

Direkte Analyse von Chlorophyllkataboliten

Vorwissenschaftliche Arbeit verfasst von

Florian Kluibenschedl

Klasse 8a



Betreuer: Mag. Mathias Scherl

BRG Telfs
Weißenbachgasse 37
6410 Telfs

Telfs, Februar 2018

Diese Arbeit wurde mit GNU Emacs geschrieben, in Palatino mit Hilfe von pdfLATEXe und Biber gesetzt.

Die LATEX Vorlage von Karl Voit basiert auf KOMA script und steht im Internet zum Download bereit: <https://github.com/novoid/LaTeX-KOMA-template>

Abstract

This is a placeholder for the abstract. It summarizes the whole thesis to give a very short overview. Usually, this the abstract is written when the whole thesis text is finished.

Vorwort und Danksagung

This is a placeholder for the abstract. It summarizes the whole thesis to give a very short overview. Usually, this the abstract is written when the whole thesis text is finished.

Inhaltsverzeichnis

Abstract

Vorwort und Danksagung

Allgemeiner Teil	1
1. Themenstellung	2
2. Das Chlorophyll und sein Abbauprozess	3
2.1. Der Abbauprozess und seine Bedeutung	3
2.2. Nummerierung von Phyllobilinen	4
3. Methoden	5
3.1. HPLC	5
3.2. Massenspektrometrie	6
3.2.1. LC-MS	7
3.2.2. MS Leafspray	7
Experimenteller Teil	9
4. Allgemeine Arbeits- und Analysemethoden	10
4.1. Herstellung von Lösungen für eine Analyse mit HPLC	10
4.2. Chemikalien und Material	10
4.3. Probenmaterial	10
4.4. Reaktion mit Essigsäureanhydrid	10
4.5. Erstellen von Fragmentierungsdiagrammen	11
5. Experimente MS-Leafspray	12
5.1. Gerätebeschreibung Massenspektrometer	12
5.2. Versuchsaufbau	12
5.3. Versuchsdurchführung	14

Inhaltsverzeichnis

5.4.	Chl-Kataboliten des Brokkoliblattes mithilfe von MS-Leafspray identifiziert	15
5.4.1.	Bo-NCC-1	15
5.4.2.	Bo-NCC-3	17
5.4.3.	Bo-DNCC	20
5.5.	Identifikation der Reaktionsprodukte	22
5.5.1.	Reaktionsprodukt von Bo-DNCC	22
5.5.2.	Reaktionsprodukt von Bo-NCC-3	26
5.5.3.	Reaktionsprodukt von Bo-NCC-1	28
6.	Experimente LC-MS	31
6.1.	HPLC-Gradient sowie Gerätebeschreibung	31
6.2.	Aufarbeitung der Probe	31
6.3.	Theoretische Auswertung der Online-UV/Vis Spektren	32
6.4.	Theoretische Auswertung der MS Spektren	32
6.5.	Chl-Kataboliten des Brokkoliblattes mithilfe von LC-MS identifiziert	33
6.6.	Identifikation der Reaktionsprodukte	39
7.	Strukturaufklärung der Chl-Kataboliten mit ESI-MS	45
7.1.	Beschreibung der Methode	45
7.2.	Identifizierte Chl-Kataboliten	45
7.2.1.	Bo-DYCC	45
7.2.2.	Bo-DNCC	48
7.2.3.	Bo-YCC	50
7.2.4.	Bo-NCC-3	51
7.2.5.	Bo-NCC-1	53
7.3.	Reaktionsprodukte der Chl-Kataboliten	54
7.3.1.	Reaktionsprodukte von Bo-DYCC	54
7.3.2.	Reaktionsprodukt von Bo-DNCC	56
7.3.3.	Reaktionsprodukt von Bo-YCC	57
7.3.4.	Reaktionsprodukt von Bo-NCC-1	57
7.3.5.	Reaktionsprodukt von Bo-NCC-3	58
Ergebnisse und Diskussion	59	
Verweise	60	
Literaturverzeichnis	61	

Inhaltsverzeichnis

Abbildungsverzeichnis	63
Abkürzungsverzeichnis	66
Anhang	67
A. Python Code-Listings für die Erstellung der Grafiken	68
A.1. Programm zur Erstellung von Chromatogrammen für LC-ESI-MS	68
A.2. Programm zur Erstellung von MS-Spektren	78
A.3. Programm zur Erstellung von Fragmentierungsdiagrammen . . .	97

Allgemeiner Teil

1. Themenstellung

Das Ziel der vorliegenden Arbeit besteht darin, die Chlorophyll Kataboliten (Chl-Katabolit) von Brokkoliblättern einer direkten massenspektrometrischen Analyse zu unterziehen und sie anhand unterschiedlicher Merkmale zu untersuchen.

Es sollte dabei untersucht werden, inwieweit eine direkte Analyse mit der noch relativ neuen Methode des MS Leafspray eine strukturelle Aufklärung von Chl-Kataboliten ermöglicht, sowie, ob das Stattfinden einer Reaktion von Essigsäureanhydrid mit den Chl-Kataboliten festgestellt werden kann.

Die Reaktion selbst sollte Aufschluss über die Reaktivitäten unterschiedlicher Carbonsäuren der Chl-Kataboliten und deren struktureller Besonderheiten geben.

Mithilfe von im Collision induced Dissociation (CID) Modus erstellten Fragmentierungsdiagrammen wird versucht, eventuelle charakteristische Eigenschaften der Chl-Kataboliten ausfindig zu machen. Diese Eigenschaften können sich zum Beispiel (z.B.) im Vergleich der aufgenommenen Diagramme zwischen den Chl-Kataboliten ergeben aber auch im intramolekularen Bereich.

Es soll somit das Potential von Fragmentierungsdiagrammen in Hinblick auf zukünftige Strukturaufklärung mithilfe eines Massenspektrometers analysiert werden.

Um die Ergebnisse von MS Leafspray zu überprüfen wurde die Methode der High performance liquid chromatography (HPLC) sowie ein hochauflösendes Massenspektrometer verwendet. Mit den im hochauflösenden Massenspektrometer erhaltenen Fragmentierungen wird versucht, diese mit den von MS Leafspray erhaltenen zu vergleichen, um einen Vergleich der beiden Methoden abgeben zu können.

Eine Analyse mit diesen Methoden setzt die Herstellung eines Blattextraktes voraus und verdient deswegen nicht mehr die Bezeichnung der direkten Analyse.

2. Das Chlorophyll und sein Abbauprozess

2.1. Der Abbauprozess und seine Bedeutung

Jedes Jahr gehen weltweit schätzungsweise 10^9 Tonnen an Chlorophyll scheinbar verloren. Der Abbauprozess des Chlorophylls ist damit ob der markanten Farbveränderungen einer der visuell am meisten wahrgenommenen biochemischen Vorgänge und kann sogar aus dem All beobachtet werden. [1] Die schönen, bunten Farben des Laubs werden dabei jedoch nicht durch die Abbauprodukte des Chlorophylls (im Folgenden Chlorophyllkataboliten) hervorgerufen [2], da die Endprodukte des Chlorophyllabbaus farblos sind [1]. Im Folgenden ist immer die Rede vom Abbauprozess in höheren Pflanzen, da gezeigt wurde, dass z.B. marine Lebensformen das Chlorophyll auf einem anderen Wege abbauen und auch dementsprechend andere Endprodukte vorzufinden sind. [1], [3], [4] Die Abbauprodukte fallen in die Klasse der Phyllobiline (bilinartige, heterocyclische Pyrrole) und sind Anzeichen für Reifung, Seneszenz und den Zelltod. Der Abbauprozess wird im Rahmen eines Entgiftungsprozesses begangen. [5]

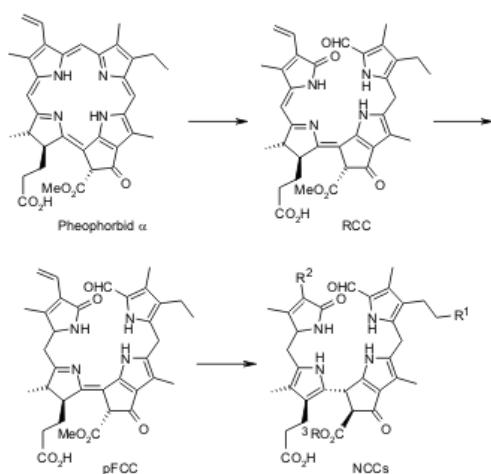


Abbildung 2.1.: Der Abbauprozess des Chlorophylls in seneszenten Blättern

2. Das Chlorophyll und sein Abbauprozess

Die Struktur eines Chlorophyllkataboliten konnte erstmalig im Jahre 1991 aufgeklärt werden. Es handelte sich hierbei um einen Hv-Non floorescent Chlorophylllic Catabolite (NCC) der Gerste (*Hordeum vulgare*) [3], das Endprodukt eines mehrstufigen Abbauprozesses.

In den folgenden Jahren fand man heraus, dass das Chlorophyll zuerst in das Pheophorbid a umgewandelt wird. Im nächsten Schritt wird der Makrozyklus oxygenolytisch (an der Reaktion beteiligtes Enzym: Pheo α Mono-oxygenase [6]) in der nördlichen *meso* Position geöffnet, woraufhin ein *Red Chlorophyll Catabolite* (RCC) entsteht. Über einen *primary "floorescent" Chlorophyll Catabolite* (pFCC) entsteht durch eine nichtenzymatische Isomerisierung ein NCC. Thermodynamische Triebkraft dieser Reaktion ist die Rearomatisierung von Ring D. [7], [8] Die unterschiedlichen Arten von NCCs ergeben sich durch Anlagerung der entsprechenden funktionellen Gruppen (z.B. Zuckerring, Hydroxylgruppen, ...) an den pFCC. [8] In Abbildung 2.2 sind die Positionen, an denen diese strukturellen Unterschiede auftreten können gekennzeichnet durch R¹, R², R³.

2.2. Nummerierung von Phyllobilinen

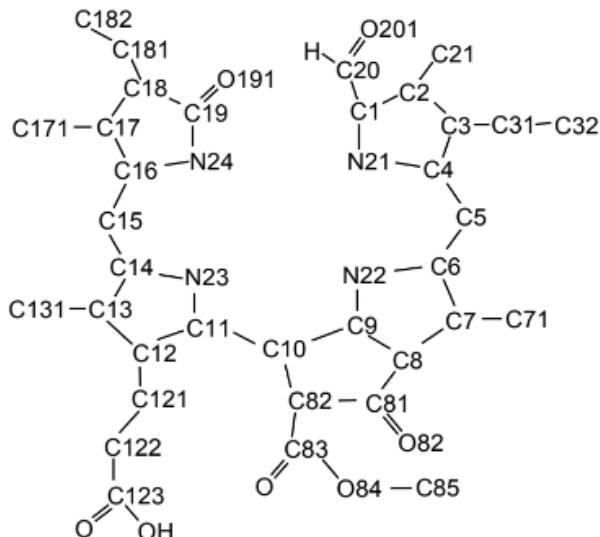


Abbildung 2.2.: Die Positionsangaben und Bezeichnung der Ringe, die in der restlichen Arbeit für Ausführungen sowie Erklärungen verwendet werden.

3. Methoden

Es folgen theoretische Erklärungen der Methoden, derer ich mich im Rahmen meiner praktischen Arbeit bediente.

3.1. HPLC

Die HPLC ist eine chromatographische Methode, um lösliche Stoffe präparativ zu trennen. Es sind dabei quantitative als auch qualitative Analysen möglich. [9, S. 165]

Das Trennen der Stoffe basiert auf ihren unterschiedlichen chemischen Eigenschaften. Die Stoffe werden gelöst und bilden zusammen mit dem Lösungsmittel (LM) die mobile Phase, die an einer, sich in der Trennsäule befindenden stationären Phase vorbeiströmt, wobei es dabei zu Wechselwirkungen zwischen den gelösten Stoffen mit der stationären Phase kommt. Aufgrund der unterschiedlichen chemischen Eigenschaften und den daraus resultierenden unterschiedlichen Wechselwirkungen hält sich jeder Stoff verschieden lange in der stationären Phase auf. Die Verweildauer eines Stoffes in der Trennsäule wird als Retentionszeit bezeichnet. [9, S. 31-32] Die Retentionszeit wird über Detektoren bestimmt, die die Änderung der Zusammensetzung der mobilen Phase feststellen und das Ergebnis in einem Chromatogramm darstellen. [9, S. 46]

Für die Experimente wurde die Methode der Reversed-phase (RP) Chromatographie angewandt. Dabei ist die mobile Phase polar und die stationäre Phase unpolär (als unpolare Phase dienen beispielsweise Silane mit langen Kohlenwasserstoffketten). [9, S. 189]

Im Rahmen meiner Arbeit wurde die HPLC verwendet, um die Stoffe im Blatt zu trennen und entsprechende Chlorophyllkataboliten zu isolieren. Die Identifikation der Chl-Katabolit erfolgte dabei durch einen UV/VIS Detektor (operierte im Wellenlängenbereich von 200nm - 800nm) sowie durch ein dazu geschaltetes Massenspektrometer (=Liquid Chromatography-Mass Spectrometry (LC-MS)).

3. Methoden

Um die Chl-Kataboliten mit einem hochauflösenden Massenspektrometer zu fragmentieren, wurden die Verbindungen zu jenen Zeiten, zu denen sie in der HPLC jeweils eluierten in Eppendorf Reaktionsgefäß (EPPI) gesammelt.

Die Herstellung eines Blattextraktes für die Analyse mit der HPLC wird in Kapitel 6.2 beschrieben. Aufgrund dieser speziellen Aufarbeitung des Blattes zählt die Methode der HPLC nicht mehr zur direkten Analyse. Sie wurde in Kombination mit einem hochauflösenden Massenspektrometer deswegen verwendet, um die Ergebnisse von MS Leafspray zu verifizieren.

3.2. Massenspektrometrie

Mithilfe eines Massenspektrometers kann die Masse eines Moleküls bestimmt werden. Aufgrund der Einfachheit der Methode und der sehr geringen benötigten Probenmenge ist das Massenspektrometer für eine Vielzahl an Anwendungen geeignet (z.B. in der Forensik, Lebensmittelprüfung, Medikamentenprüfung, Analyse von Meteoriten, ...). [10, S. 1] Der jetzige Entwicklungsstand in der Massenspektrometrie ist vor allem den Entwicklungen in den letzten vier Jahrzehnten auf diesem Gebiet zu verdanken. [10, S. 6-9]

Um die Molekulmasse der Stoffe zu bestimmen, werden sie zuerst in Gasphasen-Ionen überführt. [10, S. 15] Dabei gibt es unterschiedliche Methoden, diesen Zustand herbeizuführen, wie z.B. Electron Ionization (EI), Chemical Ionization (CI) und Field Ionization (FI). [10, S. 15-30] Nach der Ionisation werden sie im Massenanalysator nach ihrem Masse pro Ladung (m/z) Verhältnis getrennt und im Detektor der Ionenstrom gemessen. Das Ergebnis wird in einem Massenspektrum festgehalten, in dem auf der Ordinate die relative Intensität der einzelnen Peaks und auf der Abszisse das Verhältnis m/z aufgetragen werden.

In den Experimenten dieser Arbeit wurde zur Ionisation die Electrosprayionisation (ESI) Methode verwendet, die erstmalig das Messen von Proteinen mithilfe eines Massenspektrometers erlaubte und aufgrund ihrer hohen Empfindlichkeit gegenüber kleinen, polaren Molekülen mit einer HPLC kombiniert werden kann. Bei der ESI Methode wird durch Anlegen einer Spannung von 3-6kV zwischen der Kapillare, aus der die Flüssigkeit kommt und der Gegenelektrode ein elektrisches Feld mit einer Stärke in der Größenordnung von 10^6 Vm^{-1} angelegt. Die erhaltenen geladenen Tröpfchen passieren dann ein Inertgas (in den

3. Methoden

meisten Fällen Stickstoff (N₂)) beziehungsweise (bzw.) eine erhitzte Kapillare, um das LM zu entfernen. [10, S. 43-44]

Um die Chlorophyllkataboliten im Massenspektrometer zu analysieren wurde sowohl die Methode der LC-MS als auch die Methode des MS-Leafspray verwendet.

3.2.1. LC-MS

Bei der Methode der LC-MS wird eine HPLC vor ein Massenspektrometer geschaltet. Dabei trennt die HPLC die Stoffe zuvor auf und eluiert sie anschließend in das Massenspektrometer. [10, S. 217-218] Um die Flussrate bei atmosphärischem Druck zu verringern, wird nur ein Teil des direkt aus der HPLC kommenden Fluxes zum Massenspektrometer hin abgezweigt. Ansonsten wäre die Flussrate zu hoch, was eine Ionisierung der Probe mithilfe einer ESI - Quelle unmöglich machen würde. [10, S. 221]

Das Resultat ist je ein Chromatogramm der HPLC und des Massenspektrometers. Es wird somit zu jedem Zeitpunkt eines HPLC-Laufes ein UV/VIS Spektrum sowie ein Massenspektrum erzeugt. Aus dem UV/VIS Spektrum lässt sich schließen, ob es sich bei einem Chl-Katalyten um einen NCC, Decarboxylated-Non fluorescent Chlorophylllic Catabolite (DNCC) oder einen Fluorescent Chlorophylllic Catabolite (YCC) handelt. Aus dem Massenspektrum kann die Molekülmasse (in Da) mit allgegenwärtigen Fragmentierungen abgelesen werden. Unter Verwendung eines hochauflösenden Massenspektrometers wird außerdem die atomare Zusammensetzung in Form einer Summenformel ersichtlich.

3.2.2. MS Leafspray

Ambient Ionization [11] ermöglicht es, Proben ohne vorherige präparative Trennung durch chromatographische Trennverfahren direkt in ihrer *natürlichen* Umgebung mithilfe eines Massenspektrometers zu untersuchen. Eine Methode, die auf dem Prinzip der *Ambient Ionization* basiert ist *Paper Spray* [12]. Dabei kommt es zu einer Kombination der ESI sowie der *Ambient Ionisationsmethode*. [12]

3. Methoden

Die Ionisation der Probe erfolgt ausgehend von einem feuchten, porösen Material (z.B. Papier), das in einer Kupferklemme eingeklemmt wird. Zwischen der Kapillaröffnung des Massenspektrometers und der Kupferklemme liegt eine Spannung im Bereich von 3-6kV an, woraufhin kleine Tröpfchen, die Ionen der Probe enthalten von der Spitze des porösen Materials ausgesendet werden und Ionen der Probe in das Massenspektrometer befördern. [13] Durch Anlegen von Kalibrationskurven mit externen Standards wird außerdem ermöglicht, eine quantitative Bestimmung der Menge des Analyten durchzuführen. [14]

Leaf Spray ist eine Form von Paper Spray, bei der die zu analysierende Pflanze selbst als poröses Material dient und wurde im Rahmen dieser Arbeit für die Identifikation von Chl-Katabolit verwendet. Der Vorteil einer Analyse von Chl-Katabolit mit MS Leafspray ist, dass weniger Zeit für die Vorbereitung benötigt wird, was wiederum Grundlage für eine schnellere und effizientere Analyse ist. Genauere Ausführungen finden sich in Kapitel 5.

Experimenteller Teil

4. Allgemeine Arbeits- und Analysemethoden

4.1. Herstellung von Lösungen für eine Analyse mit HPLC

- Herstellung eines 3.2 mM NH₄Ac Puffers:
Es wurden 0.1542 g Ammoniumacetat abgewogen und in 0.5 L Wasser gelöst.

4.2. Chemikalien und Material

4.3. Probenmaterial

Als Probe dienten Brokkoliblätter der Brokkoliflanze aus dem Garten meiner Oma (Grundfeld Telfs, Austria). Die Brokkoliblätter wurden im Zeitraum von 03.08.2017 bis zum 13.09.2017 (die letzten Brokkoliblätter wurden am 11.09.2017 gesammelt) jeweils in der früh zwischen 07:20 und 07:30 gesammelt. Der Zeitraum, in dem die Brokkoliblätter für Analysen verwendet wurden beträgt maximal drei Tage (danach wurden neue Brokkoliblätter gesammelt).

4.4. Reaktion mit Essigsäureanhydrid

Im Rahmen der direkten Analyse der Chl-Kataboliten wurden diese einer Reaktion mit Essigsäureanhydrid unterzogen. Die Reaktion wurde dabei direkt am Blatt durchgeführt. Dazu wurden in einem Exikator getrocknete Blätter auf eine Fläche von circa (ca.) 2cm² (entspricht einer ungefähren Blattmasse

4. Allgemeine Arbeits- und Analysemethoden

von 0.01g) mit einer Rasierklinge zugeschnitten und anschließend mit 300µL Essigsäureanhydrid beidseitig benetzt. Die 300µL ergaben sich, da das Blatt nicht in der Reagenz schwimmen sollte, da sonst Teile des Blattes aus diesem herausgelöst werden, was zu einer Verringerung der Intensitäten in HPLC und MS Leafspray führt.

Die Reaktion wurde durch eine Vakuumpumpe mit zwischengeschalteter Kühlfalla gestoppt (Verweildauer in der Vakuumpumpe betrug ca. 5min). Dabei wurde die verbleibende Reagenz abgesaugt. Das Blatt wurde dann entsprechend der folgenden Analysenmethode entsprechend aufgearbeitet.

Es wurden Reaktionszeiten zwischen 7min. und 6h ausprobiert. Für die Präsentation und Diskussion der MS Leafspray Versuche wurden die Reaktionsprodukte nach 22min. Reaktionsdauer verwendet, wohingegen für die Präsentation der HPLC Versuche die Reaktionsprodukte nach 3h Reaktionsdauer.

4.5. Erstellen von Fragmentierungsdiagrammen

Zu den Chl-Kataboliten wurden Fragmentierungsdiagramme erstellt. Dazu werden die Intensitäten der einzelnen beobachteten Fragmentierungen im Massenspektrometer (aufgenommen wurde im CID Modus) zur jeweiligen aufgewendeten, normalisierten Kollisionsenergie (in %) (NKE) aufgenommen.

Auf der Abszisse des erhaltenen Diagramms befindet sich die NKE (von 0 bis 100 NKE, gemessen wurde in 5er Schritten) und auf der Ordinate die Intensität der jeweiligen Fragmentierung bezogen auf den höchsten Peak, der während der Aufnahme beobachtet wurde (ebenfalls in Prozent).

Die erhaltenen Kurven wurden mit einem Savitzky-Golay Filter [15] geglättet (Programmcode Anhang) und werden im folgenden als Fragmentierungsdiagramme bezeichnet. Ein Nachteil bei der Behandlung mit diesem Filter ist, dass in manchen Fällen die Intensitäten der Fragmentierungen bei einer NKE von null nicht gleich null sind. Es wird im folgenden angenommen, dass dies dennoch so ist.

Mit den Fragmentierungsdiagrammen wird versucht, herauszufinden, ob bestimmte Abspaltungen der Kataboliten charakteristische Muster aufweisen. Dieses Wissen könnte für eine zukünftige Strukturaufklärung sowie für die wünschenswerte Schnellidentifikation von Chl-Katalyse mittels MS Leafspray von Relevanz sein. Ebenso wird eine Verbesserung von MS Leafspray erhofft.

5. Experimente MS-Leafspray

5.1. Gerätebeschreibung Massenspektrometer

(Beschreibung Massenspektrometer)

5.2. Versuchsaufbau

Abbildung 5.1 beschreibt schematisch den Versuchsaufbau.

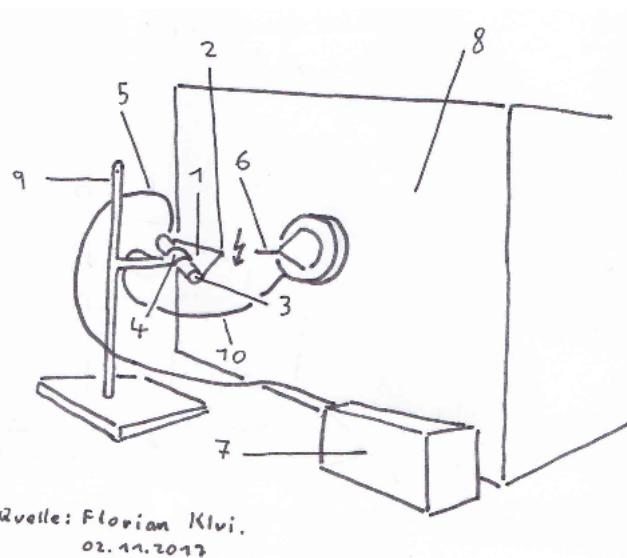


Abbildung 5.1.: Leafspray Versuchsaufbau: 1) Filterpapierdreick, 2) Spitze des Dreiecks, 3) Blattmaterial, von Filterpapier umschlossen, 4) Kupferklemme, 5) Kapillare für LM, 6) Einlass des Massenspektrometers (mit der markanten Spitze zwecks Verdeutlichung etwas übertrieben dargestellt), 7) Syringe Pump - kontrolliert den LM-Fluss durch 5), 8) DESI Massenspektrometer, 9) Stativ, 10) Kabel, mit 4) verbunden - zwischen 4) und 6) liegt eine Spannung an (3-6kV - durch Blitz zwischen 2) und 6) symbolisiert)

5. Experimente MS-Leafspray

Das zu analysierende Blatt wurde zugeschnitten und in Filterpapier eingerollt. Das Filterpapierdreieck wurde in einer Kupferklemme eingespannt (Kapitel 5.3). Die Kupferklemme wurde mit einem Kabel (10), das an einem DESI-Massenspektrometer (8) angeschlossen war, verbunden. Zwischen der Kupferklemme (4) und dem Massenspektrometer wurde eine Spannung von 3-6kV angelegt. Da das Filterpapier mit LM benetzt ist und eine Verbindung der Flüssigkeit zur Kupferklemme besteht, kommt es zu einer durch die Spannung ausgelösten Bewegung der im LM gelösten Ionen, die nun in das Massenspektrometer hineinfliegen. Der Abstand zwischen Filterpapier (2) und Einlass des Massenspektrometers (6) betrug ungefähr 0.5cm und ist damit der Flugstrecke der Ionen in natürlicher Umgebung gleichzusetzen.



Abbildung 5.2.: (a) Einlass des Massenspektrometers mit Kapillare (5), Kupferklemme (4) und Filterpapier mit Blattmaterial (1) und (3), (b) Detailansicht

In Abbildung 5.2 wird gezeigt, wie diese Anordnung umgesetzt wurde. Zu sehen sind die Kupferklemme mit dem eingespannten Filterpapier und dem darin enthaltenen Blatt, die LM-Kapillare, die Einlassöffnung des Massenspektrometers und der Abstand von Filterpapierdreicksspitze zum Massenspektrometer. Es gilt zu beachten, dass das Filterpapier in einem gewissen Winkel eingespannt wurde (Abbildung 5.2b), um zu verhindern, dass das LM nicht abfließt, was bei einer waagrechten Anordnung in Form einer *Sackbildung* des LM auftreten kann.

5. Experimente MS-Leafspray

5.3. Versuchsdurchführung

Frische, senescente Brokkoliblätter und Filterpapier wurden mit Rasierklinge und Schere wie in Abbildung 5.3a ersichtlich zugeschnitten. Anschließend wurde das Brokkoliblatt auf das Filterpapier gelegt und dieses bis zur Basis des Dreiecks eingerollt.

Diese Art der Vorbereitung zeigte sich als besonders effektiv, da mit ihr höhere Intensitäten der Signale im Massenspektrometer erreicht werden konnten, wie wenn nur das Blatt zu einem Dreieck zugeschnitten und in dieser Form vor das Massenspektrometer gehalten wird. Grund dafür ist vermutlich, dass das LM mehr Zeit hat, die Chlorophyllkataboliten aus dem Blatt heraus zu lösen und dass mehr Blattmaterial vorhanden ist. Außerdem behält das Filterpapier länger seine Steifigkeit wie ein Brokkoliblatt, weswegen längere Analysen mit konstanterem Signal möglich sind.

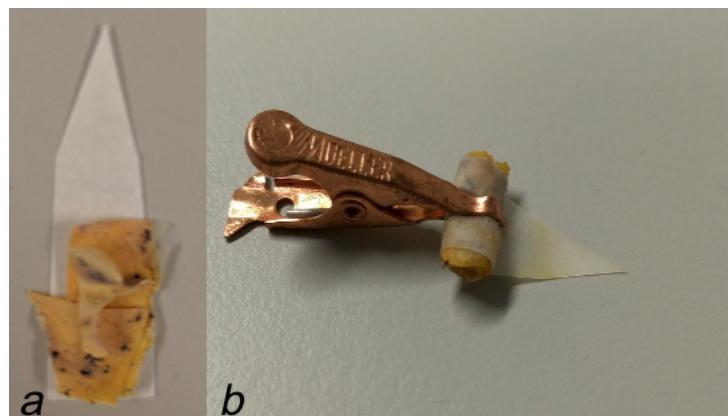


Abbildung 5.3.: MS Leafspray Blattvorbereitung: (a) zugeschnittenes Filterpapierdreieck, mit frischen, seneszenten Brokkoliblättern, (b) eingerolltes *Päckchen*, in Kupferklemme eingespannt

Das erhaltene *Päckchen* wurde durch eine Kupferklemme (Abbildung 5.3b) ca. 0.5cm vor die Kapillare des Massenspektrometers gehalten (Abbildung 5.2b). Um ein konstantes Signal zu erhalten versorgte eine Kapillare, die wie in Abbildung 5.2a befestigt war, das *Päckchen* mit einem konstanten LM-Fluss (als LM wurden Methanol (MeOH) sowie Acetonitril verwendet). Die Flussrate des LM betrug zu Beginn $12\mu\text{L min}^{-1}$, um das Blatt schneller zu befeuchten und wurde ab dem Erhalt des ersten Signals auf $5\mu\text{L min}^{-1}$ zurückgefahren. Es zeigte sich, dass bei dieser Flussrate das Signal bei annähernd gleichbleibend hoher Intensität am längsten bleibt. Der Spraystrom betrug zwischen xx und xx μA . Diese Geräteeinstellungen sowie Aufarbeitungsmethoden ermöglichen das Messen

5. Experimente MS-Leafspray

der Fragmentierungsdiagramme, da hierfür eine längere Analysezeit vonnöten ist. Das erste Signal konnte nach ca. einer Minute nach dem Einschalten der Spannung gemessen werden und erlaubte sinnvolle Messungen bis zu 25min.

(Platz für Beschreibung der Einstellungen des Gerätes)

Aufgenommen wurden die Massenspektren im Bereich von 300 m/z bis 1000 m/z, um Massenspektren zu bekommen, die nicht so stark von anderen Ionenarten gestört werden. Gemessen wurde im positiven Ionenmodus, wobei zwischendurch in den negativen Ionenmodus geschaltet wurde, wenn die Intensität des Signals im positiven Ionenmodus abnahm. Das Wechseln des Modus konnte die gewünschte Intensität wieder erhöhen. Es wurde somit ein ähnliches Verhalten der Intensitäten im Zeitverlauf beobachtet wie in [13] bereits beschrieben, wobei hier das Umschalten in den negativen Ionenmodus nicht explizit erwähnt wird, um das Problem der abnehmenden Intensitäten im positiven Ionenmodus zu beheben.

