A Gaussian Mixture Method for Modeling and Assessment of MALDI-TOF Mass Spectra for Stable Isotope Standard Quantification

John Christian Givhan Spainhour

A dissertation proposal submitted to the faculty of the Medical University of South Carolina in partial fulfillment of the requirement for the degree of Doctor of Philosophy in the College of Graduate Studies.

Department of Public Health Sciences

2014

Approved by:

Dr. Viswanathan Ramakrishnan, Ph.D Chair

Dr. John H. Schwacke, Ph. D.

Dr. Elizabeth G. Hill, Ph. D.

Dr. Juan Carlos Q. Velez, M. D.

Dr. Michael G. Janech, Ph. D

ACKNOWLEDGMENTS

I dedicate this work to my wife, Laurel E. C. Spainhour, and my grandfather, Thomas Bartram Givhan. My grandfather has always been an inspiration in many ways and has provided both support and insight in how to approach many of life's challenges. My wife has always reminded me of the end goal and that approaching any goal requires patience and will. I am grateful for their inspiration and patience.

I wish to thank a number of people for without whom this dissertation would not be possible.

Dr. Ramakrishnan for his help and guidance while finishing, work in writing and thinking through/approach and idea, support to finish and staying focused without going off to chase new ideas
Dr. Schwacke for his guidance during my first years at MUSC, the original idea for this research and willing ness to see this project through
Dr. Hill for listening to crazy ideas and making me go back and explain them until they make sense and to help strengthen my grasp of the theoretical statistic foundations underpinning this work
Dr. Velez for literary and scientific guidence that helped shape both experimental design and writing from grand ideas to concrete plans and products
Dr. Janech for his guidance at the bench, in writing and helping to cross the bridge between bench work and computation implementation

I would also like to mention Dr. Arthur, everyone in his lab and the Nephrology Proteomics group for their help and encouragement bench-side. This work would be incomplete without their assistance.
June Watson, Dr. Wolf, Larry Harbin, Dr. Lackland and rest of PHS and DBBE for their help, advice, support and opportunities for collaboration
Departments DBBE, BMB, and Division of Nephrology

I wish to thank my family and friends, especially my wife, who has been a pillar of support and without whom, I would not have finished.

Laurel

Mom and Dad, Ms. E. Givhan, Mr. T. B. Givhan, Mr. & Mrs. A. Givhan, Lauren, Mr. & Mrs. D. Spainhour, Stephen, Matt & Sarah Beth

Dr. C. Dull, Dr. C. Engels, Mr. & Mrs. A. Miller, SSgt. & Mrs. M. Charles, Mr. A. Creek, Dr. & Mrs. J. Creek, Mr. M. Owens, Ms. K. Logerman, Ms. N. New, Ms. K. Nicholas, Mrs. R. Carroll, Mrs. C. Ellerbe, Dr. C. Chiuzan, Dr. N. New, Dr. M. Stefanik, Mr. G. Chaffins, Mr. J. Small, Dr. J. Saunders, Dr. A. Nida, Drs. R. & N. Trager, Drs. M. & M. Shotwell, Dr. A. Richards, Dr. T. Qin, Dr. A. Tsoi, G. M. Tiger Kim, PhD., Dr. P. Mirabito, Dr. H. Croom, and Dr. W. Bonds

B.O.H.I.C.I.A., G.S.H & Van Winkle
To the person(s) I missed, thanks.
Finally the BTBBR Training grant and the Division of Nephrology for financial support while at MUSC.

Table of contents

Believe nothing, no matter where you read it or who said it, not even if I have said it, unless it agrees with your own reason and your own common sense.

-Anonymous

Our minds are finite, and yet even in these circumstances of finitude we are surrounded by possibilities that are infinite, and the purpose of life is to grasp as much as we can out of that infinitude.

-Alfred North Whitehead

Abstract

The use of mass spectrographic analysis in the fields of proteomics has grown to include both identification of materials in samples but also quantification of those samples. Here the Gaussian mixture model method for SIS peptide quantification in MALDI-TOF mass spectrum data is proposed and validated against past methodologies of quantification. The Gaussian mixture model takes advantage of previously known information such as the mass of the peptide being quantified and an estimation of that peptide's isotopic distribution to estimate peak area and baseline error adjustment of MALDI-TOF peptide data. This method is compared against both Peak Intensity and Riemann sum area under the curve and is shown to be at least equivalent in estimation performance while being able to quantify convolved peptides with similar levels of accuracy, where past quantification methods cannot be used. A method is described for the simulation of MALDI-TOF MS data and it is used to test the Gaussian mixture model under a number of extremes to show how this method functions. The Gaussian mixture model is shown to estimated peak area within allowable levels of error under a number of conditions. The Gaussian mixture model shows a sensitivity to signal to noise ratio whereas the ratio is decreased by either peak widening or decreases in peak area, the estimates become more error prone but are still within acceptable bounds of error found in the current literature. Improvements to the Gaussian mixture model algorithm were implemented as alternative methods for crossing the search space created by user input. Finally a maximum likely hood estimator using an expectation-maximization algorithm was implemented to estimate Gaussian mixture model parameters from the data. We show that it is possible to collect model parameters from MS data to construct a Gaussian mixture model for peptide quantification.

1. Introduction and significance

1.1. Overview and Specific Aims

The investigation of the extra cellular processing of angiotensin as a part of the study of the Renin- Angiotensin system (RAS) presents several unique difficulties. The interactions of multiple peptide substrates from a single source that compete for the same set of enzymes creates a robust network of interactions that has yet to be fully explored. Using the absolute quantification (AQUA) method of peptide quantification, relative amounts of peptides can be measured and true amounts estimated in mixtures of peptides, such as in RAS. New methods and tools for analysis of matrix-assisted laser desorption ionization-time of flight (MALDI-TOF) mass spectrometry are required to help further our understanding of results from experiments with complex networks of substrate/enzyme interactions. Complex mixtures of peptides containing overlapping peptide data from these experiments, in the form of overlapping ion currents, cannot be fully interpreted using traditional methods. A Gaussian mixture model can be used to estimate a given spectrum of a known peptide or combination of peptides if they overlap in the mass spectrum. Subsequently, this can be used to construct a model for the area under the peaks in experimental data as well as for estimating the resolution of the spectrum allowing for the quantification of all peptides in the sample. This information can then be used to reconstruct a substrate/enzyme network for future study, modeling, and drug or therapy target identification.

The main focus of this dissertation is to propose a methodology for peptide quantification in MALDI-TOF MS using stable isotope standards (SIS). This methodology is based on using a Gaussian mixture to model the peaks of the spectra and estimate the area under the peaks for a given peptide and its SIS counterpart. The Gaussian mixture model is compared to other methods of quantification, used to analyze simulated data and implemented using an expectation-maximization algorithm for finding

the maximum likelihood estimate of the model parameters, allowing for confidence interval calculation. This methodology can be extended to the analysis of other compounds and used as a foundation for future tool development in other fields of research.

Specifically, the objective of the dissertation is to describe and validate the Gaussian mixture model method for estimating peak area in MALDI-TOF mass spectra as an equivalent way to analyze SIS peptide measurement, show the limitations of the method through examination across a wide range of simulated spectra conditions, explore and compare multiple ways of implementing both the exploration of the search space and fitting of the model to data, provide methods for estimating the uncertainty in the estimation and produce a user-friendly R package for the implementation of the method.

Specific Aim 1: Describe and preform experimental validation of the Gaussian mixture model as a method of peak area estimation and compare it with other methods of peak quantification for sample quantification using SIS peptides.

Specific Aim 2: Develop a method for simulating spectra for Gaussian mixture model for stress testing the method over possible real life extremes, illustrating peak estimation over a variety of conditions.

Specific Aim 3: Provide uncertainty measures for the estimates from the Gaussian mixture model using a maximum likelihood approach.

1.2 Background

1.2.1 SIS Quantification

Quantitative proteomics is fundamental to the study of organisms at the systems level [1]. This includes identification and quantification of biomarkers, posttranslational modifications and the dissection of biochemical pathways such as the angiotensin signal extracellular preprocessing pathway. Mass spectrographic (MS) based techniques have advanced the field far beyond what is capable with traditional techniques such as 2D gels or western blots due to their capacity to identify and quantify thousands [5, 6] of proteins and posttranscriptional modifications in a single experiment [2]. This is possible due to the ability of MS based techniques to separate proteins regardless of abundance or solubility, unlike 2D gels, and to identify these proteins from their unique ion fragmentation profiles. The ability to quantify proteins is hindered by inherit physical and chemical properties associated with proteins. Differences in the charge, hydrophobicity, or posttranslational modification are a few of these chemical properties that affect the ion formation and time of flight (TOF) of a sample peptide or protein mixture. These factors that can affect peptide flyability, which affects the peptide signal strength, interfering with how well we can quantify the peptide in a sample. If the work flow of these experiments are not carefully managed, possible loss of some of the sample during preparation or contamination from unusual sources may occur that further complicate the quantification. Due to these considerations internal standards, in this case a SIS peptide labeled with heavy isotopes of Carbon, Nitrogen and Oxygen, are preferred during quantification experiments.

Methods of quantification include those based on stable isotope dilution theory and on label free methods. Stable isotope dilution theory is based on the concept that a

stable isotope labeled protein or peptide would behave exactly in the same manner as an unlabeled peptide during MS analysis. Since the mass difference between the labeled and unlabeled samples can be detected through MS, a comparison can be made based on their signal intensities and thus quantification can be achieved. Label free methods require the comparison of multiple experiments to either compare peak intensities or for counting the spectra in which a given peptide/protein of interest appears. In this dissertation focus will be on methods involving isotopically labeled samples and the data generated from these experiments.

There are multiple techniques that can be used for the stable isotope based quantification of protein in a sample. These methods depend on the Carbon isotope $^{13}C_6$, or the Nitrogen isotope $^{15}N_7$ or the Oxygen isotope $^{18}O_8$ depending on which technique is used. These techniques are generally divided into 'metabolic', 'chemical' or 'enzymatic' and 'spiked-in'.

Metabolic techniques use the earliest point possible for isotope introduction in the form of media enriched with $^{15}N_7$ and $^{13}C_6$ labeled amino acids, an example of which is the SILAC [7] (stable isotope labeling by amino acid in cell culture) method. In these methods the errors in biochemical and MS processing could be ignored since labeled and unlabeled samples are typically combined while the cells are still whole. However, metabolic strategies are not feasible in many instances due to high cost and they are not useful when dealing with higher order organisms.

Chemical and enzymatic labeling methods are post biosynthesis techniques that incorporate isotopic labels *in vivo*. These methods range from incorporation of $^{18}O_8$ during tryptic digestion [8] of a sample to iTRAQ [9] (isotope tags for relative and absolute quantification) labeling of the N-terminus of the sample, TMT [10] (tandem mass tags) or ICAT [4] (isotope-coded affinity tags).

These methods can allow for up to eight different states to be examined in one experimental run and can be applied to whole or fragmented proteins. They require fine tuning of the enzymatic reactions for a given experiment (digestion efficiency or tag attachment) and of unexpected reactions between tags that can produce products that mask the data produced for quantification.

The use of an internal standard that is spiked in the sample is most commonly referred to as AQUA [3, 4] (absolute quantification of protein) but also includes MRM [11] (multiple reaction monitoring, also selected reaction monitoring, SRM). The AQUA method, a version of the SIS method, involves placing a known quantity of labeled peptide in a sample and comparing the peak intensities between the labeled and native peptides. The MRM involves using a triple quadrupole MS to keep track of both the whole peptide and one or more specific fragment ions. The problems of adding the correct amount of labeled peptide or protein to the sample to avoid drowning out the signal of interest and the necessity of needing to add labeled versions of all peptides of interest to the sample adds to the complexity of the work flow of this method. Best results are achieved if the sample preparation is kept to a minimum to prevent loss of sample before the labeled peptide is added. This work will focus on data generated using the SIS method for the addition of isotopically labeled peptides.

SIS is a method for quantifying a given protein by using a known amount of isotopically labeled protein (an isotopologue of the peptide) as an internal standard in a MALDI TOF measurement. The sum of the intensities of the first two or three peaks (M, M+1, or M, M+1, M+2) depending on the visibility of the peaks or the sum of the areas under the curve (AUC) of the peaks, calculated using Riemann sums or pixel counting, for a given peptide are divided by the same measure of the labeled protein [4, 13]. The comparison of the monoisotopic peak (M) of the labeled and unlabeled peptide is also

used [13]. A cutoff or baseline that is used to remove the effect of signal noise is often estimated and subtracted from the peak height or area. This ratio is then multiplied by the known amount of the labeled peptide to estimate the amount of unmodified peptide.

All SIS methods of quantification suffer from error in quantification that can range from 2% to 11% error in quantification [3]. This error has several possible sources. Pipetting of material during an experiment, which involves working with due to the miniscule volumes combined with the calibration of the pipetman play a significant role in generating these errors. A significant amount of the sample can be lost during its preparation before the addition of the labeled peptide. The amount of label peptide needed for accurate quantification can vary between experiments depending on the peak intensities found in the sample. Also, there needs to be a labeled peptide for each peptide of interest in a sample to ensure an accurate estimation of that peptide. That is, the mixes of labeled peptides need to be examined for the best combination of possible for a give experimental set up. The need for multiple peptides and the quantity needed to fine tune the mixtures for the actual measurements only begin to highlight the cost and complexity of SIS peptides.

This method is also vulnerable to errors in the spectra. The calculation of the peak intensities (heights) or AUC are affected by changes in the signal-to-noise ratio and the resolution of the individual peaks within the spectra. This and the problem of quantifying individual peptides of similar mass that form a set of overlapping peaks, quantification becomes a difficult task, even with SIS methods. All of these warrant finding an alternative computational method for the automated quantification of peptides from peak data.

1.2.2 Gaussian Mixture Distribution

A Gaussian mixture distribution is based on a probability density function that is

represented as a weighted sum of Gaussian (normal) component densities. The properties of the Gaussian mixture model distribution have been discussed at length and exhaustively studied [14]. A Gaussian mixture model is defined by a series of normal components each with its own mean and standard deviation. These are then weighed by a series of probabilities that sum to 1. The density of a Gaussian mixture model with *K* components

$$f(x \mid \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \sum_{k=1}^{K} \lambda_k \phi_k(x, \mu_k, \sigma_k^2), \tag{1.1}$$

where $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, ..., \lambda_K)$ a vector of the mixture proportions, $\boldsymbol{\mu} = (\mu_1, \mu_2, ..., \mu_K)$ the component normal means, and $\boldsymbol{\sigma} = (\sigma_1^2, \sigma_2^2, ..., \sigma_K^2)$ the component variances are the unknown parameters and ϕ represents the Gaussian density given by,

$$\phi_k(x) = \frac{1}{\sigma_k \sqrt{2\pi}} e^{\frac{-(x-\mu_k)^2}{2\sigma_k^2}} \tag{1.2}$$

The Gaussian mixture model approach for modeling isotopic peaks of a peptide in mass spectra data involves finding the best fitting probability density function of a Gaussian mixture model that describes a peptide in a mass spectrum in which the intensity of the spectrum at any given isotopic mass could be modeled by a normal curve. The data generated from the MS is assumed to represent a histogram from an underlying Gaussian mixture, where the height of the bar at any given MZ is the ionic current generated by the peptide when it collides with the detector at that time. The collisions that generate the ionic current at these time points are interpreted as mass ranges through use of a set of standards that are used to calibrate the mass spectrometer. In other words, the ionic current generated at a given time point is the frequency of an ion with which a given mass has collided with the detector. The mean of the first component normal of the Gaussian mixture is defined by the known mass of the peptide and denoted by $\mu_0 + \Delta$

with each subsequent component mean is known to be separated by a mass of one neutron. However, the means are shifted by an unknown error factor (Δ, mzError) arising from standardized curve calibration errors in the machine, but each mean is shifted by the same amount. This could be written algebraically as

$$\mu_k = \mu_0 + (\mathrm{k}-1)N^0 + \Delta. \tag{1.3}$$

The weight of each individual normal distribution, say λ_k for the k^{th} component, is defined as the k^{th} isotopic point distribution of the peptide being modeled and therefore are also known. The number of normal distributions used to make up the mixture (K) is determined by the number of points in the point distribution of the isotopic distribution of the peptide being modeled. The standard deviation of each component normal could be assumed to be the same for a given peptide due to the small changes in mass over the range of the ionic current cluster being fitted. There is an assumption that the isotopic variants of the peptide have the same physical and chemical properties as the monoisotopic peptide, varying only in mass. The standard deviation also serves as an inverse measure of the spectrum's resolution. The resolution of a spectrum is the ability to distinguish two peaks, or compounds, as being separate [15]. The number of points used to fit the model is defined by the number of time points (observations) in the data, a function of the data density of the spectrum. Although the spectrum may be observed over any mass range, typically a range of one Dalton prior to the monoisotopic mass to seven Daltons after (noted as: (-1, +7)) has been used in the past [12]. The area under the peaks (AUC/AUP) of the modeled isotopic cluster for a given peptide account for close to 100% of the peptide mass in this range. The model peaks integrate closely to 1. That is,

$$\int_{-1}^{7} f(x \mid \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\sigma}) dx \sim 1 \tag{1.4}$$

1.2.3 Peak Resolution, Sigma, and GMM estimation/correction/Stable Sigma justification

In terms of mass spectra analysis and MALDI-TOF MS in specifically the resolution of a spectra is key in discussing that spectra's quality. Resolution is the ability to tell two peaks as being distinct, or the minimal distance that is needed to say that two peaks are separate. The higher the resolution, or the less distance need to separate the peaks the better the quality of the spectrum produced. Resolution can be generally defined in the convolved or single (isolated) peak case. By treating individual peaks as examples of the component normal distribution we can estimate peak area, mass error (Δ) and peak width (σ) as previously stated but also resolution. This can be used to make statements of spectrum quality and the effect it has on the ability to use the Gaussian mixture model as an effective tool.

Figure 1.1: Single Peak Resolution
Full Width at α Mamximum

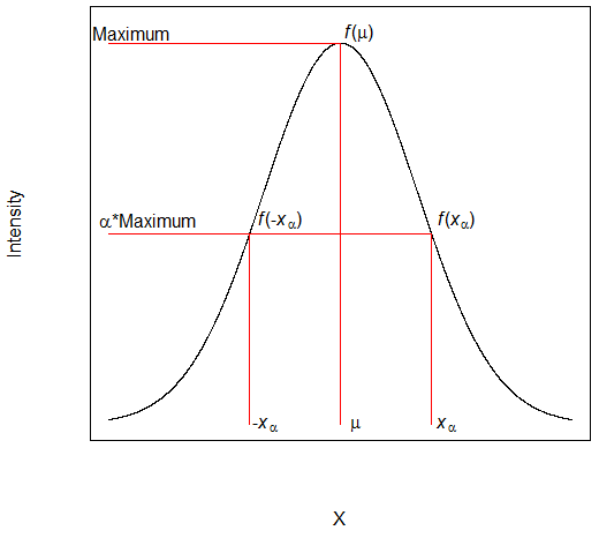
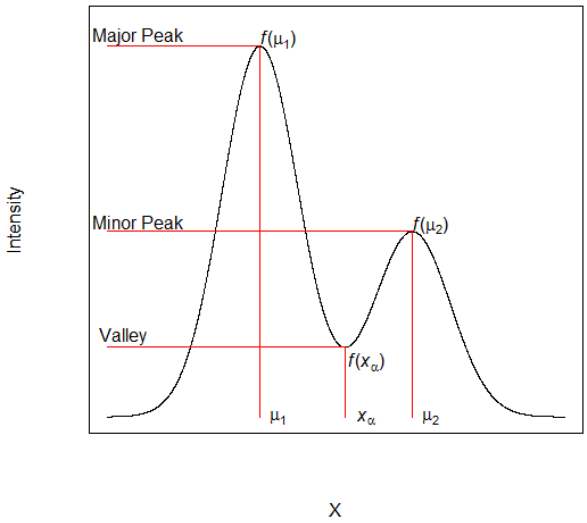


Figure 1.2: Convolved Peak Resolution
Valley Height at α Minimum



In the case of convolved peaks, the resolution is the minimum distance ($m_2 - m_1$, the means of the two peaks or the masses of the two compounds being analyzed in Da, of in the Gaussian mixture case for consecutive peaks $\mu_k - \mu_{k-1}$ in Figure 1.1) between peaks such that the valley between peaks is a minimum percent, α, of the intensity of the

larger of the two peaks.

In the single peak case resolution can be defined as $R = \frac{m}{\Delta m_\alpha}$ where m is the mass of the peptide or μ of the distribution (peptide mass plus the mass error, Δ) and Δm_α is the width of the peak at α, the maximum intensity. Typically $\alpha = .5$ or 50% of maximum peak height (intensity), and this is referred to as the Full Width-Half Maximum (FWHM, $\Delta m_{50\%}$) of the peak. This is illustrated in Figure 1.2. When estimating a peak with a normal curve, it becomes trivial to transform σ into FWHM or any other percentage of the peak intensity. This transformation of sigma can then be used to estimate a correction and gauge the impact of that correction on the area estimation of an isotopic cluster for a given peptide in MALDI-TOF spectrum since these spectra have a resolution that remains constant as mass increases. That is, the peak width increases gradually as m/z increases. This is due to the fact that these spectra are generated using time-of-flight of the analyte between matrix and sensor after the firing of the energizing laser. Larger (more massive) materials take a longer, measurable amount of time to travel that distance due to their mass using the same energization as lighter materials.

To connect peak quality, spectrum resolution and Gaussian mixture distributions for peptide quantification, the connection between peak width and resolution must be defined. Consider the normal density

$$\phi(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}. \tag{1.5}$$

When treating an isolated peak of a MALDI-TOF mass spectrum as an illustration of a normal pdf, the maximum height of that peak occurs at the estimated mean of the distribution and the proportional height at any point on the m/z axis as

$$f(x_{\max}) = f(\mu), \tag{1.6}$$

and

$$\alpha f(\mu) = f(x_\alpha). \tag{1.7}$$

The maximum of the peak is obtained by substituting $x = \mu$ in (1), which yields

$$f(\mu) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(\mu-\mu)^2}{2\sigma^2}} = \frac{1}{\sigma\sqrt{2\pi}}. \tag{1.8}$$

From equation (3) the maximum height at x_α is therefore

$$f(x_\alpha) = \alpha \frac{1}{\sigma\sqrt{2\pi}}. \tag{1.9}$$

This leads to the equation,

$$f(x_\alpha) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x_\alpha-\mu)^2}{2\sigma^2}} = \alpha \frac{1}{\sigma\sqrt{2\pi}}, \tag{1.10}$$

which simplifies to

$$e^{\frac{-(x_\alpha-\mu)^2}{2\sigma^2}} = \left(\frac{1}{\alpha}\right)^{-1}, \tag{1.11}$$

$$\frac{-(x_\alpha-\mu)^2}{2\sigma^2} = -\ln\frac{1}{\alpha}. \tag{1.12}$$

Solving for x_α as follows

$$(x_\alpha-\mu)^2 = 2\sigma^2 \ln\left(\frac{1}{\alpha}\right), \tag{1.13}$$

taking the square root

$$x_\alpha - \mu = \pm\sqrt{2\sigma^2 \ln\left(\frac{1}{\alpha}\right)}, \tag{1.14}$$

which yields

$$x_\alpha = \mu \pm \sqrt{2\sigma^2 \ln\left(\frac{1}{\alpha}\right)}. \tag{1.15}$$

Further simplification yields

$$x_\alpha = \mu \pm \sigma\sqrt{2\ln\left(\frac{1}{\alpha}\right)}. \tag{1.16}$$

Repeating this for the other side of μ and applying symmetry, the full-width to the peak μ of its α^{th} height to find the range around μ based on σ, denoted by Δm is

$$x^+ - x^- = \left[\mu + \sigma\sqrt{2\ln\left(\frac{1}{\alpha}\right)}\right] - \left[\mu - \sigma\sqrt{2\ln\left(\frac{1}{\alpha}\right)}\right], \tag{1.17}$$

cancelling μ gives,

$$x^+ - x^- = \sigma\sqrt{2\ln\left(\frac{1}{\alpha}\right)} + \sigma\sqrt{2\ln\left(\frac{1}{\alpha}\right)} \tag{1.18}$$

$$\Delta m_\alpha = \sigma\sqrt{2\ln\left(\frac{1}{\alpha}\right)} \tag{1.19}$$

Therefore, setting $\alpha = 0.5$ the FWHM is

$$\text{FWHM} = 2\sigma\sqrt{2\ln\left(\frac{1}{0.5}\right)} = 2\sigma\sqrt{2\ln 2} \approx 2.355\sigma \tag{1.20}$$

This gives us an easy calculation for $\Delta m_{0.5}$ which in combination with the estimate for the mean of our curve (peptide mass plus Δ, estimating m) gives us the estimated resolution for our isolated peak. Combining this with the equal peak resolution we see in TOF data we can then estimate the change in σ between peaks. Here our peaks are separated by the mass of a neutron since they are individual isotopes of the same

peptide. This limits the m/z range we need to consider since only the range for a single peptide or cluster of convolved peptides needs to be considered. This range changes based on the mass (atomic composition) of the peptide since more complex molecules will produce more detectable isotopic peaks over a larger m/z range.

Since we know resolution is equal across all peaks of a MALDI-TOF, the following for the i^{th} and j^{th} peaks is true:

$$\frac{\mu_i}{\Delta m_{i\alpha}} = \frac{\mu_j}{\Delta m_{j\alpha}}, \tag{1.21}$$

Substituting equation (15), equation 17 becomes (and assuming σ_i, σ_j for the two peaks and calculating $\Delta m_{k\alpha}$ for given σ_k and α using equation (15)) yields,

$$\frac{\mu_i}{2\sigma_i\sqrt{2\ln\frac{1}{\alpha}}} = \frac{\mu_j}{2\sigma_j\sqrt{2\ln\frac{1}{\alpha}}}. \tag{1.22}$$

Which is essentially,

$$\frac{\mu_i}{\sigma_i} = \frac{\mu_j}{\sigma_j}. \tag{1.23}$$

Therefore,

$$\sigma_j = \frac{\mu_j}{\mu_i}\sigma_i \tag{1.24}$$

For the Gaussian mixture model μ_k is the mean of the i^{th} peak. That is,

$$\mu_k = \mu_0 + (\mathrm{k}-1)N^0 + \Delta. \tag{1.25}$$

Substituting 21 in 20 for any two peaks i and j ($j > i$) in an isotopic cluster can be obtained by,

$$\sigma_j = \frac{\mu_0 + (j-1)N^0 + \Delta}{\mu_0 + (i-1)N^0 + \Delta}\sigma_i. \quad (1.26)$$

The change in σ between peaks of the same peptide can be simplified to

$$\sigma_j = (1 + \frac{(j-i)N^0}{\mu_i})\sigma_i, \quad (1.27)$$

We see that the only change in mass is that of the number of neutrons in the isotopic peak, $(j-i)$. For the peaks seen in Ang II spectra and for *i*=1 and *j*=2 this means

$$\sigma_2 = \left(1 + \frac{N^0}{\mu_1}\right)\sigma_1 \quad (1.28)$$

where m_1 is of the order 1046.54179 and N^0 is 1.00866. That is $\sigma_2 \sim (1.00096)\sigma_1$. Meaning σ increases by $(1.00096)\sigma$ m/z for every neutron of mass difference in an isotopic cluster of a peptide. Over the range of an isotopic cluster the minute changes in σ is negligible because the precision of the data is too low. For using Gaussian mixed model method for SIS peptide quantification this small change in σ does not significantly affect area estimation or model fitting. It should also be noted that this solution is only applicable to TOF and magnetic analyzers that have a constant resolution across spectra.

A final note, further generalized between any two peaks in a spectra with changes in μ,

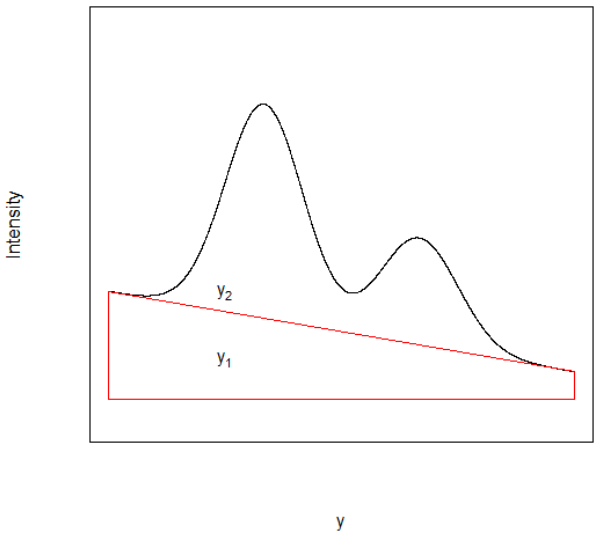
$$\sigma_{xj} = \frac{\mu_x + (j-1)N^0 + \Delta_x}{\mu_y + (i-1)N^0 + \Delta_y}\sigma_{yi} \quad (1.29)$$

For estimations of area for a single peptide as discussed previously, this means that this is a gradual increase in σ as the mass of the peak increases.

1.2.3.1 Baseline Shift in MALDI-TOF MS

Often a vertical shift is observed in MS and needs to be accounted for in SIS estimations [22-23]. This shift, hence forth called baseline shift, can be noise in the MS signal. It is believed that this noise could act in either an additive or masking (overlapping) fashion. Either boosting peaks higher or covering up their true intensity. Describing baseline shift is important in peptide quantification since all correct methods including Gaussian mixture modeling since they all depend on relative ratios of signal quantification, weather peak area or peak intensity. The shift of the signal needs to be accounted for in each of the peptide estimations so that the ratios being calculated are as accurate as possible. There is no universally accepted method in the literature for estimating or correcting for this baseline shift. However, generally this error seems to be treated as additive and is treated this way when using Gaussian mixture models. The baseline shift is described here as a slope-intercept for a straight line equation over the m/z range examined for each peptide cluster. This could be mathematically modeled using a separate mixture model of a trapezoidal and Gaussian mixture model. That is,

Figure 1.3: Trapezoidal and GMM mixed distribution model of spectrum



$$(1-\gamma) f_{y_1}(\mathrm{y}) + \gamma f_{y_2}(\mathrm{y}), 0 \le \gamma \le 1 \tag{1.30}$$

$$f_{y_1}(\mathrm{y}) = \sum_{k=1}^{K} \lambda_k \phi_k(x, \mu_k, \sigma_k^2) \tag{1.31}$$

$$f_{y_2}(\mathrm{y}) = \alpha + \beta x \tag{1.32}$$

Where y is the mz range of the MS which we restrict to a finite range based on user input

(here -1, +7 Da from $\mu_0 + \Delta$) and γ is a second mixture coefficient. When γ=0, there is no noise in the spectra. Figure 1.3 illustrates an exaggerated example of this mixture of distributions. A trapezoidal distribution is not the only way to describe the baseline shift. Since the slope and intercept are calculated from a column of ones and the m/z range, a third column of the m/z range squared could describe a quadratic function for the baseline shift. Any polynomial function could be described in this way to approximate the baseline.

1.2.4 Overview of the Gaussian Mixture Algorithm

1.2.4.1 Spectra Fitting and Area Estimation

The Gaussian mixture model is constructed by first calculating the isotopic distribution and mass of the peptide in question. This is done from the amino acid formula of the peptide and is based on known amino acid composition and estimates of universal isotope distributions. The point distributions of labeled and unlabeled versions of a given peptide are not considered to be significantly different. An idealized model of the peak cluster is then constructed using a Gaussian mixture where the isotopic point distribution is the mixture coefficient, λ, the peptide mass serves as the mean of the first peak and the peaks of the mixture are separated by one neutron mass along the mz axis in the spectra as shown in equation 1.3 (with Δ=0). At this juncture the area of the Gaussian mixture sums to one. However as described in section 1.1.3.1, data from the MALDI-TOF MS include a baseline shift that must be accounted for in our model fit. There for the unknown quanties that need to be estimated in order to estimate the area under the peaks are σ, Δ (as in equation 1.1), α and β (as in equation 1.32). This is achieved in this algorithm in two major steps where we first construct a model based on the σ and Δ parameters and then look at the estimates of α, β, and peak area for the best fitting model.

The model (or models if there are multiple peptides in the cluster) is then

described, by column, into a rectangular matrix with descriptors of the baseline equation as extra columns. This is then used to solve a linear regression using QR decomposition (*qr.solve* in R) [17, 18] where the spectra data act as the right hand side of the equation, this is the dividing line between the two major steps of the algorithm. This produces a vector describing an estimation of the sum of peaks area for each peptide and the baseline estimation (here slope and intercept). These estimates are used to reconstruct the model and compare it to the data to estimate a best fit for the model against the data. This produces an estimate of the area under the isotopic cluster in the data (the slope of the linear regression) and estimate a shared baseline error or noise adjustment (the y-intercept of the linear regression). This estimation of noise can also be calculated by subtracting the best fitting model from the data and taking the mean of what remains.

This model generation and fitting is done over a search space consisting of range of standard deviation of the normal (σ parameter or resolution) and a range of mass correction (mzError, Δ parameter based on the error from using a standard curve to calibrate mass) in the spectra data to estimate the means. Sigma changes the width of the peaks in the isotopic cluster while mzError shifts the model along the MZ axis of the

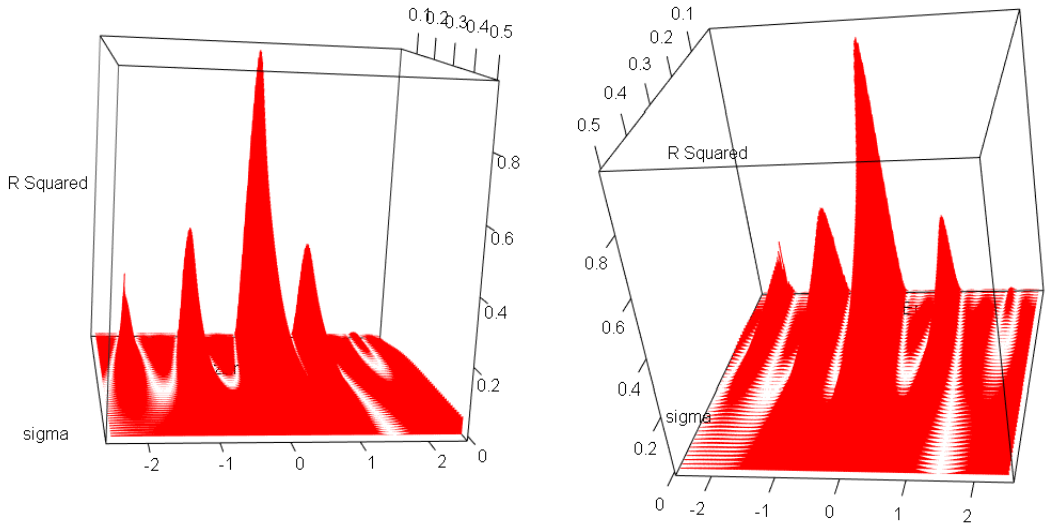


Figure 1.4: Examination of Δ&σ search space

spectra. Goodness of fit is measured by finding the model with the highest coefficient of determination (R^2) showing which model has the least unexplained variance between the expected model and the observed data. This is a brute force technique that finds the local maximum R^2 value for the provided search space made up of the combination of standard deviation (σ) and mass corrections (Δ). In chapters 3&4 improvements in parameter space crossing using quadrant ascent and Expectation-Maximization algorithms are introduced respectively.

