Automatische Auswertung von Molekularer Dynamik mittels Maschinellen Lernens

Dissertation

zur Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Christian-Albrechts-Universität zu Kiel
vorgelegt von

David Siebler

Kiel, 2018

Referent:	Prof. Dr. Bernd Hartke
Koreferentin:	Prof. Dr. Carolin König
Tag der mündlichen Prüfung:	10.10.2018

Abstract

The broad availability of huge amounts of processing power leads to bigger simulated data-sets, which are calculated much quicker. One frequent example in computational chemistry is molecular dynamics, generating many huge trajectory files. Common ways to analyze these simulation trajectories are simplistic algorithms (re-) designed for each particular case and human pattern recognition, overused by exhaustive video clip watching. A more sophisticated and more reliable way of pattern recognition and analysis of simulations is presented in this thesis: A fully automatic analysis of molecular dynamics simulation data with *machine learning* techniques.

To easily allow for generalizations and for coarse-graining, and to avoid known problems with similarity measures in molecule-internal coordinates, a different geometric approach that is voxel-based and independent of the number of atoms is shown. These voxels are generated by transforming a molecular structure into an octTree representation. Superstructures are constructed from these voxels and form the databases for *machine learning* algorithms.

The unsupervised *machine learning* technique (*Hierarchical*) *Clustering* is used for automatic ranking and selecting of a suitable training-set from a large set of trajectories. A trained *Radom-Forest-Algorithm* classifies every single snapshot in every trajectory. Finally, a *Markov-State-Model* is used for classification of each trajectory in the large set of trajectories, and to group similar ones. This eliminates most of the biased assumptions and most of the tedious work from the trajectory analysis, and allows human effort to focus on the pre-selected interesting cases and to further integrate the extracted information into a “mechanistic” picture.

This framework is developed on the example of a *trans-/cis*-photoisomerization of bridged-azobenzene derivatives. Strengths and weaknesses of various algorithm design choices are discussed and illustrated with exemplary data. Finally, the success of generalization and transferability of the resulting automatic framework is shown on the example of the photoisomerization.

Kurzzusammenfassung

Die breite Verfügbarkeit einer großen Menge an Datenverarbeitungsleistung führt zu größeren Datensätzen, welche immer schneller berechnet werden können. Ein häufiges Beispiel in der Computerchemie ist die Molekulare Dynamik, welche viele große Trajektoriendateien generiert. Die üblichen Wege der Analyse dieser Trajektorien sind simple Algorithmen, welche für jeden einzelnen Anwendungsfall (um-) konzipiert werden, und die menschliche Fähigkeit der Mustererkennung, welche durch intensives Video schauen beansprucht wird. Eine elegantere und zuverlässigere Art der Mustererkennung und der Analyse wird in dieser Arbeit vorgestellt: Eine vollautomatische Analyse von Molekularer Dynamik mit *maschinellen Lerntechniken*.

Um die Generalisierung und Vergröberung zu vereinfachen und die bekannten Probleme bei der Ähnlichkeitsmessung in internen Molekülkoordinaten zu umgehen, wird ein anderer, voxel-basierter Ansatz genutzt, welcher unabhängig von der Anzahl der Atome ist. Die Voxel werden durch die Umwandlung einer molekularen Struktur in eine octTree Repräsentation generiert. Aus diesen Voxeln werden Superstrukturen konstruiert, welche die Datenbasis für die Algorithmen des *maschinellen Lernens* bilden.

Die unüberwachte *maschinelle Lerntechnik*, (*Hierarchisches*) *Clustern*, wird für eine automatische Bewertung und eine Auswahl eines passenden Trainingssatzes aus einem großen Satz von Trajektorien verwendet. Ein trainierter *Random-Forest-Algorithmus* klassifiziert jeden einzelnen Schnappschuss in jeder Trajektorie. Schließlich wird ein *Markov-Zustands-Modell* für die Klassifikation jeder einzelnen Trajektorie in dem großen Satz von Trajektorien verwendet und ähnliche werden gruppiert. Dies eliminiert weitgehend die Voreingenommenheit und die uninteressante Arbeit bei der Trajektorienauswertung, erlaubt die Fokussierung der menschlichen Anstrengungen auf vorausgewählte interessante Fälle und die weitere Einbindung der extrahierten Information in das ›mechanistische‹ Gesamtbild.

Das vorgestellte Programmkonzept wird am Beispiel der *trans-/cis*-Photoisomerisierung von verbrückten Azobenzolderivaten entwickelt. Die Stärken und Schwächen der verschiedenen zur Wahl stehenden Algorithmen des Designs werden anhand von exemplarischen Daten diskutiert und illustriert. Abschließend wird der Erfolg der Generalisierung und Übertragbarkeit des resultierenden automatischen Programms am Beispiel der Photoisomerisierung gezeigt.

Inhalt

Kapitel 1

Einleitung

Die theoretische Chemie hat sich zur Aufgabe gemacht, die Eigenschaften von molekularen Systemen zu berechnen. Neben statischen Berechnungen von molekularen Eigenschaften nehmen Dynamiksimulationen einen großen und wichtigen Bereich ein. Diese Simulationen sind keineswegs trivial und es gibt – auch wenn manche Theorien weit mehr versprechen – nicht eine einzige Simulationstechnik, die für alle Arten von Systemen funktioniert und insbesondere auch mit den verfügbaren Computerleistungen umsetzbar ist.

Die methodischen Verbesserungen auf der einen Seite wie beispielsweise QM/MM, bei dem die maßgeblich Beteiligten mit dem Nobelpreis ausgezeichnet wurden, und auf der anderen Seite durch die kontinuierliche Leistungssteigerung im Bereich der Datenverarbeitungsgeschwindigkeit können immer aufwändigere Simulationen durchgeführt werden.[1–3] Dabei können mehr Partikel simuliert, und bessere Methoden angewendet werden. Der Umstand der heutigen Zeit, dass Smartphones mehr Rechenleistung besitzen als Supercomputer vor einigen Jahrzehnten[1], macht deutlich, dass für die Analyse von Daten ebenfalls aufwendigere Algorithmen verwendet werden können als die traditionellen statischen Implementationen.

Dieser massive Zuwachs der Rechenleistung begünstigt gleichzeitig das Feld des *Maschinellen Lernens*, welches das speziellere *Deep Learning* mit einschließt. Zusammen mit den immer größeren Datenmassen in der Forschung[2] sowie in der medialen Berichterstattung, *Big Data*, stellt dies ein omnipräsentes Thema dar.[4–8] Die Automobile sollen autonom werden, und die sogenannte *künstliche Intelligenz* erreicht in immer komplexeren Fragestellungen Erfolge in direktem Vergleich mit

[1] Beispielsweise verfügt das *Samsung Galaxy S5* im Jahr 2014 über 142 gigaflops und verfügt damit über eine vergleichbare Leistung wie *Intel's Paragon XP/S 140* mit 143.4 gigaflops aus dem Jahr 1994.

[2] Das Datenzentrum des Forschungszentrums CERN produziert ein PetaByte (10^{15}Bytes) Daten pro Tag.

dem Menschen.[9–11] Die Weltmeister der einzelnen Spiele Backgammon (1979), Schach (1997) und Go (2016) wurden nach und nach im Wettbewerb von Programmen besiegt.

Gleichzeitig wird durch freie Software der Zugriff auf die damit verbundenen Techniken zunehmend leichter. Die hiermit einhergehenden Vorteile sind, dass komplexe Probleme mit relativ einfachen Algorithmen analysiert werden können. Der nächste logische Schritt ist somit, dass Techniken der künstlichen Intelligenz und damit insbesondere des maschinellen Lernens die Analyse und Bewertung im gesamten wissenschaftlichen Umfeld weiter ergänzen und die Möglichkeiten der Forschungsgemeinschaft vergrößern.[12–18]

Bisher werden für die Auswertung von molekularer Dynamik zahlreiche Programme für die automatische Analyse angeboten, welche sich oft auf die intuitiven, meist speziellen, Parameter beschränken.[19–26] Dabei kann in Programmbibliotheken und Programme mit Benutzeroberfläche unterschieden werden. Bei letzteren wird bewusst die Hürde für den Anwender herabgesetzt.

Aktuell ist zu beobachten, dass die Techniken des maschinellen Lernens auch Einzug in den Bereich der mechanistischen Aufklärung erhalten, wie beispielsweise bei Tavadze et al.[27] speziell im Bereich der Photochemie.

Ebenfalls profitieren die Simulationsmethoden stark von den Techniken des maschinellen Lernens. Dabei werden Trainingsdaten, welche durch hochwertige quantenchemische Methoden erzeugt werden, effizient durch maschinelle Lernmethoden reproduziert und vorhergesagt.[28–31] Durch die fundierte statistische Grundlage, die die Algorithmen bieten, wird die Objektivierung der Analyse von Resultaten gefördert und stellt neben den Möglichkeiten, die durch die Verfügbarkeit von großen Datenmengen entsteht, ein wichtiges Standbein für die Durchführung zukünftiger Forschungsarbeiten dar.

In dieser Arbeit wird ein wichtiger Schritt in Richtung der vollautomatischen Auswertung von Simulationsdaten skizziert. Hierbei wird der Fokus auf unüberwachte maschinelle Lerntechniken gesetzt, welche keine Informationen über das mögliche Verhalten während der Simulation enthalten. Dazu wird das Hauptaugenmerk auf die Bewertung von geometrischen Strukturdaten gelegt und eine gitterbasierte diskrete Repräsentation von Molekülen verwendet.

Kapitel 2

Theorie

Zunächst werden die Grundlagen der Molekularen Dynamik, Abschnitt 2.1, erläutert, anschließend sowohl – nach einer kleinen Einführung in das maschinelle Lernen selbst – die mathematischen als auch die praktischen Grundlagen des maschinellen Lernens, Abschnitt 2.7, vorgestellt. Schließlich wird die verwendete Segmentierung des Raums, Abschnitt 2.8, mit welcher im weiteren Verlauf als Datengrundlage gearbeitet wird, erläutert.

2.1 Molekulare Dynamik

Die Molekulare Dynamik (MD) ist eine weit verbreitete und bekannte Simulationstechnik, bei der die Bewegungen der Partikel oder Atome als klassisch-mechanische Teilchen betrachtet werden, dazu wird die Newtonsche Differentialgleichung für die Bewegungen gelöst. Für die MD-Simulation werden Kräfte zwischen den betrachteten Teilchen benötigt, welche entweder direkt aus quantenmechanischen Berechnungen, also *ab-initio* Methoden, stammen können, oder – zur weiteren Effizienzsteigerung und Durchführbarkeit – aus Modelpotentialen, also Kraftfeldern. Die Kombination der Berechnungsmethoden mit QM/MM und dem kontinuierlichen Anstieg der Rechenleistung kann qualitativ hochwertigere Rechnungen in immer kürzerer Rechenzeit erzeugen. [1–3,17]

Unabhängig von der Methode werden in definierten Zeitschritten der Trajektorie Schnappschüsse der Geometrien und weitere Eigenschaften gespeichert, welche anschließend zur Analyse der MD dienen. Die Geometrien werden pro Zeitschritt entweder in internen Koordinaten, also den Abständen, Winkeln und Diederwinkeln, oder kartesische Koordinaten, folglich den absoluten Atompositionen, in der sogenannten Trajektoriendatei abgespeichert. Diese Trajektoriendatei, im folgenden nur noch als *Trajektorie* bezeichnet, ist die alleinige Grundlage für die

Analyse in dieser Arbeit. Da in einer einzigen, zeitlich begrenzten MD lediglich ein kleiner Ausschnitt der möglichen Reaktionsverläufe abgebildet werden kann, wird eine größere Anzahl an Simulationen durchgeführt, bei denen die Startbedingungen leicht variiert werden. Diese werden in dieser Arbeit aus einer energiearmen Trajektorie entnommen, welche die Brownsche Bewegung bei 298.15 K simuliert. Über die gesamte Schar der Trajektorien kann schließlich eine Statistik berechnet werden und daraus resultierend ein Ergebnis für den Mechanismus oder spezifische Anwendungsgebiete abgeleitet werden.

An der Universität Kiel gibt es seit Juli 2007 den *Sonderforschungsbereich 677: Funktion durch Schalten*, in dessen Rahmen verschiedene auf Azobenzolen basierte Photoschalter untersucht werden, unter anderem verbrückte Azobenzole.[32] Aus diesem Grund werden die in Abschnitte 3.2 und 3.3 gezeigten Azobenzolderivate ausgewählt und analysiert. Die in der Literatur bisher verwendeten Klassifikationsmethoden für Trajektorien solcher Moleküle werden in Abschnitt 3.1 vorgestellt und kritisch diskutiert. Für Details über die verwendeten quantenchemischen Methoden und Ergebnisse werden die Forschungsarbeiten[33–37] empfohlen.

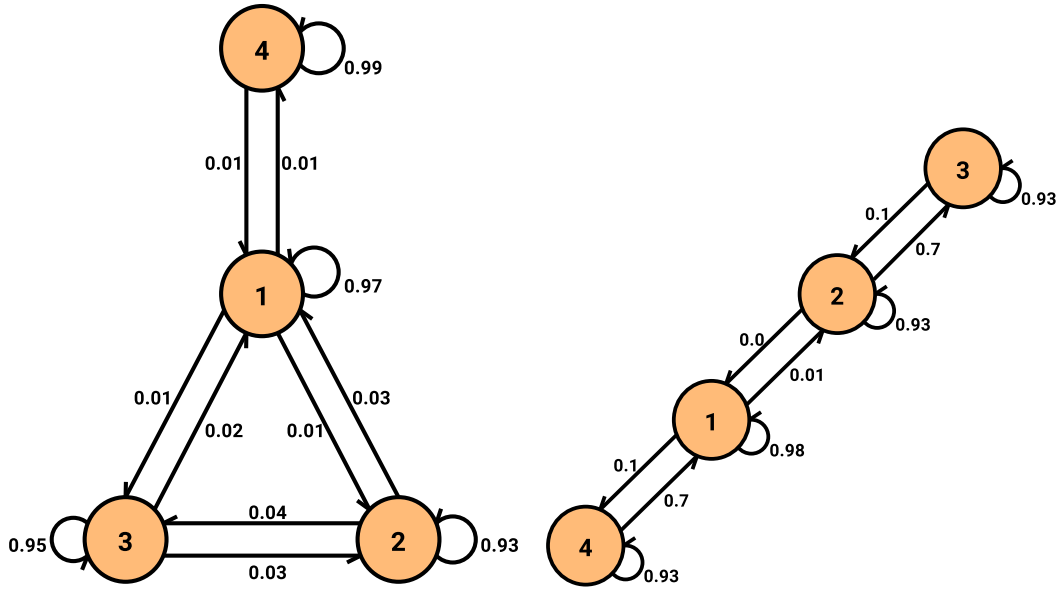
Für die Experimentatoren ist neben der mechanistischen Aufklärung einer photochemischen Reaktion ebenfalls die Effizienz eines Systems aussagekräftig, also wie viele Moleküle anteilig ein reaktives Verhalten zeigen. In photochemischen Reaktionen werden Prozesse mit Lichtquanten gestartet und damit ist die sogenannte *Quantenausbeute* der reaktive Anteil in einer Probe. Um dieser makroskopischen Größe näher zu kommen, werden diese Quantenausbeuten ermittelt, dazu wird aus Simulationen der prozentuale Anteil der Trajektorien berechnet, die einen bestimmten Zielwert erreichen.

In den vorliegenden theoretischen Berechnungen von photochemischer MD werden keine expliziten Quanten zur elektronischen Anregung simuliert, sondern das System zu Beginn der Propagation in einen elektronisch angeregten Zustand gesetzt. Die Quantenausbeute ist idealerweise 1, was bedeuten würde, dass alle berechneten Trajektorien das bestimmte Ziel erreicht haben, oder im Experiment, dass in einer Probe alle Moleküle reagieren. Um einen postulierten Reaktionsmechanismus zu untersuchen, werden die vorher als wichtig bestimmten Freiheitsgrade auf deren zeitlichen, über die Trajektorienschar gemittelten Verlauf hin untersucht. In Rahmen dieser Arbeit werden Trajektorien, welche das Ziel erreichen, als *reaktiv* und jene, welche dieses Verhalten nicht zeigen, als *unreaktiv* bezeichnet.

Die zwei im Folgenden untersuchten Moleküle sind Derivate des verbrückten Azobenzols und somit Bestandteil aktueller Forschung.

2.2 Markov-Zustandsmodell

Ein Markov-State-Model (MSM) [38–40] ist ein statisches Zustandsmodell, welches berücksichtigt, dass zuvor stattfindende Ereignisse die Wahrscheinlichkeit im aktuellen Experiment beeinflussen. Die Wahrscheinlichkeit, dass ein Ereignis eintritt, ist folglich abhängig davon, welche Ereignisse zuvor eingetreten sind. Die bedingten Wahrscheinlichkeiten, bei denen für die Wahrscheinlichkeit eines künftigen Ereignisses nur der aktuelle Zustand betrachtet wird, wird Markov-Kette der ersten Ordnung genannt, welche für diese Arbeit genutzt wird.



(a) Beispiel eines MSM bei dem der Zustand **4** nur vom Zustand **1** erreicht werden kann und der Zustand **1** eine zentrale Rolle spielt.

(b) Beispiel eines linearen MSM bei dem kein Zustand mit allen verbunden ist, also ein Übergang aus Zustand **4** zu **2** oder **3** unmöglich ist.

Abb. 2.1: Die schematischen Darstellung eines MSM; die Verbindungspfeile sind die Übergänge, die schleifenartigen Pfeile repräsentieren den Selbsterhalt, die expliziten Zahlenwerte stellen die Warscheinlichkeiten dar. Die Wahrscheinlichkeiten für den Selbsterhalt sind in diesen exemplarischen Modellen deutlich größer als die Wahrscheinlichkeiten für den Übergang in einen benachbarten Zustand.

Existiert eine feste Anzahl an Zuständen, denen definierte Wahrscheinlichkeiten für Übergänge in andere Zustände und Selbsterhalt zugeordnet sind, können direkt einfache Darstellungen wie in Abbildungen 2.1a und 2.1b erzeugt werden und Verknüpfungsmuster abgelesen werden. Im Kontext des chemischen Bildes für Moleküle einer MD ausgedrückt: Molekülgeometrien in einer MD können in einem begrenzten Zeitintervall nur eine begrenzte Anzahl weiterer Molekülgeometrien erreichen. Da es in den meisten Fällen mehrere erreichbare Geometrien gibt, kann statistisch eine Wahrscheinlichkeit angegeben werden, mit der nach ei-

ner bestimmten Zeit eine dieser Geometrien erreicht wird. Aus diesem MSM lassen sich zufällig Trajektorien errechnen, welche im Rahmen der Wahrscheinlichkeiten möglich sind. Teil dieser Arbeit ist der inverse Fall, also die Berechnung eines MSM aus vorliegenden Zustands-Trajektorien. Mittels der einfachen Zählung der Abfolge von Zuständen lassen sich deren Übergänge und der Selbsterhalt des jeweiligen Zustands bestimmen. Verschiedene MSM sind prinzipiell in der Lage, im Rahmen der Wahrscheinlichkeiten dieselbe Trajektorie zu erzeugen, daher lässt sich nicht aus einer begrenzten Anzahl an Zeitschritten in einer Trajektorie ein einziges MSM berechnen.

Für weitere Zeitreihenrepräsentationen in symbolischer Form, ähnlich zu einer aus einem MSM berechneten Trajektorie, bieten Lin et al. [41] neben einem eigenen Verfahren eine gute Übersicht.

2.3 Perron Cluster Cluster Analysis

Die Perron Cluster Cluster Analysis (PCCA)[1], welche in dem Python-Paket *PyEMMA* in einer weiterentwickelten und robusteren Version mit dem Namen *PCCA+* implementiert ist, ermöglicht es, metastabile Zustände in einem MSM zu determinieren. [42–45] Dazu wird die Übergangswahrscheinlichkeitsmatrix durch Permutationen in eine block-diagonale Form gebracht, um auf diesem Wege häufige und schnelle Übergänge zu ermitteln. Dieses Verfahren ermöglicht es, aus einer größeren Anzahl an Zuständen ein vereinfachtes MSM zu erzeugen, welches metastabile Zustände erzeugt.

Das Beispiel in Abbildung 2.2 zeigt ein MSM mit 20 Zuständen. Die Wahrscheinlichkeiten für den Selbsterhalt werden in dieser Darstellung durch die Größe des Kreises des jeweiligen Zustands dargestellt. Niedrige Übergangswahrscheinlichkeiten erhalten einen dünnen Pfeil, die größeren einen entsprechend dickeren Pfeil. Um den Zustand **0** herum gruppieren sich einige Zustände, welche Übergänge zu diesem Zustand und untereinander aufweisen. Zu den Zuständen **13** bis **19** gibt es aus dem Zustand **0** keine Übergänge.

Nach der Anwendung der PCCA für vier Zustände ergibt sich die Abbildung 2.3, in der zu erkennen ist, dass die Zustände entsprechend zusammengefasst werden. Die Zustände **0** bis **11** wurden beispielsweise zu dem neuen Zustand **A** zusammengefasst. Die jeweiligen kurzlebigen Zustände mit geringem Selbsterhalt und den vielen vereinzelten, schnellen Übergängen werden von dem meta-stabilen Zustand **A** absorbiert. Dies führt zu einer Vereinfachung des Modells und erleichtert

[1] Die Wiederholung des Wortes *Cluster* soll eine Verwechslung mit der *Principal Component Analysis (PCA)*, Hauptkomponentenanalyse, verhindern.

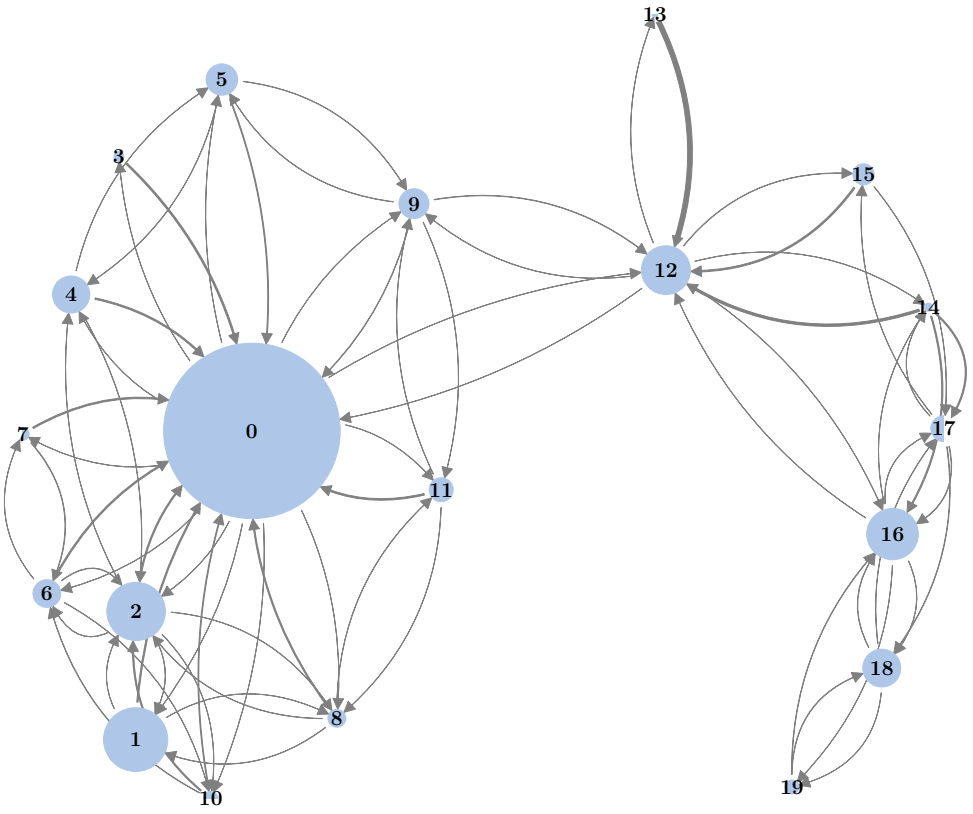


Abb. 2.2: MSM mit 20 Zuständen: Der Radius des Zustands steht für die Wahrscheinlichkeit, in diesem Zustand zu verbleiben; Pfeile mit größerer Linienstärke stehen für größere Wahrscheinlichkeit des Übergangs.

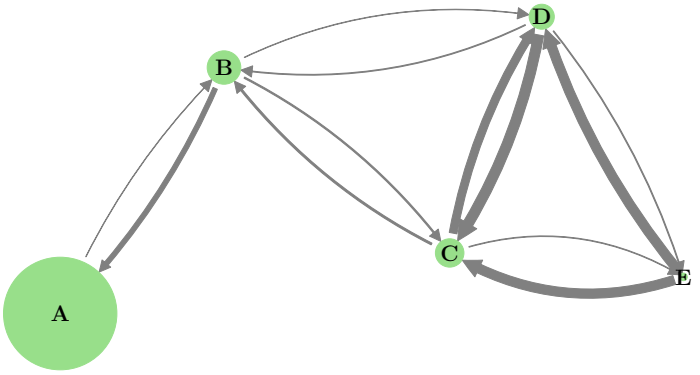


Abb. 2.3: Die 20 Zustände aus Abbildung 2.2 werden mittels PCCA zu 5 Zuständen vereinfacht.

die Analyse. In diesem Fall kann festgestellt werden, dass der Zustand **B** bzw. der urspüngliche Zustand **12** eine zentrale Rolle einnimmt und der entstandene Zustand **A** aus Mikrozuständen besteht, welche für die Reaktion in Richtung **B** eine untergeordnete Rolle einnehmen.

Die PCCA ermöglicht eine Zusammenfassung von Mikrozuständen basierend auf der zeitlichen Abfolge und finden in dieser Arbeit in der automatischen Analyse in Kapitel 4 Anwendung.

2.4 Metrik

Eine Metrik ist die Distanz zweier Punkte oder beliebiger Instanzen in einem *nD*-Raum und ist ein positiver reeller Wert, der vereinfacht als der Abstand beider Instanzen bezeichnet werden kann. Der Abstand zu einem nicht identischen Punkt muss stets ungleich Null sein, die Vertauschung der Instanzen darf den Abstand nicht verändern und es darf keinen kürzeren Weg geben als den durch die Metrik berechneten (Dreiecksungleichung [46]). Im Folgenden werden die in dieser Arbeit verwendeten Distanzen vorgestellt.

Ein anschauliches Distanzmaß ist das euklidische, da der Mensch sich in eben jenem euklidischen Raum bewegt. Die Gleichung (2.1) zeigt, dass die einzelnen Koordinaten der Punkte a und b elementweise subtrahiert, quadriert und in einer Quadratwurzel summiert werden. Diese Metrik findet in dieser Arbeit Verwendung für die Berechnung des Root-mean-square deviation of atomic positions (RMSD), Abschnitt 2.5, der multidimensionalen Skalierung, Abschnitt 2.6, und als Distanzmaß für das Clustern in Abschnitt 2.7.1.

Neben dieser gibt es weitere Metriken, welche verwendet werden können. Teilweise sind diese für definierte Anwendungsbereiche vorgesehen, so gibt es z.B. die französische Eisenbahnmetrik [47]. Diese basiert darauf, dass fast sämtliche Schienenverbindungen im 19ten Jahrhundert über Paris führten. Somit ergibt sich in dieser Metrik häufig ein Umweg über Paris, der für diesen Reiseweg eingeplant werden muss. Diese Metrik zu verwenden, ergibt verständlicherweise für jeden, der nicht auf der Schiene reist, keinen Sinn. Folglich ist die Auswahl der Metrik stark an den vorliegenden Anwendungsfall gebunden und kann beispielsweise mittels der Lage der Instanzen im Raum begründet werden.

Eine weitere Metrik, mit einem größeren Anwendungsgebiet, ist unter anderem als *Manhattan-Distanz* bekannt. Diese Bezeichnung bezieht sich auf die orthogonale Ausrichtung der *avenues* und *streets* im Stadtteil Manhattan in New York und wird auch als *Taxi-* oder *Cityblock*-Distanz bezeichnet. Ein Weg ist nur entlang der Straßen möglich und nicht quer durch die Wohnblöcke, daher ist der

Weg, der durch die euklidische Distanz beschrieben wird, nicht möglich. Es muss in der Regel ein längerer Weg zurückgelegt werden, der sich nur entlang der *avenues* und *streets* bewegt; mathematisch ergibt sich somit für einen *nD*-Raum Gleichung (2.2). Diese Metrik kommt beim Clustern zur Anwendung und kann ebenfalls für einen Multidimensionale-Skalierung-Graphen verwendet werden.

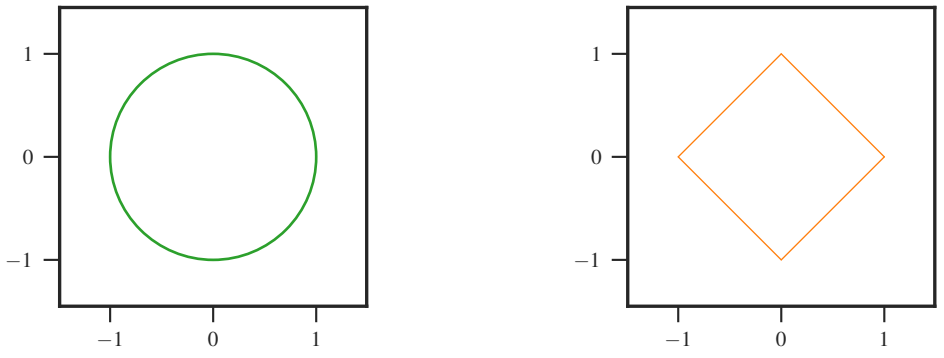
In Abbildung 2.4a ist ein Einheitskreis um den Ursprung in euklidischer und in Manhattan Distanz in der zweidimensionalen Ebene dargestellt. Jeder Punkt auf dem grünen Kreis besitzt in der euklidischen Metrik ausgedrückt einen Abstand von 1 vom Urspurng. Jeder Punkt auf dem orangenen Quadrat wiederum ist in der Manhattan-Metrik exakt 1 vom Ursprung entfernt. Erwartungsgemäß ist bei dem Vergleich dieser beiden Metriken die Differenz der berechneten Distanz entlang der Winkelhalbbierenden am größten.

$$d(\mathbf{a},\mathbf{b}) = \sqrt{\sum_{i}^{n}(\mathbf{a}_i - \mathbf{b}_i)^2} \tag{2.1}$$

euklidische Distanz

$$d(\mathbf{a},\mathbf{b}) = \sum_{i}^{n}|\mathbf{a}_i - \mathbf{b}_i| \tag{2.2}$$

Manhattan-Distanz



(a) **(b)**

Abb. 2.4: Berechnung der Equidistanz von 1 um den Ursprung: **(a)** in euklidischer Distanz und **(b)** in Manhattan-Distanz.

In einer Distanzfunktion können viele Attribute der Instanzen bereits gewichtet oder verändert werden, allerdings ist es zielführender, eine etablierte Distanzfunktion zu verwenden und die Attribute so anzupassen, dass diese eine sinnvolle Repräsentation der Abstände ermöglichen.

Die euklidische Metrik findet bei der inter- und intramolekularen Abstandsberechnung Verwendung. Mittels letzterer können Moleküle vom kartesischen Ko-

ordinatenraum zusammen mit den berechneten Winkeln und den Diederwinkeln in eine vollständige, rotations- und translationsfreie Beschreibung der Molekülgeometrie, die sogenannte Z-Matrix, überführt werden. Für diese, auch interne Koordinaten genannte, Repräsentation von Molekülen gibt es keine Metrik, welche an die Eigenschaften der Koordinaten angepasst ist und es damit ermöglicht, zwei Molekülgeometrien miteinander zu vergleichen. Der Wertebereich für einen Diederwinkel ist $[-180°;180°]$, für die Winkel gilt: $[0°;180°]$ und für die Abstände $[0;\infty[$. Eine detailliertere Auseinandersetzung wird in Abschnitt 3.3 gekoppelt mit *maschinellen Lernmethoden* vorgestellt.

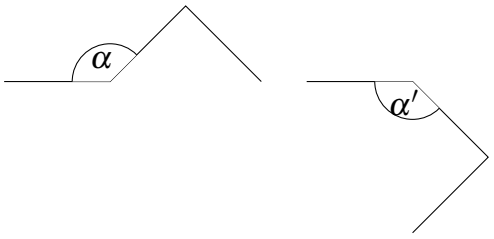


Abb. 2.5: Schematische Darstellung zweier Geometrien, welche anhand des Winkels α oder α' nicht unterschieden werden können.

Ein Gedankenbeispiel mit einer einfachen schematischen Geometrie verdeutlicht die Notwendigkeit der Verknüpfung aller internen Koordinaten miteinander. In Abbildung 2.5 sind zwei Geometrien gezeigt, welche bezüglich ihrer Winkel identisch sind. Wird der Winkel α beobachtet, kann nicht entschieden werden, ob es sich tatsächlich um α oder um α' handelt. Erst eine Analyse bezüglich des Diederwinkels kann Aufschluss geben. Verallgemeinert ausgedrückt wird für jeden Freiheitsgrad eines Moleküls im Raum eine Information benötigt, um diese eindeutig beschreiben zu können. Die Freiheitsgrade skalieren mit der Atomanzahl n mit $3n-6$ für nicht lineare und mit $3n-5$ für lineare Moleküle.

Ein Diederwinkel bringt das bekannte Problem der Periodizität mit sich, dass die Grenzen des Wertebereichs nicht den maximalen Abstand besitzen, sondern direkt benachbart sind. Zusätzlich können vier linear aneinander gereihte Atome eine starke Fluktuation im Diederwinkel und auch in den Winkeln aufweisen. Dies allein erschwert die Verwendung einer Metrik. Erschwerend dazu kommt die Skalierung der Koordinaten, da jedes Atom weitere Koordinaten mitbringt, welche sich als Attribut im Datensatz widerspiegeln.

Dieses Problem ergibt sich ebenfalls bei der Verwendung von Atompositionen in kartesischen Koordinaten. Hinzu kommt, dass die Darstellung nicht translations- und rotationsinvariant ist und ein Vergleich von Molekülgeometrien ohne vorherige Ausrichtung nicht möglich ist.[2]

[2]Die Ausrichtung zweier Konformere zueinander ist bereits nicht trivial.

2.5 Root-mean-square deviation of atomic positions

Root-mean-square deviation of atomic positions (RMSD) ist eine Standardanalysemethode in der theoretischen Chemie, Physik und Biologie, mit der die mittlere Änderung der Atomposition zweier unterschiedlicher Strukturen z.B. innerhalb einer Trajektorie beschrieben werden kann. Wie in Gleichung (2.3) gezeigt, wird die euklidische Distanz aller Atome zwischen zwei Schnappschüssen quadriert und aufaddiert und auf die Atomanzahl normiert. Damit ergibt sich ein Wert von 0 für identische und ein von 0 verschiedener Wert für veränderte Atompositionen; eine Begrenzung der Skala ist nicht vorgesehen.

$$RMSD = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\sigma_i^2} \qquad (2.3)$$
$$\sigma = \textit{euklidische Distanz}$$

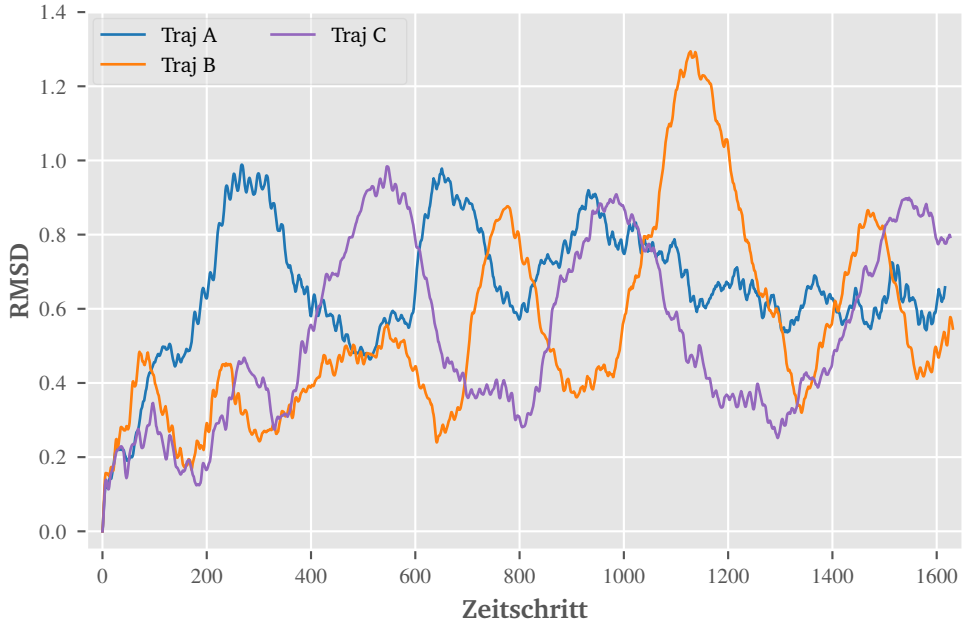


Abb. 2.6: Zeitlicher Verlauf des RMSD-Wertes zwischen der Struktur zum jeweiligen Zeitschritt und der Startstruktur beim Zeitschritt 0, nach Gleichung (2.3). Exemplarisch sind die drei Trajektorien der untersuchten Systeme aus Abschnitt 3.2 gezeigt. Es kann in allen Trajektorien eine periodische strukturelle Änderung beobachtet werden. Außerdem ist zu beobachten, dass der RMSD-Wert aller Trajektorien ansteigt.

In Abbildung 2.6 sind drei Trajektorien, welche den Anwendungsbeispielen in Abschnitt 3.2 entnommen sind, gezeigt, bei denen der RMSD-Wert über den gesamten Verlauf zwischen der Struktur beim Zeitursprung zur aktuellen Struktur berechnet wird. Es ist auffällig, dass die *Trajektorie B* einen größeren Maximalwert besitzt als *Trajektorie A* oder *C*. Der Rückschluss, dass ab einem festen Wert

von einer spezifischen strukturellen Änderung ausgegangen werden oder gar eine Reaktivität beobachtet werden kann, ist nicht uneingeschränkt möglich. Losgelöst von weiteren Analysen und Annahmen sind die Graphen nicht aussagekräftig genug, um für eine automatische Analyse ein sicheres Fundament bilden zu können.

Dieses Verfahren findet vor allem bei der Ähnlichkeitsüberprüfung von Proteinstrukturen Anwendung. Durch die Summe aller Teilchen kann und wird auch in der Regel nicht für jedes Atom exakt der jeweilige Anteil erfasst und ausgewertet. In dieser Arbeit wird der RMSD-Wert verwendet, um die Trajektorien zu visualisieren und gegebenenfalls die Zuordnung zu falsifizieren.[3]

Die Implementationen der Techniken des maschinellen Lernens der in dieser Arbeit geschriebenen Programme, Abschnitt 3.4.2 und Kapitel 4, greifen an keiner Stelle auf diesen Wert zu und können aus diesem Grund zu jedem Zeitpunkt als Referenzkriterium hinzugezogen werden.

2.6 Multidimensionale Skalierung

Für die Repräsentation von mehrdimensionalen Datensätzen gibt es eine große Anzahl an Algorithmen, welche die Dimensionalität reduzieren und die Betrachtung der Daten relativ zueinander ermöglichen. [48] Ein besonders intuitiver Fall ist die Multidimensionale Skalierung (MDS). [49] Diese erlaubt es, nD-Räume in einem Raum mit niedriger Dimensionalität abzubilden. Im Allgemeinen werden 2D-oder 3D-Räume gewählt, da diese sich für die Visualisierung eignen. Das einfache Prinzip benötigt eine vollständige Distanzmatrix aller Instanzen zueinander, welche mittels verschiedener Metriken berechnet werden kann. Diese werden nacheinander z.B. in eine zweidimensionale Ebene eingefüllt, dabei werden die Instanzen derart angeordnet, dass diese die Distanzen in der Distanzmatrix möglichst genau wiedergeben. Dabei wird für den MDS-Graphen in dieser Arbeit auf die euklidische Metrik zurückgegriffen. Für die ersten drei Instanzen ergibt sich in diesem Fall immer eine genaue Lösung, jede weitere erzeugt in der Regel den sogenannten *Stress*. Der Stress beschreibt die Differenz zwischen den realen Abständen der Instanzen und denen in der niederdimensonalen Darstellung gezeigten. Diese steigt mit einer großen Anzahl an Punkten normalerweise an und wird für eine gute Repräsentation minimiert.[4] Diese Minimierung ist lediglich lokal oder in einigen Implementationen iterativ und erzeugt nicht die global optimale Konfiguration.[5]

[3] Die Verifikation ist schwerlich möglich, systematisch falsche Ergebnisse können aber häufig einfach determiniert werden.

[4] In dieser Arbeit findet Python SKLEARN.MANIFOLD.MDS mit Standardparametern Verwendung.

[5] Mit mehreren lokalen Zufallsläufen kann das Ergebnis verbessert werden, aber ob das globale Optimum erreicht wird, kann nicht kontrolliert werden: *Globales Optimierungsproblem* [50–52]

In dieser Repräsentation kann die Nähe von Instanzen sinnvoll interpretiert werden; die exakte Position ist nicht aussagekräftig, da diese stark abhängig von der Anzahl an Instanzen ist und nur eine einzige lokal optimale Konfiguration dargestellt ist.

