analytical chemistry

Article

pubs.acs.org/ac

SpectralAnalysis: Software for the Masses

Alan M. Race,*[,†,‡] Andrew D. Palmer,[¶,‡] Alex Dexter,[†,‡] Rory T. Steven,[†] Iain B. Styles,[§,‡] and Josephine Bunch*[,†,∥]

[†]National Centre of Excellence in Mass Spectrometry Imaging (NiCE-MSI), National Physical Laboratory, Teddington, TW11 0LW, United Kingdom

[‡]PSIBS Doctoral Training Centre, School of Chemistry, University of Birmingham, Birmingham, B15 2TT, United Kingdom

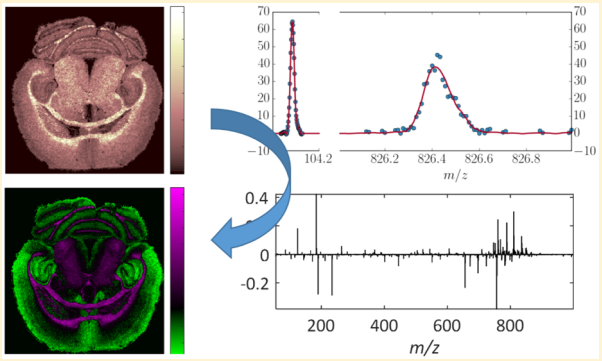
[¶]European Molecular Biology Laboratory, Meyerhofstrasse 1, Heidelberg, 69117, Germany

[§]School of Computer Science, University of Birmingham, Birmingham, B15 2TT, United Kingdom

[∥]School of Pharmacy, University of Nottingham, Nottingham, NG7 2RD, United Kingdom

Supporting Information

ABSTRACT: The amount of data produced by spectral imaging techniques, such as mass spectrometry imaging, is rapidly increasing as technology and instrumentation advances. This, combined with an increasingly multimodal approach to analytical science, presents a significant challenge in the handling of large data from multiple sources. Here, we present software that can be used through the entire analysis workflow, from raw data through preprocessing (including a wide range of methods for smoothing, baseline correction, normalization, and image generation) to multivariate analysis (for example, memory efficient principal component analysis (PCA), non-negative matrix factorization (NMF), maximum autocorrelation factor (MAF), and probabilistic latent semantic analysis (PLSA)), for data sets acquired from single experiments to large multi-instrument, multimodality, and multicenter studies. SpectralAnalysis was also developed with extensibility in mind to stimulate development, comparisons, and evaluation of data analysis algorithms.



Spectral imaging is a broad category of techniques where a spectrum is acquired at spatially resolved locations. Such techniques include mass spectrometry imaging (MSI) and Raman spectroscopy, which have also been used in combination.[1,2] Mass spectrometry imaging covers a whole suite of techniques which rely on different ionization principles, mass analyzers, and detectors to measure the m/z values of gas phase ions produced at spatially resolved locations. Different ion sources affect the classes of molecules that can readily be ionized, spatial resolution that can be achieved, and degree of fragmentation that occurs and so have the capability to provide complementary data about the chemical composition of a given sample.[3,4] Similarly, in matrix assisted laser desorption/ionization (MALDI) MSI, complementary data can be achieved through the use of different matrices, mass ranges, and/or polarities to target different classes of molecule on the same sample.[5]

Even when a single mass spectrometer is used within a study, the number of data sets being analyzed together is increasing. Recently, 9 tissue sections per time point resulting in a total of 27 sections were analyzed to investigate protein digestion;[6] 63 sections from gastric cancer and 32 from breast cancer were analyzed to investigate intratumor heterogeneity,[7] and 96 sections taken from 32 mice were analyzed to investigate the consequences of cortical spreading depression.[8]

Both the incorporation of multiple modality data, or multiple MSI modality data, and the increasing trend toward larger MSI studies present a significant challenge in the data handling, visualization, and analysis. Each MSI instrument vendor supplies software for processing and visualizing data acquired on their instruments. These software cannot analyze data from other instrument manufacturers due to the use of proprietary data formats. As there are no common preprocessing methods between any of these software (a list of which is given in Table S1), there can be no guarantee that the data have been treated equally, eliminating them from consideration as the software of choice for multimodality studies. Similar challenges exist when trying to analyze data from large studies, as these will often span multiple data acquisitions (resulting in multiple data files) which can only be analyzed one at a time in most software packages.

The first vendor neutral, and often still used, tool for visualizing MSI data was BioMAP.[9] This enabled a user-friendly means of visualizing data acquired on many mass spectrometers, even before the community agreed upon the imzML

Received: April 26, 2016
Accepted: August 25, 2016

DOI: 10.1021/acs.analchem.6b01643
Anal. Chem. XXXX, XXX, XXX−XXX

standard for sharing data.[10] A limitation of BioMAP in the processing of large MSI data is the limit to the number of m/z channels that can be loaded for any data set (32 768, due to the number format used) which has become a more significant issue as the instrumentation has improved. Since the advent of imzML, a wide number of third party software packages have been developed and released as open source software (MSiReader,[11] Cardinal,[12] and OmniSpect[13]) or freely available software (OpenMSI,[14] DataCubeExplorer,[15] and msi-Quant[16,17]) or made available to collaborators only (Mirion[18])) or as a commercial product (SCiLS Lab,[19] MALDIVision,[20] and Quantinetix[21]). The preprocessing methods available in each of these software tools are presented in Table S1. Conversion to imzML is possible through most vendor software tools as well as third party software such as imzMLConverter[22] (shown in Figures S2−S4). As most of these packages support the loading of data in the imzML format (the exception being SCiLS Lab[19]), it is now possible to process data from any instrument that has a corresponding imzML converter, while simultaneously increasing the preprocessing methods available to the analyst through the choice of software. The drawback that still remains is that these software packages do not export processed data or partial results to a format readable by other software packages, meaning that the user is restricted to the functionality included within their chosen software tool.

Here, we present software that can be used through the entire analysis workflow, from raw data through preprocessing to multivariate analysis, for data sets acquired from single experiments to large multi-instrument, multimodality, and multicenter studies. Such a wide collection of capabilities does not exist in any currently available software, which is variously limited by the instruments supported,[19] preprocessing capabilities,[15] or support for multivariate analysis.[11,20,21]

EXPERIMENTAL SECTION

All experiments were conducted in accordance with local ethical guidelines for animal care. MALDI MSI data of a sagittal section of rat brain were acquired using a QSTAR Elite (SCIEX, Ontario, Canada) as described by Carter et al.[23] Coronal rat brain sections were prepared using the protocols described by Steven and Bunch,[5] and MALDI MSI data were acquired using either an ultrafleXtreme (Bruker, Bremen, Germany) or Synapt G2 (Waters, Manchester, UK) with a pixel size of 100 μm. DESI MSI data of a fingerprint were acquired using an LTQ Orbitrap Velos (Thermo Scientific, Bremen, Germany) as described by Bailey et al.,[24] in negative ion mode. Mouse lung was sectioned at 12 μm thick and thaw mounted on ITO-coated glass slides (Bruker). SIMS MSI data were acquired using a TOF-SIMS IV (IONTOF, Muenster, Germany) equipped with a 25 keV Bi_3^+ primary ion source delivering an ion dose of 1.1×10^{10} ions per cm^2.

QSTAR Elite data were converted to mzML using MS Data Converter version 1.3 (SCIEX). Synapt G2 and LTQ Orbitrap Velos data were converted to mzML using msconvert as part of ProteoWizard.[25] ultrafleXtreme data were converted to mzML using CompassXport (Bruker). All mzML data were converted to imzML using imzMLConverter.[22] TOF-SIMS IV data were converted to GRD format using SurfaceLab 6 (IONTOF) and then to imzML using imzMLConverter.[22]

The interface for the SpectralAnalysis is shown in Figure S1 and was written primarily in MATLAB to provide an easier means of modification and custom access to and manipulation of data, with some features written in C and Java for performance improvements. The source code and an executable version (which has no additional software requirements) will be made available at https://github.com/AlanRace/SpectralAnalysis.

DISCUSSION

The remainder of the manuscript will discuss the novel combination of features included within SpectralAnalysis. Full descriptions of included algorithms for memory efficiency and ensuring a consistent m/z axis are omitted here for brevity but can be found within the Supporting Information.

Preprocessing. The purpose of preprocessing is to remove artifacts introduced during the data acquisition stage, to make spectra comparable to one another, and to improve the efficacy of peak detection routines. The common preprocessing methods applied in mass spectrometry are smoothing, baseline correction, normalization, and peak detection. Here, we include commentary on each of these methods as well as an additional step which is not often discussed, methods for ensuring a consistent m/z axis across a data set.

The suitability of certain preprocessing methods largely depends on the nature of the data to be analyzed. Preprocessing methods included in each vendor's software (for which there is a publicly available description) are also included in SpectralAnalysis, making it one of the most feature complete software tools currently available with the widest applicability to process spectra acquired using any instrument. Taking this a step further and allowing the effects of the preprocessing methods to be visualized in real time enables the user to select appropriate methods and associated parameters for optimally removing experimental artifacts and noise. The interface for performing this is shown in Figure S5.

Furthermore, it is possible to create a custom "preprocessing workflow" that allows a sequence of preprocessing methods to be applied in a user specified order using the interface shown in Figure S6. Custom workflows can be saved, shared, and reused. This not only gives the user great flexibility over the transformations applied to each spectrum but also enables the recreation of previously published routines such as LIMPIC[26] as well as in-house workflows. This allows rapid evaluation and incorporation of newly developed preprocessing workflows without the need for additional software and provides a route for methods to be published alongside articles or submitted as part of the review process. A data set preprocessed in this way can also be exported to imzML, allowing preprocessing to be performed within SpectralAnalysis and enabling subsequent processing to be performed in the analyst's software package of choice, archival of preprocessed data, or submission of preprocessed data to public repositories.

Recently Oetjen et al.[27] made a number of 3D MSI data sets publicly available to stimulate development of software capable of handling, processing, and evaluating the reproducibility of such data. The preprocessing techniques included within this software can be used to help answer one of the key questions asked by Oetjen and co-workers, what method(s) increase reproducibility of the experiments?[27] In other words, which method(s) best correct for differences in sample preparation between two sections, day-to-day variation, and pixel-to-pixel variation? This is enabled by the extensibility of the software (discussed below), providing a powerful tool supporting the benchmarking of new methods against any and all currently implemented methods. An example of preprocessing workflow

DOI: 10.1021/acs.analchem.6b01643
Anal. Chem. XXXX, XXX, XXX−XXX

applied to one of the publicly available data sets is presented in Figure S7; however, a thorough evaluation of preprocessing methods within this context is beyond the scope of this Article.

The order in which the preprocessing methods are applied has an effect on the resulting data. The widely accepted order for preprocessing time-of-flight (TOF) data is smoothing or denoising followed by baseline correction prior to peak detection.[26,28,29] Noise reduction or removal methods such as baseline correction and smoothing aim to improve the peak detection method of choice.

When comparing, averaging, or otherwise mathematically manipulating two or more spectra, it is important that they are represented by the same number of m/z bins with the same m/z intervals. This is a common requirement in data reduction routines, where one or more summary spectra, such as the mean spectrum, are used for feature detection[29,30] or peak alignment.[31] This also has the benefit of enabling spectra to be directly stored as a matrix (a 2D matrix as required for many post processing techniques such as principal component analysis (PCA) or a 3D "datacube" for efficient image generation and manipulation). Methods for achieving this are defined by Algorithms S1−S5 and visualized in Figures S8−S10, with detailed discussion on each method found elsewhere.[32]

Smoothing aims to remove small, local, fluctuations in intensity, often caused by noise, that prevent peak detection algorithms from functioning optimally. The *de facto* standard smoothing method used is Savitzky-Golay due to its intensity preserving properties.[33] This, along with other commonly used methods such as moving average and Gaussian, are window based techniques, meaning that they consider a set number of data points at once, the "window", to generate a single data point in the resulting data. The window is then "slid" along the data to the next point where a new window is considered.

Care must be taken when combining certain methods for ensuring a consistent m/z axis (a set of m/z values that is the same for every spectrum, discussed in more detail elsewhere[32]) and window based preprocessing methods as peak widths often vary across the mass range. For example, when processing TOF data and the detector based m/z axis is used, the chosen window size for the smoothing function is appropriate for a peak at m/z 826, but applying smoothing with the same window size to a peak at m/z 104 causes peak broadening and a reduction in the peak height, as shown in Figure S11a. However, when the same number of data points span both peaks, neither peak is broadened and the heights are retained, minus noise. The fwhm of the peak at m/z 104 in Figure S11a is 0.03, compared to 0.02 in Figure S11b which is equivalent to the mass resolving power being reduced to 3500 from 5200 (calculated at m/z 104.08537).

