OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

INF FAKULTÄT FÜR INFORMATIK

Toolbox zur Auflösung von Multi-Analyte-Peaks in Kapillar-Gel-Elektrophorese

Masterarbeit

Max Frick

1. März 2021

Intelligent Cooperative Systems
Computational Intelligence

Betreuende Professorin:	Prof. Dr.-Ing. habil. Sanaz Mostaghim
Betreuer:	Dr.-Ing. Christoph Steup
Betreuer:	Dr.-Ing. Heiner Zille

glyXera GmbH

Betreuer:	Dr. rer. nat. Erdmann Rapp
Betreuer:	Dipl.-Ing. Alexander Behne

Max Frick: *Toolbox zur Auflösung von Multi-Analyte-Peaks in Kapillar-Gel-Elektrophorese*

Otto-von-Guericke Universität
Intelligent Cooperative Systems
Computational Intelligence

glyXera GmbH

Magdeburg, 1. März 2021.

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Max Frick Magdeburg, 1. März 2021

Vorwort

Allen voran möchte ich mich bei Professorin Sanaz Mostaghim von der Fakultät für Informatik der Otto-von-Guericke-Universität Magdeburg und Dr. Erdmann Rapp von der glyXera GmbH für die Gelegenheit bedanken, diese Masterarbeit durchführen zu dürfen.

Besonderen Dank gilt meinen Betreuern Christoph Steup, Heiner Ziller und Alexander Behne, die wiederholt wertvolles Feedback sowie informatisches und bioinformatisches Wissen zur Anfertigung dieser Arbeit beigesteuert haben.

Des Weiteren möchte ich mich bei Robert Kottler, René Hennig, Andreas Bock und Robert Burock bedanken, die mir mit ihrem umfangreichen Detailwissen über die experimentelle Glykananalytik beratend zur Seite standen.

Ein spezieller Dank geht an Jana Tuchscherer für ihre Vorbereitung von Proben, die es am Ende leider doch nicht in die Arbeit geschafft haben.

Nicht zuletzt möchte ich mich bei meinen Freunden und meiner Familie bedanken, die mich über den Verlauf meines Studiums fortwährend unterstützt haben.

Abstract

Proteins are involved in many chemical processes essential for life. They consist of one or more chains of amino acid residues linked together by peptide bonds. These residues are often chemically modified by so-called post-translational modification (PTM). This modification alters the properties and function of the protein. The most diverse, most common and most important PTM is the glycosylation of proteins: The process of enzymatically attaching carbohydrate molecules, so-called glycans, to a protein.

It is estimated that more than half of all known human proteins are glycosylated. Glycoproteins play a key role in a variety of cellular processes ranging from protein folding to immune response. Even thought the study of glycoproteins and glycans are still in their infancy, glycan profiling is already used in a wide array of applications, such as the design of pharmaceutical products.

Today there are various techniques for analysing glycans in biological samples. Although they differ in their methodologies, they share a common goal: To distinguish and measure each component in a mixture based on their physical or chemical properties. In capillary gel electrophoresis (CGE) measurements are plotted as a time-value function in which analytes appear as peaks. The degree to which two peaks are separated is called the resolution. However, some components cannot be resolved this way due to their similar separation properties. They appear as overlapping peaks that consist of more than one analyte, referred to as multi-analyte peaks.

This work aims to provide a toolbox to resolve multi-analyte peaks in CGE-based data using Gaussian mixed models (GMM). To this end two particle swarm optimization (PSO) techniques and an expectation–maximization (EM) algorithm are investigated as a means to model glycan peaks in the form of a constrained or unconstrained optimization problem. Furthermore, two novel hyperparameter optimizer have been developed that estimate the number of overlapping peaks.

Through evaluation it was found that the best results could be achieved via a two-step approach: At first, the number of overlapping peaks are estimated using an EM-based hyperparameter optimiser. Then a GMM can be derived

by minimising the root-mean-square error between the multi-analyte peaks and the model itself. This is achieved through PSO. This approach achieved very good results for simple multi-analyte peaks which visibly consisted of more than one glycan. However, problems appeared for highly overlapping structures with the number of peaks being underestimated.

Nonetheless, the developed toolbox provides a solid and expendable foundation for future research. Various GMMs for a single multi-analyte peak can already be calculated, to guide an expert to find the best solution.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

APTS	8-Aminopyren-1,3,6-trisulfonsäure, 7	CE	capillary electrophoresis, 6–9
CFG	*Consortium for Functional Glycomics*, XI, 4	CGE	capillary gel electrophoresis, 1, 2, 6–9, 24, 26, 35, 37, 38, 47, 53, 56, 57, 59
EA	evolutionary algorithms, 18	EM-Algorithmus	Expectation-Maximization-Algorithmus, IX, 14–18, 28, 29, 52, 57
EMG	Exponentially Modified Gaussian, 9, 15, 18, 24	GMM	gaussian mixture model, IX, 10, 14–16, 24–26, 28, 31, 32, 35, 37, 42, 43, 45, 47, 49, 51, 52, 54, 56–58

GNB	Gleichungs-nebenbedingungen, 13, 14, 25, 26, 29, 40, 42, 43, 45, 46, 48, 56, 59, 95–99	HPLC	high performance liquid chromatography, 9
LC	liquid chromatography, 6, 15	LIF	laser-induced fluorescence, 7, 24, 26, 35, 37, 38, 47, 53, 56, 57, 59
LLH-Funktion	Log-Likelihood-Funktion, 15, 18, 28, 30, 42, 43, 45, 47, 48, 51, 52, 57	LOQ	limit of quantification, 27, 40
MS	mass spectrometry, 6, 15	PSO	particle swarm optimization, 18, 20, 21, 28, 30, 31, 39, 40, 43, 52, 69
RMSE	root-mean-squared error, XI, 29, 41, 42, 45–48, 51, 57, 58	SNR	signal-to-noise ratio, 40
SPSO-2011	Standard-PSO 2011, 21, 31, 41, 48, 52, 69	SSE	sum-of-squares error, IX, 12, 29

UNB	Ungleichungs-nebenbedingungen, 13, 14, 27, 31–33, 40	WLS	weighted least squares, X, 25, 40

1 Einleitung

Ein Protein ist Biomolekül, welches, als ein Baustein des Körpergewebes, an einer Vielzahl von fundamentalen Funktionen des Lebens beteiligt ist. Proteine bestehen aus Aminosäuren-Ketten, die nach ihrer Synthese weitere Veränderungen erfahren können. Diese Anpassungen werden posttranslationale Modifikationen genannt, deren häufigste, vielfältigste und wohl komplexeste Form die Glykosylierung von Proteinen ist: Die enzymatische Verknüpfung von Zucker-Strukturen, sogenannten Glykanen, mit Proteinen [1].

Es wird vermutet, dass mindestens die Hälfte aller Proteine glykosyliert sind. Dies überrascht wenig, spielt die Glykosylierung doch eine essentielle Rolle in der Funktionsweise dieser Glykoproteine, die in solch wichtigen biologischen Prozessen wie der Immunantwort involviert sind. Außerdem kann die Veränderung einer bestimmten Glykosylierung, beispielsweise aufgrund eines Gen-Defekts, Ursache einer Krankheit sein, wobei sich dieses Wissen z.B. als Biomarker für Krebs nutzen lässt. Daher ist kaum verwunderlich, dass die systematische Studie von Glykanen in biologischen Systemen und die Untersuchung der Glykosylierung von Proteinen mittlerweile einen festen Platz in der medizinischen und pharmakologischen Forschung einnimmt [2][3][4][5].

Für die Analyse von Glykanen einer biologischen Probe haben sich eine Reihe von Messverfahren etabliert, wobei in dieser Arbeit der primäre Fokus auf der Kapillar-Gel-Elektrophorese (engl.: *capillary gel electrophoresis*, CGE) liegt. Die Grundidee ist dabei, die Bestandteile des zu analysierenden Materials nach ihren physischen oder chemischen Eigenschaften zu trennen und die so separierten Analyten mit Hilfe eines Detektors zu messen. In der CGE sind diese Detektor-Reaktionen als Zeitreihe von Messspitzen (engl.: *peaks*) dargestellt, wobei der Grad der zeitlichen Separation zwischen zwei Peaks als Auflösung bezeichnet wird [6][7].

Allerdings lassen sich bestimmte Bestandteile, aufgrund von ähnlichen Trennungseigenschaften, nur unvollständig separieren. Die Folge sind schlecht auf-

gelöste Peak-Strukturen, die aus mehr als einem Analyten bestehen und fortan als Multi-Analyte-Peaks bezeichnet werden (Abbildung 1.1) .

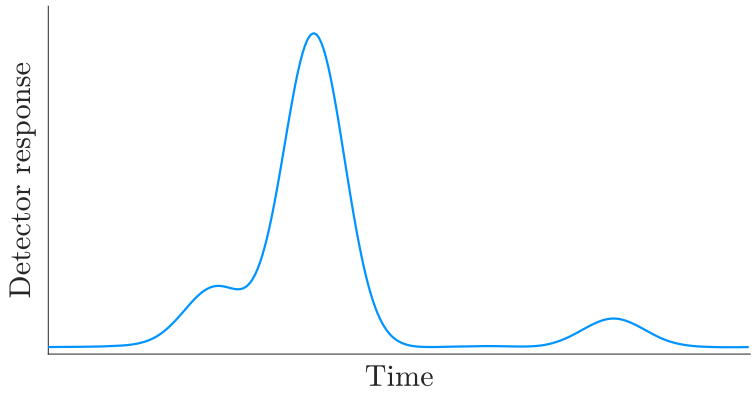


Abbildung 1.1: Darstellung eines beispielhaften Multi-Analyte-Peaks (links) und einer gut aufgelösten Peak-Struktur (rechts) im Vergleich.

In der analytischen Chemie existieren zwar Methodiken, um solche Strukturen zu trennen, jedoch sind diese mit teils beträchtlichem Mehraufwand verbunden, da Messungen wiederholt werden müssen. Infolgedessen ist eine softwareseitige Lösung des Problems, wie es diese für andere Messverfahren gibt, auch für die CGE äußerst erstrebenswert.

Der Fokus dieser Arbeit liegt daher auf dem Aspekt der Grundlagenforschung. Dabei werden bestehende Konzepte aus problemverwandten Disziplinen auf die CGE übertragen, um durch den systematischen Vergleich von Methoden einen geeigneten Ansatz zur Auflösung von Multi-Analyte-Peaks zu finden.

2 Theoretische Grundlagen

In den folgenden Abschnitten wird einleitend der biochemische Aufbau von Glykanen präsentiert, sowie eine Methode zu deren Analyse vorgestellt und erläutert, wie genau es dabei zu der Ausbildung von Multi-Analyte-Peaks kommt. Anschließend werden die technischen Grundlagen dieser Arbeit beschrieben, auf denen die entwickelten Konzepte aus Kapitel 3 basieren.

2.1 Glykane und Glykananalytik

2.1.1 Aufbau und Eigenschaften von Glykanen

Wie in der Einleitung erwähnt, stellt die Glykosylierung einen Prozess dar, bei dem Zucker-Strukturen unter Einsatz von Enzymen an ein Protein gebunden werden und so ein Glykoprotein formen. Gemeinhin wird eine solche Zucker-Verbindung als Glykokonjugat bezeichnet, wobei der Zucker des Konjugates Glykan genannt wird. Ein solches Glykan setzt sich aus Grundbausteinen, den Monosacchariden, zusammen, die über sogenannte glykosidische Bindungen miteinander verkettet sind und dadurch lineare oder verzweigende Strukturen ausbilden. Monosaccharide treten dabei häufig in Form von Hexosen auf, welche aufgrund der Anordnungsmöglichkeiten ihrer Kohlenstoff-Atome in unterschiedlichen Konfigurationen existieren können (Abbildung 2.1). Aus dieser Vielzahl von Monosaccharid-Formen, in Kombination mit den mannigfaltigen Möglichkeiten diese miteinander zu verbinden, ergibt sich die immense Vielfalt an Glykan-Strukturen [2][4][8][9].

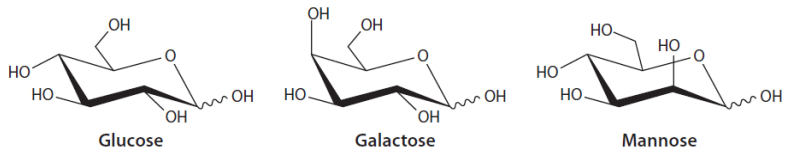


Abbildung 2.1: Illustration der chemischen Struktur der Hexosen: Glukose, Galaktose und Mannose, welche alle die selbe Summenformel $C_6H_{12}O_6$ aufweisen (aus [2]).

Da die chemische Repräsentation eines Glykans, wie in Abbildung 2.2 zu sehen, oft sehr komplex ausfällt und ungewollt detailliert ist, wurden eine Reihe von text- und grafikbasierten Notationen zur Glykan-Repräsentation entwickelt.

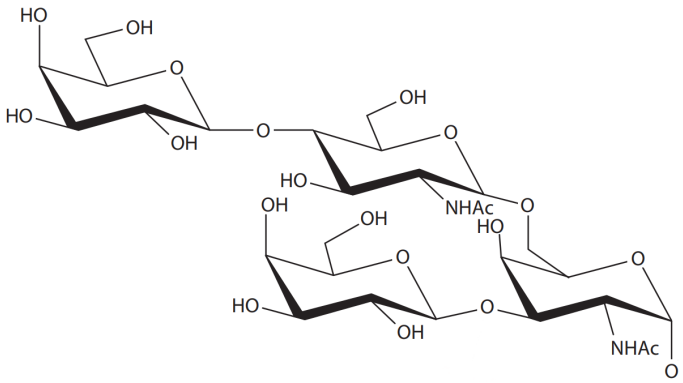


Abbildung 2.2: Chemische Darstellungsform eines Glykans (aus [2]).

Aufgrund ihrer kompakten Darstellungsform wurde für diese Arbeit auf die grafische Notation der *Consortium for Functional Glycomics* (CFG) zurückgegriffen [4][10]. In dieser werden die Monosaccharid-Bausteine, wie Tabelle 2.1 zu entnehmen ist, als simple geometrische Symbole visualisiert.

Tabelle 2.1: Auflistung der in Säugetier-Glykanen häufig auftretenden Monosaccharide in CFG-Notation mit Name und Kürzel.

Monosaccharid	Kürzel	Symbol
Glukose	Glc	●
Galaktose	Gal	●
Mannose	Man	●
N-Acetylglukosamin	GlcNAc	■
N-Acetylgalaktosamin	GalNAc	■
Fukose	Fuc	▲
N-Acetylneuraminsäure	NeuAc	◆
N-Glycolylneuraminsäure	NeuGc	◇

Diese Bausteine sind dabei durch Linien miteinander verbunden, welche bei Bedarf mit zusätzlichen Informationen, z.B. über die Art der Bindung (engl.: *linkage information*, Abbildung 2.3), versehen werden können [10].

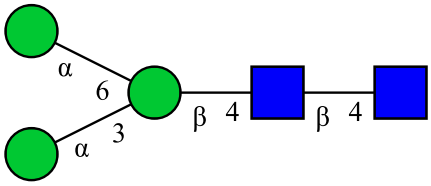


Abbildung 2.3: Darstellung der Kernstruktur der N-Glykane mit zusätzlicher linkage information an den Verbindungslinien.

Bei den proteingebundenen Glykanen wird grundsätzlich zwischen den beiden Superfamilien der N- und der O-glykosidisch gebundenen Glykane unterschieden, deren Vertreter üblicherweise verkürzt als N-Glykane und O-Glykane bezeichnet werden. Hierbei ergeben sich die jeweiligen Benennungen aus der Art der Bindung des Glykans über ein Stickstoff- (N) bzw. ein Sauerstoff-Atom (O) mit dem Protein [4].

Für diese Arbeit soll das Hauptaugenmerk ausschließlich auf der Klasse der N-Glykan-Strukturen liegen, welche alle über eine gemeinsame Kernstruktur verfügen, mit der sie am Protein gebunden sind. Wie in Abbildung 2.3 illustriert, setzt sich diese aus einem GlcNAc-Dimer zusammen, an welchem drei verzweigte Mannosen hängen. In einem komplexen Entstehungsprozess welcher Ab- und Aufbauschritte beinhaltet, wird ein jedes N-Glykan um diese Grundstruktur herum konstruiert, wobei es aufgrund seiner finalen Form zu einer von drei Subtypen zugeordnet werden kann. Dabei wird zwischen mannose-reichen Glykanen, Komplex- oder Hybrid-Typen unterschieden, für welche jeweils ein Beispiel in Abbildung 2.4 gegeben ist [2].

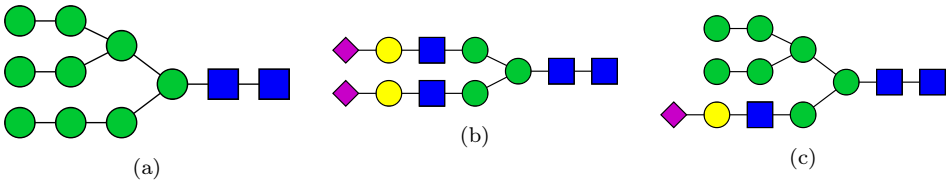


Abbildung 2.4: Beispielhafte Vertreter der drei N-Glykan-Subtypen: Mannose-reiche (a), Komplex- (b) und Hybrid-Typ (c).

2.1.2 Analytik von Glykanen

Neben der Massenspektrometrie (engl.: *mass spectrometry*, MS) [2][3][11][12] und Flüssigchromatographie (engl.: *liquid chromatography*, LC) [8][13][14][15], zählen elektromigrative Techniken zu den gängigsten Methoden zur Analyse von Glykanen und Glykoproteinen. Hierbei haben sich die Kapillar-Elektrophorese (engl.: *capillary electrophoresis*, CE) und Kapillar-Gel-Elektrophorese (engl.: *capillary gel electrophoresis*, CGE) [16][17][18][19][20] besonders hervorgetan, welche nachfolgend vorgestellt werden.

Bevor jedoch die N-Glykan-Strukturen eines Glykoproteins analysiert werden können, müssen diese zunächst von ihrem tragenden Protein abgelöst (engl.: *released*) werden. Hierzu wird, im Rahmen der Probenvorbereitung, auf enzymatische oder chemische Mittel zurückgegriffen, wobei bevorzugt das Enyzm Peptid-N-Glykosidase F zum Einsatz kommt. Dieses zählt zu den sogenannten Endoglykosidasen und ist in der Lage die spezifisch N-glykosidische Bindungen zwischen Glykan und Glykoprotein zu durchtrennen [2][8].

Der grundlegende Aufbau für CE bzw. CGE ist in Abbildung 2.5 illustriert. Ein mit zu analysierendem Material und Elektrolyten gefülltes Probengefäße (engl.: *sample vial*) ist dabei über eine Elektrolyt-beladene Kapillare mit einer Phiole am Kapillarauslass verbunden.

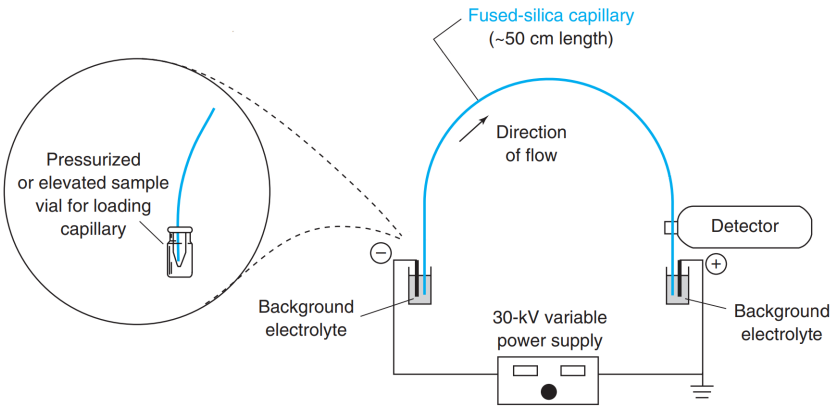


Abbildung 2.5: Aufbau einer Kapillar-Elektrophorese-Apparatur (angepasst aus [6]).

Werden die beiden Enden unter Spannung gesetzt, entsteht ein elektrisches Feld, unter dessen Einfluss die geladenen Proben-Moleküle durch die Kapillare zu migrieren beginnen und sich entsprechend ihres Masse-zu-Ladung-Verhältnisses (m/z) auftrennen (Abbildung 2.6). Die so separierten Analyten

passieren den Detektor dadurch idealerweise zu unterschiedlichen Zeitpunkten, wo diese als Messspitzen (engl.: *peaks*) von einem Computer aufgezeichnet werden. In der CGE wird das Auftrennungsvermögen zusätzlich verbessert, indem die Kapillare mit einem Gel gefüllt wird. Dies bewirkt einen Siebeffekt (engl.: *sieving effect*) der dafür sorgt, dass die Bewegungsgeschwindigkeiten der Moleküle maßgeblich von deren Form und Größe abhängen [13][6].

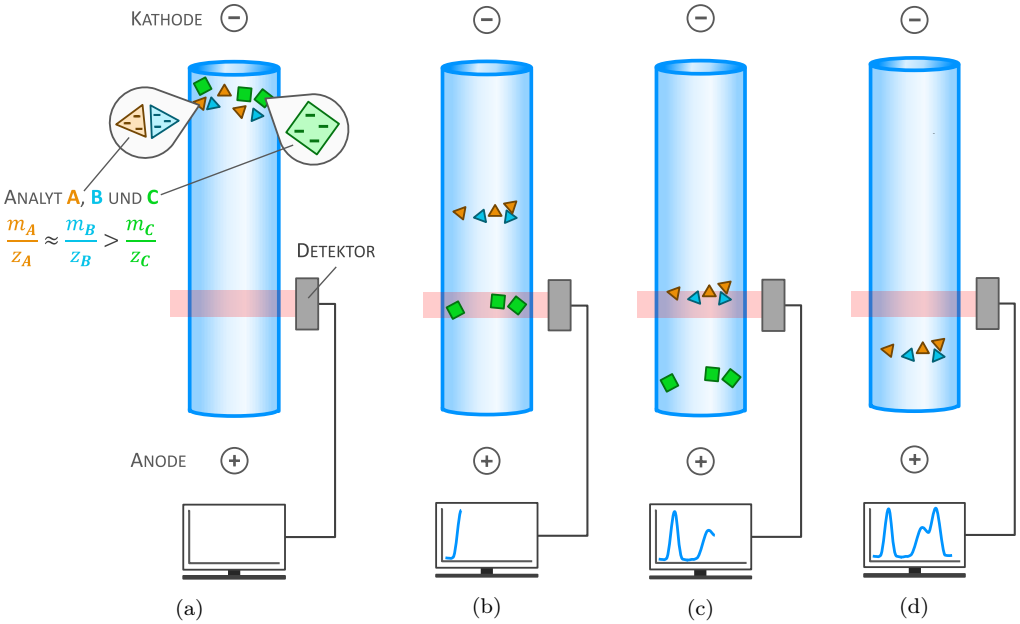


Abbildung 2.6: Funktionsprinzip der Kapillar-Elektrophorese, bei dem geladene Analyten in einer Kapillare (a) mittels eines elektrischen Felds nach ihrer elektrophoretischen Mobilität aufgetrennt werden (b und c). Während der hochgeladene Analyte C als erstes am Detektor vorübergeht, lassen sich A und B aufgrund ihres ähnlichen Masse-zu-Ladung-Verhältnisses kaum separieren, wodurch es zur Entstehung eines Multi-Analyte-Peaks kommt (d).

Damit Glykan-Strukturen in der CE bzw. CGE von einem Detektor erfasst werden können, müssen diese vorab mit einem Fluoreszenzfarbstoff markiert werden. Zu diesem Zweck hat sich der Fluoreszenztag 8-Aminopyren-1,3,6-trisulfonsäure (APTS) bewährt, welcher unter Laserlicht-Einwirkung ein detektierbares Fluoreszenzlicht (engl.: *laser-induced fluorescence*, LIF) emittiert. Darüber hinaus handelt es sich bei APTS um einen dreifach negativ geladenen Farbstoff, wodurch es erst möglich wird, Glykane, welche elektrisch neutral sind, elektrophoretisch zu trennen. Das Ergebnis der Separation, ein 1D-Signal, wird Elektropherogramm genannt und ist visuell den Chromatogrammen der chromatischen Verfahren nicht unähnlich (Abbildung 2.7) [8][14][20][21].

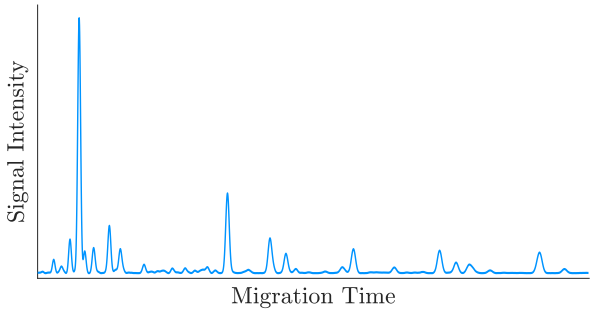


Abbildung 2.7: Repräsentation eines Elektropherogramms mit N-Glykan-Peaks.

Wie in Abbildung 2.6d zu sehen ist, können Strukturen nicht immer separiert werden, was zur Ausbildung von Multi-Analyte-Peaks führt. Zur Lösung dieser Problematik existieren in der Glykananalytik verschiedene Herangehensweisen wie beispielsweise die Kopplung von Messmethoden [3][22]. Ein weiterer Ansatz ist die Verwendung von speziellen Enzymen, den Exoglycosidases, welche die terminalen Monosaccharide eines Glykans in einem sogenannten Verdau (engl.: *digest*) nach und nach entfernen. Auf diese Weise ändern sich die Migrationszeiten der verdauten N-Glykane, welche auf Strukturen herunter gebrochen werden, die sich schneller durch die Kapillare bewegen können (Abbildung 2.8).

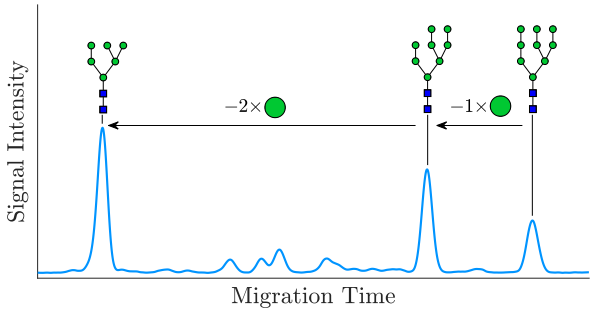


Abbildung 2.8: Funktionsprinzip der Mannosidase, welche die terminalen Mannose-Bausteine entfernt und so eine Veränderung der Migrationszeit bewirkt.

Anschließend lässt sich, auf Basis der so gewonnenen Struktur-Informationen, die ursprüngliche Zusammensetzung der Multi-Analyte-Peaks rekonstruieren. Die Mannosidasen, die Galactosidasen, und die Neuraminidasesen bzw. Sialidasen stellen dabei die bekanntesten Vertreter von Exoglycosidasen dar, welche jeweils zur Entfernung von terminalen Man-, Gal-, bzw. NeuAc-Monosacchariden herangezogen werden [2][3][8][13].

Allerdings sind all diese Ansätze mit einer Wiederholung von Messungen und damit mit Mehraufwand verbunden. Eine softwareseitige Lösung, wie es diese für andere Verfahren gibt [23][24][25][26][27][28], ist daher auch für die CE bzw. CGE äußerst erstrebenswert.

2.2 Informatische Grundlagen

Während der Einsatz von mathematischen Modellen und Funktionen zur Modellierung von vollständig separierten Peaks etabliert ist, existiert nur eine unzureichende Anzahl von Ansätzen zur Auflösung von Multi-Analyte-Peaks. Dieser Umstand, der besonders für die CGE gilt, bildet die Grundmotivation hinter dieser Arbeit. In vielen dieser Verfahren wird das Messsignal als Summe von sich partiell überlagernden Peaks nachgebildet, indem die Parameter eines Modells optimiert werden, das sich aus einer Schar von mathematischen Funktionen zusammensetzt. Auf diese Weise liefert im Idealfall jede Funktion die Beschreibung für exakt eine Struktur eines Multi-Analyte-Peaks [29][30][7].

Im Folgenden wird eine Reihe von bekannten Peak-Modellen vorgestellt, sowie ein Phänomen von Chromato- und Elektropherogrammen präsentiert, um diese zu verbessern. Anschließend wird eine kurze Einführung in das Themengebiet der Optimierung gegeben, bevor das Kapitel mit der ausführlichen Beschreibung zweier Optimierungsmethoden geschlossen wird.

2.2.1 Mathematische Modelle

Trotz einer Vielzahl an Modellen werden hochaufgelöst HPLC-, bzw. CE-Peaks häufig als einfache Normal- bzw. Gaußverteilung modelliert [6][30][7]. Diese schreibt sich für eine einzelne Variable $\mathbf{x}$ als $\mathcal{N}(\mathbf{x}|\mu, \sigma^2)$ mit

$$\mathcal{N}(\mathbf{x}|\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{x}-\mu)^2\right\}, \tag{2.1}$$

wobei μ den Erwartungswert und σ^2 die Varianz ist [31].

Allerdings beschränkt sich das Darstellungsvermögen von Gaußverteilungen, im Vergleich zu komplexeren Modellen wie beispielsweise dem Exponentially Modified Gaussian (EMG) oder dem bi-Gaussian, auf rein symmetrische Peak-Formen [29][32][33][34].

Darüber hinaus stoßen viele dieser doch eher grundlegenden Wahrscheinlichkeitsfunktion an ihre Grenzen, wenn es um die exakte Modellierung von Realdaten geht. Dabei ist ein Lösungsansatz die Verwendung von sogenannten Mischverteilungen (engl.: *mixture model*): Linearkombination von Verteilungsfunktionen, welche zusammen ein reicheres Wahrscheinlichkeitsmodell bilden.

Gauß-Funktionen, welche auf diese Weise kombiniert wurden, werden als gaußsches Mischmodell (engl.: *gaussian mixture model*, GMM) bezeichnet und können fast jede kontinuierliche Dichtefunktion beliebig genau approximieren, sofern die Funktionsanzahl und die Parameter entsprechend gewählt worden sind.

Das GMM einer Variable $\mathbf{x}$ ist als Superposition $p(\mathbf{x})$ von K Gaußverteilungen $\mathcal{N}(x|\mu_k, \sigma_k^2)$ definiert, welche mit den Mischkoeffizienten π_k gewichtet sind.

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \sigma_k^2) \tag{2.2}$$

Hierbei werden die $\pi_k \mathcal{N}(x|\mu_k, \sigma_k^2)$ für $k = 1, \ldots, K$ die Komponenten des GMMs genannt, welche jeweils über einen eigenen Satz an Parametern $\theta_k = (\pi_k, \mu_k, \sigma_k^2)$ verfügen (Abbildung 2.9).

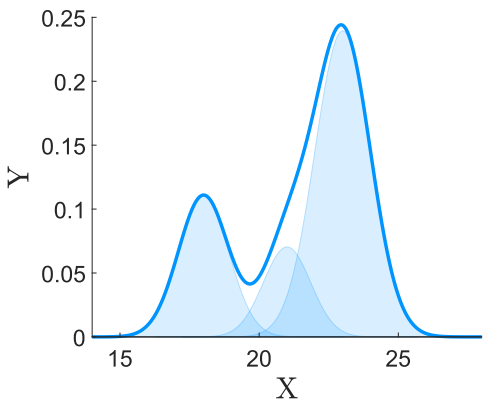


Abbildung 2.9: Visualisierung eins GMM, das sich aus den Komponenten $\theta_1 = (\pi_1 = 0.25, \mu_1 = 18, \sigma_1^2 = 0.9)$, $\theta_2 = (\pi_2 = 0.15, \mu_2 = 21, \sigma_2^2 = 0.8)$ und $\theta_3 = (\pi_3 = 0.6, \mu_3 = 23, \sigma_3^2 = 1)$ zusammensetzt.

Unter der Voraussetzung, dass sich die Fläche von $p(x)$ zu 1 summiert, liefert die beidseitige Integration von (2.2) die Eigenschaft

$$\sum_{k=1}^{K} \pi_k = 1. \tag{2.3}$$

Darüber hinaus impliziert $p(x) \geqslant 0$, dass $\mathcal{N}(x|\mu_k, \sigma_k^2) \geqslant 0$ ist und somit $\pi_k \geqslant 0$ für alle Komponenten $k = 1, \ldots, K$ gilt. Zusammen mit (2.3) folgt daraus

$$0 \leqslant \pi_k \leqslant 1. \tag{2.4}$$

Neben diesen Möglichkeiten der Modellierung existieren aber auch Konzepte, um Peak-Eigenschaften direkt als mathematische Ausdrücke zu formalisieren und so als Anforderungen an die Modell-Parameter zu nutzen. Insbesondere soll in dieser Arbeit das Augenmerk auf das Phänomen der Verbreiterung von Peaks (engl.: *peak broadening*) über die Messzeit gelegt werden. Diese Eigenheit, welche unter anderem auf thermische Diffusion zurückführen ist, wurde in verschiedenen Arbeiten als ein linearer Trend

$$\sigma_i = a_1 + a_2 t_i \tag{2.5}$$

beschrieben. Hierbei modelliert σ_i die Breite des darzustellenden Peaks i, während a_1 und a_2 vorab gewählte, feste Koeffizienten sind und t_i die Retentions- bzw. Migrationszeit ist, an welcher i maximal wird [7][34][35][36][37][38].

2.2.2 Optimierung und Nebenbedingungen

Ein Optimierungs- bzw. Minimierungsproblem lässt sich formal als der Versuch beschreiben, für eine Fitness- oder Zielfunktion $f(\mathbf{x})$ (engl.: *objective function*) mit

$$f : \mathbb{R}^n \to \mathbb{R} \tag{2.6}$$

ein $\hat{\mathbf{x}} \in \mathbb{R}^n$ zu finden, so dass

$$\forall \hat{\mathbf{x}} \in \mathbb{R}^n : f(\hat{\mathbf{x}}) \leqslant f(\mathbf{x}), \tag{2.7}$$

wobei $\hat{\mathbf{x}}$ dann das globale Minimum genannt wird. Jedes $\mathbf{x} \in \mathcal{S} \subseteq \mathbb{R}^n$ ist dabei ein n-elementiger Vektor von Entscheidungsvariablen, welcher einen möglichen Lösungskandidaten (engl.: *candidate solution*) für die Funktion $f(\mathbf{x})$ darstellt. Hierbei repräsentiert $\mathcal{S}$ den Suchraum (engl.: *search space*) der Lösungen, während das Bild von $\mathcal{S}$ als Zielraum $\mathcal{Z} \subseteq \mathbb{R}$ (engl.: *objective space*) bezeichnet wird. Da das Finden des globalen Minimums $\hat{\mathbf{x}}$ nicht immer garantiert werden kann, sind lokale Lösungskandidaten mit guter Fitness, wie in Abbildung 2.10 dargestellt, in der Praxis oft ausreichend [39][40].

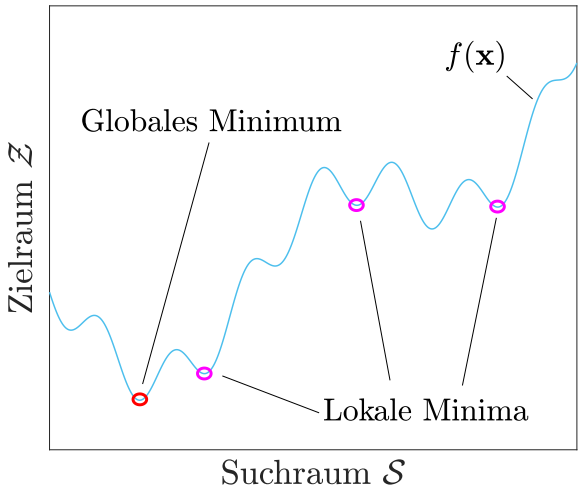


Abbildung 2.10: Ausschnitt eines Minimierungsproblems einer 1-D Funktion f, wobei die Fitnesswerte $f(\mathbf{x}) \in \mathcal{Z}$ gegen die Lösungen $\mathbf{x} \in \mathcal{S}$ aufgetragen und das globale Minimum, sowie einige der lokalen Minima, farblich hervorgehoben sind.

Ein Beispiel für $f(\mathbf{x})$ stellt die aus der Regression bekannte Fehlerfunktion $E^S(\mathbf{x})$ der Summe der Fehlerquadrate (engl.: *sum-of-squares error*, SSE) dar, welche die Ungenauigkeit zwischen einem Modell $m(\mathbf{x})$ und einer Menge von Beobachtungenswerten t_i mit $i = 1, \ldots, K$ misst.

$$E^S(\mathbf{x}) = \sum_{i=1}^{K} \{m(\mathbf{x}) - t_i\}^2 \tag{2.8}$$

Dabei ist der Fehlerwert von $E^S(\mathbf{x})$ ein nicht-negatives Gütemaß, welcher dann und nur dann zu Null wird, wenn $m(\mathbf{x})$ alle Elemente von $\mathbf{t}$ enthält (Abbildung 2.11).

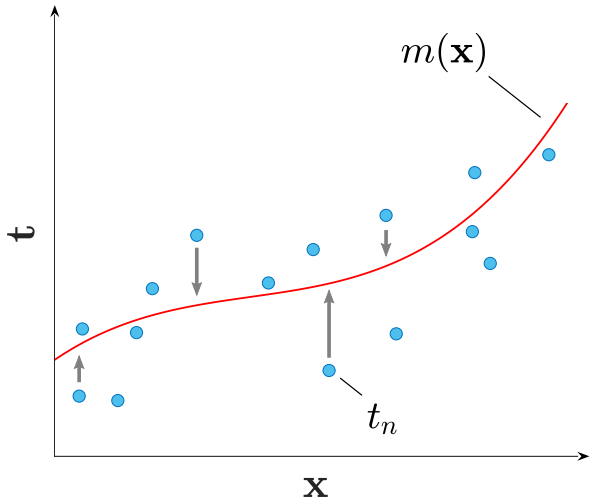


Abbildung 2.11: Veranschaulichung der SSE-Fehlerfunktion, wobei sich der Fehlerwert von $E^S(\mathbf{x})$ aus der Summe der quadrierten Abstände (grauer Pfeil) zwischen allen Datenpunkten t_n und dem Modell $m(\mathbf{x})$ errechnet.

Obwohl viele der realweltlichen Anwendungen einen sehr großen Suchraum aufspannen, sind diese oft Nebenbedingungen (engl.: *constraints*) unterworfen, welche den Raum in Teilbereiche aus *zulässigen* und *unzulässigen* Lösungen (engl.: *feasible and infeasible solutions*) separieren. Klassischerweise definiert sich ein solches Minimierungsproblem mit Nebenbedingungen als

$$\min f(\mathbf{x}) \tag{2.9}$$

s.t.

$$g_d(\mathbf{x}) \leq 0, d = 1, 2, \ldots, D \tag{2.10}$$
$$h_e(\mathbf{x}) = 0, e = 1, 2, \ldots, E, \tag{2.11}$$

wobei zwischen den Ungleichungsnebenbedingungen $g_d(\mathbf{x})$ und den Gleichungsnebenbedingungen $h_e(\mathbf{x})$ (engl.: *inequality and equality constraints*, UNB und GNB) unterschieden wird. Des Weiteren wird der Teilraum aller *zulässigen* Lösungen, d.h. die Menge der Lösungskandidaten $\mathbf{x}$, welche alle Nebenbedingungen erfüllen, mit $\mathcal{F} \subseteq \mathcal{S}$ bezeichnet (Abbildung 2.12) [41].

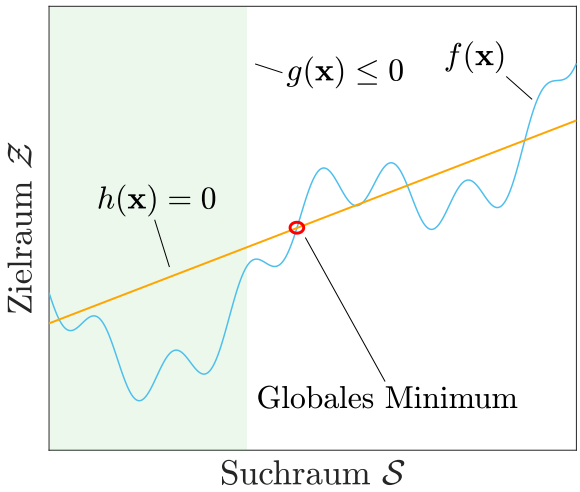


Abbildung 2.12: Darstellung der in Abbildung 2.10 illustrierten Zielfunktion f unter den Einwirkungen der Nebenbedingungen $g(\mathbf{x}) \leq 0$ und $h(\mathbf{x}) = 0$. Dabei ist $\mathcal{F}$ die Menge aller Schnittpunkte zwischen f und h, welche nicht in g liegen.