Zu beachten ist der Effekt, dass eine größere Anzahl von identischen oder nahezu identischen Instanzen den gesamten MDS-Graphen auf Grund der Minimierung des Stresses verändern kann. Diese Darstellung wird für die Visualisierung und nicht direkt für die Analyse der Trajektorien verwendet. Dies ist aus den oben genannten Gründen nicht sinnvoll, da die räumliche Anordnung nicht eindeutig ist und sich daher nicht als Datenbasis für Techniken des maschinellen Lernens eignet. Mit eingeschränkten Attributsräumen, wie z.B. die $C\alpha$-Diederwinkel in Proteinen, können MDS-Methoden gekoppelt mit Clustertechniken, Abschnitt 2.7.1, durchaus erfolgreich angewendet werden [13]. Hierbei ist die Vorauswahl der Attribute ein wichtiges Kriterium, welches in dieser Arbeit wiederum vollständig automatisch erfolgen soll.

2.7 Maschinelles Lernen

Maschinelles Lernen (ML) beschreibt die Fähigkeit eines Computers, mittels Algorithmen aus Daten Verallgemeinerungen zu extrahieren und auf neue unbekannte Datensätze anzuwenden. Ein deratiger Algorithmus ermittelt die relevanten Daten für die Beurteilung entweder aus den Trainingsdaten direkt, wobei die Kategorisierung oder der Zielwert der Daten bekannt ist, oder ermittelt in einem gegebenen Rahmen Gemeinsamkeiten ohne vorherige Information über Kategorien. [4,53,54]

Die Trainingsdaten werden als *Trainingssatz* und die damit verbundenen ML-Methoden werden als *überwachtes Lernen* bezeichnet. Dabei kann entweder für einen input-Datensatz ein kontinuierlicher Rückgabewert angelernt werden – Regressionstechnik – oder eine diskrete Kategorisierung, Klassifizierung, erzeugt werden. Die Kategorien werden im allgemeinen *Klassen* genannt und jeder Datenpunkt eines Datensatzes wird als *Instanz* bezeichnet. Jede Instanz besitzt *Attribute* und gegebenenfalls eine *Klasse*, Tabelle 2.1. Die Aufgabe eines überwachten Lernalgorithmus ist es, die Muster, die sich hinter den Instanzen einer Klasse verbergen, zu erkennen; dabei wird die Klassenzuordnung an sich nicht in Frage gestellt. Die Regeln, die ein Algorithmus aus einem Trainingssatz zur Klassifikation erstellt, sind entweder explizit und damit klar lesbar oder implizit und somit nicht

	Attribute				
	Attr. a	*Attr. b*	*Attr. c*	*Attr. ..*	*Klasse*
Instanz					
1	a_1	b_1	c_1	..	$Klasse_1$
2	a_2	b_2	..	..	$Klasse_2$
3	a_3	..	..	..	$Klasse_3$
..	..	..	..	..	..
..	..	..	..	..	..
n	a_n	b_n	c_n	..	$Klasse_{..}$

Tab. 2.1: Der schematische Datensatz für das ML: Die Instanznummer wird hier als Index der Attribute und Klassen verwendet; verschiedene Instanzen besitzen durchaus die selben Attribute oder Klassen. [55]

unmittelbar erkennbar in der Lösung enthalten.[6] Die Reproduktion des Trainingssatzes wird während des Lernens des Algortihmus optimiert und hat zum Ziel den Trainingssatz bestmöglich wiederzugeben. Die nahezu perfekte Reproduktion dieses Trainingssatzes ermöglicht keine Aussage über die Güte der Generalisierung für unbekannte Daten. Aus diesem Grund wird in der Regel mit einem Validationssatz gearbeitet, welcher aus der Aufspaltung eines vorliegenden Datensatzes in einen Trainings- und Validationsdatensatz erzeugt wird. Da dieser Datensatz nicht in den Trainingsdaten enthalten ist, lässt sich eine Aussage über die Qualität des angewandten Verfahrens machen. Im folgenden Abschnitt 2.7.2 werden in dieser Arbeit relevante Klassifizierungsalgorithmen und Arbeitsweisen erklärt.

Das *unüberwachte Lernen* benötigt keinen Trainingssatz und ermittelt in gegebenem Rahmen Muster und ermöglicht die Gruppierung von Instanzen zu Klassen. In Abschnitt 2.7.1 wird neben den hier verwendeten Clusteralgorithmen auch ein Gütekriterium vorgestellt. Jeder *unüberwachte Lernalgorithmus* kann mittels einer nachgelagerten Validation in eine überwachte Lernmethode überführt werden.

Im ML gilt zusammenfassend gesagt, dass die verwendeten Algorithmen nicht komplex sein müssen, um komplexe Datensätze zu analysieren. Die Hauptarbeit ist im Allgemeinen die Datenauf- und -vorbereitung, damit die Algorithmen diese Daten analysieren können.

[6] Inbesondere beim Deep Learning, bei dem mehrfach rekursive oder hierarchische Abstraktionen und Repräsentationen zum Lernen verwendet werden, ist diese einfache explizite Struktur nicht erkennbar. Hierfür werden besonders die so genannten künstlichen *Neuronalen Netze* als ML-Technik verwendet.

2.7.1 Clusteranalyse

Die Clusteranalyse ist eine unüberwachte Methode des ML. [56] Hierbei werden in einem Datensatz Gemeinsamkeiten und Strukturen gesucht, welche eine Separation der Daten in Untergruppen zulassen. In dem zu analysierenden nD-Raum können unterschiedliche Metriken als Grundlage dienen. Hinzu kommen Techniken, um eine sinnvolle Separation durchzuführen. Es können sogenannte *weiche* und *harte* Clustermethoden zur Anwendung kommen. Bei letzteren wird jede Instanz exakt einem einzigen Cluster zugeordnet, wie z.B. nachfolgend beim *Hierarchischem Clustern* gezeigt. In *weichen* Clustermethoden wird jeder Instanz eine Zugehörigkeitswahrscheinlichkeit zu jedem Cluster zugeordnet. Dies kommt z.B. im Folgenden beim *Expectation-Maximization*-Algortihmus zum Einsatz und ermöglicht eine Beurteilung oder Optimierung der Wahrscheinlichkeitsverteilung. Die Methoden können mittels der größten Wahrscheinlichkeit jeder Instanz in feste Clusterzuordnungen überführt werden.

Hierarchischer Cluster-Algorithmus

Das Hierarchische Clustern (HC) führt die Berechnungen der Cluster hierarchisch aus und kann durch Agglomeration oder Division erfolgen. Bei der Division werden zunächst alle Instanzen in einem einzigen Cluster vereinigt und dieser mittels definierter Regeln[7] jeweils in kleinere Cluster zerteilt, wobei entweder eine minimale Distanz zwischen Clustern oder eine definierte Clusteranzahl als Abbruchkriterium gewählt werden kann. Es ergeben sich tendenziell Cluster mit einer ähnlichen Anzahl an Instanzen bzw. das Verfahren ist wenig sensitiv gegenüber Ausreißern.

Die Agglomeration der Cluster wird von den Instanzen hin zu Clustern durchgeführt, daher wird zunächst jede Instanz einem separaten Cluster zugeordnet und diese werden iterativ zusammengeführt zu größeren Clustern. Auch hier kann ein minimaler Abstand der Cluster zueinander oder eine Clusteranzahl als Abbruchkriterium gewählt werden. Das Verfahren ist sensitiv gegenüber Ausreißern, da die absolute Distanz einer Instanz zu anderen Instanzen oder Clustern relevant dafür ist, ob eine Vereinigung stattfindet. Dies ist ein wichtiges Auswahlkriterium für die Verwendung in der automatischen Analyse. Jede vorgestellte Methode des HC kann mittels der nachfolgen Methoden und unter Verwendung der in Abschnitt 2.4 vorgestellten Distanzen durchgeführt werden.[8]

[7] z.B. wird der Cluster mit dem größten Durchmesser in zwei Cluster geteilt und dabei eine Verdichtung im Cluster isoliert.

[8] Je nach verwendetem Programmpaket sind einige Kombinationen nicht möglich, da diese nicht sinnvoll sind; andere lassen die freie Auswahl zu, ob dies wiederum dem Benutzer wirklich hilft ist

Die Art und Weise der Berechnung des Zentrums und die Wahl des Bezugspunktes eines Clusters bezüglich einer Instanz lassen sich unterschiedlich wählen. Daraus ergeben sich direkt Methodennamen, welche in dieser Arbeit für das HC eine wichtige Rolle spielen. Die Gleichungen (2.4) bis (2.9) zeigen die Berechnung des Distanzmaßes, D_{Methode}, für die einzelnen Methoden. Die Cluster **A**, **B** und **C** mit den jeweiligen Instanzen a, b und c der Cluster dienen abstrakt als Platzhalter. Der Cluster **C** wird benötigt, da sich algorithmische Unterschiede erst nach der Vereinigung zweier Cluster erkennen lassen.

Bei der sogenannten *single-linkage*-Methode wird wie in Gleichung (2.4) gezeigt eine Instanz dem Cluster zugeordnet, der eine Instanz mit dem geringsten Abstand besitzt. Diese Methode neigt zur Kettenbildung, da keine Effekte durch Median oder Schwerpunkt einfließen. Bei der *complete-linkage* Methode, Gleichung (2.5), wird im Gegensatz dazu für jeden Cluster als Referenz die weitest entfernte Instanz gewählt und somit eine Instanz dem Cluster mit der geringsten maximalen Distanz zugeordnet. Hierbei werden meistens eindeutig separierte Cluster erzeugt. Allerdings kann eine einzige Instanz, ein entfernter Ausreißer, die gesamte Zuordnung verändern.

Zum einen gibt es arithmetische Mittelwertsmethoden, welche als Abstandsmaß[9] die mittlere Distanz aller Instanzen des Clusters **A** zu allen Instanzen des Clusters **B** verwenden. Wird dieser Wert direkt verwendet, wird von Weighted Pair Group Method with Arithmetic mean (WPGMA), Gleichung (2.7), gesprochen, da die einzelnen Instanzen unterschiedlich viel Gewicht im Resultat haben. Der Cluster **C** wird an dieser Stelle als externer Bezugspunkt eingeführt, da die Unterschiede der Methoden erst bei der Vereinigung zweier Cluster mit mehreren Instanzen beobachtet werden. Bei der Berechnung des Unweighted Pair Group Method with Arithmetic mean (UPGMA) wird wiederum die Anzahl der Instanzen jedes Clusters berücksichtigt. Auf diesem Weg gehen alle ursprünglichen Distanzen gleichberechtigt in das Abstandsmaß ein.

Zum andern gibt es Methoden, welche zunächst das jeweilige Clusterzentrum berechnen und als Abstandsmaß verwenden. Hierbei gibt es wiederum eine gewichtete, Weighted Pair Group Method with Centroid averaging (WPGMC), Gleichung (2.8), und eine ungewichtete Methode, Unweighted Pair Group Method with Centroid (UPGMC), Gleichung (2.9), um die Clusterzentren aus den Instanzen zu berechnen. Dies verläuft analog zu den oben genannten Methoden WPGMA und UPGMA. Diese Anzahl zusammen mit den vorgestellten Distanzen ergibt bereits eine große Auswahl an Algorithmen, welche als HC bezeichnet werden kön-

fraglich.

[9] Dies ist nicht mit der Metrik zu verwechseln.

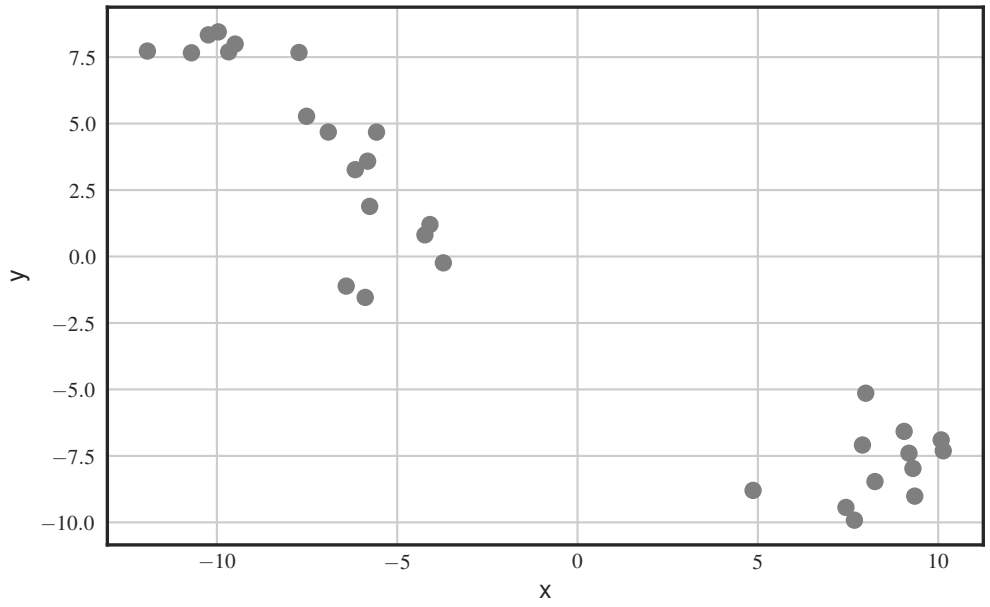


Abb. 2.7: Beispielhafter Datensatz aus 30 zufällig verteilten Instanzen im 2D-Raum.

nen.

$$\mathrm{D}_{\text{single-linkage}}(A,B) := \min_{a\in\mathrm{A},b\in\mathrm{B}}\{\mathrm{d}(\mathrm{a},\mathrm{b})\} \tag{2.4}$$

$$\mathrm{D}_{\text{complete-linkage}}(A,B) := \max_{a\in\mathrm{A},b\in\mathrm{B}}\{\mathrm{d}(\mathrm{a},\mathrm{b})\} \tag{2.5}$$

$$\mathrm{D}_{\text{UPGMA}}(A,B) := \frac{1}{|A||B|}\sum_{a\in\mathrm{A},b\in\mathrm{B}}\mathrm{d}(\mathrm{a},\mathrm{b}) \tag{2.6}$$

$$\mathrm{D}_{\text{WPGMA}}(A\cup B),C) := \frac{d(a,c)+d(b,c)}{2} \tag{2.7}$$

$$\mathrm{D}_{\text{WPGMC}}((A\cup B),C) := d(\text{Center}_C, \frac{\text{Center}_A+\text{Center}_B}{2}) \tag{2.8}$$

$$\mathrm{D}_{\text{UPGMC}}(A,B) := \frac{1}{(|A|+|B|)\,(|A|+|B|-1)}\sum_{\mathrm{a}\in\mathrm{A},\mathrm{b}\in\mathrm{B}}\mathrm{d}(\mathrm{a},\mathrm{b}) \tag{2.9}$$

Eine alternative Darstellung ist ein sogenanntes Dendrogramm, welches die gruppierten Instanzen der Cluster auf der x-Achse und die Höhe als die Distanz, bei der zwei Cluster fusionieren, die Fusionshöhe, auf der y-Achse abbildet. Diese sogenannte Fusionshöhe besitzt im Gegensatz zu den Abständen auf der x-Achse eine reale Berechnungsgrundlage, die aus dem gewählten Distanzmaß und Abstandsmaß resultiert.

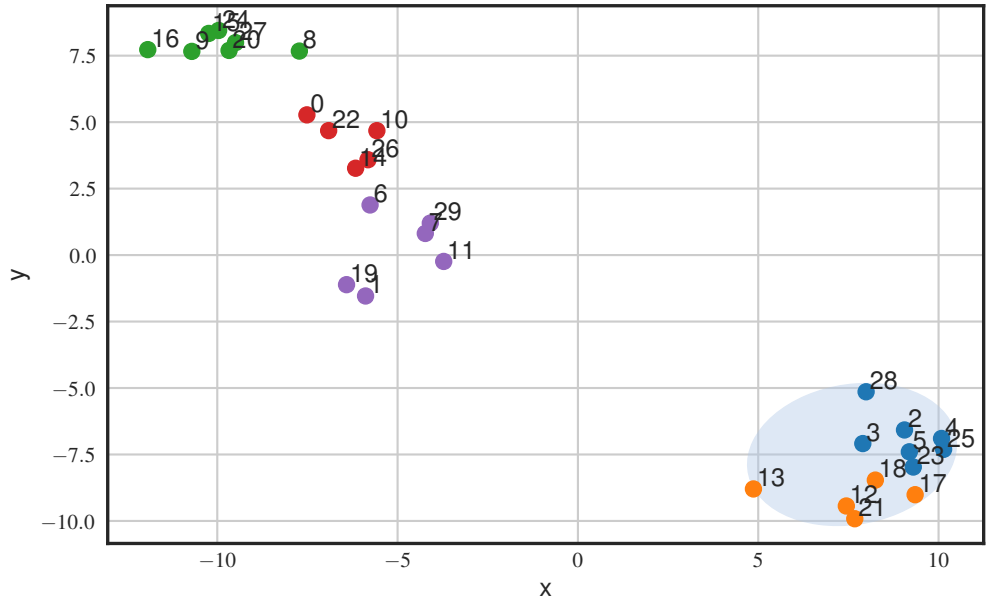


Abb. 2.8: Die Daten aus Abbildung 2.7 mittels HC: *complete-linkage/euklidisch* in fünf Cluster separiert und farblich markiert.

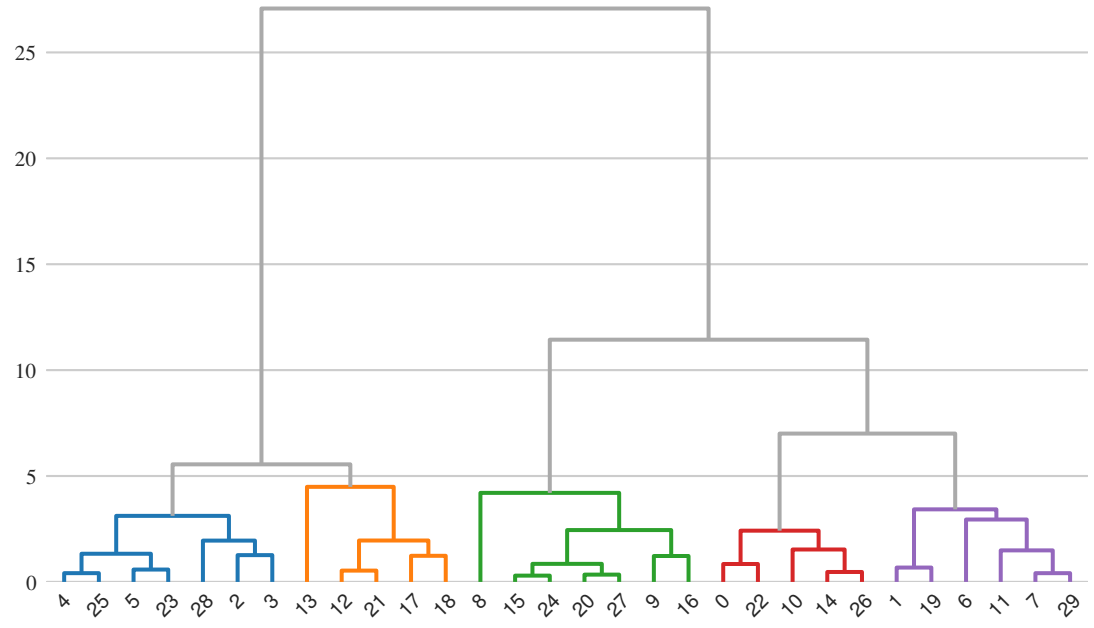


Abb. 2.9: Die Cluster aus Abbildung 2.8 als Dendrogram dargestellt. Die Cluster sind entlang der x-Achse gruppiert und die y-Achse repräsentiert die Fusionshöhe bei der Agglomeration.

Zur Veranschaulichung wird exemplarisch ein Satz aus 30 zufällig verteilten zweidimensionalen Instanzen mittels agglomerativen HC geclustert. Die vorliegenden Daten in Abbildung 2.7 werden in euklidischer Distanz, Gleichung (2.1) und der *complete-linkage*-Methode, Gleichung (2.5), geclustert. Agglomeratives HC wurde mit bis zu 5 Clustern durchgeführt und diese sind in Abbildung 2.8, im Koordinatenraum, und ebenso im Dendrogramm in Abbildung 2.9, farblich hervorgehoben. Die vorliegende Verteilung kann über die Clusteranzahl fünf, oder für eine Schnitthöhe von $[4.9; 5.2]$ erhalten werden. Bei einer Wahl von 4 Clustern oder einer Schnitthöhe $]5.2; 7]$ werden der blaue und der orangefarbene Cluster zu einem Cluster vereinigt. Dies ist ebenfalls in Abbildung 2.8 deutlich zu erkennen und durch die eingezeichnete Ellipse veranschaulicht. Die einzelnen Instanzen sind nummeriert und erlauben eine Zuordnung in den beiden Darstellungen im vorliegenden Beispiel.

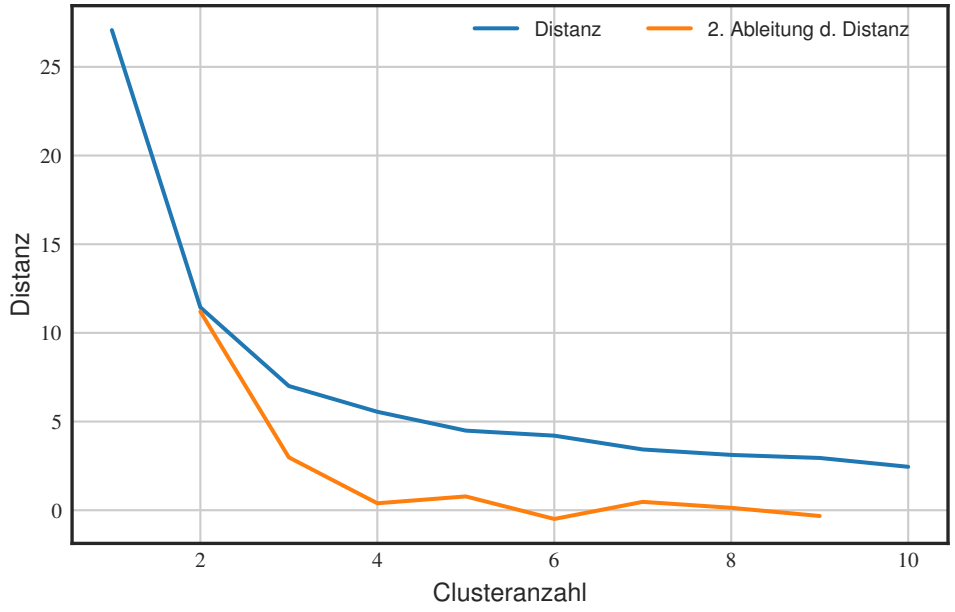


Abb. 2.10: Die Distanz aufgetragen über die veränderte Clusteranzahl, für zwei Cluster ist deutlich der *Ellenbogen* zu erkennen, die zweite Ableitung bestätigt den Eindruck.

Die Anzahl der Cluster soll in dieser Arbeit vollständig automatisiert werden, daher wird auf das sogenannte *Ellenbogenkriterium*[10] zurückgegriffen. Dabei wird die Fusionshöhe von Clustern im HC-Verfahren gegen eine veränderliche Clusteranzahl aufgetragen. Es ergibt sich ein auffälliger Knick, der sogenannte *Ellenbogen*. Über den maximalen Wert der zweiten Ableitung kann häufig eine gute

[10] Es können auch Verteilungsfunktionen [57] oder Clusterform und -größe als Kriterium Verwendung finden oder die sogenannte GAP-Statistik [58].

Clusteranzahl abgeschätzt werden. Dieses Verfahren findet bei der Implementierung der automatischen Auswertung Anwendung, Kapitel 4. Die Abbildung 2.10 zeigt Fusionshöhen, bei der zwei Cluster vereinigt werden, und deren zweite Ableitung in Abhängigkeit von der Anzahl an Clustern. Das Ergebnis bestätigt den Eindruck, der im Dendrogramm und bereits in den Datenpunkten erkennbar ist, dass die Datenpunkte um zwei klar separierte Zentren verteilt sind. Da dieser Eindruck sich bei höherdimensionalen Räumen nicht direkt überprüfen lässt, kann auf das *Ellenbogenkriterium* zurückgegriffen werden. An dieser Stelle ist das Fazit für die zufälligen Daten, dass eine Aufteilung in zwei Cluster sinnvoll ist.[11]

Ein wichtiger Vorteil der HC-Algorithmen ist, dass diese ohne zufällige Startbedingungen auskommen und daher jede Clusterung streng deterministisch ist[12] und ein einziger Durchlauf ausreichend ist, um die Instanzen algorithmisch eindeutig zu separieren. Ein Nachteil speziell für die *single-* und *complete-linkage* Methode, Gleichungen (2.4) und (2.5), ist die geringe Robustheit gegenüber Ausreißern, da eine einzelne Instanz die gesamte Zuordnung der Cluster stark beeinflusst. Diesen Nachteil federn die mittelwertsgestützten Methoden, Gleichungen (2.6) bis (2.9), ab, können aber gleichzeitig zu einer schlechteren Separation der Daten führen.

Expectation-Maximization

Der Expectation-Maximization (EM)-Algorithmus [59,60] eignet sich, um strukturell komplexe Cluster, welche in höherdimensionalen Räumen wahrscheinlicher werden, zu beschreiben. Es werden keine festen Cluster-Zuordnungen, sondern Zuordnungswahrscheinlichkeiten verwendet. Der EM ist damit folglich den *weichen* Clusteralgorithmen zuzuordnen. Die Wahrscheinlichkeiten werden, im einfachen Fall, mittels einer mehrdimensionalen Gauß-Glocke um den Clustermittelpunkt dargestellt.

Der Ablauf besteht aus einem E(xpectation)-Schritt und einem M(aximization)-Schritt, bei dem einerseits die Zuordnung der Instanzen zu den Clustern und anderseits das Clustermodell an die Daten angepasst wird. Zunächst werden zufällige Instanzen als Zentren gewählt und zu diesen eine Zuordnung aller Instanzen vorgenommen.[13] Anschließend wird die clustereinhüllende Funktion bestmöglich an die Instanzen, welche zu dem jeweiligen Cluster gehören sollen, angepasst. Dazu wird zum einen das Zentrum neu berechnet und zum anderen die Funktion

[11] Die konstruierten Zufallswerte sind in der Realität aus 5 einzelnen Gaußglockenkurven berechnet worden.

[12] Im Gegensatz zu weiteren Cluster-Algorithmen, wie *k-Means* oder Expectation-Maximization, Abschnitt 2.7.1, welche indeterministisch erfolgen.

[13] Der ungünstigste zufällige Start wäre, dass zwei Clusterzentren unmittelbar benachbart sind.

an die Instanzen des Clusters angepasst.[14] Anschaulich ausgedrückt, die clustereinhüllenden Funktionen der Cluster bewegen sich im Attributsraum. Aus diesem Grund ist die Clusterzurodnung der Instanzen nicht mehr unmittelbar richtig und es folgt wiederum eine erneute Zuordnung der Instanzen zu den wahrscheinlichsten Clustern. Anschließend beginnt wiederum eine Optimierung der clustereinhüllenden Funktion der neuen Zuordnung. Dies wird abwechselnd so lange durchgeführt, bis keine Verbesserung der Zuordnung erreicht wird, der Algorithmus konvergiert. Dieser Algorithmus findet in dem angelagerten F3-Praktikum von C. Witt dem Clustern in internen Koordinaten, Abschnitt 3.3, und in der halbautomatischen Auswertung einer Trajektorienschar, Abschnitt 3.4.2, Anwendung.

Eine sehr gute und umfassende Übersicht der Clustertechniken bieten Xu et al.[61], einen Benchmarkversuch verschiedener Clusteralgortihmen für Proteine unternehmen Shao et al.[62] und bieten ebenfalls einen guten Überblick. Eine gute Auswahl für weitere Gütekriterien in der Programmiersprache **R** bietet das Paket CLVALID[63] und eine spezieller Ansatz für HC wird in dem Paket PCLUST[64] vorgestellt.

Clusterverfahren, welche auf der Berechnung des RMSD-Wertes beruhen, sind beispielweise in Abramyan et al.[65] zu finden.

2.7.2 Klassifizierung

Die Klassifizierung von Instanzen zählt im Allgemeinen zu den sogenannten überwachten Lerntechniken des ML. Überwacht deshalb, weil für das Lernen ein bekannter vorausgewählter Datensatz, der Trainingssatz, verwendet wird, welcher bereits für jede Instanz Klassenzuordnung enthält.[15] Diese Klasse der einzelnen Instanzen stellt die Zielfunktion dar, welche mittels der eingesetzten Klassifikationsalgorithmen unter Verwendung der verfügbaren Attribute dargestellt werden soll. Während des sogenannten Lernens oder Trainings des Algorithmus werden Entscheidungsfunktionen optimiert oder Attribute gesucht, welche eine Unterteilung in unterschiedliche Klassen ermöglichen. Der in dieser Arbeit relevante Algorithmus, welcher intuitiv mittles der Attribute arbeitet, ist ein Klassifikationsbaum.[66] Dieser separiert die Instanzen an expliziten Attributswerten in zwei Teile und jeder dieser Zerteilung können weitere folgen, bis die Instanzen nach Klassen separiert sind. Für einen Trainingssatz lassen sich somit Klassifikationsbäume erstellen, welche schließlich ein einzelnes Paar Individuen anhand beliebig vieler Attribute separieren. Dies führt zu der sogenannten Überanpassung, da der Trainingssatz zu

[14] Für den aufgeführten ungünstigsten zufälligen Start würden sich diese Zentren unmittelbar auseinander bewegen.

[15] An dieser Stelle kann ebenfalls ein zu reproduzierendes Attribut als Klasse dienen.

100% repräsentiert wird, aber ein Übertrag auf weitere Datensätze unmöglich ist und eventuell Attribute, die rein zufällig verteilt auftauchen, als relevant eingestuft werden. Aus diesem Grund werden einzelne Eckpunkte festgelegt, wie z.B. die Anzahl an Individuen, welche in einem Ast vorhanden sein müssen, die Anzahl an Attributen, welche für eine einzige Zerteilung gleichzeitig herangezogen werden dürfen, oder die maximale Tiefe des Baums. Damit wird der Trainingssatz eventuell schlechter repräsentiert, aber es besteht eine höhere Wahrscheinlichkeit, dass sich dieser Klassifikator auch auf weitere Daten anwenden lässt.

Bei einer *Überanpassung* sind häufig die Ober- und Untergrenzen und Kombinationen mehrerer Attribute notwendig und grenzen den Raum eventuell unnötigerweise ein. Es muss zusätzlich, auch bei sehr guten Trainingssätzen, davon ausgegangen werden, dass sich einzelne nicht korrekte Klassenzuordnungen im Trainingssatz befinden oder zumindest sehr selten vorkommende Instanzen. Der Algorithmus soll für einen großen Teil der Instanzen robuste Ergebnisse erzielen und eine Generalisierung auf unbekannte Datensätze ermöglichen: Die Wiedergabe des Trainingssatzes für sich genommen ist kein absolutes Gütekriterium. Den Trainingssatz darzustellen wäre auch durch eine direkte Abfrage der Trainingsdaten möglich, wobei dem Algorithmus lediglich die Aufgabe zukäme die richtige Instanz auszulesen. Einen echten Mehrwert würde dies nicht bieten, daher wird in der Regel ein getrennter Trainings- und Validationssatz erstellt. Dieser nicht im Trainingssatz enthaltene Validationssatz ermöglicht die Überprüfung auf Generalisierbarkeit.

Im Folgenden werden die erkannten Strukturen aus dem unüberwachten Lernen, die Clusterbenennung der Instanzen aus Abschnitt 2.7.1 verwendet, um daraus als Trainingssatz einen Klassifikationsbaum anzulernen. Damit können weitere Daten sofort in diese Cluster eingeteilt werden, ohne dass sich die ursprüngliche Clustereinteilung aufgrund der neuen Daten verändert.[16] Es soll daher die Clusterzugehörigkeit der einzelnen Instanzen mittels eines Klassifikationsbaums und den damit verbundenen Auswahlregeln repräsentiert werden.

Der angelernte Klassifikationsbaum führt Separationen parallel zu der x- und y-Achse durch, d.h. bei jeder Aufteilung wird ein Wert eines Attributs zur Differenzierung verwendet. Der gebildete Klassifikationsbaum wird in Abbildung 2.11 gezeigt. Die erste Unterteilung kann durch einen zufällig gewählten Schnitt bei $x = -7.619$ bereits erreicht werden.[17] Der Klassifikationsbaum mit dieser Unter-

[16] Die Verknüpfung dieser beiden ML-Methoden, Clustern und nachgelagerte Klassifikation, ist nicht zwingend, ermöglicht es aber an dieser Stelle mit dem selben Datensatz die Methodik zu erklären und die Übersichtlichkeit im Rahmen dieser Arbeit zu gewährleisten. Schließlich wird eben diese Reihenfolge für die Implementation in Kapitel 4 vorgestellt.

[17] Vollständig zufällig ist dies selbstverständlich nicht, es sind die meisten Instanzen einer Klasse,

teilung wurde generiert, da durch diesen ersten Schnitt ein *reiner* Subknoten entsteht, welcher lediglich eine Klasse enthält. Dieses Verfahren stützt sich auf den sogenannten *Gini*-Index, welcher neben der Entropie als Kriterium für die Güte eines Klassifikationsbaums genutzt werden kann.[67–69] Die Gleichung (2.10) zeigt die Berechnung, wobei $p(k)$ die Wahrscheinlichkeit ist, eine Klasse in einem Datensatz zufällig zu ziehen. Dabei steht in diesem Fall der Wert von 0 dafür, dass alle Instanzen einer einzigen Klasse angehören und der Wert nahe 1 entsprechend dafür, das jede der Instanzen eine unterschiedliche Klassenzugehörigkeit besitzt.

$$S_{\text{Gini}} = 1 - \sum_{k}^{N} p^2(k) \tag{2.10}$$

Würde dies nach einem Schnitt der Fall sein, kann von einem ungeeigneten Verfahren für die Klassifikation ausgegangen werden. Sobald ein Wert von 0 erreicht wird, ist in einem Ast eine vollständige Separation erfolgt; durch z.B. eine Mindestanzahl pro Knoten oder eine maximale Tiefe des Baums kann eine Überanpassung verhindert werden. Die in diesem ersten Schritt erreichte Klassifikation erzeugt gleichzeitig einen Bereich in dem extrapoliert wird. Dieser Bereich ist in Abbildung 2.12 grün dargestellt. Wird nach erfolgreichem Training dieser Klassifikationsbaum verwendet, um eine neue Instanz mit beispielsweise den Koordinaten $(-9|1)$ zu klassifizieren, wird dies dem grünen Datensatz zugeordnet. An dieser Stelle ist leicht zu erkennen, dass eine Zuordnung zu einer der dichteren Cluster sinnvoller wäre. Diese ist den einfachen Regeln geschuldet, welche im Bereich zwischen den Instanzen einzelner Klassen zweckmäßig sind, allerdings weit entfernt an einer Generalisierung scheitern. Es sollte in jedem Fall ein sinnvoller und ausreichend großer Trainingssatz ausgewählt werden. Die Clusteralgorithmen würden diesen Datenpunkt, $(-9|1)$, einem intuitiveren Cluster zuordnen. Allerdings könnte sich, wie bereits angemerkt, damit jede Clusterzuordnung verändern.

Dieses Vorgehen macht deutlich, dass auch, je nach gültiger, zufälliger erster Separation der Daten, die nachfolgende Klassifikation vollständig unterschiedlich ausfallen kann. Somit ist der gewählte Algorithmus nicht robust und kann nicht streng deterministisch trainiert werden.[18]

Ein weiterer angelernter Klassifikationsbaum ist in Abbildung 2.13 dargestellt, die zugehörige zweidimensionale Ebene ist in Abbildung 2.14 gezeichnet worden. Es werden im Zuge der ersten Separation dieselben sieben Instanzen separiert wie im ersten Baum, allerdings wird die Entscheidungsregel bei $y = 6.467$ parallel

welche mittels eines einzigen Attributs vollständig isoliert werden können.

[18] Das Setzen des *Random-Seeds*, einer Zufallszahl, kann zwar den selben Klassifikationsbaum in unterschiedlichen Programmläufen erzwingen, deterministisch ist dies aber nicht.

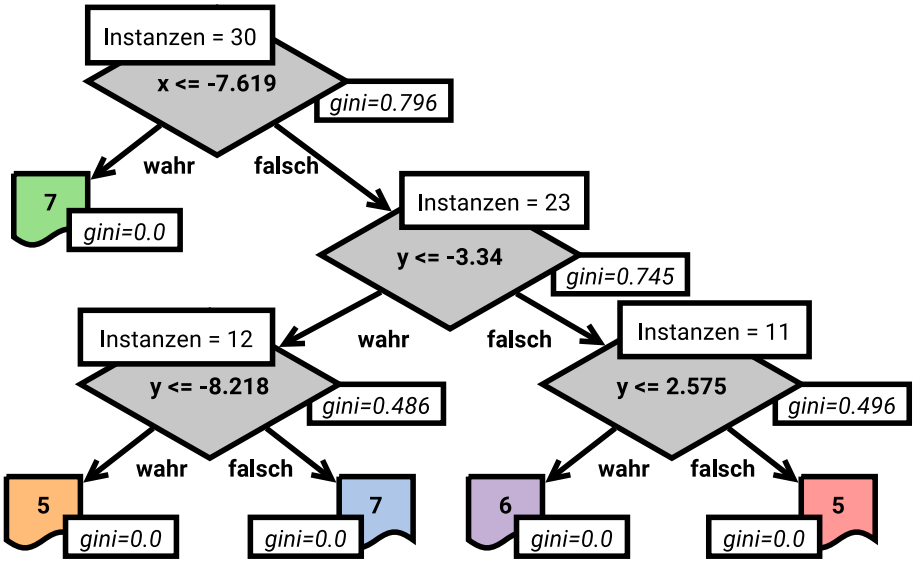


Abb. 2.11: Der gezeigte Klassifikationsbaum enthält in den farbig hinterlegten Kästen die Anzahl an Instanzen. Zunächst wird an $x = -7.619$ separiert und damit die sieben Instanzen des grünen Clusters (*gini=0.0*) separiert. Die verbleibenen 23 Instanzen besitzen ebenfalls einen niedrigeren *gini*-index.

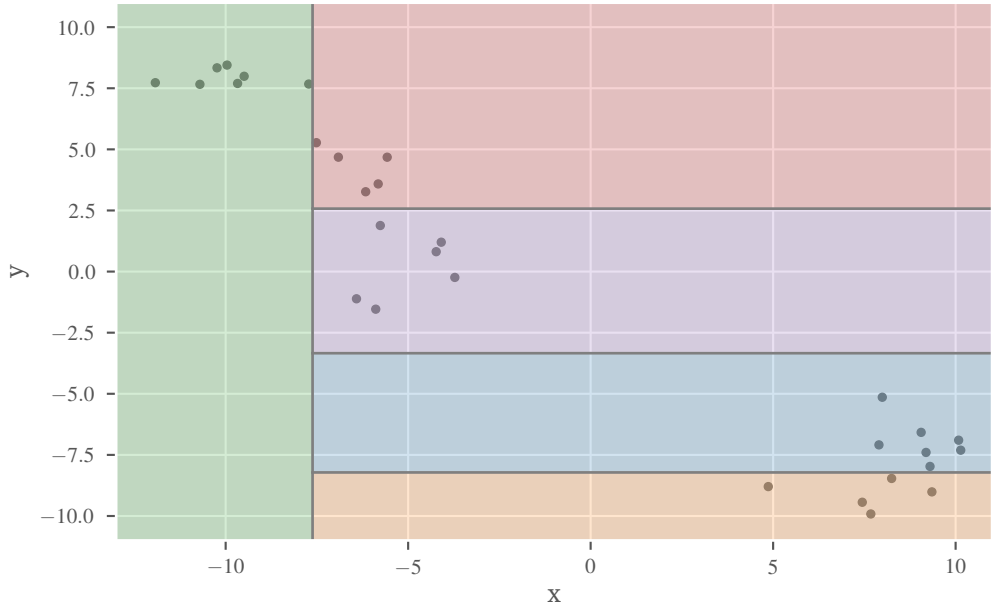


Abb. 2.12: Die Daten sind an den jeweiligen Achsen aufgespalten und ermöglichen die Separation der Individuen. Der eingefärbte Bereich veranschaulicht die Entscheidungsregeln, welche hier klar umrissende Gebiete beschreiben.

