Fraunhofer IZI

HOCHSCHULE MITTWEIDA
UNIVERSITY OF APPLIED SCIENCES

BACHELORARBEIT

Florian Lagoda

Generierung von *in silico* Modellen zur Vorhersage der Bioverfügbarkeit potentieller Arzneistoffe

Mittweida, 2017

Fakultät Angewandte Computer- und
Biowissenschaften

BACHELORARBEIT

Generierung von *in silico* Modellen zur Vorhersage der Bioverfügbarkeit potentieller Arzneistoffe

Autor:
Herr

Florian Lagoda

Studiengang:
Biotechnologie

Seminargruppe:
BT14wM-B

Erstprüfer:
Prof. Dr. Dirk Labudde

Zweitprüfer:
Dr. Mirko Buchholz

weiterer Betreuer:
Christian Jäger

Einreichung:
Mittweida, 11. Oktober 2017

Verteidigung/Bewertung:
Halle, 2017

Bibliographische Beschreibung:

Lagoda, Florian: Generierung von *in silico* Modellen zur Vorhersage der Bioverfügbarkeit potentieller Arzneistoffe. - 2017. - Seitenzahl Verzeichnisse: 12, Seitenzahl des Inhaltes: 51, Seitenzahl der Anhänge: 5, S. 60 Mittweida, Hochschule Mittweida, Fakultät Angewandte Computer- und Biowissenschaften, Bachelorarbeit, 2017

Englischer Titel

Generation of *in silico* models to predict the bioavailability of potential drugs

Kurzbeschreibung:

Ziel dieser Arbeit ist es, eine essentielle Eigenschaft (die Bioverfügbarkeit) von Verbindungen zur Auswahl von neu synthetisierten Verbindungen, vorherzusagen. Dafür wird ein QSPR Modell erarbeitet, das die orale Bioverfügbarkeit vorhersagt. Es werden verschiedene Modelle erstellt, die auf unterschiedlichen Trainingsmethoden beruhen. Die Performance der Modelle wird durch den Korrelationskoeffizienten (R^2) bewertet.

Danksagung

Bedanken möchte ich mich bei Herrn Prof. Dr. Demuth für die Möglichkeit, meine Bachelorarbeit am IZI Leipzig in der Projektgruppe MWT in Halle (Saale) anzufertigen.
Herrn Dr. Mirko Buchholz danke ich für die Betreuung während der Bachelorarbeit und die Übernahme des Zweitgutachtens.
Neben ihm möchte ich mich auch ausdrücklich bei Herrn Christian Jäger für die Beantwortung zahlreicher Fragen und die ständige Diskussionsbereitschaft bedanken.
Bei Herrn Prof. Dr. Labudde möchte ich mich für die Übernahme des Erstgutachtens bedanken.
Bei Herrn Sebastian Wussow und Herrn Felix Moorhoff möchte ich mich für die vielen hilfreichen Tipps bedanken.
Der gesamten Arbeitsgruppe Wirkstoffdesign und Analytische Chemie danke ich für die Zusammenarbeit und das gute Arbeitsklima.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Formelverzeichnis

Abkürzungsverzeichnis

ADME	Absorption, Distribution, Metabolismus, Exkretion
AUC	Fläche unter der Kurve (engl. area under curve)
BCUT	Burden's, CAS's, University of Texas
ClogP	berechneter Octanol/Wasser-Koeffizient
F	Bioverfügbarkeit
GA	Genetischer Algorithmus
GFA	genetische Funktionsnäherung (engl. genetic function appoximation)
GRNN	Allgemeines Regressions Neuronales Netz (engl. general regression neural network)
HAC	Anzahl an Wasserstoffbrücken-Akzeptoren
HDO	Anzahl an Wasserstoffbrücken-Donatoren
HKA	Hauptkomponentenanalyse (PCA, engl. principal component analysis)
i.v.	intravenöse Injektion
KNIME	Konstanz Information Miner
logP	Octanol/Wasser Verteilungskoeffizienten
MLR	multiple lineare Regression
MOE®	Molecular Operating Environment
MOPAC	Molecular Orbital PACkage
MR	molare Refraktivität
MW	molekulares Gewicht
PEOE	(engl. partial equalization of orbital electronegativities)
PCR	Hauptkomponentenregression (engl. principal component regression)
PDF	Wahrscheinlichkeitsdichtefunktion
PLS	partial least square regression

PNN	Probabilistisches Neuronales Netzwerk (engl. probabilistic neural network)
p.o.	oral/peroral (= über den Mund)
QM	quantenmechanische Deskriptoren
QSPR	Quantitatives-Struktur-Eigenschafts-Beziehungs-Modell
R^2	Bestimmtheitsmaß (quadratischer Korrelationskoeffizient)
RGB	rigide Bindungen
RNG	Anzahl an Ringen
RMSE	Wurzel des mittleren quadratischen Fehlers (engl. root-mean-square error)
RTB	Anzahl an rotierbaren Bindungen
SlogP	Octanol/Wasser-Koeffizient
SVM	Support Vector Machine
TPSA	topologische polare Molekülоberfläche

1 Einleitung

Wenn Pharmafirmen ein neues Medikament auf den Markt bringen wollen, müssen sie im Vorhinein die Wirksamkeit, Qualität und Sicherheit beweisen [1]. Dies wird durch die Durchführung zahlreicher Tests sichergestellt. Aus diesem Grund ist die Wirkstoffentwicklung ein sehr zeitintensiver und teurer Prozess. Zwischen Startpunkt, dem Suchen nach einem geeigneten Ansatzpunkt eines Zielmoleküls (target discovery) zur Therapie und dem marktreifen Medikament liegen im Durchschnitt 12 Jahre [2]. Die dadurch entstehenden Kosten betragen oft mehr als 800 Millionen US-Dollar [3]. Die hohen Kosten und der langwierige Prozess lassen sich u.a. durch die niedrige Erfolgsrate von Arzneimittelkandidaten erklären.

Die Bioverfügbarkeit ist ein essentieller pharmakokinetischer Parameter in der Wirkstoffentwicklung. Bei Medikamenten, die oral dargereicht werden, scheitern rund 30% der Wirkstoffkandidaten wegen schlechter Pharmakokinetik [4] in der Entwicklung. Während im Jahre 1991 die Medikamentenentwicklung von Wirkstoffkandidaten in den klinischen Phasen noch zu 40% aufgrund schlechter Pharmakokinetik und Bioverfügbarkeit gestoppt wurde, reduzierte sich dieser Wert bis 2000 auf 10% [5].

Dementsprechend wäre es sinnvoll, wenn diese Ausfälle schon vor der klinischen Phase prognostiziert werden könnten und so die Kosten der Wirkstoffentwicklung reduziert würden. Daher werden unterschiedlichste Methoden aus den Bereichen der *in vitro*, *in vivo* und *in silico* Forschung im frühen Stadium der Wirkstoffentwicklung angewendet, um Wirkstoffkandidaten mit geringen Erfolgschancen frühzeitig herauszufiltern. Ein Beispiel einer *in silico* Methode sind Quantitative-Struktur-Eigenschafts-Beziehungs-Modelle (QSPR, engl. quantitative-structure-property relationship). QSPRs sind mathematische Modelle, mit deren Hilfe versucht wird, eine Beziehung zwischen von der Struktur abgeleiteten Eigenschaften eines Wirkstoffkandidaten und seinen physikochemischen Eigenschaft herzustellen. Sie beruhen auf der Annahme, dass strukturell ähnliche Wirkstoffkandidaten ähnliche Eigenschaften besitzen. [6]

Folgende Schritte sind nötig, um ein QSPR-Modell zu erstellen (siehe Abbildung 1):

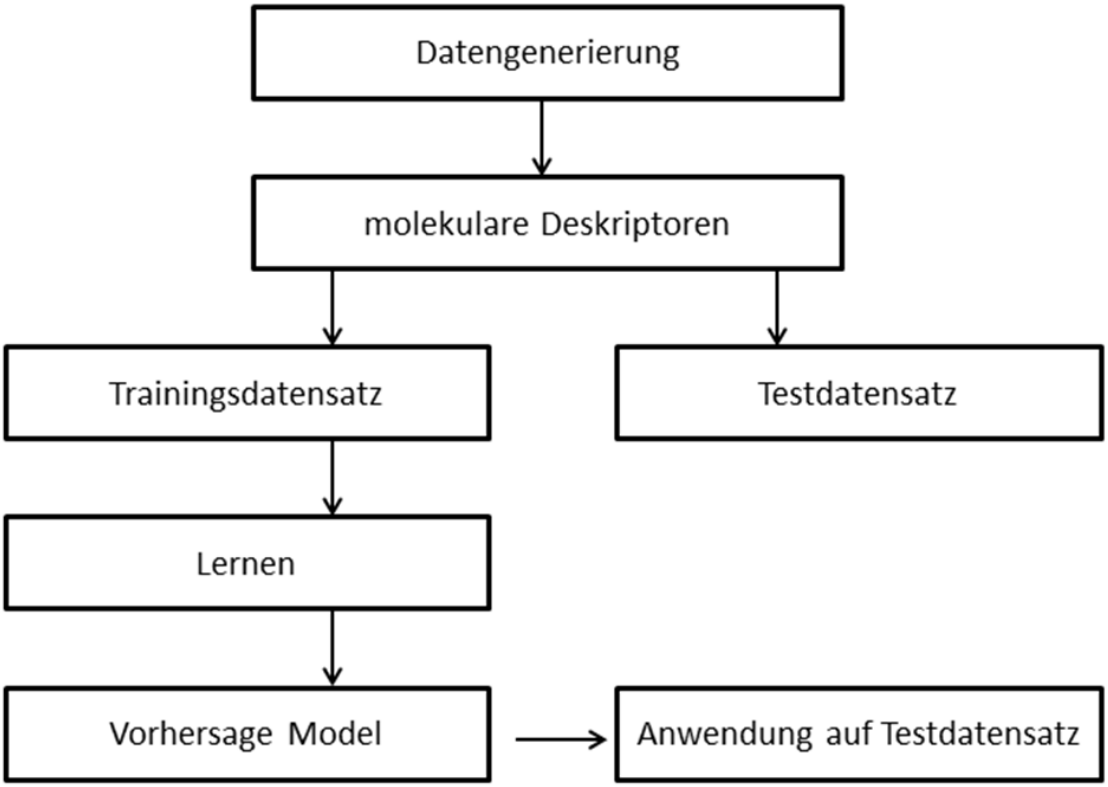

Abbildung 1 Allgemeines Schema zur Erstellung eines QSPR Modells

Die Entwicklung eines QSPR Modells beginnt mit der Zusammenstellung eines Datensatzes. In dieser Arbeit sind dies Literaturdaten von experimentell bestimmten Bioverfügbarkeitswerten. Dabei spielt die Qualität des Datensatzes eine bedeutende Rolle. Sie äußert sich in der Homogenität. Ein Datensatz ist homogen, wenn eine statistische Normalverteilung vorliegt, alle Daten auf gleiche Weise erzeugt wurden und damit vergleichbar sind. Die Repräsentation der gesammelten Daten erfolgt dann durch die Verwendung von Eigenschaften, den sogenannten molekularen Deskriptoren, die (Teil-) Informationen der Moleküle beschreiben. Anschließend wird der gesamte Datensatz für die Feinabstimmung und die Validierung des QSPR Modells in einen Trainings- und Testdatensatz unterteilt. Bei der Generierung von Modellen wird auf unterschiedliche Trainingsmethoden zurückgegriffen (siehe 3.5). Das mit dem Trainingsdatensatz trainierte Modell wird anschließend auf den Testdatensatz angewendet. Damit wird die Fähigkeit überprüft, wie gut fremde Moleküle wiedergegeben werden können. Dies wird mit Hilfe des Korellationskoeffizienten ausgedrückt.

2 Zielstellung

Ziel dieser Arbeit ist es, eine essentielle Eigenschaft (die Bioverfügbarkeit) von Verbindungen, zur Auswahl von neu synthetisierten Verbindungen, vorherzusagen. Dafür wird ein QSPR Modell erarbeitet, dass die orale Bioverfügbarkeit vorhersagt. Die Performance des Modells wird durch den Korrelationskoeffizienten (R^2) bewertet. Er gibt Auskunft darüber, wie viel Variation in den Daten durch das Regressionsmodell erklärt werden kann und nimmt Werte zwischen 0 bis 1 an (0 = kein linearer Zusammenhang, 1 = perfekter linearer Zusammenhang). Datengrundlage für das Modell ist der *hF-MWT-17* Datensatz [7].

3 Theoretischer Hintergrund

3.1 ADME und Bioverfügbarkeit

Die orale (absolute) Bioverfügbarkeit (F) beschreibt den prozentualen Anteil des dargereichten Wirkstoffs, der den pharmakologischen Wirkort, im Vergleich zur i.v. Applizierung (intravenöse Injektion, 100% Referenz), erreicht [8]. Dieser Parameter ist im ADME-Modell zusammengefasst. Die Abkürzung ADME steht für Absorption (**a**bsorption), Verteilung (**di**stribution), Metabolisierung (**m**etabolism) und Ausscheidung (**e**xcretion).

Unter Absorption (Resorption) wird die Aufnahme des Wirkstoffs vom Applizierungsort, nach vorheriger Lösung (Freisetzung, liberation) verstanden.

Distribution beschreibt die Verteilung des Wirkstoffs durch den Kreislauf in seine Kompartimente (z.B. Fettgewebe oder Liquor). Die Metabolisierung erfolgt über Enzyme. Ausgeschieden wird ein Wirkstoff hauptsächlich durch die Niere (Urin). [9]

3.2 Bioverfügbarkeit

Aus dem ADME-Modell ergeben sich weitere Paramater, wie z.B. die AUC (Plasmakonzentrations-Zeit-Profil, engl. area under the curve), die u.a. Verwendung bei der Berechnung der Bioverfügbarkeit findet.

Zur Messung der absoluten Bioverfügbarkeit wird ein Wirkstoff i.v. dargereicht und Plasmaproben werden entnommen. Nach Wiederholung mit oraler Darreichung werden die AUCs berechnet (siehe Abb. 2.).

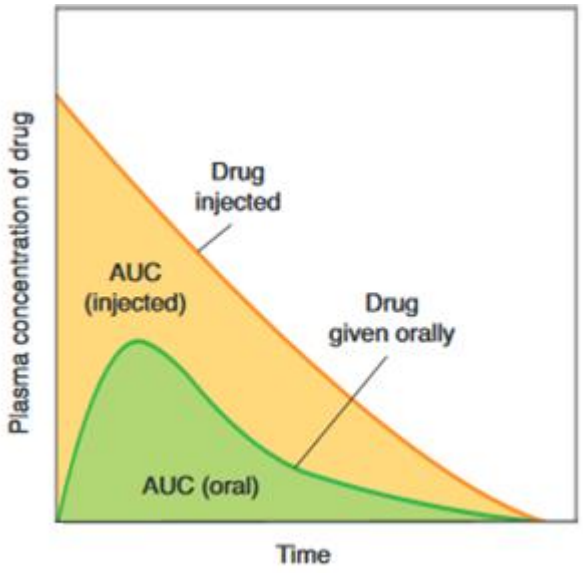

Abbildung 2 Vergleich der AUCs nach i.v. und p.o. Darreichung
Die Abbildung zeigt die Plasmakonzentration (AUC) (AUC injected und AUC oral) des Wirkstoffs nach i.v. (Drug injected) und p.o. (Drug given orally) Darreichung aufgetragen gegen die Zeit (Time). Die Bioverfügbarkeit wird über den Vergleich der beiden AUC Kurven bestimmt (siehe Formel 1). Verändert nach [10]

Die absolute Bioverfügbarkeit lässt sich mit folgender Formel berechnen:

$$\%\, F = \frac{AUC\ p.o.\ *\ Dosis\ i.v.}{AUC\ i.v.\ *\ Dosis\ p.o.} * 100 \qquad (1)$$

Legende: % F = absolute Bioverfügbarkeit in Prozent; AUC = area under the curve; p.o. = perorale Darreichung; Dosis = Menge des Wirkstoffs; i.v. = intravenöse Injektion

Als relative Bioverfügbarkeit wird der Vergleich unterschiedlicher Formulierungen des selben Wirkstoffs bei gleicher Dosis und Applizierung verstanden.
Unterschiede in den physikochemischen Eigenschaften eines Moleküls zählen zu den Hauptgründen unterschiedlicher Bioverfügbarkeiten.