Baseline correction aims to remove an experimental artifact, often attributed to chemical noise, to aid peak detection and increase comparability between spectra. The effect is more pronounced when acquiring data over a large mass range and is a common feature in protein imaging by MALDI MS due to the use of a linear TOF. The type, or lack thereof, of baseline present in the data is dependent on both the instrument and experimental parameters, such as the mass resolving power, mass range, the laser power (inducing and subsequently increasing fragmentation), the analyte, and the matrix used. The choice of the baseline correction method will depend on the style of baseline present, where some methods make certain assumptions about the shape of the baseline. Different methods and their corresponding assumptions are discussed in more detail elsewhere.[32]

Normalization is a relatively controversial topic in mass spectrometry imaging with a significant amount of debate still ongoing. A detailed review of common normalization methods is provided by Deininger et al.[34] Since then, an additional method for normalization has been proposed by Fonville et al.[35] A visual comparison of these normalization methods applied to a sagittal section of rodent brain is shown in Figure 1.

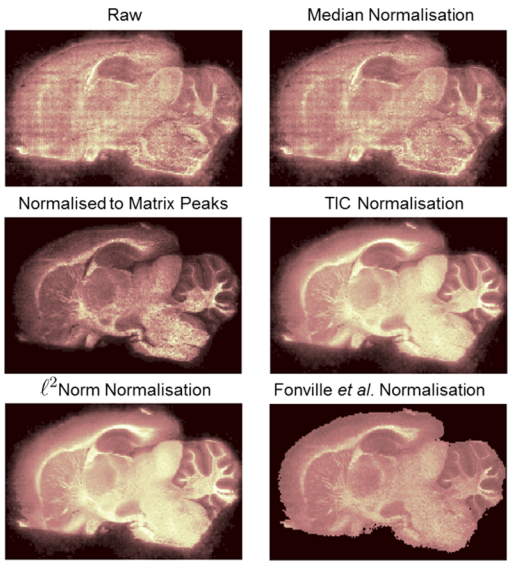


Figure 1. Comparison of normalization techniques applied to the same ion image (m/z 810 in a sagittal section of formalin fixed rat brain).

In the raw image, there are quite apparent experimental artifacts in the form of criss-cross patterns. These patterns are removed in all methods except median normalization, which normalizes to an approximate measure of the intensity of the baseline. In this data set, there is no baseline, resulting in a similar median value for every pixel (which in this case becomes a measure of noise rather than the baseline and is approximately 2 arb. unit). As the spectral sparsity of a given data set increases, the median tends toward 0, at which point this method becomes inappropriate for normalization. Alternatively, the zero values can be omitted while calculating the median value. In this case, it is likely that the estimation of the baseline, or noise level, will be an overestimate and small, low intensity, spectral features may be removed as part of the baseline correction process.

The other normalization methods considered make assumptions about the nature of the data. A frequently employed method, especially in drug quantification studies, is normalization to the intensity of an internal standard selected as a close mimic of the compound of interest, for example, a deuterated analogue.[36] In situations where an internal standard was not included, a pseudo internal standard can be selected from components that are present within the data. In MALDI MSI, normalizing to matrix peaks assumes that the matrix should be constant across the image and so by normalizing to the matrix peaks the aim is to compensate for any heterogeneity of the matrix distribution. When considering only matrix regions, the sum of all detected matrix ions (fragments, clusters,

DOI: 10.1021/acs.analchem.6b01643
Anal. Chem. XXXX, XXX, XXX−XXX

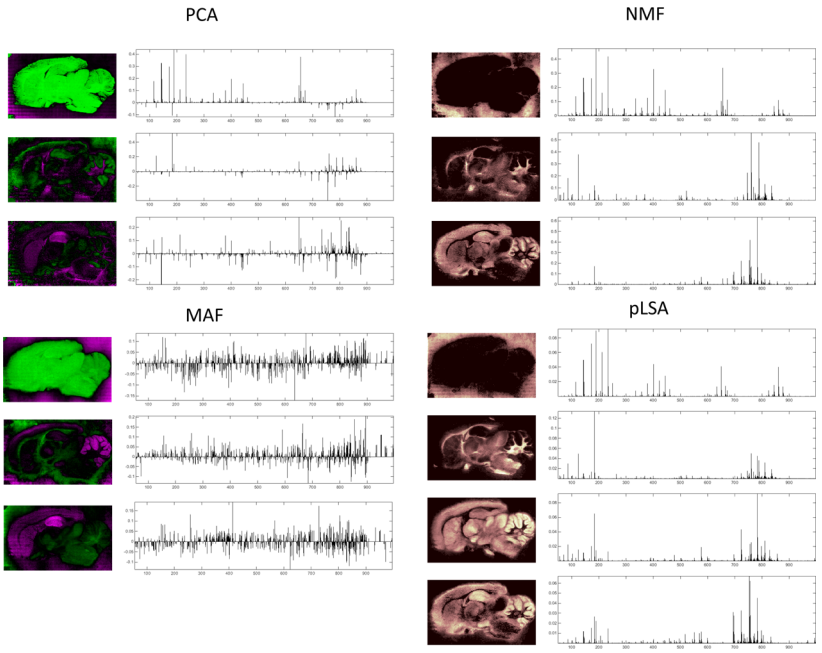


Figure 2. Selected factors from principal component analysis (PCA), non-negative matrix factorization (NMF), maximum autocorrelation factor (MAF), and probabilistic latent semantic analysis (PLSA) applied to a MALDI MS image of a sagittal section of rat brain.

and adducts) could potentially provide a good normalization factor. However, once an analyte is incorporated, suppression effects can cause ions to be detected differently, or not at all, and so this method becomes less suitable. As this relies on the matrix peaks, this method is only applicable to MALDI data; however, the matrix peaks could be replaced with other experimental constants prevalent in other techniques such as solvent peaks in desorption electrospray ionization (DESI) or liquid extraction surface analysis (LESA) but with similar caveats.

The total ion current (TIC), and similarly the l^2, normalization method makes the assumption that at every pixel location the same number of ions should be detected. In homogeneous single compound samples, this would hold true; however, any form of heterogeneity renders this assumption inappropriate. It could be argued that within a given area there is only a given amount of charge present required for the formation of ions, and so, despite the heterogeneity, this method is applicable. Due to varying proton affinities of molecules present, suppression effects, and reactions that may occur within the plume (for example, charge transfer or metastable fragmentation), this assumption is unlikely to hold true. Fonville et al.[35] attempt to provide a more robust method of normalization that does not suffer from the issues listed above, by only considering signal from the analyte when constructing the scaling factor for each pixel. However, given the heterogeneity of the analyte, this is still not an ideal solution.

Until there is a consensus on which method is most applicable in which situation, it falls to the analyst to evaluate and investigate these methods and their appropriateness in the context of their data, and so, each of the methods described above are included within SpectralAnalysis. The method employed for generating ion images from a MSI data set can result in different apparent spatial distributions, demonstrated in Figure S12. Methods that simply extract a single m/z channel are more susceptible to noise in the data, and so, ion images generated with such methods often appear to include high frequency fluctuations in intensity. Through the application of appropriate preprocessing prior to image generation, these fluctuations can be lessened, producing a smoother image, which can help reveal patterns previously masked by noise.

A difference in the spatial distribution produced by each of the ion image generation methods in both cases, with and without preprocessing being applied, can be observed in Figure S12. The difference is primarily between the methods that extract a value at a single m/z channel and those that integrate across the peak. This can likely be explained by the effect illustrated in Figure S13 where the distribution of intensities when integrating the left side of the peak is different to that of the right side, potentially due to unresolved ions having different spatial distributions.

Multivariate Analysis. Multivariate analysis techniques have been shown to be a powerful tool for aiding interpretation of these complex data sets. Despite this, very few freely available software packages include such techniques and those that do only include one or two.[12,14] The efficacy of any single technique used in isolation has recently been brought into question.[37]

To address this, SpectralAnalysis includes principal component analysis (PCA), non-negative matrix factorization (NMF), maximum autocorrelation factor (MAF),[38] and probabilistic latent semantic analysis (PLSA).[39] Selected factors from each of the techniques applied to a MALDI MS image of a sagittal section of rat brain[23] are shown in Figure 2 using diverging color schemes where appropriate.[40] Different anatomical features can be distinguished in the different techniques; for example, the hippocampus region is highlighted most prominently in the third MAF factor and is visible (but could easily be overlooked) in the fourth PLSA latent variable and third PCA component but is not prominent within any of the NMF factors. However, NMF highlights significant contrast between the gray and white matter regions in the second and third components which are less obvious in the PCA and MAF.

DOI: 10.1021/acs.analchem.6b01643
Anal. Chem. XXXX, XXX, XXX−XXX

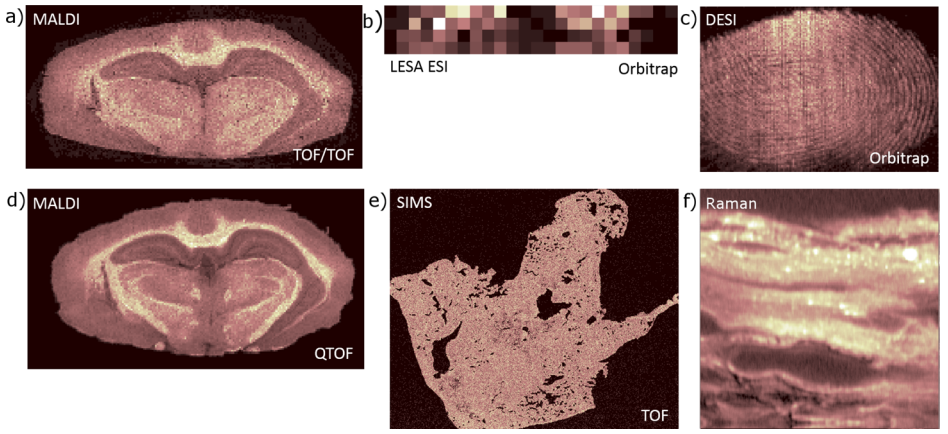


Figure 3. A selection of different modality imaging data, acquired at different length scales, processed using SpectralAnalysis. (a) MALDI MSI (m/z 826.6) of coronal rat brain acquired with a pixel size of 100 μm using an ultrafleXtreme (Bruker Daltonics). (b) LESA MSI of liver acquired with a pixel size of 1000 μm using an Orbitrap Elite (Thermo Scientific). (c) DESI MSI (m/z 509.36) of a fingerprint acquired with a pixel size of 200 μm using an LTQ Orbitrap Velos (Thermo Scientific). (d) MALDI MSI (m/z 826.6) of coronal rat brain acquired with a pixel size of 100 μm using a Synapt G2S (Waters). (e) SIMS data (m/z 104.1) of murine lung acquired with a pixel size of 7 μm using a TOF-SIMS IV (ION-TOF). (f) Spontaneous Raman scattering (SRS) of a living skin equivalent with a pixel size of 0.2 μm using a home-built system (NPL).

Depending on the question at hand, the increased ability to differentiate spectrally different regions, such as anatomy, could be very powerful, especially in drug distributions studies.

Supporting Large Scale MSI. *Multimodality Data.* It is becoming increasingly desirable to incorporate multiple additional techniques into the analysis of mass spectrometry imaging data. This can range from simply including histology images to determine colocalization with anatomy, through the inclusion of additional MSI data (either from the same instrument or a complementary one),[3] to the inclusion of other spectral imaging modalities such as Raman.[2] To cater for this scenario, SpectralAnalysis was written in such a way that enables any spectral data to rapidly be incorporated, allowing any of the core functionality (such as preprocessing and multivariate analysis) to be performed without alteration. A selection of data from different modalities processed using SpectralAnalysis is given in Figure 3. Although some preprocessing techniques are only suitable for specific styles of data, the end goal largely remains the same and the majority of algorithms for smoothing, baseline correction, and peak detection included are technique independent providing a powerful platform for multimodality processing, investigation, and visualization.

Handling Extremely Large Data Sets. As instruments develop and improvements are made in both the mass resolution and the lateral resolution, the data size is correspondingly increasing, with raw data easily capable of exceeding 10s to 100s of GB for a single MSI data set. Furthermore, the move toward larger biomedical studies with increased cohort and sample numbers (including replicates) significantly increases the data handling challenge. The vast majority of MSI software loads the data to be processed into RAM before any visualization or analysis can be performed. This then introduces a restriction on the size of the data that can be processed on the basis of the hardware of the computer being used, and as the data size is rapidly outpacing the hardware specifications, this is becoming an increasing problem and may render some software/hardware/data combinations unusable.

This problem has been addressed previously by enabling the ability to load and analyze a subsection of the data set, limiting the number of pixels, the mass range, or both.[15,18] SpectralAnalysis also includes this option and expands upon it by allowing the user to select an arbitrarily shaped region of interest as well as an optional mass range limit to be loaded into memory; the interface for this is shown in Figure S14. In order to do this, the m/z axis must be consistent and so any of the techniques discussed above can be employed to ensure this. In some cases, the parameters can be specified such that this process also contributes to the reduction of data (such as rebinning) at the cost of potentially discarding information.

However, this approach does not solve all situations as it limits the analyst's view of the data set as a whole and involves discarding data (and potentially analytically useful information) which can be detrimental to the analysis, while still being fundamentally constrained by the RAM available. This issue of datasize is compounded when multiple MS images are combined as discussed below, requiring the analyst to compromise even further to be able to visualize the data. SpectralAnalysis includes memory efficient methods that enable data sets vastly exceeding the size of the available RAM to be visualized, preprocessed, and analyzed using multivariate analysis.