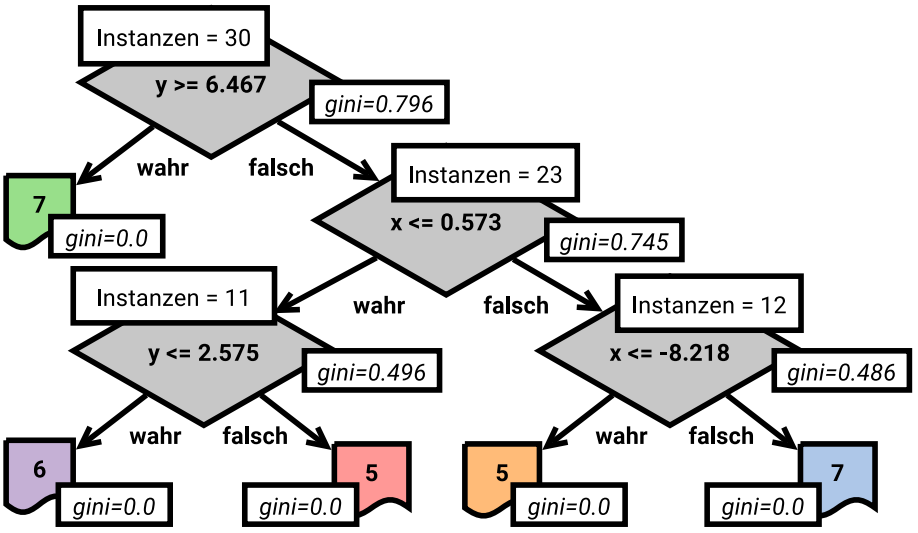


Abb. 2.13: Der gezeigte Klassifikationsbaum separiert zunächst an $y = 6.467$ und kann somit die sieben Instanzen des grünen Clusters ermitteln. Die verbleibenden 23 Instanzen werden an $y = -3.34$ separiert, die letzten Schritte werden wieder bei $x = -8.218$ und $x = 2.575$ durchgeführt.

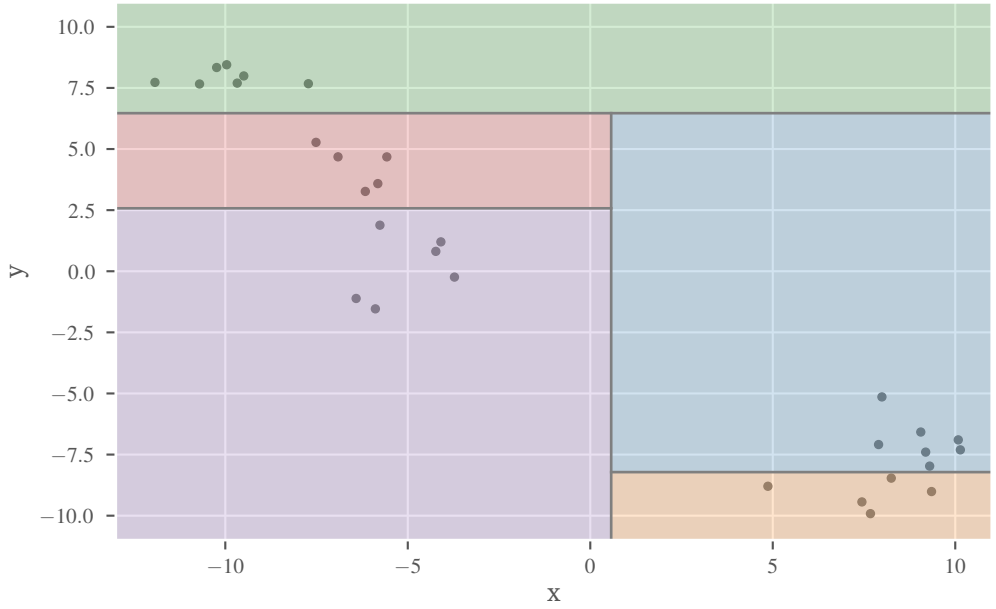


Abb. 2.14: Die Daten sind an den jeweiligen Achsen aufgespalten und ermöglichen die Separation der Individuen. Der eingefärbte Bereich veranschaulicht die Entscheidungsregeln, welche hier klar umrissende Gebiete beschreiben.

zur x-Achse aufgestellt. Die Anzahl der benötigten Separationen ist gleich und es können ebenfalls alle Instanzen richtig klassifiziert werden. Es ist folglich nicht anhand der Bäume und der Trainingsdaten zu erkennen, welcher Klassifikationsbaum besser geeignet ist. Die Extrapolation unterscheidet sich deutlich und kann unerwartete Ergebnisse erzeugen. Daher wird an diesen Schritt angeschlossen üblicherweise ein zweiter Datensatz, der *Validationssatz*, der vollständige Attribute und Klassen enthält, hinzugezogen. Auch kann über die sogenannte *Kreuzvalidation* ein Datensatz mehrfach zufällig in einen Trainings- und Validationssatz aufgespalten werden und in mehreren Durchläufen der geeignetste Klassifikationbaum ermittelt werden. In diesem Fall würden die geclusterten Instanzen mit ihrer Clusterzuordnung mehrfach in den Trainings- und den Validationsdatensatz aufgeteilt. Für jede Aufspaltung wird der Klassifikationsbaum mit dem Trainingssatz trainiert und schließlich auf den Validationssatz angewendet. Die daraus resultierende Klassifikation jeder Instanz kann mit der ursprünglichen Clusterzuordnung verglichen werden. Daraus folgt ein Anteil an richtig klassifizierten Instanzen als Gütekriterium. Da dieses für das Prozedere bekannt ist, kann es an dieser Stelle ebenfalls – seltener bei der Kreuzvalidation – zu einer Überanpassung des Validationssatzes kommen. Dabei wird der Validationssatz sehr gut repräsentiert, ermöglicht aber dennoch keine Generalisierung der Ergebnisse. Dieser kann wiederum mit Hilfe eines nicht im Optimierungsprozess enthaltenen Testsatzes überprüft werden. All dies unterstreicht nochmals die Gefahr der Überanpassung bei einer überwachten ML-Methode und stellt ein wichtiges Thema im Bereich des ML dar.

Eine weitere Möglichkeit, die Ergebnisse zu verbessern, kann durch die Verwendung einer Vielzahl an Klassifikationsbäumen erreicht werden. Diese werden unter Umständen unter Verwendung eines Attributsunterraums[19], *bagging*, erzeugt. Auch kann aus dem Trainingssatz zufällig und mit Zurücklegen ein neuer Trainingssatz erstellt werden, *bootstrapping*, welcher als Trainingssatz für einen einzelnen Klassifikationsbaum verwendet wird.[70,71] Bei einer großen Anzahl (>100 Stück) kann jeder Klassifikationsbaum einzeln für die Zuordnung einer Instanz zu einer Klasse votieren und anschließend wird die häufigste Klassifikation übernommen. Diese Verfahren sind unter dem Namen *Random-Forest*[20] bekannt. Dabei kann entschieden werden, ob für jeden Klassifikationsbaum nur ein Unterraum der Attribute verwendet werden soll. Außerdem kann die Dimension der Bäume mittels der Tiefe und der Instanzen, welche mindestens pro Entscheidung separiert werden sollen, festgesetzt werden. Auch hierbei ist eine Überanpassung gegen die Generalisierbarbeit abzuwägen.

[19] Lediglich die Verwendung einiger Attribute als Trainingssatz
[20] Ein Wald besteht schließlich aus Bäumen.

In dieser Arbeit findet der sogenannte *J48*[72], aus dem Programmpaket WEKA[73,74], in Abschnitt 3.3 Verwendung. Außerdem wird aus dem Python Paket SCIKIT-LEARN der Klassifikationsbaum und *Random-Forest*-Algorithmus in Kapitel 4 verwendet.

2.8 Rasterung

In einem nD-Raum ergeben sich auf einem beliebigen Wegstück[21] unendlich viele nicht identische Orte. Diese könnten alle theoretisch, aber nicht unbedingt numerisch, unterschieden werden; daher ergibt sich bereits für eine einfache Gerade die Möglichkeit, diese durch unendlich viele Punkte darzustellen und somit den Rechenaufwand unendlich ansteigen zu lassen. Dies ist selbstverständlich nicht zweckmäßig. Daher kann durch eine Diskretisierung des Raums die Anzahl an zu speichernden Punkten reduziert werden. Eine Rasterung oder Gitterdarstellung von Strukturen stellt eine Lösung dar.

Die Verwendung des adaptiven dreidimensionalen Gitters soll zunächst am zweidimensionalen Fall anschaulich erklärt werden. Die Beschreibung einer Konturfläche kann entweder über eine analytische Funktion oder näherungsweise durch eine ausreichende Dichte an expliziten Punkten auf der Begrenzung der Kontur erfolgen. Anstelle einer Funktion, z.B. einer Kreisbahn, kann auch die Definition der Konturlinie in einzelnen Stücken erfolgen. Dabei ist es notwendig, dass die benötigte Form eines einzelnen Stückes analytisch vorgenommen werden kann. Einerseits ist sowohl die Größe als auch die notwendige Anzahl an Punkten limitierend, andererseits die Möglichkeit der Näherung einzelner Abschnitte einer Struktur.

Werden wiederum die Konturflächen in ein diskretes orthogonales Gitter eingebracht, kann durch die Gitterpunkte eine Beschreibung erfolgen. Die Gitterpunkte überlappen nicht und die begrenzten Flächen sind parallel. Die Auflösung des Gitters muss explizit definiert werden und eine Vergröberung bedeutet eine Neuberechnung. Die Gitterpunkte zu speichern wird mit zunehmender Auflösung[22] aufwendig.

[21] Die Vorstellung einer einfachen Geraden im zwei- oder dreidimensionalen Raum sollte an dieser Stelle genügen.

[22] Gleichzeitig werden mehr und mehr Fließkommastellen notwendig, da das Gitter immer feiner wird.

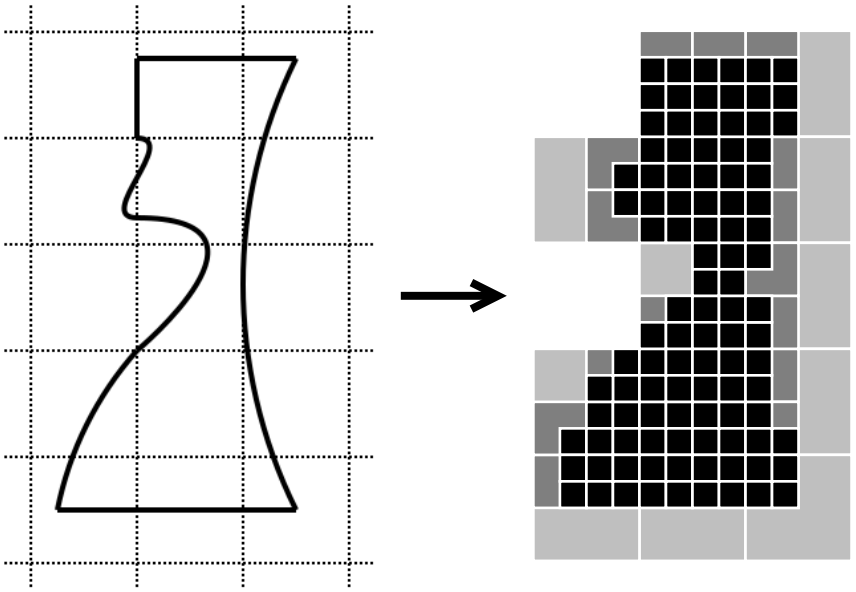


Abb. 2.15: Die Repräsentation einer Büste in Gittern unterschiedlicher Auflösung.

2.8.1 Quaternärbaum

Eine interessante Möglichkeit an dieser Stelle ist der sogenannte Quaternärbaum, *QuadTree*, welcher rekursiv einen zweidimensionalen Raum in jeweils vier exakt gleich große Unterräume zerteilt. Der erste Knotenpunkt unterteilt den gesamten Raum in vier gleichgroße, in dieser Arbeit quadratische, Unterräume und benennt diese jeweiligen Knoten mit ganzen Zahlen von 0 bis 3. Dies wird bis zu einer gewünschten Tiefe also Lagen im Baum, durchgeführt und somit der gesamte Raum kodiert. Diese Kodierung wird *Pfad* genannt und ermöglicht den Zugriff auf explizite Quadrate bzw. die *Pixel*. In dem Quadtree können an jedem Knoten Informationen über die Art und Weise der Befüllung abgelegt werden. Die Befüllung des Quadtrees mit Objekten wird mittels eines Strahlengangs[75] durchgeführt, wobei alle Pixel als besetzt markiert werden, durch die der Strahl verläuft. Die bekannten Nachbarschaftsbeziehungen der Pixel im Raum verringern den Rechenaufwand, da an Stelle von kartesischen Koordinaten direkt mit dem kodierten Pfad gearbeitet werden kann. Die Anzahl an Lagen beeinflusst die Auflösung und damit die Wiedergabegenauigkeit der Details. Es müssen lediglich die kodierten Pfade der besetzten Pixel gespeichert werden, welche wiederum aus ganzen Zahlen bestehen, wodurch eine Komprimierung erreicht wird. Eine fiktive Büste mit unterschiedlichen Auflösungen ist in Abbildung 2.15 dargestellt. Es ist zu erkennen, das je nach Auflösung einzelne Details hervorgehoben werden oder verschwinden. Ein weiterer Vorteil ist, dass nur der definierte Raum in der gewünschten Auflösung repräsentiert wird und dass durch die Hierarchie nach dem Einfüllen eine gröbere Auflösung bereits vorhanden ist. Zudem bleiben die Vorteile des Gitters

erhalten, da die Pixel parallel sind und in keinem Fall überlappen. Jeder unbesetzte Unterraum kann bereits frühzeitig im QuadTree erkannt und gegebenenfalls vernachlässigt werden. In Abbildung 2.16 sind die Subräume bis zum kleinsten Pixel dargestellt, der zugehörige Pfad 3-2-3 ist im Dendrogramm Abbildung 2.17 farblich markiert.

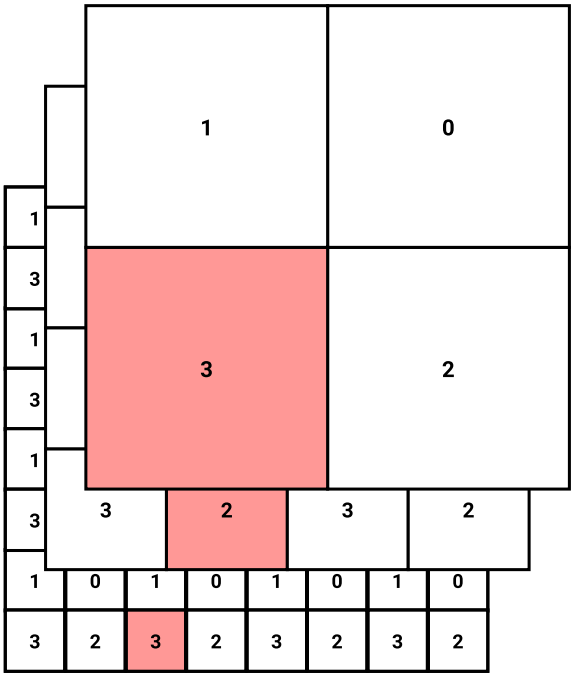


Abb. 2.16: Die eindeutige Kodierung des Pixels 3-2-3 in der dritten Lage des QuadTrees.

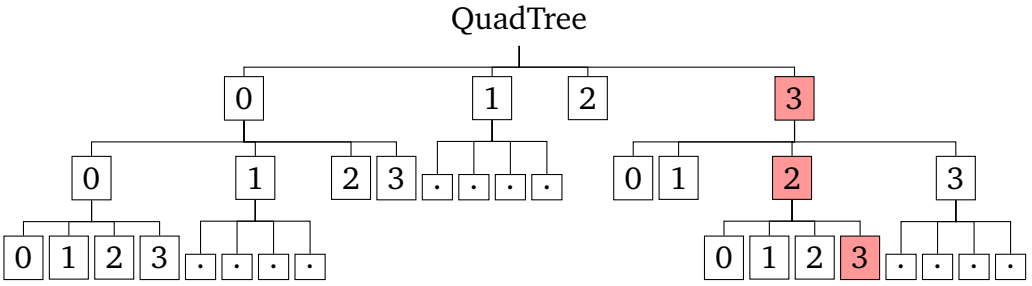


Abb. 2.17: Schematische Darstellung des rekursiven QuadTrees mit drei Lagen; Der markierte Pfad 3-2-3.

Die Problematik der Orientierung einer einfachen Linie im Raum wird in Abbildung 2.18 dargestellt. Die Pfade sind im Dendrogramm in Abbildung 2.19 farblich markiert. Die Pfade der Orientierung, die die roten Pixel erzeugen, ist: 3-3 und 3-2. Wie bereits erwähnt, lässt sich die gesamte Information in einem Vektor von ganzen Zahlen komprimieren. Das Resultat nach dem Befüllen des QuadTrees mit einer einfachen Struktur ist stark von der Orientierung abhängig. Die ideale Befüllung in diesem Fall ist in Rot dargestellt. Dabei ist die Linie parallel zum Gitter

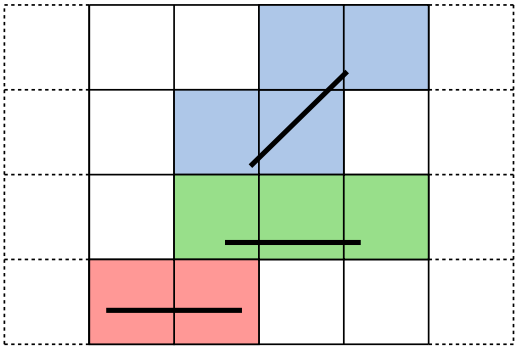


Abb. 2.18: Eine Linie im QuadTree: in *Rot*: ideale Anordnung mit minimaler Pixelanzahl, in *Grün*: nicht idealer horizontaler Fall mit deutlicher Vergrößerung der Projektion, in *Blau*: diagonale Ausrichtung bezüglich des Gitters.

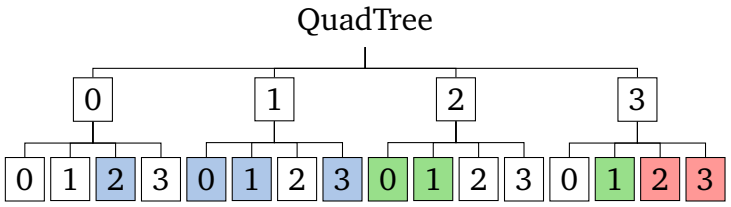


Abb. 2.19: Schematische Darstellung des rekursiven QuadTrees mit zwei Lagen; die markierten Pixel sind die besetzten Pixel bei unterschiedlicher Lage im QuadTree.

ausgerichtet und die Anzahl an benötigten Pixeln ist minimal. Die grünen Pixel zeigen, dass dieselbe Struktur trotz paralleler Ausrichtung zum Gitter größer erscheint. Im ungünstigsten Fall, in Blau dargestellt, kann die ursprüngliche Struktur vollständig verzerrt werden. Dies kann wiederum mit einer erhöhten Auflösung korrigiert werden. Für die in dieser Arbeit verfolgten Ziele ist die möglichst exakte Repräsentation nicht notwendig, sondern die Komprimierung der Daten. Daher wird ein Gleichgewicht zwischen Genauigkeit, Kompression und Übertragbarkeit angestrebt.

2.8.2 OctalBaum

Die Erweiterung in die dritte Dimension wird OctalBaum, OctTree, genannt.[76] Hierbei wird der Raum rekursiv in acht gleichgroße Unterräume unterteilt, welche *Voxel* genannt werden. Diese Voxel werden, wie in Abbildung 2.20, gezeigt in dieser Arbeit nummeriert und die Kodierung als Pfad erlaubt die Darstellung einer dreidimensionalen Struktur als Liste von ganzen Zahlen.

Die Auswertung der entstanden Voxel wurde an die Auswertung von Lidar-Scanner-Daten[77–79] angelehnt, bei denen die erfassten Punkte im Raum zunächst in einen OctTree überführt werden. Zwei Voxel können über die Seitenflächen, die Kanten oder die Ecken verknüpft sein und erzeugen, wie in Abbildung 2.19 ange-

deutet, eine Ungenauigkeit bezüglich der benötigten Voxel für die Repräsentation einer Struktur.

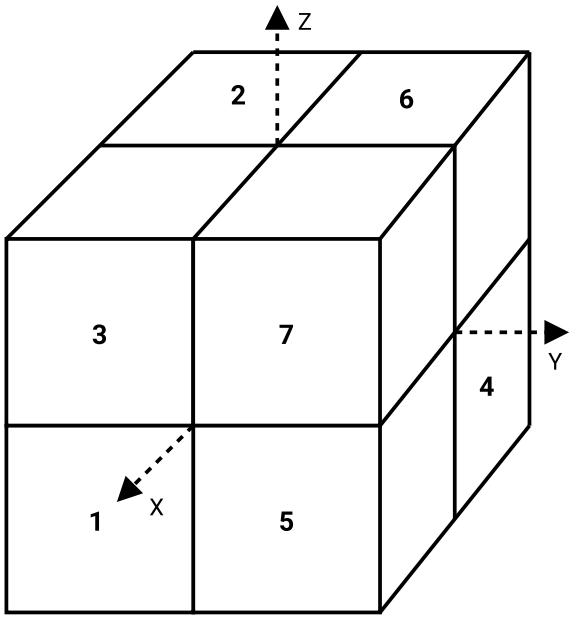


Abb. 2.20: Benennung der Voxel im kartesischem Koordinatenraum mit den ganzen Zahlen von 0 bis 7.

Der OctTree findet im Abschnitt 3.4.2 und Kapitel 4 Verwendung, das genaue Prozedere und die Überführung von molekularen Geometrien in einen OctTree und die resultierenden Parameter werden näher in Abschnitt 3.4.1 erklärt.

Kapitel 3

Konzeptionelle Entwicklung

Es werden neben der Art und Weise der klassischen Analyse der vorliegenden Phototrajektorien auch Analyseansätze mittels traditioneller Koordinaten in Verbindung mit Methoden des ML vorgestellt. Für den halb- und vollautomatischen Ansatz, welcher eine Kombination verschiedener ML-Techniken verwendet, werden die Rasterung der molekularen Geometrien in einem OctTree und die daraus berechneten Attribute detailliert vorgestellt.

Die Berechnung der Rasterung wurde eigenständig in einem Java-Programm implementiert und die Algorithmen des ML wurden aus Paketen der freien Programmiersprachen R und Python entnommen und in separaten Programmen in der jeweiligen Sprache implementiert.

3.1 Klassische Auswertung

Anhand der Beschreibungen in den Veröffentlichungen zu Photoschaltern [32] soll zunächst ein typischer, nicht-automatischer Auswertungsablauf von MD vorgestellt werden. Die Arbeiten beschäftigten sich mit der Dynamik von Molekülen, für die ein bestimmter Reaktionsmechanismus vorgeschlagen wurde. In N. O. Carstensen [32] wurde die Photoisomerisierung eines verbrückten Azobenzols von *cis* nach *trans* und der inverse Fall untersucht. Die größte Änderung des Moleküls ist in der zentralen Azo-Gruppe sowie der gegenüberliegenden Ethyl-Brücke zu finden. Das in dieser Arbeit betrachtete Derivat ist in Abbildung 3.2 gezeigt. Die Betrachtung der Isomerisierung zeigt, dass neben der größten Änderungen in der Azo-Gruppe die ebenfalls gegenüberliegende Ether-Brücke den größten Beitrag zur strukturellen Änderung besitzt. In beiden Fällen – der Studie durch N. O. Carstensen als auch die vorliegende Studie – wird ausschließlich ein einfacher Bereich der ausgewählten internen Koordinaten der besagten zwei Strukturelemente der Moleküle be-

trachtet. Allein anhand dieses Koordinatenunterraums der Moleküle wird über die erfolgreiche Photoisomerisierung entschieden. Im Allgemeinen geschieht dieses mittels kleiner Programme sowie visueller Beurteilung. Die Wahl der Programmsprache und der Umfang des Programms hängen vom Umfeld des Anwenders ab, sind spezifisch und werden in der Regel nicht zusammen mit den Veröffentlichungen publiziert. Die Auswertung basiert darauf, dass Produkt und Edukt bekannt sind. Nach der visuellen Betrachtung der Isomerisierung von *cis* $\rightleftharpoons$ *trans* sind in den vorliegenden Arbeiten[32,36] die Diederwinkel beider Brücken CNNC und CCCC ausgewählt worden.

In der Dissertation von N. O. Carstensen[37] wird über die Berechnung der Quantenausbeuten folgendes geschrieben:

> The settings used for the calculation of the quantum yields were carefully chosen by checking against the trajectory movies, if the proposed classification into *reactive* and *unreactive* was correct. A typical setting as used in the QM/MM dynamics of brAB [...] for identifying the trans-isomer is:
>
> - CNNC-dihedral $> 113^\circ$ and $< 163^\circ$ ($138^\circ \pm 25\%$)
> - CCCC-dihedral (of the ethylenic bridge) $> 61^\circ$ and $< 141^\circ$ ($101 \pm 40\%$)
>
> The trajectory is considered reactive only if the product structure is reached and held for 50 frames (= 50fs).

Diesem ist zu entnehmen, dass auch die visuelle Beurteilung zu diesem Ergebnis geführt hat. Die Grenzen des Toleranzintervalls von $138^\circ \pm 25\%$, mit einem festen prozentualen Intervall von 25% könnten, sollte daraus eine generelle Heuristik abgeleitet werden, irreführend sein, da dies den Rückschluss zuließe, dass ein kleiner Zielwert auch ein kleineres Toleranzintervall erzeugte.[1] Dem widerspricht das zweite Toleranzintervall von $101 \pm 40\%$. Ein absoluter Toleranzbereich ist zweckmäßiger. An dieser Stelle verschweigt der Autor an keiner Stelle, dass die Parameter auf Grund der Betrachtung der Visualisierung des Films ausgewählt wurden, allerdings wird dieses *Erleben* der Videos durch die festen Parameter maskiert und objektiviert.

Analog dazu müssen für das in dieser Arbeit betrachtete Derivat, die Diederwinkel der Azo- und der Ethergruppe betrachtet werden. Diese Diederwinkel einer Trajektorie ohne Isomerisierung, also *unreaktiv*, ist in Abbildung 3.1 gezeigt.

[1] Der Zielwert 0° bekommt eine Toleranz von $\pm 0^\circ$.

Aus diesem Graphen kann eindeutig abgelesen werden, dass keine Isomerisierung stattfindet, und es bleiben keine Freiräume für eine weitere Spekulation.

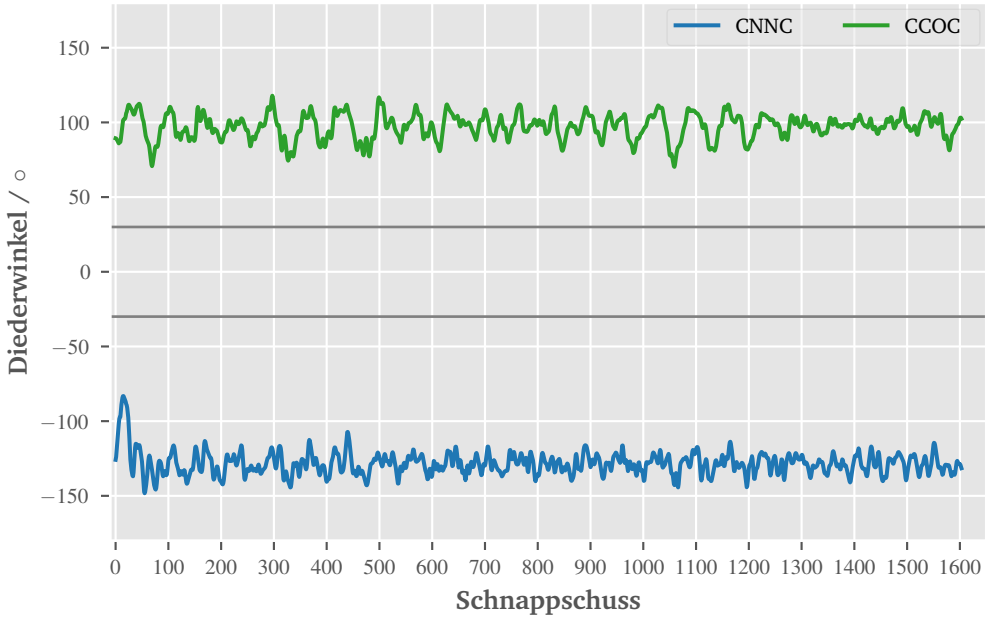
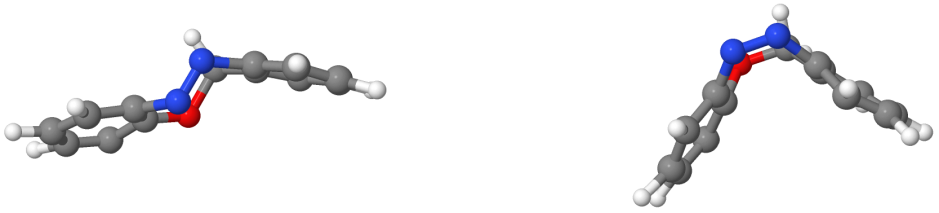


Abb. 3.1: Diederwinkel der Brückenatome des brAB-O in einer unreaktiven Trajektorie: Beide Diederwinkel nähern sich dem Zielwert von $\approx 0°$ nicht an und zeigt damit keine Photoisomerisierung.

Ein weiterer exemplarischer Graph ist in Abbildung 3.3 gezeigt. Die Trajektorie startet als *trans*-Isomer und es ist zu erkennen, dass das definierte Ziel von $\approx 0°$ für beide Diederwinkel erreicht wird, jedoch nicht für einen längeren Zeitraum und es folglich nicht stabil als *cis*-Isomer anzutreffen ist.

Für die Auswertung der MD ist eine einzelne Trajektorie nicht repräsentativ, sondern die Statistik aus einer größeren Zahl (mehrere hundert Stück), dabei wird der prozentuale Anteil an reaktiven Trajektorien, die sogenannten *Quantenausbeuten*, berechnet. Die Betrachtung der Visualisierung und die internen Koordinaten der *reaktiven* und *unreaktiven* Trajektorien zeigt einerseits eindeutig einen Zusammenhang, andererseits aber auch die große Empfindlichkeit dieser Parameter. Die Wahl eines Zielwertes des Diederwinkels benötigt aufgrund der Oszillationen der ausgewählten, reaktionsbeschreibenden Diederwinkel, in Abbildung 3.3, neben dem Zielwert eine Toleranz und zusätzlich ein Zeitintervall, nach welchem ein Übergang in das jeweils andere Isomer als abgeschlossen gilt. Damit die Betrachtung der Visualisierung entfallen kann, müssen diese zwei Parameter sinnvoll gewählt werden. Hierbei ist zu berücksichtigen, ob die quantenchemische Rechenmethode qualitativ richtige Ergebnisse liefert und die Quantenausbeuten richtig



(a) *trans*-Isomer mit ca. 180° Diederwinkel der Brücken CNNC und CCOC

(b) *cis*-Isomer mit ca. 0° Diederwinkel der Brücken CNNC und CCOC

Abb. 3.2: Isomere des brAB-O.

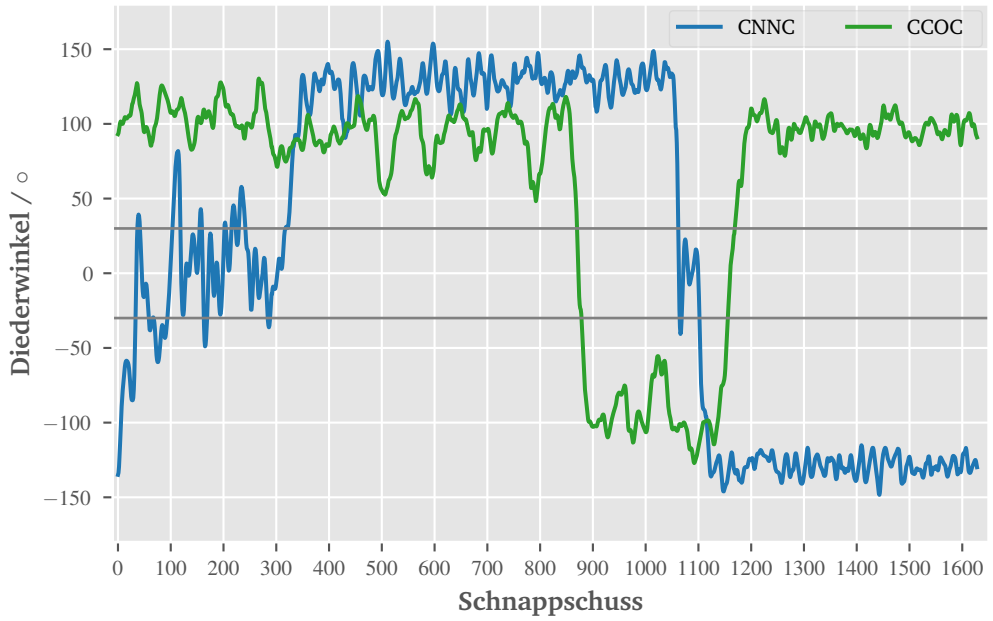


Abb. 3.3: Der zeitliche Verlauf der Diederwinkel der beiden Brücken aufgespannt durch CNNC und CCOC des brAB-O.

sind, ohne die erwarteten Quantenausbeuten künstlich zu erzeugen. Wahrscheinlich wird dieses Vorgehen ohne globalen Standard fehleranfällig sein, und bekannte experimentelle Quantenausbeuten könnten ein objektives Vorgehen erschweren. Zusätzlich findet eine Überprüfung anhand der Visualisierung statt, welche mutmaßlich die richtigen Quantenausbeuten liefert. An dieser Stelle sei erwähnt, dass diese Projektion einer dreidimensionalen Struktur auf eine zweidimensionale Fläche – dem Monitor – ebenfalls fehlerbehaftet ist. Hinzu kommt, dass für jeden Datenpunkt die menschliche Wahrnehmung als Grundlage dient. Diese lässt sich noch schlechter standardisieren und vergleichen.[2] Ebenfalls sollte erwähnt werden, dass einzelne *unvorhergesehene Ereignisse* mit einfachen Programmen oft nicht erfasst werden, da diese nicht im Erwartungsraum enthalten sind. Hierauf wird in Abschnitt 5.3 gesondert eingegangen.

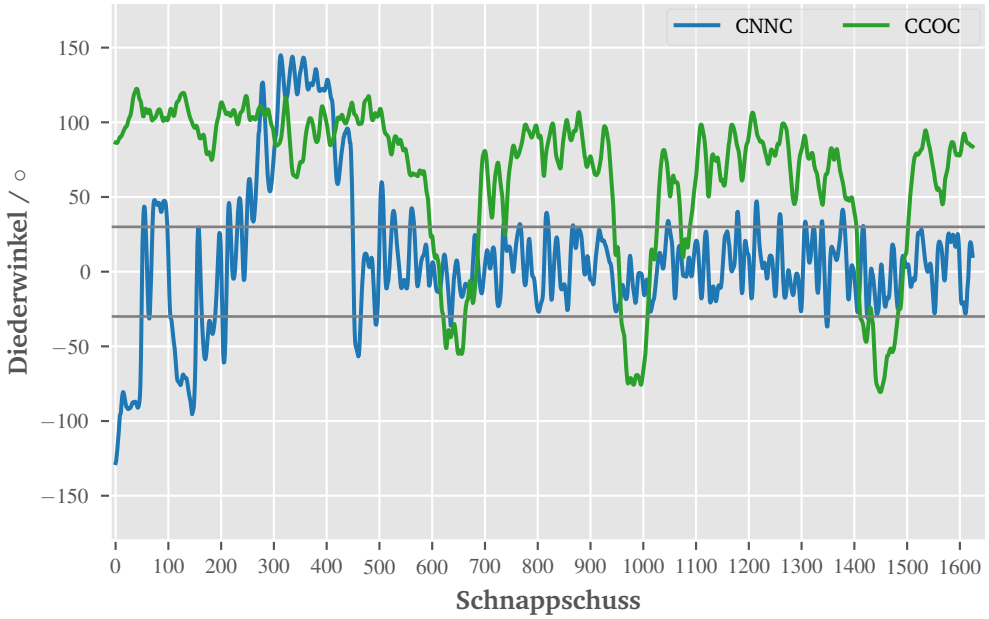


Abb. 3.4: Der zeitliche Verlauf der Diederwinkel der beiden Brücken aufgespannt durch CNNC und CCOC des brAB-O. Durch horizontale Linien ist der Toleranzbereich von 30° um den Zielwert von 0° eingezeichnet.

Die Auftragung der Diederwinkel der Brückenatome CNNC und CCOC einer reaktiven Trajektorie in Abbildung 3.4 verdeutlicht die Problematik der Bewertung einer Trajektorie. Der CNNC-Diederwinkel nähert sich dem Zielwert von 0° bei etwa Zeitschritt 450 an und fluktuiert um diesen. Der CCOC-Diederwinkel folgt erst ab Zeitschritt 600 und fluktuiert stärker, wobei der Bereich von $0° \pm 30$ deutlich

[2] Selbstversuche des Autors und eines Praktikanten zeigten Neigungen zu extrapolieren oder zu verkürzen. Der Faktor Mensch als Fehlerquelle sollte für die Reproduzierbarkeit ausgeschlossen werden.

verlassen wird. Die Auswertung dieser Trajektorie liefert je nach gewähltem Toleranzintervall und der gezählten Schnappschüsse ein unterschiedliches Ergebnis.

Im Folgenden werden 100 Trajektorien[3] über den CNNC-Diederwinkel ausgewertet. Die Trajektorien werden mit *trans*-Isomeren gestartet und der Zielwinkel beträgt 0°. Die Toleranz wurde von 0° bis 70° und das Zeitintervall, in dem sich der Diederwinkel in diesem Winkelbereich befinden muss, von 0 bis 80 Schnappschüsse variiert.[4] Nach der erreichten Anzahl von Schnappschüssen im angegebenen Toleranzintervall werden die Trajektorien, wie üblich, als *reaktiv* klassifiziert und die restlichen verbliebenen Zeitschritte ignoriert. Es ergibt sich Abbildung 3.5, in welcher zu erkennen ist, dass die berechneten Quantenausbeuten in einem Bereich von 0% bis 95% einstellbar sind. Es ist kein Plateau zu erkennen, in dem sich bei Variation der Parameter die Quantenausbeuten nicht ändern, folglich kann aus diesem Graphen keine Begründung für die Wahl der *richtigen* Parameter gefunden werden. Es kann eine steile Flanke erkannt werden, bei der sich bei kleiner Veränderung der Anzahl der Schnappschüsse die Quantenausbeuten stark verändern. Letzterer Befund ist kritisch zu beurteilen, da in einem relativ schmalen Intervall der erlaubten Toleranz eine große Veränderung der Quantenausbeuten erreicht werden kann und keine hinreichende Einschränkung derselben zu erkennen ist. Die Größe des Toleranzintervalls besitzt einen geringeren Einfluss.

In Abbildung 3.6 werden für drei mögliche Quantenausbeuten von 50, 60 und 70% mögliche Parameterkombinationen dargestellt. Es ist deutlich zu erkennen, dass verschiedenste Kombinationen die gewünschten Quantenausbeuten liefern, dabei sind nicht vereinzelte Ausreißer zu beobachten, sondern es lassen sich fast beliebige Kombinationen wählen.

Ziel im Folgenden ist es, unüberwachte ML-Methoden zu implementieren, welche die Auswertung standardisieren und die Aufklärung eines Mechanismus ermöglichen und idealerweise eine Definition der Ziel-Strukturen automatisieren. Es soll eine Analyse von MD durchgeführt werden, bei denen mögliche Edukte und Zwischenstufen unbekannt sind. Hierzu ist, wie bereits ausgeführt, eine reine Analyse der berechneten Quantenausbeuten nicht ausreichend bzw. der Vergleich mit den vorgegebenen Quantenausbeuten, welche aus den Diederwinkeln berechnet werden, nicht unmittelbar ein Gütekriterium, allerdings sollten systematische Tendenzen ebenfalls erkennbar sein.

Zur Visualisierung in dieser Arbeit werden die Diederwinkel der Brückenatome CNNC und COCC zusätzlich zur Bewertung der Klassifikation der entwickelten

[3] Basierend auf den Rechnungen des Praktikums von F. Spenke, siehe Abschnitt 3.2.

[4] Eine Besonderheit der Phototrajektorien ist, dass bei einem elektronischen Übergang ein zusätzlicher Schnappschuss gespeichert wird; dies soll hier nicht weiter vertieft werden.

