# Intelligent differential ion mobility spectrometry (iDMS): A machine learning algorithm that simplifies optimization of lipidomic differential ion mobility spectrometry parameters

Xun Xun Shi

Thesis submitted to the University of Ottawa
In partial fulfillment of the requirements for the
MASc degree in Biomedical Engineering

School of Electrical Engineering and Computer Science

Faculty of Engineering

University of Ottawa

# Acknowledgements

A few years ago, Dr. Steffany Bennett gave a lecture in my seminar class, passionately introduced me to the roles of lipids, how they help shape and define us for who we are. Her passion sparked my interest in lipidomics and inspired me to pursue a career path driven by my passion rather than defined by my past. Like lipids, she definitively shaped me into the person that I am today. Dr. Bennett was not only my thesis supervisor but both a friend and a mentor. At difficult times, she has always believed me in my potential, even though my self-doubts.  This work would not have been completed without her drive, guidance, and insights.

Furthermore, I would like to thank my co-supervisors, Dr. Theodore Perkins and Dr. Mathieu Lavallée-Adam. I am beyond lucky to have both of your wisdom and patience to guide me through this chapter of my life. Both of your intuition and knowledge are impeccable, and I hope to get there one day. None of my work could have been possible without those weekly meetings, and I will greatly miss all the silly dad jokes and great discussions we all had during those meetings.

I have been fortunate to work with three labs filled with bright and driven individuals. I would like to explicitly thank my senior mentor, Graeme Taylor, for helping me transition into the world of lipidomics.  I want to thank Thao Nguyen for running the many experiments, providing me with data, and relentlessly teaching me the ins and outs of mass spectrometry and DMS.

I would also like to thank the Ottawa Hospital Research Institute and the MATRIX CREATE training program for providing the resources and funding for this work to be available. I would also like to thank AVANTI lipids for supplying with the standards.

Additional gratitude to my undergraduate professor, Dr. Thibault Jules, for sparking my interest in neural networks. I would like to also thank other amazing professors that I've encountered during my undergraduate studies, Dr. Xudong Cao and Dr. Andre

Tremblay, for providing me with my first two research experiences and training me with critical thinking skills for my master's project.

I would like to thank Guillaume for being the best partner and supporting me through every conference talk and deadline.

Lastly, I owe the uttermost gratitude to my mom and dad. Growing up, my mom worked many retail jobs and opened a business to support my dad for his graduate studies. Even after working 16 hours shifts, she has never missed any of my birthdays or graduations. While my dad gifted me the inspiration to pursue research, my mom gifted me the privilege to do so.

# Abstract

Glycosphingolipids such as α- and β-glucosylceramides (GlcCers) and α- and β-galactosylceramides (GalCers) are stereoisomers differentially synthesized by gut bacteria and their mammalian hosts in response to environmental insult. Thus, lipidomic assessment of α- and β-GlcCers and α- and β-GalCers is crucial for inferring biological functions and biomarker discovery. However, simultaneous quantification of these stereoisomeric lipids is difficult due to their virtually identical structures. Differential mobility mass spectrometry (DMS), as an orthogonal separation to high performance liquid chromatography used in electrospray ionization, tandem mass spectrometry (LC-ESI-MS/MS), can be used to separate stereoisomeric lipids. Generating LC-ESI-DMS-MS/MS methods for lipidomic analyses is exceedingly difficult demanding intensive manual optimization of DMS parameters that depend on the availability of synthetic lipid standards. Where synthetic standards do not exist, method development is not possible. To address this challenge, I developed a supervised in silico machine learning approach to accelerate method development for ion mobility-based quantification of lipid stereoisomers.  I hypothesized that supervised neural network models could be used to learn the relationships between lipid structural characteristics and optimal DMS machine parameter values thereby reducing the total number of empirical experiments required to develop a DMS method and enabling users to “predict” DMS parameters for analytes that lack synthetic standards. Specifically, this thesis describes a supervised learning approach that learns the relationship between two DMS machine parameter values (separation voltage and compensation voltage) and two lipid structural features (N-Acyl chain length and degree of unsaturation). I describe here, iDMS, an algorithm that was trained on 17 lipid species, and can further simulate results of DMS manual method development and suggest optimal parameter values for 47 lipid species. This approach promises to greatly accelerate the development of assays for the detection of lipid stereoisomers in biological samples.

# Table of Contents

# List of Abbreviations

| | |
|---|---|
| $\phi\ (z)$ | Activation Function |
| $\beta$ | Decaying Parameters in Momentum Optimizer |
| $\beta_1, \beta_2$ | Decaying Parameters in ADAM |
| $\varepsilon$ | Smoothing Parameter in ADAM |
| $\varepsilon r_j$ | Noise |
| $u_{Glc}$ | Mean COV of GlcCer |
| $u_{Gal}$ | mean COV of GalCer |
| $\sigma_{Glc}$ | Standard Deviation of GlcCer |
| $\sigma_{Gal}$ | Standard Deviation of GalCer |
| $\bar{\sigma}_{Glc}$ | Standard Deviation of GlcCer Averaged Across Lipid Species |
| $\bar{\sigma}_{Gal}$ | Standard Deviation of GalCer Averaged Across Lipid Species |
| α-GalCers | α-linkage Galactosylceramide |
| α-GalSph | α-linkage Galactosylsphingosine |
| α-GlcCers | α-linkage Glucosylceramide |
| α-GlcSph | α-linkage Glucosylsphingosine |
| α-HexCer | α- linkage Hexosylceramides |
| β-GalCers | β-linkage Galactosylceramide |
| β-GalSph | β-linkage Galactosylsphingosine |
| β-GlcCers | β-linkage Glucosylceramide |
| β-GlcSph | β-linkage Glucosylsphingosine |
| β-HexCer | β-linkage Hexosylceramides |
| A | Area of Ionogram |
| ADAM | ADAptive Momentum Estimation |
| APC | Antigen Presenting Cells |
| c | Constant Value |
| C | Chain Length |
| Cer | Ceramide |
| COV | Compensation Voltage |
| DMS | Differential Ion Mobility Spectrometry |
| ESI | Electrospray ionization |
| HILIC | Hydrophilic Interaction Chromatography |
| HPLC | High Perfomance Liquid Chromatography |
| iNKT | Invariant natural killer T cells |
| $I$ | Measure intensity |
| $\tilde{I}$ | Normalized Intensity from Riemann Sum |
| $\tilde{I}$ | Normalized Intensity from gaussian features |
| MAE | Mean Absolute Error |
| MSE | Mean Squared Error |
| MSLE | Mean Squared Log Error |
| MS/MS | Tandem Mass Spectrometry |
| $m$ | Momentum Vector Term |
| n | Learning Rate |
| LC | Liquid Chromatography |
| ReLU | REctifier Linear Unit |
| RF | Radio Frequency |
| SCDase | Sphingolipid Ceramide N-deacylase |

| | |
|---|---|
| SV | Separation Voltage |
| t | Iteration Number |
| T | Neuron Threshold |
| U | Degree of Unsaturation |
| V | Volts |
| $w_i$ | Weights of a Neuron for Feature i |
| $x_i$ | Input to a Neuron for Feature i |
| X | Instance of a Variable/Feature from the Dataset |
| $y_{j,i}$ | Output of the $j^{th}$ Neuron from Layer i |
| z | Summed Input Towards a Neuron |

# List of Tables

# List of Figures

# List of Supplemental Figures

# Chapter 1: Introduction

## 1.1 Differential ion mobility spectrometry (DMS) enables separation and quantification of lipid stereoisomers

### 1.1.1. Glycosphingolipid structure dictates biological function yet standard lipidomic methodologies fail to separate molecular species

Glycosphingolipids are a class of membrane sphingolipids composed of a sugar head group attached to a ceramide molecule through either α-glycosidic or β-glycosidic bonds (Figure 1). The ceramide molecule is constructed from a sphingoid base and an N-acyl hydrocarbon (Figure 1) [1]. Simple neutral glycosphingolipids are defined at the molecular level by (a) their sugar headgroup (glucose or galactose), (b) the α or β glycosidic linkage, (c) their sphingoid base, and (d) the length and degree of unsaturation of their N-acyl hydrocarbon (Figure 1). When the N-acyl hydrocarbon chains are missing and only the backbone is present, these lipids are referred as the sphingoid base (Figure S1).

These features confer distinct biological functions as described in Figure 2. For example, β-Glucosylceramide (β-GlcCers) are involved in regulating neuronal axonal growth [2] while β-Galactosylceramide (β-GalCers) are synthesized by oligodendrocytes and are enriched in myelin in both the central and peripheral nervous system [3] (Figure 2A). Physiological membrane concentrations are dependent on the homeostasis of the synthesis, degradation, and regulation of these lipid stereoisomers [4]. For example, the genetic determinants of Gaucher Disease are mutations in the *GBA* gene, which encodes the β-glucocerebrosidase protein that cleaves the glucose moiety from GlcCers and glucosylsphingosine (GlcSph) but not the galactose moiety from GalCers or galactosylsphingosine (GalSph) [5]. When undesired amounts of insoluble GlcCers or GlcSph accumulate in cellular lysosomes, organelle (and cellular) membrane structure and function are disrupted resulting in Gaucher pathology [5]. Moderate accumulation enhances risk of Parkinson's Disease [6]. Conversely, mutations in the GALC gene,

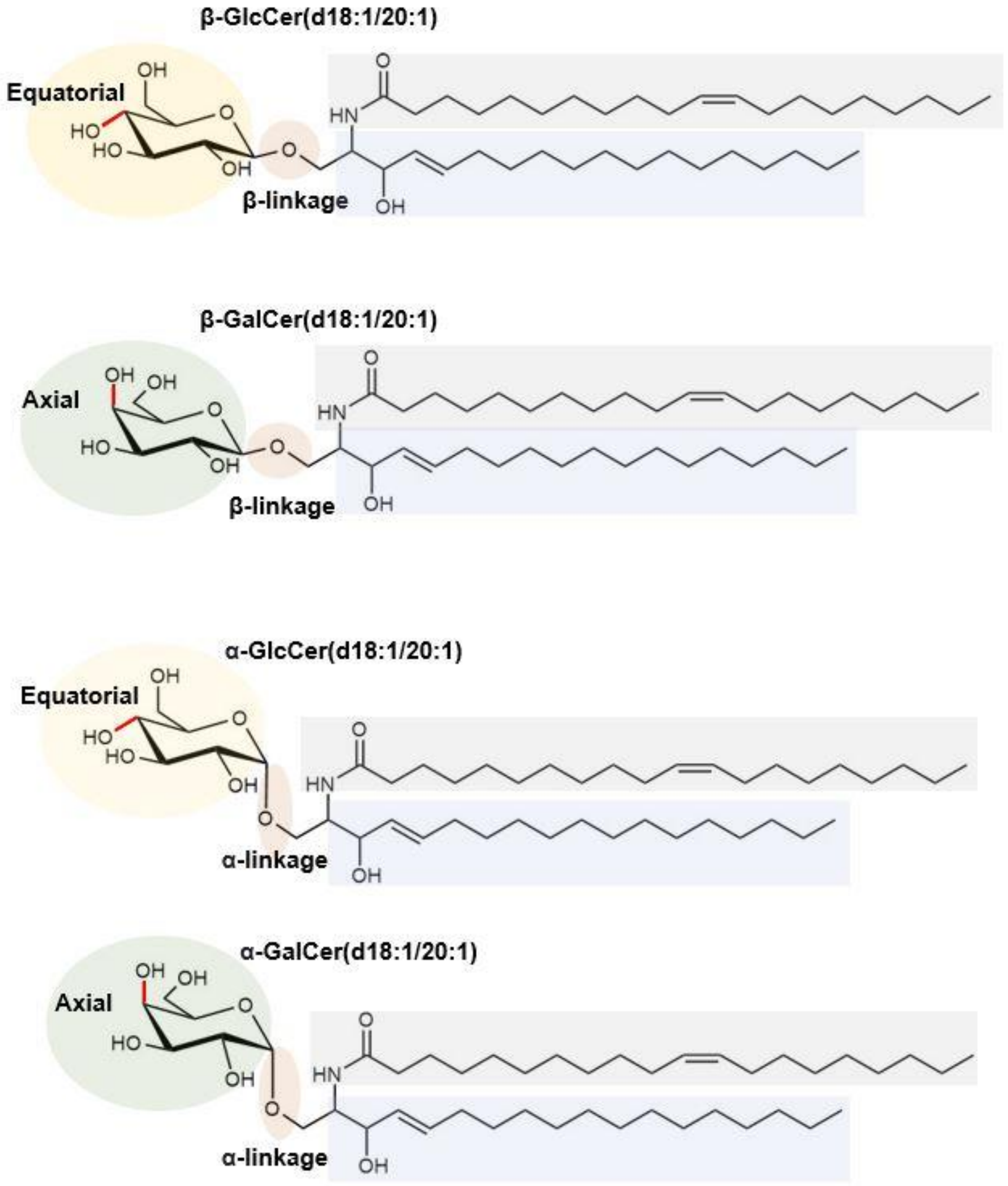


**Figure 1. Structures of β-GlcCer(18:1/20:1), β-GalCer(18:1/20:1), and α-GlcCer(18:1/20:1), and α-GalCer(18:1/20:1).**The glucose headgroup is denoted in yellow. The galactose headgroup is denoted in green. The position (equatorial or axial) of the hydroxyl group that distinguishes glucose from galactose is indicated in red. The glycosidic linkage is denoted by the salmon shaded area. The blue shaded area is the sphingoid (d18:1) base. The fatty N-acyl hydrocarbon chain (20:1) is denoted in grey. Note in the sphingolipid nomenclature β-GlcCer(18:1/20:1), β refers to the glycosidic linkage. Glc defines the sugar head group. Cer refers to a lipid with a sphingoid base and an N-acyl chain. d18:1 defines the sphingoid base, specifically sphingosine which is a sphingoid base with 18 carbons and one degree of unsaturation. 20:1 defines an N-acyl-linked hydrocarbon of 20 carbons and one degree of unsaturation.

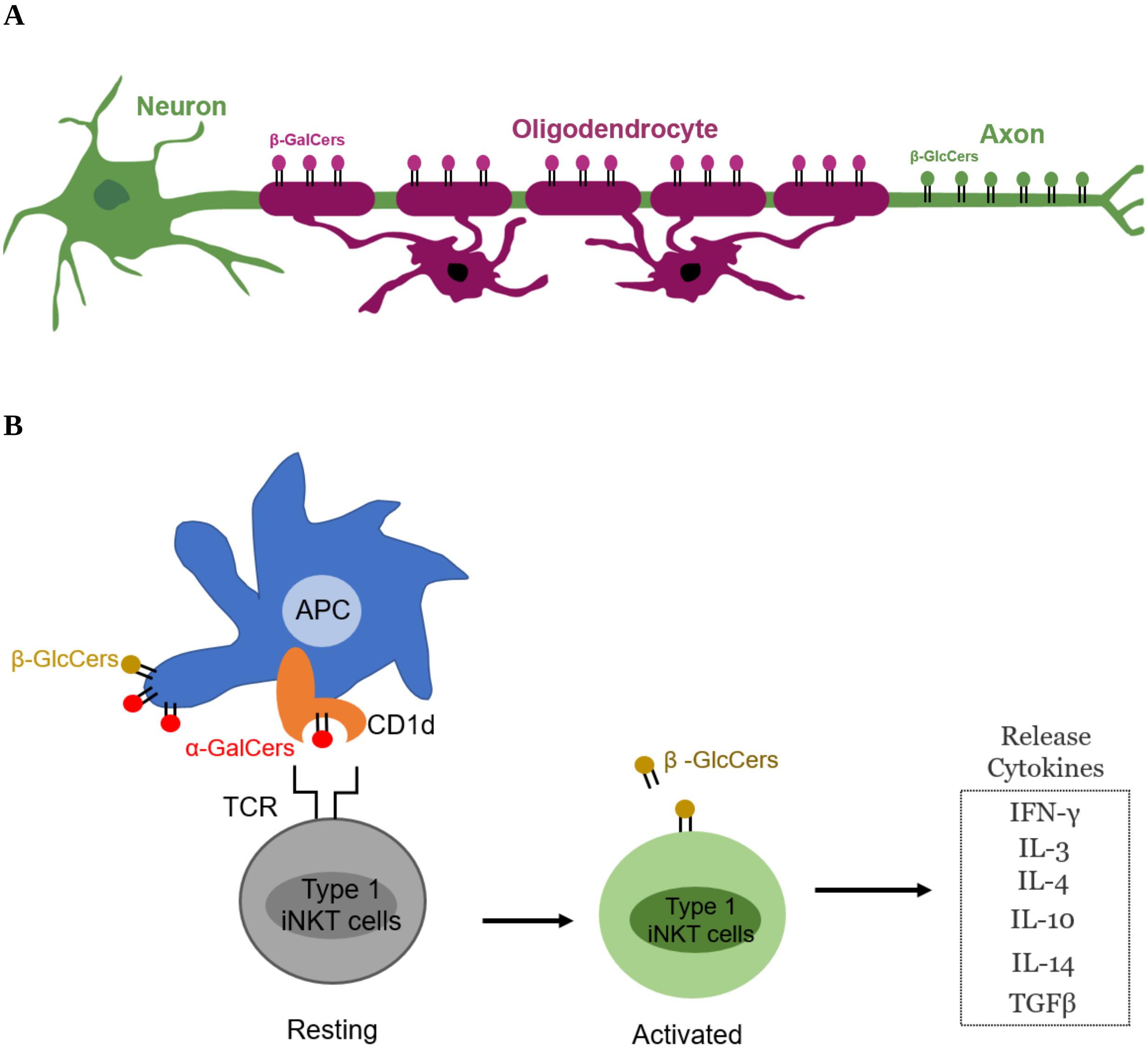


**Figure 2. Select biological functions of α- and β-glycosphingolipids.(A)** β-GlcCers play critical roles in maintaining axonal function [2]. β-GalCers, synthesized by oligodendrocytes, are primary components of myelin that insulate neuronal axons [3]. **(B)** Antigen presenting cells (APCS) synthesize both α-GalCers and α-GlcCers. When presented by the CD1, α-GalCers trigger activation of iNKT cells [7] .The mechanisms by which α-GlcCers activate iNKT cells are unclear [8]; however activated iNKT cell secrete β-GlcCers (a marker of their activation) and cytokines [7,8].

encoding for galactosylceramidase, can lead to the toxic accumulation of GalCers and GalSph (also known as psychosine), resulting in the gradual loss of myelin and the deterioration of neural functions [4]. Unlike the β-stereoiosomers, α-GlcCers and α-GalCers are synthesized by antigen-presenting cells (APCs) and promote activation of Type 1 invariant natural killer T (iNKT) cells when presented with CD1d. Activated iNKT cells and increase synthesis of β-GlcCers to sustain activation and enhance immunity [7,8] (Figure 2B). With such subtle differences in structure conferring functional diversity and cellular specificity, it is essential that researchers be able to separate and quantify all molecular species of α-GlcCers, β-GlcCers, α-GalCers, β-GalCers to infer biological function.

Lipidomic approaches enabling quantification of neutral glycosphingolipid stereoisomers would enable a nuanced investigation of sphingolipid pathobiology. Due to the near-identical structural between the glycosphingolipids isomers, each lipid pair share the same mass to charge (m/z) ratio and retention time and cannot be separated by high-performance liquid chromatography (LC) and tandem mass spectrometry (MS/MS), as shown on Figure 3. However, Xu et al have previously shown that DMS can be used as an orthogonal separation to high-performance liquid chromatography (LC) in electrospray ionization (ESI) tandem mass spectrometry (MS/MS) lipidomic approaches to separate β-GlcCers and β-GalCers at the molecular level [9]. However, optimizing the necessary machine parameters for quantification is a time-consuming, labor-intensive process that requires access to chemical standards. Thus, it remains challenging to identify and quantify α-Hexosylceramides (α-GlcCers and α-GalCers) and β-Hexosylceramides (β-GlcCers and β-GalCers) and thus these species are often quantified as mixtures (referred to as α - or β-HexCers). Moreover, it is currently not possible to optimize existing orthogonal methodologies for species where synthetic standards do not exist. To address these challenges, this thesis focuses on providing a supervised *in silico* machine learning approach to aid in the optimization of the DMS parameters required to separate and quantify neutral glycosphingolipids (HexCers). Here, I validate this approach using neutral β-HexCers; however, this supervised machine learning approach can theoretically be trained to aid in the optimization of any lipid stereoisomer defined at the molecular level by a given backbone and fatty acid constituents.

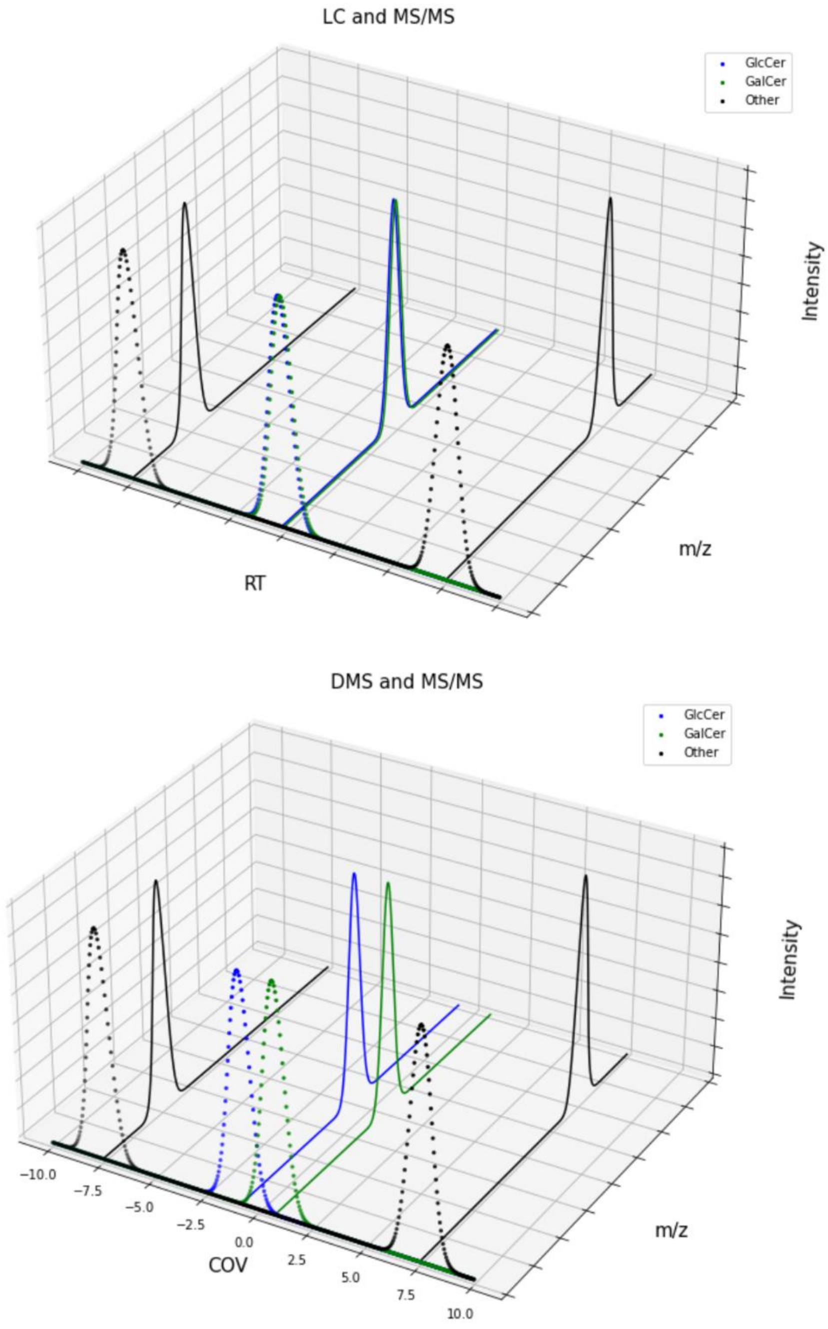


**Figure 3. Glycosphingolipids pairs can not be separated by LC-MS/MS but can be separated by DMS-MS/MS.Top)** Glycosphingolipids pairs share the same retention time (RT) and m/z ratio and cannot be separated using LC-MS/MS. **Bottom)** Glycosphingolipids pairs can be separated at different compensation voltage (COV) and can be separated using DMS.

### 1.1.2. Separation techniques for quantifying β-GlcCer and β-GalCer stereoisomers

Traditional targeted lipidomics studies involve reverse phase LC coupled to ESI and tandem MS to separate target analytes based on their unique mass to charge ratios (m/z) and hydrophobicity. However, due to the virtually identical structure of β-GlcCers and β-GalCers, HexCers cannot be adequately separated by reverse-phased-LC-ESI-MS/MS as they share virtually identical column retention times. GlcCer and GalCer subclasses can be separated by TLC plates coated in borate [10] but molecular species cannot be resolved. Thus TLC can be used for qualitative comparison but suffers from low reproducibility and cannot identify species at a molecular level[9].

Normal phase high performance liquid chromatography (HPLC) can separate GlcCers and GalCers with [13,14] and without derivatization [15,16] by exploiting the interaction of the hydrophilic head group with the nonpolar mobile phase [13]. In the normal phase LC method proposed by Kaye and Ullman, cerebrosides must first be isolated by silica gel chromatography and further synthesized into perbenzoylated derivatives of GlcCer and GalCer [11]. This process is labor-intensive and is prone to quantitative error. Kaye and Ullman further improved sensitivity by adding sphingolipid ceramide N-deacylase (SCDase) to hydrolyze the perbenzoylated derivatives of GlcCers and GalCers into their constituent sphingosine backbone and N-acyl fatty acids. However, SCDase also undesirably hydrolyzes other lipids which confounds simultaneous quantification based on the fatty acid quantities given contamination by other lipids [9]. Although recent improvement of the normal phase LC method has eliminated the need for derivatization, this revision is still confounded by ion suppression of specific ion pairs. For instance, a high abundance of the M+2 isotope of GlcCer(d18:1/24:1) or GalCer(d18:1/24:1) can mask low abundance signal of GlcCer(d18:1/24:0) or GalCers(d18:1/24:0) [9].

Two methods have emerged as the primary means of quantifying glycosphingolipid stereoisomers: DMS and Hydrophilic Interaction Chromatography (HILIC), a specific type of normal phase LC that uses a more hydrophobic solvent. The latter method has demonstrated promising results for separating α and β anomers [15]. However, HILIC still does not completely separate isobaric lipid species, notably if abundances vary in log magnitude differences [9]. By contrast, DMS can efficiently resolve these stereoisomers [9].

DMS sequentially separates different ion clusters in time and space thus temporarily restricting access of one stereoisomer to the mass spectrometer while enabling passage of the other based on their nonlinear electric field transport characteristics [16]. Figure 4 provides a schematic of this process. Briefly, the DMS cell comprises two parallel plates. A transport gas is applied at the entrance to assist ions in traveling through the cell. As analytes pass through the cell, they are exposed to an asymmetric radio frequency (RF) waveform that alternates between high and low fields but at a constant, integrated area of voltage/time [17]. The high field RF wave radially forces the ions towards one plate (electrode) while the low field, applied at the opposite polarity, reverses their trajectory, dragging the ions back towards the opposing plate (electrode). The applied field causes the ions to travel in a zig-zag pattern radially with deviation dependent on the alternation time. The waveform amplitude is referred to as the separation voltage, and the temporal wavelength of the applied field dictates how long the ion travels in one radial direction as shown on Figure 4. To ensure that the target ion trajectories do not deviate away from the center path thus can exit the DMS cell and enter the mass spectrometer while the paired stereoisomer is directed away from the DMS orifice, a direct current voltage called compensation voltage (COV) is applied. COV counteracts the differential mobility of a given ion cluster ensuring it (but not other ion clusters in the cell) reaches and passes through the DMS exit orifice. Thus, the COV is the offset potential between the two plates (electrodes). The optimal combination of SV and COV allows only the target analyte to exit the DMS cell and enter the mass spectrometry for detection while deflecting other isobaric analytes. By oscillating SV and COV, each stereoisomer is directed through the DMS exit orifice over the retention time range, enabling detection [17].