Generation of individual ion images does not require the whole data set to be loaded into memory. Instead, only the data points that fall within the peak boundaries (m_{min} and m_{max}) are required so that one of the image generation methods shown in Figure S12 can be applied. This can also be taken a step further, and since the calculation of an intensity at a given pixel is completely independent of all other pixels, only one spectrum is required in memory at a given point in time. This significantly reduces the amount of memory required, as a single spectrum ranges from 100s of kB to 1−2 MB, compared to the 10s of GBs for the whole data set. In many cases, it is desirable to preprocess the data prior to ion image generation. The algorithm presented in Algorithm S6 presents a memory efficient method of generating ion images from preprocessed data. Each spectrum is loaded in sequentially and preprocessed; then, the data points within peak limits are extracted, and an intensity is generated on the basis of the image generation method of choice. The spectrum can then be removed from memory before the next is loaded in. This reduces the amount

DOI: 10.1021/acs.analchem.6b01643
Anal. Chem. XXXX, XXX, XXX−XXX

of memory required for the size of a single spectrum, plus the size of the ion image(s) to be generated, which is orders of magnitude smaller than the whole data, allowing TBs of data to be visualized on even the most memory constrained systems.

Peak detection is often performed on spectral representations of the data.[29] As mentioned above, these only require a single spectrum to be loaded into memory at once and can be generated in a memory efficient manner using Algorithm S9. It is possible to generate multiple representations at once, requiring only a single pass through the data, by including additional update methods after line 8 of Algorithm S9. This provides a memory efficient method of generating all spectral representations proposed by McDonnell et al.[29] for optimal peak detection in a given data set.

By combining the above methods, it is possible to reduce the MS image to a "datacube" in a memory efficient manner, using Algorithm S10. In this case, only a single spectrum and the datacube is required to be in memory at any one point in time. This allows reduction of data to peak lists without a limitation applied to the number of peaks retained.

The algorithm as it is presented reduces and loads the data into memory; however, this can also be used to write the reduced data to disk by altering line 10 in Algorithm S10 to be a disk write instead of a matrix update. In this case, only a single spectrum is required to be in memory, making this process feasible on memory constrained systems where the datacube is larger than that of the RAM. Then, all methods for handling large data sets described above can be employed to visualize and further process the data.

A previously published memory efficient PCA algorithm is also fully integrated into SpectralAnalysis.[41] The capabilities of this algorithm have been expanded to include the ability to use any user defined preprocessing workflow and to allow memory efficient scaling of the data (shown in Algorithm S11) to be applied by each of the techniques investigated by Tyler et al.[42]

Multidata Set Studies. The ability to combine multiple data sets acquired separately but which together form a single experiment is extremely powerful. Consider the experiments presented by Carter et al.[23] and Griffiths et al.[43] where different sample preparation methods are being compared but the data were collected as separate mass spectrometry images. To compare these data, the analyst would have to load an image, perform any preprocessing necessary, search for an ion image of interest, and then repeat for any other data set being compared. Then, the analyst would have to ensure that the intensity scales that the images were presented on were comparable prior to any interpretation. This is a laborious and time-consuming process and would have to be repeated for each ion image that was investigated. While this is manageable for an experiment only consisting of two MS images, when trying to perform this on 14 serial sections, such as the data presented by Steven et al.,[44] it becomes impractical.

The imzMLConverter tool[22] provides the ability to tile and combine multiple imzML files together into a single imzML file. In combination with SpectralAnalysis, this feature enables the analyst to rapidly compare the spatial distributions and relative abundances of ions in the visualization software of their choice without needing to open multiple data sets and manually ensure color schemes and intensity ranges of each ion image generated for each data set are comparable. When considering this, and additionally the support for large data discussed above, the data presented by Steven et al.[44] and Oetjen et al.[27] become much more manageable to process, and much less error prone, when the whole study is considered as one large data set. Ion images can be generated in seconds (an example of which is given in Figure 4) rather than minutes to hours when having to process each data set individually and manually. This can be used to successfully analyze data from larger studies such as those published by McDonnell and co-workers.[6−8]

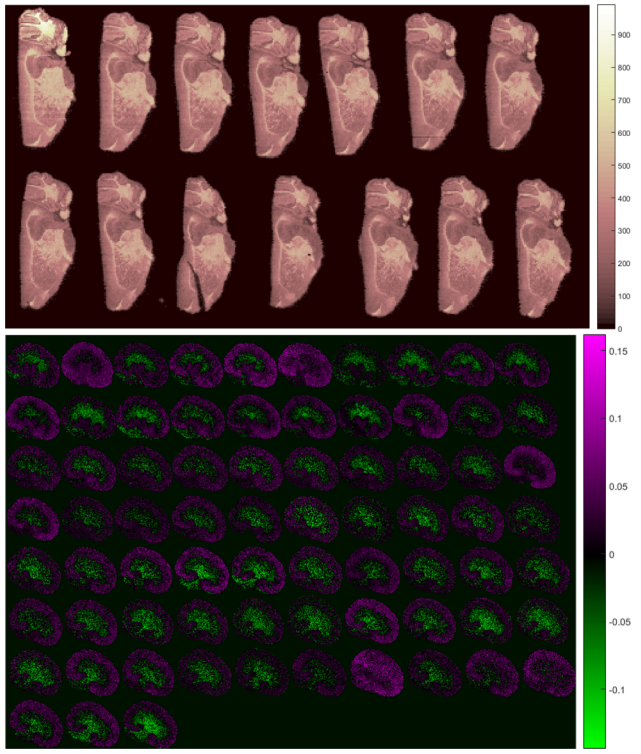


Figure 4. Visualization and processing of large data (top, 10 GB comprised of 104 916 spectra; bottom, 42 GB comprised of 1 362 830 spectra) from multiple experiments within SpectralAnalysis. Top: MSI data (m/z 826.6) of 14 serial sagittal sections of mouse brain. Data were acquired from 14 separate imaging acquisitions by Steven et al.[44] Data were subsequently combined using imzMLConverter and processed together.[22] Bottom: Principal component 2 calculated using memory efficient PCA[41] on the 3D kidney data set made publicly available by Oetjen et al.[27]

If a 3D imzML data set is opened within SpectralAnalysis, such as those recently released to the community by Oetjen et al.,[27] then this is automatically detected and the data is presented as a 2D tile. All included algorithms and features can then be applied to the data, for example, memory efficient PCA as shown in Figure 4. This provides the ability to visualize 3D data while also including the tools necessary to be able to evaluate suitability of methods for handling variations in signal intensity observed between different sections, as was noted as one of the main reasons for releasing the data. While 3D visualization is not natively included, it could be included at a later date due to the extensible nature of the software (discussed below).

Extensibility. SpectralAnalysis was developed with extensibility in mind, providing a platform for visualization and processing that it is simple to include additional data format readers, preprocessing, multivariate analysis, and clustering algorithms without the requirement to write new user interface or data visualization code. A block diagram visualizing the core components that can be extended is shown in Figure S15. A

DOI: 10.1021/acs.analchem.6b01643
Anal. Chem. XXXX, XXX, XXX−XXX

"Parser" handles the reading of a given file format, for example, imzML, to get meta information such as the image dimensions (width, height, depth) and whether the data are stored in sparse format or dense (to determine the need to ensure a consistent m/z axis as discussed previously) as well as to read parts of the data from disk. Extension of this allows data in different formats (such as older MSI formats like Analyze 7.5) as well as file formats associated with other imaging modalities to be visualized and processed. The "DataRepresentation" determines how the data are to be handled, either in memory or left on disk, and could be extended to include additional capabilities such as a hybrid of the two (cached data in memory, majority remain on disk). "Preprocessing" and associated subcomponents include all of the features discussed in the "Preprocessing" section and can be extended to include methods or algorithms that are currently omitted or to develop new algorithms and make use of the real-time visualization of the effects on spectral data. "Postprocessing" includes all multivariate analysis techniques shown in Figure 2, as well as clustering algorithms not shown, and can be extended to include additional algorithms. This provides a platform for rapid testing of algorithms at every stage of the analysis process on multiple modality data sets with instant visualization of the results.

By having this design philosophy, it is hoped that SpectralAnalysis will enable the community to evaluate current methods against one another and, most importantly, evaluate current methods against newly developed ones. This is especially important for quantification studies, where normalization plays a significant role in the data processing but remains a heavily debated and actively researched topic.

CONCLUSIONS

SpectralAnalysis provides a unique, and currently the most exhaustive, collection of algorithms for preprocessing and subsequent multivariate analysis of spectral imaging data. This, combined with the flexibility of the extensibility to include additional algorithms, results in a platform suitable for comparisons of preprocessing methods on MSI data acquired on any instrument.

Due to the capability of handling multiple spectral imaging modalities, each of which capture different information about the sample, SpectralAnalysis would be an excellent platform on which to develop and integrate multimodality processing techniques such as image fusion. Image fusion aims to combine data from multiple sources to gain information that was not present in each source in isolation. For example, the combination of a high spatial resolution, single channel image with a multispectral but low spatial resolution image resulting in a high spatial multispectral image.[45] This could be extended and applied to multiple MSI data sets to combine, for example, the high spatial resolution of SIMS data with the high mass range of MALDI data.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.analchem.6b01643.

Details on how to convert data into a suitable format (using imzMLConverter[22]) and algorithms describing preprocessing and memory efficient methods included within SpectralAnalysis (PDF)

AUTHOR INFORMATION

Corresponding Authors

*E-mail: alan.race@npl.co.uk.

*E-mail: josephine.bunch@npl.co.uk.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

A.M.R. (2010−2014), A.D.P. (2009−2013), A.D. (2012−2016), and R.T.S. (2009−2013) gratefully acknowledge financial support from the EPSRC through studentships from the PSIBS Doctoral Training Centre (EP/F50053X/1). J.B. gratefully acknowledges funding from NPL strategic research programmes 116301 and 117194 and strategic capability programme AIMS HIGHER. Thanks to Jocelyn Sarsby (University of Liverpool) for preparing and acquiring the LESA MSI data. Thanks to Peter Marshall (GSK) for preparing the lung tissue and Melissa Passerelli (NPL) for acquiring the corresponding SIMS data. Thanks to Epistem Ltd (Manchester) for preparing the living skin equivalent sample and to Alasdair Rae (NPL) for acquiring the SRS data. Data supporting this research is openly available from the University of Birmingham data archive at http://rab.bham.ac.uk/.

REFERENCES

(1) Bocklitz, T.; Crecelius, A.; Matthäus, C.; Tarcea, N.; Von Eggeling, F.; Schmitt, M.; Schubert, U.; Popp, J. *Anal. Chem.* **2013**, *85*, 10829−10834.

(2) Ahlf, D. R.; Masyuko, R. N.; Hummon, A. B.; Bohn, P. W. *Analyst* **2014**, *139*, 4578−4585.

(3) Eberlin, L. S.; Liu, X.; Ferreira, C. R.; Santagata, S.; Agar, N. Y.; Cooks, R. G. *Anal. Chem.* **2011**, *83*, 8366−8371.

(4) Eijkel, G.; Kükrer Kaletaş, B.; Van der Wiel, I.; Kros, J.; Luider, T.; Heeren, R. *Surf. Interface Anal.* **2009**, *41*, 675−685.

(5) Steven, R. T.; Bunch, J. *Anal. Bioanal. Chem.* **2013**, *405*, 4719−4728.

(6) Heijs, B.; Tolner, E. A.; Bovée, J. V.; van den Maagdenberg, A. M.; McDonnell, L. A. *J. Proteome Res.* **2015**, *14*, 5348−5354.

(7) Balluff, B.; Frese, C. K.; Maier, S. K.; Schöne, C.; Kuster, B.; Schmitt, M.; Aubele, M.; Höfler, H.; Deelder, A. M.; Heck, A. J.; Hogendoorn, P. C.; Morreau, J.; Altelaar, A. M.; Walch, A.; McDonnell, L. A. *J. Pathol* **2015**, *235*, 3−13.

(8) Carreira, R. J.; Shyti, R.; Balluff, B.; Abdelmoula, W. M.; van Heiningen, S. H.; van Zeijl, R. J.; Dijkstra, J.; Ferrari, M. D.; Tolner, E. A.; McDonnell, L. A.; Maagdenberg, A. M. J. M. *J. Am. Soc. Mass Spectrom.* **2015**, *26*, 853−861.

(9) Hosokawa, N.; Sugiura, Y.; Setou, M. *Imaging Mass Spectrometry*; Springer: New York, 2010; pp 113−126.

(10) Schramm, T.; Hester, A.; Klinkert, I.; Both, J.-P.; Heeren, R.; Brunelle, A.; Laprévote, O.; Desbenoit, N.; Robbe, M.-F.; Stoeckli, M.; Spengler, B.; Römpp, A. *J. Proteomics* **2012**, *75*, 5106−5110.

(11) Robichaud, G.; Garrard, K. P.; Barry, J. A.; Muddiman, D. C. *J. Am. Soc. Mass Spectrom.* **2013**, *24*, 718−721.