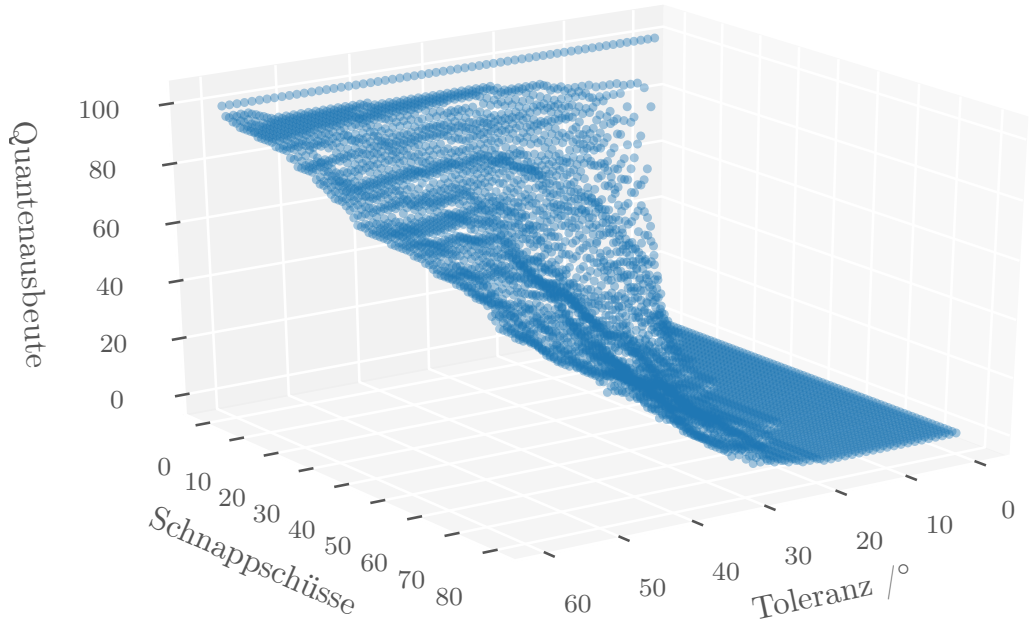


Abb. 3.5: Die berechneten Quantenausbeuten bei variierter Toleranz und Zeitintervall; es existiert, neben dem Minimum, kein weiteres Plataeu, welches die Wahl der Parameter nahe legt.

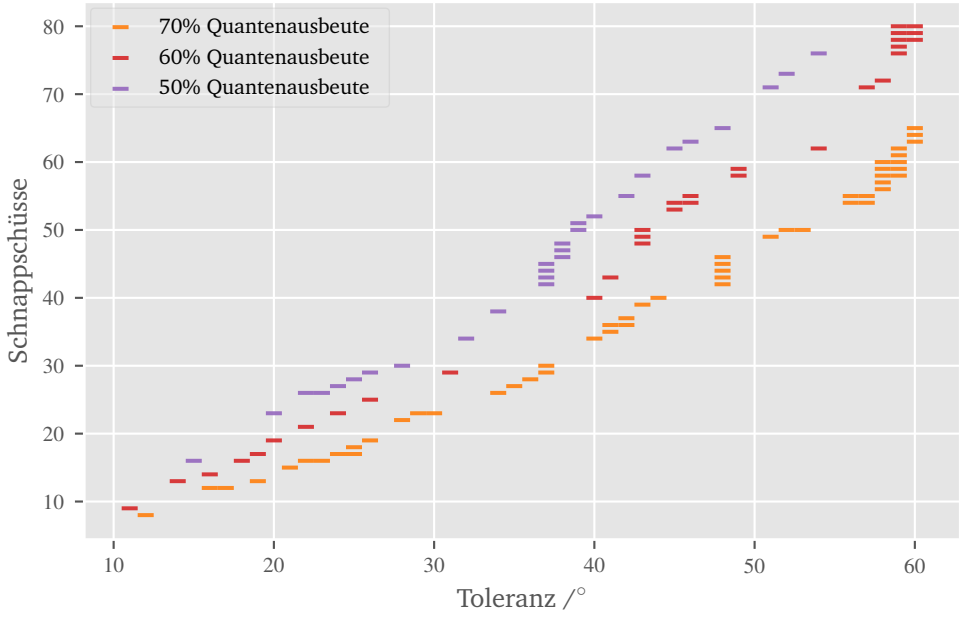


Abb. 3.6: Auftragung der möglichen Parameterkombinationen der Auswertung, welche eine Quantenausbeute von 50, 60 oder 70% erzeugen.

ML-Implementation verwendet. Diese Bewertung findet nicht in Form des Validationssatzes statt, also als Teil des ML, sondern stellt vielmehr den Testsatz dar und kommt vornehmlich zur Falsifizierung zum Einsatz.

3.2 Krafteinwirkung

Für die Simulation von MD werden in der Regel möglichst kleine sinnvolle Systeme verwendet, welche das zu untersuchende Verhalten repräsentieren. Der hier vorgestellte Fall: ›Ein Photoschalter unter Krafteinwirkung‹ ist aus der theoretischen Überlegung entstanden, dass ein in z.B. dem Molekülgerüst eines Polymers eingebauter Photoschalter unter dem Einfluss einer äußeren Kraft steht. Diese Kraft begünstigt oder verhindert den Photoisomerisationsprozess. Daher wird als Substitution für ein komplexes molekulares Gerüst in der vorliegenden Arbeit lediglich die resultierende Kraft, welche auf einen Photoschalter wirkt, vektoriell während der Simulation eingeführt. [80,81]

Das bereits erwähnte brAB-O, ein Azobenzol mit modifizierter Brücke, das 8-H-Dibenzo[b,f]-1-oxa-4,5-diazozin, Abbildung 3.7, wird unter Einwirkung von Kräften simuliert. Dabei kommen auseinander gerichtete Kräfte von 70, 140, 210, 280 und 350 pN zur Anwendung, welche an den markierten Atomen angelegt werden. Es ergeben sich die folgenden Kombinationen: $C^5C^{5'}$, $C^4C^{4'}$, $C^5C^{4'}$ und $C^{5'}C^4$. Für jede der vorliegenden Kombinationen wurden 100 Trajektorien mit einer *trans*-Startgeometrie berechnet. Somit wirkt die Kraft der strukturellen Änderung zum *cis*-Isomer entgegen. Die Simulationen werden mit dem *trans*-Isomer in *twist*- und *chair*-Konformation als Startgeometrie durchgeführt.

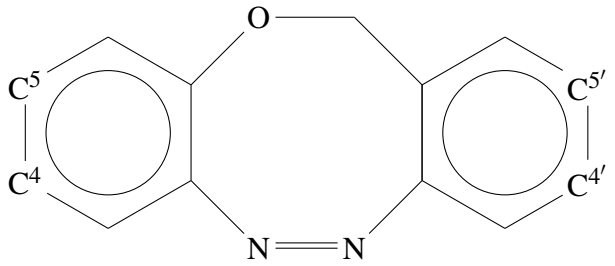


Abb. 3.7: Das Azobenzolderivat mit modifizierter Brücke: 8-H-Dibenzo[b,f]-1-oxa-4,5-diazocin (brAB-O). Die benannten Atome C^5, $C^{5'}$, C^4 und $C^{4'}$ dienen als Angriffspunkte für die angelegte Kraft.

Insgesamt liegen 4000 Trajektorien eines bekannten Systems vor, welches automatisch analysiert werden sollen. Der zu erwartende Trend, dass die Reaktivität mit steigender Kraft abnimmt, kann neben den in Abschnitt 3.1 vorgestellten Verfahren und der visuellen Beurteilung des Autors als methodische Überprüfung

dienen. Als Toleranz werden 30° und als Zeitintervall 50 fs für die Parameter der klassischen Auswertungsmethode definiert.[5] Die Tabelle 3.1 zeigt die resultierenden Quantenausbeuten der Trajektorien gestartet mit dem *chair*-Konfomer. Der CNNC-Diederwinkel eignet sich mit diesen Parametern nicht um direkt eine Tendenz abzulesen.[6] Der CCOC-Diederwinkel bestätigt das postulierte Phänomen, dass eine größere angelegte Kraft zu geringeren Quantenausbeuten führt. Die absoluten Quantenausbeuten sind, wie in Abschnitt 3.1 ausführlich gezeigt, nicht aussagekräftig, ermöglichen es allerdings, eine Vorstellung der vorliegenden Trajektorien zu entwickeln und das neue Verfahren einzuordnen.

Die gleiche Tendenz kann auch in Tabelle 3.2 für die MD des *twist*-Konfomers beobachtet werden. Die alternativen, möglichst objektiven Methoden zur Analyse sollten in jedem Fall die vorgestellten Tendenzen zeigen, welche sich auch in einer visuellen Überprüfung zeigen.

Tab. 3.1: Die Quantenausbeuten des *chair*-Konformers berechnet mit Hilfe des CCOC und CNNC-Diederwinkels. Für diese Auswertungsparameter zeigt der CNNC keinen Zusammenhang zwischen angelegter Kraft und der erhaltenen Quantenausbeute, der CCOC zeigt, dass eine größere Kraft zu geringeren Quantenausbeuten führt.

	Quantenausbeuten		Quantenausbeuten	
	CCOC	*CNNC*	*CCOC*	*CNNC*
	$C^5C^{5'}$		$C^4C^{4'}$	
70 pN	76	95	71	94
140 pN	66	95	49	94
210 pN	42	96	45	98
280 pN	29	97	32	96
350 pN	19	94	14	97
	$C^5C^{4'}$		$C^4C^{5'}$	
70 pN	73	93	64	95
140 pN	65	98	60	98
210 pN	35	90	38	93
280 pN	27	92	23	93
350 pN	21	98	12	98

[5] In Rücksprache mit Dr. T. Raeker und nach besten Wissen und Gewissen.

[6] An dieser Stelle würde der klassische Anwender die Werte in dem Sinne einstellen, dass geeignete Korrelationen erzeugt werden.

Tab. 3.2: Die Quantenausbeuten des *twist*-Konformers bestimmt mittels des CCOC und CNNC-Diederwinkels. Für diese Auswertungsparameter zeigt der CNNC keinen Zusammenhang zwischen angelegter Kraft und der erhaltenen Quantenausbeute, der CCOC zeigt, dass eine größere Kraft zu geringeren Quantenausbeuten führt.

	Quantenausbeuten		Quantenausbeuten	
	CCOC	*CNNC*	*CCOC*	*CNNC*
	$C^5C^{5'}$		$C^4C^{4'}$	
70 pN	76	96	77	98
140 pN	56	92	55	93
210 pN	50	98	42	97
280 pN	37	97	26	94
350 pN	20	97	18	97
	$C^5C^{4'}$		$C^4C^{5'}$	
70 pN	80	97	73	98
140 pN	60	94	75	99
210 pN	45	96	45	95
280 pN	30	93	29	93
350 pN	14	94	12	89

3.3 Clustern der Klassischen Parameter

Die Reduktion der Abhängigkeit der Ergebnisse von einer vorab aufgestellten Hypothese des Anwenders kann gelingen, indem Techniken des ML Anwendung finden. Zunächst wurden verschiedene Ansätze mit den internen und kartesischen Koordinaten durchgeführt, welche einen methodischen Einstieg bilden. Diese wurden in einem betreuten F3-Praktikum von C. Witt durchgeführt.[82] Verwendet wurde für diesen Ansatz das brAB-Derivat (Z)-11,12-dihydrodibenzo-[c,g]-[1,2]-diazozin-3,8-diamin, dargestellt in Abbildung 3.8.

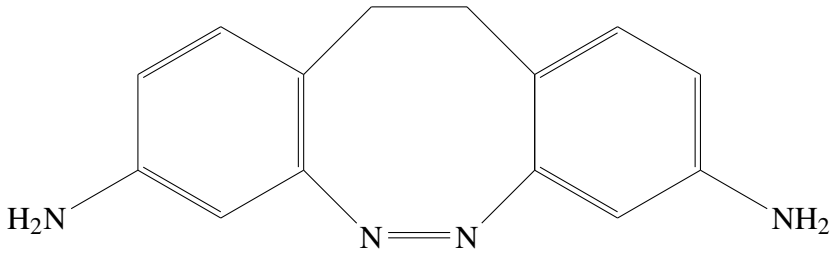


Abb. 3.8: Funktionalisiertes Azobenzol: (Z)-11,12-dihydrodibenzo-[c,g]-[1,2]-diazocin-3,8-diamin (Diamino-brAB).

Es werden zunächst Grundzustandstrajektorien in *cis*- und *trans*-Isomerie über

10 ps berechnet und daraus resultierend je 100 in den S_1-Zustand angeregte Trajektorien mit 2 ps, welche bezüglich ihres *cis*→*trans*-Übergangs untersucht werden. Zu diesem Zweck werden die Trajektorien visuell klassifiziert und mit automatischen ML-Techniken verglichen, als Datengrundlage dienen alle Winkel, Diederwinkel und Abstände.[7] Diese 109 Attribute wurden mit Trajektorien aus mehr als 4000 Schnappschüssen verwendet.[8]

Es werden 200 Trajektorien visuell händisch klassifiziert und mittels CHEMISTRY DEVELOPMENT KIT (CDK) und WEKA, ein Java-basiertes Programm, geschrieben, welches mittels Clustermethoden eine Separation der Trajektorien ermöglicht.[83] Um die Entwicklungsarbeit in Rahmen eines Praktikums durchführen zu können, wurden C. Witt funktionierende Programmcodefragmente zur Verfügung gestellt und die Verwendung der vorhandenen Anbindungen an bestehende Programme gezeigt. Auf diesem Weg konnte die Entwicklung einer eigenständigen Bewertungslogik in den Mittelpunkt gerückt werden. Es wurde lediglich in *reaktiv* und *unreaktiv* unterschieden, da eine weitergehende Analyse eine Überprüfung der Güte erschwert.

Neben einiger weiterer ML-Schemata wurde ein Trainingssatz bestehend aus allen Schnappschüssen beider Grundzustandstrajektorien mittels eines EM-Algorithmus geclustert. Die Separation in *cis*- und *trans*-Geometrie gelang in der euklidischen Distanz.[9] Mit Hilfe der resultierenden Clusterzentren wurden angepasste Klassifizierungsalgorithmen erstellt, welche sich im Wesentlichen an den in Abschnitt 3.1 vorgestellten Wegen orientieren. Diese werden auf die Phototrajektorien angewendet. Es zeigt sich, dass keine gute Extrapolation in den unbekannten Parameterraum möglich ist, da die Trainingsdaten einzig die Grundzustandtrajektorien enthalten. Eine Begrenzung der Klassifikation auf das *cis*- und *trans*-Isomer, folglich die Auslassung weiterer Geometrien, kann unter bestimmten Voraussetzungen die Ergebnisse verbessern, überführt den Ansatz allerdings in eine Art *überwachten* Lernalgorithmus mit sehr begrenztem Anwendungsgebiet. Die beiden erwarteten Isomere werden separat berechnet, in einem Trainingssatz vereinigt und wiederum in zwei Cluster unterteilt. Damit müssen für diesen Ansatz das Produkt und das Edukt bekannt sein, eine Entdeckung von neuen Geometrien ist nicht möglich. Vorstellbar ist, dass mit Hilfe dieses Ansatzes die Auswahl an relevanten internen Koordinaten verbessert oder sogar automatisiert werden kann, wie bei P. Tavadze et al.[27] gezeigt. Zusätzlich wurde untersucht, ob sich durch einen einzige Art der internen Koordinaten (Bindungslänge, Winkel oder

[7] Im Gegensatz zur Z-Matrix wurden redundante Informationen erstellt.
[8] An dieser Stelle wurde eine sehr feine Auflösung der Zeit gewählt.
[9] EM(WEKA): maxIterations: 100, minStdDev: $1 \cdot 10^{-6}$, numClusters: 2

Diederwinkel) bessere Ergebnisse erzeugen lassen und ob eine Normierung der Parameterräume die Ergebnisse verbessern kann.

Die Versuche, aus allen Parametern aller Schnappschüsse direkt die resultierenden Cluster mittels EM zu erzeugen, zeigen keinen Erfolg und sind sehr rechenintensiv.

Es zeigte sich, dass die Diederwinkel auf Grund des Wertebereichs $[-180, 180]$ nicht uneingeschränkt anwendbar sind. Die Grenzen des Periodizitätsintervalls stellen eine Singularität dar, bei der eine quasi unendliche kleine strukturelle Änderung genügt, um einen Sprung von 180 zu -180 zu vollführen. Dieses ist ein verbreitetes Problem und erschwert die Distanzberechnung der Geometrien mittels der Diederwinkel. Die Verwendung aller Abstände und aller Winkel bringt keine guten Ergebnisse, daher ist der Diederwinkel unabdingbar. Versuche der Normierung und Transformation zeigen keine Verbesserung.

Als Ergebnis kann festgehalten werden, dass die klassischen Parameter ohne eine gezielte Vorauswahl nicht direkt als Datenbasis verwendet werden können. Durch diese Vorauswahl, oder eine mögliche angepasste Gewichtung, wird wiederum eine Anpassung aller Parameter an einen spezifischen Fall notwendig. Dies ermöglicht keine hypothesenfreie Auswertung und erschwert den Übertrag eines Auswertungsschemas auf weitere Systeme. Außerdem zeigt sich, dass mit zunehmender Systemgröße eine große Anzahl von Attributen in einem Datensatz verwendet werden müssen. Unter der Verwendung einer hypothetischen idealen Metrik, welche alle internen Koordinaten kombiniert, müsste zumindest nicht wie in diesem Abschnitt mit redundanten Koordinaten gearbeitet werden. Es würde die eindeutige Beschreibung der Strukturen genügen.

Nachtrag: Ein Versuch, die Diederwinkel in die komplexe Zahlenebene zu überführen und diese in Koordinatenform als Attribute in einer euklidischen Distanz zu verwenden, zeigt keinen Erfolg. Dies ist damit zu begründen, dass ein spezielles Distanzmaß eingeführt werden müsste, welches die beiden Koordinaten (Realteil und Imaginärteil) zwingend aneinander bindet. Durch die Verdopplung der Parameter bei der Darstellung in kartesischen Koordinaten können weitere, nicht sinnvolle Kombinationen nicht nachvollziehbare Distanzen erzeugen. Eine Verdopplung der Dimensionalität ist verständlicherweise kein Garant für eine Verbesserung der Analyse. Weitere Auswahlregeln, speziell um einzelne Diederwinkel besser aneinander auszurichten und die Singularität zu beseitigen, wurden nicht speziell untersucht. Es besteht durchaus die Möglichkeit, ein Setup zu erzeugen, welches gute Ergebnisse liefert, allerdings wird dieses wiederum nicht generalisierbar sein. Hinzukommende Atome und Bindungen benötigen immer neue inter-

ne Koordinaten, und die Bedeutsamkeit für das untersuchte System kann schwer ohne weitere Hintergründe abgeschätzt werden.

3.4 Parameterfreier Ansatz

Aus den vorangegangen Erläuterungen folgt, dass konventionelle Auswertungen, zu denen, wie in Abschnitt 3.1 skizziert, neben den vorgestellten kleinen Programmen auch eine visuelle Beurteilung gehört, nicht fehlerfrei funktionieren. Dabei sind weniger die einzelnen individuellen Fehler interessant, sondern vielmehr die systematischen. Es fällt auf, dass das Wissen über das untersuchte System der systematischen Analyse der strukturellen Merkmale hinderlich sein kann. Ebenso ist ein fundiertes Wissen über die verwendete ML-Methodik notwendig.

Ein weiteres Argument ist, dass die visuelle Klassifizierungen der Trajektorien des ausgelagerten Praktikums sich nicht vollständig mit denen des Autors decken. Beispielsweise wurde für den letzten Teil eines Trajektorienfilms von C. Witt mit der Annahme gearbeitet, dass unter bestimmten, erfahrungsgestützten Bedingungen eine Photoisomerisierung eintreten *wird*, dass also ein Ereignis nach dem Trajektorienfilm eintreten wird, welches sich eventuell anbahnt, aber nicht definitiv zu beobachten ist. Ebenso wird die visuell wahrgenommene Grenze bei der Unterscheidung des *cis*- und *trans*-Isomers stark davon beeinflusst, welches Verhalten vorherige Trajektorien zeigten.[10] Dabei wird klar, dass die Differenzierung zwischen *reaktiv* und *unreaktiv* ohne definierte Rahmenbedingungen niemals konsistent erfolgen kann. Ähnlich wie in Abschnitt 3.1 gezeigt, bleiben Unsicherheiten, ob mathematisch korrekt festgehalten oder durch die visuelle Wahrnehmung und Erwartungshaltung geprägt. Das Ziel des folgenden Ansatzes ist es, aus einer reaktiven Trajektorie einen idealen Trainingssatz, unterstützt durch Clusteralgorithmen, selbst definieren zu können. Dazu wird nach erfolgreichem Clustern die händische Zusammenlegung unabhängig vom Abstandsmaß des Clusteralgorithmus erlaubt. Dieser Trainingssatz soll die Optimierung des Klassifikationsalgorithmus im Rahmen des *überwachten Lernens* ermöglichen.

Zu diesem Zweck wurde ein Java-Programm entwickelt, welches die im folgenden aufgeführten Schritte ermöglicht. Cluster- und Klassifikationsalgorithmen werden an dieser Stelle zunächst aus WEKA eingebunden, Bindungsmatrizen mittels CDK erstellt und der Jmol-Molekül-Viewer[84] in eine Benutzoberfläche

[10] Ein Selbstversuch des Autors, welchen das nachfolgend entwickelte Programm ermöglicht, war die Bewertung von einzelnen, zufällig ausgewählten Geometrien. Für diesen extremem Fall zeigte sich, dass, auch mit dem Wissen über die zufällige Auswahl der Strukturen, ein starker Einfluss der vorherigen Geometrien zu beobachten war. Der Wille zur Extrapolation scheint unermesslich zu sein.

eingebunden.[11] Weitere Funktionalitäten, wie beispielsweise nachfolgend in Abschnitt 3.4.1 beschrieben, werden direkt in Java entwickelt.

3.4.1 OctTree-Übersetzung

Auf Grund der schlechten Skalierbarkeit und Übertragbarkeit bei der Verwendung von internen Koordinaten wurde jede Geometrie in den OctTree übersetzt und für jeden Zeitschritt Attribute berechnet.

Für die Übersetzung in den OctTree muss ein absoluter Ankerpunkt und eine translations- und rotationsbefreite Darstellung erfolgen. Dafür wird der erste Schnappschuss jeder Trajektorie mittels der euklidischen Distanz in eine vollständige Distanzmatrix überführt. Die Atome mit dem größten Abstand werden auf die z-Achse transformiert und anschließend wird das Molekül so um die z-Achse rotiert, dass die größte Ausdehnung der Projektion auf die xy-Ebene parallel zur y-Achse ausgerichtet wird. Am Ende wird der geometrische Mittelpunkt auf den kartesischen Koordinatenursprung gelegt. Absolut dieselbe Transformation, mit der gleichen Verschiebung und Drehung, wird auf jeden Schnappschuss der Trajektorie angewendet. Das Einfüllen eines Schnappschusses in den OctTree überführt die Bindungen und Atompositionen der C-,N- und O-Atome[12] von kartesischen Koordinaten in einen segmentierten Raum. Damit erfolgt eine Simplifizierung des Moleküls, bei der lediglich das Molekülgerüst mit seinen Veränderungen übersetzt wird. Dies ist bei Betrachtung des untersuchten Reaktionstyps eine gute Methode für die Untersuchung, da die Photoisomerisierung unabhängig von H-Atomen stattfindet. Bei der Untersuchungen wie beispielsweise einem Protonentransfer, siehe T. Raeker[85], sollten speziell die involvierten Atome mit betrachtet werden. Die Auswahl der Atome und Atomtypen können in dem entwickelten Programm explizit gewählt werden. Die Bindungen werden für jeden Zeitschritt erneut berechnet, daher können Bindungen gebildet und gebrochen und entsprechend in der OctTree-Repräsentation aktualisiert werden. Die Atome werden ohne Volumeninformation und die Bindungen entsprechend dem konventionellen Symbolismus als Geraden behandelt. Es ergibt sich jeweils ein gerastertes Bild, wobei der hierarchisch aufgebaute Baum die Bildinformationen kodiert. An dieser Stelle wird die gesamte klassisch-chemische Information entfernt, allerdings ergibt sich aus den fehlenden Atomen und Bindungen zusätzlich eine Permutationsinvarianz,

[11] Die genannten Projekte wurden alle in Java entwickelt, daher bieten diese für ein eigenes Projekt ideale Bedingungen für eine bestmögliche Anbindung und Zugriff auf gewünschte Funktionalitäten.

[12] Der Input des geschriebenen Java-Programms erlaubt die Wahl der Atomtypen, Wasserstoffatome sind folglich nicht kategorisch ausgeschlossen.

welche für den Vergleich von ähnlichen Systemen ein entscheidender Vorteil ist. In Abschnitt 5.4 wird exemplarisch gezeigt, dass die resultierenden Attribute einen Vergleich verwandter molekularer Systeme erlauben. In Abbildung 3.9 werden die resultierenden Bilder einer *trans*- und einer *cis*-Konformation gezeigt. Hierbei wurden lediglich die Schweratome in den OctTree eingefüllt; in der Visualisierung werden besetzte Voxel rot dargestellt und unbesetzte transparent.

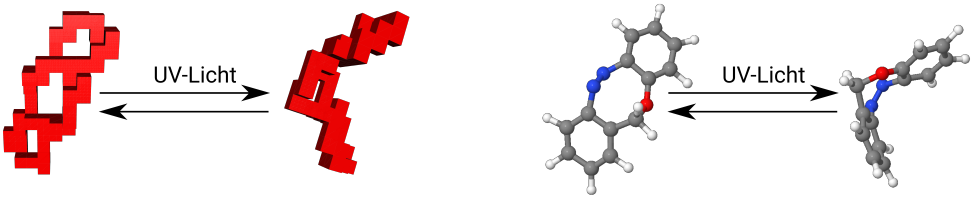


Abb. 3.9: Die resultierende gerasterte Geometrie von Edukt und Produkt einer Phototrajektorie von brAB-O (links) und die gewöhnliche Darstellung in einem Molekül-Viewer (rechts).

Die Voxel enthalten zunächst lediglich binäre Information über die Besetzung. Zur Auswertung werden verschiedene Attribute berechnet, welche im folgenden als *oct-Attribute* bezeichnet werden.

Die Gesamtordnung der Voxel eines Schnappschusses im Raum lässt sich direkt über die Kodierung der Voxel schwer erfassen und verarbeiten. Würde die Voxelposition in kartesischen Koordinaten verwendet, entstünde kaum ein echter Mehrwert durch diese Repräsentation. Bei Verwendung der kodierten Pfade im OctTree ist eine Überanpassung der ML-Algorithmen für die Auswertung an dieses gewählte Modellsystem groß; denn mit der veränderlichen Anzahl an besetzten Voxel wird ebenso die explizite Besetzung des OctTrees als Repräsentation eines Schnappschusses komplex und unflexibel. Zusätzlich erschweren explizit kodierte Voxel den Übertrag von strukturellen Merkmalen von einer Trajektorie auf andere, welche mit mehr oder weniger geringfügig veränderten Startbedinungen simuliert werden. Kurzum, es sind neue Attribute vorhanden, allerdings bringen diese lediglich andere Probleme und Unsicherheiten als die Ähnlichkeitsüberprüfungen in kartesischen oder internen Koordinaten und skalieren nicht mit der Anzahl an Atomen, sondern mit der gewählten Tiefe des OctTrees. Aus diesen Gründen bietet es sich an, aus den besetzten Voxeln eines Schnappschusses zusätzlich Superstrukturelemente zusammenzusetzen, welche eine vereinfachte übergeordnete Beschreibung erlauben. Diese können je nach Anwendungsfall und expliziter Auswertung einfache geometrische Objekte sein oder komplexe Strukturelemente, welche selbst aus einer größeren Anzahl an Voxeln bestehen. Damit dienen schließlich diese Superstrukturelemente als Analysegrundlage und ermöglichen sogar eine Skalierung der Größe, welche lediglich aus der Auflösung des OctTrees resultieren.

Ein Photoschalter mit 10-20 Atomen könnte direkt mit einem Protein verglichen werden, welches eine ähnliche strukturelle Änderung durchläuft – wenn auch auf einer anderen Zeit- und Größenskala. Diese grobe Umschreibung wäre etwa mit einem Sketch (oder Comic) vergleichbar, welcher die Grundlage für das Verständnis eines neuen naturwissenschaftlichen Phänomens darstellen soll. Dabei wären zunächst alle Details für einen Einstieg nicht unmittelbar hilfreich, stattdessen sollte zunächst die Versinnbildlichung eines Ereignisses im Vordergrund stehen.

Die einfachsten Superstrukturelemente sind die sogenannten Sticks, welche aus Seite an Seite angeordneten Voxeln bestehen. Dazu wird in dem fertig befüllten OctTree mit einem beliebigen, besetzten Voxel gestartet. In diesem wird in einer der drei Raumrichtungen begonnen, Seite an Seite benachbarte Voxel in positiver und negativer Richtung aufzuaddieren. Findet sich kein weiterer besetzter Voxel, wird der Vorgang beendet und der gebildete Stick extrahiert. Hierzu werden verwendete Voxel als unbesetzt markiert und ein Stickobjekt der entsprechenden Raumrichtung, x, y oder z, erstellt, welches die kodierten Voxel enthält. Dieser Vorgang wird solange fortgesetzt, bis der OctTree entleert ist und sich kein weiterer Stick erstellen lässt. Dieser Vorgang wird für die übrigen Raumrichtungen wiederholt. Es werden folglich für jede Raumrichtung Sticks erhalten, welche die kodierten Voxel, deren Anzahl, sowie Start- und End-Voxel enthalten. Aus diesen Strukturen lassen sich entsprechend größere voxelisierte Stukturen berechnen.

Zunächst werden aus diesen Sticks die sogenannten Pseudo-Sticks konstruiert, welche aus benachbarten Sticks einer Raumrichtung zusammengesetzt werden. Das Ziel dieses Ansatzes ist es, ein Maß für die Ordnung in jeder Raumrichtung zu erzeugen. Daher werden möglichst wenige und große Pseudo-Sticks berechnet. Die Nachbarschaft wird über die Start- und End-Voxel der jeweiligen Sticks bestimmt und ermöglicht die Berücksichtigung der Richtung. Zulässige Nachbarn sind seitenflächen-verknüpfte Voxel (in einer der verbleibenden zwei Raumrichtungen) und kanten- und eckenverknüpfte Voxel (Kapitel A). Die Position des Sticks, der den Start der Suche darstellt, bildet das Zentrum, um welches sich nur mit einem Voxel Abstand die möglich Nachbarn anordnen sollen. Folglich gibt es einen festen Bereich, der sich immer an der Position des ersten Sticks orientiert und in dem valide Nachbarn akzeptiert werden. Zum Beispiel wird bei der Kombination der Sticks in z-Richtung zu Pseudso-Sticks eine Abweichung in $\pm x$ und $\pm y$ um einen Voxel toleriert, wobei zusätzlich kombinierte Sticks echt benachbart sein müssen, d.h. seitenflächen-, kanten- oder ecken-verknüpft sein müssen. Dabei zeigt der OctTree einen weiteren Vorteil: Da lediglich die möglichen Nachbarvoxel überprüft werden müssen, ist dieses kombinatorische Problem in einer überschaubaren Zeit zu bewältigen. Außerdem ist in dieser Repräsentation zu je-

dem Zeitpunkt bekannt, welche Sticks gültige Nachbarn sind. Die Abbildung 3.10 zeigt in der 2D-Projektion exemplarisch Voxel, welche im ersten Schritt zu Sticks zusammengefasst und farblich markiert werden. Diese werden im Folgenden zu den sogenannten Pseudo-Sticks kombiniert.

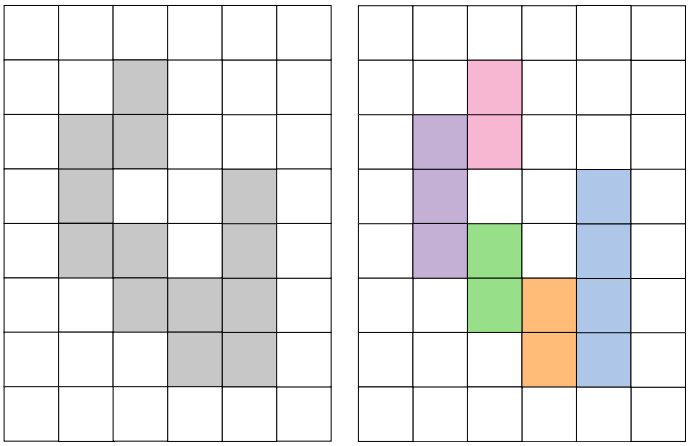


Abb. 3.10: In y-Richtung konstruierte Sticks; es ergibt sich aus den besetzten Voxeln links die Einteilung in Sticks rechts. Der blaue Stick mit vier Voxeln ist in diesem Beispiel der größte.

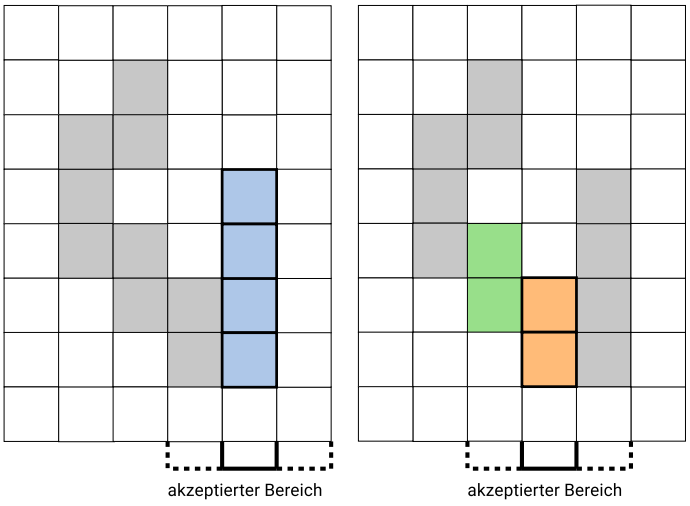


Abb. 3.11: In y-Richtung konstruierte Pseudo-Sticks, schwarz hervorgehoben sind die jeweiligen Sticks mit denen begonnen wird; daraus resultiert ein Bereich, in dem valide Nachbarn akzeptiert werden. Links: Der blaue Stick besitzt keine validen Nachbarn. Rechts: Der orangefarbene besitzt einen validen Nachbarn – den grünen. Daraus resultiert der fablich hervorgebhobene Pseudo-Stick.

In Abbildung 3.11 ist zu erkennen, dass der blaue Stick keine validen Nachbarn hat, da dieser zwar am unteren Ende einen benachbarten Stick besitzt, allerdings sind diese beiden Sticks parallel angeordnet, also die beiden Enden benachbart. Die Logik an dieser Stelle ist, dass ein Längenwachstum entlang der Koordinate bei der Kombination zweier Sticks erfolgen und die Fortsetzung des Sticks immer

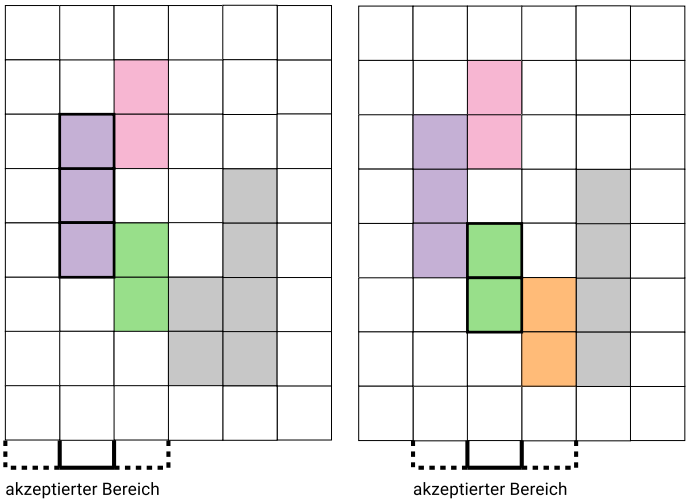


Abb. 3.12: In y-Richtung konstruierte Pseudo-Sticks, schwarz hervorgehoben sind die jeweiligen Sticks mit denen begonnen wird; daraus resultiert ein Bereich in dem valide Nachbarn akzeptiert werden. Links: Der Lilafarbene besitzt zwei valide Nachbarn und damit bilden die drei farblich hervorgehobenen Sticks einen Pseudo-Stick. Rechts: Der grüne Stick besitzt zwei valide Nachbarn – orange und lila – und kann ebenfalls um den rosafarbenen Stick ergänzt werden. Der rosafarbende Stick als Start vereint in einem Pseudo-Stick die selben farblich hervorgehobenen Sticks.

in der selben Richtung erfolgen muss. Dabei wird getrennt in positiver (oben) und negativer (unten) Richtung die Fortsetzung des Sticks ermittelt und bei Abschluss der Suche in einem Pseudo-Stick vereinigt. Bei der Fortsetzung nach unten, wäre wiederum lediglich die Nachbarn erlaubt, welche der in diesem Beispiel blaue Stick ebenfalls besitzt.[13] Damit ist der orange Stick kein valider Nachbar und beide werden nicht zu einem Pseudo-Stick kombiniert. Die Abbildung 3.12 zeigt, dass weitere benachbarte Sticks im akzeptieren Bereich hinzugefügt werden. Nach der Konstruktion aller Pseudo-Sticks wird als erstes der größte Pseudo-Stick extrahiert und mit ihm alle zugehörigen Sticks, welche ebenfalls aus den verbliebenen Pseudo-Sticks entfernt werden. Nach einer erneuten Evaluation kann der nächst größte extrahiert werden. Dies wird fortgesetzt, bis der OctTree vollständig entleert ist.[14] In diesem Fall liefert der grüne Stick als Basis für einen Pseudo-Stick in diesem Fall den größten Stick, welcher zunächst extrahiert wird. Gleiches ergäbe sich, würde der erste Stick der rosafarbene sein. Der verbliebene blaue Stick bildet

[13] Ausgenommen ist der Fall bei dem die benachbarten alle Voxel nur über Kanten oder Ecken verknüpft wären. Es ist an dieser Stelle nicht gewünscht, dass Sticks zum Pseudo-Stick hinzugefügt werden, welche in großen Teilen parallel zueinander verlaufen, Kapitel A. Die erhaltenen Ergebnisse der Pseudo-Sticks müssten in dem Fall eher den Namen ›Voxel-Haufen‹ bekommen und bei einer getrennten Suche in positiver (oben) und negativer (unten) Richtung würden Sticks eventuell doppelt erfasst.

[14] Im Extremfall besteht ein PseudoStick aus einem einzigen RealStick, welcher wiederum lediglich aus einem einzigen Voxel besteht.

einen eigenen Pseudo-Stick.[15]

Hierzu können in jedem Voxel relevante Informationen gespeichert werden. Dieses Vorgehen, bis hin zu komplexeren Objekten, ist der Auswertung von Lidar-Scanner-Daten entlehnt, welche selbstständig durch Segmentation des Raumes Objekte erkennen.[77,79,86] Die weitere Konstruktion wurde auf Grund der nicht gewissen Erfolgsaussichten dieser Arbeit vorerst zurückgestellt. Da die weitere Kombination der Objekte zu größeren Objekten rechen- und entwicklungsintensiv ist, soll vorerst mit im Folgenden berechneten *oct-Attributen* eine Überprüfung stattfinden. Die *oct-Attribute* sind, außer der absoluten Anzahl der besetzten Voxeln, `NumberOccupiedVoxel`, und der maximale Länge in Voxel, `maxLengthInVox`, für jede der drei Raumrichtungen berechnet worden. Somit ergeben sich insgesamt 38 Attribute, welche Verwendung finden und in Tabelle 3.3 aufgeführt sind. Dabei ist die Länge der Struktur in Voxeln enthalten, wobei die maximale Ausdehnung der Geometrie in dem aktuellen Zeitschritt in einem idealen parallelen OctTree-Gitter, Abbildung 2.18, mit minimierter Voxelanzahl gemeint ist.

Das `coreRegionPercentX` ist die Anzahl an besetzen Voxeln nach der Projektion aller Voxel auf die *yz*-Ebene geteilt durch die maximale Anzahl an Voxel der *yz*-Ebene. Nach der durchgeführten Ausrichtung zu Beginn jeder Trajektorie ist ein geringer Wert für z zu erwarten, hingegen wird der größte Wert für x erwartet. Das Attribut `UsedArea`**X** bezieht sich auf die möglichen besetzten Voxel entlang einer Koordinatenachse normiert auf die maximale Anzahl entlang dieser Achse. `UsedAreaX` ist die Projektion auf die x-Achse und somit der verwendete Wertebereich in x-Richtung. Zusätzlich werden Attribute aus den Sticks und Pseudo-Sticks berechnet. Die Attribute `BigPseudoPercent`**X** und `BigRealPercent`**X** geben an, wie viele aller besetzten Voxel prozentual in dem größten Pseudo-Stick bzw. größten einfachen Stick vereinigt werden. Die Attribute `PseudoStickAverage`**X** und `RealStickAverage`**X** geben entsprechend an, wie viele Voxel durchschnittlich in den Pseudo-Sticks und Sticks enthalten sind. `BigPseudoToLength`**X** stellt das Verhältnis des größten Pseudo-Sticks zu `maxLengthInVox` dar – wie groß der Anteil des längsten Pseudo-Sticks zur minimalen Anzahl an Voxel ist, welche die längste Ausdehnung des Moleküls beschreibt.

Die Sticks und die Pseudo-Sticks können als Maß für die Ordnung angesehen werden. Die Kombination der Parameter erlaubt eine Unterscheidung zwischen Strukturmerkmalen, ohne dass diese explizit implementiert werden müssen. Spezifischere Parameter erlauben eine bessere Beschreibung, können allerdings nicht flexibel eingesetzt werden und beschränken das Anwendungsgebiet. Die berech-

[15] Wobei an dieser Stelle bereits festgestellt wurde, dass sich kein valider Nachbar auffinden lässt.