Ein effizienter Ansatz, um $f(\mathbf{x})$ zu optimieren, wäre es daher ausschließlich in $\mathcal{F}$ nach Lösungen mit guter Fitness zu suchen. Jedoch ist dieser Teilraum unter UNBs nur schwer definierbar. Außerdem kann es dazu kommen, dass ein zuerst

zulässiger Lösungskandidat nach der Optimierung nicht mehr in $\mathcal{F}$ liegt. Diese und ähnliche Schwierigkeiten haben zur Entwicklung von Strategien geführt, welche in der Lage sind auch für beschränkte Probleme optimale Lösungen zu finden und von denen nun die Straffunktion (engl.: *penalty function*) im Detail erläutert werden soll [39][41].

Sei $f(\mathbf{x})$ eine Zielfunktion, die es zu minimieren gilt, und $g_d(\mathbf{x})$ bzw. $h_e(\mathbf{x})$ die UNBs und GNBs. Dann priorisiert eine Straffunktion *zulässige* gegenüber *unzulässigen* Lösungen, indem sie Verstöße gegen die Nebenbedingungen mit der Addition eines Strafterms auf die Fitness eines Lösungskandidaten $\mathbf{x}$ ahndet. Formell lässt sich dies als erweiterte Zielfunktion $\phi(\mathbf{x})$ (engl.: *expanded objective function*) mit

$$\phi(\mathbf{x}) = f(\mathbf{x}) + s(\mathbf{x}) \tag{2.12}$$

schreiben. Hierbei stellt die Straffunktion $s(\mathbf{x})$ eine Kombination der Nebenbedingungen dar, die mit den Straffaktoren r_d und c_e (engl.: *penalty factors*) gewichtet sind:

$$s(\mathbf{x}) = \sum_{d=1}^{D} r_d \max(0, g_d(\mathbf{x}))^2 + \sum_{e=1}^{E} c_e |h_e(\mathbf{x})| \tag{2.13}$$

Während die Implementierung von $\phi(\mathbf{x})$ nahezu trivial ist, muss das Strafmaß häufig problemspezifisch, durch die Wahl geeigneter Faktoren r_d und c_e, auf das jeweilige Anwendungsfeld angepasst werden [39][41].

2.2.3 Expectation-Maximization-Algorithmus

Wie in Abschnitt 2.2.1 beschrieben, ist das GMM ein mächtiges Wahrscheinlichkeitsmodell, welches u.a. im Bereich der Mustererkennung, der statistischen Analyse und dem maschinellen Lernen zum Einsatz kommt. Zur Optimierung eines GMMs wird klassischerweise die Likelihood, ein Maß für die Anpassungsgüte (engl.: *goodness of fit*) eines statistischen Modells über eine Menge von Beobachtungen, als Zielfunktion maximiert. Einen eleganten Ansatz dafür stellt der Expectation-Maximization-Algorithmus (EM-Algorithmus) dar, der aus dem namensgebenden Expectation-Schritt (E-Schritt) und Maximization-Schritt (M-Schritt) besteht [31][42].

Dijkstra *et al.* nutzten den EM, um Flächen von MS-Peaks quantifizieren zu können die mit log-normalverteilten Modellen approximiert wurden [43]. Polanski *et al.* wiederum modellierten die MS-Massenspektren als eine Menge von GMMs, wofür auch auf den EM-Algorithmus zurückgegriffen wurde [44]. Yu und Peng postulierten ein EM-ähnliches Iterationsschema zur Optimierung von bi-Gaussian-Mischmodelle in der LC-MS, während Araes *et al.* einen EM-Ansatz für EMG-basierte Mischmodell entwickelten [45][46].

Die grundlegende Funktionsweise des EM-Algorithmus kann als ein iterativer Prozess beschrieben werden, bei dem im E-Schritt für ein Parametersatz θ eine Verteilungsfunktion p geschätzt wird. Auf dieser Basis werden im nachfolgenden M-Schritt neue Parameter θ' berechnet, welche die Likelihood bezüglich p maximieren. Diese können dann wiederum von einem E-Schritt verwendet werden, um eine Verteilung p' zu berechnen und so weiter. Während sich der Vorgang prinzipiell beliebig oft wiederholen lässt, wird die Iteration traditionell nach dem Erreichen eines Konvergenzkriteriums beendet und das finale $\hat{\theta}$ oder $\hat{p}$ als Lösung zurückgegeben [31][42].

Im Folgenden wird nun die konkrete EM-Instanz für GMMs hergeleitet:

Sei $\mathbf{x} = \{x_1, \ldots, x_N\}$ ein Datensatz mit N i.i.d. Beobachtungen $x_n \in \mathbb{R}, n = 1, \ldots, N$, welche von einem GMM mit K Komponenten stammen, und $\mathbf{z}$ die unbekannte Information, welche Komponenten k die jeweilige Beobachtung x_n erzeugt hat. Dann wird $\mathbf{y} = \{\mathbf{x}, \mathbf{z}\}$ der komplette Datensatz genannt und das Ziel ist es, die Parameter $\theta = \{(\pi_k, \mu_k, \sigma_k^2)\}_{k=1}^K$ auf Basis von $\mathbf{x}$ zu optimieren, um so eine möglichst gute Approximation von $\mathbf{z}$ zu erhalten (Abbildung 2.13).

Dazu werden die Dichten $p(\mathbf{x} = x_n|\theta)$ über alle Beobachtungen x_n, in Form der Log-Likelihood-Funktion (LLH-Funktion) $L(\theta)$, maximiert. Dies ist äquivalent mit der Maximierung der Likelihood-Funktion, da jede Lösung $\hat{\theta}$ von (2.15), aufgrund der steigenden Monotonie des Logarithmus, auch eine Maximum-Likelihood-Lösung ist.

$$p(\mathbf{x} = x_n|\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \sigma_k^2) \tag{2.14}$$

$$L(\theta) = \ln p(\mathbf{x}|\theta) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \sigma_k^2) \right\} \tag{2.15}$$

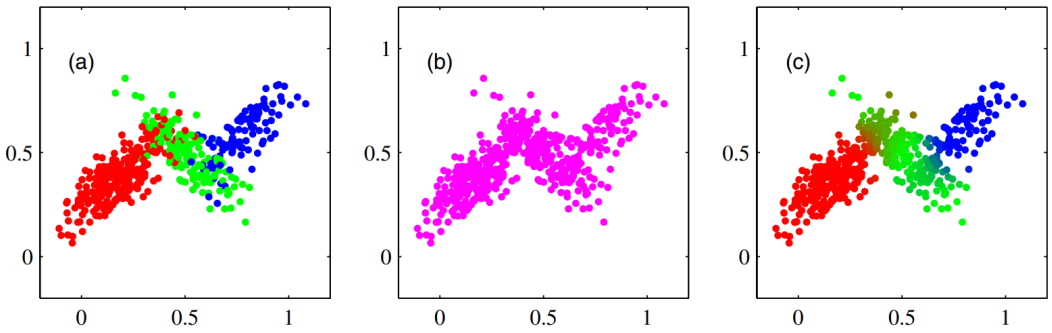


Abbildung 2.13: Illustration von 500 Beobachtungen aus [31], welche von einem GMM mit $K = 3$ gezogen worden sind. Dabei stellt (a) den gesuchten, kompletten Datensatz $\mathbf{y}$ dar, mit $\mathbf{z}$ als die farbliche Codierung der Beobachtungen $\mathbf{x}$. Diese sind wiederum in (b) zu sehen, während (c) die approximierten Daten $\mathbf{y}' = \{\mathbf{x}, \mathbf{z}'\}$ nach der Ausführung des EM-Algorithmus zeigt.

Die Werte des $\hat{\theta} = \{(\pi_k, \mu_k, \sigma_k^2)\}_{k=1}^K$ lassen sich dabei berechnen, indem $L(\theta)$ bezüglich π_k, μ_k und σ_k^2 abgleitet, Null gesetzt und umgestellt wird.

$$\pi_k = \frac{N_k}{N}. \tag{2.16}$$

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} x_n \tag{2.17}$$

$$\sigma_k^2 = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} (x_n - \mu_k)^2. \tag{2.18}$$

Während N_k die Anzahl an Beobachtungen darstellt, die von Komponente k erklärt werden, repräsentiert γ_{nk} die Wahrscheinlichkeit, dass die n-te Beobachtung von der k-ten Komponente erzeugt worden ist. D.h. γ_{nk} kann als die Verantwortung betrachtet werden, die Komponente k für die Beobachtung x_n trägt [31][47].

$$N_k = \sum_{n=1}^{N} \gamma_{nk} \tag{2.19}$$

$$\gamma_{nk} = p(\mathbf{z}_n = k | x_n, \theta) = \frac{\pi_k \mathcal{N}(x_n | \mu_k, \sigma_k^2)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i | \mu_j, \sigma_j^2)} \tag{2.20}$$

Des Weiteren ist zu beachten, dass (2.16), (2.17) und (2.18) eine geschlossene Form besitzen, obwohl N_k von γ_{nk} und damit von allen Parametern abhängt. Zur Lösung dieser Problematik bedient sich der EM-Algorithmus eines eleganten Tricks, wobei er γ_{nk} für die Berechnung der Ableitungen als konstant ansieht. Auf diese Weise bilden die Formeln ein iteratives Funktionsschema zum Finden einer Maximum-Log-Likelihood-Lösung:

1. Initialisiere die Parameter $\theta^{(t)} = \{(\pi_k, \mu_k, \sigma_k^2)\}_{k=1}^K$ in der Iteration $t = 0$.
2. E-Schritt: Schätze die Verantwortung $\gamma_{nk}^{(t)}$ mit Hilfe von (2.20) für die Werte von $\theta^{(t)}$ ab.
3. M-Schritt: Berechne das neue $\theta^{(t+1)}$ mittels (2.17), (2.18) und (2.16) bei konstantem $\gamma_{nk}^{(t)}$.
4. Aktualisiere die momentane Iteration mit $t = t + 1$.
5. Evaluiere (2.15) und wiederhole den gesamten Vorgang ab Schritt 2, bis ein Konvergenzkriterium erreicht wurde.

Diese Arbeitsweise ist in Abbildung 2.14 noch einmal verdeutlicht.

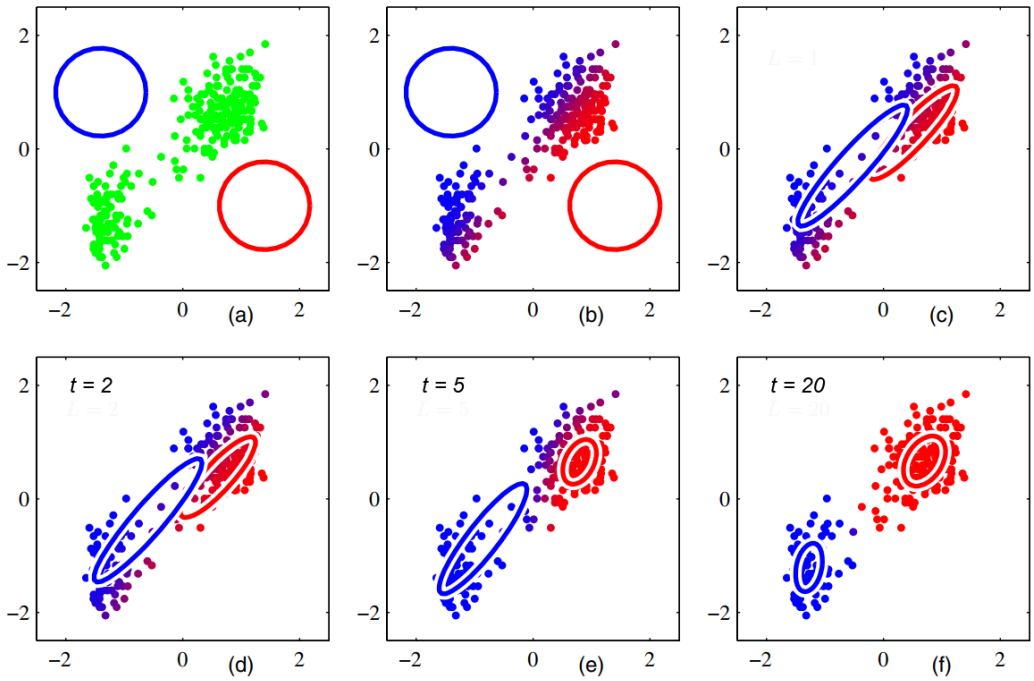


Abbildung 2.14: Visualisierung der iterativen Funktionsweise des EM zum Zeitpunkt der Initialisierung (a), nach dem ersten E-Schritt (b) und M-Schritt (c), sowie nach der Iteration $t = 2$ (d), $t = 5$ (e) und $t = 20$ (f) (angepasst aus [31]).

Abschließend ist zu betonen, dass durch die abwechselnde Ausführung des E- und M-Schritts eine Verbesserung der Log-Likelihood-Funktion garantiert ist. Allerdings hängt die finale Lösung ausschließlich von den Parametern $\theta^{(0)}$ ab, mit denen der EM-Algorithmus initialisiert worden ist. Es ist folglich nicht garantiert, dass eine gefundene Lösung tatsächlich das globale Optimum der LLH darstellt oder nur ein lokales Maximum repräsentiert [31].

2.2.4 Partikelschwarmoptimierung

Die Partikelschwarmoptimierung (engl.: *particle swarm optimization*, PSO) ist eine populäre Optimierungsmethode, welche der Superfamilie der evolutionären Algorithmen (engl.: *evolutionary algorithms*, EA) zugerechnet wird und von dieser die Eigenschaft einer Agenten-basierten, iterativen Suchstrategie übernommen hat. Ursprünglich zur Simulation eines Vogelschwarms entworfen, stellt der PSO im Kern einen Black-Box Optimierer dar, der auf dem Verhalten von sich selbstorganisierenden Agenten einer Population, dem Partikelschwarm, basiert [39][48][49][50].

Li postuliert einen dynamischen PSO-Ansatz, um Multi-Analyte-Peaks mittels EMG-basierten Konvolutionsmodellen in der Chromatographie aufzulösen [51]. Parastar *et al.* wiederum verfolgten ein Konzept, welches mit der Arbeit von Araes *et al.* vergleichbar ist, wobei ein PSO, statt eines EM-Algorithmus, zur Optimierung der Modelle verwendet wurde [52].

Der PSO unterscheidet sich insofern von traditionellen Optimierungsansätzen, als dass keine Ableitungen oder ähnliches notwendig sind, um eine Lösung zu finden. Stattdessen wird ein Schwarm von Partikeln zum Durchforsten des Suchraums herangezogen, welche sich, ausgehend von zufälligen Stratpositionen, über den Kurs der Optimierung clever durch den Raum bewegen [40][49][53].

Sei $f(\mathbf{x})$ eine zu minimierende Zielfunktion und jedes Element x_j mit $j = 1, \ldots, n$ einer Lösung $\mathbf{x} \in \mathbb{R}^n$ ist durch eine obere und eine untere Grenze $L_j \leqslant x_j \leqslant U_j$ beschränkt, welche den Suchraum $\mathcal{S}$ aufspannen. Des Weiteren wird eine Menge von M Partikeln in der Generation t der Schwarm $P^{(t)}$ genannt, wobei die Partikel $i = 1, \ldots, M$ jeweils über eine Position $\mathbf{x}_i^{(t)} \in \mathcal{S}$ und eine Geschwindigkeit $\mathbf{v}_i^{(t)} \in \mathbb{R}$ verfügen. Dann sind die Lösungskandidaten $\mathbf{x} \in \mathbb{R}^n$ für f genau die individuellen Positionen $\mathbf{x}_i^{(t)}$ der Partikel [39][41][53].

Damit sich eine Suchstrategie aus dem Verhalten der einzelnen Partikel herausbildet, sind diese in sogenannten Nachbarschaften (engl: *neighborhoods*) miteinander verbunden. Die Verbindung, welche verschiedene Formen annehmen kann, wird als Topologie des Schwarms bezeichnet (engl.: *swarm topology*) und dient dem Austausch von Wissen. Des Weiteren verfügt jedes Partikel i über einen Speicher, in welchem seine beste bisher gefundene Position, der individuelle Bestwert $\mathbf{p}_i^{(t)}$ (engl.: *personal best*), und die beste Position seiner aktuellen Nachbarschaft, der globale Bestwert $\mathbf{g}_i^{(t)}$ (engl.: *global best*) hinterlegt wird.

Auf dieser Grundlage werden nun in jeder Generation t die Positionen $\mathbf{x}_i^{(t)}$ und Geschwindigkeiten $\mathbf{v}_i^{(t)}$ angepasst. Hierbei animiert das Partikel mit dem besten $\mathbf{x}_i^{(t)}$ seine Nachbarn dazu, sich auf diese Position zuzubewegen. Die Idee dahinter ist, dass im Umfeld einer bereits gefundenen Lösung weitere, bessere Kandidaten existieren, welche durch die so angelockten Partikel aufgespürt werden. Wurde auf diese Weise tatsächlich eine besseres $\mathbf{x}_i^{(t)}$ entdeckt, werden die $\mathbf{p}_i^{(t)}$ und $\mathbf{g}_i^{(t)}$ der betroffenen Partikel entsprechend aktualisiert, bevor der gesamte Vorgang in Generation $t+1$ wiederholt wird [39][53][54].

Unter der Annahme, dass sämtliche Partikel miteinander verknüpft sind, lässt sich der Ablauf dabei wie folgt zusammenfassen:

1. Starte den Schwarm P in der Generation $t = 1$ und setzte alle Elemente $j = 1, \ldots, n$ der Positionen $\mathbf{x}_{i,j}^{(1)}$ und Geschwindigkeiten $\mathbf{v}_{i,j}^{(1)}$ auf

$$\begin{aligned}\mathbf{x}_{i,j}^{(1)} &= U(L_j, U_j)\\ \mathbf{v}_{i,j}^{(1)} &= \frac{1}{2}(U(L_j, U_j) - \mathbf{x}_{i,j}^{(1)}),\end{aligned}$$

   mit $U(L_j, U_j)$ als zufällig gezogene Zahl einer uniformen Verteilung im Wertebereich $[L_j, U_j]$. Außerdem initialisiere $\mathbf{p}_i^{(1)}$ und $\mathbf{g}_i^{(1)}$ mit den Werten

$$\begin{aligned}\mathbf{p}_i^{(1)} &= \mathbf{x}_i^{(1)}\\ \mathbf{g}_i^{(1)} &= \arg \min_{\forall \mathbf{p}' \in \{\mathbf{p}_1^{(1)}, \ldots, \mathbf{p}_M^{(1)}\}} f(\mathbf{p}').\end{aligned}$$

2. Bestimme die Fitness $f(\mathbf{x}_i^{(t)})$ für jeden Partikel und passe $\mathbf{p}_i^{(t)}$ mittels

$$\mathbf{p}_i^{(t)} = \begin{cases}\mathbf{p}_i^{(t-1)}, & f(\mathbf{p}_i^{(t-1)}) \leqslant f(\mathbf{x}_i^{(t)})\\ \mathbf{x}_i^{(t)}, & sonst\end{cases} \tag{2.21}$$

an.

3. Aktualisiere die globalen Bestwerte $\mathbf{g}_i^{(t)}$ mit Hilfe von

$$\mathbf{g}_i^{(t)} = \arg \min_{\forall \mathbf{p}' \in \{\mathbf{p}_1^{(t)}, \ldots, \mathbf{p}_M^{(t)}\}} f(\mathbf{p}'). \tag{2.22}$$

4. Berechne für alle Partikel i und alle Elemente j neue Geschwindigkeits- und Positionswerte $v_{i,j}^{(t+1)}$ und $x_{i,j}^{(t+1)}$ mit

$$v_{i,j}^{(t+1)} = \omega v_{i,j}^{(t)} + c_1 r_1 (p_{i,j}^{(t)} - v_{i,j}^{(t)}) + c_2 r_2 (g_j^{(t)} - v_{i,j}^{(t)}), \tag{2.23}$$

und

$$x_{i,j}^{(t+1)} = x_{i,j}^{(t)} + v_{i,j}^{(t+1)}. \tag{2.24}$$

Hierbei stellt das Trägheitsmoment (engl.: *inertia weight*) $\omega \in \mathbb{R}$ eine Stellschraube für globale oder lokale Exploration dar, welche den Einfluss von $\mathbf{v}_i^{(t-1)}$ auf $\mathbf{v}_i^{(t)}$ steuert. Ein nicht-null-Gewicht lässt die Partikel ihre bereits eingeschlagene Richtung in leicht veränderter Form fortsetzten und kann bei der richtiger Wahl zu einer beschleunigten Konvergenz des Algorithmus führen. Darüber hinaus wirken sich die vier Kontrollparameter $c_1 r_1$ und $c_2 r_2$ auf die Trajektorie aus und beeinflussen die Fähigkeit des PSO, ein Optimum zu finden, signifikant. Dabei werden $r_1, r_2 \sim U(0,1)$ in jeder Generation t neu berechnet, wohingegen c_1 und c_2 zwei feste Konstanten sind, welche im Bereich $0 \leqslant c_1, c_2 \leqslant 2$ initialisiert werden.

5. Erzwinge die Einhaltung der Bereichsgrenzen der Positionen $\mathbf{x}_{i,j}^{(t+1)}$ durch

$$\mathbf{x}_{i,j}^{(t+1)} = \begin{cases} L_j, & \mathbf{x}_{i,j}^{(t+1)} < L_j \\ U_j, & \mathbf{x}_{i,j}^{(t+1)} > U_j \\ \mathbf{x}_{i,j}^{(t+1)}, & sonst. \end{cases} \tag{2.25}$$

Klemme außerdem die Geschwindigkeiten $\mathbf{v}_{i,j}^{(t+1)}$ mit

$$\mathbf{v}_{i,j}^{(t+1)} = \begin{cases} 0, & \mathbf{x}_{i,j}^{(t+1)} < L_j \\ 0, & \mathbf{x}_{i,j}^{(t+1)} > U_j \\ \mathbf{v}_{i,j}^{(t+1)}, & sonst. \end{cases} \tag{2.26}$$

Auf diese Weise wird verhindert, dass die selben Partikel den Suchraum $\mathcal{S}$ in der nächsten Generation wieder verlassen.

6. Aktualisiere die momentane Generation mit $t = t + 1$.
7. Wiederhole den gesamten Vorgang ab Schritt 2, bis ein Konvergenzkriterium erreicht wurde.

Klassischerweise wird der PSO terminiert, wenn sich die beste Lösung des Schwarms $\mathbf{g}^{(t)}$ kaum noch verändert oder eine maximale Anzahl an Funktionsaufrufen bzw. Generationen erreicht wurde [39][53].

Über die Jahre hat der zuvor skizzierte Funktionsablauf eine Reihe von Verbesserungen erfahren, welche unter dem Name Standard-PSO 2011 (SPSO-2011) zusammengefasst sind und dessen zwei wichtigste Änderungen abschließend vorgestellt werden sollen.

So ist seit Jahren bekannt, dass die Aktualisierung der Geschwindigkeiten in Schritt 4 zu einer Abhängigkeit des PSO zum Koordinatensystem des jeweiligen Problems führt, wodurch sich die Partikel auf Wegen konzentrieren, welche parallel zu den Koordinatenachsen verlaufen. Die Lösung besteht darin, $v_{i,j}^{(t+1)}$ über die Formel

$$v_{i,j}^{(t+1)} = \omega v_{i,j}^{(t)} + (\mathbf{x}_i^* - \mathbf{x}_i^{(t)}) \tag{2.27}$$

zu berechnen, wobei $\mathbf{x}_i^*$ ein zufälliger Punkt in der Hypersphäre $\mathcal{H}_i$ ist, die durch den Radius $\|\mathbf{G}_i^{(t)} - \mathbf{x}_i^{(t)}\|$ und ein Gravitationszentrum $\mathbf{G}_i^{(t)}$ definiert wird.

$$\mathcal{H}_i(\mathbf{G}_i^{(t)}, \|\mathbf{G}_i^{(t)} - \mathbf{x}_i^{(t)}\|) \tag{2.28}$$

$$\mathbf{G}_i^{(t)} = \frac{1}{3}(\mathbf{x}_i^{(t)} + \mathbf{p}_i^{*(t)} + \mathbf{g}_i^{*(t)}) \tag{2.29}$$

Hierbei stellen $\mathbf{p}_i^{*(t)}$ und $\mathbf{g}_i^{*(t)}$ Positionen dar, die jenseits der Bestwerts $\mathbf{p}_i^{(t)}$ bzw. $\mathbf{g}^{(t)}$ liegen und für jeden Partikel i, in jeder Generation j, mit Hilfe von

$$\mathbf{p}_i^{*(t)} = \mathbf{x}_i^{(t)} + c_1 r_1 (\mathbf{p}_i^{(t)} - \mathbf{x}_i^{(t)}), \tag{2.30}$$

und

$$\mathbf{g}_i^{*(t)} = \mathbf{x}_i^{(t)} + c_2 r_2 (\mathbf{g}^{(t)} - \mathbf{x}_i^{(t)}), \tag{2.31}$$

neu bestimmt werden [54][55][56].

Darüber hinaus sollte die Neuberechnung der $\mathbf{v}_{i,j}^{(t+1)}$-Werte in Schritt 5 zu

$$\mathbf{v}_{i,j}^{(t+1)} = \begin{cases} -0.5 v_{i,j}^{(t+1)}, & \mathbf{x}_{i,j}^{(t+1)} < L_j \\ -0.5 v_{i,j}^{(t+1)}, & \mathbf{x}_{i,j}^{(t+1)} > U_j \\ \mathbf{v}_{i,j}^{(t+1)}, & sonst \end{cases} \tag{2.32}$$

abgewandelt wurde, um Teile der Bewegungsenergie zu erhalten [54][55].

3 Konzept

In den folgenden Abschnitten soll der konzeptuelle Teil dieser Arbeit erläutert werden, welcher auf Basis der theoretischen Grundlagen entwickelt wurde. Die Kernidee dahinter war, ein flexibles Baukasten-System (Abbildung 3.1) zu erschaffen, was leicht erweiterbar ist und dessen Elemente sich einfach und schnell austauschen lassen. Sämtliche Lösungsansätze der Arbeit wurden mit Hilfe dieses Systems konstruiert, wobei deren Ergebnisse in Kapitel 4 vollumfänglich vorgestellt und diskutiert werden.

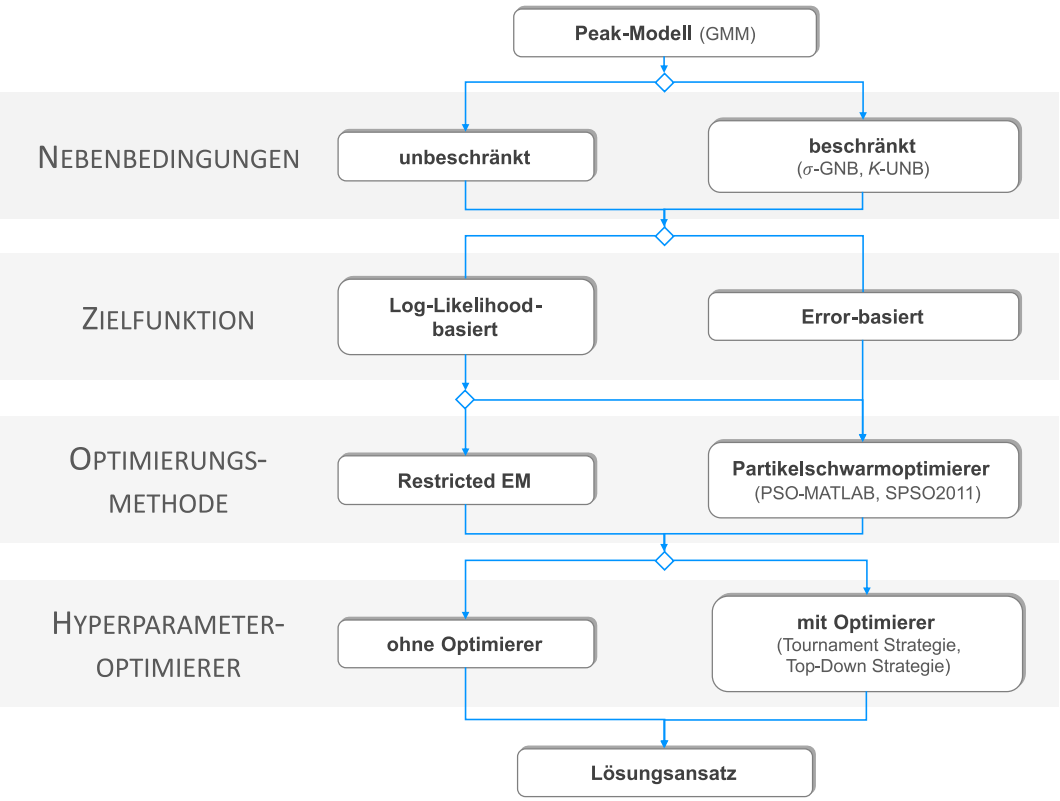


Abbildung 3.1: Konstruktionsplan der Lösungsansätze dieser Arbeit.

Dabei beschreiben die ersten Ebenen den zu modellierenden Sachverhalt als unbeschränktes oder beschränktes Optimierungsproblem. Darauf aufbauend steht eine Reihe von Methoden bereit, deren Zweck es ist, eine Lösung für die ausgewählte Zielfunktion zu finden. Diese können bei Bedarf durch sogenannte Hyperparameteroptimierer erweitert werden, welche die Problemgröße abschätzen, um so ein gutes, sinnvoll dimensioniertes Modell zu berechnen.

3.1 Problemmodellierung

3.1.1 Peak-Modell

Die CGE-LIF-basierten Multi-Analyte-Peaks wurden im Rahmen dieser Arbeit als GMMs der Form $p(\theta, \mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \sigma_k^2)$ modelliert, wobei $p(\theta, \mathbf{x})$ das Messsignal $\mathbf{y}$ an den Zeitpunkten $\mathbf{x}$ unter Zuhilfenahme der Parameter $\theta = \{(\pi_k, \mu_k, \sigma_k^2)\}_{k=1}^{K}$ abbildet. Es ist zu betonen, dass $\mathbf{y}$ vor der Modellierung auf eine Signalfläche $A = 1$ normalisiert wurde, da es sich bei $p(\theta, \mathbf{x})$ um eine Verteilungsfunktion handelt.

Unter der Annahme, dass die vorliegenden Glykan-Peaks einer symmetrischen Form folgen, wurde sich gegen die Verwendung von EMGs oder anderen asymmetrischen Modellen entschieden. Diese sind nicht nur komplexer als ein GMM, auch wurden Asymmetrien als Indiz für mehrere Strukturen und nicht als das Ergebnis schlechter Messbedingungen aufgefasst, wie in Abbildung 3.2 verdeutlicht wird.

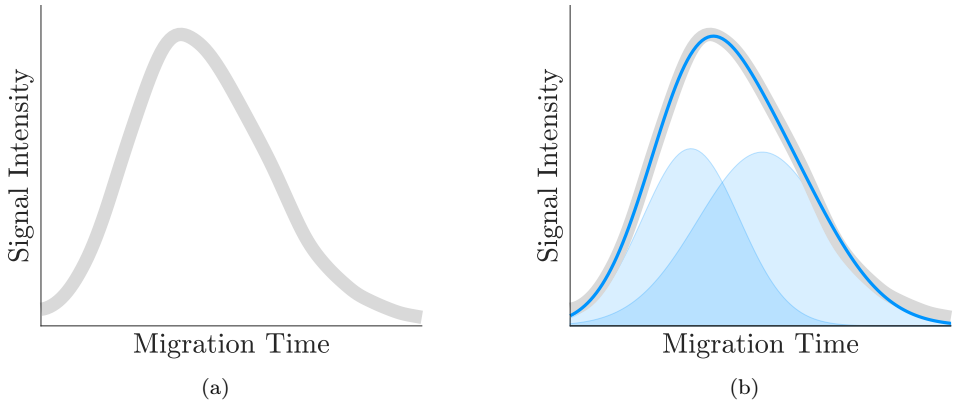


Abbildung 3.2: Visualisierung einer asymmetrischen Peak-Form vor (a) und nach (b) der Modellierung mit einem zwei-komponentigen GMM der Form $p(\theta, \mathbf{x})$ (blaue Linie).

3.1.2 Nebenbedingungen

Peak-Breite als Gleichungsnebenbedingungen

Eine Testreihe, welche in Vorbereitung der Arbeit durchgeführt wurde, zeigte, dass die unbeschränkte Optimierung von GMMs häufig zur Ausbildung von sehr schmalen Komponenten führt (Abbildung 3.3a). Des Weiteren konnte bei stark verrauschten Messsignalen eine unerwünschte, in Abbildung 3.3b illustrierte, Komponenten-Verbreiterung beobachtet werden.

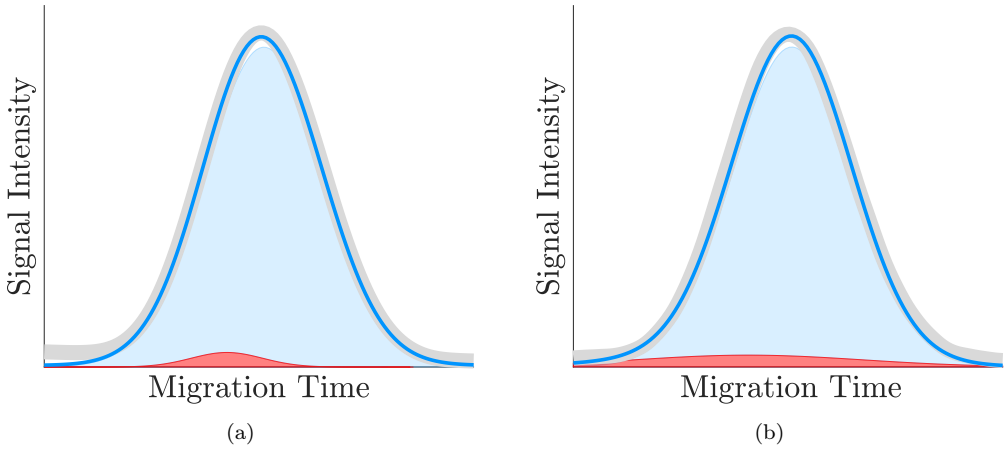


Abbildung 3.3: Darstellung von GMMs (blaue Linie), die ohne GNBs optimiert wurden und daher zu schmale (a, rote Fläche) oder zu breite Komponenten (b, rote Fläche) aufweisen.

Damit spiegelten diese Ergebnisse in keiner Weise die Realität wieder, in der zeitlich benachbarte Strukturen ähnlich breit sind. Eine Eigenschaft die sich, wie in Abschnitt 2.2.1 beschrieben, als eine lineare Funktion modellieren lässt. Im Zuge dessen wurde in Anlehnung an (2.5) eine σ-GNB $h_1(\mu_k, \sigma_k)$ mit

$$h_1(\mu_k, \sigma_k) = a_1 + a_2\mu_k - \sigma_k = 0 \tag{3.1}$$

aufgestellt, deren Koeffizienten wie folgt bestimmt wurden:

1. Modelliere und optimiere vollständig separierte Glykan-Peaks mittels ein-komponentiger GMMs und der $E^S(\mathbf{x})$-Zielfunktion.
2. Mit den Peak-Höhen als Gewichte, nutze die gewichtete Methode der kleinsten Quadrate (engl.: *weighted least squares*, WLS), um anhand der optimierten μ- und σ-Werte ein lineares Modell mit Koeffizienten a_1 und

a_2 zu approximieren. Dadurch wird das Regressionsmodell weniger stark von niedrigintensiven GMMs verzerrt, welche besonders vom Rauschen des Signals beeinflusst sind.

Da in der CGE-LIF ein solch linearer Zusammenhang aufgrund des Siebeffekts des Gels nur bedingt existiert, wurde die σ-GNB für diese Arbeit als Straffunktion $s_1(\theta)$ formuliert.

$$s_1(\theta) = \frac{1}{K} \sum_{k=1}^{K} |h_1(\mu_k, \sigma_k)| \tag{3.2}$$

Auf diese Weise werden, bei entsprechender Wahl des Straffaktors, sehr kleine oder große σ_k-Werte verhindert, ohne dass sich die Breite allzu statisch über die Migrationszeit definiert.

Komponentenanzahl als Ungleichungsnebenbedingung

Bevor allerdings ein GMM optimiert werden kann, muss zunächst die Anzahl an Komponenten K, d.h. die Dimension von $\theta = \{(\pi_k, \mu_k, \sigma_k^2)\}_{k=1}^{K}$, sinnvoll abgeschätzt werden. Jedoch ist die Wahl von K nicht trivial und insbesondere der Einsatz von hochdimensionalen Modellen ist im Fall dieser Arbeit aufgrund von redundanten (Abbildung 3.4a) oder vom Rauschen beeinflusster Komponenten äußerst problematisch (Abbildung 3.4b).

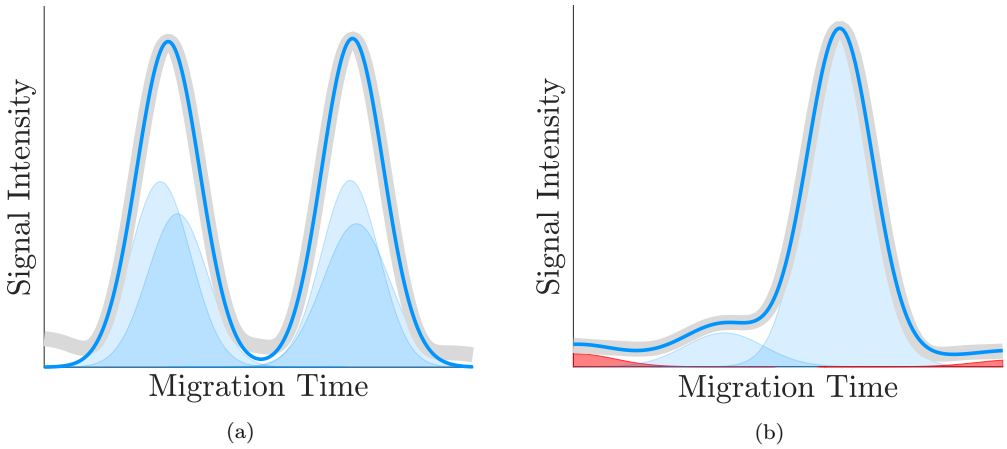


Abbildung 3.4: Illustration von GMMs (blaue Linien), die aus vier Komponenten bestehen und von denen zwei redundant sind (a) oder das Rauschen modellieren (b, rote Flächen).

Ein solcher Parameter, der vor dem eigentlichen Optimierungs- oder Trainingprozess festgelegt werden muss, wird Hyperparemeter genannt. Traditionell wird dieser optimiert, indem für eine definierte Menge von Werten eine Reihe von Modellkandidaten erzeugt wird, aus denen dann das geeignetste ausgewählt wird (engl.: *model selection*). Das Ziel dabei ist es die Anpassungsgüte gegen die Einfachheit des Modells zu balanciert, wofür gerne auf Informationskriterien (engl.: *information criteria*) zurückgegriffen wird. Diese fungieren als Strafterm der Komplexität, wodurch simple Lösungen mit guter Anpassungsgüte gegenüber komplexen Modellen bevorzugt werden [31].

Von diesem Konzept inspiriert, wurde die Straffunktion $s_2(\theta)$ entwickelt.

$$s_2(\theta) = |\Theta_{valid}| \tag{3.3}$$

Hierbei ist Θ_{valid} die Menge der *zulässigen* Komponenten

$$\Theta_{valid} = \{\theta_k \,|\, \theta_k \text{ erfüllt } g_1(\theta_k)\,\} \tag{3.4}$$

und $\Theta_{invalid}$ die Menge der *unzulässigen* Komponenten

$$\Theta_{invalid} = \{\theta_k \,|\, \theta_k \text{ erfüllt } g_1(\theta_k) \text{ nicht}\,\} \tag{3.5}$$

der K-UNB $g_1(\theta_k)$

$$g_1(\theta_k) = h_k - I_{min} \geq 0, \tag{3.6}$$

wobei $h_k = \pi_k \mathcal{N}(\mu_k|\mu_k, \sigma_k^2)$ die Höhe der Komponente $k = 1, \ldots, K$ darstellt und I_{min} ein entsprechend gewählter Minimalwert ist.

