/btq054

2. Röst, H., Rosenberger, G., Navarro, P. *et al.* OpenSWATH enables automated, targeted analysis of data-independent acquisition MS data. *Nat Biotechnol* 32, 219–223 (2014) doi:10.1038/nbt.2841

3. Bruderer, R., Bernhardt, O. M., Gandhi, T., Xuan, Y., Sondermann, J., Schmidt, M., … Reiter, L. (2017). Optimization of Experimental Parameters in Data-Independent Mass Spectrometry Significantly Increases Depth and Reproducibility of Results. *Molecular & cellular proteomics : MCP*, *16*(12), 2296–2309. doi:10.1074/mcp.RA117.000314

4. Teleman, J.; Chawade, A.; Sandin, M.; Levander, F.; Malmström, J. Dinosaur: A Refined Open-Source Peptide MS Feature Detector. *Journal of Proteome Research,* 2016, *15* (7), 2143-2151. DOI: 10.1021/acs.jproteome.6b00016

5. Smith, R.; Ventura, D.; Prince, J. Controlling for confounding variables in MS-omics protocol: why modularity matters. *Briefings in Bioinformatics.* 2014, 15:5, 768-770.

6. Smith, R.; Ventura, D.; Prince, J. Novel algorithms and the benefits of comparative validation. *Bioinformatics.* 2013, 29:12, 1583 – 1585.

7. Rami Al Shweiki, M.; Mönchgesang, S.; Majovsky, P.; Thieme, D.; Trutschel, D.; Hoehnwarter, W. Assessment of Label-Free Quantification in Discovery Proteomics and Impact of Technological Factors and Natural Variability of Protein Abundance. *Journal of Proteome Research* 2017 *16* (4), 1410-1424. DOI: 10.1021/acs.jproteome.6b00645

8. Navarro, P.; Kuharev, J.; Gillet, L.; Bernhardt, O.; MacLean, B.; Röst, H.; Tate, S.; Tsou, C.; Reiter, L.; Distler, U.; Rosenberger, G.; Perez-Riverol, Y.; Nesvizhskii, A.; Aebersold, R.; Tenzer, S. A multicenter study benchmarks software tools for label-free proteome quantification. *Nat. Biotechnol.* 2016, 34(11):1130-1136. DOI: 10.1038/nbt.3685

9. Henning, J.; Tostengard, A.; Smith, R. A Peptide-Level Fully Annotated Data Set for Quantitative Evaluation of Precursor-Aware Mass Spectrometry Data Processing Algorithms. *J. Proteome Res.* 2019, 18:1, 392-398.

10. Conley, C.; Smith, R.; Torgrip, R.; Taylor, R.; Tautenhahn, R.; Prince, J. Massifquant: open-source Kalman filter-based XCMS isotope trace feature detection. *Bioinformatics.* 2014, 30:18, 2636-2643.

11. Cox, J.; Matthias, M. MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nature Biotechnology.* 2008, 26, 1367-1372.

12. Tautenhahn, R.; Bottcher, C.; Neumann, S. Highly sensitive feature detection for high resolution LC/MS. *BMC Bioinformatics.* 2008, 9:1, 504.

13. Pluskal, T.; Castillo, S.; Villar-Briones, A.; Oresic, M. MZMine2: Modular framework for processing, visualizing, and analyzing mass spectrometry-based molecular profile data. *BMC Bioinformatics.* 2010, 11, 395.

14. Smith, C.; Want, E.; O'Maille, G.; Abagyan, R.; Siuzdak, G. XCMS: Processing Mass Spectrometry Data for Metabolite Profiling Using Nonlinear Peak Alignment, Matching, and Identification. *Anal. Chem.* 2008, 78, 779-787.

15. Sturm, M.; Bertsch A.; Gröpl, C.; Hildebrandt A.; Hussong, R.; Lange, E.; Pfeifer, N.; Schulz-Trieglaff, O.; Zerck, A.; Reinert, K.; Kohlbacher, O. OpenMS–an open-source software framework for mass spectrometry. *BMC Bioinformatics*. 2008, 9:1, 163.

16. Smith, C. LC/MS Preprocessing and Analysis with XCMS. LC/MS Preprocessing and Analysis with XCMS. 2010. Retrieved from

https://www.bioconductor.org/packages//2.7/bioc/vignettes/xcms/inst/doc/xcmsPreprocess.pdf

17. Tsou, C.; Avtonoov, D.; Larsen, B.; Tucholska, M.; Choi, H.; Gingras, A.; Nesvizhskii, A. DIA-Umpire: comprehensive computational framework for data-independent acquisition proteomics. *Nature Methods.* 2015, 12:3, 258-264: *Nature Research*.

18. Libiseller, G.; Dvorzak, M.; Kleb, U.; Gander, E.; Eisenberg, T.; Madeo, F.; Neumann, S.; Trausinger, G.; Sinner, F.; Pieber, T.; Magnes, C. IPO: a tool for automated optimization of XCMS parameters. *BMC Bioinformatics.* 2015, 16, 118.

19. Rosen, J.; Handy, K.; Gillan, A.; Smith, R. JS-MS: A cross-platform, modular JavaScript viewer for mass spectrometry signals. *BMC Bioinformatics.* 2017, 18:1.

An Evaluation of Noise Reduction Algorithms for Liquid Chromatography Mass Spectrometry Data

Annika Tostengard and Rob Smith

Abstract

Mass spectrometry (MS) is used in analysis of chemical samples to identify both what molecules are present and their quantities. This analytical technique has applications in many fields, from pharmacology to space exploration. Its impacts on medicine are particularly significant, since MS aids in the identification of molecules associated with disease; for instance, in proteomics, MS allows researchers to identify proteins that are associated with autoimmune disorders, cancers, and other conditions. Since the applications are so wide-ranging and the tool is ubiquitous across so many fields, it is critical that the analytical methods used to collect data are sound.

Data analysis in MS is challenging. Experiments produce massive amounts of raw data that need to be processed algorithmically in order to generate interpretable results in a process known as feature detection, which is tasked with distinguishing signals associated with the chemical sample being analyzed from signals associated with background noise, which can be introduced through multiple mechanisms. We examine two popular algorithms, the COmponent Detection Algorithm (CODA) and adaptive iteratively reweighted Penalized Least Squares (airPLS), for reducing background noise and compare them to the results of feature detection alone. Due to weaknesses inherent in the implementation of these algorithms, both algorithms eliminate data identified by feature detection alone as significant.

Introduction

Mass spectrometry is a quintessential tool for a variety of domains, including proteomics, lipidomics, and metabolomics. Liquid chromatography coupled to mass spectrometry (LC-MS) has emerged as a ubiquitous configuration for many experimental objectives. In LC-MS, sample analytes are separated by LC, ionized, and analyzed by the mass spectrometer. The objective of mass spectrometry experiments is to measure the presence, absence, or abundance of one or more specific molecules. Computational processing is used in many forms and to different extents to achieve this objective.

While various specific approaches for and across proteomics, lipidomics, and metabolomics exist, with some approaches being specific to the molecule type and others general across all types, these approaches require the computational processing of raw mass spectrometer output to make the results interpretable to researchers. Multiple algorithms have been produced with the goal of rendering raw data human-interpretable. Shared between many of these approaches is the need to isolate specific portions of the raw data. Isotopic envelopes are signal groups comprised of individual isotopic masses, known as features, that correspond to one or more compounds at a given mass-to-charge (m/z) and retention time (RT), and are of particular interest. Isotopic envelopes can be used to measure the abundance of the compound(s), as the abundance is proportional to the summed intensities of all points comprising the envelope; these envelopes map to MS^n events which provide information for molecule identification and reduces data from millions of points to tens of thousands or fewer envelopes.