Tab. 3.3: Die verwendeten Attibute eines OctTree-Schnappschusses, das hervorgehobene **X** zeigt an, dass dieses oct-Attribut in jeder der drei Raumrichtungen berechnet werden.

`NumberOccupiedVoxel`	Absoulute Anzahl an besetzten Voxeln (globale Variable, absolut)
`maxLengthInVox`	Maximale Ausdehnung der Struktur in theoretischen Voxeln (globale Variable, absolut)
`BigPseudoToLength`**X**	Verhältnis des größten PseudoSticks in X zu `maxLengthInVox` (relativ)
`BigPseudoPercent`**X**	Anzahl der Voxel im größten Pseudo-Stick geteilt durch `NumberOccupiedVoxel` (relativ)
`BigRealPercent`**X**	Anzahl der Voxel im größten Stick geteilt durch `NumberOccupiedVoxel` (relativ)
`coreRegionPercent`**X**	Anzahl an besetzen Voxeln nach der Projektion aller Voxel auf die *yz*-Ebene geteilt durch die möglichen Pixel der Projektionsebene (relativ)
`numberPseudoStick`**X**	Anzahl der Pseudo-Sticks in **X** (absolut)
`numberRealStick`**X**	Anzahl der Sticks (absolut)
`PseudoStickAverage`**X**	Durchschnittliche Voxelanzahl pro Pseudo-Stick (absolut)
`RealStickAverage`**X**	Durchschnittliche Voxelanzahl pro Stick (absolut)
`RealToPseudo`**X**	Durchschnittliche `numberRealStickX` pro `numberPseudoStickX` in X (absolut)
`SizeBiggestPseudo`**X**	Absolute Voxelanzahl des größten Pseudo-Sticks (absolut)
`SizeBiggestReal`**X**	Absolute Voxelanzahl des größten Pseudo-Sticks (absolut)
`UsedArea`**X**	Projektion auf die *x*-Achse (relativ)

neten Parameter sind korreliert, wobei ein wichtiger Grundsatz im ML ist, dass die Attribute voneinander unabhängig sein sollen und die Gemeinsamkeiten erst ermittelt werden sollen. Dies wird in weiten Teilen der Forschungsgemeinschaft allerdings ausgedehnt, und die Anwendungsfälle zeigen, dass dies nicht grundsätzlich zu schlechten Ergebnissen führt. Das entwickelte Modell der geometrischen Beschreibung eines Moleküls in einer Trajektorie ist eine starke Rasterung (OctTree). Diese ermöglicht eine geometrische Konstruktion von Superstrukturen und aus diesen wiederum die vorgestellten *oct-Attribute*, um direkt das geometrisch Veränderliche für das ML zu erzeugen. Dies ist der Ansatz ohne chemische oder physikalische Information. Ob dies gut funktioniert, wird sich prinzipiell erst mit einem finalen Gütekriterium herausstellen können, Kapitel 5 und Abschnitt 5.4.

3.4.2 Halbautomatischer Ansatz

Bevor ein vollautomatisches Schema, Kapitel 4, entwickelt werden konnte, stellte die Auswahl eines idealen Trainingssatzes einen wichtigen Teilschritt dieser Arbeit dar. Dazu wurden Clustertechniken zusammen mit einer visuellen Klassifikation in einem Programm vereinigt. Es wurde zufällig und nach Augenmaß *eine* reaktive Trajektorie von 100 des in Abschnitt 3.3 verwendeten Diamino-brAB ausgewählt. Mittels des EM, Abschnitt 2.7.1, wurden die Daten zu einer größeren Anzahl an Clustern zusammengefasst und diese mittels einer *Jmol*-Einbettung visualisiert. [87] Nach händischer Analyse der Individuen der einzelnen Cluster, wobei die volle Funktionalität des Molekül-Viewers zur Verfügung stand, können gewünschte Klassifizierungen bzw. Benennungen gewählt werden. Die vorliegenden 10 Cluster, welche der EM-Algorithmus gebildet hat, enthalten in verschiedenen Clustern ähnliche Geometrien. Diese können durch die Benennung der Cluster auf diesem Wege, unabhängig vom Abstand der Cluster im Attributsraum, zu einem Cluster vereinigt werden. Die Attribute, die während des Clusterns Anwendung finden, bleiben verborgen, damit diese keinen Einfluss auf die händische Klassifizierung haben. Durch die Einbettung eines etablierten Viewers kann in diesem eine Vermessung und falls gewünscht Optimierung[16] der Struktur erfolgen, wobei die Schnappschüsse der Trajektorie nicht verändert werden. Es soll eine möglichst uneingeschränkte Betrachtung aller Individuen eines Clusters ermöglicht werden, bevor eine Entscheidung gefällt werden muss.

Die auf diesem Wege klassifizierten Geometrien können als Trainingssatz für die eigentliche Klassifizierung weiterer Geometrien anderer Trajektorien dienen. Es wird ein robuster Klassifikationsbaum verwendet, welcher mittels des Trainings-

[16] Unter Verwendung von einfachen Kraftfeldmethoden.

satzes angelernt wird. Es zeigt sich, dass dieser nicht für die gesamte Schar[17] erfolgreich anwendbar ist. Dies ist darin begründet, dass die Trainingsdaten nicht vollständig sind, welches entweder zu einer Auslassung im nD-Raum führen kann oder zu einer Überanpassung des Algorithmus. Es gilt in jedem Fall, dass die zufällige Auswahl einer Trajektorie nicht zwingend alle reaktiven Instanzen enthält und diese durch den angelernten Klassifizierer als *unreaktiv* klassifiziert werden. Eine Verbesserung oder Verschlechterung der Ergebnisse kann unter der Verwendung einer anderen zufällig gewählten *reaktiven* Trajektorie erreicht werden. Eine Begründung kann nicht unmittelbar abgeleitet werden. Die einfache Betrachtung des komplexen Parameterraums ist schwer möglich, folglich lässt sich nicht mit hinreichender Sicherheit ein einziger Grund für die fehlerhafte Klassifikation ermitteln. Aus diesem Grund wird eine verbesserte Auswahl der Trainingsdaten in Kapitel 4 entwickelt und verwendet.

Die Verwendung des EM-Algorithmus aus WEKA ist durch die große Anzahl an benötigten Schritten bis zur Konvergenz langsam. Wie häufig im ML kann nicht angenommen werden, dass mehr Rechenzeit und ein komplexerer Ansatz zu besseren Ergebnissen führen muss. Mit einer kleinen Auswahl an Schnappschüssen, hier 150 Stück, ist der Arbeitsablauf vom Rechenaufwand her praktikabel, für eine ganze Trajektorie (hier >1600), ist die Performanz bereits stark eingeschränkt. Der EM ist nicht sensitiv gegenüber Ausreißern. Daher kommt es vor, dass die vorgeschlagenen Cluster nicht nach gewünschten Charakteristika unterteilt werden, da die Anzahl der Instanzen einen starken Einfluss auf die Clusterbildung hat. Der Algorithmus kann nicht zuverlässig aus den vorgegeben Attributen aus einer Trajektorie einen *reaktiven* und *unreaktiven* Cluster determinieren. Aus diesen Gründen wird dieser Algorithmus im weiteren Vorgehen vernachlässigt und im Folgenden durch einen HC-Algorithmus ersetzt. Ebenso bringen klassische Benchmarkansätze mit verschiedenen Clusteralgorithmen angewendet auf den Trainingssatz, oder weitere Klassifikationsalgorithmen an dieser Stelle, keine Verbesserung.

Ein wichtige Erkenntnis dieser Arbeit ist es, dass die Gruppierung von Geometrien bezüglich gemeinsamer Eigenschaften für den Menschen sehr schwierig ist. Aus diesem Grund lassen sich schwer Trainingssätze ermitteln, welche für leistungsstarke *überwachte* ML-Techniken benötigt werden.[18] Es kann festgehalten werden, dass der Fokus der Arbeit auf die Extraktion seltener Instanzen gelegt werden muss, ohne die restlichen Instanzen zu vernachlässigen. Die Konvertierung der

[17] Bestehend aus 100 Trajektorien.

[18] Ein Training mittels der in Abschnitt 3.1 vorgestellten Methoden ist prinzipiell möglich, verfehlt den gestellten Anspruch dieser Arbeit allerdings. Die Reproduktion der Ergebnisse auf einem Umweg (oct-Attribute und ML) böte keine generalisierbaren Erkenntnisse und wäre lediglich an die vorliegende Fragestellung (Quantenausbeuten) angepasst.

molekularen Geometrien in oct-Attribute ermöglicht mit einer festgelegten Anzahl an Parametern, Strukturen zu beschreiben und Analyseschemata zu entwickeln, welche unabhängig von der Anzahl der Atome und der strukturellen Komplexität sind. Die verwendeten oct-Attribute ermöglicht eine einfachere Darstellung von geometrischen Eigenschaften mittels Superstrukturen, welche eine eindeutige Separation innerhalb einer Trajetorie erlauben. Die Anwendung eines trajektorienübergreifenden Klassifikationsalgorithmus scheitert weniger an den gewählten Attributen als vielmehr an dem mangelhaften selektierten Trainingssatz. Dies lässt sich damit belegen, dass es möglich ist, die Ergebnisse mittels eines anderen Trainingssatzes zu verbessern, bei dem eben *eine* andere Trajektorie der vorliegenden 100 ausgewählt wird. Eine Begründung lässt sich zwar theoretisch formulieren, aber die visuelle Beurteilung der Trajektorien lässt keine eindeutige Tendenz erkennen.

3.5 Meta-Clustern

Das semi-automatische Erstellen eines Trainingssatzes soll ebenfalls automatisiert werden, wobei auch leistungsfähigere als die im letzten Abschnitt verwendeten ML-Algorithmen Anwendung finden. Dieser Ansatz mündet schließlich in einen hierarchischen Arbeitsablauf, welcher noch mehr als Clustern und Klassifizieren einschließt, Kapitel 4. Der beschriebene Weg beinhaltet vielversprechende Ansätze, welche auch negative Ergebnisse liefern, aber für die Entwicklung der gesamten Heuristik des resultierenden Programms notwendige Schritte darstellen.

Für einen automatischen Ansatz sollte idealerweise diejenige Trajektorie, welche als Trainingssatz Verwendung findet, möglichst viele verschiedene Geometrien enthalten, da nur die im Trainingssatz befindlichen Zustände im weiteren Prozess Beachtung finden können. Dies liegt vor allem daran, dass ein Klassifizierer angelernt werden muss, welcher nur in begrenztem Maße generalisiert werden kann, siehe Abschnitt 2.7.2. Zunächst wird ein wichtiger Zwischenschritt gezeigt, bei dem schließlich eine Schar von Trajektorien selbst geclustert wird und nicht nur die einzelnen Schnappschüsse in einer Trajektorie.

Zunächst werden die Instanzen jeder einzelnen Trajektorie mittels eines HC-Algorithmus geclustert. Dieser ist, wie bereits erwähnt, deterministisch und benötigt daher nur einen einzigen Turnus und ist generell effizienter, da nach einfachen Regeln, Gleichungen (2.4) bis (2.9), eine Agglomeration stattfindet. Für jede Trajektorie werden die 20 größten Fusionshöhen, im Dendrogramm in Abbildung 2.9 anschaulich als y-Achse zu erkennen, als Attribute gespeichert. Damit ergibt sich ein Datensatz für die Trajektorien-Schar, der aus Trajektorien als In-

stanzen besteht. Jede einzelne Trajektorie besitzt 20 Attribute, die Fusionshöhen des HC der Schnappschüsse. Anschließend wird die Trajektorien-Schar in dieser Repräsentation erneut mit Hilfe eines HC geclustert. Aus diesem Grund wurde die Bezeichnung: *meta-Clustern* gewählt, da die Attribute der Instanzen selbst aus dem Ergebnis des Clusterns entnommen wurden.

Die Geometrien jedes Schnappschusses werden wie in Abschnitt 3.4.1 beschrieben in *oct-Attribute* übersetzt. Anschließend wird jede Trajektorie einzeln mittels des HC-Algorithmus mit der Methode UPGMA, Gleichung (2.6), unter der Verwendung der Manhattan-Distanz, Gleichung (2.2), geclustert.[19] Für eine Beurteilung bezüglich der Reaktivität werden die 20 größten Fusionshöhen als Satz von Attributen für eine Trajektorie verwendet und im Folgenden zwei Cluster erstellt: *reaktiv* und *unreaktiv*. Die Trajektorienschar wird wiederum mittels HC mit der Methode UPGMA, Gleichung (2.6), unter der Verwendung der euklidischen Distanz, Gleichung (2.1), geclustert. Das Ergebnis wird mit dem etabliertem Verfahren, der Berechnung der Reaktivität mit Hilfe der beiden Diederwinkel der Brücken, Abschnitt 3.1 und den dafür vorgestellten Parametern für das Zeit- und das Toleranzintervall, verglichen.

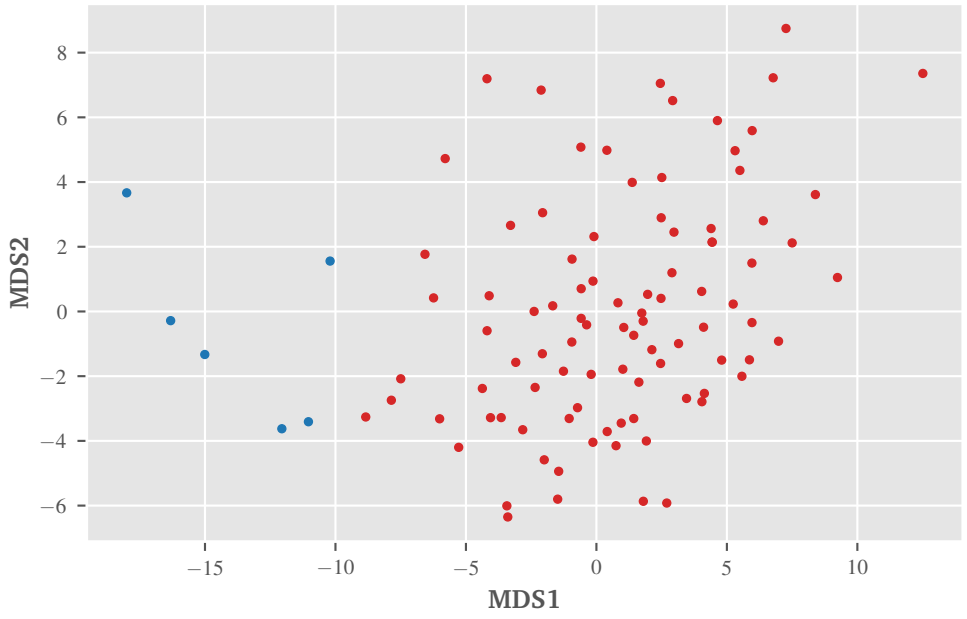


Abb. 3.13: Der MDS-Graph wird auf Basis der 20 größten Verschmelzungshöhen berechnet. Eine Separation in *reaktiv* und *unreaktiv* ist durch HC möglich, die wenigen *reaktiven* Trajektorien sind bei MDS1<-10 im Graphen zu finden und blau eingefärbt.

[19]Die Chronologie dieser Arbeit ist an dieser Stelle gebrochen, eine Argumentation für die Verwendete Metrik findet sich erst in Abschnitt 4.5. Zu diesem Zeitpunkt nehmen wir an, dass diese Wahl gut überlegt ist und die Geometrien sich gut separieren lassen.

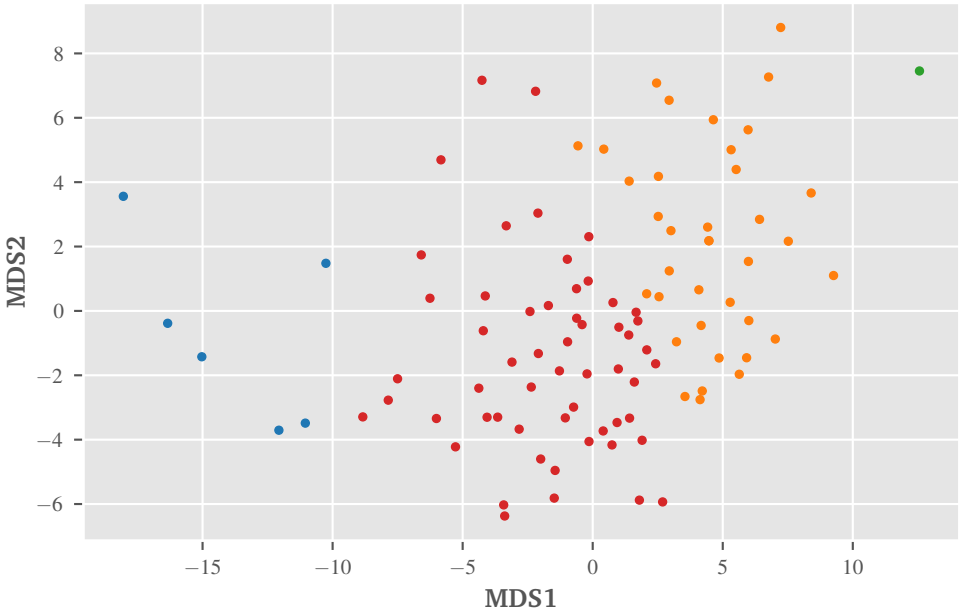


Abb. 3.14: Der MDS-Graph wird auf Basis der 20 größten Verschmelzungshöhen berechnet. Bei der Zerteilung der Trajektorien-Schar in vier Cluster wird der rote Cluster aus Abbildung 3.13 in zwei gleich große und einen Ausreißer bei MDS1>10 aufgeteilt. Der blaue Cluster mit den *unreaktiven* Trajektorien bleibt erhalten.

Die Unterteilung in zwei Cluster verläuft erfolgreich und es werden die *unreaktiven* Trajektorien eindeutig in einem Cluster vereinigt. Die als *reaktiv* gekennzeichneten Trajektorien erweisen sich erwartungsgemäß nicht alle eindeutig als reaktiv bei visueller Betrachtung und zerfallen auch bei relativ großer Fusionshöhe in Subcluster, Abbildung 3.14. Die Separation der *unreaktiven* Trajektorien ist ein wichtiger Schritt, da dies zeigt, dass ein Cluster mit nur einer einzigen Art von Trajektorien gebildet wird. Dieser Eindruck wird in einem MDS-Graphen, Abschnitt 2.6, in Abbildung 3.13 bestätigt. In diesem sind die 20 Attribute der Trajektorien in zwei Dimensionen dargestellt und jede Instanz nach ihrer Clusterzugehörigkeit eingefärbt. Blau sind die *unreaktiven* und rot die *reaktiven* eingefärbt. Auch in diesem Bild ist eine deutliche Separation zu erkennen, wobei der blaue Cluster mit nur sechs Instanzen klein ausfällt.[20] Die Verwendung dieser wenigen Trajektorien als Trainingssatz ist jedoch nicht sinnvoll, da keine *reaktiven* Instanzen enthalten sind. Ein Blick auf den zweiten Cluster ist lohnender, da auf jeden Fall *reaktive* Instanzen anzutreffen sind. Die Zerlegung in vier Cluster zeigt, dass eine Instanz isoliert bei MDS1>10 entsteht und ansonsten der große Cluster in zwei etwa gleich große Cluster zerfällt.[21] Dies ist in Abbildung 3.14 deutlich zu

[20] Zur Erinnerung: Die Achsen des MDS-Graphen sind nicht die Grundlage des HC

[21] *Zerfallen* ist beim Agglomerieren eher ein 'Nicht-weiter-Agglomerieren'. Folglich besteht der größere Cluster aus zwei etwa gleich großen Subclustern und einer einzelnen Instanz.

erkennen, gleichzeitig ist der Cluster der *unreaktiven* Trajektorien stabil und würde erst bei kleineren Fusionshöhen zerteilt. Es zeigt sich die Tendenz, dass die Cluster entlang einer einzigen Dimension aufgereiht werden können. Die Parameter, die zur Klassifikation der einzelnen Cluster benötigt werden, können über einen einfachen Klassifikationsbaum extrahiert werden. Diese Parameter erzeugen eine scheinbar harte Grenze, bei der eine Geometrie reaktiv ist, und diese wird, da die oct-Attribute nicht bijektiv sind, nicht direkt für einen einzigen Satz an internen Koordinaten zutreffen, wie die klaren Grenzen der Diederwinkel bei der klassischen Auswertung.

Die automatische Analyse einer einzelnen Trajektorie ist auf diesem Weg nicht unmittelbar möglich, da im Schritt des meta-Clusterns keine expliziten Informationen über die Geometrien der Schnappschüsse der einzelnen Trajektorien zur Anwendung kommen. Außerdem wird die Gesamtheit aller Geometrien einer Trajektorie unabhängig von der zeitlichen Information beurteilt und im ersten Schritt geclustert. An dieser Stelle kann eine einzige Instanz (je nach HC-Ansatz) eine Trajekorie von *unreaktiv* zu *reaktiv* aufwerten und umgekehrt und auf diesem Wege die Attribute der jeweiligen Trajektorie für das darauffolgende meta-Clustern verändern. Im Prinzip ist an dieser Stelle nicht unmittelbar klar, warum eine Trajektorie letztendlich in dem roten Cluster der *reaktiven* Trajektorien aus Abbildung 3.13 gelandet ist. Die hier gewählten oct-Attribute machen es nicht einfacher, allerdings kann festgehalten werden, dass es möglich ist, einen Satz an Trajektorien auf diesem Weg in Subgruppen aufzuspalten.

An dieser Stelle könnten die Trajektorien durch das Clustern kategorisiert und auf das Verhalten der internen Koordinaten hin untersucht werden. Schließlich könnte eine einfache Entscheidungsregel aufgestellt werden oder mittels ML-Techniken beispielsweise mit Entscheidungsbäumen vollautomatisch entschieden werden. Ein Blick auf die Reaktionsrichtung zeigte, dass für eine Trajektorien-Schar von *cis* nach *trans* und von *trans* nach *cis* nicht dieselben Clustermethoden und Metriken für die beiden Clusterdurchläufe zum Erfolg führten. Zusätzlich zeigt sich, dass die durchgeführte Ausrichtung aller Geometrien einer Trajektorie anhand des ersten Schnappschusses bei einer Trajektorie mit einer anfänglichen *cis*-Geometrie zu einer Verschlechterung führte. Dies ist bei näherer Betrachtung anhand des Durchmessers der Struktur zu erklären. Die größte Ausdehnung ist nicht zuverlässig entlang der Brücke zu finden, wie bei einer *trans*-Geometrie, sondern kann sich je nach Kompaktheit ändern.

Die fertige Trennung unter bestimmen Rahmenbedingungen in zwei Gruppen von Trajektorien an sich ist ein gutes Ergebnis, allerdings fehlt die detaillierte Analyse einzelner Trajektorien und die Übertragbarkeit auf weitere Systeme. Aus

diesen Gründen wurde die Erkenntnis, dass es möglich ist, zwei Cluster aus der Trajektorien-Schar zu bilden, in Abschnitt 4.1 weiter verallgemeinert.

Kapitel 4

Vollautomatischer Ansatz

Der vorgestellte Ansatz in Abschnitt 3.3, bei dem direkt die internen Koordinaten als Attribute für die ML-Techniken verwendet wurden, zeigte, dass dies aus Gründen der Metrik und der hohen Dimensionalität nicht praktikabel ist. Die entwickelten oct-Attribute in Abschnitt 3.4.1 können diese Schwierigkeiten beseitigen und als Attribute für ML-Methoden zur Anwendung kommen. Es zeigte sich in Abschnitt 3.4.2, dass das Training eines Klassifizierungsbaums gelingen kann und die Übertragung zwischen einzelnen Trajektorien möglich ist, aber stark von dem gewählten Trainingssatz abhängt. Das hier vorgestellte meta-Clustern, Abschnitt 3.5 zeigt, dass die geometrischen Eigenschaften kodiert in Superstrukturelementen eine Unterscheidung der Trajektorien prinzipiell ermöglichen. Aus diesen Ergebnissen wird im Folgenden eine Generalisierung für die Auswahl eines Trainingssatzes abgeleitet werden und ein Bewertungskriterium vorgestellt werden. Die Trajektorien werden in der gerasterten OctTree-Darstellung und den resultierenden Superstrukturelementen verwendet. Die Abbildung 4.1 zeigt den schematischen Ablauf der automatischen Auswertung, wie er schließlich zur Anwendung kommt und in den folgenden Abschnitten vorgestellt wird. Nach der erfolgreichen vollständigen Klassifizierung der einzelnen Schnappschüsse jeder Trajektorie wird eine abschließende Bewertung der Trajektorien mit Hilfe von Zeitreihenanalyse durchgeführt, welche an dieser Stelle durch die Umrechnung in ein statisches Modellsystem erfolgt.

Nach der Vorstellung des Gesamtkonzepts sollen im Folgenden die einzelnen veränderlichen Parameter und der gesamte Arbeitszyklus diskutiert werden. Da es sich um korrelierte Parameter handelt, wird in einem sinnvollen Rahmen die Wahl eingegrenzt und erklärt.

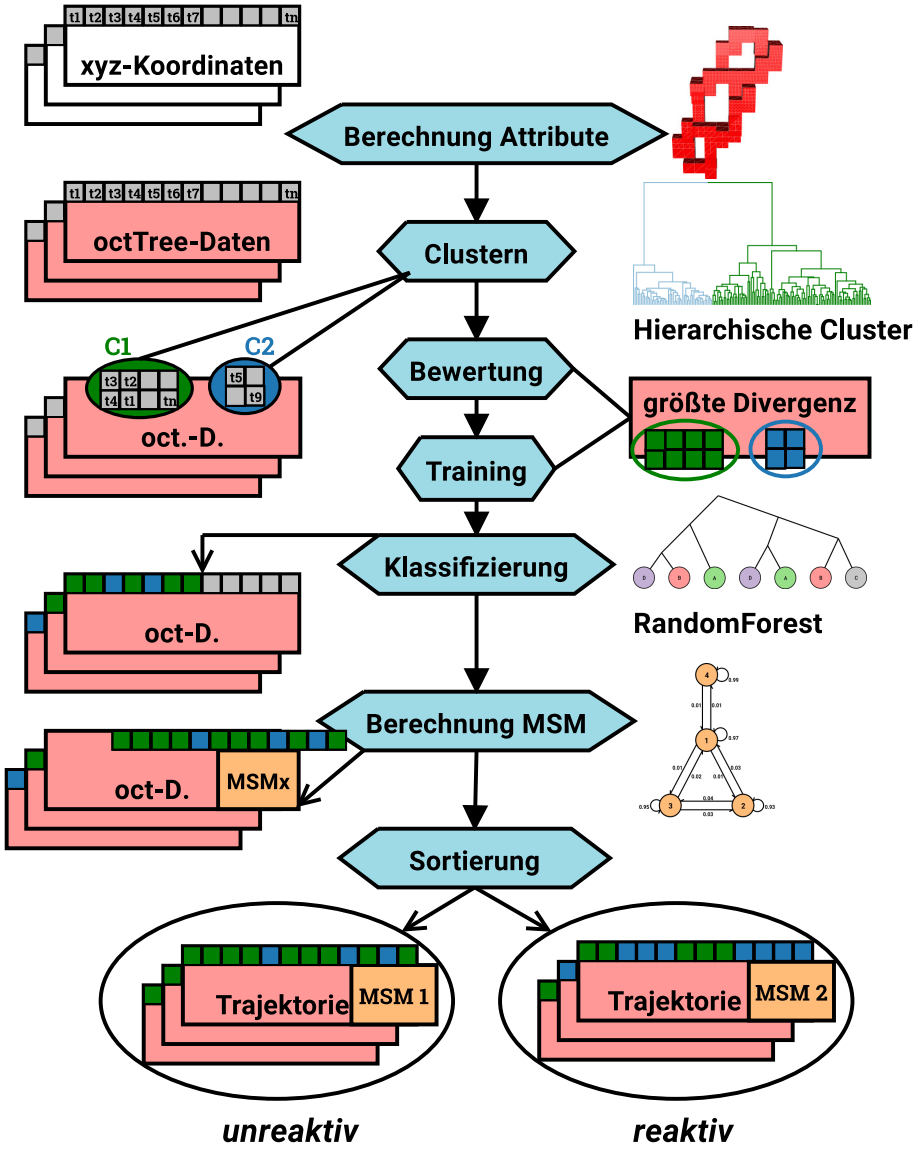


Abb. 4.1: Schematische Darstellung des Arbeitsablaufs des vollautomatischen Ansatzes: Die Trajektorien werden in octTree-Paramter überführt, die Trajektorien werden geclustert und bewertet. Die am besten bewertete Trajektorie wird geclustert, mittles PCCA zusammengefasst und als Trainingssatz für den Klassifikationsbaum verwendet. Der angelernte Algorithmus klassifiziert jede Instanz jeder Trajektorie, welche wiederum in ein MSM umgerechnet wird und als Bewertungsgrundlage dient.

4.1 Bewertung

Die 20 Fusionshöhen, welche in Abschnitt 3.5 zur Anwendung kommen, unterscheiden sich signifikant für *reaktive* und *unreaktive* Trajektorien. Da für das Clustern der Trajektorien-Schar eine euklidische Distanz verwendet wird und die Agglomeration durch gewichtete Methoden durchgeführt wird, kann aus dem vorliegenden Graphen in Abbildung 3.13 abgeleitet werden, dass sich diese auch in niedrigerer Dimension separieren lassen. Zu diesem Zweck wird eine einfache Summe der zwanzig größten Höhen gebildet und diese über die zwei Dimensionen des MDS-Graphen aus Abbildung 3.13 aufgetragen. Der resultierende Graph ist in Abbildung 4.2 gezeigt.

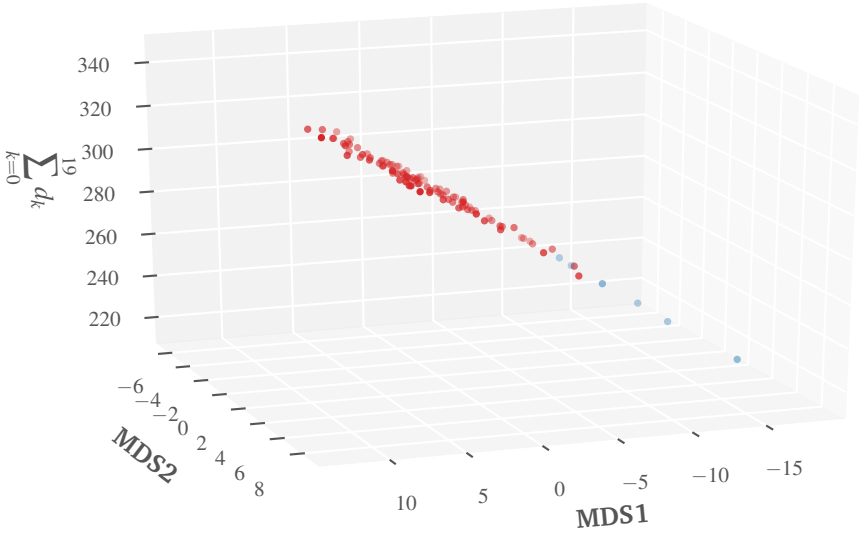


Abb. 4.2: Die Auftragung der Summe der 20 größten Fusionshöhen des HC über den zwei Dimensionen des MDS-Graphen.

Es ist zu erkennen, dass ein linearer Zusammenhang[1] existiert. Die Trajektorien, welche mit einer niedrigen resultierenden Summe aufgetragen sind, enthalten eine geringe Vielfalt an Strukturen und sind daher als Trainingssatz ungeeignet. Diese sind die bereits in blau markierten und als *unreaktiv* identifizierten Trajektorien. Die Trajektorien mit maximaler Summe enthalten eine größere Vielfalt und ermöglichen am ehesten eine Klassifikation aller vorhandenen Geometrien. Aus diesem Ergebnis lässt sich die Annahme aufstellen, dass eine größere Summe stets eine größere Wahrscheinlichkeit beinhaltet *reaktiv* zu sein, da die Strukturen ei-

[1] Die Perspektive erzeugt einen kompakteren Eindruck in der MDS1-MDS2-Ebene, als es in Wirklichkeit der Fall ist, siehe Abbildung 3.13.

ne größere Divergenz und einen größeren Ausschnitt aller möglichen Geometrien enthalten. Eine direkte Klassifikation über die angegebene Summe benötigt einen festen Wert, bei dem in *un-/reaktiv* unterteilt wird. Dieser Wert ist ähnlich wie bei dem Versuch, die Trajektorien über den RMSD-Wert zu klassifizieren, nicht global wählbar und im Zweifelsfall vollkommen beliebig. Daher lässt sich kein absoluter oder prozentualer Wert festlegen, bei dem eine Trajektorie als *reaktiv* bewertet wird. Allerdings kann davon ausgegangen werden, dass in einer Trajektorie mit einer großen Summe der zwanzig größten Fusionshöhen mehr voneinander verschiedene Strukturen enthalten sind als in einer Trajektorie mit einer kleinen Summe. Dabei kann es durchaus sein, dass die Trajektorie mit der maximalen Summe Strukturen enthält, welche nicht unmittelbar mit der untersuchten Reaktivität zusammenhängen. Es ist für einen guten Trainingssatz wichtig, dass möglichst viele verschiedene Strukturen enthalten sind, damit diese im weiteren Verlauf auch klassifiziert werden können.[2] Aus diesen Gründen werden für eine Bewertung der Trajektorien die zwanzig größten Fusionshöhen der Cluster im HC summiert und als Bewertungskriterium verwendet. Die Trajektorie mit der größten Summe bildet im Folgenden den Trainingssatz.

4.2 Klassifikation der Schnappschüsse

Bis zu dieser Stelle findet die zeitliche Information keine Verwendung, allein die strukturellen Eigenschaften der Schnappschüsse in den einzelnen Trajektorien werden verwendet. Ziel ist es, mit Hilfe eines überwachten angelernten Klassifikationsalgorithmus, welcher alle möglichen Geometrien eines molekularen Systems kennt, jeden Schnappschuss jeder Trajektorie einordnen zu können. Aus diesem Grund wird ein möglichst divergenter Trainingssatz benötigt, damit jede mögliche strukturelle Eigenschaft eines Systems zu jedem Zeitpunkt bestimmt werden kann. Für einen beliebigen Schnappschuss muss eine eindeutige Klassifikation trajektorienübergreifend anwendbar sein.

Das in Abschnitt 3.5 hergeleitete Bewertungskriterium wird auf einen Satz von Trajektorien angewendet. Die am besten bewertete Trajektorie wird, ganz analog zu dem in Abschnitt 2.7.1 und Abschnitt 2.7.2 vorgestellten Ablauf, als Trainingssatz verwendet. Dazu werden die Clusterzuordnungen der Schnappschüsse der Trajektorie als *Klassen*-Attribut übernommen. Anschließend wird ein Klassifikationsbaum mit diesem Trainingssatz trainiert. Die Verwendung eines einfachen Baums kann, wie in Abschnitt 2.7.2 gezeigt, unter Umständen ungenau sein, des-

[2] Im idealen Fall alle erdenklichen.

halb wurde – vornehmlich aus theoretischen Überlegungen – der einfache Baum durch einen *Random-Forest*-Algorithmus mit 1000 Klassifikationsbäumen ersetzt. Es ist zu beobachten, dass die Klassifizierung der einzelnen Instanzen entlang der Zeitachse mit dem *Random-Forest*-Algorithmus etwas homogener ist als mit einem einfachen Baum. Da der Rechenaufwand relativ gering ist und um spätere Fehler, eingebracht durch einen unzulänglichen Klassifikator, zu minimieren, wird der *Random-Forest*-Algorithmus weiterverwendet.

Die Annahme, dass die Ergebnisse der Klassifikation verbessert werden können, wenn der Trainingssatz vergrößert wird, wurde überprüft, indem die Strukturen der zweit- und dritthöchsten bewerteten Trajektorie hinzugefügt wurden. Es zeigten sich eher negative als positive Veränderungen, da zum einen die Instanzen, an denen die Cluster separiert werden, nicht identisch sind und zum anderen in Einzelfällen beobachtet werden kann, dass die Clusterbenennung vertauscht ist. All dies algorithmisch zu beachten und abzufangen ist aufwendig und erfordert Wissen über das analysierte System und kann die Übertragung auf beliebige Systeme erschweren. Die weitere Analyse als Zeitreihe in Abschnitt 4.3 ermöglicht die Verwendung eines relativ kleinen Trainingssatzes.

Der trainierte Klassifikator ist in der Lage, jede einzelne Instanz jeder Zeitreihe zu klassifizieren und somit eine kodierte Zeitreihe zu erzeugen. Diese kann über einen ähnlichen Ansatz wie im klassischen Auswertungsverfahren ausgewertet werden und in eine feste Aussage über die Reaktivität übersetzt werden. Die Parameter sind in diesem Fall ebenso frei wählbar wie in Abschnitt 3.1 vorgestellt und in Abschnitt 3.3 angewendet.

4.3 Klassifikation der Trajektorien als Zeitreihe

Im Weiteren wird die Klassifikation der Geometrien mit der Analyse der Zeitreihe verbunden. Dies geschieht, um nicht allein über die Distanz, welche aus der Metrik und dem Distanzmaß des HC entsteht, eine Aussage treffen zu können, sondern auch den zeitlichen Kontext zu berücksichtigen. Das Ziel ist die automatische Auswertung von MD. Bei der MD ist bekannt, dass zwei zeitlich aufeinander folgende Schnappschüsse wirklich benachbart sind.[3] Dieses Wissen kann im Folgenden ausgenutzt werden und neben der Ähnlichkeitsüberprüfung in den oct-Attributen als zusätzliche Datengrundlage für die ML-Techniken dienen.

Die Abbildung 4.3 zeigt die Clusterverteilung über die Zeit aufgetragen. Mit

[3]Die Speicherung der Schnappschüsse erfolgt in einem sinnvollen Abstand und die Bewegungsgleichungen sollten erfolgreich gelöst werden, andernfalls wäre keine einzige Auswertung wirklich sinnvoll.

gleicher Methode und Metrik des HC wie in Abschnitt 3.5 ist in Blau die Clusterzuordnung mit einer Gesamtanzahl von 20 Clustern und in Grün von 10 Clustern dargestellt; zur besseren Darstellung wurden die Cluster des blauen Graphen um 0.2 entlang der y-Achse verschoben. Die Clusterbenennung erfolgt von 0 bis 19 bzw. 9. Die Nähe der einzelnen Cluster zueinander kann nicht anhand der Benennung der Cluster, aufgetragen als y-Achse, abgelesen werden.

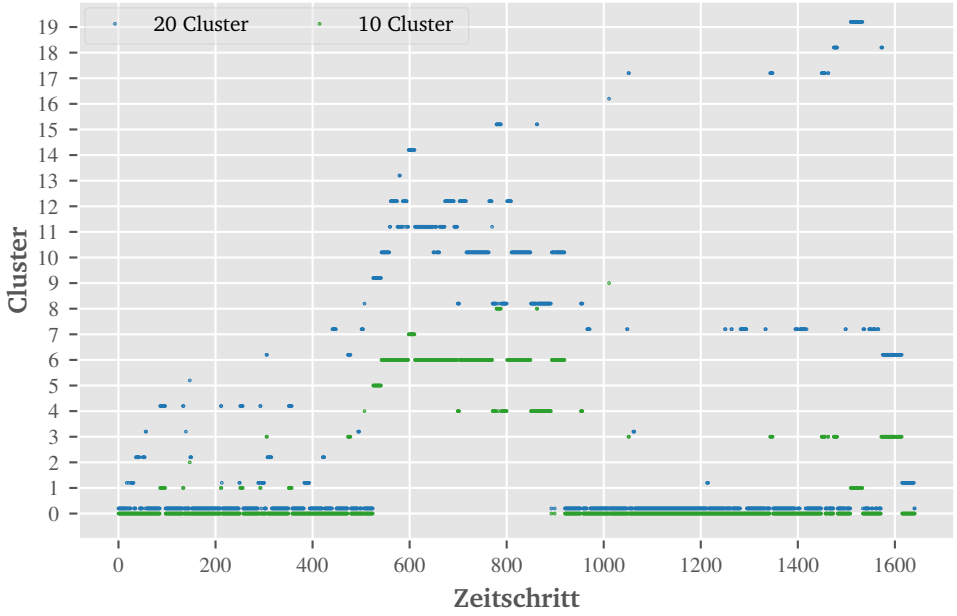


Abb. 4.3: Die Clusterzuordnung des HC einer Trajektorie mit einer Clusteranzahl von 20 in Blau und Clusteranzahl von 10 in Grün, aufgetragen über die Zeitschritte. Erkennbar sind Bereiche, in denen Cluster vom Übergang von 20 auf 10 Cluster zusammengefasst werden; die Distanz der Cluster entlang der y-Achse zueinander ist zufällig.

Die Verwendung der internen Koordinaten als Unterscheidungskriterium der klassischen Auswertung wird an dieser Stelle gezeigt, um das Cluster,n aufgetragen entlang der Zeitachse, validieren zu können. In Abbildung 4.4 sind die Diederwinkel CNNC und CCOC der Trajektorie aufgetragen. Es ist gut zu erkennen, dass die Separation durch das Clustern im Bereich von 550 bis 930 Schnappschüssen gut zu dem Verhalten der Diederwinkel passt, welche in diesem Intervall Reaktivität zeigen.