Die Idee zwischen *zulässigen* und *unzulässigen* Komponenten zu unterscheiden, entstammt der Bestimmungsgrenze (engl.: *limit of quantification*, LOQ), einem Konzept der analytischen Chemie. Dabei ist der LOQ die kleinste Menge eines zu untersuchenden Materials, die mit einer initial festgelegten Präzision quantifiziert werden kann. Analog zu dieser Definition müssen Komponenten eine Minimalintensität I_{min} aufweisen, um nicht direkt von der Optimierung ausgeschlossen zu werden. Auf diese Weise wird die Anzahl an niederintensiven Strukturen, welche besonders vom Rauschen beeinflusst sind, reduziert, während gleichzeitig redundante Komponenten mittels $s_2(\theta)$ verhindert werden.

Mit den in Abschnitt 3.3 vorgestellten Hyperparameteroptimierern, werden zwei Varianten präsentiert, die diese Mechanik implementieren. Zwar wäre es möglich gewesen, die Straffunktion $s_2(\theta)$ als weitere Zielfunktion in einer multikriteriellen Optimierung (engl.: *multi-objective optimization*) zu minimieren. Allerdings wäre dieser Ansatz rein auf die PSOs begrenzt und nicht mit der Funktionsweise des EM-Algorithmus kompatibel. Darüber hinaus wäre für die Elimination der *unzulässigen* Komponenten $\Theta_{invalid}$ ebenfalls eine Modifikation der bestehenden Verfahren vonnöten gewesen.

3.1.3 Zielfunktionen

Sämtliche erweiterte Zielfunktionen $\phi(\theta, \mathbf{x}, \mathbf{y})$ wurden als gewichtete Summen

$$\phi(\theta, \mathbf{x}, \mathbf{y}) = (1 - c_1 - c_2)f + c_1 s_1(\theta) + c_2 s_2(\theta) \tag{3.7}$$

der Straffaktoren c_1 und c_2 realisiert, wobei nachfolgend jeweils die LLH- bzw. Error-basierte Variante der Zielfunktion f vorgestellt wird.

Log-Likelihood

Die Log-Likelihood-Funktion, als der traditionelle Ansatz zur Optimierung von GMMs, war für f die ersten Wahl. Allerdings musste die Formel (2.15) angepasst werden, da die Daten nicht als Menge von Beobachtungen, sondern in Form eines Elektropherogramms, als diskrete Folge von Intensitätswerten $\mathbf{y}$ und Migrationszeiten $\mathbf{x}$ vorlagen. Dazu wurde jedes $y_n \in \mathbb{R}$ als die Häufigkeit der Beobachtung $x_n \in \mathbb{R}$ modelliert, womit sich die neue Log-Likelihood-Funktion $L(\theta, \mathbf{x}, \mathbf{y})$ als

$$L(\theta, \mathbf{x}, \mathbf{y}) = \sum_{n=1}^{N} y_n \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n | \mu_k, \sigma_k^2) \right\} \tag{3.8}$$

definiert. Darüber hinaus wurde (3.8) zum Minimierungsproblem $E^L(\theta, \mathbf{x}, \mathbf{y})$ umformuliert, um diese auch mit den PSO-Ansätzen optimieren zu können.

$$E^L(\theta, \mathbf{x}, \mathbf{y}) = -L(\theta, \mathbf{x}, \mathbf{y}) \tag{3.9}$$

Root-Mean-Square Error

Als Gegenentwurf zu einer LLH-basierten Funktion f wurde mit der Wurzel der mittleren Fehlerquadrate (engl.: *root-mean-squared error*, RMSE) auf eine herkömmliche Fehlerfunktion zurückgegriffen:

$$E^R(\theta, \mathbf{x}, \mathbf{y}) = \sqrt{\frac{1}{N} \sum_{n=1}^{N} \{p(\theta, x_n) - y_n\}^2} \tag{3.10}$$

Hierbei stellt $E^R(\theta, \mathbf{x}, \mathbf{y})$ eine Abwandlung der SSE-Funktion dar, die, dank der Division durch N, den Vergleich zwischen Datensätzen mit einer variablen Anzahl an Beobachtungen erlaubt. Außerdem befindet sich der Fehlerwert von $E^R(\theta, \mathbf{x}, \mathbf{y})$ aufgrund der Wurzel auf der selben Skala wie $\mathbf{y}$.

3.2 Optimierungsmethoden

3.2.1 Restricted EM

Mit dem EM-Algorithmus wurde für diese Arbeit der state-of-the-art Ansatz zur Maximierung der Log-Likelihood ausgewählt. Jedoch wurde nicht der in Abschnitt 2.2.3 beschriebene Algorithmus verwendet, sondern der Restricted EM von Kim und Tayler eingesetzt, welcher auch mit GNBs umzugehen vermag. Dieser findet für ein Problem mit linear unabhängigen Nebenbedingungen

$$\mathbf{A}\theta = \mathbf{b} \tag{3.11}$$

einen beschränkten Parametersatz θ^R, indem ein Newton-Raphson-Schritt auf die Parameter θ^U der unbeschränkten Lösung angewandt wird [57]. Die σ-GNB wurde dabei für alle Komponenten $i = 1, \ldots, K$ als Problem $\mathbf{A}\theta_k = \mathbf{b}$ definiert, mit

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -a_2 & 1 \end{bmatrix}, \theta_k = \begin{pmatrix} \pi_k \\ \mu_k \\ \sigma_k \end{pmatrix} \text{ und } \mathbf{b} = \begin{pmatrix} 0 \\ 0 \\ a_1 \end{pmatrix},$$

wobei a_1 bzw. a_2 die Koeffizienten von $h_1(\mu_k, \sigma_k)$ sind.

Des Weiteren wurden die Formeln (2.19), (2.17) und (2.18) bezüglich der neuen Log-Likelihood-Funktion aktualisiert. Weil die Intensitäten $\mathbf{y} = \{y_1, \ldots, y_N\}$ einfach als konstante Faktoren in (3.8) eingehen, konnten auf gleicher Weise auch die Berechnungsvorschriften zu (3.12), (3.13) und (3.14) abgewandelt werden.

$$N_k = \sum_{n=1}^{N} y_n \gamma_{nk}, \tag{3.12}$$

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} y_n \gamma_{nk} x_n, \tag{3.13}$$

$$\sigma_k^2 = \frac{1}{N_k} \sum_{n=1}^{N} y_n \gamma_{nk} (x_n - \mu_k)^2 \tag{3.14}$$

Da eine gefundene Lösung $\hat{\theta}$ einzig von den Startwerten abhängt, wurde der Restricted EM jeweils für M zufällig gewählte $\theta = \{(\pi_k, \mu_k, \sigma_k^2)\}_{k=1}^{K}$ ausgeführt. Analog zur Bestimmung von $\mathbf{g}^{(t)}$ in einem Schwarm, wurde anschließend aus der Menge $\{\hat{\theta}_1, \ldots, \hat{\theta}_M\}$ von lokalen Lösungen, das Element mit der größten Log-Likelihood als das globale Optimum ausgewählt.

3.2.2 PSO-MATLAB

Als Konkurrenz zum Restricted EM wurde der PSO der Global Optimization Toolbox[1] von MATLAB[2] (PSO-MATLAB) herangezogen. Dieser out-of-the-box Ansatz folgt dabei keinem gängigen Standard und kombiniert die alten Vorschriften (2.23) und (2.26) mit einer zufälligen Topologie, in welcher die Nachbarschaften in jeder Generation neu gebildet werden. Vergleichbar mit dem Konzept von Clerc wird hierbei die Anzahl der benachbarten Partikel je nach aktueller Schwarm-Performance vergrößert oder verkleinert [58]. Nach dem gleichen Schema wird darüber hinaus auch das Trägheitsmoment ω dynamisch angepasst [59].

[1] `https://de.mathworks.com/products/global-optimization.html`
[2] `https://ch.mathworks.com/products/matlab.html`

3.2.3 SPSO-2011

Ergänzend zum PSO-MATLAB wurde für die Arbeit ein eigener SPSO-2011, geschrieben, um eine Standardimplementierung für die Evaluation zu haben. Der Einsatz eines bestehenden SPSO-2011[3] der bereits in anderen wissenschaftlichen Arbeit referenziert wurde, hatte sich als wenig nützlich erwiesen, da dieser nicht für MATLAB optimiert worden war. Stattdessen wurde dieser als Referenz verwendet, um die Funktionsweise der eigens implementierten Methode zu validieren (siehe hierzu Anhang 6).

3.3 Hyperparameteroptimierer

3.3.1 Top-Down Strategie

Mit der Top-Down Strategie wurde ein Konzept entwickelt, dass initial mit einem hohen K-Wert gestartet wird und anschließend versucht, das Modell auf die wichtigsten Komponenten zu reduzieren. Dafür modifiziert der Ansatz den Funktionsablauf des PSOs bzw. SPSOs und ergänzt diesen um zwei zusätzliche Berechnungsschritte:

- Schritt 1 wird dabei der Auswertung der Zielfunktion vorgeschaltet, wo er die unzulässigen Komponenten $\Theta_{invalid}$ der K-UNB für die Dauer der Funktions-evaluation aus dem GMM entfernt. Dadurch erfolgt die Berechnung der Fitness ausschließlich auf Basis von Θ_{valid}, während die Verteilungen, die die niederintensiven Signalanteile modellieren, weder positiv noch negativ zur Optimierung beitragen. In Kombination mit der Straffunktion $s_2(\theta)$ sorgt diese Mechanik dafür, dass redundante Komponenten schrittweise verkleinert werden, bis sie schließlich in $\Theta_{invalid}$ landen. Auf diese Weise nimmt das GMMs eine effizientere Form an, ohne darüber an Aussagekraft zu verlieren (Abbildung 3.5).
- Schritt 2 wird als neue letzte Aktion dem Funktionsablauf angehängt, wo er die unzulässigen Verteilungen $\Theta_{invalid}$ aus dem finalen $\hat{\theta}$ entfernt.

[3] `https://www.particleswarm.info`

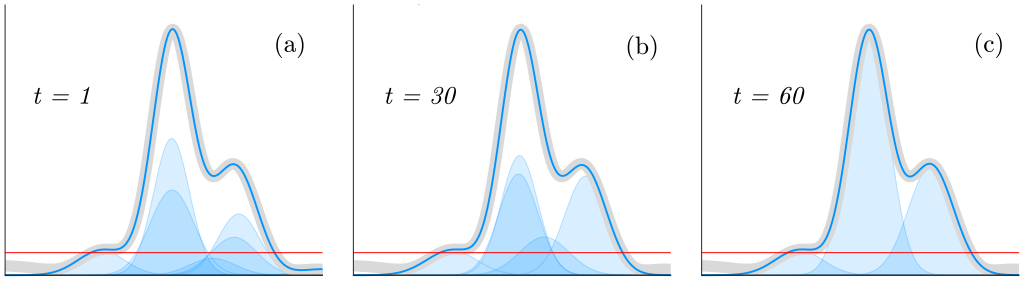


Abbildung 3.5: Visualisierung der Funktionsweise der Top-Down Strategie, wobei sich der Hyperparameter K über den Kurs der Optimierung von acht (a) auf fünf (b) und final auf drei (c) reduziert, während die Aussagekraft des Modells (blaue Linie) in den wichtigsten Signalbereichen unverändert bleibt. Hierbei stellt die rote Linie den Minimalwert I_{min} der K-UNB dar.

3.3.2 Tournament Strategie

Im Gegensatz zur Top-Down Strategie wurde mit der Tournament Strategie ein methodenunabhängiger Ansatz verfolgt, welcher mit Hilfe eines Turniersystem versucht, ein sinnvoll dimensioniertes GMM zu finden. Hierbei buhlen mehrere Instanzen der selben Optimierungsmethode um die Ausführung, wobei sich diese nur in der Größe ihres Hyperparameters K unterscheiden. Eine Instanz wird dabei umso häufiger ausgewählt, je vielversprechender ihre Lösung ist. Auf diese Weise werden gute Peak-Modelle noch weiter optimiert, wohingegen Methoden mit schlechten Ergebnissen nicht weiter ausgeführt und irgendwann ganz aus dem Wettkampf geworfen werden.

Der eigentliche Funktionsablauf lässt sich wie folgt zusammenfassen:

1. Initialisiere einen Pool $P_K = \{m_1, \ldots, m_K\}$ an Instanzen m_k der selben Methode, wobei jedes m_k die Parameter $\theta_k = \{(\pi_i, \mu_i, \sigma_i^2)\}_{i=1}^k$ optimiert. Des Weiteren sei F_{max} ein vorab definiertes Budget an Funktionsauswertungen, das für die Ausführung des Turniers zur Verfügung steht.

2. Wähle eine Instanz m_k aus P_K mittels roulette wheel selection und lasse diese für eine feste Anzahl an Funktionsevaluationen F laufen. Liefert die Optimierung ein Modell mit Komponenten $k \in \Theta_{invalid}$, so gilt dieser Ansatz als invalide und m_k wird dauerhaft aus dem Pool und damit dem Turnier entfernt.

3. Aktualisiere das Budget mit $F_{max} = F_{max} - F$.

4. Wiederhole den gesamten Vorgang ab Schritt 2, bis F_{max} erschöpft oder keine Optimierungsmethode in P_K vorhanden ist.
5. Wenn der Pool nicht leer ist, gib das θ_k mit dem besten Funktionswert aller verbliebenen Instanzen m_k als Lösung $\hat{\theta}$ zurück. Anderenfalls existiert für das gewählte I_{min} der K-UNB kein valides $\hat{\theta}$ und der gesamte Vorgang sollte mit einem anderen Minimalwert wiederholt werden.

4 Experimente und Ergebnisse

Nachfolgend wird zunächst ein neuer Ansatz beschrieben, um die Komplexität von Multi-Analyte-Peaks zu bestimmen, bevor die Datensätze dieser Arbeit vorgestellt werden. Daran schließt sich die Konfiguration der Methoden und die Parameterabschätzung der Nebenbedingungen an. Zum Schluss wird der Aufbau der Experimente erläutert, sowie die gefundenen Ergebnisse präsentiert und diskutiert.

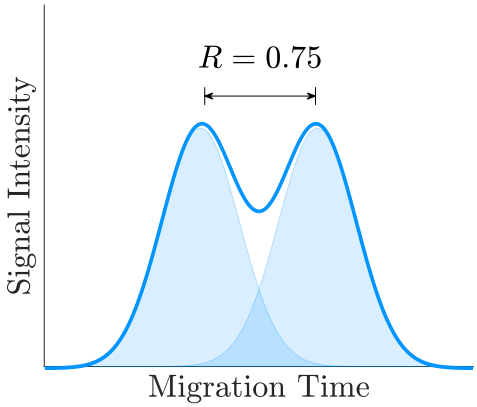
4.1 Datensätze

Aus Ermangelung einer vielfältigen Menge von CGE-LIF-Datensätzen wurden artifizielle Multi-Analyte-Peaks für die grundlegende Evaluation der Ansätze verwendet, bevor die daraus hervorgegangene *Siegermethode* final an echten Messdaten erprobt wurde.

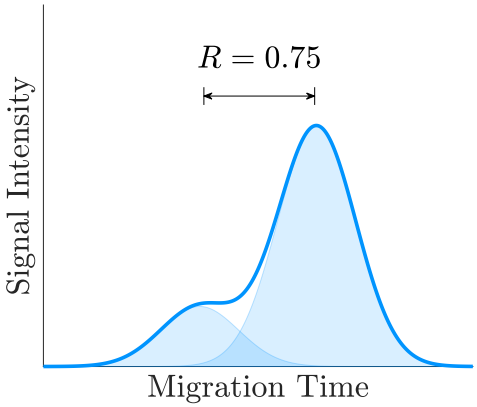
Dabei wurde anfänglich noch die Auflösung R herangezogen, um den Trenngrad zwischen Peaks zu bestimmen und so Einblick in die Komplexität der Daten zu gewinnen. Diese berechnet sich im Fall von GMMs für zwei benachbarte Komponenten k und $k+1$ nach Luckey *et al.* als

$$R = \sqrt{2\ln 2}\frac{\Delta\mu_k}{w_{\frac{1}{2}h_k} + w_{\frac{1}{2}h_{k+1}}}. \tag{4.1}$$

Hierbei ist $\Delta\mu_k = \mu_{k+1} - \mu_k$, für $\mu_k \leq \mu_{k+1}$, die zeitliche Differenz zwischen den beiden Komponenten, wohingegen $w_{\frac{1}{2}h_k}$ die Breite von k auf der halben Höhe $\frac{1}{2}h_k$ ist und sich über die Formel der Halbwertsbreite (engl.: *full width at half maximum*) $w_{\frac{1}{2}h_k} = 2\sqrt{2\ln 2}\sigma_k$ berechnen lässt [7]. Das heißt R ist im Kern ein normalisiertes Abstandsmaß, welches von der Signalintensität unbeeinflusst ist, obwohl die Peak-Höhe, wie in den Abbildungen 4.1a und 4.1b zu sehen, zweifelsohne Einfluss auf die Datenkomplexität nimmt.

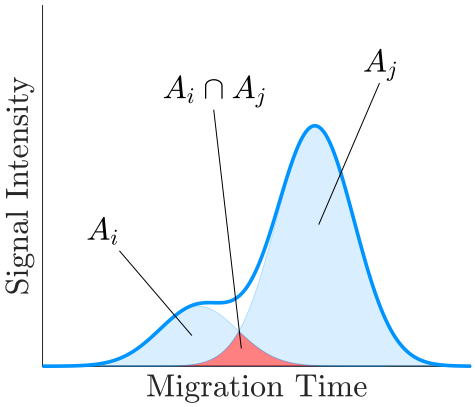


(a)



(b)

Abbildung 4.1: Illustration der Auflösung R als höhenunabhängiger Wert (a und b). Diese Art den Trenngrad zwischen Peaks zu bestimmen, steht dem neuen Ansatz gegenüber, welcher auf dem absoluten Flächen-Overlap $A_i \cap A_j$ basiert (c).



(c)

Aufgrund dieser Einschränkung wurde ein neuer Quantifizierungs-Ansatz für diese Arbeit entwickelt, welcher die Überlappung (engl.: *overlap*) zwischen den Komponenten $i, j = 1, \dots K$ als quadratische Matrix $\mathbf{O}$

$$\mathbf{O} = \begin{bmatrix} 1 & o_{12} & \dots & o_{1K} \\ o_{21} & 1 & \dots & o_{2K} \\ \vdots & \vdots & \vdots & \vdots \\ o_{K1} & o_{K2} & \dots & 1 \end{bmatrix} \text{ mit } o_{ij} = \frac{A_i}{A_i \cap A_j} \tag{4.2}$$

modelliert. Hierbei repräsentiert o_{ij} den relativen Overlap zwischen der i-ten und j-ten Komponente, mit A_i bzw. A_j als deren Flächen (Abbildung 4.1c). Darauf aufbauend wurden mit $Dist(\mathbf{O})$

$$Dist(\mathbf{O}) = \|\mathbf{O}\|_2 \tag{4.3}$$

und $Asym(\mathbf{O})$

$$Asym(\mathbf{O}) = Dist(\mathbf{O} - \mathbf{O}^{\intercal}) \tag{4.4}$$

zwei Maße zu Quantifizierung der Nähe und der Höhenunterschiede zwischen Peaks definiert. Zusammen formen beide Vorschriften einen simplen Ansatz, um die Komplexität von Messdaten zu bestimmen, der im Gegensatz zu der Auflösung R, auch die Intensitäten berücksichtigt.

4.1.1 Künstliche Messdaten

Um ein möglichst weites Spektrum an Testdaten abzudecken, wurden künstliche Multi-Analyte-Peaks auf Basis der Faktoren *Abstand* und *Intensität* für $K \in 2, 3, 4, 5$ erzeugt. Auf diese Weise konnten gezielt unterschiedlich komplexe Messungen konstruiert werden, welche einem einfachen, einem mittleren oder einem hohen Schwierigkeitslevel L_i mit $i \in$ {I,II,III} zugeordnet wurden (Abbildung 4.2). Insgesamt wurden so $3 \times 4 \times 5 = 60$ artifizielle Strukturen, verteilt auf fünf Datensätze, generiert, wobei jeder Satz $|L_i| \times |K| = 3 \times 4 = 12$ Signale beinhaltet (siehe Anhang 6).

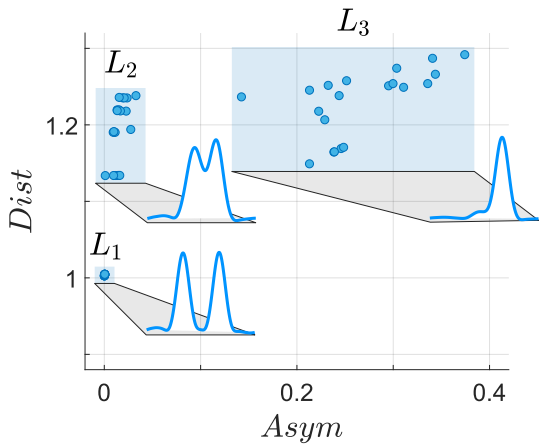


Abbildung 4.2: Die Komplexität der künstlichen Daten (blaue Kreise) sowie deren Zuordnung zu den Schwierigkeitsleveln L_I, L_{II} und L_{III}. Wie anhand der Beispielsignale zu sehen ist, sind die Level informell wie folgt definiert: L_I = Großer *Abstand* + gleiche *Intensität*; L_{II} = Kleiner *Abstand* + gleiche *Intensität*; L_{III} = Kleiner *Abstand* + unterschiedliche *Intensität*.

Diese Imitate basieren dabei auf GMMs, die in Anlehnung an echte CGE-LIF-Messdaten im Bereich $\mathbf{x} = [350, 390]$ angesiedelt worden sind und deren σ_k uniform um den Wert 1 gestreut wurden (Abbildung 4.5). Mit dem Ziel die Strukturen so komplex wie nötig zu halten, wurden die L_I-Komponenten in diesem Fenster entlang der äquidistanten Abstände $\mu_k = \mu_1 + 1.5(k-1)$, bzw.

die L_{II}- und L_{III}-Komponenten entlang $\mu_k = \mu_1 + 0.75(k-1)$, angeordnet. Darüber hinaus wurden für die Intensitäten verschiedene Peak-Muster definiert, welche an reellen Messsignalen orientiert waren und in ihrer Höhe jeweils leicht variiert wurden (Abbildung 4.3). Damit sollte der Umstand modelliert werden, dass zwei Proben des gleichen Glykoproteins selten vollkommen identisch sind und es alleine aufgrund der Sensitivität der Messmethoden zu minimalen Intensitätsänderungen kommt.

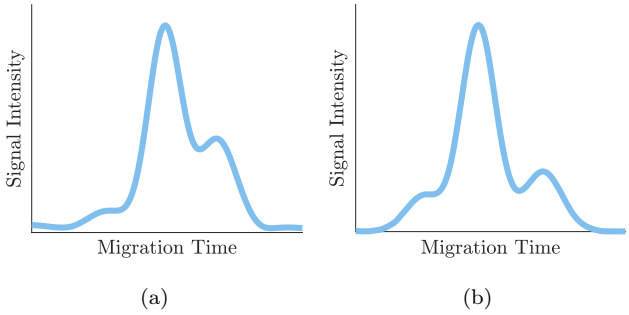


Abbildung 4.3: Darstellung eines künstlich erzeugten Multi-Analyte-Peaks (a) und seines realweltlichen Vorbilds (b) im Vergleich.

4.1.2 N-Glykan-Datensätze

Sämtliche der in dieser Arbeit verwendeten CGE-LIF-Messdaten wurden von der glyXera GmbH zur Verfügung gestellt. Von dieser wurde darüber hinaus auch die proprietäre Analyse-Software *glyXtool* bereitgestellt mit welcher die Daten vorprozessiert wurden. Dieser Vorgang umfasste dabei die Glättung des Signals, die Korrektur der Basislinie, sowie die Normalisierung der Migrationszeiten, d.h. die zeitliche Ausrichtung der Messungen anhand eines Standards. Des Weiteren wurden die Elektropherogramme, wie in Anhang 6 aufgeführt, gemäß ihrer biologischen Herkunft in Gruppen aufgeteilt:

- Glykoprotein-Typ I: Beinhaltet wenige, überwiegend mannose-reiche N-Glykane.
- Glykoprotein-Typ II: Weißt hauptsächlich mannose-reiche und Hybrid-Strukturen auf.
- Glykoprotein-Typ III: Besteht aus N-Glykanen, die den mannose-reichen und komplexen Subtypen zugeordnet werden.
- Glykoprotein-Typ IV: Setzt sich aus mannore-reichen Strukturen und Komplex-Typen zusammen.

- Glykoprotein-Typ V: Zeichnet sich durch eine hohe Glykan-Vielfalt aus und enthält sowohl mannose-reiche N-Glykane, wie auch Hybrid- und Komplex-Typen.

Aus diesen Daten wurden nur die Multi-Analyte-Peaks zur Evaluation ausgewählt, für die eine Ideallösung $\theta^* = \{(\pi_k^*, \mu_k^*, \sigma_k^{2*})\}_{k=1}^{K^*}$ anhand von Verdau-Informationen konstruiert werden konnte. Die Mehrzahl dieser N-Glykan-Strukturen wies dabei eine höhere Komplexität als die künstlich erzeugten Messsignale auf (Abbildung 4.2 und 4.4). Dies war sowohl auf eine größere Nähe der Strukturen zueinander als auch eine höhere Asymmetrie zwischen den Peaks zurückzuführen.

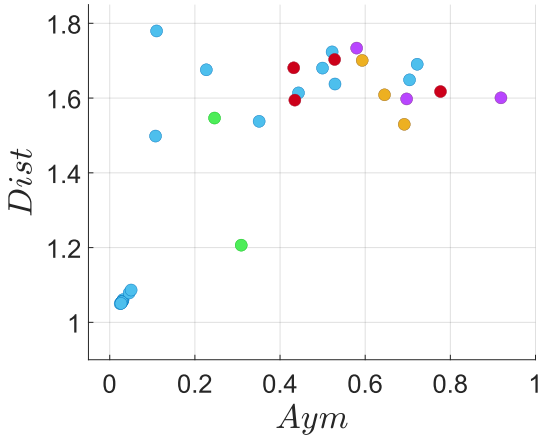


Abbildung 4.4: Die Komplexität der Multi-Analyte-Peaks des Glykanprotein-Typ I (blau), Glykanprotein-Typ II (grün), Glykanprotein-Typ III (lila), Glykanprotein-Typ IV (rot) und der Glykanprotein-Typ V (gelb).

4.2 Methodenkonfiguration und Parameterabschätzung

Die Optimierungsmethoden wurden für diese Arbeit mit den zufälligen Startwerten $\pi_k = U(0,1)$, $\mu_k = U(x_{min}, x_{max})$ und $\sigma_k = U(0.1, 2.5)$ initialisiert, wobei x_{min} die minimale und x_{max} die maximale Migrationszeit des zu modellierenden Signalbereichs ist. Des Weiteren wurde die Schwarmgröße, bzw. im Fall des Restricted EM die Kandidatenmenge, M entsprechend der Komponentenanzahl auf $M = 50 \times K$ festgelegt, während die verbliebenen PSO-Parameter auf die von MATLAB empfohlenen Standardwerte gesetzt wurden.

Die Koeffizienten der σ-GNB wurden entsprechend des in 3.1.2 beschriebenen Vorgehens anhand der Mannose-Peaks des Glykoprotein-Typ 1 approximiert. Zur Vermeidung von lokalen Minima wurde der erste Schritt für jeden Peak 30-mal wiederholt, wobei als Optimierungsmethode ein PSO-MATLAB mit $M = 50$ zufällig initialisierten Partikeln verwendet wurde. Basierend auf den μ- und σ-Werten des besten $\hat{\theta}$ aller Wiederholungen wurde anschließend, wie in Schritt 2 beschrieben, ein lineares Regressionsmodell mit den Koeffizienten $a_1 = 0.4545$ und $a_2 = 0.0015$ konstruiert (Abbildung 4.5).

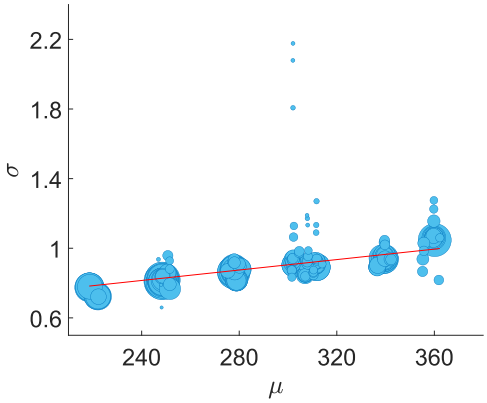


Abbildung 4.5: Darstellung des WLS-Modells (rote Linie) und der μ/σ-Paare (blau), wobei die Kreisgrößen die Peak-Höhen wiederspiegeln.

Die Minimalintensität I_{min} der K-UNB wurde wiederum als

$$I_{min} = \frac{LOQ}{A} \tag{4.5}$$

berechnet, wobei A die Fläche des zu modellierenden Messsignals $\mathbf{y}$ ist, bevor diese auf 1 normalisiert wurde. Der LOQ wurde entsprechend der Anleitung von Wenzel *et al.* mittels

$$LOQ = median(\mathbf{y}_{noise}) + 10 \cdot SNR(\mathbf{y}_{noise}) \tag{4.6}$$

bestimmt, wohingegen das Signal-Rausch-Verhältnis (engl.: *signal-to-noise ratio*, SNR) nach Ullsten *et al.* als

$$SNR(\mathbf{y}_{noise}) = median(|y_i - median(\mathbf{y}_{noise})|) \tag{4.7}$$

ermittelt wurde, mit $\mathbf{y}_{noise}$ als ein Bereich des Signals, das nur Rauschen enthält [60][61].

4.3 Experimentaufbau und Evaluation

Zur systematischen Analyse der Lösungsansätze dieser Arbeit wurden drei aufeinander aufbauende Testphasen durchgeführt:

1. Phase: Vergleich der Zielfunktionen und Optimierungsmethoden anhand der artifiziellen Messdaten. Das Ziel ist es dabei einen sinnvollen Wert für den Straffaktor c_1 der erweiterten Zielfunktion $\phi(\theta, \mathbf{x}, \mathbf{y})$

   $$\phi(\theta, \mathbf{x}, \mathbf{y}) = (1 - c_1 - c_2)f + c_1 s_1(\theta) + c_2 s_2(\theta)$$

   zu identifizieren, weshalb $c_2 = 0$ gesetzt und die Anzahl an Komponenten K für jeden Datensatz vorgeben wird.

2. Phase: Bewertung der beiden Hyperparameteroptimierer auf Basis des c_1-Werts aus Phase 1 und der künstlichen Signaldaten mit der Intention einen geeigneten Faktoren c_2 für $\phi(\theta, \mathbf{x}, \mathbf{y})$ zu finden.

3. Phase: Evaluation der aus den beiden vorangegangen Testphasen hervorgegangen *Siegermethode* mittels Realdaten.

Die wichtigsten Ergebnisse der Arbeit, auf welchen die nachfolgenden Tabellen basieren, sind in Anhang 6 zu finden. Dort sind die Verfahren jeweils durch die Mediane ihrer Funktionswerte repräsentiert, wobei die Optimierung für jeden Datensatz 31-mal wiederholt wurde, um statistisch valide Aussagen treffen zu können.

4.3.1 Phase 1

Die ausgewählten Ansätze des ersten Testphase sind in Tabelle 4.1 aufgeführt.

Tabelle 4.1: Lösungsansätze der Phase 1.

Ansatz	Zielfunktion f		Optimierungsmethode		
	LLH	RMSE	Restricted EM	PSO-MATLAB	SPSO-2011
(1)	✓		✓		
(2)	✓			✓	
(3)	✓				✓
(4)		✓		✓	
(5)		✓			✓

Da die Werte der LLH- und RMSE-Funktion mehrere Größenordnungen auseinanderliegen, wurden jeweils zwei Straffaktor-Mengen $c_1^L \in \{0, 0.5, 0.9\}$ und $c_1^R \in \{0, 0.01, 0.1\}$ definiert. Darüber hinaus wurde das Abbruchkriterium auf die maximale Anzahl von $5 \cdot 10^4$ Funktionsaufrufe festgesetzt.

Zur Evaluation der Lösungsansätze wurde als externes Bewertungsmaß der Flächenfehler $E^A(\hat{\theta}, \mathbf{x}, \mathbf{y})$ zwischen dem künstlichen Messsignal $\mathbf{y}$ und dem optimierten GMM $p(\hat{\theta}, \mathbf{x})$ berechnet.

$$E^A(\hat{\theta}, \mathbf{x}, \mathbf{y}) = \int |\mathbf{y} - p(\hat{\theta}, \mathbf{x})| \, dx \tag{4.8}$$

Auf dieser Grundlage wurden jeweils die Mediane von zwei Ansätzen paarweise mit einem zweiseitigen Wilcoxon-Vorzeichen-Rang-Test gegeneinander verglichen, wobei als Nullhypothese angenommen wurde, dass diese von der gleichen Grundgesamtheit stammen. Wie in der Arbeit von Derrac *et* al. beschrieben, summiert der Vorzeichen-Rang-Test dazu die Performancedifferenz d_i der beiden Verfahren über alle Probleme i mittels

$$R^+ = \sum_{d_i > 0} rang(d_i) + \frac{1}{2} \sum_{d_i = 0} rang(d_i) \tag{4.9}$$

auf. Hierbei wird R^+ die Rangsumme der positiven Differenzen genannt, welche angibt wie häufig der erste Ansatz den zweiten geschlagen hat [62].

Im Folgenden werden zunächst die Ergebnisse der LLH-, bzw. RMSE-basierten Lösungsansätze präsentiert, bevor die besten Verfahren aus beiden Kategorien einander gegenübergestellt werden.

Log-Likelihood

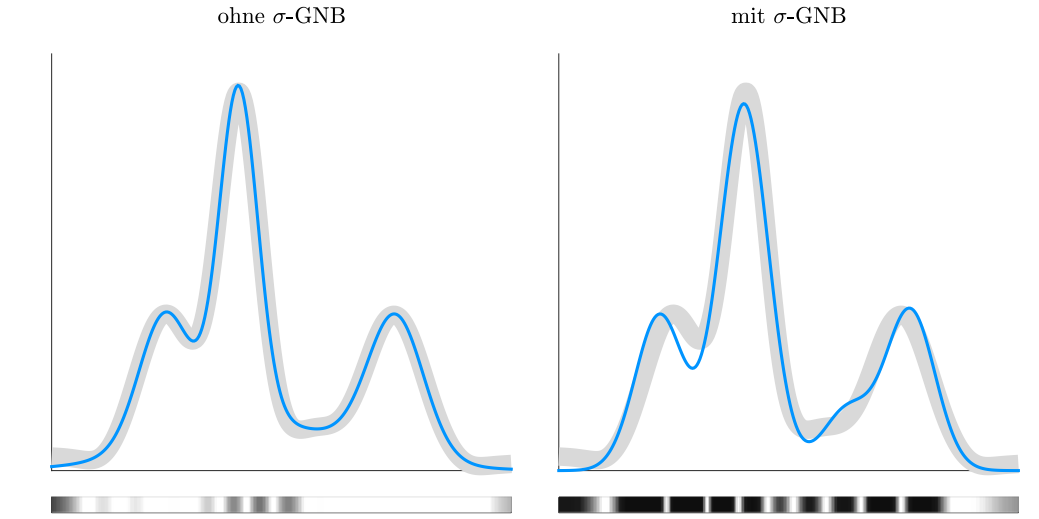
Der GNB-freie Ansatz 1 dominiert die anderen Verfahren teilweise deutlich, wie in Tabelle 4.2 und Abbildung 4.6 zu sehen ist. Das heißt wenn es darum ging eine Maximum-Likelihood-Lösung zu finden, dann war der unbeschränkte EM die effizienteste Optimierungsmethode dieser Arbeit. Damit bestätigten die Ergebnisse das, was in der Literatur beschrieben ist: Die analytische Maximierung der LLH-Funktion für GMMs ist einer Optimierung mittels heuristischer Suche vorzuziehen. Dazu passt auch, dass die Lösungsansatze 2 und 3 im Fall $c_1^L = 0$ am besten gegen Ansatz 1 performen (Tabelle 4.2, erste Spalte).

Bei den PSO-basierten Verfahren ist Ansatz 3 wiederum Ansatz 2 unterlegen, welcher im Schnitt die bessere Leistung für den gleichen Straffaktor zeigt, trotz der veralteten Rechenvorschriften. Dies spricht dafür, dass der PSO-MATLAB auf die verwendeten Standardparameter getunt war.

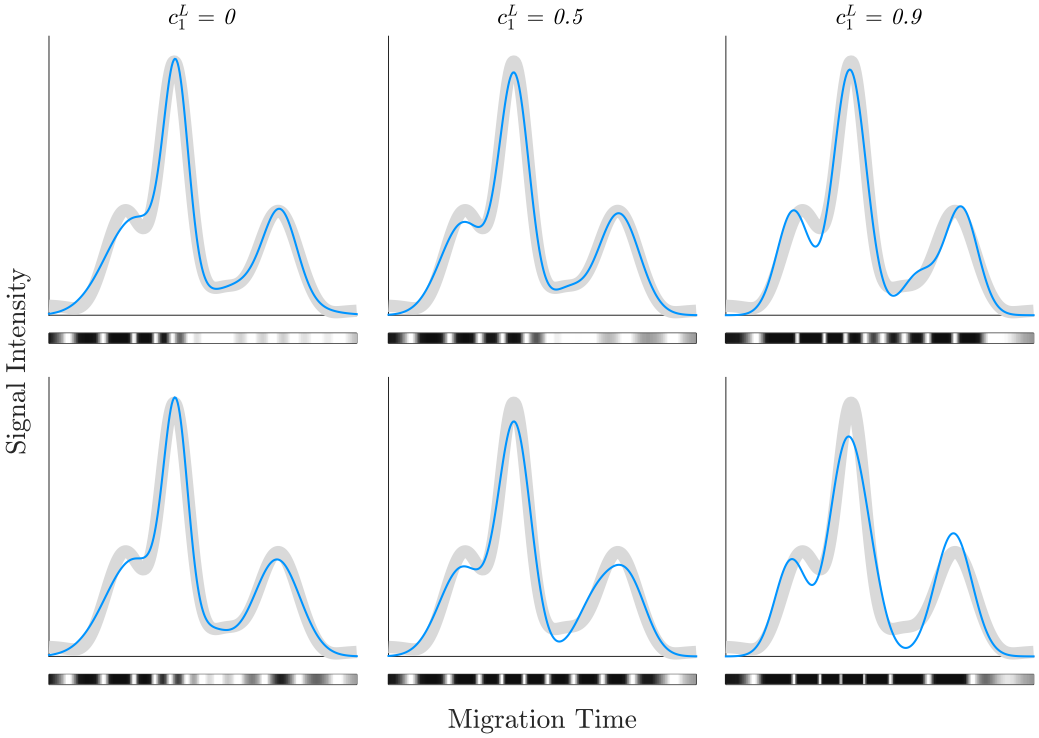
Tabelle 4.2: Die R^+-Werte des Wilcoxon-Vorzeichen-Rang-Tests für die Ansätze 1 bis 3, mit p als Indikator, ob sich zwei Lösungsansätze signifikant voneinander unterscheiden ($p < 0.05$, schwarze Schriftfarbe) oder nicht ($p \geq 0.05$, rote Schriftfarbe). Des Weiteren sind in jeder Spalte die Maxima fett hervorgehoben.