3.3 Molekulare Deskriptoren

3.3.1 Allgemeine Einleitung zu molekularen Deskriptoren

Anhand eines molekularen Deskriptors können die Eigenschaften von Molekülen mit numerischen Werten charakterisiert werden. Diese können zum Beispiel die physikochemischen Eigenschaften eines Moleküls repräsentieren. Im Laufe der Zeit sind eine Vielzahl von molekularen Deskriptoren entwickelt worden. Sie variieren in der Komplexität der Information, die sie verschlüsseln und in der benötigten Zeit, sie zu berechnen. Manche Deskriptoren beruhen auf empirischen Daten (z.B. n-Octanol-Wasser-Verteilungskoeffizient), andere auf rein algorithmischen Konstrukten (z.B. Zahl der Kohlenstoff Atome). [11]

1D-Deskriptoren beschreiben molekül-übergreifende Eigenschaften wie z.B. das Molekulargewicht, die molare Refraktivität oder das Van-der-Waals Volumen. 2D-Deskriptoren werden aus der jeweiligen Strukturformel berechnet. Beispiele dafür sind z.B. die Zahl an aromatisch gebundenen Atomen im Molekül oder die Anzahl von drehbaren Bindungen. Weitere Eigenschaften, die sich aus der Strukturformel ergeben, sind z.B. die Elektronegativtät, Partialladungen oder Konnetivitätsindizes.

Mathematisch betrachtet fügt jeder zusätzlich berechnete Deskriptor eine neue Dimension zum Raum hinzu. So bekommt jedes Molekül nach der Berechnung von *n*-Deskriptorwerten einen n-dimensionalen Koordinaten-Vektor im Deskriptorraum, der seine Eigenschaftsrichtung definiert [12].

3.3.2 Für die Modelle bedeutende Deskriptoren

VSA-Deskriptoren

Bei diesen Deskriptoren wird angenommen, dass sich die Van-der-Waals-Oberfläche von Molekülen aus deren zweidimensionaler Strukturformel berechnen lässt.

Folgende Moleküleigenschaften werden dabei verwendet:

- Der Octanol/Wasser-Koeffizient (SlogP); SlogP-Werte geben Auskunft über hydrophobe bzw. hydrophile Interaktionsmöglichkeiten des Moleküls am Protein.
- Die molare Refraktivität (MR) beschreibt die Polarisierbarkeit eines Moleküls
- Die Partialladung, die hier nach der Gasteiger-Methode (PEOE) berechnet wird, gibt Auskunft über direkte elektrostatische Interaktionen.

Da für jede dieser Eigenschaften feste Wertebereiche existieren, gibt es mehrere Deskriptoren für die Eigenschaften (10 SlogP-, 8 MR- und 14 PEOE-Deskriptoren). In Summe bilden sie die gesamte Van-der-Waals-Oberfläche ab. Alle drei oben genannten Eigenschaften haben gemeinsam, dass sie atomare Parametrisierungen besitzen. Damit sind sie unabhängig von Strukturklassen. Sie lassen sich anhand von vorgegebenen Werten für einzelne Atomtypen sehr schnell und für alle Arten von Molekülen berechnen. [13,14]

TPSA

Dieser 2D-Deskriptor ermittelt die topologische polare Molekularoberfläche (TPSA, engl. topological polar surface area) eines Moleküls und kann ohne großen Rechenaufwand aus jeder Strukturformel berechnet werden. In der Medizinalchemie wird dieser Deskriptor häufig verwendet, da er Erkenntnisse über die Fähigkeit liefert, wie gut Moleküle in Zellen eindringen können. Desweiteren ist die polare Oberfläche einer Verbindung bestimmend für den Verteilungskoeffizienten zwischen Blut und Hirn. [15]

opr_violation

Dieser 2D-Deskriptor zählt die Anzahl an Verstößen des Opere's lead-like Tests [16]. Folgende Eigenschaften sind in diesem Deskriptor zusammengefasst: das molekulare Gewicht (MW), der berechnete Octanol/Wasser-Koeffizient (ClogP), die Anzahl an rotierbaren (RTB) und rigiden Bindungen (RGB), die Anzahl an Ringen (RNG) und die Anzahl an Wasserstoffbrücken-Donatoren (HDO) und Wasserstoffbrücken-Akzeptoren (HAC). Die Eigenschaften RNG, RTB und RGB wurden zu den Lipinski-Parametern der „rule of five" hinzugefügt, da sie die molekulare Komplexität von Bibliotheken durch ihre Flexibilität und Starrheit repräsentieren [16].

Tanimoto – Fingerprints

Fingerprints sind binäre Vektoren, bei denen jedes Bit die Präsenz (1) oder die Abstinenz (0) eines bestimmten Substrukturfragments innerhalb eines Moleküls angibt. Die Similarität zwischen zwei Molekülen wird meist über binäre 2D Fingerprints unter Benutzung des Tanimoto Koeffizienten ausgegeben. Dieser gibt die Anzahl an Fragmenten aus, die in beiden Molekülen vorhanden sind. [17]

Die Tanimoto Similarität zwischen zwei Molekülen *A* und *B* wird über binäre Vektoren wie in Formel 2 wiedergegeben:

$$S_{AB} = \frac{c}{a + b - c} \tag{2}$$

Dabei wird *a* Bits die Zahl „1“ in Molekül *A* gegeben, *b* Bits wird die Zahl „1“ in Molekül *B* gegeben und *c* „1“ Bits kommen in beiden Molekülen *A* und *B* vor. Der Bereich des Tanimoto Koeffizienten liegt zwischen 0 und 1. Ein Wert von 1 weist daraufhin, dass das Molekül eine identische Fingerprint Repräsentation besitzt (das heißt nicht, dass es sich um das identische Molekül handelt) und ein Wert von 0 deutet auf eine nicht vorhandene Similarität (d.h. es gibt keine Bits, die in beiden Molekülen vorkommen) hin. In dieser Arbeit finden die Fingerprints Verwendung bei der Berechung der diversen Subsets (siehe 3.6). [11]

3.3.3 BCUT Deskriptoren

Bei der Einteilung des Datensatzes in Trainings- und Testdaten werden häufig die BCUT (**B**urden’s, **C**AS’s, **U**niversity of **T**exas) Deskriptoren verwendet, da sie Verbindungen eineindeutig beschreiben können.

Sie wurden mit der Motivation entwickelt, atomare Eigenschaften, die relevant für intermolekulare Interaktionen sind, darzustellen. Häufig finden sie Anwendung bei Diversitätsanalysen (z.B. bei der Aufgabe, eine möglichst diverse Menge an Molekülen aus einer großen Datenpopulation auszuwählen). Sie beruhen auf einem früheren Deskriptor, dem Burden-Index, der von Burden im Jahre 1989 entwickelt wurde. Dieser berechnet eine Matrixdarstellung aufgrund einer Verbindungstabelle (engl. connection table) eines Moleküls. Dabei werden Atomnummern von nicht-Wasserstoffatomen auf eine Diagonale einer Matrix aufgetragen. Die off-Diagonale beschreibt den Wert, der sich nach dem Bindungstyp des Atoms richtet. Bei gebundenen Atomen wird er mit 0,1 multipliziert und bei ungebundenen mit 0,001. Die beiden niedrigsten Eigenwerte der Matrix werden anschließend berechnet, kombiniert und als Einzelindex ausgegeben.

Aufbauend auf dieser Methode wurde von Pearlman eine neue Familie die BCUT Deskriptoren generiert, die dazu benutzt werden können, einen gering dimensionalen

chemischen Raum darzustellen [18]. Anstelle von Atomnummern wird die Diagonale der Matrix bei den BCUTs mit Eigenschaften des entsprechenden Atoms, wie z.B. Polarisierbarkeit, Hydrophobizität oder Elektronegativität, beschrieben. Die höchsten und niedrigsten Eigenwerte werden dann extrahiert und als Deskriptoren verwendet. Diese Zusammenfassung erzeugt einen mehrdimensionalen chemischen Raum, in dem ein Molekül immer die gleiche Position erhält, egal aus welchem Datensatz es stammt. Damit nehmen ähnliche Moleküle ähnliche Positionen im chemischen Raum ein. Umgekehrt finden sich chemisch unterschiedliche Moleküle in jeweils anderen Raumsektoren eines Diagramms wieder. [11]

3.4 Einführung zu QSPR Modelle

3.4.1 Geschichte der QSPR Modelle

Die Wurzeln des Aufstellens einer Korrelation der sogenannten QSAR/QSPR Modelle reichen bis ins 19. Jahrhundert zurück, als Brown und Fraser folgende leichte Formel vorschlugen, die Curare[1] ähnlich lähmende Eigenschaften für ein Set von quaternisierten[2] Strychninen[3] beschreibt:

$$\Phi = f\,(C) \qquad (3)$$

Dabei steht *f* für das Maß der biologischen Aktivität und *C* repräsentiert die strukturellen Eigenschaften, die die quaternisierende Gruppe charakterisieren [19].

Ein Jahrhundert später, in den 1960er Jahren entwickelte Corwin Hansch ein Hydrophobizität-Modellsystem auf der Basis des Octanol/Wasser Verteilungskoeffizienten (logP) [20]. Sein wichtigster Beitrag zum immer noch jungen QSAR Feld war jedoch seine Vermutung, dass eine einzige Variable (wie der logP zum Beispiel) vielleicht unzureichend für die Erklärung der Potenz, bzw. Wirkstärke eines Moleküls ist. Die generalisierte Form seiner Formel ist heute als „Hansch Formel“ [21] bekannt. Weitere wichtige Ereignisse erleichterten die breite Anwendung dieses Ansatzes. So wurde z.B. im Jahre 1986 von Wold das Benutzen der PLS (PLS, engl. parital least-squares) Analyse im Gegensatz zur PCA (PCA, engl. principal component analysis) vorgeschlagen, um Feldwerte mit der Aktivität zu korrelieren [6].

[1] alkaloidhaltige Substanz – stammt aus Brechnuss-Arten und Mondsamengewächsen – wird als Pfeilgift bei indigenen Völkern aus Südamerika benutzt
[2] Quaternisierung - chemische Reaktion, Bindungen an Atom werden auf 4 organische Reste erhöht, Beispiel: Überführung tertiärer Amine in quartäre Ammoniumverbindungen durch Alkylierung
[3] giftiges Alkaloid – bewirkt eine Starre der Muskeln

3.4.2 Literaturbekannte QSPR Modelle zur Bioverfügbarkeit

In diesem Unterkapitel werden kurz die drei QSPR Modelle erläutert, aus denen der Datensatz in der Praxismodularbeit [7] generiert wurde.

Das QSPR-Modell von Turner [22] basiert auf 169 Molekülen (10 davon bilden den Testdatensatz). 94 Deskriptorwerte wurden für alle Moleküle berechnet. Anschließend wurde über eine stufenweise Regression die relative Gewichtung der einzelnen Deskriptoren ermittelt. Mit folgenden 8 Deskriptoren wurde eine Vorhersagefähigkeit mit einem Korrelationskoeffizienten von 0,72 erzielt: molekulares Volumen (molecular volume), polare Oberfläche (HLB), Octanol/Wasser Verteilungskoeffizient (logP), Wasserstoffbindungen (Elektronen Affinität), Löslichkeit (Hansen Wasserstoffbindungslöslichkeitsparameter und berechnete Wasserlöslichkeit) und elektronische Effekte (HOMO Energie).

Das Modell von Moda [23] beruht auf einem Hologramm QSAR Modell (HQSAR). Für die Erstellung wurde von den Molekülen die 2D Struktur und die entsprechenden Bioverfügbarkeitswerte genutzt. Anschließend wurde jedes Molekül auf einige einzigartige Strukturfragmente heruntergebrochen, die dann als molekulares Hologramm zusammengefügt wurden und als eine erweiterte Form von Fingerprints betrachtet werden können. Diese Hologramme können von einer Vielzahl von Parametern beeinflusst werden (Länge, Fragmentgröße und Fragment-Unterscheidung). Die HQSAR Analyse wurde mit 12 Serien von Standardeinstellungen der Hologrammlänge durchgeführt. Der Trainingsdatensatz beinhaltete 250 Strukturen und der Testdatensatz 52. Die Vorhersagefähigkeit für die Testdaten liegt bei einem R^2 von 0,85.

Das QSPR Modell von Hou und seinen Mitarbeitern [24] beruht auf der Trainingsmethode der multiplen linearen Regression; nach vorangegangener Anwendung der *genetic function appoximation* (GFA, dt. genetische Funktionsnäherung) für die Auswahl von strukturellen Fingerprints und verschiedenen Moleküleigenschaften für die Modelle. Datengrundlage waren 1014 Moleküle. Das beste Modell hatte eine Vorhersagefähigkeit, für den Testdatensatz (80 Moleküle), von einem Regressionskoeffizienten der bei 0,71 liegt.

3.5 Trainingsmethoden für die Modellerstellung

Im Folgenden werden alle wesentlichen Trainingsmethoden, die bei der Modellerstellung von Relevanz sind, kurz erläutert.

3.5.1 Lineare Trainingsmethoden

3.5.1.1 Multiple Lineare Regression

Die multiple lineare Regression (MLR, engl. multiple linear regression) ist eine gängige Modellierungsmethode. Im Allgemeinen findet dieser Ansatz wegen seiner Einfachheit und leichten Interpretation Verwendung. Es wird davon ausgegangen, dass es eine lineare Beziehung zwischen der physikochemischen Eigenschaft $\boldsymbol{y}$ und seinem Eigenschaftsvektor $\boldsymbol{X}$, der normalerweise ein Set von molekularen Deskriptoren ist, gibt. Daher kann mit einer Vorstellung (notion) von $\boldsymbol{X}$ die Eigenschaft von einem unbekannten Molekül durch das trainierte Modell vorhergesagt werden. Folgende Formel repräsentiert die allgemeine Formulierung eines MLR Modells:

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k \qquad (4)$$

Dabei ist β_0 die Modellkonstante und $X_1,\ldots. X_k$ sind die molekularen Deskriptoren mit ihren korrespondierenden Koeffizienten $\beta_1,\ldots. \beta_k$. Die Regressionskoeffizienten können über die *least squares* Methode erhalten werden. Die Größe des Koeffizienten gibt eine Auskunft über den Grad des Einflusses des molekularen Deskriptors auf die physikochemische Eigenschaft (z.B. Bioverfügbarkeit) an. Darüber hinaus zeigt ein positiver Koeffizient, dass die jeweiligen molekularen Deskriptoren positiv zur Vorhersage der physikochemischen Eigenschaft beitragen. Negative Koeffizienten bedeuten das Gegenteil. Ein Nachteil ist, dass die Interpretationen in der Anwesenheit von co-linearen Deskriptoren fehleranfällig sind. [6]

3.5.1.2 Hauptkomponentenanalyse

Mit Hilfe der Hauptkomponentenanalyse (PCA, engl. principal component analysis) ist es möglich, eine Dimensionsreduktion von mehrdimensionalen Daten zur besseren Darstellung und Analyse zu erhalten. Dabei werden die Hauptkomponenten aus

hochdimensionalen Datenvektoren berechnet, die den größtmöglichen Anteil der Varianz der ursprünglichen Vektoren enthalten [25]. Man nutzt also die Korrelation hochdimensionaler Daten aus, um am Ende zu einer informationsverdichteten und dimensionsreduzierten Datenrepräsentation mit wenigen neuen Hauptkomponenten zu kommen [26]. Diese Dimensionsreduzierung von Datenpunkten ist auch nach der Anwendung von diversen molekularen Deskriptoren hilfreich. Ein gutes Beispiel dafür sind die BCUT Deskriptoren. Diese berechnen 12 Werte zu einem Molekül (12 Dimensionen), welche unter der Zuhilfenahme der PCA und den ersten drei Hauptkomponenten auf einen dreidimensionalen Raum reduziert werden, der als vereinfachter chemischer Raum verstanden werden kann. Damit ist es anschließend möglich, Diversitätsanalysen durchzuführen.

Auch wenn häufig einige Werte von Deskriptoren korrelieren, unterscheiden sie sich auf ihrer relativen Skala (z.B. 0,01 bis 1 vs. 1 bis 100). Dies kann zu Problemen bei einer distanzbasierten Analyse führen, da die euklidische Metrik nicht mehr direkt ohne die Betonung einer Untermenge von Deskriptoren, benutzt werden kann. Mit Hilfe einer Normierung der Deskriptorwerte kann dieses Problem behoben werden. In der PCA kommt es zu einer linearen Projektion von Daten. Diese Projektion wird so gewählt, dass die Datenstruktur in der niedrigdimensionalen Projektion die größtmögliche Varianz aufweist.