Auch im RMSD-Graphen, Abbildung 4.5, können die Cluster von 0 bis 9 farblich eingezeichnet werden und zeigen, dass in jenem Bereich ein erhöhter RMSD-Wert beobachtet werden kann. Es kann bestätigt werden, dass die vorgenommene Separation mittels Clustern erfolgreich stattgefunden hat.

Die vorliegenden Cluster können folglich als Trainingssatz für einen Klassifikationsalgorithmus dienen. Durch die Verwendung der zeitlichen Information kann

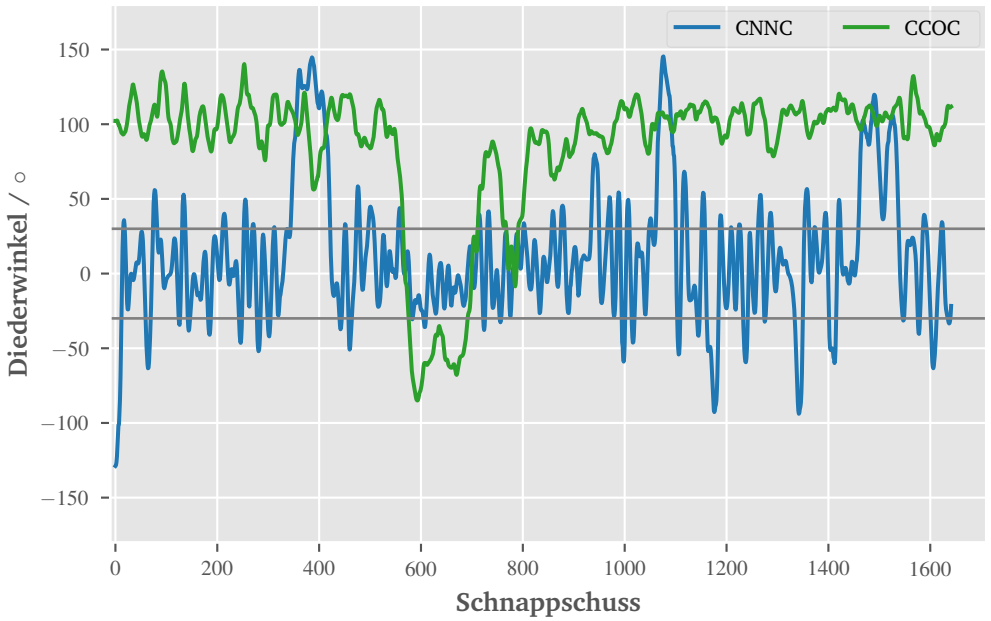


Abb. 4.4: Aufgtragung der CNNC- und CCOC-Diederwinkel der geclusterten Trajektorie aus Abbildung 4.3.

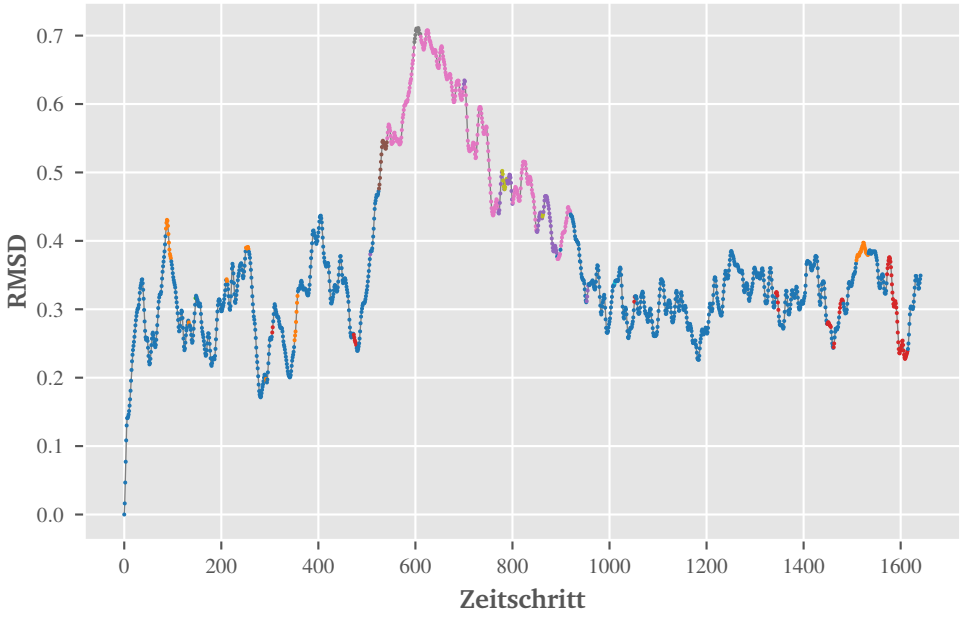


Abb. 4.5: Die Clusterzuordnung des HC mit einer Clusteranzahl von 10, wie in Abbildung 4.3 in Grün dargestellt, hier nochmal mit dem RMSD-Graphen nach Gleichung (2.3) verglichen. Die Cluster 0 bis 9 sind farblich kodiert.

eine weitere Verbesserung erreicht werden oder zunächst ein externes, attributsfernes Gütekriterium eingeführt werden. Dies ist sinnvoll, da die verwendeten Daten die vorgestellten oct-Attribute sind. Folglich treten Fehler bereits in der Datenvorbereitung auf und können – auf Grund der hohen Dimensionalität – nicht direkt ermittelt werden. Abzuwägen, welche Qualität der verwendete ML-Ablauf hat, ist aus dem Verfahren selbst schwierig, da messbare oder berechenbare Größen häufig bereits während des Verfahrens zur Anwendung kommen. An dieser Stelle ist die Überprüfung, ob ein Clusteralgorithmus ideal ist, ohne externe Kriterien schwierig. Hinzukommt, dass es keine Garantie gibt, dass eine überprüfte intrinsische Güte des HC eine Aussage über die Anwendbarkeit auf das vorliegende System erlaubt oder ob dieses Verfahren generalisierbar ist. Aus diesem Grund ist der vorgestellte Vergleich mit chemischen Eigenschaften (RMSD und Diederwinkel der Brücken), welche keine Verwendung in den ML-Methoden finden, wichtig. Bis hierhin konnte die Korrelation der Clusterung der verwendeten HC-Methode mit chemischen Eigenschaften gezeigt werden. Diese zeitliche Information kann im Folgenden die Auswertung und Klassifikation der Trajektorien als Fortführung der automatischen Auswertung dienen.

Bei der großen Gesamtanzahl an Instanzen ist zu erwarten, dass bei einer größeren Anzahl an Cluster und einem begrenztem Trainingssatz – einer einzigen Trajektorie – die Klassifikation von einzelnen Instanzen vereinzelt falsch sein wird.[4] Mittels einer einfachen Fensterfunktion, welche die zeitlich benachbarten Instanzen betrachtet, kann dieses Phänomen eliminiert werden. Da in der Simulation die Distanz limitiert ist, die das System innerhalb eines Zeitschrittes zurücklegen kann, ist dieser Ansatz durchaus vertretbar. Zusätzlich enthalten die resultierenden oct-Attribute eine Vergröberung der geometrischen Eigenschaften.

Im Folgenden wird die Abhängigkeit der Auswertung von der gewählten Metrik und der HC-Methode reduziert. Hierzu wird die ausgewählte Trainingstrajektorie, wie in Abbildung 4.3 in Blau gezeigt, zunächst beispielsweise in 20 Cluster unterteilt. Die Darstellung als kodierte Zeitreihe mit 20 unterschiedlichen Clustern von 0 bis 19 wird als Grundlage für die Berechnung eines MSM, Abschnitt 2.2, verwendet. Das erhaltene MSM wird mittels PCCA, wie in den Abbildungen 2.2 und 2.3 gezeigt, auf die gewünschte Anzahl metastabiler Zustände reduziert. Somit wird die Abhängigkeit von den rein geometrischen oct-Attributen reduziert, da für die Abstandsberechnung nicht allein das Abstandsmaß des HC mit seiner Metrik verwendet wird, sondern die zeitliche Anordnung Beachtung findet. Auf die-

[4] Durch die bereits angesprochene Rotation des Systems, gut zu erkennen im RMSD-Graphen in Abbildung 2.6 und die Ausrichtung zu Beginn der Auswertung, Abschnitt 3.4.1 ist ein zu enger Korridor bei der Klassifikation ebenfalls ungünstig.

sem Weg kann bei einer ungünstigen Bewertung der räumlichen Anordnung der Schnappschüsse im Hyperraum eine Korrektur über die zeitliche Anordnung erfolgen. Denn die Cluster, welche als metastabiler Zustand zusammengefasst werden, müssen nicht zwingend auch bei der Fusion im HC – also im Hyperraum der oct-Attribute – durchgeführt werden. Die erwähnte Filterfunktion wird an dieser Stelle nicht benötigt, da ein weitreichender und sinnvoller Ansatz gefunden wurde, um dem Problem der Missklassifikation einzelner Schnappschüsse zu begegnen.

Die metastabilen Zustände werden im nächsten Schritt als Klassen-Attribut für die einzelnen Instanzen und als Trainingssatz verwendet. Mit Hilfe dieses Trainingssatzes wird ein überwachter ML-Algorithmus, der Random-Forest-Algorithmus, Abschnitt 2.7.2, angelernt. Dieser dient zur Klassifizierung aller Schnappschüsse in der gesamten Trajektorien-Schar und ermöglicht auf diesem Weg einen Übertrag von einer beispielhaften Trajektorie auf sämtliche vorliegende oder noch nachträglich berechnete.[5]

Die auf diesem Weg klassifizierten Trajektorien werden wiederum jeweils einzeln in ein MSM umgewandelt. Dieses einfache Model mit seinen Übergangswahrscheinlichkeiten kann zur Klassifizierung der einzelnen Trajektorien dienen. Damit können direkt aus einer MD-Trajektorien-Schar wichtige Zwischenstufen gefunden werden und somit die Aufklärung des Mechanismus erleichtert werden – ohne eine direkte Annahme aufstellen zu müssen.
Dies benötigt abschließend zusätzlich eine Methodik, welche die explizite Klassifikation ermöglicht. Die mechanistische Aufklärung eines konkreten Reaktionsverlaufs ist nicht Gegenstand dieser Arbeit.

Im Folgenden werden die veränderlichen Parameter der vorgestellten Implementation anhand des brAB-O, Abschnitt 3.2, vorgestellt und für diesen Fall eine Methodik entwickelt, um *reaktive* und *unreaktive* Trajektorien zu klassifizieren. Der zusammengefasste Arbeitsablauf ist in Abbildung 4.1 gezeigt. Die Auswertung mittels der entwickelten Programme wird explizit in Kapitel 5 für das brAB-O erörtert. Die Behauptung der Generalisierbarbeit des Klassifikators wird in Abschnitt 5.4 anhand eines systemübergreifenden Klassifikators mit dem Molekül brAB-O überprüft. Außerdem wird gezeigt, dass weitere andersartige Systeme mit einem vollständig unterschiedlichen Reaktionstyp auf dem gleichen Weg ausgewertet werden können.

Nach der Vorstellung des Gesamtkonzepts sollen im Folgenden die einzelnen veränderlichen Parameter der Methoden und Konzepte und der gesamte Arbeitszyklus diskutiert werden. Da es sich um korrelierte Parameter handelt, wird in

[5]Dazu muss selbstverständlich entschieden werden, ob die vorliegende Trajektorie des Trainingssatzes eine ausreichend große Divergenz besitzt.

einem sinnvollen Rahmen die Wahl eingrenzt und erklärt.

4.4 OctTree

Die Zielsetzung, dass keine externen Parameter für eine vollständige Analyse gesetzt werden sollen, führt dazu, dass die festgelegte Größe des kubischen Raumes automatisch aus der größten Ausdehnung ermittelt werden kann. Diese wird als kleinste Kantenlänge der acht Voxel der ersten Lage gesetzt[6]. Die Übertragbarkeit zwischen einzelnen Trajektorien ist bei relativ geringen Abweichungen in den strukturellen Eigenschaften gut. Die zugrundeliegende Physik der Simulationen verlangt die Unveränderlichkeit der Erhaltungsgrößen, woraus bei korrekter Implementation und numerischer Integration eine Translations- und Rotationsinvarianz der Trajektorien resultiert. Daher kann angenommen werden, dass dieser definierte Raum ausreicht und während der Analyse einer Trajektorie nicht neu berechnet werden muss. In Einzelfällen zeigt sich, dass diese Annahme bei den vorliegenden Trajektorien nicht zutrifft und eine Rotation bzw. Translation zu erkennen ist. Aus diesem Grund reicht die auf diesem Weg ermittelte maximale Raumgröße nicht aus und es wird ein fester Wert von 12 Å für alle Trajektorien gewählt. In der aktuellen Implementation des OctTrees und der Algorithmen für das Einfüllen der molekulares Strukturen und der Berechnung der Pseudo-Sticks ist ein dynamisches Wachstum des Raums nicht möglich. Für die Simulationen, welche mit dem kompakten *cis*-Isomer gestartet werden, muss in jedem Fall mit einer festen Boxgröße gearbeitet werden, da die erreichbare Ausdehnung nicht zu Beginn bekannt ist. Die Rotation zu Beginn der Ausrichtung eines Moleküls, Abschnitt 3.4.1, stellt eine zusätzliche Fehlerquelle dar und wird in Abschnitt 4.4.3 kurz diskutiert. Die Form des *cis*-Isomers ist näherungsweise kugelsymmetrisch, und es werden nicht für jede Trajektorie die selben zwei Atome als längste Ausdehnung identifiziert. Damit liefert die Methodik, welche eigentlich die Geometrien ausrichten und vergleichbar machen soll, letztendlich zufällige Resultate. Eine sinnvollere, aber ebenso generalisierbare Ausrichtung sollte zukünftig entwickelt werden und wird in dieser Arbeit nicht vorgestellt.

Die Auflösung des OctTrees kann über die Anzahl an Lagen eingestellt werden. Dabei ist der Rechenaufwand für die Berechnungen der oct-Parameter rechenintensiver, der gesamte nachgelagerte ML-Teil dieses Projektes wird in seiner Performanz nicht beeinflusst. Es werden vier bis sieben Lagen[7] überprüft. Ziel ist es, eine

[6] Die doppelte Ausdehnung des Moleküls in jeder Raumrichtung.

[7] Gemeint ist der Input Parameter des Java-Programms, die reale Anzahl an Lagen ist um eins reduziert.

Balance zwischen Leistung und Genauigkeit zu erreichen.

Es zeigt sich, dass eine feine Auflösung die Ergebnisse nicht weiter verbessert, sondern einen gegenteiligen Effekt hat, da die große Anzahl an Voxeln und wenigen expliziten Strukturmerkmalen die Wahrscheinlichkeit erhöht, dass dieselben oct-Attribute bei unterschiedlichen Geometrien erzeugt werden. Zusätzlich steigt der Rechenaufwand exponentiell mit der Anzahl der Lagen im OctTree und damit mit der Auflösung an.[8] Eine geringe Anzahl an Lagen wird angestrebt, da diese weniger rechenintensiv sind und einen besseren Vergleich unterschiedlicher molekularer Strukturen untereinander ermöglichen. Ein weiterer Punkt, der überprüft wurde, ist, dass die oct-Attribute in der vorliegenden Form teilweise nicht normiert sind. Daraus resultiert bei unterschiedlich vielen Lagen im Raum eine Verschiebung der einzelnen Gewichtungsfaktoren bei der Berechnung der Abstände mittels der Metrik. Dies führt dazu, dass je nach Metrik und Methode einzelne Attribute einen so großen Einfluss haben, dass auch ein kleinerer Unterparameterraum Verwendung finden könnte, welches durch die Anzahl an Parametern mit zu geringem Gewicht verschleiert würde. Dies muss nicht unmittelbar schlecht sein – eine exakte Beschreibung des System mit wenigen Parametern ist einer komplexeren Beschreibung vorzuziehen – allerdings müssten die Gewichtungsfaktoren so gewählt werden, dass zumindest alle Attribute den gleichen Einfluss in der Metrik besitzen. Dies ist auch wichtig, da nicht nur Photoisomerisationen automatisch ausgewertet werden sollen, sondern ebenso Reaktionen mit unterschiedlichen Charakteristika. Die maximale Ausdehnung der Geometrie ist ein solcher Parameter mit einem großen Gewicht. Bei den vorliegenden $trans \rightarrow cis$-Übergängen ist eine deutliche Längenänderung zu beobachten, es ist daher möglich, allein über diesen einen Parameter eine Klassifikation durchzuführen. Ein allgemeingültiger Übertrag auf beliebige unbekannte molekulare Systeme sollte im besten Fall diese Längenänderung registrieren können – daher ist dieser Parameter eingeführt worden – sollte aber kein Übergewicht gegenüber weiteren geometrischen Veränderungen bekommen.

4.4.1 Globale Optimierung der Gewichtungsfaktoren

Zunächst werden visuell alle Schnappschüsse mehrerer Trajektorien des brAB-O klassifiziert. Diese Trajektorien sind *reaktiv* und haben nach dem vorgestellten Bewertungskriterium eine hohe Bewertung; es werden drei Klassen erstellt: *cis*,*trans* und *intermediär*, da es einen Bereich gibt, bei dem eine Entscheidung schwierig ist.

[8] Es ergeben sich mit der Anzahl an n Lagen entsprechend 8^n Voxel, welche anschließend mit dem Raytracing befüllt, zu Sticks und schließlich kombinatorisch zu Pseudo-Sticks vereinigt werden.

Das Ziel ist die globale Optimierung der Gewichtungsfaktoren der oct-Attribute. Zu diesem Zweck wird die Fitnessfunktion als Überlappung von händischer Klassifikation und automatischer Clusterzuweisung der einzelnen Schnappschüsse definiert. Für die Wahl einer Trajektorie als Trainingssatz lassen sich erwartungsgemäß die Ergebnisse des Cluster-Algorithmus[9] stark verbessern. Der Übertrag von einem global optimierten Satz von oct-Attributen einer Trajektorie auf die anderen Trajektorien verbessert das Clusterverhalten in der Regel nicht und ist sogar schlechter. Ein angelernter Klassifikator, nach dem vorgestelltem Schema, kann ebenfalls keine verbesserten Ergebnisse erzielen und erzeugt häufig sehr schlechte Ergebnisse bei der Klassifizierung einzelner Schnappschüsse. Erwartungsgemäß findet in diesem Fall eine Überanpassung statt, da die Gewichtungsfaktoren spezialisiert für einen einzigen Trainingssatz angepasst wurden. Es wird deutlich, dass eine Überanpassung der Daten vorgenommen wurde, welche durch das Herabsetzen des Abbruchkriteriums verhindert werden kann oder durch die Vergrößerung des Trainingssatzes. Die Vereinigung mehrerer Trajektorien zu einem größeren Trainingssatz ergibt ebenso wenig eine Verbesserung wie ein herabgesetztes Abbruchkriterium für die globale Optimierung. Die optimierten Gewichtungsfaktoren zeigen keine deutlichen Tendenzen, welche im Folgenden weiter verwendet werden können, sondern fallen von Ansatz zu Ansatz unterschiedlich aus. Daher ist weder eine globale Optimierung der Gewichtungsfaktoren während des vorgestellten ML-Schemas, noch eine einmalige globale Optimierung für die oct-Attribute im allgemeinen sinnvoll.

Sicherlich kann eine Optimierung gelingen, aber dieser Ansatz müsste viel genereller durchgeführt werden und daher mit einem viel größeren Ansatz durchgeführt werden. Dabei müssten verschiedene Systeme mit bekannter geometrischer Bewertung und im idealen Fall ebenfalls sämtliche Parameter der nachfolgend vorgestellten Methoden mit einbezogen werden.[10]

Im Folgenden werden die unterschiedlichen Gewichtungen innerhalb der oct-Attribute berücksichtigt und die Auflösung des octTrees als Grundlage verwendet.

[9] Ein größere Auswahl von Metrik und HC-Algorithmen wird unabhängig voneinander getestet.

[10] Eine globale Optimierung der Parameter – mal eben mit einem in z.B. R vorhandenen globalen Optimierer – liefert erwartungsgemäß nicht die idealen Ergebnisse. Allerdings lag an dieser Stelle mehr die Hoffnung darauf, eine Tendenz erkennen zu können, welche anschließend genutzt werden könnte.

4.4.2 *Ad hoc* Normierung der oct-Attribute

Ein Ansatz, bei dem die einzelnen oct-Attribute mittels der Tiefe im OctTree normiert werden, ist zum einen nachvollziehbarer und damit einfacher zu vertreten und zum anderen letztendlich effektiver. Dazu werden die möglichen global maximalen Werte für jeden octParameter berechnet[11] und als Norm, Gleichung (4.1), verwendet. Dreidimensional ist der Normierungsfaktor für die Anzahl an Voxel, `NumberOccupiedVoxel`, für die Projektion auf eine Seitenfläche, `coreRegionPercent`**x**, ergibt sich ein zweidimensionaler und für Parameter entlang einer Raumrichtung ein eindimensionaler Normierungsfaktor.

$$\text{NormOctParameter} = \frac{\text{octParameter}}{\text{Dimensionalität}^{\text{Tiefe}}} \tag{4.1}$$

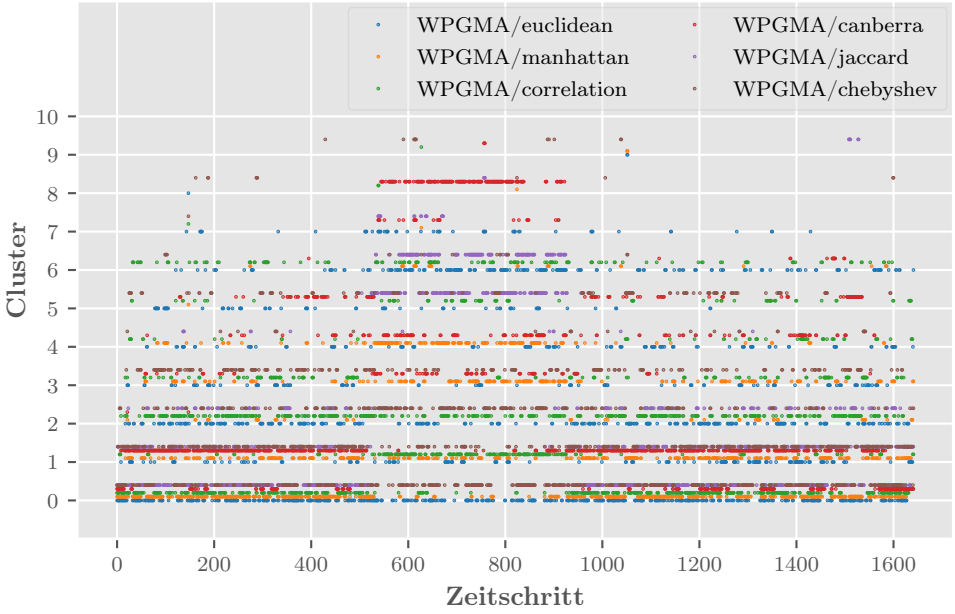


Abb. 4.6: Clusterzuordnung des HC einer Trajektorie mit einer Clusteranzahl von 10 unter der Verwendung von WPGMA, Gleichung (2.7) und verschiedenen Metriken, Abschnitt 2.4. Die oct-Attribute werden wie in Tabelle 3.3 verwendet. Es können in allen sechs Kombinationen fast durchgängig alle Clusterzuordnungen beobachtet werden. Im Bereich von 550 bis 900 kann eine Separation erahnt werden.

Es zeigt sich, dass mit den normierten Parametern mit einer größeren Anzahl an Kombinationen aus Metrik und Methode eine erfolgreiche Separation der Geometrien erreicht werden kann. Hierzu ist die Clusterzuordnung des HC einer Trajetorie mit einer Clusteranzahl von 10 unter Verwendung oct-Attribute in Abbil-

[11] Die Parameter *numberRealSticks* und *numberPseudoSticks* ermöglichten keine sinnvolle Normierung und wurden deshalb für den normierten Datensatz weggelassen, es ergeben sich damit 32 Parameter.

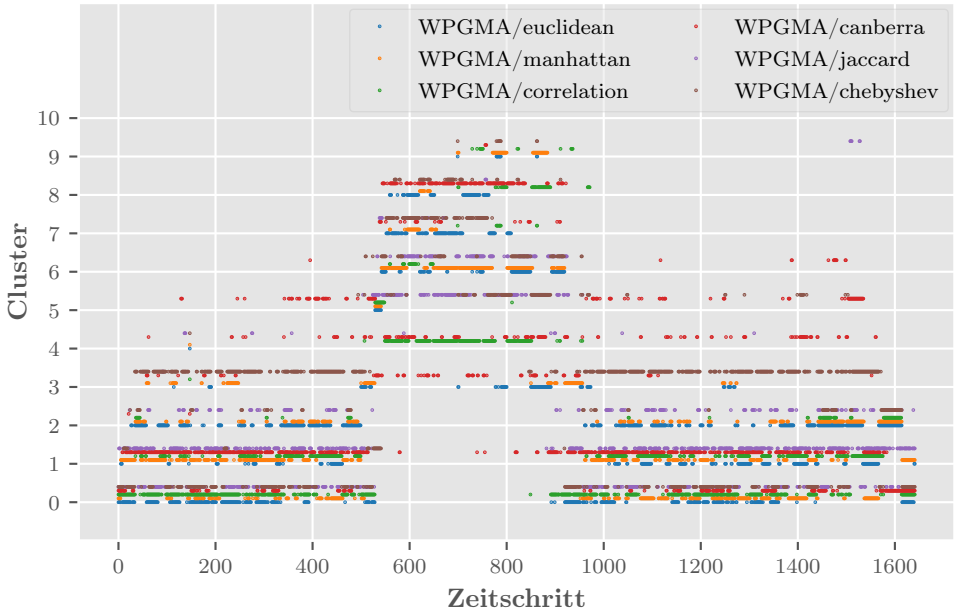


Abb. 4.7: Clusterzuordnung des HC einer Trajektorie mit einer Clusteranzahl von 10 unter der Verwendung von WPGMA, Gleichung (2.7) und verschiedenen Metriken, Abschnitt 2.4. Die oct-Attribute aus Tabelle 3.3 werden mit Hilfe von Gleichung (4.1) normiert. Es kann in allen sechs Kombinationen eine deutliche Separation im Gegensatz zu Abbildung 4.6 im Bereich von 550 bis 900 beobachtet werden.

dung 4.6, und unter Verwendung der normierten *oct-Attribute* in Abbildung 4.7 über die Zeitschritte aufgetragen. Es werden verschiedene Kombinationen aus der HC-Methode WPGMA, Gleichung (2.7), und Metrik verwendet. Es ist zu erkennen, dass in beiden Fällen eine mehr oder weniger gute Separation der Instanzen im Bereich von 550 bis 900 vom Rest der Trajektorie erreicht werden kann. Unter Verwendung der Normierung in Abbildung 4.7 ist zu erkennen, dass die Streuung der Clusterzuordnung abnimmt und sich im wesentlichen auf die drei Kerngebiete 0 bis 550, 550 bis 900 und 900 bis >1600 konzentriert. Die visuelle Überprüfung zeigt, dass es sich in der zeitlichen Abfolge um eine *trans*-Trajektorie handelt, welche zu *cis* isomerisiert und anschließend zurück nach *trans* wechselt. Am Schluss (1550) deutet sich ein weiterer unvollständiger Übergang an, welcher sofort zurück schaltet. Damit deckt sich die Beobachtung mit den Ergebnissen des HC, und die Veränderung von Abbildung 4.6 zu Abbildung 4.7 unter Verwendung der normierten oct-Attribute stellt eine Verbesserung dar. Die verschiedene HC-Methoden zusammen mit verschiedenen Metriken werden in Abschnitt 4.5 weiter diskutiert. Aus dieser erreichten Veränderung folgt, dass die direkte Abhängigkeit von den gewählten Methoden reduziert und damit die Robustheit erhöht werden kann.

4.4.3 Ausrichtung

Die Ausrichtung der Geometrie, welche in Abschnitt 3.4.1 vorgestellt wird, zeigt für eine Auswertung der *trans* $\rightleftharpoons$ *cis* Trajektorien-Schar eine Verbesserung, für die Gegenrichtung ergibt sich der gegenteilige Effekt. Der erste Schritt bei Ausrichtung des ersten Schnappschusses einer Trajektorie ist die Bestimmung der maximalen Ausdehnung. Diese ist für das *cis*-Isomer nicht in jeder Trajektorie für das gleiche Paar an Atomen zu finden. Die Begründung dafür ist die fast kugelsymmetrische Anordnung der Atome um den Schwerpunkt. Daher verändert sich die Anordnung des ganzen Moleküls in Abhängigkeit des Atompaars mit dem größten Abstand. Die nachfolgen Schritte sind von diesem ersten Schritt abhängig und es resultiert für jede Trajektorie unter Umständen eine andere Ausrichtung. Aus demselben Grund ist eine Neuausrichtung zu definierten Zeitschritten auf diesem Weg nicht praktikabel. Ansätze, durch chemische Intuition und Vorwissen eine Ausrichtung zu erzwingen, werden, da dieses Wissen nach eigener Zielsetzung aus der Auswertung herausgefiltert werden soll, nicht weiter verfolgt.[12] Die Auswahl der Startbedingungen für jede Trajektorie aus einer Grundzustandstrajektorie erlaubt die Annahme, dass diese Strukturen eine ähnliche räumliche Anordnung zueinander haben und in diesem Fall als aneinander ausgerichtet angesehen werden können. Zumindest ist zu beobachten, dass die Auswertung von einer Trajektorien-Schar mit *cis*-Isomer als Startgeometrie schlechter ausgewertet werden können bei aktiver Ausrichtung. Daher scheint die aus der Grundzustandstrajektorie resultierende Ausrichtung besser zu sein als der hier gezeigte Ansatz.

4.5 Metrik und Methode des HC

Wie im Folgenden gezeigt, ist der rein geometrische Ansatz nicht für jede Trajektorien-Schar erfolgreich, wenn die Parameter des HC und das Bewertungsverfahren konstant gehalten werden. Die Ergebnisse können für einzelne Datensätze mittels der Veränderung des Distanzmaßes oder sogar des HC-Prozederes verbessert werden, allerdings würde an dieser Stelle wiederum eine Art meta-Optimierung eingeführt werden, welche nur bei einem erwartbaren Ergebnis überhaupt durchführbar wäre.

Die Auswahl der HC-Methode ist eng mit der Wahl der Metrik verknüpft und wird deshalb jeweils in Kombinationen überprüft. Zusätzlich sind Methoden für

[12] Die Minimierung des RMSD-Werts – ein immer wieder vorgeschlagenes Instrument – würde beispielsweise für das in Abschnitt 5.4 vorgestellte System keine verbesserten Ergebnisse liefern, da es sich um einen Bindungsbruch in einem verhältnismäßig großem Molekül handelt.

eine einzige Metrik entwickelt worden oder in der verwendeten Software implementiert.[13] Zunächst werden die Kombinationen eliminiert, welche keine sinnvolle Separation der verschiedenen Strukturen ermöglichten. Dies wird mittels Vergleich der visuellen Bewertung und der fertigen Zeitreihen-Klassifikation der als Trainingssatz ausgewählten Trajektorien durchgeführt. Die Beurteilung der dafür verantwortlichen Bewertung ist im nachfolgenden Abschnitt 4.6 gezeigt. Der Graph wurde bereits in Abbildung 4.7 für das gewichtete HC, WPGMA, für die Clusteranzahl 10 gezeigt. Ein direkter Vergleich mit dem HC-Verfahren UPGMA, Gleichung (2.6), in Abbildung 4.8 zeigt, dass dieser Algorithmus die Fluktuation der Clusterzuordnung entlang der Zeitachse reduziert werden kann. Positiv ist anzumerken, dass unabhängig von der Metrik die Separation im Bereich von 550 bis 900 zuverlässig durchgeführt wird.

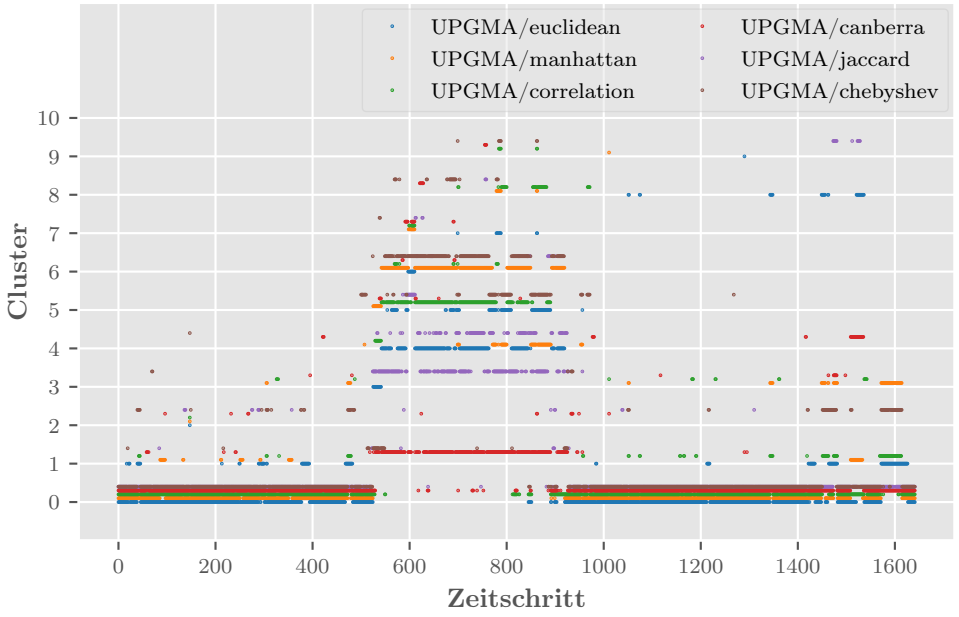


Abb. 4.8: Die Clusterzugehörigkeit der Instanzen einer Trajektorie. Es kann in allen sechs Kombinationen eine deutliche Separation im Bereich von 550 bis 900 beobachtet werden.

Diese Überprüfung wurde für eine größere Anzahl an Trajektorien durchgeführt und UPGMA zeigte die beste Überlappung zwischen den resultierenden Clustern und dem visuellen Eindruck. In erster Iteration ist dies scheinbar unabhängig von der Metrik, allerdings zeigt sich bei näherer Betrachtung, dass die Manhattan-Distanz robuster ist und sich im weiteren Verlauf als vorteilhaft herausstellt. Zu beobachten ist, dass die Manhattan-Distanz gute Ergebnisse mit mehreren HC-Techniken liefert und im fertigen Programm mit der Bewertungslogik zuverlässi-

[13] In diesem Fall beispielsweise ist für UPGMC und WPGMC, Gleichungen (2.8) und (2.9), lediglich die euklidische Distanz implementiert.

ger ist. Ob dies im Detail an den gewählten (normierten) oct-Attributen liegt oder angepasst für die vorliegenden Phototrajektorien gilt, lässt sich nicht abschließend beantworten. Eventuell ist durch diese Metrik eine etwas günstigerer Separation im Attributs-Hyperraum möglich, was sich günstig auf das berechnete Resultat auswirkt.

4.6 Bewertung der Trajektorien

Die Bewertung der einzelnen Trajektorien basierend auf der Metrik und HC-Methode wird über die Summe der Fusionshöhen des Dendrogramms durchgeführt. Die Trainingstrajektorie sollte eine mögliche große Divergenz bezüglich der vorhandenen geometrischen Strukturen aufweisen. Daher werden identische Instanzen in jeder einzelnen Trajektorie entfernt, bevor die Bewertung durchgeführt wird. Andernfalls würden, vor allem bei gewichteten Methoden, häufig auftretende Instanzen die wenigen ebenfalls wichtigen verdecken.

Für die Summe kann die Anzahl der summierten Fusionshöhen variiert werden.

$$\sum_{k=1}^{n} d_k, \text{ mit } d_1 > d_2 > ... > d_n \tag{4.2}$$

Es werden Werte für $n = 1$, 5, 10, 15 und 20 überprüft. Die Untersuchung zeigt, dass mit 1, 5 oder 10 Fusionshöhen nicht zuverlässig eine geeignete Trajektorie für das Training ausgewählt wird. Eine Verwendung aller Fusionshöhen ist nicht zweckmäßig, da durch die Abhängigkeit von der Anzahl der Instanzen die Übertragbarkeit und Vergleichbarkeit von unterschiedlichen Trajektorien nicht mehr gegeben wäre.

4.7 Clusteranzahl-Trainingstrajektorie

Die Clusteranzahl in der Trainingstrajektorie, bevor das MSM berechnet wird, kann unterschiedlich eingestellt werden. Somit kann die Auflösung bezüglich der geclusterten Geometrien erhöht werden, welche in einem nächsten Schritt mittels PCCA wiederum zusammengefasst werden. Für die Berechnung eines MSM muss eine ausreichend große Stichprobe vorliegen, weshalb eine unbegrenzt detaillierte Auflösung nicht möglich ist. Für das vorliegende System sind keine hundert Zustände möglich, da die Aussagekraft bei einer Trajektorie mit weniger als 2000 Zeitschritten begrenzt ist. Aus diesem Grund werden 20 Cluster gewählt

und diese auf die gewünschte Anzahl an Zuständen[14] im MSM durch den PCCA-Algorithmus reduziert. Das MSM wird über einfache Zählung der Abfolge der erzeugten Cluster in der Zeitreihe erstellt. Für den PCCA-Algorithmus kann die gewünschte Clusteranzahl festgelegt werden, welches über das Ellenbogenkriterium, Abbildung 2.10, erfolgen kann, dies wiederum folgt direkt aus dem verwendeten Abstandsmaß des HC-Algorithmus. Dies ist aus den bereits angesprochenen Gründen sinnvoller als direkt aus den Schnappschüssen einer Trajektorie lediglich die Clusteranzahl des Ellenbobenkriteriums zu erzeugen. Dieses Kriterium wurde für die brAB-O Trajektorien-Schar nicht verwendet und auf lediglich zwei Zustände, *reaktiv* und *unreaktiv*, beschränkt. Für die Aufklärung eines Mechanismus ist eine feinere Auflösung sinnvoll und für einen ersten Überblick bei einem unbekannten System kann die vorgeschlagene Anzahl an Clustern direkt Verwendung finden. Das Filtern und einfache Kategorisieren einer ganzen Trajektorienschar bezüglich relevanter Instanzen oder ganzen Trajektorien kann bei der Beurteilung von unbekannten Systemen bereits eine große Erleichterung sein. Obwohl es für den direkten Vergleich in dieser Arbeit keine Anwendung findet, ist dies ein wichtiger Vorschlag für eine Clusteranzahl und in dieser Art und Weise bereits implementiert.

4.8 Klassifikatoroptimierung

Der einfache Klassifikationsbaum wird aus theoretischen Überlegungen verworfen, Abschnitt 2.7.2. Der stabilere Random-Forest-Algorithmus wird weitgehend mit den Standardparametern verwendet, jedoch wird die Anzahl an verwendeten Klassifikationsbäumen bis zur Stabilisierung der Ergebnisse erhöht. Als Gütekriterium wurde die Schwankungsbreite der schließlich ermittelten Quantenausbeuten der Trajektorien-Scharen des brAB-O verwendet. Bei 100 Klassifikationsbäumen wird für die in Kapitel 5 betrachteten Trajektorien eine Schwankungsbreite von 50% beobachtet. Für 1000 Klassifikationsbäume wird eine Schwankungsbreite von 4% erreicht, dies ist in Anbetracht der Tatsache, dass dazu >1600 Schnappschüsse in 100 Trajektorien klassifiziert wurden, ein akzeptables Ergebnis. Durch die Optimierung des Klassifikators kann die Güte der Klassifikation für eine einzelne Instanz erhöht werden und somit die Abhängigkeit von einer nachgelagerten Filterfunktion minimiert werden.

[14] Für Kapitel 5 werden exakt 2 Zustände benötigt.

4.9 Das MSM einzelner Trajektorien

Das MSM, welches nach erfolgreicher Klassifizierung aller Schnappschüsse aller Trajektorien für jede Trajektorie einzeln berechnet werden kann, kann wiederum mit einer Filterfunktion oder Fensterfunktion ausgestattet werden, welche eine Glättung der Ergebnisse ermöglicht.

Es kann auf diesem Wege ein MSM für jede einzelne Trajektorie erstellt werden und als Bewertungsgrundlage dienen. Im Fall der Phototrajektorien werden direkt die Übergangswahrscheinlichkeitsmatrizen mit Fensterfunktion berechnet und als Basis für eine komplexere Beurteilung des Systems verwendet. Dieses Fenster der Fensterfunktion wird so eingestellt, dass drei zeitlich aufeinanderfolgende Schnappschüsse identisch klassifiziert werden, um gezählt zu werden. Über diese Funktion kann verhindert werden, dass ein einziger falsch klassifizierter Schnappschuss, welcher bei 99 Trajektorien und jeweils mehr als 1600 Schnappschüssen erwartet werden kann, die gesamte Auswertung beeinflusst.

