KAPITEL 4

Machine Learning mit Molekülen

In diesem Kapitel werden die Grundlagen der Anwendung von Machine Learning auf molekularen Daten vermittelt. Aber zunächst wollen wir kurz ansprechen, warum es sich lohnt, sich mit molekularem Machine Learning zu befassen. Ein Großteil der modernen Materialwissenschaften und der modernen Chemie beruht auf der Notwendigkeit, neue Moleküle mit den gewünschten Eigenschaften zu entwickeln. Obwohl bedeutende wissenschaftliche Arbeiten zu neuen Designstrategien geführt haben, ist manchmal noch viel Zufall erforderlich, um interessante Moleküle zu gestalten. Der Traum des molekularen Machine Learning ist es, solche willkürlichen Experimente durch gelenkte Suchen zu ersetzen. Hierbei können durch Machine Learning erzeugte Prädiktoren vorschlagen, welche der neuen Moleküle die gewünschten Eigenschaften haben könnten. Derartig genaue Prädiktoren könnten die Erschaffung grundlegend neuer Materialien und Chemikalien mit nützlichen Merkmalen ermöglichen.

Dieser Traum ist unwiderstehlich, aber wo fangen wir an? Der erste Schritt besteht darin, technische Methoden zur Umwandlung von Molekülen in Zahlenvektoren zu entwickeln, die dann an Lernalgorithmen übergeben werden können. Solche Methoden werden *molekulare Featurizations* genannt. Wir werden einige davon in diesem Kapitel behandeln und weitere im nächsten Kapitel. Moleküle sind komplexe Objekte, für die Forscher eine Vielzahl verschiedener Techniken entwickelt haben, um sie zu charakterisieren. Darunter sind chemische Deskriptorvektoren, 2-D-Graphendarstellungen, 3-D-Gitterdarstellungen, Orbitalbasisfunktionsdarstellungen usw.

Aus einem charakterisierten Molekül muss nach wie vor gelernt werden. Wir werden einige Algorithmen für das Lernen von Funktionen anhand von Molekülen untersuchen, einschließlich einfacher, vollständig verbundener Netze sowie ausgefeilterer Methoden wie Graph Convolutions. Wir werden auch einige Einschränkungen der Graph Convolutions erläutern und was wir von ihnen erwarten sollten und was nicht. Wir beenden das Kapitel mit einer Fallstudie zum molekularen Machine Learning anhand eines interessanten Datensatzes.

Was ist ein Molekül?

Bevor wir uns näher mit molekularem Machine Learning befassen, wäre es hilfreich, zu wissen, was genau ein Molekül ist. Diese Frage klingt ein wenig albern, da Moleküle wie H_2O und CO_2 bereits kleinen Kindern bekannt sind. Ist die Antwort nicht offensichtlich? Tatsache ist jedoch, dass wir lange Zeit nicht wussten, dass Moleküle überhaupt existieren. Betrachten wir folgendes Gedankenexperiment: Wie würden Sie einen skeptischen Außerirdischen davon überzeugen, dass es Objekte gibt, die Moleküle genannt werden? Die Antwort wird recht aufwendig. Es könnte durchaus sein, dass Sie ein Massenspektrometer hervorholen müssen!

Massenspektrometrie

Herauszufinden, welche Moleküle in einer bestimmten Probe vorhanden sind, kann durchaus schwierig sein. Die dafür derzeit beliebteste Methode ist die Massenspektrometrie. Die Grundidee der Massenspektrometrie besteht darin, eine Probe mit Elektronen zu beschießen. Durch diesen Beschuss zersplittern die Moleküle in Fragmente. Typischerweise *ionisieren* diese Bruchstücke – das heißt, sie nehmen Elektronen auf oder verlieren diese, um sich aufzuladen. Diese geladenen Fragmente werden von einem elektrischen Feld angetrieben, das sie anhand ihres Masse-zu-Ladung-Verhältnisses trennt. Die Streuung erkannter geladener Bruchstücke wird *Spektrum* genannt. In Abbildung 4-1 wird dieser Prozess illustriert. Aus der Sammlung erkannter Fragmente lassen sich häufig die exakten Moleküle bestimmen, die sich in der ursprünglichen Probe befanden. Dieser Vorgang ist jedoch immer noch verlustbehaftet und schwierig. Viele Wissenschaftler erforschen derzeit, wie die Massenspektrometrie durch Deep-Learning-Algorithmen verbessert werden kann, um die Identifizierung der ursprünglichen Moleküle aus dem erkannten geladenen Spektrum zu erleichtern.

Beachten Sie die Komplexität dieser Erkenntnis! Moleküle sind komplizierte Objekte, deren genaue Bestimmung schwierig ist.

Nehmen wir für den Anfang an, ein Molekül sei eine Gruppe von Atomen, die durch physikalische Kräfte miteinander verbunden sind. Ein Molekül ist die kleinste Grundeinheit einer chemischen Verbindung, die an einer chemischen Reaktion beteiligt sein kann. Atome eines Moleküls sind durch *chemische Verbindungen* miteinander verbunden, die sie zusammenhalten und ihre Bewegung zueinander einschränken. Moleküle gibt es in einer Vielzahl von Größen, bestehend aus nur wenigen Atomen bis hin zu mehreren Tausend. Abbildung 4-2 zeigt eine einfache Darstellung eines Moleküls.