3.5.1.3 Partial Least Squares Regression

Die Partial Least Squares (PLS) Regression ist eine häufig verwendete Methode für die Analyse von großen Datensätzen aufgrund ihrer inhärenten Fähigkeit, viele redundante Eigenschaften zu bewältigen und leicht interpretierbare Regressionskoeffizienten aus den prädiktiven Modellen hervorzubringen. Bei der PCA werden nur *X* Variablen in der mehrdimensionalen (multivarianten) Analyse betrachtet und nicht die biologischen Eigenschaften der Verbindungen. Dahingegen werden bei der PLS auch die Informationen der *Y* Variablen verwendet, um die Varianz innerhalb der Klassen zu maximieren. Die Wurzeln der PLS gehen auf den *nicht-linearen iterativen partial least squares* Algorithmus von Herman Wold zurück [27]. [28]

Die PLS Methode bestimmt ein lineares Regressionsmodell, indem sie einen Richtungsvektor im Deskriptorraum findet, der die erklärte Varianz in der Antwort

maximiert. Daraus ergibt sich eine Modellierungsmethode, die robust in Bezug auf co-lineare Deskriptoren und unkompliziert in der Durchführung ist. Der einzige Modellparameter, der optimiert werden kann, ist die Anzahl der Deskriptoren für die lineare Kombination. [6]

3.5.2. Nicht-lineare Trainingsmethoden

Nicht-lineare Trainingsmethoden, zu denen u.a. das Maschinelle Lernen gehört sind auch unter dem Begriff Black-Box Modelle [29] bekannt. Sie sind erfolgreich in der Modellierung von physikochemischen Eigenschaften, besitzen allerdings den Nachteil, dass sie nicht die zugrunde liegenden Assoziationen der einzelnen Eigenschaften mit ihrem spezifischen Ergebnis erkennen und nicht aufdecken können, welche Merkmale einen wesentlichen Beitrag zur beobachten Vorhersagegenauigkeit beitragen. [28]

3.5.2.1 Support Vector Machine

Support Vector Machines (SVM, engl. support vector machine – dt. Stützvektormaschine) werden bei Klassifizierungen oder bei Regressionsanalysen eingesetzt. Ziel ist dabei eine Menge von Objekten in zwei Klassen zu unterteilen, so dass um die gezogenen Klassengrenzen ein möglichst breiter Bereich, der frei von Objekten ist, entsteht (siehe Abb. 3). Neue Objekte können dann einer der beiden Klassen eindeutig zugewiesen werden. Es geht dabei um ein rein mathematisches Verfahren zur Mustererkennung, das maßgeblich von Vapnik und Tscherwonenkis entwickelt wurde.

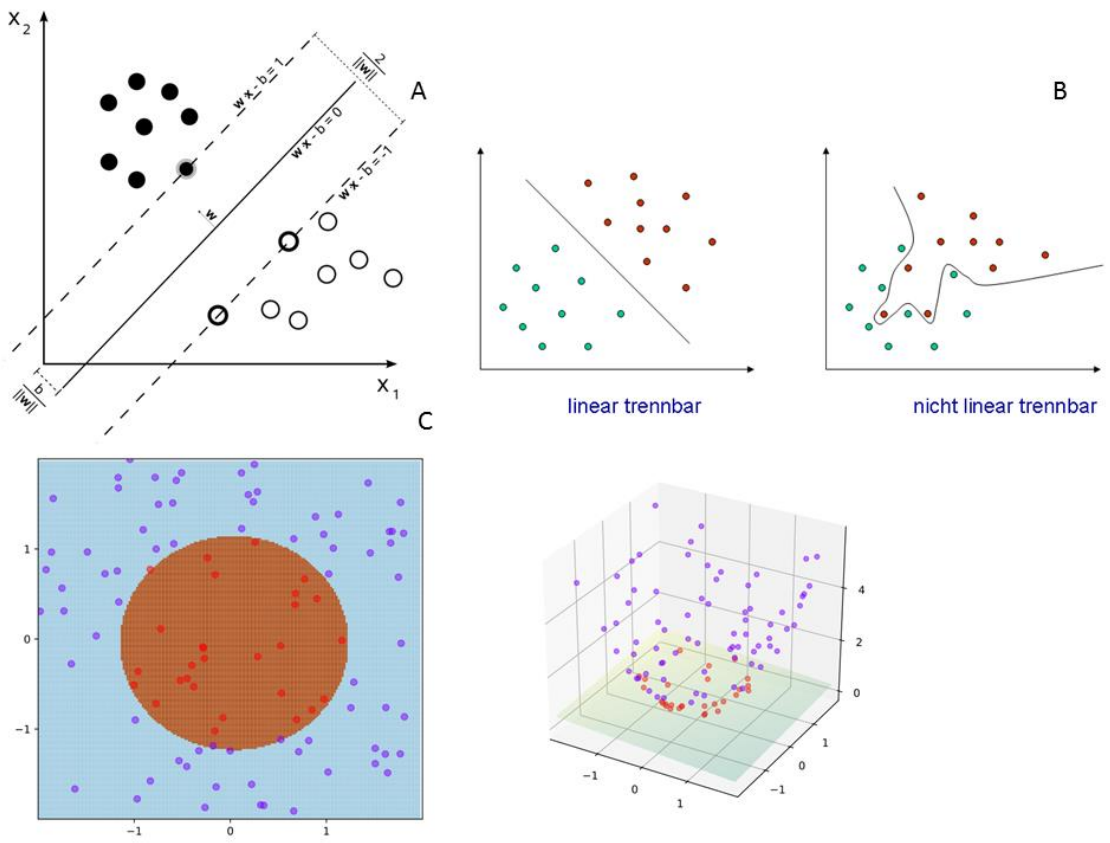

Abbildung 3 Graphik zur Support Vector Machine A) Hyperebene (durchgezogene Linie) teilt Daten in zwei Klassen ein. Auf der gestrichelten Linie befinden sich Stützvektoren (support vectors), die für die richtige Einteilung der Daten die SVM trainieren [30] **B)** Beispiel für einen linear und nicht linear trennbareren Scatter Plot. [31] **C)** Trainingsbeispiel für eine SVM mit Anwendung des Kernel Tricks ($\varphi((a, b)) = (a, b, a^2 + b^2)$) [32]

Bei nicht linear trennbaren Daten wird der Kernel-Trick angewendet. Hierfür wird der Vektorraum mit seinen Trainingsobjekten in einen höherdimensionalen Raum überführt, so dass er irgendwann linear trennbar ist und eine trennende Hyperebene bestimmt werden kann. Dies bedeutet, dass ein linearer Klassifikator auf nicht linear klassifizierbare Daten angewendet wird. Bei der Rücktransformation in den niedrigdimensionalen Raum wandelt sich die lineare Hyperebene zu einer nicht linearen Fläche, die die Trainingsobjekte eindeutig in zwei Klassen einteilt. Da eine Transformation in einen höherdimensionalen Raum normalerweise sehr rechenintensiv ist, wird eine geeignete Kernelfunktion gesucht, mit der im hochdimensionalen Raum die Hyperebene beschrieben werden kann, die sich im niedrigdimensionalen Raum nicht stark verändert, so dass eine Hin- und Rücktransformation möglich ist, ohne sie jedoch tatsächlich rechnerisch durchführen zu müssen. [33–35]

3.5.2.2 Probabilistisches Neuronales Netzwerk

Probabilistische Neuronale Netzwerke (PNN, engl. *Probabilistic neural networks*) sind einschichtige Feedforward (dt. vorwärts) Neuronale Netzwerke; das bedeutet, dass Informationen nur in einer Richtung verarbeitet werden. Sie funktionieren auf der Basis von nichtparametrischen Kalkulatoren, bedingten Wahrscheinlichkeitsdichtefunktionen (PDF, engl. probability distribution function) und der Bayes Strategie (um das erwartete Risiko zu minimieren). Ähnlich wie bei *general regression neural networks* (GRNN, dt. Allgemeines Regressions Neuronales Netzwerk), kann die PDF für jede Zieleigenschaft für einen univariaten Fall durch den nichtparametrischen Parzen's Kalkulator geschätzt werden [6]. Anschließend wird die Bayes'sche Regel angewendet, um die Klasse mit der höchsten Wahrscheinlichkeit dem neuen Daten Input zuzuweisen. Auf diese Weise wird die Wahrscheinlichkeit einer Missklassifikation minimiert. [36]

Diese Form des künstlichen Neuronalen Netzwerks ist vom *Bayes'schem Netz* und vom statistischen *Kernel Fisher discriminant analysis* Algorithmus abgeleitet und wurde 1966 von Specht eingeführt [37]. Die Netzwerkarchitektur des PNNs (siehe Abb. 5) ist zu der des GRNNs ähnlich. Innerhalb jeder Summierungsebene (s*ummation layer*) wird die geschätzte PDF für die entsprechende Zieleigenschaft durch die Summierung aller erfassten Eingaben (*input layer*) von der Musterschicht (p*attern layer*) erhalten. Diese Informationen werden anschließend dem Einzelelement der Outputebene (o*utput layer*) übergeben, bei der die PDF evaluiert und die Klasse einer unbekannten Verbindung mit der Zieleigenschaft, die den höchsten Wert hat, zugewiesen wird. [6]

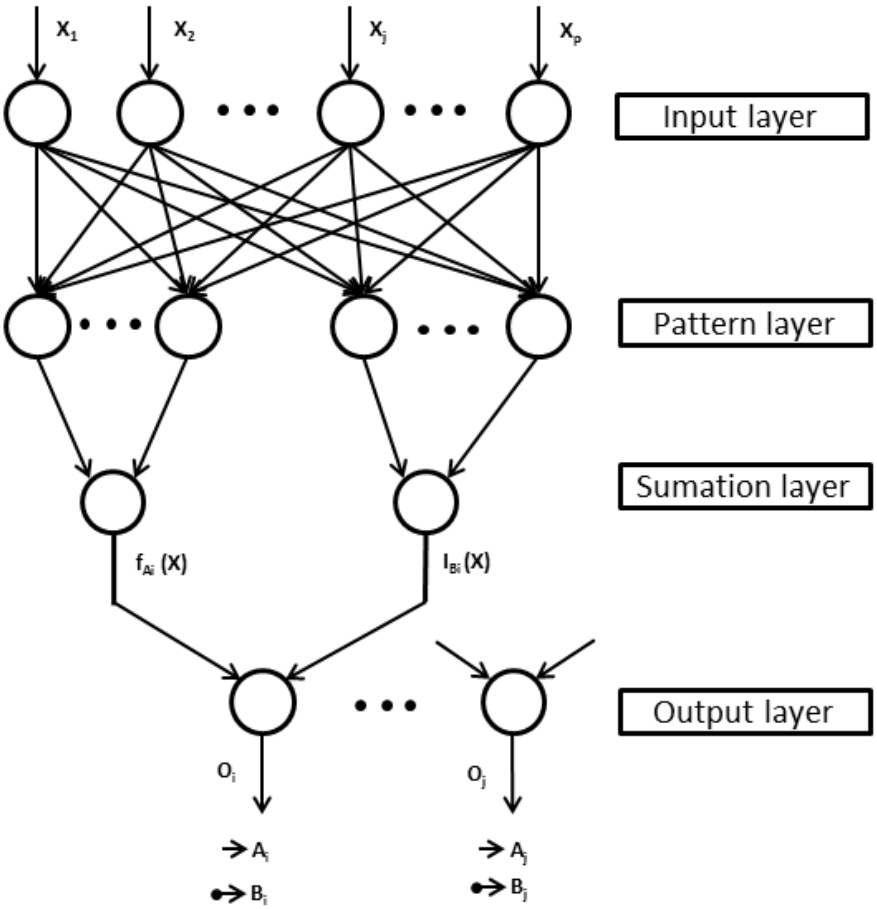

Abbildung 4 Graphische Darstellung eines theoretischen PNNs
X₁ (X₂, Xj, Xp) Variablen der verschiedenen Eingabewerte (input layer)); ***Ai, Aj*** und ***Bi, Bj*** Variablen der Trainingsmuster (pattern layer); ***fAi (X)*** und ***IBi (X)*** Variable der Summierungsebene (sumation layer); ***Oi*** und ***Oj*** Variablen der Ausgabeebene (Output layer). Verändert nach [37]

Die Vorteile des PNNs gegenüber mehrlagigen Perceptronen (engl. multilayer perceptron) liegen dabei in einer erhöhten Schnelligkeit und Korrektheit des Trainingsprozesses, der Generierung von genauen Vorhersagewahrscheinlichkeitswerten und dem optimalen Konvergieren eines Klassifikators, wenn der repräsentative Trainingsdatensatz zunimmt. [38]

3.5.2.3 Genetischer Algorithmus

Der Genetische Algorithmus (GA, engl. genetic algorithm), basiert auf der natürlichen Evolution, bei der Variablen die Rolle von Genen (in diesem Fall Deskriptoren) in einem Individuum (in diesem Fall ein Set von *d* Deskriptoren) einer Spezies übernehmen. Eine Initialgruppe von zufälligen Individuen (Population) entwickelt sich entsprechend einer Fitness Funktion, die über das „Überleben" der Individuen entscheidet. Der Algorithmus sucht nach solchen Individuen, die zu besseren Fitness Funktionen durch Selektion, Mutationen und genetischen Crossing-over Operationen führen. Der GA rastert dafür den Lösungsraum durch Kombination der Gene von zwei Individuen (Eltern), durch einen Crossing-over Operator und zusätzlich durch zufällige Mutationen vom Mutations-Operator, um zwei neue Individuen (Kinder) zu bilden, ab. So können viele Lösungen parallel entwickelt werden. [6]

In dieser Arbeit wird der GA wie folgt angewendet: Aus den zur Verfügung stehenden Deskriptoren werden im ersten Schritt zufällig 100 verschiedene Modelle erstellt. Die Vorhersagekraft jedes Modells wird durch einen Fitnesswert ausgegeben. Zur Ermittlung dieses Wertes kann entweder der Trainings- oder der Testdatensatz beitragen. Die Modelle werden nach dem Fitnesswert geordnet (gerankt). Durch genetische Operationen wie dem Crossing-over und Mutationen von zwei zufällig ausgewählten Modellen wird ein neues Modell gebildet, das über den Fitnesswert evaluiert wird. Sollte dieser Wert höher sein, erfolgt eine Neubewertung innerhalb der Tabelle und das am niedrigsten bewertete Modell wird aus der Liste gelöscht. Sollte der Fitnesswert kleiner als der des 100. Eintrags der Tabelle sein, wird das Modell verworfen. Die Modelle mit den jeweils höchsten Fitnesswerten sind von den genetischen Operationen ausgeschlossen. Über eine vorher definierte Anzahl von Generationen werden die Operationen durchgeführt, wobei sich durch den Selektionsdruck das Modell mit der höchsten Fitness durchsetzt. Die Modelle werden anschließend gespeichert und auf den Testdatensatz angewendet. Der Vorteil dieses Verfahrens besteht darin, dass immer wieder neue Kombinationen von Deskriptoren möglich sind. Dies ist bei linearen Verfahren nicht möglich, da dort durch die Datenreduktion gelöschte Deskriptoren nicht für weitere Modelle zur Verfügung stehen. [39,40]

3.6 Diverse Subset von MOE

Mit der *diverse subset* Funktion von dem Programm MOE® (Molecular Operating Environment) können Einträge einer Datenbank nach ihrer Diversität eingestuft werden. Mit dieser Funktion wird in dieser Arbeit der Datensatz in einen Trainings- und Testdatensatz eingeteilt.

Der Testdatensatz beinhaltet am Ende dieser Berechnung die diversesten Moleküle der Ursprungsdatenbank. Die Einstufung der Moleküle basiert auf der Distanz zu einem Referenzset von Verbindungen und jedem anderen Eintrag der Datenbank. Für die Berechnung der Distanz zwischen zwei Einträgen gibt es verschiedene Wege:

1. Deskriptoren: Angenommen *n* Deskriptorwerte sind für die Berechnung der Distanz ausgewählt. So wird die Distanz zwischen zwei Einträgen über die Euklidische Distanz zwischen ihren korrespondierenden Punkten im *n*-dimensionalen Deskriptorraum berechnet.

   Sollten die Deskriptorwerte aus sehr unterschiedlichen Wertebereichen stammen (z.B. 0,01 bis 0,02 vs. -1000 bis 1000), ist es sinnvoll, eine vorherige Normierung der Werte durchzuführen.
2. Fingerprints: Bei Fingerprints wird die Distanz anhand einer Ähnlichkeitsmetrik wie zum Beispiel dem Tanimoto-Koeffizienten berechnet.
3. Konformationsdaten: Die Distanz wird als der mittlere quadratische Abstand (RMSD) zwischen einem Paar molekularer Konformationen berechnet. Diese Berechnungsart benötigt eine Datenbank, in der zu allen Molekülen Konformationsdaten vorhanden sind. [14]

In dieser Arbeit wurde die erste Methode, über die Deskriptoren, für die Modelle 1 bis 25 und 30-31 und die zweite Methode, über die Fingerprints, für die Modelle 26-29 und 36-37, zur Berechnung der am entferntesten liegenden Einträge verwendet (siehe 7.3 und 7.5.3).