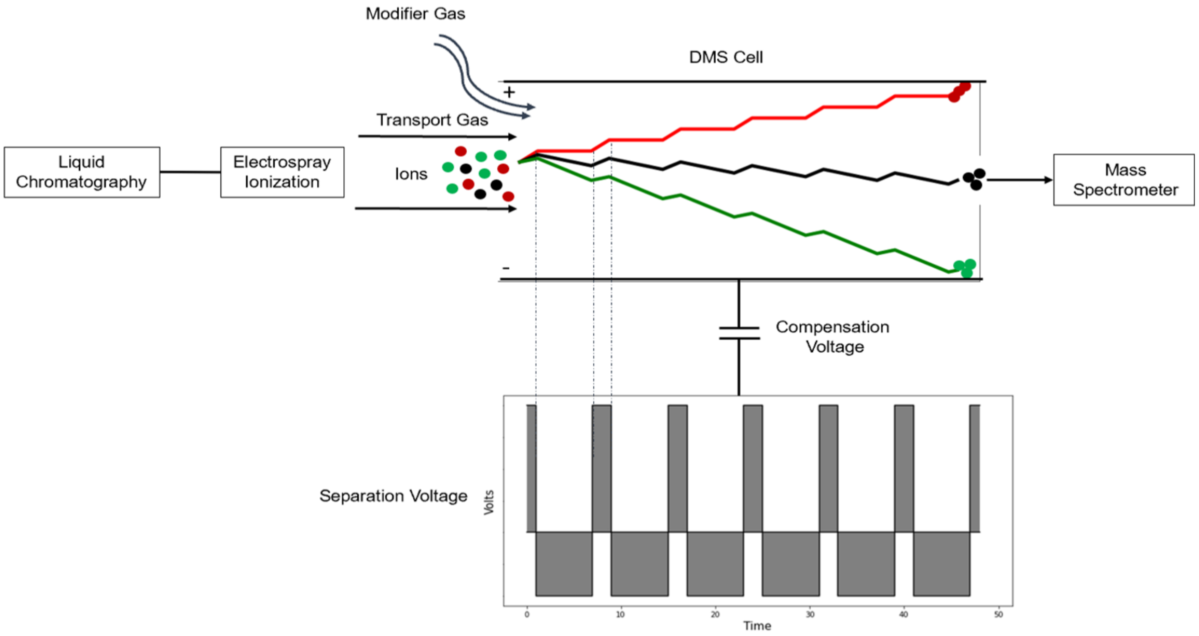


**Figure 4.Schematics of DMS.**Lipids are separated by LC and then undergo ESI. Three representative ionized molecular lipid species, shown as red, green and black circles, are transported through into the DMS cell in gaseous phase, and cluster under the influence of on the pre-determined concentration of modifier gas and according to electric field. As ion clusters pass through the DMS cell, they are exposed to alternating high and low field asymmetric radio frequencies referred to as separation voltage (SV). The temporal wavelength (depicted in lower box in gray) of the applied field dictates how long the ion travels in one radial direction (dotted lines) causing the ions to travel in a zig-zag pattern. A direct current potential called compensation voltage (CoV) is further applied to further correct the ion paths and direct the target clusters of stereoisomers out of the DMS cell and into the MS. CoV exploits the differential mobility of a given ion cluster ensuring it (but not other ion clusters in the cell) reaches (and passes through) the DMS exit orifice. Thus, SV separates isobaric stereoisomer clusters while CoV directs target clusters to the DMS exit orifice. The target lipid, shown as black circles, successfully travels through and exit the DMS cell and into the MS for detection. Other lipid isomers are not able to transverse through the DMS cell thus are never detected by the mass spectrometer. By oscillating SV and CoV over retention time windows specific for each stereoisomer, both pairs are sequentially directed through DMS exit over the retention time range, enabling detection and quantification.

### 1.1.3 DMS machine parameters and neutral glycosphingolipid features affects separation

There are six DMS machine parameters and two structural properties of neutral glycosphingolipids that must be modelled to optimize stereoisomer separation (Figure 5). The optimal parameter combination is unique for each lipid stereoisomer pair and is determined through laborious manual tuning of the machine parameters using pairs of synthetic standards. During method development, one must optimize SV, COV, DMS resolution enhancement, DMS offset, modifier gas concentration, and DMS cell temperature depending on N-acyl hydrocarbon chain length and degree of unsaturation (Figure 5). SV and COV have been defined above. DMS resolution enhancement controls the ion cluster residence time in the cell by altering the transport gas flow between the DMS orifice inlet and exit [18]. DMS offset changes the voltage differential between the electrodes at the orifice exit to the mass spectrometer which consequentially regulates the injection rate of the ion from the DMS cell exit into the mass spectrometer [18]. Thus, the DMS offset tunes the transmission of ions out of the DMS cell. Choice and concentration of modifier gas concentration as well as DMS temperature each impact on the clustering chemistry [18]. Each of these parameters are further dependent on the N-acyl chain length and degree of unsaturation of the lipids. Manual optimization can only alter one parameter at a time rendering method development labor-intensive, challenging, and dependent on availability of synthetic lipids.

The current DMS method development process requires a tedious investigation on how each DMS machine parameter combination affects the selectivity and sensitivity of separating individual β-GlcCers and β-GalCers. Before sequential optimization of the machine parameters, costly pure standards of these lipid species must be first synthesized for method development. During optimization, for each combination of the four machine parameters of DMS resolution enhancement, temperature, DMS offset, modifier gas concentration, experimental designs must be performed sequentially [9]. This process is initiated by directly injecting each lipid (each molecular species of β-

GlcCer or β-GalCer) in a DMS-ESI-MS/MS experiment, ramping the COV values at every

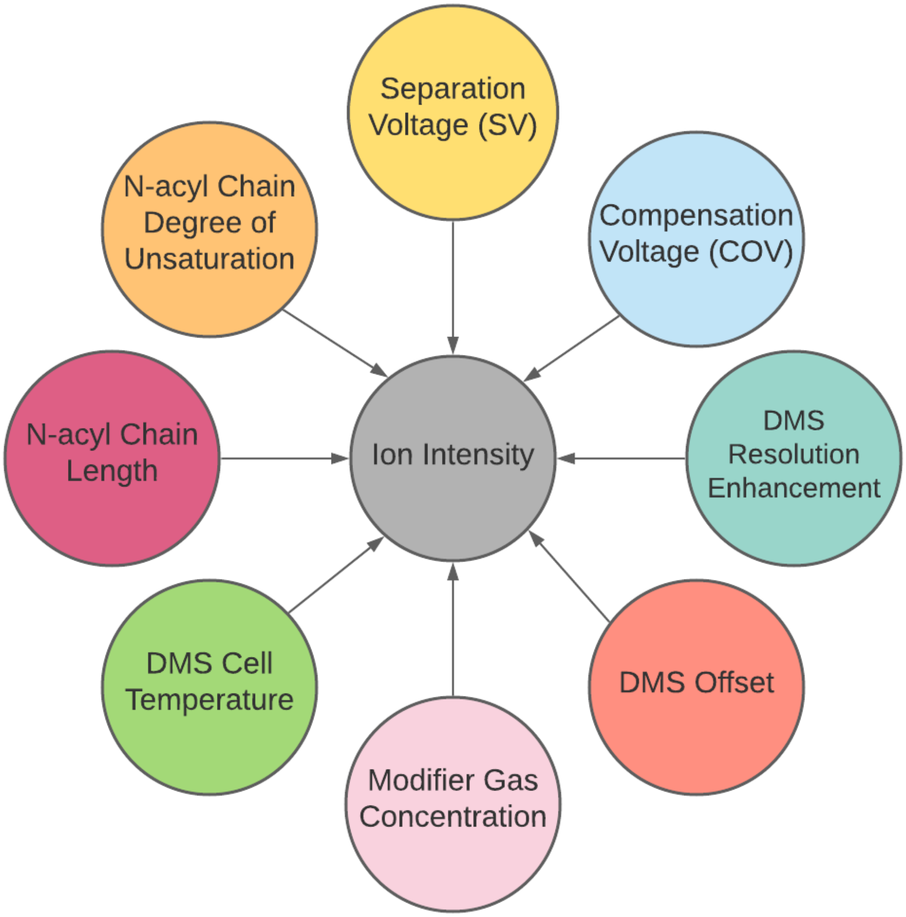


**Figure 5. Six DMS machine parameters and two lipid physiochemical features affect the DMS separation and subsequent ion intensity.**

possible SV range and monitoring the ion intensity output while keeping all other features constant. The maximal SV value is specific to the instrument but is limited to under 4100 V in this case. The optimal SV/COV combination will yield two ionograms with high ion intensity but significantly different peak COV, indicating excellent selectivity of one lipid species over the other. An example of this output is presented in Figure 6. Once optimal SV and COV are found, each of these experiments needs to be repeated to optimize the four other DMS parameter settings for each stereoisomer pair. This lengthy process highlights the challenges associated with selecting the six different DMS machine parameter choices to maximize the selectivity and sensitivity. Simply put, manual exploration of all possible DMS combinations is exhausting and would require the generation of over 3000 different ionograms for each stereoisomer pair. Therefore, developing an in silico supervised learning methodology to augment the manual experiment will save time and cost and allow for DMS method development for stereoisomers when synthetic standards are unavailable. Thus, development of a neural network capable of accelerating this process *in silico* by training on a set number of tuning experiments would be of considerable advantage.

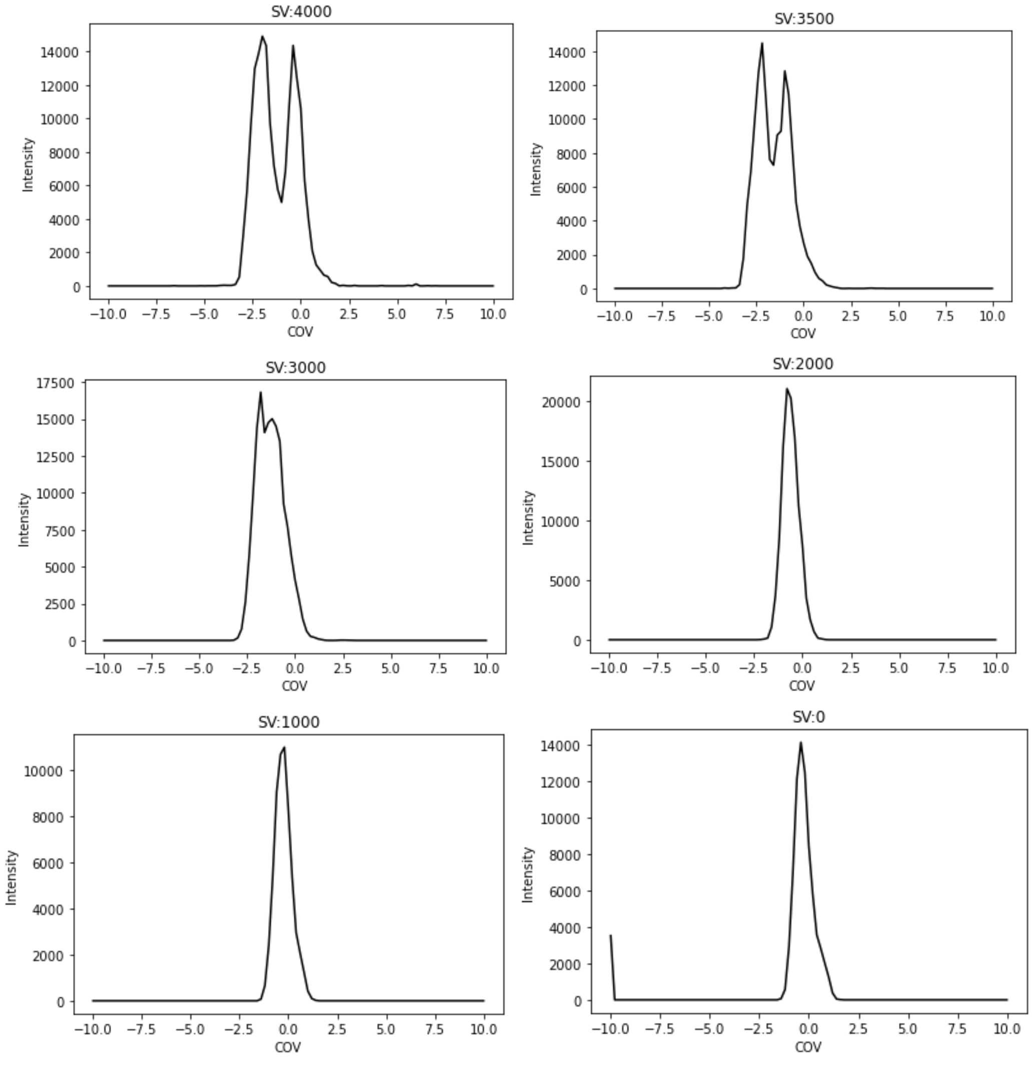


**Figure 6. Ionogram response of β-HexCer(d18:1/16:0) (β -GlcCer(d18:1/16:0) and β-GalCer(d18:1/16:0)) samples at varying SV and COV values.**

## 1.2. Neural Networks

### 1.2.1 Network history and structure

The first concept of an artificial neural network was proposed by neurophysiologist Warren McCulloch and mathematician Walter Pitts in 1943 [19]. In 1957, Frank Rosenblatt expanded this concept by introducing the perceptron, which uses a layer of threshold linear units (TLU) to model the relationship between the input and output [20]. The TLU can take in multiple numerical inputs, each multiplied by a unique associated weight. Thus, the summed product can be calculated as follows:

$$z = \sum_i x_i w_i$$

Here $z$ is the summed of the weighted output, and $x_i$ and $w_i$are the input and weight for feature $i$.

The TLU unit take the summed product into a step function. The step function is a simple activation function. It can activate the neuron if the summed input products are greater than a threshold value $T$[21]. The function is displayed in the top left panel of Figure 7. The summed product can be inputted into the step function as follows:

$$f(z) = \begin{cases} 0 & z < T \\ 1 & z \geq T \end{cases}$$

Here $f(z)$ is the output of the linear unit, and $T$ is the threshold value.

A TLU is analogous to a single neuron in the brain. A neuron receives different simultaneous inputs that can accumulate. Each input has a different influence on the neuron, which is encapsulated in the value of the assigned weights. Indeed, some inputs may be activating for the neuron, having positive weights, and some may be repressing, having negative weights. When the total weighted inputs, analogous to the environmental stimulus received by the neuron, exceed the neuron threshold, the neuron will become excited. This mechanism is similar to the regulation of action potentials (aka neuronal “firing”) of biological neurons.

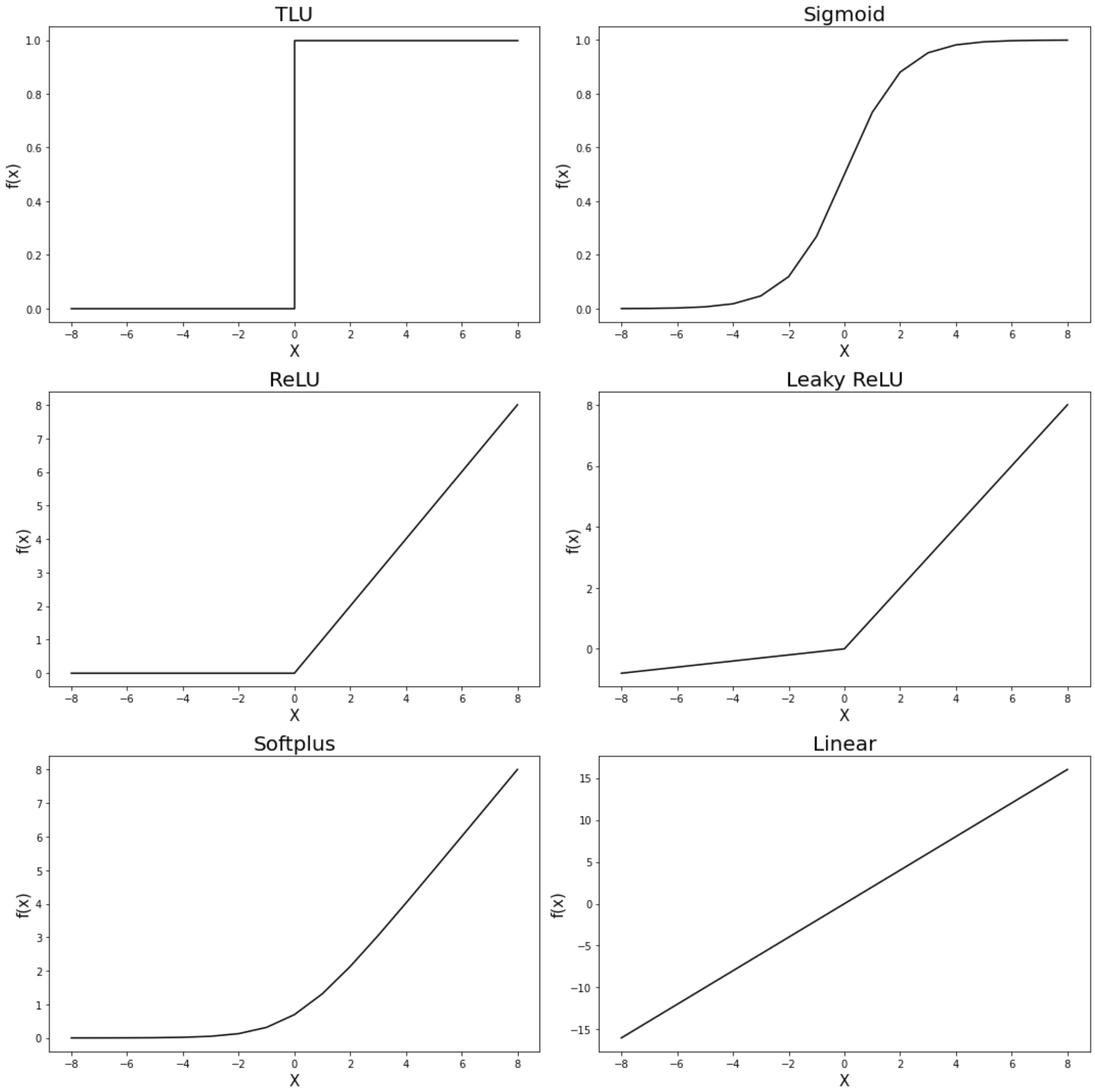


**Figure 7. Commonly applied activation functions used in neural network.**

However, the TLU is not a dynamic model. Rather, it describes a static computation of an output value based on input values. As a result, TLU can perform simple binary classification tasks but is limited to hard thresholds rather than probability. Subsequent development of functions, called activation functions ($\phi$) led to the replacement the TLU thus overcome these shortcomings.

Rosenblatt further proposed the single-layer perceptron containing multiple TLU units for multiclass prediction [20]. This model contains multiple TLU units that can take in multiple inputs and predict outcomes for multiple outputs. However, in 1969, Minksy and Seymour demonstrated that network application with a single perceptron layer was restricted to linearly separable problems, in which a line in the plane can separate all instances into either one class or the other [22]. Specifically, this limitation was demonstrated through the author's failure, using the single-layer perceptron, to find convergence when handling the nonlinearly separable XOR logic gate problem [22].

This finding led to stacking numerous layers of perceptron to overcome limitations associated with nonlinearly separable datasets [23]. This type of network, shown on Figure 8, is often referred to as the multilayer perceptron, which contains an input layer and an output layer that are separated by numerous hidden layers.

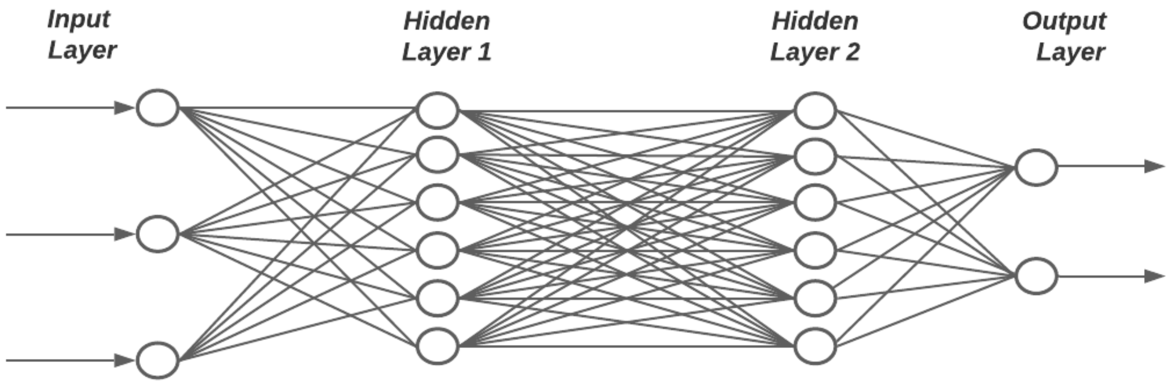


**Figure 8. Structure of a fully connected feedforward network with two hidden layers.** Here arrows represent input and output values, circles represent neurons, and solid lines represent the weights that connect the neurons.

In this network, the input layer contains a set of neurons corresponding to the features in the dataset. During forward propagation, the input values are first fed into the inner hidden layers and further propagated through each inner layer containing one to many neurons until they reach the final output layer, where the network computes the target values. Specifically, the output value of the $j^{th}$ neuron in the layer $i$ ($y_{j,i}$) can be calculated from inputting the weighted sum of all neurons in the previous layer $i$ -1 and the bias into the activation function, as follows:

$$z_{j,i} = \sum_{k=0}^{n} y_{k,i-1}\, w_{ji,ki-1} + b_{ji}$$

$$y_{j,i} = \phi\,(z_{j,i})$$

Here $y_{k,i-1}$ is the kth neuron from the previous layer $i-1$, $w_{ki,ji-1}$are the weights between the kth neuron from layer $i-1$ and the jth neuron on layer i, $b_{ji}$ is the added bias term for the jth neuron of layer $i$, $z_{j,i}$ is the weighted sum input for that same neuron, $\phi$ is the activation function.

When each neuron of every layer is connected to all neurons of the previous layer by its weighted channels, this type of network requires the process to be repeated for each neuron until the input value are propagated unidirectionally throughout the entire network. Thus, this type of network is named the fully connected feedforward network. The number of neurons and layers dictates the complexity of the model. Although there are other types of networks available, the scope of this work focuses on the feedforward network.

### 1.2.2 Activation Functions

The activation function $\phi(z)$ is the critical decision-making function that calculates the neuronal outputs based on the input. There are several activation functions, and choosing the optimal function is important for neural network learning [21]. For instance, if the network was used to predict for a set of continuous outputs, then an output layer with TLU units would fail at such task. The different characteristics of these activation function, including unique derivatives and ranges, also impact the weights of the hidden units. Thus, the optimal choices of the activation function are unique and depend on the input-output relationship that the network is intended to capture. This work compares several different functions such as sigmoid, ReLU family functions, soft plus, and linear functions as the potential activation function for the inner and outer layers of the network. Graphical figure of these activation functions was presented above in Figure 7.

#### 1.2.2.1 Sigmoid function

The sigmoid function was first presented as an important activation function for neural networks when it was introduced by Rumelhart as part of the backpropagation algorithm that will be further discussed [23]. The sigmoid function was used as the activation function, shown as below:

$$\phi(z) = \frac{1}{1 + e^{-z}}$$

Here $\phi(z)$ is the activation function output, and $z$ is the function input.

Compared to threshold linear units, the sigmoid function is differentiable and has a smooth gradient for backpropagation implementation in multiple layer perceptron [21]. The function itself is also bounded to an output range between 0 and 1, preventing explosive gradients during activations [21]. However, this bound can also create the vanishing gradient problem, as demonstrated by Xavier Glorot and Yoshua Bengio [24]. The authors demonstrated that when the weights are initialized from a normal distribution with a mean of 0 and standard deviation of 1, each layer yielded an output variance far greater than the input variance until a saturation point [24]. This phenomenon is caused by the nature of the sigmoid function, as the function is only sensitive to input values in the mid-range of [-2,2] but is not robust to input values outside of this bound. Even if the values are normalized, forward propagation throughout the network layers can eventually lead to values that saturate at the bounded values. These values will conversely result in error gradients that are near zero. The vanished gradient prevents effective backpropagation, and the network either slows down or stops learning [24]. Due to these drawbacks, in this work, the sigmoid function is only investigated as a potential function for the output layer rather than the hidden layers.

#### 1.2.2.2 ReLU

The rectified linear unit (ReLU) can improve the network learning faster than the sigmoid function. The function itself is simple, as it outputs the input value itself if the input is positive and outputs zero otherwise. The function is shown as below:

$$\phi(z) = max\ (0, z)$$

The simplicity of the function means faster computational time, both forward and backward propagation, compared to other activation functions such as the sigmoid or tanh function. An additional advantage of the ReLU function is the lack of the upper limit that could lead to vanishing gradients experienced in Sigmoid. Both factors contribute to the fast-training time. For example, Krizhevsky previously demonstrated with a four-layer convolution network, using the ReLU function can reach the same error rate six times faster than with the tanh function [25].

However, ReLU functions can suffer from the dying ReLU problem when the neural output becomes zero [21]. This situation can be caused when weight and bias are initialized with a symmetrical distribution [26]. When the weighted sum of input is negative or zero, this can lead to an output value of zero despite the input magnitude, causing the neuron to "die" [21]. Although sometimes a single dying neuron can add sparsity to the network, it can be problematic if most neurons become in the zero range, which is inevitable in deep networks [26]. When this happens, the error gradient will become minimal, and weights cannot get reupdated. Furthermore, this problem is also concerning if the ReLU activation function is used as the final output neuron, leading to predicting only outputs instances of zeros. Since this work does not investigate very deep networks, ReLU is still implemented as a potential baseline activation function for the inner layers but is omitted as the activation function for the outer layer.

#### 1.2.2.3 Leaky ReLU

To overcome the dying ReLU problem, a variant of the ReLU algorithm called the leaky rectified linear unit (Leaky ReLU) is developed. The function is as shown below:

$$\phi(z) = max\ (a * z, z)$$

Here $a$ is a small multiplier for handling instances of negative input.

This $a$ factor introduces a non-zero and differentiable slope for negative inputs and prevents the neuron from dying [27]. However, previous experiments performed by Maas et al. have demonstrated that compared to ReLU, using the leaky ReLU did not significantly impact the training optimization of a deep neural network [27]. There is still an advantage to using leaky ReLU over ReLU as the outer layer activation function, as the dying ReLU problem is more detrimental in these cases.

#### 1.2.2.4 Soft plus

The soft plus activation function is a non-linear variant of the ReLU based function as shown below:

$$\phi(z) = ln\ (1 + e^z)$$

Although soft plus function has a similar shape as ReLU, the function contains a smooth and constantly differentiable derivative that prevents dying neurons [21]. Previous work by Glorot et al has shown that in deep neural networks, ReLU based functions are equally or better than Soft plus functions at finding the global minimum of the error gradient [28]. The proposed reasoning is that in certain circumstances, the dead neurons can provide network sparsity-induced regularization [28]. As this work does not involve deep networks, soft plus is still investigated and compared with the ReLU function as the choice of activation function in the inner layers. In addition, soft plus is also used as a potential outer layer activation function to overcome the dying neuron problem.

#### 1.2.2.5 Linear Activation functions

Lastly, the linear activation function is a simplistic function with a constant derivative, as shown below:

$$\phi(z) = cz$$

Here c is a constant value.

The linear activation function is undesirable for inner layer activation functions since they lack a differentiable error gradient for backpropagation. However, they can be implemented in the outer layer as an activation function as a replacement of ReLU functions to assist in dying neuron problems [21].

### 1.2.3 Training, Validation and Test Datasets

For the network to learn the relationship between the independent variables and the output values, a set of annotated datasets must be provided to adjust its weights to fit the data trends. This type of dataset is called the training dataset, containing pairs of input vectors and the respective output vectors, also referred to as target values. The difference between the network computed outputs and the expected target values are called the error or loss value, and the total error is aggregated through a loss function. The loss value quantifies the model performance on learning the pattern in the given dataset. During training, the network aims to optimize the weights to minimalize the loss value.

While the network weights are determined during training, other parameters called hyperparameters are determined as part of the hyperparameters' learning process. Examples of hyperparameters include the number of neurons and layers, as well as activation function choice, and they play key roles in governing the model complexity. Both the network weights and hyperparameters play important roles in how well the model can accurately describe the data trends.

When the network has a high training error, the trained network fails to describe the training data pattern. This failure could be due to premature training termination before the optimal weights were found, but it could also have resulted from suboptimal hyperparameter choices. Poor hyperparameter choices can lead to an under-fitted model that results in high model bias. The under-fitted model often have an overly simplistic model architecture, such as too few neurons or layer numbers, to describe the data pattern.  Conversely, a sufficient training error indicates that the network can perform well with the training data but does not reveal how well the network is generalizing the trend. Indeed, an overfit model could perform very well on the training data but suffers when predicting new instances.  In such a case, the model can have too many neurons or be too deep and tends to incorporate noise in the training data as part of the model. Therefore, training loss alone can only provide limited insights on the model performance.

Alternatively, the annotated dataset is often divided into three subsections: the training dataset with a validation dataset, and sometimes a hold-out testing dataset. While the training dataset is used for training the network weights, the test dataset can be used to determine how well the network generalizes to new instances. Furthermore, multiple networks with different hyperparameters can be trained on the same training set. Then the network with the hyperparameter values that best performs on the same validation dataset is selected. Sometimes, the training and validation data are also partitioned to perform cross-validation. During cross-validation, the training and validation data are broken down into different complementary subsets that the network repeatedly trains and tests on. The average results from validation sets provides a performance value that reduces potential bias within a single subset of data.  However, since hyperparameters are selected using the training and validation sets from the cross-validation procedure, the final model should still be further tested on a new holdout dataset to understand how well the model can generalize trends. Therefore, a separate testing dataset was previously held out from cross-validation and can be used to assess the model variance in the final trained network.