5.4. Chl-Kataboliten des Brokkoliblattes mithilfe von MS-Leafspray identifiziert

Im Folgenden werden die Chlorophyllkataboliten beschrieben, die sich durch MS-Leafspray identifizieren ließen. Die Strukturvorschläge wurden durch ein hochauflösende Massenspektrometer überprüft (Kapitel 6.5 und 7). Sie beruhen auf den exakten Molekülmassen und den daraus errechneten möglichen Summenformeln. Eine exakte Strukturaufklärung müsste mit ¹H-NMR durchgeführt werden.

Fragmentierungsdiagramme wurden wie in Kapitel 4.5 beschrieben erstellt.

5.4.1. Bo-NCC-1

Bei diesem Kataboliten handelt es sich vermutlich um denselben, wie er auch in der Brokkolifrucht gefunden wurde, weswegen er die Bezeichnung Bo-NCC-1 erhält. [16] Gefunden wurde die protonierte Verbindung bei m/z = 793 [M+H]⁺ und das Kaliumsalz bei m/z = 831 [M+K]⁺ (Abbildung 5.4). Aufgrund der geringen Intensitäten der protonierten Verbindung war es nicht möglich, ein

5. Experimente MS-Leafspray

verwertbares Massenspektrum dieser aufzunehmen.

Der Katabolit bei $m/z = 831 [M+K]^+$ zeigte Abspaltungen von H_2O bei $m/z = 813 [M - H_2O + K]^+$, von CO_2 bei $m/z = 787 [M - CO_2 + K]^+$ und eine Folge von Abspaltungen bei $m/z = 311 [M - (Ring A, Zucker, Ring D, CO_2) + K]^+$, bei der Ring A mit einem Zucker, Ring D sowie CO_2 abgespalten wird (siehe Kapitel, Katabolit 619, hochauflösende Massenspektrometrie). Die Abspaltungen bei $m/z = 798 [M - (noch Aufklärungsbedarf (nAb.) + K]^+$, $m/z = 586 [M - (nAb.) + K]^+$ und $m/z = 551 [M - (nAb.) + K]^+$ können nicht eindeutig zugeordnet werden, da hierzu weitere experimentelle Daten und vor allem die exakten Molekülmassen vonnöten sind.

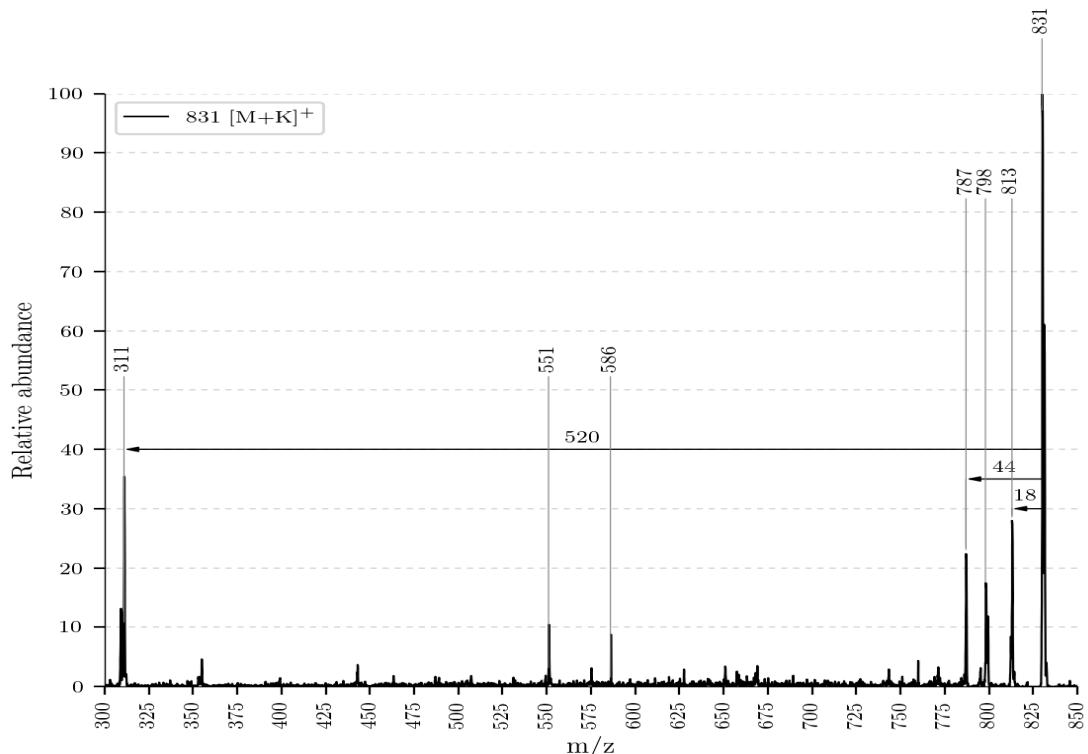


Abbildung 5.4.: ESI-MS von Bo-NCC-1 mit $m/z = 831 [M+K]^+$

Das Fragment bei $m/z = 798 [M - (MeOH?) + K]^+$ ist insofern interessant, da es sich hierbei um eine Abspaltung von MeOH (-32 Da) handeln könnte (es wird angenommen, dass die Abweichung um eins durch Ungenauigkeiten des Massenspektrometers zustandekommt), was aber nicht mit der Struktur des Bo-NCC-1 (siehe Abbildung 5.5a) vereinbar wäre. Aufgrund ihrer Fragwürdigkeit

5. Experimente MS-Leafspray

wird auf diese Abspaltung in den weiteren Ausführungen nicht näher eingegangen.

Wie aus dem Fragmentierungsdiagramm (Abbildung 5.5b) ersichtlich, erfolgt die Abspaltung von H_2O bei einer niedrigeren NKE wie jene von CO_2 und verschwindet bei höheren Energien, wohingegen die Abspaltung von CO_2 erhalten bleibt. Die Abspaltung von H_2O erreicht ein lokales Maximum bei einer NKE von 10. Die Abspaltung von CO_2 erreicht ein lokales Maximum bei 30 NKE.

Aufgrund der CO_2 Abspaltung wird an Position 8² eine Carbonsäuregruppe vermutet (wie in [17] gezeigt), die über einen Mechanismus wie unter anderem (u.a.) in Abbildung 7.5a vorgeschlagen, abgespalten wird. Die relativ große Molekülmasse weist zudem auf einen Zucker an Position 32 hin. Die Summenformel des Bo-NCC-1 konnte über die exakte Molekülmasse mit einem hochauflösenden Massenspektrometer bestimmt werden (Kapitel 7.2.5).

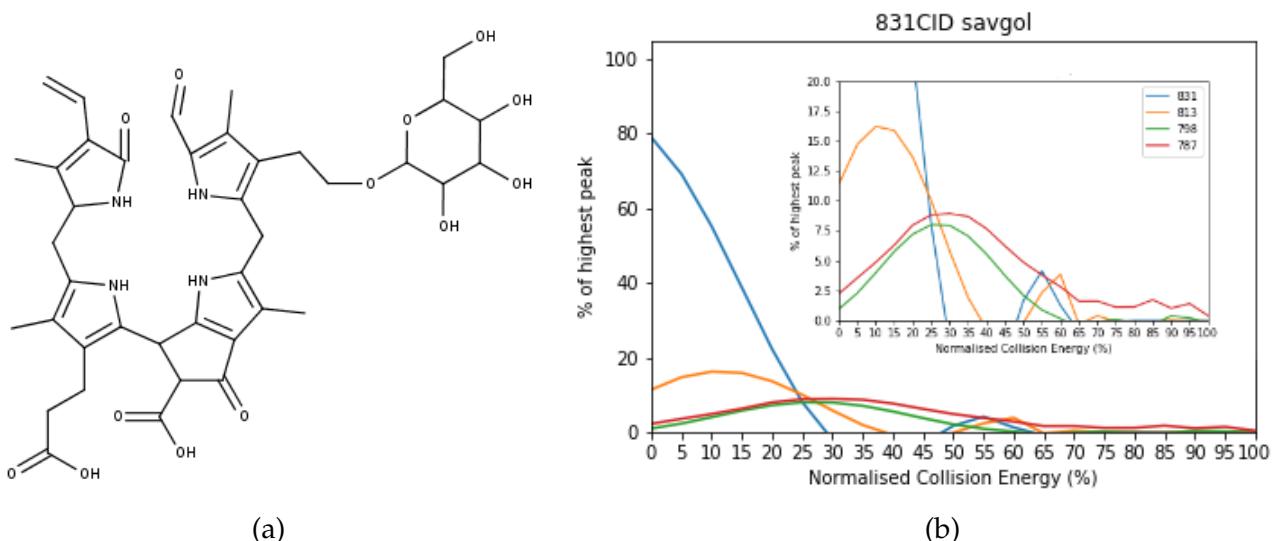


Abbildung 5.5.: (a) Strukturvorschlag des Bo-NCC-1 mit Summenformel $\text{C}_{40}\text{H}_{48}\text{N}_4\text{O}_{13}$, (b) Fragmentierungsdiagramm von Bo-NCC-1 (blau = 831 [$\text{M}+\text{K}^+$], orange = 813 [$\text{M}-\text{H}_2\text{O} + \text{K}^+$], grün = 798 [$\text{M} - (\text{MeOH} - \text{nAb.}) + \text{K}^+$], rot = 787 [$\text{M}-\text{CO}_2 + \text{K}^+$])

5.4.2. Bo-NCC-3

Beim Bo-NCC-3 handelt es sich um einen Chl-Katabolit, der bisher nicht in der Brokkolifrukt identifiziert wurde, weswegen er als dritter [16], in der

5. Experimente MS-Leafspray

Brokkolipflanze gefundener Katabolit den Index 3 erhält. Analysiert wurde das Kaliumsalz mit $m/z = 685 [M+K]^+$.

Es wurden zwei charakteristische Abspaltungen von H_2O bei $m/z = 667 [M - H_2O + K]^+$ sowie von CO_2 bei $m/z = 641 [M - CO_2 + K]^+$ beobachtet. Bei den Abspaltungen bei $m/z = 429 [M - (nAb.) + K]^+$, $m/z = 561 [M - (nAb.) + K]^+$, $m/z = 605 [M - (nAb.) + K]^+$ und $m/z = 652 [M - (nAb.) + K]^+$ ist nicht eindeutig geklärt, welche Fragmente hierbei entstanden sind. Für das Fragment bei $m/z = 652 [M - (MeOH?) + K]^+$ gilt dasselbe wie bei der Abspaltung von $m/z = 798 [M - (MeOH?) + K]^+$ von Bo-NCC-1 (Kapitel 5.4.1). Um diese Fragmente aufzuklären müssten weitere Experimente des Kaliumsalzes mit einem hochauflösenden Massenspektrometer durchgeführt werden. Fragmentierungen der protonierten Verbindung konnten mit einem hochauflösenden Massenspektrometer gemessen werden (Kapitel 7.2.4).

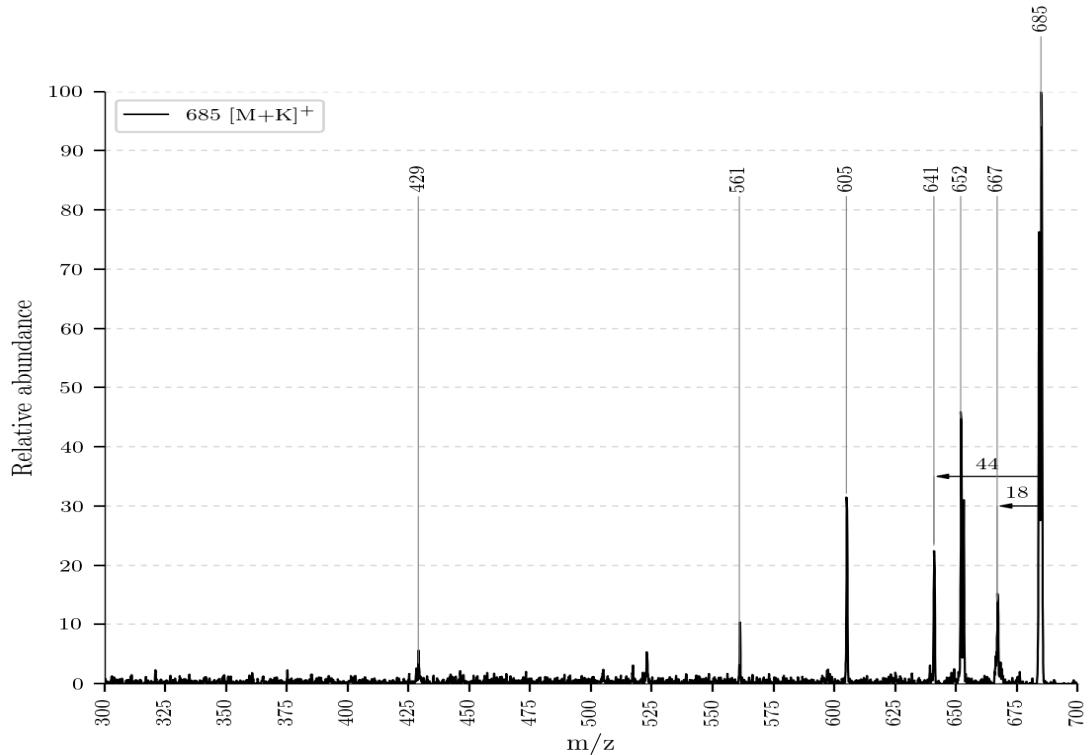


Abbildung 5.6.: ESI-MS von Bo-NCC-3 mit $m/z = 685 [M+K]^+$

Das Fragmentierungsdiagramm zeigt, dass die Abspaltung von H_2O bei einer niedrigeren NKE erfolgt, wie jene von CO_2 , da sie ihre höchste Intensität zuvor

5. Experimente MS-Leafspray

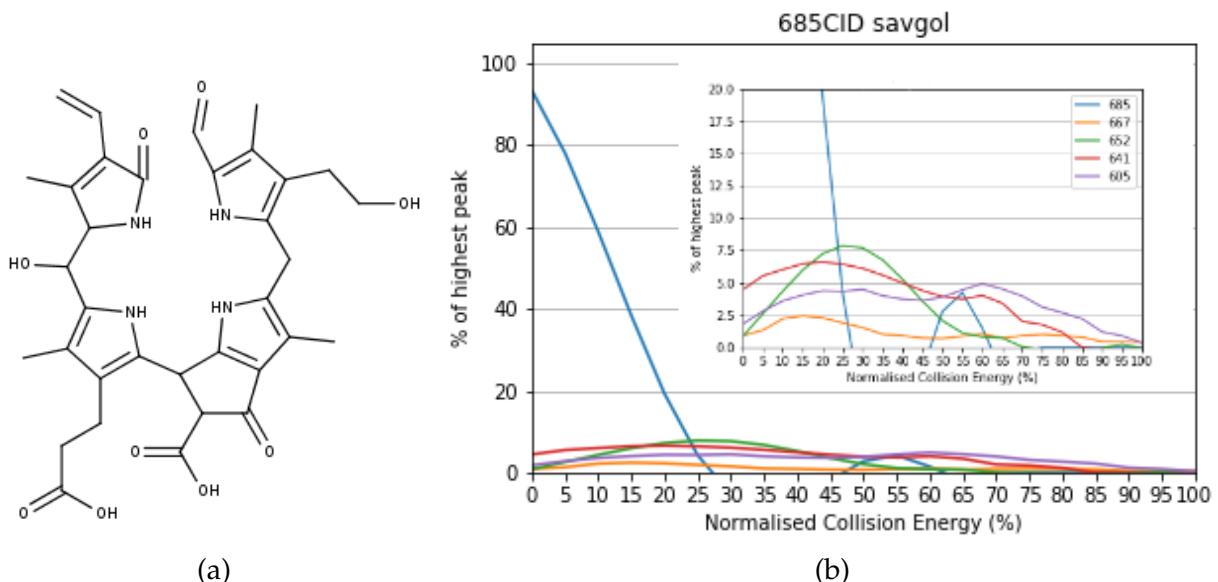


Abbildung 5.7.: (a) Strukturvorschlag von Bo-NCC-3 mit Summenformel $C_{34}H_{38}N_4O_9$, (b) Fragmentierungsdiagramm von Bo-NCC-3 (blau = $685 [M+K]^+$, orange = $667 [M - H_2O + K]^+$, grün = $652 [M - (MeOH?)] + K^+$, rot = $641 [M - CO_2 + K]^+$, violett = $605 [M - (nAb.) + K]^+$)

erreicht (bei einer NKE von 15 - H_2O im Vergleich zu 20 - CO_2).

Im Vergleich zum Bo-NCC-1 zeigt der Graph ein lokales Maximum der H_2O Abspaltung bei höheren Energien (beim Bo-NCC-3 bei 15 NKE wohingegen beim Bo-NCC-1 bereits bei 10 NKE). Das lokale Maximum der CO_2 Abspaltung verschiebt sich von 30 NKE beim Bo-NCC-1 auf 20 NKE beim Bo-NCC-3. Das lokale Maximum der potentiellen Abspaltung von MeOH würde sich von 25 NKE beim Bo-NCC-1 auf 30 NKE beim Bo-NCC-3 verschieben (Abbildungen 5.5b und 5.7b).

Wie beim Bo-NCC-1 weist die CO_2 Abspaltung auf eine freie Carbonsäure an Position 8² hin. Aufgrund der durch die Summenformel erhaltene Sauerstoffanzahl wird angenommen, dass sich an Position 15 eine Hydroxygruppe befindet (Abbildung 5.7a). Es wird vermutet, dass es sich dabei um eine Vorstufe zu einem YCC handelt. [Referenz]

5. Experimente MS-Leafspray

5.4.3. Bo-DNCC

Es wird vermutet, dass der Bo-DNCC des Brokkoliblattes ident ist mit dem Bo-DNCC der Brokkolifrukt. [16] Beobachtet wurden zwei Pseudo-Molekulare Ionen. Eines mit $m/z = 619 [M+H]^+$ (Abbildung 5.8) und mit $m/z = 657 [M+K]^+$ (Abbildung 5.9).

Der Katabolit bei $m/z = 619 [M+H]^+$ zeigte Abspaltungen von H_2O bei $m/z = 601 [M - H_2O + H]^+$, von CO_2 bei $m/z = 575 [M - H_2O + H]^+$, von Ring D (zusammen mit einer Abspaltung von CO_2) bei $m/z = 452 [M - (Ring D, CO_2) + H]^+$ und von Ring A, Ring D und CO_2 bei $m/z = 311 [M - (Ring A, Ring D, CO_2) + H]^+$ (Abbildung 5.8 - Zuordnung Kapitel 7.2.2).

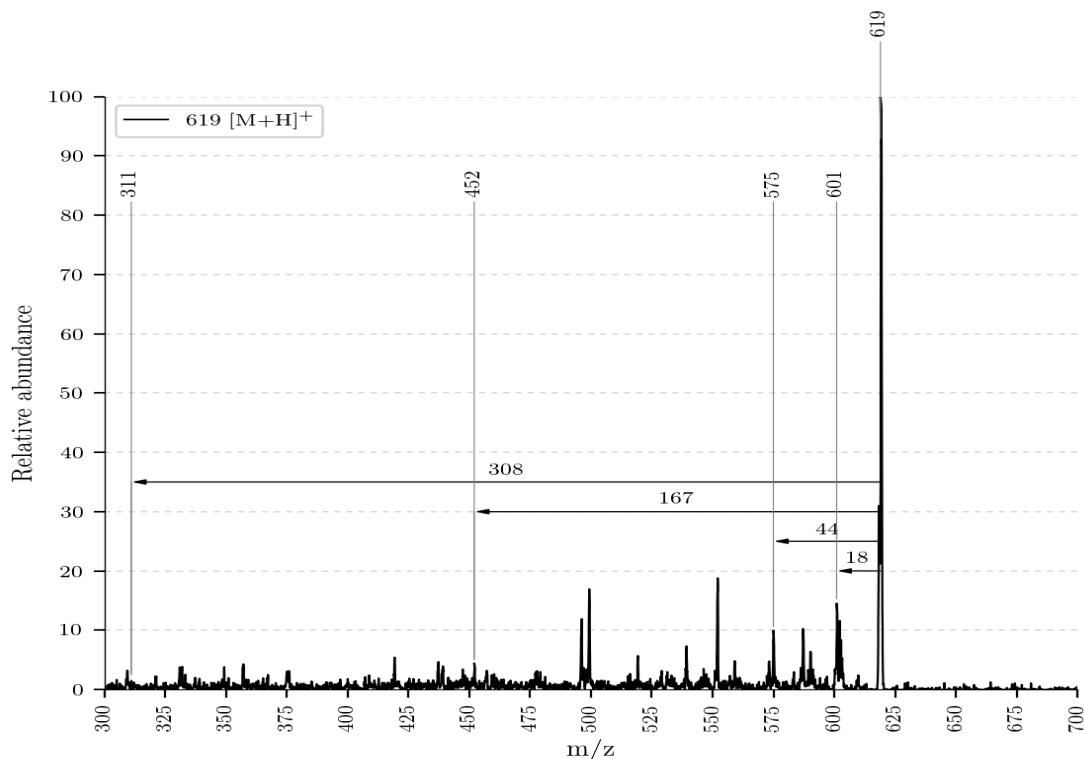


Abbildung 5.8.: ESI-MS von Bo-DNCC bei $m/z = 619 [M+H]^+$

Das Kaliumsalz des Bo-DNCC mit $m/z = 657 [M+K]^+$ zeigte eindeutige Abspaltungen von H_2O bei $m/z = 639 [M - H_2O + K]^+$ und von CO_2 bei $m/z = 613 [M - CO_2 + K]^+$ (Abbildung 5.9). Die Abspaltungen bei $m/z = 375 [M - (nAb.) + K]^+$ und $m/z = 577 [M - (nAb.) + K]^+$ können nicht eindeutig zugeordnet werden.

5. Experimente MS-Leafspray

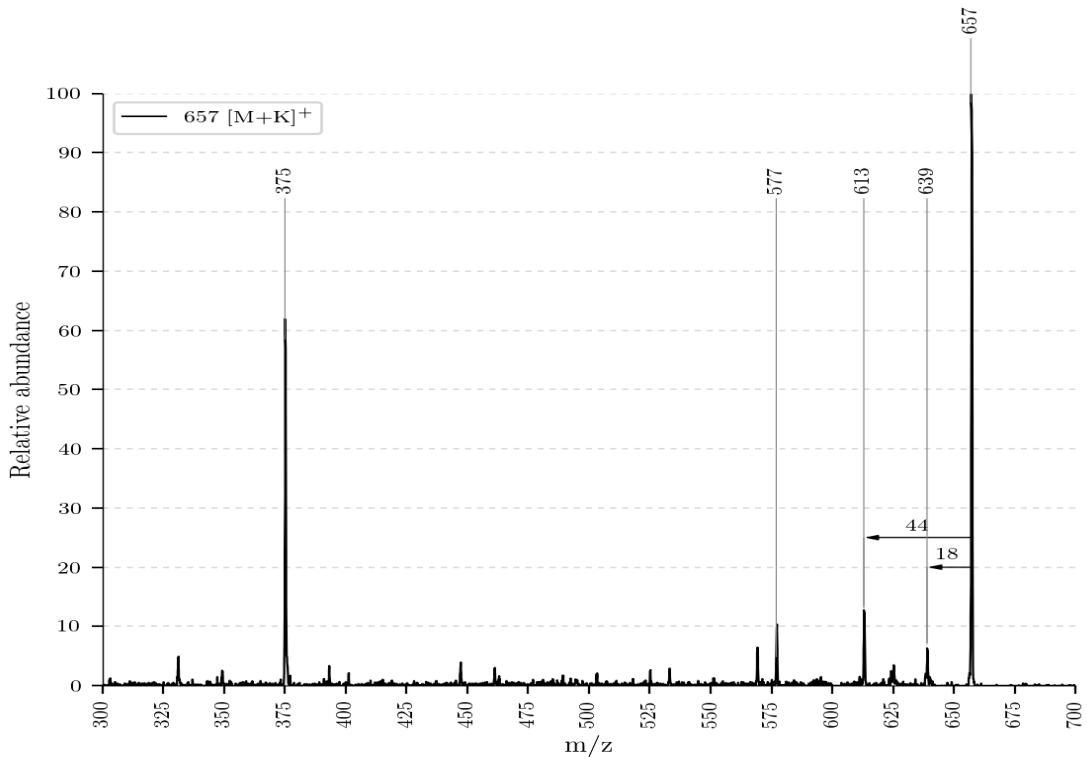


Abbildung 5.9.: ESI-MS von Bo-DNCC bei $m/z = 657$ $[M+K]^+$

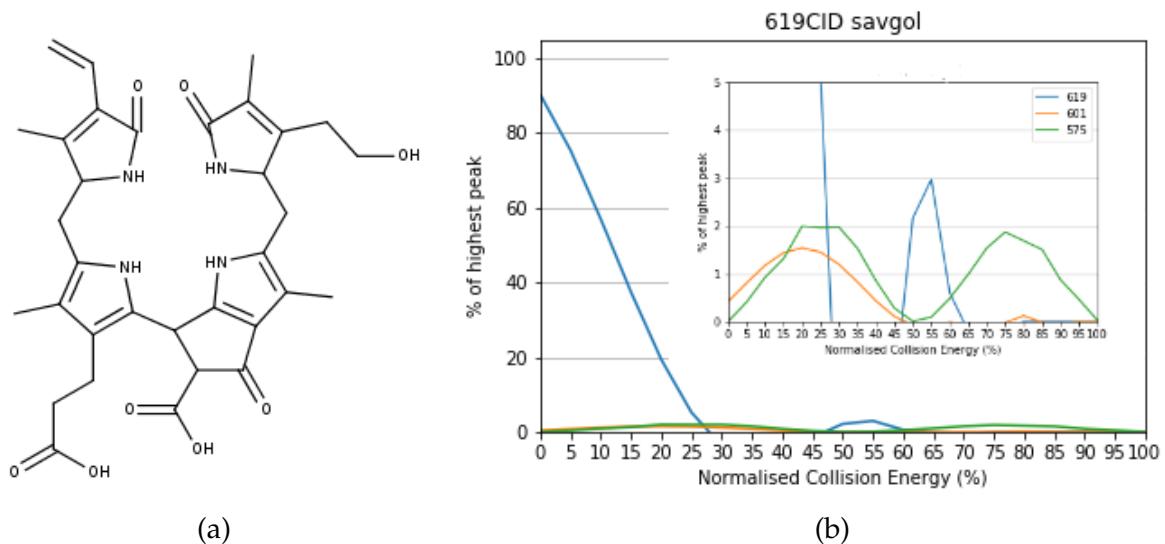


Abbildung 5.10.: (a) Strukturvorschlag von Bo-DNCC mit Summenformel $C_{33}H_{38}N_4O_8$, (b) Fragmentierungsdiagramm von Bo-DNCC (blau = 619 $[M+H]^+$, orange = 601 $[M - H_2O + H]^+$, grün = 575 $[M - H_2O + H]^+$)

5. Experimente MS-Leafspray

Die H_2O Abspaltung beim Bo-DNCC erreicht ein lokales Maximum bei 20 NKE und erfolgt damit im Vergleich zum Bo-NCC-1 und Bo-NCC-3 bei der höchsten NKE. Die Abspaltung von CO_2 weist beim Bo-DNCC zwei lokale Maxima, bei 25 NKE und 75 NKE auf. Das lokale Maximum an der Stelle 75 NKE ist dabei etwas weniger intensiv ausgeprägt wie jenes an der Stelle 25 NKE. Das erste lokale Maximum befindet sich damit an der gleichen Stelle wie bei Bo-NCC-1 und Bo-NCC-3 (Abbildungen 5.5b und 5.7b). Das zweite Maximum kann noch nicht geklärt werden, da es bei den anderen bisher analysierten Kataboliten nicht beobachtet wurde.

5.5. Identifikation der Reaktionsprodukte

Für den Nachweis des Stattfindens der Reaktion der Chl-Kataboliten mit Essigsäureanhydrid, wurde der gleiche Versuchsaufbau wie in Kapitel 5.2 beschrieben, verwendet. Das Anhydrid als Reaktionsprodukt konnte durch Verwendung von Acetonitril als LM isoliert werden. Um eine bessere Identifikation der Reaktionsprodukte zu erreichen, wurden Fragmentierungsdiagramme erstellt.

5.5.1. Reaktionsprodukt von Bo-DNCC

Das Reaktionsprodukt von Bo-DNCC konnte mit $m/z = 699 [\text{M}+\text{K}]^+$ bestimmt werden (Strukturvorschlag - Abbildung 5.11). Identifiziert wurde es über die charakteristische Abspaltung von Essigsäure ($M = 60 \text{ Da}$) bei $m/z = 639 [\text{M} - \text{CH}_3\text{COOH} + \text{K}]^+$. Ein Mechanismus für die Abspaltung wird in Abbildung 5.14a vorgeschlagen. Dieser Mechanismus ähnelt dem Mechanismus der Abspaltung von MeOH (z.B. beobachtbar bei einem Cj-NCC), wie in [17] publiziert.

Es wurden Abspaltungen von H_2O bei $m/z = 681 [\text{M} - \text{H}_2\text{O} + \text{K}]^+$, von CH_3COOH bei $m/z = 639 [\text{M} - \text{CH}_3\text{COOH} + \text{K}]^+$ und von Ring A und Ring D mit CO_2 bei $m/z = 311 [\text{M} - (\text{Ring A, Ring D, CO}_2) + \text{K}]^+$ beobachtet.

Zur Identifikation der Reaktionsprodukte wurde die CH_3COOH Abspaltung aufgrund ihrer Dominanz und Eindeutigkeit herangezogen (u.a. Abbildung 5.13b). Das Fragment bei $m/z = 599 [\text{M} - (\text{nAb.}) + \text{K}]^+$ ist interessant, da eine Abspaltung von 100 Da bei anderen Kataboliten ebenfalls beobachtet wurde. Die anderen Fragmentierungen in Abbildung 5.12 konnten nicht zugeordnet werden.