More specifically, feature detection attempts to directly extract isotopic envelopes from the raw data. This is most often a two-stage process in which individual isotopes are identified and then grouped into isotopic envelopes. Historically, due to low mass spectrometer resolution, feature detection algorithms were not effective in directly extracting isotopic envelopes from raw data. However, recent innovations have shown vast improvement over prior algorithms.

Another approach treats isotopic envelope detection as a modular process with many different configurations possible [1]. In this approach, prior to feature detection, the raw data is subjected to the ordered application of several processes that are treated as independent; this modular process often includes noise removal. Noise is generally present in two forms: background noise of very low intensity signals, and noise peaks of moderately low intensity that resemble actual features. Both are generally introduced through the mobile phase of the chemical matrix in which the sample of interest is introduced to the MS instrument. Noise removal involves identifying noisy regions and then either removing the regions entirely or reducing the intensity of the region by a certain amount.

To our knowledge, the efficacy of current noise removal algorithms has never been quantitatively evaluated. In this manuscript, we investigate whether noise reduction is necessary or helpful given modern feature detection algorithms and high resolution mass spectrometry.

Algorithms

GridMass

GridMass is a feature detection algorithm included in the popular open source mass spectrometry analysis software MZMine2. GridMass employs probes assigned to a rectangular area of the entire chromatogram to find local maxima; probes that converge on the same feature are used to provide an estimation of feature boundaries. [2]

GridMass is included here because it outperformed several other popular feature detection algorithms in an evaluation on a published ground truth MS dataset [3].

XFlow

XFlow is a recently published feature detection algorithm that extracts ion chromatograms from MS1 LC-MS data without using any parameters. It can be used on both profile or centroided data and is agnostic in regard to both resolution and instrument. XFlow identifies features by identifying points which are local maxima; it then assigns nearby points a confidence score based on both intensity and distance from the local maxima. Points with the highest confidences are then assigned to the same feature. [3]

As with GridMass, XFlow is included here because it outperformed several popular feature detection algorithms when evaluated on a published ground truth MS dataset [3].

CODA

The component detection algorithm is one that aims to be useful for noisy data with a high background level. Furthermore, it does so without transforming the original data and instead selects high-quality regions within a dataset. It does this by first smoothing the data, then calculating the average intensity across each chromatogram. The average intensity is subtracted from the smoothed intensities, after which a normalized output of the difference between the average and smoothed data is calculated to determine the similarity threshold against which all the spectra can be compared. Only those signals whose similarity values are above the threshold value are retained.

The theory behind CODA is that in regions of data containing noise spikes, the smoothed data will differ substantially from the original data; therefore, a high similarity between the smoothed and original data indicates high-quality peaks that represent pertinent information, while a low similarity index indicates a noise peak. When comparing data that has been both smoothed and averaged with the original data, a high-quality peak has a low mean value in comparison to a poor-quality peak. As when comparing the smoothed and original data, the smoothed, averaged data that has a high similarity to the original data indicates a quality peak. The two different measurements are combined into a single similarity index, called a “mass quality index” (MCQ). In the case that a noise peak has a high similarity in either case, the combination of the two measurements will indicate that it is a poor-quality peak. Again, only regions which are greater than a certain MCQ threshold will be considered high-quality. Data regions that do not meet this threshold are discarded entirely. [4]

airPLS

Penalized least squares has long been used as a smoothing technique for spectroscopic data. However, in 2005, it was suggested that it be used for a number of other purposes, including baseline correction [5]. A well-known method of least squares for baseline correction uses asymmetry as a parameter which needs to be optimized, but it was subsequently shown to have a tendency to produce negative regions in complex data where things like overlapping signals occur frequently [6, 7] and required that feature detection be performed prior to smoothing. airPLS, on the other hand, is based on asymmetric least squares but does not require feature detection. Instead, airPLS adds a penalty item to control smoothing. Eilers’ method also uses the same asymmetry parameter across the entire baseline, but adaptive iteratively reweighted least squares introduced a new way to calculate weights based on the difference between the previously fitted baseline and the original signals. [8]

Methods

The fundamental signal unit in MS is the extracted ion chromatogram, also known as a feature. Features represent a molecule of a certain mass that occurs at a certain time as a sample passes through a mass spectrometer. Isotopic masses are then grouped into isotopic envelopes that represent all isotopic masses for a particular molecule. Therefore, quality feature detection is the ultimate goal of MS data processing algorithms. An example of extracting features from background noise is shown in Figure 1. Algorithms tasked with distinguishing masses of interest from noise are known as feature detection algorithms. To make the feature detection process more accurate, many preprocessing algorithms are used to reduce dataset size and make data more scrutable. One important set of preprocessing algorithms are noise reduction algorithms, which often both reduce noise and smooth data.

Noise reduction is traditionally applied as an independent process that occurs before feature detection. While there is value in treating every subproblem in a complex analysis as independent, in order to maximize the overall result [9], a rational argument can be made that the separately-treated problems of noise reduction and feature detection are really just simplified interdependent approximations of the 3D feature finding problem. It can be argued that treating these problems separately is a lossy approach compared to leveraging all available information into 3D feature finding.

For example, when feature detection occurs on data that has already been separated into independent bins, then denoised and possibly smoothed, it has gone through at least two successive lossy processes, none of which was aware of the assumptions made or information lost in each step [1].

Denoising and smoothing are typically used to remove or minimize the abundances of points that do not pertain to an isotopic envelope [1]. Typically, some filter is applied to the data to attenuate the intensities of points not pertaining to an isotopic envelope while (hopefully) minimally affecting points pertaining to isotopic envelopes. Filters include the Savitzky-Golay [10], some variation of wavelet [11], moving average [12], Gaussian [13], or kernel density [14].