### 1.2.4 Loss Functions

As mentioned above, the network performance can be quantified by the loss value and provides guidance for the weight optimization process. The loss function is the formula that measures the residual or error between the predicted values and the target values. There are many different types of loss functions for calculating the error gradient which the network can optimize on. While each loss function can have its advantage, they can also have their drawbacks. The graphical summary of these functions is shown on Figure 9.

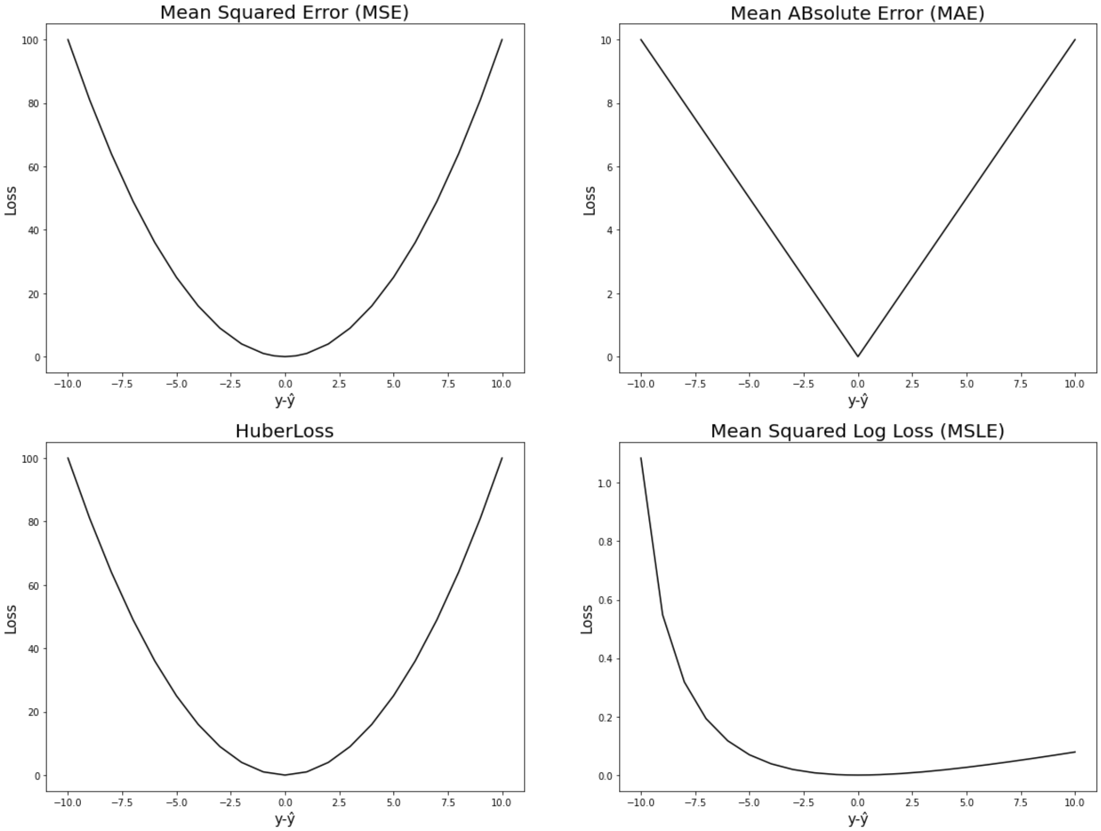


**Figure 9. Commonly applied loss functions used in neural network.**Each functional loss is plotted against the difference between the predicted value (y) and the empirical value ($\hat{y}$).

### 1.2.4.1 Mean squared error

Mean squared error (MSE) loss is modeled from the least squared loss from Gauss and is a very common loss function shown as below:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Here $\hat{y}_i$ is the empirical value of a single neuron output for instance i, and $y_i$ is same neuron output predicted by the network output at instance i.

The MSE is zero when all predictions are correct and quadratically increases with the error when predictions are wrong. As a result, mean squared error tend to penalize large errors more significantly than smaller errors [29], outliers can have a more significant effect on the loss value and thereby influencing the solution [30]. This case is especially worse when the error distribution is long tailed. On the other hand, mean squared error can be an appropriate choice of loss function when the dataset does not have many outliers, handling symmetrical error that follows normal distribution [29] .

### 1.2.4.2 Mean absolute error

Mean absolute error (MAE) is another commonly used loss function, as shown below:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \overset{\wedge}{y_i}|$$

Comparatively, the MAE is robust to outliers, as it gives proportional penalization to errors of all magnitudes. However, the absolute function is not differentiable at zero and introduces problems with gradient calculations. Thus, models trained with mean absolute error could take longer to converge and lack precision compared to MSE.

### 1.2.4.3 Huber Loss

In 1964, Huber proposed a mixed model of mean absolute error and mean squared loss [31], shown as below:

$$L_i = \begin{cases} \frac{1}{2}\left(\left(y_i - \overset{\wedge}{y_i}\right)^2\right) for \left|\overset{\wedge}{y} - y\right| \leq \delta \\ \delta \left|y_i - \overset{\wedge}{y_i}\right| - \frac{1}{2}\delta^2 otherwise \end{cases}$$

$$Huberloss = \frac{1}{n}\sum_{i=1}^{n} |L_i|$$

Here $L_i$ is the loss value for a single input-output pair, and $\delta$ is a threshold loss parameter.

Huber loss models mean absolute error for larger errors that are greater than $\delta$, but models mean squared error for a small error. Thus, this function is more robust to outliers than mean squared error but is also more differentiable than mean absolute error. However, determining the threshold loss parameter can be challenging and requires tuning.

#### 1.2.4.5 Mean Squared Log Error Loss

Mean squared log error (MSLE) can be described by the formula below:

$$MSLE = \frac{1}{n}\sum_{i=1}^{n}(log\ (\hat{y}_i + 1) - log(y_i + 1))^2$$

The log value scales errors to a similar magnitude, allowing mean squared log error to be robust to outliers. However, as the mean squared logged error is an asymmetrical function, underestimation and overestimation of the target values are not subjected to the same penalty magnitudes. Specifically, cases when the target values are underestimated will result in a larger loss compared to the when the target values are overestimated.

### 1.2.5 Solving for The Optimal Weights

As described by the section above, the loss function provides performance metrics for comparing the expected output value and the value predicted by the network for a given input set. Since the objective is to determine a set of weights that would yield predicted outputs closest to the expected outputs, the weights should be adjusted to minimize the total loss value (E). Gradient descent can achieve this by computing the error derivative with respect to the weights ($\frac{\partial E}{\partial w}$) in the network.

### 1.2.5.1 Back Propagation and Gradient Descent

In 1985, Rumelhart showed how the error gradient can be calculated for the outer layer and propagated backward for all layers [23]. The partial error gradient with respect to the neuron output ( $\frac{\partial E}{\partial y}$) can be calculated as the derivative of the loss function. Thus, for example, in the case of the mean squared error, this derivative $\frac{\partial E}{\partial y_{j,i}}$, which is the error with respect to the output value for the j$^{th}$ neuron on the output i$^{th}$ layer can be represented as follows:

$$\frac{\partial E}{\partial y_{j,i}} = 2(y_{j,i} - \hat{y}_{j,i})$$

Furthermore, the partial derivative $\frac{\partial y_{j,i}}{\partial z_{j,i}}$, which is the neuron output $y_{j,i}$ with respect to the same neuron input $z_{j,i}$ , is the derivative of the activation function. For instance, when the sigmoid function is used, this equation becomes:

$$\frac{\partial y_{j,i}}{\partial z_{j,i}} = y_{j,i}(1 - y_{j,i})$$

The partial derivative $\frac{dz_{j,i}}{dw_{ji,ki-1}}$, which is the input for the same neuron $z_{j,i}$ with respect to the weight between this neuron and the kth neuron from the previous i-1 layer, is a linear function. The derivative becomes the output of the kth neuron from the previous layer ($y_{k,i-1}$):

$$\frac{dz_{j,i}}{dw_{ji,ki-1}} = y_{k,i-1}$$

From chain rule, the error gradient with respect to the weight ($\frac{\partial E}{\partial w_{ji,ki-1}}$) can be calculated as the product of the three partial derivatives described above:

$$\frac{\partial E}{\partial w_{ji,ki-1}} = \frac{\partial E}{\partial y_{j,i}} \frac{\partial y_{j,i}}{\partial z_{j,i}} \frac{\partial z_{j,i}}{\partial w_{ji,ki-1}}$$

The error gradient with respect to the j$^{th}$ neuron from any hidden i$^{th}$ layer that is connected to the i+1$^{th}$ layer with n number of neurons can be derived as follows:

$$\frac{\partial E}{\partial y_{j,i}} = \sum_{m=0}^{n} \frac{\partial E}{\partial y_{m,i+1}} * \frac{\partial y_{m,i+1}}{\partial y_{j,i}}$$

$$\frac{\partial E}{\partial y_{j,i}} = \sum_{m=0}^{n} \frac{\partial E}{\partial y_{m,i+1}} * \frac{\partial y_{m,i+1}}{\partial z_{m,i+1}} * \frac{\partial z_{m,i+1}}{\partial y_{i,j}}$$

$$\frac{\partial E}{\partial y_{j,i}} = \sum_{m=0}^{n} \frac{\partial E}{\partial y_{m,i+1}} * \frac{\partial y_{m,i+1}}{\partial z_{m,i+1}} * w_{mi+1,ji}$$

$$\frac{\partial E}{\partial y_{j,i}} = \sum_{m=0}^{n} \frac{\partial E}{\partial z_{m,i+1}} * w_{mi+1,ji}$$

Following this process, the error gradient with respect to weights of all neurons in all previous layers can be calculated iteratively. Finally, in simple gradient descent, the weights are adapted based on the error gradient based on the formula below:

$$W_{t+1} = W_t - n\frac{\partial E}{\partial W_t}$$

Here the updated weight and bias vector ($W_{t+1}$)is based on the current weight and bias vector ($W_t$) and the gradient of the error for the current weights and bias vector ($\frac{dE}{dW_t}$) multiplied by a constant learning rate $n$ [23].

### 1.2.5.2 Stochastic Gradient Descent and The Momentum Optimizer

Gradient descent calculates the error rate with respect to all pairs of input-output values at every iteration, causing the process to be very slow at convergence, especially for large training datasets [23]. Stochastic gradient descent can accelerate this process by calculating the gradient and updating the weights using a single random pair of input-output values for each iteration. However, stochastic gradient descent has the same learning rate across all dimensions. This means that a large learning rate may cause undesirable oscillation on one dimension of the error gradient surface. Figure 10 shows that for a concave error gradient, a large learning rate may cause a direction change in each iteration leading to the gradient to oscillates horizontally across the valley rather than directly traveling downward towards the solution at the valley bottom. Although a smaller learning rate could reduce oscillation in the horizontal direction, this simultaneously will also reduce the step size for the downward progress [32].

The momentum optimization, initially proposed by Boris Polyak in 1964 [33], helps to overcome this problem. The momentum concept can be used with a stochastic gradient to reduces undesirable oscillation by introducing a momentum term $(m)$ that incorporates an exponentially decaying average of previous gradients. The weighted update of the weights for iteration $t$ can be represented by the equation below:

$$m_t = \beta(m_{t-1}) - n\frac{\partial E}{\partial W_t}$$

$$W_{t+1} = W_t + m$$

Here $m$ is a momentum vector term, $\beta$ is a decaying hyperparameter between 0 and 1.

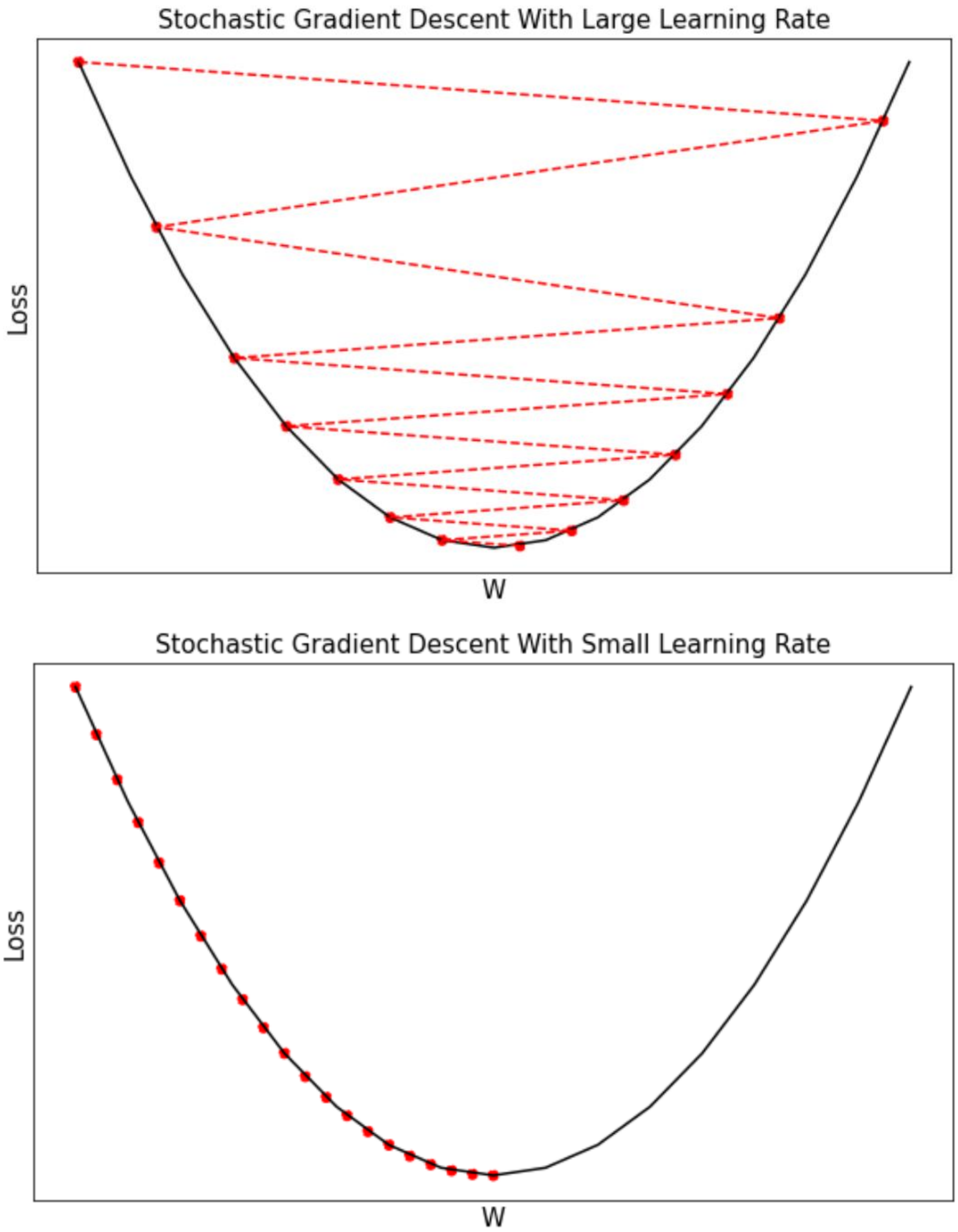


**Figure 10.Effect of learning rate on the back propagation process when determining error minimum.Top)** Learning rates too large leads to unstable directional changes across the gradient surface. **Bottom)** Learning rates too small leads to slow convergence when finding error minimum.

The accumulated average gradient helps slow down learning in the dimension where direction changes frequently and prevents unnecessary oscillation [32]. At the same time, the momentum term also allows for accelerated learning for the dimension of a more stable direction [32]. Therefore, momentum optimization with stochastic gradient descent allows for much faster convergence compared to without momentum.

#### 1.2.5.3 ADAM Solver

ADAM, short for ADAptative Momentum estimation, is a recently developed optimizer that also computes an adaptive learning rates for each parameter [34]. ADAM builds on both the momentum algorithm, and AdaGrad algorithm [35] and the RMSProp algorithm [36] for updating weights [34]. ADAM updates its weights using the following equations:

$$m_{t+1} = \beta_1 m_t - (1 - \beta_1)\frac{dE}{dW_t}$$

$$s_{t+1} = \beta_2 s_t - (1 - \beta_2)\left(\frac{dE}{dW_t}\right)^2$$

$$\hat{m} = \frac{m_{t+1}}{1 - \beta_1}$$

$$\hat{s} = \frac{s_{t+1}}{1 - \beta_2}$$

$$w_{t+1} = W_t + \frac{n\hat{m}}{\sqrt{\hat{s} + \varepsilon}}$$

Here $m$ is the momentum vector, $\frac{dE}{dW_t}$ is the gradient vector, $\beta_1, \beta_2$ are the decaying hyperparameters, $\varepsilon$ is a smoothing term to prevent division by zero,$n$ is the default learning parameter, and t is for the iteration number.

Equations 1 and 2 calculates the exponentially decaying moving averages of gradient and the squared gradient, respectively. These gradients are the first and second moments of the gradient, representing the mean and uncentered variances. Equations

3 and 4 are implemented to overcome bias associated with initializing the m and s terms as zero during the early search. Lastly, each weight is updated from the previous weight based on the historical momentum [34].

ADAM solver is implemented in this work with its default hyperparameters, as it was reported that its default hyperparameters work well in most cases, and little tuning is usually needed. Additional advantages for implementing ADAM include its fast convergence speed and ability to handle noisy and sparse gradients [34].

### 1.2.6 Additional Hyperparameters

As previously mentioned, hyperparameters are components of the network characteristics that can be determined as part of the training selection process. Some important hyperparameters previously mentioned includes the activation function, loss function, and the type of solver. There are additional hyperparameters such as the network size and structure, the type of weight initialization process that can also dictate both the training process and the final solution.

Different network size also reflects the complexity of the relationship. As mentioned previously, a network with too few layers may lead to an underfit model with high bias but low variance. Conversely, a network that is too complex can lead to overfitting, resulting in a model with a high variance but low bias. Therefore, during hyperparameter tuning, networks of varying sizes should be implemented on both the training data and the validation dataset to determine the appropriate model with the optimal bias and variance trade-off. In this work, we investigated different network sizes ranging from one to five layers, with each layer containing ten neurons. Since the dataset handled is not of high dimensionality, we hypothesized that a shallow network of up to five layers will be a sufficient model.

As previously mentioned, neural networks with sigmoid activation can encounter the vanishing gradient problem. Xavier Glorot et al was the first to propose a root reason for this problem in his 2010 paper. He highlighted that initializing the weights with a gaussian distribution with sigmoid functions, which introduce more significant output variance for each layer compared to input variance [24].The normal distribution has a mean of zero, the sigmoid function has a mean of 0.5, which can cause the variance to increase until saturation at the end layers, reducing the gradient close to zero [24]. Glorot weight initialization was further introduced, where the weights of each layer are proportionally scaled to the number of input layers to allow for variance entering each layer to be equal to the variance exiting each layer [24]. He et al also developed the He weights initialization method for overcoming similar challenges associated with ReLU activation functions, where a scaling factor of 2 was also applied [37]. In this work, both He and Glorot weight initialization methods are investigated as potential weight seeding methods.

## 1.3 Objectives and hypothesis

The overall goal of my thesis is to accelerate method development for DMS quantification of lipid stereoisomers. To achieve this, training data was collected from performing exhaustive experiments covering different combinations of COV/SV machine values for 17 lipid species with varying N-acyl chain length and degree of unsaturation. These data allow the neural network to learn the relationship between lipid structural characteristics and optimal DMS machine parameter values. Furthermore, the tuned algorithm, iDMS, can be further applied to simulate for DMS manual tuning values for 47 lipid species pairs where no standard exists. These simulations provide the users with the optimal DMS machine parameter values (SV, COV) based on the physicochemical structure of lipid of interest and allow them to perform LC-DMS-MS/MS separation and quantification of each lipid species.

# Chapter 2: Methods

## 2.1 Pipeline Process Overview

A summary of the algorithm development process is shown in Figure 11. Training data was first generated using direct injection of lipid species into ESI-DMS-MS/MS. The measured ion signals levels were collected in the form of ion intensities for different neutral glycosphingolipid species as a function of different DMS machine parameters. The data was aggregated and preprocessed into its ionogram form. As depicted in Figure 6, ionograms display MS detected intensities at different COV values for a constant SV for a given period of direct injection [17]. Each ionogram can also be approximated as a gaussian function, as the measured ion intensities follow a normal distribution across the CV values. Gaussian features were extracted with peak COV values were used to estimate the mean value and the peak widths at the half maximal intensity were used to estimate the standard deviation value. The feature engineered dataset was further split into a training dataset and a validation dataset. The training dataset was used for hyper-parameter selection via cross-validation and to evaluate the effect of training dataset size. The validation dataset was used to assess training bias in the final tuned model. To complete the study, the final network was trained on all lipid species for which synthetic standards are available and then applied to determine the DMS parameter values for lipid species lacking synthetic standards.

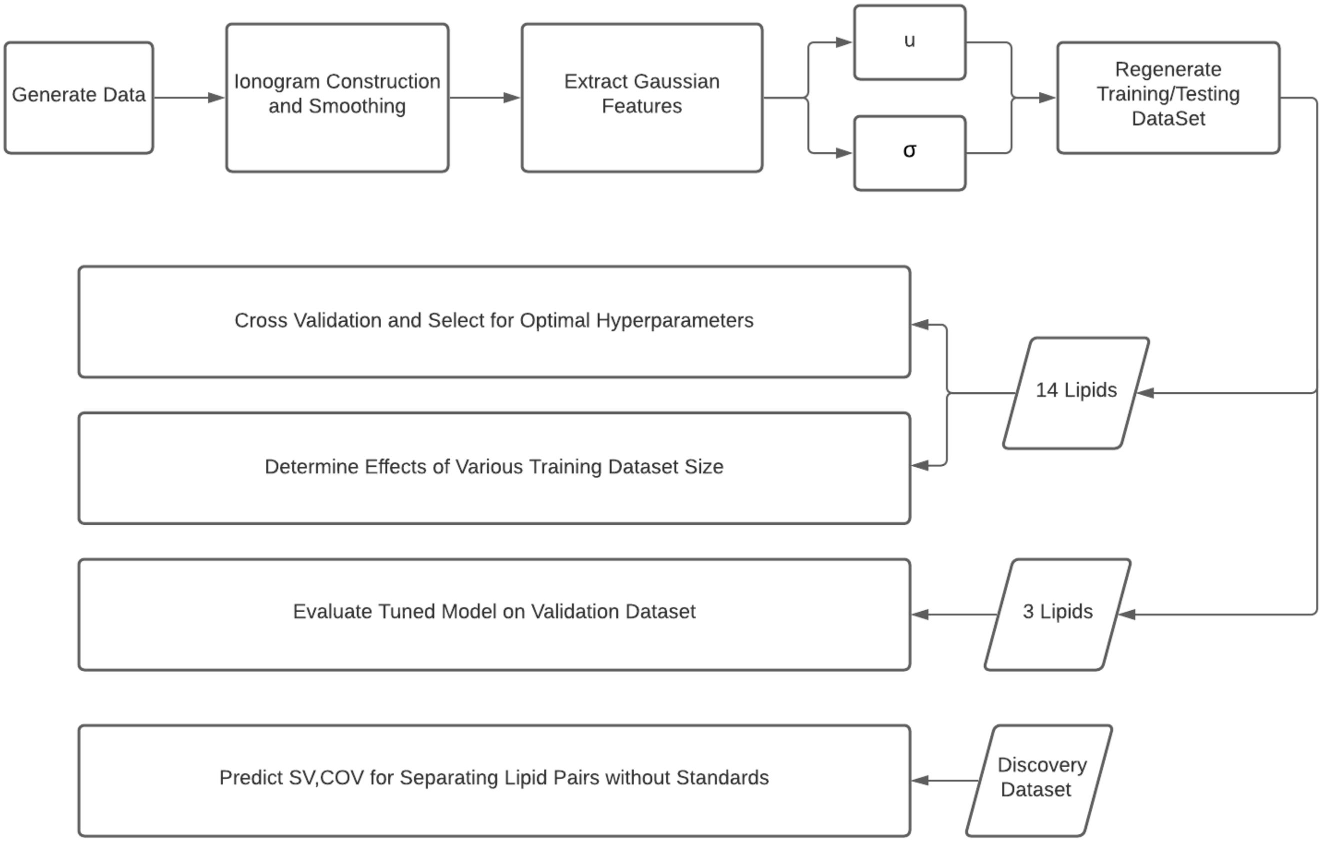


**Figure 11. Flowchart of the Experimental Pipeline.**

## 2.2 Direct Infusion (DI)-DMS-ESI-MS/MS

DMS optimization of 17 pairs of lipid isomers were performed in the Bennett laboratory by Ph.D candidate Thao Nguyen following the procedure of Xu et al [10]. All lipids had a d18:1 sphingosine backbone modified by a glucose or galactose headgrop via a β-glycosidic linkage. One stereoisomer pair represented the sphingosine backbone with no N-acyl hydrocarbon (β-GalSph(d18:1/0:0) and β-GlcSph(d18:1/0:0). Fourteen stereoisomer pairs (β-GalCer(d18:1/8:0), β-GlcCer(d18:1/8:0), β-GalCer(d18:1/14:0), β-GlcCer(d18:1/14:0), β-GalCer(d18:1/16:0), β-GlcCer(d18:1/16:0), β-GalCer(d18:1/18:0), β-GlcCer(d18:1/18:0), β-GalCer(d18:1/20:0), β-GlcCer(d18:1/20:0), β-GalCer(d18:1/21:0), β-GlcCer(d18:1/21:0), β-GalCer(d18:1/22:0), β-GlcCer(d18:1/22:0), β-GalCer(d18:1/23:0), β-GlcCer(d18:1/23:0), β-GalCer(d18:1/24:0), β-GlcCer(d18:1/24:0), GalCer(d18:1/25:0), β-GlcCer(d18:1/25:0) , β-

GalCer(d18:1/26:0), β-GlcCer(d18:1/26:0), β-GalCer(d18:1/31:0), β-GlcCer(d18:1/31:0) had fully saturated N-acyl hydrocarbons of varying lengths (aka no degrees of unsaturation). Three stereoisomer pairs were monounsaturated β-GalCer(d18:1/16:1), β-GlcCer(d18:1/16:1), β-GalCer(d18:1/18:1), β-GlcCer(d18:1/18:1), β-GalCer(d18:1/22:1), β-GlcCer(d18:1/22:1). All lipids were custom synthesized by Avanti Polar Lipids (Alabaster, USA).

DMS separation was performed using a SelexION DMS device (SCIEX, Concord, Ontario) interfaced with a triple quadrupole-linear ion trap mass spectrometer QTRAP 5500 (SCIEX). The DMS cell was mounted in the atmospheric pressure region between the Turbo V ion source (SCIEX) curtain and the orifice plate of the MS. Nitrogen was used as the transport gas. Before sample injection, lipid samples were diluted with ethanol to a final concentration of 1 μM. Lipid standards were T-infused using a syringe pump set at 5 μL/min into the flow of 10 μL/min of 100% solvent B (acetonitrile/isopropanol at 5:2 v/v ratio containing 0.1% formic acid and 10 mM ammonium acetate) delivered by an Agilent Infinity II system (Agilent, Mississauga, ON). DI-ESI-DMS-MS/MS was performed as described in Xu et al [20]. The following DMS parameters were held constant: DMS cell temperature was set at 150°C (low). Modifier gas concentration was set to low concentration (1.5% v/v in transport gas) and consisted of isopropanol. DMS resolution enhancement was set to 30 psi (Medium). DMS offset was set to - 3.0 V, and nitrogen resolving gas = 30 psi. All other MS parameters were as described in [9]. To determine the effect of SV, COV, and lipid structure on the sensitivity and selectivity of the lipid analyte, each lipid species was injected into the ESI-DMS-MS/MS at these settings mentioned above while ramping the COV for each SV adjusted in 100 V steps from 0V to 4100V. COV ramping occurred from -10 V to 10 V in 0.2 V. Ion intensity was measured at each SV/COV combination.

## 2.3 Dataset

For each lipid species at each SV, intensity data were collected.  Data were aggregated to generate an ionogram with Y-axis displaying intensity and X-axis representing COV between -10 V to 10 V in 0.2 V increments. Figure 12 A displays examples of the aggregated ionograms for β-GalCer(d18:1/16:0) and β-GlcCer(d18:1/16:0), generated at three different SV values. Data were collected at every SV in 100 V steps (Figure 12 B).  For all 17 stereoisomer pairs (34 lipid species), data were collected for 42 different unique SV values and 101 unique COV values . Each dataset was generated in duplicates . The datasets were loaded and processed in Python 3.7.10 using packages Pandas 1.2.5 and NumPy 1.20.2. All data manipulations in the following work were performed using these packages.