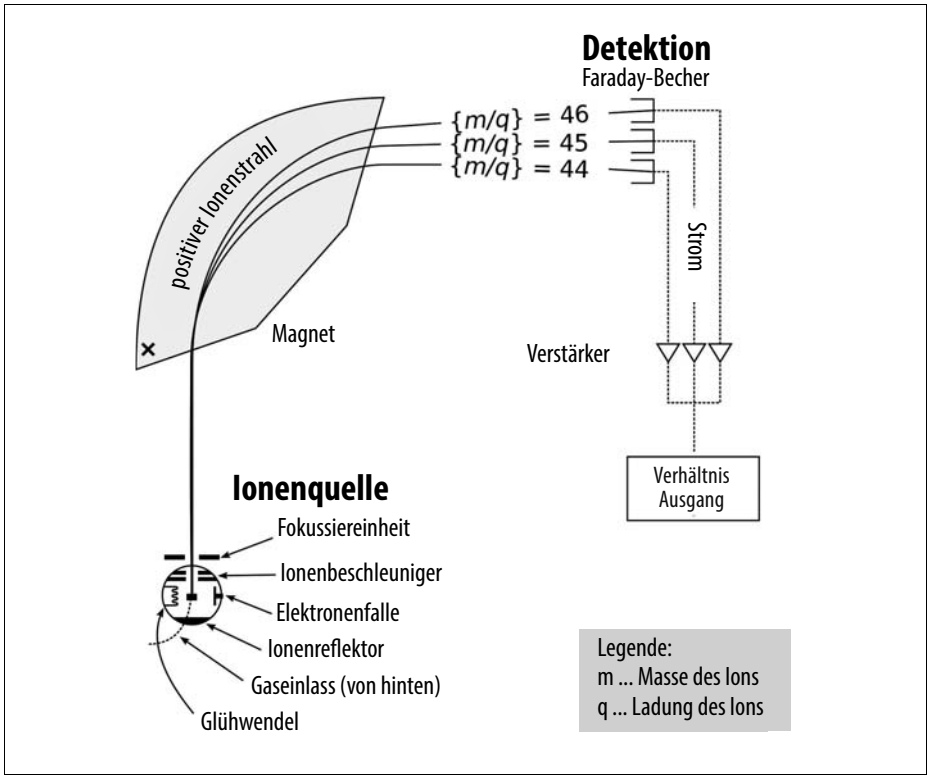

Abbildung 4-1: Schematische Skizze eines Massenspektrometers. (Quelle: Wikimedia, https://commons.wikimedia.org/wiki/File:Mass_Spectrometer_Schematic.svg)

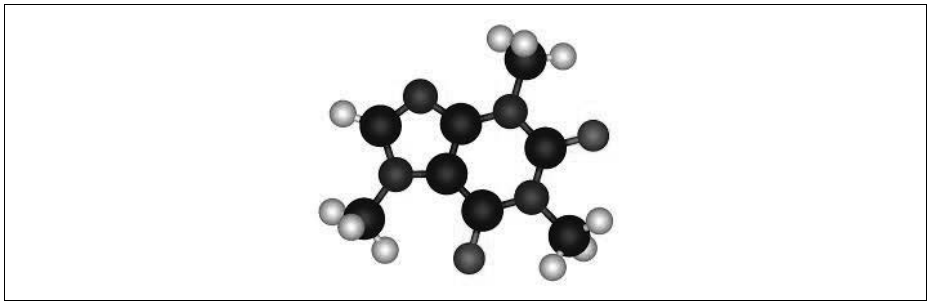

Abbildung 4-2: Eine einfache Darstellung eines Koffeinmoleküls als Kugel-Stab-Modell. Atome werden als farbige Kugeln dargestellt (Schwarz für Kohlenstoff, Rot für Sauerstoff, Blau für Stickstoff, Weiß für Wasserstoff), die mit Stäbchen verbunden sind, die chemische Bindungen symbolisieren.

Nach dieser grundlegenden Beschreibung werden wir uns in den nächsten Abschnitten eingehender mit verschiedenen Aspekten der molekularen Chemie befassen. Es ist nicht notwendig, dass Sie all diese Konzepte bereits nach dem ersten

Lesen des Kapitels verinnerlicht haben, aber es kann hilfreich sein, auf einige Grundkenntnisse der Chemie zurückgreifen zu können.

Moleküle sind dynamische Quantenobjekte

Wir haben gerade eine vereinfachte Beschreibung von Molekülen anhand von Atomen und Bindungen gegeben. Es ist sehr wichtig, im Hinterkopf zu behalten, dass in jedem Molekül viel mehr vor sich geht. Zum einen sind Moleküle dynamische Objekte, sodass sich alle Atome innerhalb eines bestimmten Moleküls schnell aufeinander zubewegen. Die Bindungen selbst dehnen sich aus und ziehen sich wieder zusammen, sie können in ihrer Länge schnell schwanken. Es ist durchaus üblich, dass Atome von Molekülen abbrechen und sich wieder verbinden. Wir werden mehr über die Dynamik von Molekülen erfahren, wenn wir die molekularen Konformationen besprechen.

Noch seltsamer ist, dass Moleküle Quanten sind. Es gibt viele Möglichkeiten, zu sagen, dass eine Struktur ein Quant ist, aber einfach ausgedrückt ist es wichtig, zu beachten, dass »Atome« und »Bindungen« viel weniger gut definiert sind, als dies ein einfaches Kugel-Stab-Modell implizieren könnte. Die Definitionen hierfür sind nicht eindeutig. Es ist nicht wichtig, dass Sie diese Komplexitäten zum jetzigen Zeitpunkt erfassen können, aber beachten Sie, dass unsere Darstellung von Molekülen sehr vage ist. Dies kann praktische Auswirkungen haben, da je nach Lernaufgabe Moleküle möglicherweise unterschiedlich beschrieben werden müssen.

Was sind molekulare Bindungen?

Es mag eine Weile her sein, seit Sie sich mit den Grundlagen der Chemie befasst haben. Daher werden wir immer wieder Zeit darauf verwenden, grundlegende chemische Konzepte aufzufrischen. Die grundlegendste Frage lautet: Was ist eine chemische Bindung?

Die Moleküle, die den Alltag ausmachen, bestehen aus Atomen, oftmals sehr vielen davon. Diese Atome sind durch chemische Bindungen miteinander verbunden. Diese Bindungen »kleben« Atome regelrecht durch ihre gemeinsamen Elektronen zusammen. Es gibt viele verschiedene Arten molekularer Bindungen, einschließlich kovalenter Bindungen, und verschiedene Arten nicht kovalenter Bindungen.

Kovalente Bindungen

Bei kovalenten Bindungen werden Elektronen zwischen zwei Atomen geteilt, sodass die gleichen Elektronen Zeit um beide Atome verbringen (siehe Abbildung 4-3). Grundsätzlich sind kovalente Bindungen die stärksten chemischen Bindungen. Sie entstehen und zerbrechen bei chemischen Reaktionen. Kovalente Bindungen sind in der Regel sehr stabil: Sobald sie sich gebildet haben, braucht es sehr viel Energie, um sie wieder zu lösen, sodass die Atome sehr lange gebunden bleiben können. Aus diesem Grund verhalten sich Moleküle wie individuelle Objekte anstatt wie lose Ansammlungen nicht verwandter Atome. Tatsächlich werden Moleküle durch

kovalente Bindungen definiert: Ein Molekül ist eine Ansammlung von Atomen, die durch kovalente Bindungen verbunden sind.