VS.		Ansatz 1		Ansatz 2 mit c_1^L			Ansatz 3 mit c_1^L		
		ohne σ-GNB	mit σ-GNB	0	0.5	0.9	0	0.5	0.9
Ansatz 1	ohne σ-GNB		**1817**	**1812**	**1253**	**1815**	**1801.5**	**1280**	1824
	mit σ-GNB	13		139	94	459.5	299	131	909 ($p > 0.5$)
Ansatz 2 mit c_1^L	0	1320	1691		794 ($p > 0.2$)	1671	1752.5	902 ($p > 0.5$)	1776
	0.5	577	1736	1036 ($p > 0.2$)		1729	1169 ($p > 0.05$)	1422.5 ($p > 0.05$)	1811
	0.9	15	1430.5	159	101		346	134	1131 ($p > 0.2$)
Ansatz 3 mit c_1^L	0	1038.5	1531	1087.5	661 ($p > 0.05$)	1484		769 ($p > 0.2$)	1652
	0.5	550	1699	928 ($p > 0.5$)	1061.5 ($p > 0.05$)	1696	1061 ($p > 0.2$)		**1828**
	0.9	6	921 ($p > 0.5$)	54	19	818 ($p > 0.2$)	178	2	

Während die Straffaktoren $c_1^L = 0$ und $c_1^L = 0.5$ im direkten Vergleich ähnliche Ergebnisse liefern (Tabelle 4.2, rote Zellen), geht für alle Verfahren die durchschnittliche Performance unter einer zu strikten Verwendung der GNB zurück (Tabelle 4.2, Zeile 2, 5 und 8). Dabei implizieren die überwiegend dunklen Heatmaps mit den kurzen, hellen Unterbrechungen in Abbildung 4.6, dass die GMMs zwischen einer ständigen Über- und Unterschätzung der Fläche oszillieren und das Messsignal nur schneiden. Dieses Verhalten lässt sich durch die gegensätzlichen Ziele der Log-Likelihood-Funktion und der σ-GNB erklären. Bei Ersterer wollen selbst kleinste y_i von der Maximum-Likelihood-Lösung abgebildet werden, wohingegen die Zweiter vorgibt wie viel Signal von einer Komponente maximal modelliert werden kann. Infolgedessen stellen die GMMs reine Kompromisslösungen dar, welche die Peak-Formen nur grob approximieren.



(a)



(b)

Abbildung 4.6: Beispielhafte Ergebnisse (blaue Linie) der Ansätze 1 (a), 2 und 3 (b, von oben nach unten). Für jedes Modell ist der Verlauf der Funktion $E^A(\hat{\theta}, \mathbf{x}, \mathbf{y})$ als Heatmap dargestellt. Dabei gilt: Je besser der Funktionswert, desto heller die Farbe.

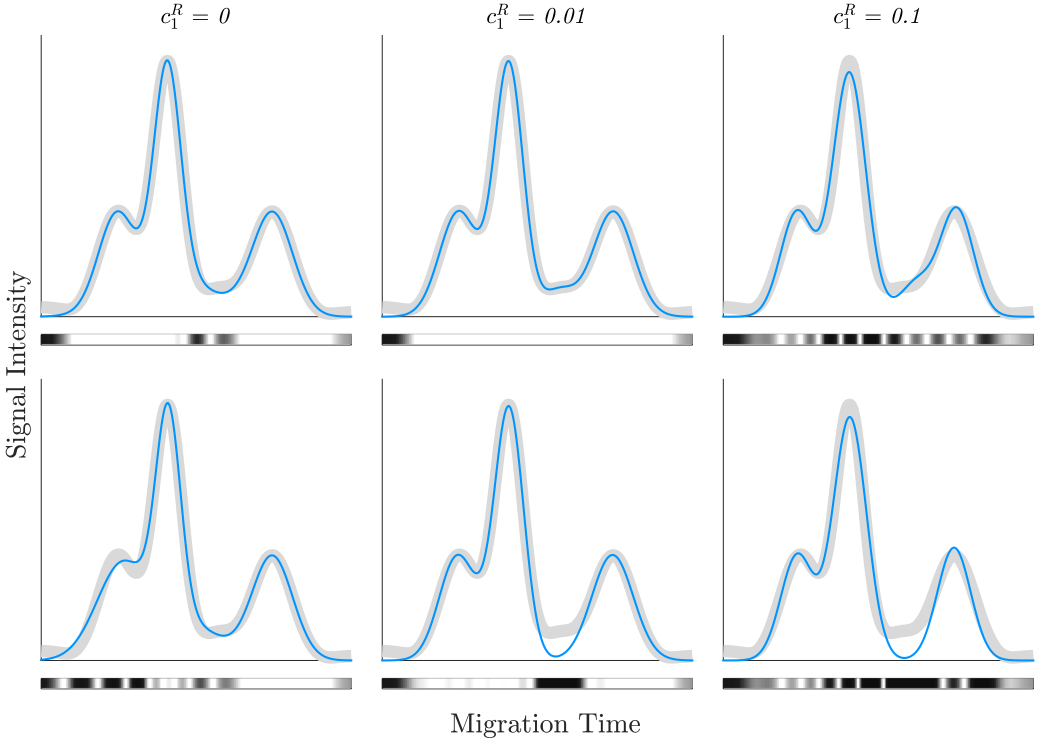
Root-Mean-Square Error

Im Gegensatz zur den LLH-basierten Ansätzen wirkt sich die σ-GNB positiv auf die Resultate der RMSE-Verfahren aus (Tabelle 4.3 und Abbildung 4.7). Während sich die Ergebnisse der Faktoren $c_1^L = 0$ und $c_1^L = 0.01$ im Vergleich kaum unterscheiden (Tabelle 4.3, rote Zellen), zeigt Ansatz 4, wie auch der unterlegene Ansatz 5, seine durchschnittlich beste Performance bei einem c_1^R-Wert von 0.01 (Tabelle 4.3, Zeile 2 und 4).

Tabelle 4.3: Die R^+-Werte des Wilcoxon-Vorzeichen-Rang-Tests für die Ansätze 4 und 5, mit p als Indikator, ob sich zwei Lösungsansätze signifikant voneinander unterscheiden ($p < 0.05$, schwarze Schriftfarbe) oder nicht ($p \geq 0.05$, rote Schriftfarbe). Des Weiteren sind in jeder Spalte die Maxima fett hervorgehoben.

VS.		Ansatz 4 mit c_1^R			Ansatz 5 mit c_1^R		
		0	0.01	0.1	0	0.01	0.1
Ansatz 4 mit c_1^R	0		823 ($p > 0.2$)	1622	**1509.5**	1204	1821
	0.01	1007 ($p > 0.2$)		**1742**	1318	**1493**	**1830**
	0.1	208	88		381	406	1761
Ansatz 5 mit c_1^R	0	497.5	512	1449		873 ($p > 0.5$)	1789
	0.01	626	571	1424	957 ($p > 0.5$)		**1830**
	0.1	9	0	69	41	0	

Anders als noch bei der Log-Likelihood-Funktion steht die σ-GNB dabei nicht im direkten Konflikt zur E^R-Fehlerfunktion. Stattdessen reduziert diese den Parameterraum der σ_k, wodurch sich die Suche nach Lösungen, welche die Signalform gut modellieren, vereinfacht (Abbildung 4.7, mittlere Ergebnisse). Analog zu den Ansätzen 1 bis 3 (Tabelle 4.2, Zeile 2, 5 und 8) kehrt sich dieser Effekt ins Gegenteil, sobald der Straffaktor c_1^R zu groß gewählt wird (Tabelle 4.3, Zeile 3 und 6) und die Peak-Breiten sich zu sehr über die GNB definieren. Die Folge sind GMMs, welche erneut zwischen einer Über- und Unterschätzung des Signals schwanken (Abbildung 4.7, rechte Ergebnisse) und damit große Ähnlichkeit mit den Resultaten aus Abbildung 4.6 haben.



(a)

Abbildung 4.7: Beispielhafte Ergebnisse (blaue Linien) der Ansätze 4 und 5 (von oben nach unten). Für jedes Modell ist der Verlauf der Funktion $E^A(\hat{\theta}, \mathbf{x}, \mathbf{y})$ als Heatmap dargestellt. Dabei gilt: Je besser der Funktionswert, desto heller die Farbe.

Log-Likelihood und Root-Mean-Square Error im Vergleich

Wie Tabelle 4.4 zeigt, geht Lösungsansatz 4 als klarer Sieger aus der Gegenüberstellung der besten LLH- und Error-basierten Verfahren hervor.

Tabelle 4.4: Die R^+-Werte des Wilcoxon-Vorzeichen-Rang-Tests für den besten LLH- und RMSE-Ansatz der ersten Testphase. Hierbei gibt die schwarze Schriftfarbe an, dass sich die beiden Lösungsansätze statistisch signifikant voneinander unterscheiden ($p < 0.05$).

VS.	Ansatz 1 ohne σ-GNB	Ansatz 4 mit $c_1^R = 0.01$
Ansatz 1 ohne σ-GNB		372
Ansatz 4 mit $c_1^R = 0.01$	1458	

Grund dafür ist die stärkere Orientierung der GMMs der RMSE-Funktion an der eigentlichen Form der Multi-Analyte-Peaks (Abbildung 4.8, rechtes Ergebnis). Während die Funktion in der Lage ist, kleinere Fehler durch die Modellierung der prominentesten Strukturen zu kompensieren, haben selbst niedrige Werte großen Einfluss auf eine Maximum-Likelihood-Lösung und wollen von dieser erklärt werden. Die Eigenschaft, Verantwortung für jede Beobachtung übernehmen zu wollen, macht die Log-Likelihood-Funktion dabei extrem anfällig für Störsignale. So approximiert Ansatz 1 den linken Randbereich in Abbildung 4.8 besser (hellere Heatmap) als Ansatz 4. Dies ist insbesondere im Hinblick auf CGE-LIF-Messungen ein kritischer Aspekt, da diese Daten nie vollkommen rauschfrei sind. Darüber hinaus sind die Ergebnisse der LLH-Funktion dadurch maßgeblich von dem gewählten Signalbereich abhängig, wohingegen Error-basierte Verfahren automatisch die größten Strukturen modellieren.

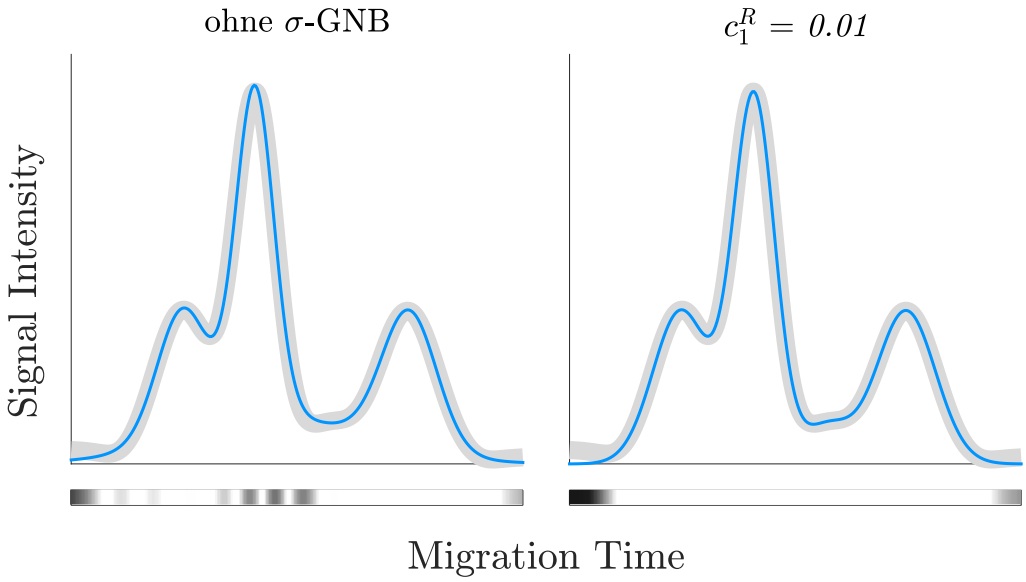


Abbildung 4.8: Beispielhafte Ergebnisse (blaue Linien) der Ansätze 1 (rechts) und 4 (rechts). Für jedes Modell ist der Verlauf der Funktion $E^A(\hat{\theta}, \mathbf{x}, \mathbf{y})$ als Heatmap dargestellt. Dabei gilt: Je besser der Funktionswert, desto heller die Farbe.

4.3.2 Phase 2

Die untersuchten Lösungsansätze der Phase 2 sind in Tabelle 4.5 aufgelistet.

Tabelle 4.5: Lösungsansätze der Phase 2.

Ansatz	Zielfunktion f		Optimierungsmethode			Hyperparameter-optimierer	
	LLH	RMSE	Restricted EM	PSO-MATLAB	SPSO-2011	Tournament Strategie	Top-Down Strategie
(1)	✓		✓			✓	
(2)	✓			✓		✓	
(3)	✓			✓			✓
(4)	✓				✓	✓	
(5)	✓				✓		✓
(6)		✓		✓		✓	
(7)		✓		✓			✓
(8)		✓			✓	✓	
(9)		✓			✓		✓

Obwohl die Ergebnisse aus Tabelle 4.2 eine unbeschränkte Optimierung der Log-Likelihood-Funktion nahelegen, wurden die LLH-basierten Ansätze der zweiten Testphase dennoch mit der σ-GNB bzw. mit einem Straffaktor von $c_1^L = 0.5$ initialisiert. Auf diese Weise sollten realitätsferne Lösungen mit zu schmalen oder zu breiten Komponenten verhindert werden. Im Gegensatz dazu wurde c_1^R entsprechend der Resultate aus Tabelle 4.4 auf 0.01 gesetzt. Darüber hinaus wurden analog zur ersten Testphase zwei Mengen $c_2^L \in \{0.01, 0.1, 0.2\}$ und $c_2^R \in \{0.0001, 0.001, 0.01\}$ definiert. Des Weiteren wurde eine Obergrenze von $K = 10$ Komponenten für die Hyperparameteroptimierer gewählt, sowie das Abbruchkriterium auf $1 \cdot 10^5$ Funktionsaufrufe festgelegt.

Zur Evaluation der Ansätze wurden als Vergleichskriterium die Differenzen ΔK zwischen der tatsächlichen Anzahl an Peaks K und der Anzahl an Komponenten $\hat{K}$ der gefunden Lösung $\hat{\theta} = \{(\hat{\pi}_k, \hat{\mu}_k, \hat{\sigma}_k^2)\}_{k=1}^{\hat{K}}$ berechnet.

$$\Delta K = K - \hat{K} \tag{4.10}$$

Für die Repräsentation der Ergebnisse wurden Heatmaps verwendet, welche nach dem in Abbildung 4.9 illustrierten Schema eingefärbt wurden.

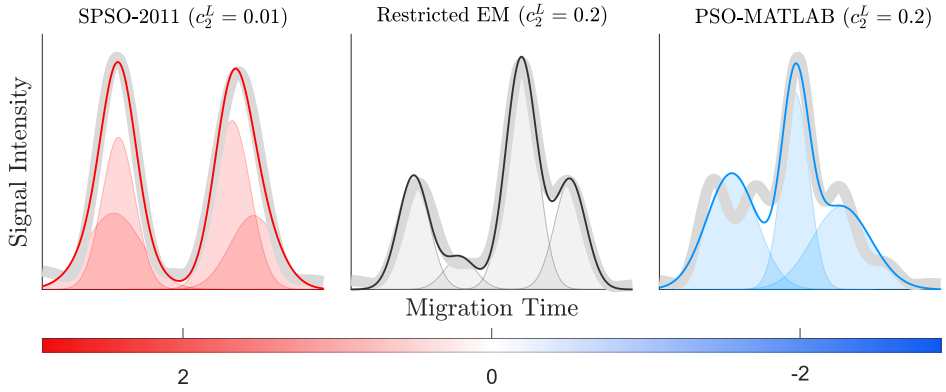


Abbildung 4.9: Die farbliche Codierung von ΔK, visualisiert anhand von Beispielmodellen (rot, schwarze, und blaue Linie). Dabei steht die Farbe Weiß für GMMs mit $\Delta K = 0$, wohingegen Modelle, welche die Komponentenanzahl über- ($\Delta K > 0$) bzw. unterschätzen ($\Delta K < 0$) in Rot bzw. Blau gehalten sind.

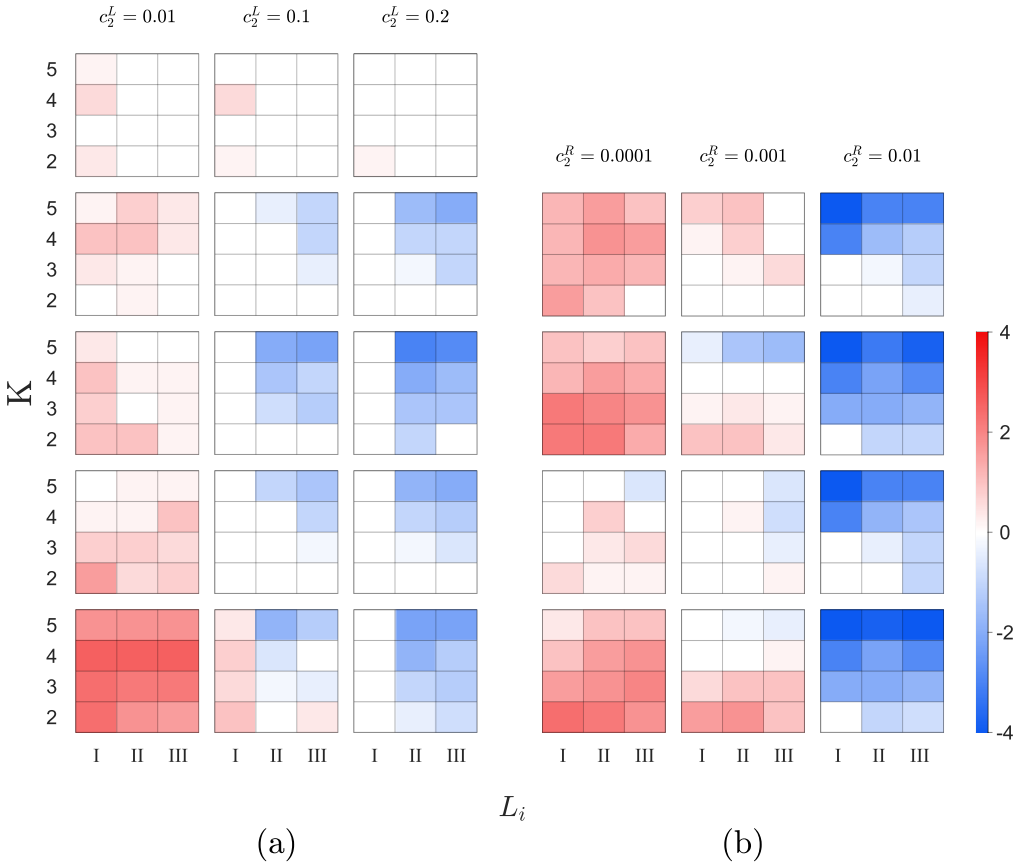


Abbildung 4.10: Heatmap-Darstellung der über die Datensätze gemittelten ΔK-Werte für die Ansätze 1 bis 5 (a, von oben nach unten) und 6 bis 9 (b, von oben nach unten).

Wie in Abbildung 4.10 zu sehen ist, ist Ansatz 1 den anderen Verfahren bei der Abschätzung von K deutlich überlegen, wobei deren Ergebnisse maßgeblich von der Wahl des Straffaktors c_2^L bzw. c_2^R abhängen. Wird der Faktor dabei zu klein gewählt ($c_2^L = 0.01$, $c_2^R = 0.0001$), gibt es für den Optimierer keinen Anreiz, um redundante Komponenten zu eliminieren und so eine effizientere Modellform zu finden. Anders ausgedrückt, K wird überschätzt. Fällt die Strafe hingegen zu streng aus ($c_2^L = 0.2$, $c_2^R = 0.01$), wird die Anzahl an Komponenten auf ein Minimum reduziert, was eine Unterschätzung von K zur Folge hat. Hierbei scheint ΔK direkt mit den, in Abbildung 4.11 dargestellten, gemittelten Ergebnissen der E^A-Funktion zu korrelieren. Dabei ist der Flächenfehler dort besonders klein (grün) bzw. groß (orange), wo sehr viele bzw. sehr wenige Komponenten für die Modellierung des Signals zur Verfügung stehen und dieses daher eher besser bzw. schlechter abgebildet werden kann.

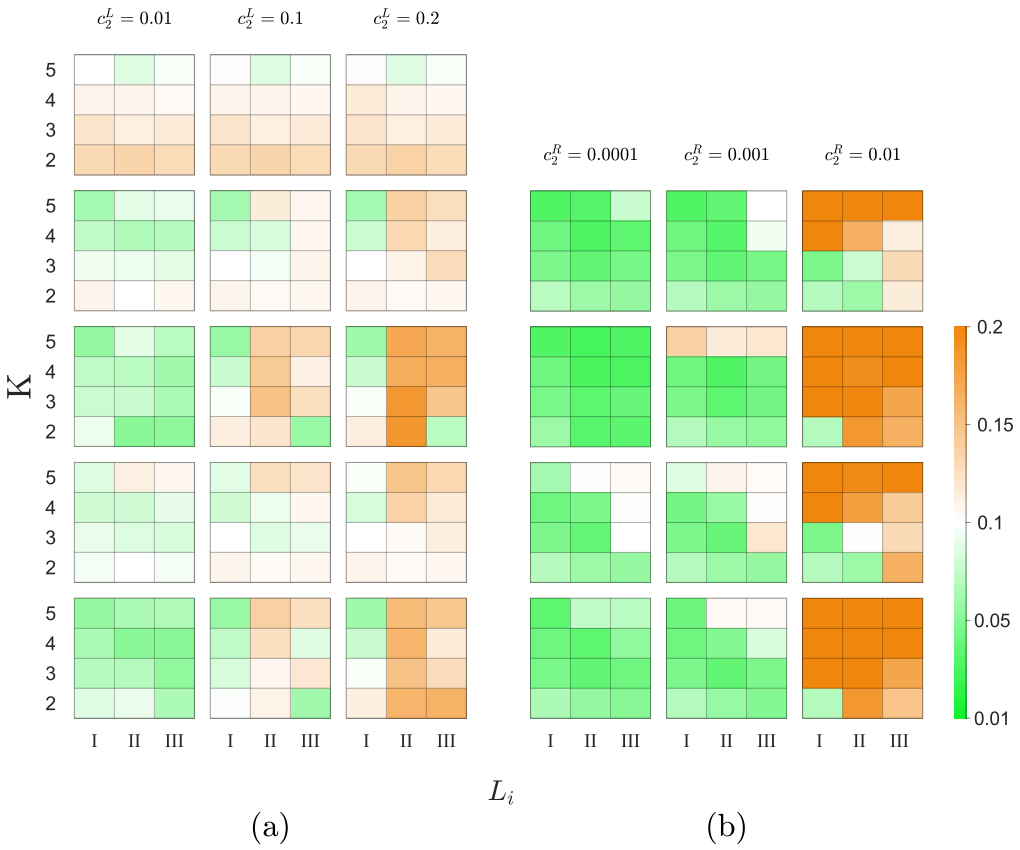


Abbildung 4.11: Heatmap-Darstellung der über die Datensätze gemittelten Werte der E^A-Funktion für die Ansätze 1 bis 5 (a, von oben nach unten) und 6 bis 9 (b, von oben nach unten).

Die Wahl eines sinnvollen Straffaktors für die Verfahren 2 bis 9 wird dadurch verkompliziert, dass die konkrete Höhe des Strafterms, d.h. der Einfluss des Faktors auf das Ergebnis, an die Komplexität der zu modellierenden Daten gebunden ist. Wie beispielhaft in der mittleren Heatmap der zweiten Zeile in Abbildung 4.10b dargestellt ist, führt der selbe Faktor je nach Datensatz zu einer Über- ($K = 2$, $L_i \in \{I, II\}$) bzw. einer Unterschätzung von K ($K = 5$, $L_i \in \{II, III\}$). Infolgedessen fällt E^A entsprechend besser oder schlechter aus (Abbildung 4.11b, Zeile 2, mittlere Heatmap).

Werden die Ansätze 2 bis 5 und 6 bis 9 miteinander verglichen, fällt auf, dass die LLH-basieren Verfahren besser dazu in der Lage sind den Hyperparameter zu approximieren (mehr weiße Zellen). Die Erklärung dafür ist, dass die Optimierung der Log-Likelihood-Funktion zu Lösungen führt, welche versuchen das Signal zu erklären, statt ein Abstand- oder Fehlermaß zu minimieren. Weil $\mathbf{y}$ dabei multiplikativ in die Berechnung eingeht, ist E^L direkt von der zu modellierenden Signalmenge abhängig und wächst mit dieser mit (Abbildung 4.12, E^L-Werte).

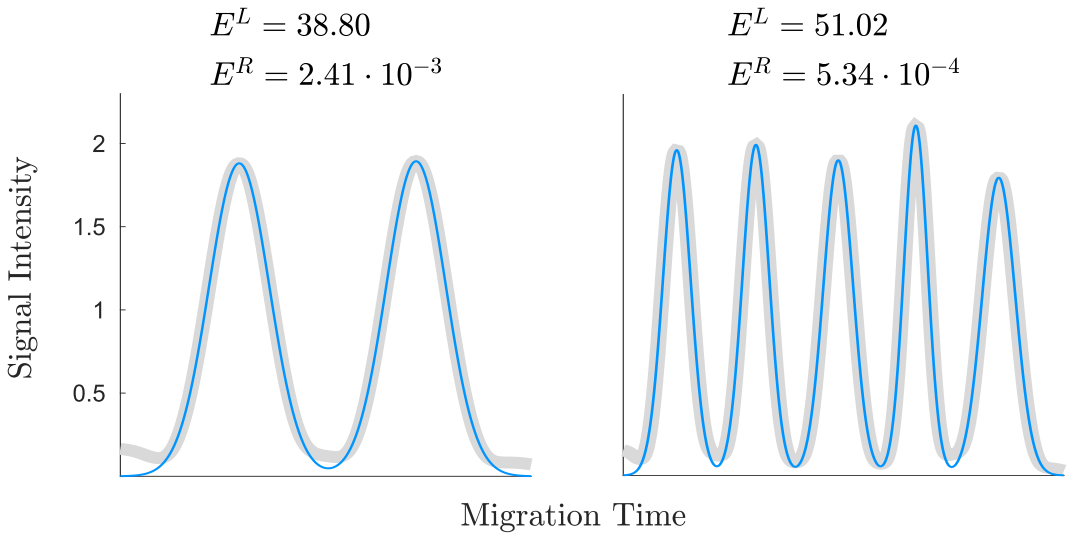


Abbildung 4.12: Zwei beispielhafte GMMs (blaue Linie) ausgewertet für die LLH-Funktion E^L und die RMSE-Funktion E^R.

Auf diese Weise bleibt das Werteverhältnis zwischen Ziel- und Straffunktion tendenziell bestehen, da nicht mehr nur der Strafterm mit der tatsächlichen Komponentenanzahl K skaliert. Die Verwendung der RMSE-Funktion bewirkt

hingegen das genaue Gegenteil: Je mehr Komponenten Teil des Multi-Analyte-Peaks sind, umso kleiner fällt der Fehler, d.h. der relative Anteil des Rauschens am Signal, aus (Abbildung 4.12, E^R-Werte). Folglich müsste der Straffaktor bei steigendem K verkleinert werden, um den gleichen Strafeffekt für verschiedene Daten zu erzielen.

Das heißt, was im vorherigen Abschnitt 4.3.1 als kritische Eigenschaft der LLH-Funktion aufgefasst wurde, entpuppt sich für die Abschätzung von K als großer Vorteil. Es ist daher naheliegend zu vermuten, dass Ansatz 1 deswegen so gut performt, weil der EM-Algorithmus die effizienteste Methode zum Finden einer Maximum-Likelihood-Lösung ist. Analog zu den Resultaten der ersten Testphase ist der PSO-MATLAB dem SPSO-2011 überlegen (Abbildung 4.10a und 4.10b, Zeile 2 und 4 im Vergleich zu 3 und 5). Dies spricht erneut dafür, dass dieser auf die verwendeten Standardparameter getunt war.

Werden die Ergebnisse anhand des Hyperparameteroptimierers unterschieden, ist festzustellen, dass sämtliche Extremwerte, dargestellt in Tiefrot bzw. Tiefblau, bei der Top-Down Strategie zu finden sind (Abbildung 4.10a und 4.10b, Zeilen 3 und 5). Dies liegt darin begründet, dass die Tournament Strategie sehr viel restriktiver mit möglichen Lösungskandidaten umgeht und diese entweder nicht weiter optimiert (wenn deren Funktionswert zu schlecht wird) oder sofort dauerhaft entfernt (wenn diese invalide werden). Im Gegensatz dazu schließt die Top-Down Strategie unzulässige Komponenten nur vorübergehend für die Fitnessevaluation aus, wodurch unterschiedlich dimensionierte GMMs gleichzeitig, als Teil des selben Prozesses, optimiert werden. Auf diese Weise haben Lösungskandidaten eher die Möglichkeit in das eine ($c_2^L = 0.01$, $c_2^R = 0.0001$) oder andere Extrem ($c_2^L = 0.2$, $c_2^R = 0.01$) auszuschlagen, weshalb K stärker über- bzw. unterschätzt wird.

4.3.3 Phase 3

Für die abschließende Testphase wurde ein Hybrid-Ansatz auf Basis der vorangegangenen Ergebnisse konstruiert, der die folgenden Schritte beinhaltet:

1. Approximiere einen Hyperparameter $\hat{K}$ mittels des ersten Ansatzes der zweiten Phase für einen maximalen Wert von $K = 10$ und einen Straffaktor von $c_2^L = 0.2$.

2. Berechne eine finale Lösung $\hat{\theta}$ anhand des abgeschätzten $\hat{K}$ und des zweiten Lösungsansatzes der ersten Phase für $c_1^R = 0.01$.

Analog zu den Phasen 1 und 2 wurden dabei die Funktionsaufrufe auf $1 \cdot 10^5$ für Schritt 1, bzw. auf $5 \cdot 10^4$ für Schritt 2, festgesetzt. Wie eingangs erläutert, wurde der Lösungsansatz anhand der CGE-LIF-Messdaten evaluiert, wobei die Resultate in Qualitätsklassen unterteilt wurden:

- Klasse 1: Der Hyperparameter $\hat{K}$ über- oder unterschätzt das K^* der konstruierten Ideallösung θ^*, weshalb $\hat{\theta}$ ebenfalls fehlerhaft ist.

- Klasse 2: Der $\hat{K}$-Wert stimmt mit K^* überein, aber $\hat{\theta}$ weicht von θ^* ab.

- Klasse 3: Sowohl $\hat{K}$ als auch $\hat{\theta}$ sind identisch zu K^* bzw. θ^*.

Zwei Lösungen $\hat{\theta}$ und θ^* wurden hierbei als identisch angesehen, wenn die totale Flächendifferenz zwischen den Modell- und Glykan-Peaks weniger als 10% der zu modellierenden Signalfläche beträgt.

Zweidrittel der Ergebnisse konnten, wie Tabelle 4.6 zu sehen, der Klasse 1 und 2 zu geordnet werden. Die Verteilung korreliert dabei direkt mit der Komplexität der untersuchten Multi-Analyte-Peaks (Abbildung 4.13).

Tabelle 4.6: Verteilung der 32 Ergebnisse der dritten Testphase.

	Klasse		
	1	2	3
Anzahl	17	3	11

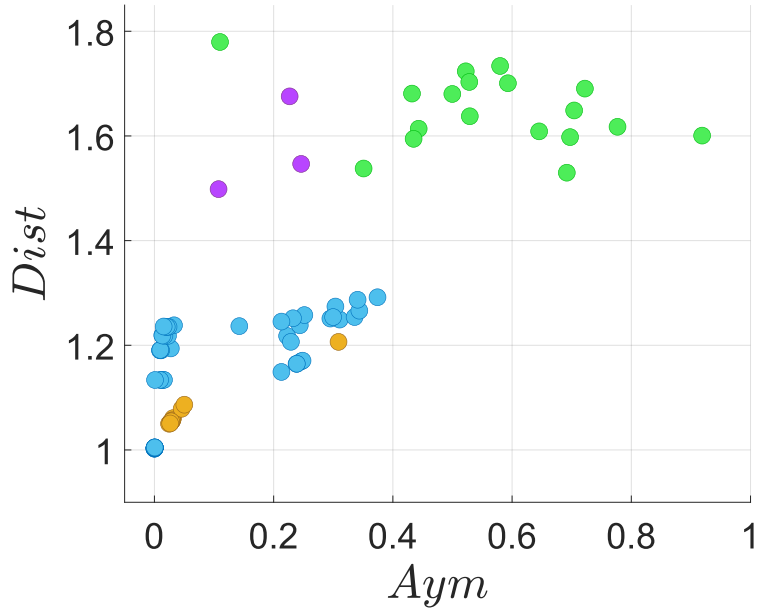


Abbildung 4.13: Die Komplexität der Multi-Analyte-Peaks der Qualitätsklassen 1 (grün), 2 (lila) und 3 (gelb), sowie der künstlichen Messdaten (blau).

Wie in Abbildung 4.14 visualisiert, sind die Multi-Analyte-Peaks der Klasse-1-Daten aufgrund ihrer starken Überlagerung optisch nicht mehr von einer einzelnen Peak-Form zu unterscheiden, weshalb diese auch als ein-komponentige GMMs modelliert wurden. Dies zeigt die Grenzen des Hybrid-Ansatzes auf, welcher ohne zusätzliche Struktur-Informationen bei besonders komplexen Daten nur ein unteres Limit für K finden kann.

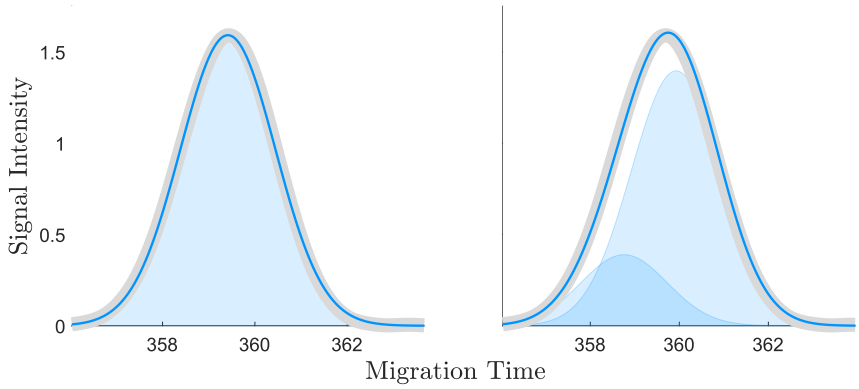


Abbildung 4.14: Beispielhaftes Ergebnis der Klasse-1-Daten des Hybrid-Ansatzes (links, blaue Linie) im Vergleich zur Ideallösung θ^* (rechts, blaue Linie).

Der schlechte Funktionswert der idealisierten Lösung in Abbildung 4.15 macht außerdem deutlich, dass solche Zusatzinformationen, welche von komplexen biologischen Prozessen stammen, nicht fehlerfrei sind und die θ^* nur bessere Approximationen darstellen. Infolgedessen ist eine tiefergehende Analyse der Klasse-2-Daten vonnöten, um abschließend sagen zu können, ob die abweichenden E^R-Werte auf fehlerhaften θ^*-Lösungen basieren oder das Problem doch beim Hybrid-Ansatz zu finden ist.

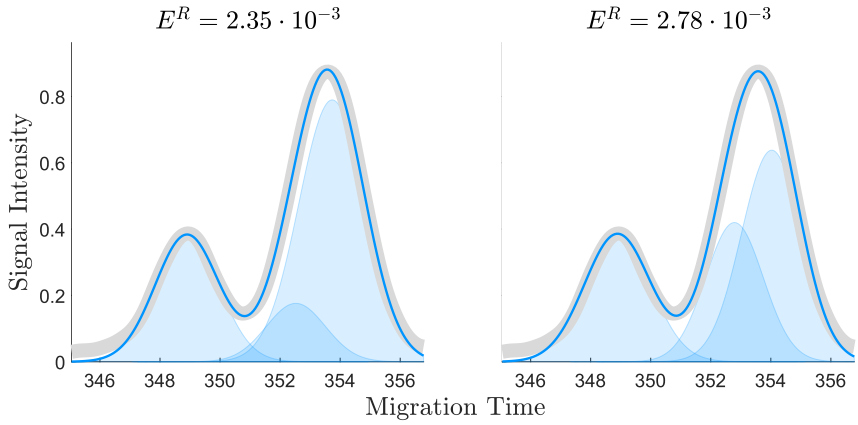


Abbildung 4.15: Beispielhaftes Ergebnis der Klasse-2-Daten des Hybrid-Ansatzes (links, blaue Linie) im Vergleich zur Ideallösung θ^* (rechts, blaue Linie).

Davon abgesehen konnten mit dem Ansatz aber auch Erfolge erzielt werden. So wurden für Multi-Analyte-Peaks, welche eine ähnliche Komplexität wie die künstlichen Messdaten aufwiesen, durchweg sehr gute Lösungen gefunden (Abbildung 4.16).

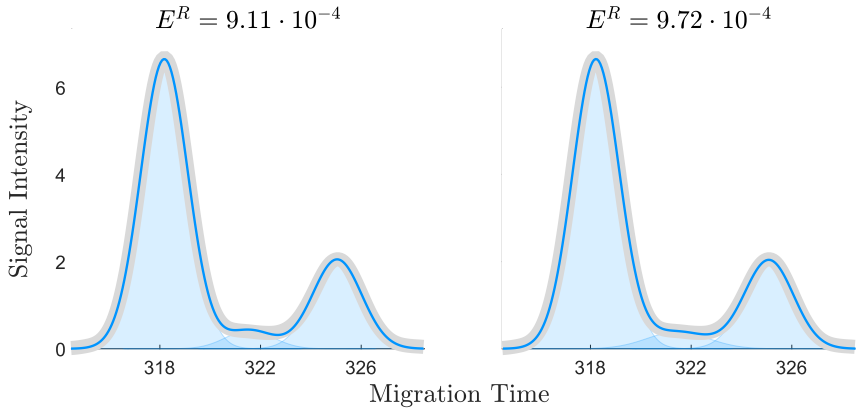


Abbildung 4.16: Beispielhaftes Ergebnis der Klasse-3-Daten des Hybrid-Ansatzes (links, blaue Linie) im Vergleich zur Ideallösung θ^* (rechts, blaue Linie).

4.3.4 Fazit

Abschließend ist festzustellen, dass die CGE-LIF-basierten Messungen häufig bedeutend komplexer als die artifiziell erzeugten Daten waren. Daher war es wenig überraschend, dass die final konstruierte *Siegermethode*, der Hybrid-Ansatz, die ideale Anzahl an Komponenten oft unterschätzte und dadurch $\hat{\theta}$ lieferte, welche von einer Ideallösung θ^* abwichen. Dies zeigt die Grenzen dieser Arbeit auf, in welcher Lösungsansätze nur die σ-GNB als Struktur-Informationen zur Auflösung von Multi-Analyte-Peaks verwenden.

Abbildung 4.17 untermauert allerdings auch das Potenzial des Hybrid-Ansatzes als semi-automatischer Algorithmus. Bereits jetzt können Lösungsvorschläge für viele, schlechte aufgelöste Strukturen berechnet werden, aus denen abschließend das glaubwürdigste GMM durch einen Experten auswählt wird.

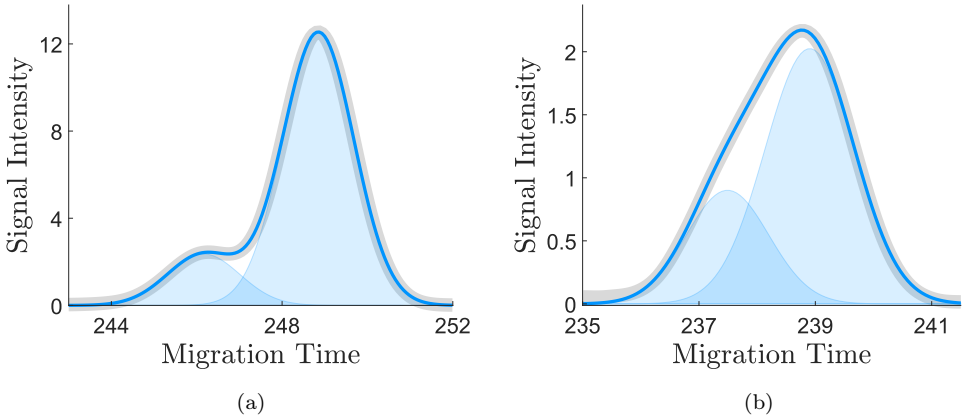


Abbildung 4.17: Ergebnisse des Hybrid-Ansatzes (blaue Linien) für die keine Exoglycosidase-Verdaus vorlagen und für die daher keine Ideallösung θ^* konstruiert werden konnte.

5 Zusammenfassung

In dieser Arbeit wurde eine Toolbox präsentiert, womit Ansätze zur Auflösung von Multi-Analyte-Peaks in CGE-LIF-Messungen konstruiert werden können. Ein solcher Lösungsansatz bildet dabei das Messsignal, als Summe von sich partiell überlagernden Peak-Formen, nach, indem die Parameter eines GMMs, als unbeschränktes oder beschränktes Problem, optimiert werden. Auf diese Weise beschreibt jede Komponente im Idealfall genau einen Peak.