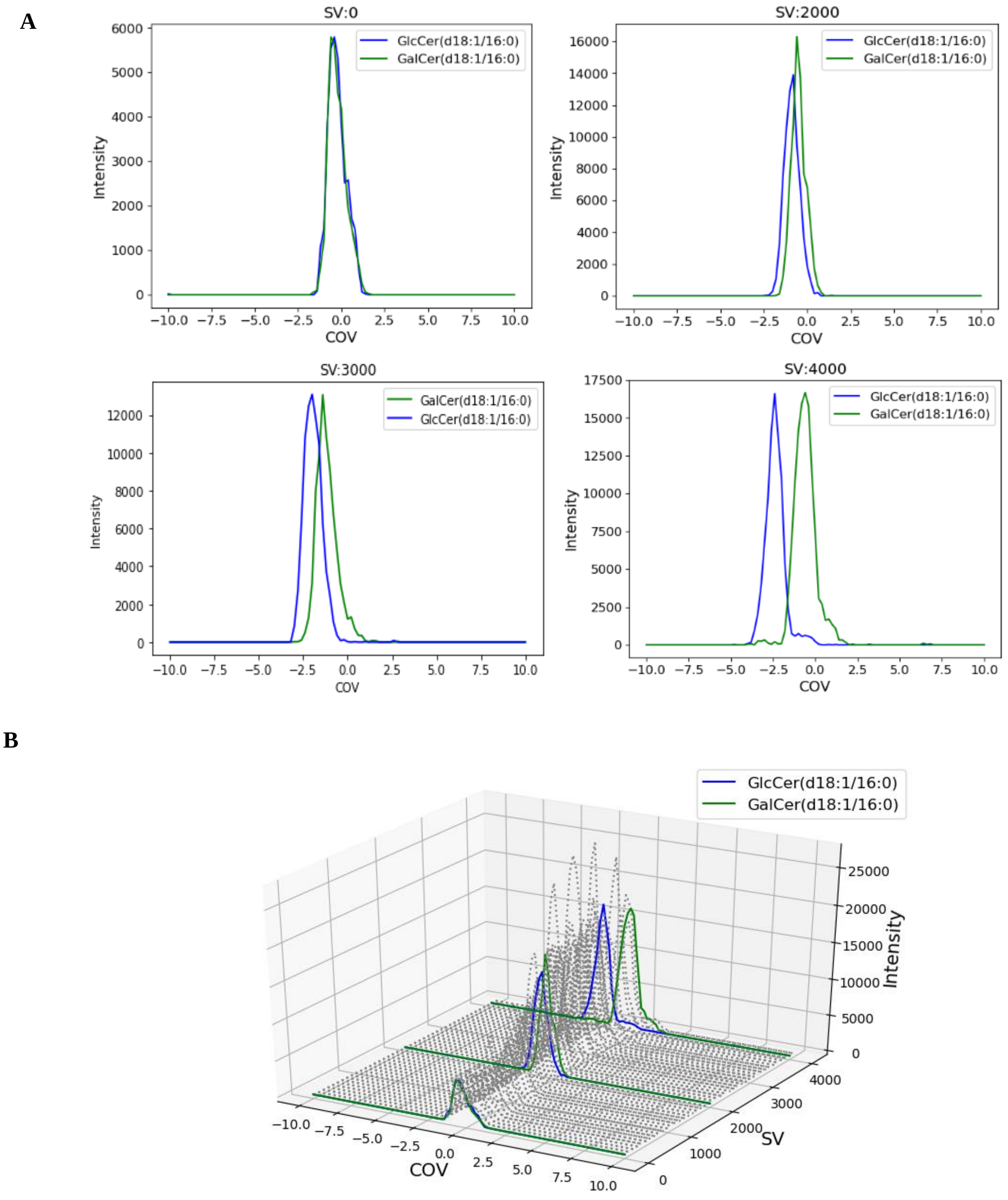


**Figure 12. Example of the β-GalCer(d18:1/16:0) and β-GlcCer(d18:1/16:0) dataset. A)** Ionograms were generated and overlaid for each stereoisomer at each SV. SV= 0 V, SV=2000 V, SV=3000,and SV=4000 V are shown. B) The entire dataset for this stereoisomer pair is depicted from SV= 0 V to SV= 4100 V in 100 V steps. The colored ionograms are the 3-D representations of the three ionograms from the top figure.

## 2.4 Data Normalization and Gaussian Features Engineering

Each ionogram was treated as a unique dataset composed of 101 observations, with intensity values dependent on the COV value. Each ionogram was described by the following sample space $\{COV_j, I_j : j = 1, \ldots, 101\}$. The relationship between COV at the $j^{th}$ instance $(COV_j)$ and intensity at the $j^{th}$ instance $(I_j)$ was assumed to be described by function g($x_j$) and random noise $\varepsilon r_j$, as follows: $I_j$=g($COV_j$)+ $\varepsilon r_j$. For each ionogram, I approximated for function g($COV_j$) without the noise $\varepsilon r_j$ , by solving for the cubic smoothing spline $f$ using De Boor's approach [38]. The cubic spline interpolation was performed using the CSAP Python package V1.0.0.

During the process, the cubic spline function $f$ that approximates function g($COV_j$), was determined by minimizing an error term that dictated goodness of fit, and a second term that dictated the function roughness, as follows:

$$p \sum_{j=1}^{n=101} w_j \left| I_j - f\left(COV_j\right) \right|^2 + (1-p) \int \lambda(t) |f''(t)|^2 \, dt$$

Here $I_j$ and $COV_j$ are the $j^{th}$ instance in the dataset. $w_j$ is a set of error measure weights set to the default value of 1. $f''$ is the second derivative of the function f. The piecewise constant weight function $\lambda$ is the constant function 1. $p$ is the smoothing parameter with a range of [0,1] that dictates the minimalization priority between function fit and roughness. When the smoothing parameter p is 1, function f will only minimalize for the fit term and thus converges to the natural spline interpolant. Conversely, when the smoothing parameter p is 0, function f converges to the least-squares straight-line fit of the data. In this study, a smoothing parameter of 0.98 was chosen to avoid over-smoothing. The cubic spline smoothing function f was determined for each ionogram to smooth random noise and improve feature engineering (Figure 13A).

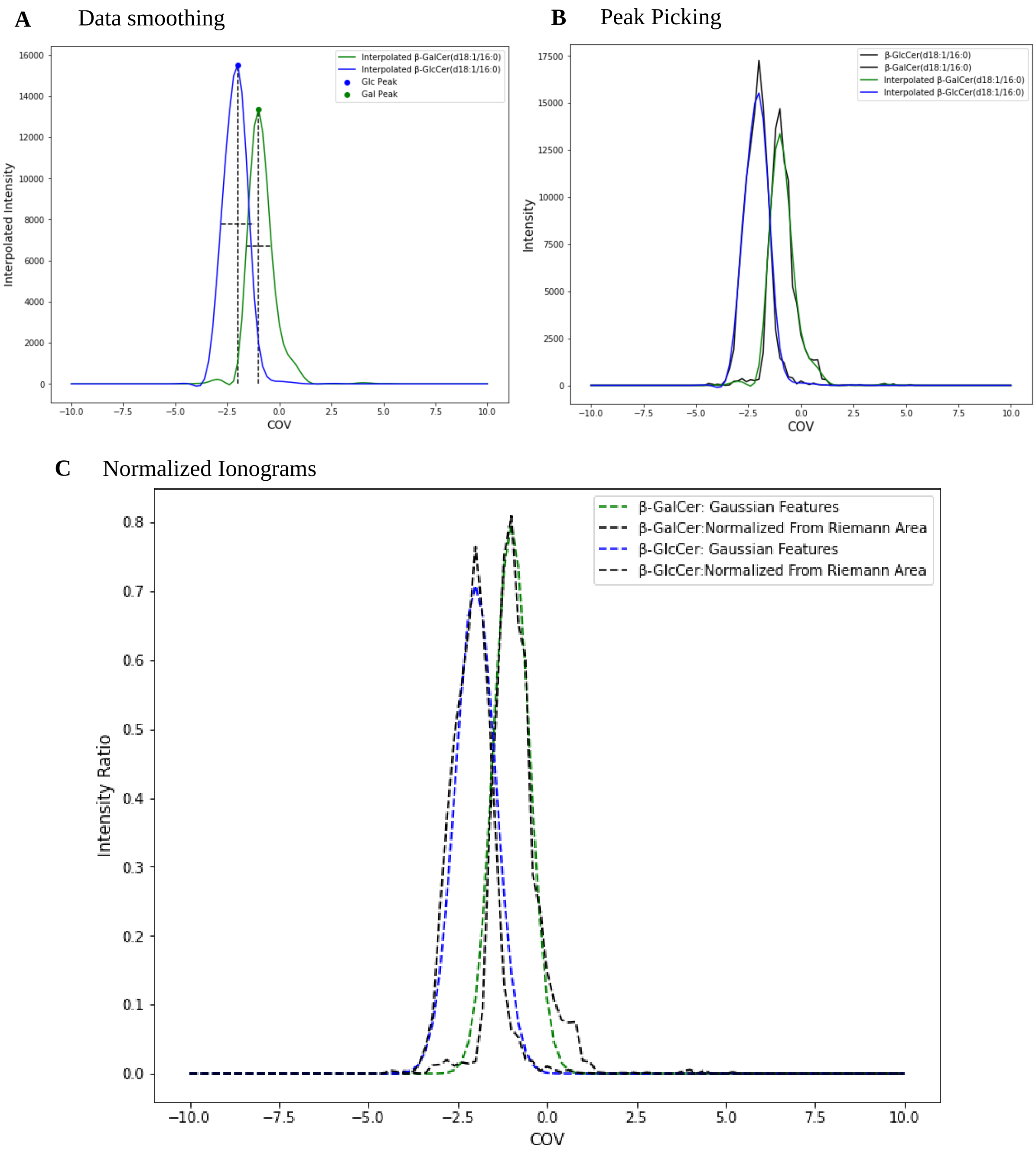


**Figure 13. Data preprocessing and feature engineering.** A) Cubic smoothing splines were applied to the aggregated data of each stereoisomer in each pair to smooth the data. B) The peak COV and gaussian width were identified from the recapitulated ionogram to estimate $\mu,\sigma$ values. C) Normalized lipid peaks were approximated from the gaussian intensity ($\bar{I}$) and compared to lipid peaks calculated from the area under the curve ($\tilde{I}$).

Following smoothing, the maximum intensity of each ionogram was selected as the peak intensity. For each lipid and specified SV, we further determined the four features that described each peak to aggregate both stereoisomer datasets into a single ionogram: β-GlcCer peak intensity, β-GlcCer peak COV, β-GalCer peak intensity, and β-GalCer peak COV(Figure 13B).

The mean ($u_{Glc}, u_{Gal}$) and standard deviation ($\sigma_{Glc}, \sigma_{Gal}$) were estimated from the peak COV and peak width at half maximum from the aggregated ionograms, as shown by the dotted lines on the ionograms of Figure 14 A.

The features were extracted following the equation below:

$$u \text{ =Peak CoV, } \sigma=\frac{\textit{Full Width Half Max}}{2\sqrt{2ln2}}$$

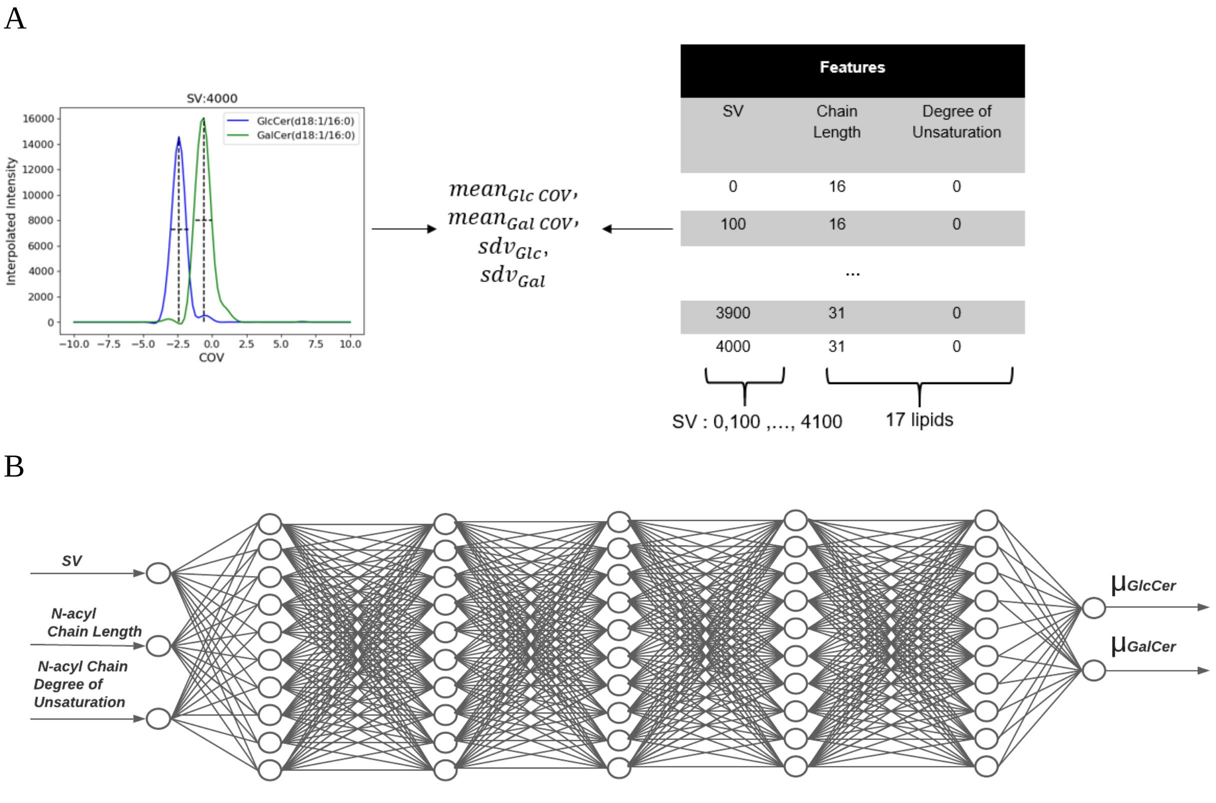


**Figure 14. Gaussian feature engineering. A)**Gaussian features were extracted for each peak pair in each aggregated ionogram at each SV between 0-4100 V for the 17 pairs of lipid stereoisomers. **B)**Problem formulation using neural network.

For all 17 lipid species, gaussian features were extracted from the 42 datasets at each unique SV value in replicate experiments. The average $u_{Glc}, u_{Gal}$ were calculated across the two replicates, unique to each lipid species at each SV. The average $\sigma_{Glc}, \sigma_{Gal}$ values across two replicates were also estimated for each SV value and for each lipid. However, since the $\sigma_{Glc}, \sigma_{Gal}$ value did not greatly vary across lipid species as opposed to across SV, the $\sigma_{Glc}, \sigma_{Gal}$ values were further averaged across all lipid species at the corresponding SV.

To validate that normalized intensity can be correctly approximated using Gaussian features, empirical ionograms were also normalized for comparison. The empirical normalization was achieved by dividing the intensity at each COV value by the area under the curve defined by each peak. The area under the curve was roughly approximated for each peak using the Riemann sum, as follows:

$$A = \sum_{j=-10}^{10} I_j(COV_j) * \Delta COV_{j-1,j}$$

Here $(COV_j)$ is the COV value at the j$^{th}$ instance, $I_j$ is the corresponding intensity value, and A is the area under the ionograms. Furthermore, $\Delta COV$ is the CV step size, where $\Delta COV_{j-1,j}$=$COV_j - COV_{j-1}$. Since intensity were collected at CV increments of 0.2 Volts, $\Delta COV_{j-1,j}$ is 0.2.

The normalized intensity at the j$^{th}$ instance $\tilde{I}(COV_j)$ was calculated for the corresponding COV value using the approximated area:

$$\tilde{I}(COV_j) = \frac{I_j(COV_j)}{A}$$

Similarly, the normalized ionograms were also approximated from the gaussian features using the probability density function as follows:

$$\bar{I}(COV_j) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{COV\,j-\mu}{\sigma}\right)^2}$$

Here $(\bar{I})$ is the normalized intensity approximated from the gaussian parameters. The normalized intensity with respect to Riemann summed area $(\tilde{I})$ was further compared to normalized intensity approximated from gaussian features $(\bar{I})$ , as shown by Figure 13C. The close correspondence between two normalized curves for β-GlcCer(d18:1/14:0) and β-GalCer(d18:1/14:0) validated that gaussian features can approximate normalized intensity.

It was assumed that there was an underlying relationship between each lipid peak mean values $u_{Glc}, u_{Gal}$, each DMS parameter (SV and CoV) and lipid structure (N-acyl hydrocarbon length and degree of unsaturation). The relationship was described as:

$$u_{Glc}, u_{Gal} = f(SV, C, U)$$

Here the C and U represent the lipid's N-acyl chain length and degree of unsaturation, respectively. Function f encapsulates the relationship between the features and predictions shown in Figure 14A. A neural network was also utilized to learn these relationships as displayed on Figure 14B.

Before training, the input and output values of the dataset were normalized using the min/max normalizer, according to the equation below:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Here $X_{min}$ is the minimum value across the dataset, $X_{max}$ is the maximal value across the dataset. These normalized datasets were further used for training.

## 2.5 Network Training and Grid Search Hyperparameters

A fully connected neural network model was used to learn the relationship between SV, chain length, degree of unsaturation, and the mean values of $u_{Glc}, u_{Gal}$. The open-source library TensorFlow 2.0.0 and Keras 2.2.4 were implemented to develop and train the feedforward network. During training, different pairs of inputs (SV, chain length, degree of unsaturation) and output ( $u_{Glc}, u_{Gal}$) values were fed into the specified network to solve for the weights using the ADAM solver.

Hyperparameters were selected using a grid search. These parameters included the network space, where one to five fully connected layers with ten neurons per layer were implemented. ReLU and soft plus functions were compared as the inner layers' activation functions. ReLU, soft plus, Sigmoid, and linear functions were compared as the outer layers' activation functions. Various loss functions were also compared for penalization, including the mean absolute error, mean squared loss error, Huber loss, and mean squared log loss error. Other hyperparameters were also explored, such as using He or Glorot as the weight initialization step.

The dataset was first divided into a training dataset of 14 lipid pairs for performing grid search with cross-validation. A testing dataset of three lipid pairs was left out to measure the performance of the final tuned network (Figure 15). The training set was used to perform a leave one lipid pair out cross-validation during grid-search. Next, separate networks, with unique hyperparameter pairings, were cross validated during grid search by training on 13 lipid pairs and testing the remaining single lipid pair. All individual experiments were repeated ten times with different weights seeded to overcome any randomness associated with the weight initialization step. For each lipid pair at every unique hyperparameter combination, the average test results of the ten experiments were calculated. The average test results across all the lipid pairs were further calculated and compared to choose the optimal hyper-parameter performance. The exact criteria for how these results were compared and evaluated is described in section 2.6.

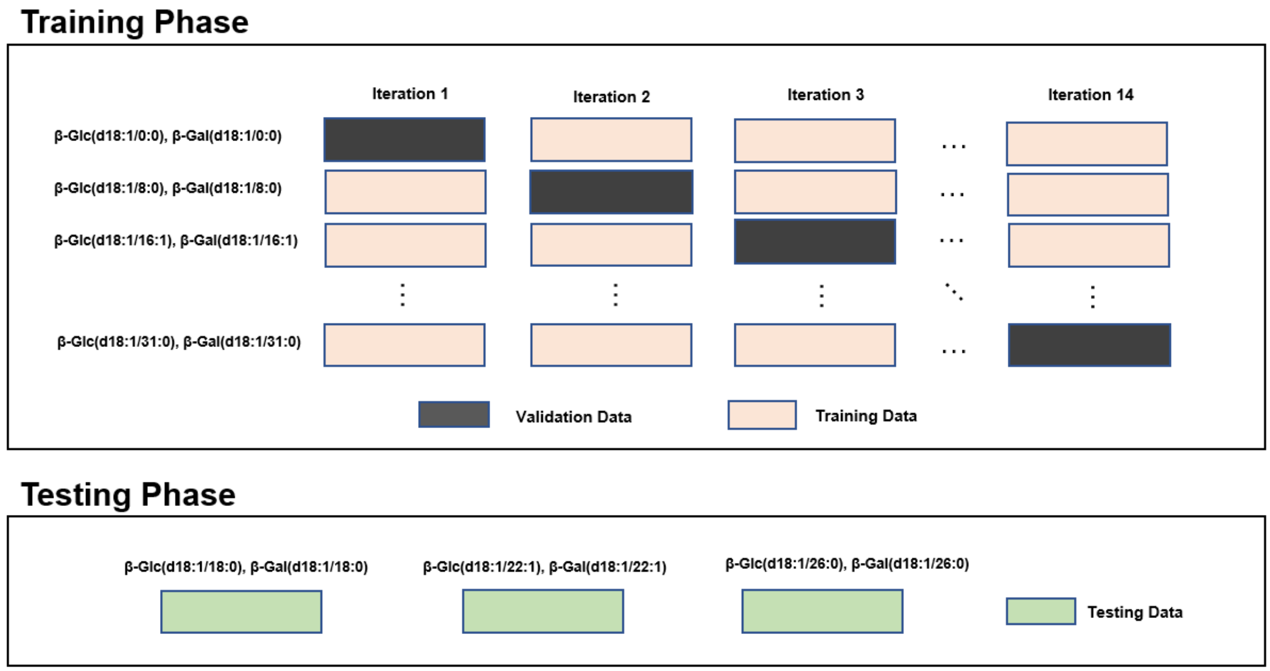


**Figure 15. Dataset split into training datasets for cross-validation (with holdouts) and test datasets.** Fourteen stereoisomer pairs were used in the training data phase, and three lipid pairs were left out for validation.

The tuned hyperparameters were also evaluated. The network with the optimal parameters was retrained on the training dataset and was used to predict and evaluate the three “hold-out” pairs in the test set.

## 2.6 Evaluation and Metrics of Network Results

Since the loss function was a hyperparameter, a standardized method was required for comparing the various training results. Therefore, a universal criterion was implemented to compare the network results across different loss functions.

From the prediction results, ionograms were generated for each lipid pair at each SV. Firstly, the normalized intensity aggregated from prediction was denoted as $\ddot{I}$ , not to be confused with normalized empirical intensity denoted as $\bar{I}$.. Similar to how $\bar{I}$ was calculated following the gaussian distribution equation in section 2.4, $\ddot{I}$ was also determined from the predicted $u_{Glc}$, predicted $u_{Gal}$, average $\sigma_{Glc}$ and average $\sigma_{Gal}$ values (Figure 13C).

To compare the characteristics of normalized ionograms ($\ddot{I}$) from empirical intensity ($\bar{I}$), three types of features were extracted from both types of ionograms to provide a summary attribute. First, the maximum normalized intensity of each lipid pair in a given ionogram was selected as the peak value. Thus, for ionograms of β-GlcCer data, this intensity was referred to as β-GlcCer peak relative intensity, and the corresponding values were referred to as β-GlcCer peak CoV. Similarly, the maximum intensity and corresponding COV values for β-GalCer data were referred to as β-GalCer peak relative intensity and β-GalCer peak COV.

Furthermore, where the β-GlcCer curves and β-GalCer curves intersect was defined as the valley value between two curves. The characteristic (location and intensity) of this valley can provide insights into the separability of the two stereoisomers. The corresponding COV and intensity values at the valley were referred to as valley COV and relative valley intensity.

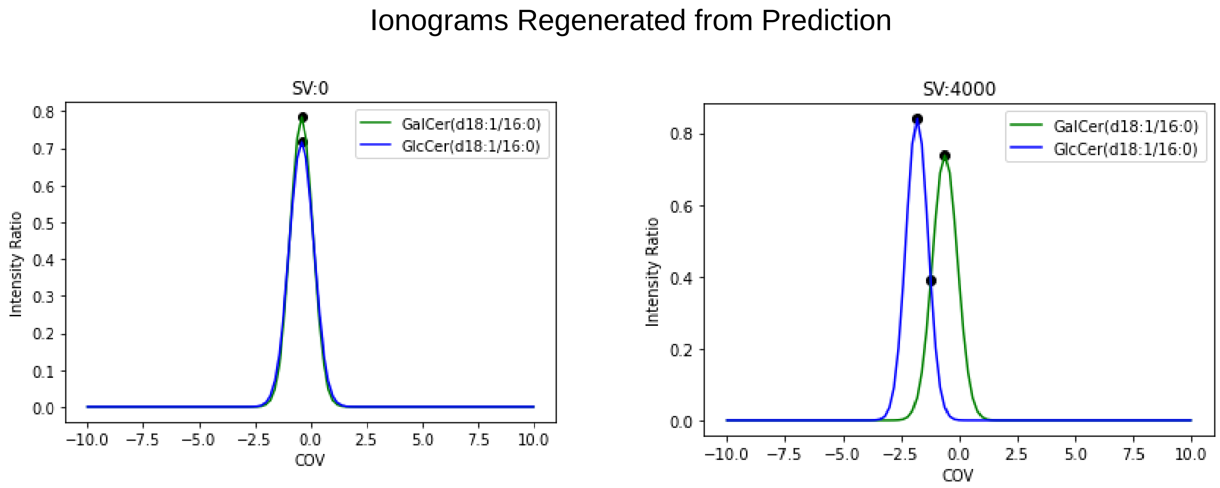
Extracting the valley features required a longer process than extracting the lipid peak values. One observation from the data was that all β-GlcCers peaks had a mean COV value to the left of β-GalCers. As such, the valley COV must be bounded within the β-GlcCer peak COV and β-GalCer peak COV. Furthermore, the left of the valley had a higher normalized β-GlcCer intensity, whereas to the right of the valley had a higher normalized β-GalCer intensity. The valley bounds were determined by incrementally searching the space between the two lipid peak intensities. Finally, the valley COV and relative valley intensity values were determined from linearly interpolating the valley bound values.

The above processes were repeated across all empirical and aggregated ionograms to determine all six key features: β-GlcCer peak relative intensity, β-GlcCer peak COV, β-GalCer peak relative intensity, β-GalCer peak COV, valley COV, and relative valley intensity. For each feature, correlation parameters ($R^2$) were determined between the empirical values and the values obtained from prediction.

Both empirical ionograms and predicted normalized ionograms were subsequently annotated with separation labels. These labels determine whether β-GlcCer and β-GalCer had distinct peaks that allowed them to be separated at each SV. The annotation was based on the 50% evaluation rule at each SV for each pair of lipid species and was calculated as:

$$\frac{Valley(Rel)Intensity}{Min(GlcCerPeak(Rel)Intensity, GalCerPeak(Rel)Intensity)} = \begin{cases} \leq 0.5 Seperation \\ > 0.5 NoSeperation \end{cases}$$

Examples of the annotated separation labels are shown in Figure 16. It was important to consider that annotation based on this rule naturally yields an imbalanced dataset, as 10% of the dataset is annotated as separation. The annotated binary labels from predictions were further compared against the empirical labels at each SV for each lipid species. Finally, the interpreted result of separation was reported as a form of binary classification, where the metrics of F1, accuracy, sensitivity, and specificity were reported.



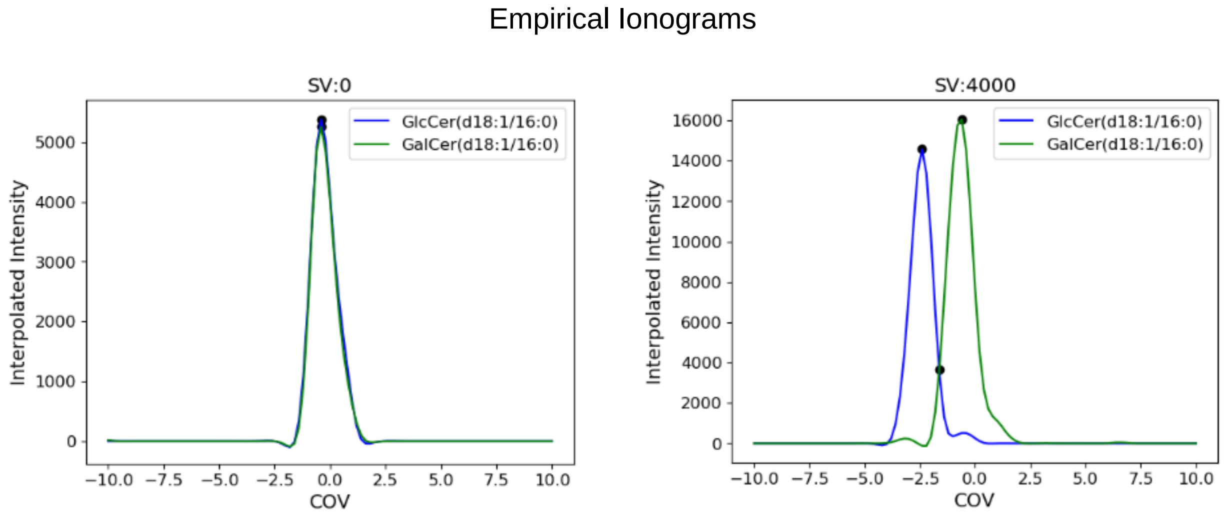


**Figure 16. Peaks and valleys detected in aggregated empirical and predicted ionograms for β-GlcCer(d18:1/16:0) and β-GalCer(d18:1/16:0) at SV=0 and SV=4000.**
**Top**) Ionograms with intensity ($\ddot{I}$) generated based on the prediction $\mu$ and average $\sigma$ values. **Bottom**) Empirical ionograms (I) of the lipid species.