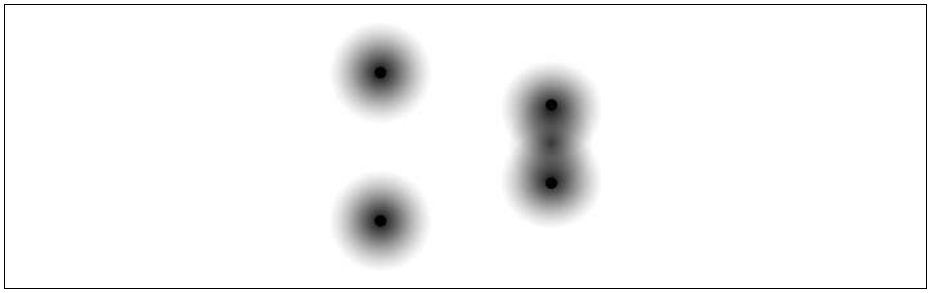

Abbildung 4-3: Links: zwei Atomkerne, jeder umgeben von einer Elektronenwolke. Rechts: Nähern die Atome sich an, verbringen die Elektronen mehr Zeit im Raum zwischen den Kernen. Dies zieht die Kerne zusammen und bildet eine kovalente Bindung zwischen den Atomen.

Nicht kovalente Bindungen

Bei nicht kovalenten Bindungen teilen sich Atome Elektronen nicht direkt, aber sie bilden schwache elektromagnetische Wechselwirkungen. Da sie nicht so stark sind wie kovalente Bindungen, sind sie kurzlebiger, zerfallen laufend und bilden sich neu. Nicht kovalente Bindungen »definieren« Moleküle nicht auf die gleiche Weise wie kovalente Bindungen, aber sie haben eine große Auswirkung auf die Formen, die Moleküle annehmen können, und die Art, wie sich verschiedene Moleküle miteinander verbinden.

»Nicht kovalente Bindungen« ist ein Oberbegriff, der verschiedene Arten von Interaktionen abdeckt. Beispiele nicht kovalenter Bindungen sind Wasserstoffbrücken (siehe Abbildung 4-4), Ionenbrücken, π-π-Wechselwirkungen usw. Die hier genannten Wechselwirkungen spielen häufig eine entscheidende Rolle bei der Arzneimittelentwicklung, da die meisten Wirkstoffe durch nicht kovalente Wechselwirkungen mit biologischen Molekülen im menschlichen Körper interagieren.

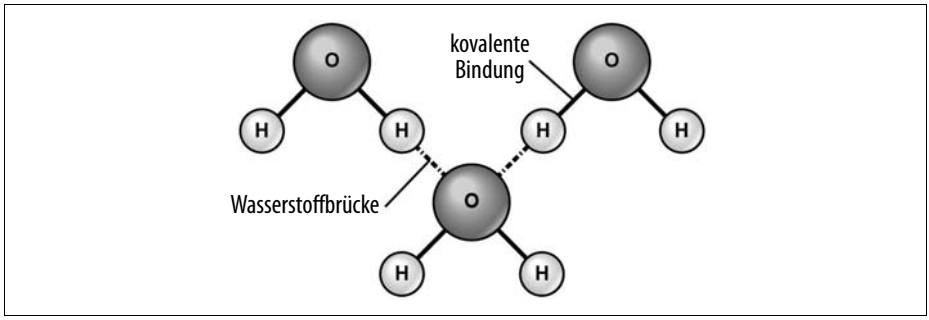

Abbildung 4-4: Wassermoleküle bilden starke Wasserstoffbrücken zwischen Wasserstoff und Sauerstoff an benachbarten Molekülen aus. Ein solides Netzwerk von Wasserstoffbrücken trägt zum Teil zur Stärke des Wassers als Lösungsmittel bei. (Quelle: Wikimedia, https://commons.wikimedia.org/wiki/File:SimpleBayesNet.svg)

Wir kommen an anderen Stellen des Buchs wieder auf diese Bindungsarten zurück. In diesem Kapitel werden wir uns hauptsächlich mit kovalenten Bindungen befassen. Nicht kovalente Wechselwirkungen werden deutlich wichtiger, wenn wir damit beginnen, uns mit biophysikalischen Deep-Learning-Modellen auseinanderzusetzen.

Molekülgraphen

Ein *Graph* ist eine mathematische Datenstruktur, die aus *Knoten* besteht, die durch *Kanten* (siehe Abbildung 4-5) verbunden werden. Graphen sind äußerst nützliche Abstraktionen in der Informatik. Tatsächlich gibt es einen ganzen Zweig der Mathematik, der als Graphentheorie bezeichnet wird und sich zum Ziel gesetzt hat, die Eigenschaften von Graphen zu verstehen sowie Wege zu finden, diese zu manipulieren und zu analysieren. Graphen werden für die Beschreibung von allem genutzt, von den Rechnern, aus denen ein Netzwerk besteht, über Pixel, die sich zu einem Bild zusammensetzen, bis hin zu Schauspielern, die in Filmen mit Kevin Bacon mitgewirkt haben.

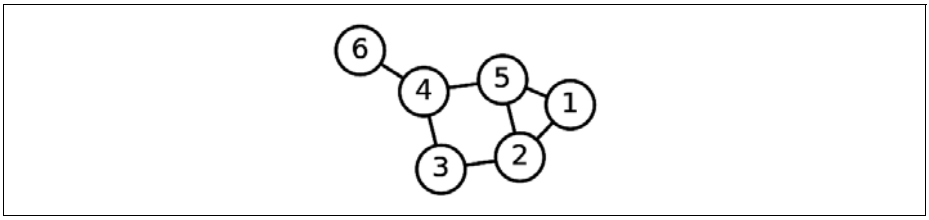

Abbildung 4-5: Beispiel eines mathematischen Graphen mit sechs Knoten, der durch Kanten verbunden wird. (Quelle: Wikimedia, https://commons.wikimedia.org/wiki/File:6n-graf.svg)

Wichtig ist, dass Moleküle auch als Graphen betrachtet werden können (siehe Abbildung 4-6). In dieser Beschreibung stellen die Knoten Atome im Graphen dar, während die Kanten die chemischen Bindungen abbilden. Jedes Molekül kann in einen entsprechenden Molekülgraphen umgewandelt werden.