Für den Aufbau eines solchen Lösungsansatzes bot die Toolbox eine Reihe von Konfigurationsmöglichkeiten an, wobei für die Signalmodellierung eine LLH- oder RMSE-basierte Zielfunktion zur Auswahl stand. Darauf aufbauend konnte für die Optimierung zwischen einem EM-Algorithmus, sowie zwei Arten von Partikelschwarmoptimierern, gewählt werden. Diese ließen sich wiederum mit der Top-Down bzw. Tournament Strategie erweitern, zwei speziell für diese Arbeit entwickelte Hyperparameteroptimierer, mit denen die Anzahl an Peaks K in einem Multi-Analyte-Peak abgeschätzt wurde.

Zur Evaluation der Ansätze dieser Arbeit wurden drei aufeinander aufbauende Testphasen durchgeführt, wobei in den beiden ersten Phase künstliche Daten für die grundlegende Analyse verwendet wurden. Auf Basis der dadurch gewonnenen Erkenntnisse wurde anschließend eine *Siegermethode* konstruiert, welche final an echten N-Glykan-Messdaten erprobt wurde.

Der unbeschränkte EM-Algorithmus war, wie in der Literatur beschrieben, die Optimierungsmethode der Wahl, wenn es darum ging die Log-Likelihood-Funktion zu maximieren. Allerdings nahm das Signalrauschen einen sehr viel größeren Einfluss auf die Ergebnisse der LLH-basierten Ansätze, als dies bei den RMSE-Verfahren der Fall war, deren GMMs stärker an den Peak-Formen der künstlichen Messdaten orientiert waren.

Wenn es hingegen darum ging die Komponentenanzahl in den artifiziell erzeugten Multi-Analyte-Peaks abzuschätzen, blieb der EM-basierte Lösungsansatz

in Kombination mit der Tournament Strategie außer Konkurrenz. Die Ergebnisse der anderen Verfahren hingen hierbei maßgeblich von der Komplexität der zu modellierenden Daten ab.

Auf Basis dieser Erkenntnisse wurde die *Siegermethode* wie folgt konstruiert:

1. Approximation von K mit Hilfe des EMs und der Tournament Strategie.
2. Modellierung des Multi-Analyte-Peaks als K-komponentiges GMM, welches mittels RMSE-Verfahren optimiert wird.

Während mit diesem Hybrid-Ansatz sehr gute Ergebnisse für einfache Realdaten erzielt werden konnten, wurde die tatsächliche Komponentenanzahl bei komplexeren N-Glykan-Messungen häufig unterschätzt. Dies zeigt die Grenzen der, mit der Toolbox erzeugbaren, Ansätze auf, welche, ohne weitere Struktur-Informationen, als semi-automatische Algorithmen am ehesten ihr Potenzial entfalten können. So lassen sich bereits jetzt verschiedene Lösungsvorschläge für ein Multi-Analyte-Peak berechnen, aus denen ein Experte dann das beste GMM auswählen kann.

6 Ausblick

Die Auflösung von Multi-Analyte-Peaks in CGE-LIF-Messungen ist grundsätzlich möglich, wie die vorgestellten Ergebnisse belegen. Jedoch kommen die entwickelten Ansätze dieser Arbeit an ihre Grenzen, wenn die Strukturen zu einer einzigen Peak-Form verschmelzen.

Eine Möglichkeit dieser Problematik entgegen zu wirken, wäre der Einsatz von weiteren Struktur-Informationen. So existieren Datenbanken, in welchen die Migrationszeiten der Intensitätsmaxima von N-Glykanen hinterlegt sind, jedoch weichen die Peak-Positionen aufgrund von Messungenauigkeiten häufig von diesen idealisierten μ-Werten ab. Dieser Umstand, in Kombination mit dem Aspekt, dass selbst kurze Signalbereiche oft mit so vielen Einträgen assoziiert sind, macht die direkte Verwendung dieser Werte schwierig. Stattdessen wäre es denkbar, solche Informationen als Straffunktion in ein Modell zu integrieren, um so Glykan- besser von Rausch-Peaks, die nicht in der Datenbank vertreten sind, unterscheiden zu können.

Des Weiteren ließe sich das Konzept der σ-GNB weiterentwickeln. So könnte beispielsweise eine Datenbank angelegt werden, in welcher die Breiten von bekannten Glykan-Strukturen abgespeichert werden. Alternativ wäre es auch vorstellbar, für jedes N-Glykan ein eigenes lineares σ-Modell zu konstruieren, was für die Optimierung immer wieder herangezogen werden kann. Auf diese Weise könnten die strukturellen Eigenschaften einzelner Peak-Formen besser berücksichtigt und modelliert werden.

Darüber hinaus ist das Potenzial der künstlich erzeugten Multi-Analyte-Peaks längst nicht ausgeschöpft. Nicht nur wäre es möglich artifizielle Messsignale zu generieren, welche von ihrer Komplexität stärker an den Realdaten orientierst wären, auch könnte, bei entsprechender Datenbasis, ein Klassifikator entwickelt werden, um Messungen bereits vorab in Problemklassen einzuteilen und damit ihre Lösbarkeit abzuschätzen.

7 Literaturverzeichnis

[1] A. Varki, R. D. Cummings, J. D. Esko, P. Stanley, G. W. Hart, M. Aebi, A. G. Darvill, T. Kinoshita, N. H. Packer, J. H. Prestegard, R. L. Schnaar, and P. H. Seeberger, *Essentials of Glycobiology, 3rd edition.* Cold Spring Harbor Laboratory Press, 2015.

[2] J. F. Rakus and L. K. Mahal, "New Technologies for Glycomic Analysis: Toward a Systematic Understanding of the Glycome," *Annual Review of Analytical Chemistry*, vol. 4, pp. 367–392, 2011.

[3] D. Vanderschaeghe, N. Festjens, J. Delanghe, and N. Callewaert, "Glycome profiling using modern glycomics technology: technical aspects and applications," *Biological Chemistry*, vol. 391, pp. 149–161, 2010.

[4] M. E. Taylor and K. Drickamer, *Introduction to Glycobiology.* Oxford University Press, 2011.

[5] B. Tissot, S. J. North, A. Ceroni, P.-C. Pang, M. Panico, F. Rosati, A. Capone, S. M. Haslam, A. Dell, and H. R. Morris, "Glycoproteomics: past, present and future," *FEBS Letters*, vol. 583, pp. 1728–1735, 2009.

[6] D. C. Harris, *Quantitative Chemical Analysis.* W. H. Freeman and Company, 2010.

[7] J. A. Luckey, T. B. Norris, and L. M. Smith, "Analysis of resolution in DNA sequencing by capillary gel electrophoresis," *Journal of Physical Chemistry*, vol. 97, pp. 3067–3075, 1993.

[8] H. Geyer and R. Geyer, "Strategies for analysis of glycoprotein glycosylation," *Biochimica et Biophysica Acta*, vol. 1764, pp. 1853–1869, 2006.

[9] The Editors of Encyclopaedia Britannica, "Monosaccharide," Dezember 2019. https://www.britannica.com/science/monosaccharide.

[10] A. Varki, R. D. Cummings, J. D. Esko, H. H. Freeze, P. Stanley, J. D. Marth, C. R. Bertozzi, G. W. Hart, and M. E. Etzler, "Symbol nomenclature for glycan representation," *Proteomics*, vol. 9, pp. 5398–5399, 2009.

[11] L. Han and C. E. Costello, "Mass Spectrometry of Glycans," *Biochemistry (Moscow)*, vol. 78, pp. 710–720, 2013.

[12] J. Zaia, "Mass Spectrometry and Glycomics," *OMICS: A Journal of Integrative Biology*, vol. 14, pp. 401–418, 05 2010.

[13] P. J. Domann, A. C. Pardos-Pardos, D. L. Fernandes, D. I. R. Spencer, C. M. Radcliffe, L. Royle, R. A. Dwek, and P. M. Rudd, "Separation-based glycoprofiling approaches using fluorescent labels," *Proteomics*, vol. 7, pp. 70–76, 2007.

[14] L. Royle, M. P. Campbell, C. M. Radcliffe, D. M. White, D. J. Harvey, J. L. Abrahams, Y.-G. Kim, G. W. Henry, N. A. Shadick, M. E. Weinblatt, D. M. Lee, P. M. Rudd, and R. A. Dwek, "HPLC-based analysis of serum N-glycans on a 96-well plate platform with dedicated database software," *Analytical Biochemistry*, vol. 376, pp. 1–12, 2008.

[15] M. Wuhrer, M. I. Catalina, A. M. Deelder, and C. H. Hokke, "Glycoproteomics based on tandem mass spectrometry of glycopeptides," *Journal of Chromatography B*, vol. 849, pp. 115–128, 2007.

[16] W. Laroy, R. Contreras, and N. Callewaert, "Glycome mapping on DNA sequencing equipment," *Nature protocols*, vol. 1, pp. 397–405, 2006.

[17] N. Callewaert, S. Geysens, F. Molemans, and R. Contreras, "Ultrasensitivie profiling and sequencing of N-Linked oligosaccharides using standard DNA-sequencing equipment," *Glycobiology*, vol. 11, pp. 275–81, 05 2001.

[18] F.-T. A. Chen, T. S. Dobashi, and R. A. Evangelista, "Quantitative analysis of sugar constituents of glycoproteins by capillary electrophoresis," *Glycobiology*, vol. 8, pp. 1045–1052, 11 1998.

[19] D. Reusch, M. Haberger, T. Kailich, A.-K. Heidenreich, M. Kampe, P. Bulau, and M. Wuhrer, "High-throughput glycosylation analysis of therapeutic immunoglobulin G by capillary gel electrophoresis using a DNA analyzer," *mAbs*, vol. 6, pp. 185–196, 2014.

[20] L. R. Ruhaak, R. Hennig, C. Huhn, M. Borowiak, R. J. E. M. Dolhain, A. M. Deelder, E. Rapp, and M. Wuhrer, "Optimized Workflow for Preparation of APTS-Labeled N-Glycans Allowing High-Throughput Analysis of Human Plasma Glycomes using 48-Channel Multiplexed CGE-LIF," *Journal of Proteome Research*, vol. 9, pp. 6655–6664, 2010.

[21] J. Schwarzer, E. Rapp, and U. Reichl, "N-glycan analysis by CGE-LIF: Profiling influenza A virus hemagglutinin N-glycosylation during vaccine production," *Electrophoresis*, vol. 29, pp. 4203–4214, 2008.

[22] M. Wuhrer, A. M. Deelder, and C. H. Hokke, "Protein glycosylation analysis by liquid chromatography–mass spectrometry," *Journal of Chromatography B*, vol. 825, pp. 124–133, 2005.

[23] B. C. Jansen, K. Reiding, A. Bondt, A. Ederveen, M. Palmblad, D. Falck, and M. Wuhrer, "MassyTools: A High-Throughput Targeted Data Processing Tool for Relative Quantitation and Quality Control Developed for Glycomic and Glycoproteomic MALDI-MS," *Journal of Proteome Research*, vol. 14, pp. 5088–5098, 2015.

[24] B. C. Jansen, D. Falck, N. de Haan, A. L. Hipgrave Ederveen, G. Razdorov, G. Lauc, and M. Wuhrer, "LaCyTools: A Targeted Liquid Chromatography-Mass Spectrometry Data Processing Package for Relative Quantitation of Glycopeptides," *Journal of Proteome Research*, vol. 15, pp. 2198–2210, 2016.

[25] B. C. Jansen, L. Hafkenscheid, A. Bondt, R. Gardner, J. Hendel, M. Wuhrer, and D. Spencer, "HappyTools: A software for high-throughput HPLC data processing and quantitation," *PLoS One*, vol. 13, p. e0200280, 2018.

[26] C.-Y. Yu, A. Mayampurath, Y. Hu, S. Zhou, Y. Mechref, and H. Tang, "Automated annotation and quantification of glycans using liquid chromatography-mass spectrometry," *Bioinformatics*, vol. 29, pp. 1706–1707, July 2013.

[27] Y. Hu, S. Zhou, C.-Y. Yu, H. Tang, and Y. Mechref, "Automated annotation and quantitation of glycans by liquid chromatography/electrospray ionization mass spectrometric analysis using the MultiGlycan-ESI computational tool," *Rapid Communications in Mass Spectrometry*, vol. 29, pp. 135–142, 2015.

[28] E. Maxwell, Y. Tan, Y. Tan, H. Hu, G. Benson, K. Aizikov, S. Conley, G. O. Staples, G. W. Slysz, R. D. Smith, and J. Zaia, "GlycReSoft: a software package for automated recognition of glycans from LC/MS data," *PLoS One*, vol. 7, p. e45474, 2012.

[29] V. Di Marco and G. Bombi, "Mathematical functions for the representation of chromatographic peaks," *Journal of Chromatography A*, vol. 931, pp. 1–30, 2001.

[30] B. Durney, C. Crihfield, and L. Holland, "Capillary electrophoresis applied to DNA: determining and harnessing sequence and structure to advance bioanalyses (2009–2014)," *Analytical and Bioanalytical Chemistry*, vol. 407, pp. 6923–6938, 2015.

[31] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2007.

[32] E. Grushka, "Characterization of exponentially modified Gaussian peaks in chromatography," *Analytical Chemistry*, vol. 44, pp. 1733–1738, 1972.

[33] M. S. Jeansonne and J. P. Foley, "Review of the Exponentially Modified Gaussian (EMG) Function Since 1983," *Journal of Chromatographic Science*, vol. 29, pp. 258–266, 1991.

[34] H. Kong, F. Ye, X. Lu, L. Guo, J. Tian, and G. Xu, "Deconvolution of overlapped peaks based on the exponentially modified Gaussian model in comprehensive two-dimensional gas chromatography," *Journal of Chromatography A*, vol. 1086, pp. 160–164, 2005.

[35] J. Baeza-Baeza and M. García-Álvarez Coque, "Prediction of peak shape as a function of retention in reversed-phase liquid chromatography," *Journal of Chromatography A*, vol. 1022, pp. 17–24, 2004.

[36] E. Benická, J. Krupcík, J. Lehotay, P. Sandra, and D. W. Armstrong, "Selectivity Tuning in an HPLC Multicomponent Separation," *Journal of Liquid Chromatography & Related Technologies*, vol. 28, pp. 1453–1471, 2005.

[37] L. Komsta, Y. V. Heyden, and J. Sherma, *Chemometrics in Chromatography*. CRC Press, 2018.

[38] Y. Shen and M. L. Lee, "General Equation for Peak Capacity in Column Chromatography," *Analytical Chemistry*, vol. 70, pp. 3853–3856, 1998.

[39] S. Mostaghim, W. Halter, and A. Wille, "Linear Multi-Objective Particle Swarm Optimization," *Stigmergic Optimization*, vol. 31, pp. 209–238, 2006.

[40] M. E. H. Pedersen, "Good Parameters for Particle Swarm Optimization," Tech. Rep. HL1001, Hvass Laboratories, 2010.

[41] E. Mezura-Montes and C. A. C. Coello, "Constraint-handling in nature-inspired numerical optimization: Past, present and future," *Swarm and Evolutionary Computation*, vol. 1, pp. 173–194, 2011.

[42] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, pp. 1–38, 1977.

[43] M. Dijkstra, H. Roelofsen, R. J. Vonk, and R. C. Jansen, "Peak quantification in surface-enhanced laser desorption/ionization by using mixture models," *Proteomics*, vol. 6, pp. 5106–5116, 2006.

[44] A. Polanski, M. Marczyk, M. Pietrowska, P. Widlak, and J. Polanska, "Signal Partitioning Algorithm for Highly Efficient Gaussian Mixture Modeling in Mass Spectrometry," *PLoS One*, vol. 10, p. e0134256, 2015.

[45] T. Yu and H. Peng, "Quantification and deconvolution of asymmetric LC-MS peaks using the bi-Gaussian mixture model and statistical model selection," *BMC Bioinformatics*, vol. 11, p. 559, 2010.

[46] S. Arase, K. Horie, T. Kato, A. Noda, Y. Mito, M. Takahashi, and T. Yanagisawa, "Intelligent peak deconvolution through in-depth study of the data matrix from liquid chromatography coupled with a photo-diode array detector applied to pharmaceutical analysis," *Journal of Chromatography A*, vol. 1469, pp. 35–47, 2016.

[47] M. R. Gupta and Y. Chen, "Theory and Use of the EM Algorithm," *Foundations and Trends in Signal Processing*, vol. 4, pp. 223–296, 2011.

[48] J. Kennedy and R. Eberhart, "Particle swarm optimization," vol. 4, pp. 1942–1948, 1995.

[49] M. E. H. Pedersen, *Tuning & Simplifying Heuristical Optimization*. PhD thesis, School of Engineering Sciences, 01 2010.

[50] Y.-D. Zhang, S. Wang, and G. ji, "A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications," *Mathematical Problems in Engineering*, vol. 2015, pp. 1–38, 2015.

[51] Y. Li, "Dynamic Particle Swarm Optimization Algorithm for Resolution of Overlapping Chromatograms," vol. 3, pp. 246–250, 2009.

[52] H. Parastar, H. Ebrahimi, and M. Jalali-Heravi, "Multivariate curve resolution-particle swarm optimization: A high-throughput approach to exploit pure information from multi-component hyphenated chromatographic signals," *Analytica Chimica Acta*, vol. 772, pp. 16–25, 2013.

[53] U. Paquet and A. P. Engelbrecht, "Particle Swarms for Linearly Constrained Optimisation," *Fundamenta Informaticae*, vol. 76, pp. 147–170, 2007.

[54] M. Zambrano-Bigiarini, M. Clerc, and R. Rojas, "Standard Particle Swarm Optimisation 2011 at CEC-2013: A baseline for future PSO improvements," in *Proceedings of the 2013 IEEE Congress on Evolutionary Computation*, pp. 2337–2344, 2013.

[55] M. Clerc, "Standard Particle Swarm Optimisation," Tech. Rep. hal-00764996, HAL, 2012. https://hal.archives-ouvertes.fr/hal-00764996 [Online: März 2020].

[56] W. Spears, D. Green, and D. Spears, "Biases in Particle Swarm Optimization," *International Journal of Swarm Intelligence Research*, vol. 1, pp. 34–57, Jan. 2010.

[57] D. K. Kim and J. M. G. Taylor, "The Restricted EM Algorithm for Maximum Likelihood Estimation under Linear Restrictions on the Parameters," *Journal of the American Statistical Association*, vol. 90, pp. 708–716, 1995.

[58] M. Clerc, *Particle Swarm Optimization.* ISTE (International Scientific and Technical Encyclopedia), 2005.

[59] M. Liu, D. Shin, and H. I. Kang, "Parameter Estimation in Dynamic Biochemical Systems Based on Adaptive Particle Swarm Optimization," in *Proceedings of the 2009 International Conference on Information, Communications and Signal Processing*, pp. 1–5, 2009.

[60] T. Wenzl, J. Haedrich, A. Schaechtele, P. Robouch, and J. Stroka, *Guidance Document on the Estimation of LOD and LOQ for Measurements in the Field of Contaminants in Feed and Food.* Publications Office of the European Union, 2016.

[61] S. Ullsten, R. Danielsson, D. Backstrom, P. Sjoberg, and J. Bergquist, "Urine profiling using capillary electrophoresis-mass spectrometry and multivariate data analysis," *Journal of Chromatography A*, vol. 1117, pp. 87–93, 2006.

[62] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing

evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, pp. 3–18, 2011.

[63] Omran, M. G. H. and Clerc, M., "Particle Swarm Central," April 2020. http://www.particleswarm.info [Online. Last accessed 2020].

Anhang A – Methodenvergleich

Um den SPSO-2011 auf Fehler zu überprüfen, wurde dessen Performance gegen die Leistung der Implementierung von [63] verglichen. Dazu wurden die beiden Methoden, wie in der Arbeit von Derrac *et* al. beschrieben, für eine Reihe von Benchmark-Funktionen evaluiert und die Ergebnisse mit einem zweiseitigen Wilcoxon-Vorzeichen-Rang-Test gegeneinander verglichen [62].

Für die Nullhypothese H_0 wurde angenommen, dass die paarweise betrachteten Fehler-Mediane, welche in Tabelle A2 aufgelistet sind, von der gleichen Grundgesamtheit stammen. Die zwei PSO-Ansätze wurden jeweils mit einer Schwarmgröße von $M = 100$ initialisiert und für $2.5 \cdot 10^5$ Funktionsaufrufe ausgeführt.

Wie Tabelle A1 zu entnehmen ist, konnte H_0 bei einem Signifikanzniveau von $\alpha = 0.05$ nicht verworfen werden, d.h. es konnte kein signifikanter Unterschied zwischen der Performance des selbgeschriebenen SPSO-2011 und seiner Referenz festgestellt werden.

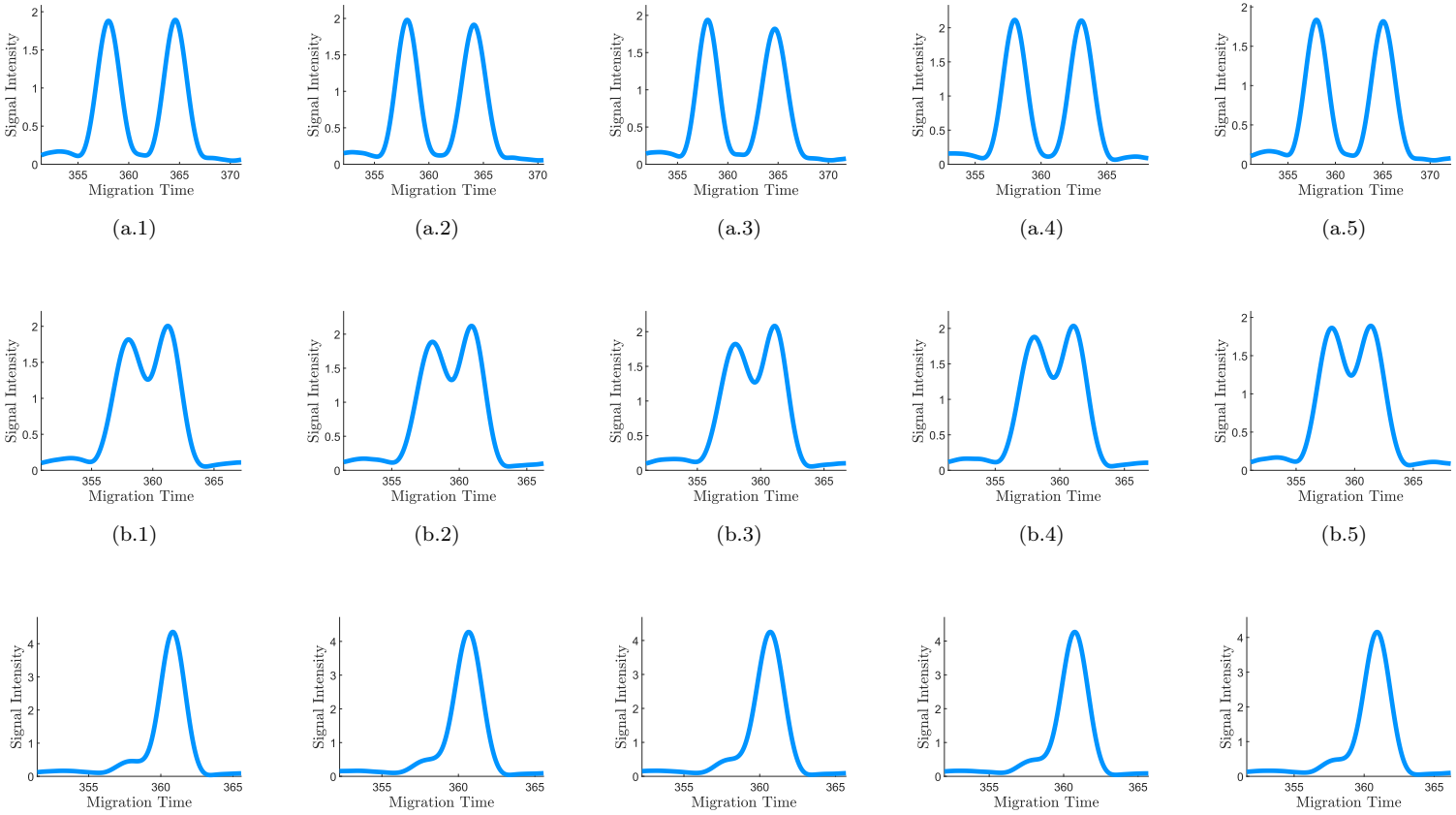
Tabelle A1: Die R^+-Werte des Wilcoxon-Vorzeichen-Rang-Tests mit p als Indikator, ob sich die beiden Methoden signifikant voneinander unterscheiden ($p < 0.05$, schwarze Schriftfarbe) oder nicht ($p \geq 0.05$, rote Schriftfarbe)

VS.	SPSO-2011	Ref. SPSO-2011
SPSO-2011		835 ($p > 0.5$)
Ref. SPSO-2011	867 ($p > 0.5$)	

Tabelle A2: Die Median-Werte der 43 Benchmark-Funktionen.

Funktion	Grenzen	Dimension	SPSO-2011	Ref. SPSO-2011
Ackley	$(-32.7680, 32.7680)$	30	1.04	$4.44 \cdot 10^{-15}$
Beale	$(-4.50, 4.50)$	5	0	0
Bohachevsky 1	$(-100, 100)$	2	0	0
Bohachevsky 2	$(-100, 100)$	2	0	0
Bohachevsky 3	$(-100, 100)$	2	0	0
Booth	$(-10, 10)$	2	0	0
Branin	$(-5, 10) \times (0, 15)$	2	$3.58 \cdot 10^{-7}$	$3.58 \cdot 10^{-7}$
Colville	$(-10, 10)$	4	0	0
Dixon-Price	$(-10, 10)$	30	0.67	0.67
Easom	$(-100, 100)$	2	0	0
Foxholes	$(-65.5360, 65.5360)$	2	$2.22 \cdot 10^{-16}$	$2.22 \cdot 10^{-16}$
Goldstein Price	$(-2, 2)$	2	0	0
Griewank	$(-600, 600)$	30	$1.61 \cdot 10^{-15}$	0
Hartman 3	$(0, 1)$	3	0	0
Hartman 6	$(0, 1)$	6	0.12	0
Kowalik	$(-5, 5)$	4	0	$2.69 \cdot 10^{-4}$
Langerman	$(0, 10)$	2	0	0
Matyas	$(-10, 10)$	2	0	0
Michalewicz 10	$(0, \pi)$	10	0.87	0.77
Michalewicz 2	$(0, \pi)$	2	0	0
Michalewicz 5	$(0, \pi)$	5	$4.18 \cdot 10^{-2}$	0
Perm	$(-D, D)$	4	$2.25 \cdot 10^{-3}$	$1.49 \cdot 10^{-3}$
Powell	$(-4, 5)$	24	$2.13 \cdot 10^{-3}$	$1.07 \cdot 10^{-2}$
PowerSum	$(0, D)$	4	$1.53 \cdot 10^{-4}$	$1.6 \cdot 10^{-4}$
Quartic	$(-1.28, 1.28)$	30	0.65	0.54
Rastrigin	$(-5.12, 5.12)$	30	28.36	42.78
Rosenbrock	$(-30, 30)$	30	19.97	25.78
Schaffer	$(-100, 100)$	2	0	0
Schwefel 1.2	$(-10, 10)$	30	$7.02 \cdot 10^{-13}$	$2.21 \cdot 10^{-5}$
Schwefel 2.22	$(-10, 10)$	30	$2.57 \cdot 10^{-16}$	$8.39 \cdot 10^{-3}$
Shekel 10	$(0, 10)$	4	$4.58 \cdot 10^{-2}$	$4.58 \cdot 10^{-2}$
Shekel 5	$(0, 10)$	4	$4.58 \cdot 10^{-2}$	$4.58 \cdot 10^{-2}$
Shekel 7	$(0, 10)$	4	$4.58 \cdot 10^{-2}$	$4.58 \cdot 10^{-2}$
Shubert	$(-10, 10)$	2	0	0
Six Hump Camel Back	$(-3, 3)$	2	0	0
Sphere	$(-100, 100)$	30	0	0
Step	$(-100, 100)$	30	6.5	0
Stepint	$(-5.12, 5.12)$	5	1	0
SumSquares	$(-10, 10)$	30	0	$1.14 \cdot 10^{-13}$
Three Hump Camel Back	$(-5, 5)$	2	0	0
Trid10	$(-D^2, D^2)$	10	$1.64 \cdot 10^{-11}$	0
Trid6	$(-D^2, D^2)$	6	0	0
Zakharov	$(-5, 10)$	10	0	0

Anhang B – Künstliche Messdaten

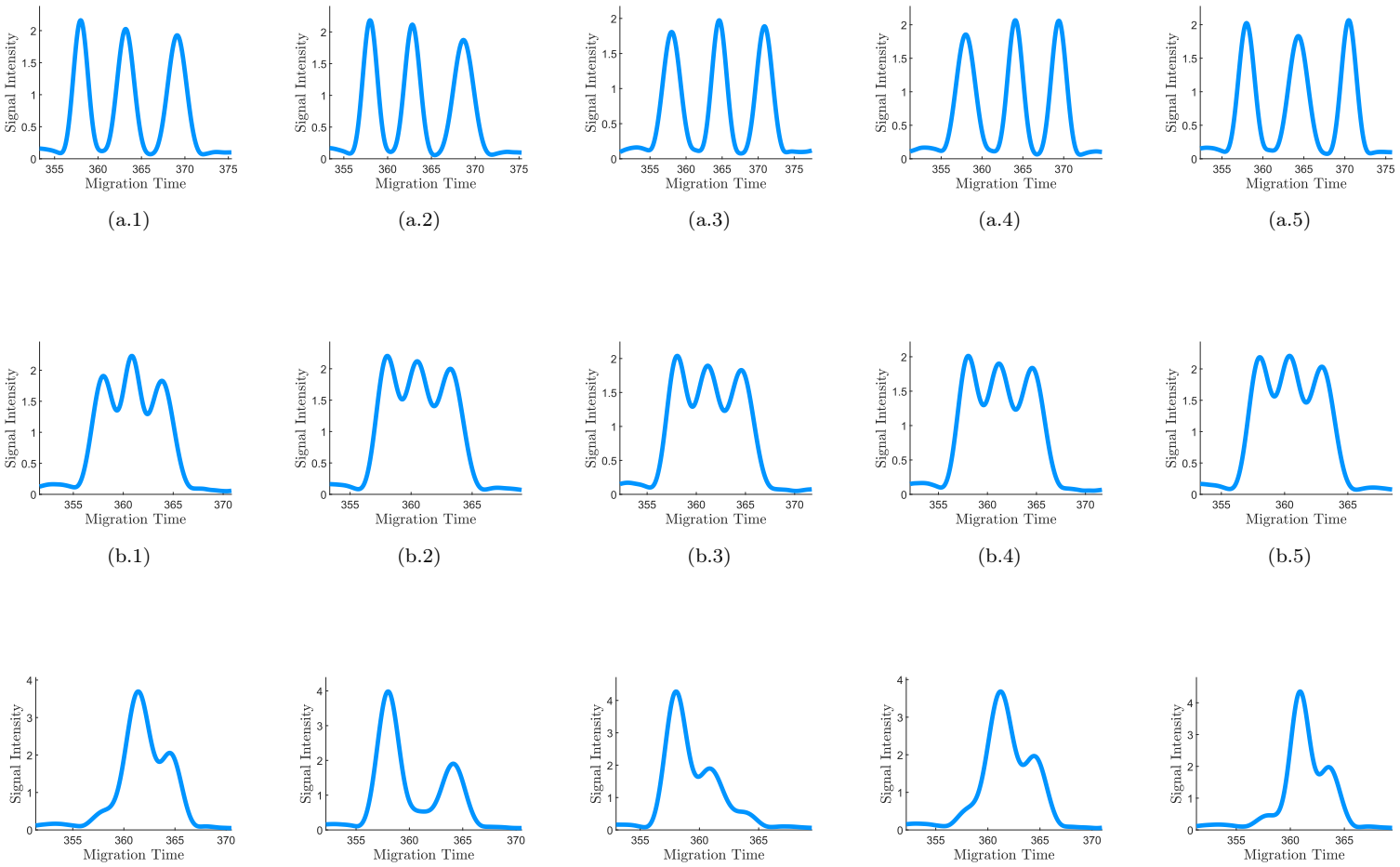


(a.1) (a.2) (a.3) (a.4) (a.5)

(b.1) (b.2) (b.3) (b.4) (b.5)

(c.1) (c.2) (c.3) (c.4) (c.5)

Abbildung B1: Die künstlichen Messdaten (1, 2, 3, 4, 5) der Schwierigkeitslevel L_I, L_{II} und L_{III} (a, b, c) für $K = 2$.

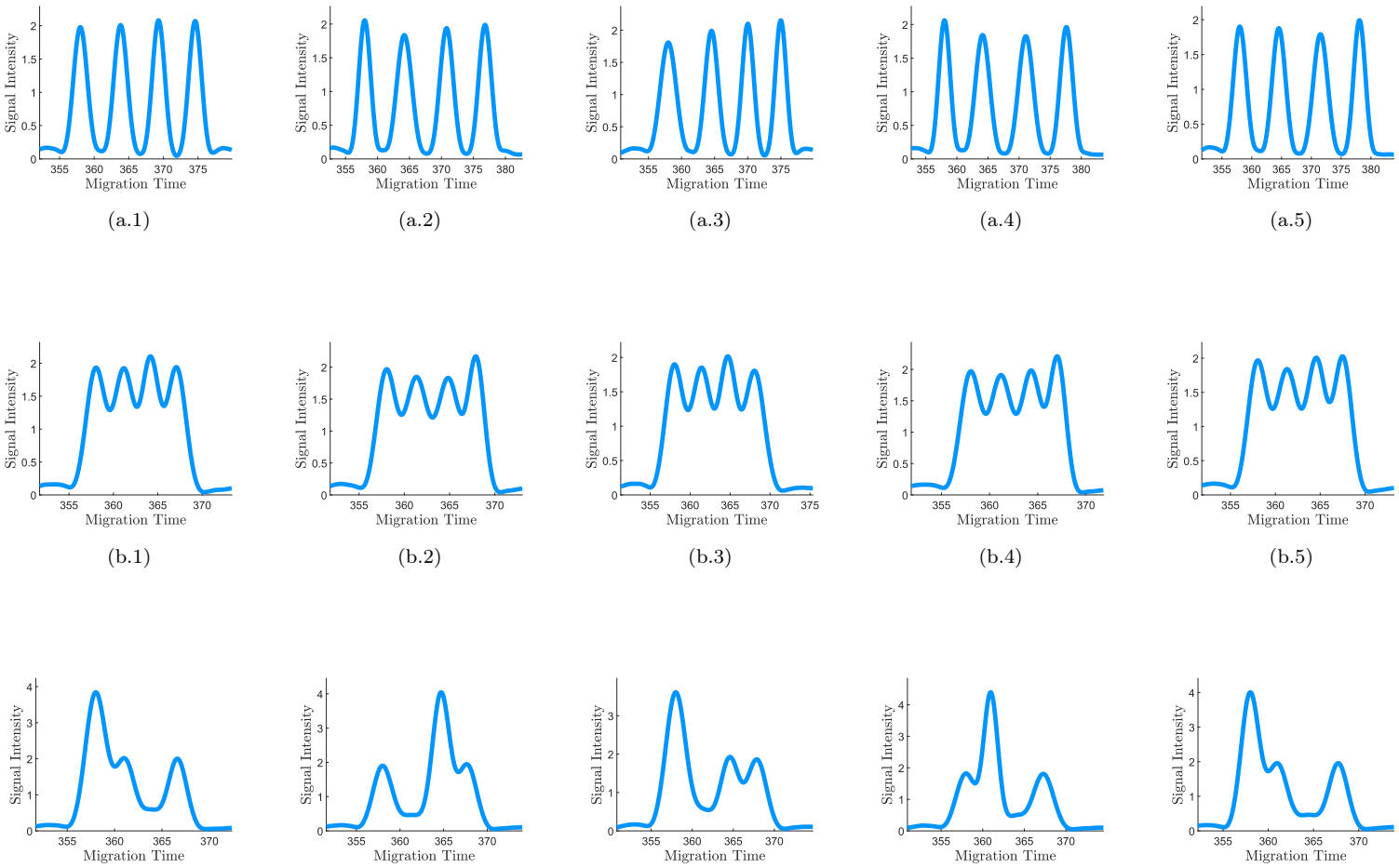


(a.1) (a.2) (a.3) (a.4) (a.5)

(b.1) (b.2) (b.3) (b.4) (b.5)

(c.1) (c.2) (c.3) (c.4) (c.5)

Abbildung B2: Die künstlichen Messdaten (1, 2, 3, 4, 5) der Schwierigkeitslevel L_I, L_{II} und L_{III} (a, b, c) für $K = 3$.



(a.1) (a.2) (a.3) (a.4) (a.5)

(b.1) (b.2) (b.3) (b.4) (b.5)

(c.1) (c.2) (c.3) (c.4) (c.5)

Abbildung B3: Die künstlichen Messdaten (1, 2, 3, 4, 5) der Schwierigkeitslevel L_I, L_{II} und L_{III} (a, b, c) für $K = 4$.

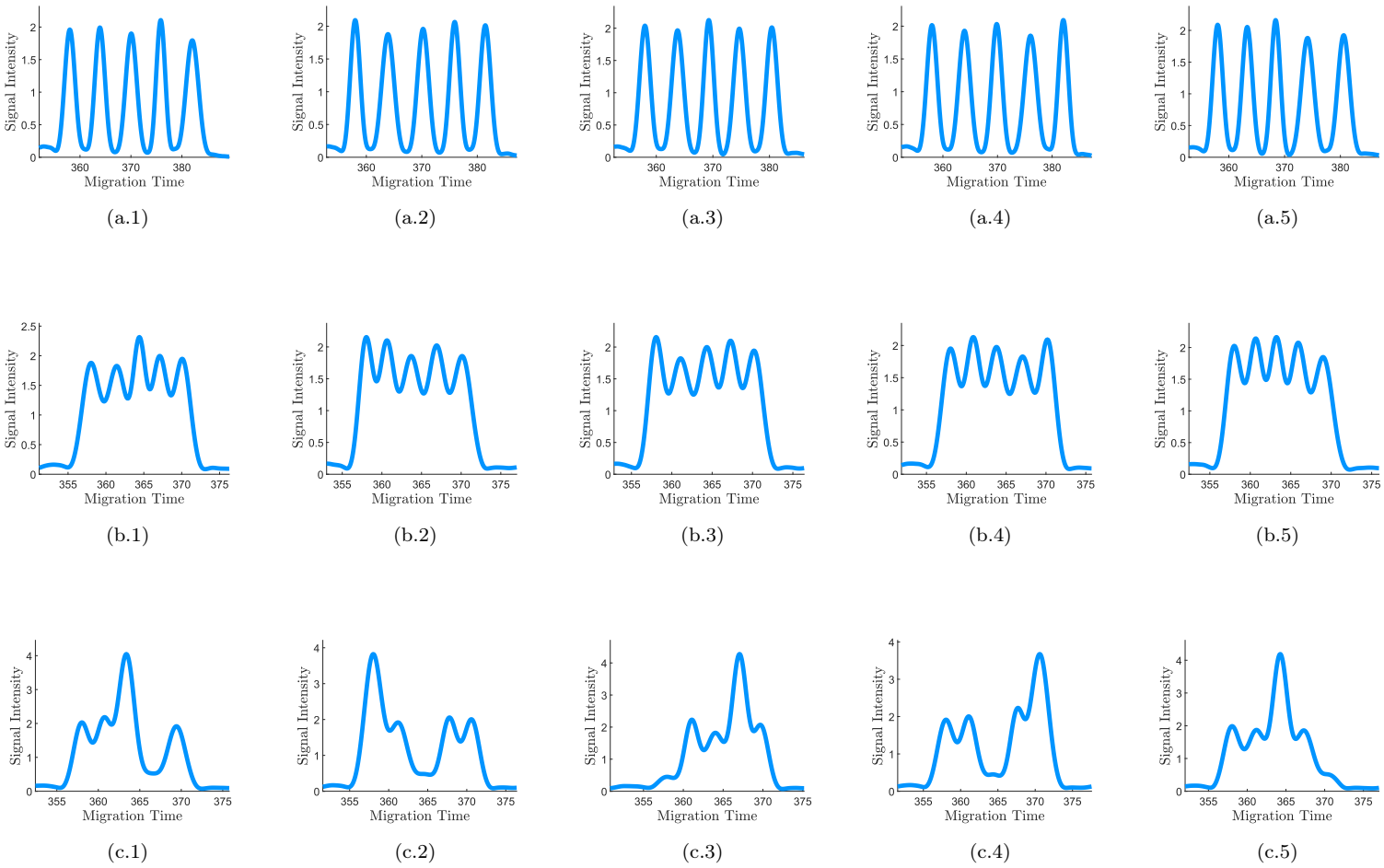


Abbildung B4: Die künstlichen Messdaten (1, 2, 3, 4, 5) der Schwierigkeitslevel L_I, L_{II} und L_{III} (a, b, c) für $K = 5$.