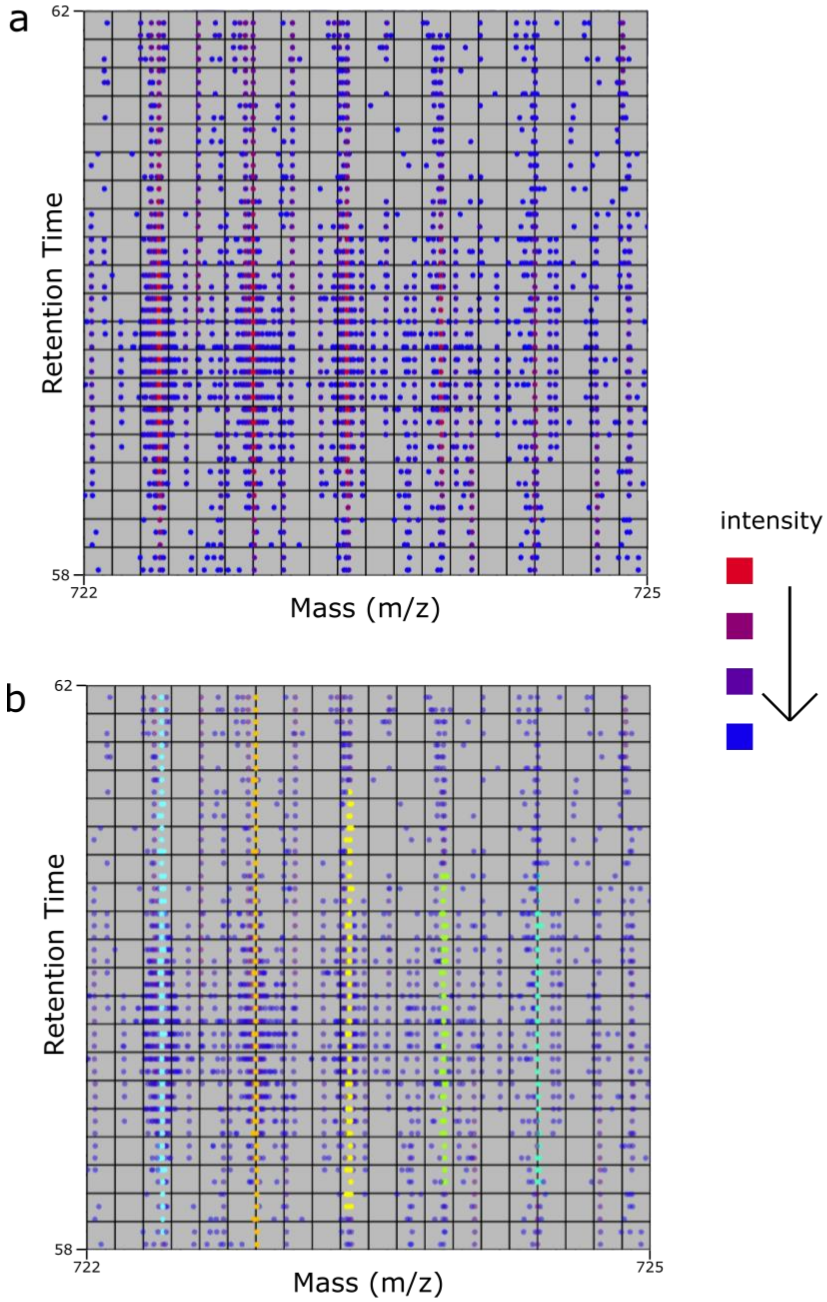

Figure 1. An example of detected features, where color indicates intensity. Panel a shows an isotopic envelope with five features before they have been extracted from background noise. Panel b shows extracted features.

Treating scans as informationally independent is convenient when performing noise reduction, as it considerably reduces dataset size. However, the size savings comes at the cost of ignoring the majority of relevant information contained in the run. For example, both noise reduction algorithms included in this evaluation bin the entire chromatogram by both mass (m/z) and retention time. However, isotopic envelopes can occur across two or more masses, and the individual isotopic masses that make up the envelope often occur across multiple scans; these algorithms therefore lose the opportunity to leverage mutual information among masses/scans. In addition, most noise reduction algorithms have parameters, which not only require setting and optimization, but also inherently impart the constraint that parameters that work well on some parts of an experimental output will necessarily work poorly on others. Total Ion Chromatogram (TIC) approaches, in which all masses in each scan are summed into a single representation (shown in Figure 2), are common to noise reduction and smoothing algorithms and discard much of the information contained in the three dimensional full run data [1].

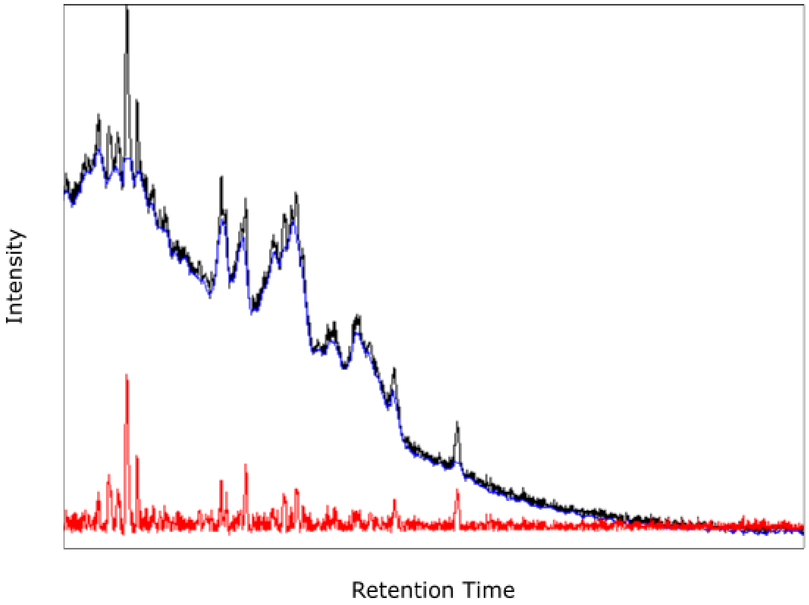

Figure 2. An example of Total Ion Chromatogram (TIC) noise reduction via intensity reduction of noisy regions.

To demonstrate the differences, strengths and weaknesses of the algorithms evaluated here, we include the changes made by each algorithm across the entire chromatogram, as well as find individual mass chromatograms in feature detection disagrees with noise reduction as to which data to retain or by how much the original data needs to be transformed.

Data

We used several files to compare the data processing of the noise reduction algorithms with the data processing of the feature detection algorithms. The files included are:

- a published UPS2 protein standard spiked into *E. coli* acquired on an AB Sciex TripleTOf 5600, which can be found in PRIDE repository PXD001587 under filename 18185_REP2_4pmol_UPS2_IDA_1.mzML and is hereafter referred to as UPS2;
- a published panel of seven cancer cell lines acquired on an Orbitrap Fusion Lumos, which can be found in PRIDE repository PXD008952 under filename 01_CPTAC_TMTS1-NC17_P_JHUZ_20180509_LUMOS.mzML and is hereafter referred to as LUMOS;
- a published green alga cell culture acquired on a Q Exactive, which can be found in Pride repository PXD003236 under filename

Cre_PLPGS_0_100_mix_FASP_fraction_FT_ph11.mzML and is hereafter referred to as FASP;

- a published baker's yeast protein sample acquired on an LTQ Orbitrap, which can be found in PRIDE repository PXD000792 under filename 000.mzML and is hereafter referred to as 000;
- a published mouse murine myoblast cell line acquired on an LTQ Orbitrap Elite, which can be found in PRIDE repository PXD000790 under filename OEII12347.mzML and is hereafter referred to as OEII;
- a set of three experimental replicates consisting of a commercial proteomics dynamic range standard (UPS2, sigma) analyzed using liquid chromatography MS with an Orbitrap FusionTM Lumos mass spectrometer. Publication of the data is in progress. This file is hereafter referred to as PRICE.

Parameters

GridMass

GridMass has seven parameters, four required and three optional. Required parameters include the minimum height threshold, an intensity value below which points will be ignored; width, a time parameter that determines the distance between probes in the retention time dimension; m/z tolerance, a mass parameter that determines the distance between probes in the m/z dimension; and the intensity similarity ratio, which detects features that have a similar intensity and mass. The three optional parameters include ignore times, a list of time ranges that will be ignored; smoothing time, the time over which the chromatogram will be smoothed via averaging; and smoothing m/z, the m/z range over which the chromatogram will be smoothed via averaging [2]. The default parameters achieved high performance in an evaluation including XFlow, and therefore the default parameters were used here as well.