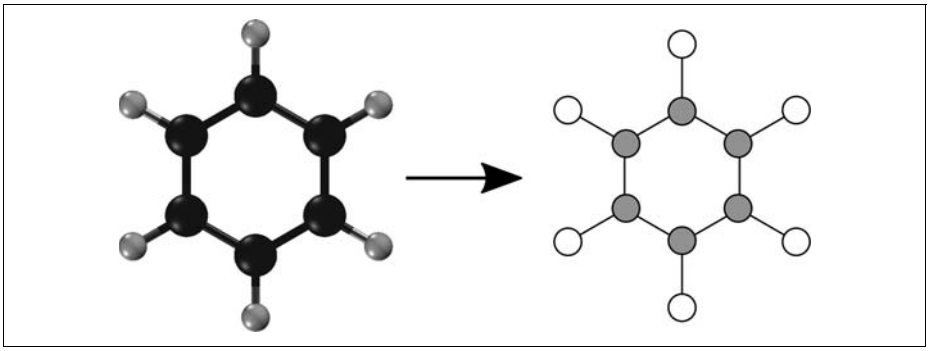

Abbildung 4-6: Ein Beispiel für die Umwandlung eines Benzolmoleküls in einen Molekülgraphen. Beachten Sie, dass Atome in Knoten und chemische Bindungen in Kanten umgewandelt werden.

Im restlichen Kapitel werden wir wiederholt Moleküle in Graphen umwandeln, um sie zu analysieren und Vorhersagen zu treffen.

Molekulare Konformationen

Ein Molekülgraph beschreibt die in einem Molekül enthaltenen Atome und deren Bindungen untereinander. Aber es gibt noch eine weitere Sache, die wir wissen müssen: die räumliche Anordnung der Atome, die als *Konformation* des Moleküls bezeichnet wird.

Atome, Bindungen und Konformationen stehen in Beziehung zueinander. Sind zwei Atome kovalent gebunden, kann dies den Abstand zwischen ihnen festlegen und mögliche Konformationen stark einschränken. Die Winkel, die durch drei oder vier gebundene Atome gebildet werden, sind ebenfalls oft beschränkt. Manchmal gibt es ganze Atomcluster, die völlig starr sind und sich alle als eine Einheit bewegen. Andere Molekülteile wiederum sind flexibel und ermöglichen den Atomen, sich relativ zueinander zu bewegen. Zum Beispiel erlauben viele (aber nicht alle) kovalente Bindungen den Atomgruppen, die sie verbinden, sich frei um die Achse der Bindung zu drehen. Dadurch kann das Molekül viele verschiedene Konformationen annehmen.

Abbildung 4-7 zeigt ein sehr beliebtes Molekül: Saccharose, auch als Haushaltszucker bekannt. Es wird sowohl als chemische 2-D-Struktur als auch als 3-D-Konformation dargestellt. Saccharose besteht aus zwei miteinander verbundenen Ringen. Jeder Ring ist ziemlich starr, sodass sich seine Form im Laufe der Zeit kaum ändert. Aber das Bindeglied, das sie verbindet, ist deutlich flexibler, sodass sich die Ringe relativ zueinander bewegen können.

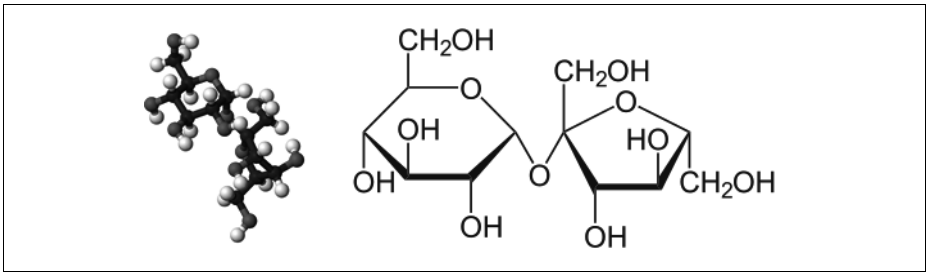

Abbildung 4-7: Saccharose, dargestellt als 3-D-Konformation und chemische 2-D-Struktur. Bilder aus Wikimedia (https://commons.wikimedia.org/wiki/File:Sucrose-3D-balls.png) und Wikipedia (https://en.wikipedia.org/wiki/File:Saccharose2.svg).

Je größer die Moleküle werden, desto mehr mögliche Konformationen können sie annehmen. Große Makromoleküle wie Proteine (siehe Abbildung 4-8) benötigen zur Untersuchung möglicher Konformationen derzeit sehr kostspielige Simulationen.

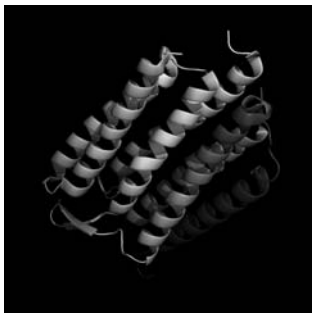

Abbildung 4-8: Eine in 3-D gerenderte Konformation von Bacteriorhodopsin (zur Erfassung von Lichtenergie). Proteinkonformationen sind besonders komplex und enthalten mehrere geometrische 3-D-Motive. Sie erinnern daran, dass Moleküle zusätzlich zu ihren chemischen Formeln auch eine Geometrie aufweisen. (Quelle: Wikimedia, https://upload.wikimedia.org/wikipedia/commons/thumb/d/dd/1M0K.png/480px-1M0K.png)

Chiralität von Molekülen

Einige Moleküle (darunter viele Medikamente) liegen in zwei Formen vor, die sich spiegeln. Das wird *Chiralität* genannt. Wie in Abbildung 4-9 dargestellt, hat ein chirales Molekül beides, eine »rechtshändige« Form (auch bekannt als »R«-Form) und eine »linkshändige« Form (auch bekannt als »S«-Form).

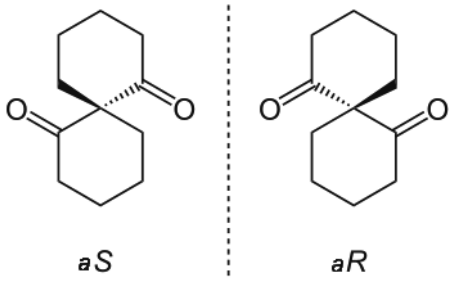

Abbildung 4-9: Axiale Chiralität einer Spiroverbindung (einer Verbindung aus zwei oder mehr miteinander verbundenen Ringen). Beachten Sie, dass die beiden chiralen Varianten jeweils mit »R« und »S« gekennzeichnet sind. Diese Konvention ist in der chemischen Literatur sehr verbreitet.

Chiralität ist sehr wichtig und sowohl für Laborchemiker als auch für Cheminformatiker eine Quelle großer Frustration. Zunächst einmal unterscheiden die chemischen Reaktionen, die chirale Moleküle produzieren, häufig nicht zwischen den Formen und produzieren beide Chiralitäten in gleichen Mengen. (Diese Produkte werden *Razemat* genannt.) Will man also nur eine der beiden Formen haben, wird der Herstellungsprozess schnell komplizierter. Des Weiteren sind diverse physikalische Eigenschaften für beide Chiralitäten identisch, sodass viele Experimente

nicht zwischen den chiralen Formen eines bestimmten Moleküls unterscheiden können. Gleiches gilt für Rechenmodelle. Beispielsweise haben beide Chiralitäten identische Molekülgraphen, sodass jedes Machine-Learning-Modell, das nur auf dem Molekülgraphen basiert, nicht zwischen ihnen unterscheiden kann.