Kapitel 5

Anwendung

Das vorgestellte automatische Gesamtkonzept wird im folgenden speziell für die Photoisomerisierung unter Kraft, Abschnitt 5.1, vorgestellt. Dabei werden Besonderheiten und Vorteile des in dieser Arbeit verwendeten Ansatzes in Abschnitt 5.3 weiter erläutert. Anschließend wird überprüft, ob die Behauptung der Übertragbarkeit des angelernten Klassifikationsalgorithmus, Abschnitt 5.4, zutreffend ist. Schließlich wird die automatische Auswertung auf ein größeres molekulares System mit einer anderen Reaktion angewendet, Abschnitt 5.5.

5.1 Auswertung des brAB-O unter Krafteinwirkung

Wie in Abschnitt 3.2 beschrieben, wird im Folgenden das System brAB-O, Abbildung 3.7 mit angewendeter Kraft von 70 bis 350 pN als Datengrundlage verwendet. Anhand dieser Trajektorien kann der erfolgreiche Einsatz des Gesamtkonzepts, welches in Abbildung 4.1 schematisch gezeigt ist, illustriert werden.

Die vorliegenden Trajektorien-Scharen unter Krafteinwirkung zeigen, dass der automatische Ansatz funktioniert und die Abnahme der Quantenausbeuten bei steigender entgegenwirkender Kraft zu beobachten ist. Die automatische Auswertung wurde mit den folgenden bereits erörterten Einstellungen durchgeführt:

- Box-Größe: 12 Å, octTree-Tiefe: 6,
- HC: complete-linkage, Metrik: Manhattan-Distanz,
- Fusionshöhen für die Bewertung: 20,
- Clusteranzahl 20, PCCA: 2 metastabile Zustände,
- Random-Forest mit Standardparametern aus dem Python-Paket *scikit-learn* mit 1000 Klassifikationsbäumen,

- MSM: 3 zeitlich aufeinanderfolgende Schnappschüsse müssen für die Zählung identisch sein.

Schließlich wird die automatische Auswertung mit der klassischen Auswertung verglichen. Aus diesem Grund wird das MSM mittels PCCA auf zwei Zustände reduziert und schließlich die möglichen MSM auf *reaktiv* und *unreaktiv* abgebildet. Hierbei wurden im einfachsten Fall, unter der Annahme, dass das *trans*-Isomer durch den Zustand 1 beschrieben wird, unabhängig von den Wahrscheinlichkeiten die zwei Klassen aus Abbildung 5.1 generiert.

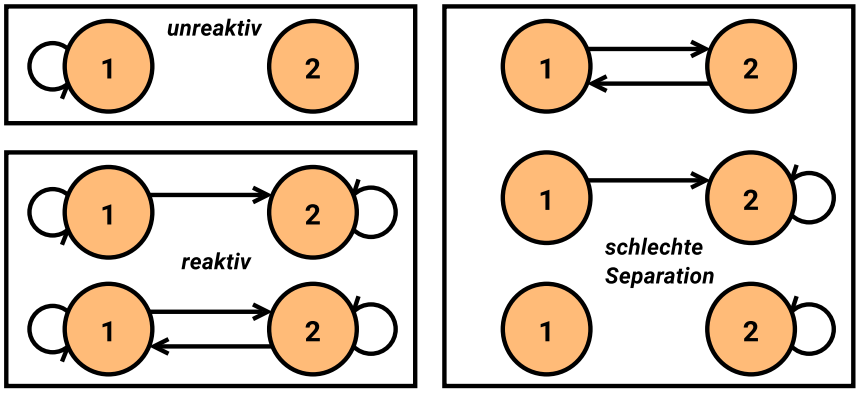


Abb. 5.1: Die reaktiven, unreaktiven Trajektorien als MSM und die durch schlechte Separation der Trainingsdaten berechneten MSM. Ausgehen von der Annahme, dass eine Trajektorie im Zustand 1 starten und einen Selbsterhalt besitzen sollte.

Unreaktiv ist eine Trajektorie, wenn vollständiger Selbsterhalt im Zustand 1 vorliegt. Reaktive Trajektorien besitzen zusätzlich Übergänge $1 \rightarrow 2$ und gegebenenfalls $2 \rightarrow 1$ und den Selbsterhalt in Zustand 2. Mögliche weitere MSM deuten auf eine unzulängliche Auswertung hin, welche häufig durch eine schlechte Separation der Trainingsdaten erzeugt wird. Der Fall eines ständigen Wechsels zwischen Zustand 1 und 2 ist eindeutig, das Fehlen des Selbsterhalts im Ausgangszustand ist in einer MD im Prinzip nicht möglich, da zu Beginn der Zustand vorliegt und es durch Trägheit der Atome kaum vorstellbar ist, dass diese in einem einzigen Schritt die strukturellen Eigenschaften grundlegend verändern.[1] Sollte lediglich der Zustand 2 ermittelt werden, so ist die Klassifikation der Schnappschüsse ebenfalls fehlerhaft. Diese Übersetzung in *reaktive* und *unreaktive* Trajektorien dient dazu, direkt mit denen in Abschnitt 3.1 vorgestellten Verfahren für die Bestimmung der Quatenausbeuten, welche in Abschnitt 3.2 angewendet werden, einen Vergleich vorzunehmen. Zu erwähnen ist an dieser Stelle, dass sich aus dem Ellenbogenkriterium für die Mehrzahl der verwendeten Kombinationen aus HC und

[1] Sollte dies beobachten werden, ist entweder die Frequenz des Abspeicherns der MD zu gering oder die mögliche Filter- bzw. Fensterfunktion des MSM wurde fehlerhaft verwendet.

Metrik drei und nicht zwei Cluster ergeben. Deshalb muss der Übertrag auf eine einfache Aussage bezüglich der Reaktivität unter Vorbehalt erfolgen.

Tab. 5.1: Die Quantenausbeuten der mit dem *chair*-Konformer gestarteten Trajektorien-Schar, bestimmt mit Hilfe des CCOC-, des CNNC-Diederwinkels (Abschnitt 3.1) und der automatischen Auswertung mit den erörterten Einstellungen. Jede Trajektorien-Schar eines Kraftmusters und einer Kraft wurde separat analysiert.

	Quantenausbeuten			Quantenausbeuten		
	CCOC	*CNNC*	*auto*	*CCOC*	*CNNC*	*auto*
$C^5C^{5'}$				$C^4C^{4'}$		
70 pN	76	95	70	71	94	79
140 pN	66	95	85	49	94	90
210 pN	42	96	64	45	98	61
280 pN	29	97	41	32	96	53
350 pN	19	94	34	14	97	52
$C^5C^{4'}$				$C^4C^{5'}$		
70 pN	73	93	77	64	95	93
140 pN	65	98	81	60	98	52
210 pN	35	90	93	38	93	80
280 pN	27	92	99	23	93	90
350 pN	21	98	90	12	98	99

In der Tabelle 5.1 werden die Quantenausbeuten exemplarisch miteinander verglichen: Dabei kommt lediglich ein einziger Parametersatz, der mit dem ML verbundenen Techniken, für die automatische Bewertung zur Anwendung. Die erwartete Korrelation zwischen angelegter Kraft und Quantenausbeuten kann für $C^5C^{5'}$ und $C^4C^{4'}$ beobachtet werden. Für $C^5C^{4'}$ und $C^4C^{5'}$ kann entgegen der Erwartungen keine Abnahme der Quantenausbeuten bei steigender Kraft beobachtet werden. Die Folgerung, dass dies an dem Kraftmuster liegt, bestätigt sich visuell nicht und widerspricht auch dem Klassifikationskriterium für den CCOC-Diederwinkel.

Aus einer anderen Perspektive kann formuliert werden, *dass* eine große Auswirkung der angelegten mechanischen Kraft auf die Reaktivität beobachtet werden kann. Diese große Auswirkung führt folglich zu dem Umstand, dass keine einzelne Trajektorie die erforderliche maximale Diversität widerspiegeln kann, die notwendig wäre, um einen guten Trainingssatz bilden zu können. Im Trainingssatz fehlen Informationen, um die strukturellen Eigenschaften gegenüber anderen einzugrenzen. Aus diesem Grund wird der angelernte Random-Forest-Algorithmus unzureichende Ergebnisse liefern, bezogen auf die global möglichen geometrischen Eigen-

schaften, da der Trainingssatz selbst nur einen Unterraum der strukturellen Eigenschaften bildet. Folgerichtig liegt die Entscheidungsgrenze des Klassifikators im Attributsraum entlang eines Bereiches mit nur unreaktiven Trajektorien und nicht *zwischen* reaktiven und unreaktiven Trajektorien. Dies ist damit zu begründen, dass jeder Satz von Trajektorien separat analysiert wird und somit die Divergenz der geometrischen Strukturen bei steigender Kraft abnimmt. Das *cis*-Isomer in Abbildung 3.2 wird auf Grund der starken, entgegengesetzten Kräfte nicht erreicht. Somit fehlen diese Instanzen für das Clustern und daher im Trainingssatz und die Klassifikation der Instanzen verändert sich verglichen mit den Trajektorien unter geringem Krafteinfluss. Dieser Effekt scheint stärker bei den diagonal angelegten Kräften, $C^5C^{4'}$ und $C^4C^{5'}$, zu sein, als bei den parallel zur Brücke ausgerichteten Kräften, $C^5C^{5'}$ und $C^4C^{4'}$.

Die Anwendung des mit geringen Kräften trainierten Klassifikators auf die übrigen Trajektorien mit gleichem Kraftmuster zeigt, Tabelle 5.2, dass die erwartete Tendenz, die Abnahme der Quantenausbeute bei ansteigender Kraft, reproduziert werden kann. Das Ergebnis ist identisch zu dem Ansatz, jeweils alle 500 Trajektorien eines angelegten Kraftmusters in einer Trajektorien-Schar zu vereinigen, zu analysieren und schließlich wieder in die angelegten Kräfte zu unterteilen. Dies ist auf dem vorgestellten Bewertungsverfahren begründet, da die Trajektorien mit niedriger Kraft erwartungsgemäß eine größere Divergenz der geometrischen Strukturen besitzen. Daher liefert die Bewertung der 100 Trajektorien mit 70 pN dieselbe Trajektorie als Trainingssatz wie die Bewertung der 500 Trajektorien aller verschiedener Kräfte.

Der numerische Wert der Quantenausbeuten ist für sich genommen kein Gütekriterium, da ein vollständig anderer Satz an Trajektorien durch das automatische Verfahren als reaktiv klassifiziert werden könnte, als es der CCOC-Diderwinkel nahelegen würde. Deshalb wird in jeder Trajektorien-Schar überprüft, wie groß die direkte Schnittmenge der Klassifikation der beiden Verfahren ist. Folglich wird der prozentuale Anteil der durch das klassische Verfahren mit dem CCOC-Diederwinkel als reaktiv klassifizierten Trajektorien, welche ebenfalls durch die automatische Auswertung als reaktiv erkannt werden, $\mathrm{CCOC} \cap \mathit{auto}$, berechnet. Würde sich keine Überlappung zeigen würden die Quantenausbeuten in dem hier entwickelten Verfahren wahrscheinlich aus vollkommen beliebigen Gründen entstehen. Die Vollständige Überlappung ist zwar ebenfalls kein absolutes Gütekriterium, da im schlimmsten Fall beide Verfahren immer noch unsinnige Ergebnisse liefern könnten. Es ist anzumerken, dass die Auswertung mittels des klassischen Verfahrens in der Literatur Verwendung findet und sollte es sich um echte Zufallszahlen handeln, liegt die Hoffnung nahe, dass dies bemerkt würde. Außerdem

wurden die vorliegenden Trajektorien visuell überprüft und die Betrachtung der Trajektorien durch den Leser kann an dieser Stelle nicht erfolgen, aber es kann versichert werden, dass diese Quantenausbeuten als Referenz in diesem Fall durchaus eine Daseinsberechtigung haben.

Tab. 5.2: Die Quantenausbeuten der mit dem *chair*-Konformer gestarteten Trajektorien-Schar, bestimmt mit Hilfe des CCOC-Diederwinkels (Abschnitt 3.1) und der automatischen Auswertung mit den erörterten Einstellungen. Der Klassifkator zur Trajektorien-Schar bei einer Kraft von 70 pN wird auch für alle Trajektorien-Scharen bei höheren Kräften angewendet.

	Quantenausbeuten			Quantenausbeuten		
	CCOC	*auto*	CCOC ∩ *auto*	CCOC	*auto*	CCOC ∩ *auto*
$C^5C^{5'}$				$C^4C^{4'}$		
70 pN	76	70	0.707	71	78	1.000
140 pN	66	66	0.742	49	71	0.980
210 pN	42	51	0.643	45	64	0.867
280 pN	29	55	0.655	32	49	0.813
350 pN	19	48	0.632	14	23	0.786
$C^5C^{4'}$				$C^4C^{5'}$		
70 pN	73	77	0.932	64	90	0.984
140 pN	65	77	0.969	60	88	0.950
210 pN	35	44	0.800	38	79	0.974
280 pN	27	30	0.815	23	65	0.783
350 pN	21	21	0.667	12	57	0.833

In Tabelle 5.3 sind die Quantenausbeuten für das *twist*-Konformer angegeben.[2]

Bei allen Klassifikationen wird in der automatischen Auswertung eine *reaktive* Trajektorie im MSM mit allen Übergängen: $1 \to 1$, $1 \to 2$, $2 \to 2$ und $2 \to 1$, beobachtet. Eine Übertragung von einer Trajektorienschar auf weitere funktioniert ebenfalls und ist in Tabelle 5.4 aufgeführt.

Weitere Ansätze für die automatische Auswertung wurden mit veränderlichen HC-Algorithmen durchgeführt. Mit WPGMA lassen sich mit gleicher Metrik, wie in Tabelle 5.5 exemplarisch gezeigt, ähnliche Ergebnisse berechnen. In dieser Arbeit wurde die Relation der Klassifikation durch den CCOC-Diederwinkel zur automatischen Auswertung verwendet, um die Sinnhaftigkeit der vorgestellten Auswertung zu zeigen. Eine a-priori-Quantenausbeute ohne Modellcharakter als Referenz, we-

[2] † Für das Muster $C^4C^{5'}$ des *twist*-Konformers sind bei 210 pN nur 99 Trajektorien verfügbar, da eine Trajektorie dissoziiert ist und wegen der resultierenden räumlichen Ausdehnung nicht in die *octParameter* übersetzt werden konnte.

Tab. 5.3: Die Quantenausbeuten der mit dem *twist*-Konformer gestarteten Trajektorien-Schar, bestimmt mit Hilfe des CCOC-Diederwinkels (Abschnitt 3.1) und der automatischen Auswertung mit den erörterten Einstellungen. Der Klassifkator zur Trajektorien-Schar bei einer Kraft von 70 pN wird auch für alle Trajektorien-Scharen bei höheren Kräften angewendet.

	Quantenausbeuten			Quantenausbeuten		
	CCOC	*auto*	*CCOC* ∩ *auto*	*CCOC*	*auto*	*CCOC* ∩ *auto*
$C^5C^{5'}$				$C^4C^{4'}$		
70 pN	76	87	0.987	77	96	1.000
140 pN	56	82	0.982	55	84	1.000
210 pN	50	85	0.920	42	83	0.952
280 pN	37	75	0.892	26	80	0.962
350 pN	20	63	0.850	18	65	0.944
$C^5C^{4'}$				$C^4C^{5'}$		
70 pN	80	92	0.988	73	92	0.986
140 pN	60	91	0.967	75	94	0.987
210 pN	45	86	0.956	45	$85^{\dagger}$	0.78
280 pN	30	78	0.933	29	84	0.897
350 pN	14	76	0.929	12	73	0.917

Tab. 5.4: Die Quantenausbeuten der mit dem *chair*-Konformer gestarteten Trajektorien-Schar, bestimmt mit Hilfe des CCOC-Diederwinkels (Abschnitt 3.1) und der automatischen Auswertung mit den erörterten Einstellungen. Der Klassifkator zur Trajektorien-Schar bei einer Kraft von 70 pN wird auch für alle Trajektorien-Scharen bei höheren Kräften angewendet.

	Quantenausbeuten			Quantenausbeuten		
	CCOC	*auto*	*CCOC* ∩ *auto*	*CCOC*	*auto*	*CCOC* ∩ *auto*
$\underline{C^5C^{5'}}$				$C^4C^{4'}$		
70 pN	76	83	0.947	71	85	1.000
140 pN	66	86	0.985	49	82	0.959
210 pN	42	78	0.976	45	82	0.911
280 pN	29	73	0.931	32	75	0.969
350 pN	19	60	0.790	14	57	0.786
$C^5C^{4'}$				$C^4C^{5'}$		
70 pN	73	83	0.959	64	85	0.984
140 pN	65	90	0.969	60	92	1.000
210 pN	35	74	0.943	38	80	0.921
280 pN	27	65	1.000	23	71	0.783
350 pN	21	62	0.952	12	60	0.917

der methodisch von einer klassischen Auswertung, noch von statistischen Verfahren stammend, gibt es nicht. Ebenso wenig bringt eine intrinsische Beurteilung der Güte der verwendeten ML-Methoden keinen Mehrwert, da allein die Anwendbarkeit und das endgültige Ergebnis als Referenz dienen können. Die exakten strukturellen Eigenschaften, bei denen eine Separation durch die gewählte HC-Methode und Metrik durchgeführt wird, sind unterschiedlich. Aus diesem Grund kommt es zu einer Verschiebung bei der Beurteilung der einzelnen Strukturen. Anschaulich kann eine einfache unidirektionale ideale Reaktion betrachtet werden, bei der nur ein Übergang vom *trans*-Isomer zum *cis*-Isomer auf dem schnellstmöglichen bzw. kürzesten Weg stattfindet. Die Geometrien entlang der Zeitachse sind automatisch benachbart, und je nach verwendeter Clustertechnik wird sich die Clustergrenze auf dieser vereinfachten Reaktionsachse bei lediglich zwei Clustern irgendwo zwischen dem *cis*- und *trans*-Isomer befinden. Davon ist schließlich die spätere Klassifikation der Schnappschüsse in diesem automatischen Arbeitsablauf abhängig, welche schließlich die ermittelte Reaktivität beeinflusst. Die Verwendung einer größeren Anzahl an Clustern kann dieses Phänomen entlang der zeitlichen Veränderung abschwächen.[3] An dieser Stelle kann schließlich das MSM mit der PCCA Abhilfe schaffen, da auf diesem Weg die rein strukturelle Bewertung in den Hintergrund tritt und eine zeitliche Nähe in den Vordergrund rückt.

Tab. 5.5: Die Quantenausbeuten der mit dem *chair*-Konformer gestarteten Trajektorien-Schar, bestimmt mit Hilfe des CCOC-Diederwinkels (Abschnitt 3.1) und der automatischen Auswertung mit den erörterten Einstellungen und WPGMA als HC. Der Klassifkator zur Trajektorien-Schar bei einer Kraft von 70 pN wird auch für alle Trajektorien-Scharen bei höheren Kräften angewendet.

	Quantenausbeuten			Quantenausbeuten		
	CCOC	*auto*	*CCOC* ∩ *auto*	*CCOC*	*auto*	*CCOC* ∩ *auto*
$C^5C^{5'}$				$C^4C^{4'}$		
70 pN	76	80	0.933	71	82	1.000
140 pN	66	73	0.864	49	74	0.980
210 pN	42	58	0.857	45	74	0.889
280 pN	29	49	0.759	32	56	0.844
350 pN	19	35	0.632	14	40	0.786

[3] Abstellen lässt sich dieses Phänomen bei MD-Trajektorien nicht, da ein Atom nur ein begrenztes Stück Weg innerhalb eines Zeitschrittes bewegen kann und weniger gespeicherte Zeitschritte eine MD sinnlos machen würde.

5.2 Euklidische Metrik als Alternative

Unter Verwendung der euklidischen Metrik lassen sich, wie in Tabelle 5.6 dargestellt, keine konsistenten Ergebnisse mit dem WPGMA-Algorithmus erhalten. Die Trends lassen sich, ebenso wie für den HC-Algorithmus complete-linkage, Gleichung (2.5), gezeigt in Tabelle 5.7, nicht für jeden Satz an Trajektorien reproduzieren. Außerdem ist die Überlappung, CCOC $\cap$ *auto*, beispielsweise für das Kraftmuster $C^5C^{5'}$, gering. Dies zeigt, dass die oct-Attribute für das vorgestellte chemische System in normierter Form mittels der Manhattan-Distanz deutlich zuverlässiger separiert werden können als mittels der euklidischen.

Wenigstens für den betrachteten Photoschalter ist die Verwendung der normierten oct-Attribute ein Verbesserung, welche zusätzlich unter Verwendung der Manhattan-Distanz zusammen mit der HC-Methode complete-linkage gute Ergebnisse liefert. Dies ist, da die abschließende Bewertungsmethode, Abbildung 5.1, restriktiv ist und die Beträge für Übergangswahrscheinlichkeiten nicht explizit mit einbezieht, ein gutes Ergebnis. Ob diese Kombination der ML-Einstellungen für ein breiteres Anwendungsfeld direkt verwendbar sind, kann aus diesen Ergebnissen nicht abgeleitet werden. Allerdings lässt sich abschätzen, dass die Datenvorbereitung durch den OctTree und den resultierenden oct-Attribute erfolgreich durchgeführt werden kann. Die konkrete Wahl der ML-Methoden und Submethoden bis hin zum berechneten MSM und der anschließenden Bewertungslogik, sollten für explizite chemische Systeme nochmals angepasst werden. Da es sich in diesem automatisierten Arbeitsablauf um eine unüberwachte ML-Methode handelt, ist je nach Einstellung der gewonnene Informationsgehalt unterschiedlich. Dies ist generell im Bereich des ML neben der Datenauf- und vorbereitung eine der Kernaufgaben bei der Analyse von neuen unbekannten Daten.

Tab. 5.6: Die Quantenausbeuten der mit dem *chair*-Konformer gestarteten Trajektorien-Schar, bestimmt mit Hilfe des CCOC-Diederwinkels (Abschnitt 3.1) und der automatischen Auswertung mit den erörterten Einstellungen und WPGMA mit euklidischer Distanz als HC. Der Klassifkator zur Trajektorien-Schar bei einer Kraft von 70 pN wird auch für alle Trajektorien-Scharen bei höheren Kräften angewendet.

	Quantenausbeuten			Quantenausbeuten		
	CCOC	*auto*	*CCOC* ∩ *auto*	*CCOC*	*auto*	*CCOC* ∩ *auto*
$C^5C^{5'}$				$C^4C^{4'}$		
70 pN	76	37	0.387	71	82	1.000
140 pN	66	36	0.333	49	77	0.959
210 pN	42	46	0.405	45	76	0.889
280 pN	29	39	0.414	32	57	0.844
350 pN	19	42	0.474	14	41	0.786

Tab. 5.7: Die Quantenausbeuten der mit dem *chair*-Konformer gestarteten Trajektorien-Schar, bestimmt mit Hilfe des CCOC-Diederwinkels (Abschnitt 3.1) und der automatischen Auswertung mit den erörterten Einstellungen und complete-linkage mit euklidischer Distanz als HC. Der Klassifkator zur Trajektorien-Schar bei einer Kraft von 70 pN wird auch für alle Trajektorien-Scharen bei höheren Kräften angewendet.

	Quantenausbeuten			Quantenausbeuten		
	CCOC	*auto*	*CCOC* ∩ *auto*	*CCOC*	*auto*	*CCOC* ∩ *auto*
$C^5C^{5'}$				$C^4C^{4'}$		
70 pN	76	43	0.400	71	79	1.000
140 pN	66	39	0.424	49	72	0.959
210 pN	42	31	0.381	45	63	0.867
280 pN	29	43	0.448	32	42	0.750
350 pN	19	44	0.421	14	20	0.786

5.3 Besondere Ereignisse

An dieser Stelle wird explizit auf ein auftretendes seltenes Ereignis einiger Trajektorien des brAB-O unter Krafteinwirkung, Abschnitt 3.2, eingegangen. Es zeigte sich, dass durch die gewählte Art der quantenmechanischen Beschreibung eine Trajektorie eine Besonderheit aufweist. Das klassische Auswertungsverfahren aus Abschnitt 3.1 verwendet die Diederwinkel der Brückenatome, welche in Abbildung 5.2 gezeigt sind. Es ist zu erkennen, dass ab etwa dem Schnappschuss 1620 starke Ausschläge beider Diederwinkel zu beobachten sind. Die vorgestellten klassischen Verfahren untersuchen lediglich, ob sich die Diederwinkel für das gewählte Zeitintervall in dem definierten Toleranzintervall befinden. Mit denen in Abschnitt 3.1 gewählten Parametern[4] wird diese Trajektorie bezüglich beider Diederwinkel als *reaktiv* bewertet.

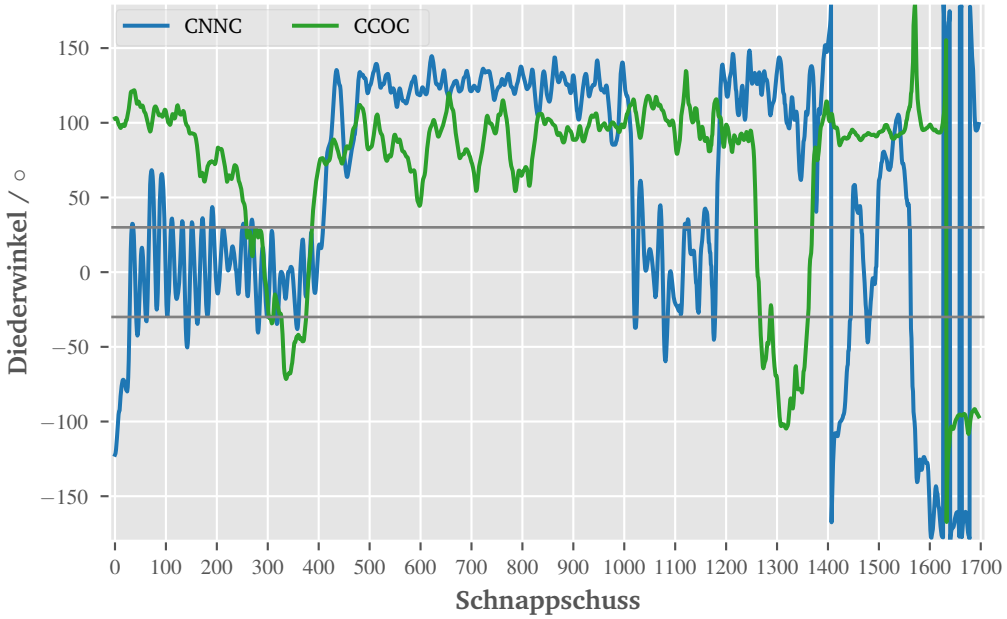


Abb. 5.2: Der zeitliche Verlauf der Diederwinkel der beiden Brücken aufgespannt durch *CNNC* und *CCOC* des BrAB-O. Ab etwa Schnappschuss 1620 sind starke Ausschläge in beiden Brücken zu beobachten.

Die Betrachtung des Trajektorienfilms zeigt ab etwa 1114, dass zwei Bindungen in einem Benzolring zeitweise über 2 Å lang sind. Anschließend setzt sich dieses Phänomen in die Brücke fort und resultiert ab etwa 1380 in einer Dissoziation der CCOC-Brücke. Ab diesem Zeitpunkt ist die Berechnung dieses Diederwinkels unzweckmäßig, außerdem ist zu beobachten, dass nun die zweite Brücke frei drehbar ist. Das gesamte Ereignis lässt sich anschaulich mittels des RMSD-Graphen in

[4] 20 Schnappschüsse mit 30° Toleranz

Abbildung 5.3 illustrieren. Zum Zeitpunkt des Bindungsbruchs ist der RMSD-Wert verhältnismäßig gering, anschließend ist die Öffnung der Brücke und ab Schnappschuss 1500 die Rotation um die CNNC-Brücke zu erkennen.

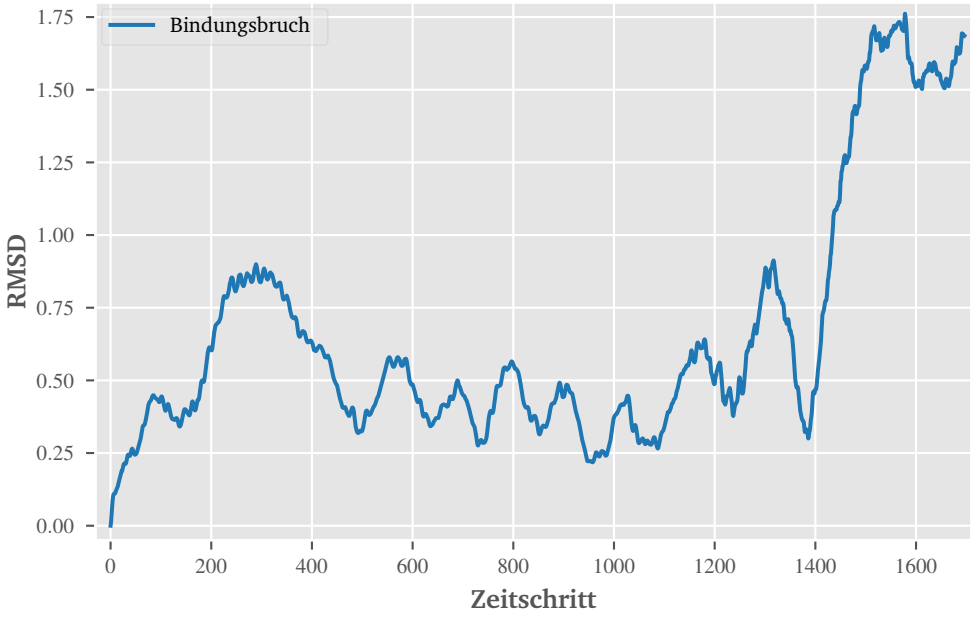


Abb. 5.3: Der zeitliche Verlauf des RMSD-Werts des BrAB-O während eines Bindungsbruches. Ab etwa Schnappschuss 1480 werden zuvor nicht erreichte Werte beobachtet. Anschließend wird eine Rotation zu vollführt.

Die theoretisch-chemische Folgerung kann sein, dass methodisch ein zu großer Zeitschritt in der berechneten Simulation verwendet wurde. Es könnte ebenso sein, dass sich fälschlicherweise Energie akkumulieren konnte, oder dass sich dieses seltene Ereignis einer Energieakkumulation tatsächlich beobachten lässt. Diese Akkumulation könnte wiederum ein tatsächlich seltenes Ereignis in der untersuchten Photochemie sein, welche sehr selten zu beobachten ist, oder aber dieses Ereignis ist in Wirklichkeit nicht so selten und die Dissoziationsbarriere des vorliegenden Systems wird unterschätzt und stellt einen konkurrierenden Prozess zur Photoisomerisierung dar. Dieser Sachverhalt soll an dieser Stelle nicht weiter vertieft werden, im Vordergrund steht, dass dieses Ereignis detektiert werden konnte.

Die Folgerung bezüglich der klassischen Auswertung unter Verwendung handverlesener interner Koordinaten ist, dass diese zu angepasst an die Erwartungen entwickelt werden. Diese starke Fokussierung auf das chemisch Erwartete und das Ausblenden alles anderen erzeugt unter Umständen eine Art ›Tunnelblick‹. Eine angelagerte Überprüfung, ob z.B. das Bindungsmuster erhalten geblieben ist, erfordert mehr Entwicklungs- und Rechenzeit und verlangsamt die Forschungsarbeit

vordergründig, sichert Resultate aber zusätzlich ab.

Mit dem automatischen Arbeitsablauf aus Kapitel 4, in Abbildung 4.1 schematisch gezeigt, ist folglich das Problem der chemischen Voreingenommenheit, die gezielt Prozesse bzw. Nichterwartetes ausklammert und sogar zu offensichtlichen Fehlevaluationen führen kann, umgangen. Daher lassen sich mit diesem *generellen* automatischen Verfahren derartige Trajektorien sofort erkennen. Es ist in diesem Arbeitsschema nicht möglich, diese Trajektorie zu ›übersehen‹, da die gezeigte Trajektorie diejenige mit der größten Diversifikation der Geometrien ist und damit die höchste Bewertung erhält. Aufgefallen ist die hier vorgestellte Trajektorie ohne die Eingrenzung auf lediglich die zwei Zustände *un-/reaktiv* sondern mit beispielsweise fünf oder mehr Zuständen. Hierbei nahm schließlich die Trajektorie mit dem Bindungsbruch Zustände ein, welche keine weitere Trajektorie sonst erreichte. Allerdings fehlte an dieser Stelle eine automatische Heuristik für die Einteilung der MSM mit mehr als zwei Zuständen und daher wurden zu diesem Zeitpunkt die Zeitreihen noch vollständig händisch überprüft.[5] Für die automatische Analyse wurde diese Trajektorie in der vorliegenden Trajektorien-Schar belassen. Auf diesem Weg konnten eventuelle Auswirkungen auf die Gesamtbeurteilung beobachtet werden. Da dies in den vorliegenden 100 Stück die einzige ist, verschiebt sich die Klassifikation der übrigen Trajektorien und die Quantenausbeuten fallen geringer aus. Da für die Klassifikation der Trajektorien als *un-* und *reaktiv* ein vorhandener Übergang ausreicht und die Anzahl an resultierenden metastabilen Zustände auf exakt 2 gesetzt wird, sind die Auswirkungen relativ gering. Die Zustände, welche isoliert in der besagten Trajektorie auftreten, werden von einem metastabilen Zustand absorbiert und auf diesem Wege maskiert. Bei einer vollständig automatischen Auswertung, ohne fester Anzahl an metastabilen Zuständen, würde die Anzahl für den PCCA, Abschnitt 2.3, aus dem vorgestellten Ellenbogenkriterium resultieren, Abbildung 2.10. An dieser Stelle würde – mutmaßlich – ein metastabiler Zustand gebildet werden, der nur in der Trainingstrajektorie enthalten ist. Mit Hilfe einer angepassten, im idealen Fall automatischen, Bewertungslogik für die fertigen berechneten MSM der einzelnen Trajektorien würde sich vermutlich keine weitere Trajektorie dieser Schar finden, welche ähnlich zu derjenigen des Trainingssatzes ist. Dieser Zwang auf exakt 2 metastabile Zustände beeinflusst verständlicherweise die Bestimmung der restlichen Quantenausbeuten. Aus diesem Grund wurde die besagte Trajektorie aus der Trajektorien-Schar entfernt.

Damit ergeben sich für die Trajektorienschar mit dem Kraftmuster $C^5C^{5'}$ veränderte Quantenausbeuten, Tabelle 5.8, welche als absolute Zahl von 99 Trajektorien

[5] Zu diesem Zeitpunkt wurden alle Trajektorien als Zeitreihenauftragung angeschaut und die auffälligen Trajektorien visuell überprüft.

angegeben sind. Es ist zu beobachten, dass zwar die Quantenausbeuten scheinbar schlechter passen, allerdings ist die Überlappung des CCOC-Klassifikators mit dem automatischen Ablauf größer. Damit werden die Quantenausbeuten aus den richtigen Gründen erzeugt – der absolute Wert ist keine globales Gütekriterium. Es ist zu erkennen, dass die Schnittmenge von CCOC- und *auto*-Klassifikation deutlich größer ist als vor der Entfernung der Trajektorie. Gleichzeitig ist die Tendenz der Quantenausbeuten erhalten geblieben.

Tab. 5.8: Die Quantenausbeuten des *chair*-Konfomers, ohne die vorgestellte Trajektorie mit Bindungsbruch, mit den erörterten Einstellungen aus Kapitel 5. Gezeigt ist die absolute Anzahl an Trajektorien von 99. Der ermittelte Klassifikator aus 70pN wird auf die größeren Kräfte angewendet. Die Überlappung ist deutlich vergrößert.

	Quantenausbeuten		
	CCOC	*auto*	*CCOC* ∩ *auto*
$C^5C^{5'}$			
70 pN	76	83	0.947
140 pN	66	86	0.985
210 pN	42	78	0.976
280 pN	29	73	0.931
350 pN	19	60	0.790

Trajektorien, welche komplett dissoziieren, erreichen eine große räumliche Ausdehnung und werden bereits bei der Berechnung der oct-Attribute detektiert und können nicht übersetzt werden. Daher sind diese Art von Trajektorien unproblematisch und werden, wie bereits in Tabelle 5.3 angemerkt, bei der Datenvorbereitung erkannt und nicht konvertiert. Bei dem vorliegenden System unter Krafteinwirkung ist dies ein einziges Mal bei 2000 MD-Simulationen geschehen.

5.4 Übertragung zwischen MD-Systemen

Durch die oct-Attribute, welche die Datengrundlage für die vorgestellten ML-Techniken bilden, ist es möglich, die Klassifikation von einem Molekül auf modifizierte Derivate zu übertragen und direkt anzuwenden. Diese Übetragung ist in diesem Fall umsetzbar, da anders als in anderen chemoinformatischen Datenrepräsentationen, insbesondere für weitere ML-Ansätze[12,18,29–31,88], der hier verwendete Ansatz einen rein geometrischen Attributsraum verwendet. Es wird vielmehr ein hierarischer, diskreter OctTree-Raum verwendet, dessen abgeleitete Attribute daher vergröberte geometrische Tendenzen oder Dynamikveränderungen

erkennen sollen. Dies ist damit aber grundlegend unabhängig von den faktisch vorhandenen Atomen: Es werden also keine (qualitativ) unterschiedlichen Repräsentationen entworfen, abhängig vom Periodensystem der Elemente. Die typische chemische Fragestellung, was an Dynamikänderungen geschieht, bei kleinerer Konfigurationsänderungen des Systems, können damit auch direkt beantwortet werden, häufig ohne einen neuen Klassifikator anzulernen. Daher soll hier der Transfer eines angelernten Klassifikators des brAB-O auf das ebenfalls vorgestellte, zweifachfunktionalisierte brAB-Derivat, das Diamino-brAB in Abbildung 3.8, vorgestellt werden.

Tab. 5.9: Die Quantenausbeuten des Diamino-brAB-Isomers bestimmt mittels des CCOC-, des CNNC-Diederwinkels (Abschnitt 3.1) und der automatischen Auswertung mit den erörterten Einstellungen aus Kapitel 5. Der trainierte Klassifkator zur Trajektorien-Schar des *chair*-brAB-O ($C^5C^{5'}$, 70 pN) wird ebenfalls auf die Trajektorien-Scharen der Diamino-brAB-Isomere angewendet.

	Quantenausbeuten			
	CCOC	*CNNC*	*auto*	*CCOC* ∩ *auto*
brAB-O	76	95	83	0.947
cis-Diamino	83	99	50	0.542
trans-Diamino	56	78	57	0.982

Die Tabelle 5.9 zeigt, dass sich der Klassifikator auf das *trans*-Diamino gut übertragen lässt, für das *cis*-Diamino ergibt sich durch die fehlende, bzw. nicht mögliche Ausrichtung des Systems eine deutlich geringere Überlappung. Damit ist gezeigt, dass trotz der oct-Attribute eine Rotationsinvarianz nicht gegeben ist und somit eine Vorausrichtung notwendig ist.

5.5 MD-Analyse eines größeren Systems

Ein weiterer Versuch, welcher allerdings ohne vergleichbares Ergebnis durchgeführt wurde, ist die exemplarische Untersuchung des Abrissexperiments aus der Doktorarbeit von J. Müller[89]. Dabei geht es in erster Linie darum zu überprüfen, ob deutlich größere Systeme und damit verbundene Räume überhaupt in oct-Attribute umgewandelt werden können und weitere Reaktionstypen analysierbar sind. Das verwendete Molekül ist in Abbildung 5.4 dargestellt, die Polyethylenglycolketten, PEG, dienen als Aufhängung, an denen die Kraft, vektoriell auseinander wirkend, angelegt wird. Die zentrale Einheit ist die Disulfidbrücke, welche im idealen Fall dissoziiert. Durch eine weitere Brücke, welche als Sicherheitsleine be-

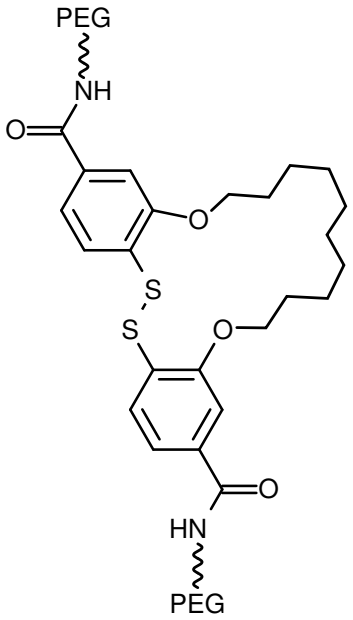


Abb. 5.4: Das Molekül aus der Arbeit von J. Müller.[89] Die Polyethylenglycolketten, PEG, werden als Aufhängung in dieser Simulation verwendet. Die Kraft wird an den Enden des PEG angelegt, vektoriell in entgegengesetzte Richtungen zeigend (auseinander). Zentral ist die Disulfidbrücke, welche unter Kraft geplant dissoziieren soll. Überbrückt wird diese durch die sogenannte Sicherheitsleine, welche den vollständige Abriss verhindern soll.

zeichnet wird, soll der vollständige Abriss verhindert werden. Als HC wurde, wie in Kapitel 4, complete-linkage zusammen mit der Manhattan-Distanz verwendet. Es wurde mit einer periodischen Box mit 100 Å Kantenlänge gearbeitet. Durch die periodischen Randbedingungen ergibt sich der Vorteil, dass die benötigte maximale Ausdehnung bekannt ist. Gleichzeitig ergibt sich der Nachteil, dass sich, durch die periodischen Randbedingungen, gebundene oder ungebundene Atome von einem zum nächsten Simulationsschritt von einer Seite der Box zu der gegenüberliegenden bewegen. Es werden 204 Atome, bzw. 94 Schweratome, in 250 Trajektorien mit jeweils 5000 Schnappschüssen analysiert.[6] Dazu wurde die Auflösung, wegen der deutlichen Systemvergrößerung, erhöht und damit eine Lage im octTree hinzugefügt. Zusammenfassend ausgedrückt wird an dieser Stelle ein deutlich größeres System mit mehr Trajektorien und mehr Schnappschüssen pro Trajektorie analysiert. Dazu wird zusätzlich eine feinere Auflösung des OctTrees verwendet, wodurch die Analyse rechenintensiver wird.