CODA

CODA examines masses individually and discards data across entire masses that do not achieve a quality threshold. Here, we refer to masses that were not discarded as retained mass chromatograms. To determine the best CODA results, we used the number of retained chromatograms, with a higher number of retained chromatograms being preferred. The primary parameters for the CODA algorithm are window size, a rectangular window over which smoothing is applied, and mass quality index level (MCQ), the threshold chromatograms must meet to be considered high-quality. For each file, we also binned chromatograms by m/z using different bin sizes for comparison. The parameters tested are shown in Table 1; every combination of the parameters was evaluated.

The parameter setting which retained the highest number of chromatograms was then further compared to XFlow and GridMass in two ways: first, by summing the intensity of all points that were clustered into peaks by XFlow and GridMass, respectively, but were not included in any chromatograms retained by CODA, and second, by summing the intensity of all points that were included in any chromatograms retained by CODA that were not clustered into a peak by XFlow and GridMass, respectively. We also report the total intensity of all points that were retained by CODA and by feature detection.

Table 1. Parameters Tested for CODA

window size	3	5	
MCQ	0.69	0.79	0.89
bin size	0.5	1	10

airPLS

The primary setting in airPLS is lambda, which dictates how smoothed the result is. As with CODA, we binned the data by m/z to examine the effects of binning. The parameters tested are shown in Table 2. Since airPLS returns a modified chromatogram and does not discard any regions of data, there is no objective measure by which to judge whether a given parameter setting retains more information; therefore, we report the intensity differences across all settings, and between feature detection and airPLS in Figures 5 and 6. Figures 5 and 6 show the intensity differences between XFlow and GridMass by summing the total intensity that was retained in the modified airPLS chromatogram that did not overlap with the signals reported by both XFlow and GridMass, and also the total intensity reported by XFlow and GridMass that was not retained in the airPLS chromatogram. We also report the total intensity of the points that were reported by airPLS and feature detection.

Table 2. Parameters Tested for airPLS

lambda	10e3	10e5	10e7	10e9	10e11
bin size	0.5	1	10		

Results

CODA

The number of retained chromatograms for each file at each bin size are shown in Figure 3 below. CODA retained the highest number of chromatograms using the same parameter settings for all files; these settings are a bin size of 0.5 m/z, a window size of 3 and an MCQ of 0.69. The total intensity comparison results are shown in Figures 5 and 6.

When compared to the chromatograms obtained by feature detection with XFlow, there is a significant difference in what data is retained, suggesting that CODA is discarding pertinent information that may be experimentally significant. Features detected by XFlow and GridMass but discarded by CODA are shown in Figures 7 and 8.

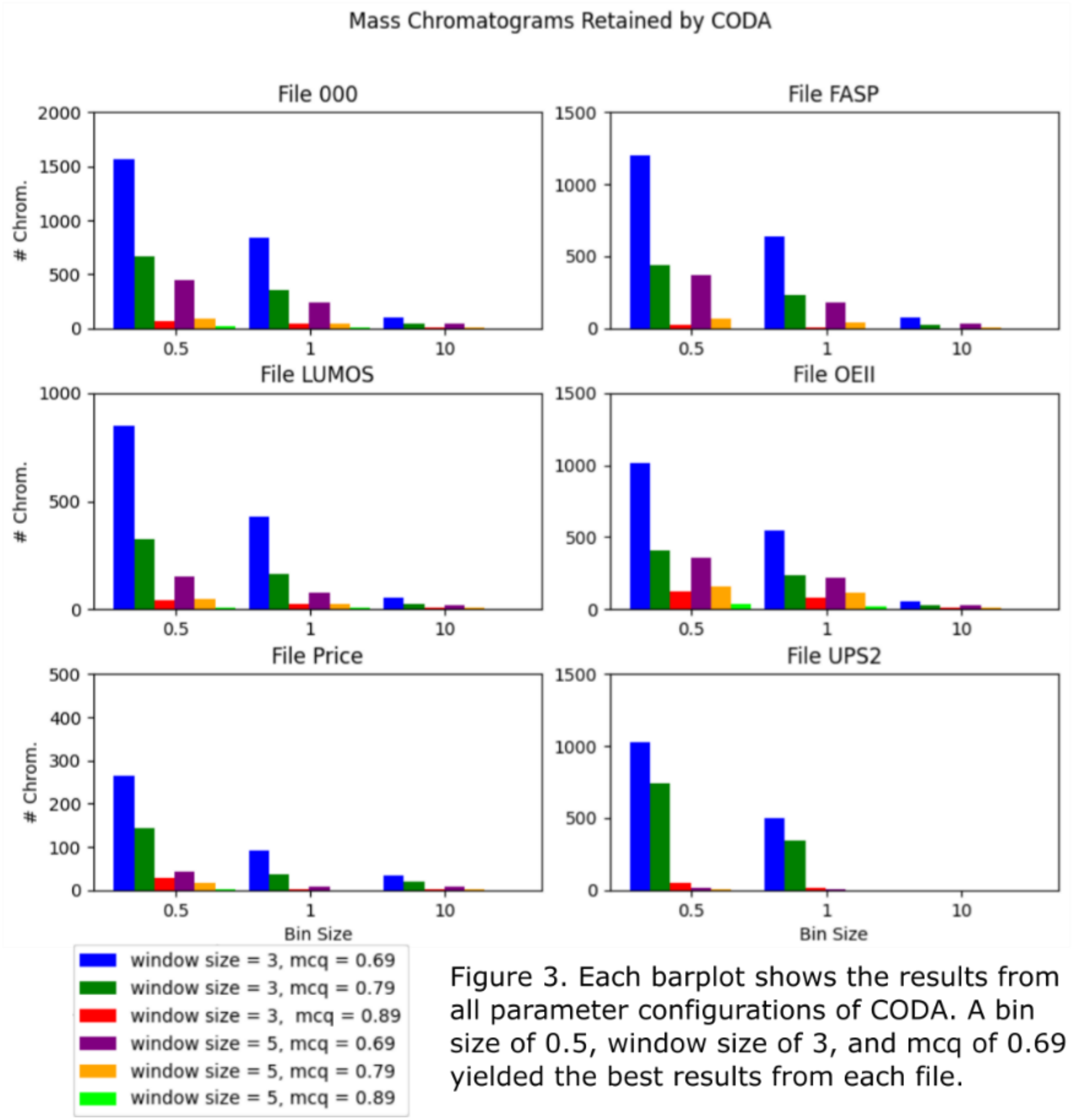

Figure 3. Each barplot shows the results from all parameter configurations of CODA. A bin size of 0.5, window size of 3, and mcq of 0.69 yielded the best results from each file.

airPLS

The total intensity differences for airPLS are shown in the tables below. These tables include the total intensity differences between airPLS and both XFlow and GridMass, respectively.

As with CODA, airPLS discarded features detected by XFlow and GridMass. airPLS significantly reduced the intensity of many sections of data so that features which were apparent in the original data were no longer discernible, shown in Figures 7 and 8.

Comparisons Between Algorithms

In order to demonstrate the changes made across an entire chromatogram by each algorithm and to get a sense of individual algorithm performance, Figure 4 made heat maps of the UPS2 data, showing both the original, unadulterated data and the chromatogram after it had been processed by each algorithm separately, shown in Figure 4.

We have also included the total intensity difference results from all files. The GridMass results are shown in Figure 5 and the XFlow results are shown in Figure 6. These figures display the total intensity that was reported by the noise reduction algorithms that was not reported by the feature detection algorithms, the total intensity that the feature detection algorithms reported that the noise reduction algorithms did not, and the total intensity that both reported.