Das wäre unerheblich, würden sich die beiden Formen in der Praxis identisch verhalten. Aber dies ist häufig nicht der Fall. Die beiden chiralen Formen eines Medikaments können durchaus an völlig verschiedene Proteine binden und im menschlichen Körper sehr unterschiedliche Wirkungen haben. In vielen Fällen hat nur eine Form eines Arzneimittels den gewünschten therapeutischen Effekt, während die andere Form nur Nebenwirkungen erzeugt, ohne Nutzen zu haben.

Ein konkretes Beispiel für die unterschiedliche Wirkung von chiralen Verbindungen ist das Medikament Thalidomid, das in den 1950er- und 1960er-Jahren als Beruhigungsmittel verschrieben wurde. Dieses Medikament war später rezeptfrei erhältlich, um Übelkeit und morgendliche Übelkeit bei Schwangerschaften zu behandeln. Die R-Form von Thalidomid ist ein wirksames Sedativum, während die S-Form fruchtschädigend ist und schwere Geburtsfehler verursacht. Diese Schwierigkeiten werden weiter verstärkt, da Thalidomid im Körper zwischen den beiden Formen wechselt bzw. racemisiert.

Featurization eines Moleküls

Wie können uns diese beschriebenen Grundlagen der Chemie dabei helfen, Merkmale für Moleküle zu entwickeln? Um Machine Learning mit Molekülen durchzuführen, müssen wir sie in Merkmalsvektoren umwandeln, die als Eingabe in Modellen verwendet werden können. In diesem Abschnitt stellen wir das DeepChem-Submodul `dc.feat` vor und erklären, wie es verwendet wird, um Moleküle auf verschiedene Arten zu charakterisieren.

SMILES-Strings und RDKit

SMILES ist eine beliebte Methode, um Moleküle als Zeichenketten anzugeben. Die Abkürzung steht für *Simplified Molecular-Input Line-Entry System*. Ein SMILES-String beschreibt die Atome und die chemischen Bindungen eines Moleküls auf eine Weise, die für Chemiker sowohl knapp als auch intuitiv ist. Für Nichtchemiker sehen diese Zeichenketten in der Regel wie sinnlose Muster aus zufälligen Zeichen aus. Zum Beispiel beschreibt »OCCc1c(C)[n+](cs1)Cc2cnc(C)nc2N« den wichtigen Nährstoff Thiamin, auch als Vitamin B1 bekannt.

DeepChem verwendet SMILES-Strings als Format für die Darstellung von Molekülen in Datensätzen. Es gibt einige Deep-Learning-Modelle, die SMILES-Strings direkt als Eingabe akzeptieren und versuchen, sinnvolle Merkmale in der Textdarstellung zu identifizieren. Viel öfter jedoch muss die Zeichenfolge zuerst in eine andere Darstellung konvertiert werden, die besser für die jeweilige Aufgabenstellung geeignet ist.

DeepChem ist auf ein anderes Open-Source-Paket der Cheminformatik, RDKit, angewiesen, um sich die Handhabung der Moleküle zu erleichtern. RDKit stellt viele Funktionen für die Arbeit mit SMILES-Strings bereit. Es spielt eine zentrale Rolle bei der Konvertierung der Strings in Molekülgraphen und andere, unten beschriebene Darstellungen.

Konnektivitäts-Fingerprints

Chemische Fingerabdrücke sind Vektoren, die aus Einsen und Nullen bestehen und das Vorhandensein oder Fehlen spezifischer Merkmale eines Moleküls anzeigen. Konnektivitäts-Fingerprints (ECFPs) sind eine Klasse von Featurizations, die mehrere nützliche Funktionen kombinieren. Sie wandeln Moleküle beliebiger Größe in Vektoren fester Länge um. Das ist wichtig, da viele Modelle nur mit Eingaben gleicher Größe arbeiten können. Mit ECFPs können Moleküle unterschiedlicher Größe mit dem gleichen Modell verwendet werden. ECFPs sind auch sehr einfach zu vergleichen. So können die entsprechenden Elemente anhand der Fingerabdrücke zweier Moleküle verglichen werden. Je mehr Elemente übereinstimmen, desto ähnlicher sind die Moleküle. Außerdem sind ECFPs schnell zu berechnen.

Jedes Element des Fingerabdruckvektors zeigt das Vorhandensein oder Fehlen eines bestimmten Molekülmerkmals an, das durch eine lokale Anordnung der Atome definiert ist. Der Algorithmus beginnt damit, jedes Atom einzeln zu prüfen und einige seiner Eigenschaften zu betrachten: sein Element, die Anzahl der von ihm gebildeten kovalenten Bindungen usw. Jede spezifische Kombination dieser Eigenschaften ist ein Merkmal, und die entsprechenden Elemente des Vektors werden auf 1 gesetzt, um ihre Anwesenheit anzuzeigen. Der Algorithmus kombiniert dann von innen nach außen arbeitend jedes Atom mit allen Atomen, an die es bindet. Dadurch wird ein neues Set größerer Merkmale definiert, und die entsprechenden Vektorelemente werden gesetzt. Die am weitesten verbreitete Variante dieser Methode ist der ECFP4-Algorithmus, bei dem Teilfragmente einen Radius von zwei chemischen Bindungen um ein zentrales Atom haben.

Die RDKit-Bibliothek bietet Werkzeuge zur Berechnung von ECFP4-Fingerabdrücken für Moleküle. DeepChem stellt praktische Wrapper für diese Funktionen zur Verfügung. Die Klasse `dc.feat.CircularFingerprint` erbt von `Featurizer` und bietet eine Standardschnittstelle, um Moleküle zu kennzeichnen:

```
smiles = ['C1CCCCC1', 'O1CCOCC1'] # Cyclohexan und Dioxan
mols = [Chem.MolFromSmiles(smile) for smile in smiles]
feat = dc.feat.CircularFingerprint(size=1024)
arr = feat.featurize(mols)
# arr ist ein 2 x 1024-Array, das die Fingerabdrücke der
# beiden Moleküle enthält
```

ECFPs haben einen bedeutenden Nachteil: Der Fingerabdruck verarbeitet eine große Menge an Informationen über das Molekül, einige Angaben gehen dabei jedoch verloren. Es ist möglich, dass zwei unterschiedliche Moleküle identische

Fingerabdrücke haben. Hat man einen Fingerabdruck, ist es unmöglich, eindeutig zu bestimmen, von welchem Molekül dieser stammt.