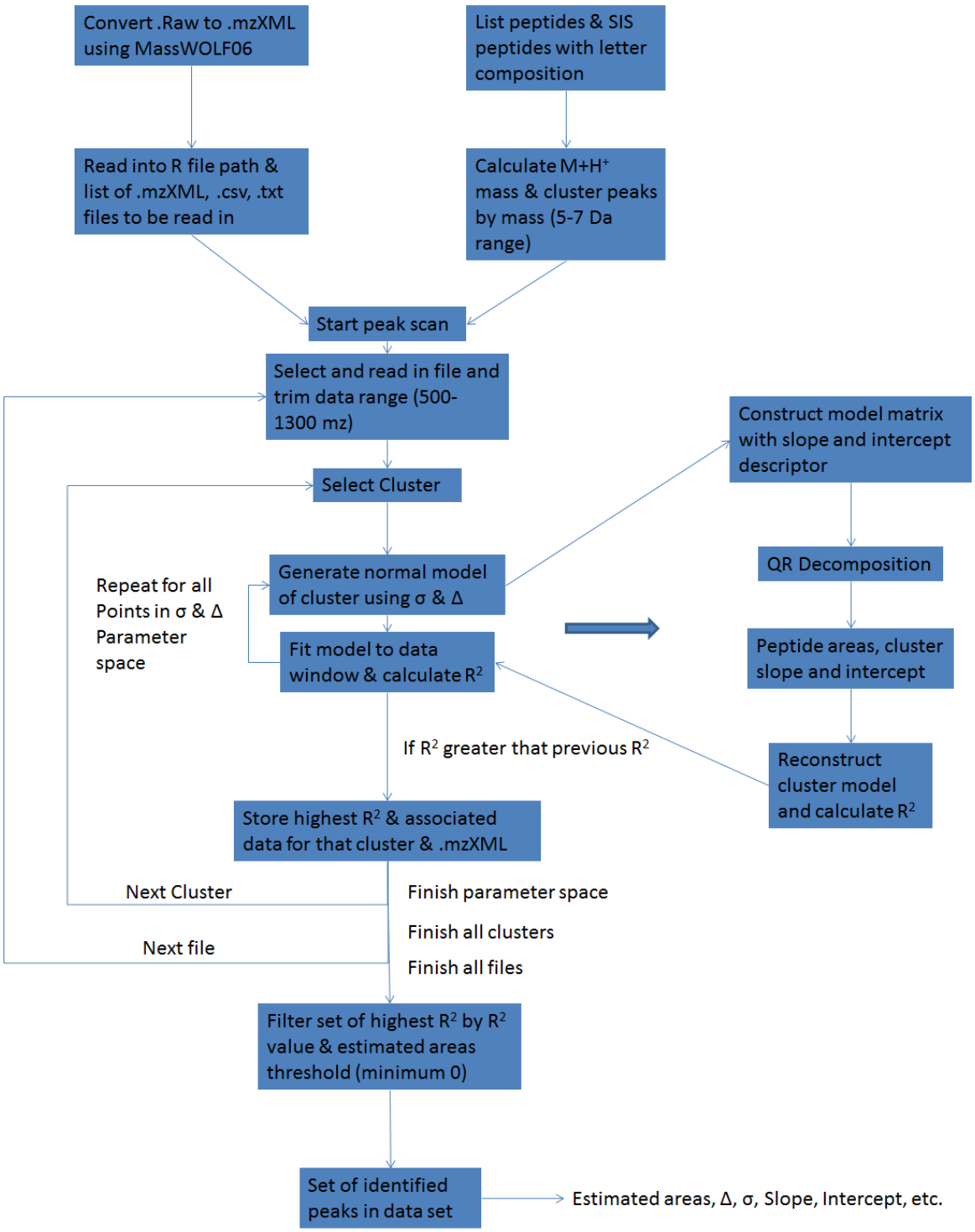
Since this process will produce an answer for each peptide being searched for, a post search filtering step needs to be performed to remove false positives. Removing those estimations of a negative or small area and those fits with low R^2 value removes most false positive fits. These filtering parameters are set by the user and are affected by spectrum background noise and resolution. Figure 1.4 shows an examination of a sample search space of standard deviation and mass adjustment using the coefficient of determination as a measure of fit. This shows that only a relatively small parameter range need to be considered to find the best fitting, and therefore most accurate estimate of peak area, due to constraints of peak structure. The distribution of R^2 is quadratic around the maximum. Each local fit maximum is separated by a single neuron mass. If the mz error estimation is greater than one half of an mz in the data, this is indicative of a need for maintenance of the spectrograph.

A full workflow for the Gaussian mixture model algorithm as described is shown in Figure 1.5.

1.2.4.2 Spectrum Simulation

The simulation of peptide peaks was accomplished by using an algorithm based on the one used for peak area estimation. The point distribution for our simulated peptide is calculated in the same manner and is used to define the number of normal distributions

Figure 1.5: MALDI-TOF MS data processing Flowchart for GMM



drawn from to simulate individual peaks of a peptide. A total number of samples (picks) is set and divided up between each normal distribution based on the point distribution of the simulated peptide. This is done using rmultinorm in R, where a multinomial distribution based on the isotopic point distribution is sampled by the total sample number of times. This gives the total number of samples per peak or normal distribution that

represents a peak.

The normal distributions are described by the mass plus an error in mass estimation with an additional neutron mass for every peak after the first, until there is a normal distribution for each element of the point distribution. These samples from these normal distributions represent activations of the detector at that given mz. Once all of the samples are collected they are binned to form a histogram that simulates data from a mass spectrum. The bin size mimics the range between mz readings form a mass spectra. Some sampling from the normal is lost by setting the mz range that is being simulated, samples that are outside this range are discounted.

Once the histogram is constructed it is normalized to have a total area of one by dividing by the total number of samples taken. This has the effect of making the sum of the area under the peaks close to one. This vector is then multiplied by the peak area desired for the simulation. This modified set or peaks (or modified histogram) is then treated with a two-step noise addition process. First AR(1) noise is added then a baseline shift is performed. AR(1) noise is generated over the mz range desired and is added by either replacing elements in the vector that are zero or less than the generated noise at that point in the mz range. The baseline shift is added to the peaks by addition to the vector, either a flat lift or an application of a regression line over the mz range is used.

1.2.5 Dealing with Adducts

There are multiple different adducts that can be found in mass spectra analysis of various compounds. In peptide analysis, common adducts can include Na, Cl, K, Br, and other elements. These adducts can be dealt with by treating them as extra atoms in the peptide and calculating mass and isotopic distribution accordingly. In this algorithm, to construct the correct peptide molecular formula, each amino acid of the peptide is treated as being dehydrated and then a water molecule is added to complete the chemical formula

of the peptide. The constituent atoms of each amino acid are summed for the peptides chemical formula and isotopic distributions for each element are then convolved using a Fast Discrete Fourier Transform of a vector of the atomic isotope distribution raised to the power of the number of that atom in the peptide. The individual vectors describing the distribution of each set of atoms are then multiplied together and a second transform is preformed to get the isotopic distribution of the peptide. Adducts are dealt with in the exact same way, they are treated as extra groups of atoms and are calculated accordingly. New adducts can be added easily as long as isotopic distributions are known.

1.2.6 Identifying Questionable peaks

Since the Gaussian mixture model method depends of knowing what compound to look for, mystery compounds from the sample can interfere with quantification estimations. Compounds with different ionizations, such as M+2H or M+3H compounds can have multiple peaks per Dalton and the can be between peaks of a known compound or contribute to the intensity of one or more peaks of a known compound. Estimations of Peak AUC using GMM could be done using an identified unaltered (ex. M+2H if M and M+1 are convolved with an unknown) peak but this estimation would be error prone. These 'contaminated' peaks can be detected through a filtering process where A low R2 but high area (high intensity) set of peaks are detected.

The low R^2 would reflect both contaminant peaks appearing between peaks of interest and the contaminant peaks adding to peaks of interest throwing of estimation based on the isotopic distribution of the compound being searched for. Since it is believed that the R^2 statistic falls in a bathtub distribution (Beta, ~a=b=.5) when used in the fitting of Gaussian mixture models to spectra data, a cut off there is easy to estimate. Combining a R^2 cutoff and minimum peak area or peak intensity can be used to filter for possible contaminated peaks. These peaks can be flagged by the software as needing

review by the used to detect contaminants and still be filtered for cases where the algorithm is fitting noise.

It is important to note that the difference between local distributions and the universal distributions used in the calculation of estimated peak ratios may lead to a small amount of error. Local elemental isotope distributions may be different from the universal distributions used to construct λ. This can play a role when trying to fit models to spectra data but it should be noted that these differences in distribution are likely equivalent to the differences between a SIS peptide and its native twin. These are small, calculable and ultimately unable to be seen in real data.

1.3 Bench Analysis and Data Generation

In the process of generating data to test GMM several skills needed to be learned and obstacles overcome. The original purpose of GMM was to be able to quantify peptides with overlapping ionic currents in MALDI-TOF MS. There data were originally generated to facilitate the analysis of the angiotensin extracellular preprocessing pathway in mouse podocytes as a part of the exploration of the Renin Angiotensin System and its role in the control of blood pressure. This research focused on the production of angiotensin peptides from a network of enzymes with the focus being placed on finding peptides being produced by the network [12] that have not been reported on extensively and to also understand how the network reacts to modification from enzyme inhibitors. Three different experiments were done while verifying GMM. First, cell culture exposures to various angiotensin fragments under various inhibitors were performed to gauge the production of fragment peptides. Second, isolated mixtures of recombinant enzymes and angiotensin fragments were preformed to gauge enzyme activity and peptide production. Finally, MALDI-TOF samples were generated using know amounts of peptide to test GMM predictions against known data. These experiments are not gone

into great detail since they are not the main argument of this document but are reported to provide a more robust account of the work done in validating GMM as a method of quantification.

While involved in the bench side portion of this work. I designed experiments focusing on the degradation of Angiotensin-1-9 and Angiotensin-1-7. The skill set required for these experiments started with the culture of immortalized mouse podocytes, including the sterile technique and care time tables required to grow the cells for experiments. This included preparing the cell media and triggering the appropriate cell metamorphosis for mature cells to be available for the experiments. Once the cells were ready for experimentation, samples of cell media were collected after initial exposure to an angiotensin peptide. Once the samples were collected, preparation for MALDI-TOF analysis was done in triplicate with three spots on the MALDI plate for each time point. Each cell culture experiment was done with one control well, well for the individual inhibitors, and a well for any combinations of inhibitors being examined.

The recombinant enzyme activity and product assays were done by time series sampling of mixtures of single enzymes and a single angiotensin peptide. Recombinant human Aminopeptidase A and Neprilysin were individually exposed to Ang-1-7 and Ang-2-10 respectively, with and without inhibitors (Amastatin and SCH39370 respectively). These were done to check the activity of the recombinant human enzyme against the known activity of the mouse podocyte analog and the effectiveness of the inhibitors used. These experiments also provided information on how the enzyme interacted with the peptide products produced from previous enzyme-peptide interactions along with the dampening effects of inhibitors.

The known mixture preparations were a series of angiotensin-2-10 and angiotensin-1-9 mixtures at different ratios with SIS-angiotensin-1-9. The ratios of

peptide ranged from 1:1 to 10:1 and 1:10. The amounts of peptide had to be changed to keep them with in detectable limits and the SIS peptide was kept at a constant amount throughout all ratios. These data were created to gauge the ability of GMM to detect and quantify the second peak of a convolved mixture.

These bench experiments had a body of work involved in trouble shooting. The proper technique for MALDI-TOF sample preparation needed to be explored. A sandwich technique where a layer of matrix was spotted followed by the sample and a second layer of matrix was determined to deliver the best sample ionization with angiotensin peptides. Peptides concentrations required continual adjustment to maintain final concentrations within detection limits and required adjustment pre-experiment to maintain experimental cohesion. Alternate buffer solutions needed to be used since MALDI analysis formed natural polymers from certain buffer components. Finally, due to normal use a alternative MALDI-TOF Mass Spectrometer was required to finish the analysis of these bench experiments and alternative data management methods were required to incorporate this data.

This hands on knowledge of the bench experiments involved in data generation and the difficulties that can occur, allowed for a better understanding were errors in the experimental process can occur and possible causes of variations in the data produced. It provided insight into how data can influence further experiment progression and problem solving it the experimental framework.

1.4.1 The Overall Picture

To place this problem in its proper context, it is important to describe and attempt to summarize the fields that are involved in the asking of the question as well as in its answering. This is partly due to the wide range of subjects involved as well as their contributions.

Proteomics is the large-scale experimental analysis of proteins. The focus of proteomics is to understand the role and function of proteins in biological processes through examination of protein structure (sequence), modification, function, or quantity. These examinations usually entail the analysis of samples using immune precipitation, purification, gel electrophoresis or mass spectrometry. The role played by mass spectrometry includes those of protein identification and quantification. Matrix-assisted laser desorption/ionization (MALDI) time-of-flight (TOF) mass spectrometry (MS) is one version of MS used that has the unique ability to analyze biological molecules without breaking them up during analysis. MALDI-TOF is used for identification and when possible quantification of these molecules. Quantification can be done by using an internal standard in the sample preparation. This is normally done when the material in question is already identified. In the example of peptides in biological samples, a stable isotope standard (SIS) is used. This is a version of the peptide to be quantified that has been created using heavy labeled elements. It is chemically identical to the normal (native) peptide but is more massive due to the extra neutrons added by the heavy elements.

Systems biology is the study of biological systems or networks and the web of interactions between the components of these systems and the systems themselves in living organisms. Proteomics has been used to shed light on these interactions by looking as large bodies of data to make inferences about biological networks. One attempt to do this is the construction of a mathematical model of a network, for example an enzymatic pathway, based on the known substrate kinetics of the enzymes in that pathway. One methodology for this is biochemical systems theory, where pathway being modeled is represented by ordinary differential equations where the enzyme-substrate reactions (biochemical processes) are individual equations. This means that an entire series of

reactions that are interconnected at different levels, where multiple enzymes share multiple substrates, can be modeled with a degree of accuracy. We can then begin to think of a biochemical pathway as a network or processing of material.

Here the example of a biochemical pathway is the extra-cellular preprocessing of Angiotensin that occurs as a part of the Renin-Angiotensin System (RAS) before signaling occurs by receptor binding at the cell surface. This network of substrates and enzymes is an example of a pathway that is ubiquitous in human biology, complex, and not fully understood. It has competition between enzymes and substrates, causing cross-talk with in the pathway, making it flexible, allowing it the versatility to be used in different instances in human physiology but it has one starting point at the cleavage of Angiotensinogen from Renin and ends after Angiotensin is broken down. This limit on network size makes it an ideal candidate as well.

Once a model is constructed from known data it can then be perturbed *in silico* (in machine) with either substrate material at different points or by inhibiting different enzymes along the pathway. These mathematical predictions of the behavior of the pathway can then be tested in a laboratory setting (bench side). This is a cyclical process with the data generated from bench-side with wet experiments that can be used to improve the model. These improvements take the form of adding or removing nodes from the network (enzyme-substrate reactions) or changing the reaction rates at these nodes. The improved model can then be used to inform the next set of biological experiments. This process can be used to identify new drug targets in a pathway or identify new combinations of drugs tailors to a specific biochemical outcome, where in this case these drugs are inhibitor or activators of enzyme activity. In the example of adjusting RAS preprocessing, the aim could be to pool peptide production towards peptides that are known to lower blood pressure (BP) instead of just blocking the

production of peptides that increase BP as is currently done with current prescriptions. This combination of drugs approach is more interesting since there is already a large library of drugs that already exist that can be used without the need for going through the drug discovery process.

This process of model construction, bench experimentation and then model refinement can be used to gain a greater understanding of the pathway being examined and to start asking larger, more interesting questions. In RAS, for example, is it the blocking of BP increasing peptides or the resulting increase in BP lowering peptides as material processing in the pathway is changed, or the combination of the two, that causes the decrease in BP from the use of a given drug.

This process of mathematically defining a biochemical pathway is dependent on several factors including the ability to measure the amount of material in a sample from an experiment. This can be done using the previously mentioned proteomics methodologies. This is important since enzyme kinetics and inhibitor effectiveness can be gauged from a time series experiment where substrate quantification tells how much of a given material is being used or produced at different points in a given biochemical pathway. This use and production can be thought of as the flux of the material at a given point in the pathway. Drugs (inhibitors or activators) act to change this material flow, increasing or decreasing the net amount of material at different points in the pathway.

Thus the problem of quantification becomes important not only due to the need to understand the flow or flux of material through a biochemical network or system but also becomes a computing problem since a large number of data points (multiple experiments, each with multiple time points) and associated secondary experiments (individual reactions, not the entire network) are needed to illustrate the properties of the pathway in question. The implementation of a computational method for quantification prevents the

large amount of data from becoming a bottle neck where the data itself becomes too large to analyze. It becomes necessary to have versatile methods for the quantification of material from proteomics experiments. The original problem of quantifying convolved peptides in MALDI-TOF MS becomes a small but necessary step in the overall process of biochemical systems analysis using proteomic methods.

In this workvfocus is on a statistical approach to quantifying peptides in MALDI-TOF MS data that is capable of analyzing both single and multiple (convolved, overlapping) peptides in spectra and give estimates of the peptides using internal standards. The aim is to provide a method that does not require any previous information except the peptide to be quantified and the spectra to be analyzed while producing the parameters of a Gaussian mixture model that describes the peaks seen in the data. This description giving a quantification of the peaks as individual peptides, using areas under the peaks, that can be used with SIS to quantify these peptides. This description included an estimate of the area under the peaks for a given peptide, which is used for its quantification, and the uncertainty of the model parameters estimated from the data. This method can be used to replace past methods of SIS quantification since it can process both single peptide and peptides that overlap one another in the MS.

1.5 Significance

The Gaussian mixture model allows the estimation of peak areas for individual in a group of peptides when these peptides have peak clusters that overlap, masking part of the lower mass peptide with the signal from a larger mass peptide. Overlapping clusters of peptides cannot be quantified by previously established methods of AQUA interpretation because those methods rely on isolated, unmasked peptides to be clearly seen in the spectrum. This method can be easily implemented in a computational environment and used to scan large numbers of spectra accurately with minimal input

from the user.

Besides those topics mentioned above there are topics that will be covered multiple times in the following chapters that will be used to remind the reader of specific points of importance. This is in part due to the fact that the following chapters are written for publication to the wider community. This work has three parts. First, an analysis of peak intensity and Riemann sum AUC methods of peak quantification with SIS peptides which is not found in current literature and compare these to the Gaussian mixture model method. Second, a method for simulations is introduced and simulations are preformed to stress test the Gaussian method. Finally an EM algorithm is introduced for a maximum likelihood estimator for the parameters of the Gaussian mixture model to replace the grid search method if estimation. Among these papers shared topics will include the importance of baseline estimation and its roll in proper estimation of peptides in a sample.

This method is usable on more than just peptides, any compound that has a known chemical formula can be modeled in this manner and if the spectra do not have the same normal shape, this method can be done other distributions id they more closely resemble the spectra peak shape.

The Gaussian mixture method will be published as a software package in R using Bioconductor hosting.

2. **Manuscript one:** The Application of Gaussian Mixture Models for Signal Quantification in MALDI-ToF Mass Spectrometry of Peptides

2.1 Abstract

Matrix assisted laser desorption/ionization time-of-flight (MALDI-TOF) coupled with stable isotope standards (SIS) has been used to quantify native peptides. This peptide quantification by MALDI-TOF approach has difficulties quantifying samples containing peptides with ion currents in overlapping spectra. In these overlapping spectra the currents sum together, which modify the peak heights and make normal SIS estimation problematic. An approach using Gaussian mixtures based on known physical constants to model the isotopic cluster of a known compound is proposed here. The characteristics of this approach are examined for single and overlapping compounds. The approach is compared to two commonly used SIS quantification methods for single compound, namely Peak Intensity method and Riemann sum area under the curve (AUC) method. For studying the characteristics of the Gaussian mixture method, Angiotensin II, Angiotensin-2-10, and Angiotenisn-1-9 and their associated SIS peptides were used. The findings suggest, Gaussian mixture method has similar characteristics as the two methods compared for estimating the quantity of isolated isotopic clusters for single compounds. All three methods were tested using MALDI-TOF mass spectra collected for peptides of the renin-angiotensin system. The Gaussian mixture method accurately estimated the native to labeled ratio of several isolated angiotensin peptides (5.2% error in ratio estimation) with similar estimation errors to those calculated using peak intensity and Riemann sum AUC methods (5.9% and 7.7%, respectively). For overlapping angiotensin peptides, (where the other two methods are not applicable) the estimation error of the Gaussian mixture was 6.8%, which is within the acceptable range. In summary, for single compounds the Gaussian mixture method is equivalent or marginally superior compared

to the existing methods of peptide quantification and is capable of quantifying overlapping (convolved) peptides within the acceptable margin of error.

2.2 Introduction

MALDI-TOF is a convenient tool for determining peptide abundance in high-throughput workflows. MALDI-TOF MS is a t a solid-state ionization technique in which the sample is mixed with a chemical (matrix) that is excited by an ultraviolet or infrared laser. The laser excites the matrix leading to the transfer a proton to the analytes in the sample. The time of flight required for a given analyte to be detected is proportional to the mass of the analyte. The ions produced by this technique are primarily singly charged. Principles underlying each ionization technique have been well described elsewhere [1] Visualizing specific peptides or other analytes by exact mass allows for a greater degree of specificity in quantification and identification. In past work, MALDI-TOF has been used to measure angiotensin (Ang) peptides in cell culture or tissue samples and profile these peptides within their network [2-10]. Ang peptides belong to the renin angiotensin system (RAS), a hormonal system of major significance in human biology. The main effector of the system is Ang II [Ang-(1-8)], an octapeptide that is formed through sequential cleavage of the substrate angiotensinogen[11]. Among many other pathophysiological roles, Ang II is known to stimulate blood vessels to raise arterial blood pressure [12], activate mechanisms of sodium retention in the kidney [13] and induce proliferation in cardiac myocytes [14]. Interestingly, Ang peptides that are generated through alternative pathways of enzymatic processing, such as the heptapeptide Ang-(1-7), may elicit biological effects that are counteracting to those of Ang II [15]. Therefore, accurate visualization and quantification of Ang peptides is of utmost importance to adequately study the RAS. Proteins from tissue sections have been also

analyzed, allowing for the localization of biological molecules to distinct regions of tissue [16-17], demonstrating the diversity and flexibility of MALDI-TOF analysis.

Mass spectrometry (MS) based techniques have advanced the field far beyond antibody-based methods with the capacity of identifying and quantifying multiple [18-20] peptides and posttranslational modifications in a single experiment [21]. The ability to quantify peptides is hindered by their physical and chemical properties. Differences in charge, hydrophobicity, or posttranslational modification are some of the properties that effect the ion formation and time of flight of a sample peptide or peptide mixture. Similar peptides can have widely varying differences in ionization within a sample, leading to differential matrix suppression or 'flyability' [2] between peptides. Flyability refers to the differences in ionization and post-source decay between similar peptides. Similar peptides may be more or less prone to ionize and therefore will generate a higher or lower signal, respectively. . In quantification, flyability can be obtained through a constant based on known differences between peaks of different peptides [2].

Traditional methods of peptide quantification utilize the specific binding properties of antibodies to estimate abundance. Enzyme-linked immunosorbent assay (ELISA) and radioimmunoassay (RIA) are popular methods that indirectly measure the amount of bound antibody to the native peptide by a colorimetric reaction or radioactive decay [22-23]. One of the drawbacks of antibody-based methods is the potential for cross-reactivity with non-target peptides. Peptide quantification by mass spectrometry is direct, thereby avoiding issues associated with antibody cross-reactivity, and include those based on stable isotope dilution theory, although label free methods have been described [24-35]. Stable isotope dilution theory is based on the concept that a stable isotope labeled protein or peptide behaves exactly the same during MS analysis. Because the mass difference between the labeled and unlabeled samples can be detected through

direct comparisons of signal intensities (*e.g.* time-of-flight) or area under extracted ion chromatograms (*e.g.* LC-MS/MS) are used for quantification. For the case of LC-MS/MS based quantification the sample is first separated prior to being introduced into the mass spectrometer thereby reducing the complexity and competition for ionization [31-34]. Due to lower complexity, the likelihood of overlapping peptide masses is also reduced and extraction of peptide specific fragment ion intensity over time can further increase the specificity of the measurement.

For monitoring biological reaction product formation, such as peptide metabolism, MALDI-TOF MS is ideally suited and takes advantage of the internal standardization commonly referred to as AQUA [35-38] (Absolute QUAntification of protein). Although MALDI-TOF does have limitations in its reproducibility due to the effects of uneven matrix-analyte mixture and matrix interactions, the use of SIS quantification allows for the circumvention of some of these difficulties. All SIS methods involve placing a known quantity of isotopically labeled peptide in a sample and comparing the peak intensities between the labeled and native peptide. The synthetic SIS peptide is identical to the native peptide with the exception that one amino acid is comprised of stable isotopes of carbon (^{13}C) and nitrogen (^{15}N). In practice both peptides are chemically identical with respect to ionization and decomposition, but the stable isotope labeled peptide is heavier and is detected as a different m/z window in the mass spectrometer thus allowing simultaneous comparison with the native. One or more amino acids can be labelled imparting further flexibility in the monitoring of peptide metabolism. The sum of the intensities of the first two or three peaks (M, M+1, or M, M+1, M+2) depending on the visibility of the peaks or the sum of the areas under the curve (AUC) of the peaks, calculated using Riemann sums or pixel counting, for a given peptide are divided by the same measure of the labeled peptide [36]. The peak intensity is defined as the maximum

height of the peak. The Riemann sum AUC is the trapezoidal sum of the area under each peak. In both approaches, a cutoff or baseline is used to remove the effect of signal noise and is subtracted from the peak height or AUC. This ratio of native to labeled peptide is then multiplied by the known amount of the labeled peptide and sometimes corrected for response based on an external standard curve to estimate the amount of unmodified peptide.

This method of quantification is not without its difficulties; error in quantification can range from 2% to 12% [35]. This error has several possible sources from both methodology used for quantification and from the analysis itself. A significant amount of the sample can be lost during preparation due to manipulation before the addition of the labeled peptide. The amount of SIS peptide needed for accurate quantification can vary between experiments depending on the peak intensities found in the sample. The ratio between native and SIS peptide need to be less than 10 to aid in accurate estimation [38]. There is also the fact that there needs to be a SIS peptide for each peptide of interest in a sample to insure an accurate estimation of that peptide. This means, mixtures of SIS peptides should be balanced with endogenous levels for a given experiment. The need for multiple peptides and the quantity needed to fine tune the mixtures and preform the actual measurements begin to highlight the costs of peptide quantification by SIS peptides. In SIS quantification, the peptide(s) being quantified are known beforehand. This is necessary to produce the SIS version of the peptide, bypassing any problems that may occur by matrix suppression (or difference in ionization). The Gaussian mixture method incorporates the chemical properties of the known peptides by parametrizing the probability density function. The approach also provides discrete peak separation and provides the characteristics through the estimates of the unknown parameters of the isotopic distribution of the peptide being examined. An efficient algorithm for estimating

the baseline is also incorporated into this approach. Unlike the existing Gaussian mixture approaches [38-43] by incorporating the known chemical information in the parameterization our approach reduces the dimensionality of the unknown parameter space. In addition to providing a more accurate quantification, the approach considerably speeds up the computations. In the convolved peptide situation peak intensity and Riemann sum AUC cannot be used to accurately quantify the separate peptides. Since this method can be automated over a large number of spectra and peptides, bottlenecks in the data processing pipeline are avoided.

Past methods are vulnerable to errors in data processing. The estimation of a baseline and cutoff regions for measurement are often end-user dependent or automated by proprietary software, both of which are often accepted and unquestioned. The calculation of the peak intensities (heights) or AUC are affected by changes in the signal-to-noise ratio and the resolution of the individual peaks within the spectra. Combine this with the problem of quantifying individual peptides of similar mass that form sets of overlapping peaks and even with SIS methods, quantification can become a difficult task using either of these signal intensity measures. Methods like liquid chromatography can be used to isolate convolved peptides but this adds additional sources of sample loss, are expensive in terms of both manpower and funding and do not scale easily to high-throughput workflows.

In the study of the renin-angiotensin system (RAS), the use of SIS methods for peptide quantification have been used to map out the extracellular processing of angiotensinogen prior to its biological action [2-9] at a given cellular target. Ang peptides from the RAS serve as good examples of isolated isotopic clusters (Ang-II: octopeptide, molecular weight (MW) 1046 Da) and convolved clusters (Ang-(2-10): nonapeptide, MW1081 Da and Ang-(1-9): nonapeptide, MW 1083 Da) detected by mass spectrometry.

While Gaussian mixtures have been used in the analysis of mass spectra [38-43], as mentioned earlier, these methods do not incorporate known characteristics, such as probable isotopic distribution of a compound, and have not been used to address the issues of peptide quantification. The proposed Gaussian mixture method takes into account the known physical constraints, such as isotopic mass separation and point distributions. The peak areas are estimated with the same error range as the peak intensity and peak AUC methods of SIS quantification in Ang peptide data. The range of error is carried over to convolved groups of peptides where no direct comparison of methods can be made. This method is easily automated using an R-package, for implementation of a multiple peptide search over several spectra.

Due to the abundance of data generated from MS analysis, there are several software packages that can be used to aid in the analysis of MS data. These include commercial packages such as Progenesis MALDI (Nonlinear USA Inc., Durham, NC) and many open platform packages that have been produced individually [44-49]. Each of these uses different methods for identifying and quantifying mass spectra, the details of which have not been published. Here, the aim is to show the versatility of partially known Gaussian mixture method in dealing with overlapping peptide clusters in the framework of SIS peptide quantification.

Materials and Methods

The purpose of this study was to provide a new approach to the problem of quantifying single and convolved peptides in MALDI-TOF MS data using a Gaussian mixture model to measure and compare native peptides to SIS peptides. This approach will be compared to the established methods of single peptide quantification: peak intensity and Riemann sum AUC peak quantification for single peptide quantification.

Peptides of the RAS were used for studying the characteristics of our approach and for comparing with other approaches.

Mass Spectra Collection

Samples were examined using MALDI-TOF MS. Ratios of native and SIS peptides (Sigma-Aldrich,St. Louis,MO) were mixed in 2% aqueous trifluoroacetic acid (TFA). The SIS peptides are 6 Da larger than the native peptide as a result of [^{13}C.^{15}N]-valine incorporation into the amino-acid sequence. Concentrations of native and labeled peptides ranged from 20 to 1000 nmol/L (Tables 1 & 3) depending on the ratio required. SIS-Ang-(1-9) has a MW of 1189.56, SIS-Ang-(2-10) a MW of 1187.71 and SIS-Ang-II a MW of 1052.59.

Samples were applied to a MALDI target with a sandwich of α–cyano-4-hydroxycinnamic acid (cyano matrix) mixed in a one to one ratio (10 g/L) with 50% acetonitrile/0.1% TFA. The sandwich consisted of 2 µL of cyano matrix, 2 µL of sample, then another 2 µL of matrix. Each application was allowed to dry prior to the application of an additional layer. Spectra were collected in reflectron mode using a M@LDI MALDI-TOF mass spectrometer (Waters Corp., Milford, MA and AB SCIEX, Farmingham, MA). Twenty spectra were combined for analysis and were converted from MassLynx .raw directories to .mzXML files using MassWolf [48] or from .mgf to .mzXML using ProteoWizard [49] MSConvert for import into an R computing environment, version 3.01, [50] for analysis.

Mass Spectra Data Processing

Once the data were imported into an R environment, the XCMS [51-53] package is used to load the .mzXML file and isolate the region around the known mass of the group of peptides of interest. This range is -1 m/z from the monoisotopic mass [M+H$^+$]

of the smallest peptide to +5 m/z from the monoisotopic mass of the largest peptide in the group. These ranges of peptide masses were grouped together based on the overlap of isotopic clusters of individual peptides within the above range. Peak area estimation was performed by constructing a Gaussian mixture model for each peptide.

The Gaussian mixture is a multimodal distribution the density of which is produced by a weighted sum of Gaussian densities. In the MS context, the density could be written

$$f(x;\Delta,\sigma)=\sum_{k=1}^{K}\lambda_k f_k(x;\mu_0+(k-1)\nu^0+\Delta,\sigma), \tag{2.1}$$

where $f_k(x;\mu_0+(k-1)N^0+\Delta,\sigma)$ is the Gaussian density with mean $\mu_0+(k-1)N^0+\Delta$ and variance σ^2 for the *k*th component of the Gaussian mixture, Δ is the mass error (or accuracy) of the spectra due to error in the standard curve calibration of the mass spectra, λ_k is the proportion of the *k*th component as defined by the isotopic distribution of the peptide that is limited by the total amount of the peptide accounted for, 99.99% in most instances, N^0 is the mass of a neutron (1.00866912 Da). The square root of the variance, namely the standard deviation, σ, could also be interpreted as the peak width. Peak resolution can be obtained from the standard deviation by simple transformation of the 'Full-Width Half-Maximum' of the peak and it can be explained in terms of σ as $2\sqrt{2\ln 2}\sigma$. Although general Gaussian mixtures allow for the variance of each component Gaussian density to vary, since the resolution of individual peaks in MALDI-TOF are equal [1] with the change in variance between peaks increasing ($\sigma_i=\frac{m_i}{m_{i-1}}\sigma_{i-1}$), it is reasonable to set the variance across the examined m/z range to be equal since the change in variance is very small over the range being examined. The mass difference between peaks of a single peptide is N^0. The addition due to this mass is

negligible over the small m/z range seen in SIS quantification. For clusters of peptides, the Gaussian mixture of each peptide is combined across the mass range without additional weighing of the individual peptides. The peptide peak areas associated with the mass error and peak width (namely Δ and σ) yielding the best fit is used as the area estimates for that cluster of peptides. Goodness of fit of the model is determined by the R^2, the coefficient of determination (computed as the average of the squared distance between the observed and estimated peaks). The R^2 is calculated over a range of mass error (Δ) and peak width (σ) values and the area corresponding to the combination of Δ and σ that yields the highest R^2 is considered the final estimate of the peak area.