## 2.7 Determination of Minimum Training Sample

To determine the minimum number of samples required for training, I evaluated the performance of the tuned neural networks on the cross-validation data using varying training set sizes. The 14 lipid pairs were further split into training sets, varying between 2 and 12 pairs. For each training dataset size, up to 100 different lipid combinations of training data were drawn from the cross-validation dataset. The remainder of the lipids was used as the test set for each combination. Thus, ten separate networks were trained for each unique combination.

## 2.8 Application of Tuned Neural Network

Optimal hyperparameters determined from cross-validation were implemented in the final network. Ten separate networks with these parameters had weights initialized with the optimal weight setting method and were trained using the entire 17 stereoisomer pairs. The final tuned network was then used to determine the DMS machine parameter values of lipid pairs where synthetic standards did not exist. The network simulated the normalized ionogram features for all possible new lipid pairs with chain length between 0 to 32 and degree of unsaturation between 0 and 1. An ensemble method was implemented for the final prediction, where the average of the ten network predictions was reported to overcome variability due to the weight initialization step or training results. Ionograms were regenerated from the prediction results indicating the COV/SV values that would yield separation of each lipid species.

These SV/COV combinations were then further used as DMS machine parameter values to quantify these lipid species in biological samples. LC-ESI-DMS-MS/MS was implemented with DMS using specified COV/SV values for targeted quantification of β-GlcCers and β-GalCers pairs where no standards were available.

# Chapter 3: Results

## 3.1 Extracted Features

As previously mentioned, the measured ion intensity is a quantitative measure of the DMS performance as dictated by machine parameters settings and the structural properties of the lipid species under investigation. The ability of iDMS to aggregate empirical data and generate composite ionograms of stereoisomer pairs enabled both quantitative and qualitative evaluation of the impact of lipid structure on both ion intensity and DMS separation. For example, differences in ion intensities are depicted in Figure 17 under different SV and COV combinations for three pairs of neutral sphingolipid stereoisomers with different chain lengths. Plots present the following lipid species: β-GlcCer(d18:1/8:0), β-GalCer(d18:1/8:0), β-GlcCer(d18:1/16:0), β-GalCer(d18:1/16:0)), β-GlcCer(d18:1/31:0) , β-GalCer(d18:1/31:0), and the ion intensity sum of β-GlcCer(d18:1/8:0) and β-GalCer(d18:1/8:0), β-GlcCer(d18:1/16:0) and β-GalCer(d18:1/16:0), β-GlcCer(d18:1/31:0) and β-GalCer(d18:1/31:0).

As expected, lipids exited the DMS cell and were detected by the MS within a narrow range of COVs for a given SV (Figures 17); however, little to no discrimination of stereoisomers was observed for all lipids employing SV under 1500 V. This is evident in the overlapping intensities depicted in Figure 17 (right panel). By contrast, separation was achieved for all pairs at higher SV values (Figure 17). This observation is also highlighted in Figure 18. For all stereoisomer pairs, the COV value with the maximal intensity, referred to as the peak COV defining the apex of a given lipid peak in each ionogram, had a value around 0 at low SV. Here, β-GlcCers and β-GalCers were inseparable (Figure 17). The peak COV value moved away from 0 as SV was increased. Separation was only achieved at high SV values as shown by the bifurcation of peak intensities in Figures 17 and 18.

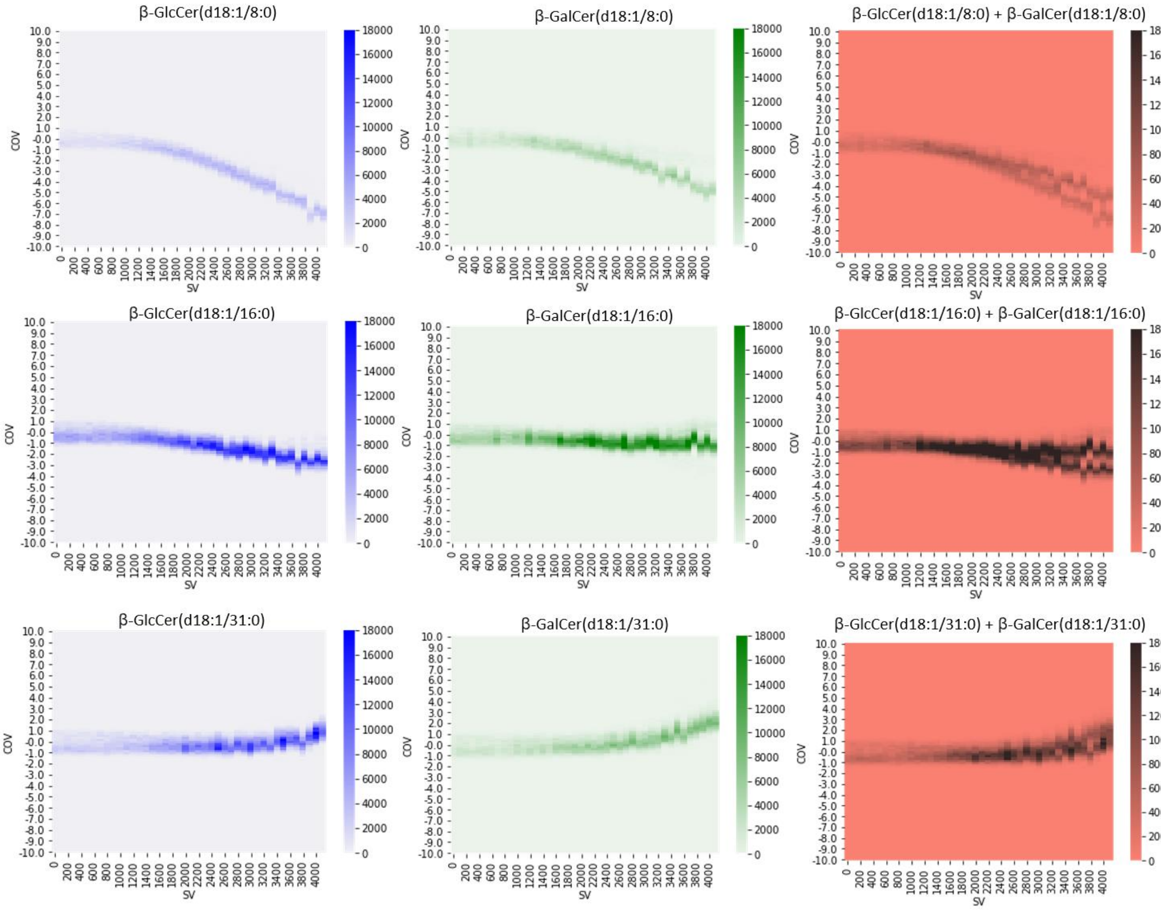


**Figure 17. Heatmap of measured ion intensities of three different lipid stereoisomers across different SV and COV combinations.** The left column shows heatmap for β-GlcCers. The middle column shows heatmaps of the corresponding β-GalCer stereoisomers, The right column shows sum of the intensities of both stereoisomer pairs across varying SV and COV.

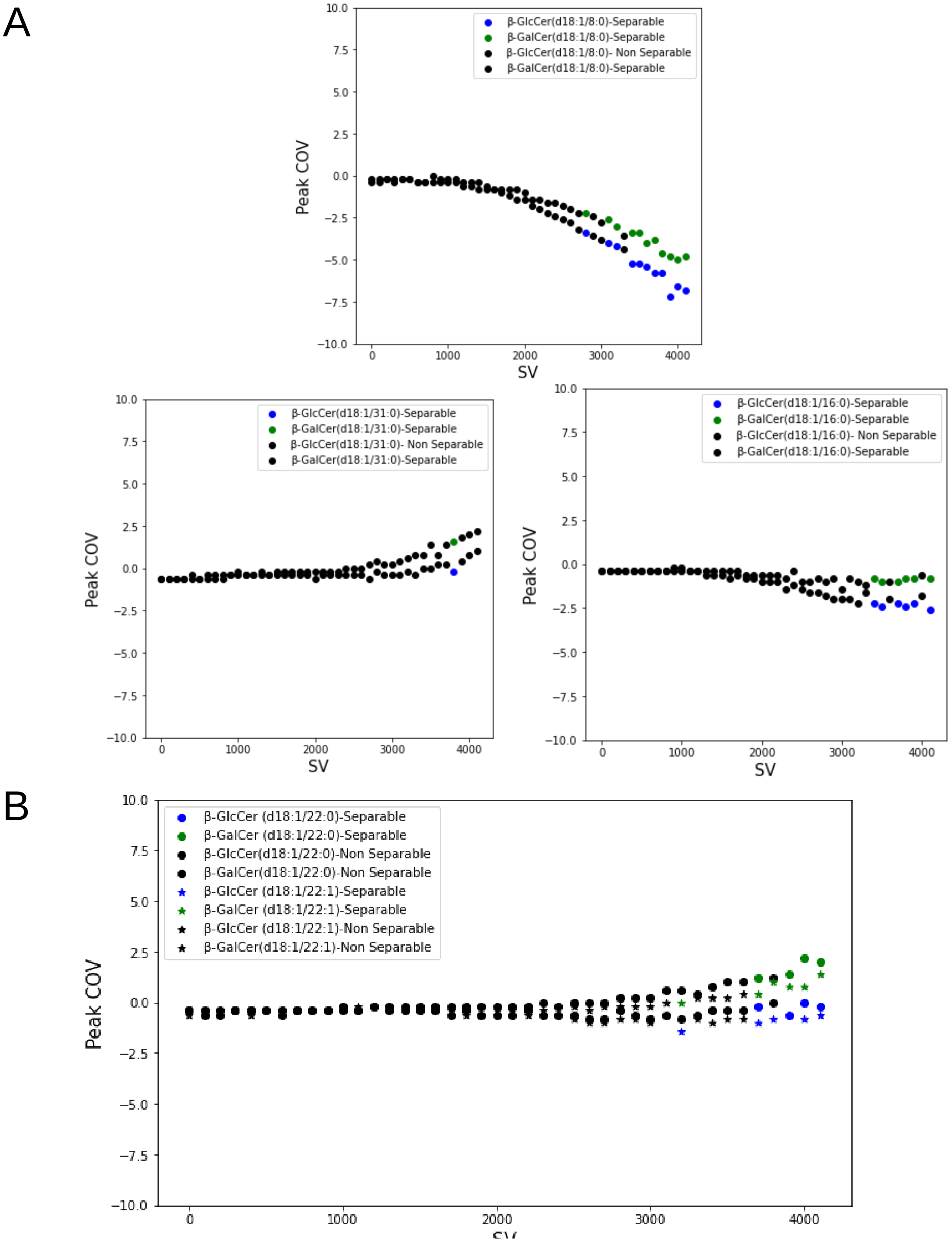


**Figure 18. SV, N-acyl chain length, and N-acyl chain degree of unsaturation impact on the location of the peak CoV for the lipid stereoisomers.** A) Lipids with longer N-acyl chain lengths require increasingly positive COV to separate. B) Increases in the number of double bonds requires more negative COV. Data represent the CoV value at the apex of each peak. In all plots, black markers indicate instances where the indicated β-GlcCer and β-GalCer did not separate. Separation criteria required the valley intensity to be less than 50% of the smallest peak height of the two lipid species at the given SV. Green and blue markers indicate instances where the stereoisomer pairs met the separation criteria at the given SV.

Separation was further impacted by the length of the N-acyl chain with stereoisomers with longer hydrocarbons increasingly difficult to separate. Note that separation required a higher SV value for β-GalCer(d18:1/16:0) and β-GlcCers(d18:1/16:0) compared to β-GalCer(d18:1/8:0) and β-GlcCer(d18:1/8:0) (Figures 17 and 18). Moreover, summed intensities of β-GlcCer(d18:1/31:0) and β-GalCer(d18:1/31:0) continued to overlap across COV ranges at high SV values as shown in Figure 17. Only a single SV value met the separation criteria as shown in Figure 18. These observations support the idea that SV required to separate neutral glycosphingolipids is proportional to N-acyl hydrocarbon length.

The identity of sugar headgroup was also found to impact on COV.β-GlcCers required more negative peak COV value compared to β-GalCers to be directed to the DMS orifice and thus be detected, as shown in Figure 18A. Moreover, species with shorter N-acyl chains required a more negative COV value to separate than species with longer N-acyl chains, as shown both in both Figure17 and Figure 18A.Finally, addition of double bonds further negatively shifted optimal CoV values (Figure 18B).

Data presented in Figures 17 and 18 represent the average of replicate experiments at each measurement point. While the general trend described above was reproducibly observed, noise was apparent. The CoV/SV trajectories were not smooth but rather oscillated back and forth with as SV was increased (Figures 17 and 18). These data indicate variability in response from day to day. In context of analytical chemistry, the impact of the DMS parameters (impact of modifier concentration on ion clustering and thus response to SV and CoV) may be attributed to fluctuations in room temperature or humidity, for example, the contribute to this variation in response. This noise and reproducibility must be considered when establishing optimal method parameters.

Interestingly, different lipid species displayed varying magnitudes of ion response as depicted in the heatmap (Figure 17). For instance, the ion intensity range for β-GlcCer(d18:1/8:0) and β-GalCer(d18:1/8:0) was found to be between 0-7000 ions, as opposed to around 0-18000 ions detected for β-GlcCer(d18:1/16:0) and β-GalCer(d18:1/16:0). The varying dynamic range of ion response could be attributed to factors such as the unique ionization efficiency of each lipid species. Lipids that ionize more efficiently by the ESI upstream of the DMS will result in a greater abundance within the DMS cell, thus enhanced cluster capacity, and with greater probability of exiting the DMS cell and reaching the mass spectrometer detector. Therefore, although the ion intensity is dictated by the DMS parameters chosen, the value itself is also influenced the impact of lipid structure on ionization, regulated in part by the source parameters upstream of the DMS cell [39]. Thus, directly predicting absolute ion intensity is difficult. As a result, I opted for a more rational approach and reformulated the problem to predict Gaussian features that resemble normalized ion intensity values and not absolute intensity.

To optimize this approach, gaussian parameters $u_{Glc}, u_{Gal}, \sigma_{Glc}, \text{and } \sigma_{Gal}$ were extracted for each lipid species. The extracted $u_{Glc}$ and $u_{Gal}$ approximated the peak COV values of each lipid species at given SVs. The extracted values of the $\sigma_{Glc}$ and $\sigma_{Gal}$ across SV values were plotted for five particular lipid species (Figure 19). Although the extracted values $\sigma$ values varied across SV values, the $\sigma$ values did not vary significantly with the chain length at high SV values, where separation occurs (i.e SV>1500). Therefore, it was not necessary to predict for the $\sigma_{Glc}$ and $\sigma_{Gal}$ values for all input features. The $\bar{\sigma}_{Glc}$ and $\bar{\sigma}_{Gal}$ values were estimated for each lipid species at each SV by averaging all $\sigma_{Glc}$ and $\sigma_{Gal}$ values across all lipid species at the corresponding SV.

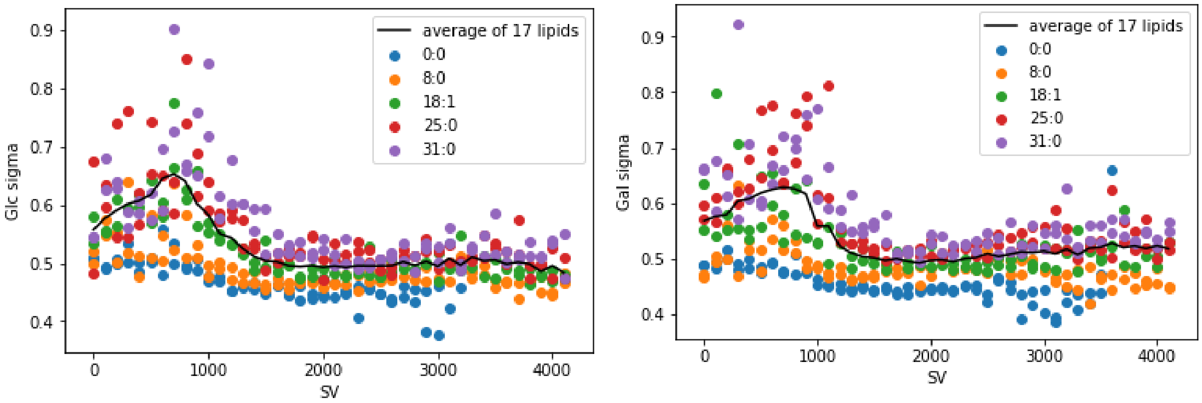


**Figure 19. Extracted values of $\sigma_{gal}, \sigma_{glc}$ do not vary significantly across lipid species at a given SV.** Selective dispersion plots of the $\sigma_{gal}$ and $\sigma_{glc}$ for five different lipid species lacking an N-acyl chain (β-GlcSph and β-GalSph) or defined by medium-chain (6-12 hydrocarbons), long-chain (13-21 hydrocarbons), and very long-chain (greater than 22 hydrocarbons) N-acyl chains are presented. Data represent average COV $\bar{\sigma}$ plotted by SV. The black line indicates the dispersion plot of the average value across all 17 lipid pairs.

## 3.2 Cross-Validation Results and Optimal Parameters Selection

As described in chapter 2, section 2.6, the neural network was trained to predict for $u_{Glc}$ and $u_{Gal}$ as a function of SV, N-acyl chain length, and their degree of unsaturation. The network was trained with different architectures of between 1-5 layers with ten neurons each, and the weights were initialized using either the He or Glorot method. Furthermore, ReLU and Softplus were used as potential inner activation functions, while leaky ReLU, sigmoid, Softplus, and linear functions were tested as the outer layer activation function. Finally, networks were also trained with four different loss functions and performance of all these combinations was evaluated.

To determine the optimal combination of the hyperparameter, cross-validation was performed with a training dataset of 14 lipid pairs for each hyperparameter combination. During cross-validation, one lipid species was left out at a time to be the validation dataset. Ten separate networks with the same hyperparameters were repeated on the test set to account for any instability in the weight initialization.

Each test set result was evaluated based on the methods described in section 2.6. The average value of the test set results across the ten networks was reported for each hyperparameter combination for every single lipid pair. The average test set results of each lipid species across the training set were averaged and further reported as the final result for the given hyperparameter combinations. This process was repeated for all 961 unique hyperparameter combinations, and the average performance across all tests was reported for each combination.

Hyperparameters were ranked based on the average F1 value for annotated separation labels and the average $R^2$ value of key ionograms features, including Glc peak COV, Gal peak COV, valley COV, relative Glc peak intensity, relative Gal peak intensity, and relative valley intensity. The top ten performing hyperparameter combinations are listed in Table 1. The details of the classification metrics for separation at a given SV are listed in Table 2. The average test set results, such as the classification metrics and the correlation values, are reported as the average across all networks and all lipid species. The details of the peak feature regression values are shown in Table 3. All ten of the algorithms had a soft plus function as both the inner and outer activation functions, which indicated that the selection of the soft plus function had a significant performance impact compared to other hyperparameter choices. Furthermore, the number of layers

in the top ten combinations was between 3-5 demonstrating that a network of at least three layers is required to learn the relationship between feature inputs and predictions. The optimal tuned network had five layers, with soft plus as the activation function for all layers, mean absolute error as the loss function, initialized weights based on the Glorot method, and an early stopping loss of 1E-10.

***Table 1. Summary of cross-validation results for networks with the top ten optimal hyperparameter combinations.***

| Inner Activation | Outer Activation | loss Function | Layer # | Stop-loss | weight Int method | Avg Rank | F1 rank | Avg $R^2$ rank | Avg Peak/ Valley Intensity $R^2$ | Avg Peak/ Valley COV $R^2$ |
|---|---|---|---|---|---|---|---|---|---|---|
| softplus | softplus | mae | 5 | 1E-10 | Glorot | 9.75 | 14 | 5.5 | 0.933 | 0.885 |
| softplus | softplus | mse | 5 | 1E-15 | Glorot | 24.75 | 12 | 37.5 | 0.931 | 0.885 |
| softplus | softplus | mse | 4 | 1E-15 | Glorot | 44.25 | 65 | 23.5 | 0.932 | 0.885 |
| softplus | softplus | huberLoss | 4 | 1E-10 | He | 46.25 | 91 | 1.5 | 0.932 | 0.889 |
| softplus | softplus | mae | 5 | 1E-15 | Glorot | 50.75 | 58 | 43.5 | 0.934 | 0.882 |
| softplus | softplus | huberLoss | 3 | 1E-15 | He | 51.25 | 69 | 33.5 | 0.933 | 0.883 |
| softplus | softplus | MSLE | 4 | 1E-15 | Glorot | 64.75 | 84 | 45.5 | 0.932 | 0.883 |
| softplus | softplus | huberLoss | 5 | 1E-15 | Glorot | 68 | 118.5 | 17.5 | 0.932 | 0.885 |
| softplus | softplus | mse | 4 | 1E-10 | Glorot | 70.75 | 28 | 113.5 | 0.932 | 0.880 |
| softplus | softplus | mse | 3 | 1E-15 | Glorot | 73.75 | 144 | 3.5 | 0.932 | 0.886 |
| | | | | | ... | | | | | |

***Table 2. Separation performance for the cross-validation results for networks with the top ten hyperparameter combinations.***

| Inner Activation | Outer Activation | loss Function | Layer # | Stop-loss | weight Int method | F1 | TN | FP | FN | TP | Sen | Selec | Acc |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| softplus | softplus | mae | 5 | 1E-10 | Glorot | 0.729 | 33.2 | 2.71 | 0.757 | 4.67 | 0.894 | 0.927 | 0.917 |
| softplus | softplus | mse | 5 | 1E-15 | Glorot | 0.73 | 33.2 | 2.74 | 0.736 | 4.70 | 0.898 | 0.925 | 0.916 |
| softplus | softplus | mse | 4 | 1E-15 | Glorot | 0.722 | 33.1 | 2.86 | 0.743 | 4.69 | 0.893 | 0.921 | 0.912 |
| softplus | softplus | Huber Loss | 4 | 1E-10 | He | 0.721 | 33.0 | 2.94 | 0.714 | 4.71 | 0.901 | 0.920 | 0.912 |
| softplus | softplus | mae | 5 | 1E-15 | Glorot | 0.723 | 33.1 | 2.78 | 0.779 | 4.65 | 0.894 | 0.925 | 0.915 |
| softplus | softplus | Huber Loss | 3 | 1E-15 | He | 0.722 | 33.2 | 2.72 | 0.821 | 4.61 | 0.887 | 0.926 | 0.915 |
| softplus | softplus | MSLE | 4 | 1E-15 | Glorot | 0.722 | 33.0 | 2.89 | 0.736 | 4.69 | 0.899 | 0.920 | 0.912 |
| softplus | softplus | Huber Loss | 5 | 1E-15 | Glorot | 0.719 | 33.0 | 2.91 | 0.75 | 4.68 | 0.899 | 0.920 | 0.911 |
| Softplus | softplus | mse | 4 | 1E-10 | Glorot | 0.726 | 33.1 | 2.79 | 0.743 | 4.69 | 0.897 | 0.924 | 0.915 |
| Softplus | softplus | mse | 3 | 1E-15 | Glorot | 0.717 | 33.1 | 2.84 | 0.807 | 4.62 | 0.888 | 0.922 | 0.911 |

...

**Table 3. Correlation performance of peak features extracted from cross-validation results and empirical ionograms for the top ten ranked networks.**

| Inner Activation | Outer Activation | loss Function | Layer # | Stop-loss | weight Int method | R$^2$ Gal Peak COV | R$^2$ Glc Peak COV | R$^2$ Valley COV | R$^2$ Gal Peak Intensity | R$^2$ Glc Peak Intensity | R$^2$ Valley Intensity |
|---|---|---|---|---|---|---|---|---|---|---|---|
| softplus | softplus | mae | 5 | 1E-10 | Glorot | 0.900 | 0.843 | 0.835 | 0.981 | 0.985 | 0.913 |
| softplus | softplus | mse | 5 | 1E-15 | Glorot | 0.911 | 0.829 | 0.827 | 0.981 | 0.984 | 0.915 |
| softplus | softplus | mse | 4 | 1E-15 | Glorot | 0.913 | 0.828 | 0.830 | 0.981 | 0.985 | 0.913 |
| softplus | softplus | huber Loss | 4 | 1E-10 | He | 0.914 | 0.837 | 0.831 | 0.981 | 0.985 | 0.915 |
| softplus | softplus | mae | 5 | 1E-15 | Glorot | 0.898 | 0.839 | 0.835 | 0.981 | 0.985 | 0.908 |
| softplus | softplus | huber Loss | 3 | 1E-15 | He | 0.912 | 0.827 | 0.833 | 0.981 | 0.985 | 0.910 |
| softplus | softplus | MSLE | 4 | 1E-15 | Glorot | 0.913 | 0.822 | 0.831 | 0.982 | 0.984 | 0.914 |
| softplus | softplus | huber Loss | 5 | 1E-15 | Glorot | 0.914 | 0.823 | 0.831 | 0.981 | 0.984 | 0.913 |
| softplus | softplus | mse | 4 | 1E-10 | Glorot | 0.904 | 0.823 | 0.830 | 0.982 | 0.984 | 0.912 |
| softplus | softplus | mse | 3 | 1E-15 | Glorot | 0.932 | 0.887 | 0.914 | 0.831 | 0.83` | 0.914 |

...

The final tuned network yielded a prediction measure very close to empirical data across the majority of lipid species. Table 4 demonstrates that the final network prediction yielded relative peak intensity measures that were highly correlated with the empirical values, with an $R^2$ value of 0.98. Furthermore, average (MAE values) between 0.006 and 0.007 were observed for each relative peak intensity for all lipid species (Table 5). These results suggested that the gaussian height was not significantly impacted by using the average $\overline{\sigma}_{Glc}\,\text{and}\,\overline{\sigma}_{Gal}$ value across all the lipid pairs. The MAE of both the relative peak and valley intensity was also plotted considering N-acyl chain length of each stereoisomer pair tested by the network (Figure 20, top panel). The MAE of the relative peak intensity measurements was constant across each of the test datasets and showed independence from the N-acyl chain length, further justifying the approach of using an average $\overline{\sigma}_{Glc}\,\text{and}\,\overline{\sigma}_{Gal}$ value. However, as shown in both Table 5 and Figure 20, higher errors values were observed for relative valley intensities than relative peak intensities. Figure 21 further shows that the estimated peak intensity values had much lower residuals than the estimated valley intensity values. Since relative valley intensity is the crossing point between the two lipids, this value is dependent on both the Gaussian width and mean. Therefore, it may be more sensitive to systematic noise and thus more difficult to predict.