Molekulare Deskriptoren

Ein alternativer Denkansatz besagt, dass es nützlich ist, Moleküle anhand physikochemischer Deskriptoren zu beschreiben. Diese entsprechen üblicherweise verschiedenen berechneten Größen, die die Struktur des Moleküls beschreiben. Diese Größen, wie der logarithmische Verteilungskoeffizient oder die polare Oberfläche, leiten sich häufig aus der klassischen Physik oder Chemie ab. RDKit berechnet viele dieser physikalischen Deskriptoren für Moleküle. Der DeepChem-Featurizer `dc.feat.RDKitDescriptors()` bietet eine einfache Möglichkeit, die gleichen Berechnungen durchzuführen:

```
feat = dc.feat.RDKitDescriptors()
arr = feat.featurize(mols)
# arr ist ein 2 x 111-Array, das die Eigenschaften
# der beiden Moleküle enthält
```

Diese Featurization ist offensichtlich für einige Aufgabenstellungen nützlicher als für andere. Es eignet sich am besten für Vorhersagen, die auf relativ allgemeinen Eigenschaften der Moleküle beruhen. Es ist weniger geeignet, um Eigenschaften vorherzusagen, die von der genauen Anordnung der Atome abhängen.

Graph Convolutions

Die im vorhergehenden Abschnitt erläuterten Featurizations wurden von Menschen entworfen. Ein Experte dachte sorgfältig darüber nach, auf welche Weise Moleküle dargestellt werden können, um sie als Eingabe für Machine-Learning-Modelle verwenden zu können, und programmierte die Darstellung dann von Hand. Wäre es stattdessen möglich, das Modell selbst herausfinden zu lassen, wie Moleküle am besten dargestellt werden können? Darum geht schließlich beim Machine Learning: Anstatt manuell eine Funktion zu entwerfen, können wir versuchen, eine automatisch anhand der Daten zu erstellen.

Als Analogie betrachten wir ein Convolutional Neural Network für die Bilderkennung. Die Eingabe in das Netz ist das Bild. Es besteht aus einem Zahlenvektor für jedes Pixel, zum Beispiel die drei Farbkomponenten. Das ist eine sehr einfache, allgemeine Darstellung des Bilds. Der erste Convolution-Layer lernt, einfache Muster, wie vertikale oder horizontale Linien, zu erkennen. Seine Ausgabe ist erneut ein Zahlenvektor für jedes Pixel, der aber abstrakter dargestellt wird. Jede Zahl steht für ein lokales geometrisches Merkmal.

Das Netz setzt sich durch eine Reihe von Schichten fort. Jede gibt eine neue Darstellung des Bilds aus, die abstrakter als die der vorhergehenden Schicht ist und weniger an die ursprünglichen Farbkomponenten erinnert. Und diese Darstellungen werden automatisch anhand der Daten gelernt, nicht von einem Menschen

entworfen. Niemand gibt dem Modell vor, nach welchen Mustern es suchen soll, um festzustellen, ob das Bild eine Katze enthält. Das Modell begreift das durch das Training.

Graph Convolutional Networks wenden die gleiche Idee auf Graphen an. So wie ein gewöhnliches CNN mit einem Zahlenvektor beginnt, beginnt ein Graph Convolutional Network mit einem Zahlenvektor für jeden Knoten und/oder jede Kante. Stellt der Graph ein Molekül dar, kann es sich bei diesen Zahlen um chemische Eigenschaften jedes Atoms handeln, wie z. B. sein Element, seine Ladung und seinen Hybridisierungszustand. So wie eine Konvolutionsebene einen neuen Vektor für jedes Pixel basierend auf einem lokalen Bereich ihrer Eingaben berechnet, berechnet ein Graph-Convolution-Layer einen neuen Vektor für jeden Knoten und/oder jede Kante. Die Ausgabe wird berechnet, indem ein gelernter Convolution-Kernel auf jeden lokalen Bereich des Graphen angewendet wird, wobei »lokal« nun als Kanten zwischen Knoten definiert ist. Beispielsweise kann ein Ausgabevektor jedes Atoms auf dem Eingabevektor desselben Atoms und allen anderen Atomen, an die es direkt gebunden ist, berechnet werden.

Das ist die Grundidee. Was die Details betrifft, wurden viele verschiedene Varianten vorgeschlagen. Glücklicherweise enthält DeepChem Implementierungen vieler dieser Architekturen, sodass Sie sie ausprobieren können, auch ohne jedes Detail zu verstehen. Beispiele sind Graph Convolutions (`GraphConvModel`), Weave-Modelle (`WeaveModel`), Message Passing Neural Networks (`MPNNModel`), Deep Tensor Neural Networks (`DTNNModel`) sowie weitere.

Graph Convolutional Networks sind leistungsstarke Tools zur Analyse von Molekülen, haben jedoch eine wichtige Einschränkung: Die Berechnung basiert ausschließlich auf dem Molekülgraphen. Sie erhalten keine Angaben zur Konformation des Moleküls, sodass sie nichts vorhersagen können, das von der Konformation abhängt. Das macht sie besonders für kleine, meist starre Moleküle geeignet. Im nächsten Kapitel werden wir Methoden vorstellen, die für große, flexible Moleküle, die viele Konformationen annehmen können, besser geeignet sind.

Trainieren eines Modells zur Vorhersage der Löslichkeit

Wir fügen nun alle Teile zusammen und trainieren ein Modell anhand eines realen, chemischen Datensatzes, um eine wichtige molekulare Eigenschaft vorherzusagen. Zuerst laden wir den Datensatz:

```
tasks, datasets, transformers = dc.molnet.load_delaney(featurizer='GraphConv')
train_dataset, valid_dataset, test_dataset = datasets
```

Dieser Datensatz enthält Angaben zur Löslichkeit, die ein Maß dafür ist, wie leicht sich ein Molekül in Wasser löst. Diese Eigenschaft ist von entscheidender Bedeutung für alle Chemikalien, die man als Arzneimittel verwenden möchte. Sollte es

sich nicht leicht lösen, dürfte es unmöglich sein, genug davon in den Blutkreislauf eines Patienten einzubringen, um eine therapeutische Wirkung zu erzielen. Medizinische Chemiker verbringen viel Zeit damit, Moleküle zu modifizieren, um ihre Löslichkeit zu erhöhen.