Examples of model fits using the Gaussian mixture are shown in Figures 1, 3 and 4 where the fit of a single peptide (Ang II, Figure 1) and a convolved set of peptides

Figure 1: Different Fitting Methods

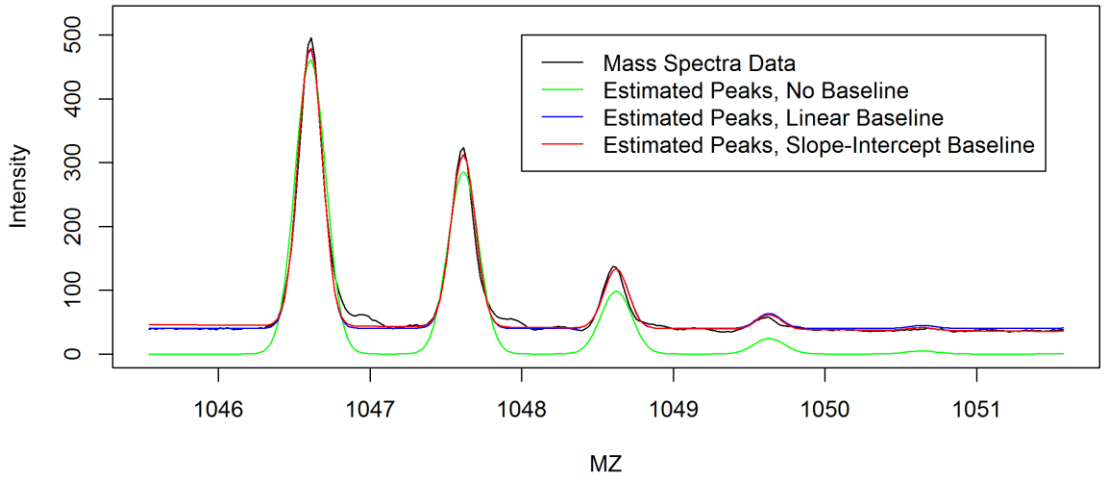


Figure 1: Different methods used for fitting a single peptide isotopic cluster to a MALDI-TOF spectrum of unlabeled Angiotensin II. The inclusion of a slope-intercept form baseline (red estimation) increases the fit over a flat baseline (blue estimation). Both of which are better than not including a baseline (green estimation).

(Ang-(2-10) and Ang-(1-9) and SIS peptides, Figure 3 and 4). The estimated individual contribution of each peptide can be seen in in Figure 4. Often a vertical shift is observed in spectra and needs to be accounted for in SIS estimations [54-55]. This shift, hence forth called baseline shift, could be caused by many sources one of which might be the

noise from the MS signal. It is believed that this noise could act in either an additive or masking (overlapping) fashion. There is no universally accepted method in the literature for estimating or correcting for this baseline shift. However, generally this error seems to be treated as additive and is treated as such with all methods discussed in this paper. The baseline shift is described here as a slope-intercept form of a linear function over the m/z range examined for each peptide cluster. Figure 1is presented to illustrate the importance of including the baseline correction in the quantification. For this example, there is a less adequate fit (R^2=.64) when no baseline is assumed, and the fit improves as the baseline is estimated, which improves further changed from a simple vertical shift (R^2=.97) to a shift on a gradient (R^2=.99) described as a slope intercept equation. Figures 3 and 4 shows the fit of a convolved peptide model (R^2=.94) using a slope intercept baseline shift.

The proposed approach estimates the mass error and peak width (Δ, σ respectively), over a range decided by the researcher. Then peak area for each individual peptide and the slope intercept of the baseline shift are simultaneously estimated using a QR decomposition of a linear model between each individual peptide, the intercept, and the m/z range the cluster covers, and the spectrum data over that same m/z range. Then the mass and isotropic distribution based on the composition of each labeled and native peptide is estimated. Peptides with masses within 5 DA of each other are then convolved to estimate their abundance. (An R [49] package implementing this algorithm to estimate the mass error, peak width, and peak area has been written and will be made available subsequently.) A full workflow for the Gaussian mixture method algorithm as described is shown in supplemental figure 1.

Two currently available methods for SIS quantification are considered for comparison against the Gaussian mixture method. The methods are the peak intensity measure and the Riemann sum AUC methods of quantification. The peak intensity

measures the height of the identifiable peaks in a given isotopic cluster and sums them. The Riemann sum AUC is trapezoidal sum of the area under the identifiable peaks. For single peptide clusters, the peak intensities of the visible major peaks (usually monoisotopic, M+1, M+2) and a Riemann sum of the major peaks were estimated for comparison (Table 1). Both peak intensity and Riemann sum methods require a baseline

Table 1: Method Comparison Summary

Method	MPE	MSE	Variance	Bias	95 % CI
Gaussian Mixture	0.05172	0.00018	0.00201	0.05154	[0.03449,0.06895]
Peak Intensity	0.05876	0.00030	0.00465	0.05845	[0.03255,0.08497]
Riemann Sum	0.07691	0.00045	0.00598	0.07646	[0.04718,0.10664]
Gaussian Mixture, Convolved Peptide	0.06801	0.00064	0.00264	0.06737	[0.03765,0.09837]

Table 1: The mean percent error (MPE), MSE, Variance and bias of each method's percent error of the peptide ratio for various methods of ratio quantification. While all methods fall within the error parameters of the SIS method The Gaussian mixture model produces estimates in both single and convolved peptides while the peak intensity and Riemann sum methods of estimation cannot be used in convolved peptides.

shift correction, which are often made using one of several methods in the literature [26-27, 44-47, 54-55]. (For the comparisons made in this manuscript, the baseline shift estimated from the Gaussian method will be used.)

Theoretically, the expected locations of the peaks are known to be at the mass for each peak (M, M+1, M+2) shifted by the error adjustment. The new masses (M+Δ, M+1+Δ, M+2+Δ) are each then used to center a ν^0 range on the mass-to-charge ratio axis (MZ). The peak intensities within this range are searched for the maximum intensity. Once the estimated baseline at this MZ is found, it is subtracted from the overall intensity at this point and this is used as the peak intensity for that peak. Riemann sums are calculated over these same three MZ ranges using the formula

$$\sum_{i=1}^{n}\left[\frac{(h_i+h_{i+1})\cdot(x_{i+1}-x_i)}{2}-\frac{(BL_i+BL_{i+1})\cdot(x_{i+1}-x_i)}{2}\right], \qquad (2.2)$$

where h_i is the intensity of the spectrum at point x_i on the MZ axis and BL is the baseline estimate used for noise subtraction at this point on the MZ axis.

Once the peak areas for the best fitting model are collected the ratios of native to labeled peptide are calculated form the peak intensity, Riemann sum areas and Gaussian mixture areas. (Table 1) Since peak intensity and Riemann sums cannot be estimated for all individual peptides in overlapping isotopic clusters of peptides, only the Gaussian mixture method estimates are obtained for overlapping peptides.

Once ratios were calculated for the various measures of signal intensity for the pairs of native and labeled peptides these were used to calculate percent error form the know ratio present in the sample. The absolute difference between the known and estimated ration was divided by the known ratio. These percent errors in ratio estimation were then used to compare between samples and peptides. This was done to limit the interference inherent in samples with a range of signal intensities.

Statistical Analysis

To compare the various methods, the fits are expressed as percent error of a given ratio and the mean, mean square error (MSE), variance and bias for each of the three methods are presented. (Table 1, Supplemental 1) Scatter plots of the percent error of the true ratio between methods for each spectra analyzed (n=26, Supplemental Data 2) will be presented for graphical illustration of the agreement between methods (Figure 2). A two-way ANOVA with peptide (Ang II and Ang-(2-10)) and method (Peak Intensity, Riemann

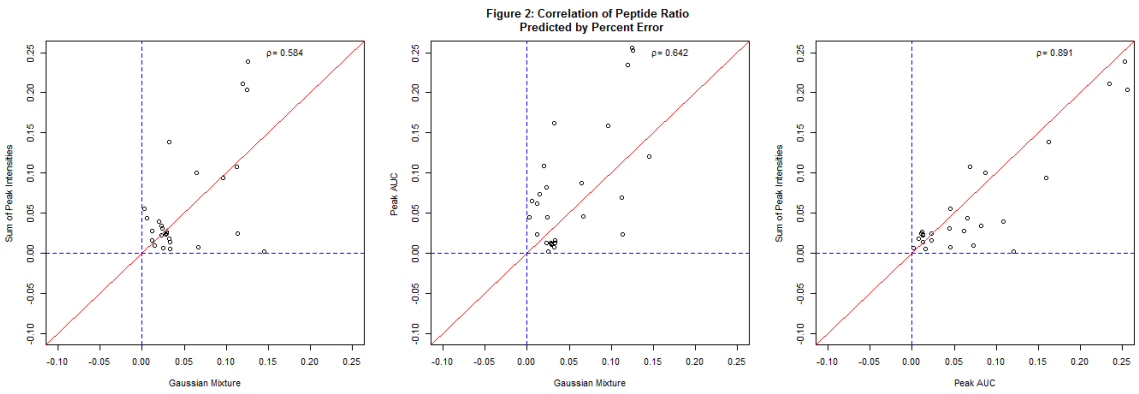


Figure 2: Correlation plots showing the difference in estimation error of the peak ratio for a given spectrum when different methods of peak ratio measurement. The red line denotes a correlation of ρ=1 and the blue lines denote 0% error in ratio estimation for that given method. Here we see that the Peak intensity and Riemann sum AUC methods of quantification correlate more highly with one another than with the Gaussian mixture method. Note that the GMM estimates tend to cluster closer to the blue line suggesting lower error.

Sum AUC and Gaussian mixture model) as independent factors and adjusting for the subsampling (sample replicates) by including a random effects term in the model will be used to formally test the null-hypothesis that the methods are similar. From this analysis the least squares estimates of the mean for each method, along with post-hoc confidence intervals adjusted using Tukey's approach will be presented. (Table 2)

Table 2:Two way ANOVA with pairwise testing

Method	**Mean Estimate**	**SE**	**Pr>|t|**	**95 % CI**
Peak Sum	0.07181	0.01159	<0.0001	[0.049,0.095]
AUC Sum	0.08999	0.01159	<0.0001	[0.067,0.113]
Gaussian	0.05562	0.01159	<0.0001	[0.032,0.079]

Pairwise comparison of means			
Method	**Mean Estimate Difference**	**Pr>|t|**	**95 % CI**
Peak Sum v. AUC Sum	-0.01818	0.2712	[-0.051,0.015]
Peak Sum v. Gaussian	0.01619	0.327	[-0.017,0.049]
AUC Sum v. Gaussian	0.03437	0.0399	[0.002,0.067]

Table 2: The Two-way ANOVA analysis takes into account that several samples are replicates of a single mixture of peptides (Supplemental Data 1) and that there may be differences between peptides used and not just the methods of peak quantifcation.

Since the currently available methods are not applicable for the estimation of convolved peptides, there cannot be a direct comparison between the above methods using convolved peptide data. An analysis of mean, mean square error (MSE), variance and bias was used to compare single and convolved peptide estimation using the Gaussian mixture method. (Table 1)

Results

Single Peak Analysis

As a proof of principle, the Peak Intensity and Riemann sum AUC methods of signal measure and the Gaussian mixture method were used to examine 26 spectra (9 Ang-(2-10)/ SIS-Ang-(2-10) and 17 Ang-(1-9)/SIS-Ang-(1-9)) that consisted of replicate MALDI-TOF analysis of seven different mixtures. These measures were then used to back calculate a ratio of native to labeled peptide which was then compared to the true ratio. Since varying ratios were involved, the percent error of the true ratio was used to measure predictive capacity of all three methods. Peak intensity and Riemann sum used the first three visible peaks for comparison (M, M+1, M+2). The Peak Intensity method was found to have a mean error of estimation of 5.9[3.3, 8.5] %. The Riemann sum method was found to have a mean error of estimation of 7.7[4.7, 10.7] %. The Gaussian mixture method was found to have a mean error of estimation of 5.2[3.4, 6.9] % (Table 1). The mean errors seem to fall within the range of accepted SIS accuracy [34] and share low variances across all three methods.

Correlation plots between methods show that the peak intensity measure and Riemann sum are highly correlated (ρ=0.89) and that the Gaussian mixture method is similarly correlated (ρ=0.58,0 .64) with the other methods implemented (Figure 2).

The two-way random effects ANOVA gives more accurate estimates (Least Square Means) for the comparison of the methods (Table 2). The estimated means of the methods are 7.2 [4.9, 9.5] % for Peak Intensity, 9.0 [6.7, 11.3] % for Riemann sum and 5.6 [3.2, 7.9] % for Gaussian mixture. The difference between Peak Intensity and Riemann sum was not significant and the difference between Peak Intensity and Gaussian was not significant ($p>0.05$), but the difference between Riemann sum and Gaussian methods was significant ($p<0.04$).

The analyses of different ratios of Ang-(2-10) and Ang-(1-9) that were considered are summarized in Table 3. Ten samples of the three ratios were used for a total of 30 spectra. The 2-10 was increased while 1-9 and both SIS peptides were kept constant. The mean error estimate was 2.97[2.5,3.4]% for 1:1 ratio, 5.7[.57,10.7]% for a 2:1 ratio, and 5.3[4.2,6.4]% for a 10:1 ratio.

Figure 3: Fitting a Peptide Cluster

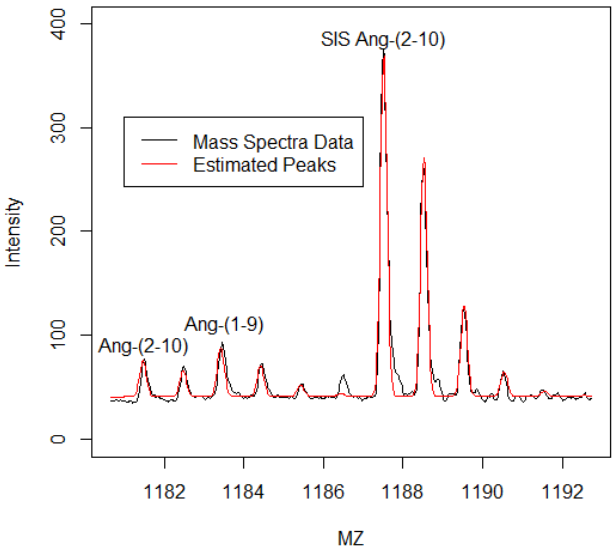


Figure 3: A MALDI-TOF mass spectrum from the analysis of Ang I extracellular breakdown [2] by rat glomeruli in the presence of amastatin (APA inhibitor) and thiorphan (NEP inhibitor) at 60 minutes. The sample contains a mixture of Ang-(2-10), Ang-(1-9), and SIS-Ang-(2-10) that overlap forming one cluster. These peaks are fit and the individual areas for each isotopic cluster can be decomposed from the spectrum.

Convolved Peak Analysis

Convolved peaks are formed by overlapping ionic currents as described earlier. An example of a typical convolved peak problem consisting of multiple peptides is shown in figure 3. The Gaussian mixture method is the only method capable of decomposing convolved peptides. To examine how well the Gaussian mixture method can be used to estimate peptide ratios for this type of quantification, eleven spectra

Table 3: Secondary peptide ratio estimation

2-10:1-9 Ratio	MPE	MSE	Variance	Bias	95 % CI
1:1	0.02971	0.00100	0.00013	-0.02146	[0.0254,0.03401]
2:1	0.05684	0.01975	0.01835	-0.05555	[0.00574,0.10795]
10:1	0.05315	0.00361	0.00087	-0.02585	[0.04201,0.06428]

Table 3: Here the Gaussian mixture method is used to recover the peptide ratio of the second peptide in a convolved set. The error in the estimation of an Ang-(1-9) peak ratio against its corresponding SIS peptide when convolved with various amounts of Ang-(2-10) and its corresponding SIS is used to test the Gaussian mixture method. The initial concentration of 300 nM of each peptide (for a 1:1:1:1 ratio of peptides, Supplemental Data 2) is modified by changing the amount of Ang-(2-10). Here we see that the Gaussian mixture method can recover the second peptide from a series of different peptide ratios.

representing convolved peptides (5 replicate Ang-(2-10)/SIS-Ang-(2-10)/SIS-Ang-(1-9) and 6 replicate Ang-(2-10)/ Ang-(1-9)/ SIS-Ang-(2-10)/ SIS-Ang-(1-9)) were analyzed. Ang-(2-10) ratios and Ang-(1-9) ratios were calculated and compared to the true ratios. The mean error of estimation from these eleven spectra was found to be 6.8[3.8, 9.8] %.

Discussion

Our findings demonstrate, the Gaussian mixture method is capable of handling both single and convolved peptides for the estimation of SIS ratios with similar accuracy but the performance of the method is sensitive to peak resolution and signal to noise ratio. The single peptide estimations produce similar (or better) results compared to the two previously used methods. For convolved peaks, the Gaussian mixture method produced similarly accurate results, while previous two methods treat those situations intractable. All of the means fall within the acceptable levels [18, 30, 35-37] of error for SIS quantification and provide a basis for the equivalence of the results from the Gaussian mixture model method for estimating convolved and non-convolved peptides. Gaussian mixture method is more advantageous because it can be applied to both single peptide and multiple, overlapping peptides with at least the same accuracy as past methods. It also supplies a mathematical justification for baseline estimations instead of an *ad hoc* approach.

There are a few limitations in using all the three methods and some are specific to the Gaussian mixture. A closer examination of the correlation plot (Figure 2) reveals a

Figure 4a: Fitting a all Peptides

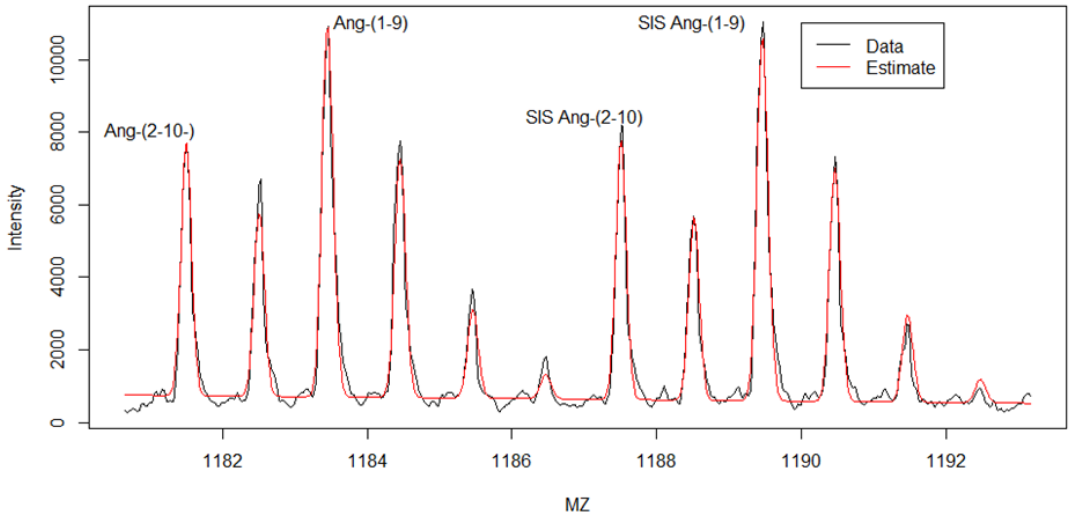


Figure 4b: Ang-(2-10) Contribution

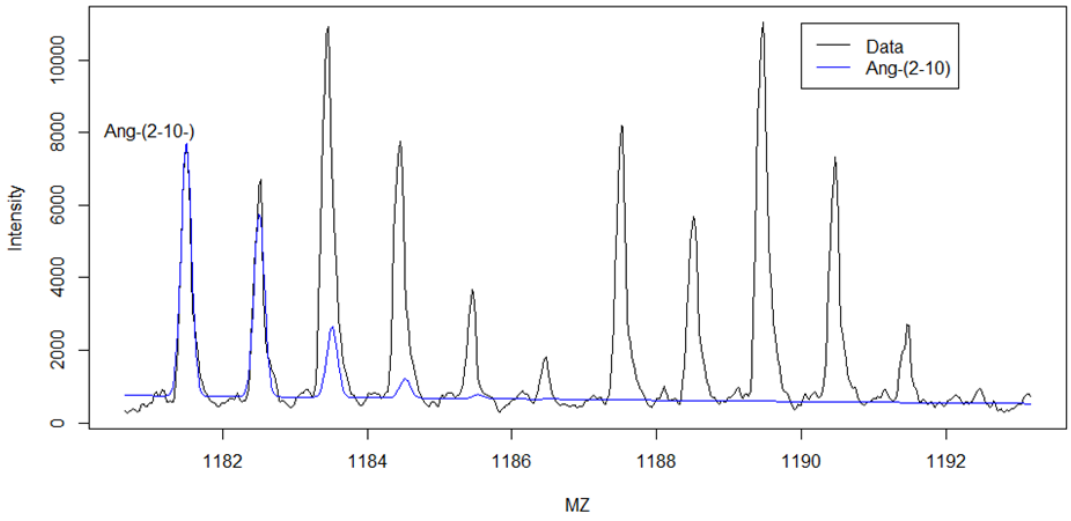


Figure 4c: Ang-(1-9) Contribution

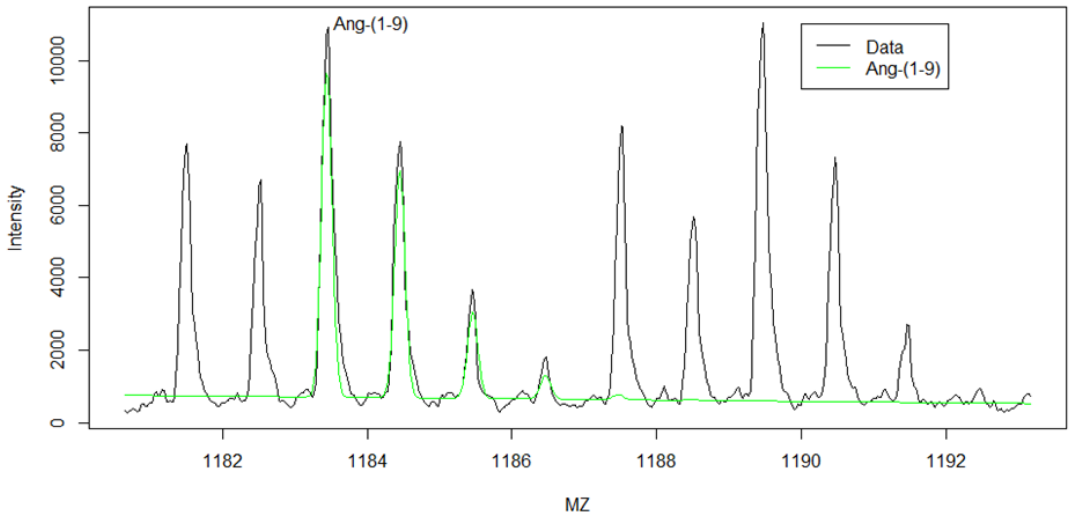


Figure 4d: SIS Ang-(2-10) Contribution

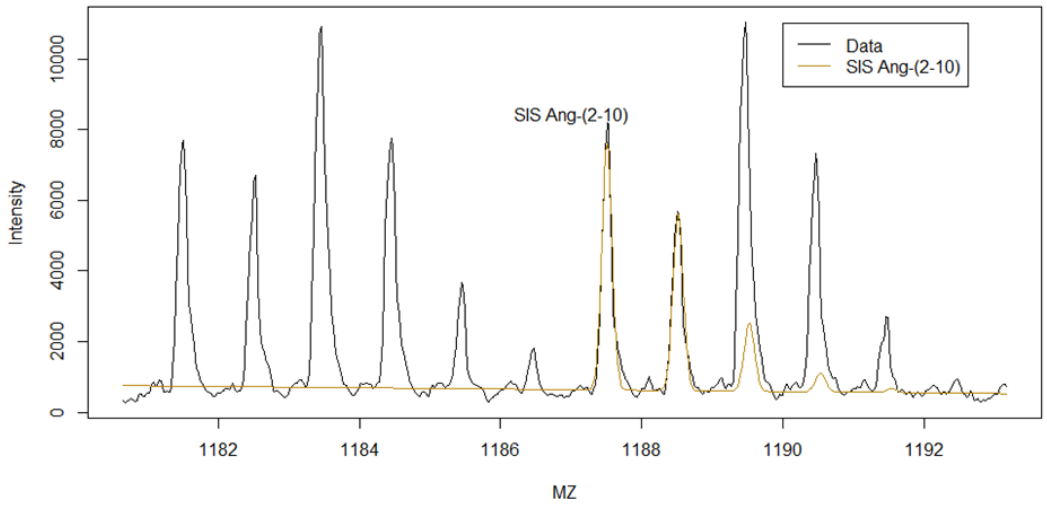


Figure 4e: SIS Ang-(1-9) Contribution

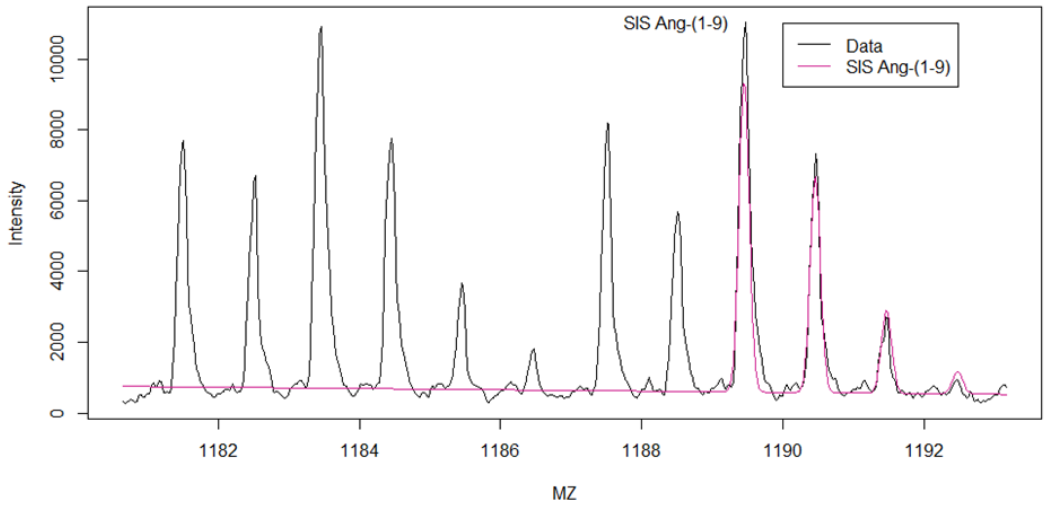


Figure 4: A MALDI-TOF mass spectrum of a known ratio of 1:1:1:1 peptides consisting of 300nM Ang-(2-10), Ang-(1-9), SIS-Ang-(2-10) and SIS-Ang-(1-9). The spectrum has been fit using GMM and the figure shows how each estimated peptides contributes to the whole spectrum. Since all peptides are estimated simultaneously, each peptide is presented here separately to illustrate the individual contribution of each peptide to the spectrum as a whole. (A) Figure 4a shows the entire estimation as a whole, preformed as a single fit to a single cluster of four overlapping peptides. The data is shown in black with the estimated peaks superimposed in red. (B) Figure 4b shows the estimated contribution of Ang-(2-10) to the spectra superimposed in blue. (C) Figure 4c shows the estimated contribution of Ang-(1-9) to the spectra superimposed in green. (D) Figure 4d shows the estimated contribution of SIS-Ang-(2-10) to the spectra superimposed in dark yellow. (E) Figure 4e shows the estimated contribution of SIS-Ang-(1-9) to the spectra superimposed in dark purple.

grouping of points that seem to be outliers. These points that cluster furthest from the diagonal represent samples that on closer examination had lower resolution and/or exhibited skewed peaks. This cluster of three data points is farther from the main cluster of data, suggesting a poor estimation of the ratio using all methods of peak quantification. The ability to calculate the native:SIS ratio is affected by the quality of the data being examined. Quality can be quantified by the resolution (or variance of the component normal of the Gaussian mixture distribution) of the peaks. Low quality (high variance, low resolution and/or misshapen) peaks are harder to quantify using the Gaussian mixture method. In other words, if the underlying assumption of normality under each peak is violated the Gaussian mixture method might produce larger errors. The Gaussian method is more sensitive to the resolution, returning higher error ratio estimates with the lower resolution spectra than previous methods. The Gaussian mixture method does predict ratios with better accuracy with higher resolution spectra than previous methods. This needs to be explored further by analysis of signal to noise ratios and there correlation with resolution. It is anticipated that the higher resolution will produce larger signal to noise ratios, which would explain the sensitivity. This method is dependent on knowing the exact mass of peptides being quantified in a given sample. Because GMM data derived from MALDI-ToF alone analyzes only the intact charged mass, not reacted to produce highly specific fragment ions (e.g. b/y ions) like that for other mass spectrometry modalities, unknown compounds that are nearly identical in mass can confound the accuracy of the measurement. Only the highest resolution instruments, such as MALDI-Fourier transform ion cyclotron resonance mass spectrometers, can achieve peak resolution that can minimize this overlap. Furthermore, due to their low abundance in plasma, enrichment strategies are often necessary to measure vasocactive peptides by MALDI-TOF [29,56] which is a low sensitivity detection system in the presence of a

high matrix environment. In experiments attempting to profile the metabolism of vasoactive peptides and quantify the end-products, where GMM is most relevant, requires the addition of an exogenous peptide to a high concentration necessary to elevate the signal to detectable levels [2-5, 7,8,10]. Whether or not GMM is applicable to the analysis of native biological samples is likely to require testing on a case by case basis using high resolution instruments prior to the use of MALDI-ToF alone.

The error in estimation increases for convolved peaks (compared to single peak estimation errors which ranged between 0.051% and 0.068% (Table 1). The analysis of Ang-(2-10) with a convolved set of SIS peptides shows that convolved peaks when decomposed can be estimated within the same error range as single peptide peaks but sets of convolved peptides (Figure 3,4) show an increase in the error of estimation (Supplemental Data 1). This estimation error may be corrected in MALDI-TOF data by adjusting the peak width estimation by a correction based on the static resolution of the data. This will be explored in future research. Even with these increased errors in estimation of the convolved peptide ratios, the ratios are estimated within the same allowable error range [35]. It is possible that additional mixture parameters, such as a variable peak sigma that narrowly increases across peptides' m/z range or the use of flyability constant as in previous work [2] may need to be incorporated to estimate multiple sets of convolved peptides.

Future refinement of the Gaussian mixture method will require the examination of several aspects of the algorithm. The appropriate cutoff for the number of isotopic peaks that constitutes the significant majority of the peptide in the sample also affects the minimum m/z that needs to be considered for calling two peptides separate. This defines their status as a convolved cluster or as peptides to be considered individually. This is expected to be a function of the atomic composition of a given compound, where the

more atoms comprising the molecule lead to a larger and more complex isotopic distribution. Methods for adjusting estimations of peptide isotopic distributions that reflect possible local variations will need to be considered to see if they are viable and make a significant addition to area estimations. Implementation of a maximum likelihood estimator of the Gaussian parameters will increase both speed and accuracy of this method, but other measures of 'goodness of fit' need to be explored. Implementation of a quadrant search algorithm for exploring the parameter space needs to be implemented to accelerate peak quantification for larger data sets. Finally, simulation studies are required to validate this method over a wide range of extremes in spectra composition. Such a study is being considered and will appear in a subsequent publication.

The use of informed Gaussian mixture method is a novel approach to peptide quantification with the tangible benefits of the flexibility to tackle traditional single peptide cases and overlapping peptides as well. It also provides baseline estimation with mathematical justification. This process can also be automated for multiple peptides over multiple spectra allowing for a high through put quantification analysis. The Gaussian mixture method is comparable to both Peak Intensity and Riemann sum methods of signal measure in SIS quantification. When dealing with convolved peptides we show similar levels of error relative to non-convolved peptide area and ratio estimates with the Gaussian method. The Gaussian method is equivalent, will remove the *ad hoc* baseline estimations used else-where, and will give estimations that fall within the range of acceptable SIS error for both convolved and non-convolved peptides. This method could be implemented in a reasonable amount of time for quantification of any compound, with known composition, examined using mass spectrometry and an internal standard. The use of the Gaussian mixture is also variable since mixtures of other distributions could be used to better describe other spectra where necessary.

3. Manuscript two: Algorithmic Improvements and Simulations of partially known Gaussian mixture model estimates for stable isotope standard quantification in MALDI-TOF MS

3.1 Abstract

The quantification of peptides in Matrix assisted laser desorption/ionization time-of-flight (MALDI-TOF) mass spectrum (MS) analysis coupled with stable isotope standards (SIS) has been used to quantify native peptides under many experimental conditions. This peptide quantification by MALDI-TOF approach has difficulties quantifying samples containing peptides with ion currents in overlapping (convolved) spectra. Here we present an examination of the Gaussian mixture model (GMM) method for peak quantification using simulated spectra and an associated algorithmic method for mass spectra simulation. The algorithmic implementations of GMM and spectra simulation are discussed in detail. Enhancements in the GMM algorithm implementation including expansion of the baseline estimation to a *p*-polynomial function and a quadrant ascent search method for crossing the parameter search space. The Monte Carlo simulations of mass spectrum containing randomly generated peptides with auto regressive modeled noise and baselines were algorithmically generated to test the ability of Gaussian mixture models to preform area estimation under different parameters. We show that under a wide variety of conditions we are able to estimate peak areas and peak ratios within an acceptable range (2-12%) of error. In both single (1.7-10.3% area error) and convolved sets (0.3-9.98% ratio error) of peptides, estimation error is acceptable across all simulations with GMM displaying distinct patterns of behavior. We illustrate the behavior of GMM and pinpoint its dependence on signal to noise ratio in peak quantification. GMM is a valuable tool in MALDI-TOF MS peptide quantification with SIS. The GMM is sensitive to signall to noise ratios but is able to estimate peak area and area ratios with in published limits of SIS quantification. GMM is accurate in both single peptide and convolved peptide simulations GMM is a valuable tool in MALDI-TOF MS

peptide quantification with SIS under multiple conditions.

3.2 Background

Mass spectrographic (MS) based techniques have advanced the field of Quantitative proteomics far beyond what is capable with 2D gels or western blots with the capacity to identify and quantify thousands [1-2] of proteins and postranscriptional modifications in a single experiment [3,4]. MS based techniques can separate proteins regardless of abundance or solubility, unlike 2D gels, and to identify these proteins from their unique ion fragmentation profiles. The ability to quantify proteins using MS is hindered by inherent physical and chemical properties. Differences in the charge, matrix effects, hydrophobicity, or posttranslational modification are just some of the properties that effect the ion formation and time of flight of a sample peptide or mixture. The work flow of these experiments must be carefully managed due to possible loss of some of the sample during preparation or because of contamination from unusual sources. Due to these considerations internal standards with properties as similar as possible to the material in question are preferred during quantification experiments. Peptides labeled with heavy carbon and nitrogen are one source of internal standards and are referred to as SIS peptides.

Stable isotope dilution theory is based on the concept that a stable isotope labeled protein or peptide would behave exactly the same during MS analysis since their only difference form their native, unlabeled counterparts is the addition of extra neutrons. Since the mass difference between the labeled and unlabeled samples can be detected through MS, a comparison can be made based on their signal intensities and quantification achieved. The SIS method involves placing a known quantity of labeled peptide in a sample and comparing the peak intensities between the labeled and native peptide. SIS has an error range from 2% to 12% [3] depending on the quantification

method used. The MS intensities can be measured in several ways and have been compared [5] in previous work. GMM is one method for quantifying MS SIS data. Previous work [5-6] has shown that in a limited set of peptides, in real data, that GMM is capable of both single peptide and convolved peptide quantification within a reasonable error margin. Past attempts to simulate mass spectra have been done to aid in sequence prediction of ion fragments and the identification of peptides in mass spectra.[7-10] Several tools have been developed with this task in mind.[11-12] Here we aim to simulate spectra to test the GMM under multiple conditions and to go into further detail about the implementation of a GMM algorithm. Improvements in this implementation are explained along with simulation studies to show the flexibility of GMM. Future possibilities for improvement and expansion of the GMM are also explored.