Anhang C – Realdaten

Glykoprotein-Typ I

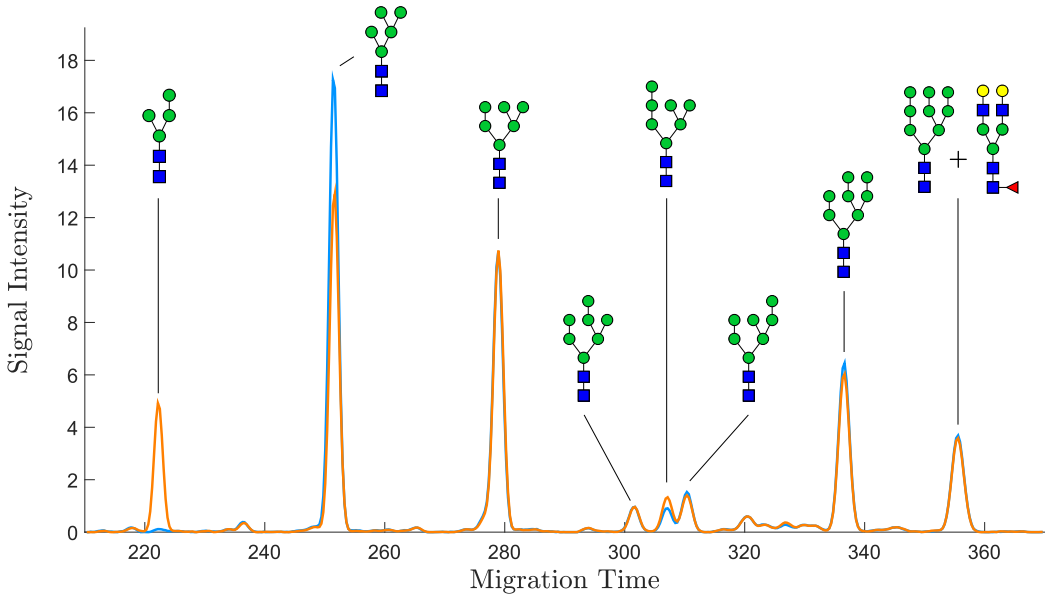


Abbildung C1: Annotierte N-Glykan-Profile vor (blau) und nach einem 1-minütigen Mannosidase-Verdau (orange).

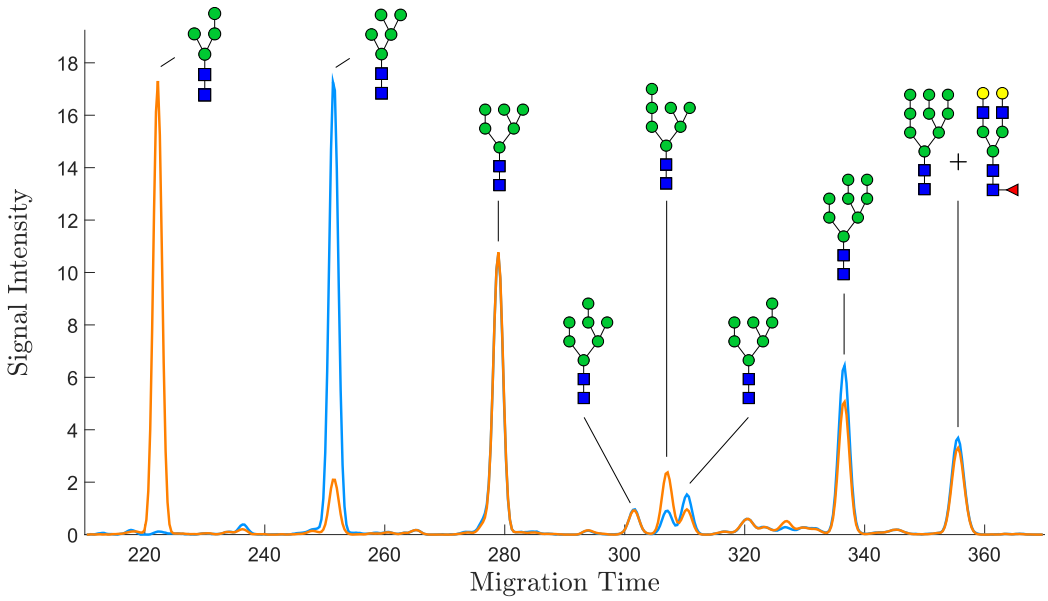


Abbildung C2: Annotierte N-Glykan-Profile vor (blau) und nach einem 10-minütigen Mannosidase-Verdau (orange).

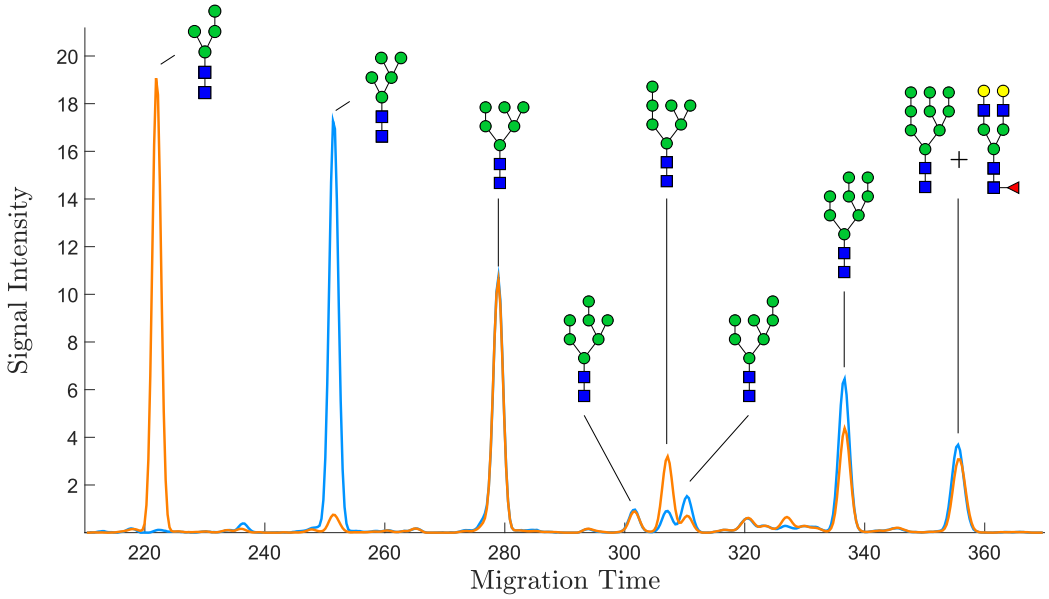


Abbildung C3: Annotierte N-Glykan-Profile vor (blau) und nach einem 20-minütigen Mannosidase-Verdau (orange).

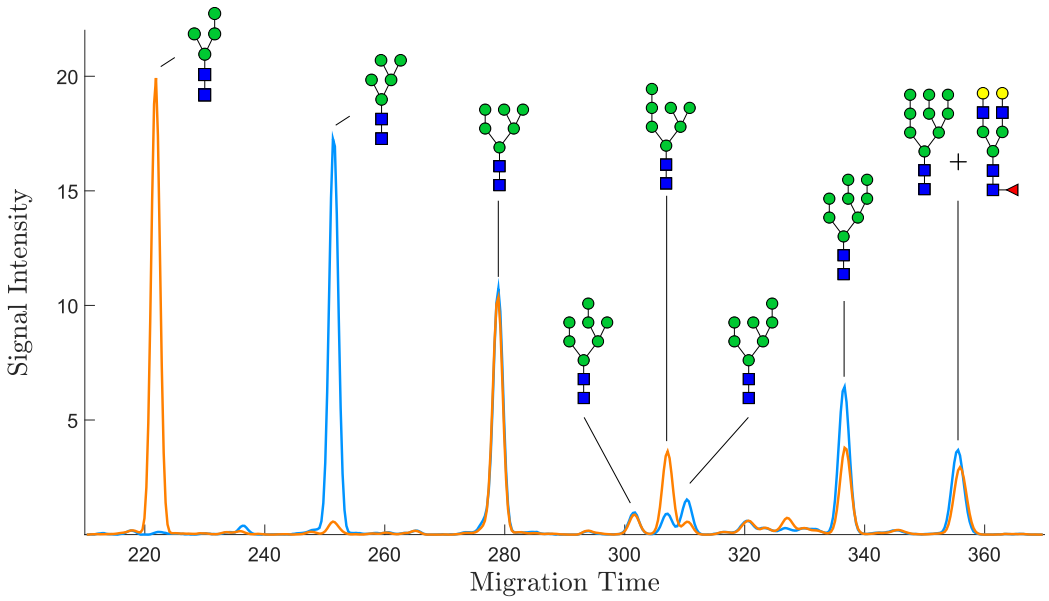


Abbildung C4: Annotierte N-Glykan-Profile vor (blau) und nach einem 30-minütigen Mannosidase-Verdau (orange).

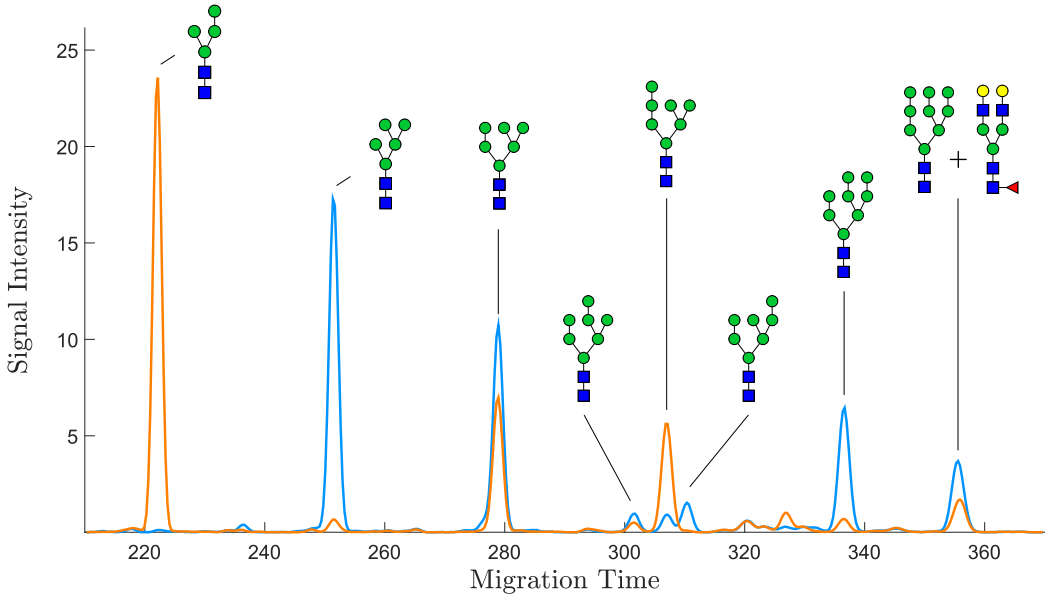


Abbildung C5: Annotierte N-Glykan-Profile vor (blau) und nach einem 200-minütigen Mannosidase-Verdau (orange).

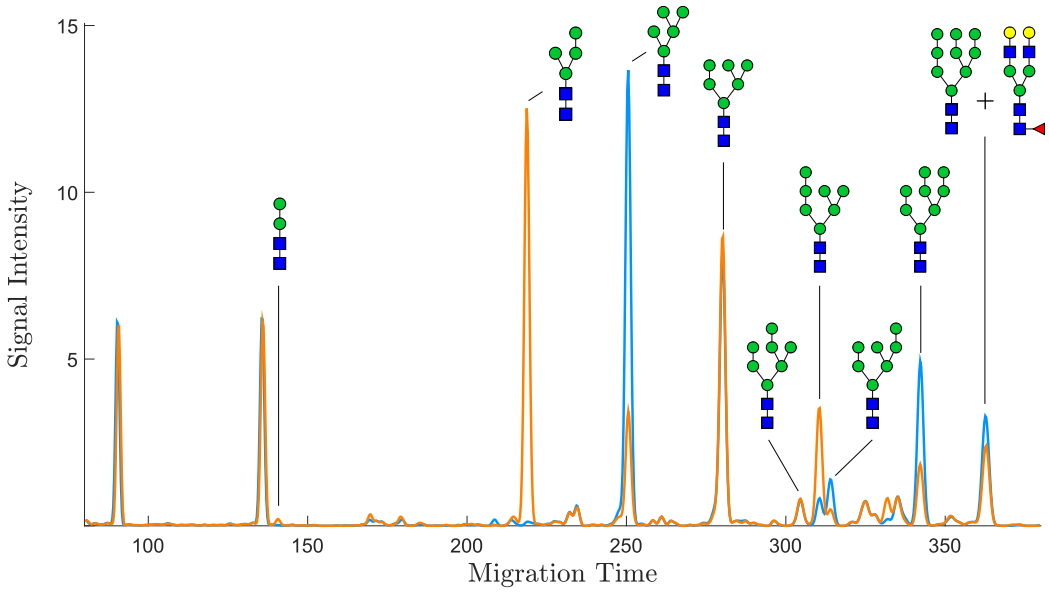


Abbildung C6: Annotierte N-Glykan-Profile vor (blau) und nach einem 1-minütigen Mannosidase-Verdau (orange).

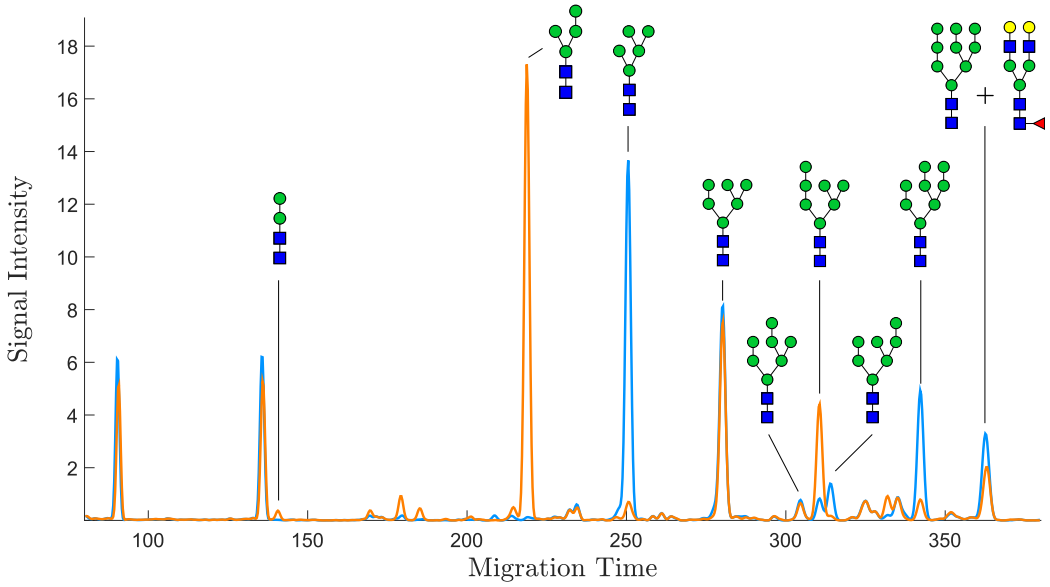


Abbildung C7: Annotierte N-Glykan-Profile vor (blau) und nach einem 10-minütigen Mannosidase-Verdau (orange).

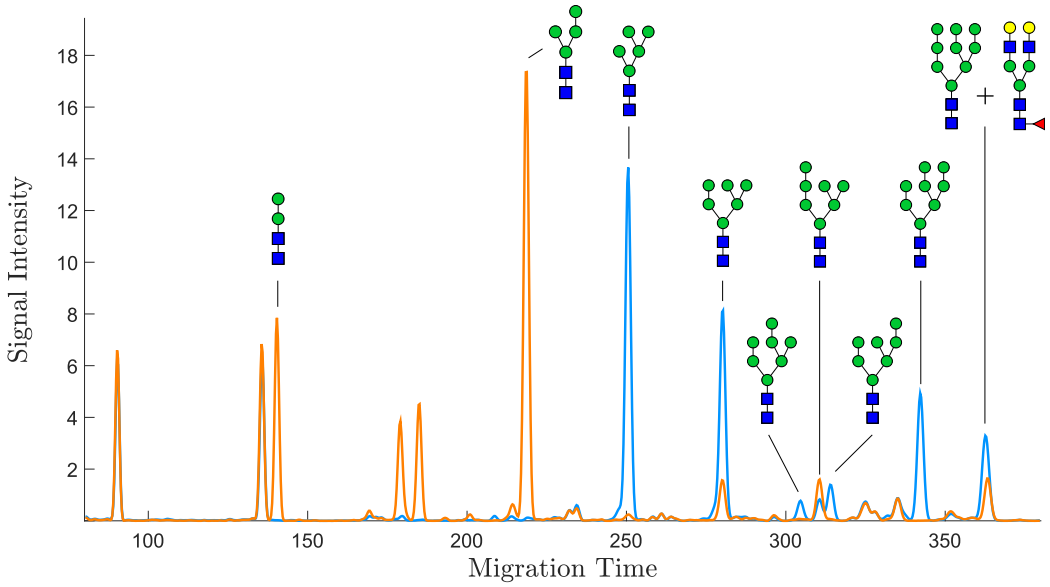


Abbildung C8: Annotierte N-Glykan-Profile vor (blau) und nach einem 2-stündigen Mannosidase-Verdau (orange).

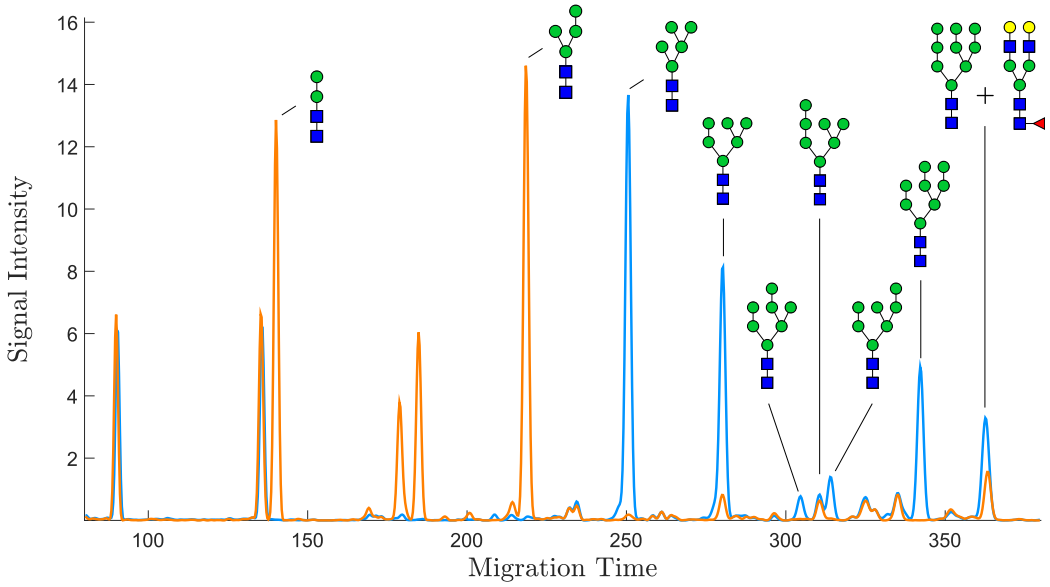


Abbildung C9: Annotierte N-Glykan-Profile vor (blau) und nach einem 3-stündigen Mannosidase-Verdau (orange).

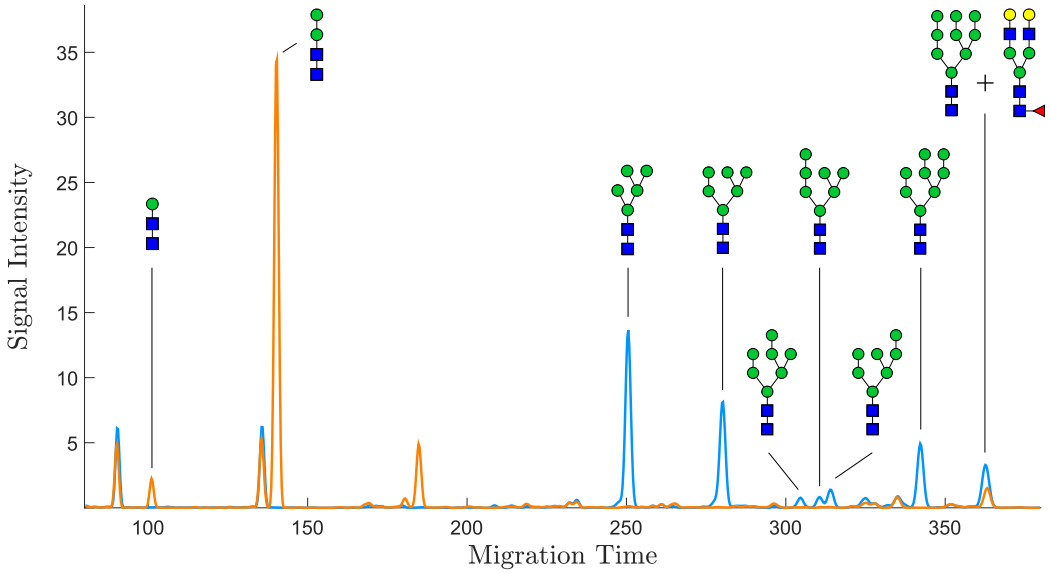


Abbildung C10: Annotierte N-Glykan-Profile vor (blau) und nach einem 15-stündigen Mannosidase-Verdau (orange).

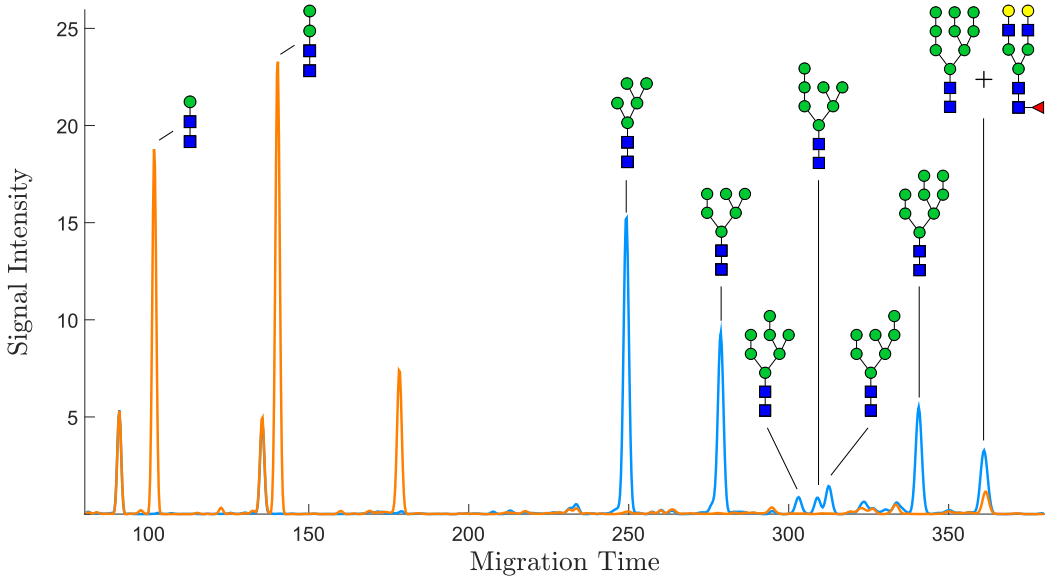


Abbildung C11: Annotierte N-Glykan-Profile vor (blau) und nach einem 200-minütigen Mannosidase-Verdau (orange).

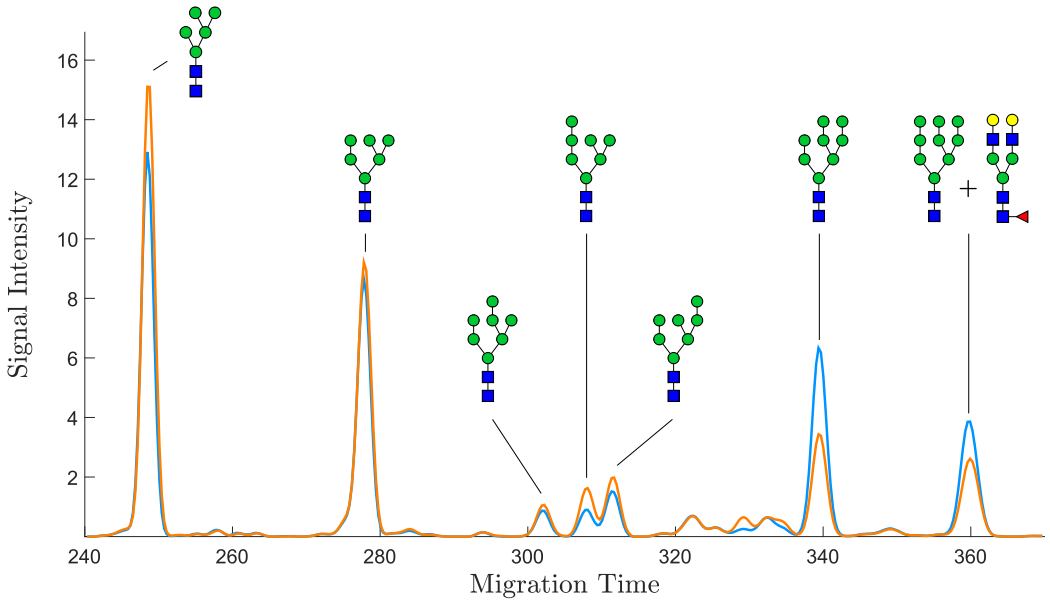


Abbildung C12: Annotierte N-Glykan-Profile vor (blau) und nach einem 10-minütigen Mannosidase-Verdau (orange).

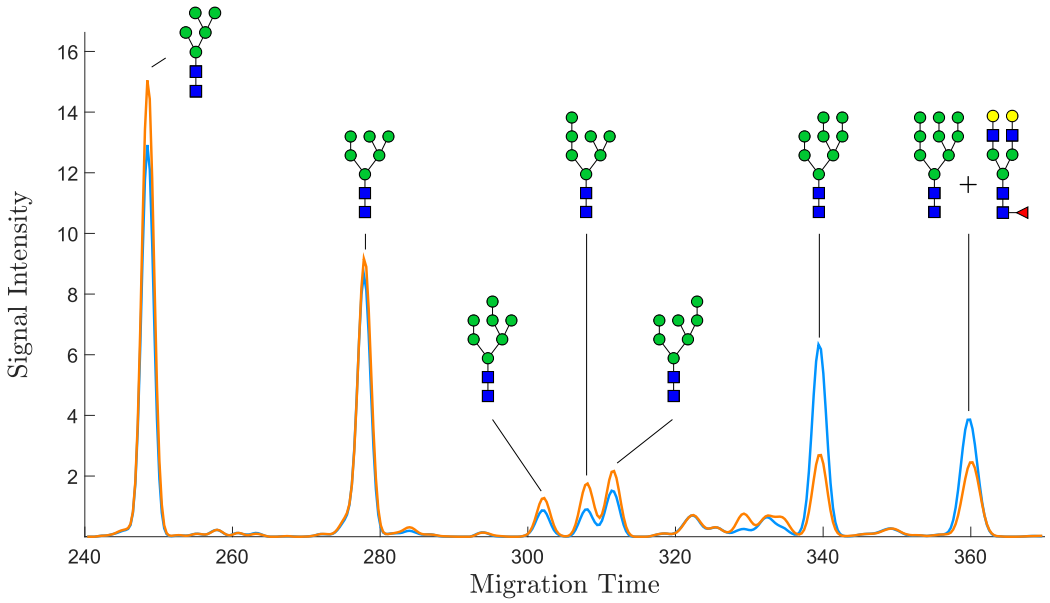


Abbildung C13: Annotierte N-Glykan-Profile vor (blau) und nach einem 1-stündigen Mannosidase-Verdau (orange).

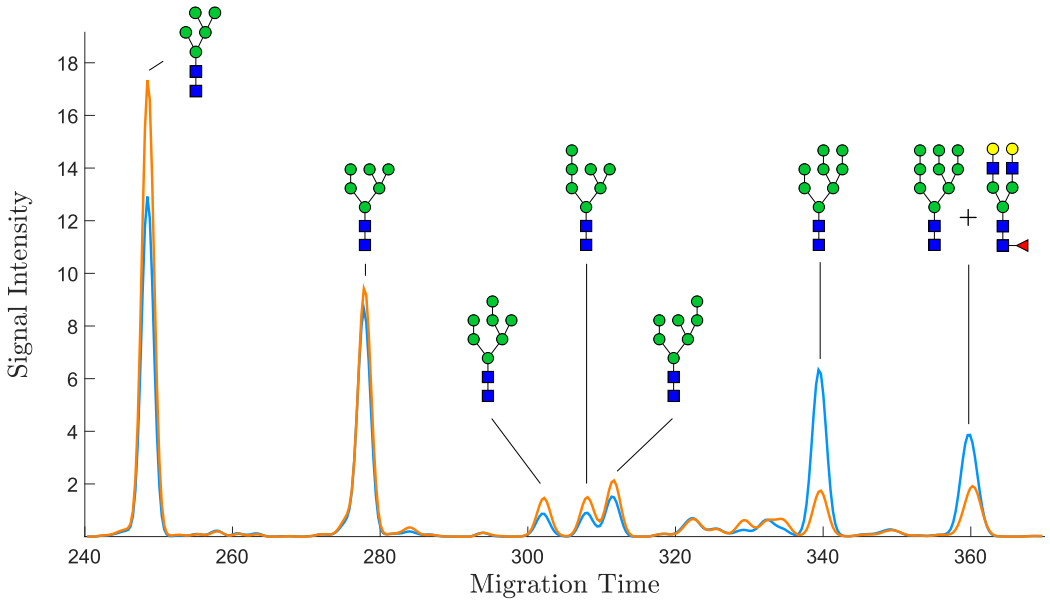


Abbildung C14: Annotierte N-Glykan-Profile vor (blau) und nach einem 3-stündigen Mannosidase-Verdau (orange).

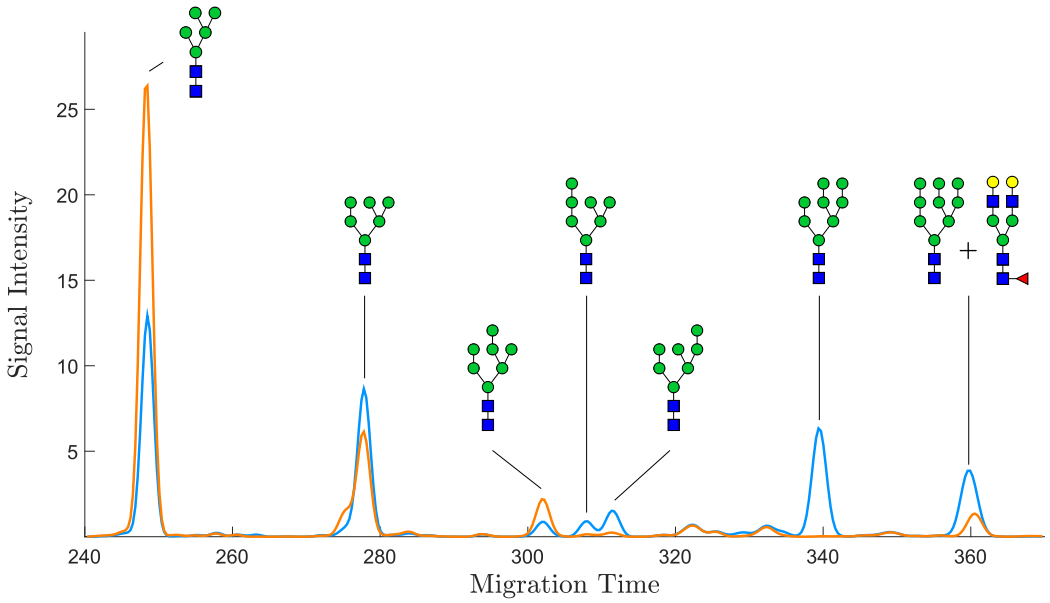


Abbildung C15: Annotierte N-Glykan-Profile vor (blau) und nach einem 23-stündigen Mannosidase-Verdau (orange).

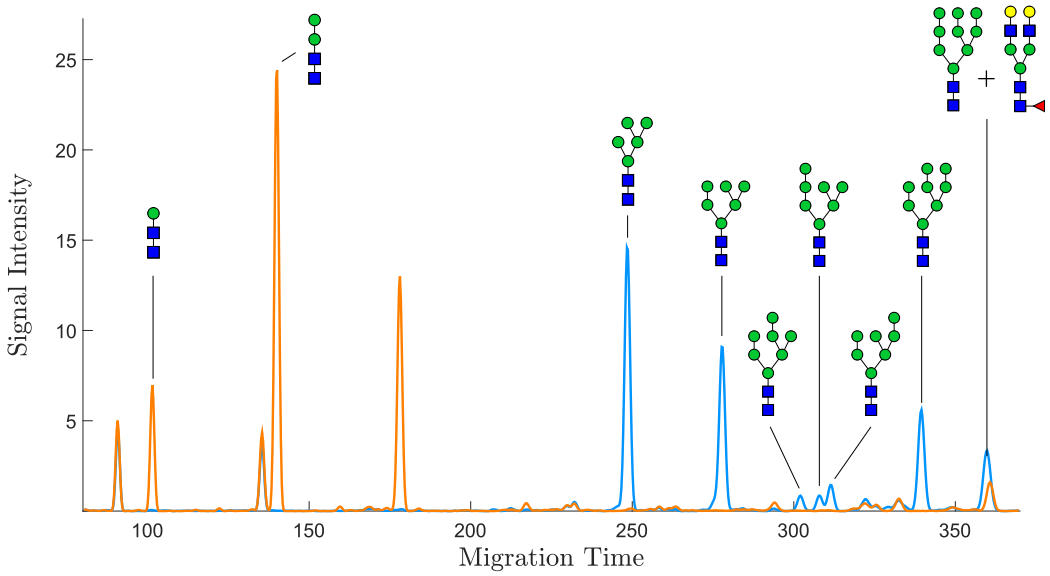


Abbildung C16: Annotierte N-Glykan-Profile vor (blau) und nach einem 1-stündigen Mannosidase-Verdau (orange).

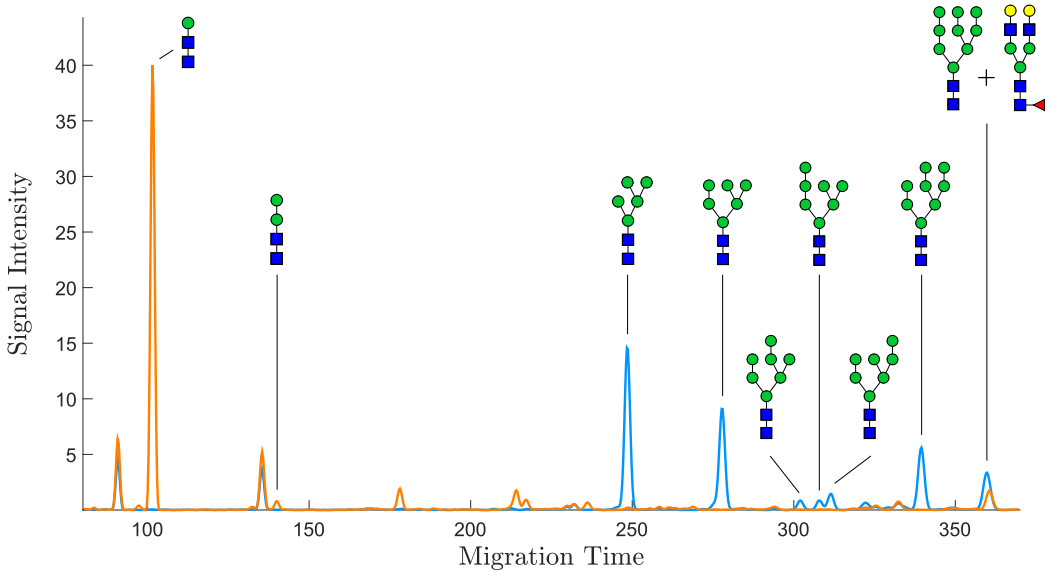


Abbildung C17: Annotierte N-Glykan-Profile vor (blau) und nach einem 19-stündigen Mannosidase-Verdau (orange).

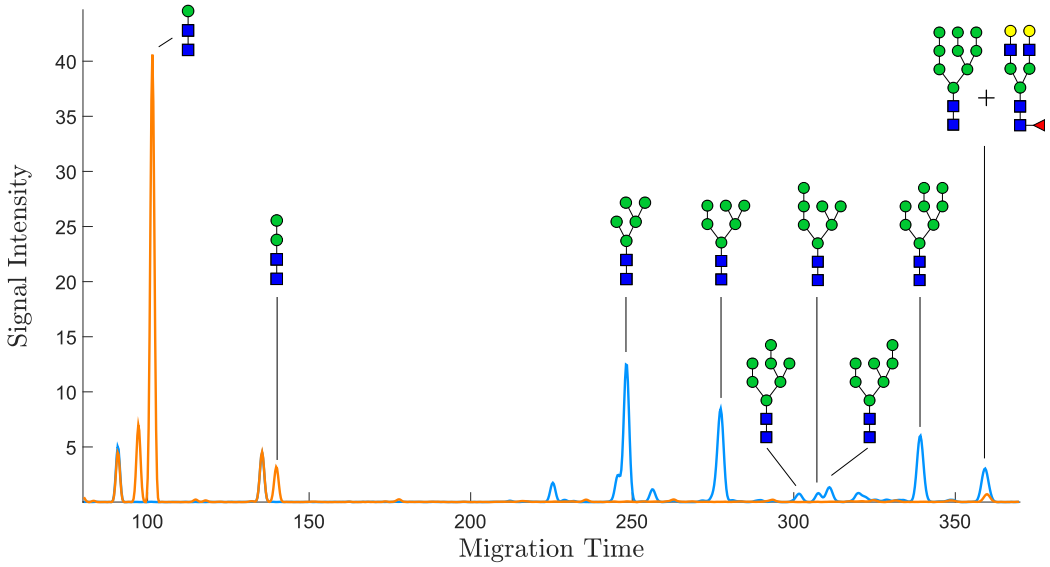


Abbildung C18: Annotierte N-Glykan-Profile vor (blau) und nach einem 3-stündigen Mannosidase-Verdau (orange).

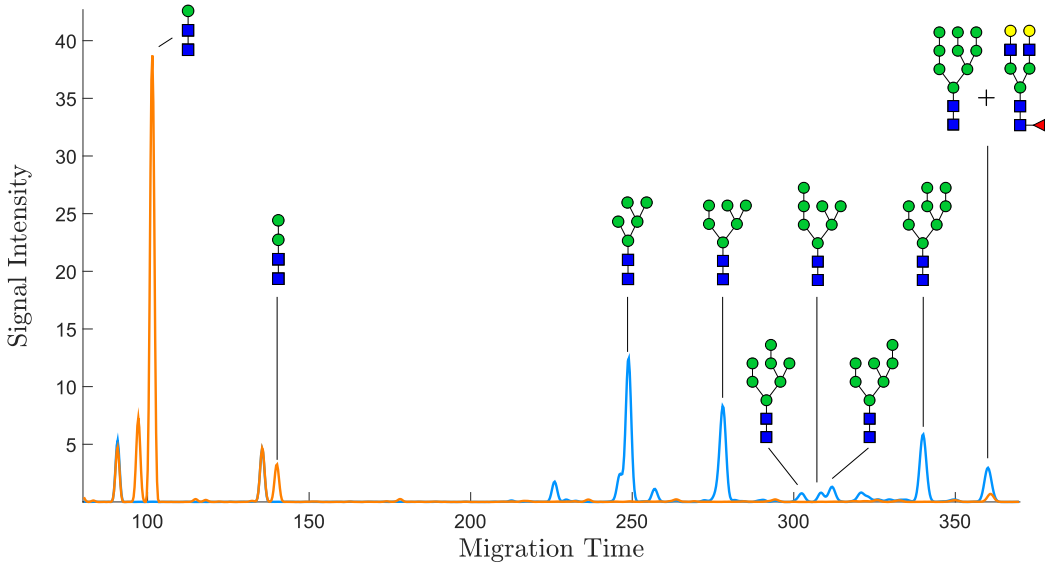


Abbildung C19: Annotierte N-Glykan-Profile vor (blau) und nach einem 3-stündigen Mannosidase-Verdau (orange).