**Table 4. Correlation performance of peak features extracted from cross-validation results of the optimal network versus empirical ionograms for each of the lipid pairs.**

| Lipid N-Acyl Chain | $R^2$ Gal Peak COV | $R^2$ Glc Peak COV | $R^2$ Valley COV | $R^2$ Gal Relative Intensity | $R^2$ Glc Relative Intensity | $R^2$ Valley Relative Intensity |
|---|---|---|---|---|---|---|
| **0:0** | 0.983 | 0.987 | 0.985 | 0.985 | 0.987 | 0.89 |
| **8:0** | 0.987 | 0.991 | 0.986 | 0.983 | 0.984 | 0.807 |
| **14:0** | 0.96 | 0.98 | 0.931 | 0.978 | 0.984 | 0.818 |
| **16:0** | 0.639 | 0.978 | 0.783 | 0.982 | 0.985 | 0.804 |
| **16:1** | 0.842 | 0.976 | 0.918 | 0.983 | 0.983 | 0.895 |
| **18:1** | 0.669 | 0.966 | 0.692 | 0.98 | 0.983 | 0.936 |
| **20:0** | 0.951 | 0.841 | 0.935 | 0.981 | 0.986 | 0.785 |
| **21:0** | 0.938 | 0.739 | 0.923 | 0.981 | 0.987 | 0.85 |
| **22:0** | 0.984 | 0.7 | 0.947 | 0.978 | 0.985 | 0.961 |
| **23:0** | 0.979 | 0.711 | 0.949 | 0.981 | 0.986 | 0.83 |
| **24:0** | 0.973 | 0.441 | 0.953 | 0.987 | 0.981 | 0.816 |
| **24:1** | 0.957 | 0.59 | 0.876 | 0.984 | 0.983 | 0.877 |
| **25:0** | 0.972 | 0.831 | 0.974 | 0.98 | 0.984 | 0.661 |
| **31:0** | 0.98 | 0.875 | 0.972 | 0.982 | 0.983 | 0.782 |
| **Average** | 0.915 | 0.829 | 0.916 | 0.982 | 0.984 | 0.837 |

***Table 5. MAE of peak features extracted from cross-validation results of the optimal network versus empirical ionograms for each of the lipid pairs.***

| Lipid N-Acyl Chain | Gal Peak COV | Glc Peak COV | Valley COV | Gal Relative Intensity | Glc Relative Intensity | Valley Relative Intensity |
|---|---|---|---|---|---|---|
| **0:0** | 0.839 | 0.843 | 0.667 | 0.006 | 0.006 | 0.027 |
| **8:0** | 0.265 | 0.394 | 0.253 | 0.006 | 0.006 | 0.046 |
| **14:0** | 0.130 | 0.150 | 0.058 | 0.006 | 0.007 | 0.057 |
| **16:0** | 0.132 | 0.168 | 0.139 | 0.006 | 0.006 | 0.028 |
| **16:1** | 0.135 | 0.120 | 0.075 | 0.006 | 0.006 | 0.024 |
| **18:1** | 0.094 | 0.085 | 0.051 | 0.006 | 0.006 | 0.022 |
| **20:0** | 0.095 | 0.105 | 0.042 | 0.007 | 0.006 | 0.031 |
| **21:0** | 0.102 | 0.116 | 0.045 | 0.007 | 0.006 | 0.034 |
| **22:0** | 0.110 | 0.142 | 0.069 | 0.007 | 0.006 | 0.025 |
| **23:0** | 0.102 | 0.115 | 0.047 | 0.006 | 0.007 | 0.034 |
| **24:0** | 0.096 | 0.098 | 0.035 | 0.006 | 0.007 | 0.030 |
| **24:1** | 0.091 | 0.113 | 0.041 | 0.006 | 0.007 | 0.025 |
| **25:0** | 0.077 | 0.100 | 0.055 | 0.007 | 0.006 | 0.037 |
| **31:0** | 0.095 | 0.244 | 0.107 | 0.006 | 0.005 | 0.051 |
| **Average** | 0.169 | 0.199 | 0.120 | 0.00629 | 0.00621 | 0.0336 |

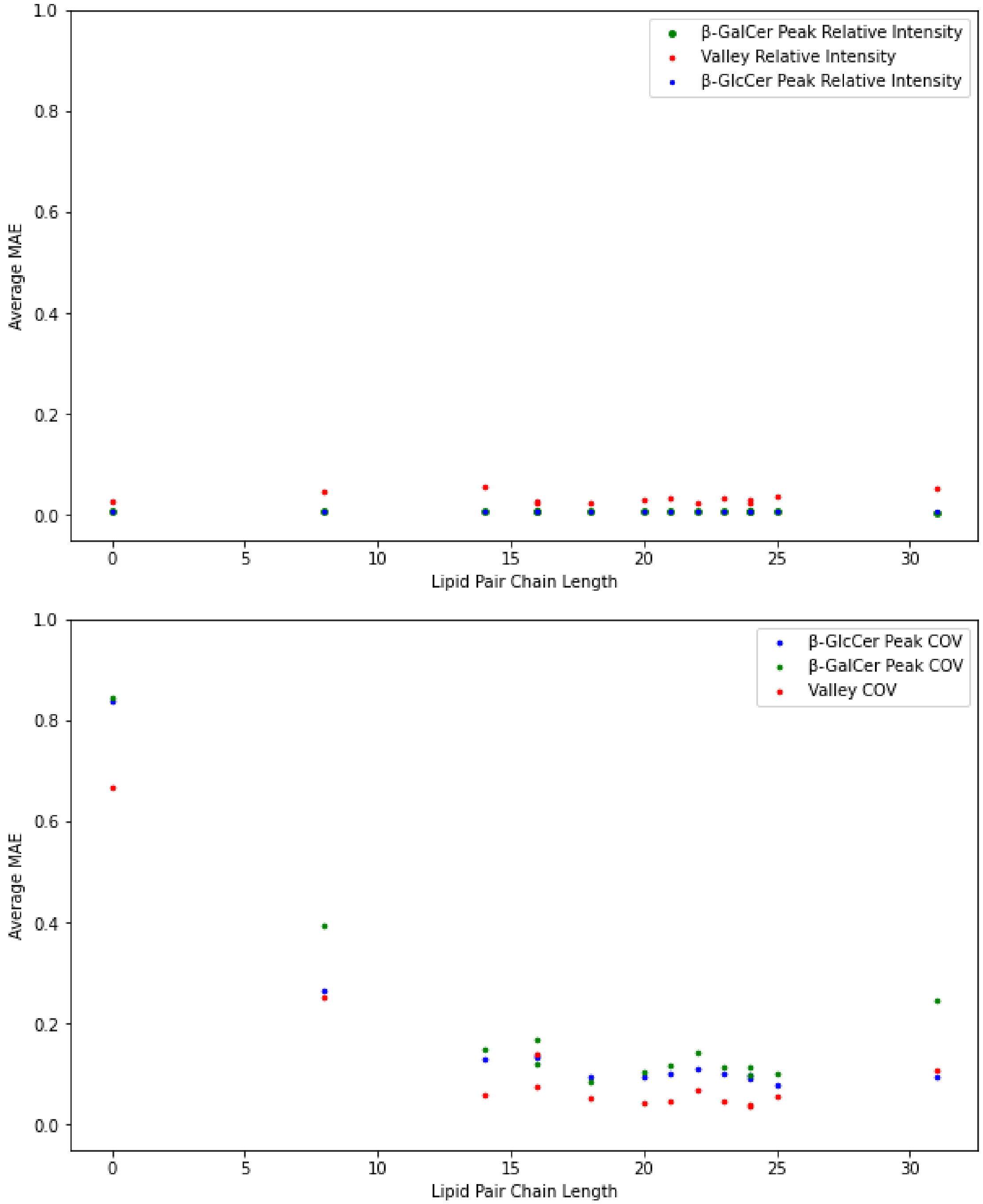


**Figure 20. Average MAE for different measurements inferred from 10 network predictions for each of the test sets left out during cross-validation.Top panel)** Average MAE for different peak and valley COV relative intensities inferred from network predictions and plotted for each unique lipid chain length in the test data. **Bottom Panel)** Average MAE for different peak and valley COV locations inferred from network predictions plotted for each unique lipid chain length in the test data.

Figure 21 also demonstrated that the predicted peak and valley COVs were close to the empirical data at high COV ranges. The network was able to correctly estimate the peak and valley COV values for the majority of the test set lipid pairs. An average MAE of 0.169 was achieved for β-GlcCer peaks; 0.199 was achieved for β-GalCer peaks; 0.120 was achieved for valley COV values, as shown in Table 5 and Figure 20.

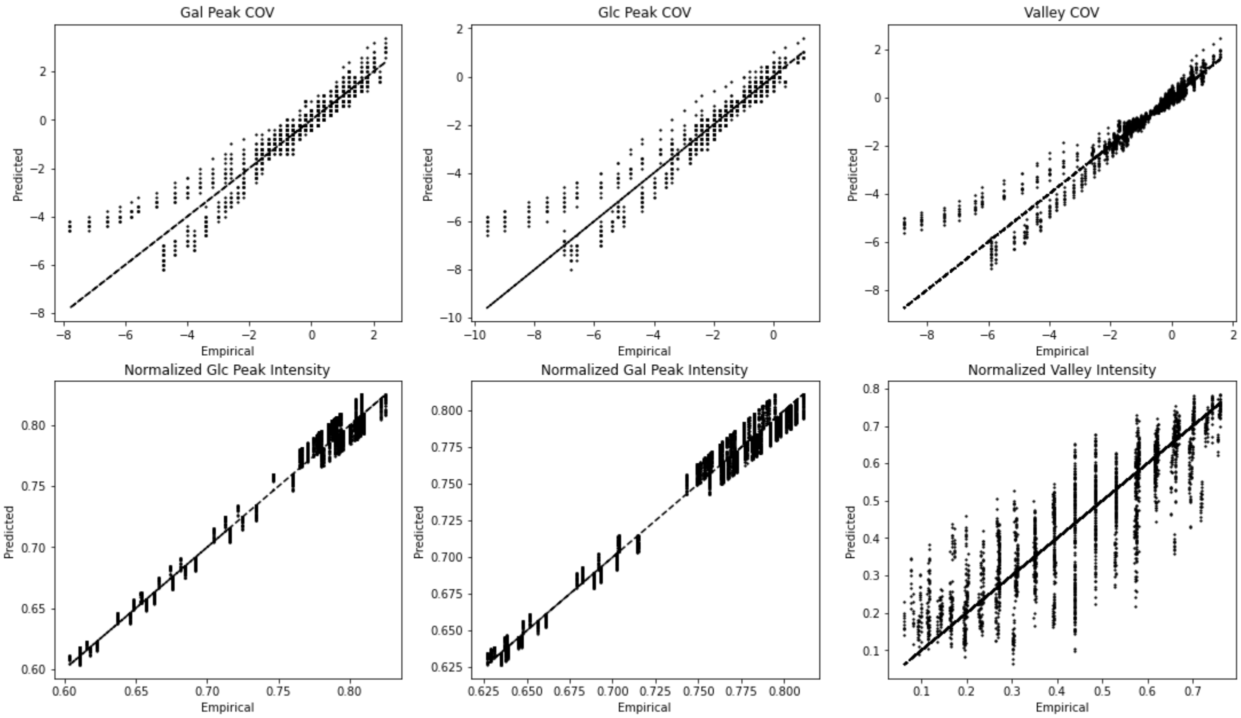


**Figure 21. Aggregated features based on the results of the test set left out during cross-validation versus empirical feature values.** Predicted features (peak COV) and additional features (valley COV and normalized intensities) were extracted from the ionograms generated using predicted gaussian features. Empirical ionograms features were extracted from normalized ionograms ($\tilde{I}$). The trendline was estimated based on the empirical and predicted values shown as the dashed lined.

However, at very low COV values, the estimated peak and COV values deviated from the empirical values. These instances were likely caused by the test sets with β-GlcSph and β-GalSph, lipids without any N-acyl chain as they presented with significantly higher MAE loss than the average MAE of all lipids. Specifically, the corresponding errors for estimating peak COV of β-GlcSph(d18:1/0:0) and β-GalSph(d18:1/0:0) were 0.839 and 0.843, whereas the corresponding errors for estimating the peak COV of β-

GlcCer(d18:1/8:0) and β-GalCer(d18:1/8:0) were 0.265 an 0.394. Further insights into these errors are provided by Figures S2-S5, which display a set of bifurcation diagrams showing the predicted peak and valley COV values from one of the ten networks plotted at each SV for each lipid pair. While the peak and valley COV values of β-GlcSph(d18:1/0:0) and β-GalSph(d18:1/0:0) were estimated to be higher than the actual values, the peak and valley COV values for β-GlcCer(d18:1/8:0) and β-GalCer(d18:1/8:0) were estimated to be lower than those empirically determined.

Multiple reasons could explain the poor prediction performance observed for the lipid pairs with no or shorter N-acyl chains. The most likely reason is that the training dataset contained long to very long-chain lipid stereoisomers. Consequently, the network under fitted these edge case instances. The higher MAE observed in the other edge case of β-GlcCer(d18:1/31:0) and β-GalCer(d18:1/31:0) supports this explanation as it represents the extreme very-long chain edge case (Figure 20). However, given that the error directions for the two datasets were different, lipid-specific reasons could have attributed to this error. For instance, within the training dataset, the COV ramp was restricted to a range of -10 Volts to 10 Volts. Due to the min-max standardization of the values, -10 Volts became the lower bound of prediction. With soft plus function as the activation function, the predicted outputs could never be less than 0 and thus could never be less than the minimum bounded value. Therefore, the network cannot predict COV values lower than -10 and will have trouble estimating COV values close to this boundary. Future work is required to investigate whether normalization of the COV values to a broader range would help reduce such errors. Lastly, β-GlcSph(d18:1/0:0) and β-GalSph(d18:1/0:0) are in fact as distinct subclass (hexosyl-sphingoid bases) of neutral sphingolipids compared to the hexosylceramide subclasses (β-GlcCers and β-GalCers). The lipid pairs are specifically referred to as β-GlcSph(d18:1) and β-GalSph(d18:1), with their structure shown in supplemental Figure S5. These lipid pair have the sugar head group and the sphingoid (d18:1) backbone but lack the N-acyl chain. These structural differences may impact on clustering behavior in the DMS cell thus not conform to what is seen in the hexosyl-ceramide structures.

Furthermore, compared to the empirical data, the average predicted lipid peaks have a smoother relationship with the input features (chain length, SV), indicating that the algorithm had eliminated some noise during the prediction (Figure 20 and Figure 21). This observation was further evident by the distinct SV boundary, where separation occurs in the predicted values compared to the “blurred” empirical boundary. Although the overall predicted separation was close to the empirical values, achieving an average F1 value of 0.72, the mispredictions could be attributed to the noise in the dataset itself. Multiple repeats of these experiments may result in a separation pattern that is closer to the predicted boundaries.

Furthermore, during the empirical annotation of separation labels, data instances with conflicting separation results from the duplicate experiments were labeled as not separable. This cautious labeling possibly had led to some false negatives annotated in the dataset itself, explaining certain instances of the low F1 value but high sensitivity values. Specifically, F1 values were significantly lower for very high chain-length lipid species. High numbers of false positives associated with β-GlcCer(d18:1/25:0) and β-GalCer(d18:1/25:0), β-GlcCer(d18:1/31:0), as well as β-GalCer(d18:1/31:0) resulted in F1 values of 0.448 and 0.718 respectively as reported in Table 6. Empirically, very high chain lipid species have fewer separation instances overall since they require higher SV to achieve separation, meaning that false-positive misclassifications can impact the F1 more greatly than other lipid species. This phenomenon may also be another case of challenges associated with predicting edge cases, as previously discussed.

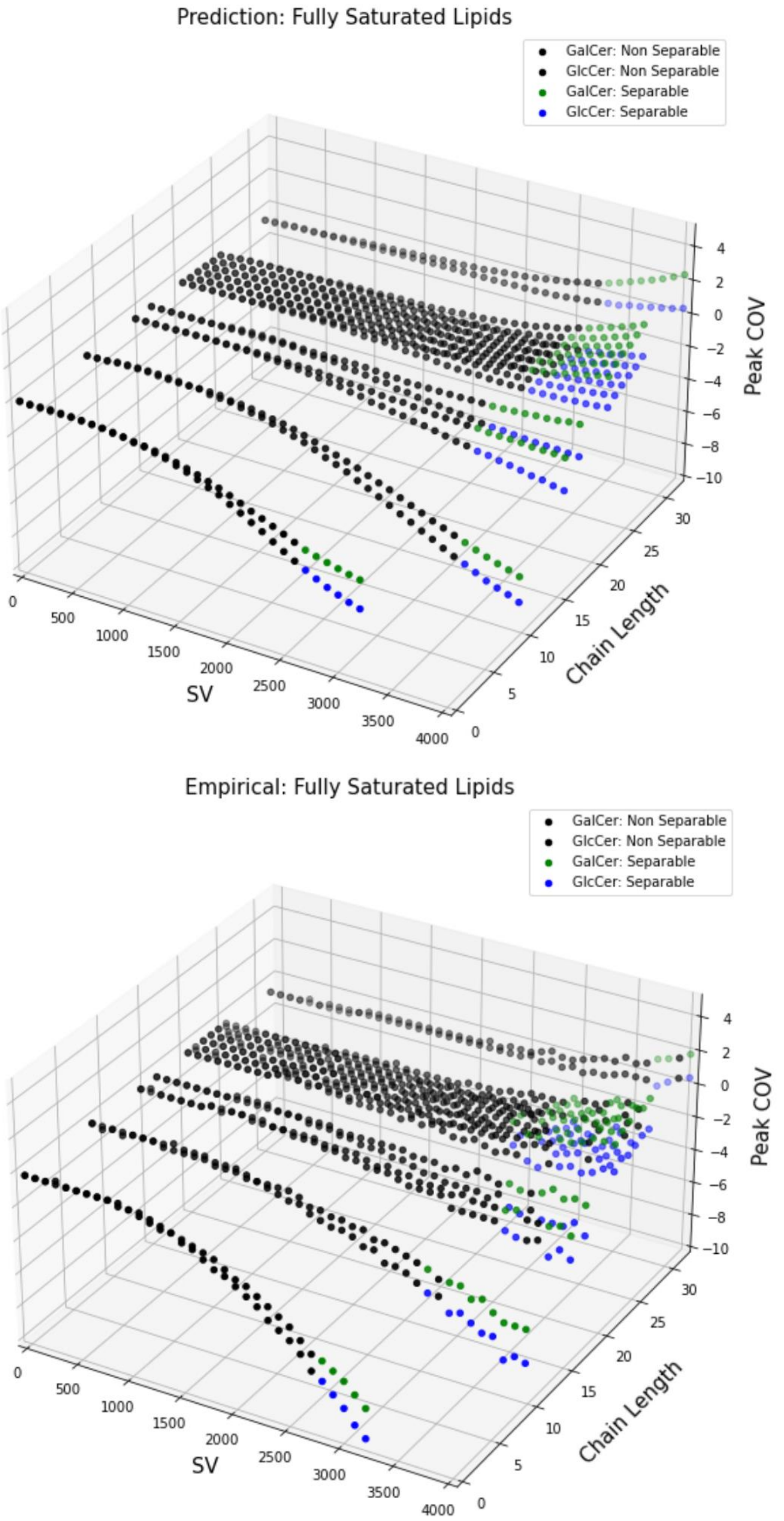


**Figure 22. Three-dimension plot of the average peak COV values and separability for 11 fully saturated lipid pairs across different SVs for the test set left out during cross-validation versus empirical feature values.** ***Top****) Predicted* $u_{Glc}, u_{Gal}$ *as peak COV.* ***Bottom****) Empirical* $u_{Glc}, u_{Gal}$ *as peak COVs determined from peak picking.*

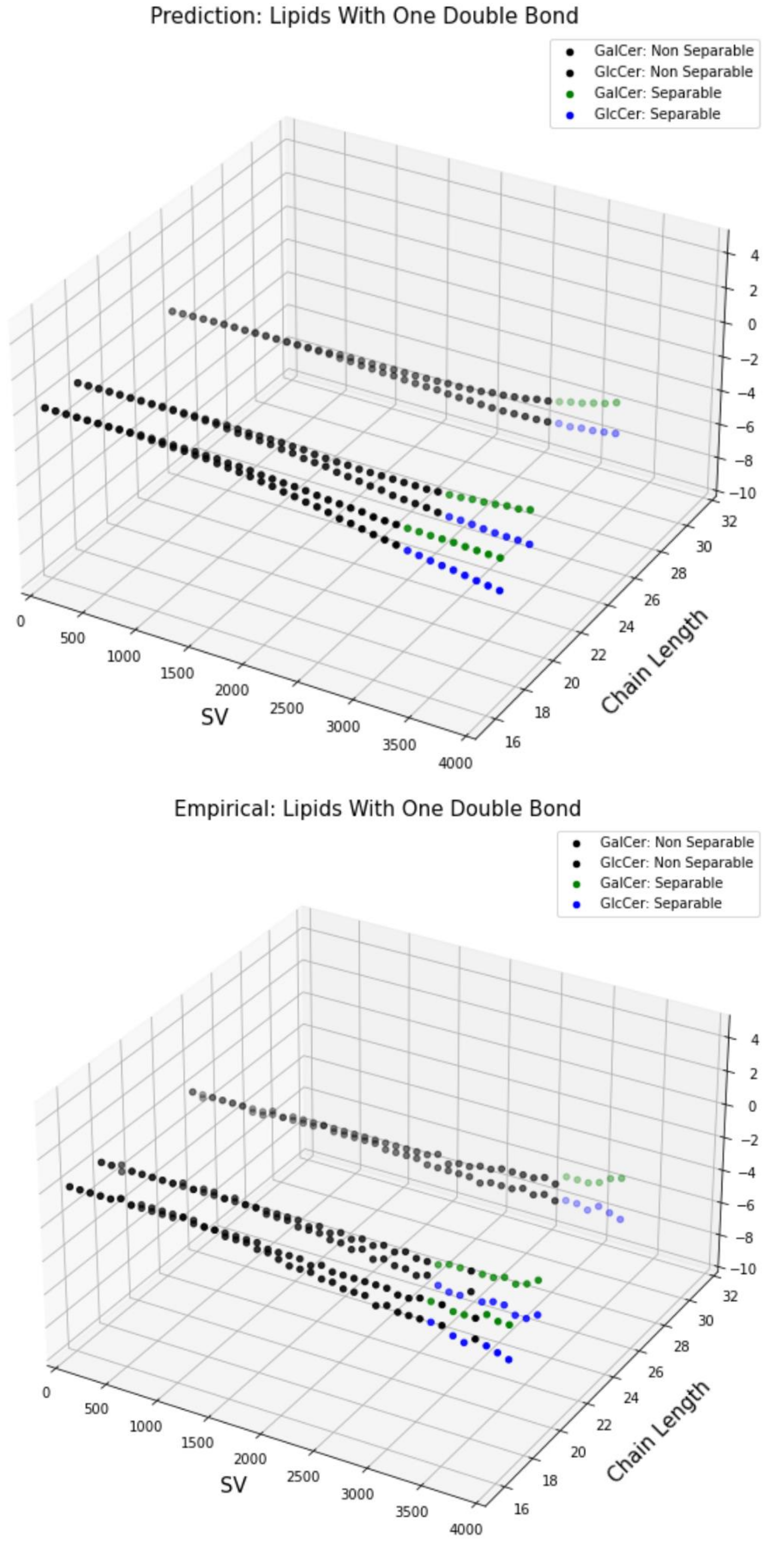


**Figure 23. Three-dimension plot of the average peak COV values and separability for three lipid pairs with one degree of unsaturation across different SVs for the test set left out during cross-validation versus empirical feature values. *Top*)** *Predicted $u_{Glc}, u_{Gal}$ as peak COV.* ***Bottom*)** *Empirical $u_{Glc}, u_{Gal}$ as peak COVs determined from peak picking.*

**Table 6. Separation performance for cross-validation result for each of the lipid pairs predicted by the optimal network.**

| Lipid N-Acyl Chain | F1 | Sensitivity | Specificity | Accuracy |
|---|---|---|---|---|
| **0:0** | 0.794 | 1 | 0.905 | 0.919 |
| **8:0** | 0.727 | 0.633 | 0.969 | 0.889 |
| **14:0** | 0.642 | 1 | 0.877 | 0.889 |
| **16:0** | 0.825 | 0.792 | 0.971 | 0.937 |
| **16:1** | 0.786 | 1 | 0.907 | 0.921 |
| **18:1** | 0.779 | 0.889 | 0.935 | 0.929 |
| **20:0** | 0.698 | 0.778 | 0.926 | 0.905 |
| **21:0** | 0.632 | 0.8 | 0.901 | 0.889 |
| **22:0** | 0.871 | 0.944 | 0.963 | 0.96 |
| **23:0** | 0.833 | 1 | 0.946 | 0.952 |
| **24:0** | 0.777 | 0.778 | 0.963 | 0.937 |
| **24:1** | 0.927 | 1 | 0.972 | 0.976 |
| **25:0** | 0.448 | 1 | 0.875 | 0.881 |
| **31:0** | 0.315 | 1 | 0.87 | 0.873 |
| **Average** | 0.718 | 0.901 | 0.927 | 0.918 |

## 3.3 Effect of Training Dataset Size

The previous section demonstrated that with thirteen pairs of lipid species and the optimized hyperparameters described above, the iDMS neural network can predict gaussian parameters that describe the peak features for new lipid species. However, the minimum number of lipid pairs required to train the network to achieve these results still needed to be investigated. To understand the impact of the number of lipid species on network performance, the network was further trained with different permutations of lipid subsets. All permutation subsets were taken from the cross-validation dataset containing fourteen lipid species. For each permutation, between 2-12 lipid pairs were taken for training, and the remainder of the lipid species from the cross-validation dataset was used for testing. Each permutation was tested on ten separate networks with weights randomly initialized.

The network performance was first averaged across all ten separate networks with weights randomly initialized for each unique permutation. Then, the performance metrics were further averaged across the 100 permutations for each training dataset size. Each evaluated result and confidence interval across training set sizes are shown in Figure 24. The average $R^2$ of β-GlcCer Peak COV stabilized at 0.82 when the training dataset size included ten different lipid species. The average $R^2$ of β-GalCer Peak COV also reached 0.85 using a training set composed of 10 lipid pairs, with slight improvement afforded by the addition of 1 or 2 lipids: $R^2$ of 0.88 for 11 lipid species and $R^2$ of 0.89 for 12 lipid species. Lastly, the network also achieved stable separation results based on training on ten lipid species, as shown by the reported sensitivity, accuracy, and F1 values reaching a consistent value of 0.87, 0.90, and 0.69.

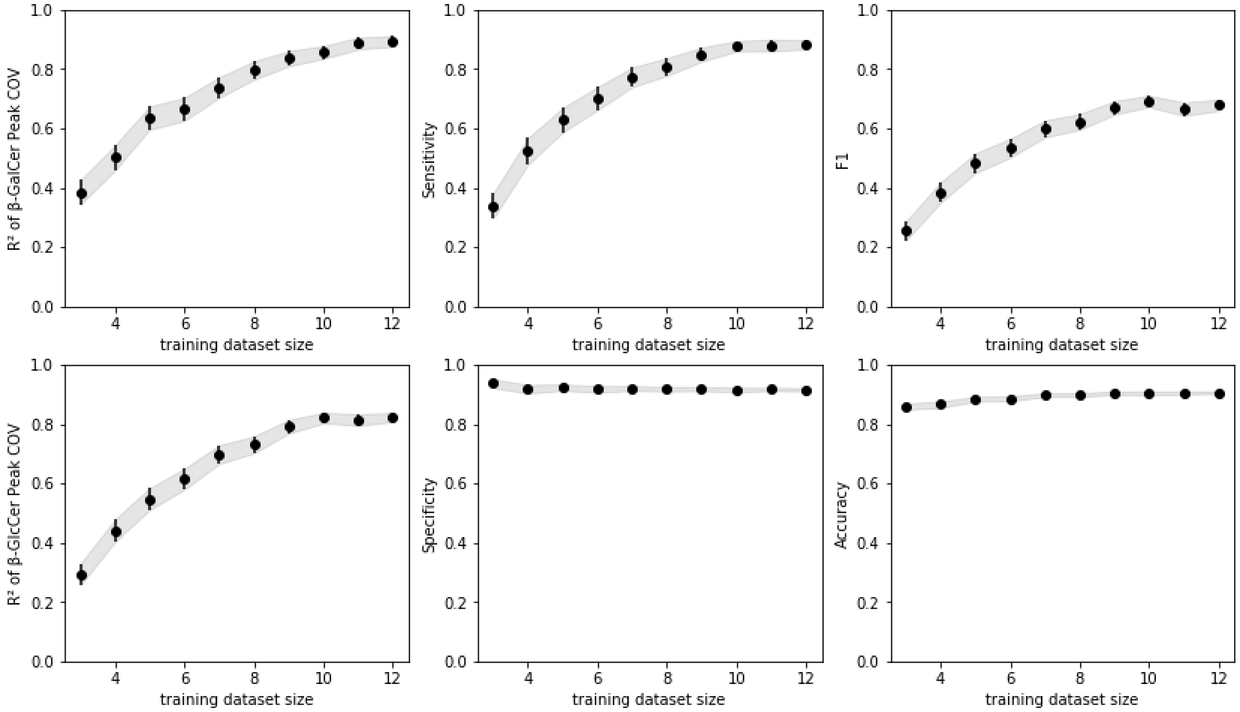


**Figure 24. Effect of the training data size on network performance.** Confidence intervals and means were aggregated from training and testing on 100 different permutations of the lipid combinations.

It is important to note that this is an imbalanced dataset, containing mostly negative instances of non-separating ionograms that failed to meet the 50% peak to intensity evaluation criterion. Specifically, the average percentage of positive instances is only 10% across all lipid species, and, as such, the dataset introduces some limitations in predicting these classes. As such, the reduction in specificity is inevitable due to the trade-off between recall and precision. When the model improves its true positive prediction, it will also introduce new errors with misclassifying true negatives. However, the specificity only slightly decreases from 0.93 to 0.91 as the number of lipid species in the training dataset increases from 3 to 12. Therefore, this slight reduction is not concerning.

## 3.4 Testing Data Results

To ensure that the hyperparameters selected for the final tuned network was not biased towards the cross-validation dataset, the network performance under these hyperparameters were further assessed using a testing dataset that was never used to train any of the networks. Thus, ten separate networks with the same optimum hyperparameters were trained on fourteen lipid species and tested on the remainder of three lipid species during this process.