3.3 Methods

3.3.1Mixture Models

The Gaussian mixture model is a combination of two or more Gaussian distributions and it consists of the Gaussian parameters (μ_k, σ_k^2) for each individual distribution and a mixture coefficient (λ_k) that describes the contribution of the individual Gaussians to the overall distribution. An observation from this distribution will have density of the form

$$\sum_{k=1}^{K} \lambda_k f_k(x; \mu_k, \sigma_k^2) \tag{3.1}$$

where $f_k(x; \mu_k, \sigma_k^2)$ is the Gaussian density with mean μ_k and variance σ_k^2.

In the context of mass spectra analysis of peptides this density can be parameterized using known characteristics of the peptides. The mixture coefficients (λ_k) can be determined by the isotopic distribution of the peptide. The mean of the first Gaussian (μ_1) can be determined within an unknown mass error Δ to account for error in

the standardization of the spectrum. Subsequent means are known to be separated by the mass of a single neutron. That is, $\mu_k = \mu_0 + (k-1)\nu^0 + \Delta$, where μ_0 is the mean of the first Gaussian (or where the first peak occurs) and ν^0 is the mass of a neutron (1.00866912 Daltons). The variance of the *k* Guassians can be assumed to be equal due to the small range of MZ being examined in SIS quantification [4,13]. This leads to a new parameterization and a reduced dimensionality in the GMM to fit a mass spectrum of a given peptide or series of overlapping peptides, where the unknown parameters are the mass error, Δ, and the variance (peak width, σ). In the case of overlapping peptides we mix two or more Gaussian mixtures depending on the number of convolved peptides being estimated.

Since mass spectra have some form of baseline error or noise a second trapezoidal distribution can be combined with the Gaussian mixture (or the mixture of Gaussian mixtures in the case of overlapping peptides) to mimic this noise. This also can be treated as a mixture of trapezoidal and Gaussian mixtures and estimated as a straight line equation in slope-intercept form with the parameters of slope (α) and intercept (β). These noise parameters can be estimated for each combination of the parameters Δ and σ while estimating the peak area for the quantification. The proposed algorithm does this by implementing a search of Δ and σ for the best model as defined by the highest coefficient of determination found in the parameter space.

3.3.2 GMM Algorithm

The GMM algorithm is a method for the quantification of peptides from MALDI-TOF MS data. The algorithm is designed to estimate the peak area or area under the curve of all peaks making up a single peptide. In cases of convolved peptides, where the isotopic clusters of individual peptides overlap, this algorithm simultaneously estimates the peak areas for each convolved peptide separately. To begin a search of spectra, a list

of peptides needs to be provided, described using the single letter code for the constituent amino acids of the peptides. This algorithm was implemented in R [14] and will be released as an R package for future public use. A workflow for this algorithm can be seen in in figure 1.5.

The algorithm starts with estimating the $M+H^+$ mass of the peptides being searched for and ordering them by mass and placing them in clusters based on mass. These clusters are used to represent convolved peptides and a maximum mass difference is specified to distinguish peptides that are convolved (in the same cluster) from those that do not overlap. An example would be the masses of Angiotensin-2-10 (Ang-2-10, molecular weight (MW) 1181.7 Daltons (Da)) and Angiotensin-1-9 (Ang-1-9, MW 1183.6 Da) which when seen in a mixture would overlap (or be convolved) in a mass spectra while Ang-2-10 and Angiotensin-1-8 (Ang-1-8, MW 1046.5 Da) would not. The spectrum is then scanned for the peptide or peptides in question by looking at a window based on the mass range including the lightest and heaviest peptides in the cluster. This is prespecified but is generally set at one Dalton upstream from the monoisotopic mass of the lightest peptide and five to seven Daltons downstream from the monoisotopic mass of the heaviest peptide. In our previous example of Ang-2-10 and Ang-1-9, the window would extend from MZ 1180.7 to 1190.6 in the spectrum. This is the search window the algorithm uses for estimating the peak area.

Once the search window is defined, a search space of σ and Δ that are used to construct Gaussian mixture models of the peptide(s) is determined. The Δ is, as mentioned earlier, used to estimate the mass error due to standardization. For each combination of σ and Δ models of the peptide are constructed. These models start with estimating the isotopic distribution of the peptide based on the isotopic distribution of its constituent elements. This estimation is capped at 99.99% of the peptide being accounted

for. This prevents a model from being constructed that is too cumbersome to use. It also saves computational time sense the peaks being unaccounted for constitute so little of the peptide as to be too small (<0.01%) to be seen in any real data (lost in the noise) and are outside the window being searched. Each Gaussian curve in the model is defined by the monoisotopic mass of the peptide plus the neutron mass that increases with every isotope of the peptide. The Gaussian mixture model is a multimodal distribution, the density of which is produced by a weighted sum of Gaussian densities. As mentioned in the previous section, using the parameterization described under the MS context, the mixture Gaussian density in (1.1) reduces to

$$f(x;\Delta,\sigma)=\sum_{k=1}^{K}\lambda_k f_k(x;\mu_0+(k-1)\nu^0+\Delta,\sigma), \tag{3.2}$$

where $f_k(x;\mu_0+(k-1)\nu^0+\Delta,\sigma)$ is the Gaussian density with mean $\mu_0+(k-1)\nu^0+\Delta$ and variance σ^2 for the *k*th component of the Gaussian mixture, Δ is the mass error, as mentioned earlier, of the spectra due to error in the standard curve calibration of the mass spectra, the weights λ_k is the proportion of the *k*th component as defined by the isotopic distribution of the peptide that is limited by the total amount of the peptide accounted for, 99.99% in most instances, ν^0 is the mass of a neutron (1.00866912 Da). The square root of the variance, namely the standard deviation, σ, could also be interpreted as the peak width. Although general Gaussian mixtures allow for the variance of each component Gaussian density to vary, and the resolution of individual peaks in MALDI-TOF are equal [4], meaning the σ gradually increases. This can be treated a constant in the model since the mz range being examined is small. The mass difference between peaks of a single peptide is ν^0. For clusters of peptides, the Gaussian mixture of each peptide is combined across the mass range without additional weighing of the individual peptides.

Each peptide in the cluster is constructed separately and represented in a matrix that we call a model matrix. The baseline noise in the mass spectrum is also included in the model matrix. The baseline noise is defined by descriptors of the slope and intercept represented in the model matrix by a vector of 1's and the corresponding MZ values in the window being examined respectively. Each of the peptides modeled and baseline descriptors (α, β) are columns in the matrix.

To estimate the peak areas and baseline noise the model matrix is equated to a vector of peak intensities and a QR decomposition is applied to produce the solution which constitutes a vector containing estimates of peak area for each peptide in the cluster and the α and β baseline parameters. The use of QR decomposition acts as a linear regression where we estimate the peak areas and baseline. A more through explanation of QR decomposition can be found in the supplemental material.

The peptide peak areas, cluster slope and cluster intercept are then used to construct a model to be compared against the data. The cluster model is used to calculate the coefficient of determination (R^2) as a goodness-of-fit metric. The combination of σ and Δ that returns the highest R^2 is then stored along with the associated peak areas as the best estimate of the spectrum.

Once all peptides in all spectra have been examined over the σ and Δ search space, a user based filtering step is done to remove false positives. This is a two-step process where threshold peak areas and R^2 are sought and any estimation that does not meet these thresholds are rejected. The peak area threshold is needed since it is possible to fit noise in a spectrum with very small peaks that can return either positive or negative areas.

This is used in peptide quantification were ratios to internal standards are used to quantify the amount of peptide in the sample.

3.3.3 Algorithm Enhancements

Two enhancements to the algorithms may be considered. First one is to speed up the Δ and σ parameter search and the second is to expand the ability to describe the baseline error in the model fit by allowing non-linear shits.

The difficulty in applying the GMM to large data sets of spectra to quantify many peptides may arise due to the computational time required to examine the full search space of Δ and σ across all peptides for all spectra. To speed up the search space of possible peak models an exhaustive search of all possible combinations of Δ and σ could be replaced by a search using a quadrant ascent method. The quadrant ascent method operates from an initial set of values of the two parameters within the specified ranges and searches through the range of one of the two parameters for a fixed value of the other parameter. When a best fit (highest R^2) is found in the range of the first parameter, at that value the search continues in the range of the second parameter and so on until convergence. Specifically, starting with an initial value for Δ, all possible σ's are searched and the σ that produces the best fit is determined. Then, for this σ, all possible Δ's are searched and the Δ that produces the best fit is obtained. This process is repeated until the R^2 has reached a maximum within the resolution of the specified parameter grid or it does not increase beyond a certain tolerance. The quadrant ascent method is useful in speeding up the convergence to the best fitting parameters when the distribution of the criterion (here R^2) is quadratic around the maximum. The distribution of R^2 can be shown to be quadratic around the maximum for estimating the two parameters under the Guassian mixtures. The acceptable parameter range is influenced by experimental conditions and spectra resolution. Therefore, it is possible that the best fitting model may not be within the selected parameter range. In such cases, the best fit may occur for large values of Δ and σ and this is most likely to be due to tuning errors with the mass spectrometer that may need to be addressed before applying the algorithm. In other words the assumption

made in our algorithm is that the maximum R^2 occurs within the Δ and σ ranges searched.

The linear baseline estimation could be improved by setting higher order polynomial functions. Higher order polynomials could be included in the QR decomposition by simply adding columns that are powers of the MZ range.

3.3.4 Analysis of Existing Data for Selection of Simulation Parameters

Data characteristics and subsequently the simulation parameters were determined through an examination of 1789 MALDI-TOF MS spectra from previously published work containing singular and convolved peptides [15-19]. Based on the figures of the various spectra we observed several noise and signal characteristics and the corresponding estimates of the noise and model parameters were calculated. The following are highlights of some of these characteristics; 1) The tails of the spectra around peptides demonstrated autocorrelations and baseline shifts that were often linearly changing, 2) this noise was constant between peaks and consistently exhibited auto-correlations, 3) The noise seemed additive, 4) the signal appeared to drown the noise out with relatively smooth increases in intensity, and 5) the peaks for a given compound were separated by constant intervals and the peaks appeared to have the same width regardless of intensity. The fifth characteristic observed confirms our reparametrization of the Gaussian mixture given in 1.1.

The generation of noise for our simulated spectra was an important part of replicating data that accurately represent real experiments so that the characteristics of the GMM could be studied in the context of applying it to real situations. The inclusion of noise component in the simulations was accomplished by using a time-series approach. Based on the observations mentioned above, a first order autoregressive model (AR(1)) was used to add the noise in the tails and between peptide peaks. Windows of two Daltons were selected both up and down stream of the known ionic cluster or peaks in the 1789

spectra, to determine the magnitude of the autocorrelation and the white noise parameter, ζ. This analysis showed that the noise was best fit by and AR(1) model with one parameter, no constant, and uniform white noise. That is

$$X_n = \rho X_{n-1} + U(0,\zeta) \qquad (3.3)$$

where X_n is the intensity of the simulated spectrum at the *n*th entry of the MZ range, and $U(0,\zeta)$ stands for a uniform distribution in the interval $(0,\zeta)$. The most common estimate of ρ in these data was 0.7 and the mean of ζ was found to be approximately two units of intensity. In the following simulations, these parameters were used to generate model noise over a window of 400 elements longer that than the window of spectra being simulated to allow for a burn in of the autoregressive model and for pseudo-random behavior to be seen. All noise generation was seeded at X_0=10. This was done to since seed selection does not overly influence noise generation after burn in and is of minimal importance in this context since it is located in a tail of the noise generation vector and is not added to the simulation. The tails of 200 elements of this model noise vector are removed before combining with the peptide simulation.

Simulation pick count, or the number of samples from the Gaussian distributions that are used to model the individual peaks was selected at 100,000. This comes from an analysis of the behavior of histograms with the average bin size estimated from the examined spectra. An average bin size of 0.02 was selected since the MZ increments ranged from 0.16 to 0.24 Daltons in these spectra. The number of samples gives a close to normal appearance to our peaks but allows for some non-normal features to appear in the generated histogram at random.

3.5 Simulation Algorithm

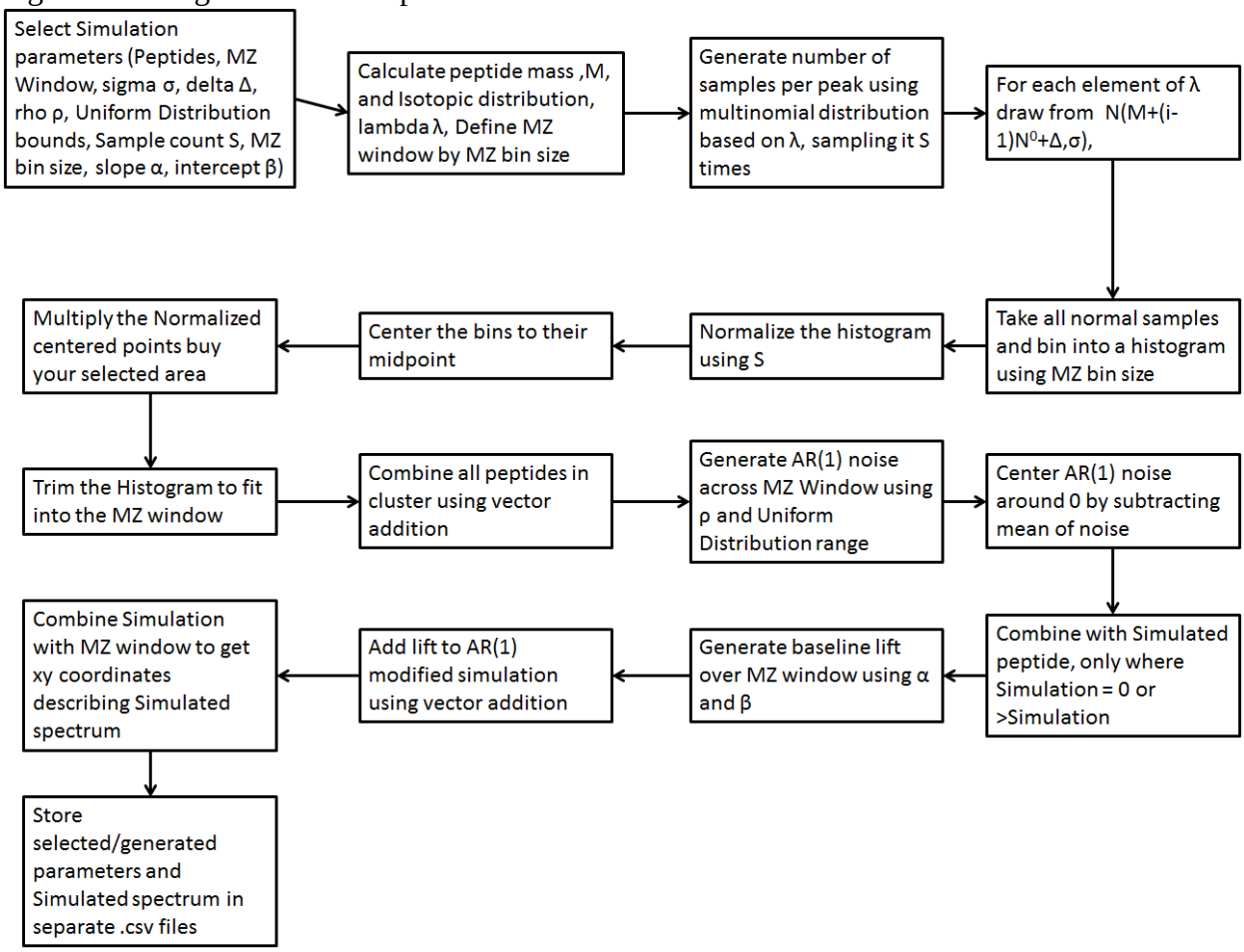
This simulation algorithm consists of three main parts. First is the generation of the isotopic peaks in the spectrum. Second is the addition of fine level noise using an

auto regressive (AR(1)) model. The third is a baseline shift that includes the AR(1) noise, is generated to model the gross changes in the spectrum due to noise. This was done for separate peptides, native and SIS, and for convolved peptides, where the isotopic peaks overlap one another. This algorithm was implemented in an R computational environment [14] and used to generate spectra under a number of parameters to test the flexibility of the Gaussian mixture method. The simulation algorithm will be made user-friendly and subsequently published as an R function in the GMM package.

This algorithm breaks a spectrum down into several characteristics, which will be referred, simulation conditions. These consist of: the peptide(s) being simulated, total peak area (A) ,the size of the window and its bin width (step size) around that peptide on the mz axis, peak width (σ), mass error adjustment (Δ), AR(1) parameter (ρ), uniform distribution for white noise, sample count (S), bin size, slope (σ), and intercept (β). In the simulation R package these conditions could be randomly generated or specified based on prior knowledge. For the purposes of the examination of GMM, the combinations of values under these simulation conditions were selected based on the experience from the existing data, listed earlier. While this is not an exhaustive list of possible influences of spectrum production, it does provide enough variation to produce a large range of spectrum to test the GMM to adequately study the properties of the GMM approach. A summary of this algorithm workflow can be seen in Figure 4.

The algorithm starts by calculating the mass (μ_0) and isotopic distribution (λ_k) of the peptide being simulated. Isotopic distribution is estimated by convolving the elemental distributions for each atom in the peptide being simulated. Individual elemental isotopic distributions come from the CRC [20] and are universal distributions that may have local variations. A multinomial distribution based on λ is then sampled S times to determine an empirical probability vector (λ_k') describing the number of samples

Figure 4: Data generation for Spectra simulation



per isotopic peak (i.e., the *k*th mixture normal component) in the peptide. For each λ_k, the number of samples in λ_k is taken from the corresponding Gaussian density, namely $N(\mu_0 + (k-1)\nu^0 + \Delta, \sigma)$. All of the samples from every Gaussian distribution sampled in this way (one for each isotope of the peptide being modeled) for this peptide are then binned into a histogram for a specified bin size. This histogram is then normalized by dividing by S so that the area under the peaks approximates 1. The histogram bins are then centered to the bin midpoints, matching the normalized bin count to a single point on the MZ axis of a spectrum. The centered, normalized histogram is then multiplied by the selected peak area A for the peptide being simulated. The modified histogram is then truncated to fit into the selected MZ window. Although, this essentially truncates the mixture density at both ends of the tail, the density under the trimmed region is expected to be negligible. Once each of the individual peptides in a cluster are

generated in this fashion they are combined by vector addition of the individual peptide modified histograms to create the cluster of overlapping isotopic peaks in the cluster.

Generating noise starts with the generation of an AR(1) noise sampled along the MZ window, using the bin size to measure the steps within the window, with an extra 2 Dalton (200 element) addition at either end of the window. This AR(1) uses the autocorrelation ρ and the uniform distribution selected on a specified range. Once generated the AR(1) noise is centered on zero by subtracting the mean of the noise generated within the MZ window. This centered noise is then combined with the simulated peaks with a simple substitution, where if the peak simulation is zero or less than the noise being generated it is replaced in the simulation with the noise. This mimics our treatment of noise being an additive factor and not a masking one.

The baseline bulk noise is added using a straight line constructed over the MZ window using intercept α and slope β over the range of the window. This is then added to the AR(1) modified simulation to finish modeling the noise seen in MALDI-TOF spectra. The parameters and simulated spectra are stored in separate files.

3.5.1 Simulation parameters

To test the predictive accuracy of GMM for peak parameter description a series of simulation were performed using the algorithm detailed above. In this series of single peptide simulations peak area (50, 100, 250 units), σ (0.05, 0.1, 0.2), and Δ (-0.1, 0, 0.1) were varied and 1000 spectra were simulated for each combination of parameters (Table 1). Baseline, noise generation, and peptide (Ang-1-8) were kept the same.

To test GMM ability to detect peak contribution in cases of convolved peptides, simulations of Ang-(2-10) and Ang-(1-9) were performed using variable, reversed ratios of peaks ranging from 1:1 to 1:10 (Table 2). All parameters (σ=.1, Δ=.1, slope=-2, Intercept=50) were kept the same except for the peak areas for each peptide. Each set of

ratios were simulated 1000 times.

To test the efficacy of peak area ratio calculations, simulation of Ang-(2-10) and Ang-(1-9) with SIS- Ang-(2-10) and SIS-Ang-(1-9) were performed (Tale 3) to see what effect different ratios of the native peptides had on the ratio back calculations. All parameters (σ=0.1, Δ=0.1, slope=-2, Intercept=50) except for peak area were kept stable and 1000 samples were obtained in each case.

To test the range of convolved peptides capable of being estimated using GMM a series of two by two, four, eight and 16 sets of convolved peptides (Table 4) were simulated with each peptide having a peak area of 200 units with all other parameters kept the same. These peptides were randomly generated and then ordered by mass. Peptides that had a minimum of two Daltons in difference of mass were used. The two by two peptide simulation refers to a series where the second and third peptide are separated by a minimum mass difference of 5 Da but all four peptides are considered one cluster. A complete list of the peptides generated and used can be found in the supplemental information. All parameters (σ=0.1, Δ=0.1, slope=-2, Intercept=50) were kept the same and 1000 simulations of each set of peptides were performed.

All simulations used a random number seed of 031851.

3.4 Statistics

To summarize the performance of the GMM in these simulations bias, mean square error (MSE), for the estimation of the model parameters and percent error (Tables 1-4) for the peak area were computed under various simulation conditions. These are also used to discuss the peak area ratio estimations for convolved peptides and sets of convolved peptides as seen in SIS quantification. A graphical representation showing a comparison of the calculated MSE and the Bias squared (Figures 1-3) in a scatter plot to illustrate the correlation and accuracy of simulation estimation.

3.5 Results and Discussion

Previous work in publication using this method has shown that GMM has similar estimative capacity [5] to other methods of peak quantification. These SIS quantification problems are dependent on estimations using comparisons of peak evaluation. GMM uses an estimation of peak area based on a constructed model of MS peaks for that given peptide to quantify the amount of peptide in a sample by estimating the total area under the peaks that describe a given peptide in a spectrum. These quantification problems can be broken up into single peptide or multiple convolved peptides problems. This simulation analysis aims to discover the behavior of GMM estimation over a variety of different conditions.

In the single peptide simulations (Table 1-2, Figure 2) several trends can be seen. Any parameter change that decreases the signal to noise ratio, which can be seen as decreasing peak (signal) intensity, has the effect of decreasing the accuracy of peak

Table 1:Simulation of Single Peptide Peaks

	mzError		-0.1			0			0.1	
Sigma	Area	50	100	250	50	100	250	50	100	250
	Bias	0.079	0.14	0.363	0.058	0.107	0.222	0.046	0.061	0.154
0.05	MSE	0.051	0.186	1.08	0.048	0.165	1.027	0.043	0.159	1.005
	Percent Error	0.393	0.381	0.37	0.373	0.348	0.349	0.346	0.33	0.333
	Bias	0.075	0.05	-0.044	0.037	0.02	-0.092	0.023	-0.017	-0.126
0.1	MSE	0.049	0.1	0.485	0.039	0.089	0.469	0.039	0.09	0.429
	Percent Error	0.353	0.25	0.22	0.314	0.231	0.213	0.309	0.235	0.204
	Bias	0.123	0.062	-0.075	0.136	0.042	-0.07	0.137	0.02	-0.142
0.2	MSE	0.143	0.223	0.866	0.169	0.221	0.712	0.162	0.217	0.745
	Percent Error	0.595	0.375	0.298	0.651	0.377	0.267	0.642	0.37	0.275

Table 1: GMM tried under combinations of several parameters. Note that as sigma increases or area decreases, GMM predictive ability falls. This is due to the noise to signal ratio inherent in the given simulation. As the peaks become more distinct from the baseline noise they become easier to estimate. Each combination of parameters was simulated 1000 times.

prediction. In the single peptide simulations (Table 1) we see that error in estimation increases as peak width is increased or as peak area is decreased. Both of these

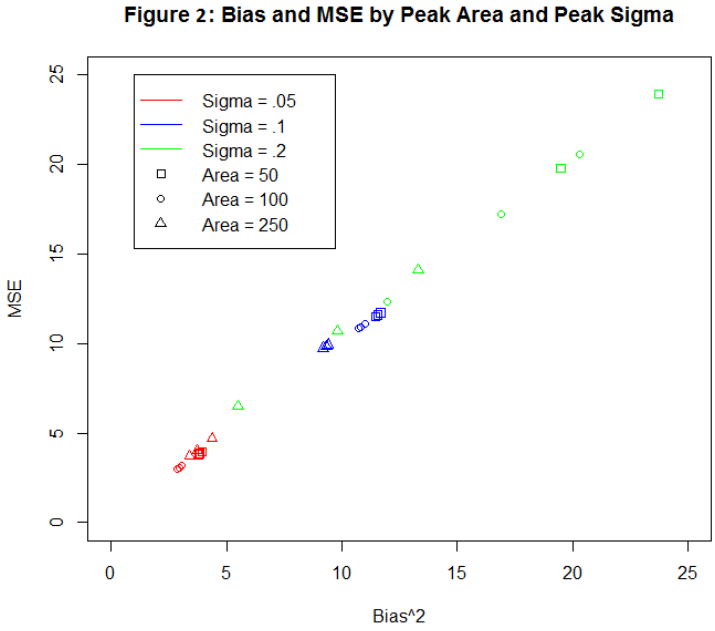


Figure 2: This figure shows the correlation between MSE and Bias squared as shown above with the parameters that most directly affect the signal to noise ratio. As this ratio is decreased (area decrease, sigma increase) we see an increase in both MSE and Bias.

parameters decrease the signal to noise ratio. The higher error seen in the smallest peak width is due to the fact that the resolution of the peaks is too high for this bin size and thus we have fewer points to estimate the peak area. There is still accurate estimation but a noticeable increase in error. This can also be seen in Figure 2, where it is clear that the increases in bias and MSE follow the same pattern. This also shows that the accuracy and precision also suffer as the signal to noise ratio decreases. The simulation of a single set of convolved peptides (Ang-(2-10) and Ang-(1-9), Table 2) shows similar behavior in the ratio estimations.

When analyzing simulated convolved peptide data, GMM shows a tendency to be more error prone in the estimation of the last peptide in the convolved cluster. For different ratios of convolved peptides (Table 3, Figure 3) similar effects to single peptide estimation, where the signal to noise ratio plays an

Table 2: Convolved Peptide Peak Area Estimation						
Bias	0.017	0.029	0.009	-0.046	0.007	0.007
MSE	0	0.001	0	0.005	0	0
Percent Error	1.668	0.738	3.414	0.592	6.866	6.884
Ratio	1:1	1:4	4:1	1:10	10:1	10:1
Peak Areas	200:200	200:800	800:200	50:500	500:50	1000:100

Table 2: Convolved Peptide Peak Area Estimation. Here the peak areas were estimated and the ratio of peaks was estimated and compared to the real ratio.

important role in the quality of peak estimation are observed. This is compounded when the second peptide has less contributing area to the cluster (Table 3 500:50, 1000:100 ratios). The same decrease in accuracy and precision can be seen in figure three where the second peak ratio

is harder to estimate in every example. It is also apparent that the effect of baseline estimation on the smaller of the two amounts (areas) of peptide on estimation area is greater. The peaks that are due to the peptides with the larger amount of material are proportionately less affected by errors in baseline estimation. These changes in error are a measure of degrees however, since all estimation errors are within the allowed range.

Table 3:Separation of Convolved Peaks

	Ang-2-10 ratio			Ang-1-9 ratio		
Ratio	**Bias**	**MSE**	**Percent Error**	**Bias**	**MSE**	**Percent Error**
200:200	-0.884	1.049	0.453	2.467	6.395	1.233
200:800	0.075	0.678	0.317	6.17	41.287	0.771
800:200	-5.189	31.574	0.65	5.577	33.922	2.788
50:500	0.557	0.419	1.149	3.331	12.295	0.666
500:50	-3.48	13.842	0.696	3.11	10.488	6.22
1000:100	-7.121	58.097	0.713	6.223	42.107	6.223

Table 3: Separate estimations of convolved peak simulations of Ang-2-10 and Ang-1-9. Here we look at the ability of GMM to pull apart convolved peptides and estimate independent peak ratios. The ratio refers to the native simulated peptides compared to the SIS peaks each had 200 units of area under each peptide.

The last simulation was a series of convolved peptides that demonstrated the ability of GMM to estimate long chains of convolved peptides for analysis. Each of the simulations shows the same pattern of having the

Figure 3: Bias and MSE of Convolved Peak Estimations

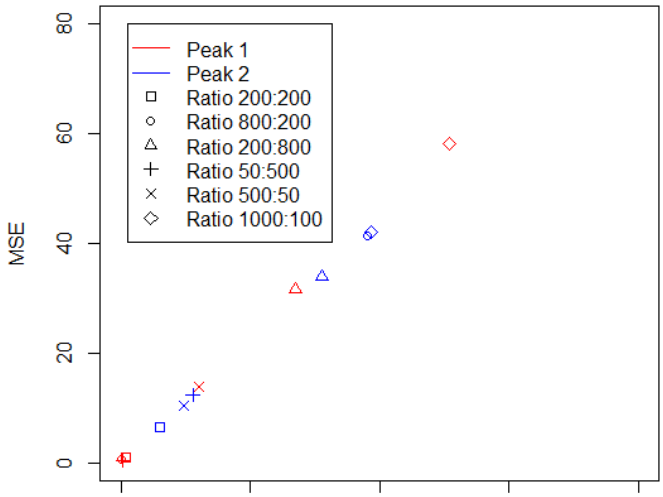


Figure 3: This figure shows the correlation between MSE and Bias squared for the separation of the convolved peaks

most estimation error pool in the last peptide being estimated. We show the mean estimation error (Table 4) and the individual peak estimations (Figure 4). Here we see that while estimation error is within the established bounds we see that the last peptide estimation of each simulation cluster together in the upper right corner of Figure 4 showing an increase in error estimation but still maintain an estimated error within the acceptable bounds for SIS quantification.

Each set of simulations highlight the behavior of GMM. In almost all conditions, the estimation error is acceptable but follows a predictable pattern that shows the importance of signal to noise ratio and its effects on peak quantification.

Table 4: Summation different sets of convolved peptides

Peaks	Bias	MSE	Percent Error
2 by 2	2.182	5.125	1.091
4	2.424	6.198	1.211
8	2.373	6.031	1.187
16	2.311	5.71	1.155

Table 4: An analysis of multiple sets of convolved peaks by examining the peak area of each peptide. The peptides were randomly generated to be 2 Daltons apart in mono-isotopic mass.

3.6 Conclusions

GMM is a robust method for peptide quantification in MALDI-TOF experiments using SIS peptides. Simulations confirm, both single and convolved peptide areas and area ratios can be estimated within an acceptable range of error. While GMM shows some sensitivity to signal to noise ratio, as reflected with increases in estimation error with wider peaks or lower peak areas that decrease the signal strength, this method still has a robust tolerance and can estimate peak areas and subsequent ratios for quantification with precision and confidence. The unique properties of MALDI-TOF and the nature of fitting the peptide model reduce the complexity of using GMM and make GMM an ideal tool for this class of problems. The algorithm described above is meant for high-throughput use, scanning multiple spectra for multiple peptides and their SIS counterparts. Here we have shown its application in a subset of circumstances to illustrate its utility. However, this method is not perfect and can be improved in several

ways.

The isotopic distribution used in this algorithm is based on a universal elemental distribution from literature. Since local isotopic distributions for a given element or compound can be different, regional elemental distributions need to be considered. We assume that all simulated peptides have equal flyability in these simulations but this can be relaxed in a strait forward manner by applying a flyability constant to the data to reflect differences in matrix effects or ionization.

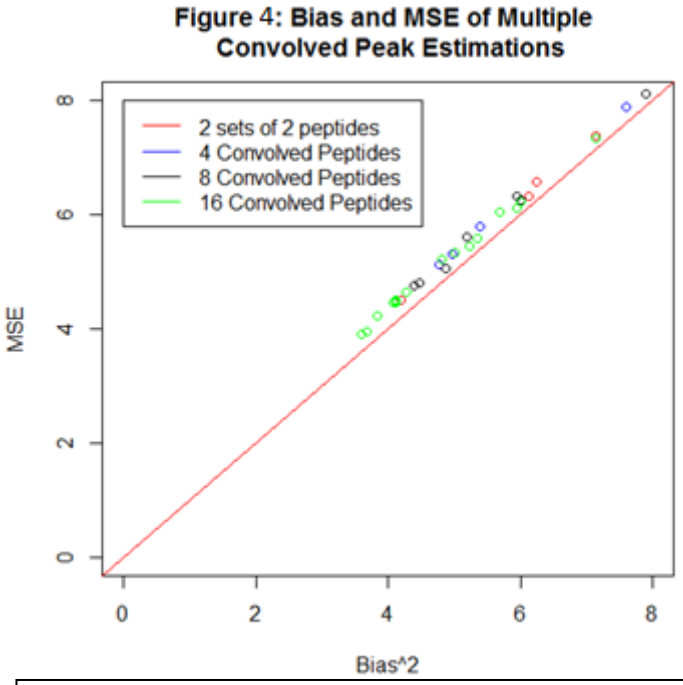


Figure 4: Here we see the individual peptide estimation error for each set of convolved peaks regardless of which set of convolved peptides are estimated.

Future work with GMM will include an analysis of the minimal separation between peaks that is needed for accurate peak quantification. Since this method has an obvious weakness with peptides or other compounds with identical (or nearly so) monoisotopic mass, a possible minimal peak separation based on spectra resolution needs to be explored. As noted previously the estimation error is greater in second peak estimation of convolved sets of peptides. This will need to be addressed by exploring alternate methods for peak analysis including solving for first two peaks of a cluster to estimate unseen peaks using GMM then subtracting estimated peaks from the convolved peaks. This allows for an estimation of the 2nd peptide as a single isotopic cluster. This is

expected to compound errors over long chains of convolved peptides but needs to be explored using GMM.

Future improvements will include a maximum likely-hood estimator (MLE) for parameter estimation using an expectation-maximization (EM) algorithm. This is feasible since there are only two parameters being considered for influencing the GMM. Previous work using EM algorithms has been limited due to the number of model parameters being considered. Since MALDI-TOF has unique restrictions governing peak width across the spectrum and individual known peptides have defined distances between peaks, a MLE becomes the next logical step for parameter estimation.

GMM is a versatile and robust method for peak quantification that has utility in multiple instances of peptide quantification in MALDI-TOF MS. It is ideally designed for high-throughput analyses and can be implemented in a straight forward manor.

4. Manuscript three: The expectation-maximization of the maximum likely hood estimator of parameters for informed Gaussian mixture models used in the quantification of peptides from MALDI-TOF mass spectra

4.1 Abstract

We present an implementation of a Gaussian mixture model that is used to quantify peptides from Matrix-Assisted Laser Desorption/Ionization Time-of-Flight (MALDI-TOF) Mass spectrometry (MS) by estimating peak area for individual peptides. We estimate a maximum likelihood estimator (MLE) using an expectation-maximization (EM) algorithm for the unknown parameters in our Gaussian mixture model (GMM). This GMM already has several parameters that can be estimated from known chemical and physical data but the estimation of the mean shift and Gaussian mean can be inferred from available data within the spectra and does not need to be searched for. This method allows for the automated scanning of mas spectra for the quantification of peptides within the spectra.