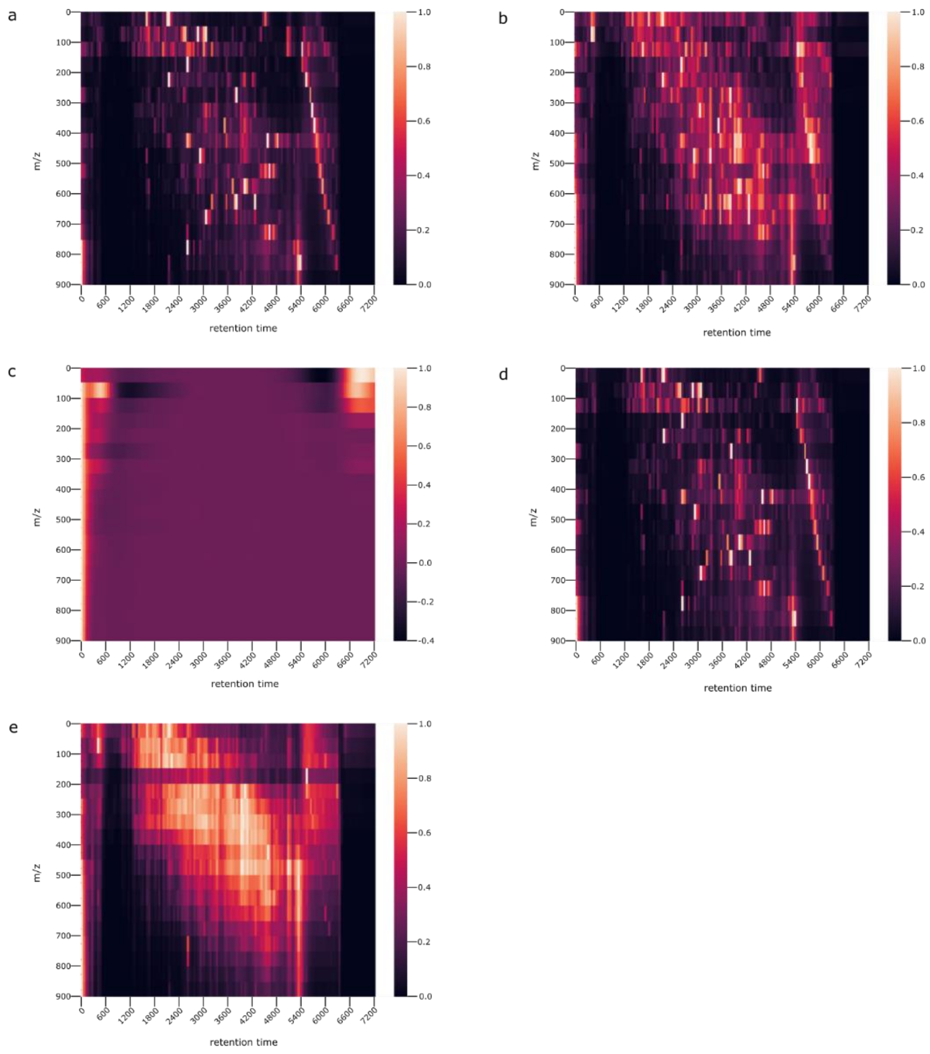

Figure 4. Heat maps depicting the entire UPS2 chromatogram after algorithmic processing. The original chromatogram is shown in a). Algorithms include b) CODA, c) airPLS, d) MZMine2, and e) XFlow. The data has been normalized and clustered into bins of 50x50 m/z.

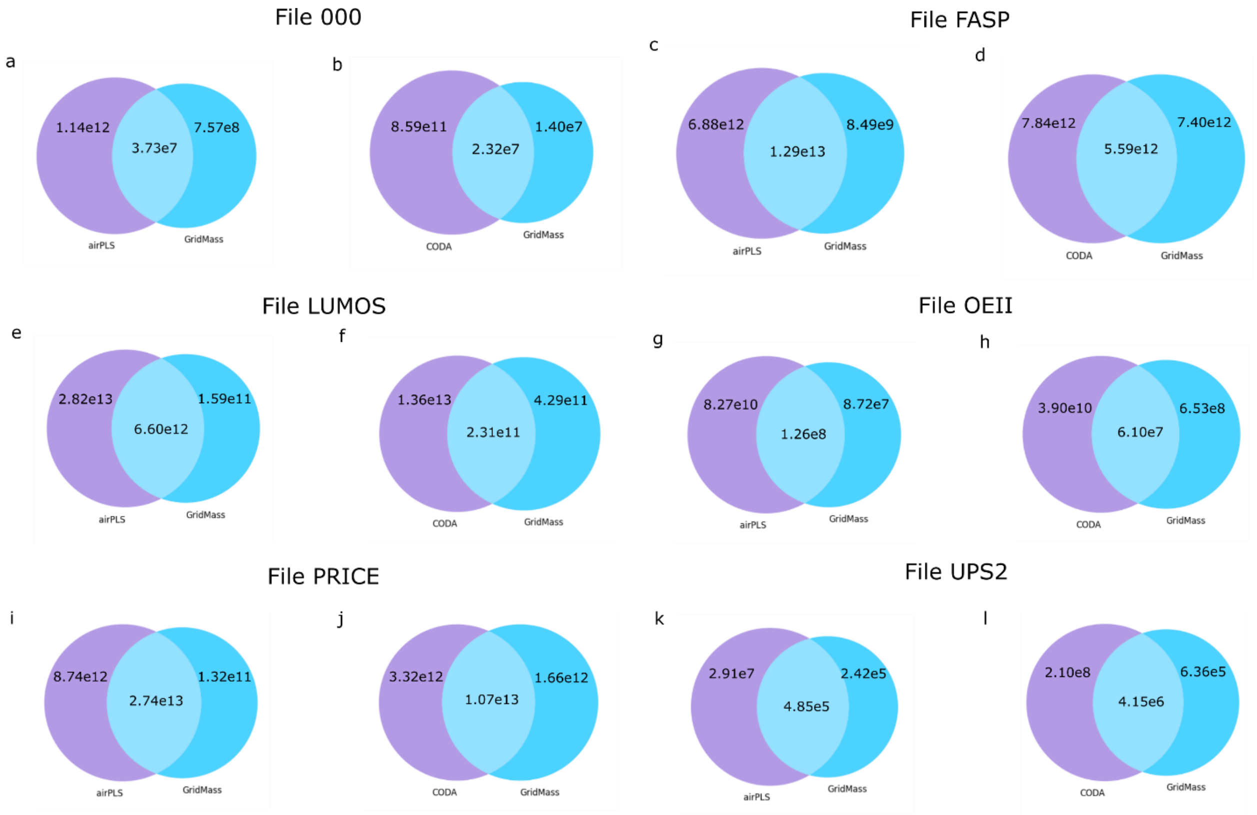

Figure 5. Venn diagrams displaying the total intensity of all points that were clustered into peaks by GridMass but were not included in any chromatogram retained by the respective noise reduction algorithm (blue) and the total intensityof all points that were included in any chromatogram retained by the noise reduction algorithm but were not clustered into a peak by GridMass (purple). The parameter combination with the best performance was chosen for this comparison, where the best parameter combination for CODA is the parameter combination with the highest number of retained chromatograms and the best parameter combination for airPLS is the one in which the highest intensity is shared by the two algorithms. These parameter combinations are: a) lambda: 10e3, bin size: 1 m/z; b) window size: 3, mcq: 0.69, binsize: 0.5 m/z; c) lambda: 10e5, bin size: 10 m/z; d) window size: 3, mcq: 0.69, bin size: 0.5 m/z; e) lambda: 10e7, bin size: 10 m/z; f) window size: 3, mcq: 0.69, bin size: 0.5 m/z; g) lambda: 10e9, bin size: 1 m/z; h) window size: 3, mcq: 0.69, bin size: 0.5 m/z; i) lambda: 10e11, bin size: 10 m/z; j) window size: 3, mcq: 0.69, bin size: 0.5 m/z; k) lambda: 10e7, bin size: 1 m/z; l) window size: 3, mcq: 0.69, bin size: 0.5 m/z.