Wir geben die Option `featurizer='GraphConv'` an. Wir wollen Graph Convolution verwenden, und diese Option teilt MoleculeNet mit, den SMILES-String jedes Moleküls in das für das Modell benötigte Eingabeformat umzuwandeln.

Als Nächstes erstellen und trainieren wir das Modell:

```
model = GraphConvModel(n_tasks=1, mode='regression', dropout=0.2)
model.fit(train_dataset, nb_epoch=100)
```

Wir geben an, dass es für jeden Datenpunkt nur eine Aufgabe, d.h. einen Ausgabewert (die Löslichkeit), gibt. Außerdem geben wir an, dass es sich um ein Regressionsmodell handelt, d.h., die Labels sind fortlaufende Zahlen, und das Modell sollte versuchen, diese so genau wie möglich wiederzugeben. Das ist der Unterschied zu einem Klassifikationsmodell, bei dem versucht wird, vorherzusagen, zu welcher der vorgegebenen Klassen jeder Datenpunkt gehört. Um Overfitting zu verringern, geben wir eine Drop-out-Rate von 0.2 an, was bedeutet, dass 20 % der Ausgaben jedes Convolution-Layers zufällig auf 0 gesetzt werden.

Das war es schon! Jetzt können wir das Modell evaluieren, um zu sehen, wie gut es funktioniert. Hierzu verwenden wir den Pearson-Korrelationskoeffizienten als unsere Metrik:

```
metric = dc.metrics.Metric(dc.metrics.pearson_r2_score)
print(model.evaluate(train_dataset, [metric], transformers))
print(model.evaluate(test_dataset, [metric], transformers))
```

Der Korrelationskoeffizient beträgt 0.95 für den Trainingsdatensatz und 0.83 für den Testdatensatz. Anscheinend liegt etwas Overfitting vor, aber nicht zu stark. Außerdem ist ein Korrelationskoeffizient von 0.83 durchaus ordentlich. Unser Modell prognostiziert erfolgreich die Löslichkeit von Molekülen anhand ihrer Struktur!

Nun, da wir ein Modell haben, können wir es verwenden, um die Löslichkeit neuer Moleküle vorherzusagen. Angenommen, uns interessierten die folgenden fünf Moleküle, die als SMILES-Strings vorliegen:

```
smiles = ['COC(C)(C)CCCC(C)CC=CC(C)=CC(=O)OC(C)C',
          'CCOC(=O)CC',
          'CSc1nc(NC(C)C)nc(NC(C)C)n1',
          'CC(C#C)N(C)C(=O)Nc1ccc(Cl)cc1',
          'Cc1cc2ccccc2cc1C']
```

Um diese als Eingabe in unserem Modell verwenden zu können, müssen wir zunächst RDKit zum Parsen der SMILES-Strings einsetzen. Anschließend setzen wir einen DeepChem-Featurizer ein, um sie in das von der Graph Convolution erwartete Format zu konvertieren:

```
from rdkit import Chem
mols = [Chem.MolFromSmiles(s) for s in smiles]
featurizer = dc.feat.ConvMolFeaturizer()
x = featurizer.featurize(mols)
```

Im nächsten Schritt übergeben wir `x` an das Modell, um die Löslichkeit vorherzusagen:

```
predicted_solubility = model.predict_on_batch(x)
```

MoleculeNet

Wir haben bereits zwei Datensätze aus dem Modul `molnet` geladen: den Toxizitätsdatensatz Tox21 im vorhergehenden Kapitel und den Delaney-Löslichkeitsdatensatz in diesem Kapitel. MoleculeNet ist eine große Sammlung von Datensätzen, die für das molekulare Machine Learning nützlich sind. Wie Abbildung 4-10 zeigt, enthält es Daten zu verschiedenen molekularen Eigenschaften. Sie reichen von physikalischen Eigenschaften, die mit Quantenmechanik berechnet werden können, bis hin zu Informationen über Wechselwirkungen im menschlichen Körper, wie Toxizität und Nebenwirkungen.

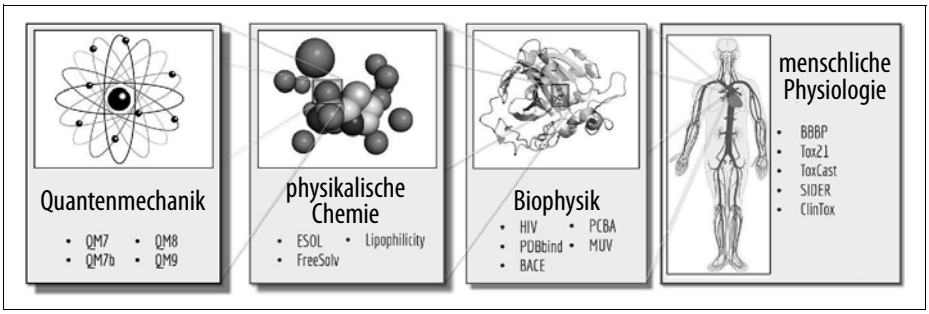

Abbildung 4-10: MoleculeNet beherbergt viele verschiedene Datensätze aus unterschiedlichen Bereichen der molekularen Wissenschaften. Wissenschaftler finden es nützlich, physikalische, biophysikalische und physiologische sowie Quanteneigenschaften von Molekülen vorherzusagen.

Bei der Entwicklung neuer Machine-Learning-Methoden kann MoleculeNet als Sammlung von Richtwerten zum Testen dieser Methoden verwendet werden. Unter *http://moleculenet.ai* können Sie einsehen, wie gut verschiedene Standardmodelle für die einzelnen Datensätze funktionieren, und vergleichen, wie gut Ihr eigenes Modell im Vergleich zu etablierten Methoden abschneidet.

SMARTS-Strings

In vielen häufig verwendeten Anwendungen, wie etwa der Textverarbeitung, wollen wir nach einem bestimmten String suchen. In der Cheminformatik begegnen wir ähnlichen Situationen, in denen wir bestimmen wollen, ob Atome eines Mole-

küls einem bestimmten Muster entsprechen. Hierfür gibt es eine Reihe von Anwendungsfällen:

- Durchsuchen einer Moleküldatenbank, um Moleküle zu identifizieren, die eine bestimmte Substruktur enthalten.
- Ausrichten mehrerer Moleküle auf einer gemeinsamen Substruktur zur Verbesserung der Visualisierung.
- Hervorheben einer Substruktur in einem Plot.
- Beschränkung einer Substruktur während einer Berechnung.