4.2 Introduction

Proteomics is the large-scale experimental analysis of proteins. The focus of proteomics is to understand the role and function of proteins in biological processes through examination of protein structure (sequence), modification, function, or quantity. These examinations usually entail the analysis of samples using immune precipitation, purification, gel electrophoresis or mass spectrometry. The role played by mass spectrometry includes those of protein identification and quantification. Matrix-assisted laser desorption/ionization (MALDI) time-of-flight (TOF) mass spectrometry (MS) is one version of MS used that has the unique ability to analyze biological molecules without breaking them up during analysis. MALDI-TOF is used for identification and quantification. Quantification can be done by using an internal standard in the sample preparation. This is normally done when the material in question is already identified. In

the example of peptides in biological samples, a stable isotope standard (SIS) is used [1]. This is a version of the peptide to be quantified that has been created using heavy labeled elements. It is chemically identical to the normal (native) peptide but is more massive due to the extra neutrons added by the heavy elements.

Systems biology is the study of biological systems or networks and the web of interactions between the components of systems and the systems themselves in living organisms. Proteomics has been used to shed light on these interactions by looking as large bodies of data to make inferences about biological networks. One attempt to do this is the construction of a mathematical model of a network, for example an enzymatic pathway, based on the known substrate kinetics of the enzymes in that pathway. One methodology for this is biochemical systems theory, where pathway being modeled is represented by ordinary differential equations where the enzyme-substrate reactions (biochemical processes) are individual equations. This means that an entire series of reactions that are interconnected at different levels, where multiple enzymes share multiple substrates, can be modeled with a degree of accuracy. We can then begin to think of a biochemical pathway as a network or processing of material.

Once a model is constructed it can then be perturbed *in silico* (in machine) with either substrate material at different points or by inhibiting different enzymes along the pathway. These mathematical predictions of the behavior of the pathway can then be tested in a laboratory setting (bench side). This is a cyclical process with the data generated from bench-side, wet experiments can be used to improve the model, and then the improved model can then be used to inform the next set of biological experiments. This process can be used to identify new drug targets in a pathway or identify new combinations of drugs tailors to a specific biochemical outcome, where in this case these drugs are inhibitor of enzyme activity. This combination of drugs approach is more

interesting since there is already a large library of drugs that already exist that can be used without the need for going through the drug discovery process.

This process of mathematically defining a biochemical pathway is dependent on several factors including the ability to measure the amount of material in a sample from an experiment. This can be done using the previously mentioned proteomics methodologies. This is important since enzyme kinetics and inhibitor effectiveness can be gauged from a time series experiment where substrate quantification tells how much of a given material is being used or produced at different points in a given biochemical pathway. This use and production can be thought of as the flux of the material at a given point in the pathway. Drugs (inhibitors) act to change this material flow, increasing or decreasing the net amount of material at different points in the pathway.

Thus the problem of quantification becomes important not only due to the need to understand the flow or flux of material through a biochemical network or system but also becomes a computing problem since a large number of data point and associated experiments are needed to illustrate the properties of the pathway in question. The implementation of a computational method for quantification prevents the large amount of data from becoming a bottle neck where the data itself becomes too large to analyze. It becomes necessary to have a versatile method for the quantification of material from proteomics experiments.

In this work we will focus on a statistical approach to quantifying peptides in MALDI-TOF MS data that is capable of analyzing both single and multiple (convolved, overlapping) peptides in spectra and give estimates of the peptides using internal standards. We aim to provide a method that does not require any previous information except the peptide to be quantified and the spectra to be analyzed while producing the parameters of a Gaussian mixture model that describes the peaks seen in the data. This

description included an estimate of the area under the peaks for a given peptide, which is used for its quantification, and the uncertainty of the model parameters estimated from the data. This method can be used to replace past methods of SIS quantification since it can process both single peptide and peptides that overlap one another in the MS. Past implementations of this method have used set ranges describing several parameters and we aim to remove the need of these ranges using MLE's implemented in an EM algorithm.

In past analysis of MS data, statistical analysis has focused on peak detection and peak identification with quantification being left to simpler methods [2-3]. While detection and identification are of the utmost importance in the wider field of proteomics, this method aims to allow for quantification of peptides known to be in the spectra using the statistical underpinning of a Gaussian mixture distribution. Treating the spectra as a GMM allows us to use the well understood mechanisms behind MLE's and EM [4] to incorporate prior knowledge of the peptides of interest to simplify the estimations.

4.3 Methods

4.3.1 Deriving the maximum likelihood estimators for GMM parameters

In MS we only observe for any given x_i, the mass to charge ratio (mz) and h_i which is the frequency of x_i's as detected in the machine as ionic current. The mass spectra data is observed in the form of an n x 2 matrix where the first column represents the random variable (x_i) as the mass to charge ratio (in Daltons) and the second column is the intensity for that x_i. In terms of statistical distributions, the intensity represents the frequency of the i^{th} observation. The frequency histogram often represents a mixture normal distribution. Therefore, this could be modeled using a K component mixture normal of the form,

$$f(x;\Delta,\sigma^2)=\sum_{k=1}^{K}\lambda_k f_k(x;\mu_0+(k-1)\nu^0+\Delta,\sigma^2), \tag{1.1}$$

where Δ is the m/z error of the spectra due to error in the standard curve calibration the mass spectra, σ is the peak width as standard deviation (peak resolution can be substituted with a simple transformation for FWHM), λ_k is the proportion of the *k*th component (isotope of the compound), ν^0 is the mass of a neutron (1.00866912(43) Da), and $f_k(x;\mu_0+(k-1)N^0+\Delta,\sigma)$ is the normal density with mean $\mu_0+(k-1)N^0+\Delta$ and variance σ^2. In proteomics the number of components (K) and the proportion in each component λ_k could be determined through theoretical analysis of elemental distribution. In order to model a frequency histogram of the mixture normal, the h_i's standardized, so that the sum is 1.

Then Likelihood of the Gaussian mixture including the normalized intensities is,

$$L(\sigma^2,\Delta\,|\,\underline{x})=\prod_{i=1}^{n} f_k(x_i;\sigma^2,\Delta)^{h_i} \quad . \tag{1.2}$$

The expanded likelihood including the full expression for the mixture means is,

$$L(\sigma^2,\Delta\,|\,\underline{x})=\prod_{i=1}^{n}\left[\sum_{k=1}^{K}\lambda_k f_k(x_i;\sigma^2,\mu_0+(k-1)\nu^0+\Delta)\right]^{h_i} \quad . \tag{1.3}$$

The log likelihood is,

$$\ell(\sigma^2,\Delta\,|\,\underline{x})=\ln L(\sigma^2,\Delta\,|\,\underline{x}) \tag{1.4}$$

$$\ell(\sigma^2,\Delta\,|\,\underline{x})=\sum_{i=1}^{n} h_i \ln\left[\sum_{k=1}^{K}\lambda_k f_k(x_i;\sigma^2,\mu_0+(k-1)\nu^0+\Delta)\right] \tag{1.5}$$

Let $\mu_k=\mu_0+(k-1)\nu^0+\Delta$. Then,

$$\ell(\sigma^2,\Delta\,|\,\underline{x})=\sum_{i=1}^{n} h_i \ln\left[\sum_{k=1}^{K}\lambda_k f_k(x_i;\sigma^2,\mu_k)\right] \tag{1.6}$$

First derivative of the log likelihood with respect to Δ,

$$\frac{d\ell}{d\Delta} = \sum_{i=1}^{n} h_i \frac{d}{d\Delta} \ln \sum_{k=1}^{K} \lambda_k f_k(x_i; \sigma^2, \mu_k), \tag{1.7}$$

Where

$$f_k(x_i; \sigma^2, \Delta) = \sum_{k=1}^{K} \lambda_k f_k(x_i; \sigma^2, \mu_k) \ . \tag{1.8}$$

Then,

$$\frac{d\ell}{d\Delta} = \sum_{i=1}^{n} h_i \frac{\frac{d}{d\Delta} f_k(x_i; \sigma^2, \Delta)}{f_k(x_i; \sigma^2, \Delta)} \tag{1.9}$$

Since μ_k is dependent on Δ and the derivative of the μ_k with respect to Δ is equil to one we can substitute Δ for μ_k in the dirivative

$$\frac{d}{d\Delta} f_k(x_i; \sigma^2, \Delta) = \frac{d}{d\mu_k} \sum_{k=1}^{K} \lambda_k f_k(x_i; \sigma^2, \mu_k) \tag{1.10}$$

$$\frac{dl}{d\Delta} = \frac{dl}{d\mu_k} \frac{d\mu_k}{d\Delta},$$

$$\frac{d\mu_k}{d\Delta} = 1$$

$$\frac{d}{d\mu_k} \sum_{k=1}^{K} \lambda_k f_k(x_i; \sigma^2, \mu_k) = \sum_{k=1}^{K} \lambda_k \frac{d}{d\mu_k} f_k(x_i; \sigma^2, \mu_k). \tag{1.11}$$

The base Gaussian distribution is

$$f_k(x_i; \sigma^2, \mu_k) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)} \tag{1.12}$$

Since the mixture proportion λ_k is independent of μ_k the derivative can be moved inside the equation for the derivative of the basic Gaussian mixture

$$\sum_{k=1}^{K}\lambda_k \frac{d}{d\mu_k} f_k(x_i;\sigma^2,\mu_k) = \sum_{k=1}^{K}\lambda_k \frac{d}{d\mu_k}\frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}$$
$$= \sum_{k=1}^{K}\lambda_k \frac{1}{\sqrt{2\pi\sigma^2}}\frac{(x_i-\mu_k)}{\sigma^2} e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)} \quad (1.13)$$

The derivative of the full mixture

$$\frac{d\ell}{d\mu_k} = \sum_{i=1}^{n} h_i \frac{\sum_{k=1}^{K}\lambda_k \frac{1}{\sqrt{2\pi\sigma^2}}\frac{(x_i-\mu_k)}{\sigma^2} e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}}{\sum_{k=1}^{K}\lambda_k \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}}$$
$$= \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \frac{\lambda_k \frac{1}{\sqrt{2\pi\sigma^2}}\frac{(x_i-\mu_k)}{\sigma^2} e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}}{\sum_{k=1}^{K}\lambda_k \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}}$$
$$= \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \left(\frac{\lambda_k \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}}{\sum_{k=1}^{K}\lambda_k \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}} \frac{(x_i-\mu_k)}{\sigma^2} \right) \quad (1.14)$$

Here we simplify part of the first derivative by assigning α_{ik}. This is the probability that a given observation (h_i) is in a given Gaussian (K) in the mixture. This serves a role in the implementation of the EM algorithm for parameter estimation.

$$\alpha_{ik} \equiv P(x_i \in K) \quad (1.15)$$

$$\alpha_{ik} = \frac{\lambda_k f_{ik}(\mathrm{x})}{\sum \lambda_k f_{ik}(\mathrm{x})} = \frac{\lambda_k \frac{-1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}}{\sum_{k=1}^{K}\lambda_k \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}} = \frac{\lambda_k e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}}{\sum_{k=1}^{K}\left(\lambda_k e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}\right)} \quad (1.16)$$

Note that

$$\sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} = \sum_{k=1}^{K} \alpha_{ik} = 1 \tag{1.17}$$

therefore

$$\frac{d\ell}{d\Delta} = \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} \frac{(x_i - \mu_k)}{\sigma^2} \tag{1.18}$$

MLE of Δ, expanding out μ_k in the above equation and setting it to zero is,

$$\begin{aligned} &\sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} \frac{(x_i - \mu_k)}{\sigma^2} = 0 \\ &\sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} (x_i - \mu_k) = 0 \\ &\sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} x_i - \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} \mu_k = 0 \end{aligned} \tag{1.19}$$

$$\begin{aligned} \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} x_i &= \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} \mu_k \\ &= \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} (\mu_0 + (k-1)\nu^0 + \Delta) \\ &= \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} \mu_0 + \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} (k-1)\nu^0 + \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} \Delta \\ &= \mu_0 \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} + \nu^0 \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} (k-1) + \Delta \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} \end{aligned} \tag{1.20}$$

$$\sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} = 1$$

$$\sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} x_i = \mu_0 + \nu^0 \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} (k-1) + \Delta. \tag{1.21}$$

Recall that α_{ik} is dependent on Δ, therefore 1.21 is nonlinear and does not lead to an explicit solution. Alternatively an EM algorithm can be applied to simplify the solution. For the EM algorithm complete data needs to be constructed. This complete data consists of the observed data (that is, the intensities and mz ratios) S = {hi, xi}. The

missing data *X*, is in indicator function of the i^{th} observation belonging to the k^{th} normal distribution (peak). That is, $I_{ik} = \begin{Bmatrix} 1, x_i \in K^{th} \\ 0, o.w. \end{Bmatrix}$ for every observation. Such that the complete data is $Y = \{S, X\}$. Then in the E-step of the algorithm we will estimate the missing data conditional on the observed data as,

$$E[I_{ik} \mid X] = \mathrm{P}(I_{ik} = 1 \mid X) = \alpha_{ik} \ . \tag{1.22}$$

Given initial values of delta and sigma we can compute α_{ik}. Then in the M-step, for given α_{ik}, we can solve for Δ using 1.21 but fixing α_{ik}. That is,

$$\hat{\Delta} = \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} x_i - \nu^0 \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} (k-1) - \mu_0. \tag{1.23}$$

Note that

$$\begin{aligned} &\sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} (k-1) = \sum_{i=1}^{n} \sum_{k=1}^{K} h_i \alpha_{ik} (k-1) \\ &= \sum_{k=1}^{K} (k-1) \sum_{i=1}^{n} h_i \alpha_{ik} \\ &= \sum_{k=1}^{K} (k-1) \lambda_k = E(\lambda_k) - 1 \end{aligned} \tag{1.24}$$

Therefore

$$\begin{aligned} \hat{\Delta} &= \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} x_i - \nu^0 \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} (k-1) - \mu_0 \\ &= \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} x_i - \nu^0 (E(\lambda_k) - 1) - \mu_0 \\ &= \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} x_i - E(\lambda_k) \nu^0 + \nu^0 - \mu_0 \end{aligned} \tag{1.25}$$

$$\sum_{k=1}^{K} \alpha_{ik} = \sum_{k=1}^{K} \lambda_k = 1 \tag{1.26}$$

$$\sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} x_i = \sum_{i=1}^{n} \sum_{k=1}^{K} h_i \lambda_k x_i = \sum_{i=1}^{n} \sum_{k=1}^{K} h_i x_i \tag{1.27}$$

$$\hat{\Delta} = \sum_{i=1}^{n}\sum_{k=1}^{K} h_i x_i - E(\lambda_k)\nu^0 + \nu^0 - \mu_0. \tag{1.28}$$

Similarly, to obtain the MLE for σ^2 in the M-step, we take the first derivative of the log likelihood with respect to σ^2. That is,

$$\frac{d\ell}{d\sigma^2} = \sum_{i=1}^{n} h_i \frac{d}{d\sigma^2} \ln \sum_{k=1}^{K} \lambda_k f(x;\sigma,\mu_0 + (k-1)\nu^0 + \Delta) \tag{1.29}$$

$$\begin{aligned} \frac{d\ell}{d\sigma^2} &= \sum_{i=1}^{n} h_i \frac{d}{d\sigma^2} \ln f_k(x_i;\sigma^2,\Delta) \\ &= \sum_{i=1}^{n} h_i \frac{\frac{d}{d\sigma^2} f_k(x_i;\sigma^2,\Delta)}{f_k(x_i;\sigma^2,\Delta)} \end{aligned} \tag{1.30}$$

$$\frac{d}{d\sigma^2} f_k(x_i;\sigma^2,\Delta) = f_k(x_i;\sigma^2,\Delta)\left[\frac{(x_i-\mu_k)^2}{2\sigma^4} - \frac{1}{2\sigma^2}\right] \tag{1.31}$$

$$\begin{aligned} \frac{d\ell}{d\sigma^2} &= \sum_{i=1}^{n} h_i \frac{\sum_{k=1}^{K} \lambda_k \frac{1}{\sqrt{2\pi\sigma^2}}\left[\frac{(x_i-\mu_k)^2}{2\sigma^4} - \frac{1}{2\sigma^2}\right] e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}}{\sum_{k=1}^{K} \lambda_k \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{1}{2}\bullet\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}} \\ &= \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \frac{\lambda_k \frac{1}{\sqrt{2\pi\sigma^2}}\left[\frac{(x_i-\mu_k)^2}{2\sigma^4} - \frac{1}{2\sigma^2}\right] e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}}{\sum_{k=1}^{K} \lambda_k \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}} \\ &= \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \left(\frac{\lambda_k \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}}{\sum_{k=1}^{K} \lambda_k \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}} \left[\frac{(x_i-\mu_k)^2}{2\sigma^4} - \frac{1}{2\sigma^2}\right] \right) \end{aligned} \tag{1.32}$$

$$\frac{d\ell}{d\sigma^2} = \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \left[\alpha_{ik} \left[\frac{(x_i-\mu_k)^2}{2\sigma^4} - \frac{1}{2\sigma^2}\right]\right] \tag{1.33}$$

Sigma MLE

$$\sum_{i=1}^{n} h_i \sum_{k=1}^{K} \left[\alpha_{ik} \left[\frac{\left(x_i - \mu_k\right)^2}{2\sigma^4} - \frac{1}{2\sigma^2} \right] \right] = 0$$

$$\sum_{i=1}^{n} h_i \sum_{k=1}^{K} \left[\alpha_{ik} \left[\left(x_i - \mu_k\right)^2 - \sigma^2 \right] \right] = 0 \tag{1.34}$$

$$\sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} \left(x_i - \mu_k\right)^2 - \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} \sigma^2 = 0$$

$$\sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} \left(x_i - \mu_k\right)^2 = \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} \sigma^2 \tag{1.35}$$

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} \left(x_i - \mu_k\right)^2}{\sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik}} = \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} \left(x_i - \mu_k\right)^2 \tag{1.36}$$

$$\hat{\sigma}^2 = \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} \left(x_i - \mu_0 - k\nu^0 + \nu^0 - \Delta\right)^2$$

We can then substitute in our estimator for μ_k for the EM algorithm

4.3.2 Confidence Intervals for $\hat{\Delta}$ and $\hat{\sigma}^2$ based on the asymptotic distribution of the MLEs.

The calculation of the asymptotic Variance-Covariance matrix for the creation of confidence intervals for sigma and delta the asymptotic variance of the MLE's are necessary. To obtain this we use the information matrix, which is the expected value of the second derivatives and partial derivatives. That is,

$$\begin{bmatrix} \frac{d^2\ell}{d\Delta^2} & \frac{d\ell}{d\Delta d\sigma^2} \\ \frac{d\ell}{d\Delta d\sigma^2} & \frac{d^2\ell}{d\left(\sigma^2\right)^2} \end{bmatrix} \tag{1.37}$$

First we must find the Second derivative of the log likelihood with respect to Δ.

$$
\begin{aligned}
\frac{d^2\ell}{d\Delta^2} = \frac{d^2\ell}{d{\mu_k}^2} &= \frac{d\ell}{d\Delta}\sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} \frac{(x_i - \mu_k)}{\sigma^2} \\
&= \frac{d\ell}{d\Delta}\frac{1}{\sigma^2}\left[\sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} x_i - \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik}\mu_k\right] \\
&= \frac{1}{\sigma^2}\left[\sum_{i=1}^{n} h_i \sum_{k=1}^{K} \frac{d\ell}{d\Delta}\alpha_{ik} x_i - \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \frac{d\ell}{d\Delta}\alpha_{ik}\mu_k\right] \\
&= \frac{1}{\sigma^2}\left[\sum_{i=1}^{n} h_i \sum_{k=1}^{K} \frac{d\ell}{d\Delta}\alpha_{ik} x_i - \sum_{i=1}^{n} h_i \sum_{k=1}^{K}\left[\left(\frac{d\ell}{d\Delta}\alpha_{ik}\right)\mu_k + \alpha_{ik}\right]\right] \\
&= \frac{1}{\sigma^2}\left[\sum_{i=1}^{n} h_i \sum_{k=1}^{K} \frac{d\ell}{d\Delta}\alpha_{ik} x_i - \sum_{i=1}^{n} h_i \sum_{k=1}^{K}\left[\left(\frac{d\ell}{d\Delta}\alpha_{ik}\right)\mu_k\right] + \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik}\right] \\
&= \frac{1}{\sigma^2}\left[\sum_{i=1}^{n} h_i \sum_{k=1}^{K} \frac{d\ell}{d\Delta}\alpha_{ik} x_i - \sum_{i=1}^{n} h_i \sum_{k=1}^{K}\left[\left(\frac{d\ell}{d\Delta}\alpha_{ik}\right)\mu_k\right] + 1\right]
\end{aligned}
\tag{1.38}
$$

A_{ik} is the derivative of α_{ik} with respect to Δ is

$$
A_{ik} = \frac{d\alpha_{ik}}{d\Delta} = \frac{\lambda_k e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}\frac{(x_i-\mu_k)}{\sigma^2}}{\sum_{k=1}^{K}\lambda_k e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}} - \frac{\lambda_k e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}}{\left(\sum_{k=1}^{K}\lambda_k e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}\right)^2}\sum_{k=1}^{K}\lambda_k e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}\frac{(x_i-\mu_k)}{\sigma^2}
$$

(1.39)

$$
\begin{aligned}
A_{ik} &= \alpha_{ik}\frac{(x_i-\mu_k)}{\sigma^2} - \alpha_{ik}\frac{\sum_{k=1}^{K}\lambda_k e^{\left(-\frac{1}{2}\bullet\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}\frac{(x_i-\mu_k)}{\sigma^2}}{\sum_{k=1}^{K}\lambda_k e^{\left(-\frac{1}{2}\bullet\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}} \\
&= \alpha_{ik}\frac{(x_i-\mu_k)}{\sigma^2} - \alpha_{ik}\sum_{k=1}^{K}\alpha_{ik}\frac{(x_i-\mu_k)}{\sigma^2}
\end{aligned}
\tag{1.40}
$$

A_{ik} can then be substituted in

$$
\frac{1}{\sigma^2}\left[\sum_{i=1}^{n} h_i \sum_{k=1}^{K} \frac{d\ell}{d\Delta}\alpha_{ik} x_i - \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \frac{d\ell}{d\Delta}\alpha_{ik}\mu_k\right] = \frac{1}{\sigma^2}\left[\sum_{i=1}^{n} h_i \sum_{k=1}^{K} A x_i - \sum_{i=1}^{n} h_i \sum_{k=1}^{K} A\mu_k + 1\right] \tag{1.41}
$$

and the Second derivative of the log likelihood with respect to delta becomes

$$
\begin{aligned}
\frac{d^2\ell}{d\Delta^2} &= \frac{1}{\sigma^2}\left[\sum_{i=1}^{n} h_i \sum_{k=1}^{K} A_{ik}\left(x_i-\mu_k\right)+1\right] \\
&= \frac{1}{\sigma^2}\sum_{i=1}^{n} h_i \sum_{k=1}^{K} A_{ik}\left(x_i-\mu_k\right)+\frac{1}{\sigma^2} \\
&= \frac{1}{\sigma^2}\sum_{i=1}^{n} h_i \sum_{k=1}^{K}\left(\alpha_{ik}\frac{\left(x_i-\mu_k\right)}{\sigma^2}-\alpha_{ik}\sum_{k=1}^{K}\alpha_{ik}\frac{\left(x_i-\mu_k\right)}{\sigma^2}\right)\left(x_i-\mu_k\right)+\frac{1}{\sigma^2} \\
&= \frac{1}{\sigma^2}\sum_{i=1}^{n} h_i \sum_{k=1}^{K}\left(\alpha_{ik}\frac{\left(x_i-\mu_k\right)^2}{\sigma^2}-\alpha_{ik}\left(x_i-\mu_k\right)\sum_{k=1}^{K}\alpha_{ik}\frac{\left(x_i-\mu_k\right)}{\sigma^2}\right)+\frac{1}{\sigma^2} \\
&= \frac{1}{\sigma^2}\left[\sum_{i=1}^{n} h_i \sum_{k=1}^{K}\alpha_{ik}\frac{\left(x_i-\mu_k\right)^2}{\sigma^2}-\sum_{i=1}^{n} h_i\left(\sum_{k=1}^{K}\alpha_{ik}\left(x_i-\mu_k\right)\right)\left(\sum_{k=1}^{K}\alpha_{ik}\frac{\left(x_i-\mu_k\right)}{\sigma^2}\right)\right]+\frac{1}{\sigma^2} \\
&= \frac{1}{\sigma^4}\left[\sum_{i=1}^{n} h_i \sum_{k=1}^{K}\alpha_{ik}\left(x_i-\mu_k\right)^2-\sum_{i=1}^{n} h_i\left(\sum_{k=1}^{K}\alpha_{ik}\left(x_i-\mu_k\right)\right)^2\right]+\frac{1}{\sigma^2}
\end{aligned}
\tag{1.42}
$$

Now, to obtain the expected value of the RHS, we use a well-known result of expectations;

$$
E\left[E\left[X \mid Y\right]\right] = E[\mathrm{X}] \ . \tag{1.43}
$$

Here, we apply that by conditioning on α_{ik}. Then it is simple to show, that the expected value of

$$
\frac{d^2\ell}{d\Delta^2} = \frac{1}{\sigma^2}. \tag{1.44}
$$

The derivative of the log likelihood with respect to Δ and σ^2 is

$$
\frac{d\ell}{d\Delta d\sigma^2} = \sum_{i=1}^{n} h_i \sum_{k=1}^{K}\frac{d\ell}{d\sigma^2}\alpha_{ik}\frac{\left(x_i-\mu_k\right)}{\sigma^2} = \sum_{i=1}^{n} h_i \sum_{k=1}^{K}\left(\frac{d\ell}{d\sigma^2}\alpha_{ik}\right)\frac{\left(x_i-\mu_k\right)}{\sigma^2}+\alpha_{ik}\frac{-\left(x_i-\mu_k\right)}{\sigma^4} \quad , \tag{1.45}
$$

where B_{ik} is the derivative of α_{ik} with respect to σ^2

$$B_{ik} = \frac{d\alpha_{ik}}{d\sigma^2} = \frac{\lambda_k e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}\frac{(x_i-\mu_k)^2}{2\sigma^4}}{\sum_{k=1}^{K}\lambda_k e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}} - \frac{\lambda_k e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}}{\left(\sum_{k=1}^{K}\lambda_k e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}\right)^2}\sum_{k=1}^{K}\lambda_k e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}\frac{(x_i-\mu_k)^2}{2\sigma^4}$$

$$= \alpha_{ik}\frac{(x_i-\mu_k)^2}{2\sigma^4} - \alpha_{ik}\frac{\sum_{k=1}^{K}\lambda_k e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}}{\sum_{k=1}^{K}\lambda_k e^{\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)}}\frac{(x_i-\mu_k)^2}{2\sigma^4}$$

$$= \alpha_{ik}\frac{(x_i-\mu_k)^2}{2\sigma^4} - \alpha_{ik}\sum_{k=1}^{K}\alpha_{ik}\frac{(x_i-\mu_k)^2}{2\sigma^4}$$

(1.46)

and the Second derivative of the log likelihood with respect to Delta and Sigma becomes

$$\frac{d\ell}{d\Delta d\sigma^2} = \sum_{i=1}^{n} h_i \sum_{k=1}^{K}\left(\frac{d\ell}{d\sigma^2}\alpha_{ik}\right)\frac{(x_i-\mu_k)}{\sigma^2} + \alpha_{ik}\frac{-(x_i-\mu_k)}{\sigma^4}$$

$$= \sum_{i=1}^{n} h_i \sum_{k=1}^{K}(B_{ik})\frac{(x_i-\mu_k)}{\sigma^2} + \alpha_{ik}\frac{-(x_i-\mu_k)}{\sigma^4}$$

$$= \sum_{i=1}^{n} h_i \sum_{k=1}^{K}\left(\alpha_{ik}\frac{(x_i-\mu_k)^2}{2\sigma^4} - \alpha_{ik}\sum_{k=1}^{K}\alpha_{ik}\frac{(x_i-\mu_k)^2}{2\sigma^4}\right)\frac{(x_i-\mu_k)}{\sigma^2} + \alpha_{ik}\frac{-(x_i-\mu_k)}{\sigma^4}$$

$$= \left(\sum_{i=1}^{n} h_i \sum_{k=1}^{K}\alpha_{ik}\frac{(x_i-\mu_k)^2}{2\sigma^4} - \sum_{i=1}^{n} h_i \sum_{k=1}^{K}(\alpha_{ik})\sum_{k=1}^{K}\left(\alpha_{ik}\frac{(x_i-\mu_k)^2}{2\sigma^4}\right)\right)\frac{(x_i-\mu_k)}{\sigma^2} + \alpha_{ik}\frac{-(x_i-\mu_k)}{\sigma^4}$$

$$= \sum_{i=1}^{n} h_i \sum_{k=1}^{K}\alpha_{ik}\frac{(x_i-\mu_k)^2}{2\sigma^4} - \sum_{i=1}^{n} h_i \sum_{k=1}^{K}\alpha_{ik}\frac{(x_i-\mu_k)^2}{2\sigma^4} + \sum_{i=1}^{n} h_i \sum_{k=1}^{K}\alpha_{ik}\frac{-(x_i-\mu_k)^2}{\sigma^4}$$

$$= \sum_{i=1}^{n} h_i \sum_{k=1}^{K}\alpha_{ik}\frac{-(x_i-\mu_k)^2}{\sigma^4}$$

(1.47)

The expected value of $\frac{d\ell}{d\Delta d\sigma^2}$ is zero. This is similar to the MLE's of gaussian mean and variance, which are independent. We use the above as part of the information matrix for the calculation of the confidence interval.

The second derivative of the log likelihood with respect to sigma

$$\frac{d^2\ell}{d\left(\sigma^2\right)^2} = \sum_{i=1}^{n} h_i \sum_{k=1}^{K}\left[\frac{d\ell}{d\sigma^2}\alpha_{ik}\left[\frac{\left(x_i-\mu_k\right)^2}{2\sigma^4}-\frac{1}{2\sigma^2}\right]\right]$$

$$= \sum_{i=1}^{n} h_i \sum_{k=1}^{K}\left[\alpha_{ik}\left[\frac{-\left(x_i-\mu_k\right)^2}{2\sigma^6}-\frac{1}{2\sigma^4}\right]+\left[\frac{\left(x_i-\mu_k\right)^2}{2\sigma^4}-\frac{1}{2\sigma^2}\right]\bullet\left[\frac{d\ell}{d\sigma^2}\alpha_{ik}\right]\right]$$

$$= \sum_{i=1}^{n} h_i \sum_{k=1}^{K}\left[\alpha_{ik}\left[\frac{-\left(x_i-\mu_k\right)^2}{2\sigma^6}-\frac{1}{2\sigma^4}\right]+\left[\frac{\left(x_i-\mu_k\right)^2}{2\sigma^4}-\frac{1}{2\sigma^2}\right]\left[\alpha_{ik}\frac{\left(x_i-\mu_k\right)^2}{2\sigma^4}-\alpha_{ik}\sum_{k=1}^{K}\alpha_{ik}\frac{\left(x_i-\mu_k\right)^2}{2\sigma^4}\right]\right]$$

$$= \sum_{i=1}^{n} h_i \sum_{k=1}^{K}\left[\alpha_{ik}\left[\frac{-\left(x_i-\mu_k\right)^2}{2\sigma^6}-\frac{1}{2\sigma^4}\right]+\alpha_{ik}\left[\frac{\left(x_i-\mu_k\right)^2}{2\sigma^4}-\frac{1}{2\sigma^2}\right]\left[\frac{\left(x_i-\mu_k\right)^2}{2\sigma^4}-\sum_{k=1}^{K}\alpha_{ik}\frac{\left(x_i-\mu_k\right)^2}{2\sigma^4}\right]\right]$$

$$= \sum_{i=1}^{n} h_i \sum_{k=1}^{K}\alpha_{ik}\left[\left[\frac{-\left(x_i-\mu_k\right)^2}{2\sigma^6}-\frac{1}{2\sigma^4}\right]+\left[\frac{\left(x_i-\mu_k\right)^2}{2\sigma^4}-\frac{1}{2\sigma^2}\right]\left[\frac{\left(x_i-\mu_k\right)^2}{2\sigma^4}-\sum_{k=1}^{K}\alpha_{ik}\frac{\left(x_i-\mu_k\right)^2}{2\sigma^4}\right]\right]$$

$$= \sum_{i=1}^{n} h_i \sum_{k=1}^{K}\alpha_{ik}\left[\left[\frac{-\left(x_i-\mu_k\right)^2}{2\sigma^4}-\frac{1}{2\sigma^2}\right]\left[\frac{-1}{2\sigma^2}+\frac{\sum_{i=1}^{n}\left(x_i-\mu_k\right)^2}{2\sigma^4}-\sum_{k=1}^{K}\alpha_{ik}\frac{\left(x_i-\mu_k\right)^2}{2\sigma^4}\right]\right]$$

(1.48)

Using the same result of expectations conditioning on α_{ik} it is simple to show, that the expected value of

$$\frac{d^2\ell}{d\left(\sigma^2\right)^2} = \frac{1}{2\sigma^4} \tag{1.49}$$

With the original derivations of the log likelihood, the information matrix become

$$\begin{bmatrix} \frac{1}{\sigma^4}\left[\sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik}\left(x_i-\mu_k\right)^2 - \sum_{i=1}^{n} h_i \left(\sum_{k=1}^{K} \alpha_{ik}\left(x_i-\mu_k\right)\right)^2\right] + \frac{1}{\sigma^2} & \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} \frac{-\left(x_i-\mu_k\right)^2}{\sigma^4} \\ \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} \frac{-\left(x_i-\mu_k\right)^2}{\sigma^4} & \sum_{i=1}^{n} h_i \sum_{k=1}^{K} \alpha_{ik} \left[\left[\frac{-\left(x_i-\mu_k\right)^2}{2\sigma^4} - \frac{1}{2\sigma^2}\right] \left[\frac{-1}{2\sigma^2} + \frac{\sum_{i=1}^{n}\left(x_i-\mu_k\right)^2}{2\sigma^4} - \sum_{k=1}^{K} \alpha_{ik} \frac{\left(x_i-\mu_k\right)^2}{2\sigma^4}\right]\right] \end{bmatrix}$$

(1.50)

Using the Rao's Score test, the inverse of the information matrix gives the variance of $\hat{\Delta}$ and $\hat{\sigma}^2$. This can be calculated easily

$$\begin{bmatrix} \frac{d^2\ell}{d\Delta^2} & \frac{d\ell}{d\Delta d\sigma^2} \\ \frac{d\ell}{d\Delta d\sigma^2} & \frac{d^2\ell}{d\left(\sigma^2\right)^2} \end{bmatrix}^{-1} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \frac{1}{(\mathrm{AD}-\mathrm{CB})} \begin{bmatrix} D & -C \\ -B & A \end{bmatrix} = \begin{bmatrix} Var(\hat{\Delta}) & Var(\hat{\Delta},\hat{\sigma}^2) \\ Var(\hat{\Delta},\hat{\sigma}^2) & Var(\hat{\sigma}^2) \end{bmatrix}.$$

(1.51)

The variance is used for the calculation of the confidence interval of the estimated model parameters. While not intractable, this calculation can be cumbersome and there can be simplification of eqn. 1.48 by taking conditional expectations based on α_{ik} as shown in eqn. 1.49. First, we must review several features of our model. First since all means in the mixture are defined by μ_0 and Δ the variance of Δ is the same for each component of the mixture. Second since we assume constant variance across all components of the mixture, the variance of the variance is also the same. Since we can relate μ_k and Δ by a series of constants (eqn. 1.6), the variance of μ_k (the mean of our Gaussian component k) is the variance of Δ. The variance of the mean is σ^2.