(12) Bemis, K. D.; Harry, A.; Eberlin, L. S.; Ferreira, C.; van de Ven, S. M.; Mallick, P.; Stolowitz, M.; Vitek, O. *Bioinformatics* **2015**, *31*, 2418−20.

(13) Parry, R. M.; Galhena, A. S.; Gamage, C. M.; Bennett, R. V.; Wang, M. D.; Fernández, F. M. *J. Am. Soc. Mass Spectrom.* **2013**, *24*, 646−649.

(14) Rübel, O.; Greiner, A.; Cholia, S.; Louie, K.; Bethel, E. W.; Northen, T. R.; Bowen, B. P. *Anal. Chem.* **2013**, *85*, 10354−10361.

DOI: 10.1021/acs.analchem.6b01643
Anal. Chem. XXXX, XXX, XXX−XXX

(15) Klinkert, I.; Chughtai, K.; Ellis, S. R.; Heeren, R. *Int. J. Mass Spectrom.* **2014**, *362*, 40−47.
(16) Källback, P.; Shariatgorji, M.; Nilsson, A.; Andrén, P. E. *J. Proteomics* **2012**, *75*, 4941−4951.
(17) Källback, P.; Nilsson, A.; Shariatgorji, M.; Andrén, P. E. *Anal. Chem.* **2016**, *88*, 4346.
(18) Paschke, C.; Leisner, A.; Hester, A.; Maass, K.; Guenther, S.; Bouschen, W.; Spengler, B. *J. Am. Soc. Mass Spectrom.* **2013**, *24*, 1296−1306.
(19) *SCiLS/SCiLS Lab - The statistical analysis software*; http://scils.de/software/; 2014; Accessed: 21/05/2014.
(20) Biosoft, P. *MALDIVision*; http://www.premierbiosoft.com/maldi-tissue-imaging/index.html; Accessed: 20/03/2016.
(21) imabiotech. *Quantinetix*; https://www.imabiotech.com/Benefits; Accessed: 20/03/2016.
(22) Race, A. M.; Styles, I. B.; Bunch, J. *J. Proteomics* **2012**, *75*, 5111−5112.
(23) Carter, C. L.; McLeod, C. W.; Bunch, J. *J. Am. Soc. Mass Spectrom.* **2011**, *22*, 1991−1998.
(24) Bailey, M. J.; Bradshaw, R.; Francese, S.; Salter, T. L.; Costa, C.; Ismail, M.; Webb, R. P.; Bosman, I.; Wolff, K.; de Puit, M. *Analyst* **2015**, *140*, 6254−6259.
(25) Chambers, M. C.; et al. *Nat. Biotechnol.* **2012**, *30*, 918−920.
(26) Mantini, D.; Petrucci, F.; Pieragostino, D.; Del Boccio, P.; Di Nicola, M.; Di Ilio, C.; Federici, G.; Sacchetta, P.; Comani, S.; Urbani, A. *BMC Bioinf.* **2007**, *8*, 101.
(27) Oetjen, J.; et al. *GigaScience* **2015**, *4*, 1−8.
(28) Gibb, S.; Strimmer, K. *Bioinformatics* **2012**, *28*, 2270−2271.
(29) McDonnell, L. A.; Van Remoortere, A.; De Velde, N.; Van Zeijl, R. J.; Deelder, A. M. *J. Am. Soc. Mass Spectrom.* **2010**, *21*, 1969−1978.
(30) Morris, J. S.; Coombes, K. R.; Koomen, J.; Baggerly, K. A.; Kobayashi, R. *Bioinformatics* **2005**, *21*, 1764−1775.
(31) Alexandrov, T.; Kobarg, J. H. *Bioinformatics* **2011**, *27*, i230−i238.
(32) Race, A. M. Investigation and interpretation of large mass spectrometry imaging datasets. Ph.D. thesis, University of Birmingham, 2016.
(33) Savitzky, A.; Golay, M. J. *Anal. Chem.* **1964**, *36*, 1627−1639.
(34) Deininger, S.-O.; Cornett, D. S.; Paape, R.; Becker, M.; Pineau, C.; Rauser, S.; Walch, A.; Wolski, E. *Anal. Bioanal. Chem.* **2011**, *401*, 167−181.
(35) Fonville, J. M.; Carter, C.; Cloarec, O.; Nicholson, J. K.; Lindon, J. C.; Bunch, J.; Holmes, E. *Anal. Chem.* **2012**, *84*, 1310−1319.
(36) Pirman, D. A.; Reich, R. F.; Kiss, A.; Heeren, R. M.; Yost, R. A. *Anal. Chem.* **2013**, *85*, 1081−1089.
(37) Jones, E. A.; van Remoortere, A.; van Zeijl, R. J.; Hogendoorn, P. C.; Bovee, J.; Deelder, A. M.; McDonnell, L. A. *PLoS One* **2011**, *6*, e24913.
(38) Nielsen, A. A. *Image Processing, IEEE Transactions on* **2011**, *20*, 612−624.
(39) Hanselmann, M.; Kirchner, M.; Renard, B. Y.; Amstalden, E. R.; Glunde, K.; Heeren, R. M.; Hamprecht, F. A. *Anal. Chem.* **2008**, *80*, 9649−9658.
(40) Race, A. M.; Bunch, J. *Anal. Bioanal. Chem.* **2015**, *407*, 2047−2054.
(41) Race, A. M.; Steven, R. T.; Palmer, A. D.; Styles, I. B.; Bunch, J. *Anal. Chem.* **2013**, *85*, 3071−3078.
(42) Tyler, B. J.; Rayal, G.; Castner, D. G. *Biomaterials* **2007**, *28*, 2412−2423.
(43) Griffiths, R. L.; Sarsby, J.; Guggenheim, E. J.; Race, A. M.; Steven, R. T.; Fear, J.; Lalor, P. F.; Bunch, J. *Anal. Chem.* **2013**, *85*, 7146−7153.
(44) Steven, R. T.; Race, A. M.; Bunch, J. *J. Am. Soc. Mass Spectrom.* **2013**, *24*, 801−804.
(45) Van de Plas, R.; Yang, J.; Spraggins, J.; Caprioli, R. M. *Nat. Methods* **2015**, *12*, 366−372.

DOI: 10.1021/acs.analchem.6b01643
Anal. Chem. XXXX, XXX, XXX−XXX

Supporting information for:

SpectralAnalysis: software for the masses

Alan M. Race,[*,†,‡] Andrew D. Palmer,[¶,‡] Alex Dexter,[†,‡] Rory T. Steven,[†] Iain B. Styles,[§,‡] and Josephine Bunch[*,†,‖]

†*National Centre of Excellence in Mass Spectrometry Imaging (NiCE-MSI), National Physical Laboratory, Teddington, UK*

‡*PSIBS Doctoral Training Centre, School of Chemistry, University of Birmingham, Birmingham, UK*

¶*European Molecular Biology Laboratory, Meyerhofstrasse 1, 69117 Heidelberg, Germany*

§*School of Computer Science, University of Birmingham, Birmingham, UK*

‖*School of Pharmacy, University of Nottingham, Nottingham, UK*

E-mail: alan.race@npl.co.uk; josephine.bunch@npl.co.uk

Contents

Content Overview

The content within this Supporting Information provides detail on how to convert data into a suitable format (using imzMLConverter[S1]) for use with SpectralAnalysis, with commentary on optional compression and hardware for either optimising time to access or disk size. Then, preprocessing and memory efficient methods included within SpectralAnalysis are discussed in detail, with accompanying algorithms, in the context of processing MSI data.

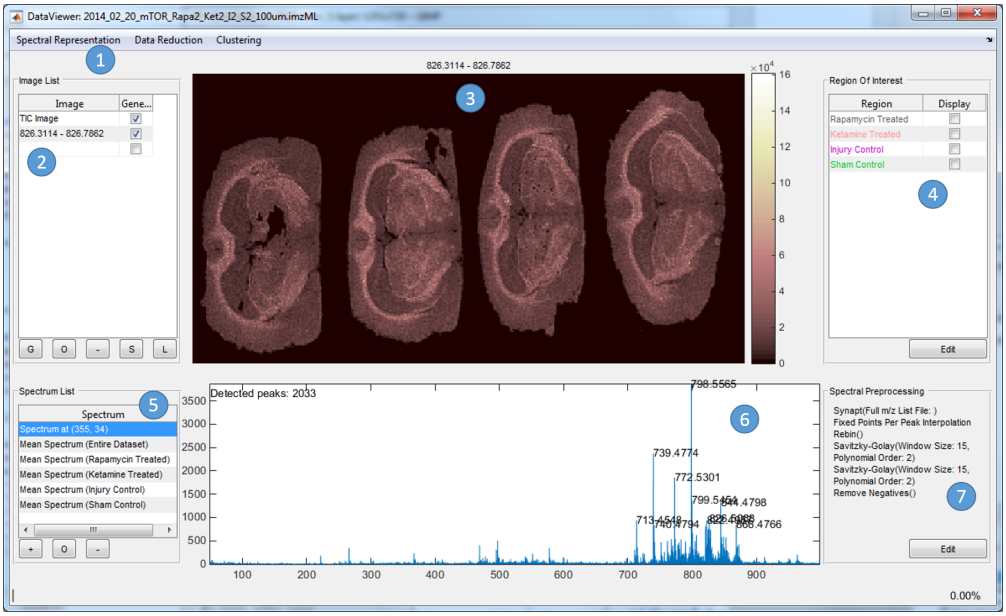


Figure S1: Screenshot of SpectralAnalysis interface. 1) Postprocessing options. 2) List of previously generated ion images, with options to generate from a previously saved list or overlay. 3) Current ion image being displayed. 4) List of created regions of interest. 5) List of generated spectra, with options to overlay. 6) Current spectrum view, with peak detection turned on labelling the top 10 intensity peaks out of the 2033 detected. 7) Preprocessing workflow applied to every spectrum.

Data Formats for MSI

Converters translate data stored using a certain specification (a format) into another. Tools exist for the conversion of most mass spectrometry formats to mzML. To compliment this work rather than repeat it, imzMLConverter was designed to use mzML as an intermediary

Instrument Vendor	Software	Data Format	Baseline Correction	Smoothing	Normalisation	Peak Detection
AB SCIEX	oMALDI / Analyst	wiff	None	Gaussian[S2] 3 Weighted Points[S2]	TIC[S2] Mass Window(s)[S2]	IWA[S2]
	TissueView		Derivative[S3]	User Defined Filter[S3]	Mass Window[S3]	
Bruker	flexImaging / flexAnalysis	Bruker	Convex Hull[S4] Top Hat[S4] Local Median[S5]	Savitzky-Golay[S4] Gaussian[S4] Chemical Noise[S5]	RMS[S4] TIC[S4] Median[S5] Mass Window(s)[S4]	Centroid[S5] SNAP[S5] Sum[S5]
Thermo Fisher Scientific	ImageQuest / Xcalibur	RAW	Curve fitting[S6]	Moving Mean[S6] Gaussian[S6]	TIC[S7] Mass Window[S7]	Genesis[S8] ICIS[S8] Avalon[S8]
Waters	High Definition Imaging / MassLynx	raw	Curve Fitting[S9]	Savitzky-Golay[S9] Moving Mean[S9]		Normal[S9] Apex[S9]
N/A	BioMAP[S10]	Analyze 7.5[S11] imzML	Derivative[S11]	Sinc[S11] Savitzky-Golay[S11]	Reference Image[S11]	
N/A	Cardinal[S12]	Analyze 7.5[S13] imzML[S13]	Local Median[S13] Local Minimum[S13]	Moving Mean[S13] Gaussian[S13] Savitzky-Golay[S13]	TIC[S13]	Simple[S13] Adaptive[S13] LIMPIC[S13 S14]
N/A	DataCubeExplorer[S15]	Analyze 7.5[S16] Datacube[S16] imzML[S16]				
N/A	Mirion[S17]	imzML[S17] raw[S17] udf[S17]				Unknown Method
N/A	msiQuant[S18]	imzML	Unknown Method	Savitzky-Golay[S18]	TIC Median RMS Labelled	
N/A	MSiReader[S19]	Analyze 7.5[S20] mzXML[S20] mzML[S20] imzML[S20]	msbackadj()[S20]		TIC[S20] Mass Window[S20] Custom Data[S20]	Parabolic Centroid[S20] mspeaks()[S20]
N/A	OmniSpect[S21]	Analyze 7.5[S21] imzML[S21] mzXML[S21] netCDF[S21]				
N/A	OpenMSI[S22]	OpenMSI[S22]				
N/A	Quantinetix	imzML Bruker				
N/A	SCiLS Lab[S23]	Bruker	Unknown Method		TIC[S23]	Unknown Method

Table S1: Preprocessing methods available in MSI instrument vendor's and third party software. The 'raw' file format supported by Mirion is Thermo Fisher Scientific 'raw' only. It is worth noting that the user manual for Mirion and SCiLS Lab were not easily available (without permission from the author and without purchasing respectively) and so the list of available functions may not be complete. Abbreviations used in the table are total ion current (TIC), root mean square(RMS), intensity-weighted average (IWA), sophisticated numerical annotation procedure (SNAP), interactive chemical information system (ICIS) and linear MALDI-TOF-MS peak indication and classification (LIMPIC).

format between the proprietary vendor's formats and imzML, the interface of which is shown in Figure S2. This relies on tools such as msconvert (part of ProteoWizard[S24]), which can convert data from all major instrument vendor's proprietary formats to mzML.