5. Experimente MS-Leafspray

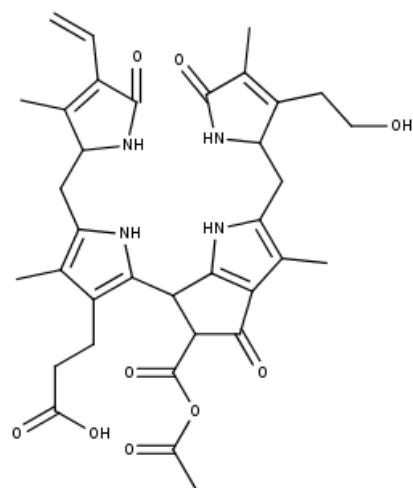


Abbildung 5.11.: Strukturvorschlag des Reaktionsproduktes mit Summenformel $C_{33}H_{40}N_4O_9$

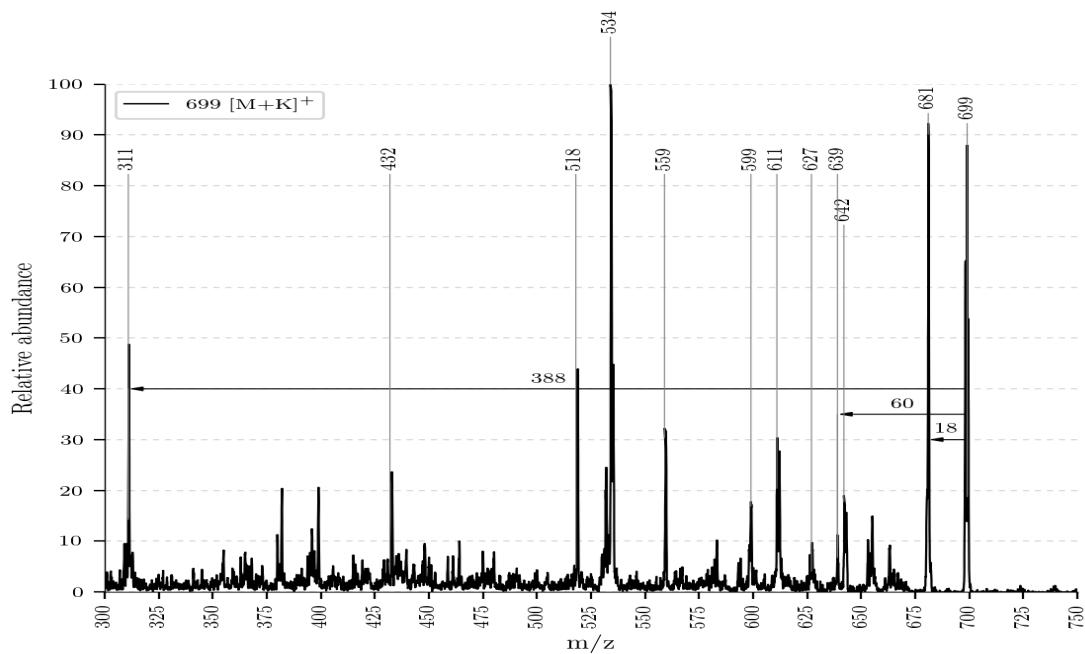


Abbildung 5.12.: ESI-MS Spektrum des Reaktionsproduktes mit $m/z = 699 [M+K]^+$

5. Experimente MS-Leafspray

Diskussion der Abspaltung bei $m/z = 599 [M - (nAb.) + K]^+$: Die Abspaltung von 100 Da bei $m/z = 599 [M - (nAb.) + K]^+$ erreicht im Fragmentierungsdiagramm lokale Maxima bei 15 NKE und 30 NKE. Lokale Minima befinden sich bei 17 NKE und 40 NKE, an jenen Stellen, an der die Abspaltung von CH_3COOH lokale Maxima aufweist (Abbildung 5.13b). Daraus könnte man Informationen über den Mechanismus der Abspaltung ableiten. Man könnte sagen, dass die Abspaltung von 100 Da einhergeht mit jener von CH_3COOH und dass sie mechanistisch miteinander verknüpft sind, also, dass bevor einer Abspaltung des Fragments mit 100 Da CH_3COOH abgespalten werden muss. Es ließe sich damit erklären, warum bei einem Maximum der einen Abspaltung die andere Abspaltung ein Minimum aufweist.

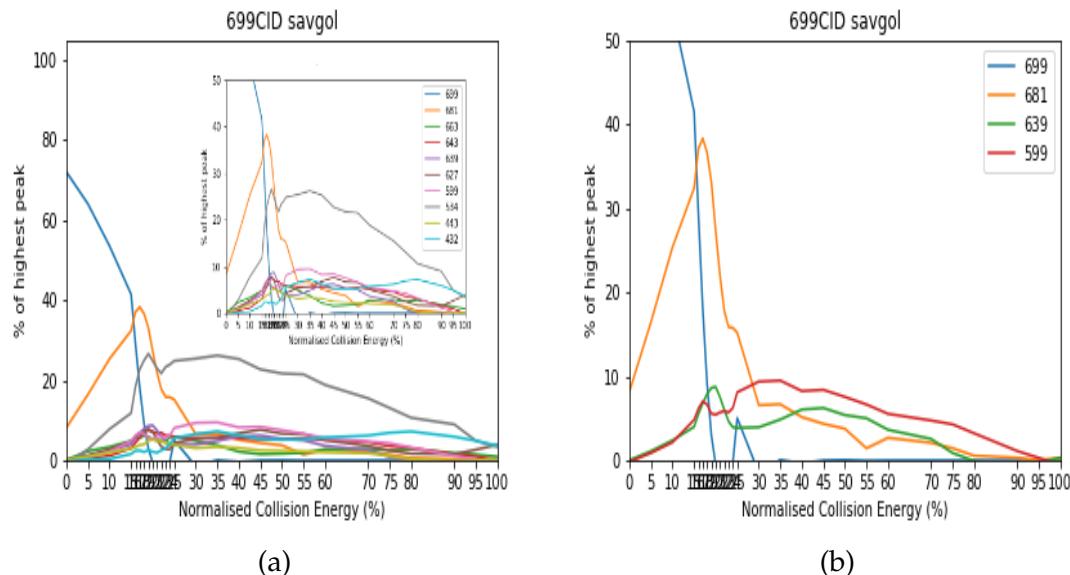


Abbildung 5.13.: (a) Fragmentierungsdiagramm des Bo-NCC-3 mit allen beobachteten Abspaltungen (blau = $699 [\text{M}+\text{K}]^+$, orange = $681 [\text{M} - \text{H}_2\text{O} + \text{K}]^+$, grün = $663 [\text{M} - (2\text{xH}_2\text{O}) + \text{K}]^+$, rot = $643 [\text{M} - (\text{nAb.}) + \text{K}]^+$, violett = $639 [\text{M} - \text{CH}_3\text{COOH} + \text{K}]^+$, braun = $627 [\text{M} - (\text{nAb.}) + \text{K}]^+$, pink = $599 [\text{M} - (\text{nAb.}) + \text{K}]^+$, grau = $534 [\text{M} - (\text{nAb.}) + \text{K}]^+$, hellgrün = $443 [\text{M} - (\text{nAb.}) + \text{K}]^+$, türkis = $432 [\text{M} - (\text{nAb.}) + \text{K}]^+$), (b) Fragmentierungsdiagramm mit ausgewählten Abspaltungen (blau = $699 [\text{M}+\text{K}]^+$, orange = $681 [\text{M} - \text{H}_2\text{O} + \text{K} - \text{H}_2\text{O}]^+$, grün = $639 [\text{M} - \text{CH}_3\text{COOH} + \text{K}]^+$, rot = $599 [\text{M} - (\text{nAb.}) + \text{K}]^+$)

Im Fragmentierungsdiagramm erreicht die H_2O Abspaltung ein lokales Maximum bei 17 NKE. Die Abspaltung nimmt bis zu 30 NKE stark ab und bleibt bis zu 90 NKE erhalten. Im Vergleich zum Fragmentierungsdiagramm des nicht reagierten Bo-DNCC erfolgt die H_2O Abspaltung bei einer niedrigeren NKE

5. Experimente MS-Leafspray

und ist länger beobachtbar (vergleiche Abbildungen 5.10b und 5.13b). Es gilt zu bedenken, dass beim nicht reagierten Bo-DNCC das $[M+H]^+$ -Ion aufgenommen wurde, wohingegen man beim reagierten Bo-DNCC das $[M+K]^+$ -Ion analysierte. Der Unterschied im Verlauf der Kurven könnte somit auch durch diesen Umstand hervorgerufen werden.

Die Abspaltung von CH_3COOH besitzt lokale Maxima bei 20 NKE und 45 NKE. Das Maximum bei 45 NKE ist weniger intensiv. Die Intensität der Abspaltung nimmt dabei kontinuierlich bis zu einer von 80 NKE ab (Abbildung 5.13b). Ein lokales Minimum der Abspaltung befindet sich zwischen 23 NKE und 30 NKE.

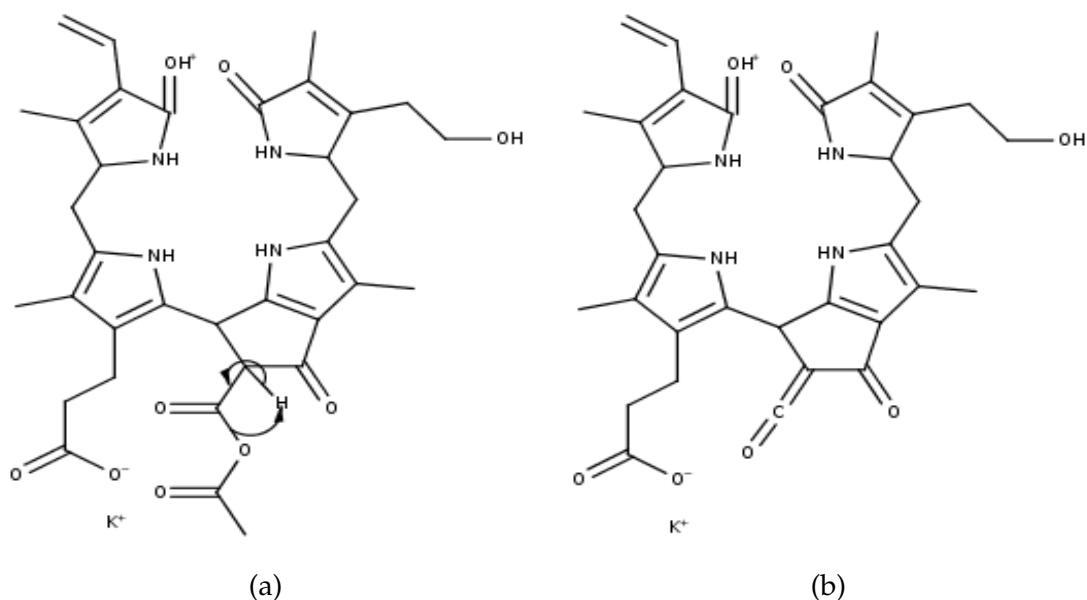


Abbildung 5.14.: (a) vorgeschlagener Mechanismus der Essigsäureabspaltung und (b) das Produkt, wobei CH_3COOH als stabiles Neutralteilchen abgespalten wird

5. Experimente MS-Leafspray

5.5.2. Reaktionsprodukt von Bo-NCC-3

Die Molekülmasse des Reaktionsproduktes von Bo-NCC-3 konnte mit $m/z = 727 [M+K]^+$ bestimmt werden. Eine Abspaltung von Essigsäure wurde bei $m/z = 667 [M+K]^+$ beobachtet. Weiters wurde eine Abspaltung von H_2O bei $m/z = 709 [M - H_2O + K]^+$ beobachtet. Bei der Abspaltung bei $m/z = 627 [M - (nAb.) + K]^+$ könnte es sich um die gleiche Abspaltung wie beim Reaktionsprodukt des Bo-DNCC handeln, da auch ein Fragment mit $M = 100$ Da abgespalten wird. Die anderen Abspaltungen (Abbildung 5.15) konnten nicht eindeutig zugeordnet werden.

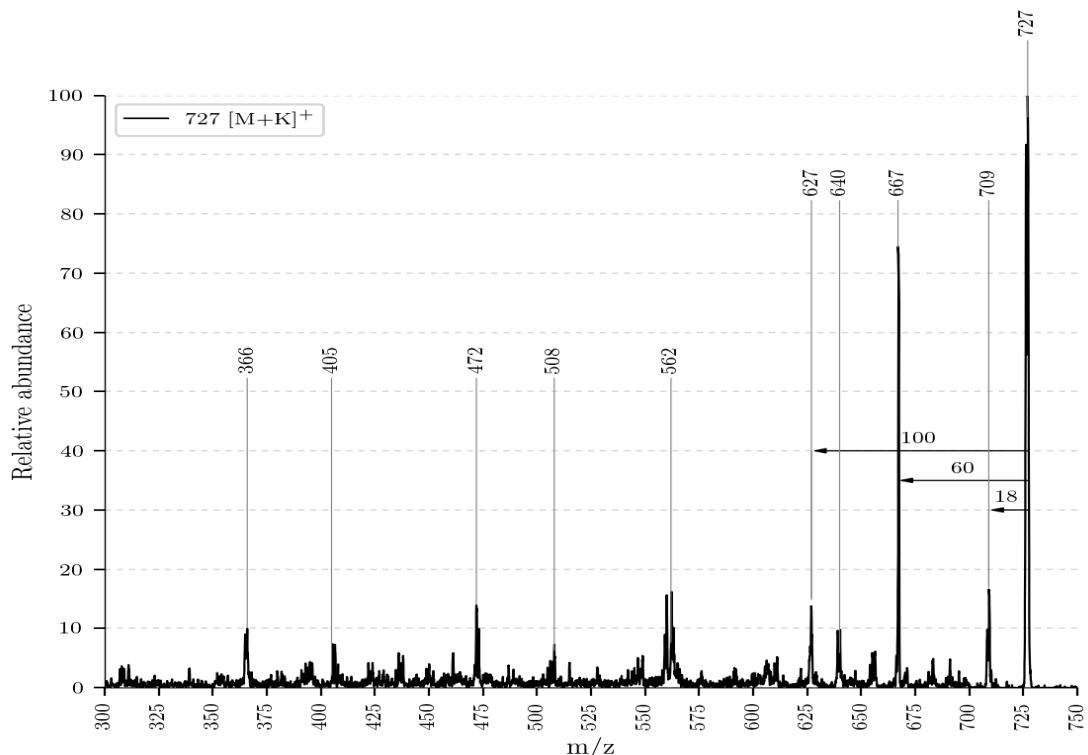


Abbildung 5.15.: ESI-MS Spektrum des Reaktionsproduktes bei $m/z = 727 [M+K]^+$

Es wurde beobachtet, dass die Abspaltung von H_2O bei niedrigeren Energien erfolgt wie jene von CH_3COOH . Im Vergleich zum Fragmentierungsdiagramm des Reaktionsproduktes des Bo-DNCC (Abbildung 5.13a) kann als Charakteristikum der CH_3COOH Abspaltung ein lokales Maximum bei 45 NKE gedeutet werden (Abbildung 5.13b und Abbildung 5.17). Die Abspaltung von H_2O weist bei beiden Kataboliten ein lokales Maximum bei 15 NKE auf und besitzt einen ähnlichen Kurvenverlauf (Abbildung 5.13b und Abbildung 5.17). Dies lässt

5. Experimente MS-Leafspray

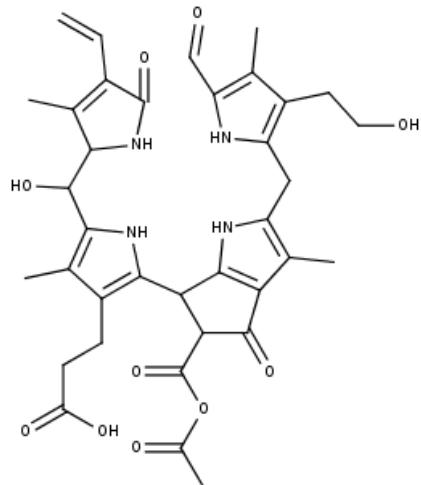


Abbildung 5.16.: Strukturvorschlag des Reaktionsproduktes mit Summenformel $C_{36}H_{40}N_4O_{10}$

darauf schließen, dass es sich bei dieser H_2O -Abspaltung um eine Abspaltung auf ein und derselben Position handelt. Als Position der Abspaltung wird die Hydroxygruppe an Position 32 des Chl-Kataboliten vorgeschlagen.

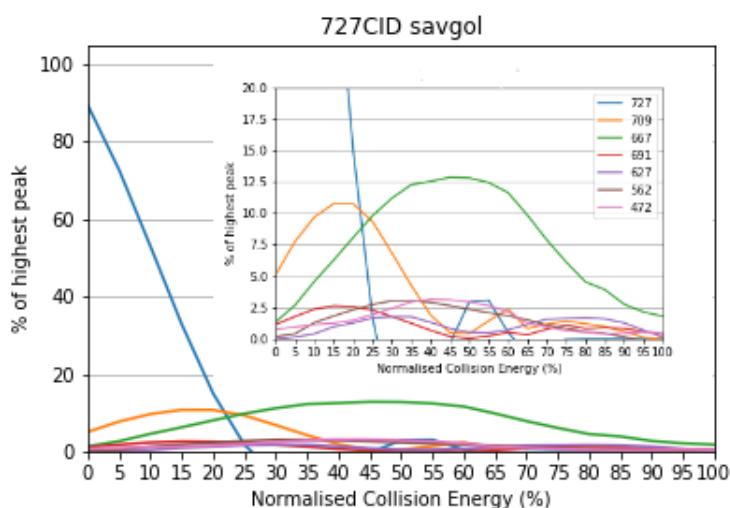


Abbildung 5.17.: Fragmentierungsdiagramm des Reaktionsproduktes (lbau = 727 $[M+K]^+$, orange = 709 $[M - H_2O + K]^+$, grün = 667 $[M - CH_3COOH + K]^+$, rot = 691 $[M - ? + K]^+$, violett = 627 $[M - ? + K]^+$, braun = 562 $[M - ? + K]^+$, pink = 472 $[M - ? + K]^+$)

5. Experimente MS-Leafspray

5.5.3. Reaktionsprodukt von Bo-NCC-1

Erwartungsgemäß konnte das Reaktionsprodukt des Bo-NCC-1 bei $m/z = 873 [M+K]^+$ gefunden werden. Es zeigt Abspaltungen von H_2O bei $m/z = 855 [M - H_2O + K]^+$, von Essigsäure bei $m/z = 813 [M - CH_3COOH + K]^+$ und von CH_3COOH , Ring A, Ring D, zweimal MeOH und CO bei $m/z = 309 [M - (Ring A, Ring D, 2mal MeOH, CO) + K]^+$ (diesselbe Abspaltung wurde beim Reaktionsprodukt $m/z = 661 [M+H]^+$ beobachtet - Kapitel 7.3.5). Beim Fragment $m/z = 441 [M - (Ring D, 2mal MeOH, H_2O) + K]^+$ könnte es sich um eine Abspaltung von Ring D, zweimal MeOH und H_2O handeln.

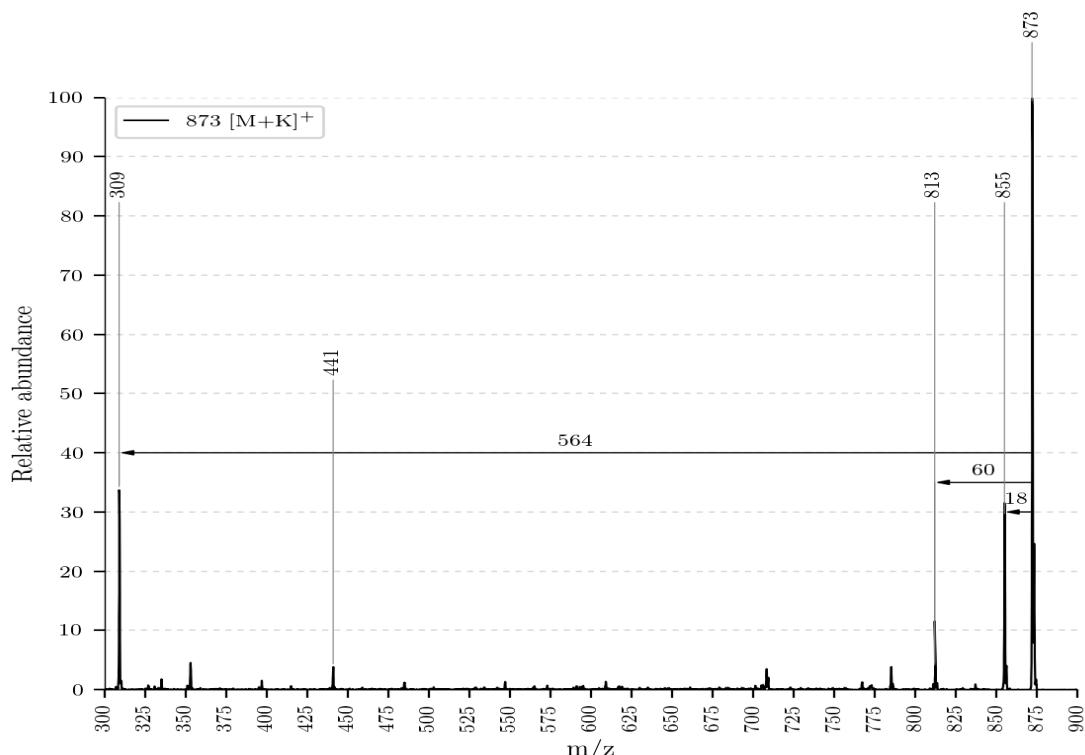


Abbildung 5.18.: ESI-MS des Reaktionsproduktes bei $m/z = 873 [M+K]^+$

Im Fragmentierungsdiagramm sieht man, dass sich das lokale Maximum der Essigsäureabspaltung hin zu niedrigeren Energien verschoben hat. Es befindet sich nun bei 35 NKE. Auch die H_2O Abspaltung verschiebt sich zu niedrigeren Energien und besitzt ein lokales Maximum bei 10 NKE. Im Vergleich zum Bo-DNCC und Bo-NCC-3 nahmen diese Werte um 10 bzw. 5 Einheiten an NKE ab. Dieser Zusammenhang wurde in zwei voneinander unabhängigen Experimenten beobachtet (Abbildung 5.20a und Abbildung 5.20b). Die Ursache

5. Experimente MS-Leafspray

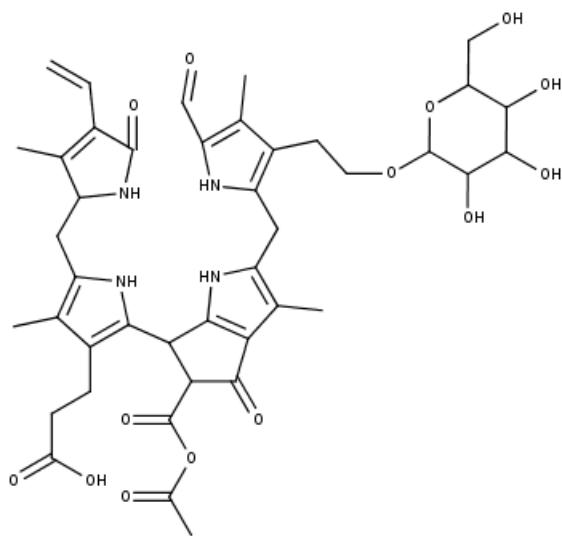


Abbildung 5.19.: Strukturvorschlag des Reaktionsproduktes mit Summenformel $C_{42}H_{50}N_4O_{14}$

könnte beim Zuckerring liegen, der die Elektronenverteilung vermutlich so beeinflusst, dass die Abspaltungen bereits bei niedrigeren Energien erfolgen.
(Einfügen von 3D Bildern, die die sterischen Zusammenhänge vorschlagen).

5. Experimente MS-Leafspray

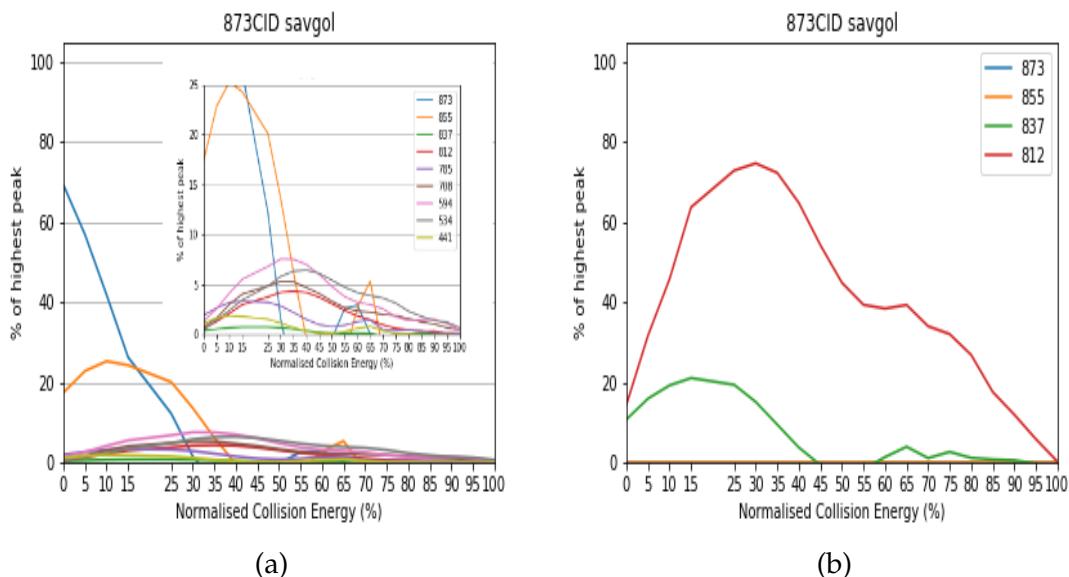


Abbildung 5.20.: Fragmentierungsdiagramm des Reaktionsproduktes: (a) Experiment am 13.09.2017 (11:00) - (blau = 873, orange = 855, grün = 837, rot = 812, violett = 765, braun = 708, pink = 594, grau = 534, hellgrün = 441), (b) Experiment am 13.09.2017 (09:45) - schlechter gelungen, weswegen die Abspaltungen nicht so schön wie in Experiment (a) zu sehen sind (blau = 873, orange = 855, grün = 837, rot = 812)

6. Experimente LC-MS

Die Analyse mit LC-MS diente dazu, die Ergebnisse von MS Leafspray zu überprüfen. In Kombination mit einem hochauflösenden Massenspektrometer wurde zudem die Ermittlung der Strukturen der Chl-Kataboliten erleichtert. Weiters konnten mit MS Leafspray nicht gefundene Chl-Kataboliten identifiziert werden.

6.1. HPLC-Gradient sowie Gerätebeschreibung

6.2. Aufarbeitung der Probe

Um ein Blattextrakt zu erhalten wurde ein Brokkoliblatt auf eine Größe von ca. 2 cm² mit einer Rasierklinge zugeschnitten und mithilfe von Mörser und Pistill aufgerieben und mit 2-5 mL MeOH vermischt (Anm.: um möglichst hohe Intensitäten in der HPLC zu erhalten wurde versucht, eine möglichst hohe Konzentration des Blattextraktes zu erreichen). Die Lösung wurde für 2min. bei 3000 rpm abzentrifugiert und anschließend mit Wasser im Verhältnis 20:80 verdünnt und nach kurzem Homogenisieren für 7min. (3000 rpm) abzentrifugiert. Von der erhaltenen Lösung wurden 50 µL in die 20 µL Schleife der HPLC eingespritzt.

Beim Einspritzen wurde versucht, die ungelösten Bestandteile im Blattextrakt nicht mitzunehmen, da diese die HPLC mit der Zeit verunreinigen könnten. Eine Filterung des Blattextraktes erwies sich als Intensitätsverringern.

Bevor die Blätter wie oben beschrieben aufgerieben wurden, wurden sie in einem Exikator getrocknet (ca. einen halben bis ganzen Tag). Die Blätter, an denen die Reaktion (Kapitel 4.4) stattgefunden hatte, wurden nach dem Stoppen der Reaktion der gleichen Aufarbeitungsmethode unterzogen.

6. Experimente LC-MS

6.3. Theoretische Auswertung der Online-UV/Vis Spektren

Mithilfe einer HPLC kann bestimmt werden, ob es sich bei einem bestimmten Chl-Katabolit um einen NCC, DNCC oder YCC handelt. Man erhält zu jedem Peak im HPLC Chromatogramm ein Online-UV/Vis Spektren, das von einem an die HPLC angeschlossenen UV/Vis Detektor gemessen wurde (Kapitel 6.2).

Ein NCC kann über eine charakteristische Bande bei 315 nm eindeutig bestimmt werden. Die Bande geht dabei auf das konjugierte System von Ring A zurück. Ein DNCC besitzt aufgrund seiner decarboxylierten Carbonylgruppe dieses konjugierte System nicht mehr, weswegen die Bande bei 315 nm verschwindet, der sonstige für einen NCC typische Kurvenverlauf jedoch erhalten bleibt. Bei einem YCC führt die Reduktion der Verbindung zwischen Ring C und D (Einführung einer Doppelbindung zwischen Position 15 und 16) zu einer Erweiterung des konjugierten Systems (nun bestehend aus Ring C und D) und damit zu einer Bande bei 415 nm.

Alle folgenden UV/Vis Spektren und HPLC Chromatogramme wurden bei 254 nm detektiert.

6.4. Theoretische Auswertung der MS Spektren

Über das an die HPLC gekoppelte Massenspektrometer wird zu jedem Zeitpunkt des HPLC Chromatogramms ein Massenspektrum aufgenommen und ein dementsprechendes Chromatogramm erzeugt. Die hervorgehobenen Peaks in den folgenden Chromatogrammen zeigen an, zu welchem Zeitpunkt welcher Katabolit in Bezug auf seine Molekülmasse gefunden wurde. Da das Massenspektrometer erst nach 10min. an die HPLC gekoppelt wurde, muss man, um die entsprechende Retentionszeit im HPLC Chromatogramm zu erhalten, zu jedem Zeitpunkt im Chromatogramm des Massenspektrometers ca. 11min. dazuzählen (1min. steht für den Weg von HPLC zu MS).

Man kann somit den über Online-UV/Vis Spektren identifizierten Chl-Kataboliten (Kapitel 6.3) eine Molekülmasse zuordnen. Im Rahmen meiner Vorwissenschaftlichen Arbeit erwies es sich jedoch als schwierig, die Resultate der HPLC mit

6. Experimente LC-MS

denen des Massenspektrometers im Rahmen eines LC-MS Versuches in Einklang zu bringen (die Retentionszeiten der über HPLC identifizierten Chl-Kataboliten stimmten mit jenen des Massenspektrometers oft nicht überein). Da die Verwendung von Daten aus der HPLC zur Analyse der Chl-Kataboliten nicht das primäre Ziel meiner Arbeit war, spielt dies auch keine wesentliche Rolle. Aus Gründen der wissenschaftlichen Vollständigkeit, werden die Daten der HPLC trotzdem präsentiert. Ebenso wird versucht, die Probleme, die sich aus den Daten ergeben, darzustellen.

Der Typ des Chl-Kataboliten wurde somit, sofern möglich durch ein Online-UV/Vis Spektrum bestimmt und mit der vom Massenspektrometer erhaltenen Summenformel und den sich daraus ergebenden strukturellen Möglichkeiten überprüft. War die Zuordnung anhand UV/Vis Spektren aufgrund von Unklarheiten nicht möglich, wurde zur Strukturbestimmung auf die Daten des Massenspektrometers zurückgegriffen.

6.5. Chl-Kataboliten des Brokkoliblattes mithilfe von LC-MS identifiziert

Das HPLC Chromatogramm in Abbildung 6.1 zeigt, welche der Kataboliten mithilfe ihrer UV/Vis Spektren eindeutig identifiziert werden konnten. Es dürfte sich dabei ob ihrer etwas höheren Intensitäten um die Hauptkataboliten des Brokkoliblattes handeln. Dies müsste jedoch in gezielten quantitativen Messungen weiter und genauer untersucht werden.