Normally the variance of a Gaussian is considered to be distributed Chi-squared with n-1 degrees of freedom; this means that the variance can be calculated as follows

$$Var(\sigma^2) = \frac{2\sigma^4}{n-1} \tag{1.52}$$

But since we are estimating an MLE for σ^2, *n-1* becomes n. The spectra is treaded as a histogram of frequency data and has been normalized, so *n* can be considered to be $\sum_{i=1}^{n} h_i$, or one. Hence our variance co variance matrix becomes;

$$\begin{bmatrix} Var(\hat{\Delta}) & Var(\hat{\Delta},\hat{\sigma}^2) \\ Var(\hat{\Delta},\hat{\sigma}^2) & Var(\hat{\sigma}^2) \end{bmatrix} = \begin{bmatrix} \hat{\sigma}^2 & 0 \\ 0 & 2\hat{\sigma}^4 \end{bmatrix} \tag{1.53}$$

And our confidence intervals are

$$95\%CI\Delta = \hat{\Delta} \pm 1.96\sqrt{Var(\hat{\Delta})} = \hat{\Delta} \pm 1.96\hat{\sigma} \tag{1.54}$$

$$95\%CI\sigma^2 = \hat{\sigma}^2 \pm 1.96\sqrt{Var(\hat{\sigma}^2)} = \hat{\sigma}^2 \pm 2.772\hat{\sigma}^2 \tag{1.55}$$

Where our estimators are equations: 1.16, 1.23, and 1.36.

4.3.3 Summary of the EM algorithm

To begin to formulate an EM algorithm based one the GMM you must first suppose the baseline shift has been corrected. It can be shown that the probability that the *i*th observation will be in the *k*th component Gaussian is equation 1.16. Since in MALDI-TOF we observe the frequencies at each m/z (instead of the actual x_i) we introduce the notation h_i representing the frequency of each x_i for $i = 1, 2, \ldots, N$. Then, it can be shown that the MLE's of Δ and σ^2, when α_{ik} are known are shown in equations 1.23 and 1.36. The MLE's of Δ and σ^2 when α_{ik} are unknown do not have an explicit form, which is the case in Gaussian mixture data. An EM algorithm could be used here, in which an indicator function of whether an observation is from the *k*th Gaussian density could be used as the missing component of the complete data. Then the following expectation maximization steps are then used to obtain the MLE's. In some situations where there is a

baseline shift, data preprocessing is essential. For instance, The Data is modified by removing a baseline estimation based on the slope between the first and last points of the spectra. In removing this baseline one may encounter over correction and negative values may result. Since, this violates the distribution properties one may have to correct them by taking absolute values. Also, for simplification of calculations the data is normalized by $\sum_{i=1}^{n} h_i$. This is equivalent to dividing each observation by the total number of observations.

Step 1. Set the initial values for Δ and σ^2 say Δ_0 and σ_0^2.

Step 2. E-Step: In the *r*th iteration (starting with *r* = 1) compute the expected value of the indicator function I given the observed data, which is $\alpha_{ik,r}$ using Δ_{r-1} and σ_{r-1}^2 as in equation 1.16. That is, for a given Δ_{r-1} and $\sigma_{r-1}^2, \alpha_{ik,r}$ is computed under the influence of λ_k.

Step 3. M-Step: Using $\alpha_{ik,r}$ obtained in step 2, obtain the Δ_r and σ_r^2 from formulas shown in equations 1.23 and 1.36. With the exception of $E(\lambda_k)$ being replaced with $E\left(\sum_{i=1}^{n} h_i \alpha_{ik}\right) = E\left(\hat{\lambda}_k\right)$.

Step 4. Repeat steps 2 and 3 until convergence within the allowed tolerance. Once the EM algorithm converges, a second iterative computational algorithm is used to calculate the slope-intercept form of the baseline.

Often in proteomics estimating the peak Area (under the curve) is of interest. Once the MLE's $\hat{\Delta}$ and $\hat{\sigma}^2$ have been obtained the peak area between m/z = b and m/z = a could be estimated using

$$Area = F_K(b) - F_K(a). \tag{1.56}$$

Here,

$$F_K(x) = \sum_{k=1}^{K} \lambda_k \Phi_k(x; \mu_k, \sigma_k^2), \tag{1.57}$$

where $\Phi_k(x; \mu_k, \sigma_k^2)$ is the cdf of a Gaussian. The cdfs could be obtained from most software.

4.4. Application of the EM algorithm to MS spectra

Under the GMM analyses of MS data have been performed previously [5] of MS data of Angiotensin II (Ang II) and Angiotensin-2-10 (Ang-2-10), where the criterion for estimation of Δ and σ^2 was the highest R^2, a measure of goodness-of-fit. This is done using the best fit results from a grid search of the Δ and σ^2 parameters as the closest current true estimates of the parameters. Individual measures from the original grid search and the MLE are shown for the individual spectrum with the 95% confidence intervals for each MLE (Table 1). This is check that the original gird estimation is found within the confidence intervals. The σ^2 confidence intervals will be bounded by zero since there cannot be a σ^2 of zero or less. An analysis of previously published simulation data [7] using the MLE and confidence intervals will show the function of the MLE over a range of Δ, σ^2, and peak areas (Table 2).

4.5. Results and Discussion

The calculation of the MLE under the constraints of the GMM have given estimates of the Δ and σ^2 parameters in real data that contain the parameters estimated in the grid search (Table 1). This shows that the MLE has promise as a possible estimator for the parameters of a GMM for peak quantification. There are several factors that make the current attempt at producing MLE's of the GMM parameters sub-optimal. An

analysis of simulated data (Table 2) shows that the mean of the estimators $\hat{\Delta}$ and $\hat{\sigma}^2$, $\bar{\hat{\Delta}}$ and $\bar{\hat{\sigma}}^2$ do estimate the true Δ and σ^2 of the simulations under most conditions. The $\bar{\hat{\Delta}}$ is accurate under all conditions. The $\bar{\hat{\sigma}}^2$ becomes easier to detect as the σ^2 increases and as the peak area increases. It should be noted that in all simulations the confidence interval contains the true Δ and σ^2 for each set of simulations regardless of simulation parameters.

There are multiple assumptions that come with using a GMM to attempt to quantify peaks in MS data. A Gaussian mixture does not have a component that allows for the noise seen in MS data. This can be added to the model [5] but it is only an estimate that acts as a linear regression of the noise in a slope intercept form. This estimate treats the noise as an additive effect which the signal of interest sits on top of. The noise estimator is added outside the description of the Gaussian mixture and this separation is not possible in MLE estimation.

In the instance of a single Gaussian distribution, all predicted probabilities from the pdf are positive and only reach zero at infinity. The intensities in the MS data are treated as a frequency distribution that mirrors the idealized Gaussian at each peak in GMM. This acts to increase the signal-to-noise ratio. Preprocessing steps must be applied to the data to fit the analysis. In this work, preprocessing was limited to a three step process. First, calculation of the slope-intercept line describing the noise between the first and last points in the spectrum range on the MZ axis being examined. Second, this estimation of the noise was subtracted from the data and the absolute value of the resulting intensities was used to serve as the estimated signal. This is done to remove any negative values. Finally the data are normalized by the sum of the estimated intensities. This aids in the calculation of the MLE.

This preprocessing removes the majority of gross noise and negative values from

the fine noise but this still skews the calculation of the MLE for both parameters.

Table 1: Comparison of individual spectra quantification parameter estimations						
	Grid Search Estimator		Maximum Likelihood Estimator			
Spectrum	Δ	σ^2	Δ	95% CI	σ^2	Bound 95 % CI
1	0.0883	0.0116	0.1157	[0.0610,0.1703]	0.0279	(0,0.1052]
2	0.064	0.0063	0.0768	[0.0419,0.1116]	0.0178	(0,0.0670]
3	0.0832	0.0192	0.1151	[0.0557,0.1745]	0.0303	(0,0.1143]
4	0.1228	0.0095	0.1952	[0.0972,0.2933]	0.0500	(0,0.1887]
5	0.1069	0.0109	0.1207	[0.0808,0.1607]	0.0204	(0,0.0769]
6	0.0813	0.0062	0.0816	[0.0496,0.1137]	0.0163	(0,0.0616]
7	0.1307	0.0565	0.1544	[0.0374,0.2713]	0.0597	(0,0.2251]
8	0.3094	0.1011	0.3581	[0.1642,0.5520]	0.0989	(0,0.3732]
9	0.1151	0.0327	0.1630	[0.0598,0.2663]	0.0527	(0,0.1987]
10	0.3795	0.0216	0.3790	[0.3125,0.4456]	0.0339	(0,0.1280]
11	0.3103	0.0135	0.3217	[0.2596,0.3838]	0.0317	(0,0.1196]
12	0.2281	0.0129	0.2407	[0.1944,0.2870]	0.0236	(0,0.0891]
13	0.2439	0.0233	0.2510	[0.1836,0.3184]	0.0344	(0,0.1298]
14	0.2795	0.0180	0.2878	[0.2260,0.3495]	0.0315	(0,0.1189]
15	0.1169	0.0161	0.1364	[0.0841,0.1888]	0.0267	(0,0.1007]
16	0.1726	0.0154	0.1862	[0.1307,0.2416]	0.0283	(0,0.1067]
17	0.1793	0.0168	0.2013	[0.1450,0.2575]	0.0287	(0,0.1082]
18	0.1885	0.0712	0.2095	[0.0699,0.3492]	0.0713	(0,0.2688]
19	0.2289	0.0913	0.2548	[0.0927,0.4169]	0.0827	(0,0.3119]
20	0.2461	0.0502	0.2399	[0.1338,0.3459]	0.0541	(0,0.2041]
21	0.1963	0.0359	0.1949	[0.0914,0.2983]	0.0528	(0,0.1990]
22	0.0399	0.0234	0.0421	[-0.0150,0.0992]	0.0291	(0,0.1099]
23	0.2259	0.0613	0.2285	[0.0911,0.3658]	0.0701	(0,0.2643]
24	0.4082	0.1041	0.3822	[0.1692,0.5953]	0.1087	(0,0.4100]
25	0.0912	0.0250	0.0943	[0.0083,0.1803]	0.0439	(0,0.1655]
26	0.1306	0.0171	0.1495	[0.0858,0.2131]	0.0325	(0,0.1225]

Table 1: This table contains the individual estimations from the grid search of parameters and from the maximum likelihood estimators. Note that all estimations from the grid parameter search done previously are found within the confidence intervals for both Δ and σ^2. The σ^2 confidence intervals are bound by zero since σ or σ^2 cannot be zero or negative in a Gaussian distribution. All σ^2 confidence intervals were found to cross into negative numbers.

Table 2:MLE test on Simulated Data				
	Δ	**-0.1**		
	Area	**50**	**100**	**250**
	0.025	-0.098 [-0.345, 0.149 0.126 (0,0.060]	-0.099[-0.289,0.091] 0.097 (0,0.036]	-0.099[0.247,0.048] 0.006 (0,0.021]
σ^2	**0.01**	-0.098[-0.387,0.191] 0.022 (0,0.082]	-0.099[-0.346,0.147] 0.016(00.06]	-0.1[-0.317,0.118] 0.012(0,0.046]
	0.04	-0.098[-0.517,0.322] 0.046(0,0.173]	-0.098[-0.503,0.308] 0.046(0,0.173]	-0.1[-0.496,0.297] 0.041(0,0.154]
	Δ	**0**		
	Area	**50**	**100**	**250**
	0.025	0[-0.249,0.249] 0.016 (0,0.061	0[-0.1980.198] 0.010 (0,0.039]	0.002[-0.161,0.164] 0.007(0,0.026]
σ^2	**0.01**	0[-0.287,0.287] 0.021 (0,0.081]	-0.001[-0.245,0.244] 0.016 (0,0.059]	0[-0.219,0.219] 0.013 (0,0.047]
	0.04	0[-0.423,0.424] 0.047(0,0.176]	0[-0.406,0.406] 0.043(0,0.162]	0[-0.397,0.396] 0.041(0,0.154]
	Δ	**0.1**		
	Area	**50**	**100**	**250**
	0.025	0.097[-0.161,0.355] 0.017(0,0.065]	0.098[-0.101,0.297] 0.010(0,0.039]	0.099[-0.048,0.247] 0.006(0,0.021]
σ^2	**0.01**	0.097[-0.195,0.39] 0.022 (0,0.084]	0.098[-0.154,0.35] 0.016(0,0.062]	0.099[-0.122,0.32] 0.013(0,0.048]
	0.04	0.097[-0.332,0.526] 0.048 (0,0.181]	0.099[-0.308,0.506] 0.043(0,0.163]	0.099[-0.298,0.497] 0.041(0,0.155]

Table 2: Here the MLE is used to estimate $\hat{\Delta}$ and $\hat{\sigma}^2$ for 1000 simulations under a range of conditions. The mean $\hat{\Delta}$ is presented in the top of cell and the mean $\hat{\sigma}^2$ in the bottom for each set of simulation parameters. The mean for these estimators are presented over the 1000 simulations with the confidence interval around that mean. Note that as the peak width increases, the width estimation becomes more accurate. This is also apparent with the increase in peak area. The estimates of $\hat{\Delta}$ all converge on the true Δ of the simulation. The confidence intervals contain the true Δ and σ^2 for each set of simulations.

4.6. Conclusions

The calculation of an MLE to describe the parameters of a GMM for peak quantification is possible but there are several points that need to be addressed. The calculation of a confidence interval is a good measure of the accuracy of the estimator when combined with knowledge of the tolerances of the MS instrument. This gives an interesting way to measure the accuracy of the estimate since we know specific bounds on the ranges of the parameters from the tolerances of the MS instruments being used. Since the resolution and $\hat{\sigma}^2$ are inexorably linked and bounded by zero we can make value statements on $\hat{\sigma}^2$ based on the confidence interval. The same can be said of $\hat{\Delta}$ since a given MS instrument has allowable tolerances on $\hat{\Delta}$, as error from the application of a standard curve to the data. The confidence intervals allow a user with sufficient knowledge of their MS instruments to judge the MLE's. This alone makes the calculation of MLE's and theirs associated confidence intervals valuable to the analysis of MS data. These confidence intervals also have the interesting benefit of being able to inform the grid search. Every calculated confidence interval contained the best estimator from the grid search. This means that a grid search preformed within the MLE confidence intervals would find a fit that is close if not closer to the true best fitting GMM for a given spectrum.

Future analysis of different data preprocessing methods will improve the estimation of the MLE's. The use of a simple slope-intercept calculation and application of the absolute value is a crude method for dealing with the non-Gaussian behavior seen in the data. This shows in the range of error seen in the MLE's. Replacing the abs function with a more finessed approach such as time series pre-whitening, sliding window mean or other smoothing procedure will improve baseline estimation and aid in its removal from the data. This needs to be balanced with the need to keep the signal-to-

noise ratio as high as possible by avoiding modification to the isotopic cluster of peaks. Methods of separating noise in the tails of the spectra and between the peeks from the peaks need to be applied before smoothing.

The estimation of the MLE's has been done only is single isotopic clusters. Previous use of GMM has included analysis and quantification of convolved isotopic clusters of peptides in MS data. This necessitates looking at a wider MZ range in the spectra and my need to include allowing the σ^2 to be variable. In MALDI-TOF data this is an easy calculation to account for as the peaks increase in width a linear fashion associated with increase in MZ. In other forms of MS this is not as straight forward a relationship. Since the convolved peaks have separate μ_0 and Δ parameters along with a possibly changing σ^2, the dimensionality of the problem increases and MLE's need to be reexamined.

The use of MLE's in the application of GMM to quantify peaks in MALDI-TOF MS data for sample peptide quantification is feasible with more work. It is a valuable tool for researcher in the field of proteomics and will aid in speeding up of the workflow of data processing for systems biology research involving these problems.

5. Discussion and Conclusion

The Gaussian mixture model is a novel approach for the estimation of peptide peak area. This method acts to quantify individual peptides by estimating the total peak area for a given peptide or set of convolved peptides. This is a novel approach to quantify a peptide from MALDI-TOF MS data by estimating the area under the peaks (under the isotopic cluster) by treating the data as the probability density function (pdf) of a Gaussian mixture. This mixture has the unique properties of:

- The mixture weights are assumed to be equivalent to the isotopic distribution of the peptide.
- The means of the individual component Gaussians are defined by the mean of the first Gaussian, based on peptide monoisotopic mass
- All of the Gaussian variances are treated as being equal (the differences are known and can be accounted for easily in Malid-tof data).
- The MS data can be treated as a frequency histogram that is mimicked by the GMM pdf

These peak quantifications are then used to estimate the amount of a peptide in a given sample by comparing the native peptide estimated area to the estimated area of a SIS copy of that peptide in the same spectra. This use of an internal standard allows for the user to account for many of the reproducibility issues seen in the use of Madli-tof ms. In this work the GMM method for peak estimation has been compared to past methods of peak quantification and been shown to be at least equivalent in peptide estimation if not better. The limitations of GMM have been explored in simulation and its dependence on signal-to-noise ratio in analyzed spectra and spectra resolution has been highlighted. Finally, a maximum likelihood estimator implemented in an expectation maximization algorithm has been used to estimate both model input parameters (Δ, σ) with confidence intervals for a measure of uncertainty.

GMM has been shown to accurately estimate both single isotopic clusters and convolved sets of isotopic clusters, giving area estimations for individual peptides and

peptides that overlap. These area estimates can be used to quantify the amount of peptide in a sample using a SIS internal standard. Using the GMM algorithm we can estimate noise by using a baseline lift or error described as an n-polynomial. This work has used a slope-intercept form but it can easily be expanded to any n-polynomial desired. This moves away from the black box or *ad hoc* approaches used to account for error in previous work. GMM also allows for a measure of uncertainty in the parameter estimations by the calculation of a confidence interval round the Δ and σ estimations. These confidence estimations can be combined with knowledge of the MS equipment (i.e. the maximum allowable limit on error due to the standard curve) to help gauge the overall quality of the estimation. All of these features have been automated such that multiple spectra can be analyzed for multiple peptides with minimal input from a user.

5.1 Weaknesses

While GMM is a robust method for peak quantification it does have limitations that require future work or slightly different methods to overcome. The difficulties involved with proper baseline estimation, the required prior knowledge of the peptides in the sample, and the dependence on signal to noise ratios coupled to spectra resolution limit what can be accomplished using GMM. The baseline as described in slope-intercept form can also be thought of as a trapezoidal mixture that must be accounted for in the mixture distribution. This needs to be included in future research as part of the MLE and future modeling attempts need to reflect this.

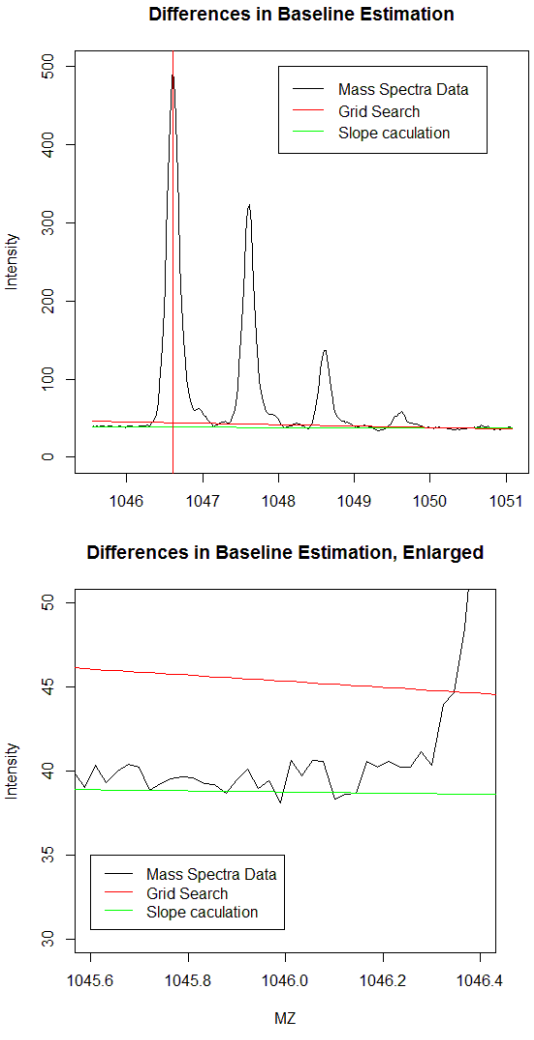
GMM takes advantage of prior knowledge to fill in several blanks in the Gaussian mixture parameters, but the peptides must be suspected to be present before GMM becomes useful. GMM is a quantification tool for known or suspected peptides in a sample. It can be used to search for peptides that are thought to be present in a spectrum but is incapable of giving a definitive answer as to a peak's identity. Analysis of a

spectrum that turns up a peak with a large area and a good model fit does not necessitate that it is that specific peptide. Further analysis of that peak using MS/MS or other methods is still required for identification. GMM is only a targeted search tool, it is not designed for searching an entire spectrum for any possible peptides, only for looking specific peptides as described by the user.

As stated previously, GMM is dependent on signal-to-noise ratio in the spectrum as well as the spectrum resolution. Since MALDI-TOF has a fixed resolution, meaning a slow, steady change in peak width as mass increases, this problem is somewhat of set. However, GMM produces poor results with sub optimal data, and is not very robust outside of these conditions. Since GMM is based on the underlying assumptions inherent in a Gaussian distribution, these assumptions become more violated as the data becomes less Gaussian in appearance. This occurs as the peaks become wider (resolution decreases) and the signal to noise ratio decreases. This becomes even more apparent in the maximum likelihood estimation of the parameters. Finally, it should be noted that GMM was designed with MALDI-TOF data in mind and that other forms of MS data are accurately analyzable do to the different behavior patterns in the data. One example of this is that the peak width is not constant in all forms of MS, violating one of our chief assumptions of GMM as it is currently implemented. It must be remembered that the MALDI data is ultimately not Gaussian. The data contains at least two types of noise. The first is a fine noise that can be seen to mimic an auto regressive model and the second a gross lift of the signal. In this work the noise is treated as such but may be far more complex than this.

5.2 Future Expansion

Future work with GMM has several different avenues for discovery. The measure of the model fit needs to be expanded and new methods explored for finding a better way to measure goodness-of-fit of the GMM to the data and expand the confidence interval estimation to all model parameters. The current methods of GMM implementation use the coefficient of determination (R^2) as the way to gauge model fit. As shown in the introduction this produces a convincing curve across the Δ and σ parameter space during a grid search, showing a local maximum of fit near the true mean with smaller peaks in fit at regular intervals. This does not always give the most apparent answer. While the model may fit the best to the data as a whole it does not produce a convincing description. As seen in the figure, the grid search has produced a very close estimate of the true mean (red vertical line) of the monoisotopic peak but estimates the baseline (red horizontal line) with a larger slope than what is calculated from a simple examination of the first and last points in the spectra (green horizontal line). While this is not a large difference we still discount part of the peak area with the red line that we do not using the green. Different methods of baseline estimation need to be explored and the role of the baseline in peak area estimation needs to be accounted for in estimation error. There is also a need for confidence intervals for the baseline estimators and the isotopic



cluster areas. Confidence intervals around the area estimation will allow for a final calculation of a confidence interval around the final peak quantification. These relationships and descriptive mathematics have yet to be examined.

Another avenue of baseline estimation requires expanding on the notion of the contribution of a trapezoidal distribution, creating an extra layer of mixture in the overall distribution being modeled. Currently we view the Gaussian mixture as

$$f_k(x_i;\sigma^2,\Delta)=\sum_{k=1}^{K}\lambda_k f_k(x_i;\sigma^2,\mu_k)\ . \tag{2.1}$$

But estimating the trapezoidal distribution requires a second mixture proportion that needs to be estimated, such that

$$\gamma g(\alpha,\beta,x)+(1-\gamma) f_k(\lambda_k,x,\Delta,\sigma)\ . \tag{2.2}$$

Where $g(\alpha,\beta,x)$ is the trapezoidal distribution and γ is the mixture between the trapezoidal distribution and the Gaussian mixture where

$$g(\alpha,\beta,x)=c(\alpha+\beta x), \tag{2.3}$$

where $c\in\int c(\alpha+\beta x)=1$. Besides the additional complexity and new parameter estimation required to inform model construction, there is loss of flexibility that is found in the current GMM in terms of the n-polynomial quadratic baseline description. Using a trapezoidal baseline limits our estimate since the ability to describe a point of inflection is lost.

Alternative methods for dealing with noise in the data need to be explored. This includes model modification before fitting and possible data preprocessing such as smoothing, especially in the case of the MLE approach. Since the MLE treats the MS data as being normal for the purposes of estimation smoothing of the noise and baseline estimation and removal become much more important. This connects with alternate

forms of baseline estimation as mentioned previously.

Finally alternative distributions can be explored and other data types examined. The basis for the GMM approach can be applied to other forms of data using other mixtures of distributions. The GMM method has application beyond MALDI-TOF MS data analysis. Any data that has a Gaussian appearance can be quantified using a form of this method. GMM can be theoretically applied florescence data, high performance liquid chromatography analysis using absorption analysis, or any other data that uses flow rate or time of flight that might contain data that appears approximately normal in its distribution. This can be expanded to using other distributions instead of the Gaussian to build the mixture. One example is the use of a Chi-squared distribution to describe data that may pear close to normal but has a decreasing slope as mass increases (going down stream on the spectrum). This is seen in some MS data and is possibly due to the instrument needing tuning. This presents unique difficulties such as an increase in the dimensionality of the parameter space being searched and the need to reassess model fitness but is feasible if computationally intensive. In the above example the *k* parameter will need to be considered but also a gross adjustment of the whole pdf to move it across the spectra will be needed, but since the apex of the pdf does not occur at the mean fitting the Chi-squared distribution requires more thought. Even in this instance the basic principles of GMM could be applied to obtain a peak area estimate to quantify the peak (or associated peptide).

5.3 R Package Implementation

In implementing the GMM for publication and use by others in an R package, several methods can be considered for a functional piece of software. Publishing the grid search method used discussed in manuscript one is easiest but requires the user to not only input the peptides being searched for but also to define the parameter grid in both

range and step size. This may not be ideal if the person scanning the spectrum is not familiar with the tolerances of the instrument used; hence they may lack the experience to define a proper parameter space. The grid search does not include a confidence interval. The grid method is also more computationally expensive (slower) than the MLE even after the improvements presented in manuscript two. While the quadrant ascent method could be replaced with another method for crossing the parameter space, such as a step-wise or genetic algorithm, the MLE should both be faster and more accurate in its estimation. The problem of not getting the MLE to converge due to the noise in the spectrum still presents some problems. The MLE in its current form does not produce a better estimator of Δ and σ when compared to the grid search.

This can be used to our advantage however with a combination of the two methods to remove the need for parameter input. The algorithm can be altered to look at a raw estimation of the MLE for Δ and σ, the confidence intervals can then be used to define the boundaries of the parameter space and the step size in the MZ range can be used to define the step size of the grid search. Once the best fir in the grid search is found a second confidence interval can be reported using the data, modified by the new baseline estimation. This combination of methods, while taking more time for estimation than the MLE provides the confidence intervals with the accuracy of the grid search.

This is a possible step for the publication of the R package.

5.4 Closing remarks

Ultimately the purpose of GMM is to estimate isotopic cluster (peptide) area in MALDI-TOF spectra, while allowing the user to pull apart convolved peaks and get separate estimates for each isotopic cluster. This has been accomplished and has shown to be useful in quantifying peptides from MALDI-TOF MS. There is potential in GMM for further study and application. While there are limitations on what can be

accomplished with GMM in its current form, there is ample room for future growth and change in the method.

References

Chapter 1: Introduction

1. Bantscheff M, Schirle M, Sweetman G, RickJ, Kuster B (2007) Quantitative mass spectrometry in proteomics: a critical review. Analytical and Bioanalytical Chemistry 389:1017-1031. doi:10.1007/s00216-007-1486-6

2. Bantscheff M, Lemeer S, Savitski MM, Kuster B (2012) Quantitative mass spectrometry in proteomics: critical review update from 2007 to the present. Analytical and Bioanalytical Chemistry 404:939-965 doi:10.1007/s00216-012-6203-4

3. Gerber SA, Rush J, Stemman O, Kirschner MW, Gygi SP (2003) Absolute quantification of proteins and phosphoproteins from cell lysates by tandem MS. Proceedings of the National Academy of Natural Sciences of the United States of Americs 100:6940-6945 doi:10.1073/pnas.0832254100

4. Gygi SP1, Rist B, Gerber SA, Turecek F, Gelb MH, Aebersold R (1999) Quantitative analysis of complex protein mixtures using isotope-coded affinity tags. 17:994-999 doi:10.1038/13690

5. Beck M, Schmidt A, Malmstroem J, Claassem M, Ori A, Szymborska A, Herzog F, Rinner O, Ellenberg J, Aebersold R (2011) The quantitative proteome of a human cell line. Molecular Systems Biology 7:549 doi:10.1038/msb.2011.82

6. Nagaraj N, Wisniewski JR, Geiger T, Cox J, Kircher M, Kelso J, Paabo S, Mann M (2011) Deep proteome and transcriptome mapping of a human cancer cell line. Molecular Systems Biology 7:548 doi:10.1038/msb.2011.81

7. Ong S-E, Blagoev B, Kratchmarova I, Kristensen DB, Steen H, Pandey A, Mann M (2002) Stable Isotope Labeling by Amino Acids in Cell Culture, SILAC, as a Simple and Accurate Approach to Expression Proteomics. Molecular and Cellular Proteomics 1:376-386 doi:10.1074/mcp.M200025-MCP200

8. Yao X, Freas A, Ramirez J, Demirev PA, Fenselau C (2001) Proteolytic 18O labeling for comparative proteomics: model studies with two serotypes of adenovirus. Analytical Chemistry 73:2836-2842 doi:10.1021/ac001404c

9. Ross PL, Huang YN, Marchese JN, Williamson B, Parker K, Hattan S, Khainovski N, Pillai S, Dey S, Daniels S, Purkayastha S, Juhasz P, Martin S, Bartlet-Jones M, He F, Jacobson A, Pappin DJ (2004) Multiplexed Protein Quantitation in Saccharomyces cerevisiae Using Amine-reactive Isobaric Tagging Reagents.Mollecular and Cellular Proteomics 3:1154-1169 doi:10.1074/mcp.M400129-MCP200

10. Thompson A, Schäfer J, Kuhn K, Kienle S, Schwarz J, Schmidt G, Neumann T, Johnstone R, Mohammed AK, Hamon C (2003) Tandem mass tags: a novel quantification strategy for comparative analysis of complex protein mixtures by MS/MS. Analytical Chemistry 75:1895-1904 doi:10.1021/ac0262560

11. Kirkpatrick DS1, Gerber SA, Gygi SP (2005) The absolute quantification strategy: a general procedure for the quantification of proteins and post-translational modifications.

Methods 35:265-273 doi:10.1016/j.ymeth.2004.08.018

12. Schwacke JH, Spainhour JCG, Ierardi JL, Chaves JM, Arthur JM, Janech MG, Velez JC (2013) Network modeling reveals steps in angiotensin peptide processing. Hypertension 61:690-700doi:10.1161/HYPERTENSIONAHA.111.00318

13. Velez JC, Ryan JK, Harbeson CE, Bland AM, Budisavljevic MN, Arthur JM, Fitzgibbon WR, Raymond JR, Janech MG (2009) Angiotensin I Is Largely Converted to Angiotensin (1-7) and Angiotensin (2-10) by Isolated Rat Glomeruli. Hypertension 53:790-797 doi:10.1161/HYPERTENSIONAHA.109.128819

14. Dempster AP, Laird NM, Rubin DM (1977) Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B 39:1-31

15. Hoffmann, E, Vincent Stroobant, V (2007) Mass Spectrometry: Principles and Applications. Wiley-Interscience pg.250

16. R Core Team (2013) R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.

17. Francis JGF (1961) The QR Transformation A Unitary Analogue to the LR Transformation—Part 1. The Computer Journal 4:265-271. doi: 10.1093/comjnl/4.3.265

18. Francis JGF (1962) The QR Transformation—Part 2. The Computer Journal 4:332-345 doi: 10.1093/comjnl/4.4.332

19. Plechawska M, Polańska J (2009) Simulation of the usage of Gaussian mixture models for the purpose of modelling virtual mass spectrometry data. Stud Health Technol Inform 150:804-8 doi: 10.3233/978-1-60750-044-5-804

20. Coombes KR, Koomen JM, Baggerly KA, Morris JS, Kobayashi R (2005) Understanding the characteristics of mass spectrometry data through the use of simulation. Cancer Informatics 1:41-52

21. Morris JS, Coombes KR, Koomen JM, Baggerly KA, Kobayashi R (2005) Feature extraction and quantification for mass spectrometry in biomedical applications using the mean spectrum. Bioinformatics 21:1764-1775 doi: 10.1093/bioinformatics/bti254

22. Barkauskas DA, Rocke DM (2010) A general-purpose baseline estimation algorithm for spectroscopic data. Analytica Chimica Acta 657:191–197 doi:10.1016/j.aca.2009.10.043.