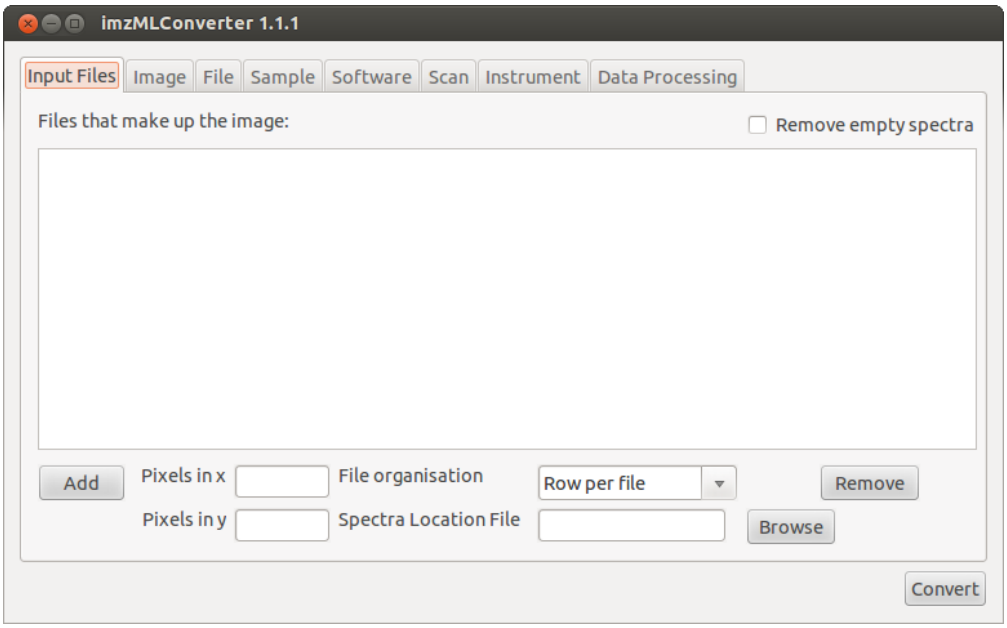


Figure S2: imzMLConverter 1.1.1 graphical user interface.

There are three different ways that mzML file(s) can be generated from imaging data, which depend on both the instrument and the imaging parameters used to acquire the data. The first is a mzML file for every row within the image. This option occurs when images are acquired in raster mode, where the sample holder is moved under the sampling probe continuously for each row and pixels are spectra are acquired at fixed time intervals, such as in desorption electrospray ionisation (DESI) and raster mode MALDI. The second is a single mzML file which contains all of the spectra that make up the MS image. The final way is for each spectrum in the image to be stored in its own mzML file. Options in imzMLConverter exist for each of these situations by selecting the 'File organisation' to be 'Row per file', 'Image per file' and 'Spectrum per file' respectively.

The final piece of information required to be able to reconstruct an image from the mzML file(s), is the relative spatial location for each of the spectra. In the case of raster mode imaging where the the imaged area was rectangular, this is trivial and the image dimensions

can be automatically detected from the mzML files (number of pixels in x is the number of spectra in each mzML file and the number of pixels in y is the number of mzML files). It is equally trivial when a spot mode image has been acquired over a rectangular area, however the user must define the image dimensions. However, it is becoming increasingly common for instruments to allow the user to select arbitrarily shaped regions of interest to acquire data from. Two such instruments that allow this are the ultrafleXtreme (Bruker Daltonics) and the Synapt G2S (Waters). The pixel location for each spectrum in an image acquired on an ultrafleXtreme can be determined from the meta information stored in the mzML file, where each spectrum has an associated name with the format of '0_R00X170Y127' (for region of interest 00, x coordinate 170 and y coordinate 127). Setting up an imaging experiment on the Synapt G2S produces a '*.pat' file, an XML based file containing the start and end coordinate of each row of the image and the pixel size. This information can then be used to assign a spatial location to each spectrum in the image.

Compression

One feature of the mzML specification that is omitted from all other available imzML exports or converters is the use of compression. One of the main benefits cited for the use of imzML over other open formats such as mzML is the disk space savings. As shown in Table S2, imzML only beats compressed mzML when it too is compressed. The use of compression makes the format more usable for its other major goal, data sharing. The reduction in disk space saves costs in both data storage and data transfer, especially when considering transfer across a network such as the Internet. The optimal choice between compressed or uncompressed data depends on the storage medium and processor used. A solid state disk (SSD) is sufficiently fast in returning data that it would be slower to read a smaller amount of data and decompress than it could be to read the uncompressed data. This is also true for HDDs arranged in RAID configurations, as demonstrated in Table S2. As the transfer time of the storage medium used increases, the benefit of using compressed data begins to

outweigh uncompressed in both size and response time, as is shown in the random access, uncompressed data stored on a hard disk drive in Figure S3.

Table S2: Data size of typical MS image as acquired from a QSTAR XL (SCIEX, *.wiff) and a Synapt G2 (Waters, *.raw) in the corresponding proprietary format as well as the open formats mzML and imzML (both with and without compression).

Raw Data Format	Raw Size	mzML Size	Compressed mzML Size	imzML Size	Compressed imzML Size
SCIEX (*.wiff)	773 MB	11.50 GB	5.04 GB	8.67 GB	3.80 GB
Waters (*.raw)	3.13 GB	8.38 GB	1.93 GB	6.30 GB	1.46 GB

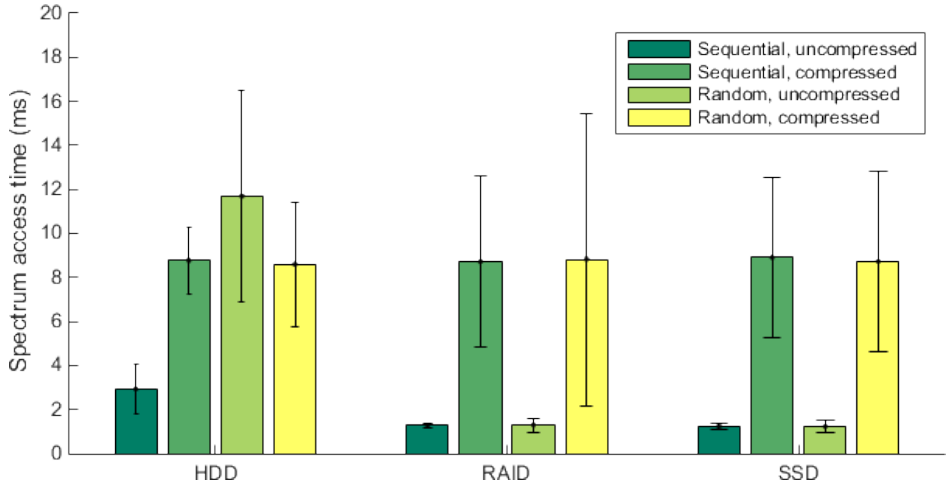


Figure S3: Data access times averaged over accessing 1000 spectra either randomly or sequentially from the SCIEX dataset from Table S2 stored in compressed and uncompressed imzML. Data stored on either hard disk drive (HDD), redundant array of independent disks (RAID) or solid state disk (SSD).

Algorithm S1. Rebinning: generate new axis

Require: Minimum, m_{min}, and maximum, m_{max}, m/z values for new m/z axis
Require: Bin size Δm
1: Generate new m/z axis $M_{\text{rebin}} \leftarrow \{m_{\text{min}}, m_{\text{min}} + \Delta m, m_{\text{min}} + 2\Delta m, ..., m_{\text{max}}\}$

Memory Efficient Ion Image Generation

Generation of individual ion images does not require the whole dataset to be loaded into memory. Instead only the data points that fall within the peak boundaries (m_{min} and m_{max}) are required so that one of the methods shown in Figure S12 can be applied. This can also be

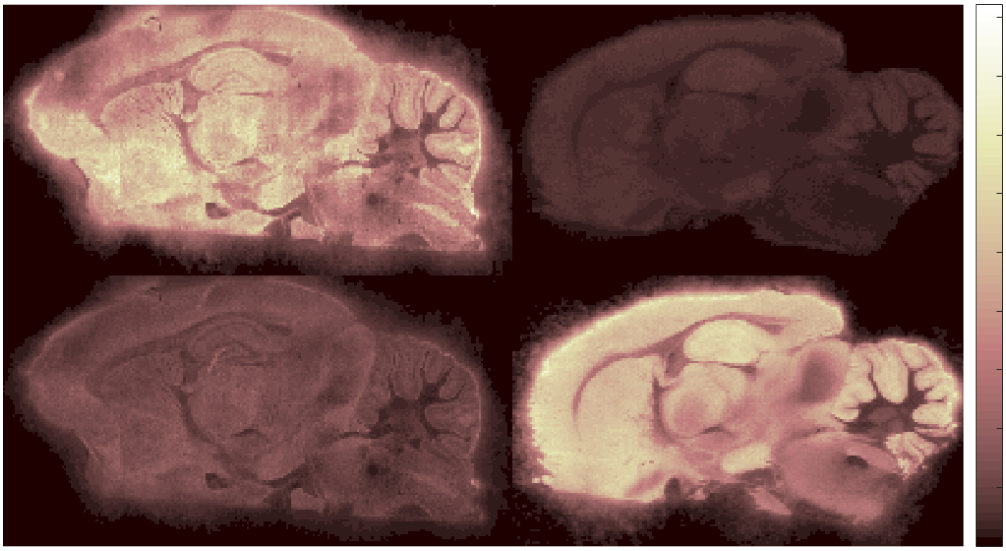


Figure S4: Replication of Figure 2a from,[S25] generated by combining two separate imzML files together and then generating two ion images (PC 32:0 $[M+K]^+$ m/z 772 top row and PC 32:0 $[M+Na]^+$ m/z 756 bottom row).

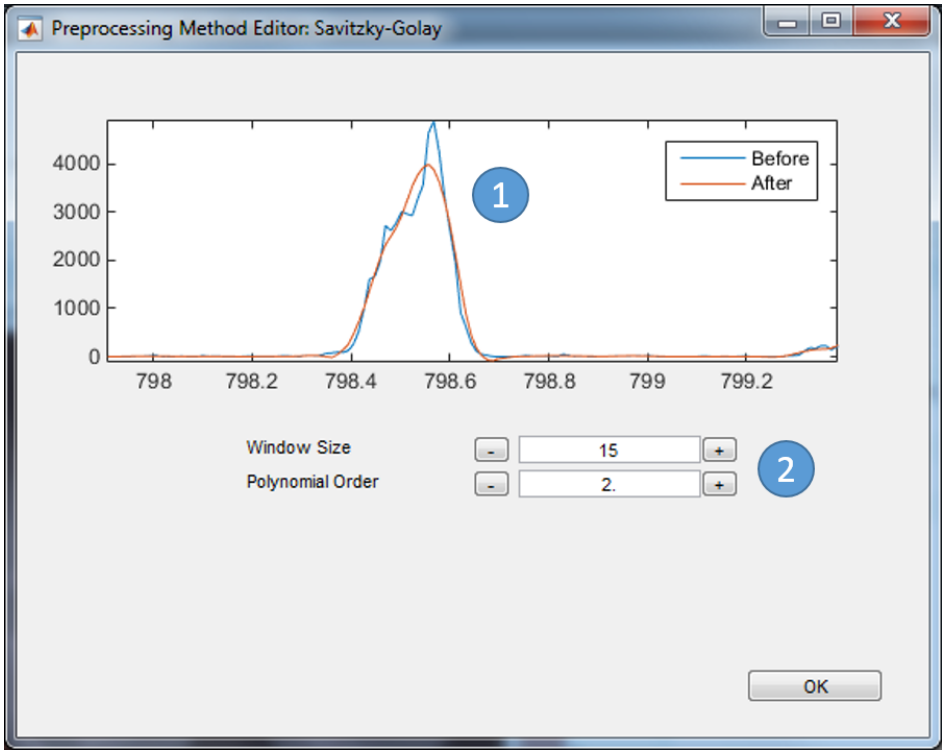


Figure S5: Screenshot of the real time update of preprocessing effects. 1) Raw spectrum ('Before') overlayed with the preprocessed spectrum ('After'). 2) Preprocessing method's parameters. Changing these values updates the 'After' spectrum so their effect is visualised in real time.