The structures of these lipid pairs in the testing dataset are shown in Figure 25A. The separation results of the ten networks are shown as box and whisker plots in Figure 25B. β-GlcCer(d18:1/18:0) and β-GalCer(d18:1/18:0) yielded the lowest sensitivity of 0.9, while the other two lipid species yielded a sensitivity of 1.0 (Figure 25). Similar to the cross-validation results, the F1 value was noticeably reduced for longer chain lipid species, as the β-GlcCer(d18:1/18:0) and β-GalCer(d18:1/18:0) pair achieved an F1 value of 0.873, while the β-GlcCer(d18:1/26:0) and β-GalCer(d18:1/26:0) pair achieved an F1 value of 0.506. Thus, while the sensitivity metric resulted in a heavier penalty towards false negatives, the F1 metric resulted in half of the false negative penalty weights towards false positive values. The low F1 value and high sensitivity from the β-GlcCer(d18:1/26:0) and β-GalCer(d18:1/26:0) pairs were presumably due to the presence of false positives in the prediction results. Specifically, the network predicted separation at a lower SV range that was not empirically labeled as separating. As previously mentioned, because conflicting results between duplicated experiments were labeled as not separating, the annotated dataset could have a higher number of false negatives that consequentially led to a greater number of the estimated instances to be counted as false positives.

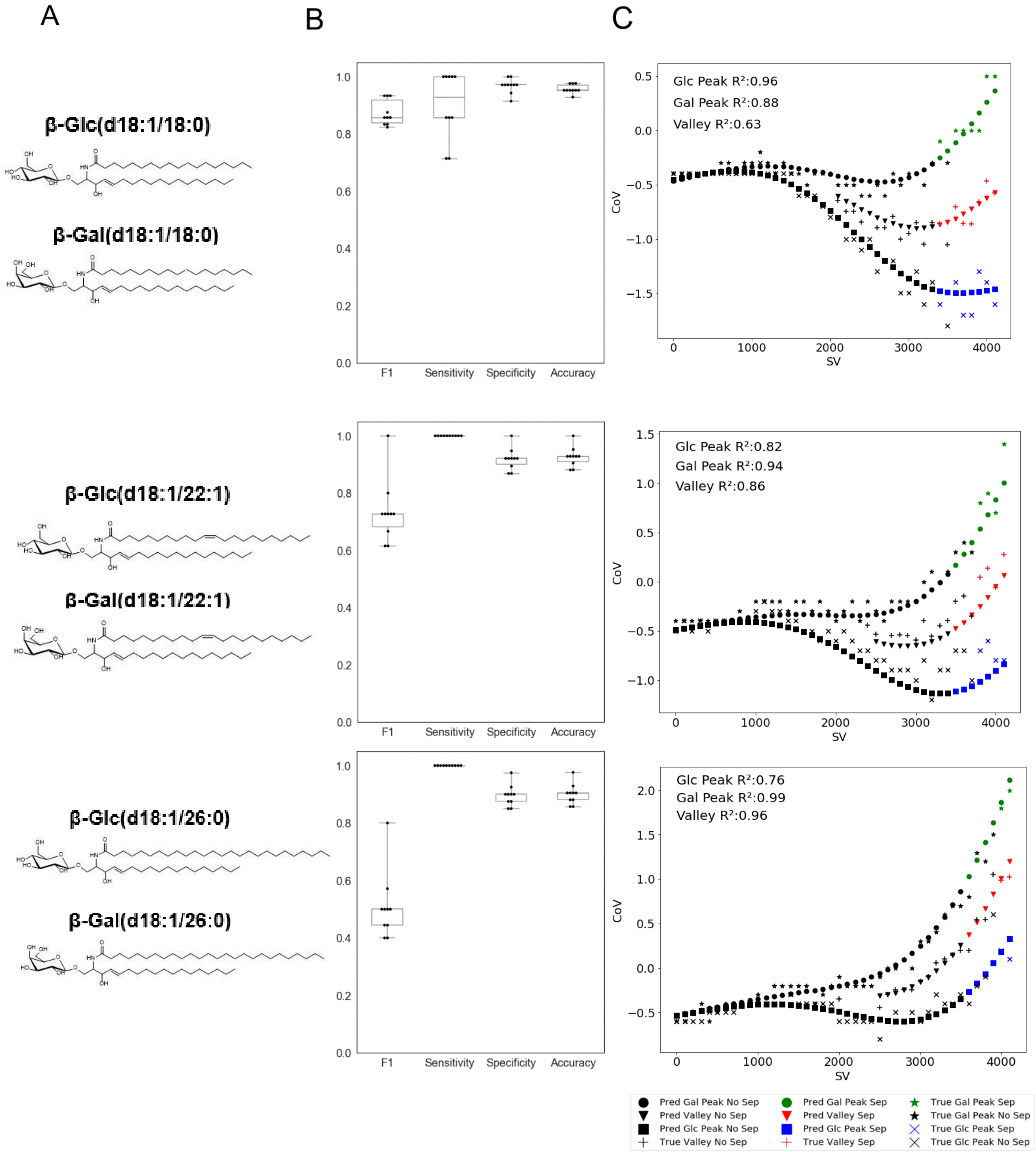


**Figure 25. Test set results of the tuned network.A)** Chemical structure of the lipid pairs of the corresponding test set. **B)** Data are presented as box and whisker plots demonstrating the annotated separation quality based on the ten prediction outputs of the tuned neural network. The box indicates the first and third interquartile, and the whisker indicates the min and max values**. C)** Dispersion plots of both the empirical and average (n=10) predicted β-GlcCer peak COV, β-GalCer peak COV, and valley COV at each SV.

The randomness across the ten separate networks caused by the weight initialization step is revealed from the whiskers in Figure 25B. For example, there were two networks for the β-GlcCer(d18:1/26:0) and β-GalCer(d18:1/26:0) pairs and three networks for each of the remaining two lipid pairs, that resulted in F1 values below the third interquartile range of the ten predictions. In addition, the long lower whiskers for the sensitivity value of β-GlcCer(d18:1/18:0) and β-GalCer(d18:1/18:0) further highlighted how certain initialized weights occasionally underperformed. As expected, the accuracy and specificity values have short whiskers due to the imbalanced classes containing significantly more negative separation data points.

Performance reduction due to weight initialization was rare since only two to three networks significantly underperformed. However, this can still be concerning when the network is implemented to predict new lipid species without synthetic standards. Although one can train on the cross-validation data and store the optimal performing weights based on the testing set, this method favors weights representing the training dataset, resulting in a higher bias towards the testing set. One method to avoid selecting a network with underperforming weights during prediction would be to use a simple ensemble method. Here, the average prediction of the ten separate networks was taken to reduce model bias. Rather than predicting each of the ten networks and averaging the performance measures, the average predictions were made from ten separate networks. The performance was evaluated based on the average prediction. Figure 25C demonstrated that the extracted peak and valley COV values based on the ensembled prediction across the ten networks were very close to their empirical values for all three lipid species. Table 7 and Table 8 showed that using ensembled network results improved the $R^2$ value and reduced the MAE for all three lipid stereoisomer peak/valley COV estimations. Although using the ensembled method did not significantly improve the separation metric of all lipid species, as shown in Table 9, the reported separation metrics are very close to the average value of the ten networks. Therefore, using this ensembled approach could prevent the bias prediction of a single network due to the weight initialization step.

**Table 7. Test set resulting R$^2$ values with and without ensemble methods.** Comparison of the average performance of 10 network predictions against performance of the averaged ensembled predictions of 10 networks.

| R$^2$ | GlcCer d(18:1/18:0) & GalCer d(18:1/18:0) | | GlcCer d(18:1/22:1) & GalCer d(18:1/22:1) | | GlcCer d(18:1/26:0) & GalCer d(18:1/26:0) | | Average of Three Lipids | |
|---|---|---|---|---|---|---|---|---|
| | Average of 10 iterations | Ensemble prediction score | Average of 10 iterations | Ensemble prediction score | Average of 10 iterations | Ensemble prediction score | Average of 10 iterations | Ensemble prediction score |
| **Glc Peak COV** | 0.960 | 0.962 | 0.820 | 0.842 | 0.756 | 0.763 | 0.845 | 0.856 |
| **Gal Peak COV** | 0.859 | 0.879 | 0.942 | 0.945 | 0.984 | 0.985 | 0.928 | 0.933 |
| **Valley COV** | 0.580 | 0.632 | 0.859 | 0.862 | 0.957 | 0.958 | 0.799 | 0.817 |
| **Glc Peak Relative Intensity** | 0.986 | 0.990 | 0.981 | 0.981 | 0.986 | 0.988 | 0.984 | 0.986 |
| **Gal Peak Relative Intensity** | 0.981 | 0.982 | 0.983 | 0.983 | 0.983 | 0.980 | 0.982 | 0.982 |
| **Valley Relative Intensity** | 0.907 | 0.923 | 0.864 | 0.855 | 0.764 | 0.767 | 0.845 | 0.848 |

**Table 8. Test set resulting MAE with and without ensemble methods.** Comparison of the average performance of 10 network predictions against performance of the averaged ensembled predictions of 10 networks.

| MAE | GlcCer d(18:1/18:0) & GalCer d(18:1/18:0) | | GlcCer d(18:1/22:1) & GalCer d(18:1/22:1) | | GlcCer d(18:1/26:0) & GalCer d(18:1/26:0) | | Average of Three Lipids | |
|---|---|---|---|---|---|---|---|---|
| | Average of 10 iterations | Ensemble prediction score | Average of 10 iterations | Ensemble prediction score | Average of 10 iterations | Ensemble prediction score | Average of 10 iterations | Ensemble prediction score |
| **Glc Peak COV** | 0.0876 | 0.0787 | 0.130 | 0.124 | 0.0868 | 0.0748 | 0.101 | 0.093 |
| **Gal Peak COV** | 0.0990 | 0.0694 | 0.118 | 0.0969 | 0.101 | 0.0741 | 0.106 | 0.08 |
| **Valley COV** | 0.108 | 0.0842 | 0.157 | 0.147 | 0.0992 | 0.0690 | 0.121 | 0.1 |
| **Glc Peak Relative Intensity** | 0.00564 | 0.0061 | 0.00724 | 0.00959 | 0.00594 | 0.00570 | 0.006 | 0.007 |
| **Gal Peak Relative Intensity** | 0.00739 | 0.00615 | 0.00883 | 0.00591 | 0.007556 | 0.00598 | 0.008 | 0.006 |
| **Valley Relative Intensity** | 0.0577 | 0.0473 | 0.0706 | 0.0672 | 0.0712 | 0.0623 | 0.067 | 0.059 |

**Table 9 Test set separation results with and without ensemble methods.**Comparison of the average performance of 10 network predictions against performance of the averaged ensembled predictions of 10 networks.

| | GlcCer d(18:1/18:0) & GalCer d(18:1/18:0) | | GlcCer d(18:1/22:1) & GalCer d(18:1/22:1) | | GlcCer d(18:1/26:0) & GalCer d(18:1/26:0) | |
|---|---|---|---|---|---|---|
| | Average of 10 iterations | Ensemble prediction score | Average of 10 iterations | Ensemble prediction score | Average of 10 iterations | Ensemble prediction score |
| **F1** | 0.873 | 0.933 | 0.733 | 0.727 | 0.506 | 0.500 |
| **Sensitivity** | 0.900 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| **Accuracy** | 0.957 | 0.976 | 0.926 | 0.928 | 0.900 | 0.905 |
| **Specificity** | 0.970 | 0.970 | 0.91 | 0.92 | 0.89 | 0.90 |

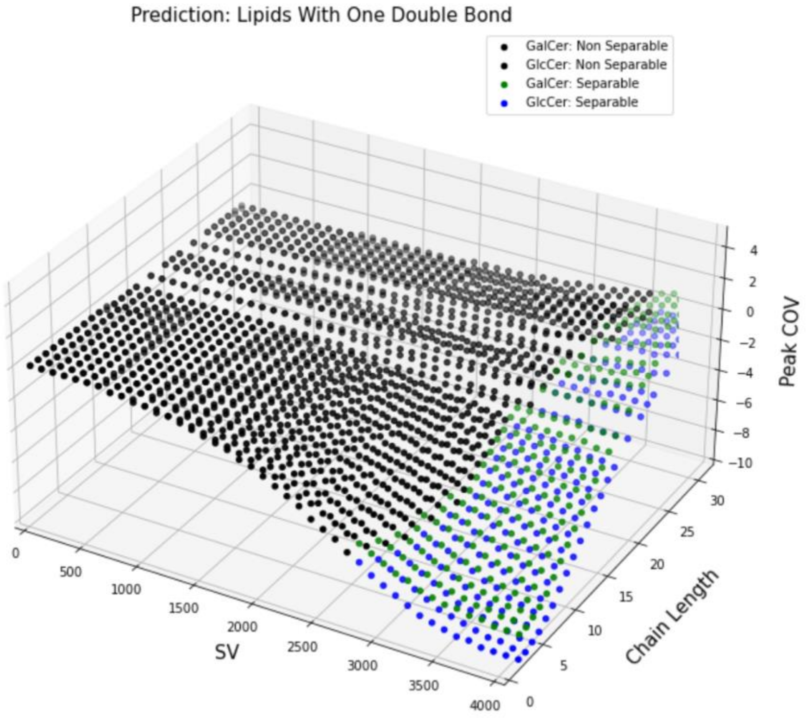
These results demonstrated that aggregating prediction from ten tuned networks improves overall performance and reduces randomness associated with weight initialization. As such, the final tuned model includes the ensemble of ten different network predictions, while each network was trained across the entire training dataset of 17 lipid species.

## 3.5 Prediction of DMS Parameters for Lipid Species

To summarize, the optimized network has five layers containing ten neurons each and uses the soft plus function as both the inner and outer activation functions and implemented the Glorot method for weight initialization. During cross-validation, the tuned network predicted the COV values that maximized the relative intensity of a lipid isomer at a given SV with an $R^2$ value of greater than 0.84. Furthermore, from the network predictions we were also able to infer the SV values where separation was possible most of the time, explicitly achieving an F1 value of 0.71 across all lipid species. The final ensemble network, which used the average of ten network predictions, further reduced randomness in the weight initialized steps and improved the testing dataset results. Predictions from the final ensemble network were made on three separate lipid species from the testing dataset. The final prediction determined the Glc Peak COV and Gal Peak COV location accurately and yielded an average $R^2$ value of 0.845 and 0.928, respectively, for Glc Peak COV and Gal Peak COV. Furthermore, the prediction also improved the separation metrics resulted in an average F1 value of 0.86. These results demonstrated that the ensemble network successfully simulated DMS behavior for lipid pairs that the network has not seen.

As such, the final ensemble network was retrained with the entire dataset to learn the relationships between DMS machine parameters, lipid structure, and normalized ion intensity. The network was further used to simulate DMS behavior for lipid pairs for which no synthetic standard is available to empirically optimize DMS parameters but are expected to be present in biological matrices. These results are shown in Figure 26.

All these theoretical lipids were not separable at SV ranges lower than 2000. With the increase in chain length, the SV values, where separation was theoretically observed, become more limited to the higher ranges. A lower peak COV value was observed across all lipid species at separable SV ranges for β-GlcCer as compared to β-GalCer. The peak COV values for fully saturated lipids yielded slightly more positive values than lipids with one double bond. At time of thesis submission, Avanti Polar Lipids had agreed to synthesize three of these novel lipids to validate empirically the capacity of iDMS to report machine parameters for lipids that do not have synthetic standards. The final predicted results can be implemented to determine the optimal COV and SV values for LC-ESI-DMS-MS/MS for separating and subsequent selected reaction mode methods for quantifying these glycosphingolipid isomeric species. Further discussion of how these results can be used is elaborated in Chapter 4.



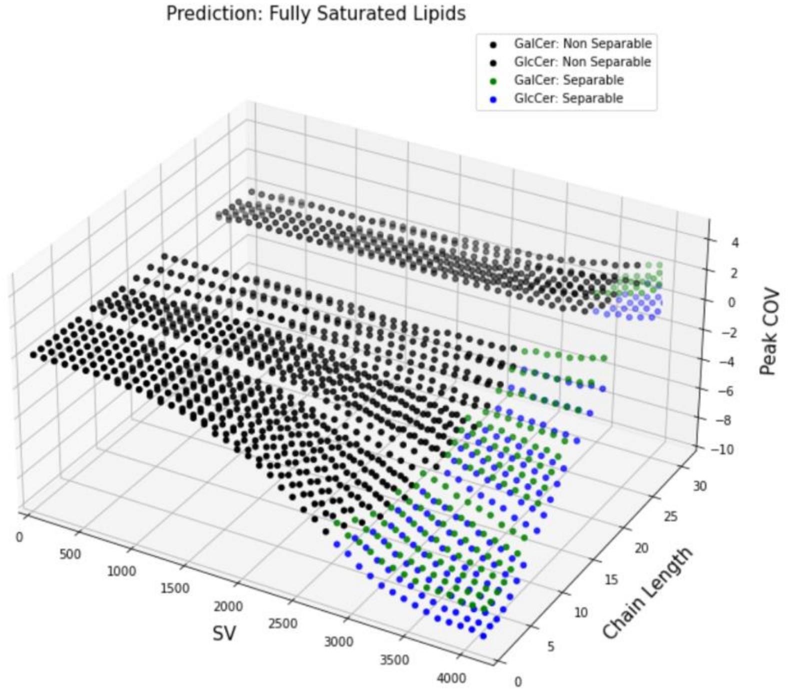


**Figure 26. Three-dimension plot of the average peak COV values and separability predicted by ten ensemble tuned network for lipids for which no synthetic standards are available.Top)** Predicted $u_{Glc}, u_{Gal}$ as peak COV for lipids with one degree of unsaturation. **Bottom)** Predicted$u_{Glc}, u_{Gal}$as peak COV for fully saturated lipids.

# Chapter 4: Discussion

## 4.1 Algorithm Application

This work presents a machine learning approach to learn the relationships between lipid molecular structures and the DMS machine parameters. This *in-silico* simulation method can replace the labor-intensive and expensive manual method development described previously (see Xu et al[9]) that is required for DMS-MS/MS method development. The final tuned network was applied to determine normalized ionograms of 47 pairs of glycosphingolipids stereoisomers where synthetic standards were not available.

In development of iDMS, application of the trained network for determining machine parameters of these lipid species first requires the generation of normalized ionograms from the predicted mean at each SV. The 50% separation rule was further applied to each generated normalized ionogram, informing the mass spectrometer user of the SV range where the stereoisomeric pair can achieve separation. For SV values that are separable, the ionograms can also inform the user of the analyte selectivity and normalized sensitivity response of the lipid species at each COV value. The selectivity mentioned in this section refers to the purity of the ion intensity detected, whereas the sensitivity refers to the measured magnitude of the ion intensity, rather than separation metrics for predictions. The user can make an informed decision to select the SV and COV parameter values for quantifying target lipid. The DMS machine parameter values can be further used in LC-ESI-DMS-MS/MS in selected reaction monitoring mode to separate the target lipid species from its isomer and other isobaric lipid species and achieve accurate relative quantification of the lipid species.

Based on the cross-validation results and the testing set results, the prediction yields the highest F1 and $R^2$ values for prediction features for lipids with chain lengths between 8-25 carbons. Previous cross-validation results also suggested that predicted peak COV values tend to be more positive for lipids with chain length under eight carbons. The prediction may slightly overestimate the SV separation range for lipids with chain lengths above 25 carbons. These observations should be considered by the user when selecting the optimal SV/COV values.

In this thesis, the final tuned network was applied to pairs of lipid species with varying chain lengths between 0-32 carbons and either zero or one degree of unsaturation. Although some of these species such as β-GlcCer(d18:1/17:0) and β-GlcCer(d18:1/19:0) [46] have previously been quantified using normal phase HPLC, determining the DMS machine parameter values of these lipid species allows for LC-DMS-MS/MS targeted quantification of these lipids in the presence of other isobaric species. At time of submission, we are awaiting custom-generated synthetic standards in collaboration with Avanti Polar Lipids to definitively demonstrate that iDMS can be used to identify DMS machine parameters for lipids that do not have synthetic standards readily available for empirical optimization. Furthermore, many of these lipid species can theoretically exist in nature, but their physical presence has not been validated due to presence other isobaric species. The simulated DMS machine parameter can be implemented to validate these lipid existences and better understand their biological importance.

## 4.2 Limitations

As previously mentioned, features predicted by the network were used to generate normalized ionograms for the simulations. The main limitation when predicting normalized ionograms is that the normalized intensity does not inform the user about the absolute signal levels, which can also vary as a function of DMS parameters. For example, one might theoretically get separation of a lipid pair at a certain SV, but the signal level might be lower than desired for accurate quantification. Particularly if lipid abundances in real LC-DMS-MS/MS samples are lower than in the direct-infusion DMS-calibration experiments, absolute signal levels may be a concern. Therefore, users should also consider the absolute signal levels measured in the calibration data, and in the future, absolute signal level could be added as a prediction target of the network.

Furthermore, both empirical ionograms and predicted normalized ionograms were labelled based on the criteria that the intensity ratio between the valley and smallest peak must be less than 0.5. However, this annotation is based on the naïve assumption that the two lipid stereoisomers share similar concentrations in the biological sample and the lipid species have similar ionization efficiency. When a sample has significantly higher amounts of a single lipid species over its isomer or contains a pair of lipids with significantly different ionization efficiency, the valley-to-peak ratio can significantly shift from its normalized values.  Consequentially, the separation criteria should be limited to situations where the lipid isomers have a similar range of measured intensities. Otherwise, it could result in different annotations when applied to the normalized intensity and the empirically measured intensity (Figure 27).

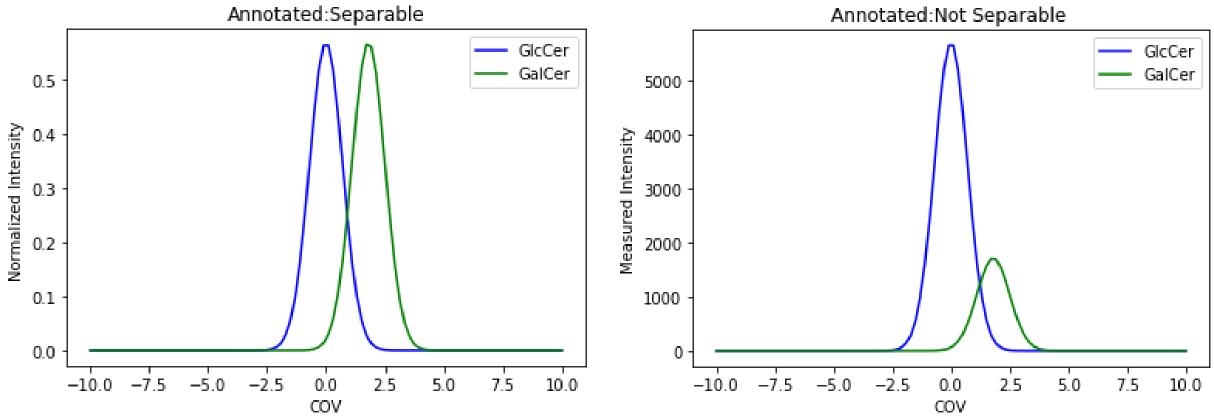


**Figure 27.Different separation annotation for empirical ionograms vs normalized ionograms.** Sample data was generated with two gaussian with mean values of 0 and 1.8, and standard deviation values of 0.7 and 0.7 respectively for β-GlcCer and β-GalCer. The measured intensity was generated by multiplying the β-GlcCer sample by 3000 and β-GalCer sample by 10000.

Because ionograms were normalized as Gaussian distributions, the normalized values can be considered to be the fraction in which a single lipid species is detected at a specific COV value. Therefore, when selecting the optimal COV value that maximizes signal impurity, the user should not only consider maximizing the fraction of the targeted analyte but must also consider minimizing the fraction of its isomers. Empirically, at the peak of a lipid's COV value, the likelihood of finding the other lipid species is approximately 5-10%. Selecting the maximal peak COV value of a lipid species as the DMS quantification parameter is not concerning when there are similar quantities of the lipid isomers in the sample. However, as previously mentioned, caution must be taken when the measured intensity value of the two lipid isomers differs in magnitudes. Figure 28 shows the normalized ionograms observed when empirically, ten times greater amounts of β-GlcCers were present in the mixture compared to β-GalCers. In this situation, while the normalized ionograms demonstrate that selecting the β-GalCer peak at a COV value of 1.2 results in a small proportion (0.067) of β-GlcCer and a great proportion of β-GalCer (0.78), the ratio of the measured β-GlcCer intensity to measured β-GalCer intensity is still 0.53.

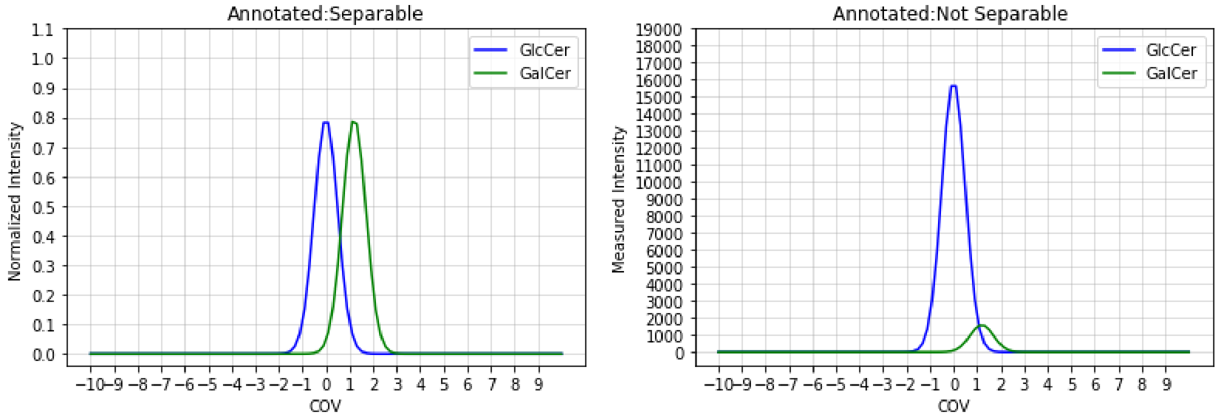


**Figure 28. Quantification using the predicted COV at maximal normalized intensity for samples with high concentration or ionization efficiency of its isomer may lead to low empirical selectivity of the target lipid.**Sample data was generated with two gaussian with mean values of 0 and 1.2, and standard deviation values of 0.5 and 0.5 respectively for β-GlcCer and β-GalCer. The measured intensity was generated by multiplying the β-GlcCer sample by 20000 and β-GalCer sample by 2000.

Therefore, when handling standards that may contain isomers of magnitudes difference in ionization efficiency or concentration, it is recommended that when selecting for the COV value, one should prioritize minimizing the target isomer's likelihood over simply maximizing the likelihood of the target analyte. To achieve this, users could consider choosing a COV value that is slightly more positive than the Gal Peak COV value or slightly more negative than the Glc Peak COV value. In these situations, selectivity of the analyte should be considered over the ion sensitivity.

## 4.3 Promises and Future Work

This work aimed to determine the optimal SV/COV parameter combination for sphingolipid isomeric pairs of β-GlcCer(d18:1/X:0) and β-GalCer(d18:1/X:0), with the primary objective focused on optimizing for the selectivity of a target analyte for its targeted quantifications. As this work was built on a previous optimization method by Xu et al., the four DMS machine parameter values (cell temperature, transport gas concentration, DMS resolution enhancement, DMS offset) were optimized sequentially with a single lipid; these four parameters were then kept constant during the selection of

SV/COV values [10]. However, these four parameter combinations also depend on the structural properties of the lipid that can affect their behavior in the DMS cell, which, in turn, further affects sensitivity and selectivity. Therefore, the optimal parameter combinations may change between lipid species. Further incorporation of these four features into the current network model will require additional experimental runs to generate training data. As demonstrated by Figure 24, ten lipid species pairs may be required for each set of training data. Each set contains ionograms that span the SV range of 2000-4000 for each combination of the additional DMS machine parameter values. Moreover, should these number of lipid standards not be available; in that case, I propose that transfer learning could be implemented to expand on the previous training by fixing the inner layer and training explicitly for the outer layer. While this approach can be applied to other datasets for determining selectivity, a different approach incorporating additional features is required to predict for machine sensitivity. These features may incorporate additional parameters outside the scope of DMS, such as ionization efficiency of the lipid or collision energy of the mass spectrometer.

In addition, future work could also revolve around developing a similar machine learning approach to determine DMS machine parameter values for separating the β-glycosphingolipid isomers from their alpha anomers. Previous studies have shown that α-GalCer were derived from Bacteroids fragilis found in the human gut microbiome[15]. Furthermore, it has been suggested that low levels of α-GalCer could simulate the activation of natural killer T-Cells (iNKT) when presented by CD1d and is hypothesized to be essential for the homeostasis of those cells [40]. Currently, there is therapeutic interest in the use of α-GalCers for tumor treatments [40]. The development of a machine learning model specific for quantifying α-HexCer would therefore be of considerable utility but would still require the synthesis of at least some standards of the alpha anomers. However, as with incorporating additional DMS features, it may be possible to apply transfer learning to determine the selectivity of different DMS parameters for the alpha anomers in cases where only a small subset of lipid standards is available.