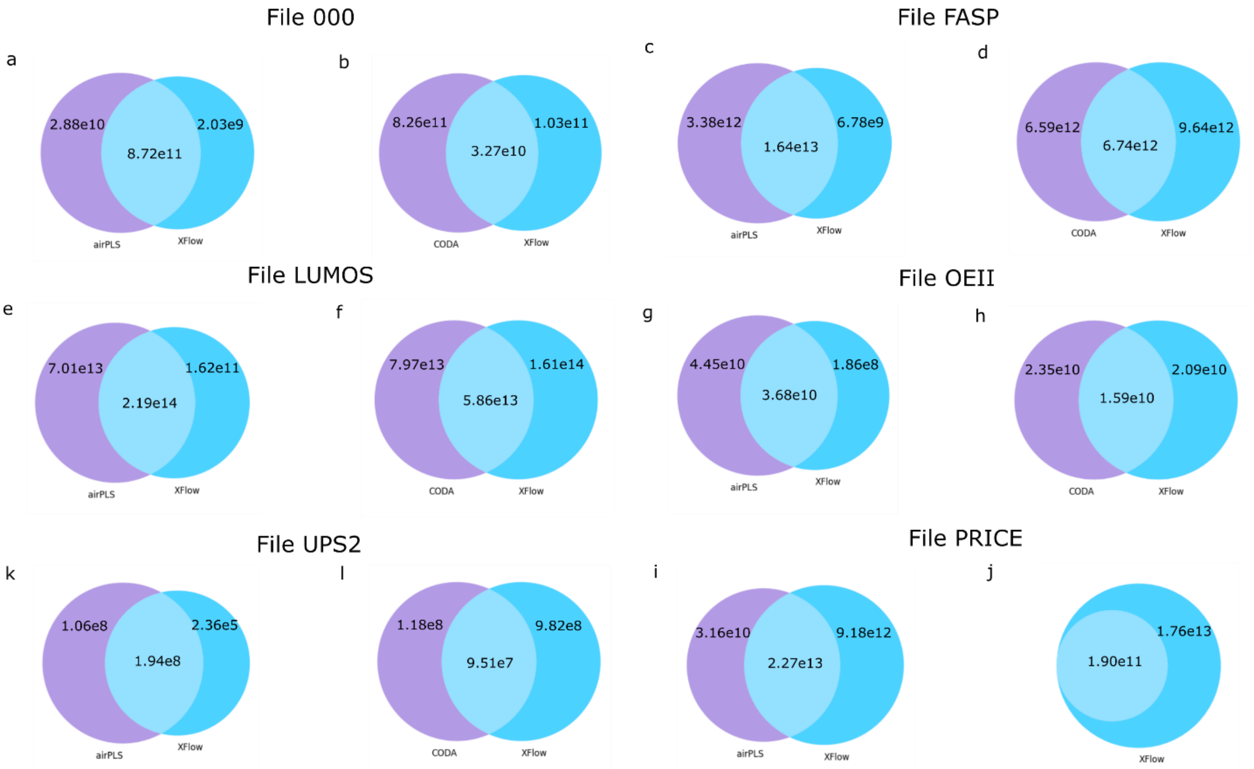

Figure 6. Venn diagrams displaying the total intensity of all points that were clustered into peaks by XFlow but were not included in any chromatogram retained by the respective noise reduction algorithm (blue) and the total intensity of all points that were included in any chromatogram retained by the noise reduction algorithm but were not clustered into a peak by XFlow (purple). The parameter combination with the best performance was chosen for this comparison, where the best parameter combination for CODA is the parameter combination with the highest number of retained chromatograms and the best parameter combination for airPLS is the one in which the highest intensity is shared by the two algorithms. These parameter combinations are: a) lambda: 10e9, bin size: 10 m/z; b) window size: 3, mcq: 0.69, bin size: 0.5 m/z; c) lambda: 10e5, bin size: 10 m/z; d) window size: 3, mcq: 0.69; bin size: 0.5 m/z; e) lambda: 10e9, bin size = 0.5 m/z; f) window size: 3, mcq: 0.69, bin size: 0.5 m/z; g) lambda: 10e9, bin size: 10 m/z; h) window size: 3, mcq: 0.69, bin size: 0.5 m/z; i) lambda: 10e11, bin size: 10 m/z; j) window size: 3, mcq: 0.69, bin size: 0.5 m/z; k) lambda: 10e9, bin size: 0.5 m/z; l) window size: 3, mcq: 0.69, bin size: 0.5 m/z.