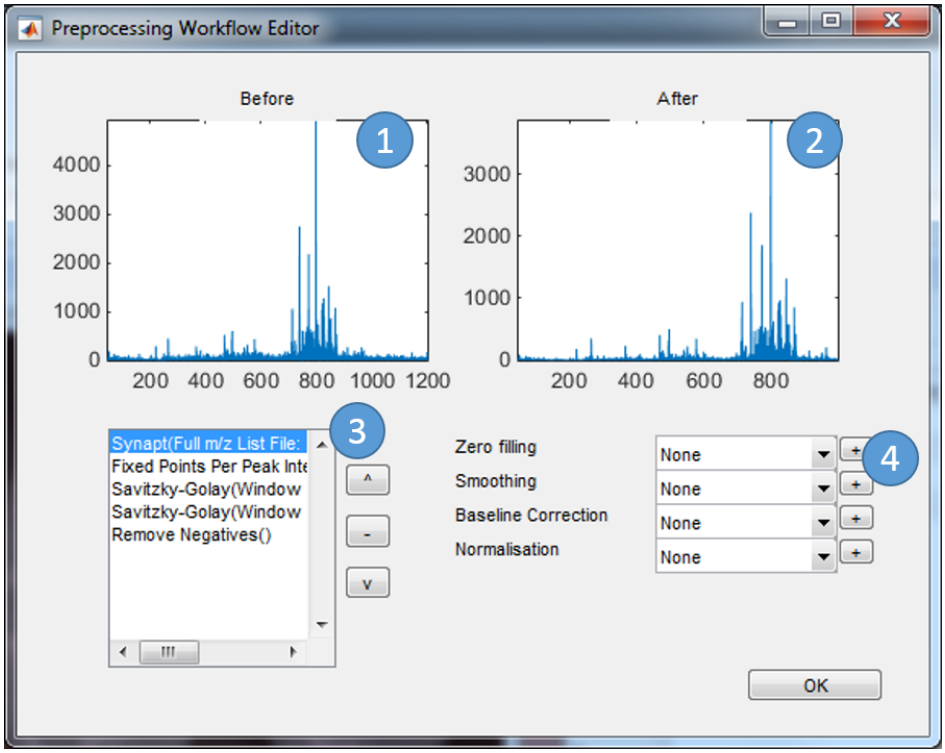


Figure S6: Screenshot of the workflow generation. 1) Spectrum prior to any preprocessing applied. 2) Spectrum after full preprocessing workflow applied. 3) Preprocessing workflow, a list of methods and parameters in a set order. 4)

Algorithm S2. Rebinning: apply new axis

Require: Spectrum S with m/z axis M
Require: New m/z axis M_{rebin}
$j \leftarrow 0$
$h \leftarrow \Delta m/2$
Create new spectrum S' to have same size as M_{rebin}
for $i \leftarrow 0$ to $|S| - 1$ **do**
  if $M(i) < m_{min}$ **then**
    continue
  end if
  if $M(i) > m_{max}$ **then**
    break
  end if
  $t_1 \leftarrow M(i) - h$
  $t_2 \leftarrow M(i) + h$
  while $j < |M_{\text{rebin}}|$ and $M_{\text{rebin}}(j) < t_1$ **do**
    $j \leftarrow j + 1$
  end while
  if $j < |M_{\text{rebin}}|$ and $M_{\text{rebin}}(j) < t_2$ **then**
    $S'(j) \leftarrow S'(j) + S(i)$
  end if
end for

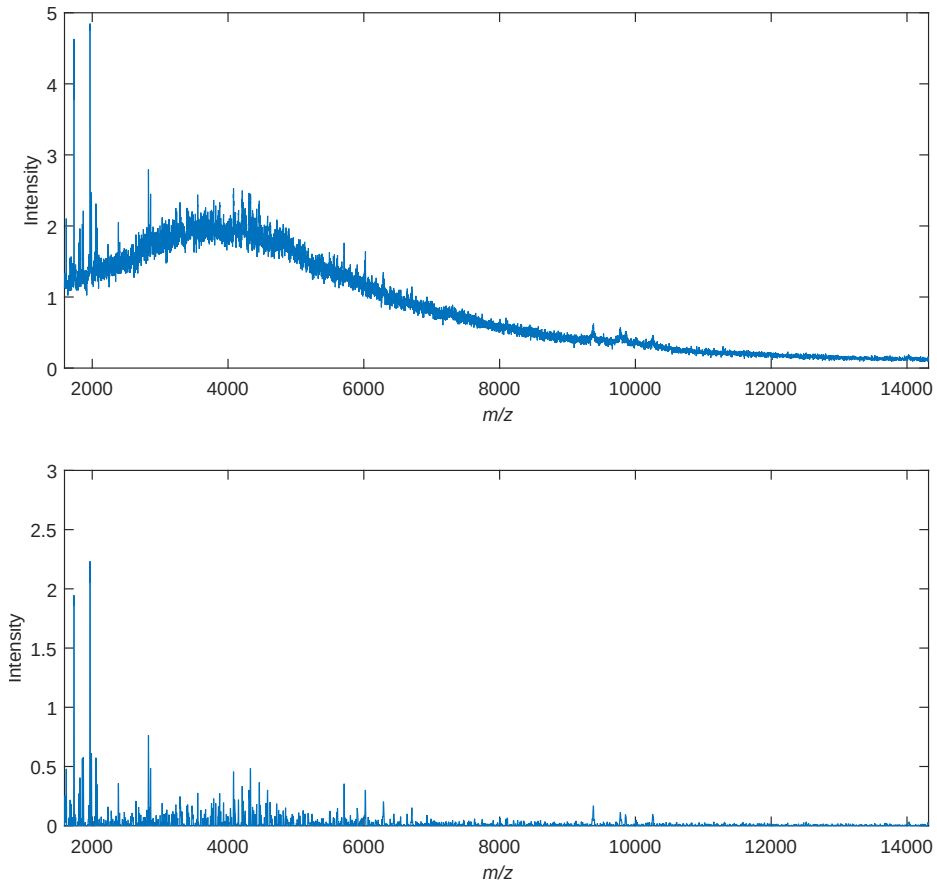
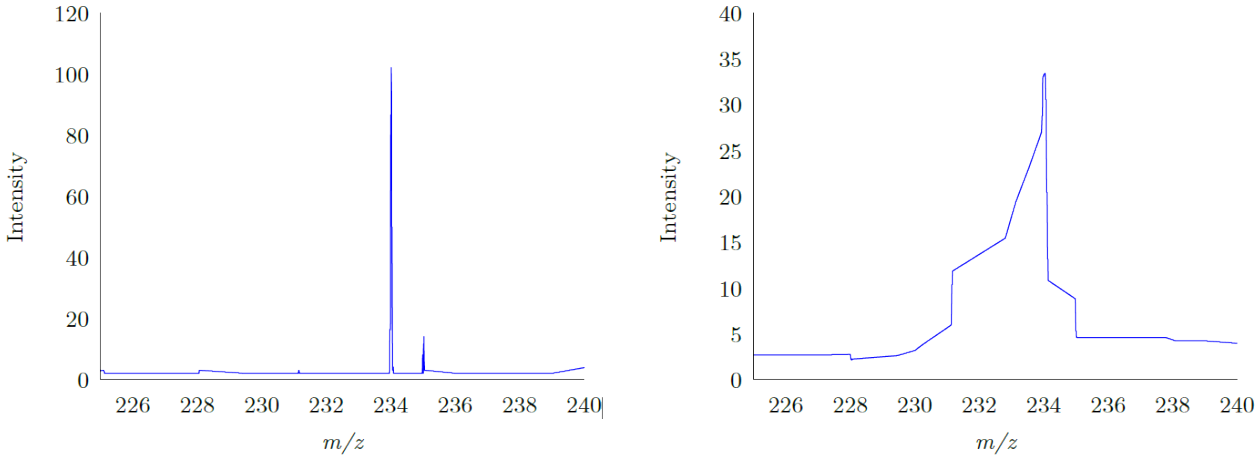


Figure S7: Example preprocessing workflow applied to publicly available dataset. (Top) A representative spectrum from 3D_Mouse_Pancreas.imzML[S26] (spectrum identifier in the imzML file "spectrum=26695"). (Bottom) Spectrum after applying the following preprocessing workflow: 1) Savitzky-Golay smoothing with window size 11 and polynomial order 2; 2) Savitzky-Golay smoothing with window size 11 and polynomial order 2; 3) Median baseline correction with window size 100; 4) Remove negatives baseline correction.

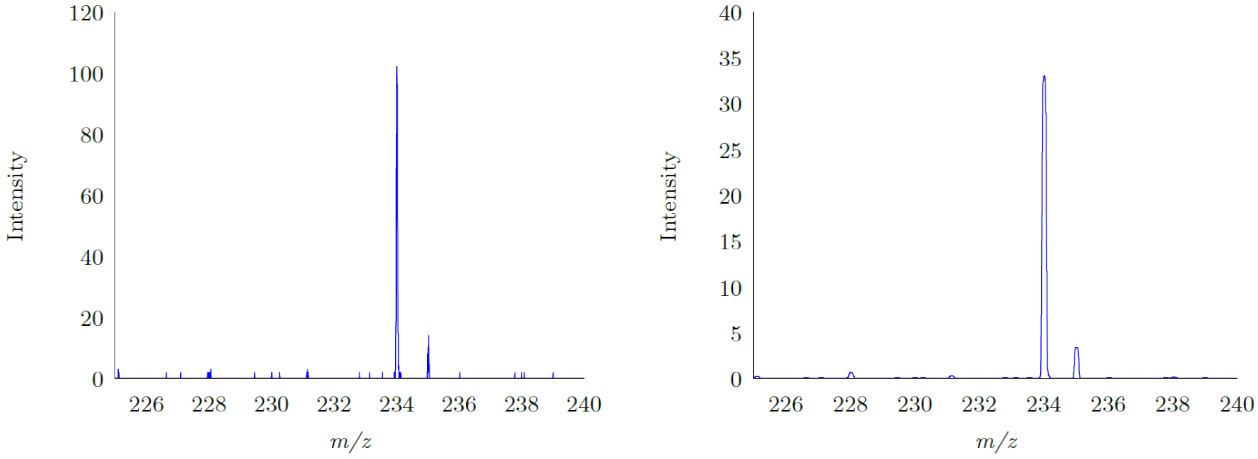
Algorithm S3. Calculate detector sampling interval for QSTAR from spectrum

Require: m/z axis M
$T \leftarrow \sqrt{M}$
$\Delta T \leftarrow T(1, ..., |T|) - T(0, ..., |T| - 1)$
$\Delta T' \leftarrow \{\}$
for each Δt in ΔT **do**
  if $\Delta t < \frac{3}{2} \min \Delta T$ **then**
    Insert Δt into $\Delta T'$
  end if
end for
$\delta \leftarrow \text{mode}\{\Delta T'\}$



(a) Raw spectrum

(b) Smoothed raw spectrum



(c) Zero filled spectrum

(d) Smoothed zero filled spectrum

Figure S8: Spectrum displayed in sparse format (a), where peak shapes are distorted and artificial baseline is displayed but is not part of the data, which is then corrupted through application of window based smoothing (b). Spectrum displayed with zero values replaced (c) and correctly smoothed (d).

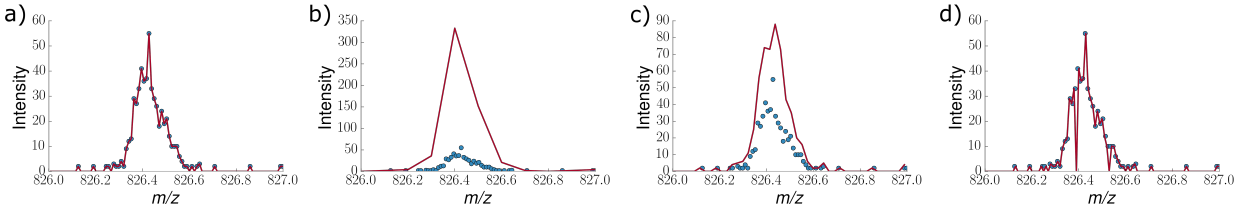


Figure S9: The effects of different bin sizes applied to the peak at m/z 826 in a single spectrum. The recorded data from the mass spectrometer shown as blue circles. Red line shows the standard representation of a mass spectrum (linear interpolation between each data point as a guide for the eye) with a) the raw data b) data binned at 0.1 m/z c) data binned at 0.023 m/z d) data binned at 0.01 m/z.