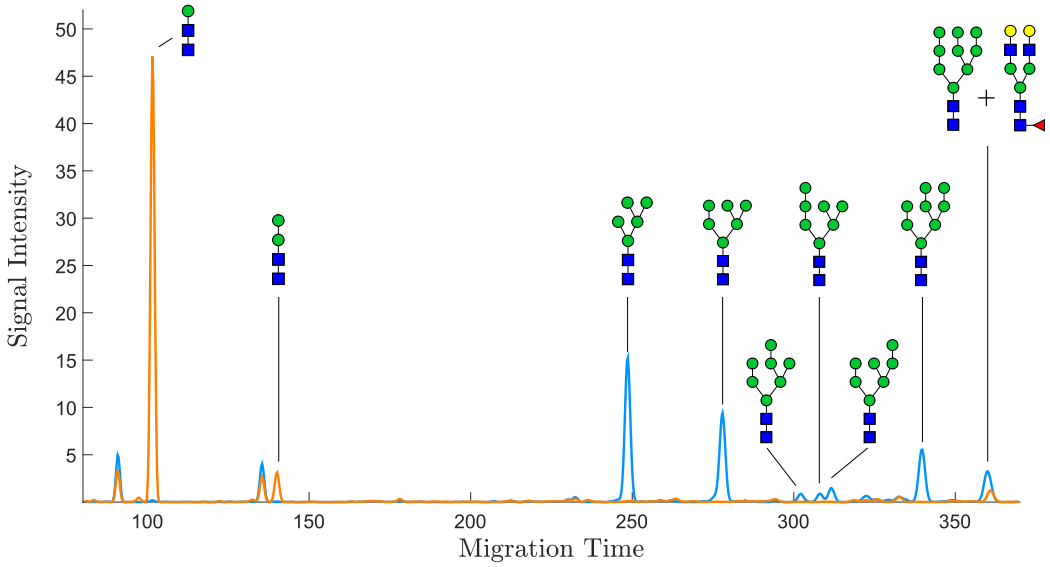


Abbildung C20: Annotierte N-Glykan-Profile vor (blau) und nach einem 4-stündigen Mannosidase-Verdau (orange).

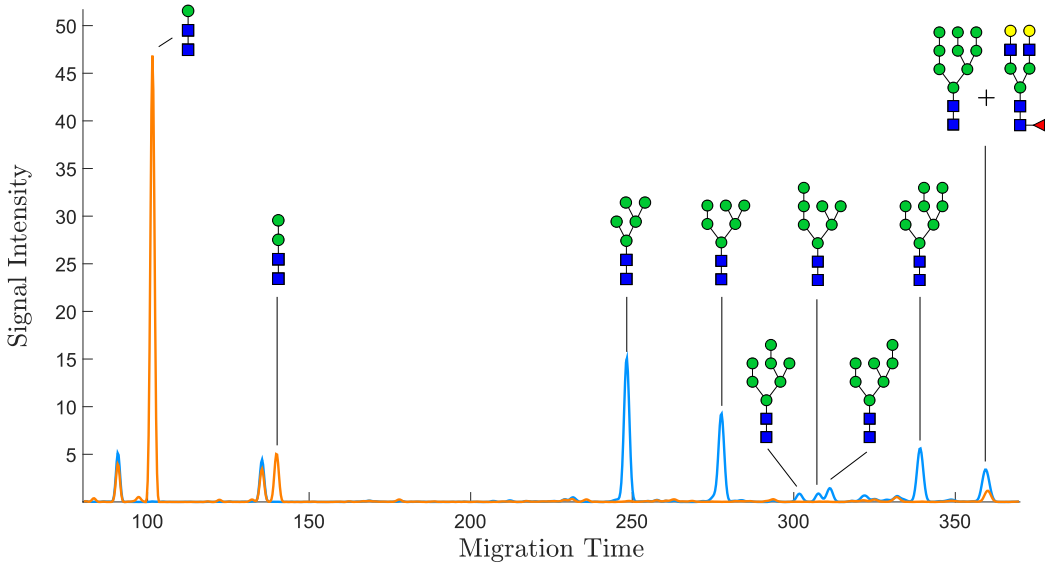


Abbildung C21: Annotierte N-Glykan-Profile vor (blau) und nach einem 4-stündigen Mannosidase-Verdau (orange).

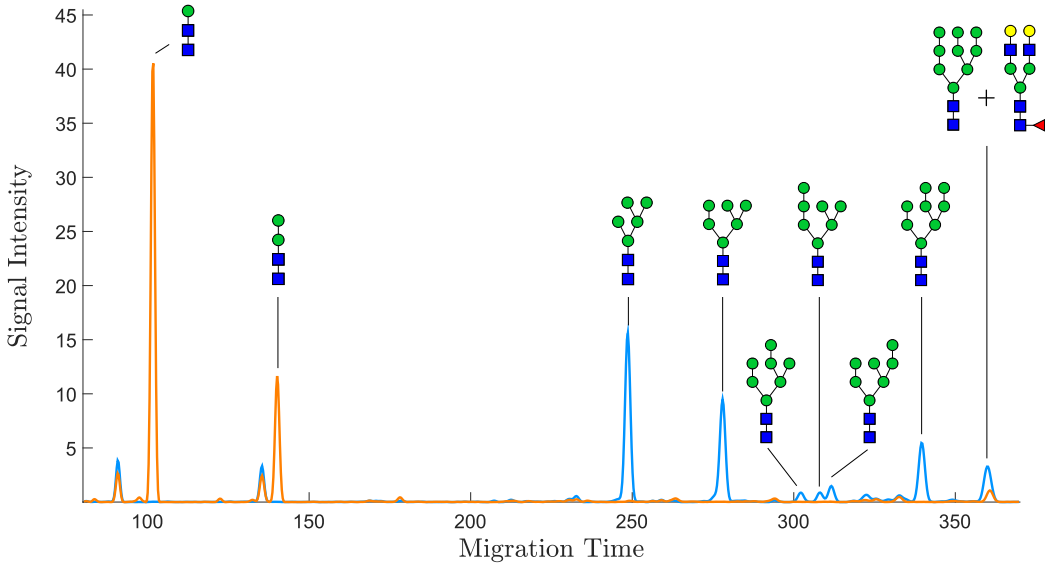


Abbildung C22: Annotierte N-Glykan-Profile vor (blau) und nach einem 15-stündigen Mannosidase-Verdau (orange).

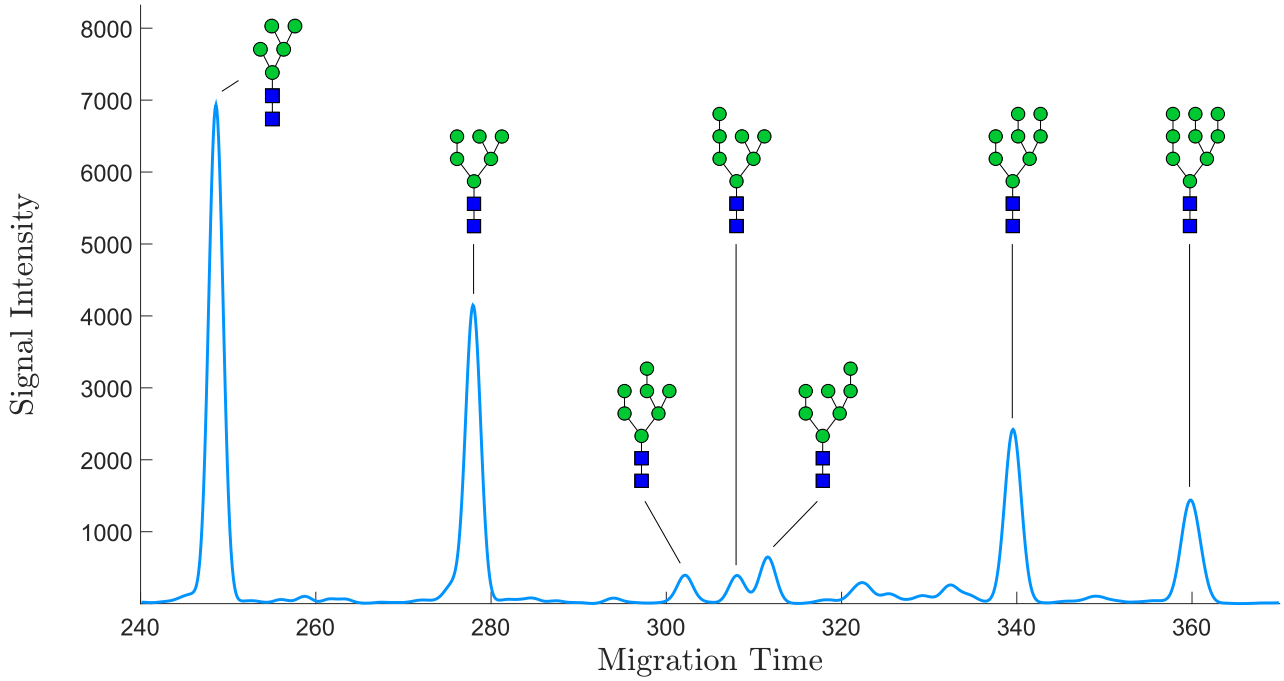


Abbildung C23: Annotiertes N-Glykan-Profil.

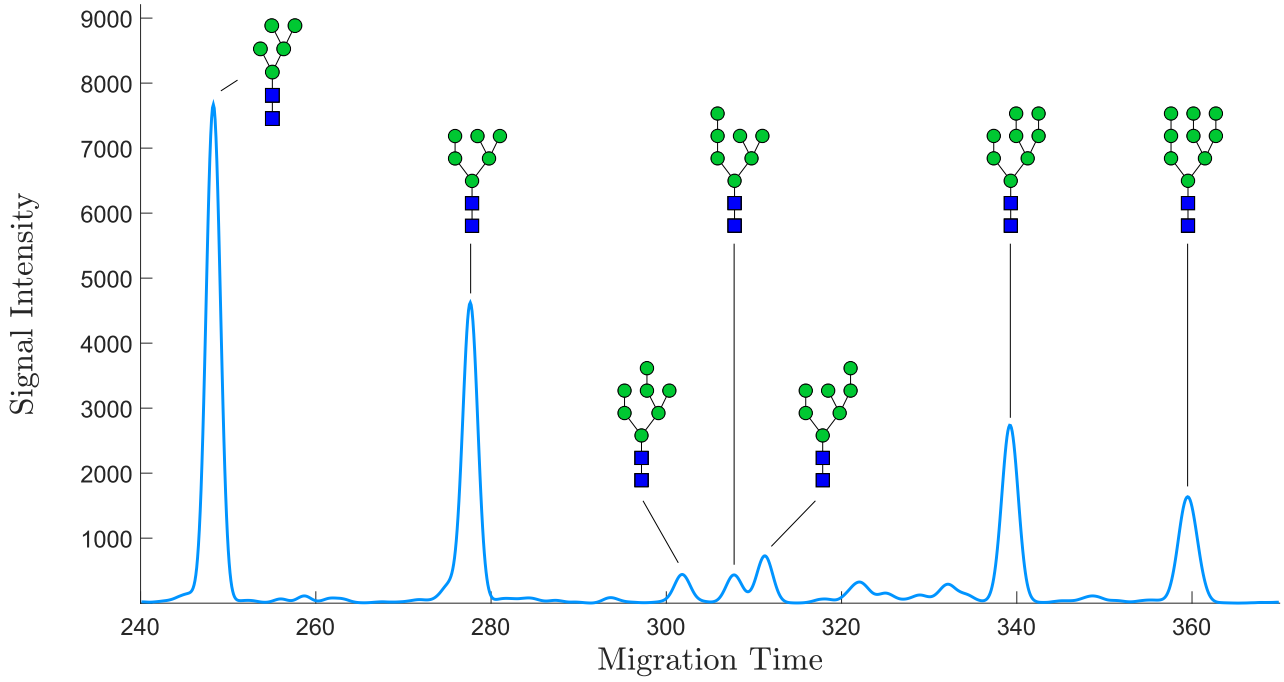


Abbildung C24: Annotiertes N-Glykan-Profil.

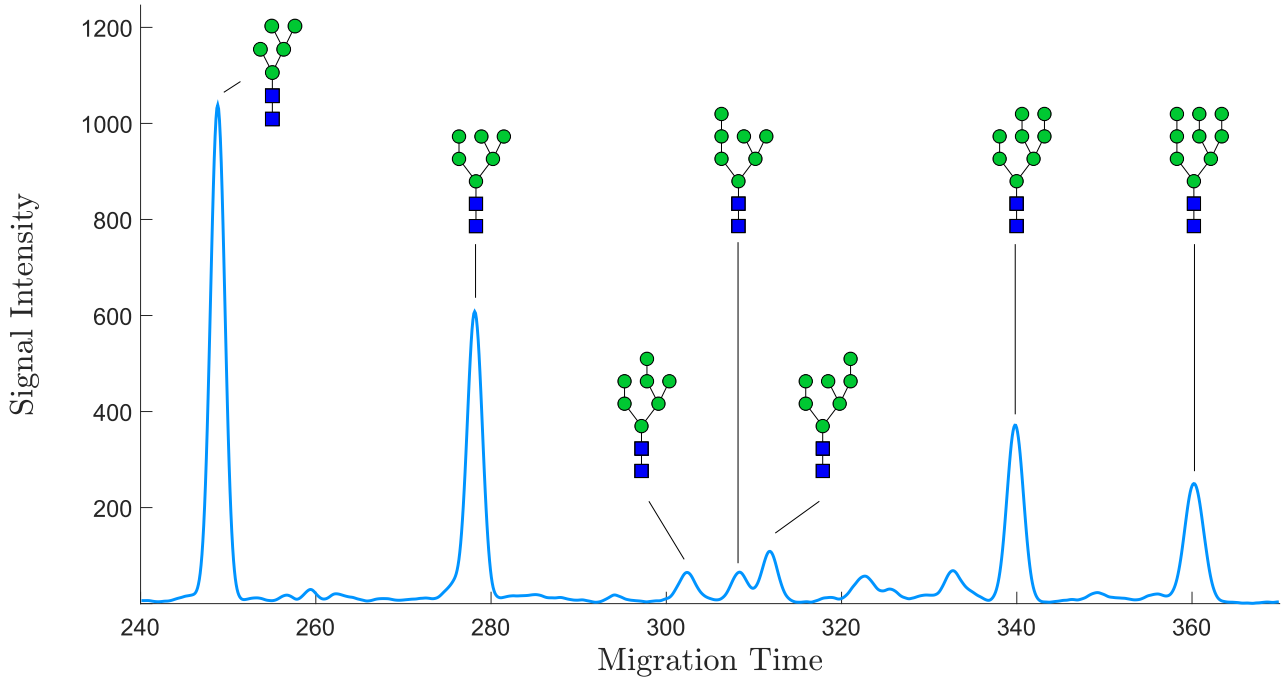


Abbildung C25: Annotiertes N-Glykan-Profil.

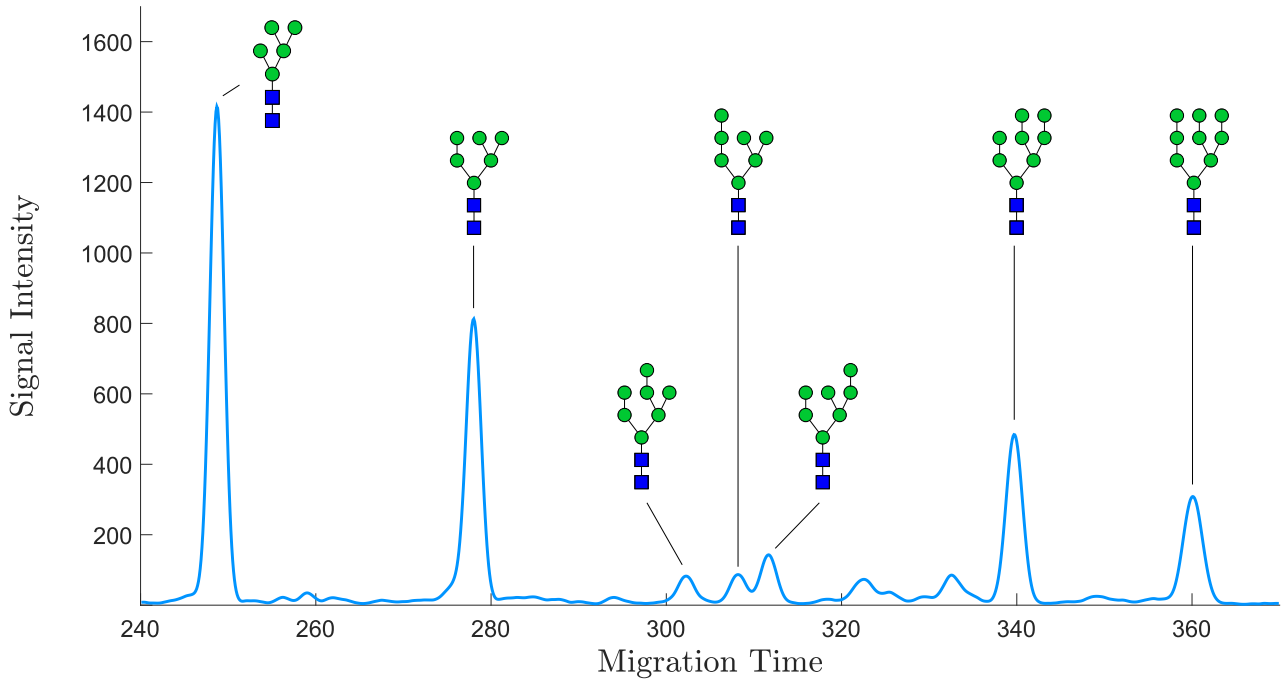


Abbildung C26: Annotiertes N-Glykan-Profil.

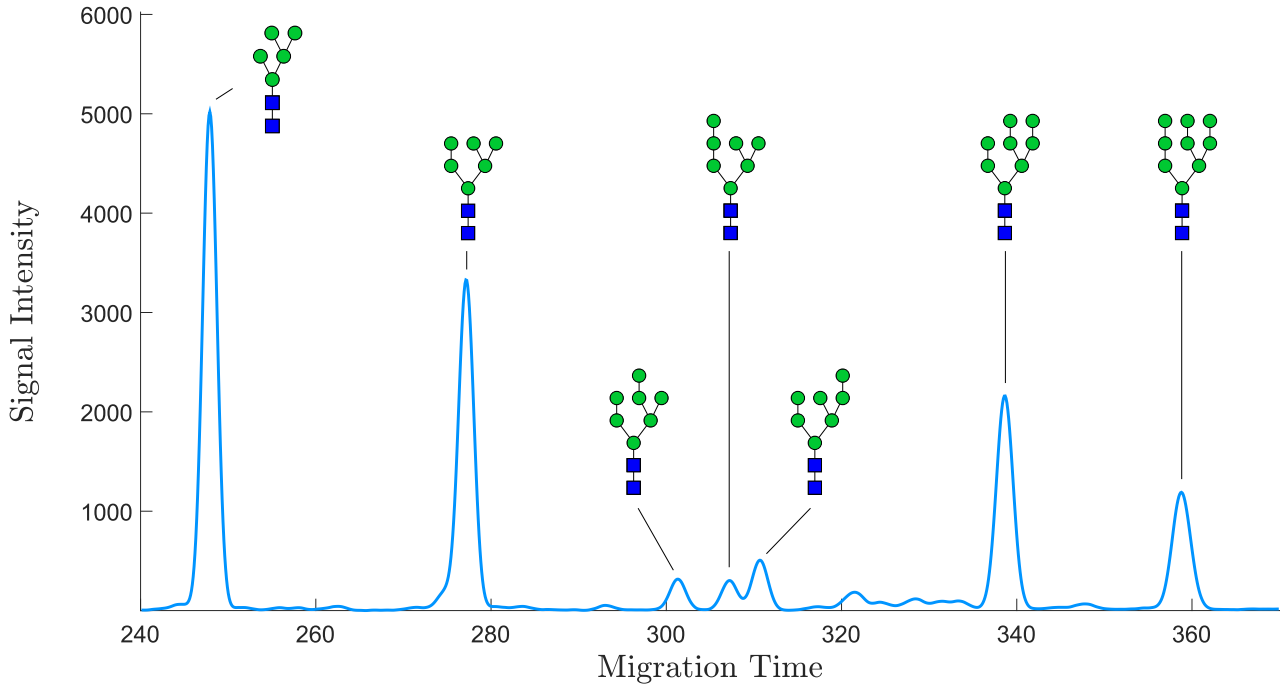


Abbildung C27: Annotiertes N-Glykan-Profil.

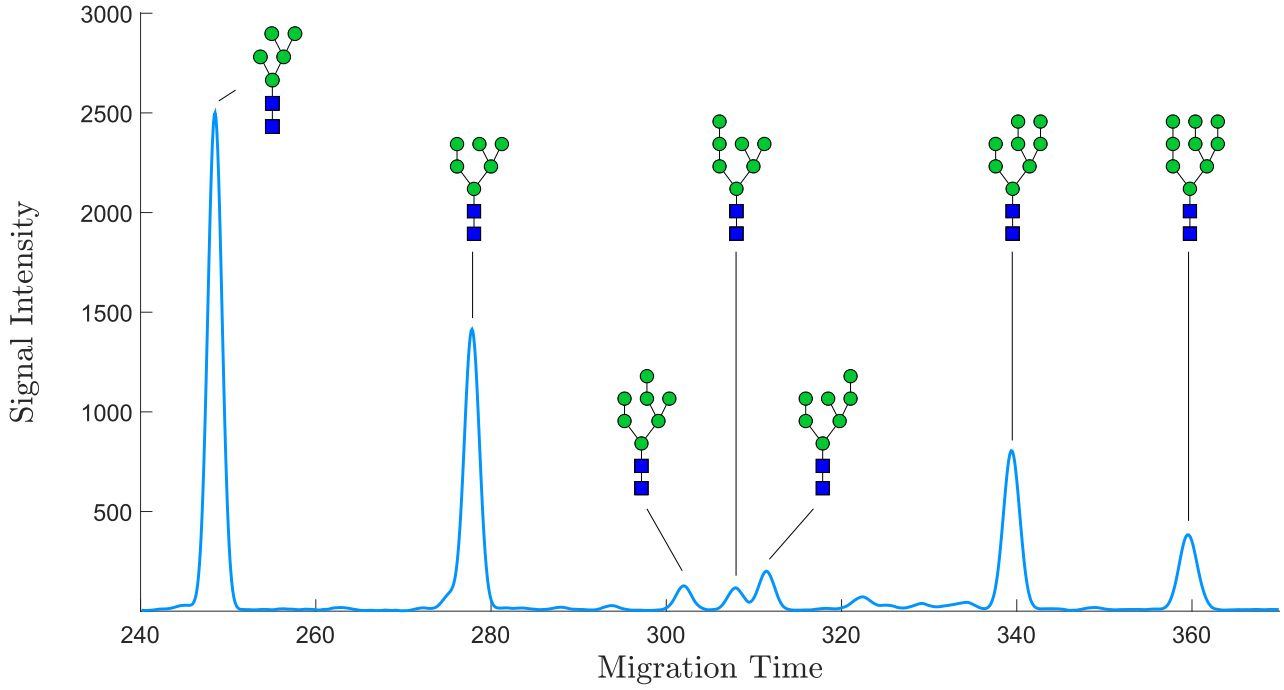


Abbildung C28: Annotiertes N-Glykan-Profil.

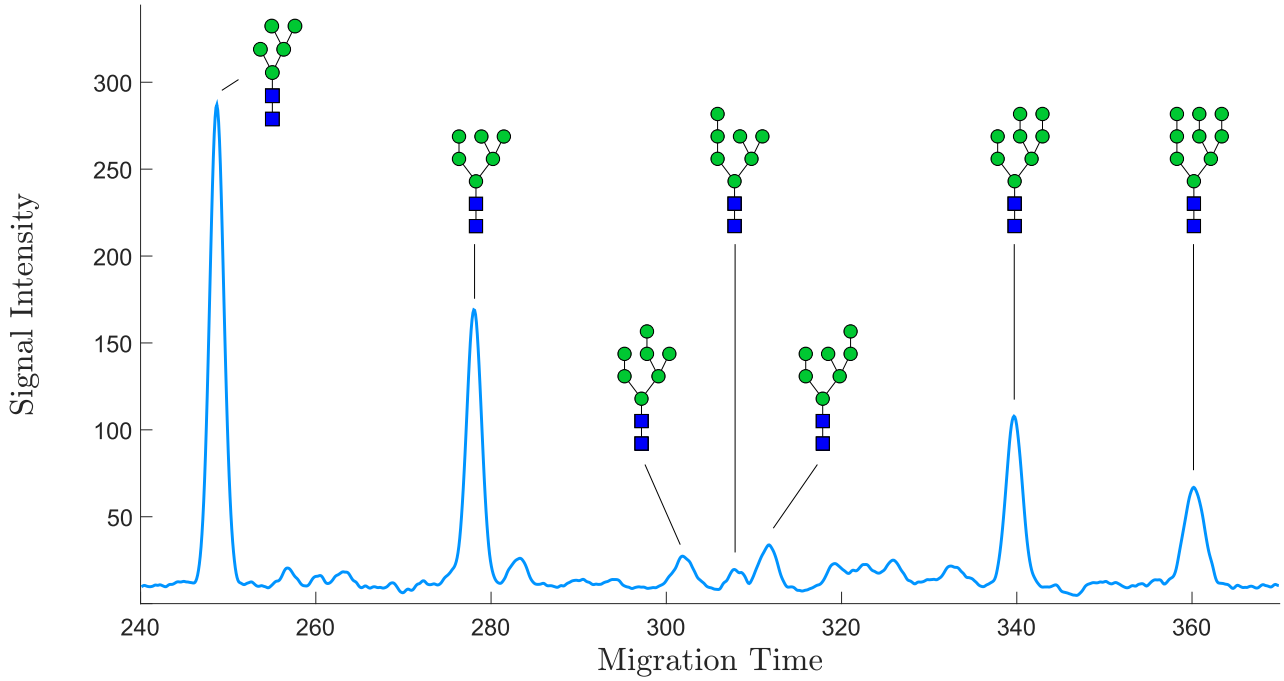


Abbildung C29: Annotiertes N-Glykan-Profil.

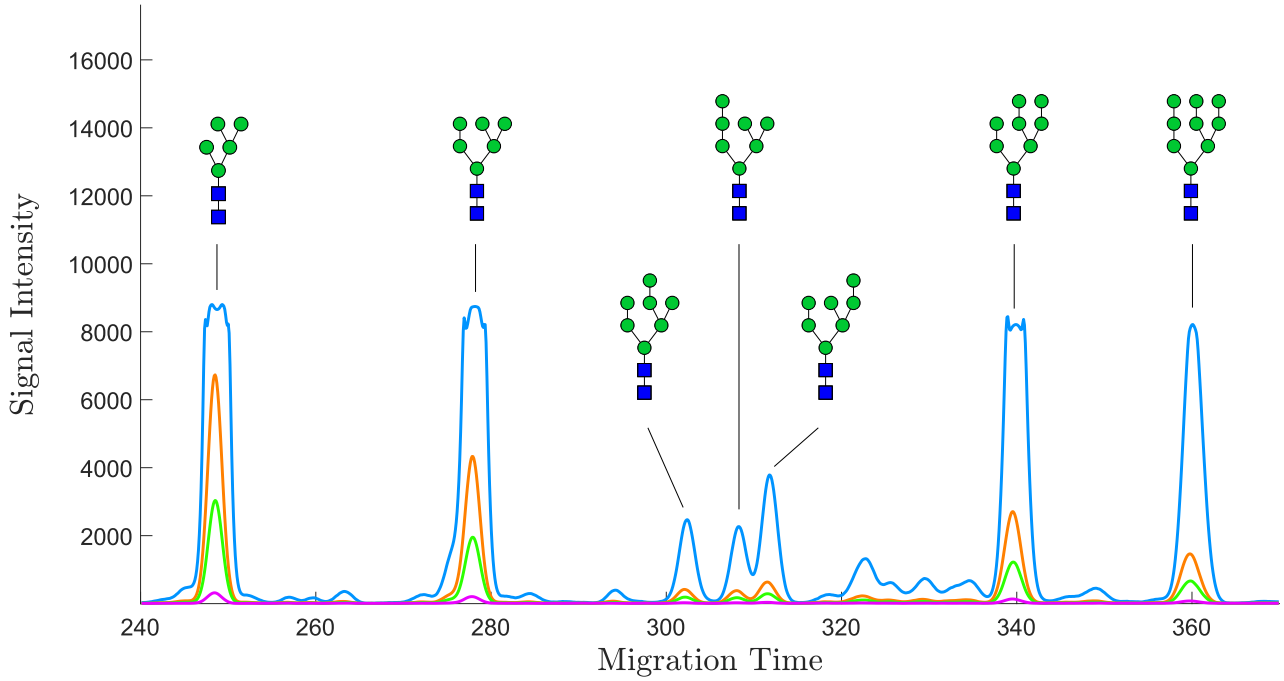


Abbildung C30: Annotiertes N-Glykan-Profil einer Verdünnungsreihe.

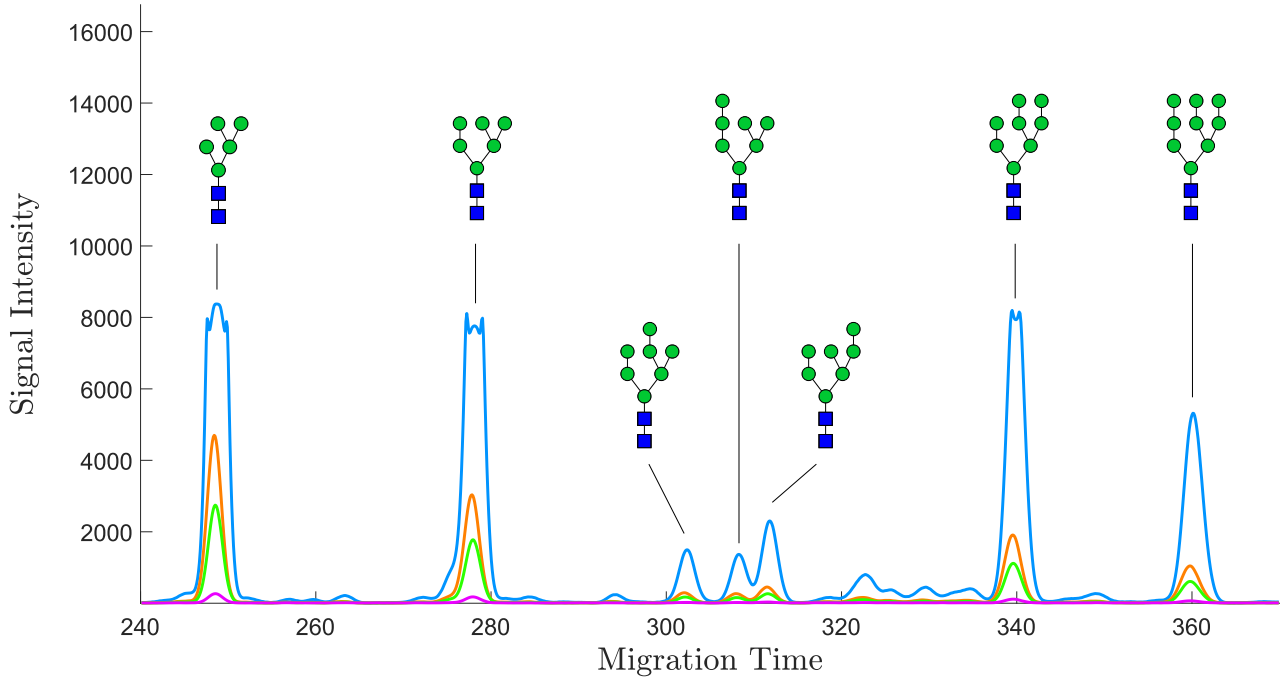


Abbildung C31: Annotiertes N-Glykan-Profil einer Verdünnungsreihe.

Glykoprotein-Typ II

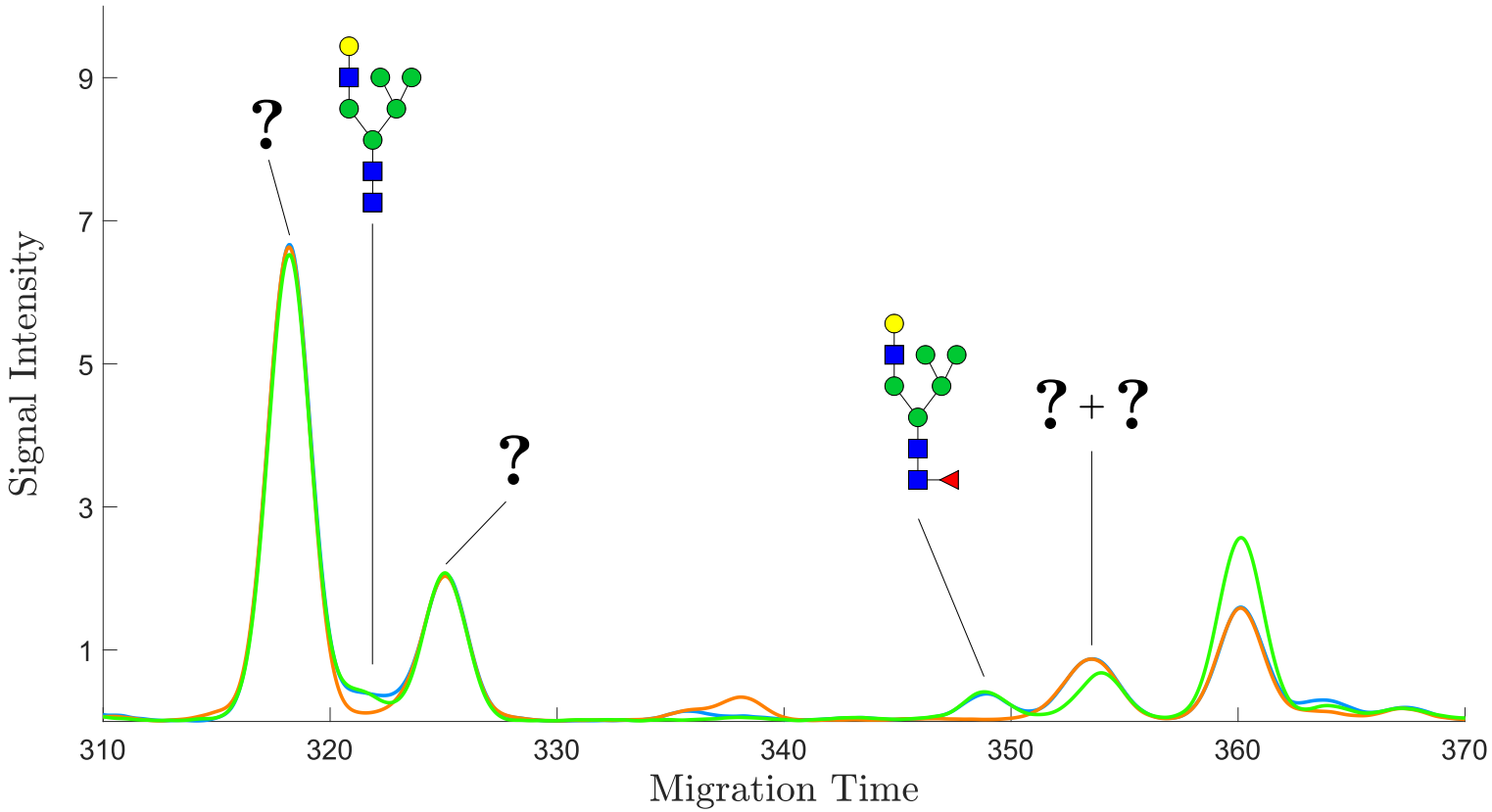


Abbildung C32: Annotierte N-Glykan-Profile vor (blau) und nach einem 4-stündigen Mannosidase- (orange) bzw. einem 3-stündigen Sialidase-Verdau (grün).

Glykoprotein-Typ III

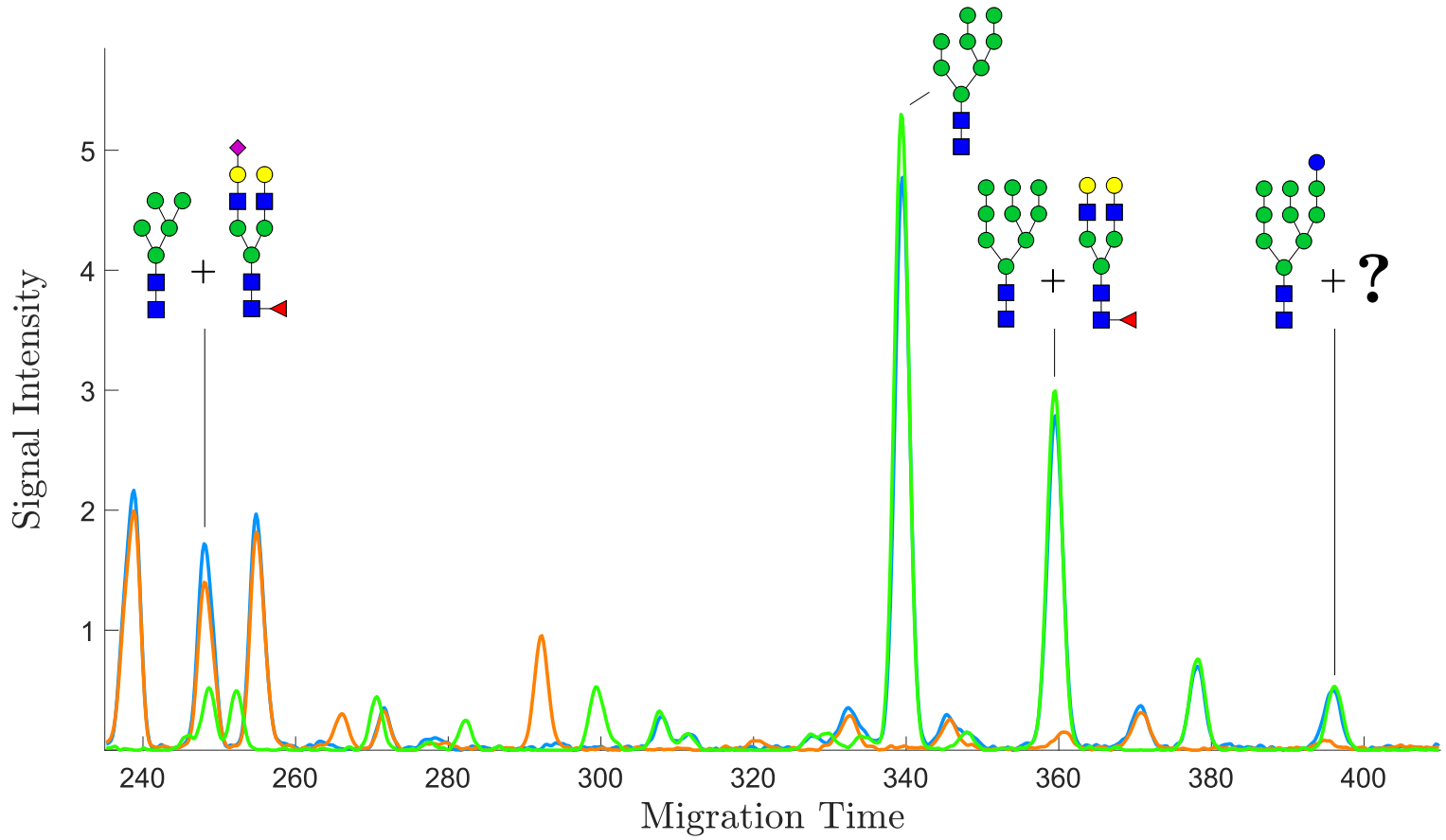


Abbildung C33: Annotierte N-Glykan-Profile vor (blau) und nach einem 15-stündigen Mannosidase- (orange) bzw. einem Galactosidase-Verdau (grün).