3.7 Performance Evaluierung und Validierungsmethoden

QSAR/QSPR Modelle können auf unterschiedliche Weise validiert werden. Es gibt eine große Anzahl an Performance Parametern, zu denen z.B. die korrelations-basierten (R^2, Q^2) oder auch die Fehlerbasierten (error-like) (MAE, RMSE) gehören. Mit ihnen lassen sich sich z.B. die Genauigkeit, Stabilität und Reversibilität messen.

Beim Erstellen von Vorhersagemodellen wird die Qualität des Modells durch die Performance bewertet. Sie gibt Auskunft, wie gut das trainierte Modell die physikochemische Eigenschaft wiedergeben kann. Dies kann durch den Korrelationskoeffizienten R^2 oder der Wurzel des mittleren quadratischen Fehlers (RMSE, root-mean-square error) zwischen der experimentellen Aktivität und der vorhergesagten Aktivität angegeben werden (siehe Tabelle 1).

Die Validierung des Modells kann auf interne sowie auf externe Weise erfolgen. Bei der internen Validierung wird häufig die Cross-Validierung benutzt. Dabei wird eine Probe (leave-one-out, LOO) oder eine Gruppe von Proben (group-leave-out, GLO) in einer systematischen oder zufälligen Art und Weise aus dem Datensatz herausgelassen und vorausgesagt, um vorbeugend gegen das Problem des overfittings (Überanpassung) zu wirken. Diese Überanpassung tritt auf, wenn das Modell nur auf den Trainingsdatensatz hin trainiert wurde und daher neue, d.h. fremde Daten, die des Testdatensatzes zum Beispiel, nur ungenau prognostizieren kann.

Tabelle 1 Validierungsmethoden
Legende: $\boldsymbol{R^2}$ Korrelationskoeffizient; **EXT** externe Validierung; ***RMSE*** Wurzel des mittleren quadratischen Fehlers; $\boldsymbol{Q^2_{LOO}}$ Leave-one-out Kreuzvalidierung; ***RSS*** Residuenqaudratsumme; ***TSS*** Gesamte Abrechnungsqaudratsumme;$\boldsymbol{y_i}$ Einzelexperiment Wert; $\boldsymbol{\overline{y}}$ Durchschnitt der experimentellen Werte; $\boldsymbol{\hat{y}_i}$ Einzel Vorhersagewert; $\boldsymbol{\hat{y}}$ Durchschnitt der Vorhersagewerte; $\boldsymbol{\hat{y}_{i/i}}$ Vorhersagewert für das i-te Besipiel, wenn das i-te Beispiel aus dem Training herausgelassen ist; ***n*** Zahl der Beispiele, ***i*** Beispiel index. Verändert nach [28]

Performance Parameter	**Anwendung**	**Formel**
R^2, R^2_{ext}	Training, externe Validierung	$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \overline{y})^2} = 1 - \frac{RSS}{TSS}$
RMSE	Training, interne und externe Validierung	$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}}$
Q^2_{LOO}	interne Validierung	$Q^2_{Loo} = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_{i/i})^2}{\sum_{i=1}^{n} (y_i - \overline{y})^2} = 1 - \frac{PRESS}{TSS}$
PRESS	interne und externe Validierung	$PRESS = \sum_{i=1}^{n} (y_i - \hat{y}_{i/i})^2$

4 Ergebnisse

Die Modellierungen wurden mit den Programmen MOE® [41] und KNIME (Konstanz Information Miner, Open-Source Programm) [42] durchgeführt. Im Programm MOE wurde auf Erweiterungen der Chemical Computing Group-Support, sogenannte svl-Skripte, zurückgegriffen. Diese waren zum Einen das Autoqsar Skript [39] und zum Anderen das QSAR Evolution Skript [43]. Für eine größere Auswahl an Deskriptoren wurden in KNIME die Erweiterungen von RDKit (Open-Source Cheminformatics Software) [44] und CDK (Open-Source Chemistry Development Kit) [45] hinzugezogen.

4.1. Vorbereitung MWT-Lagoda Datensatz

Für den in der Praxismodularbeit erstellten hF-MWT-17 Datensatz, der im folgendem als MWT-Lagoda Datensatz bezeichnet wird, wurden Mittelwerte für die Bioverfügbarkeitsdaten berechnet. Einträge, die eine Standardabweichung von mehr als 10 hatten, wurden entfernt (siehe Abb. 5). Dies traf auf 34 Einträge zu. Desweiteren wurden die Einträge mit einer prozentualen Abweichung von mehr als 10 % gelöscht, wenn ihre Standardabweichung höher als 5 war. Dadurch verringerte sich der Datensatz um 5,1% auf 930 Einträge (von Anfangs 980 Einträgen).

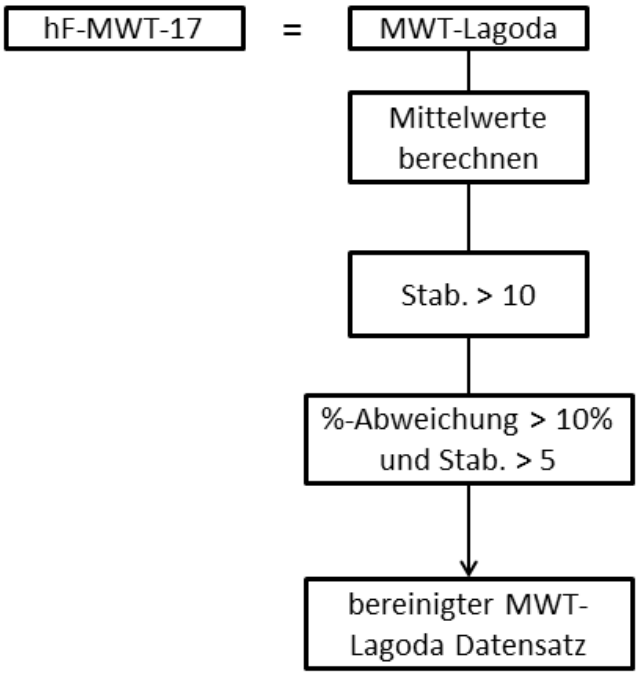

Abbildung 5 Schema zur Mittelwert Berechnung/Bereinigung

4.2. QSPR Modelle

Zur besseren Übersicht über die erstellten Modelle findet sich im Anhang eine tabellarische Zusammenfassung aller Modelle (siehe Tab. 11). Teilausschnitte der Tabelle finden sich in den jeweiligen Unterkapiteln wieder.

Unter 4.2.2 - 4.2.5 werden nur die Modellergebnisse beschrieben, die auf den Trainingsmethoden *PNN, SVM, PLS* und *PCR* beruhen, während in 4.2.6 alle mit dem MOE *Autoqsar* Knoten erstellten Modelle (8, 9, 14, 15, 20, 21) beschrieben werden.

4.2.1 2D-QSPR Modelle – in MOE

Modell 1 ergibt sich aus der Anwendung der *PLS* Methode (siehe 7.4.2.1) auf die 850 Verbindungen des Trainingsdatensatzes (siehe 7.4.1). Es enthält 206 2D-Deskriptoren. Das Skript erzeugt auch eine Protokolldatei, die neben der Geradengleichung eine Abschätzung der Relevanz der verwendeten Deskriptoren für das Modell aufzeichnet. An den für das Modell einflussreichsten Deskriptor wird der Wert 1 (oder 100%) vergeben. Alle weiteren Deskriptoren erhalten abgestufte Werte. Tabelle 2 zeigt 10 Deskriptoren, die die höchste relative Wichtung für das Modell 1 besitzen.

Tabelle 2 Die 10 Deskriptoren mit der höchsten relativen Wichtung und ihrem Vorzeichen von Modell 1

Deskriptor	Vorzeichen	Relative Wichtung [%]
vsa_hyd	+	100
vdw_vol	-	52,9
vdw_area	+	30,7
Q_VSA_POS	+	30,7
Q_VSA_HYD	+	30,7
vsa_pol	-	29,9
apol	-	26,5
vsa_other	-	24,8
Weight	+	22,3
TPSA	-	15,4

Der VSA-Deskriptor *vsa_hyd* enthält die höchste relative Wichtung für Modell 1. Er beschreibt die Summe der Van der Waals Oberfläche von hydrophoben Atomen. Die molekulare Oberfläche ist eine wichtige Größe in der Beschreibung von Molekülen und der Quantifizierung ihrer Interaktionseigenschaften, wie z.B. mit anderen Molekülen oder Lösungsmitteln.

Der Deskriptor *vdw_vol*, der das Van der Waals Volumen berechnet, hat ebenfalls eine hohe relative Wichtung in Modell 1, doch anders als der *vsa_hyd* Deskriptor besitzt er ein negatives Vorzeichen. Berechnet wird er unter der Verwendung einer Konnektivitätstabellen-Näherung. Ähnlich berechnet wird der *vdw_area* Deskriptor. Er gibt die Umgebung der Van der Waals Oberfläche an und steht in der Wichtung der Deskriptoren an dritter Stelle. Eine gleich hohe Bedeutung haben die *Q_VSA_POS* und –*_HYD* Deskriptoren. Sie beschreiben die partiellen Ladungen jedes Atoms und geben die gesamte positive (*Q_VSA_POS*) bzw. hydrophobe (*Q_VSA_HYD*) Van der Waals Oberfläche an.

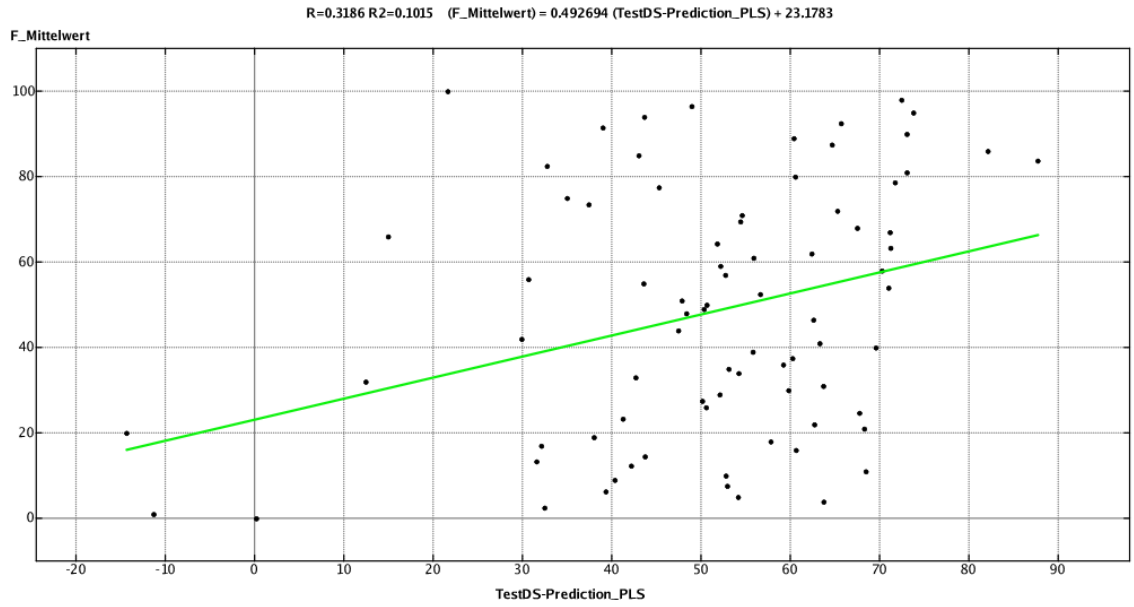

Abbildung 6 Korrelationsplot von Modell 1 Aufgetragen auf der x-Achse sind die Bioverfügbarkeitsmittelwerte des Testdatensatzes gegen die auf der y-Achse aufgetragenen Werte der mit dem QSPR-Modell vorhergesagten Werte. Das Bestimmtheitsmaß hat einen Wert von 0,10.

Die Fähigkeit des Modells, die Bioverfügbarkeit der Moleküle des Trainingssatzes wiederzugeben, wird mit Hilfe des Korrelationskoeffizienten $R_{Training}$ ausgedrückt. Dieser ist mit 0,37 im Vergleich zu publizierten Daten anderer Modelle [22–24] niedrig. Ebenso ist die Kenngröße $R^2_{Training}$ für das kreuzvalidierte Modell mit 0,23 gering. Modell 1 besitzt eine geringe Vorhersagekraft und spiegelt damit die Bioverfügbarkeitswerte des Traingsdatensatzes nur ungenügend wieder. Die geringe Leistungsfähigkeit des Modells zeigt sich ebenfalls bei der Anwendung auf die 80 Verbindungen des Testdatensatzes (siehe Abb. 6). Für diese nicht in den Modellbildungsprozess involvierten

Bioverfügbarkeitswerte beträgt der Regressionskoeffizient R_{Test} 0,32 und der Regressionskoeffizient R^2_{Test} 0,10.

Modell 2 ist eine Reproduktion von Modell 1 mit dem Unterschied, dass statt der *PLS* Methode die *PCR* Methode angewendet wird. Auch hier handelt es sich um eine lineare Trainingsmethode, so dass die Ergebnisse direkt vergleichbar sind. Beide Werte $R_{Training}$ und $R^2_{Training}$ die die Modellgüte ausdrücken sind mit 0,25 und 0,18 geringer als die von Modell 1. Bei den Werten für die Vorhersage des fremden Testdatensatz besitzt der Regressionskoeffizient R_{Test} einen Wert von 0,32 und der Regressionskoeffizient R^2_{Test} einen Wert von 0,10. Damit liegen sie auf einem gleichen Niveau mit Modell 1.

Tabelle 3 Die 10 Deskriptoren mit der höchsten relativen Wichtung und ihrem Vorzeichen von Modell 2

Deskriptor	Vorzeichen	Relative Wichtung [%]
vdw_vol	-	100
TPSA	-	73,9
weinerPath	+	64,7
SlogP_VSA0	-	54,8
SMR_VSA5	-	52,3
SMR_VSA6	-	51,5
vsa_hyd	+	40,3
SlogP_VSA5	-	31,8
vsa_acc	+	30,2
vsa_don	+	29,8

Bei Modell 2 haben alle 10 Deskriptoren mit der höchsten relativen Wichtung (siehe Tabelle 3) Prozentwerte, die über 29 liegen. Bei Modell 1 trifft dies nur auf die ersten 5 Deskriptoren zu. Dies bedeutet, dass bei Modell 2 viele Deskriptoren einen hohen Einfluss auf das Modell ausüben, während bei Modell 1 dies auf nur wenige Deskriptoren zutrifft.

Der Deskriptor mit der höchsten relativen Wichtung ist der *vdw_vol* Deskriptor, der das Van der Waals Volumen des Moleküls ausgibt. Einen weiteren hohen Einfluss auf das Modell nimmt der *TPSA* Deskriptor ein. Er ermittelt die topologische polare Molekularoberfläche eines Moleküls. Sein Wert gibt Aufschluss darüber, wie gut ein Molekül Zellen durchdringen kann. Der *weinerPath* Deskriptor zählt zu den ältesten

topologischen Deskriptoren. Definiert ist der Wiener Index, auf dem der *weinerPath* beruht, als die Summe der Längen der kürzesten Wege zwischen allen Paaren von Eckpunkten im chemischen Graphen, die nicht Wasserstoffatome im Molekül darstellen. Deskriptoren, die die Hydrophobizität beschreiben, stehen an vierter, siebter und achter Position der Tabelle (*SlogP_VSA0, vsa_hyd* und *SlogP_VSA5*). Die beiden Deskriptoren an fünfter und sechster Position der Tabelle beschreiben die Oberflächenanteile von Molekülen mit einer hohen Polarisierbarkeit. Die beiden Deskriptoren an letzter Position der Tabelle (*vsa_acc* und *vsa_don*) beschreiben eine Näherung der Summe der Van der Walls Oberfläche von Wasserstoffbrücken-Akzeptoren und Wasserstoffbrücken-Donatoren.