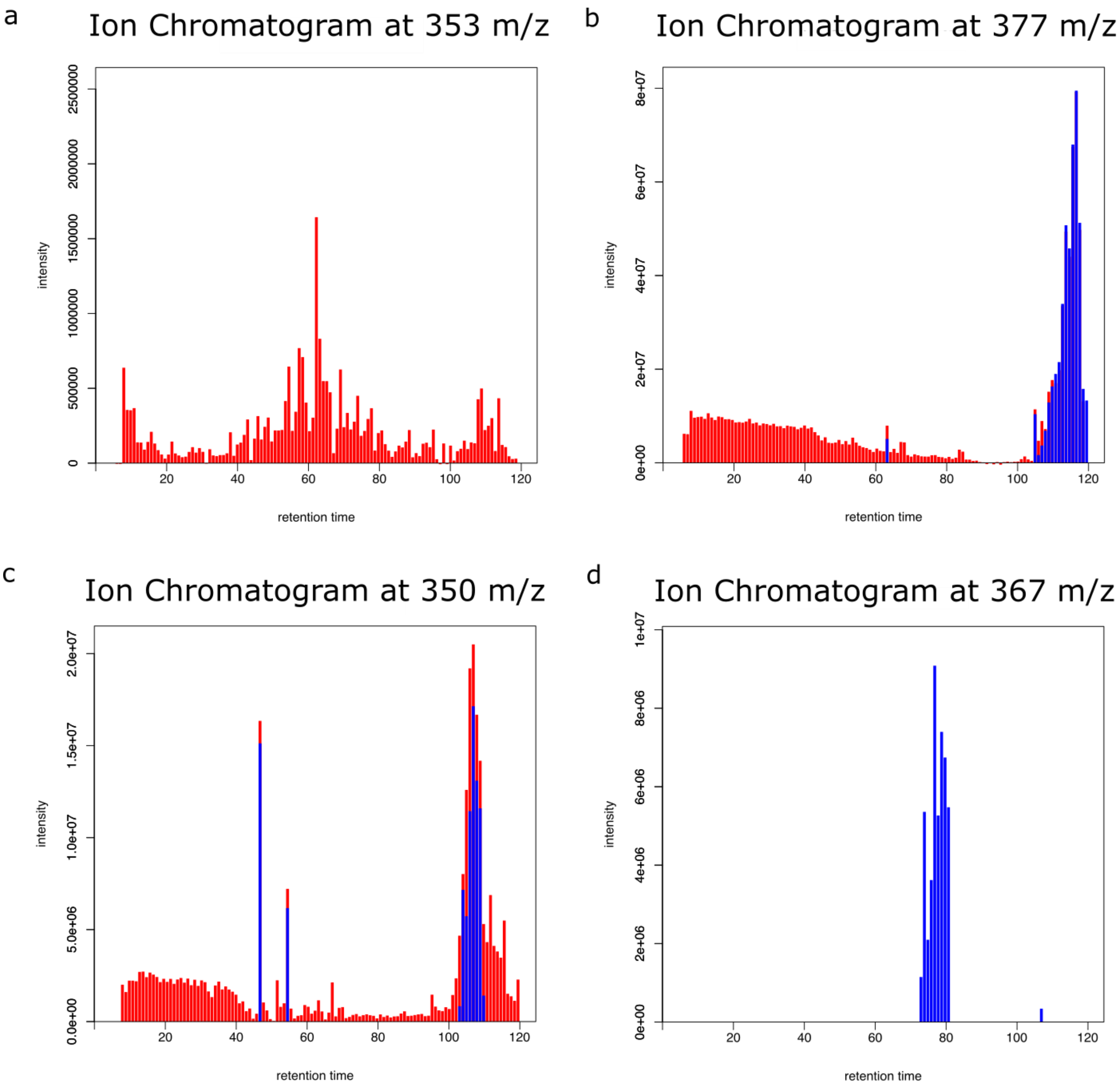

Figure 7. Bar plots of individual mass chromatograms from File 000 that have been processed by XFlow and noise reduction algorithms. A mass chromatogram in which XFlow did not identify any peaks, but in which airPLS retained data is shown in a). An example in which peaks that were identified by XFlow were not retained by airPLS is shown in b). The parameter settings for airPLS in a) and b) are lambda = 10e7 and bin size = 0.5 m/z. A mass chromatogram in which CODA retained data that was not included in peaks found by XFlow is shown in c). For other masses, CODA did not retain the chromatogram, while XFlow found meaningful peaks, shown in d). The parameter settings used for CODA in c) and d) are window size = 3, mcq = 0.69, and bin size = 0.5 m/z.

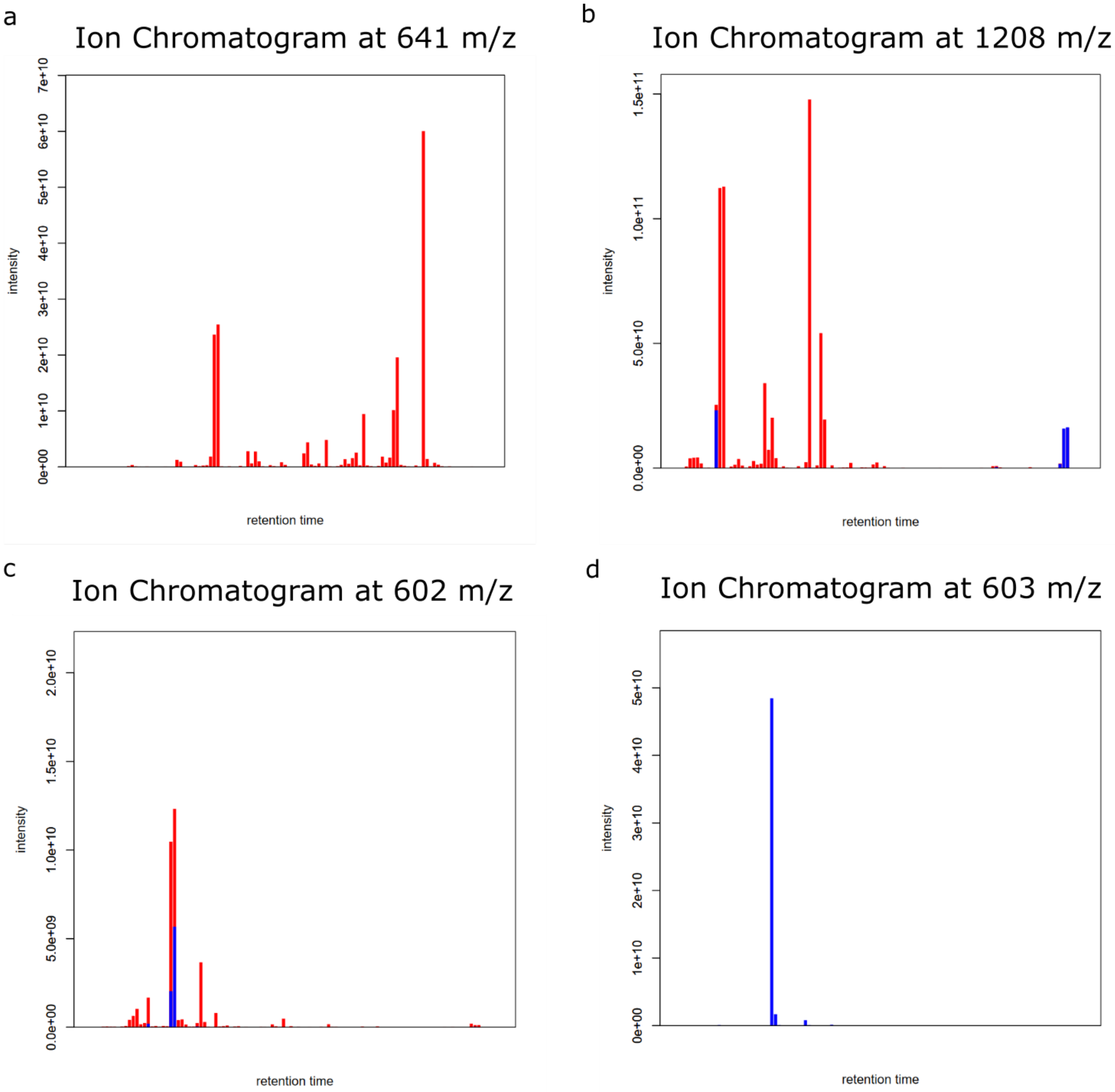

Figure 8. Bar plots of individual mass chromatograms from File LUMOS that have been processed by GridMass and noise reduction algorithms. A mass chromatogram in which GridMass did not identify any peaks, but in which airPLS retained data is shown in a). An example in which peaks that were identified by GridMass were not retained by airPLS is shown in b).The parameter settings for airPLS in a) and b) are lambda = 10e7 and bin size = 1 m/z. A mass chromatogram in which CODA retained data that was not included in peaks found by GridMass are shown in c). CODA did not retain an entire chromatogram that was found to contain meaningful peaks by GridMass in d), while in e), CODA retained the chromatogram but did not include the intensity included in the peak found by GridMass. The parameter settings used CODA in c), d) and e) are window size = 3, mcq = 0.69 and bin size = 0.5 m/z.

Discussion

There are several drawbacks from pre-processing raw data prior to feature detection. Among all approaches, information is lost. Preprocessing algorithms necessarily consider only a subset of the feature detection problem, and as such are likely to repeatedly mischaracterize a point or collection of points as noise in situations where a good feature detection algorithm might leverage mutual information to extract information from the same data subset. For example, while the noise reduction algorithms here treat each scan as an independent data source, 3D feature detection can make point clustering decisions using the mutual information across multiple scans.