Finally, a similar machine learning approach could be applied to additional lipid families where many pairs of isomers exist. In this work, we investigated the β-glucosylceramide and β-galactosylceramide lipid subclasess with a d18:1 sphingosine backbone. However, many other lipid isomeric pairs also differ in sphingoid base. For instance, β-glucosylceramide and β-galactosylceramide can also be found in nature with a d18:2 sphingoid backbone. iDMS can be implemented to further determine DMS machine parameters for these other pairs of neutral glycosphingolipids.

Previously, Walker et al.[41] demonstrated, using machine learning, that a drug's physiochemical properties (e.g. cell permeability, solubility, polar surface area, etc.) could be used to predict optimized DMS machine parameter values. The high correlation of his model further suggests the strong relationship between the DMS machine parameter values and their dependence on the physicochemical properties and analyte structure. Although this work was restricted to the physical structure of the N-acyl chain, it may also be desirable to incorporate other physiochemical properties in the future. One advantage to the direct use of physicochemical properties could be method generalization across different lipid families. However, this work would require the availability of a training dataset that includes different categories and subclasses of lipid species.

In summary, the current thesis demonstrates the possibility of using machine learning for DMS machine parameter optimization in lipidomics. Laboratories can further use the *in silico*-optimized parameters to perform accurate targeted mass spectrometry to separate lipid isomers without synthesizing standards.

# References

[1] A. H. Merrill, “Sphingolipid and glycosphingolipid metabolic pathways in the era of sphingolipidomics,” *Chem. Rev.*, vol. 111, no. 10, pp. 6387–6422, 2011.

[2] A. Schwarz and A. H. Futerman, “Distinct roles for ceramide and glucosylceramide at different stages of neuronal growth,” *J. Neurosci.*, vol. 17, no. 9, pp. 2929–2938, 1997.

[3] M. Eckhardt, “The role and metabolism of sulfatide in the nervous system,” *Mol. Neurobiol.*, vol. 37, no. 2–3, pp. 93–103, 2008.

[4] A. E. Ryckman, I. Brockhausen, and J. S. Walia, “Metabolism of glycosphingolipids and their role in the pathophysiology of lysosomal storage disorders,” *Int. J. Mol. Sci.*, vol. 21, no. 18, pp. 1–31, 2020.

[5] J. Do, C. McKinney, P. Sharma, and E. Sidransky, “Glucocerebrosidase and its relevance to Parkinson disease,” *Mol. Neurodegener.*, vol. 14, no. 1, pp. 1–16, 2019.

[6] W. Westbroek *et al.*, “A new glucocerebrosidase-deficient neuronal cell model provides a tool to probe pathophysiology and therapeutics for Gaucher disease,” *DMM Disease Models and Mechanisms*, vol. 9, no. 7. pp. 769–778, 2016.

[7] Y. Cui and Q. Wan, “NKT cells in neurological diseases,” *Front. Cell. Neurosci.*, vol. 13, no. May, pp. 1–10, 2019.

[8] S. Ishihara *et al.*, “ α-Glycosylceramides Enhance the Antitumor Cytotoxicity of Hepatic Lymphocytes Obtained from Cancer Patients by Activating CD3 − CD56 + NK Cells In Vitro ,” *J. Immunol.*, vol. 165, no. 3, pp. 1659–1664, 2000.

[9] H. Xu *et al.*, “DMS as an orthogonal separation to LC / ESI / MS / MS for quantifying isomeric cerebrosides in plasma and cerebrospinal fluid,” vol. 60, 2019.

[10] E. L. Kean, “Separation of gluco- and galactocerebrosides by means of borate thin-layer chromatography,” *J. Lipid Res.*, vol. 7, no. 3, pp. 449–452, 1966.

[11] E. M. Kaye and M. D. Ullman, “Separation and quantitation of perbenzoylated glucocerebroside and galactocerebroside by high-performance liquid chromatography,” *Anal. Biochem.*, vol. 138, no. 2, pp. 380–385, 1984.

[12] K. Zama *et al.*, “Simultaneous quantification of glucosylceramide and galactosylceramide by normal-phase HPLC using O-phtalaldehyde derivatives prepared with sphingolipid ceramide N-deacylase,” *Glycobiology*, vol. 19, no. 7, pp. 767–775, 2009.

[13] A. H. Merrill, M. C. Sullards, J. C. Allegood, S. Kelly, and E. Wang, “Sphingolipidomics: High-throughput, structure-specific, and quantitative analysis of sphingolipids by liquid chromatography tandem mass spectrometry,” *Methods*, vol. 36, no. 2 SPEC. ISS., pp. 207–224, 2005.

[14] R. L. Shaner *et al.*, “Quantitative analysis of sphingolipids for lipidomics using triple quadrupole and quadrupole linear ion trap mass spectrometers,” *J. Lipid Res.*, vol. 50, no. 8, pp. 1692–1707, 2009.

[15] J. Von Gerichten *et al.*, “Diastereomer-specific quantification of bioactive hexosylceramides from bacteria and mammals,” *J. Lipid Res.*, vol. 58, no. 6, pp. 1247–1258, 2017.

[16] A. P. Bowman, R. R. Abzalimov, and A. A. Shvartsburg, “Broad Separation of Isomeric Lipids by High-Resolution Differential Ion Mobility Spectrometry with Tandem Mass Spectrometry,” *J. Am. Soc. Mass Spectrom.*, vol. 28, no. 8, pp. 1552–1561, 2017.

[17] B. Schneider, E. Nazarov, F. Londry, P. Vouros, and T. Covey, “Differential Mobility Spectrometry/Mass Spectrometry History, Design Optimization, Simulations, And Applications,” *Wiley Online Libr.*, vol. 35, pp. 687–737, 2014.

[18] “SelexION / SelexION + Technology for SCIEX Triple Quad ™ and QTRAP ® Systems User Guide,” no. January, 2019.

[19] M. Warren and W. Pitts, “A logical calculus of the ideas immanent nervous activity,” *Bull. Math. Biophys.*, vol. 5, pp. 115–133, 1943.

[20] F. Rosenblatt, “The perceptron: probabilistic model for information storage and organization in the brain,” *Psychol. Rev.*, vol. 386, no. 65, 1958.

[21] T. Szandała, “Review and comparison of commonly used activation functions for deep neural networks,” *Stud. Comput. Intell.*, vol. 903, pp. 203–224, 2021.

[22] M. L. Minsky and S. A. Papert, *Perceptrons: An Introduction to Computational Geometry*. 1969.

[23] D. E. Rumelhart, G. E. Hintont, and R. J. Williams, “Learning Representations by Back-Propagating Errors,” *Nature*, vol. 323, no. 9, pp. 533–535, 1986.

[24] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” *Proc. 13th Int. Conf. Artifical Intell. Stat. 2010*, vol. 9, pp. 249–256, 2010.

[25] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet Classification with Deep Convolution Neural Networks,” *Proc. 25th Int. Conf. neural Inf. Process. Syst.*, vol. 1, pp. 1097–1105, 2012.

[26] L. Lu, Y. Shin, Y. Su, and G. E. Karniadakis, “Dying ReLU and initialization: Theory and numerical examples,” *Commun. Comput. Phys.*, vol. 28, no. 5, pp. 1671–1706, 2020.

[27] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” *JMLR Work. Conf. Proc.*, vol. 28, 2013.

[28] X. Glorot, A. Bordes, and Y. Bengio, “Deep Sparse Rectifier Neural Network,” *Proc. fourteenth Int. Conf. Artif. Intell. Stat.*, vol. 15, pp. 315–323, 2011.

[29] M. Ghosh and J. Berger, *Statistical Decision Theory and Bayesian Analysis.*, vol. 83, no. 401. 1988.

[30] S. Bermejo and J. Cabestany, “Oriented principal component analysis for large margin classifiers,” *Neural Networks*, vol. 14, no. 10, pp. 1447–1461, 2001.

[31] P. J. Huber, “Robust Estimation of a Location Parameter,” *Ann. Math. Stat.*, vol. 35, no. 1, pp. 73–101, 1964.

[32] M. D. Zeiler, “ADADELTA: An Adaptive Learning Rate Method,” *CoRR abs/1212.5701*, 2012.

[33] B. T. Polyak, “Some methods of speeding up the convergence of iteration methods,” *USSR Comput. Math. Math. Phys.*, vol. 4, no. 5, pp. 1–17, 1964.

[34] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.

[35] J. C. Duchi, E. Hazan, and Y. Singer, “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization,” *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, 2011.

[36] T. Tieleman and G. Hinton, “COURSERA: Neural Networks for Machine Learning. Technical report,” 2012.

[37] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, pp. 1026–1034, 2015.

[38] C. R. Boor, *A practical guide to splines*. 1978.

[39] M. Oss, A. Kruve, K. Herodes, and I. Leito, “Electrospray ionization efficiency scale of organic compound,” *Anal. Chem.*, vol. 82, no. 7, pp. 2865–2872, 2010.

[40] T. Kawano *et al.*, “CD1d-restricted and TCR-mediated activation of V(α)14 NKT cells by glycosylceramides,” *Science (80-. ).*, vol. 278, no. 5343, pp. 1626–1629, 1997.

[41] S. W. C. Walker *et al.*, “Determining molecular properties with differential mobility spectrometry and machine learning,” *Nat. Commun.*, vol. 9, no. 1, pp. 1–7, 2018.

# Appendix: Supplemental Figures

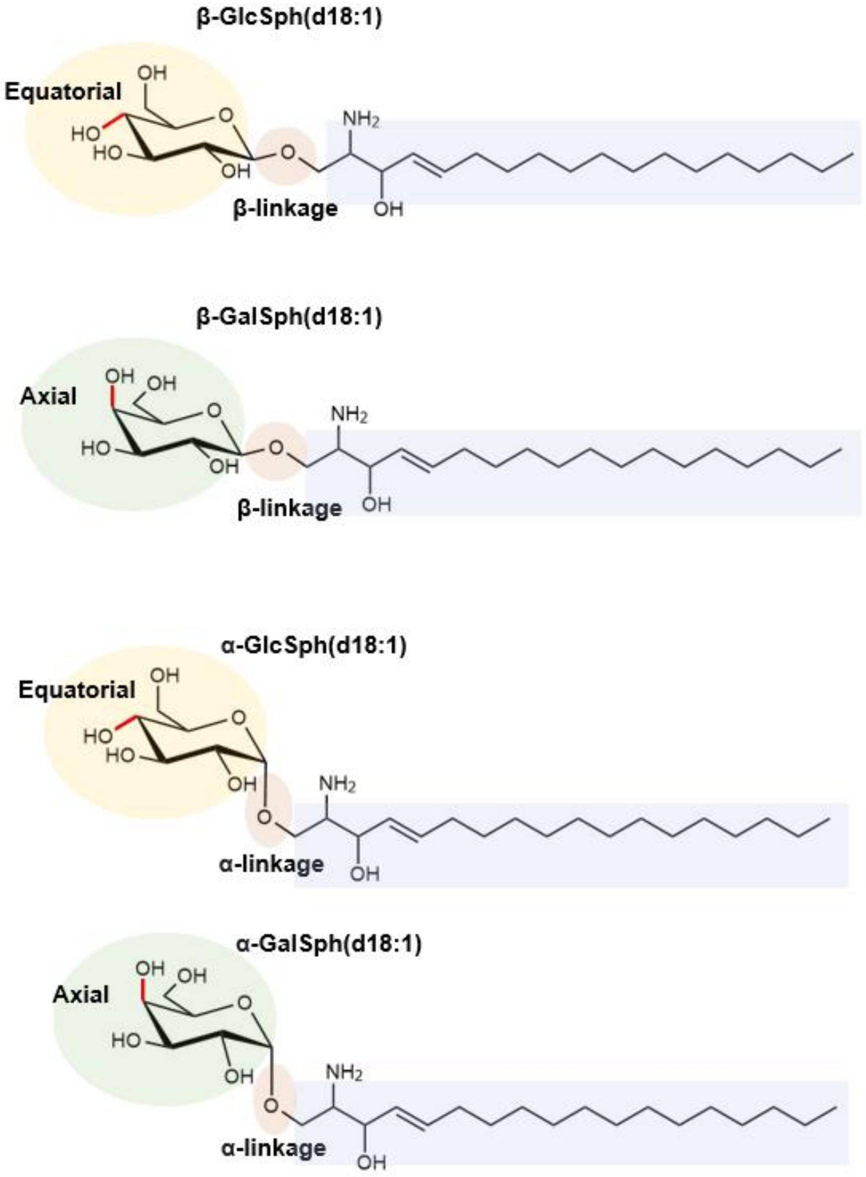


**Fig S1. Physiochemical structure of glycosphingolipid β-GlcSph, β-GalSph, β-GlcSph, α-GalSph and α-GlcSph.** *The sugar head group is denoted by the yellow shaded area, where green is for the galactose head group and yellow is the glucose headgroup. The glycosidic linkage is denoted by the salmon shaded area. The blue shaded area is the sphingoid (d18:1) base.*

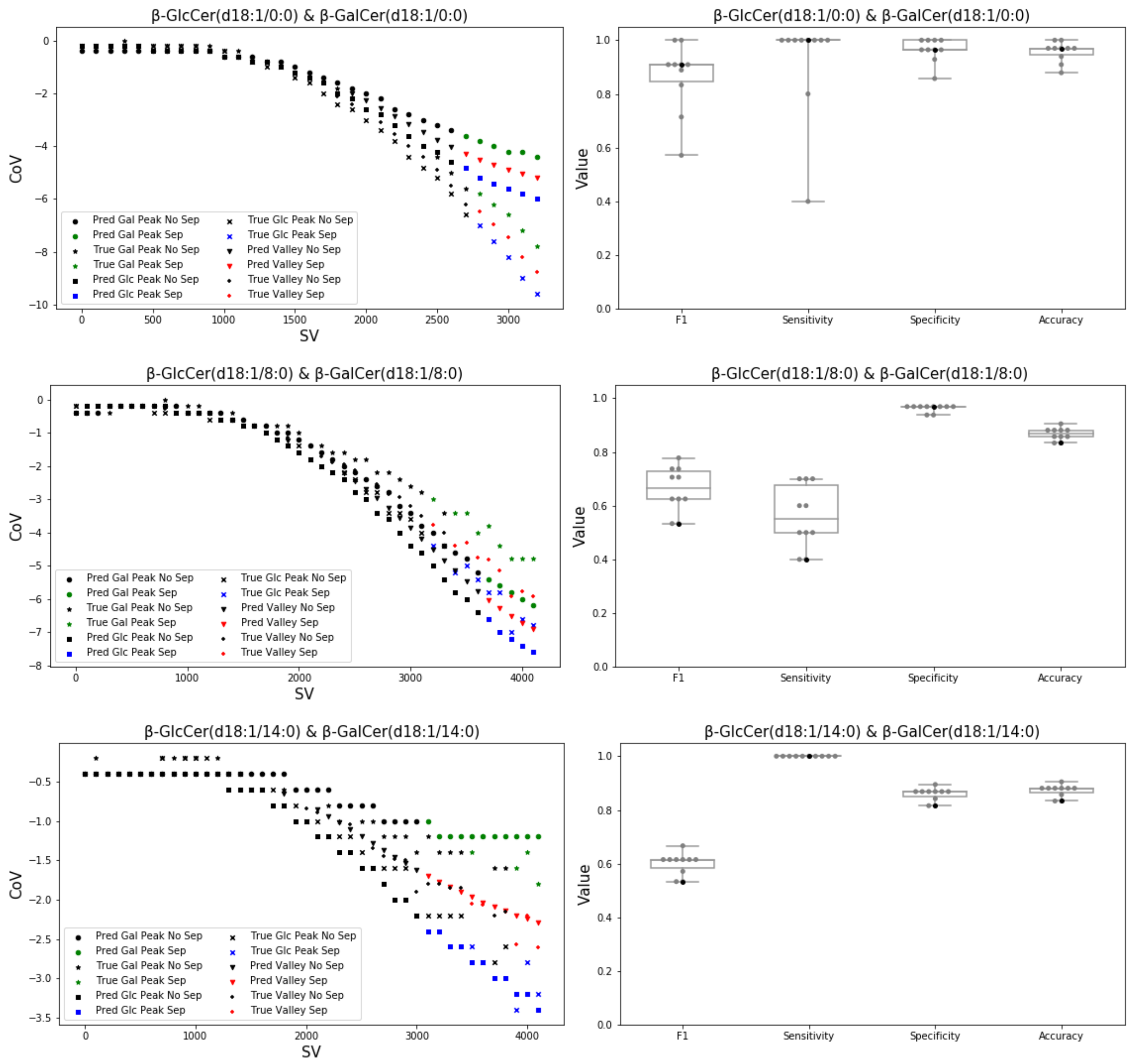


**Fig S2. Dispersion plot of test results of four lipid pairs (β-HexSph(d18:1/0:0), β-HexCer(d18:1/8:0), β-HexCer(d18:1/14:0)) from one of the ten networks compared to the separation results aggregated from all ten networks.**

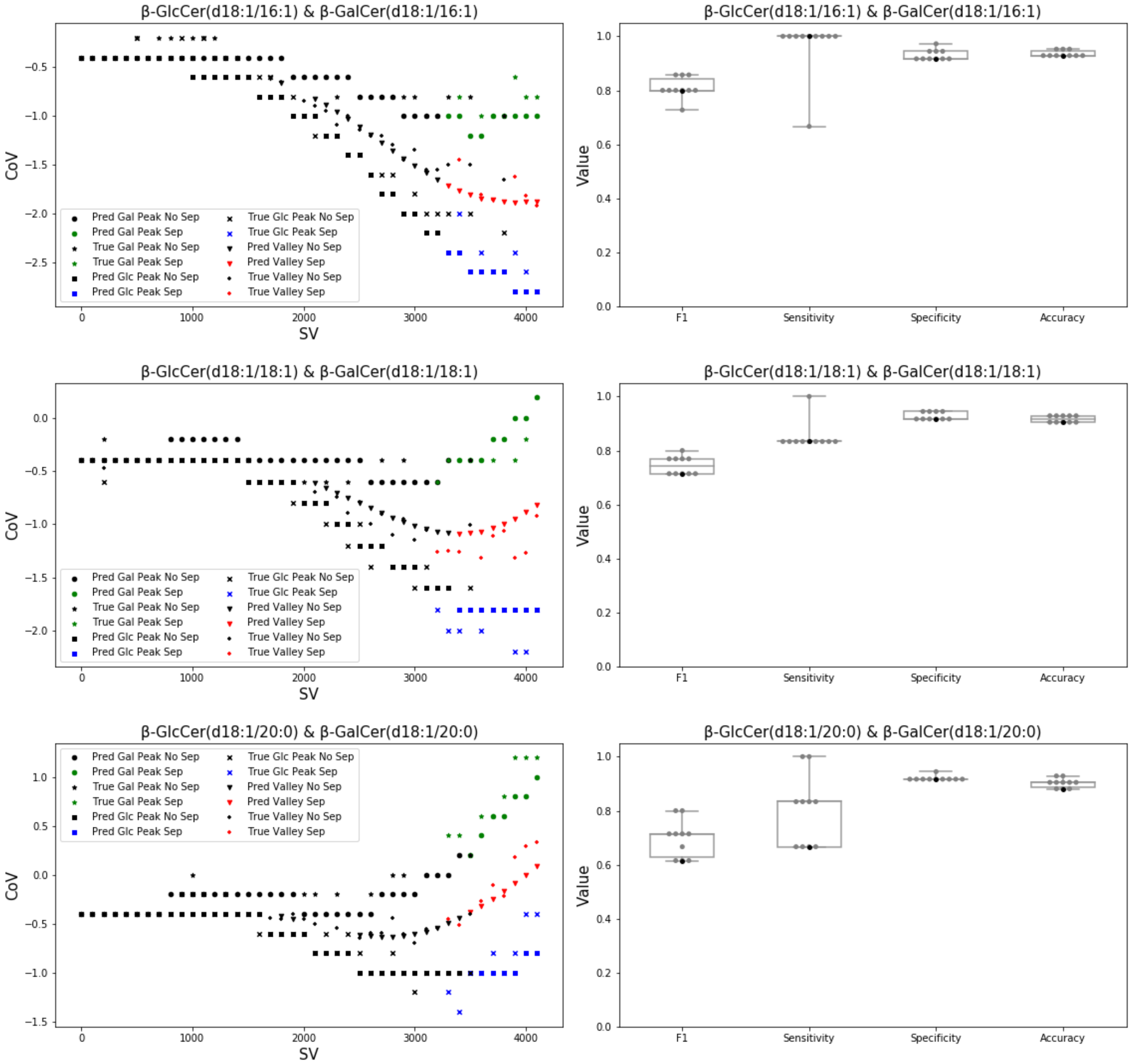


**Fig S3. Dispersion plot of test results of four lipid pairs (β-HexCer(d18:1/16:0), β-HexCer (d18:1/16:1), β-HexCer(d18:1/18:1), β-HexCer(d18:1/20:0)) from one of the ten networks compared to the separation results aggregated from all ten networks.**

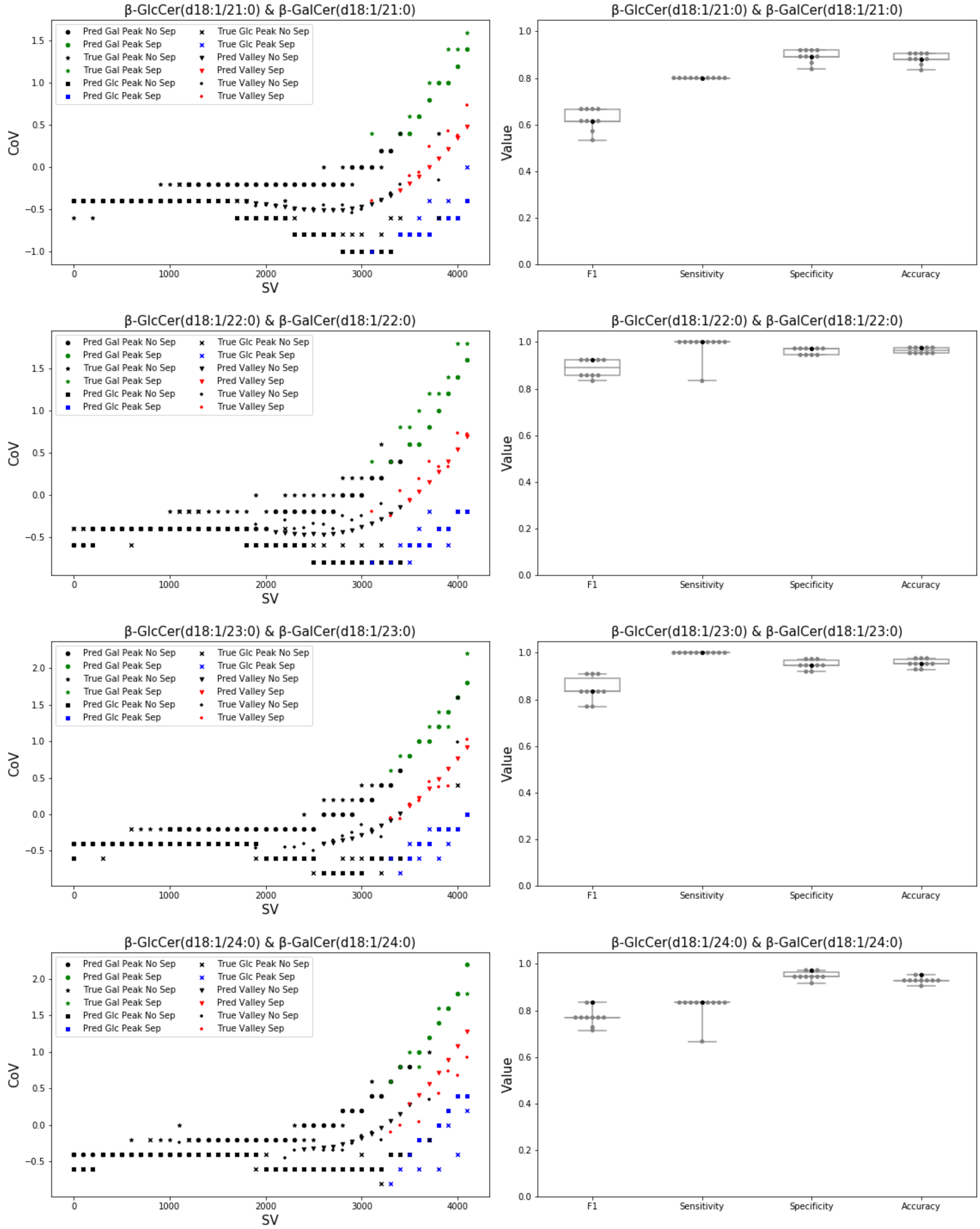


**Fig S4. Dispersion plot of test results of four lipid pairs (β-HexCer(d18:1/21:0), β-HexCer(d18:1/22:0), β-HexCer(d18:1/23:0), β-HexCer(d18:1/24:0)) from one of the ten networks and the separation results aggregated from all ten networks.**

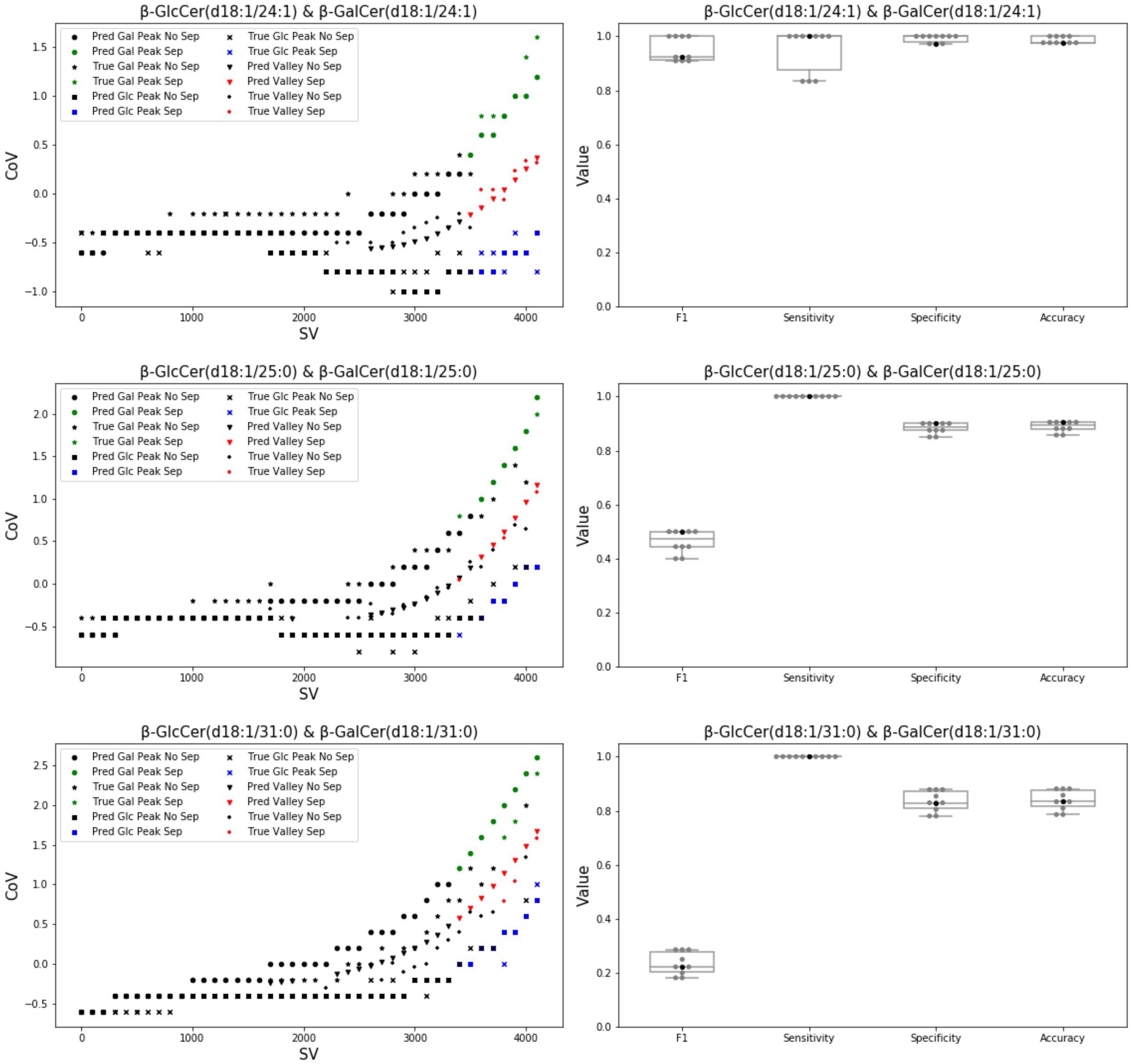


**Fig S5. Dispersion plot of test results of four lipid pairs (β-HexCer(d18:1/24:1), β-HexCer (d18:1/25:0), β-HexCer(d18:1/31:0)) from one of the ten networks and the separation results aggregated from all ten networks.**