Es zeigt sich, dass mit der Ausrichtung der Struktur und einer Reduktion der Zustände mittels PCCA auf drei Zustände, eine Separation der Trajektorien, welche keinen Abriss aufweisen, eindeutig erfolgt. Die Trajektorie mit der höchsten

[6]Die Trajektorien werden für 25 ps simuliert, die exakte angelegte Kraft ist weniger entscheidend, einzig der Sachverhalt, dass an dieser Stelle etwas passiert, ist von Bedeutung. Für weitere Details und Ergebnisse siehe Mueller et al.[89].

Bewertung, welche als Trainingssatz dient, ist eine mit einem Bindungsbruch beider Brücken. Anschließend lässt sich beobachten, wie sich die entstandenen Fragmente durch die periodische Box bewegen. Dies ist mit den oben aufgestellten Annahmen zu erwarten und kann als Indiz für eine sinnvolle Analyse der vorliegenden Trajektorien gewertet werden. Auch ohne eine weitere explizite händische Analyse aller Trajektorien, kann festgehalten werden, dass das vorgestellte Verfahren auch für größere Systeme als 10 Å funktioniert und die gewählte Kompression der Schnappschüsse auch bei 94 Schweratomen systematisch erwartete Ergebnisse liefert. Dies gelang bei einer gänzlich anderen Reaktion, welche während der Entwicklung und Testung des Programms zu keinem Zeitpunkt eine Referenz darstellte. Daher kann erwartet werden, dass das Anwendungsspektrum dieses Verfahrens relativ breit ist. Allein die Möglichkeit 250 Trajektorien grob zu kategorisieren, ohne explizites Wissen einbringen zu müssen, bedeutet eine Arbeitsentlastung für den Anwender. Der gewählte Zeitschritt in den Phototrajektorien und den an dieser Stelle analysierten ist nicht identisch, daher bieten sich weitere Anpassungen an. Dies könnte über die Fensterfunktion des MSM und die Anzahl der erzeugten Cluster ebenso wie über die durch PCCA erzeugten Zustände erfolgen.

Kapitel 6

Zusammenfassung und Ausblick

In dieser Arbeit konnte gezeigt werden, dass die Einbindung des maschinellen Lernens (ML) in den Arbeitsablauf der Auswertungen von Molekularer Dynamik (MD) gelingen kann und auf diesem Weg die Analyse vereinfacht und standardisiert. Ohne explizites Vorwissen der erreichbaren Zustände oder der Energetik eines zu simulierenden Systems, rein mittels der erhaltenen strukturellen Daten, ist eine vollautomatische Auswertung möglich. Die Überführung eines molekularen Systems in eine dreidimensionale gerasterte Darstellung des Octalbaums (OctTree) konnte erfolgreich durchgeführt werden. Die besetzten Voxel des OctTrees können in Superstrukturelemente umgewandelt werden, welche nicht mit der Atomanzahl skalieren und permutationsinvariant sind. Diese Superstrukturelemente konnten erfolgreich als Datenbasis für eine vollständig automatische Auswertung verwendet werden. Durch das vorgestellte Bewertungsschema, welches auf der Auswertung von Hierarchischen Clustern (HC) basiert, kann eine maximal divergente Trajektorie herausgefiltert werden und anschließend als Trainingssatz für einen Random-Forest-Algorithmus verwendet werden. Dieser Random-Forest-Klassifikator kann erfolgreich jede einzelne Geometrie einer Trajektorie klassifizieren.

Die Superstrukturelemente erlauben eine Übertragung des angelernten Klassifikationsalgorithmus auf alle Trajektorien einer Schar und damit auf ein identisches Molekül unter leicht veränderten Simulationsbedingungen. Zusätzlich konnte gezeigt werden, dass leicht modifizierte molekulare Systeme durch denselben Klassifikationsalgorithmus erfolgreich klassifiziert werden können. Ein deutlicher Vorteil dieses Ansatzes ist, dass keine zusätzlichen Informationen oder Erwartungen für die automatische Auswertung benötigt werden. Daher ist es möglich, vollkommen unvoreingenommen die Ergebnisse auszuwerten und auf diesem Weg keine Ereignisse, wie in Abschnitt 5.3 explizit gezeigt, zu übergehen. Die vorgestellten Methoden dienen zur schnellen Überprüfung und Sortierung der berechneten Simulationsdaten, bevor die erste Betrachtung überhaupt stattfindet. Die

Einführung einer zeitlichen Auflösung in die vorhandenen Daten ermöglicht eine Klassifikation der Trajektorien als *un-/reaktiv* mittels eines Markov-State-Models (MSM). Die Wahl einer größeren Anzahl an Clustern kann die Separation von wichtigen Intermediaten direkt aus Dynamiksimulationen ermöglichen. Diese Auswertung kann auf einem handelsüblichen Desktopcomputer durchgeführt werden.

Die weitere Anpassung der Methode zur Bewertung der resultierenden MSM erlaubt vollautomatische Analysen und wurde in dieser Arbeit für die Darstellung der Ergebnisse als sogenannte Quantenausbeuten erfolgreich angewendet. Die Photoisomerisationsreaktionen des Diamino-brAB und des brAB-O konnten automatisch bezüglich ihrer Quantenausbeuten ausgewertet werden. Dabei zeigten sich die erwarteten Tendenzen, dass eine Kraft, welche einer Photoisomerisierung entgegenwirkt, die Quantenausbeuten verringert. Mit Hilfe der vorgestellten Algorithmen gelang die Analyse, ohne der Vorgabe für das Auswertungssystem, dass überhaupt eine Photoreaktion vorliegt. Es konnte gezeigt werden, dass eine Trajektorie mittels unüberwachter ML-Methoden nach strukturellen Eigenschaften eingeteilt werden kann. Der erstellte Trainingssatz aus einer einzigen Trajektorie als Grundlage für das Anlernen eines Klassifikators ist in den vorgestellten Anwendungsfällen ausreichend. Das eingeführte Bewertungskriterium für die Auswahl einer Trainingstrajektorie konnte erfolgreich angewendet werden und zeigt sich auch bei der Untersuchung eines größeren bindungsbrechenden Systems, Abschnitt 5.5, als zuverlässig. Damit konnte herausgestellt werden, dass es nicht nur speziell für die Photoisomerisierung anwendbar ist, sondern auch für eine vollständig unterschiedliche Reaktion. Aus diesem Grund kann als Fazit festgehalten werden, dass dieser Ansatz höchstwahrscheinlich ein breiteres und universelleres Anwendungsfeld besitzt und zukünftig Beachtung finden sollte.

Ausblick

Im Rahmen eines Projektes zum automatischen Design von molekularen Maschinen – molekulare Effektoren – ist die Beurteilung der molekularen Bewegungen essentiell.[90] In einem laufenden Projekt, zu welchem eine vom Autor betreute Masterarbeit zählt, werden aus einer Molekülfragmentdatenbank erzeugte Molekülfragmente bezüglich ihrer mechanischen Stabilität überprüft. Zur Zeit findet beispielsweise die GDB-Datenbank Verwendung.[91–98] Die Datenbank GDB-11 enthält alle kleinen organischen Moleküle mit bis zu 11 Atomen, bestehend aus Kohlen-, Stick- und Sauerstoff, sowie Fluor. Damit sind 26.4 Millionen Strukturen und fast 110 Millionen Stereoisomere verfügbar, welche analysiert werden kön-

nen.[1] Dabei kommt die automatische Auswertung zum Einsatz, mit deren Hilfe der direkte Vergleich zwischen Fragmenten unterschiedlicher atomarer Zusammensetzung aber ähnlicher Geometrie möglich ist. Die prinzipielle Anwendbarkeit wurde in Abschnitt 5.4 gezeigt. Bei dem Design von molekularen Maschinen geht es in erster Linie um die strukturellen und damit mechanischen Eigenschaften. Daher ist die Verwendung einer Repräsentation in Form der vorgestellten oct-Attribute ein naheliegendes Vorgehen. Zumal die Permutationsinvarianz es ermöglicht, direkt kleinere Konfigurationsänderungen vorzunehmen oder einzelne Atome durch andere zu ersetzten, ohne dass der angelernte Bewertungsalgorithmus erneut trainiert werden muss. Auf diesem Weg ist es möglich, ein bekanntes Fragment mit bekannten Eigenschaften als Referenz zu verwenden und ähnliche molekulare Strukturen mit ähnlichen und gegebenenfalls verbesserten Eigenschaften zu finden.

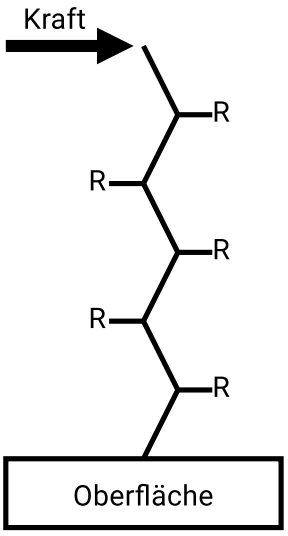


Abb. 6.1: Ein orthogonal zur Oberfläche ausgerichtetes Molekülfragment mit seitlich angelegter Kraft. Die Reste R und auch die Kohlenstoffatome in diesem Schema können durch beliebige Atome ersetzte werden. Die diskrete Auswahl erfolgt durch Moleküle aus einer beispielsweise GDB-11/-13 Datenbank.

Die untersuchten Systeme werden orthogonal auf einer Oberfläche festgehalten, und am äußeren Ende werden in unterschiedlichen Richtungen Kräfte angelegt, Abbildung 6.1. Zu diesem Zweck wurde bereits in das MD-Programm Tinker eine zusätzliche Funktion implementiert.[99] Auf diesem Weg ist es möglich in unterschiedlichen Winkeln und zu definierten Zeiten Kräfte auf ein System wirken zu lassen. Ein Teil der Untersuchung ist die Stabilitätsüberprüfung eines Moleküls gegenüber seitlich angelegten Kräften. Dabei kann die Simulation mit stetig steigender Kraft oder wechselnder Richtung für kurze Simulationszeiten wiederholt

[1] Ebenso gibt es entsprechend eine GDB-13 oder GDB-17 Datenbank mit folglich bis zu 13 oder sogar 17 Schweratomen.

werden. Zunächst wird in dieser Arbeit exemplarisch die in Raeker et al.[100] vorgestellte Cilie für den molekularen Transport als Referenzsystem verwendet. Das in dieser Arbeit vorgestellte Verfahren erlaubt es, die Referenz-Trajektorie mit gewünschter Stabilität unter Krafteinfluss zu verwenden. Iterativ werden anschließend Fragmente aus der vorliegenden Datenbank mit diesem Verhalten verglichen. All dies kann im Gegensatz zu statischen Rechnungen direkt aus den MD-Simulationen abgeleitet werden.

Das Ziel ist, zunächst die zufällig generierten Moleküle mittels einer MD-Simulation bezüglich der Stabilität bewerten zu können. Darauf folgen soll schließlich die Verkürzung der Simulationszeit und die Optimierung der benötigten Rechenzeit für die automatische Auswertung. Münden soll dies idealerweise in einer Verknüpfung der Evaluation der Ergebnisse mit der Auswahl der Fragmente in der GDB-11/-13 Datenbank, möglicherweise mittels Methoden der globalen Optimierung oder mit Hilfe von ML-Techniken.[101]

Als Anwendungsbeispiel dienen Bauteile von molekularen Maschinen, welche für den erfolgreichen Einsatz definierten Kräften standhalten müssen. Über eine gewählte Referenz können neuartige Strukturen mit definierter Stabilität gefunden werden, welche unter Umständen beispielsweise bessere chemische Eigenschaften besitzen. Auf diesem Weg könnte es schließlich gelingen, abstrakte geometrische Muster durch passende Molekülfragmente darzustellen, welche sich in einer MD-Simulation durch gewünschte Eigenschaften auszeichnen. Dabei kann nicht nur ein festes Gerüst gebildet werden, sondern durch die Analyse der zeitlichen und räumlichen Eigenschaften können direkt molekulare Effektoren aus einer Datenbank zusammen gesetzt werden. Vorliegen muss hierzu lediglich ein einfaches räumliches und zeitliches Muster einer einfachen Struktur, welches nicht einmal chemisch sinnvoll sein muss. Die Übersetzung in den OctTree und die resultierenden oct-Attribute ermöglichen eine Übertragung zwischen den vorliegenden Systemen.

Ein abstrakt generiertes Bewegungsmuster eines Systems, in kartesischen Koordinaten vorliegend, kann als repräsentativer Reaktionsverlauf Anwendung finden, Abbildung 6.2. Der unmittelbare Vergleich mit beliebigen Molekülen unter definierten Einflüssen kann auf diesem Weg automatisch durchgeführt werden. Dabei ergeben sich neben den standardisierten quantenchemischen Eigenschaften neue Möglichkeiten der Bewertung – speziell bezüglich der dynamischen Eigenschaften.

Neben der Optimierung kann die Güte der reinen Analyse verbessert und weiterentwickelt werden. Dies kann bis hin zu einem wirklich automatischen Prozedere vorangetrieben werden, wobei der gesamte Arbeitsablauf mit seinen wählbaren ML-Parametern einer globale Optimierung oder meta-Optimierung während des

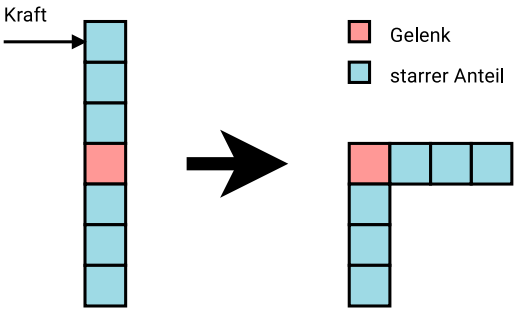


Abb. 6.2: Die abstrakte zweidimensionale Darstellung eines molekularen Gelenks unter der Einwirkung einer äußeren Kraft. Die blauen Kästen sollen unbeweglich sein, der rote soll eine definierte Flexibilität haben. Dieses kann in Form einer Trajektorie als Vorlage oder Blaupause für die Optimierung eines Moleküls dienen.

eigentlich Ablaufs unterzogen würde.

Zusätzlich werden vorliegende Verfahren, Algorithmen und Programmpakete ständig weiter entwickelt und können für eine Verbesserung eingebunden werden. Beispielsweise für den verwendeten PCCA wurde kürzlich von Reuter et al. [102] eine generalisierte Version namens *G-PCCA+* veröffentlicht. In jedem Fall muss für die automatische Auswertung von vielen Trajektorien eine Anpassung der vorgestellten Parameter der ML-Algorithmen vorgenommen werden. Ebenfalls können einzelne ML-Algorithmen modifiziert oder komplett ausgetauscht werden. Es gibt eine große Anzahl an Clusteralgorithmen, welche eventuell unabhängig von der vorgestellten Bewertung der Trajektorien und dem daraus resultierenden Trainingssatz Verwendung finden können. Der Klassifikationsalgorithmus Random-Forest kann bezüglich Subattributsräumen und der zufälligen Auswahl an Instanzen während des Trainings sicherlich verbessert oder ganz ausgetauscht werden. Beispielsweise können künstliche Neuronale Netze mit Hilfe des Trainingssatzes angelernt werden. All dies wird das Resultat des vorgestellten Schemas verändern und kann mit großen Testsätzen auf diesem Weg robuster und generalisierbarer gemacht werden.

Ebenso kann die wiederholte Anwendung des gesamten Bewertungssystems nach einem ersten erfolgreichen Durchlauf auf ein Subset von Trajektorien erfolgen. Dies kann weitere Details und Muster in den Trajektorien herausarbeiten und gegebenenfalls helfen, das vorliegende molekulare System zu verstehen. Dazu wird die abschließende Methodik zur Bewertung des MSM dazu verwendet, die Trajektorien-Schar in Subgruppen zu sortieren, welche anschließend erneut mit dem automatischen Auswertungsprogramm analysiert werden. Zusätzlich können die aus den Subgruppen trainierten Klassifikationsalgorithmen auf weitere Subgruppen angewendet werden und daraus ein Modell über Diversifikation innerhalb der Subgruppen gebildet werden. Im einfachen Fall würden zunächst alle

Trajektorien, welche ein auffälliges strukturelles Merkmal besitzen (z.B. einen Bindungsbruch) in einer Subgruppe zusammengefasst. Die verbleibenden Trajektorien können dann in einem erneuten Programmdurchlauf auf weitere Merkmale hin untersucht werden, da der Klassifikator sich ebenfalls in dieser Subgruppe nur an der Trajektorie orientiert, welche die größte Divergenz beinhaltet. An dieser Stelle ist dies allerdings nicht mehr der zuvor beispielsweise separierte Bindungsbruch, sondern es können neue zuvor unauffällige Merkmale sein, welche nun die Divergenz des Systems beeinflussen. Dieses kann mit nur zwei metastabilen Zuständen folglich so weit betrieben werden, bis zu wenige Trajektorien für eine Aussage verbleiben oder strukturelle Unterschiede lediglich das Schwingungsmuster sind. All dies kann in einem vollautomatischen Zyklus vereint werden und mit einem definierten Abbruchkriterium allgemein anwendbar gemacht werden. Dafür wird ein Gütekriterium benötigt, welches zunächst noch unbestimmt ist, allerdings viele Möglichkeiten für zukünftige Arbeiten bietet.

Ein wichtiger Teil, der das Anwendungsgebiet erweitern würde, ist die Implementation des dynamischen Wachstums des analysierten Raums. Der naheliegende Weg ist, sobald ein randständiger Pixel besetzt ist, eine ganze Lage im OctTree hinzuzufügen und somit den Raum zu vergrößern. Dies ist in Abbildung 6.3 im zweidimensionalen Fall für den QuadTree skizziert. Der Pfad 3-3 ist ein Pixel am Außenrand des beschriebenen Bereichs.[2] Der Pixel mit dem Pfad 2-2 existiert, ist allerdings auf der gegenüberliegenden Seite der Fläche lokalisiert. Beim tatsächlichem Wechsel zu 2-2 würde in der aktuellen Implementation richtigerweise ein Fehler herausgegeben werden. In einer zukünftigen Ausführung würde, wie in Abbildung 6.4 gezeigt, eine komplette Lage im QuadTree oben hinzugefügt werden, in Grau dargestellt. Somit würde sich die mit dem erweiterten Baum beschreibbare Fläche vervierfachen. Der blau markierte Pfad 2-2 beschreibt den benachbarten Pixel zu 3-3, welcher hinzugekommen ist. Die Pfade müssen entsprechend erweitert werden von 3-3 zu 3-3-3 und von 2-2 zu 2-2-2. Die weiteren Algorithmen für die Berechnung der Superstrukturelemente müssten ebenfalls angepasst werden und gegebenenfalls eine Funktion für die Eliminierung von gänzlich unbesetzten Bereichen ergänzt werden. Auf diesem Weg kann die benötigte Rechenzeit in Grenzen gehalten werden, und das dynamische Gitter würde real mit dem untersuchten molekularen System wandern. Außerdem lassen sich weitere Superstrukturelemente implementieren, welche auf Basis der Sticks aufgebaut werden können. Beispielsweise die in Abbildung 6.2 rechts abgebildete Struktur als abgewinkeltes Element.

[2] In Abbildung 2.16 ist die verwendete räumliche Zuordnung im einem QuadTree gezeigt. Die Nachbarschaften sind im gezeigten Baum nicht unmittelbar ersichtlich.

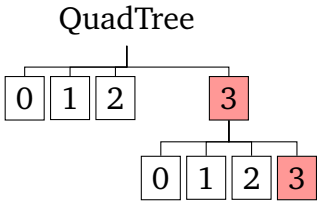


Abb. 6.3: Schematische Darstellung des rekursiven QuadTrees mit zwei Lagen; die markierten Pixel sind die besetzten Pixel bei unterschiedlicher Lage im QuadTree. Der gezeigte Pfad 3-3 ist ein randständiger Pixel im QuadTree.

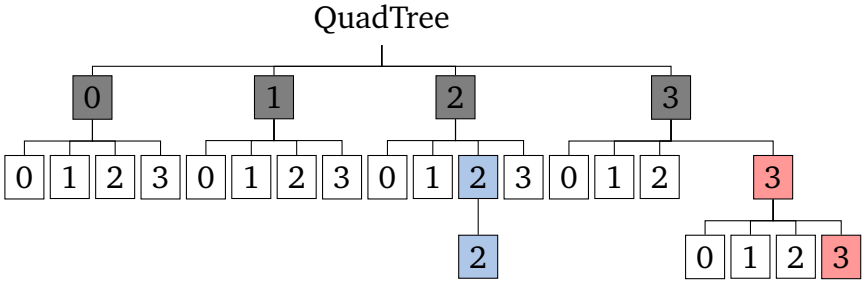


Abb. 6.4: Schematische Darstellung des erweiterten rekursiven Quadtrees mit drei Lagen; die markierten Pixel sind die besetzten Pixel bei unterschiedlicher Lage im QuadTree. Der besetzte Pixel (3)-3-3 ist nun nicht mehr außen und hat einen Nachbarn (2)-2-2.

Auch ist bereits eine Suche nach der längsten Molekülkette in einem Molekül implementiert, welche im resultierenden Programm keine direkte Verwendung findet. Die Analyse könnte an dieser Stelle auf diese längste Kette gestützt werden und somit z.B. dem Rückgrat eines Proteins als Basis dienen. Auf diesem Wege würde sich eine Alternative zu der allgemein üblichen Analyse der Cα-Atome bieten. Diese würde nicht statisch, sondern dynamisch während der Simulation durchgeführt werden und fundamentale Veränderungen in der strukturellen Anordnung auswerten. Der Beweis, dass 100 Å große Boxen, Abschnitt 5.5, analysiert werden können, ist noch kein unmittelbarer Leistungsnachweis, allerdings sind die Grenzen des Möglichen nicht erreicht. Für weitere natürliche und synthetische Polymere kann dieser Ansatz verständlicherweise ebenfalls angewendet werden.

Abkürzungsverzeichnis

MD Molekulare Dynamik

ML Maschinelles Lernen

MDS Multidimensionale Skalierung

HC Hierarchische Clustern

EM Expectation-Maximization

MSM Markov-State-Model

PCCA Perron Cluster Cluster Analysis

RMSD Root-mean-square deviation of atomic positions

UPGMA Unweighted Pair Group Method with Arithmetic mean

WPGMA Weighted Pair Group Method with Arithmetic mean

WPGMC Weighted Pair Group Method with Centroid averaging

UPGMC Unweighted Pair Group Method with Centroid

Literaturverzeichnis

[1] A. Warshel, M. Karplus, M. Levitt, *Angew. Chem. Int. Ed.* **1998**, *37*.

[2] J. Bentzien, R. P. Muller, J. Florián, A. Warshel, *J. Phys. Chem. B* **1998**, *102*, 2293–2301.

[3] C. A. Mack, *IEEE Trans. Semicond. Manuf.* **2011**, *24*, 202–207.

[4] C. M. Bishop, *Pattern Recognition and Machine Learning*, von *Information science and statistics*, Springer, New York, **2006**.

[5] Y. LeCun, Y. Bengio, G. Hinton, *Nature* **2015**, *521*, 436–444.

[6] G. Montavon, W. Samek, K.-R. Müller, *Digit. Signal Process.* **2018**, *73*, 1–15.

[7] K. Kambatla, G. Kollias, V. Kumar, A. Grama, *J. Parallel Distrib. Comput.* **2014**, *74*, 2561–2573.

[8] *Computing | CERN*, https://home.cern/about/computing, **2018**.

[9] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, K. Zieba, *ArXiv160407316 Cs* **2016**.

[10] C. E. Shannon, *Ann. Math. AI* **2000**, *28*, 27–30.

[11] E. Gibney, *Nature* **2016**, *529*, 445–446.

[12] M. Rupp, R. Ramakrishnan, O. A. von Lilienfeld, *J. Phys. Chem. Lett.* **2015**, *6*, 3309–3313.

[13] A. Rajan, P. L. Freddolino, K. Schulten, *PLoS ONE* **2010**, *5*, e9890.

[14] G. Pilania, C. Wang, X. Jiang, S. Rajasekaran, R. Ramprasad, *Sci. Rep.* **2013**, *3*, 1–6.

[15] T. Mueller, A. G. Kusne, R. Ramprasad, *Rev. Comput. Chem.* **2016**, *29*, 186–273.

[16] R. L. Melvin, R. C. Godwin, J. Xiao, W. G. Thompson, K. S. Berenhaut, F. R. Salsbury, *J. Chem. Theory Comput.* **2016**, *12*, 6130–6146.

[17] V. Botu, R. Ramprasad, *Int. J. Quantum Chem.* **2015**, *115*, 1074–1083.

[18] M. Rupp, O. A. von Lilienfeld, K. Burke, *ArXiv180602690 Phys.* **2018**.

[19] M. Brehm, B. Kirchner, *J. Chem. Inf. Model.* **2011**, *51*, 2007–2023.

[20] G. Bouvier, N. Desdouits, M. Ferber, A. Blondel, M. Nilges, *Bioinformatics* **2015**, *31*, 1490–1492.

[21] T. D. Romo, A. Grossfield, *Eng. Med. Biol. Soc. IEEE* **2009**, 2332–2335.

[22] A. V. Popov, Y. N. Vorobjev, D. O. Zharkov, *J. Comput. Chem.* **2013**, *34*, 319–325.

[23] J. C. Jeong, S. Jo, E. L. Wu, Y. Qi, V. Monje-Galvan, M. S. Yeom, L. Gorenstein, F. Chen, J. B. Klauda, W. Im, *J. Comput. Chem.* **2014**, *35*, 957–963.

[24] B. Keller, X. Daura, W. F. van Gunsteren, *J. Chem. Phys.* **2010**, *132*, 074110.

[25] N. Michaud-Agrawal, E. J. Denning, T. B. Woolf, O. Beckstein, *J. Comput. Chem.* **2011**, *32*, 2319–2327.

[26] P. I. Koukos, N. M. Glykos, *J. Comput. Chem.* **2013**, *34*, 2310–2312.

[27] P. Tavadze, G. Avendaño Franco, P. Ren, X. Wen, Y. Li, J. P. Lewis, *J. Am. Chem. Soc.* **2018**, *140*, 285–290.

[28] T. L. Jacobsen, M. S. Jørgensen, B. Hammer, *Phys. Rev. Lett.* **2018**, *120*, 026102.

[29] K. Hansen, G. Montavon, F. Biegler, S. Fazli, M. Rupp, M. Scheffler, O. A. von Lilienfeld, A. Tkatchenko, K.-R. Müller, *J. Chem. Theory Comput.* **2013**, *9*, 3404–3419.

[30] K. Hansen, F. Biegler, R. Ramakrishnan, W. Pronobis, O. A. von Lilienfeld, K.-R. Müller, A. Tkatchenko, *J. Phys. Chem. Lett.* **2015**, *6*, 2326–2331.

[31] F. A. Faber, L. Hutchison, B. Huang, J. Gilmer, S. S. Schoenholz, G. E. Dahl, O. Vinyals, S. Kearnes, P. F. Riley, O. A. von Lilienfeld, *J. Chem. Theory Comput.* **2017**, *13*, 5255–5264.

[32] N. O. Carstensen, *Phys. Chem. Chem. Phys.* **2013**, *15*, 15017.

[33] G. Granucci, M. Persico, A. Toniolo, *J. Chem. Phys.* **2001**, *114*, 10608–10615.

[34] G. Granucci, M. Persico, *J. Chem. Phys.* **2007**, *126*, 134114.

[35] G. Floß, G. Granucci, P. Saalfrank, *J. Chem. Phys.* **2012**, *137*, 234701.

[36] T. Raeker, N. O. Carstensen, B. Hartke, *J. Phys. Chem. A* **2012**, *116*, 11241–11248.

[37] N. O. Carstensen, PhD Thesis, CAU Kiel, **2014**.

[38] B. E. Husic, V. S. Pande, *J. Am. Chem. Soc.* **2018**, *140*, 2386–2396.

[39] V. S. Pande, K. Beauchamp, G. R. Bowman, *Methods* **2010**, *52*, 99–105.

[40] J. D. Chodera, N. Singhal, V. S. Pande, K. A. Dill, W. C. Swope, *J. Chem. Phys.* **2007**, *126*, 155101.0–155101.16.

[41] J. Lin, E. Keogh, L. Wei, S. Lonardi, *Data Min. Knowl. Discov.* **2007**, *15*, 107–144.

[42] P. Deuflhard, M. Weber, *Linear Algebra Its Appl.* **2005**, *398*, 161–184.

[43] M. K. Scherer, B. Trendelkamp-Schroer, F. Paul, G. Pérez-Hernández, M. Hoffmann, N. Plattner, C. Wehmeyer, J.-H. Prinz, F. Noé, *J. Chem. Theory Comput.* **2015**, *11*, 5525–5542.

[44] S. Röblitz, M. Weber, *Adv. Data Anal. Classif.* **2013**, *7*, 147–179.

[45] F. Noé, H. Wu, J.-H. Prinz, N. Plattner, *J. Chem. Phys.* **2013**, *139*, 184114.

[46] Wikipedia, *Dreiecksungleichung — Wikipedia, Die Freie Enzyklopädie*, https://de.wikipedia.org/w/index.php?title=Dreiecksungleichung&oldid=168986384, **2017**.

[47] M. Dobrowolski, *Angewandte Funktionalanalysis: Funktionalanalysis, Sobolev-Räume Und Elliptische Differentialgleichungen 2. Ed.*, Springer, Berlin, **2010**.

[48] L. Van Der Maaten, E. Postma, J. Van den Herik, *J Mach Learn Res* **2009**, *10*, 66–71.

[49] I. Borg, P. J. F. Groenen, *Modern Multidimensional Scaling: Theory and Applications*, Springer Science & Business Media, **2005**.

[50] B. Hartke, *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2011**, *1*, 879–887.

[51] T. Weise, *Global Optimization Algorithms - Theory and Application 3. Ed.*, **2011**.

[52] D. E. Goldberg, *Genetic Algorithms*, Pearson Education India, **2006**.

[53] D. Livingstone (Ed.), *Artificial Neural Networks: Methods and Applications*, von *Methods in molecular biology*, Humana Press, Totowa, NJ, **2008**.

[54] I. H. Witten, E. Frank, M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques 3. Ed.*, von *Morgan Kaufmann series in data management systems*, Morgan Kaufmann, Burlington, MA, **2011**.

[55] W. McKinney, *Proc. 9th Python Sci. Conf.* **2010**, 51–56.

[56] A. K. Jain, M. N. Murty, P. J. Flynn, *ACM Comput. Surv. CSUR* **1999**, *31*, 264–323.

[57] J. Wu, H. Xiong, J. Chen, *Neurocomputing* **2009**, *72*, 2319–2330.

[58] R. Tibshirani, G. Walther, T. Hastie, *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2001**, *63*, 411–423.

[59] A. P. Dempster, N. M. Laird, D. B. Rubin, *J. R. Stat. Soc. Ser. B Methodol.* **1977**, *39*, 1–38.

[60] T. Benaglia, D. Chauveau, D. R. Hunter, D. Young, *J. Stat. Softw.* **2009**, *32*.

[61] R. Xu, D. WunschII, *IEEE Trans. Neural Netw.* **2005**, *16*, 645–678.

[62] J. Shao, S. W. Tanner, N. Thompson, T. E. Cheatham, *J. Chem. Theory Comput.* **2007**, *3*, 2312–2334.

[63] G. Brock, V. Pihur, S. Datta, S. Datta, *J. Stat. Softw.* **2008**, *25*, 1–22.

[64] R. Suzuki, H. Shimodaira, *Bioinformatics* **2006**, *22*, 1540–1542.

[65] T. M. Abramyan, J. A. Snyder, A. A. Thyparambil, S. J. Stuart, R. A. Latour, *J. Comput. Chem.* **2016**.

[66] W.-Y. Loh, *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2011**, *1*, 14–23.

[67] Wikipedia, *Gini-Koeffizient*, https://de.wikipedia.org/w/index.php?title=Gini-Koeffizient&oldid=172017208, **2017**.

[68] L. Breiman, J. Friedman, C. J. Stone, R. A. Olshen, *Classification and Regression Trees*, CRC press, **1984**.

[69] L. Breiman, *Classification and Regression Trees*, Routledge, **2017**.

[70] B. Efron, R. Tibshirani **1985**, 35.

[71] L. Breiman, *Mach. Learn.* **1996**, *24*, 123–140.

[72] H. Chauhan, A. Chauhan, *Int. J. Sci. Res. Publ.* **2013**, *3*, 1–3.

[73] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, *ACM SIGKDD Explor. Newsl.* **2009**, *11*, 10–18.

[74] S. Drazin, M. Montag, *Mach. Learn.-Proj. II Univ. Miami* **2012**, 1–3.

[75] J. Amanatides, A. Woo, *Eurographics* **1987**, *87*, 10.

[76] B. B. Chaudhuri, *Pattern Anal. Mach. Intell. IEEE Trans. On* **1985**, *6*, 652–661.

[77] Z. Lari, A. Habib, E. Kwak, *ISPRS Workshop Laser Scanning* **2011**, 29–31.

[78] B. Wu, B. Yu, W. Yue, S. Shu, W. Tan, C. Hu, Y. Huang, J. Wu, H. Liu, *Remote Sens.* **2013**, *5*, 584–611.

[79] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, W. Burgard, *Auton. Robots* **2013**, *34*, 189–206.

[80] S. Frick, TC-2 Praktikum, CAU Kiel, **2009**.

[81] F. Spenke, F3-Praktikum, CAU Kiel, **2014**.

[82] C. Witt, F3-Praktikum, CAU Kiel, **2015**.

[83] C. Steinbeck, Y. Han, S. Kuhn, O. Horlacher, E. Luttmann, E. Willighagen, *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 493–500.

[84] *Jmol: An Open-Source Browser-Based HTML5 Viewer and Stand-Alone Java Viewer for Chemical Structures in 3D*, http://www.jmol.org/, **2018**.

[85] T. Raeker, PhD Thesis, CAU Kiel, **2018**.

[86] J. Papon, A. Abramov, M. Schoeler, F. Worgotter, *Comput. Vis. Pattern Recognit. Conf. IEEE* **2013**, 2027–2034.

[87] F. T. Marchese, J. Mercado, Y. Pan, *Inf. Vis. Seventh Int. Conf. IEEE* **2003**, 252–257.

[88] O. A. von Lilienfeld, *Angew. Chem. Int. Ed.* **2018**, *57*, 4164–4169.

[89] J. Müller, PhD Thesis, CAU Kiel, **2017**.

[90] S. Erbas-Cakmak, D. A. Leigh, C. T. McTernan, A. L. Nussbaumer, *Chem. Rev.* **2015**, *115*, 10081–10206.

[91] T. Fink, H. Bruggesser, J.-L. Reymond, *Angew. Chem. Int. Ed.* **2005**, *44*, 1504–1508.

[92] T. Fink, J.-L. Reymond, *J. Chem. Inf. Model.* **2007**, *47*, 342–353.

[93] L. Ruddigkeit, R. van Deursen, L. C. Blum, J.-L. Reymond, *J. Chem. Inf. Model.* **2012**, *52*, 2864–2875.

[94] L. Ruddigkeit, L. C. Blum, J.-L. Reymond, *J. Chem. Inf. Model.* **2013**, *53*, 56–65.

[95] L. C. Blum, J.-L. Reymond, *J. Am. Chem. Soc.* **2009**, *131*, 8732–8733.

[96] L. C. Blum, R. van Deursen, S. Bertrand, M. Mayer, J. J. Bürgi, D. Bertrand, J.-L. Reymond, *J. Chem. Inf. Model.* **2011**, *51*, 3105–3112.

[97] E. Luethi, K. T. Nguyen, M. Bürzle, L. C. Blum, Y. Suzuki, M. Hediger, J.-L. Reymond, *J. Med. Chem.* **2010**, *53*, 7236–7250.

[98] R. Visini, M. Awale, J.-L. Reymond, *J. Chem. Inf. Model.* **2017**, *57*, 700–709.

[99] J. Ponder, F. Richards, *J Comput Chem* **1987**, *8*, 1016–1024.

[100] T. Raeker, B. Jansen, D. Behrens, B. Hartke, *J. Comput. Chem.* **2018**, *39*, 1433–1443.

[101] W. Jin, R. Barzilay, T. Jaakkola, *ArXiv180204364 Cs Stat* **2018**.

[102] B. Reuter, M. Weber, K. Fackeldey, S. Röblitz, M. E. Garcia, *J. Chem. Theory Comput.* **2018**.

[103] R. Munroe, *Xkcd: Machine Learning*, https://xkcd.com/1838/, **2017**.

Anhang A

QuadTree und OctTree

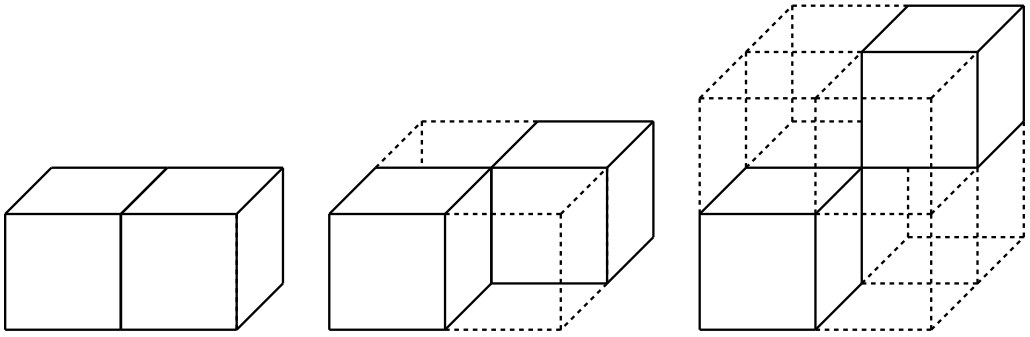


Abb. A.1: Nachbarschaften der Voxel in OctTree. Die einfachen flächen-benachbarten Voxel (links) bilden die Basis für die Sticks. Die kanten-verknüpften Voxel (mitte) und die ecken-verknüpften Voxel (rechts) bilden die Grundlage für die Pseudo-Sticks.

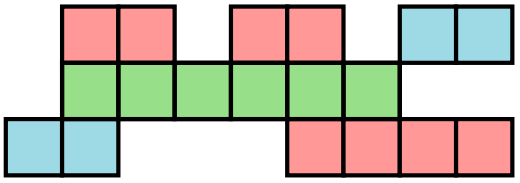


Abb. A.2: Der Stick in Grün wird als Basis für einen Pseudo-Stick verwendet. Die getrennte Suche nach links und rechts fügt die blauen Sticks hinzu. Die Rot eingefärbten Sticks werden nach den vorliegenden Regeln nicht hinzugefügt.

Abbildungen

Tabellen

Danke,
Sabine und Konrad für die Möglichkeit,
Bernd für das Vertrauen und die schöne Zeit der Forschung,
Mark für die Anregungen während der Promotion und beim Abfassen der Arbeit,
Tim dessen Quantenausbeuten immer wieder den Kopf hinhalten mussten,
an den ehemaligen und aktuellen AK Hartke für die schöne Zeit,
Dr. Matze im fernen Kanada,
und zu guter Letzt
Julia.

Abb. 6.3: Machine Learning[103]

Versicherung der Versicherung

Hiermit erkläre ich, David Siebler, die vorliegende Dissertation selbständig und ausschließlich unter Zuhilfenahme der angegebenen Quellen und Hilfsmittel angefertigt zu haben. Inhalt und Form der Arbeit habe ich, abgesehen von Beratung durch meinen Betreuer, Prof. Dr. Bernd Hartke, eigenständig erarbeitet und verfasst. Bei der Entstehung der Arbeit wurden die Regeln guter wissenschaftlicher Praxis der Deutschen Forschungsgemeinschaft eingehalten. Weder die gesamte Arbeit noch Teile hiervon wurden an anderer Stelle im Rahmen eines Prüfungsverfahrens eingereicht. Diese Dissertation ist mein erster Promotionsversuch.

Kiel, 24.07.2018 David Siebler