Among all approaches, bias is imputed. All algorithms impute bias into a problem since solutions rely on certain assumptions about the data. The more algorithms you apply, the more bias you impute. Bias is not necessarily bad, but the more of it you have, the more likely it will diminish results. The more you subdivide a problem, the more likely you are producing a suboptimal result.

With all approaches, you compound and propagate error. Most mass spectrometry experimental results come with statistical measures of confidence and error. However, modular approaches typically have no mechanism for feeding forward confidence or error. Typically, subsequent processes assume that previous processes were completely correct. Therefore, confidence is likely to be overreported, and error underreported.

In most mass spectrometry experiments, feature detection is a necessary step. While there is always the possibility that some identified peaks are noise peaks, most noise peaks are present with low intensity [15]. As seen in Figures 7 and 8, the features that were discarded by CODA or significantly reduced by airPLS are high intensity, except for Figure 8 panel b (suggesting that in this case a noise peak was removed by airPLS). Furthermore, looking at the high level of intensity in Figures 6 and 7 that was reported by feature detection but not by noise reduction makes it highly likely that not all the features removed by noise reduction are noise peaks. The total intensity retained by feature detection that was discarded by noise reduction is, for most files, within a few orders of magnitude of the total intensity retained by noise reduction that was not clustered into features by feature detection algorithms. One would expect noise reduction to retain a higher total intensity than peak detection because by its nature, it is meant to simply reduce the intensity all the existing data, or in the case of CODA, retain all the noise in peaks it finds to be high-quality. Feature detection, by its nature, is meant to retain a much smaller subset of the data. The fact that the total intensities retained by noise reduction and feature detection are so similar is highly suggestive that a great deal of meaningful data is being discarded.

Furthermore, CODA in particular is a very popular algorithm, and is included in popular mass spectrometry analysis software suites, such as OpenChrom, the Waters Empower Software, the PerkinElmer TurboMass software, and the ACD Labs Mass Processor software. Its treatment of masses as independent information slices and subsequent removal of entire masses across all scans reduces the ability of feature detection algorithms to find isotopic envelopes, as these can occur across masses. Both GridMass and XFlow found peaks in chromatograms that were discarded by CODA and suggests that binning the data in this way results in data loss.

airPLS makes less dramatic changes, and it can be seen in Figure 4 that, on average, it does not significantly modify the data. However, in the case of individual features, the changes that it is making are still reducing the intensity of features detected by XFlow and GridMass such that they become undetectable. Figures 6 and 7 demonstrate that the total intensities found by feature detection that were not included in the total intensities of CODA and airPLS are within a single order of magnitude of each other for most files, suggesting that they have similar performance in terms of discarding features that would that been discoverable otherwise.

While feature detection by itself still runs the risk of detecting noise peaks, depending on desired experimental outcome such as resolution, the potential for data loss may not be worth the reduced noise. Additional attention to noise peak identification and mitigation is needed and it may be the case that improved feature detection is preferable to maintaining noise reduction as part of the MS data processing workflow.

References

[1] Podwojski, K., Eisenacher, M., Kohl, M., Turewicz, M., Meyer, H. E., Rahnenführer, J., & Stephan, C. (2010). Peek a Peak: A Glance at Statistics for Quantitative Label-Free Proteomics. *Expert Review of Proteomics, 7*(2), 249-261.

[2] Trevino, V., Yanez-Garza, I., Rodriguez-Lopez, C. E., Urrea-Lopez, R., Garza-Rodriguez, M., Barrera-Saldana, H., Tamez-Pena, J. G., Winkler, R., & Diaz de-la-Garza, R. (2015). GridMass: a fast two-dimensional feature detection method for LC/MS. *Journal of Mass Spectrometry, 50*(1), 165-174.

[3] Gutierrez, M., & Smith, R. (2002). XFlow: An Algorithm for Extracting Ion Chromatograms. *PLoS ONE, 15*(10).

[4] Windig, W., Phalp, M. J., & Payne, A. W. (1996). A Noise and Background Reduction Method for Component Detection in Liquid Chromatography/Mass Spectrometry. *Analytical Chemistry, 68*(20), 3602-3606.

[5] Eilers, P. H. C., & Boelens, H. F. M. (2005). Baseline Correction with Asymmetric Least Squares Smoothing. *Leiden University Medical Centre Report*.

[6] Zhang, Z., Chen, S., Liang, Y., Liu, Z., Zhang, Q., Ding, L., Ye, F., & Zhou, H. (2009). An intelligent background-correction algorithm for highly fluorescent samples in Raman spectroscopy. *Journal of Raman Spectroscopy, 41*(6), 659-669.

[7] Cobas, J. C., Bernstein, M. A., Mart-Pastor, M., & Tahoces, P. G. (2006). A new general-purpose fully automatic baseline correction procedure for 1D and 2D NMR data. *Journal of Magnetic Resonance, 183*(1), 145-151.

[8] Zhang, Z., Chen, S., & Liang, Y. (2010). Baseline Correction Using Adaptive Iteratively Reweighted Penalized Least Squares. *Analyst, 135*(5), 1138-1146.

[9] Smith, R., Ventura, D., & Prince, J. T. (2013). Controlling for confounding variables in MS-omics protocol: why modularity matters. *Briefings in Bioinformatics, 15*(5), 768-770.

[10] Wang, W., Zhou, H., Lin, H., Roy, S., Shaler, T. A., Hill, L. R., Norton, S., Kumar, P., Anderle, M., & Becker, C. H. (2003). Quantification of Proteins and Metabolites by Mass Spectrometry without Isotopic Labeling or Spiked Standards. *Analytical Chemistry, 75*(18), 4818-4826.

[11] Barclay, V. J., Bonner, R. F., & Hamilton, I. P. (1997). Application of Wavelet Transforms to Experimental Spectra: Smoothing, Denoising, and Data Set Compression. *Analytical Chemistry, 69*(1), 78-90.

[12] Radulovic, D., Jelveh, S., Ryu, S., Hamilton, T. G., Foss, E., Mao, Y., & Emili, A. (2004). Informatics Platform for Global Proteomic Profiling and Biomarker Discovery Using Liquid Chromatography – Tandem Mass Spectrometry. *Molecular & Cellular Proteomics, 3*(10), 984-997.

[13] Kohlbacher, O., Reinert, K., Gröpl, C., Lange, E., Pfeifer, N., Schulz-Trieglaff, O., & Sturm, M. (2007). TOPP – the OpenMS Proteomics Pipeline. *Bioinformatics, 23*(2), e191-e197.

[14] Noy, K., & Fasulo, D. (2007). Improved Model-Based, Platform-Independent Feature Extraction for Mass Spectrometry. *Bioinformatics, 23*(19), 2528-2535.

[15] Zhang, J., Gonzalez, E., Hestilow, T., Haskings, W., & Huang, Y. (2009). Review of Peak Detection Algorithms in Liquid-Chromatography-Mass Spectrometry. *Current Genomics, 10*(6), 388-401.

[16] Naylor, B. C., Porter, M.T., Wilson, E., Herring, A., Lofthouse, S., Hannemann, A., Piccolo, S. R., Rockwood, A. L., & Price, J. C. (2017). DeuteRater: A Tool for Quantifying Peptide Isotope Precision and Kinetic Proteomics. *Bioinformatics, 33*(10), 1514-1520.