23. Russell DH, Edmondson RD (1997) High-resolution Mass Spectrometry and Accurate Mass Measurements with Emphasis on the Characterization of Peptides and Proteins by Matrix-assisted Laser Desorption/Ionization Time-of-Ñight Mass Spectrometry. Journal of Mass Spectrometry 32:263-276 DOI: 10.1002/(SICI)1096-9888(199703)32:3

Chapter 2: Manuscript One: Method Comparison

1. Hoffmann, E, Vincent Stroobant, V (2007) Mass Spectrometry: Principles and Applications. Wiley-Interscience

2. Schwacke JH, Spainhour JCG, Ierardi JL, Chaves JM, Arthur JM, Janech MG, Velez JC (2013) Network modeling reveals steps in angiotensin peptide processing. Hypertension 61:690-700 doi:10.1161/HYPERTENSIONAHA.111.00318

3. Velez JC, Ryan JK, Harbeson CE, Bland AM, Budisavljevic MN, Arthur JM, Fitzgibbon WR, Raymond JR, Janech MG (2009) Angiotensin I Is Largely Converted to Angiotensin (1-7) and Angiotensin (2-10) by Isolated Rat Glomeruli. Hypertension 53:790-797 doi:10.1161/HYPERTENSIONAHA.109.128819

4. Velez JC, Bland AM, Arthur JM, Raymond JR, Janech MG (2008) Characterization of renin-angiotensin system enzyme activities in cultured mouse podocytes. American Journal of Physiology - Renal Physiology 295:398-407 doi:10.1152/ajprenal.00050.2007

5. Velez JC, Ierardi JL, Bland AM, Morinelli TA, Arthur JM, Raymond JR, Janech MG (2012) Enzymatic processing of angiotensin peptides by human glomerular endothelial cells. American Journal of Physiology - Renal Physiology 302:1583-1594 doi:10.1152/ajprenal.00087.2012

6. Reid JD1, Holmes DT, Mason DR, Shah B, Borchers CH (2012) Towards the development of an immuno MALDI (iMALDI) mass spectrometry assay for the diagnosis of hypertension. J Am Soc Mass Spectrom 21:1680-6 doi: 10.1016/j.jasms.2010.01.024

7. Li N, Zimpelmann J, Cheng K, Wilkins JA, Burns KD (2005) The role of angiotensin converting enzyme 2 in the generation of angiotensin 1-7 by rat proximal tubules. Am J Physiol Renal Physiol 288:F353-62 DOI: 10.1152/ajprenal.00144.2004

8. Donoghue M, Hsieh F, Baronas E, Godbout K, Gosselin M, Stagliano N, Donovan M, Woolf B, Robison K, Jeyaseelan R, Breitbart RE, Acton S (2000) A novel angiotensin-converting enzyme-related carboxypeptidase (ACE2) converts angiotensin I to angiotensin 1-9. Circ Res 87:E1-9 doi: 10.1161/01.RES.87.5.e1

9. Gurley SB, Allred A, Le TH, Griffiths R, Mao L, Philip N, Haystead TA, Donoghue M, Breitbart RE, Acton SL, Rockman HA, Coffman TM (2006) Altered blood pressure responses and normal cardiac phenotype in ACE2-null mice. J Clin Invest 116:2218-25 doi: 10.1172/JCI16980

10. Grobe N, Weir NM, Leiva O, Ong FS, Bernstein KE, Schmaier AH, Morris M, Elased KM (2013) Identification of prolyl carboxypeptidase as an alternative enzyme for processing of renal angiotensin II using mass spectrometry. Am J Physiol Cell Physiol. 304:C945-53 doi: 10.1152/ajpcell.00346.2012

11. Fyhrquist F, Saijonmaa O (2008) Renin-angiotensin system revisited. J Intern Med 264:224-36 doi: 10.1111/j.1365-2796.2008.01981.x.

12. Weir MR, Dzau VJ (1999) The renin-angiotensin-aldosterone system: a specific target for hypertension management. Am J Hypertens 12:205S-213S

13. Hollenberg NK (1984) The renin-angiotensin system and sodium homeostasis. J Cardiovasc Pharmacol 6:S176-83

14. Bouzegrhane F, Thibault G (2002) Is angiotensin II a proliferative factor of cardiac fibroblasts? Cardiovasc Res 53:304-12

15. Santos RA, Ferreira AJ (2007) Angiotensin-(1-7) and the renin-angiotensin system. Curr Opin Nephrol Hypertens 16:122-8

16. Caprioli RM, Farmer TB, and Gile J (1997) Molecular Imaging of Biological Samples: Localization of Peptides and Proteins Using MALDI-TOF MS. Analytical Chemistry 69:4751–4760 DOI: 10.1021/ac970888i

17. Powers TW , Jones EE, Betesh LR, Romano PR, Gao P, Copland JA, Mehta AS, Drake RR (2013) Matrix Assisted Laser Desorption Ionization Imaging Mass Spectrometry Workflow for Spatial Profiling Analysis of N-Linked Glycan Expression in Tissues. Analytical Chemistry 85:9799–9806 DOI: 10.1021/ac402108x

18. Bantscheff M, Schirle M, Sweetman G, RickJ, Kuster B (2007) Quantitative mass spectrometry in proteomics: a critical review. Analytical and Bioanalytical Chemistry 389:1017-1031. doi:10.1007/s00216-007-1486-6

19. Beck M, Schmidt A, Malmstroem J, Claassem M, Ori A, Szymborska A, Herzog F, Rinner O, Ellenberg J, Aebersold R (2011) The quantitative proteome of a human cell line. Molecular Systems Biology 7:549 doi:10.1038/msb.2011.82

20. Nagaraj N, Wisniewski JR, Geiger T, Cox J, Kircher M, Kelso J, Paabo S, Mann M (2011) Deep proteome and transcriptome mapping of a human cancer cell line. Molecular Systems Biology 7:548 doi:10.1038/msb.2011.81

21. Bantscheff M, Lemeer S, Savitski MM, Kuster B (2012) Quantitative mass spectrometry in proteomics: critical review update from 2007 to the present. Analytical and Bioanalytical Chemistry 404:939-965 doi:10.1007/s00216-012-6203-4

22. Yalow R, Berson S (1960) Immunoassay of endogenous plasma insulin in man. J. Clin. Invest 39: 1157–75 doi:10.1172/JCI104130

23. Lequin R (2005) Enzyme immunoassay (EIA)/enzyme-linked immunosorbent assay (ELISA) Clin. Chem 51: 2415–8 doi:10.1373/clinchem.2005.051532

24. Ong S-E, Blagoev B, Kratchmarova I, Kristensen DB, Steen H, Pandey A, Mann M (2002) Stable Isotope Labeling by Amino Acids in Cell Culture, SILAC, as a Simple and Accurate Approach to Expression Proteomics. Molecular and Cellular Proteomics 1:376-386 doi:10.1074/mcp.M200025-MCP200

25. Yao X, Freas A, Ramirez J, Demirev PA, Fenselau C (2001) Proteolytic 18O labeling for comparative proteomics: model studies with two serotypes of adenovirus. Analytical Chemistry 73:2836-2842 doi:10.1021/ac001404c

26. Ross PL, Huang YN, Marchese JN, Williamson B, Parker K, Hattan S, Khainovski N, Pillai S, Dey S, Daniels S, Purkayastha S, Juhasz P, Martin S, Bartlet-Jones M, He F, Jacobson A, Pappin DJ (2004) Multiplexed Protein Quantitation in Saccharomyces

cerevisiae Using Amine-reactive Isobaric Tagging Reagents. Mollecular and Cellular Proteomics 3:1154-1169 doi:10.1074/mcp.M400129-MCP200

27. Thompson A, Schäfer J, Kuhn K, Kienle S, Schwarz J, Schmidt G, Neumann T, Johnstone R, Mohammed AK, Hamon C (2003) Tandem mass tags: a novel quantification strategy for comparative analysis of complex protein mixtures by MS/MS. Analytical Chemistry 75:1895-1904 doi:10.1021/ac0262560

28. Kirkpatrick DS1, Gerber SA, Gygi SP (2005) The absolute quantification strategy: a general procedure for the quantification of proteins and post-translational modifications. Methods 35:265-273 doi:10.1016/j.ymeth.2004.08.018

29. Camenzind AG, van der Gugten JC, Popp R, Holmes DT, Borchers CH (2013) Development and evaluation of an immuno-MALDI (iMALDI) assay for angiotensin I and the diagnosis of secondary hypertension. Clinical Proteomics 10:20. doi:10.1186/1559-0275-10-20

30. Ong S-E, Mann M (2005) Mass spectrometry–based proteomics turns quantitative. Nature Chemical Biology 1:252-262 doi:10.1038/nchembio736

31. Cui L1, Nithipatikom K, Campbell WB (2007) Simultaneous analysis of angiotensin peptides by LC-MS and LC-MS/MS: metabolism by bovine adrenal endothelial cells. Anal Biochem. 369:27-33

32. Lortie M, Bark S, Blantz R, Hook V (2009) Detecting low-abundance vasoactive peptides in plasma: progress toward absolute quantitation using nano liquid chromatography-mass spectrometry. Anal Biochem. 394:164-70 doi: 10.1016/j.ab.2009.07.021

33. Allen DK, Evans BS, Libourel IG (2014) Analysis of isotopic labeling in peptide fragments by tandem mass spectrometry. PLoS One 9:e91537 doi: 10.1371/journal.pone.0091537

34. Bronsema KJ, Bischoff R, van de Merbel NC (2012) Internal standards in the quantitative determination of protein biopharmaceuticals using liquid chromatography coupled to mass spectrometry. J Chromatogr B Analyt Technol Biomed Life Sci. 893-894:1-14 doi: 10.1016/j.jchromb.2012.02.021.

35. Gerber SA, Rush J, Stemman O, Kirschner MW, Gygi SP (2003) Absolute quantification of proteins and phosphoproteins from cell lysates by tandem MS. Proceedings of the National Academy of Natural Sciences of the United States of Americs 100:6940-6945 doi:10.1073/pnas.0832254100

36. Gygi SP1, Rist B, Gerber SA, Turecek F, Gelb MH, Aebersold R (1999) Quantitative analysis of complex protein mixtures using isotope-coded affinity tags. 17:994-999 doi:10.1038/13690

37. Silva JC, Gorenstein MV, Li GZ, Vissers JP, Geromanos SJ (2006) Absolute quantification of proteins by LCMSE: a virtue of parallel MS acquisition. Molecular and Cellular Proteomics 5:144-156 doi:10.1074/mcp.M500230-MCP200

38. Kuzyk MA, Smith D, Yang J, Cross TJ, Jackson AM, Hardie DB, Anderson NL, Borchers CH (2009) Multiple Reaction Monitoring-based, Multiplexed, Absolute Quantitation of 45 Proteins in Human Plasma. Molecular and Cellular Proteomics 8:1860–1877. doi:10.1074/mcp.M800540-MCP200

39. Melnykov V (2013) Finite mixture modelling in mass spectrometry analysis. Journal of the Royal Statistical Society: Series C 62:573–592 doi:10.1111/rssc.12010

40. Polanska J, Plechawska M, Pietrowska M, Marczak L (2012) Gaussian mixture decomposition in the analysis of MALDI-TOF spectra. Expert Systems 29:216-231 doi:10.1111/j.1468-0394.2011.00582.x

41. Nezami Ranjbar MR, Zhao Y, Tadesse MG, Wang Y, Ressom HW (2013) Gaussian process regression model for normalization of LC-MS data using scan-level information. Proteome Sci 11:S13 doi: 10.1186/1477-5956-11-S1-S13.

42. Plechawska M, Polańska J (2009) Simulation of the usage of Gaussian mixture models for the purpose of modelling virtual mass spectrometry data. Stud Health Technol Inform 150:804-8.

43. Dempster AP, Laird NM, Rubin DM (1977) Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B 39:1-31

44. Markus K (2012) Methods in Molecular Biology: Quantitative Methods in Proteomics. Springer. 85-100, 489-499 p. doi:10.1007/978-1-61779-885-6_29

45. Gonzalez-Galarza FF, Lawless C, Hubbard SJ, Fan J, Bessant C, Hermjakob H, Jones AR (2012) A Critical Appraisal of Techniques, Software Packages, and Standards for Quantitative Proteomic Analysis. 16:431–442 doi:10.1089/omi.2012.0022

46. Karpievitch VY, Hill EG, Smolka AJ, Morris JS, Coombes KR, Baggerly KA, Almeida JS (2007) PrepMS: TOF MS data graphical preprocessing tool. Bioinformatics 23:264-265. doi: 10.1093/bioinformatics/btl583

47. Parry RM, Galhena AS, Gamage CM, Bennett RV, Wang MD, Fernández FM (2013) omniSpect: an open MATLAB-based tool for visualization and analysis of matrix-assisted laser desorption/ionization and desorption electrospray ionization mass spectrometry images. Journal of the American Society for Mass Spectrometry 24:646-649 doi:10.1007/s13361-012-0572-y

48. Deutsch EW, Lam H, Aebersold R (2008) Data analysis and bioinformatics tools for tandem mass spectrometry in proteomics. Physiological Genomics 33:18-25 doi:10.1152/physiolgenomics.00298.2007

49. Chambers MC, MacLean B, Burke R, Amode D, Ruderman DL, Neumann S, Gatto L, Fischer B, Pratt B, Egertson J, Hoff K, Kessner D, Tasman N, Shulman N, Frewen B, Baker TA, Brusniak MY, Paulse C, Creasy D, Flashner L, Kani K, Moulding C, Seymour SL, Nuwaysir LM, 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 EW, Moritz

RL, Katz JE, Agus DB, MacCoss M, Tabb DL, Mallick P (2012) A cross-platform toolkit for mass spectrometry and proteomics. Nature Biotechnology 30:918-920 doi:10.1038/nbt.2377

50. R Core Team (2013). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.

51. Smith CA, Want EJ, O'Maille G, Abagyan R, Siuzdak G (2006) XCMS: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. Analytical Chemistry 78:779-787 doi:10.1021/ac051437y

52. Tautenhahn R, Böttcher C, Neumann S (2008) Highly sensitive feature detection for high resolution LC/MS. Bioinformatics 9:504 doi:10.1186/1471-2105-9-504

53. Benton HP, Want EJ, Ebbels TMD (2010) Correction of mass calibration gaps in liquid chromatography–mass spectrometry metabolomics data. Bioinformatics 22:2488-2489 doi:10.1093/bioinformatics/btq441

54. Barkauskas DA, Rocke DM (2010) A general-purpose baseline estimation algorithm for spectroscopic data. Analytica Chimica Acta 657:191–197 doi:10.1016/j.aca.2009.10.043.

55. Russell DH, Edmondson RD (1997) High-resolution Mass Spectrometry and Accurate Mass Measurements with Emphasis on the Characterization of Peptides and Proteins by Matrix-assisted Laser Desorption/Ionization Time-of-Ñight Mass Spectrometry. Journal of Mass Spectrometry 32:263-276 DOI: 10.1002/(SICI)1096-9888(199703)32:3

56. Jankowski V, Vanholder R, van der Giet M, Tölle M, Karadogan S, Gobom J, Furkert J, Oksche A, Krause E, Tran TN, Tepel M, Schuchardt M, Schlüter H, Wiedon A, Beyermann M, Bader M, Todiras M, Zidek W, Jankowski J (2007) Mass-spectrometric identification of a novel angiotensin peptide in human plasma. Arterioscler Thromb Vasc Biol. 27:297-302

Chapter 3: Manuscript Two: Algorithm Method, Simulation Method and Simulations

1. Bantscheff M, Schirle M, Sweetman G, RickJ, Kuster B (2007) Quantitative mass spectrometry in proteomics: a critical review. Analytical and Bioanalytical Chemistry 389:1017-1031. doi:10.1007/s00216-007-1486-6

2. Beck M, Schmidt A, Malmstroem J, Claassem M, Ori A, Szymborska A, Herzog F, Rinner O, Ellenberg J, Aebersold R (2011) The quantitative proteome of a human cell line. Molecular Systems Biology 7:549 doi:10.1038/msb.2011.82

3. Bantscheff M, Lemeer S, Savitski MM, Kuster B (2012) Quantitative mass spectrometry in proteomics: critical review update from 2007 to the present. Analytical and Bioanalytical Chemistry 404:939-965 doi:10.1007/s00216-012-6203-4

4. Gerber SA, Rush J, Stemman O, Kirschner MW, Gygi SP (2003) Absolute quantification of proteins and phosphoproteins from cell lysates by tandem MS. PNAS 100:6940-6945 doi:10.1073/pnas.0832254100

5. Spainhour JC, Janech MG, Schwacke JH, Velez JC, Ramakrishnan R (2014) The Application of Gaussian Mixture Models for Signal Quantification in MALDI-ToF Mass Spectrometry of Peptides accepted for publication to PLOS One

6. Schwacke JH, Spainhour JCG, Ierardi JL, Chaves JM, Arthur JM, Janech MG, Velez JC (2013) Network modeling reveals steps in angiotensin peptide processing. Hypertension 61:690-700 doi:10.1161/HYPERTENSIONAHA.111.00318

7. Siepen JA,Keevil EJ, Knight D, Hubbard SJ. (2007) Prediction of missed cleavage sites in tryptic peptides aids protein identification in proteomics. J. ProteomeRes. 6:399-408.

8. Pfeifer N, Leinenbach A, Huber CG, Kohlbacher O. (2007) Statistical learning of peptide retention behavior in chromatographic separations: a new kernel - based approach for computational proteomics. BMC bioinformatics 8:468.

9. Lan K, Jorgenson JW. 2001) A hybrid of exponential and gaussian functions as a simple model of asymmetric chromatographic peaks. J. Chromatography A 915:1-13.

10. Zhou C, Bowler LD, Feng J. (2008) A machine learning approach to explore the spectra intensity pattern of peptides using tandem mass spectrometry data. BMC bioinformatics 9:325

11. Chris Bielow, Stephan Aiche, Sandro Andreotti, Knut Reinert (2011) MSSimulator: Simulation of Mass Spectrometry Data. J.ProteomeRes. 10:2922–2929 dx.doi.org/10.1021/pr200155f

12. Schulz-Trieglaff O, Pfeifer N, Gröpl C, Kohlbacher O, Reinert K. (2008) LC-MSsim--a simulation software for liquid chromatography mass spectrometry data. BMC bioinformatics 9:423.

13. Hoffmann, E, Vincent Stroobant, V (2007) Mass Spectrometry: Principles and Applications. Wiley-Interscience
14. R Core Team (2013). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.

15. Velez JC. The importance of the intrarenal renin-angiotensin system. *Nat Clin Pract Nephrol*. 2009;5:89–100.

16. Velez JC, Bland AM, Arthur JM, Raymond JR, Janech MG. Characterization of renin-angiotensin system enzyme activities in cultured mouse podocytes. *Am J Physiol Renal Physiol*. 2007;293:F398–F407.

17. Velez JC, Ryan KJ, Harbeson CE, et al. Angiotensin I is largely converted to angiotensin (1-7) and angiotensin (2-10) by isolated rat glomeruli. *Hypertension*. 2009;53:790-797.

18. Velez JC, Ryan KJ, Harbeson CE, Bland AM, Budisavljevic MN, Arthur JM, Fitzgibbon WR, Raymond JR, Janech MG. Angiotensin I is largely converted to angiotensin (1-7) and angiotensin (2-10) by isolated rat glomeruli. *Hypertension*. 2009;53:790–797.

19. Velez JC, Ierardi JL, Bland AM, Morinelli TA, Arthur JM, Raymond JR, Janech MG. Enzymatic processing of angiotensin peptides by human glomerular endothelial cells. *Am J Physiol Renal Physiol*. 2012;302:F1583–F1594.

20. CRC 91st ed (2010)

Chapter 4: Manuscript Three: MLE based estimator of parameters using an EM algorithm approach

1. Gevaert K, Impens F, Ghesquière B, Van Damme P, Lambrechts A, et al. (2008) Stable isotopic labeling in proteomics. *Proteomics* 8: 4873–4885.

2. Noble WS, MacCoss MJ (2012) Computational and Statistical Analysis of Protein Mass Spectrometry Data. *PLoS Comput Biol* 8: e1002296. doi:10.1371/journal.pcbi.1002296

3. Oberg AL and Mahoney DW (2012) Statistical methods for quantitative mass spectrometry proteomic experiments with labeling. *BMC Bioinformatics* 13:S7 doi:10.1186/1471-2105-13-S16-S7

4. Redner RA and Walker HF (1984) Mixture Densities, Maximum Likelihood and the Em Algorithm. *SIAM Review* 26:195-239

5. Spainhour JC, Janech MG, Schwacke JH, Velez JC, Ramakrishnan R (2014) The Application of Gaussian Mixture Models for Signal Quantification in MALDI-ToF Mass Spectrometry of Peptides accepted for publication to *PLOS One*

6. Schwacke JH, Spainhour JCG, Ierardi JL, Chaves JM, Arthur JM, Janech MG, Velez JC (2013) Network modeling reveals steps in angiotensin peptide processing. *Hypertension* 61:690-700 doi:10.1161/HYPERTENSIONAHA.111.00318