Als dritte Methode wurde bei Modell 3 die nicht lineare Methode des genetischen Algorithmuses (GA-MLR) verwendet. Es wird wie unter 7.4.2.3 beschrieben vorgegangen. Ein großer Unterschied zu den bisherigen Methoden *PLS* und *PCR* ist, dass eine immer wieder neue Kombination der Deskriptoren in der Modellbildungsphase möglich ist. Bei den anderen Methoden steht ein durch die Datenreduktion entfernter Deskriptor für die weitere Modellbildung nicht mehr zur Verfügung. Zum Startzeitpunkt der Modellbildungsphase ist die Anzahl der Deskriptoren auf 4 fixiert. Diese Fixierung kann jedoch während des Modellierens beliebig erhöht werden.
Das Ergebnis der Vorhersagefähigkeit ist mit einem $R_{Trainings}$ Wert von 0,20 und einem $R^2_{Trainings}$ Wert von 0,19 leicht niedriger als die der Modelle 1 und 2. Die Werte für den Testdatensatz liegen ebenfalls mit 0,30 (R_{Test}) und 0,09 (R^2_{Test}) leicht unter denen von Modell 1 und 2.
Der Beitrag, den ein Deskriptor zur Vorhersagefähigkeit liefert, lässt sich aufgrund der verwendeten Methode des genetischen Algorithmuses nur schwer abschätzen. Eine Möglichkeit besteht darin, sich den Verlauf der Wichtung einzelner Deskriptoren während des genetischen Algorithmuses anzuschauen. Ein solcher Verlauf ist in Abbildung 7 abgebildet. Das Diagramm zeigt die Wichtung einzelner Deskriptoren von Modell 3, während der Modellbildungsphase über einen Zeitraum von 32.500 Generationen. Zu sehen ist, dass es Deskriptoren gibt, die von Anfang an einen hohen Einfluss besitzen und andere, deren Einfluss über die gesamte Modellbildungsphase gering bleibt.

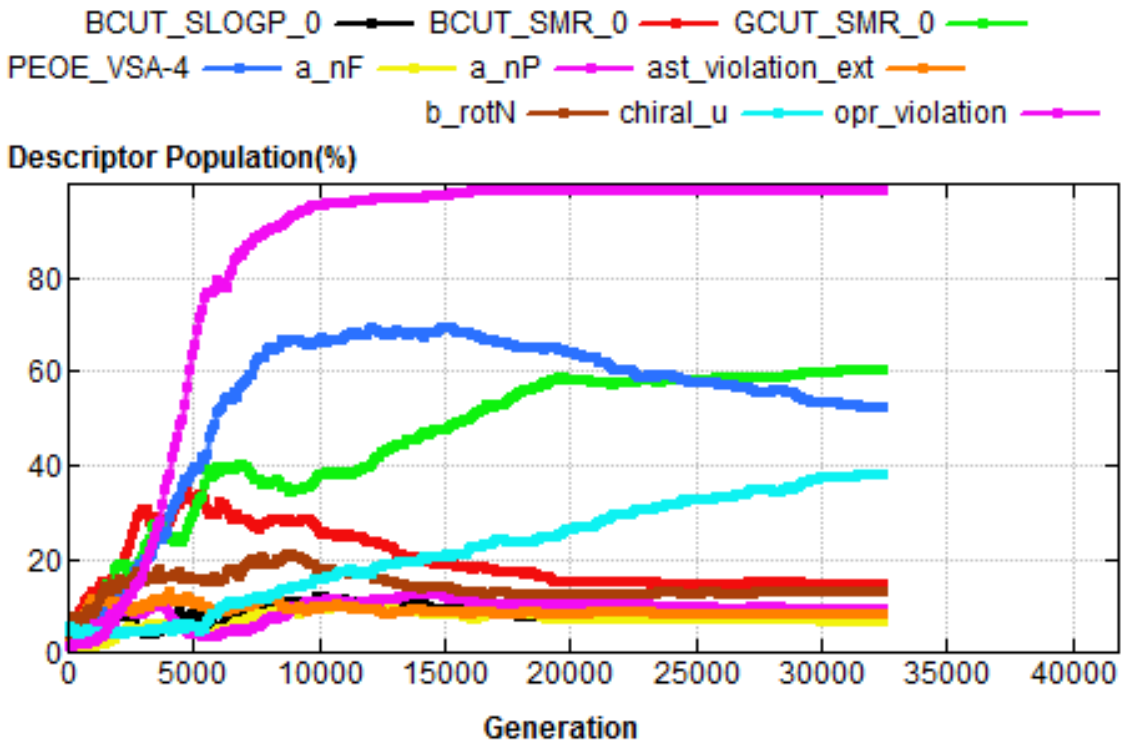

Abbildung 7 Wichtung einzelner Deskriptoren bei Modell 3. Abgebildet ist der Verlauf der Wichtung einzelner Deskriptoren während 32.500 Generationen bei Erstellung von Modell 3.

Ein Deskriptor, der vom Start des Algorithmuses an Gewicht gewinnt und auch bis zum Ende seine hohe Wichtung behält, ist z.B der Deskriptor, der die Anzahl an Verstößen des „*Oprea's lead-like test*“ zählt (*opr_violation*). Dieser Deskriptor ist eine Weiterentwicklung des „rule of five“ Deskriptors und beinhaltet zudem die Anzahl an rotierbaren und rigiden Bindungen sowie die Anzahl an Ringen.

Andere Deskriptoren (*PEOE_VSA-4* und *BCUT_SMR_0*) scheinen zunächst eine entscheidende Rolle für das Modell zu spielen, verlieren jedoch im Lauf der Optimierung wieder ein wenig an Bedeutung und damit an Gewicht. Der Deskriptor *PEOE_VSA-4* beschreibt Areale mit negativen Partialladungen. Der andere *BCUT_SMR_0* Deskriptor beruht auf einem atomaren Beitrag zur molaren Refraktivität anstatt auf Partialladungen. Eine dritte Klasse von Deskriptoren (z.B. *b_rotN, a_nF* und *a_nP*) hat von Beginn an wenig Gewicht auf das Modell und verändert diesen Status auch nicht. Dieser *b_rotN* Deskriptor beschreibt die Anzahl an rotierbaren Bindungen eines Moleküls, während der Deskriptor *a_nF* die Anzahl an Fluoratomen zählt und der Deskriptor *a_nP* die Anzahl an Phosphoratomen zählt.

Die folgende Tabelle 4 fasst die Ergebnisse der drei Modelle, die im Programm MOE erstellt wurden, noch einmal zusammen.

Tabelle 4 Modelle mit 2D Deskritptoren - MOE

Modellnummer	Trainingsmethode	Deskriptoranzahl	R_{Test}	R^2_{Test}
1	PLS	206	0,32	0,10
2	PCR	206	0,32	0,10
3	GA	206	0,30	0,09

4.2.2 2D-QSPR Modelle – in KNIME

Mit dem Programm KNIME wurden weitere 2D-QSPR Modelle erstellt. Dieses Programm wurde ausgewählt, da hiermit weitere nicht-lineare Trainingsmethoden wie das *PNN* oder die *SVM* für die Modellierung zur Verfügung stehen. Zudem ist neben weiteren Integrationen von verschiedenen Programmen eine mit dem Programm MOE vorhanden. Dies ermöglicht es, die fast exakt gleichen Deskriptoren wie unter 4.2.1 beschrieben, auszuwählen. Die Ergebnisse mit dieser Deskriptorauswahl sind in Tabelle 5 zusammengefasst.

Tabelle 5 Modelle mit 2D Deskriptoren - KNIME

Modellnummer	Trainingsmethode	Deskriptoranzahl	R_{Test}	R^2_{Test}
4	PNN	204	0,27	0,08
5	SVM	204	0,00	0,00
6	PLS	204	0,33	0,11
7	PCR	204	0,30	0,09

Darüberhinaus können weitere molekulare Deskriptoren durch die Integration mit dem RDKit und CDK in weiteren Modellen verwendet werden.

Modell 4 ist durch die Anwendung der *PNN* Methode nach der Versuchsvorschrift von 7.5.1.1 erstellt worden. Es enthält 204 Deskriptoren. Einen mathematischen Zusammenhang der Gewichtung der einzelnen Deskriptoren, wie unter 4.2.1 beschrieben ist, erstellt das Programm nicht. Die Fähigkeit des Modells, die Bioverfügbarkeit eines fremden Datensatzes (des Testdatensatzes) wiederzugeben, ist nicht höher als bei den

vorherigen Modellen 1-3. Der Korrelationskoeffizient R_{Test} liegt bei 0,27 und das R^2_{Test} bei 0,08.

Modell 5 ist mit Anwendung der *SVM* Methode nach 7.5.1.2 erstellt worden. Eine Vorhersagefähigkeit ist mit einem Korrelationskoeffizient R_{Test} von 0,0 und einem R^2_{Test} von 0,0 nicht gegeben.

Als Trainingsmethode bei Modell 6 ist die *PLS* verwendet worden. Das Modell wurde wie in 7.5.1.3 beschrieben erstellt. Die Vorhersagefähigkeit ist mit einem Korrelationskoeffizienten von 0,33 (R_{Test}) und einem Bestimmtheitsmaß von 0,11 (R^2_{Test}) höher als die der Modelle 5-6.

Die Trainingsmethode für die Erstellung von Modell 7 ist die *PCR*. Das Modell wurde nach Anwendung der Versuchsvorschrift von 7.5.1.4 erstellt. Die Vorhersagefähigkeit ist mit folgenden Werten R_{Test} 0,33 und R^2_{Test} 0,09 ähnlich hoch wie die von Modell 4.

4.2.3 Modelle mit 2D und 3D Deskriptoren - in KNIME

Die Modelle 10-13 wurden ebenfalls in KNIME erstellt. Dabei gilt die beschriebene Versuchsvorschrift aus Kapitel 7.5.2 Die Ergebnisse der einzelnen Methoden sind in Tabelle 6 aufgelistet.

Tabelle 6 Modelle mit 2D und 3D Deskriptoren - KNIME

Modellnummer	Trainingsmethode	Deskriptoranzahl	R_{Test}	R^2_{Test}
10	PNN	321	0,32	0,10
11	SVM	308	0,20	0,04
12	PLS	321	0,33	0,11
13	PCR	321	0,30	0,09

Die höchsten R_{Test} und R^2_{Test} Werte finden sich bei Modell 12 mit R_{Test} 0,33 und R^2_{Test} 0,11 wieder. Die eingesetzte Trainingsmethode ist dabei die *PLS* Methode. Die niedrigsten Werte der vier Modelle 10-13 sind mit 0,20 (R_{Test}) und 0,04 (R^2_{Test}) bei Modell 13, das mit der *SVM* Methode erstellt wurde, zu finden. Eine verbesserte Vorhersagefähigkeit, aufgrund der größeren Auswahl an Deskriptoren die zur Verfügung standen, ist damit ausgeblieben. Bei Modell 11 konnte zudem nur auf eine um 13

verringerte Anzahl von Deskriptoren für die Modellierung zurückgegriffen werden. Von diesen 13 Deskriptoren konnte bei 10 der Datentyp (Double oder Integer) nicht richtig interpretiert werden. Die restlichen 3 Deskriptoren berechnen die Anzahl an Bor Atomen, die Anzahl an Iod Atomen und die Anzahl an Molekülen. Eine Erklärung, warum diese 3 Deskriptoren nicht verwendet werden konnten, könnte darin liegen, dass die beiden Atome Bor und Iod nicht in den Molekülen vorkommen und das es sich nie um mehrere Moleküle bei einem Datenbankeintrag handelt.

4.2.4 Modelle mit fast allen Deskriptoren - in KNIME

Die Modelle 16-19 ergeben sich aus der Anwendung der Versuchsvorschrift von 7.5.2. Tabellarisch zusammengefasst sind die Ergebnisse in Tabelle 7.

Tabelle 7 Modelle mit fast allen Deskriptoren – KNIME

Modellnummer	Trainingsmethode	Deskriptoranzahl	R_{Test}	R^2_{Test}
16	PNN	484	0,35	0,12
17	SVM	470	0,17	0,03
18	PLS	484	0,14	0,02
19	PCR	484	0,14	0,02

Von den Modellen 16-19 verfügt das Modell, das mit der *PNN* Trainingsmethode erstellt wurde, über die höchsten Vorhersagewerte. Der R_{Test} Wert liegt bei 0,35 und R^2_{Test} bei 0,12. Die niedrigsten Werte sind bei den Modellen 18 und 19, die mit den linearen Trainingsmethoden *PLS* und *PCR* erstellt wurden, zu finden. Die Vorhersagewerte beider Modelle sind dabei das erste Mal deutlich geringer im Vergleich zu den vorherigen Modellen, die auf den gleichen Trainingsmethoden (PLS und PCR) und weniger Deskriptoren beruhen (Modell 7, 8 und 12, 13).

Bei Modell 17 konnten zudem 14 Deskriptoren weniger für die Modellierung eingesetzt werden. Von diesen 14 Deskriptoren konnte bei 10 der Datentyp (Double oder Integer) nicht richtig interpretiert werden. Von einem Deskriptor, der die chemische Formel berechnet, konnte der String nicht verarbeitet werden und die restlichen 3 Deskriptoren berechnen die Anzahl an Bor Atomen, die Anzahl an Iod Atomen und die Anzahl an Molekülen. Eine Erklärung, warum diese 3 Deskriptoren nicht verwendet werden konnten, könnte darin liegen, dass die beiden Atome Bor und Iod nicht in den Molekülen

vorkommen und dass es sich nie um mehrere Moleküle bei einem Datenbankeintrag handelt.

4.2.5 Modelle mit allen Deskriptoren - in KNIME

Für die Modelle 22-25 wurden weitere Deskriptoren benutzt, die auf dem Programm MOPAC (Molecular Orbital PACkage) beruhen. Es wurde entwickelt um semi-empirische quantenchemische Algorithmen zu implementieren. 15 weitere Deskriptoren konnten damit für die Modellierung, die nach 7.5.2 durchgeführt wurde, hinzugefügt werden. Die Ergebnisse der verschiedenen Methoden sind in Tabelle 8 aufgelistet.

Tabelle 8 Modelle mit allen Deskriptoren - KNIME

Modellnummer	Trainingsmethode	Deskriptoranzahl	R_{Test}	R^2_{Test}
22	PNN	499	0,06	0,00
23	SVM			
24	PLS	499	0,37	0,14
25	PCR	499	0,37	0,14

Anders als bei den Modellen aus 4.2.4 haben die Modelle in 4.2.5, die auf den linearen Trainingsmethoden *PLS* und *PCR* beruhen, höhere Performancewerte. Sie liegen bei beiden Methoden bei 0,37 (R_{Test}) und 0,14 (R^2_{Test}). Die beiden anderen Modelle (22 und 23), die auf nicht linearen Trainingsmethoden beruhen, haben dabei die niedrigsten Vorhersagewerte von allen bisherigen Modellen (Model 4-7, 10-13 und 16-19). Bei Modell 23, das auf der *SVM* Methode beruht, konnte kein Ergebnis erzielt werden, da die Deskriptorwerte der MOPAC Deskriptoren nicht korrekt vom *SVM learner* interpretiert werden konnten.

4.2.6 Modelle mit Autoqsar in KNIME

Die Modelle aus Tabelle 9 sind nach Anwendung der Versuchsvorschrift aus Kapitel 7.5.1.5 erstellt worden. Als Trainingsmethoden wurden die PLS und die PCR Methode verwendet. Der Unterschied zu den bisher erstellten Modellen mit diesen Methoden liegt darin, dass hier auf das gleiche Autoqsar svl-Skript, jedoch im Programm KNIME, zurückgegriffen wurde, welches bei Modell 1 und 2 Verwendung fand.

Modelle mit Einbeziehung der 15 MOPAC Deskriptoren konnten nicht durchgeführt werden. Dies hängt mit den berechneten Deskriptorwerten zusammen, die vom Autoqsar Skript nicht richtig interpretiert werden konnten.

Tabelle 9 Modelle mit Autoqsar Funktion - KNIME

Modellnummer	Trainingsmethode	Deskriptoranzahl	R_{Test}	R^2_{Test}
8	PLS-Autoqsar	204	0,33	0,11
9	PCR-Autoqsar	204	0,30	0,09
14	PLS-Autoqsar	321	0,28	0,08
15	PCR-Autoqsar	321	0,28	0,08
20	PLS-Autoqsar	484	0,14	0,02
21	PCR-Autoqsar	484	0,14	0,02

Die Fähigkeit der Modelle 8 und 9, die Bioverfügbarkeit eines fremden Datensatzes (des Testdatensatzes) wiederzugeben ist niedriger als die der Modelle 1 und 2, die mit dem gleichen Autoqsar Skript, jedoch in einem anderen Programm, erstellt wurden. Für Modell 8 liegt der Korrelationskoeffizient R_{Test} bei 0,33 und der R^2_{Test} Wert bei 0,11 und für Modell 9 liegen die Werte bei 0,30 (R_{Test}) und 0,09 (R^2_{Test}).

Die Vorhersagefähigkeit der Modelle 14 und 15 hat sich durch die zusätzliche Verwendung von weiteren 117 Deskriptoren nicht erhöht. Für Modell 14 liegt der Korrelationskoeffizient R_{Test} bei 0,28 und der R^2_{Test} Wert bei 0,08 und für Modell 15 liegen die Werte bei 0,28 (R_{Test}) und 0,08 (R^2_{Test}). Im Vergleich zu den Modellen 12 und 13, die mit der gleichen Trainingsmethode und Deskriptoranzahl erstellt wurden, jedoch nicht mit dem Autoqsar Skript und nicht mit dem *PLS/PCR Learner* und *Predictor*, sind die erzielten Performancewerte leicht niedriger.