Mit dem Massenspektrometer wurden die in Tabelle 6.1 aufgelisteten Phyllo-biline identifiziert. In dieser Tabelle werden neben den Summenformeln auch die exakten Molekülmassen (in Da), die Art des Chl-Katalolit (NCC, DNCC, YCC, DYCC) und die Retentionszeit in der HPLC (soweit eindeutig feststellbar) angegeben.

Eine so große Anzahl an Chl-Kataboliten wie in Tabelle 6.1 vorzufinden wäre ungewöhnlich. Bei einer Betrachtung der Summenformeln und exakten Molekülmassen fällt jedoch auf, dass sich einige Chl-Katalolit um genau ein C-Atom und zwei H-Atome unterscheiden (entspricht einem Massenunterschied von 14 Da).

6. Experimente LC-MS

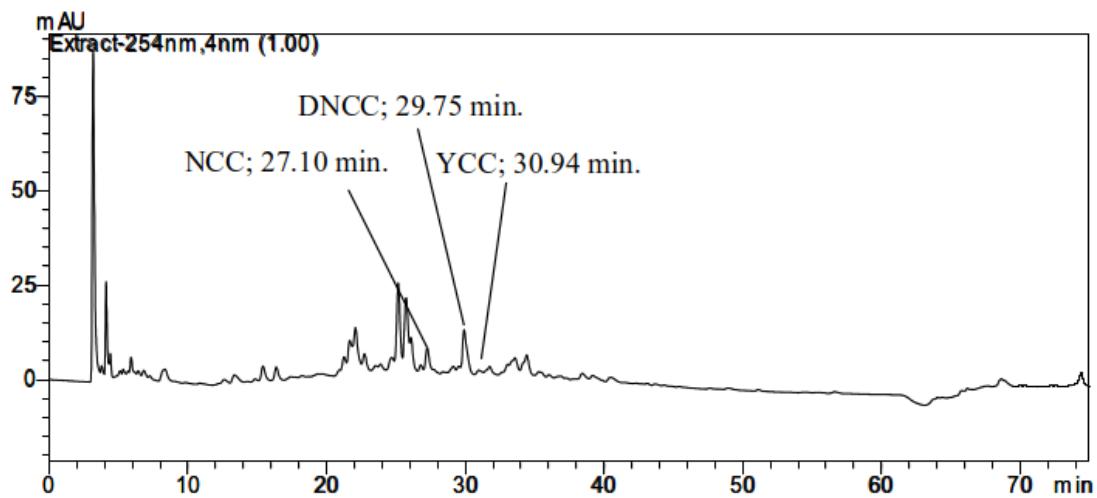


Abbildung 6.1.: HPLC Chromatogramm - die hervorgehobenen Peaks entsprechen den Retentionszeiten und der Art der Chl-Kataboliten, die über ein Online-UV/Vis Spektrum bestimmt wurden; gefunden wurden ein NCC bei 27.10min. (Abbildung 7.10a), ein DNCC bei 29.75min. (Abbildung 7.10b) und ein YCC bei 30.94min. (Abbildung 6.4c)

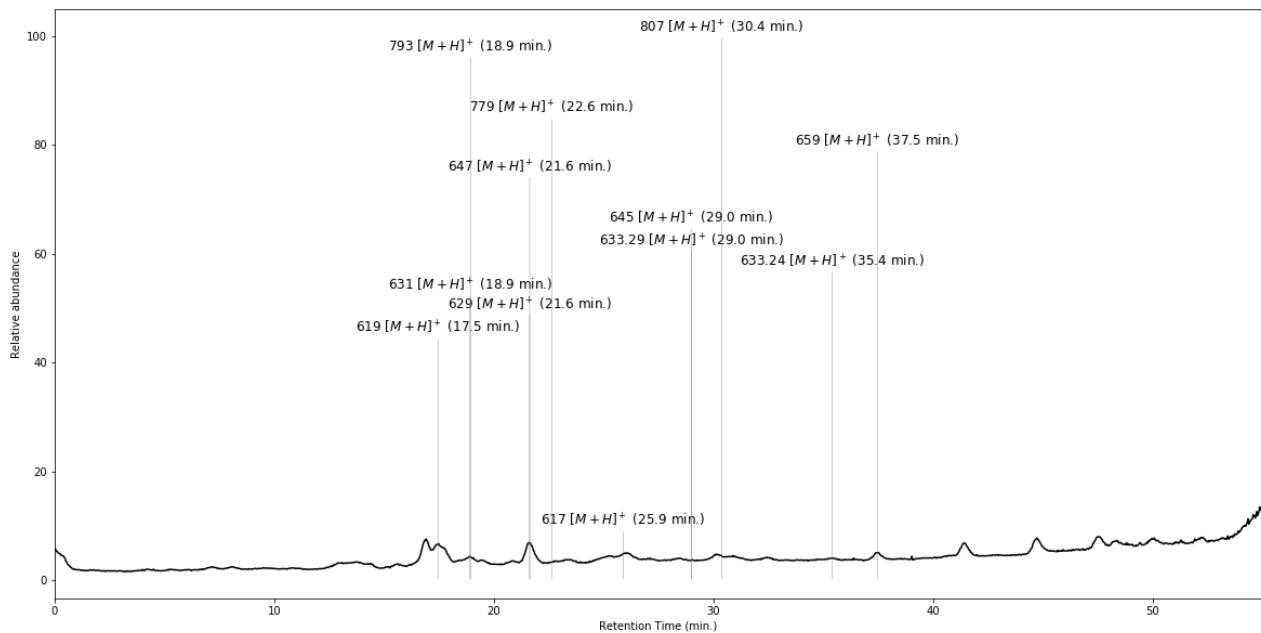


Abbildung 6.2.: LC-MS Chromatogramm

6. Experimente LC-MS

Bezeichnung	Summenformel	M (in Da)	Typ	RT _{HPLC} (in min.)	H.
Bo-DYCC	C ₃₃ H ₃₇ O ₈ N ₄	617.2635	DYCC	30.94?	-
Bo-DNCC	C ₃₃ H ₃₉ O ₈ N ₄	619.2793	DNCC	26.72	-
•	C ₃₄ H ₃₇ O ₈ N ₄	629.2639	•	-	-
-	C ₃₄ H ₃₉ O ₈ N ₄	631.2795	DYCC	29.91, 30.94	Bo-DYCC
-	C ₃₄ H ₄₁ O ₈ N ₄	633.2955	DNCC	-	Bo-DNCC
•	C ₃₆ H ₃₃ O ₇ N ₄	633.2339	•	•	-
Bo-YCC	C ₃₄ H ₃₇ O ₉ N ₄	645.2593	YCC	-	-
Bo-NCC-3	C ₃₄ H ₃₉ O ₉ N ₄	647.2748	NCC	33.04	-
-	C ₃₅ H ₃₉ O ₉ N ₄	659.2741	YCC	-	Bo-YCC
Bo-DNCC-2	C ₃₉ H ₄₇ O ₁₃ N ₄	779.3181	DNCC	•	-
Bo-NCC-1	C ₄₀ H ₄₉ O ₁₃ N ₄	793.3336	NCC	29.91	-
-	C ₄₁ H ₅₁ O ₁₃ N ₄	807.3491	NCC	-	Bo-NCC-1

Tabelle 6.1.: Übersicht über die gefundenen Chl-Kataboliten des Brokkoliblattes und ihren Methylestern, die sich aus der Reaktion der freien Carbonsäure mit MeOH ergeben (die Summenformeln und die exakten Molekülmassen beziehen sich auf die [M+H]⁺ Ionen)

Da alle identifizierten Chl-Katabolit eine freie Carbonsäuregruppe an Position 8² besitzen, wird angenommen, dass diese bei der Aufarbeitung der Probe mit MeOH (Kapitel 6.2) mit diesem reagieren und einen Methylester ausbilden. In der Spalte Herkunft (abgekürzt mit H.) der Tabelle 6.1 wird demnach festgehalten, von welchem Chl-Kataboliten die jeweilige Verbindung stammt. Es handelt sich dabei also um keine Chl-Kataboliten, sondern nur um deren Reaktionsprodukte mit MeOH. In der HPLC konnten sie jedoch nicht identifiziert werden.

Bei einer Retentionszeit von 27.10min. konnte über Online-UV/Vis ein NCC (Abbildung 7.10a) identifiziert werden, da er bei einer Wellenlänge von 315nm eine charakteristische Bande aufweist. Der von den Retentionszeiten dazugehörige Chl-Katabolit im Massenspektrum wäre der Bo-DNCC (mit einer Retentionszeit von 17.5min im Massenspektrometer - Abbildung 6.2). Bei diesem handelt es sich jedoch um einen DNCC. Es wurde versucht, das unlogische Ergebnis durch Überlagerungen mehrerer Chl-Katabolit zu erklären, was aber nicht möglich war (Abbildung 6.3). Es bleibt somit das Zustandekommen dieses UV/Vis Spektrums ungeklärt.

6. Experimente LC-MS

Bei einer Retentionszeit von 29.75min. konnte ein UV/Vis Spektrum eines DNCCs (Abbildung 7.10b) aufgenommen werden. Nach den Retentionszeiten im Massenspektrometer (Abbildung 6.2) kann diesem UV/Vis Spektrum der Chl-Katabolit Bo-NCC-1 zugeordnet werden. Auch der Methylester des Bo-YDNCC ist zu dieser Retentionszeit im Massenspektrometer vorzufinden und trägt damit vermutlich zur Entstehung des Signals bei, was die Verzerrungen bewirken könnte (Abbildung 6.3).

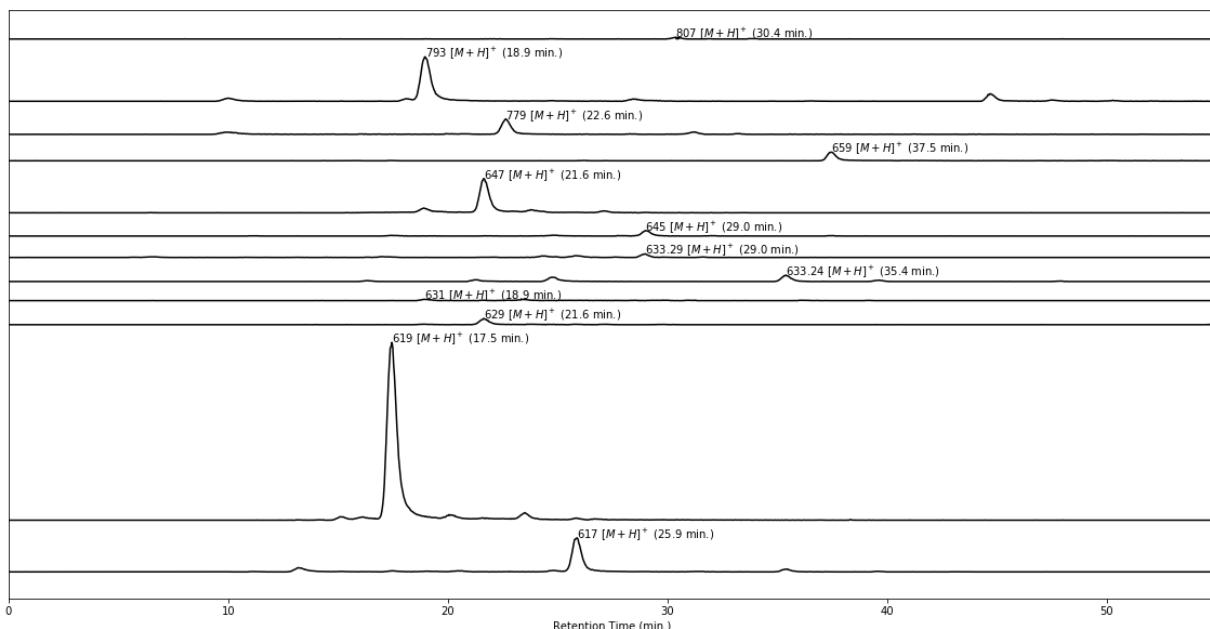


Abbildung 6.3.: LC-MS Chromatogramm zur besseren Darstellung von Überlagerungen von Chlorophyllkataboliten, um diverse unlogische Schlüsse besser verstehen zu können

Bei einer Retentionszeit von 30.94min. ist das UV/Vis Spektrum charakteristisch für einen YCC (Abbildung 7.10b). Im Massenspektrometer wurde zu dieser Retentionszeit der Methylester des Bo-YDNCC gefunden (bei einer Retentionszeit von 18.9min). Auch hier lässt sich keine Verbindung finden, bei der die Retentionszeiten von HPLC und Massenspektrometer exakt zusammenpassen. Es könnte auch hier wieder zu einer Überlagerung kommen (vielleicht mit dem Bo-YDNCC). Diese Überlagerungen könnten durch Isomere der einzelnen Chl-Kataboliten bedingt sein.

6. Experimente LC-MS

Um das Zustandekommen der nicht identifizierbaren UV/Vis Spektren zu erklären wurden Diagramme wie in Abbildung 6.3 erstellt. Es handelt sich dabei um ein Chromatogramm jedes einzelnen im Massenspektrometer während eines LC-MS Laufes identifizierten Chl-Kataboliten. Die Intensitäten wurden auf den höchsten im Zeitraum vorkommenden Peak skaliert. Bei den gekennzeichneten Peaks handelt es sich um jene, bei denen die jeweilige Verbindung die höchste Intensität im Chromatogramm zeigte. Peaks etwaiger Stereoisomere werden nicht beachtet. Mithilfe dieser Abbildung sollten etwaige Überlagerungen ersichtlich werden.

6. Experimente LC-MS

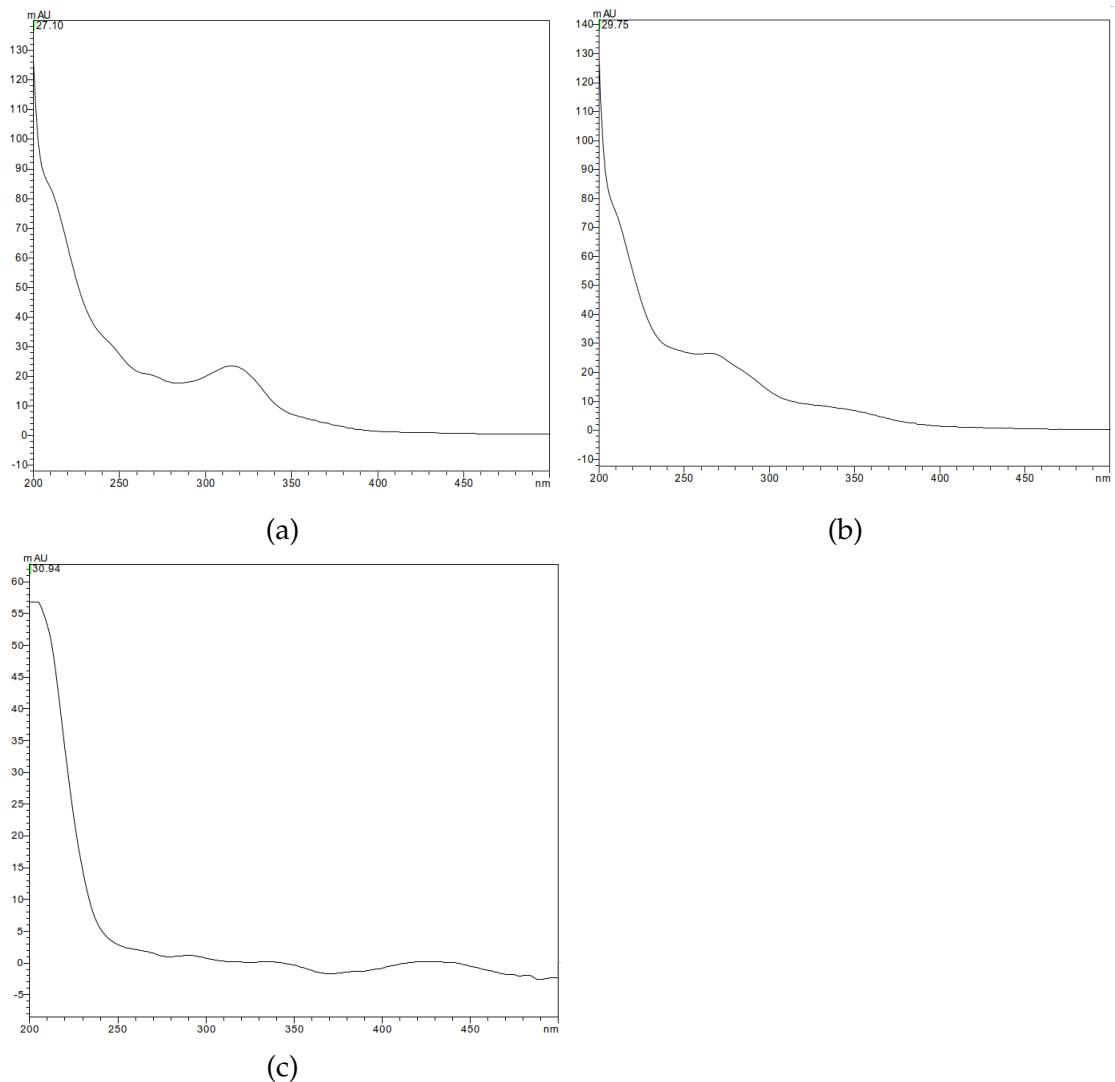


Abbildung 6.4.: Online-UV/Vis Spektren: (a) charakteristisch für einen NCC - RT = 27.25min.,
(b) charakteristisch für einen DNCC - RT = 29.91min., (c) charakteristisch für
einen YCC - RT = 30.94min.

6.6. Identifikation der Reaktionsprodukte

Die Produkte der Reaktion mit Essigsäureanhydrid konnten ebenfalls mithilfe von LC-MS identifiziert werden. In Abbildung 6.5 sind die Reaktionsprodukte, die mittels Online-UV/Vis Spektren identifiziert wurden, dargestellt. Die dazugehörigen UV/Vis Spektren werden in Abbildungen 6.8a-e dargestellt. Es handelt sich dabei um die Hauptreaktionsprodukte, die in der HPLC dadurch charakterisiert sind, dass sich ihre Retentionszeiten nach hinten verschieben. Sie dürften somit apolarere Eigenschaften besitzen wie die Chl-Kataboliten, was vermutlich durch den Methylester bedingt ist. Über die Verschiebung der Peaks im HPLC Chromatogramm wird das Stattfinden der Reaktion auf einen Blick ersichtlich (vergleiche Abbildung 6.1 und Abbildung 6.5). Zudem konnten im Vergleich zu den HPLC Läufen ohne Reaktion mehr Verbindungen über Online-UV/Vis Spektren beobachtet und identifiziert werden, da es zu einer größeren Auftrennung kommt.

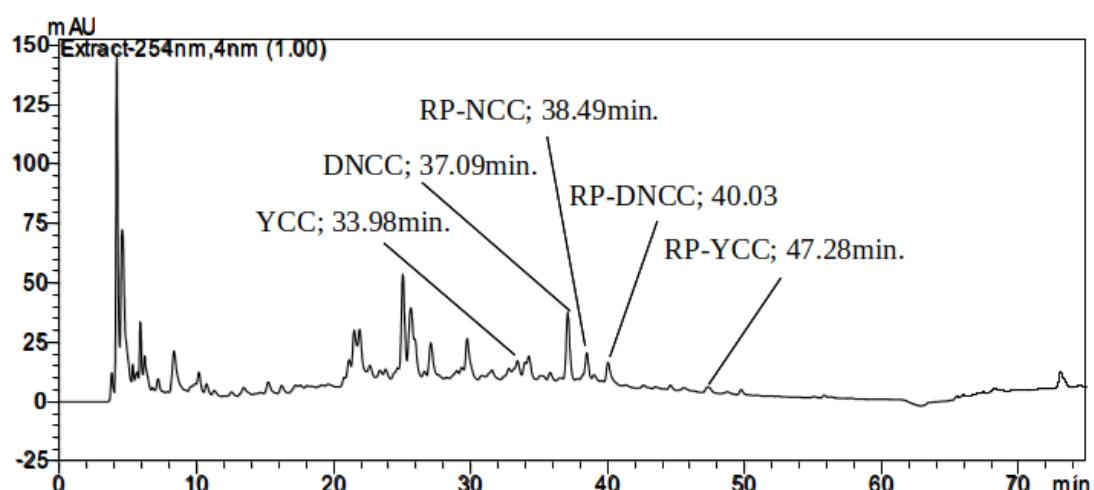


Abbildung 6.5.: HPLC Chromatogramm, gefunden wurden Reaktionsprodukte mit den Eigen-schaften eines NCCs (RT = 38.49min.), eines DNCCs (RT = 40.03min.), eines YCCs (RT = 47.28min.) und eines weiteren DNCCs (37.09min.); beim YCC bei RT = 33.98min. könnte es sich um einen weiteren Chl-Kataboliten handeln

Das Chromatogramm des Massenspektrometers des LC-MS Laufes (Abbildung 6.6) zeigt die Massen aller Chl-Kataboliten und die Zeitpunkte, zu denen sie jeweils eluieren. Durch die stattgefunden Reaktion sind dementsprechend mehr Signale vorhanden. Zu beachten ist, dass aufgrund der Aufarbeitung mit MeOH nicht ein Essigsäureanhydrid beobachtet wird, sondern der Methylester, der sich offensichtlich aufgrund der vermeintlich guten Abgangsgruppe CH_3COOH

6. Experimente LC-MS

schnell bildet (auch Kapitel 6.5 und 5.5).

Auffallend ist, dass manche Verbindungen in ihren Retentionszeiten verschoben worden sind. So eluiert Verbindung mit $m/z = 631 [M+H]^+$ nun bei 39.0min. im Vergleich zu 18.9min., Verbindung mit $m/z = 629 [M+H]^+$ bei 31.8min. im Vergleich zu 21.6min. und Verbindung mit $m/z = 645 [M+H]^+$ bei 34.5min. im Vergleich zu 29.0min. (vergleiche Abbildungen 6.6 und 6.2). Gründe für diese Verschiebungen müssten weiter untersucht werden bzw. müsste überprüft werden, ob es bei den Versuchen, aus denen einer zu Abbildung 6.2 führte, nicht einen Messfehler gab. Eine Überprüfung und erneute Durchführung der Messung im Rahmen meiner Vorwissenschaftlichen Arbeit führte zum selben Ergebnis (siehe Anhang).

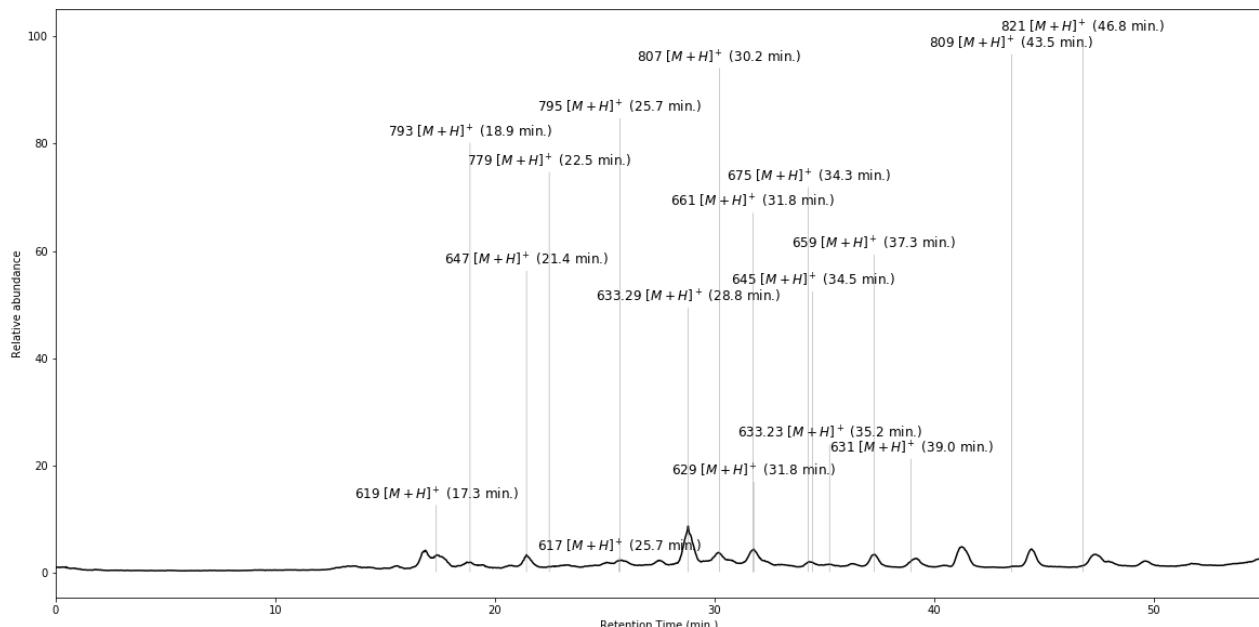


Abbildung 6.6.: LC-MS Chromatogramm

In Tabelle 6.2 werden die Chl-Kataboliten dargestellt, die nach der Reaktion gefunden wurden. Dabei wurde neben dem erwarteten Reaktionsprodukt bei $m/z = 659.2741 [M+H]^+$ eine weitere Verbindung bei $m/z = 659.2348 [M+H?]^+$ entdeckt, jedoch mit einer anderen Summenformel. Wie dieses Signal zustandekommt bleibt ungeklärt. Da bei $m/z = 633 [M+H]^+$ ebenfalls zwei

6. Experimente LC-MS

Molekülmassen beobachtet wurden ($m/z = 633.2955$ und $m/z = 633.2339$), könnte es sich um ähnliche Charakteristiken handeln, die jedoch weiter untersucht werden müssten. Von einer anderen Verbindungsklasse ist aufgrund der passenden Summenformel mit entsprechender Doppelbindungsäquivalenz und dem Massendefekt nicht auszugehen!

In Abbildung 6.7 wird nur unter Verwendung von Massenspektrometerdaten ersichtlich, dass die Reaktion stattgefunden hat. Es wird angenommen, dass die Peakhöhen in direkt proportionaler Abhängigkeit mit der quantitativen Anzahl der jeweiligen Ionen steht.

Man sieht, dass die Intensität des Bo-DNCC zurückgegangen ist, wohingegen jene von $m/z = 633 [M+H]^+$, dem Reaktionsprodukt viel größer ist. Ähnliche Verschiebungen sind von Bo-NCC-1 auf $m/z = 807 [M+H]^+$ und $821 [M+H]^+$ sowie von Bo-NCC-3 auf $m/z = 661 [M+H]^+$ und $675 [M+H]^+$ sichtbar. Die anderen Intensitätsverschiebungen sind nicht so groß, dass sie in diesem Kontext interpretierbar wären. Sie sind aber trotzdem vorhanden.

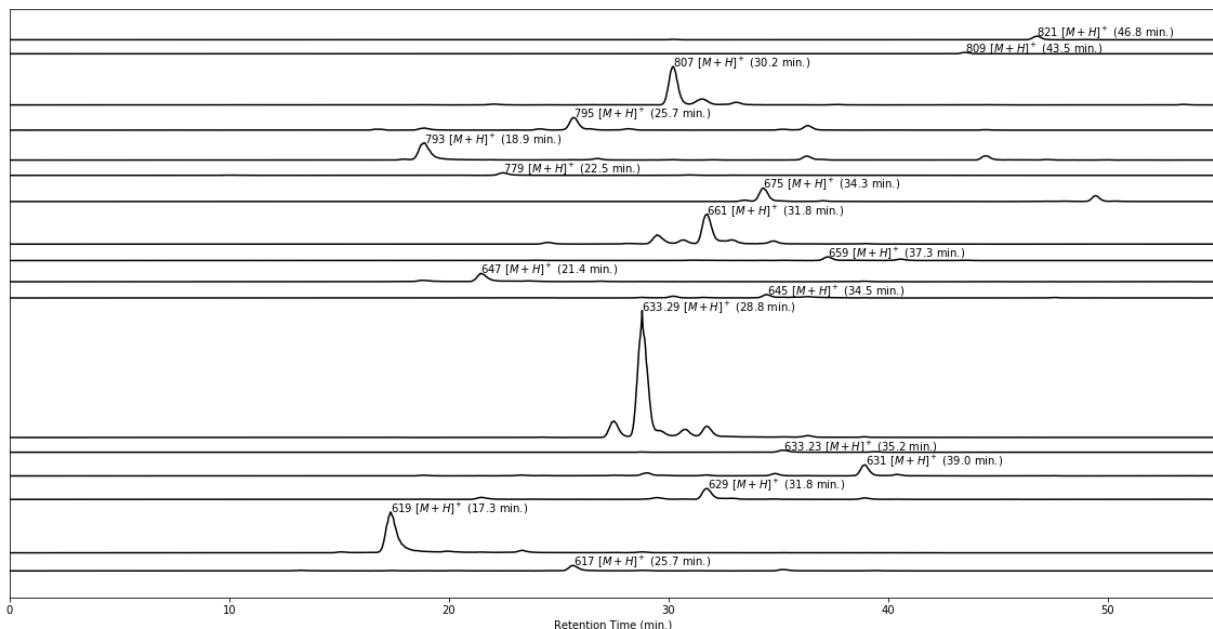


Abbildung 6.7.: LC-MS Chromatogramm

6. Experimente LC-MS

Um das Stattfinden der Reaktion zu sehen, vergleiche mit Abbildung 6.3.

Bezeichnung	Summenformel	M (in Da)	Typ	RT _{HPLC} (in min.)	H.
Bo-DYCC	C ₃₃ H ₃₇ O ₈ N ₄	617.2599	DYCC	30.94?	-
Bo-DNCC	C ₃₃ H ₃₉ O ₈ N ₄	619.2798	DNCC	26.72	-
•	C ₃₄ H ₃₇ O ₈ N ₄	629.2641	•	-	-
-	C ₃₄ H ₃₉ O ₈ N ₄	631.2795	DYCC	29.91, 30.94	Bo-DYCC
-	C ₃₄ H ₄₁ O ₈ N ₄	633.2955	DNCC	28.8	Bo-DNCC
•	C ₃₆ H ₃₃ O ₇ N ₄	633.2339	•	-	-
Bo-YCC	C ₃₄ H ₃₇ O ₉ N ₄	645.2593	YCC	-	-
-	C ₃₅ H ₄₁ O ₈ N ₄	645.2953	DYCC	-	Bo-DYCC
Bo-NCC-3	C ₃₄ H ₃₉ O ₉ N ₄	647.2748	NCC	33.04	-
•	C ₃₄ H ₃₅ O ₁₀ N ₄	659.2348	•	-	-
-	C ₃₅ H ₃₉ O ₉ N ₄	659.2741	YCC	37.09	Bo-YCC
-	C ₃₅ H ₄₁ O ₉ N ₄	661.2902	-	-	Bo-NCC-3
-	C ₃₆ H ₄₃ O ₉ N ₄	675.306	-	-	Bo-NCC-3
Bo-DNCC-2	C ₃₉ H ₄₇ O ₁₃ N ₄	779.3181	DNCC	-	-
Bo-NCC-1	C ₄₀ H ₄₉ O ₁₃ N ₄	793.3336	NCC	29.91	-
-	C ₄₀ H ₅₁ O ₁₃ N ₄	795.3491	-	-	-
-	C ₄₁ H ₅₁ O ₁₃ N ₄	807.3491	NCC	40.03	Bo-NCC-1
-	C ₄₁ H ₅₃ O ₁₃ N ₄	809.3649	-	-	795
-	C ₄₂ H ₅₃ O ₁₃ N ₄	821.3652	NCC	47.28	Bo-NCC-1

Tabelle 6.2.: Übersicht über die gefundenen Chl-Kataboliten des Brokkoliblattes und ihren Methylestern, die sich aus der Reaktion der freien Carbonsäure mit Essigsäureanhydrid und der anschließenden Aufarbeitung mit MeOH ergeben. Durch die Aktivierung der Reaktion durch Essigsäureanhydrid sind mehr Produkte zu sehen und diese sind in größeren Intensitäten vorhanden. (die Summenformeln und die exakten Molekülmassen beziehen sich auf die [M+H]⁺ Ionen)

6. Experimente LC-MS

Mithilfe von UV/Vis Spektren konnten ein YCC (RT = 33.98min.), ein DNCC (RT = 37.09min.), ein NCC (RT = 38.94min.), ein weiterer DNCC (RT = 40.03min.) und ein YCC (RT = 47.28min.) identifiziert werden (Abbildungen 6.8a-e).