7. Spainhour et al *BMC bioinformatics*, yet to be published

Appendix
I. Sample Code

```
##reworked Pepnfo for easier table reconstruction from v8, more user friendly
##and allows for easier change of peptides outside angiotensin searches of mass spectra

library(xcms)
############################################################
UnlabeledSequence = NA
UnlabeledPeptides = NA
LabeledSequence = NA
LabeledPeptides = NA

############################################################
#Construct AAcomp globally used as a lookup tabel for amino acid compositions, avoid repeated reconstruction in loops
	AAComp=rbind(c(3,2,1,1,0,0),c(5,3,1,1,0,0),c(5,3,1,2,0,0),c(7,5,1,1,0,0),c(9,5,1,1,0,0),c(7,4,1,2,0,0),c(5,3,1,1,0,1),c(11,6,1,1,0,0),c(11,6,1,1,0,0),c(6,4,2,2,0,0),c(5,4,1,3,0,0),c(8,5,2,2,0,0),c(12,6,2,1,0,0),c(7,5,1,3,0,0),c(9,5,1,1,0,1),c(7,6,3,1,0,0),c(9,9,1,1,0,0),c(12,6,4,1,0,0),c(9,9,1,2,0,0),c(10,11,2,1,0,0))
	rownames(AAComp)=c('G','A','S','P','V','T','C','L','I','N','D','Q','K','E','M','H','F','R','Y','W')
	colnames(AAComp)=c('H','C','N','O','P','S')
	AA=c('G','A','S','P','V','T','C','L','I','N','D','Q','K','E','M','H','F','R','Y','W')
############################################################
#Construct matrix of atom totals for a given peptide, includes a water molecule
AtomMatrix=function(pep, Water=TRUE){
	AtomMatrix = matrix(0,nchar(pep)+1,6)  ##+1 for extra water at ends of peptide/single amino acid

	for (i in 1:nchar(pep)){
		AtomMatrix[i,]= AAComp[substring(pep,i,i),] ##Extract substrings in a character vector
		}
		if(Water){
	AtomMatrix[nchar(pep)+1,]=c(2,0,0,1,0,0)  ##Add water to count
			}
	AtomMatrix
}
############################################################
##Gives monotopic mass (isotope 0) of a given peptide summed as H,C,N,O,P,S
MassCalc <- function(pep, AAComp, proton=1, MH=TRUE){
	AtomMatrix <- AtomMatrix(pep)

	Mass <- sum(AtomMatrix[,1])*1.007825032+sum(AtomMatrix[,2])*12+sum(AtomMatrix[,3])*14.003074004+sum(AtomMatrix[,4])*15.994914619+sum(AtomMatrix[,5])*30.973761632+sum(AtomMatrix[,6])*31.972071002##CHECK CRC
	if(MH){
		Mass <- (Mass+proton*1.007276467)/proton##Note peaks become(MassNuetron)/#Protons apart
```

```
	}
	Mass
}
#############################################################
##Generates point distrobutins for a given combination of atoms (peptide)
IsotopicCountFFT = function (pep,cutoff=.9999){

	AtomMatrix=AtomMatrix(pep)

	Molecule=c(H=sum(AtomMatrix[,1]),C=sum(AtomMatrix[,2]),N=sum(AtomMatr
ix[,3]),O=sum(AtomMatrix[,4]),P=sum(AtomMatrix[,5]),S=sum(AtomMatrix[,6]))

	##check for all atoms being present using molecule to check for 0's
	#ScanMolecule = Molecule[Molecule!=0]

	##CRC values for isotopic point distrobutions for each element
	hd = c(0.99985, 0.00015) ##1,2
	cd = c(0.98900, 0.01100) ##12,13
	nd = c(0.99634, 0.00366) ##14,15
	od = c(0.99760, 0.00039, 0.00201) ##16,17,18
	pd = c(1.00000) ##31
	sd = c(0.95020, 0.00750, 0.04210, 0.00000, 0.00020) ##32,33,34,35,36
CHANGETHIS!!!!

	## Counting each occurance of a given atom in the selected molecule
	hcount = Molecule['H']
	ccount = Molecule['C']
	ncount = Molecule['N']
	ocount = Molecule['O']
	pcount = Molecule['P']
	scount = Molecule['S']

	##Creating vector for Fast Fourier Transform based on number of atoms of a
given type present
	##FFT work faster on vectors that have a length of 2^x, use 256 here since we
look at low dalton (mass) proteins
	##look at 128 innstead of 256 for speed

	MoleculeFFT=1
	if (hcount>0){
		hd2 = append(hd,rep(0,256-length(hd)))
		hd2=fft(hd2)^hcount
		MoleculeFFT=MoleculeFFT*hd2
	}
	if (ccount>0){
		cd2 = append(cd,rep(0,256-length(cd)))
		cd2 = fft(cd2)^ccount
		MoleculeFFT=MoleculeFFT*cd2
	}
	if (ncount>0){
```

```
		nd2 = append(nd,rep(0,256-length(nd)))
		nd2 = fft(nd2)^ncount
		MoleculeFFT=MoleculeFFT*nd2
	}
	if (ocount>0){
		od2 = append(od,rep(0,256-length(od)))
		od2 = fft(od2)^ocount
		MoleculeFFT=MoleculeFFT*od2
	}
	if (pcount>0){
		pd2 = append(pd,rep(0,256-length(pd)))
		pd2 = fft(pd2)^pcount
		MoleculeFFT=MoleculeFFT*pd2
	}
	if (scount>0){
		sd2 = append(sd,rep(0,256-length(sd)))
		sd2 = fft(sd2)^scount
		MoleculeFFT=MoleculeFFT*sd2
	}

	distri = fft(MoleculeFFT,inverse=TRUE)/length(MoleculeFFT)

	##Remove the complex part of each entry in dristi
	distri = Re(distri)

	# Remove those peaks (higher masses) that contribute less than 1/100% of the
total mass
	stop = min(which(cumsum(distri)>cutoff))
	distri[1:stop]

	###distri
	##This produces slight differences from old convolution method due rounding
errors to transformation, power raise, inverse transformation
	##still produces - negnumbers use rnd if this becomes a problem
}
################################################
################################################
##Generate a Mixture of Normals model, Mean separated by one Neutron standars
deviations
##are equal this curve is used to model the peaks of a mass spectrum
MultiNormCalc=function(SpectraRange,peptide,mass,sigma,mzError=0){

	Mass=mass ##This calulated previously
	h=IsotopicCountFFT(peptide)
	mz=SpectraRange#x-axis, the mass charge
	peaks=rep(0,length(mz))
	for(i in 1:length(h)){
		peaks=peaks+dnorm(mz-Mass,((i-1)*1.008664916)+mzError,sigma)*h[i]
	}
	Predicted=matrix(0,length(mz),2)
```

```
	Predicted[,1]=mz
	Predicted[,2]=peaks
	Predicted
}
##############################################
PeptideInfo2=function(UnlabeledPeptides,UnlabeledSequence,LabeledPeptides,LabeledS
equence,AQUAState="Both",Neutron=0,ClusterCut=5){
switch(AQUAState,
	Both = {##Makes list for both labeled and unlabeled peptides
	Peptides=cbind(UnlabeledSequence)
	rownames(Peptides)=UnlabeledPeptides
	HeavyPeptides=cbind(LabeledSequence)
	rownames(HeavyPeptides)=LabeledPeptides

PeptideInformation=data.frame(Peptide=NA,Sequence=NA,Mass=NA,Spectra=NA,Spec
traCount=NA,Cluster=NA,mzError=NA,Sigma=NA,Area=NA,NoiseAdjustSlope=NA,N
oiseAdjustIntercept=NA,Rsq=NA,ResidualMean=NA,ResidualVariance=NA,TIC=NA,N
ormalizedArea=NA)

	for(i in 1:length(Peptides)){
	PeptideInformation[i,1] <- rownames(Peptides)[i]
	PeptideInformation[i,2] <- Peptides[i]
	PeptideInformation[i,3] <- MassCalc(Peptides[i])
	}
for(i in 1:length(HeavyPeptides)){
	PeptideInformation[i+length(Peptides),1] <-
paste('AQUA',rownames(HeavyPeptides)[i])
	PeptideInformation[i+length(Peptides),2] <- HeavyPeptides[i]
	PeptideInformation[i+length(Peptides),3] <-
MassCalc(HeavyPeptides[i])+Neutron*1.008664916
}

##Order peptides based on mass
PeptideInformation <- PeptideInformation[order(-PeptideInformation$Mass),]

##Assign clusters to each group of peaks less than ClusterCut mz apart
counter=0
for(i in 1:length(PeptideInformation$Mass)){
	if(i<=length(PeptideInformation$Mass)-1){
		if ((PeptideInformation$Mass[i]-
PeptideInformation$Mass[i+1])<ClusterCut){
			PeptideInformation$Cluster[i]=counter+1
			PeptideInformation$Cluster[i+1]=counter+1
		}
		else{
			PeptideInformation$Cluster[i]=counter+1
			PeptideInformation$Cluster[i+1]=counter+2
			counter=counter+1}
	}
	else
```

```
		if ((PeptideInformation$Mass[i-1]-
PeptideInformation$Mass[i])<ClusterCut){
			PeptideInformation$Cluster[i]=counter+1
		}
		else{
			PeptideInformation$Cluster[i]=counter+1
		}
}
PeptideInformation
		},
	AQUA = {##Makes list for just labeled peptides
	HeavyPeptides=cbind(LabeledSequence)
	rownames(HeavyPeptides)=LabeledPeptides

PeptideInformation=data.frame(Peptide=NA,Sequence=NA,Mass=NA,Spectra=NA,Spec
traCount=NA,Cluster=NA,mzError=NA,Sigma=NA,Area=NA,NoiseAdjustSlope=NA,N
oiseAdjustIntercept=NA,Rsq=NA,ResidualMean=NA,ResidualVariance=NA,TIC=NA,N
ormalizedArea=NA)


for(i in 1:length(HeavyPeptides)){
			PeptideInformation[i,1] <-
paste('AQUA',rownames(HeavyPeptides)[i])
			PeptideInformation[i,2] <- HeavyPeptides[i]
			PeptideInformation[i,3] <-
MassCalc(HeavyPeptides[i])+Neutron*1.008664916
	}


##Order peptides based on mass
PeptideInformation <- PeptideInformation[order(-PeptideInformation$Mass),]

##Assign clusters to each group of peaks less than ClusterCut mz apart
counter=0
if(length(PeptideInformation$Mass) < 2){
PeptideInformation$Cluster[1]=1
}
else{
for(i in 1:length(PeptideInformation$Mass)){
	if(i<=length(PeptideInformation$Mass)-1){
		if ((PeptideInformation$Mass[i]-
PeptideInformation$Mass[i+1])<ClusterCut){
			PeptideInformation$Cluster[i]=counter+1
			PeptideInformation$Cluster[i+1]=counter+1
		}
		else{
			PeptideInformation$Cluster[i]=counter+1
			PeptideInformation$Cluster[i+1]=counter+2
			counter=counter+1}
	}
```

```
	else
		if ((PeptideInformation$Mass[i-1]-
PeptideInformation$Mass[i])<ClusterCut){
			PeptideInformation$Cluster[i]=counter+1
		}
		else{
			PeptideInformation$Cluster[i]=counter+1
		}
}}
PeptideInformation
		},
	Normal = {##Makes list for just unlabeled peptides
	Peptides=cbind(UnlabeledSequence)
	rownames(Peptides)=UnlabeledPeptides

PeptideInformation=data.frame(Peptide=NA,Sequence=NA,Mass=NA,Spectra=NA,Spec
traCount=NA,Cluster=NA,mzError=NA,Sigma=NA,Area=NA,NoiseAdjustSlope=NA,N
oiseAdjustIntercept=NA,Rsq=NA,ResidualMean=NA,ResidualVariance=NA,TIC=NA,N
ormalizedArea=NA)

for(i in 1:length(Peptides)){
	PeptideInformation[i,1] <- rownames(Peptides)[i]
	PeptideInformation[i,2] <- Peptides[i]
	PeptideInformation[i,3] <- MassCalc(Peptides[i])
}

##Order peptides based on mass
PeptideInformation <- PeptideInformation[order(-PeptideInformation$Mass),]

##Assign clusters to each group of peaks less than ClusterCut mz apart
counter=0
if(length(PeptideInformation$Mass) < 2){
PeptideInformation$Cluster[1]=1
}
else{
for(i in 1:length(PeptideInformation$Mass)){
	if(i<=length(PeptideInformation$Mass)-1){
		if ((PeptideInformation$Mass[i]-
PeptideInformation$Mass[i+1])<ClusterCut){
			PeptideInformation$Cluster[i]=counter+1
			PeptideInformation$Cluster[i+1]=counter+1
		}
		else{
			PeptideInformation$Cluster[i]=counter+1
			PeptideInformation$Cluster[i+1]=counter+2
			counter=counter+1}
	}

	else
```

```
		if ((PeptideInformation$Mass[i-1]-
PeptideInformation$Mass[i])<ClusterCut){
			PeptideInformation$Cluster[i]=counter+1
		}
		else{
			PeptideInformation$Cluster[i]=counter+1
		}
}}
PeptideInformation
		})
}

##############################################
##Calculate R squared of a given fit from the data
RsqCalc <- function(SpectraPeptideRange,yhat){

	r1=SpectraPeptideRange[,2]-yhat
	r2=SpectraPeptideRange[,2]-mean(SpectraPeptideRange[,2])
	R=1-(sum(r1^2)/sum(r2^2))

	R
}
##############################################
##Brute force search of parameter space consisting of sigma and mzError
##For looking at flat baseline
MultiPeakPeptideScan=function(FileList,PeptideInformation,SigmaRange,mzErrorRange
,SpectraMin=500,SpectraMax=1350,RsqMin=0,NormalizeMethod="NoNormalization"){
##Modifying Peptide Information from PeptideInfo2 to fit this function
PeptideInformation=PeptideInformation[,-11]
colnames(PeptideInformation)[10] = "NoiseAdjust"

##Construct data frame for inforamtion on peptides looked for
TotalPeptideInformation=data.frame(Peptide=NA,Sequence=NA,Mass=NA,Spectra=NA,
SpectraCount=NA,Cluster=NA,mzError=NA,Sigma=NA,Area=NA,NoiseAdjust=NA,Rs
q=NA,ResidualMean=NA,ResidualVariance=NA,TIC=NA,NormalizedArea=NA)

##Run through each of the spectra in the file, comparingthe data to the constructed
PeptideInformation data frame and filling in the blanks
Sigma=SigmaRange
mzError=mzErrorRange

for(x in 1:length(FileList)){
	SpectraImput <- xcmsRaw(FileList[x])
	TIC <- mean(SpectraImput@tic)
	SpectraAverage <- getSpec(SpectraImput,mzrange=c(SpectraMin,SpectraMax))

	cat("Loaded",x,"Spectra of",length(FileList),"\n")

	for(i in 1:max(PeptideInformation$Cluster)){
		Cluster <- PeptideInformation[PeptideInformation$Cluster==(i),]
```

```
		PeptideInformation[PeptideInformation$Cluster==(i),]$Spectra <-
FileList[x]
		SpectraPeptideRange <-
SpectraAverage[SpectraAverage[,1]>=(min(Cluster$Mass)-1.008664916) &
SpectraAverage[,1]<=(max(Cluster$Mass)+5*1.008664916),]

		Rsq <- RsqMin
		for(j in 1:length(Sigma)){
			for(k in 1:length(mzError)){
				PredictedData <- rep(1,length(SpectraPeptideRange[,1]))
##generate new PredictedData
				for(l in 1:length(Cluster$Mass)){
					PD <-
MultiNormCalc(SpectraPeptideRange[,1],Cluster$Sequence[l],Cluster$Mass[l],Sigma[j],
mzError[k])
					PredictedData=cbind(PD[,2],PredictedData)
				}

				Soln=qr.solve(PredictedData,SpectraPeptideRange[,2])
				yhat=PredictedData %*% Soln  ##generate fit from model
###Correct by using model...??

				R=RsqCalc(SpectraPeptideRange,yhat)

				if(R>Rsq){
					Rsq=R

	PeptideInformation[PeptideInformation$Cluster==(i),]$mzError=mzError[k]

	PeptideInformation[PeptideInformation$Cluster==(i),]$Sigma=Sigma[j]
						MatrixAreas=Soln[-length(Soln)]
						ReturnedAreas=rev(MatrixAreas)

	PeptideInformation[PeptideInformation$Cluster==(i),]$Area=ReturnedAreas

	PeptideInformation[PeptideInformation$Cluster==(i),]$NoiseAdjust=Soln[length(
Soln)]

	PeptideInformation[PeptideInformation$Cluster==(i),]$Rsq=Rsq

	PeptideInformation[PeptideInformation$Cluster==(i),]$SpectraCount=x
						Residuals=SpectraPeptideRange[,2]-yhat
						mR=mean(Residuals)
						lR=length(Residuals)

	PeptideInformation[PeptideInformation$Cluster==(i),]$ResidualMean=mean(Resi
duals)
						variance=(1/(lR - 1))*(sum((Residuals -
mR)^2))
```

```
        PeptideInformation[PeptideInformation$Cluster==(i),]$ResidualVariance=varianc
e

        PeptideInformation[PeptideInformation$Cluster==(i),]$TIC=TIC
                                }
                        }
                }
        }
##Use Peak or Peptide for comparisons with in a spectra, Ion to compare across spectra
        switch(NormalizeMethod[1],
                Peak =
{PeptideInformation$NormalizedArea=PeptideInformation$Area/max(PeptideInformatio
n$Area)}, ##Normalizes by largest peak looked for
                Peptide =
{PeptideInformation$NormalizedArea=PeptideInformation$Area/
PeptideInformation[PeptideInformation$Sequence==NormalizeMethod[2],]$Area},
##Normalizes bu selected peptide
                Ion =
{PeptideInformation$NormalizedArea=PeptideInformation$Area/TIC}, ##Normalizes by
Total Ion Count/Current
                NoNormalization = {PeptideInformation$NormalizedArea=NA}) ##No
normalization asked for

        TotalPeptideInformation=rbind(TotalPeptideInformation,PeptideInformation)
##Link all generated PeptideInformation data frames together
}
TotalPeptideInformation=TotalPeptideInformation[-1,]
TotalPeptideInformation
}
################################################
##Psudo Newton-Raphson fitting method used for fitting model, looking at sigma and
mzError parameters
##quandrant gradient assnet method fo crossing search space implemented
##Root bysection in two dimensions
##add. wif,.t2d,.mgf formats with text,csv,tsd
MultiPeakPeptideScan2=function(FileList,PeptideInformation,SigmaRange,mzErrorRang
e,RangeCutoff=c(1,5),SpectraMin=500,SpectraMax=1350,RsqMin=0,NormalizeMethod=
"NoNormalization",InterationLimit=100,StartPoint=c(0,.1,0)){

TotalPeptideInformation=data.frame(Peptide=NA,Sequence=NA,Mass=NA,Spectra=NA,
SpectraCount=NA,Cluster=NA,mzError=NA,Sigma=NA,Area=NA,NoiseAdjustSlope=N
A,NoiseAdjustIntercept=NA,Rsq=NA,ResidualMean=NA,ResidualVariance=NA,TIC=N
A,NormalizedArea=NA)

IntChoice = 1:InterationLimit %% 2 == 0

for(i in 1:length(FileList)){
        SpectraImput <- xcmsRaw(FileList[i])
        TIC <- mean(SpectraImput@tic)
        SpectraAverage <- getSpec(SpectraImput,mzrange=c(SpectraMin,SpectraMax))
```

```
	cat("Loaded",i,"Spectra of",length(FileList),"\n")
	for(j in 1:max(PeptideInformation$Cluster)){
		SavePnt=StartPoint
		Cluster <- PeptideInformation[PeptideInformation$Cluster==(j),]
		PeptideInformation[PeptideInformation$Cluster==(j),]$Spectra <-
FileList[i]
		SpectraPeptideRange <-
SpectraAverage[SpectraAverage[,1]>=(min(Cluster$Mass)-
RangeCutoff[1]*1.008664916) &
SpectraAverage[,1]<=(max(Cluster$Mass)+RangeCutoff[2]**1.008664916),]

		Rsq <- RsqMin
		RCheck=rep(0,InterationLimit)
		for(k in 1:InterationLimit){
			##Here we check a range of mzError
			if(IntChoice[k] == TRUE){
				for(l in 1:length(mzError)){
					PredictedData <-
cbind(SpectraPeptideRange[,1],rep(1,length(SpectraPeptideRange[,1])))
					for(m in 1:length(Cluster$Mass)){
						PD <-
MultiNormCalc(SpectraPeptideRange[,1],Cluster$Sequence[m],Cluster$Mass[m],sigma=
SavePnt[2],mzError=mzError[l])
						PredictedData=cbind(PD[,2],PredictedData)
					}


	Soln=qr.solve(PredictedData,SpectraPeptideRange[,2])
					yhat=PredictedData %*% Soln  ##generate fit from
model

					R=RsqCalc(SpectraPeptideRange,yhat)

					if(R>Rsq){
						Rsq=R

	PeptideInformation[PeptideInformation$Cluster==(j),]$mzError=mzError[l]

	PeptideInformation[PeptideInformation$Cluster==(j),]$Sigma=SavePnt[2]
							MatrixAreas=Soln[-c(length(Soln)-
1,length(Soln))]
							ReturnedAreas=rev(MatrixAreas)

	PeptideInformation[PeptideInformation$Cluster==(j),]$Area=ReturnedAreas

	PeptideInformation[PeptideInformation$Cluster==(j),]$NoiseAdjustIntercept=Sol
n[length(Soln)]

	PeptideInformation[PeptideInformation$Cluster==(j),]$NoiseAdjustSlope=Soln[le
```

```
ngth(Soln)-1]

        PeptideInformation[PeptideInformation$Cluster==(j),]$Rsq=Rsq

        PeptideInformation[PeptideInformation$Cluster==(j),]$SpectraCount=i
                                                Residuals=SpectraPeptideRange[,2]-
yhat
                                                mR=mean(Residuals)
                                                lR=length(Residuals)

        PeptideInformation[PeptideInformation$Cluster==(j),]$ResidualMean=mean(Resi
duals)
                                                variance=(1/(lR - 1))*(sum((Residuals
- mR)^2))

        PeptideInformation[PeptideInformation$Cluster==(j),]$ResidualVariance=varianc
e

        PeptideInformation[PeptideInformation$Cluster==(j),]$TIC=TIC

                                                SavePnt[1]=mzError[l]
                                                SavePnt[3]=Rsq
                                                RCheck[k]=Rsq
                                        }
                                        }
                                }

                        ##Start with mzError, here we check a range of Sigma
                        if(IntChoice[k] == FALSE){
                                for(n in 1:length(Sigma)){
                                        PredictedData <-
cbind(SpectraPeptideRange[,1],rep(1,length(SpectraPeptideRange[,1])))
                                        for(o in 1:length(Cluster$Mass)){
                                                PD <-
MultiNormCalc(SpectraPeptideRange[,1],Cluster$Sequence[o],Cluster$Mass[o],sigma=S
igma[n],mzError=SavePnt[1])
                                                PredictedData=cbind(PD[,2],PredictedData)
                                        }


        Soln=qr.solve(PredictedData,SpectraPeptideRange[,2])
                                        yhat=PredictedData %*% Soln  ##generate fit from
model

                                        R=RsqCalc(SpectraPeptideRange,yhat)

                                        if(R>Rsq){
                                                Rsq=R

        PeptideInformation[PeptideInformation$Cluster==(j),]$mzError=SavePnt[1]
```

```
        PeptideInformation[PeptideInformation$Cluster==(j),]$Sigma=Sigma[n]
                                                        MatrixAreas=Soln[-c(length(Soln)-
1,length(Soln))]
                                                        ReturnedAreas=rev(MatrixAreas)

        PeptideInformation[PeptideInformation$Cluster==(j),]$Area=ReturnedAreas

        PeptideInformation[PeptideInformation$Cluster==(j),]$NoiseAdjustIntercept=Sol
n[length(Soln)]

        PeptideInformation[PeptideInformation$Cluster==(j),]$NoiseAdjustSlope=Soln[le
ngth(Soln)-1]

        PeptideInformation[PeptideInformation$Cluster==(j),]$Rsq=Rsq

        PeptideInformation[PeptideInformation$Cluster==(j),]$SpectraCount=i
                                                        Residuals=SpectraPeptideRange[,2]-
yhat
                                                        mR=mean(Residuals)
                                                        lR=length(Residuals)

        PeptideInformation[PeptideInformation$Cluster==(j),]$ResidualMean=mean(Resi
duals)
                                                        variance=(1/(lR - 1))*(sum((Residuals
- mR)^2))

        PeptideInformation[PeptideInformation$Cluster==(j),]$ResidualVariance=varianc
e

        PeptideInformation[PeptideInformation$Cluster==(j),]$TIC=TIC
                                                        SavePnt[2]=Sigma[n]
                                                        SavePnt[3]=Rsq
                                                        RCheck[k]=Rsq
                                        }
                                }
                        }

                        if(k>1){
                                if(RCheck[k-1]==RCheck[k]){
                                        break
                                }
                        }
                }
        }
        ##Use Peak or Peptide for comparisons with in a spectra, Ion to compare across
spectra
        switch(NormalizeMethod[1],
                Peak =
{PeptideInformation$NormalizedArea=PeptideInformation$Area/max(PeptideInformatio
```

```
n$Area)}, ##Normalizes by largest peak looked for
		Peptide =
{PeptideInformation$NormalizedArea=PeptideInformation$Area/
PeptideInformation[PeptideInformation$Sequence==NormalizeMethod[2],]$Area},
##Normalizes bu selected peptide
		Ion =
{PeptideInformation$NormalizedArea=PeptideInformation$Area/TIC}, ##Normalizes by
Total Ion Count/Current
		NoNormalization = {PeptideInformation$NormalizedArea=NA}) ##No
normalization asked for

	TotalPeptideInformation=rbind(TotalPeptideInformation,PeptideInformation)
##Link all generated PeptideInformation data frames together
	}
TotalPeptideInformation=TotalPeptideInformation[-1,]
TotalPeptideInformation
}
################################################
##For removing given nth row of a data
RowRemove <- function(Record, n){
Record[-(seq(n,to=nrow(Record),by=n)),]
}
################################################
##For editing results of peptide search, looking at rsq of fit and predicted area
QuickEditResults <- function(Record,RsqCutoff=.7,AreaCutoff=10){
	OutPut <- Record[Record$Rsq>=(RsqCutoff),]##Filters by R squared values
	OutPut <- OutPut[OutPut$Area>=(AreaCutoff=10),]##Filters for negative/low
areas, nonsensical information in this context
	OutPut
}
################################################
##Looks at .mzXML files
##Must set file destination, use .pdf in FileTitle and Record for fit from model,
##will dump pdf into file containing spectra
ViewRawSpectra=function(FileTitle,FileList,PageRow=3,PageCol=2,SpectraStart=650,S
pectraFinish=1350){
	pdf(file = FileTitle)
	par(mfrow = c(PageRow,PageCol))

	for(i in 1:length(FileList)){
		SpectraImput <- xcmsRaw(as.character(FileList[i]))
		SpectraAverage <-
getSpec(SpectraImput,mzrange=c(SpectraStart,SpectraFinish))

	plot(SpectraAverage[,1],SpectraAverage[,2],type='l',xlab="MZ",ylab="Intensity",
main=paste(FileList[i]))
	}
	dev.off()
	cat("Please remember to shutdown R before opening any large .PDF, thank
you.","\n")
```

```
}
##For Spectra with multiple peptide clusters
##Producing NA titles??##NEEDS REFINEMENT AND CORRECDTION
ViewFitResults=function(SpectraFitRecord,FileTitle,PageRow=3,PageCol=2,SpectraStart
=650,SpectraFinish=1350){
	pdf(file = FileTitle)
	par(mfrow = c(PageRow,PageCol))

	for(i in 1:max(SpectraFitRecord$SpectraCount)){
		SampleSpectra <-
SpectraFitRecord[SpectraFitRecord$SpectraCount==(i),]
		SpectraImput <- xcmsRaw(as.character(SampleSpectra$Spectra[1]))

		SpectraAverage <-
getSpec(SpectraImput,mzrange=c(SpectraStart,SpectraFinish))##Get spectra data

		for(j in 1:max(SampleSpectra$Cluster)){ ##NUmber of clusters is = to
number of pics per spectra we want
			Cluster <- SampleSpectra[SampleSpectra$Cluster==(j),]
			SpectraPeptideRange <-
SpectraAverage[SpectraAverage[,1]>=(min(Cluster$Mass)-1.008664916) &
SpectraAverage[,1]<=(max(Cluster$Mass)+5*1.008664916),]
			if(length(SpectraFitRecord[1,])<16){##for use with flat baseline,
change to if there is only NoiseAdjust
				ModelPrediction <- rep(0,length(SpectraPeptideRange[,1]))
				for(k in 1:length(Cluster$Mass)){
					ModelCurve <-
MultiNormCalc(SpectraPeptideRange[,1],as.character(Cluster$Sequence[k]),Cluster$Mas
s[k],Cluster$Sigma[k],Cluster$mzError[k])
					Curve <- (ModelCurve[,2]*Cluster$Area[k])
					ModelPrediction <- ModelPrediction+Curve
				}
				ModelPrediction <-
ModelPrediction+Cluster$NoiseAdjust[j]
			}
			else{
				ModelPrediction <- rep(0,length(SpectraPeptideRange[,1]))
				for(k in 1:length(Cluster$Mass)){
					ModelCurve <-
MultiNormCalc(SpectraPeptideRange[,1],as.character(Cluster$Sequence[k]),Cluster$Mas
s[k],Cluster$Sigma[k],Cluster$mzError[k])
					Curve <- (ModelCurve[,2]*Cluster$Area[k])
					ModelPrediction <- ModelPrediction+Curve
				}

	BL=Cluster$NoiseAdjustSlope[k]*SpectraPeptideRange[,1]+Cluster$NoiseAdjust
Intercept[k]
				ModelPrediction <- ModelPrediction+BL
			}
```

```
	plot(SpectraPeptideRange[,1],SpectraPeptideRange[,2],type='l',xlab="MZ",ylab="
Intensity")
			title(substr(as.character(Cluster$Spectra[k]),1, 25),line=3)
			title(as.character(Cluster$Peptide[k]),line=2)
			lines(SpectraPeptideRange[,1],ModelPrediction,col='red')
		}

	}
	dev.off()
	cat("Please remember to shutdown R before opening any large .PDF, thank
you.","\n")
}
##############################################
##Looking at the first 3 peak heights (intensies) and
##calculates Riemann sums for the peaks, Must set file destination
##for .mzXML files
PeakAreaHeightScan=function(SpectraFitRecord){
HeightStore=data.frame(SpectraFile=NA,Peptide=NA,Peak1=NA,Peak2=NA,Peak3=NA
,Sum=NA,AUCMinusBaseline1=NA,AUCMinusBaseline2=NA,AUCMinusBaseline3=N
A,AUCTotal=NA)

	for(i in 1:length(SpectraFitRecord[,1])){
		SpectraImput <- xcmsRaw(as.character(SpectraFitRecord$Spectra[i]))

			cat("Loaded",i,"Spectra of",length(SpectraFitRecord[,1]),"\n")

		SpectraAverage <- getSpec(SpectraImput,mzrange=c(650,1350))

		HeightStore[i,1]=as.character(SpectraFitRecord$Spectra[i])
		HeightStore[i,2]=as.character(SpectraFitRecord$Peptide[i])
		PeakEstimate=SpectraFitRecord$Mass[i]+SpectraFitRecord$mzError[i]-
1.008664916  ##Fixed twice

		if(length(SpectraFitRecord[1,])<16){ ##for use with linear baseline
estimate
			for(j in 1:3){
				counter=j*1.008664916
				PeakRange=PeakEstimate+counter
				Peak=SpectraAverage[SpectraAverage[,1]>=(PeakRange-
0.5043325) & SpectraAverage[,1]<=(PeakRange+0.5043325),]

				HeightStore[i,2+j]= max(Peak[,2]) -
SpectraFitRecord$NoiseAdjust[i]

	HeightStore[i,6+j]=CurveAreaCalc(Peak,SpectraFitRecord$NoiseAdjust[i])
			}
		}
		else{
			for(j in 1:3){
				counter=j*1.008664916
```

```
                                PeakRange=PeakEstimate+counter
                                Peak=SpectraAverage[SpectraAverage[,1]>=(PeakRange-
0.5043325) & SpectraAverage[,1]<=(PeakRange+0.5043325),]


        BL=SpectraFitRecord$NoiseAdjustSlope[1]*Peak[,1]+SpectraFitRecord$NoiseA
djustIntercept[1]

                                HeightStore[i,2+j]= max(Peak[,2]) -
BL[which.max(Peak[,2])]
                                HeightStore[i,6+j]=CurveAreaCalc(Peak,BL)
                        }
                }
                HeightStore[i,6]=sum(HeightStore[i,3:5])
                HeightStore[i,10]=sum(HeightStore[i,7:9])
        }
        HeightStore
}
##############################################
##Trapazoidal AUC calc
CurveAreaCalc = function(test,baseline){
        PArea=0

        if(length(baseline)==1){
                baseline=rep(baseline,length(test[,1]))
        }
        if(length(baseline)!=length(test[,1])){
                print('CurveAreaCalc has stopped because baseline and vector with peak
data are of unequal lenght,
                \nbaseline must describe whole area under peak of be one element to
describe flat baseline')
                break
        }
        for(i in 1:(length(test[,1])-1)){
                area=.5*(test[i+1,2]+test[i,2])*(test[i+1,1]-test[i,1])
                area=area-(.5*(test[i+1,1]-test[i,1])*(baseline[i+1]+baseline[i]))
                PArea=PArea+area
        }
        PArea
}


##############################################
##############################################
##############################################
##############################################
##Spectra Simulation Work
set.seed(31851)
PeakSim=function(SimCount,PickCount,Pep,mzError,Sigma,Slope,Intercept,Area,StepBi
nSize=.02,rho=.7,UnifRange=c(0,2),titles,SIS=0,mzR=c(1,5)){
```

```
InfoStore=list()

Mass=rep(0,length(Pep))
for(n in 1:length(Pep)){
Mass[n]=MassCalc(Pep[n])+SIS*1.008664916
}
Mass=sort(Mass)

SpectraData=data.frame(x=seq(min(Mass)-
(mzR[1]*1.008664916),max(Mass)+(mzR[2]*1.008664916),StepBinSize))
mzRange=seq(min(Mass)-
(mzR[1]*1.008664916),max(Mass)+(mzR[2]*1.008664916),StepBinSize)+.5*StepBinSiz
e

for(i in 1:SimCount){
SpectraInfo=data.frame(Simulation=NA,Peptide=NA,Mass=NA,mzError=NA,Sigma=N
A,Area=NA,Slope=NA,Intercep=NA,PeakCount=NA,RemovedCount=NA)
SpectraInfo[1:length(Mass),]=0

AreaCurrent=list()
for(m in 1:length(Pep)){
        lambda=IsotopicCountFFT(Pep[m])
        StorePeakDraws=list()
        Peak=rmultinom(1,PickCount,lambda)
                for(j in 1:length(lambda)){
                        PeakDraws=rnorm(Peak[j],Mass[m]+((j-
1)*1.008664916)+mzError[m],Sigma[m])
                        StorePeakDraws[[j]]=PeakDraws
}
Points=unlist(StorePeakDraws)

CurrentCount=sum(table(cut(Points,breaks=seq(min(Mass)-
(mzR[1]*1.008664916),max(Mass)+(mzR[2]*1.008664916),StepBinSize))))

RawCurrent=as.vector(table(cut(Points,breaks=seq(min(Mass)-
(mzR[1]*1.008664916),max(Mass)+(mzR[2]*1.008664916)+StepBinSize,StepBinSize)))
)
NormalizedCurrent=(RawCurrent/CurrentCount)*(1/StepBinSize)

AreaCurrent[[m]]=NormalizedCurrent*Area[m]
}
TotalAreaCurrent=Reduce('+',AreaCurrent)

##AR(1) Noise
if(rho>0){
Base=rep(10,(length(mzRange)+200))
for(k in 1:(length(Base)-1)){
        Base[k+1]=rho*Base[k]+runif(1,UnifRange[1],UnifRange[2])
}
```

```
AR1Noise=Base[101:(length(Base)-100)]-mean(Base[101:(length(Base)-100)])
##adjust to mean of noise to 0

##Combine AR(1) and peaks
for(l in 1:length(TotalAreaCurrent)){
if(TotalAreaCurrent[l]==0){
        TotalAreaCurrent[l]=AR1Noise[l] ##remove l+100
}
if(TotalAreaCurrent[l]<AR1Noise[l]){
        TotalAreaCurrent[l]=AR1Noise[l]
}
}
}

ModelCurrent = TotalAreaCurrent

if(abs(Slope)>0){
##ABS() since slope can be +/-
Lift = Slope*mzRange+((min(Mass)*-1*Slope)+Intercept)
ModelCurrent = TotalAreaCurrent+Lift
}

SpectraData[2*i-1] <- mzRange
SpectraData[2*i] <- ModelCurrent

SpectraInfo[,1]=i
SpectraInfo[,2]=Pep
SpectraInfo[,3]=Mass
SpectraInfo[,4]=mzError
SpectraInfo[,5]=Sigma
SpectraInfo[,6]=Area
SpectraInfo[,7]=Slope
SpectraInfo[,8]=((Mass*-1*Slope)+Intercept)
SpectraInfo[,9]=PickCount
SpectraInfo[,10]=PickCount-CurrentCount

InfoStore[[i]]=SpectraInfo
}

SumInfo=data.frame(Simulation=NA,Peptide=NA,Mass=NA,mzError=NA,Sigma=NA,A
rea=NA,Slope=NA,Intercep=NA,PeakCount=NA,RemovedCount=NA)
for(i in 1:length(InfoStore)){
SumInfo=rbind(SumInfo,InfoStore[[i]])
}
SumInfo=SumInfo[-1,]

write.csv(SumInfo,titles[1])
write.csv(SpectraData,titles[2])
}
##For csv files
```

```
MultiPeakPeptideScan3=function(FileList,PeptideInformation,SigmaRange,mzErrorRang
e,RangeCutoff=c(1,5),RsqMin=0,NormalizeMethod="NoNormalization",InterationLimit
=100,StartPoint=c(0,.1,0)){

TotalPeptideInformation=data.frame(Peptide=NA,Sequence=NA,Mass=NA,Spectra=NA,
SpectraCount=NA,Cluster=NA,mzError=NA,Sigma=NA,Area=NA,NoiseAdjustSlope=N
A,NoiseAdjustIntercept=NA,Rsq=NA,ResidualMean=NA,ResidualVariance=NA,TIC=N
A,NormalizedArea=NA)

IntChoice = 1:InterationLimit %% 2 == 0

SpectraData <- read.csv(FileList[1])
SpectraNumber=length(SpectraData[1,])/2

if(SpectraNumber <1){
SpectraData <- read.csv(FileList,sep='\t')
print('File in tab delimited form')
}
if(SpectraNumber %% 2 == .5){
SpectraData <- SpectraData[,-1]
SpectraNumber=length(SpectraData[1,])/2
print('File has uneven number of columns, removing first column')
}

for(i in 1:SpectraNumber){

        cat("Loaded",i,"Spectra of",SpectraNumber,"\n")
        TIC <- mean(SpectraData[,2*i])
        SpectraAverage <- cbind(SpectraData[,2*i-1],SpectraData[,2*i])

        for(j in 1:max(PeptideInformation$Cluster)){
                SavePnt=StartPoint
                Cluster <- PeptideInformation[PeptideInformation$Cluster==(j),]
                PeptideInformation[PeptideInformation$Cluster==(j),]$Spectra <- FileList
                SpectraPeptideRange <-
SpectraAverage[SpectraAverage[,1]>=(min(Cluster$Mass)-
RangeCutoff[1]*1.008664916) &
SpectraAverage[,1]<=(max(Cluster$Mass)+RangeCutoff[2]**1.008664916),]

                Rsq <- RsqMin
                RCheck=rep(0,InterationLimit)
                for(k in 1:InterationLimit){
                        ##Here we check a range of mzError
                        if(IntChoice[k] == TRUE){
                                for(l in 1:length(mzError)){
                                        PredictedData <-
cbind(SpectraPeptideRange[,1],rep(1,length(SpectraPeptideRange[,1])))
                                        for(m in 1:length(Cluster$Mass)){
                                                PD <-
MultiNormCalc(SpectraPeptideRange[,1],Cluster$Sequence[m],Cluster$Mass[m],sigma=
```

```
SavePnt[2],mzError=mzError[l])
                                        PredictedData=cbind(PD[,2],PredictedData)
                                    }


        Soln=qr.solve(PredictedData,SpectraPeptideRange[,2])
                                    yhat=PredictedData %*% Soln  ##generate fit from
model

                                    R=RsqCalc(SpectraPeptideRange,yhat)

                                    if(R>Rsq){
                                        Rsq=R

        PeptideInformation[PeptideInformation$Cluster==(j),]$mzError=mzError[l]

        PeptideInformation[PeptideInformation$Cluster==(j),]$Sigma=SavePnt[2]
                                                MatrixAreas=Soln[-c(length(Soln)-
1,length(Soln))]
                                                ReturnedAreas=rev(MatrixAreas)

        PeptideInformation[PeptideInformation$Cluster==(j),]$Area=ReturnedAreas

        PeptideInformation[PeptideInformation$Cluster==(j),]$NoiseAdjustIntercept=Sol
n[length(Soln)]

        PeptideInformation[PeptideInformation$Cluster==(j),]$NoiseAdjustSlope=Soln[le
ngth(Soln)-1]

        PeptideInformation[PeptideInformation$Cluster==(j),]$Rsq=Rsq

        PeptideInformation[PeptideInformation$Cluster==(j),]$SpectraCount=i
                                                Residuals=SpectraPeptideRange[,2]-
yhat
                                                mR=mean(Residuals)
                                                lR=length(Residuals)

        PeptideInformation[PeptideInformation$Cluster==(j),]$ResidualMean=mean(Resi
duals)
                                                variance=(1/(lR - 1))*(sum((Residuals
- mR)^2))

        PeptideInformation[PeptideInformation$Cluster==(j),]$ResidualVariance=varianc
e

        PeptideInformation[PeptideInformation$Cluster==(j),]$TIC=TIC

                                                SavePnt[1]=mzError[l]
                                                SavePnt[3]=Rsq
                                                RCheck[k]=Rsq
```

```
                                        }
                                        }
                                }

                        ##Start with mzError, here we check a range of Sigma
                        if(IntChoice[k] == FALSE){
                                for(n in 1:length(Sigma)){
                                        PredictedData <-
cbind(SpectraPeptideRange[,1],rep(1,length(SpectraPeptideRange[,1])))
                                        for(o in 1:length(Cluster$Mass)){
                                                PD <-
MultiNormCalc(SpectraPeptideRange[,1],Cluster$Sequence[o],Cluster$Mass[o],sigma=S
igma[n],mzError=SavePnt[1])
                                                PredictedData=cbind(PD[,2],PredictedData)
                                        }


        Soln=qr.solve(PredictedData,SpectraPeptideRange[,2])
                                        yhat=PredictedData %*% Soln  ##generate fit from
model

                                        R=RsqCalc(SpectraPeptideRange,yhat)

                                        if(R>Rsq){
                                                Rsq=R

        PeptideInformation[PeptideInformation$Cluster==(j),]$mzError=SavePnt[1]

        PeptideInformation[PeptideInformation$Cluster==(j),]$Sigma=Sigma[n]
                                                        MatrixAreas=Soln[-c(length(Soln)-
1,length(Soln))]
                                                        ReturnedAreas=rev(MatrixAreas)

        PeptideInformation[PeptideInformation$Cluster==(j),]$Area=ReturnedAreas

        PeptideInformation[PeptideInformation$Cluster==(j),]$NoiseAdjustIntercept=Sol
n[length(Soln)]

        PeptideInformation[PeptideInformation$Cluster==(j),]$NoiseAdjustSlope=Soln[le
ngth(Soln)-1]

        PeptideInformation[PeptideInformation$Cluster==(j),]$Rsq=Rsq

        PeptideInformation[PeptideInformation$Cluster==(j),]$SpectraCount=i
                                                        Residuals=SpectraPeptideRange[,2]-
yhat
                                                        mR=mean(Residuals)
                                                        lR=length(Residuals)

        PeptideInformation[PeptideInformation$Cluster==(j),]$ResidualMean=mean(Resi
```

```
duals)
                                                    variance=(1/(lR - 1))*(sum((Residuals
- mR)^2))

        PeptideInformation[PeptideInformation$Cluster==(j),]$ResidualVariance=varianc
e

        PeptideInformation[PeptideInformation$Cluster==(j),]$TIC=TIC
                                                    SavePnt[2]=Sigma[n]
                                                    SavePnt[3]=Rsq
                                                    RCheck[k]=Rsq
                                        }
                                }
                        }

                        if(k>1){
                                if(RCheck[k-1]==RCheck[k]){
                                        break
                                }
                        }
                }
        }
        ##Use Peak or Peptide for comparisons with in a spectra, Ion to compare across
spectra
        switch(NormalizeMethod[1],
                Peak =
{PeptideInformation$NormalizedArea=PeptideInformation$Area/max(PeptideInformatio
n$Area)}, ##Normalizes by largest peak looked for
                Peptide =
{PeptideInformation$NormalizedArea=PeptideInformation$Area/
PeptideInformation[PeptideInformation$Sequence==NormalizeMethod[2],]$Area},
##Normalizes bu selected peptide
                Ion =
{PeptideInformation$NormalizedArea=PeptideInformation$Area/TIC}, ##Normalizes by
Total Ion Count/Current
                NoNormalization = {PeptideInformation$NormalizedArea=NA}) ##No
normalization asked for

        TotalPeptideInformation=rbind(TotalPeptideInformation,PeptideInformation)
##Link all generated PeptideInformation data frames together
        }
TotalPeptideInformation=TotalPeptideInformation[-1,]
TotalPeptideInformation
}
##Random Peptide Generation and Filtering
PeptideGenerator=function(PeptideNumber,PeptideSize,MassFilterStart,MassFilter=2*1.
008664916){

Seq=rep(0,PeptideNumber)
Masses=rep(0,PeptideNumber)
```

```
PepSeqMass=data.frame()
for(z in 1:PeptideNumber){
	pep=rep(0,PeptideSize)
	for (i in 1:PeptideSize){
		pep[i] = AA[ceiling(runif(1,0,20))]
	}
	peps=paste(pep,collapse = "")
	PepSeqMass[z,1]=peps
	PepSeqMass[z,2]=MassCalc(peps)
}
PepSeqMassSeq=PepSeqMass[order(PepSeqMass[,2]),]

Store1=MassFilterStart
Store2='Start'
Spacer=MassFilter
PepSelection=data.frame()
for(i in 2:length(PepSeqMassSeq[,2])){
	if(PepSeqMassSeq[i,2]>=(tail(Store1,n=1)+Spacer)){
		Store1=c(Store1,PepSeqMassSeq[i,2])
		Store2=c(Store2,PepSeqMassSeq[i,1])
	}
}
PepSelection=cbind(Store2,Store1)
PepSelection
}

################################################
################################################
################################################
################################################
##MLE estimation delta and sigma
#############
##probability matrix for point x_i being in peak k
AlphaIK=function(MuNot,Delta,Sigma,lambda,x){
AlphaIK=data.frame()  ##N*K matrix
AlphaIKDr=data.frame()
for(k in 1:length(lambda)){
	MuK=MuNot+((k-1)*1.008664916)+Delta
	for(i in 1:length(x)){
		AlphaIK[i,k]=lambda[k]*dnorm(x[i],MuK,Sigma)
}}
for(i in 1:length(x)){
for(k in 1:length(lambda)){
##Do not over write AlphaIK, causes rounding errors
AlphaIKDr[i,k]=AlphaIK[i,k]/sum(AlphaIK[i,])
}}
AlphaIKDr
}
#############
##Delta Hat
```

```
DeltaHat=function(x,h,AlphaIK,MuNot){ ##hi must be normalized, indepenant of AIK
ExLk = AlphaIK[1,]
HiAlphaIK=AlphaIK
for(k in 1:length(AlphaIK[1,])){
for (i in 1:length(x)){
        HiAlphaIK[i,k] <-h[i]*AlphaIK[i,k]
}
ExLk[k] <- sum(HiAlphaIK[,k])
}
##Essentially we calculate a MLE for lambda from Aik, then get this expected value of
lambda^k^hat
Exx=0
for(k in 1:length(ExLk)){
Exx <- Exx + ExLk[k]*k
}
DoubleSum=0
for (i in 1:length(x)){
                DoubleSum <- DoubleSum+h[i]*x[i]
}
mzErrorHat = DoubleSum - MuNot - Exx*1.008664916 + 1.008664916
mzErrorHat
}
#############
##SigmaHat
SigmaHat=function(x,h,AlphaIK,MuNot,DeltaH){
SH=AlphaIK
for(i in 1:length(x)){
for(k in 1:length(AlphaIK[1,])){
SH[i,k]=AlphaIK[i,k]*((x[i]-MuNot-k*1.008664916+1.008664916-DeltaH)^2)
}
}
SH2=x
for(i in 1:length(x)){
SH2[i]=h[i]*sum(SH[i,])
}
Sig2Hat=sum(SH2)
SigHat=sqrt(Sig2Hat)
SigHat
}
#############
##EMFunction for Delta and sigma hat, Data in form of MZ,Intensity (x,h)
##Tolerance should be based on step-size in x
EMDeltaSigma=function(MuNot,Delta,Sigma,lambda,Data,Tolerance=.0001){
##Data preprocessing/smoothing
#remove gross baseline lift
m=(Data[length(Data[,1]),2] - Data[1,2])/(Data[length(Data[,1]),1] - Data[1,1])
b=(Data[1,2])-(m*Data[1,1])
TrapArea=m*Data[,1]+b
ModData=Data[,2]-TrapArea
#Remove negative values, our assumption of the Gaussian basis of the peaks is violated if
```

```
we have negative values
AbsModData=abs(ModData)

DataNorm=AbsModData/sum(AbsModData)  ##Input data before normalization
for(z in 1:20){
AIK=AlphaIK(MuNot=PepMass,Delta=Delta,Sigma=Sigma,lambda=PepIsotope,x=Data[
,1])
DHp=DeltaHat(x=Data[,1],h=DataNorm,AlphaIK=AIK,MuNot=PepMass)
SHp=SigmaHat(x=Data[,1],h=DataNorm,AlphaIK=AIK,MuNot=PepMass,DeltaH=DHp)
if(abs(Delta-DHp)<Tolerance && abs(Delta-DHp)<Tolerance)break ##Tolerance, can be
broken up into vector for ease of use
Delta=as.numeric(DHp)
Sigma=as.numeric(SHp)
}
Report=c(Delta,Sigma)
Report
}
```