Durch die weitere Erhöhung der Deskriptoranzahl auf 484 konnte keine erhöhte Vorhersagefähigkeit erzielt werden. Der Korrelationskoeffizient von Modell 20 liegt bei 0,14 (R_{Test}) und der R^2_{Test} Wert bei 0,02. Modell 21 besitzt die gleichen Performancewerte (wie Modell 20).

4.2.7 Andere Aufteilung des Datensatzes - Modelle mit Autoqsar in KNIME

Die Modelle 26-39 ergeben sich aus der Anwendung der Versuchsvorschrift von 7.5.3. Anders als bei der Erstellung der ersten Modelle (Modell 1-25) lag hier auch der Fokus auf der Frage, wie groß der Einfluss der Methode ist, mit der der Datensatz in Trainings- und Testdatensatz eingeteilt wird und ob es einen Unterschied macht, wenn die Deskriptoren in Programm MOE oder KNIME für die die Modellgenerierung berechnet werden. Tabellarisch zusammengefasst sind die Ergebnisse in Tabelle 10.

Tabelle 10 Modelle mit anderer Aufteilung des Datensatzes. Erklärung: **in KNIME** = 2D Deskriptorberechnung in KNIME und als *.mdb abgespeichert, **in KNIME-fit** = 2D Deskriptorberechnung in KNIME und als *.fit abgespeichert, **in MOE** = 2D Deskriptorberechnung in MOE und als *.mdb abgespeichert und **in MOE-fit** = 2D Deskriptorberechnung in MOE und als *.fit abgespeichert.

Modellnummer	Deskriptor-Berechnung	Datenbank-Split Methode	R_{Test}	R^2_{Test}
26	in KNIME	Fingerprint	0,62	0,38
27	in KNIME-fit	Fingerprint	0,62	0,38
28	in MOE	Fingerprint	0,52	0,27
29	in MOE-fit	Fingerprint	0,62	0,38
30	in KNIME	Deskriptoren	0,36	0,13
31	in KNIME-fit	Deskriptoren	0,53	0,28
32	in KNIME	Random Sample1	0,28	0,08
33	in KNIME-fit	Random Sample1	0,53	0,28
34	in KNIME	Random Sample2	0,38	0,14
35	in KNIME-fit	Random Sample2	0,41	0,17
36	in KNIME	Random Sample3	0,43	0,19
37	in KNIME-fit	Random Sample3	0,61	0,37
38	in KNIME	Fingerprint normalisiert	0,12	0,01
39	in MOE-fit	Fingerprint normalisiert	0,66	0,44

Die Modelle 26-29 wurden nach der Versuchsvorschrift von 7.5.3.1 erstellt, dass heißt, der Datensatz wurde zu Beginn der Modellierung nach der Fingerprint Methode in Trainings- und Testdatensatz gesplittet.

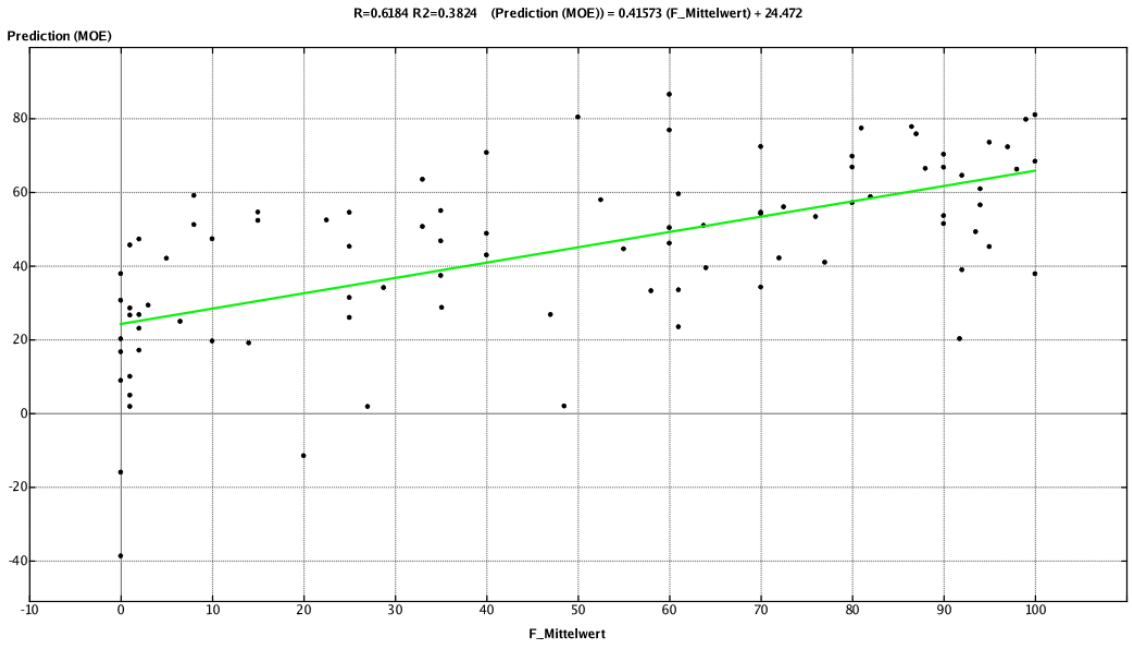

Abbildung 8 Korrelationsplot von Modell 26
Aufgetragen auf der x-Achse sind die Bioverfügbarkeitsmittelwerte des Testdatensatzes gegen die auf der y-Achse aufgetragenen Werte der mit dem QSPR-Modell vorhergesagten Werte. Das Bestimmtheitsmaß hat einen Wert von 0,38

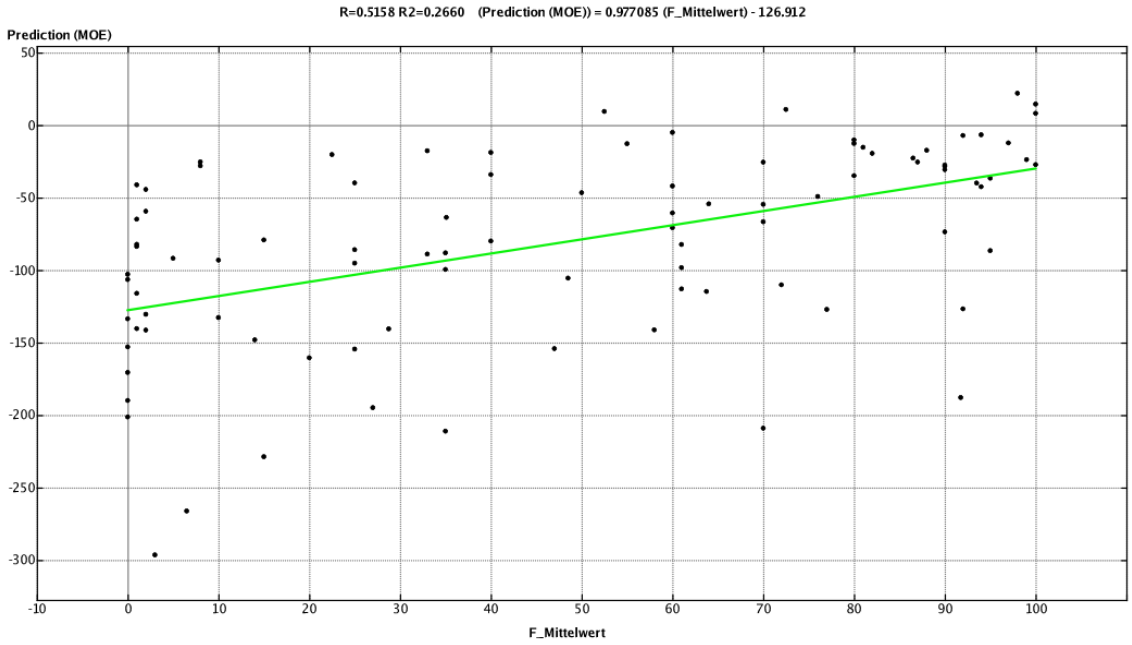

Abbildung 9 Korrelationsplot von Modell 28
Aufgetragen auf der x-Achse sind die Bioverfügbarkeitsmittelwerte des Testdatensatzes gegen die auf der y-Achse aufgetragenen Werte der mit dem QSPR-Modell vorhergesagten Werte. Das Bestimmtheitsmaß hat einen Wert von 0,27

Die Fähigkeit der Modelle 26-29, die Bioverfügbarkeit eines fremden Datensatzes (des Testdatensatzes) wiederzugeben, ist deutlich höher als bei allen vorherigen Modellen (1-25). Der Korrelationskoeffizient R_{Test} von Modell 26 liegt bei 0,62 und das R^2_{Test} bei 0,38 (siehe Abb. 8), während der R_{Test} von Modell 28 bei 0,52 und das R^2_{Test} bei 0,27 liegt (siehe Abb. 9). Der Unterschied zwischen diesen beiden Modellen liegt nur in der anfänglichen Deskriptorberechnung. Während bei Modell 26 dies in KNIME durchgeführt wurde, fand die Berechnung bei Modell 28 in MOE statt. Es wirkt sich also auf die Modellierung in KNIME aus, wenn die Deskriptorwerte vorher berechnet werden. Ein weiterer Punkt, der beim Vergleich der beiden Diagramme (Abb. 8 und 9) auffällt ist, dass die Vorhersagewerte von der als *.mdb gespeicherten Datei (Abb. 9) sehr weit in den negativen Bereich gehen (bis -300).

Die Modelle 30-31 wurden nach der Versuchsvorschrift von 7.5.3.2 erstellt, das heißt, der Datensatz wurde nach der Deskriptor Methode zu Beginn der Modellierung in Trainings- und Testdatensatz gesplittet. Die Fähigkeit der beiden Modelle, die Bioverfügbarkeit eines fremden Datensatzes wiederzugeben, ist im Vergleich zu den Modellen 26-29 geringer. Der Korrelationskoeffizient R_{Test} liegt von Modell 30 bei 0,36 und das R^2_{Test} bei 0,13; während von Modell 31 der R_{Test} bei 0,53 und das R^2_{Test} bei 0,28 liegt. Eigentilich sollten beide Modelle das gleiche Ergebnis erzielen. Weiter fällt auf, dass der R^2_{Test} Wert von Modell 30 in einem ähnlichen Bereich liegt wie der von den vorherigen Modellen 6, 12 und 24 (0,11 – 0,11 – 0,14), die auch mit der *PLS* Trainingsmethode erstellt wurden.

Die Modelle 32-37 wurden nach der Versuchsvorschrift von 7.5.3.3 erstellt, das heißt, der Datensatz wurde nach der Random Sampling Methode zu Beginn der Modellierung in Trainings- und Testdatensatz gesplittet. Auch hier zeigt sich das gleiche Bild wie bei den vorherigen Modellen: Solche, die im Programm KNIME erstellt wurden (Modelle 32, 34, 36) haben geringere Vorhersagewerte als die, die in MOE mit dem Testdatensatz validiert wurden (Modell 33, 35, 37). Der Korrelationskoeffizient der in KNIME erstellten Modelle besitzt einen R_{Test} Wert der zwischen 0,28 und 0,43 und ein R^2_{Test} Wert der zwischen 0,08 und 0,19 liegt. Der Korrelationskoeffizient von den in MOE validierten Modellen liegt mit einem R_{Test} Wert der zwischen 0,41 und 0,17 und einem R^2_{Test} Wert zwischen 0,61 und 0,37 im Vergleich viel höher.

Die Modelle 38 und 39 wurden nach der Versuchsvorschrift von 7.5.3.4 erstellt. Die Modellierung unterscheidet sich zu den Modellen 26 und 27 in nur einem Punkt, nämlich der Normalisierung der Daten. Die Normalisierung der Daten wurde vor dem Einteilen des Datensatzes unternommen. Die Vorhersagewerte sind durch diesen Schritt gesunken (Modell 38) sowie gestiegen (Modell 39). Die Vorhersagefähigkeit von Modell 38, das einen R_{Test} Wert von 0,12 und ein R^2_{Test} von 0,01 hat, ist von allen Modellen (26-39), die nach der Versuchsvorschrift von 7.5.3 erstellt wurden, am geringsten. Modell 39 hat dahingegen von allen Modellen (26-39) die höchste Vorhersagefähigkeit, mit einem R_{Test} Wert von 0,66 und ein R^2_{Test} von 0,44. Die Diskrepanz zwischen diesen beiden R^2_{Test} Werten (0,01 und 0,44) ist im Vergleich zu allen erstellten Modellen am höchsten.

5 Diskussion

Die Ergebnisse der 2D QSPR Modelle, die mit dem Programm MOE erzielt wurden (siehe 4.2), besitzen im Vergleich zu literaturbekannten Modellen [22–24] eine sehr viel niedrigere Vorhersagefähigkeit. Die Regressionskoeffizienten der zitierten Modelle liegen zwischen einem R^2 von 0,71 und 0,85.

Die besten Modelle, die mit dem Programm MOE erstellt wurden, Modell 1 und 2, besitzen eine Vorhersagefähigkeit für den Testdatensatz von einem R^2, das bei 0,10 liegt. Dass der Wert für das R^2_{Test} beider Modelle gleich ist, lässt sich in der Ähnlichkeit ihrer zugrunde liegenden Trainingsmethoden erklären. Beide Methoden erstellen Vorhersagevariablen (Komponenten) auf der Basis linearer Kombinationen der Originalvorhersagevariable. Sie unterscheiden sich nur in der Art, wie sie diese konstruieren. Die PCR Methode erstellt Komponenten, die die beobachtete Variabilität in den Vorhersagevariablen erklären ohne dabei die Antwortvariable zu berücksichtigen. Die PLS Methode dagegen berücksichtigt die Antwortvariable und führt daher oft zu Modellen, die in der Lage sind, die Antwortvariable in weniger Komponenten zu erklären.

Auch aufgrund der geringen Vorhersagefähigkeit der Modelle 1 und 2 wurde als weitere Trainingsmethode für die Modellierung der genetische Algorithmus verwendet. Anders als bei der PCR und PLS Methode handelt es sich hier um eine nicht lineare Methode. Diese lässt neue Kombinationen der Diskriptoren zu und löscht nicht wie die beiden anderen Methoden im Schritt der Datenreduktion Deskriptoren, die dann für die folgenden Modellierungsschritte nicht mehr zur Verfügung stehen. Die Vorhersagefähigkeit dieser Methode (Modell 3) ist mit einem R^2_{Test} Wert von 0,09 jedoch nicht höher.

Da die Vorhersagegüte der drei Modelle so niedrig ausfiel, wurde die Idee verfolgt, weitere Trainingsmethoden in die Modellierung mit einzubeziehen. Eine andere Idee war es, die Deskriptoranzahl zu erhöhen, um mehr Informationen für die Modellierung zu besitzen. Für diesen Ansatz wurde mit der Modellierung in das Programm KNIME gewechselt. Dieses besitzt neben vielen anderen auch eine MOE Implementierung. Aufgrund des CDKs und RDKits konnte die Deskriptoranzahl auf bis zu 499 erhöht werden. An weiteren Trainingsmethoden kamen die Methoden des Probabilistischen Neuronalen Netzwerks (PNN) und die der Support Vector Machine (SVM) hinzu.

In einem ersten Schritt wurden Modelle, die die gleiche Anzahl an Deskriptoren wie die ersten drei Modelle hatten, erstellt. Als Methoden wurden neben der bekannten PLS und PCR nun auch die PNN und die SVM Methode für die Modellierung verwendet. Die höchste Vorhersagefähigkeit dieser vier Modelle wurde mit der PLS Methode (Modell 6) erzielt. Der R^2_{Test} Wert liegt bei 0,11 und ist damit leicht höher als das R^2 der Modelle 1 und 2. Dahingegen ist die Vorhersagefähigkeit der zweiten linearen Trainingsmethode PCR mit seinem R^2_{Test} Wert von 0,09 (Modell 7) geringer.
Weiter fällt der große Performance-Unterschied zwischen den beiden nicht-linearen Trainingsmethoden auf. Während die Vorhersagefähigkeit von Modell 4, das mit der PNN Methode erstellt wurde, einen R^2_{Test} Wert von 0,08 besitzt, liegt der R^2_{Test} Wert von Modell 5, das mit der SVM Trainingmethode erstellt wurde, bei 0,0.

In einem weiteren Zyklus wurden Modelle, die neben 2D auch 3D Deskriptoren beinhalteten, erstellt. Eine Verbesserung der Vorhersagefähigkeit blieb trotz erhöhter Deskriptoranzahl aus. Zwar erhöhten sich die R^2_{Test} Werte auf 0,11 (PNN, Modell 10) und 0,04 (SVM, Modell l1), doch wurden bei den anderen beiden Methoden keine Verbesserungen gemessen. Sie erzielten dieselben Performancewerte 0,11 (PLS, Modell l2) und 0,09 (PCR, Modell l3) wie bei der Verwendung von 2D Deskriptoren. Dass eine Verbesserung ausblieb, liegt an der Tatsache, dass der Einsatz von 3D Deskriptoren erst Sinn ergibt, wenn von den verwendeten Molekülen Konfomere vorliegen.