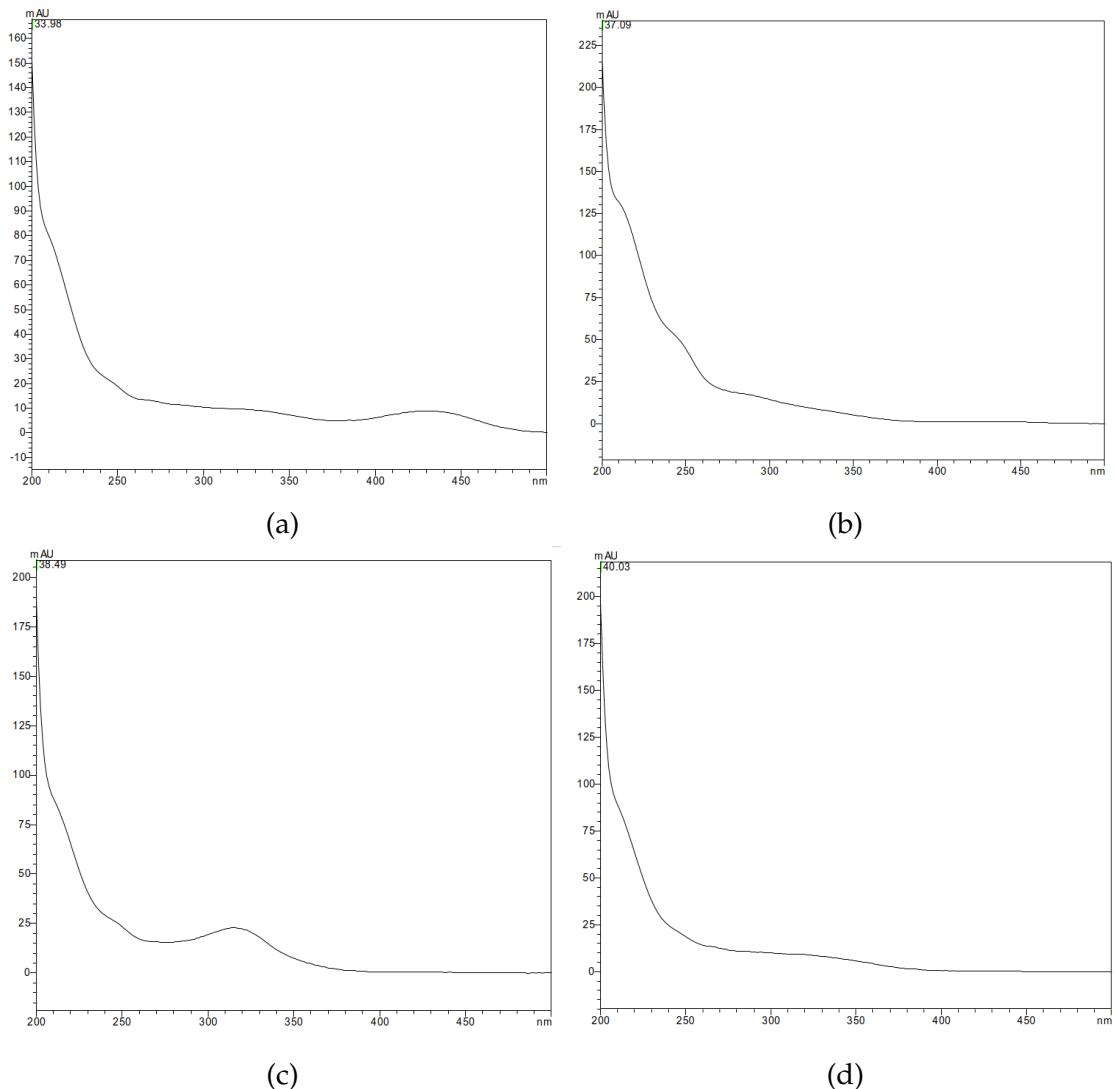


Abbildung 6.8.: Online-UV/Vis Spektren: charakteristisch für (a) YCC - RT = 33.98min., (b) DNCC - RT = 37.09min., (c) NCC - RT = 38.94min., (d) DNCC - RT = 40.03min., (e) YCC - RT = 47.28min.

6. Experimente LC-MS

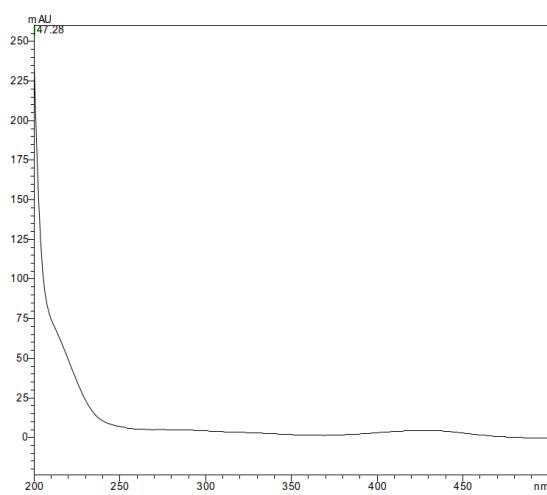


Abbildung 6.9.

Abbildung 6.10.: Online-UV/Vis Spektrum charakteristisch für einen YCC - RT = 47.28min.

7. Strukturaufklärung der Chl-Kataboliten mit ESI-MS

7.1. Beschreibung der Methode

Mit dem Wissen über die ungefähren Retentionszeiten der Chl-Kataboliten in der HPLC konnten diese in EPPIs gesammelt werden. Dabei wurde die aus der HPLC eluierte Flüssigkeit im Zeitrahmen des Peaks aufgenommen, in dem das Auftreten des jeweiligen Chl-Kataboliten vermutet wird (ca. 0.5min.). Die erhaltene Lösung wurde in das Massenspektrometer eingespritzt und analysiert, wobei besonderes Augenmerk auf die Fragmentierung und insbesondere der Erstellung von Fragmentierungsdiagrammen gelegt wurde.

Die Fragmentierungsdiagramme werden hier nicht diskutiert, da dies den Umfang der Arbeit deutlich sprengen würde.

7.2. Identifizierte Chl-Kataboliten

Im Folgenden finden sich jene Chl-Kataboliten, die über diese Methode genauer charakterisiert werden konnten (dies umfasst nicht alle mit LC-MS beobachteten - siehe Tabelle 6.2).

7.2.1. Bo-DYCC

Dieser Chl-Katabolit konnte mit der Methode von MS Leafspray nicht gefunden werden. Mit einem hochauflösenden Massenspektrometer wurde er mit $m/z = 617 [M+H]^+$ identifiziert. Es wurden Abspaltungen von H_2O bei $m/z = 599 [M - (H_2O) + H]^+$, von CO_2 bei $m/z = 573 [M - (CO_2) + H]^+$, von Ring A bei $m/z = 432 [M - (Ring A) + H]^+$ und von Ring C und D bei $m/z = 331 [M - (Ring C, Ring D) + H]^+$ (Abbildung 7.1) beobachtet.

7. Strukturaufklärung der Chl-Kataboliten mit ESI-MS

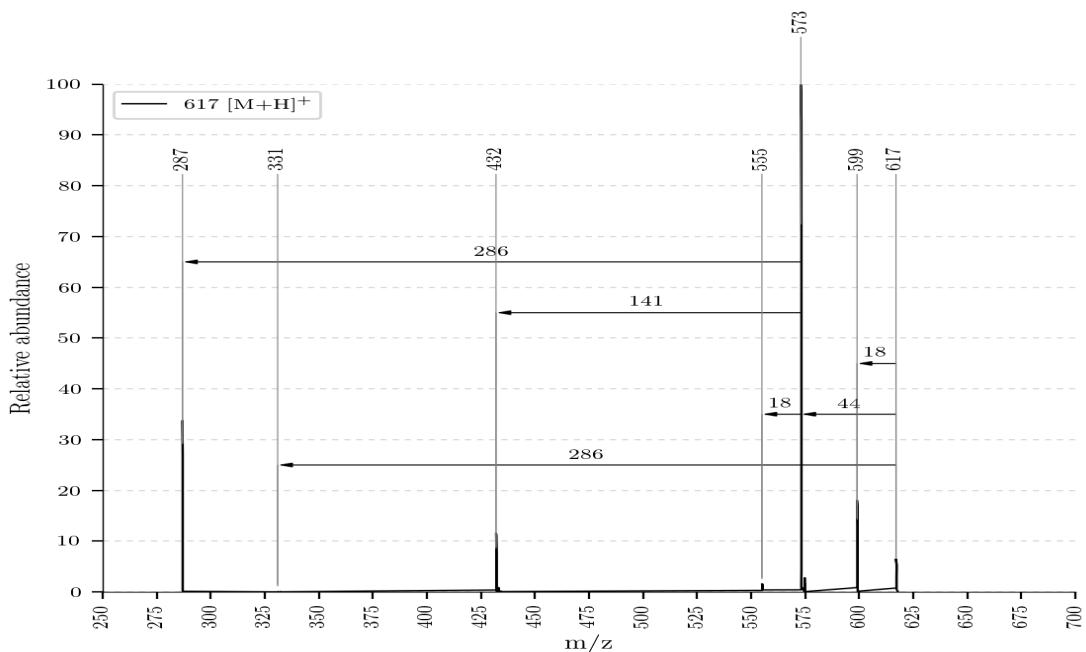


Abbildung 7.1.: ESI-MS Spektrum von Bo-DYCC mit $m/z = 617$ $[M+H]^+$

Die Struktur des Bo-DYCC wird wie in Abbildung 7.2 vorgeschlagen. Aufgrund der CO_2 Abspaltung wird eine freie Carbonsäure an Position 8² angenommen. Die zwei fehlenden H-Atome im Vergleich zum Bo-DNCC und die sinnvollen Zuordnungen der anderen Fragmentierungen weisen auf eine Doppelbindung an Position 16 hin.

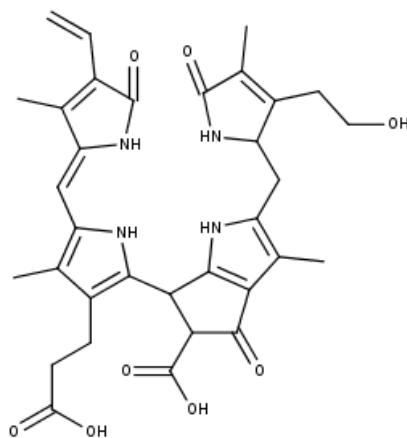


Abbildung 7.2.: Strukturvorschlag von Bo-DYCC mit Summenformel $C_{33}H_{36}N_4O_8$

7. Strukturaufklärung der Chl-Kataboliten mit ESI-MS

Der Mechanismus der Abspaltung von Ring C zusammen mit Ring D wird wie in den Abbildungen 7.3a und 7.3b dargestellt angenommen.

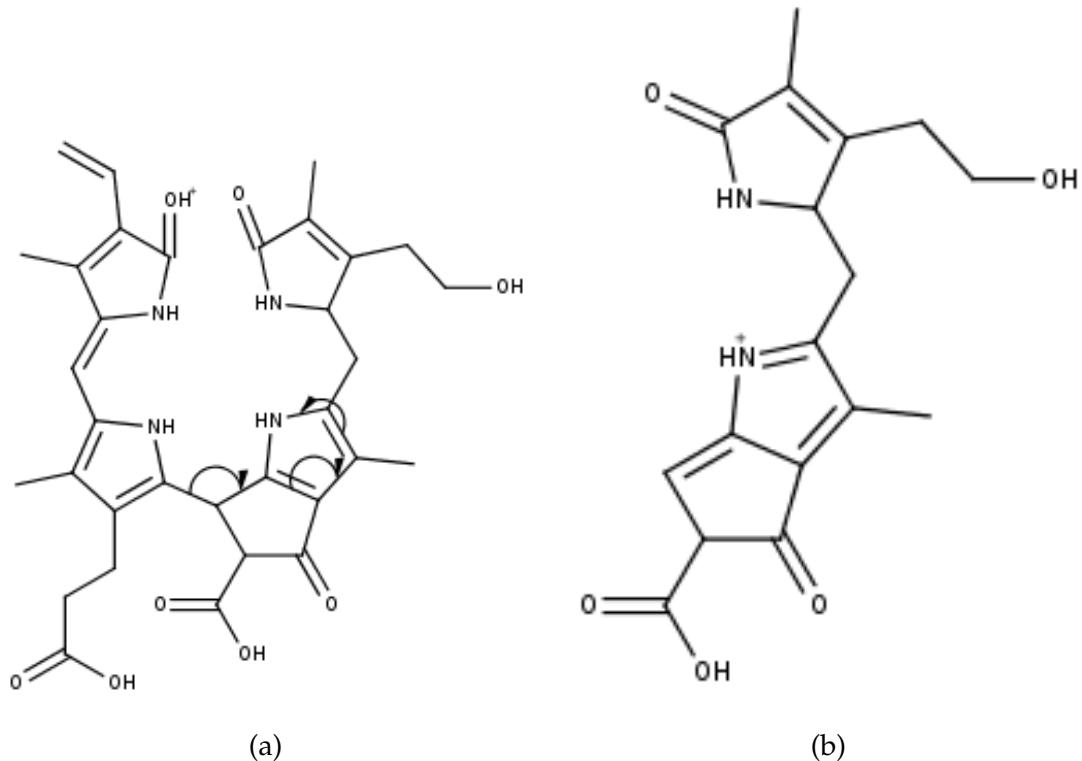


Abbildung 7.3.: mechanistischer Vorschlag für die Abspaltung von Ring C und Ring D: (a) vorgeschlagene Bewegung der Elektronen, (b) Resultat der Abspaltung bei $m/z = 331 [M - (\text{Ring C, Ring D}) + \text{H}]^+$ und Summenformel $C_{17}\text{H}_{19}\text{O}_5\text{N}_2$

7. Strukturaufklärung der Chl-Kataboliten mit ESI-MS

7.2.2. Bo-DNCC

Vom Bo-DNCC wurde die protonierte Verbindung bei $m/z = 619 [M+H]^+$ aufgenommen. Sie zeigt Abspaltungen von H_2O bei $m/z = 601 [M - (H_2O) + H]^+$, von CO_2 bei $m/z = 575 [M - (CO_2) + H]^+$, von Ring D bei $m/z = 452 [M - (Ring D) + H]^+$ und eine gemeinsame Abspaltung von Ring D und Ring A bei $m/z = 311 [M - (Ring D, Ring A) + H]^+$ (Abbildung 7.4).

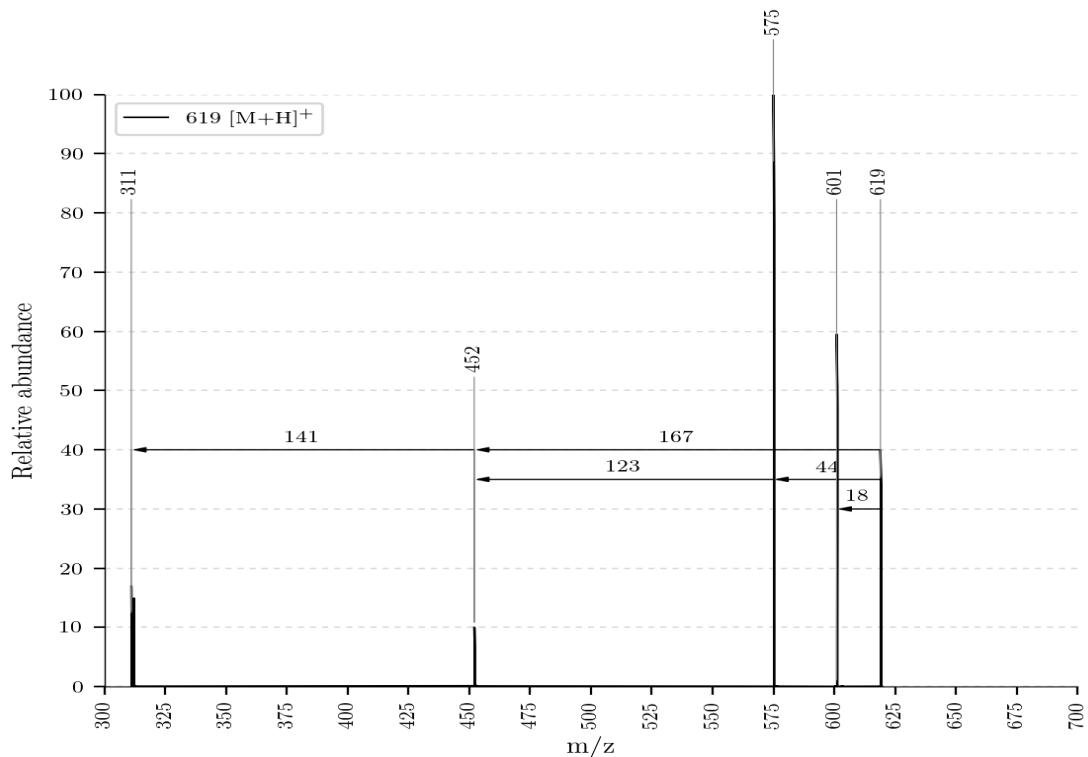


Abbildung 7.4.: ESI-MS Spektrum von Bo-DNCC bei $m/z = 619 [M+H]^+$

Da für die Abspaltung von Ring A und Ring D jeweils ein Mechanismus in [17] vorgeschlagen wird, gibt es unterschiedliche Möglichkeiten für das Abspaltungsprodukt, wenn, wie hier beobachtet, beide Ringe gleichzeitig abgespalten werden (Abbildung 7.5a). In den Abbildungen 7.5b, 7.6a-b werden die einzelnen Mesomere vorgeschlagen, die sich aus den Betrachtungen ergeben.

Als stabiler werden die Mesomere bei 7.6a-b aufgrund eines stabilen konjugierten Systems erachtet.

7. Strukturaufklärung der Chl-Kataboliten mit ESI-MS

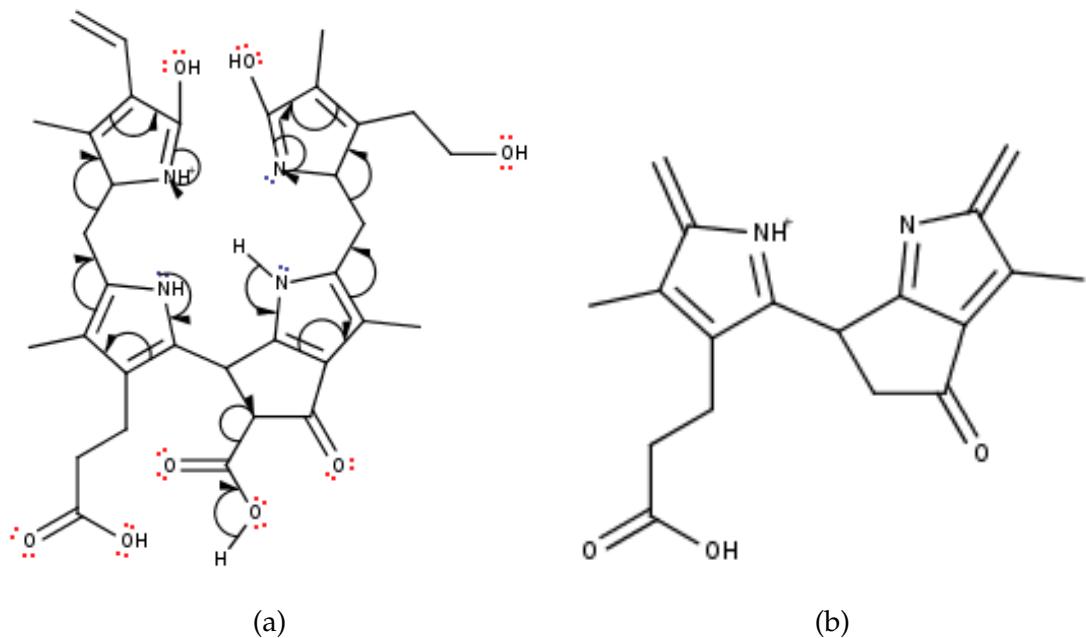


Abbildung 7.5.: vorgeschlagener Abspaltungsmechanismus von Ring D und Ring A: (a) vorgeschlagene Elektronenbewegung, (b) Mesomer 1 der Abspaltung mit $m/z = 311$ $[M - (\text{Ring D, Ring A}) + \text{H}]^+$ und Summenformel $C_{18}\text{H}_{19}\text{O}_3\text{N}_2$

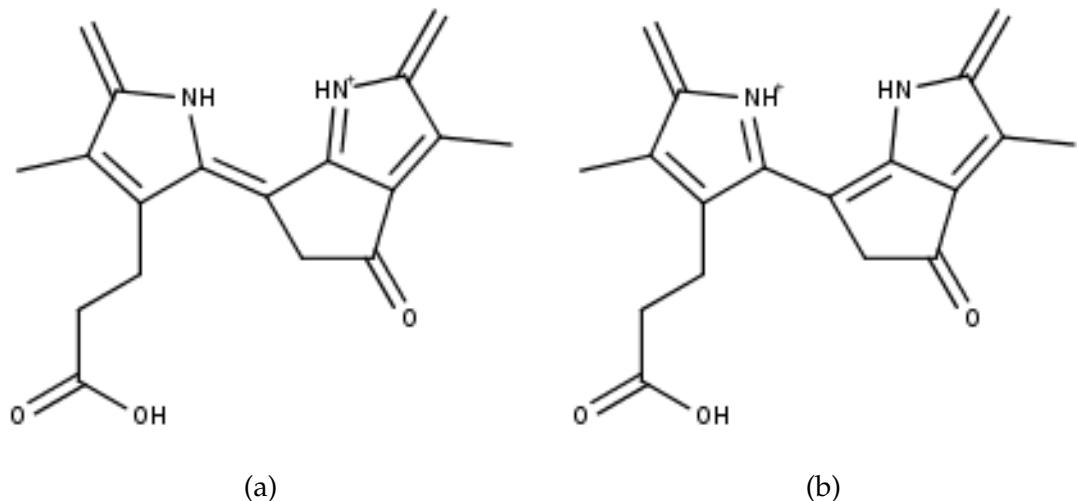


Abbildung 7.6.: Mesomere des Abspaltungsproduktes: (a) Mesomer 2, (b) Mesomer 3

7. Strukturaufklärung der Chl-Kataboliten mit ESI-MS

7.2.3. Bo-YCC

Der Bo-YCC konnte, so wie der Bo-DYCC nicht mit MS Leafspray aufgenommen werden, dafür jedoch im Rahmen der hier verwendeten Methode. Er wurde mit $m/z = 645 [M+H]^+$ identifiziert und zeigt Abspaltungen von H_2O bei $m/z = 627 [M - (H_2O) + H]^+$ und von CO_2 bei $m/z = 601 [M - (CO_2) + H]^+$ (Abbildung 7.13).

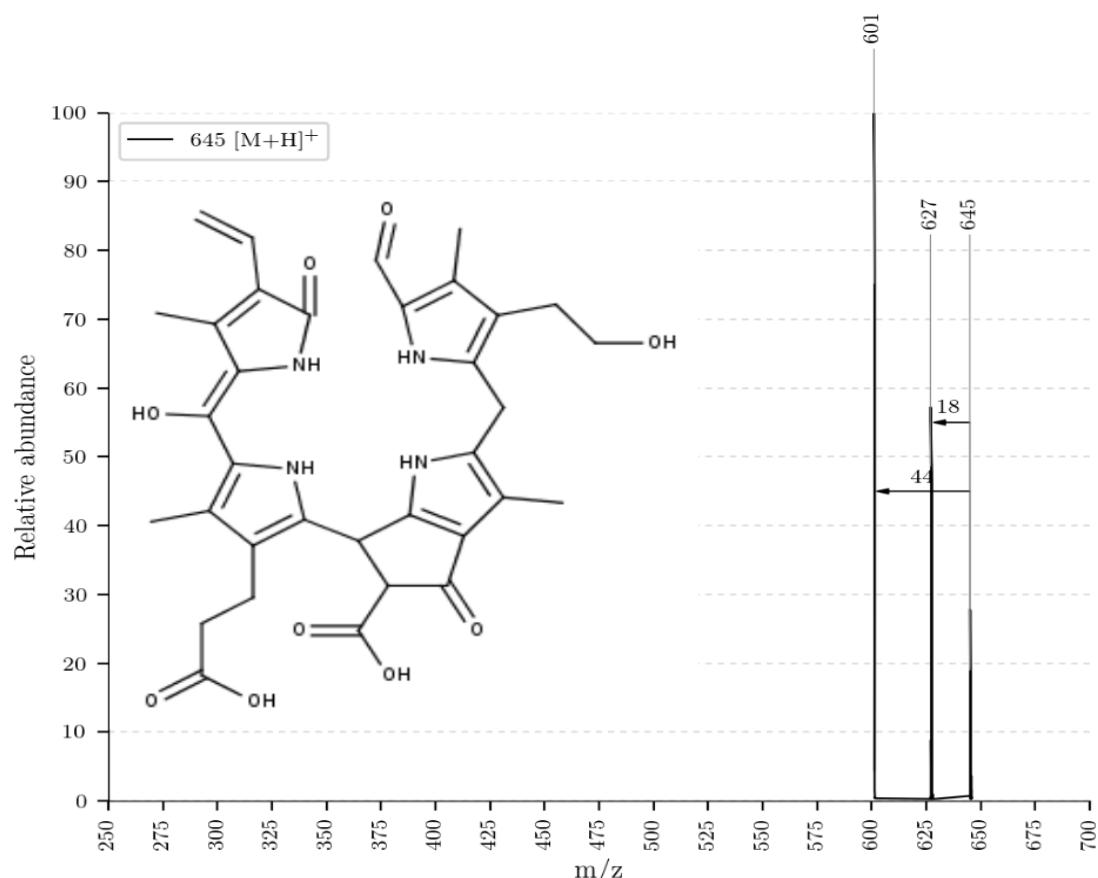


Abbildung 7.7.: ESI-MS Spektrum von Bo-YCC mit $m/z = 645 [M+H]^+$ und Strukturvorschlag mit Summenformel $C_{34}H_{36}O_9N_4$

Die Abspaltung von CO_2 hilft, ihn vom Dimethylierungsprodukt des Bo-DYCC zu unterscheiden. Es wird deswegen an Position 8^2 eine freie Carbonsäure und an Position 16 eine Doppelbindung mit einer Hydroxygruppe angenommen (Abbildung 7.13).

7. Strukturaufklärung der Chl-Kataboliten mit ESI-MS

7.2.4. Bo-NCC-3

Der Bo-NCC-3 wurde ebenfalls wie mit MS Leafspray gefunden. Die Masse des Chl-Kataboliten konnte bei $m/z = 647 [M+H]^+$ bestimmt werden. Abspaltungen von H_2O bei $m/z = 629 [M - (H_2O) + H]^+$, von CO_2 bei $m/z = 603 [M - (CO_2 + H)]^+$ und von Ring D mit CO_2 bei $m/z = 480 [M - (Ring D-CO_2) + H]^+$ wurden beobachtet.

Aufgrund der vorgeschlagenen Hydroxygruppe an Position 16 wird angenommen, dass das Abspaltungsprodukt in einem Keto-/Enolgleichgewicht steht, wie in Abbildung 7.9b vorgeschlagen. Mithilfe von Fragmentierungsdiagrammen könnte man argumentieren, dass diese mögliche Tautomerie aufgrund ihrer Stabilität als Triebkraft für die Abspaltung gesehen werden kann.