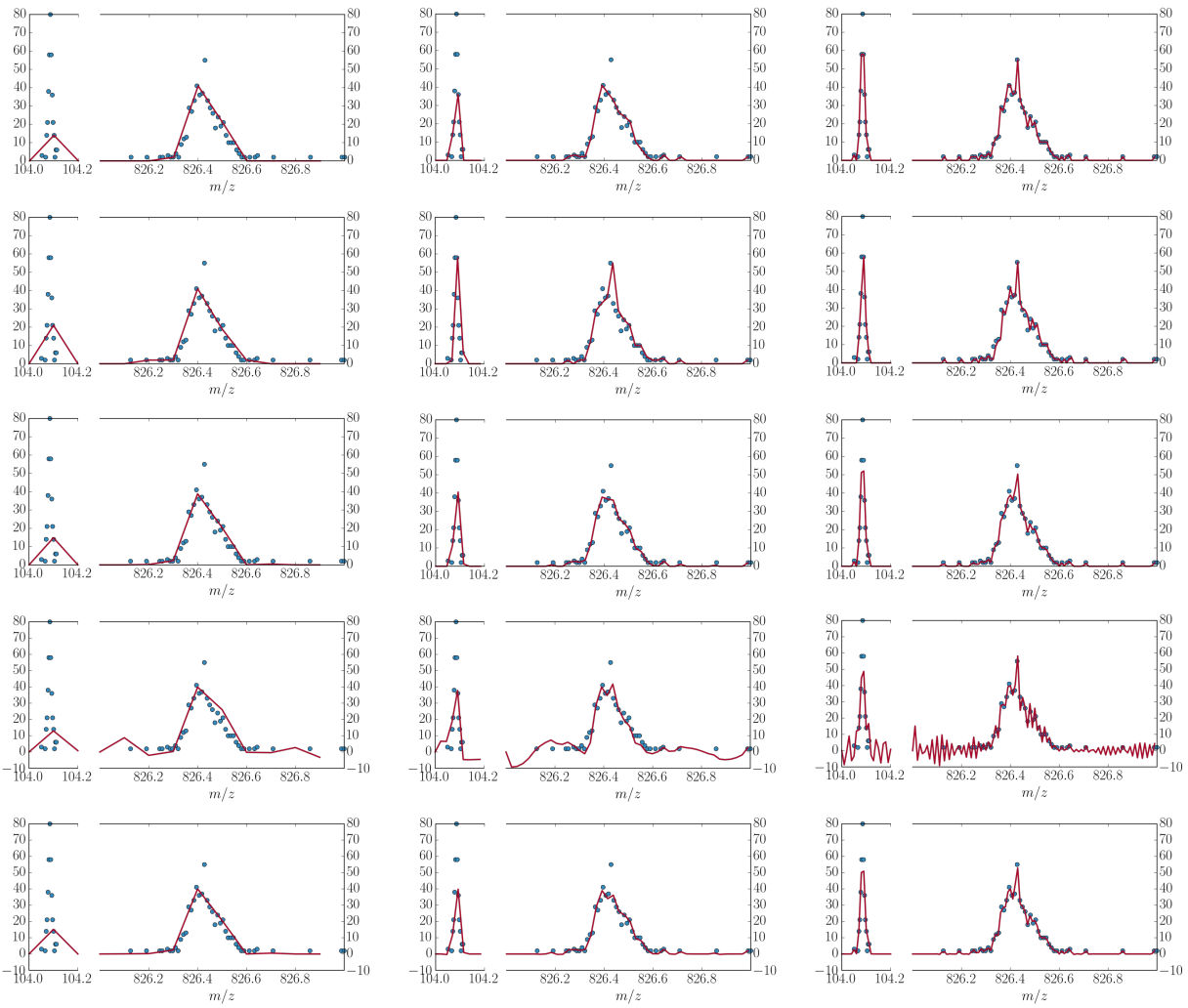


Figure S10: The effects of different interpolation methods (row 1 - nearest neighbour, row 2 - zero order spline, row 3 - linear, row 4 quadratic spline and row 5 cubic spline) with different bin sizes (column 1 $\Delta m = 0.1$ m/z, column 2 $\Delta m = 0.023$ m/z and column 3 $\Delta m = 0.01$ m/z).

Algorithm S4. Generate new axis for QSTAR

Require: Minimum, m_{min}, and maximum, m_{max}, m/z values for new m/z axis
Require: Detector sampling interval δ
1: $t_{min} \leftarrow \sqrt{m_{\text{min}}}$
2: $t_{max} \leftarrow \sqrt{m_{\text{max}}}$
3: Generate time axis $T \leftarrow \{t_{\text{min}}, t_{\text{min}} + \delta, m_{\text{min}} + 2\delta, ..., t_{\text{max}}\}$
4: $M_{\text{qstar}} \leftarrow T^2$

Algorithm S5. Set union of all m/z bins

1: $M_g \leftarrow \{\}$
2: **for each** spectrum S in dataset **do**
3: Get m/z axis M for current spectrum S
4: $M_g \leftarrow M_g \cup M$
5: **end for**

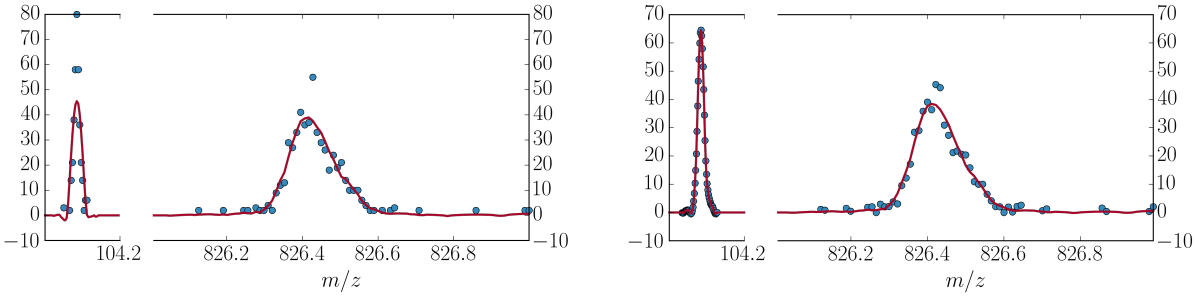


Figure S11: Demonstration of the effects of Savitzky-Golay smoothing (window size 15) applied to data on data rebinning in the detector domain (left) and data interpolated to ensure a peak spans approximately 30 mass bins across the whole mass range (right).

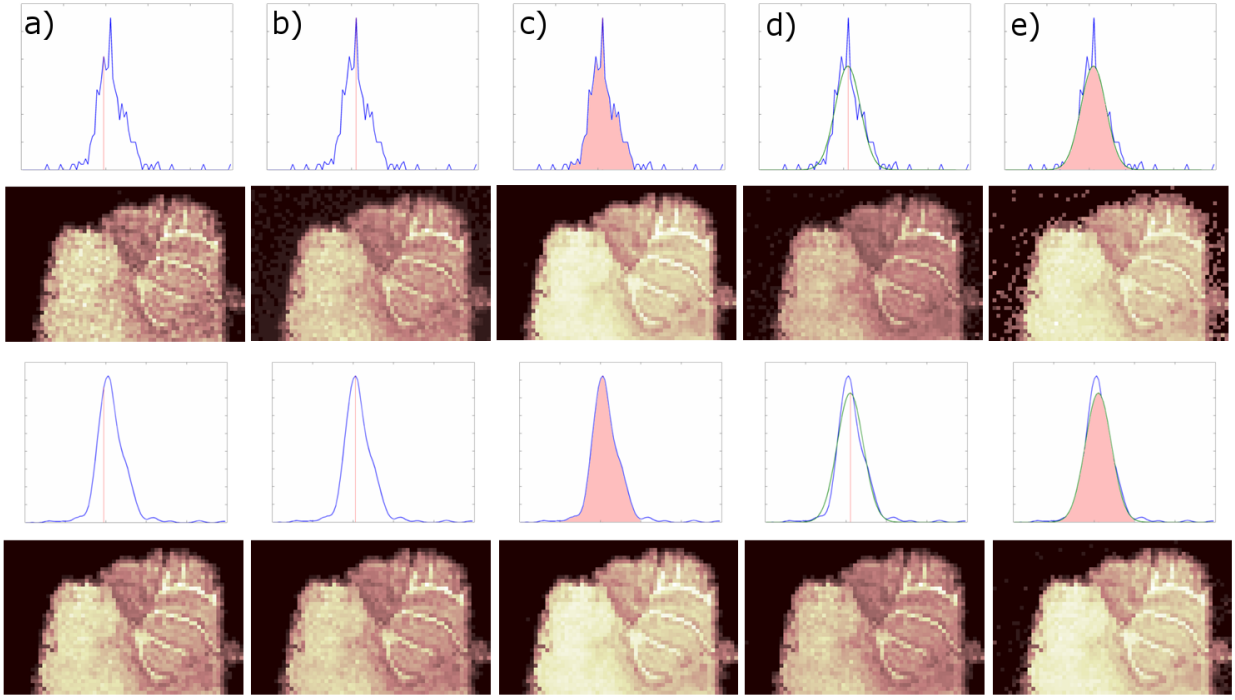


Figure S12: Comparison of ion images generated with various methods with and without preprocessing applied. a) Extract values at specific m/z location determined from total spectrum. b) Maximum in m/z range. c) Integrate over m/z range. d) Maximum in fitted function (Gaussian function used). e) Integrate over fitted function (Gaussian used). Row 1) Raw data shown in blue trace. Extracted value(s) shown in pink. Fitted function shown in green trace. Row 2) Generated ion images from raw data in Row 1). Row 3) Preprocessed data shown in blue trace. Extracted value(s) shown in pink. Fitted function shown in green trace. Row 4) Generated ion images from raw data in Row 3).

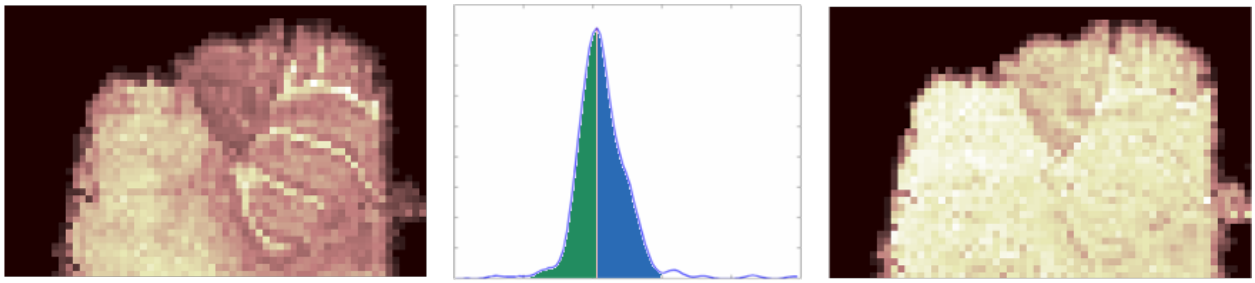


Figure S13: Comparison of ion images generated by summing values a) highlighted in green and c) highlighted in blue.

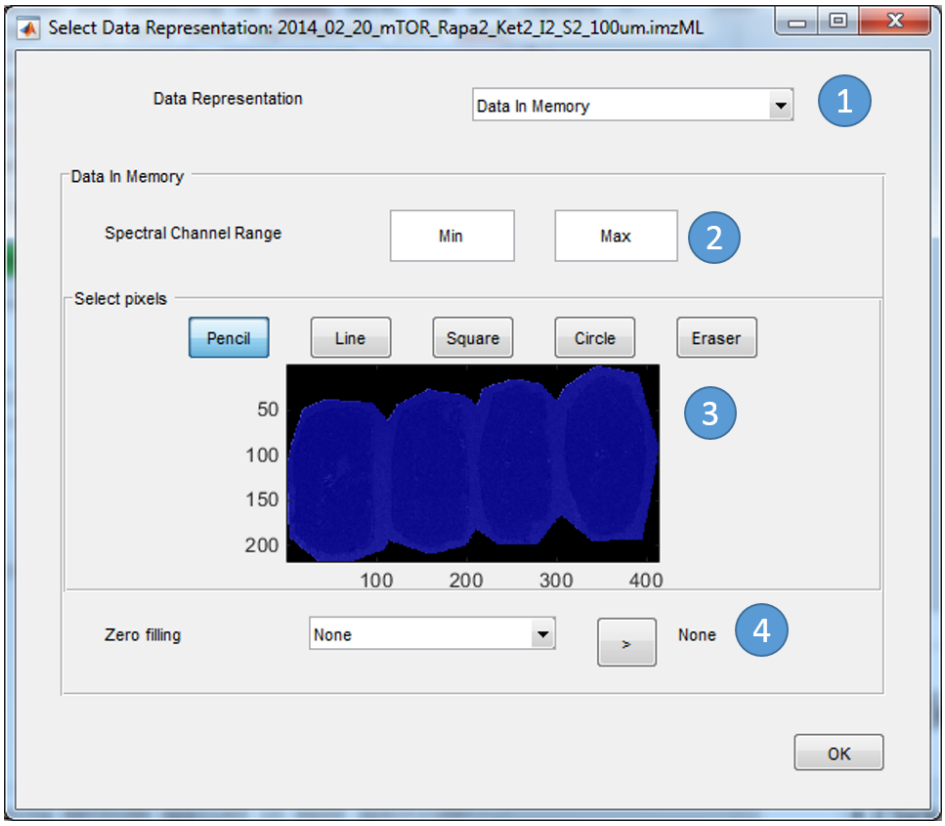


Figure S14: Interface presented when opening a dataset in SpectralAnalysis providing the user with options for how to handle large data. 1) Option between loading the data into memory or leaving the data on disk and using memory efficient methods (discussed in the main manuscript). 2) Limit the spectral domain. 3) Select a region of interest to load data within only that area. 4) Ensure consistent m/z axis to enable loading the data into memory and for optional data size reduction.

taken a step further, and since the calculation of an intensity at a given pixel is completely independent of all other pixels, only one spectrum is required in memory at a given point in time. This significantly reduces the amount of memory required, as a single spectrum ranges from 100s kB to 1-2 MB, compared to the 10s of GBs for the whole dataset. In many cases it is desirable to preprocess the data prior to ion image generation. The algorithm presented in Algorithm S6 presents a memory efficient method of generating ion images from preprocessed data. Each spectrum is loaded in sequentially, preprocessed and then the data points within peak limits are extracted and an intensity is generated based on the image generation method of choice. The spectrum can then be removed from memory before the next is loaded in. This reduces the amount of memory required to the size of a single spectrum, plus the size of the ion image(s) to be generated, which is orders of magnitude smaller than the whole data, allowing TBs of data to be visualised on even the most memory constrained systems.