Mit Hilfe des RDKits und der CDK Erweiterung wurde in KNIME die Deskriptorenanzahl um weitere 163 Deskriptoren erhöht. Erneut wurden vier neue Modelle mit den nun 484 Deskriptoren erstellt. Bei einem der vier neuen Modelle konnte eine leichte Erhöhung der Vorhersagefähigkeit erzielt werden. Diese traf auf das Modell 16 zu, das mit der PNN Methode erstellt wurde und einen R^2_{Test} Wert von 0,12 hatte. Bei den weiteren drei Modellen verringerten sich die Performancewerte auf einen R^2_{Test} Wert von 0,03 (SVM, Modell 17) und 0,02 (PLS, Modell 18 und PCR, Modell 19). Dass die Erhöhung der Deskriptoranzahl auf nun 484 nur bei der PNN Methode zu einem besseren Performancewert führte, lässt vermuten, dass für die anderen Methoden kein Informationsgewinn in den 163 neuen Deskriptorwerten lag.
In einem letzten Schritt wurden weitere 15 quantenchemische Deskriptoren zur Deskriptorauswahl hinzugefügt. Die Berechnung dieser 15 Deskriptoren beruht auf semi-

empirischen quantenchemischen Algorithmen und nahm 12 Stunden innerhalb der Modellierung der Modelle ein. Eine Verbesserung der Vorhersagefähigkeit konnte mit diesen Deskriptoren nicht erzielt werden. Zudem konnte mit der SVM Methode kein Modell erstellt werden. Womit dies zusammenhängt konnte nicht zufriedenstellend ermittelt werden. Es ist davon auszugehen, dass das Problem mit den Deskriptorwerten zusammenhängt.
Die Vorhersagefähigkeit des Modells 22, das mit der PNN Methode erstellt wurde, verringerte seinen R^2_{Test} Wert auf 0,0. Dies ist im Vergleich zu allen anderen Modellen, die mit der PNN Methode erstellt wurden, der kleinste R^2_{Test} Wert. Einzig die beiden Modelle 23 und 24, die mit linearen Trainingsmethoden erstellt wurden, verbesserten ihre Performancewerte von 0,02 (Modell 18 und 19) auf einen R^2_{Test} Wert von 0,14.

Der Modellierungsaufbau der Modelle 8 und 9 ist mit dem der Modelle 1 und 2 vergleichbar. Es wurde jeweils die gleiche Trainingsmethode und die fast gleiche Deskriptoranzahl (2 weniger bei Modell 8 und 9) verwendet. Nur das Programm war ein anderes (KNIME statt MOE). Dies führte im Vergleich zu leicht niedrigeren Performancewerten. Durch die Erhöhung der Deskriptoranzahl bei den Modellen 14 und 15 verringerte sich die Vorhersagefähigkeit leicht. Stark verringerte sie sich bei der Verwendung von 484 Deskriptoren (Modell 20 und 21). Hier lagen der R^2_{Test} Werte jeweils nahe null mit 0,02.

Verschiedene Gründe können für die niedrige Vorhersagefähigkeit der Modelle aufgeführt werden. Einer liegt mit großer Wahrscheinlichkeit in der Beschaffenheit des MWT-Lagoda Datensatzes. Dieser Datensatz verfügt zwar über 930 Moleküle, womit er größer ist als der von Turner und Moda. Jedoch liegt nicht annähernd eine Normalverteilung der Bioverfügbarkeitswerte vor. Die Extreme liegen im niedrigen und hohen Bioverfügbarkeitsbereich wie Abbildung 13 (siehe Anhang) zeigt. Für das Trainieren des Modells ist dies offenbar eine unvorteilhafte Ausgangslage, da die Extreme dabei im Vergleich zum Rest (Bioverfügbarkeitsbereich zwischen 20-80) überpräsentiert sind und das Modell damit Bioverfügbarkeitswerte, die nicht in diesen Bereichen liegen, weniger gut voraussagen kann.
Bei der Analyse, warum literaturbekannte QSPR Modelle viel höhere Performancewerte erzielen, wurde das Paper von Turner [22] im Vergleich zu den Modellen 1 bis 25 analysiert.

Der Korrelationskoeffizient von Turner liegt bei 0,72. Er ist damit um 0,35 höher, als der von den Modellen (14, 15, 24 und 25), bei denen der Korrelationskoeffizient bei 0,37 liegt. Turners Wert basiert auf der Vorhersage von 10 ausgewählten Molekülen, deren Bioverfügbarkeitswerte in einem Bereich zwischen 22 und 90 liegen (siehe Abb. 14, Anhang). Wenn man sich den zugrunde liegenden Trainingsdatensatz anschaut, fällt auf, dass in diesem Abschnitt (Bioverfügbarkeitsbereich 20-90) annähernd eine Normalverteilung vorliegt. Es kann davon ausgegangen werden, dass der Korrelationskoeffizient geringer ausfallen würde, wenn die 10 Moleküle aus dem gesamten Bioverfügbarkeitsbereich (1 bis 100) stammen würden. Die 80 vorhergesagten Moleküle der Modelle 1 bis 25 stammen hingegen vom gesamten Bioverfügbarkeitsbereich und der zugrunde liegende Trainingsdatensatz (siehe Abb. 13) besitzt eher die Form einer gestauchten Parabel.

Ein zweiter Erklärungsansatz ist in der Methode zur Auswahl des Trainings- und Testdatensatzes zu suchen. Es ist davon auszugehen, dass diese Aufteilung einen nicht zu unterschätzenden Einfluss auf die spätere Vorhersagefähigkeit des Modells ausübt. Die 80 diversesten Moleküle wurden bei den Modellen 1-25 als Testdatensatz aus dem MWT-Lagoda Datensatz (930 Moleküle) nach der MOE *diverse Subset* Funktion (siehe 3.6) ausgewählt. Ein Problem bei dieser Methode könnte darin liegen, dass die trainierten QSPR Modelle auf diese Diversität nicht hinreichend ausgelegt sind und daher die Vorhersage nur unzureichend ausfällt. Diese Hypothese könnte mit der Anwendung einer anderen Methode für die Einteilung des Datensatzes überprüft werden.
Genau dies wurde mit den Modellen 26-39 untersucht. Wie die Ergebnisse der Korrelationskoeffizienten zeigen, hängen diese von der verwendeten Datenbank-Split Methode ab und variieren stark. Zudem scheint es zum Teil einen Unterschied zu machen, wenn die Modelle (26, 30, 32, 34, 36 und 38) nur in KNIME modelliert werden. Die Modelle (27, 29, 30, 31, 33, 35, 37 und 39) die nach dem Knoten *Row to model* abgespeichert und in MOE auf ihre Vorhersagefähigkeit untersucht wurden, erzielen im direkten Vergleich bei den Modellen 29, 31, 33, 35, 37 und höhere Performancewerte.
Allgemein kann die Varianz der Vorhersagefähigkeit der Modelle 32-39 mit der zugrunde liegenden Methode des Random sampling erklärt werden. Hierbei werden die Moleküle willkürlich bei der Einteilung des Datensatz in Trainings- und Testdaten gewählt. Jeder Eintrag besitzt dabei die gleiche Wahrscheinlichkeit, ausgewählt zu werden.

Die Modelle mit der Einteilung nach Deskriptoren erzielten ähnlich hohe Performancewerte (Modell 30) wie Modell 6, das nach dem gleichen Aufbau erstellt wurde, sowie höhere Ergebnisse (Modell 31).
Bei dem Modell 26 zeigt sich die höchste Vorhersagefähigkeit von den nur in KNIME erstellten Modellen (26, 30, 32, 34, 36 und 38). Es besitzt einen R^2_{Test} Wert von 0,38. Modell 27, dessen Datenbank-Split auch nach der Fingerprint-Methode durchgeführt wurde aber in Programm MOE evaluiert wurde, besitzt die gleiche Vorhersagefähigkeit. Ein Unterschied in der Vorhersagefähigkeit tritt erst zwischen den Modellen 28 (R^2_{Test} Wert von 0,27) und 29 (R^2_{Test} Wert von 0,38) auf. Bei diesen Modellen wurden die Deskriptoren vorher in MOE berechnet. Es wirkt sich also auf die Modellierung in KNIME aus, wenn die Deskriptorwerte vorher in einem fremden Programm berechnet werden. Dies könnte in der unterschiedlichen Architektur der Programme liegen, so ist KNIME Java basiert und MOE basiert auf der SVL Programmiersprache.
Der niedrigste und höchste Performancewert wurde mit der Einteilung nach Fingerprints und normalisierten Deskriptorwerten bei Modell 38 (R^2_{Test} Wert von 0,01) und 39 (R^2_{Test} Wert von 0,44) erzielt. Der große Unterschied der zu diesem unterschiedlichen Ergebnis führt, liegt wahrscheinlich am MOE Model Predictor Knoten (siehe Abb. 12). Denn bis zu diesem Punkt sind beide Modelle in ihrem Aufbau identisch. Es ist denkbar, dass dieser Knoten mit normalisierten Werten aufgrund seiner Architektur Schwierigkeiten besitzt. Dahingegen wird bei Modell 39 (wie auch bei den Modellen 27, 29, 31, 33, 35 und 37) die Formel als *.fit Datei gespeichert. Mit ihr wird dann im Programm MOE der Testdatensatz evaluiert.

Festhalten lässt sich, dass das Modell mit Einteilung des Datensatzes nach der Fingerprint Methode und mit normalisierten Deskriptorwerten zu den höchsten Performancewerte führte. Die Einteilung nach der Random sampling Methode führt zwar auch zu hohen Performancewerten, jedoch nicht sicher im ersten Durchgang. Daher sollte diese Methode nicht weiter in der Modellierung verfolgt werden.

6 Zusammenfassung und Ausblick

In dieser Arbeit wurden insgesamt 39 unterschiedliche QSPR Modelle, die alle die Bioverfügbarkeit vorhersagen, erstellt. Sie unterscheiden sich grundlegend in den für die Modellierung verwendeten Trainingsmethoden, dem verwendeten Programm, die in Anspruch genommene Anzahl an Deskriptoren und in der Einteilung des Datensatzes (in Trainings- und Testdatensatz).

An verwendeten Trainingsmethoden wurde auf lineare (PLS, PCR) sowie auf nicht-lineare Methoden (GA, PNN und SVM) zurückgegriffen. Die folgende Tabelle 11 führt jeweils das Modell auf, welches die höchste Vorhersagefähigkeit der einzelnen Trainingsmethode besitzt.

Tabelle 11 Höchste Performancewerte nach Trainingsmethode

Modellnummer	Trainingsmethode	Deskriptoranzahl	R_{Test}	R^2_{Test}
39	PLS-Autoqsar	206	0,66	0,44
25	PCR	499	0,37	0,14
3	GA	206	0,30	0,09
16	PNN	484	0,35	0,12
11	SVM	308	0,20	0,04

Es konnte gezeigt werden, dass die Methode, nach dem der Datensatz in Trainings- und Testdatensatz eingeteilt wird, einen erheblichen Einfluss auf die Vorhersagefähigkeit des Modells besitzt und das die nicht-linearen Trainingsmethoden den linearen nicht überlegen sind.

Außerdem ist festzuhalten, dass das Modell 39, das den höchsten Performancewert besitzt, im Vergleich zu literaturbekannten QSPR Modellen über eine relativ niedrige Vorhersagefähigkeit verfügt. Dieser Fakt lässt sich auf verschiedene Gründe zurückführen. Einer liegt im verwendeten Datensatz. Dieser, wie in Abbildung 13 sichtbar, verfügt über keine Normalverteilung, sondern nimmt eher die Form einer gestauchten Parabel ein. Für die Modelle 1-25 liegt ein weiterer Grund in der Aufteilung des Datensatz in Trainings- und Testdaten. Denn wie die Modelle 26-39 zeigen, kann mit einer anderen Einteilung eine höhere Vorhersagefähigkeit erzielt werden.

Ausblickend sind folgende Punkte festzuhalten:

- Der Datensatz könnte dahingehend erweitert werden, dass weitere Bioverfügbarkeitsdaten zum Datensatz hinzugefügt werden, so dass sich eine Normalverteilung ergibt, die womöglich zu einer höheren Vorhersagefähigkeit führt.
- Es ist zu klären, warum bei den Modellen 38 und 39 grundlegend unterschiedliche Performancewerte erzielt wurden. In diesem Zusammenhang sollte der Einfluss des MOE Model Predictor Knotens genauer untersucht werden. Da er bei den Modellen 28, 30, 32, 36 und 38 zu stark unterschiedlichen Ergebnissen, im Vergleich zur Vorhersage in MOE, führt.
- Ein weiterer Ansatz wäre es, den Datensatz nach chemischen Klassen zu unterteilen. In der Medizinalchemie wird bei der Forschung nach neuen Wirkstoffkandidaten hauptsächlich nach Molekülen, die sich in der gleichen chemischen Klasse befinden, gesucht. Eine Modellierung der Bioverfügbarkeit von Molekülen, die nur einer chemischen Klasse angehörig sind, würde daher mit großer Wahrscheinlichkeit zu einer erhöhten Vorhersagegenauigkeit führen.
- Eine weitere Idee wäre die Vorhersage nach unterschiedlichen Bioverfügbarkeitsklassen in Anlehnung an das Paper von Moda [23]. Ein solcher Ansatz ist dann sinnvoll, wenn z.B. nur geklärt werden soll, ob ein Molekül eine Bioverfügbarkeit die z.B. über 30 liegt, besitzt (Werte darunter werden meist in der Wirkstoffentwicklung nicht weiterverfolgt).

7 Experimenteller Teil

7.1 Software

Zur Anfertigung dieser Praxismodularbeit wurden folgende Programme verwendet:

- Molecular Operating Environment (MOE®, Version 2015.10; Chemical Computing Gorup Inc., Montreal, Kanada) [41]
- KNIME (Version 3.4.0 2017) [42]

7.2 Hardware

Alle Berechnungen wurden auf folgendem Server durchgeführt:
Ubuntu Linux 14.04 x64 LTS Server mit installiertem X-Desktop
24 logische CPUs
160 GB RAM, 3TB lokaler Speicher (als RAID5) -> RW Performance > 300 MB/s
2x NVIDIA Quadro K2200 (2x 640 Cuda Kerne)

7.3 Vorbetrachtung Datensatz MWT-Lagoda

Für den aus der Praxismodularbeit erarbeiteten Datensatz hF-MWT-17, der im folgendem als MWT-Lagoda bezeichnet wird, wurden Mittelwerte für die Bioverfügbarkeitswerte berechnet. Da nicht für jeden Eintrag zwei oder drei Bioverfügbarkeitswerte vorhanden sind, musste dies bei der Berechnung berücksichtigt werden. Damit wird ausgeschlossen, dass nicht falsche Mittelwerte berechnet werden.
In einem weiteren Schritt wurden die Standardabweichung und die prozentuale Standardabweichung berechnet. Einträge mit einer zu großen Abweichung wurden aus dem Datensatz entfernt. Als Cut-off wurde eine mehrstufige Regel angewendet. Erst wurden alle Einträge mit einer Standardabweichung über 10 gelöscht. Im zweiten Schritt wurden alle mit einer prozentualen Standardabweichung von über 10% entfernt, es sei denn ihre Standardabweichung war niedriger als 5.

7.4 Modellierung in MOE

7.4.1 Vorbereitung Datensatz MWT-Lagoda

Für den *MWT-Lagoda* Datensatz werden alle in MOE® verfügbaren 2D Deskriptoren (206) mit Hilfe der *Descriptors* Funktion berechnet. Anschließend wird mit der Funktion *Diverse Subset* der Datensatz in einen Trainings- und Testdatensatz eingeteilt (850 Trainingsdaten, 80 Testdaten). Als Methode wurde die Berechnung der euklidischen Distanz über die Deskriptoren gewählt (siehe 3.6).

7.4.2 AutoQuaSAR

Die Modellerstellung wurde mit dem svl-Skript *AutoQuaSAR* [39] durchgeführt, das vorher über die *SVL code exchange site for the MOE user community* [46] heruntergeladen wurde. Nach dem Laden und Ausführen des Skriptes in MOE® wurden verschiedene Parameter für die Modellierung gesetzt (siehe Abb. 10).