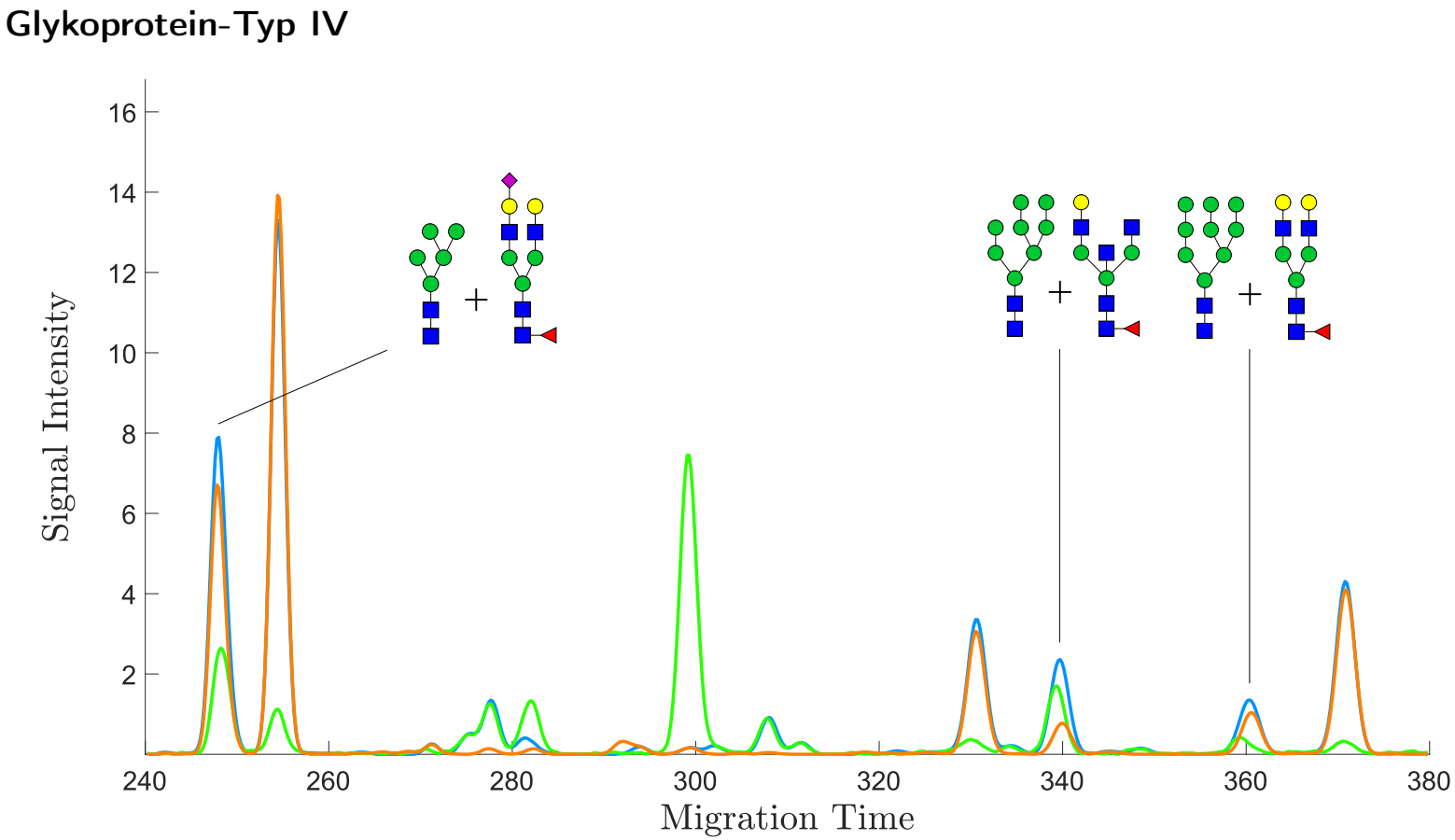


Abbildung C34: Annotierte N-Glykan-Profile vor (blau) und nach einem Mannosidase- (orange) bzw. einem Galactosidase-Verdau (grün).

Glykoprotein-Typ V

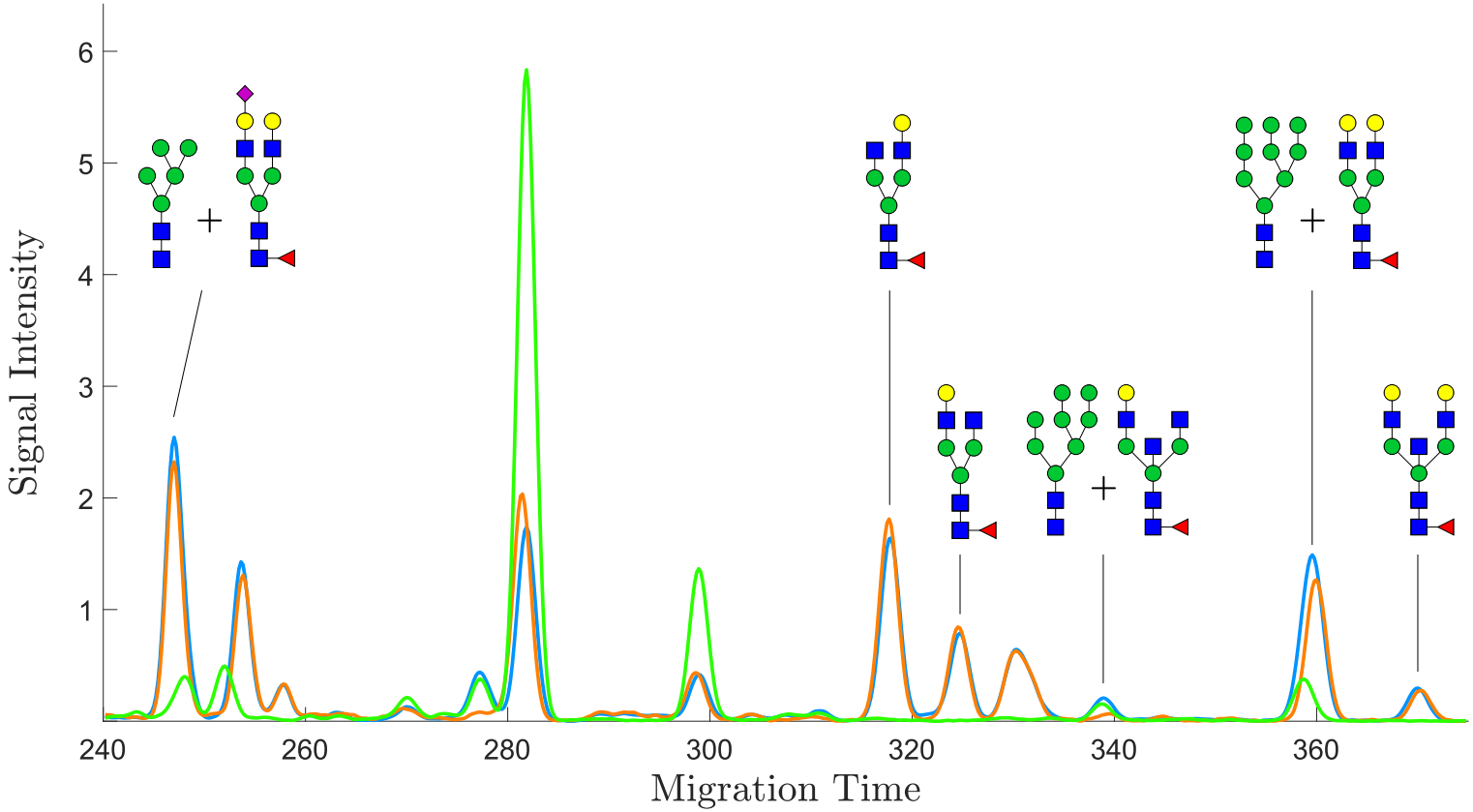


Abbildung C35: Annotierte N-Glykan-Profile vor (blau) und nach einem 15-stündigen Mannosidase- (orange) bzw. einem Galactosidase-Verdau (grün).

Anhang D – Ergebnisse

Phase 1

Tabelle D1: Die Median-Werte der E^A-Funktion der Datensätze 1 bis 5 (a, b, c, d, e).

(a)

		Ansatz 1		Ansatz 2			Ansatz 3			Ansatz 4			Ansatz 5		
		ohne	mit	für c_1^L			für c_1^L			für c_1^R			für c_1^R		
K	L_i	σ-GNB	σ-GNB	0	0.5	0.9	0	0.5	0.9	0	0.01	0.1	0	0.01	0.1
2	I	0.1302	0.1639	0.1302	0.1059	0.1384	0.1302	0.1059	0.1384	**0.0657**	0.0663	0.1512	**0.0657**	0.0663	0.1513
	II	0.1306	0.1807	0.1306	0.0880	0.1596	0.1306	0.0880	0.1591	**0.0513**	0.0515	0.1253	**0.0513**	0.0515	0.1247
	III	0.0438	0.1904	0.0438	0.0946	0.1903	0.0438	0.0946	0.1877	0.0522	0.0563	0.0562	**0.0415**	0.0562	0.2089
3	I	0.0681	0.0921	0.0681	0.0759	0.0922	0.0681	0.0759	0.0933	0.0305	**0.0302**	0.0785	0.0304	**0.0302**	0.0790
	II	0.0856	0.1249	0.0851	0.0870	0.1249	0.0861	0.0871	0.1196	0.0822	**0.0395**	0.0699	0.0847	0.0397	0.0712
	III	**0.0310**	0.1634	0.0522	0.0637	0.1632	0.0521	0.0639	0.1577	0.0401	0.0456	0.1067	0.0405	0.1142	0.1534
4	I	0.0738	0.0922	0.0738	0.0738	0.0921	0.0738	0.0738	0.0914	0.0332	0.0331	0.0706	0.0332	**0.0330**	0.5043
	II	0.0514	0.0860	0.0850	0.0404	0.0858	0.1202	0.0413	0.0859	0.0798	**0.0228**	0.0507	0.0993	0.0243	0.0558
	III	**0.0204**	0.0636	0.0568	0.0681	0.0636	0.0610	0.0810	0.1065	0.0457	0.0684	0.0964	0.0532	0.0800	0.0935
5	I	0.0526	0.1064	0.0526	0.0529	0.1063	0.0527	0.0533	0.1056	**0.0281**	0.0285	0.0945	0.0328	0.0293	0.4793
	II	0.0663	0.1000	0.1036	0.0828	0.0998	0.1293	0.0912	0.1125	0.0886	**0.0268**	0.0642	0.1010	0.0933	0.1142
	III	0.0364	0.0528	0.0535	0.0759	0.0531	0.0845	0.0858	0.1189	0.0367	0.0619	**0.0340**	0.0751	0.0689	0.1010

(b)

		Ansatz 1		Ansatz 2			Ansatz 3			Ansatz 4			Ansatz 5		
		ohne	mit	für c_1^L			für c_1^L			für c_1^R			für c_1^R		
K	L_i	σ-GNB	σ-GNB	0	0.5	0.9	0	0.5	0.9	0	0.01	0.1	0	0.01	0.1
2	I	0.1175	0.1084	0.1175	0.0942	0.1083	0.1175	0.0942	0.1081	0.0587	0.0591	0.0962	0.0587	0.0591	0.0961
	II	0.1236	0.1206	0.1236	0.0978	0.1203	0.1236	0.0978	0.1189	0.0457	0.0458	0.0657	0.0457	0.0458	0.0655
	III	0.0365	0.1435	0.0365	0.0948	0.1435	0.0365	0.0948	0.1406	0.0391	0.0407	0.0399	0.0333	0.0408	0.1811
3	I	0.0695	0.1466	0.0695	0.0743	0.1466	0.0695	0.0743	0.1465	0.0267	0.0265	0.1292	0.0267	0.0264	0.1302
	II	0.0609	0.0770	0.0609	0.0794	0.0770	0.0673	0.0794	0.0761	0.0490	0.0230	0.0640	0.0542	0.0231	0.0646
	III	0.0217	0.0737	0.0565	0.0534	0.0737	0.0565	0.0924	0.1008	0.0244	0.0262	0.0491	0.0306	0.0865	0.1253
4	I	0.0693	0.0937	0.0693	0.0620	0.0938	0.0693	0.0620	0.0932	0.0355	0.0354	0.0898	0.0355	0.0354	0.5111
	II	0.0518	0.1012	0.0625	0.0611	0.1010	0.0775	0.0616	0.0966	0.0457	0.0189	0.0798	0.0834	0.0262	0.0859
	III	0.0211	0.0830	0.0572	0.0418	0.0829	0.0570	0.0940	0.1104	0.0264	0.0232	0.0532	0.0580	0.0910	0.1334
5	I	0.0460	0.0767	0.0460	0.0484	0.0767	0.0460	0.0488	0.0798	0.0243	0.0241	0.0686	0.0265	0.0240	0.4020
	II	0.0683	0.0801	0.1086	0.0461	0.0802	0.1255	0.0513	0.0905	0.0891	0.0204	0.0603	0.1046	0.0669	0.2803
	III	0.0520	0.0802	0.0698	0.1158	0.0804	0.0893	0.0841	0.1331	0.0521	0.0777	0.0969	0.0947	0.0706	0.1566

(c)

		Ansatz 1		Ansatz 2			Ansatz 3			Ansatz 4			Ansatz 5		
		ohne	mit	für c_1^L			für c_1^L			für c_1^R			für c_1^R		
K	L_i	σ-GNB	σ-GNB	0	0.5	0.9	0	0.5	0.9	0	0.01	0.1	0	0.01	0.1
2	I	0.1231	0.1741	0.1231	0.0983	0.1490	0.1231	0.0983	0.1490	0.0630	0.0635	0.1670	0.0630	0.0635	0.1671
	II	0.1231	0.1568	0.1231	0.0904	0.1506	0.1231	0.0905	0.1501	0.0500	0.0502	0.1025	0.0500	0.0502	0.1014
	III	0.0377	0.1361	0.0377	0.1007	0.1360	0.0377	0.1007	0.1321	0.0418	0.0424	0.0418	0.0348	0.0425	0.1789
3	I	0.1046	0.1597	0.1046	0.0892	0.1169	0.1046	0.0892	0.1163	0.0531	0.0541	0.1516	0.0531	0.0541	0.1566
	II	0.0624	0.1323	0.0616	0.0470	0.1324	0.0754	0.0472	0.1306	0.0587	0.0285	0.1051	0.0295	0.0285	0.1045
	III	0.0330	0.0860	0.0659	0.0766	0.0861	0.0678	0.0744	0.0761	0.0271	0.0259	0.0544	0.0307	0.1395	0.1540
4	I	0.0804	0.1630	0.0804	0.0829	0.1269	0.0804	0.0829	0.1247	0.0384	0.0383	0.1311	0.0384	0.0383	0.4938
	II	0.0579	0.1555	0.0922	0.0512	0.1554	0.1042	0.0525	0.1488	0.0857	0.0264	0.1197	0.1037	0.0322	0.1288
	III	0.0209	0.1595	0.0756	0.0816	0.1360	0.0756	0.0629	0.1678	0.0328	0.0795	0.1419	0.0762	0.0752	0.1602
5	I	0.0467	0.0671	0.0467	0.0506	0.0670	0.0468	0.0503	0.0712	0.0227	0.0224	0.0611	0.0240	0.0223	0.4111
	II	0.0613	0.0515	0.0888	0.0452	0.0515	0.1133	0.0563	0.0730	0.0845	0.0203	0.0330	0.0991	0.0890	0.2180
	III	0.0452	0.0839	0.0965	0.0716	0.0838	0.1504	0.0701	0.1063	0.0893	0.0344	0.0488	0.1146	0.1097	0.1638

(d)

		Ansatz 1		Ansatz 2			Ansatz 3			Ansatz 4			Ansatz 5		
		ohne	mit	für c_1^L			für c_1^L			für c_1^R			für c_1^R		
K	L_i	σ-GNB	σ-GNB	0	0.5	0.9	0	0.5	0.9	0	0.01	0.1	0	0.01	0.1
2	I	0.0899	0.1207	0.0899	0.1127	0.1207	0.0899	0.1127	0.1205	0.0394	0.0390	0.1186	0.0394	0.0390	0.1186
	II	0.1337	0.1353	0.1337	0.0850	0.1349	0.1337	0.0850	0.1342	0.0479	0.0481	0.0856	0.0479	0.0481	0.0856
	III	0.0369	0.1449	0.0369	0.0965	0.1449	0.0369	0.0965	0.1416	0.0431	0.0459	0.0452	0.0344	0.0459	0.1880
3	I	0.0884	0.1411	0.0884	0.0962	0.1175	0.0884	0.0962	0.1170	0.0429	0.0430	0.1120	0.0429	0.0430	0.6528
	II	0.0629	0.1371	0.0623	0.0505	0.1372	0.0625	0.0508	0.1358	0.0596	0.0313	0.1056	0.0369	0.0312	0.1062
	III	0.0277	0.1553	0.0457	0.0588	0.1551	0.0440	0.0578	0.1472	0.0287	0.0624	0.1180	0.0318	0.0800	0.1585
4	I	0.0638	0.1145	0.0638	0.0553	0.1146	0.0638	0.0553	0.1134	0.0331	0.0334	0.1177	0.0331	0.0335	0.5013
	II	0.0490	0.0885	0.0750	0.0562	0.0883	0.0773	0.0566	0.0905	0.0456	0.0188	0.0561	0.0977	0.0216	0.0660
	III	0.0257	0.1216	0.0602	0.0690	0.1214	0.0722	0.1175	0.1645	0.0338	0.0276	0.0795	0.0527	0.0877	0.1553
5	I	0.0478	0.0875	0.0478	0.0625	0.0875	0.0479	0.0623	0.0881	0.0253	0.0252	0.0750	0.0260	0.0254	0.4359
	II	0.0645	0.0636	0.0905	0.0723	0.0635	0.1093	0.0671	0.0814	0.0830	0.0218	0.0366	0.1037	0.0704	0.1328
	III	0.0217	0.1033	0.0986	0.0874	0.1034	0.1012	0.0968	0.1383	0.0739	0.0814	0.1053	0.0958	0.0782	0.1669

(e)

		Ansatz 1		Ansatz 2			Ansatz 3			Ansatz 4			Ansatz 5		
		ohne	mit	für c_1^L			für c_1^L			für c_1^R			für c_1^R		
K	L_i	σ-GNB	σ-GNB	0	0.5	0.9	0	0.5	0.9	0	0.01	0.1	0	0.01	0.1
2	I	0.1260	0.2233	0.1260	0.1007	0.1614	0.1260	0.1007	0.1614	0.0648	0.0654	0.2143	0.0648	0.0654	0.2145
	II	0.1556	0.2008	0.1556	0.0984	0.1803	0.1396	0.0984	0.1804	0.0538	0.0539	0.1586	0.0538	0.0539	0.1588
	III	0.0387	0.1221	0.0387	0.1017	0.1220	0.0387	0.1016	0.1181	0.0494	0.0525	0.0522	0.0380	0.0524	0.2056
3	I	0.0810	0.1105	0.0810	0.0809	0.1104	0.0810	0.0809	0.1100	0.0402	0.0401	0.1037	0.0402	0.0401	0.1057
	II	0.0589	0.0840	0.0589	0.0840	0.0839	0.0589	0.0834	0.0871	0.0535	0.0232	0.0803	0.0540	0.0233	0.0812
	III	0.0378	0.1105	0.0599	0.1024	0.1105	0.0601	0.1023	0.1072	0.0381	0.0515	0.0810	0.0445	0.1484	0.1530
4	I	0.0638	0.1315	0.0638	0.0587	0.1249	0.0638	0.0587	0.1245	0.0332	0.0339	0.1338	0.0332	0.0340	0.5228
	II	0.0505	0.0844	0.0803	0.0434	0.0842	0.0845	0.0430	0.0851	0.0640	0.0202	0.0620	0.0872	0.0268	0.0706
	III	0.0208	0.0546	0.0602	0.0366	0.0546	0.0780	0.0815	0.1419	0.0206	0.0184	0.0404	0.0607	0.0908	0.1162
5	I	0.0467	0.1030	0.0467	0.0500	0.1030	0.0468	0.0502	0.1047	0.0210	0.0208	0.3775	0.0658	0.3774	0.4012
	II	0.0488	0.0717	0.0893	0.1122	0.0719	0.1144	0.0723	0.0902	0.0853	0.0864	0.0488	0.1063	0.0819	0.1035
	III	0.0415	0.0885	0.0682	0.0585	0.0885	0.0842	0.0844	0.1304	0.0572	0.0929	0.0465	0.0952	0.1038	0.1523

Phase 2

Tabelle D2: Die Median-Werte von ΔK der Datensätze 1 bis 5 (a, b, c, d, e).

(a)

		Ansatz 1 für c_2^L			Ansatz 2 für c_2^L			Ansatz 3 für c_2^L			Ansatz 4 für c_2^L			Ansatz 5 für c_2^L		
K	L_i	0.01	0.1	0.2	0.01	0.1	0.2	0.01	0.1	0.2	0.01	0.1	0.2	0.01	0.1	0.2
2	I	0	0	0	0	0	0	1	0	0	2	0	0	3	1	0
	II	3	1	0	1	0	0	1	0	−1	0	0	0	2	0	0
	III	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0
3	I	0	0	0	0	0	0	1	0	0	0	0	0	2	0	0
	II	0	0	0	0	0	−1	0	−1	−2	1	0	−1	2	−1	−1
	III	0	0	0	0	−1	−2	0	−2	−1	1	0	0	2	−1	−2
4	I	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0
	II	0	0	0	1	0	−1	0	−2	−2	0	0	−1	2	−1	−2
	III	0	0	0	0	−1	−1	0	−1	−1	1	−1	−1	2	0	−1
5	I	0	0	0	1	0	0	1	0	0	0	0	0	2	1	0
	II	0	0	0	0	0	−2	0	−2	−3	0	−1	−2	1	−1	−2
	III	0	0	0	0	−1	−2	0	−2	−2	0	−2	−2	2	0	−2

		Ansatz 6 für c_2^R			Ansatz 7 für c_2^R			Ansatz 8 für c_2^R			Ansatz 9 für c_2^R		
K	L_i	0.0001	0.001	0.01	0.0001	0.001	0.01	0.0001	0.001	0.01	0.0001	0.01	0.1
2	I	2	0	0	3	1	0	1	0	0	3	2	0
	II	1	0	0	2	1	−1	1	0	0	3	2	−1
	III	0	0	0	2	1	−1	0	0	−1	1	1	−1
3	I	2	0	0	2	0	−2	0	0	0	2	0	−2
	II	1	0	0	2	0	−2	0	0	0	1	1	−2
	III	1	0	−1	2	0	−2	0	0	−1	2	1	−2
4	I	1	0	−3	2	0	−3	0	0	−3	1	0	−3
	II	2	1	−2	1	0	−3	1	1	−2	1	0	−2
	III	2	0	−1	1	0	−3	0	−1	−1	2	0	−3
5	I	2	0	−4	1	−1	−4	0	0	−4	1	0	−4
	II	2	1	−3	0	−1	−4	0	0	−3	1	0	−4
	III	1	0	−2	1	−1	−4	0	−1	−2	1	0	−4

(b)

		Ansatz 1 für c_2^L			Ansatz 2 für c_2^L			Ansatz 3 für c_2^L			Ansatz 4 für c_2^L			Ansatz 5 für c_2^L		
K	L_i	0.01	0.1	0.2	0.01	0.1	0.2	0.01	0.1	0.2	0.01	0.1	0.2	0.01	0.1	0.2
2	I	0	0	0	0	0	0	1	0	0	2	0	0	2	0	0
	II	0	0	0	0	0	0	1	0	−1	0	0	0	1	0	0
	III	0	0	0	0	0	0	0	0	0	1	0	0	2	0	−1
3	I	0	0	0	0	0	0	0	0	0	1	0	0	2	0	0
	II	0	0	0	0	0	0	0	−1	−1	0	0	0	2	0	−1
	III	0	0	0	0	−1	−1	1	−1	−1	1	−1	−1	2	0	0
4	I	1	1	0	1	0	0	1	0	0	0	0	0	3	1	0
	II	0	0	0	2	0	−1	1	−1	−2	0	0	−1	3	0	−2
	III	0	0	0	1	−1	−1	1	−1	−2	1	−1	−1	3	0	−1
5	I	1	0	0	0	0	0	0	0	0	0	0	0	2	0	0
	II	0	0	0	1	0	−2	0	−2	−3	0	−1	−2	2	−2	−2
	III	0	0	0	0	−1	−2	0	−2	−3	0	−1	−2	2	−1	−2

		Ansatz 6 für c_2^R			Ansatz 7 für c_2^R			Ansatz 8 für c_2^R			Ansatz 9 für c_2^R		
K	L_i	0.0001	0.001	0.01	0.0001	0.001	0.01	0.0001	0.001	0.01	0.0001	0.01	0.1
2	I	1	0	0	2	1	0	0	0	0	2	1	0
	II	1	0	0	3	1	−1	0	0	0	2	2	−1
	III	0	0	0	1	0	−1	0	0	−1	2	1	0
3	I	1	0	0	2	0	−2	0	0	0	1	0	−2
	II	1	0	0	2	0	−2	1	0	−1	2	1	−2
	III	1	0	−1	2	1	−1	1	0	−1	2	1	−1
4	I	1	0	−3	1	0	−3	0	0	−3	1	0	−3
	II	2	1	−1	1	0	−2	1	0	−1	2	0	−2
	III	2	0	−1	2	0	−3	0	−1	−1	2	0	−3
5	I	1	1	−4	1	0	−4	0	0	−4	0	0	−4
	II	2	1	−3	1	−2	−3	0	0	−3	1	0	−3
	III	1	1	−3	1	−2	−3	0	0	−3	1	0	−4

(c)

		Ansatz 1 für c_2^L			Ansatz 2 für c_2^L			Ansatz 3 für c_2^L			Ansatz 4 für c_2^L			Ansatz 5 für c_2^L		
K	L_i	0.01	0.1	0.2	0.01	0.1	0.2	0.01	0.1	0.2	0.01	0.1	0.2	0.01	0.1	0.2
2	I	1	1	1	0	0	0	1	0	0	2	0	0	3	2	0
	II	0	0	0	0	0	0	1	0	−1	1	0	0	2	0	0
	III	0	0	0	0	0	0	0	0	0	1	0	0	1	1	−1
3	I	0	0	0	1	0	0	1	0	0	1	0	0	3	1	0
	II	0	0	0	0	0	0	0	0	−1	2	0	0	2	0	−1
	III	0	0	0	0	0	−1	0	−1	−1	1	0	−1	2	0	−1
4	I	0	0	0	1	0	0	1	0	0	0	0	0	2	1	0
	II	0	0	0	0	0	−1	0	−2	−2	1	0	−1	3	0	−2
	III	0	0	0	1	−1	−1	0	−1	−2	1	−1	−2	3	0	−2
5	I	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0
	II	0	0	0	2	−1	−1	0	−2	−3	1	−1	−2	2	−2	−3
	III	0	0	0	0	0	−2	0	−3	−3	1	−1	−2	2	−2	−3

		Ansatz 6 für c_2^R			Ansatz 7 für c_2^R			Ansatz 8 für c_2^R			Ansatz 9 für c_2^R		
K	L_i	0.0001	0.001	0.01	0.0001	0.001	0.01	0.0001	0.001	0.01	0.0001	0.01	0.1
2	I	1	0	0	2	1	0	1	0	0	2	2	0
	II	1	0	0	2	1	−1	0	0	0	2	2	−1
	III	0	0	−1	2	0	−1	1	0	−1	2	1	−1
3	I	1	0	0	2	0	−2	0	0	0	1	1	−2
	II	1	0	0	2	1	−2	0	0	0	2	1	−2
	III	2	1	−1	1	0	−2	1	−1	−1	2	1	−2
4	I	1	0	−3	1	0	−3	0	0	−3	1	0	−3
	II	1	0	−2	2	0	−2	1	0	−2	2	0	−3
	III	2	0	−2	1	0	−2	0	0	−2	1	1	−2
5	I	1	1	−4	1	0	−4	0	0	−4	0	0	−4
	II	2	1	−3	1	−1	−3	0	0	−3	1	0	−4
	III	0	0	−3	1	−2	−4	−1	−1	−3	1	−1	−4

(d)

		Ansatz 1 für c_2^L			Ansatz 2 für c_2^L			Ansatz 3 für c_2^L			Ansatz 4 für c_2^L			Ansatz 5 für c_2^L		
K	L_i	0.01	0.1	0.2	0.01	0.1	0.2	0.01	0.1	0.2	0.01	0.1	0.2	0.01	0.1	0.2
2	I	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
	II	0	0	0	0	0	0	1	0	−1	1	0	0	2	0	−1
	III	0	0	0	0	0	0	1	0	0	0	0	0	2	0	−1
3	I	0	0	0	0	0	0	1	0	0	1	0	0	2	1	0
	II	0	0	0	1	0	0	0	−1	−1	1	0	0	3	0	−1
	III	0	0	0	0	0	−1	0	−2	−2	0	0	−1	3	−1	−2
4	I	2	2	0	2	0	0	2	0	0	0	0	0	3	1	0
	II	0	0	0	1	0	−1	0	−1	−2	0	0	−1	2	−1	−1
	III	0	0	0	0	−1	−1	0	−1	−2	1	−1	−1	2	0	−1
5	I	0	0	0	0	0	0	1	0	0	0	0	0	2	1	0
	II	0	0	0	1	0	−1	0	−2	−3	0	−1	−1	2	−2	−2
	III	0	0	0	1	−2	−2	0	−2	−3	0	−2	−2	2	−2	−2

		Ansatz 6 für c_2^R			Ansatz 7 für c_2^R			Ansatz 8 für c_2^R			Ansatz 9 für c_2^R		
K	L_i	0.0001	0.001	0.01	0.0001	0.001	0.01	0.0001	0.001	0.01	0.0001	0.01	0.1
2	I	1	0	0	2	1	0	0	0	0	2	1	0
	II	1	0	0	2	1	−1	0	0	0	2	2	−1
	III	0	0	0	1	0	−1	0	0	−1	2	1	−1
3	I	1	0	0	3	0	−2	0	0	0	2	1	−2
	II	2	0	0	2	1	−2	0	0	0	2	1	−2
	III	1	1	−1	2	0	−2	1	0	−1	2	1	−2
4	I	1	0	−3	1	0	−3	0	0	−3	1	0	−3
	II	2	1	−2	2	0	−2	0	0	−2	2	0	−2
	III	1	0	−1	2	0	−3	0	−1	−2	2	0	−3
5	I	1	1	−4	1	0	−4	0	0	−4	1	0	−4
	II	1	1	−3	1	−1	−3	0	0	−3	1	0	−4
	III	2	−1	−4	1	−1	−4	−1	−1	−4	1	0	−4

(e)

		Ansatz 1 für c_2^L			Ansatz 2 für c_2^L			Ansatz 3 für c_2^L			Ansatz 4 für c_2^L			Ansatz 5 für c_2^L		
K	L_i	0.01	0.1	0.2	0.01	0.1	0.2	0.01	0.1	0.2	0.01	0.1	0.2	0.01	0.1	0.2
2	I	1	0	0	0	0	0	1	0	0	2	0	0	3	2	0
	II	3	2	0	0	0	0	1	0	−1	1	0	0	2	0	−1
	III	0	0	0	0	0	0	0	0	0	1	0	0	2	1	−1
3	I	0	0	0	1	0	0	1	0	0	1	0	0	3	1	0
	II	0	0	0	0	0	0	0	−1	−2	0	0	0	2	0	−1
	III	0	0	0	0	0	0	0	0	−2	0	0	0	2	0	−1
4	I	0	0	0	1	0	0	1	0	0	1	0	0	3	1	0
	II	0	0	0	1	0	−1	0	−1	−2	0	0	−1	3	−1	−2
	III	0	0	0	0	−1	−1	0	−1	−1	1	−1	−1	3	0	−1
5	I	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	II	0	0	0	0	−1	−2	0	−2	−3	0	−1	−2	2	−2	−2
	III	0	0	0	1	−1	−2	0	−2	−3	0	−1	−2	1	−1	−2

		Ansatz 6 für c_2^R			Ansatz 7 für c_2^R			Ansatz 8 für c_2^R			Ansatz 9 für c_2^R		
K	L_i	0.0001	0.001	0.01	0.0001	0.001	0.01	0.0001	0.001	0.01	0.0001	0.01	0.1
2	I	3	0	0	2	1	0	1	0	0	3	2	0
	II	1	0	0	2	1	−1	0	0	0	2	1	−1
	III	0	0	−1	1	1	−1	0	1	−1	2	1	−1
3	I	1	0	0	2	1	−2	0	0	0	2	1	−2
	II	2	1	−1	2	0	−2	1	0	−1	2	1	−2
	III	1	1	−1	2	0	−2	0	−1	−1	2	1	−2
4	I	2	1	−3	1	0	−3	0	0	−3	1	0	−3
	II	2	1	−1	2	0	−2	1	0	−2	1	0	−2
	III	1	0	−1	1	0	−3	0	−1	−1	2	0	−3
5	I	1	1	−4	1	−1	−4	0	0	−4	0	0	−4
	II	1	1	−3	1	−2	−3	0	0	−3	1	−1	−4
	III	1	0	−3	1	−2	−4	−1	0	−3	1	−1	−4

Phase 3

Tabelle D3: Die Verdau- und Hybrid-Ansatz-basierten Lösungen θ^* bzw. $\hat{\theta}$ der Messdaten des Glykoprotein-Typs I.

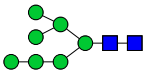
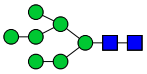
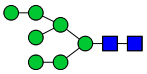
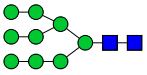
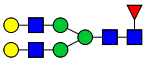
Glykane	Signal	k	θ^*			$\hat{\theta}$		
			π_k	μ_k	σ_k	π_k	μ_k	σ_k
 +  + 	(1)	1	0.28	301.54	0.84	0.28	301.54	0.85
		2	0.27	307.06	0.84	0.27	307.08	0.85
		3	0.44	310.37	0.84	0.44	310.37	0.83
	(2)	1	0.26	304.58	0.96	0.26	304.58	0.95
		2	0.27	310.57	0.92	0.27	310.57	0.92
		3	0.46	314.05	0.92	0.46	314.05	0.92
	(3)	1	0.28	303.07	0.93	0.28	303.07	0.93
		2	0.26	309.02	0.91	0.27	309.01	0.91
		3	0.46	312.53	0.92	0.46	312.53	0.92
	(4)	1	0.26	302.03	0.91	0.26	302.03	0.91
		2	0.27	307.95	0.90	0.27	307.95	0.91
		3	0.47	311.47	0.93	0.47	311.47	0.92
	(5)	1	0.27	302.02	0.90	0.27	302.02	0.91
		2	0.27	307.95	0.90	0.27	307.95	0.90
		3	0.46	311.45	0.90	0.46	311.45	0.90
	(6)	1	0.26	301.61	0.99	0.26	301.61	0.98
		2	0.26	307.62	0.91	0.26	307.62	0.91
		3	0.47	311.02	1.00	0.47	311.02	0.99
	(7)	1	0.26	302.39	1.01	0.26	302.41	0.99
		2	0.26	308.43	0.92	0.26	308.43	0.92
		3	0.47	311.83	1.03	0.47	311.83	1.03
	(8)	1	0.27	302.17	0.92	0.27	302.17	0.91
		2	0.27	308.10	0.90	0.27	308.10	0.90
		3	0.46	311.63	0.91	0.46	311.64	0.93
	(9)	1	0.27	301.75	0.91	0.27	301.75	0.91
		2	0.27	307.64	0.89	0.27	307.64	0.90
		3	0.46	311.16	0.92	0.46	311.16	0.92
	(10)	1	0.27	302.19	0.91	0.27	302.19	0.91
		2	0.27	308.12	0.88	0.27	308.12	0.89
		3	0.46	311.62	0.89	0.46	311.62	0.89
 + 	(1)	1	0.53	355.31	0.93	0.99	355.52	0.99
		2	0.46	355.82	1.02			
	(2)	1	0.57	362.32	1.01	0.48	362.25	1.00
		2	0.41	363.17	0.98	0.50	363.09	1.00
	(3)	1	0.68	360.82	1.05	0.94	361.05	1.00
		2	0.31	361.57	0.97			
	(4)	1	0.69	359.48	0.99	0.94	359.75	0.99
		2	0.31	360.52	1.00			
	(5)	1	0.57	359.31	0.90	0.62	359.44	0.99
		2	0.46	360.58	0.97	0.37	360.37	1.00
	(6)	1	0.78	359.05	1.02	0.96	359.20	0.99
		2	0.21	359.80	0.98			
	(7)	1	0.80	359.93	1.02	0.94	360.10	0.99
		2	0.20	360.95	0.97			
	(8)	1	0.67	359.68	0.95	0.93	359.99	0.99
		2	0.33	360.90	0.99			
	(9)	1	0.69	359.16	1.03	0.94	359.41	0.99
		2	0.30	360.03	0.98			
	(10)	1	0.72	359.73	0.99	0.95	359.99	0.99
		2	0.28	360.73	0.95			

Tabelle D4: Die Verdau- und Hybrid-Ansatz-basierten Lösungen θ^* bzw. $\hat{\theta}$ der Messdaten des Glykoprotein-Typs II.

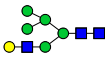
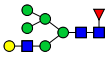
		θ^*			$\hat{\theta}$		
Glykane	k	π_k	μ_k	σ_k	π_k	μ_k	σ_k
? +  + ?	1	0.71	318.19	0.93	0.71	318.20	0.94
	2	0.05	321.43	1.22	0.05	321.59	0.94
	3	0.23	325.06	0.97	0.23	325.07	0.97
 + ? + ?	1	0.27	348.90	1.10	0.27	348.90	1.06
	2	0.27	352.79	0.96	0.12	352.53	0.98
	3	0.43	354.03	1.01	0.59	353.74	1.11

Tabelle D5: Die Verdau- und Hybrid-Ansatz-basierten Lösungen θ^* bzw. $\hat{\theta}$ der Messdaten des Glykoprotein-Typs III.

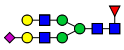
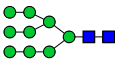
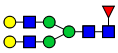
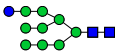
		θ^*			$\hat{\theta}$		
Glykane	k	π_k	μ_k	σ_k	π_k	μ_k	σ_k
 + 	1	0.70	248.10	0.95	0.90	248.11	0.83
	2	0.29	248.61	0.94			
 + 	1	1.0	359.44	0.97	0.98	359.52	1.00
	2	0.04	360.96	0.96			
 + ?	1	0.18	394.91	1.32	0.97	395.83	1.12
	2	0.85	396.00	1.06			

Tabelle D6: Die Verdau- und Hybrid-Ansatz-basierten Lösungen θ^* bzw. $\hat{\theta}$ der Messdaten des Glykoprotein-Typs IV.

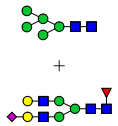
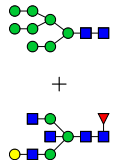
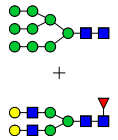
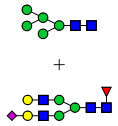
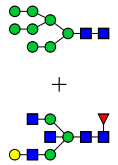
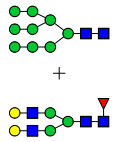
		θ^*			$\hat{\theta}$		
Glykane	k	π_k	μ_k	σ_k	π_k	μ_k	σ_k
	1	0.66	247.84	0.76	0.99	247.97	0.83
	2	0.33	248.37	0.93			
? + ?	1	0.14	330.01	1.36	0.95	330.63	0.95
	2	0.85	330.65	0.95			
	1	0.69	339.31	0.94	0.99	339.68	0.98
	2	0.30	340.00	0.94			
	1	0.32	359.49	1.16	1.00	360.35	1.09
	2	0.71	360.58	1.00			

Tabelle D7: Die Verdau- und Hybrid-Ansatz-basierten Lösungen θ^* bzw. $\hat{\theta}$ der Messdaten des Glykoprotein-Typs V.

		θ^*			$\hat{\theta}$		
Glykane	k	π_k	μ_k	σ_k	π_k	μ_k	σ_k
	1	0.77	247.12	0.73	0.98	247.06	0.83
	2	0.12	248.25	0.77			
	1	0.74	338.74	0.94	1.00	338.92	0.95
	2	0.28	339.43	0.92			
	1	0.21	358.66	0.93	1.00	359.54	1.07
	2	0.80	359.79	0.97			