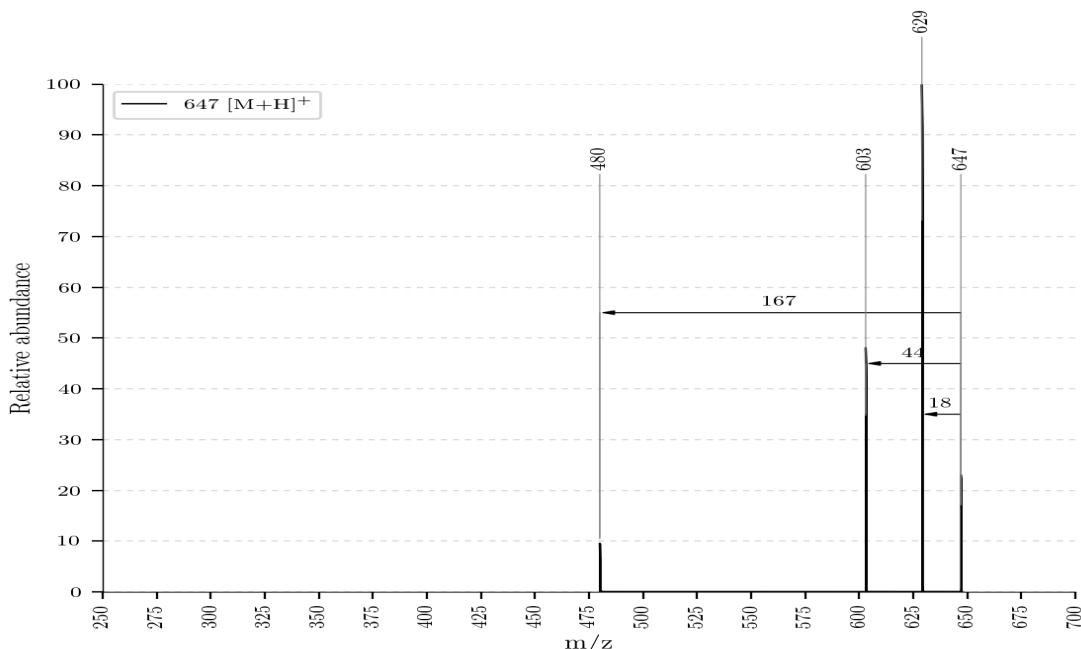


Abbildung 7.8.: ESI-MS Spektrum von Bo-NCC-3 mit $m/z = 647 [M+H]^+$

7. Strukturaufklärung der Chl-Kataboliten mit ESI-MS

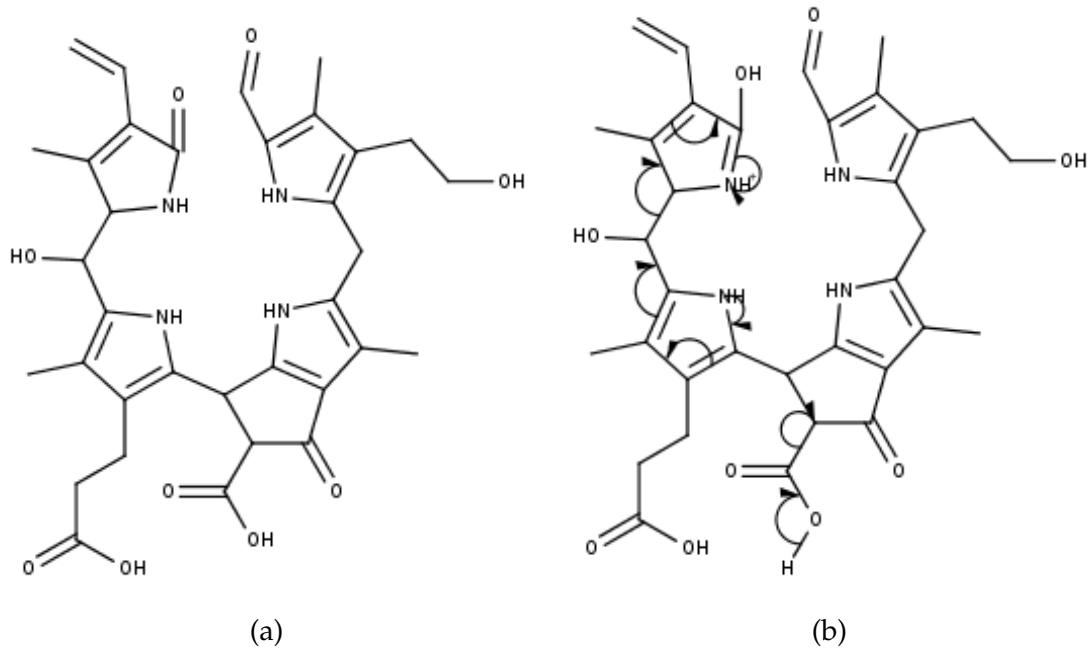


Abbildung 7.9.: (a) Strukturvorschlag von Bo-NCC-3 mit Summenformel C₃₄H₃₉O₉N₄, (b) vorgeschlagener Mechanismus für die Abspaltung von Ring D mit CO₂

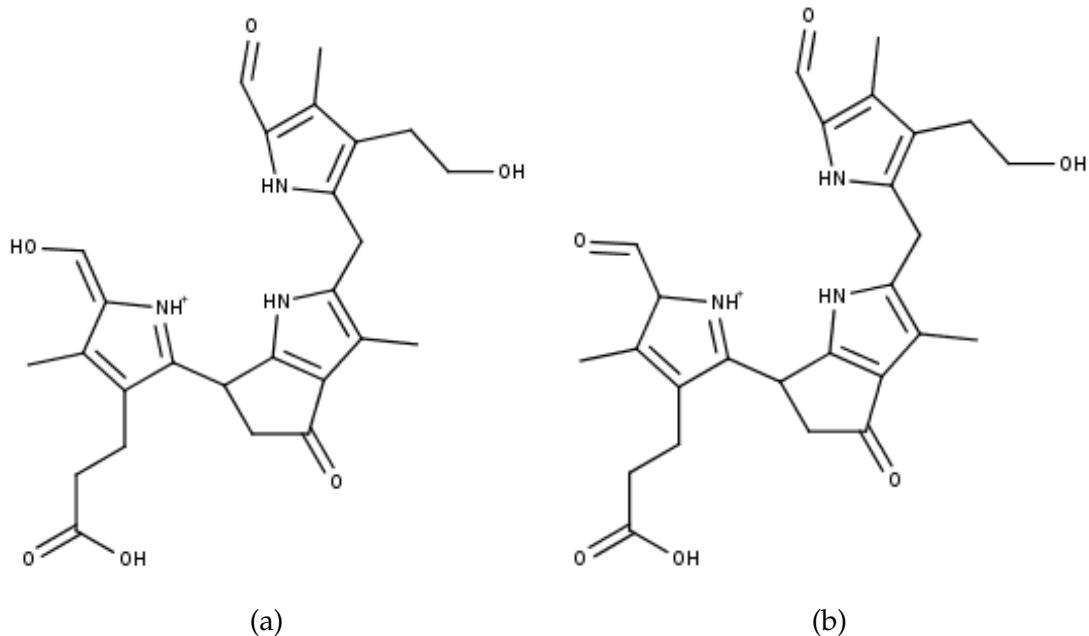


Abbildung 7.10.: vorgeschlagene Keto-/Enoltautomerie beim Abspaltungsprodukt von Ring D mit Summenformel C₂₆H₃₀O₆N₃; (a) Enolform, (b) Aldehyd (vermutlich stabiler)

7. Strukturaufklärung der Chl-Kataboliten mit ESI-MS

7.2.5. Bo-NCC-1

Der Bo-NCC-1 konnte bei $m/z = 793 [M+H]^+$ aufgenommen werden. Er zeigte im Vergleich zu MS Leafspray mehr Abspaltungen, nämlich von H_2O bei $m/z = 775 [M - (H_2O) + H]^+$, von CO_2 bei $m/z = 749 [M - (CO_2) + H]^+$, von Zucker bei $m/z = 631 [M - (Zucker) + H]^+$, von Zucker zusammen mit H_2O bei $m/z = 613 [M - (H_2O) + H]^+$ und von Zucker zusammen mit CO_2 bei $m/z = 587 [M - (CO_2) + H]^+$.

Die Struktur wird wie in Abbildung 5.5a vorgeschlagen.

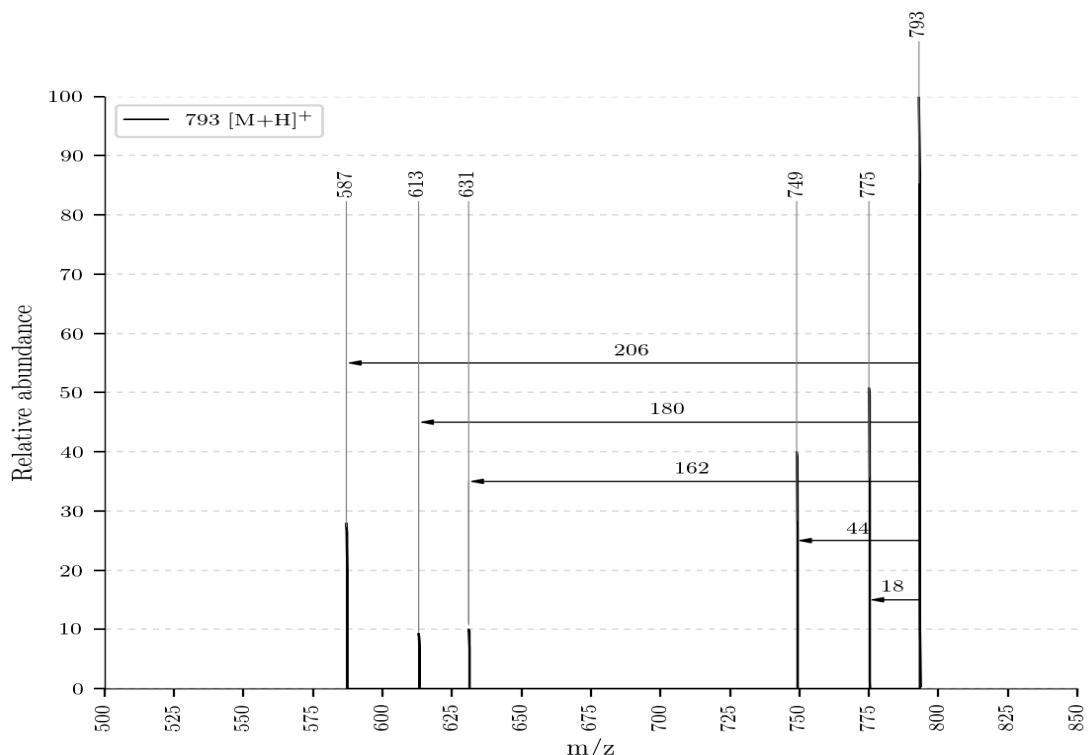


Abbildung 7.11.: ESI-MS Spektrum von Bo-NCC-1 mit $m/z = 793 [M+H]^+$

7. Strukturaufklärung der Chl-Kataboliten mit ESI-MS

7.3. Reaktionsprodukte der Chl-Kataboliten

7.3.1. Reaktionsprodukte von Bo-DYCC

Vom Bo-DYCC konnten zwei Reaktionsprodukte gefunden werden, eines bei $m/z = 631 [M+H]^+$ und eines bei $m/z = 645 [M+H]^+$. Beim ersten handelt es sich um einen Methylester. Dieser wird an Position 12^3 vermutet, da das Ion eine Abspaltung von CO_2 bei $m/z = 587 [M - (\text{CO}_2) + \text{H}]^+$ zeigt (Abbildung 7.12).

Beim Reaktionsprodukt bei $m/z = 645 [M+H]^+$ handelt es sich um einen Dimethylester, an den Positionen 8^2 und 12^3 . Dies zeigt die Abspaltung von MeOH bei $m/z = 613 [M - (\text{MeOH}) + \text{H}]^+$ (Abbildung 7.13).

Dies ist überraschend, da bei den meisten anderen untersuchten Chl-Kataboliten die Bildung eines Monomethylesters an Position 8^2 beobachtet wurde, welche durch das häufige natürliche Vorhandensein von Methylestern als günstiger angesehen werden kann.

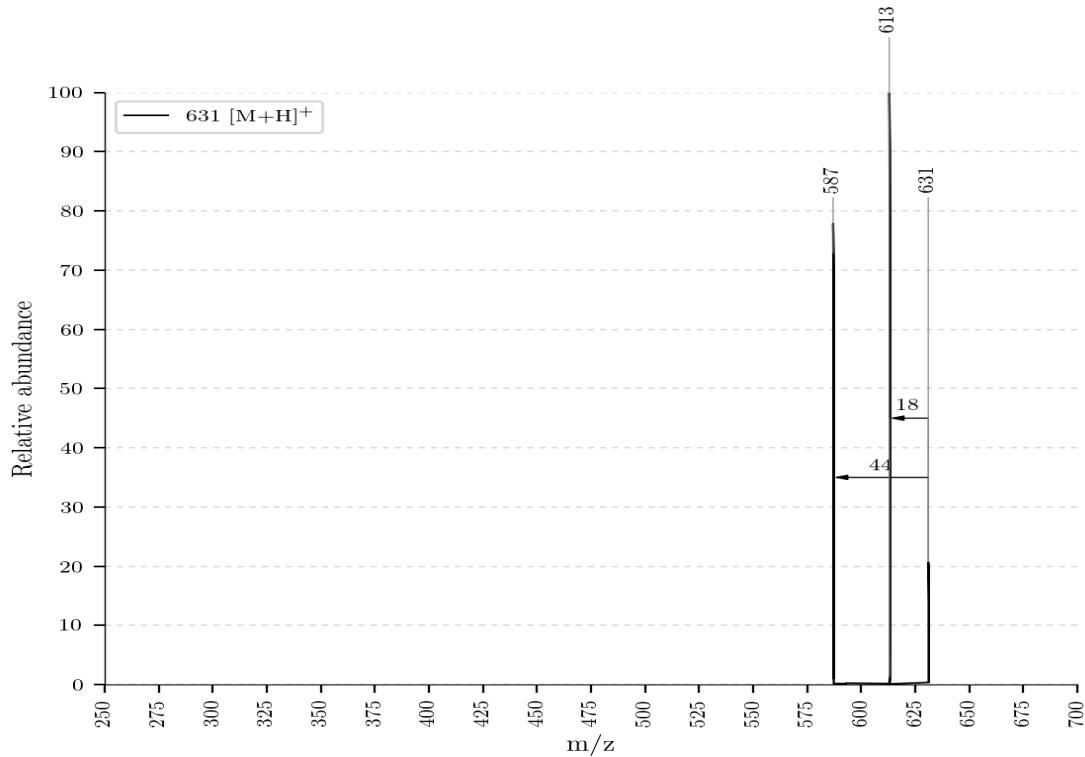


Abbildung 7.12.: ESI-MS Spektrum des Monomethylesters von Bo-DYCC bei $m/z = 631 [M+H]^+$

7. Strukturaufklärung der Chl-Kataboliten mit ESI-MS

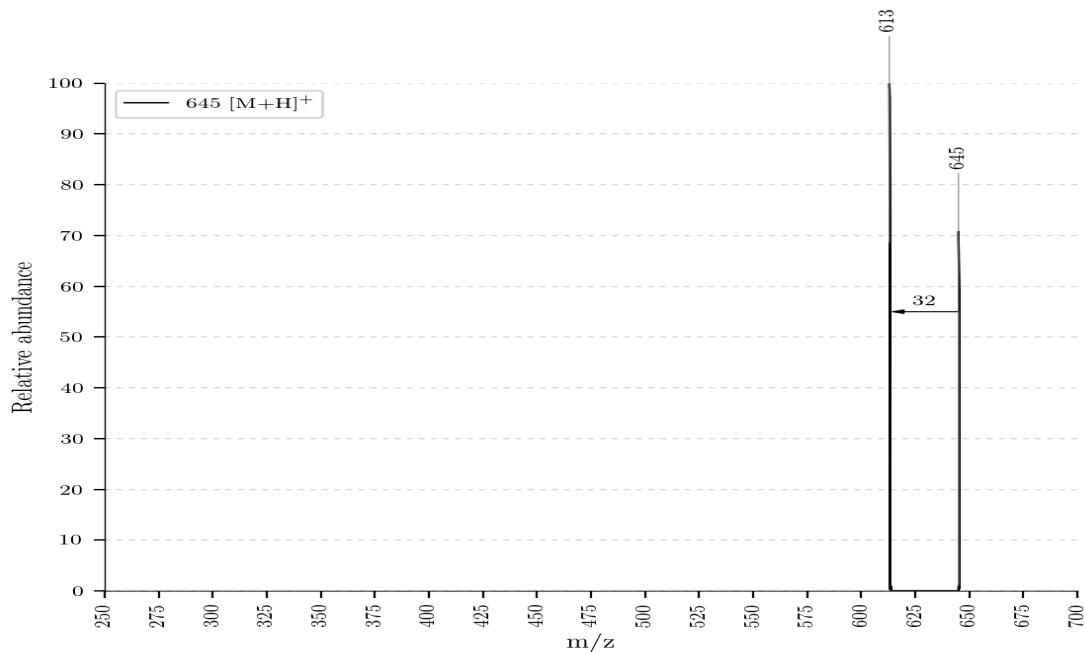


Abbildung 7.13.: ESI-MS Spektrum des Dimethylesters von Bo-DYCC bei $m/z = 645 [M+H]^+$

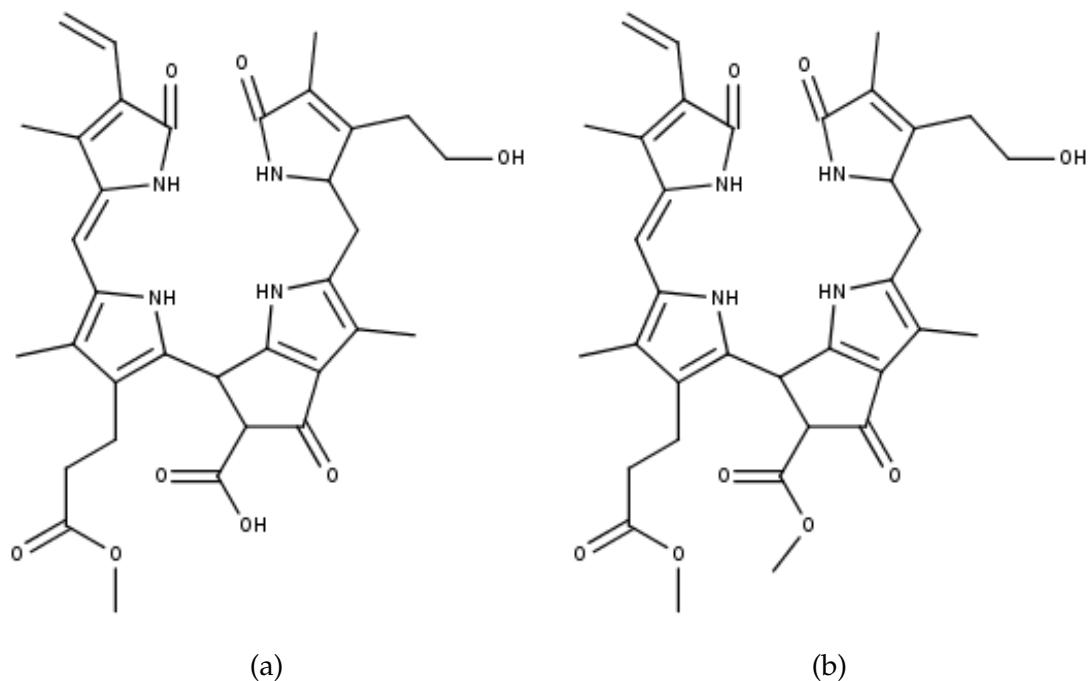


Abbildung 7.14.: Strukturvorschläge: (a) Monomethylierungsprodukt mit Summenformel $C_{34}H_{38}O_8N_4$, (b) Dimethylierungsprodukt mit Summenformel $C_{35}H_{41}O_8N_4$

7. Strukturaufklärung der Chl-Kataboliten mit ESI-MS

7.3.2. Reaktionsprodukt von Bo-DNCC

Das Reaktionsprodukt des Bo-DNCC wurde bei $m/z = 633 [M+H]^+$ gefunden. Aufgrund der Abspaltung von meoh wird angenommen, dass sich der Methylester an Position 8^2 ausbildet. Auf eine Diskussion aller in Abbildung 7.15 ersichtlichen Fragmentierungen wird ob der Menge verzichtet.

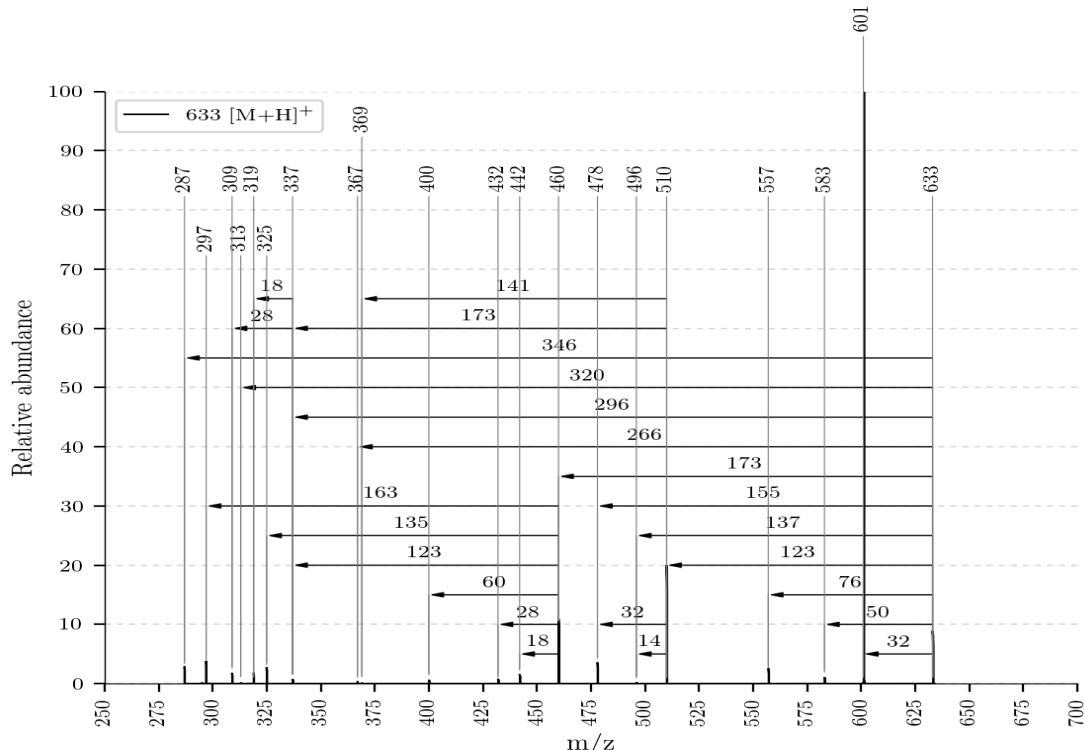


Abbildung 7.15.: ESI-MS Spektrum des Reaktionsproduktes des Bo-DNCC mit $m/z = 633 [M+H]^+$

7. Strukturaufklärung der Chl-Kataboliten mit ESI-MS

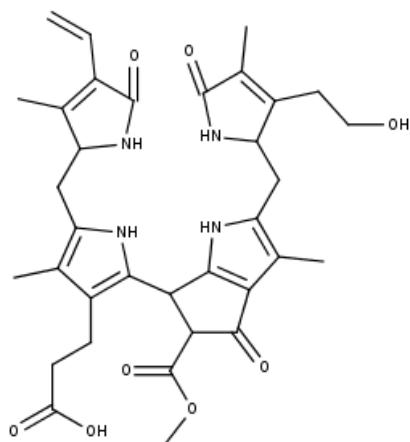


Abbildung 7.16.: Strukturvorschlag für das Reaktionsprodukt von Bo-DNCC mit Summenformel $C_{34}H_{41}O_8N_4$

7.3.3. Reaktionsprodukt von Bo-YCC

Das Reaktionsprodukt des Bo-DYCC konnte nicht gesammelt werden, weshalb keine weiteren strukturellen Informationen vorhanden sind. Es wurde lediglich im LC-MS Lauf identifiziert (Tabelle 6.2).

7.3.4. Reaktionsprodukt von Bo-NCC-1

Das Reaktionsprodukt des Bo-NCC-1 konnte ebenfalls nicht gesammelt und näher identifiziert werden. Es wurde aber im LC-MS identifiziert (Tabelle 6.2).

7. Strukturaufklärung der Chl-Kataboliten mit ESI-MS

7.3.5. Reaktionsprodukt von Bo-NCC-3

Das Reaktionsprodukt des Bo-NCC-3 konnte bei $m/z = 661 [M+H]^+$ identifiziert werden und zeigte, so wie das Reaktionsprodukt von Bo-DNCC eine große Anzahl an Fragmentierungen (Abbildung 7.17).

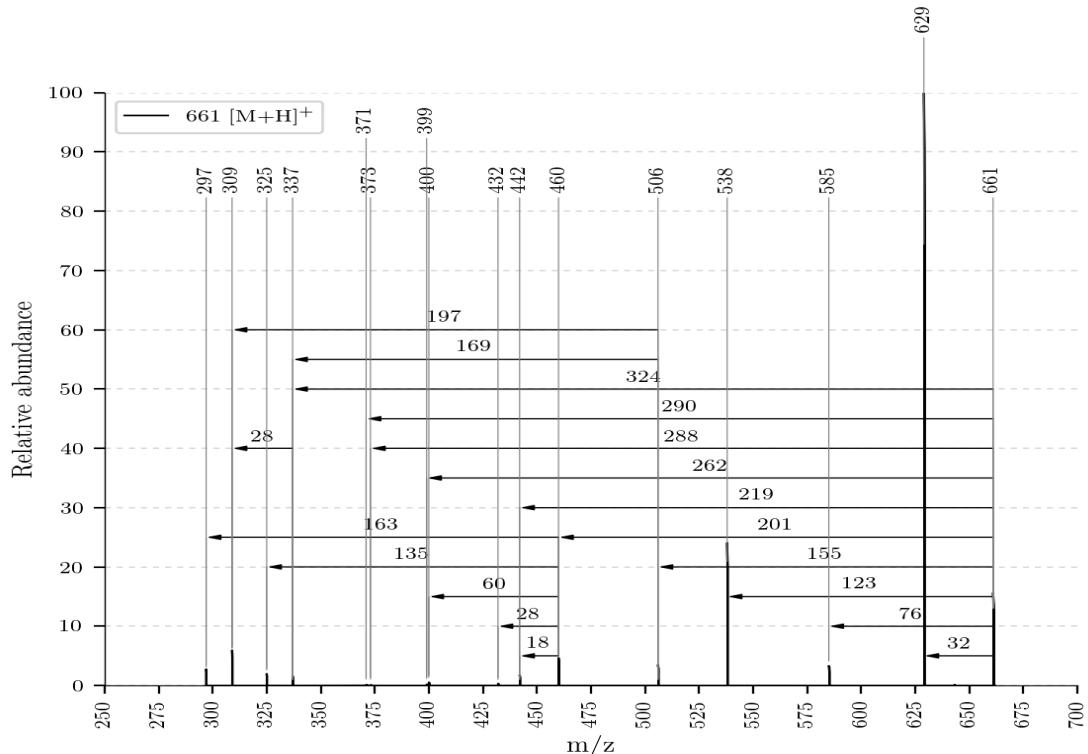


Abbildung 7.17.: ESI-MS des Reaktionsproduktes von Bo-NCC-3 bei $m/z = 661 [M+H]^+$

Ergebnisse und Diskussion

Verweise

Literaturverzeichnis

- [1] Bernhard Kräutler und Philippe Matile. „Solving the Riddle of Chlorophyll Breakdown“. In: *Accounts of Chemical Research* 32 (1 1999), S. 35–43. doi: 10.1021/ar970225n (siehe S. 3).
- [2] George A. F. Hendry, Jennifer D. Houghton und Stanley B. Brown. „The Degradation of Chlorophyll-a biological enigma“. In: *New Phytologist* 107 (2 1987), S. 255–302. doi: 10.1111/j.1469-8137.1987.tb00181.x (siehe S. 3).
- [3] Dr. Bernhard Kräutler, Dr. Bernhard Jaun, Karlheinz Bortlik, Maja Schellenberg und Philippe Matile. „Zum Rätsel des Chlorophyllabbaus: Die Konstitution eines secoporphinoiden Kataboliten“. In: *Angewandte Chemie* 103 (10 1991), S. 1354–1357. doi: 10.1002/ange.19911031020 (siehe S. 3, 4).
- [4] Simone Moser, Thomas Müller, Michael Oberhuber und Bernhard Kräutler. „Chlorophyll Catabolites – Chemical and Structural Footprints of a Fascinating Biological Phenomenon“. In: *European Journal of Organic Chemistry* 2009 (1 2009), S. 21–31. doi: 10.1002/ejoc.200890104 (siehe S. 3).
- [5] Bernhard Kräutler. „Der Chlorophyllabbau in höheren Pflanzen – Phyllobilin als weitverbreitete, aber kaum sichtbare Zeichen von Reifung, Seneszenz und Zelltod“. In: *Angewandte Chemie* 128 (16 2016), S. 4964–4990. doi: 10.1002/ange.201508928 (siehe S. 3).
- [6] Bernhard Kräutler und Stefan Hörtensteiner. „Chlorophyll Catabolites and the Biochemistry of Chlorophyll Breakdown“. In: *Chlorophylls and Bacteriochlorophylls: Biochemistry, Biophysics, Functions and Applications* 25 (2007), S. 237–260. doi: 10.1007/1-4020-4516-6_17 (siehe S. 4).
- [7] Univ.-Doz. Dr. Karl-Hans Ongania Walter Mühlecker, Prof. Dr. Bernhard Kräutler, Prof. Dr. Philippe Matile und Dr. Stefan Hörtensteiner. „Dem Chlorophyllabbau in Pflanzen auf der Spur – Konstitutionsaufklärung eines „fluoreszierenden“ Chlorophyllkataboliten“. In: *Angewandte Chemie* 109 (4 1997), S. 401–404. doi: 10.1002/ange.19971090424 (siehe S. 4).

Literaturverzeichnis

- [8] Simone Moser, Thomas Müller, Michael Oberhuber und Bernhard Kräutler. „Chlorophyll Catabolites – Chemical and Structural Footprints of a Fascinating Biological Phenomenon“. In: *European Journal of Organic Chemistry* 2009 (1 2008), S. 21–31. doi: 10.1002/ejoc.200800804 (siehe S. 4).
- [9] Jürgen Böcker. *Chromatographie*. 1. Auflage. Würzburg: Vogel, 1997. ISBN: 3-8023-1582-0 (siehe S. 5).
- [10] Edmond de Hoffmann und Vincent Stroobant. *Mass Spectrometry*. 3. Auflage. West Sussex: John Wiley und Sons, Ltd., 2007 (siehe S. 6, 7).
- [11] R. Graham Cooks, Zheng Ouyang, Zoltan Takats und Justin M. Wiseman. „Ambient Mass Spectrometry“. In: *Science* 311 (5767 2006), S. 1566–1570 (siehe S. 7).
- [12] He Wang, Jiangjiang Liu, R. Graham Cooks, Zheng Ouyang und Zoltan. „Paper Spray for Direct Analysis of Complex Mixtures Using Mass Spectrometry“. In: *Angewandte Chemie* 49 (5 2010), S. 877–880. doi: 10.1002/anie.200906314 (siehe S. 7).
- [13] Thomas Müller und R. Graham Cooks. „Differential Rapid Screening of Phytochemicals by Leaf Spray Mass Spectrometry“. In: *Analytische Methoden* 35 (3 2014), S. 919–924 (siehe S. 8, 15).
- [14] Naim Malaj, Zhen Ouyang, Giovanni Sindona und R. Graham Cooks. „Analysis of pesticide residues by leaf spray mass spectrometry“. In: *Analytische Methoden* 4 (7 2012), S. 1913–1919 (siehe S. 8).
- [15] Warren Weckesser. *savitzkygolay.py*. Version v0.17.1. scipy. Dez. 2017. URL: https://github.com/scipy/scipy/blob/v0.17.1/scipy/signal/_savitzky_golay.py#L228-L349 (siehe S. 11).
- [16] Matthias H. Roiser, Thomas Müller und Bernhard Kräutler. „Colorless Chlorophyll Catabolites in Senescent Florets of Broccoli (*Brassica oleracea* var. *italica*)“. In: *Journal of Agricultural and Food Chemistry* 63 (2015), S. 1385–1392. doi: 10.1021/jf5055326 (siehe S. 15, 17, 20).
- [17] Thomas Müller, Stefan Vergeiner und Bernhard Kräutler. „Structure elucidation of chlorophyll catabolites (phyllobilins) by ESI-mass spectrometry - Pseudo-molecular ions and fragmentation analysis of a nonfluorescent chlorophyll catabolite (NCC)“. In: *International Journal of Mass Spectrometry* 365-366 (2014), S. 48–55. doi: 10.1016/j.ijms.2013.12.028 (siehe S. 17, 22, 48).