Algorithm S6. Memory Efficient Ion Image Generation

Require: Peak limits $m_{\min}$ and $m_{\max}$
Require: Preprocessing workflow $W \leftarrow \{w_1(S), ..., w_n(S)\}$ containing n preprocessing methods
Require: Image generation method $G(S, m_{\min}, m_{\max})$ from Figure S12
1: $I \leftarrow \{\}$
2: **for each** spectrum index i in dataset **do**
3: Read in spectrum S_i from disk
4: **for each** preprocessing method w in W **do**
5: $S_i \leftarrow w(S_i)$
6: **end for**
7: Get spatial location of spectrum from header information (x, y, z)
8: $I(x, y, z) \leftarrow G(S_i, m_{\min}, m_{\max})$
9: **end for**

Memory Efficient Spectral Representation Generation

Peak detection is often performed on spectral representations of the data.[S27] As above, these only require a single spectrum to be loaded into memory at once and can be generated in a memory efficient manner using Algorithm S9. It is possible to generate multiple representations at once, requiring only a single pass through of the data, by including additional update methods after line 8 of Algorithm S9. This provides a memory efficient method of generating

all spectral representation proposed by McDonnell *et al.*[S27] for optimal peak detection in a given dataset.

Algorithm S7. $U_t(S_R, S)$ Update method for total/mean spectrum generation

Require: $|S_R| = |S|$
1: **for each** index i in S_R **do**
2: $S_R(i) \leftarrow S_R(i) + S(i)$
3: **end for**

Algorithm S8. $U_b(S_R, S)$ Update method for basepeak spectrum generation

Require: $|S_R| = |S|$
1: **for each** index i in S_R **do**
2: $S_R(i) \leftarrow \max(S_R(i), S(i))$
3: **end for**

Algorithm S9. Memory Efficient Spectral Representation Generation

Require: Preprocessing workflow $W \leftarrow \{w_1(S), ..., w_n(S)\}$ containing n preprocessing methods
Require: Spectral representation update method $U(S_R, S)$ from Algorithm S7 or S8
1: $S_R \leftarrow \{\}$
2: $n \leftarrow 0$
3: **for each** spectrum index i in dataset **do**
4: Read in spectrum S_i from disk
5: **for each** preprocessing method w in W **do**
6: $S_i \leftarrow w(S_i)$
7: **end for**
8: $S_R \leftarrow U(S_R, S_i)$
9: $n \leftarrow n + 1$
10: **end for**
11: If mean spectrum required, $S_R \leftarrow S_R/n$

Memory Efficient Datacube Generation

By combining the above methods it is possible to reduce the MS image to a 'datacube' in a memory efficient manner, using Algorithm S10. In this case, only a single spectrum and the datacube is required to be in memory at any one point in time. This allows reduction of data to peak lists without a limitation applied to the number of peaks retained.

The algorithm as it is presented reduces and loads the data into memory, however this can also be used to write the reduced data to disk by altering line 10 in Algorithm S10 to be a disk write instead of a matrix update. In this case only a single spectrum is required to be in memory, making this process feasible on memory constrained systems where the datacube

Algorithm S10. Memory Efficient Datacube Generation

Require: Preprocessing workflow $W \leftarrow \{w_1(S), ..., w_n(S)\}$ containing n preprocessing methods
Require: Image generation method $G(S, m_{\min}, m_{\max})$ from Figure S12
Calculate spectral representation S_R using Algorithm S9
Peak pick on S_R using chosen peak detection method to get m/z limits $M_{\min}$ and $M_{\max}$
$D \leftarrow \{\}$
for each spectrum index i in dataset **do**
    Read in spectrum S_i from disk
    for each preprocessing method w in W **do**
        $S_i \leftarrow w(S_i)$
    end for
    for each peak index j in $M_{\min}$ **do**
        $D(i, j) \leftarrow G(S_i, M_{\min}(j), M_{\max}(j))$
    end for
end for

is larger than that of the RAM. The methods for handling large datasets described in the main manuscript can then be employed to visualise portions of the data.

Algorithm S11. Memory Efficient Scaling Generation

Require: Preprocessing workflow $W \leftarrow \{w_1(S), ..., w_n(S)\}$ containing n preprocessing methods
Require: Image generation method $G(S, m_{\min}, m_{\max})$ from Figure S12
Calculate spectral representation S_R using Algorithm S9
Peak pick on S_R using chosen peak detection method to get m/z limits $M_{\min}$ and $M_{\max}$
$N \leftarrow 0$
$M \leftarrow 0$
$M' \leftarrow 0$
for each spectrum index i in dataset **do**
    Read in spectrum S_i from disk
    for each preprocessing method w in W **do**
        $S_i \leftarrow w(S_i)$
    end for
    if Shift variance scaling **then**
        Read in and preprocess spectrum as above for the spectrum that is horizontally and/or vertically shifted S'_i
        $S_i \leftarrow S_i - S'_i$
    end if
    $N \leftarrow N + 1$
    $\delta \leftarrow S_i - M$
    $M \leftarrow M + \frac{\delta}{N}$
    $M' \leftarrow M' + \delta(S_i - M)$
end for
if Auto-scaling or shift variance scaling **then**
    $scaling \leftarrow \sqrt{\frac{M'}{(n-1)}}$
elseRoot mean scaling
    $scaling \leftarrow \sqrt{M}$
end if

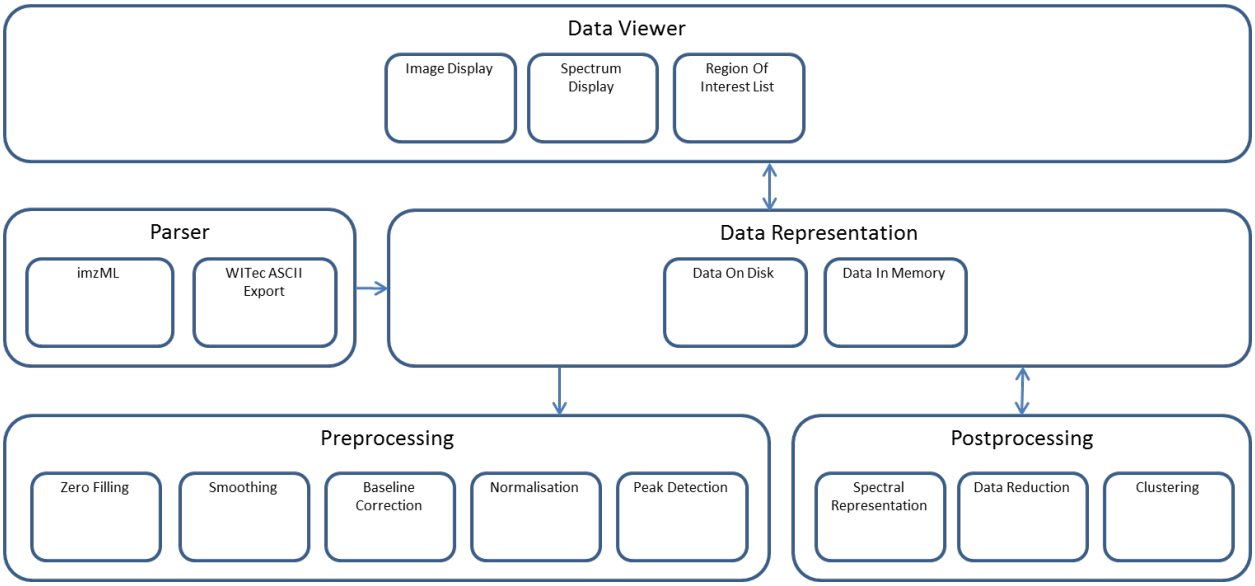


Figure S15: Block diagram describing the software interaction. Easily extensible sections to add additional functionality are 'Parser', 'Preprocessing' and 'Postprocessing'.

References

(S1) Race, A. M.; Styles, I. B.; Bunch, J. *Journal of proteomics* **2012**, *75*, 5111–5112.

(S2) Analyst QS Administrator's Guide. Applied Biosystems, 2004.

(S3) TissueView. Applied Biosystems, 2010.

(S4) fleximaging 3.0 User Manual. Bruker Daltonics, 2011.

(S5) flexanalysis 3.3 User Manual. Bruker Daltonics, 2009.

(S6) Thermo Xcalibur Qualitative Analysis User Guide. Thermo Fisher Scientific, 2010.

(S7) Thermo ImageQuest Version 1.0.1 User Guide. Thermo Fisher Scientific, 2009.

(S8) Thermo Xcalibur Acquisition and Processing User Guide. Thermo Fisher Scientific, 2010.

(S9) MassLynx 4.1 Getting Started Guide. Waters Corporation, 2005.

(S10) Hosokawa, N.; Sugiura, Y.; Setou, M. *Imaging Mass Spectrometry*; Springer, 2010; pp 113–126.

(S11) BioMAP 3x. Novartis, 2005.

(S12) Bemis, K. D.; Harry, A.; Eberlin, L. S.; Ferreira, C.; van de Ven, S. M.; Mallick, P.; Stolowitz, M.; Vitek, O. *Bioinformatics* **2015**, btv146.

(S13) Cardinal: Analytic tools for mass spectrometry imaging. Kyle D. Bemis and April Harry, 2016.

(S14) Mantini, D.; Petrucci, F.; Pieragostino, D.; Del Boccio, P.; Di Nicola, M.; Di Ilio, C.; Federici, G.; Sacchetta, P.; Comani, S.; Urbani, A. *BMC bioinformatics* **2007**, *8*, 101.

(S15) Klinkert, I.; Chughtai, K.; Ellis, S. R.; Heeren, R. *International Journal of Mass Spectrometry* **2014**, *362*, 40–47.

(S16) Datacube Explorer User Manual. FOM Institute AMOLF, 2014.

(S17) Paschke, C.; Leisner, A.; Hester, A.; Maass, K.; Guenther, S.; Bouschen, W.; Spengler, B. *Journal of The American Society for Mass Spectrometry* **2013**, *24*, 1296–1306.

(S18) Källback, P.; Nilsson, A.; Shariatgorji, M.; Andrén, P. E. *Analytical Chemistry* **2016**,

(S19) Robichaud, G.; Garrard, K. P.; Barry, J. A.; Muddiman, D. C. *Journal of The American Society for Mass Spectrometry* **2013**, *24*, 718–721.

(S20) MSiReader User's Manual. NC State University, 2012.

(S21) Parry, R. M.; Galhena, A. S.; Gamage, C. M.; Bennett, R. V.; Wang, M. D.; Fernández, F. M. *Journal of The American Society for Mass Spectrometry* **2013**, *24*, 646–649.

(S22) Rübel, O.; Greiner, A.; Cholia, S.; Louie, K.; Bethel, E. W.; Northen, T. R.; Bowen, B. P. *Analytical chemistry* **2013**, *85*, 10354–10361.

(S23) SCiLS / SCiLS Lab - The statistical analysis software. `http://scils.de/software/`, 2014; Accessed: 21/05/2014.

(S24) Chambers, M. C.; Maclean, B.; Burke, R.; Amodei, D.; Ruderman, D. L.; Neumann, S.; Gatto, L.; Fischer, B.; Pratt, B.; Egertson, J.; Hoff, K.; Kessner, D.; Tasman, N.; Shulman, N.; Frewen, B.; Baker, T. A.; Brusniak, M.-Y.; Paulse, C.; Creasy, D.; Flashner, L.; Kani, K.; Moulding, C.; Seymour, S. L.; Nuwaysir, L. M.; Lefebvre, B.; Kuhlmann, F.; Roark, J.; Rainer, P.; Detlev, S.; Hemenway, T.; Huhmer, A.; Langridge, J.; Connolly, B.; Chadick, T.; Holly, K.; Eckels, J.; Deutsch, E. W.; Moritz, R. L.; Katz, J. E.; Agus, D. B.; MacCoss, M.; Tabb, D. L.; Mallick, P. *Nature biotechnology* **2012**, *30*, 918–920.

(S25) Carter, C. L.; McLeod, C. W.; Bunch, J. *Journal of the American Society for Mass Spectrometry* **2011**, *22*, 1991–1998.

(S26) Oetjen, J.; Veselkov, K.; Watrous, J.; McKenzie, J. S.; Becker, M.; Hauberg-Lotte, L.; Kobarg, J. H.; Strittmatter, N.; Mróz, A. K.; Hoffmann, F.; Trede, D.; Palmer, A.; Schiffler, S.; Steinhorst, K.; Aichler, M.; Goldin, R.; Guntinas-Lichius, O.; Eggeling, F.; Thiele, H.; Maedler, K.; Walch, A.; Maass, P.; Dorrestein, P. C.; Takats, Z.; Alexandrov, T. *GigaScience* **2015**, *4*, 1–8.

(S27) McDonnell, L. A.; Van Remoortere, A.; De Velde, N.; Van Zeijl, R. J.; Deelder, A. M. *Journal of the American Society for Mass Spectrometry* **2010**, *21*, 1969–1978.