SMARTS ist eine Erweiterung von SMILES, die wir bereits vorgestellt haben und die zur Erstellung von Abfragen verwendet wird. Man kann sich SMARTS-Muster ähnlich wie reguläre Ausdrücke in der Textsuche vorstellen. Bei der Suche in einem Dateisystem kann beispielsweise "foo*.bar" angegeben werden, das foo.bar, foo3.bar und foolish.bar finden würde. Ein SMILES-String darf ebenfalls als SMARTS-String verwendet werden. Der SMILES-String "CCC" ist auch ein gültiger SMARTS-String und findet Sequenzen von drei benachbarten aliphatischen Kohlenstoffatomen. Das folgende Codebeispiel zeigt, wie wir Moleküle aus SMILES-Strings definieren, diese Moleküle anzeigen und die Atome hervorheben können, die zu einem SMARTS-Muster passen.

Zuerst importieren wir die benötigten Bibliotheken und erstellen eine Liste von Molekülen basierend auf einer Liste von SMILES-Strings. Das Ergebnis ist in Abbildung 4-11 zu sehen:

```
from rdkit import Chem
from rdkit.Chem.Draw import MolsToGridImage

smiles_list = ["CCCCC","CCOCC","CCNCC","CCSCC"]
mol_list = [Chem.MolFromSmiles(x) for x in smiles_list]
```

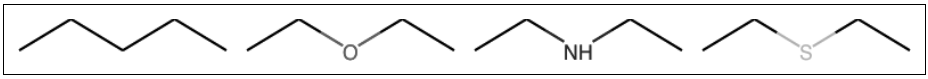

Abbildung 4-11: Chemische Strukturen aus SMILES.

Jetzt können wir sehen, welche SMILES-Strings zum SMARTS-Muster "CCC" passen (siehe Abbildung 4-12):

```
query = Chem.MolFromSmarts("CCC")
match_list = [mol.GetSubstructMatch(query) for mol in
mol_list]
MolsToGridImage(mols=mol_list, molsPerRow=4,
highlightAtomLists=match_list)
```

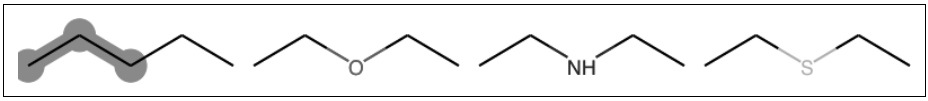

Abbildung 4-12: Moleküle, die zum SMARTS-Ausdruck "CCC" passen.

In dieser Abbildung sind einige Dinge zu beachten. Zunächst einmal entspricht der SMARTS-Ausdruck nur der ersten Struktur. Keine der anderen Strukturen enthält drei benachbarte Kohlenstoffatome. Beachten Sie auch, dass es mehrere Möglichkeiten gibt, wie das SMARTS-Muster mit dem ersten Molekül in dieser Abbildung übereinstimmen könnte – es könnte drei benachbarten Kohlenstoffatomen entsprechen, indem am ersten, zweiten oder dritten Kohlenstoffatom begonnen wird. RDKit enthält zusätzliche Funktionen, die alle möglichen SMARTS-Übereinstimmungen anzeigen, aber diese werden wir jetzt nicht vorstellen.

Zusätzliche Platzhalterzeichen können eingesetzt werden, um bestimmte Atomgruppen abzugleichen. Wie bei Text kann das Sternchen (*) verwendet werden, um jedes Atom zu finden. Das SMARTS-Muster "C*C" findet einen aliphatischen Kohlenstoff, der an ein Atom gebunden ist, das an einen weiteren aliphatischen Kohlenstoff bindet (siehe Abbildung 4-13).

```
query = Chem.MolFromSmarts("C*C")
match_list = [mol.GetSubstructMatch(query) for mol in
mol_list]
MolsToGridImage(mols=mol_list, molsPerRow=4,
highlightAtomLists=match_list)
```

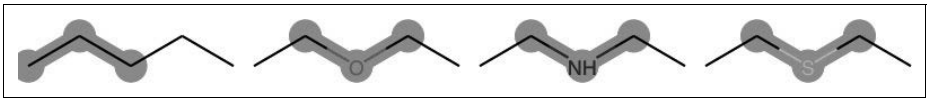

*Abbildung 4-13: Moleküle, die dem SMARTS-Ausdruck "C*C" entsprechen.*

Die SMARTS-Syntax kann erweitert werden, um nur bestimmte Atomgruppen zuzulassen. Zum Beispiel findet der String "C[C,O,N]C" Kohlenstoff, der an Kohlenstoff, Sauerstoff oder Stickstoff gebunden ist, der an einen weiteren Kohlenstoff bindet (siehe Abbildung 4-14):

```
query = Chem.MolFromSmarts("C[C,N,O]C")
match_list = [mol.GetSubstructMatch(query) for mol in
mol_list]
MolsToGridImage(mols=mol_list, molsPerRow=4,
highlightAtomLists=match_list)
```

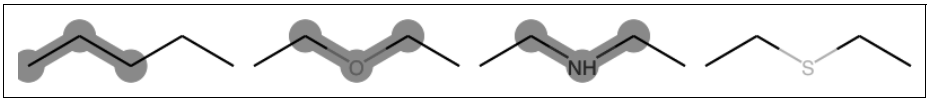

Abbildung 4-14: Moleküle, die dem SMARTS-Ausdruck "C[C,N,O]C" entsprechen.

SMARTS bietet noch viel mehr, als in dieser kurzen Einführung beschrieben wird. Interessierte Leser werden gebeten, das »Daylight Theory Manual« zu lesen, um einen tieferen Einblick in SMILES und SMARTS zu erhalten.[1] Wie wir in Kapitel 11 sehen werden, kann SMARTS verwendet werden, um komplexe Abfragen zu

1 Daylight Chemical Information Systems, Inc. »Daylight Theory Manual.« *http://www.daylight.com/dayhtml/doc/theory/*. 2011.

erstellen, mit denen Moleküle identifiziert werden können, die in Assays problematisch sein könnten.

Fazit

In diesem Kapitel haben Sie die Grundlagen des molekularen Machine Learning erlernt. Nach einem kurzen Überblick über die Grundlagen der Chemie haben wir untersucht, wie Moleküle traditionell für Computersysteme dargestellt wurden. Des Weiteren haben Sie Graph Convolutions kennengelernt, die einen neueren Ansatz zur Modellierung von Molekülen im Deep Learning darstellen. Anschließend haben wir ein vollständiges Beispiel für die Verwendung des Machine Learning mit Molekülen zur Vorhersage einer wichtigen physikalischen Eigenschaft gesehen. Diese Methoden dienen als Grundlagen, auf denen spätere Kapitel aufbauen werden.