Abbildungsverzeichnis

2.1. Abbauprozess des Chlorophylls, Quelle: http://www.organische-chemie.ch/chemie/2007nov/antioxidantien.shtml (Zugegriffen am: 05.11.2017)	3
2.2. Nummerierung von Phyllobilinen, Quelle: Mathias Scherl	4
5.1. MS Leafspray Versuchsaufbau, Quelle: Autor	12
5.2. MS Leafspray Versuchsaufbau Detailfotos, Quelle: Autor	13
5.3. MS Leafspray Blattvorbereitung, Quelle: Autor	14
5.4. ESI-MS Spektrum von Bo-NCC-1, Quelle: Autor	16
5.5. Strukturvorschlag von Bo-NCC-1 und Fragmentierungsdiagramm, Quelle: Autor	17
5.6. ESI-MS von Bo-NCC-3, Quelle: Autor	18
5.7. Strukturvorschlag von Bo-NCC-3 und Fragmentierungsdiagramm, Quelle: Autor	19
5.8. ESI-MS von Bo-DNCC, Quelle: Autor	20
5.9. ESI-MS von Bo-DNCC, Quelle: Autor	21
5.10. Strukturvorschlag von Bo-DNCC mit Fragmentierungsdiagramm, Quelle: Autor	21
5.11. Strukturvorschlag des Reaktionsproduktes von Bo-DNCC, Quelle: Autor	23
5.12. ESI-MS Spektrum des Reaktionsproduktes von Bo-DNCC, Quelle: Autor	23
5.13. Fragmentierungsdiagramme des Reaktionsproduktes von Bo-DNCC, Quelle: Autor	24
5.14. Vorschlag des Mechanismus der CH ₃ COOH Abspaltung, Quelle: Autor	25
5.15. ESI-MS des Reaktionsproduktes von Bo-NCC-3, Quelle: Autor . .	26
5.16. Strukturvorschlag des Reaktionsproduktes von Bo-NCC-3, Quelle: Autor	27
5.17. Fragmentierungsdiagramm des Reaktionsproduktes von Bo-DNCC, Quelle: Autor	27
5.18. ESI-MS des Reaktionsproduktes von Bo-NCC-1, Quelle: Autor . .	28

Abbildungsverzeichnis

5.19. Strukturvorschlag des Reaktionsproduktes von Bo-NCC-1, Quelle: Autor	29
5.20. Fragmentierungsdiagramm des Reaktionsproduktes von Bo-NCC-1, Quelle: Autor	30
6.1. HPLC Chromatogramm vor der Reaktion, Quelle: Autor	34
6.2. LC-MS Chromatogramm vor der Reaktion, Quelle: Autor	34
6.3. LC-MS Chromatogramm vor der Reaktion - Aufspaltung der Signale, Quelle: Autor	36
6.4. Online-UV/Vis Spektren mit der Charakteristik eines NCC bei 27.10min., eines DNCC bei 29.75min. sowie eines YCC bei 30.94min., Quelle: Autor	38
6.5. HPLC Chromatogramm nach 3h Reaktionsdauer, Quelle: Autor	39
6.6. LC-MS Chromatogramm nach 3h Reaktionsdauer, Quelle: Autor	40
6.7. LC-MS Chromatogramm nach 3h Reaktionsdauer - Aufspaltung, Quelle: Autor	41
6.8. Online-UV/Vis Spektren mit der Charakteristik eines YCC bei 33.98min., eines DNCC bei 37.09min. eines NCC bei 38.94min. sowie eines DNCC bei 40.03min., Quelle: Autor	43
6.9.	44
6.10. Online-UV/Vis Spektren mit der Charakteristik eines YCC bei 47.28min., Quelle: Autor	44
7.1. ESI-MS Spektrum von Bo-DYCC, Quelle: Autor	46
7.2. Strukturvorschlag von Bo-DYCC, Quelle: Autor	46
7.3. Abspaltungsmechanismus von Ring C und Ring D bei Bo-DYCC, Quelle: Autor	47
7.4. ESI-MS Spektrum von Bo-DNCC, Quelle: Autor	48
7.5. Abspaltungsmechanismus von Ring D und Ring A und Mesomer 1, Quelle: Autor	49
7.6. 2 Mesomere für potentielle Abspaltungsprodukte von Bo-DNCC, Quelle: Autor	49
7.7. ESI-MS Spektrum von Bo-YCC, Quelle: Autor	50
7.8. ESI-MS Spektrum von Bo-NCC-3, Quelle: Autor	51
7.9. Strukturvorschlag von Bo-NCC-3 und Vorschlag für Mechanismus der Abspaltung von Ring D, Quelle: Autor	52
7.10. vorgeschlagene Keto-/Enoltautomerie der Ring D Abspaltung von Bo-NCC-3, Quelle: Autor	52
7.11. ESI-MS Spektrum von Bo-NCC-1, Quelle: Autor	53
7.12. ESI-MS Spektrum des Monomethylesters von Bo-DYCC, Quelle: Autor	54

Abbildungsverzeichnis

7.13. ESI-MS Spektrum des Diethylesters von Bo-DYCC, Quelle: Author	55
7.14. Strukturvorschläge für das Mono- und Dimethylierungsprodukt des Bo-DYCC, Quelle: Autor	55
7.15. ESI-MS Spektrum des Reaktionsproduktes von Bo-DNCC, Quelle: Autor	56
7.16. Strukturvorschlag für das Reaktionsprodukt von Bo-DNCC, Quel- le: Autor	57
7.17. ESI-MS des Reaktionsproduktes von Bo-NCC-3, Quelle: Author .	58

Abkürzungsverzeichnis

bzw. beziehungsweise. 7

ca. circa. 10, 11, 31

Chl-Katabolit Chlorophyll Kataboliten. 2, 5–8, 10, 11, 17, 22, 31–37, 39, 40, 45, 51, 54

CI Chemical Ionization. 6

CID Collision induced Dissociation. 2

DNCC Decarboxylated-Non flourescent Chlorophyllic Catabolite. 7, 32, 35, 36, 38, 43

EI Electron Ionization. 6

EPPI Eppendorf Reaktionsgefäß. 6

ESI Electrosprayionisation. 6, 7

FI Field Ionization. 6

HPLC High performance liquid chromatography. 2, 5–7, 11, 31–36, 39, 45

LC-MS Liquid Chromatography-Mass Spectrometry. 5, 7, 31, 33, 34, 36, 37, 39–41, 45, 57

LM Lösungsmittel. 5, 7, 12–14, 22

m/z Masse pro Ladung. 6, 15

MeOH Methanol. 14, 16, 18, 19, 22, 28, 31, 35, 39, 42, 54

N2 Stickstoff. 7

nAb. noch Aufklärungsbedarf. 16–20, 22, 24, 26

NCC Non flourescent Chlorophyllic Catabolite. 4, 7, 32, 34, 35, 38, 43

NKE normalisierten Kollisionsenergie (in %). 11, 17–19, 22, 24–26, 28

RP Reversed-phase. 5

u.a. unter anderem. 17, 22

YCC Flourescent Chlorophyllic Catabolite. 7, 19, 32, 36, 38, 43, 44

z.B. zum Beispiel. 2–4, 6, 8, 22

Anhang

Anhang A.

Python Code-Listings für die Erstellung der Grafiken

Im Folgenden findet sich der Programmcode, den ich im Rahmen der Datenauswertung für meine VWA schrieb. Die Referenzen für diverse verwendete Bibliotheken befinden sich im jeweiligen Codeteil.

A.1. Programm zur Erstellung von Chromatogrammen für LC-ESI-MS

```
1 # coding: utf-8
2
3 # # Generate LC-MS chromatograms for analysis of overlaying effects
4
5 def cc(arg):
6     return mcolors.to_rgba(arg, alpha=0.6)
7
8 # Detecting peaks from dataset and scaling:
9
10 def detect_peaks(x, mph=None, mpd=1, threshold=0, edge='rising',
11                  kpsh=False, valley=False, show=False, ax=None):
12
13     """Detect peaks in data based on their amplitude and other
14     features.
15
16     Parameters
17     -----
18     x : 1D array_like
19         data.
20     mph : {None, number}, optional (default = None)
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
21     detect peaks that are greater than minimum peak height.
22     mpd : positive integer, optional (default = 1)
23         detect peaks that are at least separated by minimum peak
24         distance (in
25             number of data).
26         threshold : positive number, optional (default = 0)
27             detect peaks (valleys) that are greater (smaller) than '
28             threshold '
29                 in relation to their immediate neighbors.
30             edge : {None, 'rising', 'falling', 'both'}, optional (default =
31             'rising')
32                 for a flat peak, keep only the rising edge ('rising'), only
33                 the
34                     falling edge ('falling'), both edges ('both'), or don't
35                     detect a
36                     flat peak (None).
37             kpsh : bool, optional (default = False)
38                 keep peaks with same height even if they are closer than 'mpd'
39             .
40             valley : bool, optional (default = False)
41                 if True (1), detect valleys (local minima) instead of peaks.
42             show : bool, optional (default = False)
43                 if True (1), plot data in matplotlib figure.
44             ax : a matplotlib.axes.Axes instance, optional (default = None).
45
46     Returns
47
48     _____
49     ind : 1D array-like
50         indeces of the peaks in 'x'.
51
52     Notes
53
54     _____
55     The detection of valleys instead of peaks is performed internally
56     by simply
57     negating the data: 'ind_valleys = detect_peaks(-x)'
58
59     The function can handle NaN's
60
61     See this IPython Notebook [1]_.
62
63     References
64
65     _____
66     .. [1] http://nbviewer.ipython.org/github/demotu/BMC/blob/master/
67     notebooks/DetectPeaks.ipynb
68
69     Examples
70
71     _____
72     >>> from detect_peaks import detect_peaks
73     >>> x = np.random.randn(100)
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
62 >>> x[60:81] = np.nan
63 >>> # detect all peaks and plot data
64 >>> ind = detect_peaks(x, show=True)
65 >>> print(ind)
66
67 >>> x = np.sin(2*np.pi*5*np.linspace(0, 1, 200)) + np.random.
randn(200)/5
68 >>> # set minimum peak height = 0 and minimum peak distance = 20
69 >>> detect_peaks(x, mph=0, mpd=20, show=True)
70
71 >>> x = [0, 1, 0, 2, 0, 3, 0, 2, 0, 1, 0]
72 >>> # set minimum peak distance = 2
73 >>> detect_peaks(x, mpd=2, show=True)
74
75 >>> x = np.sin(2*np.pi*5*np.linspace(0, 1, 200)) + np.random.
randn(200)/5
76 >>> # detection of valleys instead of peaks
77 >>> detect_peaks(x, mph=0, mpd=20, valley=True, show=True)
78
79 >>> x = [0, 1, 1, 0, 1, 1, 0]
80 >>> # detect both edges
81 >>> detect_peaks(x, edge='both', show=True)
82
83 >>> x = [-2, 1, -2, 2, 1, 1, 3, 0]
84 >>> # set threshold = 2
85 >>> detect_peaks(x, threshold = 2, show=True)
86 """
87
88 x = np.atleast_1d(x).astype('float64')
89 if x.size < 3:
90     return np.array([], dtype=int)
91 if valley:
92     x = -x
93 # find indices of all peaks
94 dx = x[1:] - x[:-1]
95 # handle NaN's
96 indnan = np.where(np.isnan(x))[0]
97 if indnan.size:
98     x[indnan] = np.inf
99     dx[np.where(np.isnan(dx))[0]] = np.inf
100 ine, ire, ife = np.array([[], [], []], dtype=int)
101 if not edge:
102     ine = np.where((np.hstack((dx, 0)) < 0) & (np.hstack((0, dx))
> 0))[0]
103 else:
104     if edge.lower() in ['rising', 'both']:
105         ire = np.where((np.hstack((dx, 0)) <= 0) & (np.hstack((0,
dx)) > 0))[0]
106         if edge.lower() in ['falling', 'both']:
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```

107         ife = np.where((np.hstack((dx, o)) < o) & (np.hstack((o,
108 dx)) >= o))[o]
109     ind = np.unique(np.hstack((ine, ire, ife)))
110     # handle NaN's
111     if ind.size and indnan.size:
112         # NaN's and values close to NaN's cannot be peaks
113         ind = ind[np.in1d(ind, np.unique(np.hstack((indnan, indnan-1,
114 indnan+1))), invert=True)]
115     # first and last values of x cannot be peaks
116     if ind.size and ind[o] == o:
117         ind = ind[1:]
118     if ind.size and ind[-1] == x.size-1:
119         ind = ind[:-1]
120     # remove peaks < minimum peak height
121     if ind.size and mph is not None:
122         ind = ind[x[ind] >= mph]
123     # remove peaks - neighbors < threshold
124     if ind.size and threshold > 0:
125         dx = np.min(np.vstack([x[ind]-x[ind-1], x[ind]-x[ind+1]]),
126 axis=o)
127         ind = np.delete(ind, np.where(dx < threshold)[o])
128     # detect small peaks closer than minimum peak distance
129     if ind.size and mpd > 1:
130         ind = ind[np.argsort(x[ind])][::-1] # sort ind by peak
131         height
132         idel = np.zeros(ind.size, dtype=bool)
133         for i in range(ind.size):
134             if not idel[i]:
135                 # keep peaks with the same height if kpsh is True
136                 idel = idel | (ind >= ind[i] - mpd) & (ind <= ind[i]
137 + mpd) & (x[ind[i]] > x[ind] if kpsh else True)
138             idel[i] = o # Keep current peak
139         # remove the small peaks and sort back the indices by their
140         # occurrence
141         ind = np.sort(ind[~idel])
142
143     if show:
144         if indnan.size:
145             x[indnan] = np.nan
146         if valley:
147             x = -x
148         _plot(x, mph, mpd, threshold, edge, valley, ax, ind)
149
150     return ind
151
152
153 def _plot(x, mph, mpd, threshold, edge, valley, ax, ind):
154     """Plot results of the detect_peaks function, see its help."""

```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
149 try:
150     import matplotlib.pyplot as plt
151 except ImportError:
152     print('matplotlib is not available.')
153 else:
154     if ax is None:
155         _, ax = plt.subplots(1, 1, figsize=(8, 4))
156
157     ax.plot(x, 'b', lw=1)
158     if ind.size:
159         label = 'valley' if valley else 'peak'
160         label = label + 's' if ind.size > 1 else label
161         ax.plot(ind, x[ind], '+', mfc=None, mec='r', mew=2, ms=8,
162                 label='%d %s' % (ind.size, label))
163         ax.legend(loc='best', framealpha=.5, numpoints=1)
164     ax.set_xlim(-.02*x.size, x.size*.02-1)
165     ymin, ymax = x[np.isfinite(x)].min(), x[np.isfinite(x)].max()
166     yrangle = ymax - ymin if ymax > ymin else 1
167     ax.set_ylim(ymin - 0.1*yrangle, ymax + 0.1*yrangle)
168     ax.set_xlabel('Data #', fontsize=14)
169     ax.set_ylabel('Amplitude', fontsize=14)
170     mode = 'Valley detection' if valley else 'Peak detection'
171     ax.set_title("%s (mph=%s, mpd=%d, threshold=%s, edge=%s)" %
172                  (mode, str(mph), mpd, str(threshold), edge))
173     # plt.grid()
174     plt.show()
175
176
177 # Import needed modules:
178
179 import numpy as np
180 import pandas as pd
181 import matplotlib.pyplot as plt
182 from matplotlib import colors as mcolors
183 from scipy.signal import argrelextrema
184 from scipy.signal import find_peaks_cwt
185
186 from numpy import trapz
187
188
189 # Load and specify dataset:
190
191 data = pd.io.parsers.read_csv("Kuerbis_Analyse_7min_LC-ESI-MS_12min.csv")
192
193 helpvalue = ['617', '619', '629', '631', '633.23', '633.29', '645', '',
194     '647.27', '647.31', '659.23', '659.27', '661', '675', '793', '795',
195     '807', '809', '821']
#helpvalue = ['619', '633.23', '645', '647', '675', '821']
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
195 xs = data[['Zeit']].values
196 xs = np.hstack(xs)
197
198 list_x_values = []
199 list_y_values = []
200 list_catabolite = []
201 list_isomer = []
202
203
204 # Retrieve assumed constants for developing an equilibrium theory (
205 # though not working as found out):
206
207 ys = 0
208 i = 0
209 highest_peak = []
210
211 fig = plt.figure(figsize=(20,10))
212 ax = plt.axes()
213
214 area_617 = 0
215 area_619 = 0
216 area_631 = 0
217 area_63329 = 0
218 area_645 = 0
219 area_647 = 0
220 area_659 = 0
221 area_661 = 0
222 area_675 = 0
223 area_793 = 0
224 area_795 = 0
225 area_807 = 0
226 area_809 = 0
227 area_821 = 0
228
229 height_617 = 0
230 height_619 = 0
231 height_631 = 0
232 height_63329 = 0
233 height_645 = 0
234 height_647 = 0
235 height_659 = 0
236 height_661 = 0
237 height_675 = 0
238 height_793 = 0
239 height_795 = 0
240 height_807 = 0
241 height_809 = 0
242 height_821 = 0
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
243 distance_for_integration = 0.1
244
245 for h in helpvalue:
246     ys = data[[helpvalue[i]]].values
247     ys = np.hstack(ys)
248
249     highest_peak.append(ys.max())
250
251     if (i == 0):
252         print(helpvalue[i])
253         area_617 = trapz(ys, dx=distance_for_integration)
254         height_617 = ys.max()
255     elif (i == 1):
256         print(helpvalue[i])
257         area_619 = trapz(ys, dx=distance_for_integration)
258         height_619 = ys.max()
259     elif (i == 3):
260         print(helpvalue[i])
261         area_631 = trapz(ys, dx=distance_for_integration)
262         height_631 = ys.max()
263     elif (i == 5):
264         print(helpvalue[i])
265         area_63329 = trapz(ys, dx=distance_for_integration)
266         height_63329 = ys.max()
267     elif (i == 6):
268         print(helpvalue[i])
269         area_645 = trapz(ys, dx=distance_for_integration)
270         height_645 = ys.max()
271     elif (i == 7):
272         print(helpvalue[i])
273         area_647 = trapz(ys, dx=distance_for_integration)
274         height_647 = ys.max()
275     elif (i == 10):
276         print(helpvalue[i])
277         area_659 = trapz(ys, dx=distance_for_integration)
278         height_659 = ys.max()
279     elif (i == 11):
280         print(helpvalue[i])
281         area_661 = trapz(ys, dx=distance_for_integration)
282         height_661 = ys.max()
283     elif (i == 12):
284         print(helpvalue[i])
285         area_675 = trapz(ys, dx=distance_for_integration)
286         height_675 = ys.max()
287     elif (i == 13):
288         print(helpvalue[i])
289         area_793 = trapz(ys, dx=distance_for_integration)
290         height_793 = ys.max()
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
292 elif (i == 14):
293     print(helpvalue[i])
294     area_795 = trapz(ys, dx=distance_for_integration)
295     height_795 = ys.max()
296 elif (i == 15):
297     print(helpvalue[i])
298     area_807 = trapz(ys, dx=distance_for_integration)
299     height_807 = ys.max()
300 elif (i == 16):
301     print(helpvalue[i])
302     area_809 = trapz(ys, dx=distance_for_integration)
303     height_809 = ys.max()
304 elif (i == 17):
305     print(helpvalue[i])
306     area_821 = trapz(ys, dx=distance_for_integration)
307     height_821 = ys.max()
308
309     i = i+1
310
311 Ka_617 = area_631/area_617
312 Kh_617 = height_631/height_617
313 Ka_619 = area_63329/area_619
314 Kh_619 = height_63329/height_619
315 Ka_645 = area_659/area_645
316 Kh_645 = height_659/height_645
317 Ka_647 = area_661/area_647
318 Kh_647 = height_661/height_647
319 Ka2_647 = area_675/area_647
320 Kh2_647 = height_675/height_647
321 Ka_793 = area_807/area_793
322 Kh_793 = height_807/height_793
323 Ka2_793 = area_821/area_793
324 Kh2_793 = height_821/height_793
325 Ka_795 = area_809/area_795
326 Kh_795 = height_809/height_795
327
328 print('Ka_617')
329 print(Ka_617)
330 print(Kh_617)
331 print('Ka_619')
332 print(Ka_619)
333 print(Kh_619)
334 print('Ka_645')
335 print(Ka_645)
336 print(Kh_645)
337 print('Ka_647')
338 print(Ka_647)
339 print(Kh_647)
340 print('Ka2_647')
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
341 print(Ka2_647)
342 print(Kh2_647)
343 print('Ka_793')
344 print(Ka_793)
345 print(Kh_793)
346 print('Ka2_793')
347 print(Ka2_793)
348 print(Kh2_793)
349 print('Ka_795')
350 print(Ka_795)
351 print(Kh_795)

352
353 highest_peak = np.array(highest_peak).max()
354
355 ys = o
356 i=0
357 previous_mass = o
358
359 for h in helpvalue:
360     ys = data[[helpvalue[i]]].values
361     ys = np.hstack(ys)
362
363     y = []
364     for x in ys:
365         y.append(((x/highest_peak)*100)+previous_mass+10)
366
367     y = np.array(y)
368
369     previous_mass = y.max()
370     x_max_arg = y.argmax()
371
372     #local_maxima = argrelextrema(y, np.greater, order=200)
373     #local_maxima = find_peaks_cwt(y, np.arange(1, 350))
374     local_maxima = detect_peaks(y, mph=0.04, mpd=350)
375     #first_isomer = np.partition(np.array(y).flatten(), -1)[-1]
376     #first_isomer = np.round(np.array(np.hstack(y)[local_maxima])).argsort()[-o:]
377     #print(first_isomer)
378     #y_help = np.sort(y[local_maxima])
379
380     line = plt.plot(xs, y, color = 'black', label = h + '[M+H]^+')
381     ax.annotate(xy=(xs[x_max_arg], previous_mass), s = h + r' $[M+H]^{+'} + $ (' + str(np.round(xs[x_max_arg], 1)) + ' min.)')
382
383     list_x_values.append(xs[x_max_arg])
384     list_y_values.append(previous_mass)
385     list_catabolite.append(h)
386
387
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
388 #u=1
389 #for x in xs[local_maxima]:
390     #ax.annotate(xy=(x, y[np.where(x)]), s = h + r' $[M+H]^+$-
391     #Isomer '+str(u)+('+'+str(np.round(x, 1))+ ' min.)')
392
393     #list_x_values.append(x)
394     #list_y_values.append(y[np.where(x)])
395     #list_catabolite.append(h)
396     #list_isomer.append(u)
397
398     #u = u+1
399
400
401 plt.yticks([])
402 plt.xlabel('Retention Time (min.)')
403 plt.xlim(0,65)
404 plt.savefig('Kuerbis-Analyse_Reaktion3h_LC-ESI-MS.png')
405 plt.show()
406
407
408 # Load and specify another dataset:
409
410 data = pd.io.parsers.read_csv("Kuerbis-Analyse_7min_LC-ESI-
411 MS_12min_Spektrum.csv")
412
413 xs = data[['Time']].values
414 xs = np.hstack(xs)
415
416 ys = data[['Intensity']].values
417 ys = np.hstack(ys)
418
419 # Print whole chromatogram for completion purpose:
420
421 fig = plt.figure(figsize=(20,10))
422 ax = plt.axes()
423
424
425 #ax.annotate(xy=(xs[x_max_arg], previous_mass), s = h + r' $[M+H]^+$-
426     #('+'+str(np.round(xs[x_max_arg], 1))+ ' min.)')
427
428 highest_peak = np.array(ys).max()
429 y = []
430 for z in ys:
431     y.append(((z/highest_peak)*100))
432 ys = y
433 line = plt.plot(xs, ys, color = 'black', label = '')
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
434 highest_peak = np.array(list_y_values).max()
435 helper = []
436 for t in list_y_values:
437     helper.append(((t/highest_peak)*100))
438 list_y_values = helper
439
440 i = 0
441 for index in list_x_values:
442     h = list_catabolite[i]
443     #u = list_isomer[i]
444     text = h + r' $[M+H]^+' + str(np.round(index, 1)) + ' min.'
445     plt.annotate(
446         text, xy=(index, 0), xycoords='data',
447         xytext=(index, list_y_values[i]), textcoords='data',
448         rotation=0, size=12, horizontalalignment='center',
449         verticalalignment='bottom',
450         arrowprops=dict(arrowstyle='-', color="#808080",
451                         linewidth=0.4, shrinkA=0.05, shrinkB=1))
451     i = i+1
452
453 #ax.annotate(xy=(x, y[np.where(x)]), s = h + r' $[M+H]^+' + Isomer +
454 #              str(u) + ' (' + str(np.round(x, 1)) + ' min.)')
455
456 plt.xlabel('Retention Time (min.)')
457 plt.ylabel('Relative abundance')
458 plt.xlim(0,65)
459 plt.savefig('Kuerbis_Analyse_Ganzes_Spektrum.png')
460 plt.show()
```

content/Anhang/Code/VWA_Analyse_LC-ESI-MS.py

A.2. Programm zur Erstellung von MS-Spektren

```
1 # coding: utf-8
2
3 # # Drawing nice-looking MS-spectra with fragmentation paths
4
5 # Import needed modules:
6
7 import pandas as pd
8 import numpy as np
9 import math as mt
10 import matplotlib.pyplot as plt
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
12 from matplotlib import rcParams
13
14
15 # Define filetypes and other general stuff:
16 filetype = "png"          # "png" or "pdf"
17 """str: Filetype of the outputfile
18
19 Tested with "pdf" and "png"
20 """
21
22
23 figdpi = 200
24 """int: DPI of the image output file
25 """
26
27
28 # correct DPI of figure if using pdf
29 if filetype.lower() == "pdf":
30     figdpi = 72
31
32 params = {'backend': 'pdf',
33            'figure.dpi': figdpi, # 72 for pdf
34            'axes.labelsize': 10,
35            'font.size': 10,
36            'legend.fontsize': 8,
37            'legend.frameon': True,
38            'xtick.labelsize': 8,
39            'ytick.labelsize': 8,
40            'font.family': 'serif',
41            'text.usetex': True,
42            'text.latex.unicode': True,
43            'axes.linewidth': 0.5,
44            'xtick.major.size': 4, # major tick size in points
45            'xtick.minor.size': 2, # minor tick size in points
46            'xtick.direction': 'out',
47            'ytick.major.size': 4, # major tick size in points
48            'ytick.minor.size': 2, # minor tick size in points
49            'ytick.direction': 'out',
50        }
51 plt.rcParams.update(params)
52
53
54 # Retrieve the maximum peak on which to scale all the others:
55
56 def __get_max_peak(x_value, raw_values):
57     """Search for the local peak next to the x-value.
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
60     The value searches in the near area of the given x-value for the
61     maximum (local)
62     peak, which corresponds to the data.
63
64     Args:
65         x_value (float): m/z value (x-coordinate) of the point, to be
66         annotated
67         raw_values (Pandas.DataFrame): Data values, which should be
68         annotated
69
70     Returns:
71         y value of the peak, next to the given x value
72
73     """
74     #print(x_value)
75     #print(raw_values[raw_values["m/z"] == x_value])
76     raw_values_index = raw_values[raw_values["m/z"] == x_value].index
77     [0]
78
79     #print(raw_values_index)
80     value, index = float(raw_values.loc[raw_values_index - 1, "intensity_normalized"]), raw_values_index - 1
81
82     #for z in range(-5, 15):
83     #if float(raw_values.loc[raw_values_index + z, "intensity_normalized"]) > value:
84     #    value, index = float(raw_values.loc[raw_values_index + z, "intensity_normalized"]), raw_values_index + z
85     #return value
86
87
88 # Retrieve the positions of the points:
89
90 def annotate_point(x1, y_pos=0, text=' ', raw_values=None):
91     """Annotate a specific point.
92
93     Annotate a point with a label. The label will be placed
94     vertically
95     with an additional line.
96
97     The function uses the data values and searches for the peak of
98     the
99     value to be annotated. Therefore the ''raw_values'' parameter is
100    used
101
102    Args:
103        x1 (float): m/z value (x-coordinate) of the point, to be
104        annotated
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
97     y_pos (Optional[float]): Position of the label beginning (y-
98     coordinate).
99         The value uses the scale of the datapoints.
100    text (str): Label text
101    raw_values (Pandas.DataFrame): Data values , which should be
102    annotated
103
104    """
105
106    if raw_values is not None:
107
108        if __get_max_peak(x1, raw_values) < (y_pos -
109            delta_point_annotate_line_pixel):
110            xy_text = (x1, y_pos)
111        else:
112            xy_text = (x1, __get_max_peak(x1, raw_values) +
113            delta_point_annotate_line_pixel)
114
115            plt.annotate(
116                text, xy=(x1, __get_max_peak(x1, raw_values) +
117                delta_point_annotate_line_pixel), xycoords='data',
118                xytext=xy_text, textcoords='data',
119                rotation=90, size=8, horizontalalignment='center',
120                verticalalignment='bottom',
121                arrowprops=dict(arrowstyle='-', color="#808080",
122                linewidth=0.4, shrinkA=0.05, shrinkB=1))
123
124
125 # Calculate distance between the annotated points:
126
127 def annotate_distance(x1=0, x2=0, y_pos=0, text=' ', raw_values=None,
128     rotate_text=0):
129     """Annotate the distance between two peaks
130
131     Annotate the distance between two given peaks. The text can be
132     placed with a
133     given angle.
134
135     Args:
136         x1 (float): m/z value (x-coordinate) of the left point
137         x2 (float): m/z value (x-coordinate) of the right point
138         y_pos (Optional[float]): Position of the label beginning (y-
139             coordinate).
140             The value uses the scale of the datapoints.
141         text (str): Label text
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
135     raw_values (Pandas.DataFrame): Data values , which should be  
136     annotated  
137     rotate_text (Optional[int]): Rotation of the label , should be  
138     0 or 90  
139     """  
140     delta_point_annotate_line_pixel = 1  
141  
142     if raw_values is not None:  
143         if __get_max_peak(x1, raw_values) < (y_pos -  
144             delta_point_annotate_line_pixel):  
145             plt.annotate(  
146                 '', xy=(x1, __get_max_peak(x1, raw_values) +  
147                     delta_point_annotate_line_pixel), xycoords='data',  
148                 xytext=(x1, y_pos), textcoords='data',  
149                 arrowprops=dict(arrowstyle='-' , color="#808080",  
150                     linewidth=0.4, shrinkA=0.05, shrinkB=0.05))  
151  
152         if __get_max_peak(x2, raw_values) < (y_pos -  
153             delta_point_annotate_line_pixel):  
154             plt.annotate(  
155                 '', xy=(x2, __get_max_peak(x2, raw_values) +  
156                     delta_point_annotate_line_pixel), xycoords='data',  
157                 xytext=(x2, y_pos), textcoords='data',  
158                 arrowprops=dict(arrowstyle='-' , color="#808080",  
159                     linewidth=0.4, shrinkA=0.05, shrinkB=0.05))  
160  
161             plt.annotate(  
162                 text , xy=((x1 if x1 <= x2 else x2) + mt.fabs((x1 - x2)) / 2,  
163                     y_pos), xycoords='data',  
164                     rotation=rotate_text , size=8,  
165                     horizontalalignment='center' , verticalalignment='bottom',  
166                     xytext=(0, 2), textcoords='offset points')  
167  
168 # Helper functions  
169  
170 # Set the size and margins of the figure:  
171 def __figsize_and_margins(plotsize , subplots=(1, 1) , **  
    absolute_margins):
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
172     """Determine figure size and margins from plot size and absolute
173     margins
174
175     Args:
176         plotsize: (width, height) of plot area in inch
177         subplots: (nrows, ncols) of subplots
178         left, right, top, bottom: absolute margins around plot area
179         wspace, hspace: width and height spacing between subplots
180
181     Returns:
182         size: figure size for figsize argument of figure()
183         margins: relative margins dict suitable for subplots_adjust
184         ()
185
186     Example: making 2x2 grid of 3" square plots with specific
187     spacings:
188
189     sz, rm = figsize_and_margins((3,3), (2,2),
190                                 left=.1, right=.5,
191                                 top=.5, bottom=.1,
192                                 wspace=.5, hspace
193                                 =.5)
194     figure(figsize=sz)
195     subplots_adjust(**rm)
196     subplot(221); subplot(222)
197     subplot(223); subplot(224)
198
199     Source: http://scipy-central.org/item/65/1/absolute-plot-size-and
200     -margins-in-matplotlib
201     """
202
203     pw, ph = plotsize
204     nr, nc = subplots
205     amarg = absolute_margins
206     # dictionary for relative margins
207     # initialize from rcParams with margins not in amarg
208     rmarg = dict((m, rcParams['figure.subplot.' + m])
209                  for m in ('left', 'right', 'top', 'bottom', 'wspace',
210                           'hspace'))
211     if m not in amarg:
212         )
213
214     # subplots_adjust wants wspace and hspace relative to plotsize:
215     if 'wspace' in amarg:
216         rmarg['wspace'] = float(amarg['wspace']) / pw
217     if 'hspace' in amarg:
218         rmarg['hspace'] = float(amarg['hspace']) / ph
219
220     # in terms of the relative margins:
221     # width * (right - left)
222     #     = ncols * plot_width + (ncols - 1) * wspace * plot_width
223     # height * (top - bottom)
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```

215     #      = nrows * plot_height + (nrows - 1) * hspace * plot_height
216     # solve for width and height, using absolute margins as necessary
217     :
218         width = float((nc + (nc - 1) * rmarg['wspace']) * pw + amarg.get(
219             'left', 0) + amarg.get('right', 0)) / (
220                 rmarg.get('right', 1) - rmarg.get('left', 0))
221         height = float((nr + (nr - 1) * rmarg['hspace']) * ph + amarg.get(
222             ('top', 0) + amarg.get('bottom', 0)) / (
223                 rmarg.get('top', 1) - rmarg.get('bottom', 0)))
224
225     # now we can get any remaining relative margins
226     if 'left' in amarg:
227         rmarg['left'] = float(amarg['left']) / width
228     if 'right' in amarg:
229         rmarg['right'] = 1 - float(amarg['right']) / width
230     if 'top' in amarg:
231         rmarg['top'] = 1 - float(amarg['top']) / height
232     if 'bottom' in amarg:
233         rmarg['bottom'] = float(amarg['bottom']) / height
234     # return figure size and relative margins
235     return (width, height), rmarg
236
237 def __get_ax_size(ax, fig):
238     """
239     Source: http://scipy-central.org/item/65/1/absolute-plot-size-and-
240     -margins-in-matplotlib
241     """
242     bbox = ax.get_window_extent().transformed(fig.dpi_scale_trans.
243     inverted())
244     width, height = bbox.width, bbox.height
245     width *= fig.dpi
246     height *= fig.dpi
247     return width, height
248
249 def __conv_inch(length_mm):
250     """Converts a length from millimeters to inch
251
252     Args:
253         length_mm (int, float): length in millimeters to be converted
254
255     Returns:
256         float: The converted length in inch
257     """
258     return length_mm / 25.4
259
260 def generate_massspectra_plot_automatic_labels(input_filename,
261     output_filename):
262     print("Generate {0} mass spectra plot ({1}.{2}) from {3}.[xy/txt]"
263     ".format(filetype.upper(), output_filename,

```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
257     filetype , input_filename)))
258
259 # min and max value of the x axis
260 min_xaxis = 50
261 max_xaxis = 425
262
263 # dimensions of the single plot
264 figwidth_mm = 141
265 left_margin_mm = 14
266 right_margin_mm = 2
267 top_margin_mm = 26
268 bottom_margin_mm = 13
269
270 plotwidth_mm = figwidth_mm - left_margin_mm - right_margin_mm
271 plotheight_mm = plotwidth_mm * 0.8
272
273 hspace_mm = 0.1 # horizontal space between two plots
274
275 # read xy input data
276 raw_values = pd.read_csv(input_filename + ".xy", sep=" ", names=[ "m/z", "intensity"],
277                                     dtype={ "m/z": np.float64, "intensity": np.float64 })
278
279 # normalize input data
280 max_val = np.max(raw_values[ 'intensity' ])
281 raw_values[ "intensity_normalized" ] = raw_values[ 'intensity' ] / max_val * 100.0 # normalize intensity to percent
282
283 # read txt captions
284 caption_values = pd.read_csv(input_filename + ".txt", sep=" ;",
285                               names=[ "m/z", "caption" ], dtype={ "m/z": np.float64 })
286
287 # find maximum value for each annotate peak - necessary for later
288 # calculation of annotation lines
289 for i in caption_values.index:
290     raw_values_index = raw_values[ raw_values[ "m/z" ] == caption_values.ix[i][ "m/z" ] ].index[0]
291     value, index = float(raw_values.loc[raw_values_index - 5, "intensity_normalized"]), raw_values_index - 5
292
293     for z in range(-5, 15):
294         if float(raw_values.loc[raw_values_index + z, "intensity_normalized"]) > value:
295             value, index = float(raw_values.loc[raw_values_index + z, "intensity_normalized"]), raw_values_index + z
296
297         caption_values.loc[i, "m/z"] = float(raw_values.loc[index, "m/z"])
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
295     caption_values.loc[i, "intensity_normalized"] = value # add
296     intensity to caption table
297
298     # dimension for the annotations in pixel
299     delta_x_text_labels_mm = 3.8
300     delta_point_annotate_line_mm = 1
301     label_annotate_arm_high_mm = 5.3
302     delta_y_diagonal_annotate_arm_mm = 3.5
303     delta_y_baseline_axistop_mm = 6.3
304     delta_label_mznumber_pixel_mm = 7
305
306     fsize, margins = __figsize_and_margins(plotsize=__conv_inch(
307         plotwidth_mm), __conv_inch(plotheight_mm)),
308                     subplots=(1, 1),
309                     left=__conv_inch(
310                         left_margin_mm), right=__conv_inch(right_margin_mm),
311                         top=__conv_inch(
312                             top_margin_mm), bottom=__conv_inch(bottom_margin_mm),
313                         wspace=0.01, hspace=
314                         __conv_inch(hspace_mm))
315     fig = plt.figure(figsize=fsize, )
316     plt.subplots_adjust(**margins)
317     ax = fig.add_subplot(111)
318
319     # delta_dimension in pixel calculate from mm
320     delta_x_text_labels_pixel = __conv_inch(delta_x_text_labels_mm) *
321         fig.dpi
322     delta_point_annotate_line_pixel = __conv_inch(
323         delta_point_annotate_line_mm) * fig.dpi
324     label_annotate_arm_high_pixel = __conv_inch(
325         label_annotate_arm_high_mm) * fig.dpi
326     delta_y_diagonal_annotate_arm = __conv_inch(
327         delta_y_diagonal_annotate_arm_mm) * fig.dpi
328     delta_y_baseline_axistop = __conv_inch(
329         delta_y_baseline_axistop_mm) * fig.dpi
330     delta_label_mznumber_pixel = __conv_inch(
331         delta_label_mznumber_pixel_mm) * fig.dpi
332
333     # plot spectra line
334     ax.plot(raw_values["m/z"], raw_values["intensity_normalized"],
335             color="black", linewidth=0.8)
336
337     # set x axes range
338     if min_xaxis is not None and max_xaxis is not None:
339         ax.set_xlim([int(min_xaxis), int(max_xaxis)])
340
341     length_x_axis_pixel, length_y_axis_pixel = __get_ax_size(ax, fig)
342     x_axis_min, x_axis_max = ax.get_xlim()
343     y_axis_min, y_axis_max = ax.get ylim()
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```

332
333     list_text_pos = list()
334     if len(caption_values.index) > 0:
335
336         # annotate from basepeak
337         basepeak_index = caption_values[caption_values["intensity_normalized"] >= 100.0].index
338         if len(basepeak_index) == 0:
339             basepeak_index = [len(caption_values) / 2]
340
341         # annotate peaks left from basepeak
342         for z in caption_values.loc[:basepeak_index[0], "m/z"].sort_index(ascending=False):
343             x_value_pixel = (z - x_axis_min) / (x_axis_max - x_axis_min) * length_x_axis_pixel
344
345             if len(list_text_pos) == 0:
346                 list_text_pos.append(x_value_pixel)
347             else:
348                 if x_value_pixel > list_text_pos[-1] + delta_x_text_labels_pixel:
349                     list_text_pos.append(x_value_pixel)
350                 else:
351                     list_text_pos.append(list_text_pos[-1] + delta_x_text_labels_pixel)
352
353         # annotate peaks right from basepeak
354         if basepeak_index[0] + 1 in caption_values.index:
355             for z in caption_values.loc[basepeak_index[0] + 1:, "m/z"]:
356                 x_value_pixel = (z - x_axis_min) / (x_axis_max - x_axis_min) * length_x_axis_pixel
357
358                 if x_value_pixel < list_text_pos[0] - delta_x_text_labels_pixel:
359                     list_text_pos.insert(0, x_value_pixel)
360                 else:
361                     list_text_pos.insert(0, list_text_pos[0] - delta_x_text_labels_pixel)
362
363         # move annotation to left, if the label is out of the axes range
364         if len(list_text_pos) > 0 and max(list_text_pos) >= length_x_axis_pixel:
365             print("move labels to left")
366             calc_delta = max(list_text_pos) - length_x_axis_pixel
367             for index in range(len(list_text_pos) - 1, 0, -1):
368                 print("change", list_text_pos[index], calc_delta)

```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```

369         if index > 1 and list_text_pos[index] - calc_delta >
list_text_pos[index - 1] + delta_x_text_labels_pixel:
370             list_text_pos[index] = list_text_pos[index] -
calc_delta
371             break
372         else:
373             list_text_pos[index] = list_text_pos[index] -
calc_delta
374
375     for i in caption_values.index:
376
377         label_x_pos = list_text_pos.pop()
378         x_value = float(caption_values.loc[i, "m/z"])
379         y_value = float(caption_values.loc[i, "
intensity_normalized"])
380         label_text = str(caption_values.loc[i, "caption"])
381
382         x_value_pixel = (x_value - x_axis_min) / (x_axis_max -
x_axis_min) * length_x_axis_pixel
383         y_value_pixel = (y_value - y_axis_min) / (y_axis_max -
y_axis_min) * length_y_axis_pixel
384         label_y_pos_pixel = length_y_axis_pixel +
delta_y_baseline_axistop
385         length_y_annotation_line = label_y_pos_pixel -
y_value_pixel - delta_point_annotation_line_pixel
386         label_annotation_arm_low_pixel = length_y_annotation_line -
label_annotation_arm_high_pixel - delta_y_diagonal_annotation_arm
387
388         if label_annotation_arm_low_pixel < 0:
389             label_annotation_arm_low_pixel = 0
390
391         ax.annotate(" {:.0f} ".format(x_value),
392                     xy=(x_value_pixel, y_value_pixel +
delta_point_annotation_line_pixel),
393                     xycoords='axes pixels', rotation=90,
394                     xytext=(label_x_pos, label_y_pos_pixel),
395                     textcoords='axes pixels', size=8,
396                     arrowprops=dict(arrowstyle="",
397                                     connectionstyle="arc ,angleA
=-90,armA=" + str(label_annotation_arm_high_pixel) +
398                                     ",angleB= 90,
399                                     armB=" + str(label_annotation_arm_low_pixel) +
400                                     ",rad=0",
401                                     linewidth=0.5, color="#808080"),
402                     horizontalalignment='center',
403                     verticalalignment='bottom')
404         ax.annotate(label_text,
405                     xy=(x_value_pixel, y_value_pixel +
delta_point_annotation_line_pixel),

```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
403         xycoords='axes pixels', rotation=90,
404         xytext=(label_x_pos, label_y_pos_pixel +
405             delta_label_mznumber_pixel),
406         textcoords='axes pixels', size=8,
407         horizontalalignment='center',
408         verticalalignment='bottom'
409     )
410
411     # remove top and right axis
412     ax.spines['top'].set_visible(False)
413     ax.spines['right'].set_visible(False)
414     ax.get_xaxis().tick_bottom()
415     ax.get_yaxis().tick_left()
416
417     # label axes
418     ax.set_xlabel(r"$m/z$")
419     ax.set_ylabel(r"$Intensity \,[\%]$")
420
421     # set x labels
422     plt.xticks(rotation='vertical')
423     start, end = ax.get_xlim()
424     ax.xaxis.set_ticks(np.arange(start, end + 1, 25))
425
426     # set y labels
427     ax.set_ylimits(0, 100)
428     start, end = ax.get_ylimits()
429     ax.yaxis.set_ticks(np.arange(start, end + 1, 10))
430
431     # set grid
432     plt.grid(True, axis="y", color='black', linestyle=':', linewidth=0.1)
433     plt.show()
434
435     plt.savefig(output_filename + "." + filetype, dpi=fig.dpi, format=
436     filetype)
437     plt.close()
438
439 def generate_massspectra_two_plot_manual_annotation(input_filename1,
440     input_filename2, output_filename):
441     print("Generate {0} mass spectra plot ({1}.{2}) from {3}.xy and
442     {4}.xy".format(filetype.upper(), output_filename,
443                     filetype, input_filename1,
444                     input_filename2))
445
446     # min and max value of the x axis
447     min_xaxis = 50
448     max_xaxis = 475
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
444
445 # labels
446 label1 = "Substance 1"
447 label2 = "Substance 2"
448
449 # dimensions of the single plot
450 figwidth_mm = 141
451 left_margin_mm = 14
452 right_margin_mm = 2
453 top_margin_mm = 5
454 bottom_margin_mm = 13
455
456 plotwidth_mm = figwidth_mm - left_margin_mm - right_margin_mm
457 plotheight_mm = plotwidth_mm * 0.8
458
459 hspace_mm = 25 # horizontal space between two plots
460
461 # read xy input data
462 raw_values1 = pd.read_csv(input_filename1 + ".xy", sep=" ", names=
463 ["m/z", "intensity"],
464                               dtype={"m/z": np.float64, "intensity":
465 np.float64})
466 # normalize input data
467 max_val = np.max(raw_values1['intensity'])
468 raw_values1["intensity_normalized"] = raw_values1['intensity'] /
469 max_val * 100.0 # normalize intensity to percent
470
471 # read xy input data
472 raw_values2 = pd.read_csv(input_filename2 + ".xy", sep=" ", names=
473 ["m/z", "intensity"],
474                               dtype={"m/z": np.float64, "intensity":
475 np.float64})
476 # normalize input data
477 max_val = np.max(raw_values2['intensity'])
478 raw_values2["intensity_normalized"] = raw_values2['intensity'] /
479 max_val * 100.0 # normalize intensity to percent
480
481 fsize, margins = __figsize_and_margins(plotsize=(__conv_inch(
482 plotwidth_mm), __conv_inch(plotheight_mm)),
483                                         subplots=(1, 1),
484                                         left=__conv_inch(
485 left_margin_mm), right=__conv_inch(right_margin_mm),
486                                         top=__conv_inch(
487 top_margin_mm), bottom=__conv_inch(bottom_margin_mm),
488                                         wspace=0.01, hspace=
489                                         __conv_inch(hspace_mm))
490 fig = plt.figure(figsize=fsize, )
491 plt.subplots_adjust(**margins)
492
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
483 ax = fig.add_subplot(211)
484
485 # plot spectra line
486 plt1, = ax.plot(raw_values1["m/z"], raw_values1[
487 "intensity_normalized"], color="black", linewidth=0.8, label=label1
488 )
489 legend = plt.legend(handles=[plt1], loc=2)
490 legend.get_frame().set_linewidth(0)
491
492 # set x axes range
493 if min_xaxis is not None and max_xaxis is not None:
494     ax.set_xlim([int(min_xaxis), int(max_xaxis)])
495
496 # annotate plot1
497 annotate_point(253, 60, r'Peak 1', raw_values1)
498 annotate_distance(385, 403, 45, r'$-18$', raw_values1,
499 rotate_text=90)
500 annotate_distance(367, 385, 45, r'$-18$', raw_values1,
501 rotate_text=90)
502 annotate_distance(321, 349, 75, r'$xy$', raw_values1, rotate_text
503 =0)
504 annotate_point(403, 60, r'Peak 2', raw_values1)
505
506 # remove top and right axis
507 ax.spines['top'].set_visible(False)
508 ax.spines['right'].set_visible(False)
509 ax.get_xaxis().tick_bottom()
510 ax.get_yaxis().tick_left()
511
512 # label axes
513 ax.set_ylabel(r"$Intensity \,[\%]$")
514
515 # set x labels
516 plt.xticks(rotation='vertical')
517 start, end = ax.get_xlim()
518 ax.xaxis.set_ticks(np.arange(start, end + 1, 25))
519
520 # set y labels
521 ax.set_ylimits(0, 100)
522 start, end = ax.get_ylimits()
523 ax.yaxis.set_ticks(np.arange(start, end + 1, 10))
524
525 # set grid
526 plt.grid(True, axis="y", color='black', linestyle=':', linewidth
527 =0.1)
528
529 # set x labels
530 plt.xticks(rotation='vertical')
531 start, end = ax.get_xlim()
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
526     ax.xaxis.set_ticks(np.arange(start, end + 1, 25))
527     # set y labels
528     ax.set_ylimits(0, 100)
529     start, end = ax.get_ylimits()
530     ax.yaxis.set_ticks(np.arange(start, end + 1, 10))
531     # set grid
532     plt.grid(True, axis="y", color='black', linestyle=':', linewidth=0.1)
533
534     # generate plot2
535     ax = fig.add_subplot(212)
536
537     plt2, = ax.plot(raw_values2["m/z"], raw_values2["intensity_normalized"], color="#606060", linewidth=0.8,
538                      label=label2)
539     legend = plt.legend(handles=[plt2], loc=2)
540     legend.get_frame().set_linewidth(0)
541
542     # set x axes range
543     if min_xaxis is not None and max_xaxis is not None:
544         ax.set_xlim([int(min_xaxis), int(max_xaxis)])
545
546     # annotate
547     annotate_point(253, 60, r'Peak I', raw_values2)
548     annotate_point(189, 60, r'189', raw_values2)
549     annotate_point(215, 60, r'215', raw_values2)
550     annotate_distance(385, 445, 30, r'Multiple'"\n"r'line'"'\n'"'
551     annotation', raw_values2)
552     annotate_distance(367, 385, 45, r'-18', raw_values2, rotate_text=90)
553     annotate_distance(349, 367, 45, r'-18', raw_values2, rotate_text=90)
554     annotate_distance(321, 349, 75, r'xy', raw_values2, rotate_text=0)
555     annotate_point(445, 60, r'Peak 2', raw_values2)
556
557     # remove top and right axis
558     ax.spines['top'].set_visible(False)
559     ax.spines['right'].set_visible(False)
560     ax.get_xaxis().tick_bottom()
561     ax.get_yaxis().tick_left()
562
563     # label axes
564     ax.set_xlabel(r"$m/z$")
565     ax.set_ylabel(r"$Intensity \,[\%]$")
566
567     # set x labels
568     plt.xticks(rotation='vertical')
569     start, end = ax.get_xlim()
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
569     ax.xaxis.set_ticks(np.arange(start, end + 1, 25))
570
571     # set y labels
572     ax.set_ylim(0, 100)
573     start, end = ax.get_ylim()
574     ax.yaxis.set_ticks(np.arange(start, end + 1, 10))
575
576     # set grid
577     plt.grid(True, axis="y", color='black', linestyle=':', linewidth
578             =0.1)
579     plt.show()
580
581     plt.savefig(output_filename + "." + filetype, dpi=fig.dpi, format
582                 =filetype)
583     plt.close()
584
585
586 # Function used to draw most of my MS-spectra -> is calculating the
587 # distance and plotting:
588
589 def generate_massspectra_plot_distance_peak_manual_annotation(
590     input_filename, output_filename):
591     print("Generate {} mass spectra plot ({}.{}) from {}.[xy/txt]
592         ".format(filetype.upper(), output_filename,
593                  filetype, input_filename))
594
595     # min and max value of the x axis
596     min_xaxis = 300
597     max_xaxis = 700
598
599     plotSubstance = '619 [M+H]^{+}'
600
601     # dimensions of the single plot
602     figwidth_mm = 141
603     left_margin_mm = 14
604     right_margin_mm = 2
605     top_margin_mm = 5
606     bottom_margin_mm = 13
607
608     plotwidth_mm = figwidth_mm - left_margin_mm - right_margin_mm
609     plotheight_mm = plotwidth_mm * 0.8
610
611     hspace_mm = 0.1 # horizontal space between two plots
612
613     # read xy input data
614     raw_values = pd.read_csv(input_filename + ".xy", sep=" ", names=[
615         "m/z", "intensity"],
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
610                               dtype={"m/z": np.float32, "intensity":  
611                               np.float32})  
612 # normalize input data  
613 max_val = np.max(raw_values['intensity'])  
614 raw_values["intensity_normalized"] = (raw_values['intensity'] /  
615                                         max_val) * 100.0 # normalize intensity to percent  
616  
617 fsize, margins = __figsize_and_margins(plotsize=(__conv_inch(  
618                                         plotwidth_mm), __conv_inch(plotheight_mm)),  
619                                         subplots=(1, 1),  
620                                         left=__conv_inch(left_margin_mm),  
621                                         right=__conv_inch(right_margin_mm),  
622                                         top=__conv_inch(top_margin_mm),  
623                                         bottom=__conv_inch(bottom_margin_mm),  
624                                         wspace=0.01, hspace=  
625                                         __conv_inch(hspace_mm)))  
626 fig = plt.figure(figsize=fsize, )  
627 plt.subplots_adjust(**margins)  
628 ax = fig.add_subplot(111)  
629  
630 # plot spectra line  
631 line1 = ax.plot(raw_values["m/z"], raw_values["  
632 intensity_normalized"], color="black", linewidth=0.8)  
633 plt.legend([plotSubstance], loc=2)  
634  
635 # set x axes range  
636 if min_xaxis is not None and max_xaxis is not None:  
637     ax.set_xlim([int(min_xaxis), int(max_xaxis)])  
638  
639 # annotations  
640 # length in data value (in %)  
641 annotate_distance(619, 601, 30, '$18$', raw_values, rotate_text  
642 =0)  
643 annotate_distance(619, 575, 35, '$44$', raw_values, rotate_text  
644 =0)  
645 annotate_distance(619, 452, 40, '$167$', raw_values, rotate_text  
646 =0)  
647 annotate_distance(575, 452, 35, '$123$', raw_values, rotate_text  
648 =0)  
649 annotate_distance(452, 311, 40, '$141$', raw_values, rotate_text  
650 =0)  
651  
652 annotate_point(619, 83, r'619', raw_values)  
653 annotate_point(601, 83, r'601', raw_values)  
654 annotate_point(575, 110, r'575', raw_values)  
655 annotate_point(452, 53, r'452', raw_values)  
656 annotate_point(311, 83, r'311', raw_values)  
657  
658 #633
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
647     #annotate_distance(633, 601, 5, '$32$', raw_values, rotate_text
648     =o)
649     #annotate_distance(633, 583, 10, '$50$', raw_values, rotate_text
650     =o)
651     #annotate_distance(633, 557, 15, '$76$', raw_values, rotate_text
652     =o)
653     #annotate_distance(633, 510, 20, '$123$', raw_values, rotate_text
654     =o)
655     #annotate_distance(633, 496, 25, '$137$', raw_values, rotate_text
656     =o)
657     #annotate_distance(633, 478, 30, '$155$', raw_values, rotate_text
658     =o)
659     #annotate_distance(633, 460, 35, '$173$', raw_values, rotate_text
660     =o)
661     #annotate_distance(633, 367, 40, '$266$', raw_values, rotate_text
662     =o)
663     #annotate_distance(633, 337, 45, '$296$', raw_values, rotate_text
664     =o)
665     #annotate_distance(633, 313, 50, '$320$', raw_values, rotate_text
666     =o)
667     #annotate_distance(633, 287, 55, '$346$', raw_values, rotate_text
668     =o)
669     #annotate_distance(510, 496, 5, '$14$', raw_values, rotate_text
670     =o)
671     #annotate_point(633, 83, r'633', raw_values)
672     #annotate_point(601, 110, r'601', raw_values)
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
673 #annotate_point(583, 83, r'583', raw_values)
674 #annotate_point(557, 83, r'557', raw_values)
675 #annotate_point(510, 83, r'510', raw_values)
676 #annotate_point(496, 83, r'496', raw_values)
677 #annotate_point(478, 83, r'478', raw_values)
678 #annotate_point(460, 83, r'460', raw_values)
679 #annotate_point(442, 83, r'442', raw_values)
680 #annotate_point(432, 83, r'432', raw_values)
681 #annotate_point(400, 83, r'400', raw_values)
682 #annotate_point(369, 93, r'369', raw_values)
683 #annotate_point(367, 83, r'367', raw_values)
684 #annotate_point(337, 83, r'337', raw_values)
685 #annotate_point(325, 73, r'325', raw_values)
686 #annotate_point(319, 83, r'319', raw_values)
687 #annotate_point(313, 73, r'313', raw_values)
688 #annotate_point(309, 83, r'309', raw_values)
689 #annotate_point(297, 73, r'297', raw_values)
690 #annotate_point(287, 83, r'287', raw_values)

691
692 # remove top and right axis
693 ax.spines['top'].set_visible(False)
694 ax.spines['right'].set_visible(False)
695 ax.get_xaxis().tick_bottom()
696 ax.get_yaxis().tick_left()

697
698 # label axes
699 ax.set_xlabel("m/z")
700 ax.set_ylabel("Relative abundance") #$Relative abundance\,[\%]$

701
702 # set x labels
703 plt.xticks(rotation='vertical')
704 start, end = ax.get_xlim()
705 ax.xaxis.set_ticks(np.arange(start, end + 1, 25))

706
707 # set y labels
708 ax.set_ylim(0, 100)
709 start, end = ax.get_ylim()
710 ax.yaxis.set_ticks(np.arange(start, end + 1, 10))

711
712 # set grid
713 plt.grid(True, axis="y", color='black', linestyle=':', linewidth=0.1)
714 plt.savefig(output_filename + ".png")

715
716 plt.show()
717 plt.close()

718
719 # Specify dataset and function to generate the MS-spectra:
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
721 if __name__ == "__main__":
722     # generate_massspectra_plot_automatic_labels("examples/inputs/
723     # substance1", "examples/ex1")
724
725     generate_massspectra_plot_distance_peak_manual_annotation("data/
726     raw/Kuerbis_Analyse_Bindungskinetik_Fragmente61927", "data/ex2")
727
728     # generate_massspectra_two_plot_manual_annotation("examples/
729     # inputs/substance1", "examples/inputs/substance2", "examples/ex3")
730
731 content/Anhang/Code/VWA_generate_mz_diagrams.py
```

A.3. Programm zur Erstellung von Fragmentierungsdiagrammen

```
1 # coding: utf-8
2
3 # # Savitzky-Golay Filter for smoothing Fragmentation-diagrams
4
5 # Function in which savitzky-golay filter is applied:
6
7 def savgolay_filter_catabolites(ms_data, ms_name, mode_value, mode):
8     np.set_printoptions(precision=2)
9     ms_savgol = savgol_filter(ms_data, 23, 6, mode=mode_value) # 
10    apply sav-gol filter
11
12    area_initial = trapz(ms_data, dx=5)
13    area_savgol = trapz(ms_savgol, dx=5)
14
15    # x_values = np.array
16    # ([0,5,10,15,20,25,30,35,40,45,50,55,60,65,70,75,80,85,90,95,100])
17    x_values = np.array
18    # ([0,5,10,15,25,30,35,40,45,50,55,60,65,70,75,80,85,90,95,100])
19
20    ax = plt.axes()
21    ax.set_xlim([0, 100])
22    ax.set_ylim([0, 105])
23    ax.set_xticks(x_values)
24
25    if(mode == 'initial'):
26        line = plt.plot(x_values, ms_data, label = i)
27    elif(mode == 'initial-savgol'):
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
26     line1 = plt.plot(x_values, ms_data, label = i + ' initial ,  
27     area: ' + str(round(area_initial)))  
28     line2 = plt.plot(x_values, ms_savgol, label = i + ' savgol ,  
29     area: ' + str(round(area_savgol)))  
30     elif(mode == 'savgol'):  
31         line = plt.plot(x_values, ms_savgol, label = i)  
32  
33         # Following code could be used for SVM/SVR but is not "  
34         # smoothing" function  
35         # svr_rbf = SVR(kernel='rbf', C=1e3, gamma=0.1)  
36         # y_rbf = svr_rbf.fit(x_values.reshape(-1, 1), np.array(  
37         # ms_data)).predict(x_values.reshape(-1, 1))  
38         # plt.plot(x_values, y_rbf, color='navy', label='RBF model')  
39  
40     plt.xlabel('Normalised Collision Energy (%)')  
41     plt.ylabel('% of highest peak')  
42     plt.legend()  
43     ax.yaxis.grid()  
44  
45     return ms_savgol  
46  
47  
48 # Import needed modules:  
49  
50 import numpy as np  
51 from numpy import trapz  
52 import pandas as pd  
53 import matplotlib.pyplot as plt  
54 from scipy.signal import savgol_filter  
55  
56 from sklearn.svm import SVR # import support-vector machine  
57  
58 data = pd.io.parsers.read_csv("csv/  
59     Kuerbis-Analyse_Bindungskinetik_Fragmente873_Version1.csv")  
60  
61  
62 # Specify dataset on which to apply the filter:  
63  
64 catabolite = input("Specify fragmentation path: ")  
65 fragmentation_mode = input("Specify fragmentation mode: ")  
66  
67 catabolites_string = input("Specify [M]-fragments of above catabolite  
68     : ")  
69 catabolites = catabolites_string.split(",")  
70 print(catabolites)  
71  
72  
73 # Call above defined function and print output:  
74
```

Anhang A. Python Code-Listings für die Erstellung der Grafiken

```
69 specify_diagrams = ['initial', 'initial-savgol', 'savgol']
70
71 for x in specify_diagrams:
72     for i in catabolites:
73         value = data[[i]].values
74         diag = np.array(value[3:23].astype(np.int))
75         value_new = savgolay_filter_catabolites(np.hstack(diag), i,
76                                              'nearest', x)
77
78     plt.title(catabolite+fragmentation_mode+' '+x)
79
80     diag_name = 'Diagrams_output/' + catabolite+fragmentation_mode+'-' +
81     catabolites[o]+x
82     plt.savefig(diag_name+'.png')
83     plt.show()
```

content/Anhang/Code/VWA_Savitzky–Golay–Filter.py

Eidesstattliche Erklärung

Ich, Florian Klubenschedl, erkläre hiermit eidesstattlich, dass ich diese vorwissenschaftliche Arbeit selbstständig und ohne Hilfe Dritter verfasst habe. Insbesondere versichere ich, dass ich alle wörtlichen und sinngemäßen Übernahmen aus anderen Werken als Zitate kenntlich gemacht und alle verwendeten Quellen angegeben habe.

Telfs, am

Datum

Unterschrift