7.4.2.1 PLS mit MOE

Für die Modellierung mit der PLS Methode wird diese unter *Method* ausgewählt. Als *Original File* dient der Trainingsdatensatz aus 7.4.1. Im *Activity Feld* wird das Feld mit den Bioverfügbarkeitsmittelwerten selektiert. Unter *Settings* werden alle 2D Deskriptoren ausgewählt. Weitere Einstellungen werden nicht vorgenommen. Das erstellte Modell wird als *.fit Datei und der Report als *.txt Dati gespeichert.
Im Testdatensatz (siehe 7.4.1) wird die Aktivität durch die *Model-Evaluate* Funktion mit der Auswahl der entsprechenden *.fit Datei vorausgesagt. Die Korrelationsanalyse wird durch die *Analysis – Correlation Plot* Funktion durchgeführt. Dafür werden die Spalten mit den realen und den vorausgesagten Bioverfügbarkeitswerten markiert. Ausgegeben wird der entsprechende Graph mit den Kenngrößen R und R^2.

7.4.2.2 PCR mit MOE

Für die Modellierung mit der PCR Methode, wird diese unter *Method* ausgewählt. Als *Original File* dient der Trainingsdatensatz aus 7.4.1. Im *Activity Feld* wird das Feld mit den Bioverfügbarkeitsmittelwerten selektiert. Unter *Settings* werden alle 2D Deskriptoren ausgewählt. Weitere Einstellungen werden nicht vorgenommen. Das erstellte Modell wird als *.fit Datei und der Report als *.txt Dati gespeichert.

Im Testdatensatz (siehe 7.4.1) wird die Aktivität durch die *Model-Evaluate* Funktion mit der Auswahl der entsprechenden *.fit Datei vorausgesagt. Die Korrelationsanalyse wird durch die *Analysis – Correlation Plot* Funktion durchgeführt. Dafür werden die Spalten mit den realen und den vorausgesagten Bioverfügbarkeitswerten markiert. Ausgegeben wird der entsprechende Graph mit den Kenngrößen R und R^2.

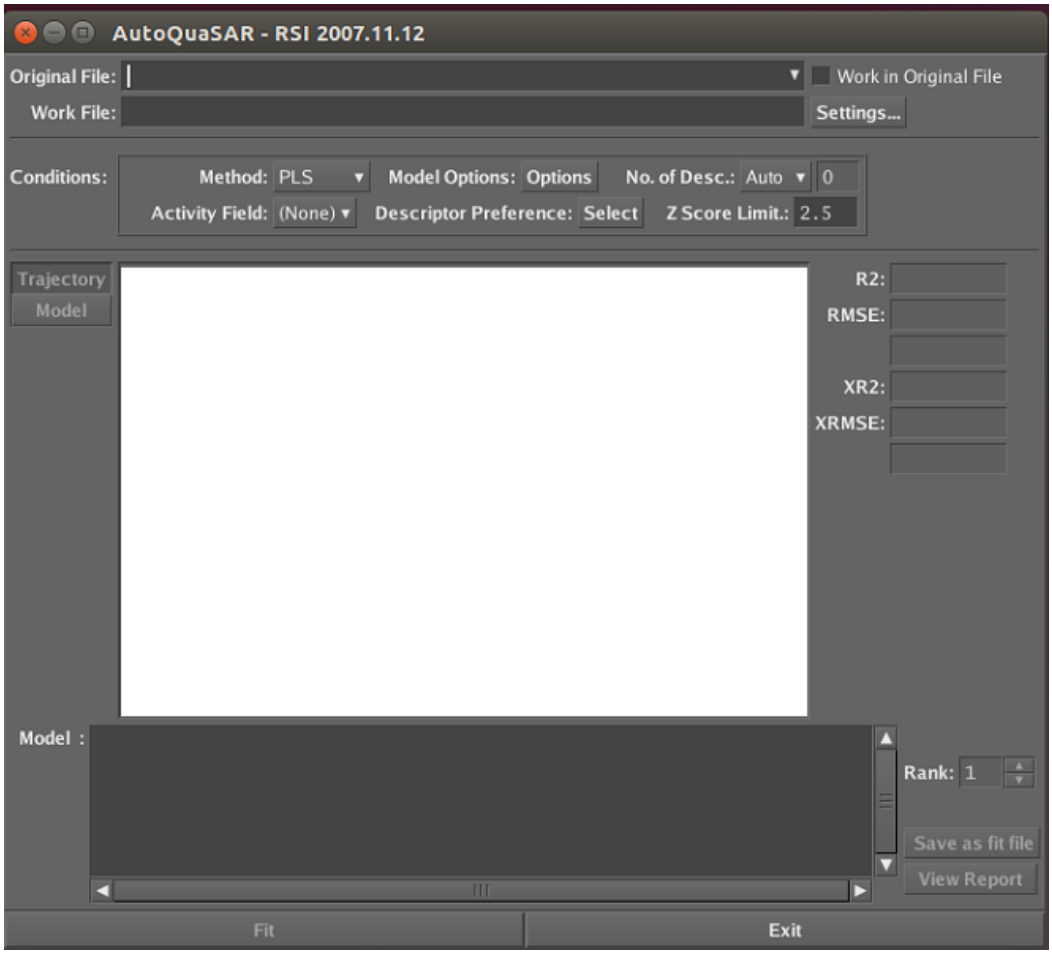

Abbildung 10 svl-Skript AutoQuSAR Unter *Original File* wird der Trainingsdatensatz ausgewählt. In der Einstellung *Setting* kann ausgewählt werden welche Deskriptoren für die Modellierung berücksichtigt werden sollen und welchen *wash options* die Moleküle vorher unterzogen werden. In der Rubrik *Conditions* wird die Trainingsmethode ausgewählt. Zur Auswahl stehen *PLS*, *PCR*, *Binary* und die *GA-MLR* Methode. Unter *Model Options* ist es möglich das *Component limit* zu setzen sowie die Gewichtung von einzelnen Deskriptoren festzulegen. Ob die Anzahl an Deskriptoren fix oder automatisch bestimmt werden soll kann unter *No. of Desc.* festgelegt werden. Im *Activity Field* wird der Wert (in diesem Fall die Bioverfügbarkeit) festgelegt der durch die Deskriptoren vorhergesagt werden soll. Wenn einzelne Deskriptoren unbedingt einbezogen oder herausgelassen werden sollen, kann dies unter *Descriptor Preference* eingestellt werden.

7.4.2.3 GA-MLR mit MOE

Für die Modellierung mit dem genetischen Algorithmus wird zunächst eine Erweiterung für das svl-Skript AutoQuSAR geladen, das svl-Skript QSAR Evolution [43].

Anschließend wird wie unter 7.4.2.1 beschrieben vorgegangen mit dem Unterschied, dass als *Methode* GA-MLR anstatt PLS ausgewählt wird.

Im Testdatensatz (siehe 7.4.1) wird die Aktivität durch die *Model-Evaluate* Funktion mit der Auswahl der entsprechenden *.fit Datei vorausgesagt. Die Korrelationsanalyse wird durch die *Analysis – Correlation Plot* Funktion durchgeführt. Dafür werden die Spalten mit den realen und den vorausgesagten Bioverfügbarkeitswerten markiert. Ausgegeben wird der entsprechende Graph mit den Kenngrößen R und R^2.

7.5 Modellierung in KNIME

Der Allgemeine Aufbau der Modellierung in KNIME lässt sich in vier Teilschritte einteilen (siehe Abb. 11). Im ersten Schritt wird der Datensatz eingelesen, die Moleküle werden gewaschen und nicht relevante Werte werden für die spätere Modellierung herausgefiltert. Dies geschieht für den Trainings- sowie für den Testdatensatz unabhängig voneinander. Anschließend werden im zweiten Schritt, nach vorheriger Auswahl, die molekularen Deskriptoren berechnet. Im dritten Schritt wird mit Hilfe einer Trainingsmethode die Bioverfügbarkeit vorhergesagt. Der Trainingsdatensatz dient zum Lernen, d.h. dem Erstellen einer Auswahl von Deskriptoren und deren Gewichtung, die die Bioverfügbarkeit am besten vorhersagen. Die Vorhersage wird dann auf den Testdatensatz angewendet. Im letzten Schritt wird das Ergebnis in eine Datenbank geschrieben.

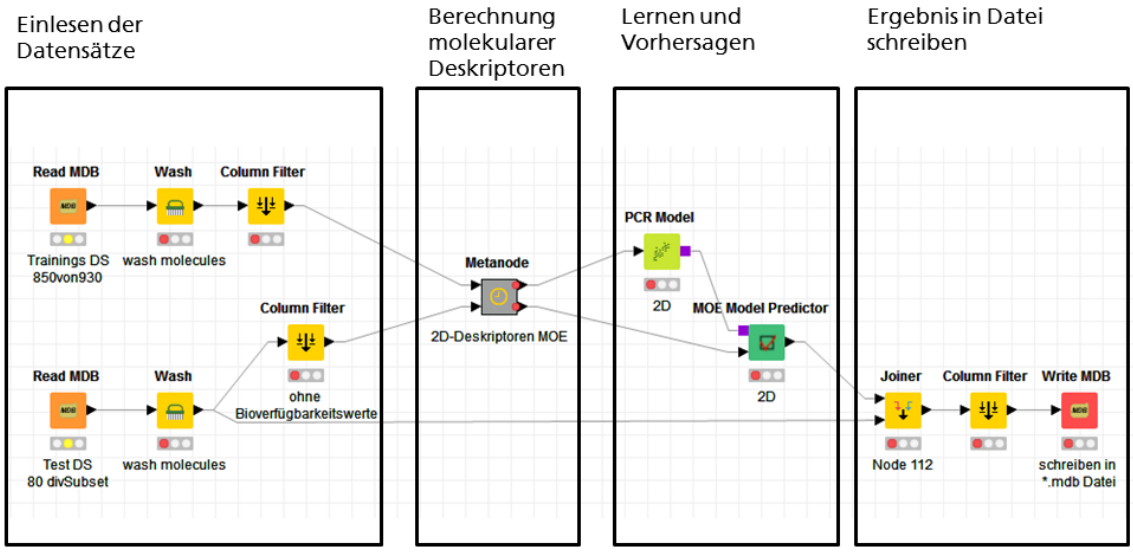

Abbildung 11 Schema für die Modellierung in KNIME. In diesem Beispiel dient die *PNN* Methode als Trainingsmethode.

7.5.1 Modelle mit 2D Deskriptoren

7.5.1.1 PNN in KNIME

Nach dem Einlesen *(Read MDB)* des unter 7.4.1 generierten Trainings- und Testdatensatzes werden im Wasch-Schritt *(Wash)* eventuell vorhandene Gegenionen gelöscht, Moleküle mit Protonen versehen und Säuren und Basen deprotoniert bzw. protoniert. Nach dem Filter Schritt, bei dem nur die Felder mol_wasched, Molecule_name und F_Mittelwert behalten werden, wird in einem *Metanode* eine Auswahl von Deskriptoren getroffen. Es werden alle MOE bekannten 2D Deskriptoren ausgewählt. Als Kraftfeld ist das MMFF94* eingestellt. Es beruht auf dem von Halgren entwickelten MMFF94 Kraftfeld [47]. Die für den Trainingsdatensatz berechneten Werte werden dem *PNN Learner* übergeben. Das berechnete Ergebnis wird zusammen mit den berechneten Deskriptorwerten des Testdatensatzes an den *PNN Predictor* gegeben. Die neu vorhergesagten Bioverfügbarkeitswerte des Testdatensatzes werden im *Joiner* mit den Molekülnamen kombiniert. Dieses Ergebnis wird anschließend gefiltert in eine *.mdb Datei geschrieben. Die Korrelationsanalyse wird anschließend im Programm MOE, wie unter 7.4.2.1 beschrieben, durchgeführt.

7.5.1.2 SVM in KNIME

Es wird wie unter 7.5.1.1 beschrieben vorgegangen. Als einziger Unterschied wird der *SVM Learner* und der *SVM Predictor* für die Modellierung verwendet.

7.5.1.3 PLS in KNIME

Es wird wie unter 7.5.1.1 beschrieben vorgegangen mit dem Unterschied, dass der *PLS Learner* und der *MOE Model Predictor* für die Modellierung verwendet werden.

7.5.1.4 PCR in KNIME

Es wird wie unter 7.5.1.1 beschrieben vorgegangen mit dem Unterschied, dass der *PCR Learner* und der *MOE Model Predictor* für die Modellierung verwendet werden.

7.5.1.5 Modelle mit MOE Knoten AutoQSAR in KNIME

Ähnlich wie unter 7.5.1.1 beschrieben, vollzieht sich der Aufbau der Modellierung mit dem AutoQSAR MOE Knoten. Es werden mehrere Modelle mit unterschiedlichen Deskriptorauswahlen durchgeführt. Neben allen 2D Deskriptoren werden Modelle mit drei weiteren Deskriptorauswahlen (siehe 7.5.2) durchgeführt. Als Trainingsmethode dienen die *PLS*, sowie die *PCR* Methode. Für die Erstellung des Modells mit dem *Autoqsar* Knoten, wird als *Activity Feld* das Feld der Bioverfügbarkeitswerte ausgewählt. Das Modell mit dem höchsten R^2 Wert wird über einen *Row Filter* selektiert und dem *Row to Model* Knoten übergeben. Das Ergebnis wird als *.fit Datei gespeichert und zusätzlich wird das Modell, zusammen mit dem Testdatensatz, dem *MOE Model Predictor* Knoten zur Vorhersage gegeben. Das Ergebnis des *Model Predictors* wird in eine *.mdb Datei geschrieben und eine Korrelationsanalyse wird anschließend im Programm MOE, wie unter 7.4.2.1 beschrieben, durchgeführt. Zusätzlich wird das Modell mit der *.fit Datei evaluiert. Dafür wird in MOE unter *Compute – Model – Evaluate* die *.fit Datei ausgewählt. Anschließend wird eine Korrelationsanalyse durchgeführt. Dies wird zur Überprüfung, ob mit beiden Modellen das gleiche Ergebnis erzielt wird, durchgeführt.

7.5.2 Modelle mit mehr als nur 2D Deskriptoren

Die Modellierung wird im Programm KNIME durchgeführt. Der allgemeine Aufbau ist wie unter 7.5.1.1 beschrieben mit dem Unterschied, dass im Meatanode mehr als nur 2D Deskriptoren aus MOE ausgewählt werden. Insgesamt werden bis zu 499 Deskriptoren verwendet. Die Liste von Deskriptoren ist im Anhang auf S.57 ff einsehbar. Neben den Deskriptoren aus MOE, zu denen auch die 15 quantenchemischen MOPAC Deskriptoren gehören, deren Berechnungszeit 12 Stunden beträgt, werden folgende verwendet: *CDK* (*Fingerprints, Molecular Properties, XLogP)* [45]*, RDKit (Deskriptoren, Fingerprint, Calculate Charges)* [44].

Zu folgenden 3 Deskriptorauswahlen:

1. 2-3D MOE Deskriptoren;
2. 2-3D MOE Deskriptoren + CDK, RDKit Deskriptoren;
3. 2-3D MOE Deskriptoren + CDK, RDKit Deskriptoren + quantenchemische Deskriptoren;

werden jeweils mit den vier Trainingsmethoden *PNN* (siehe 7.5.1.1), *SVM* (siehe 7.5.1.2), *PLS* (siehe 7.5.1.3) und *PCR* (siehe 7.5.1.4) weitere Modelle erstellt.
Zusätzlich werden mit den Deskriptorauswahlen 2 und 3 wie in 7.5.1.5 beschrieben weitere Modelle erstellt.

7.5.3 Modelle mit anderer Datensatz Einteilung

7.5.3.1 Datensatzsplit nach Fingerprint Methode

Datengrundlage ist der *MWT-Lagoda* Datensatz. Bei den Modellen 26, 27, 38 und 39 werden die 2D Deskriptoren im Programm KNIME berechnet und bei den Modellen 28 und 29 werden die 2D Deskriptoren im Programm MOE® berechnet. Anschließend wird der Datensatz nach der Fingerprint Methode (siehe 3.6) in einen Trainingsdatensatz (90% = 837 Einträge) und einen Testdatensatz (10% = 93 Einträge) eingeteilt. Beide neuen Datensätze werden anschließend als *.mdb gespeichert. Mit dem *Autoqsar* Knoten wird das Modell erstellt. Als Trainingsmethode dient die *PLS* Methode, als *Activity Feld* wird das Feld der Bioverfügbarkeitswerte ausgewählt und als verwendete Deskriptoren dienen die vorher berechneten 2D Deskriptoren. Das Modell mit dem höchsten R2 Wert wird dem *Row to Model* Knoten übergeben. Es wird als *.fit Datei gespeichert und das Modell wird auf den Testdatensatz, mit dem *MOE Model Predictor* Knoten, angewendet. Das Ergebnis des *Model Predictors* wird in einer *.mdb geschrieben (siehe Abb. 12) und eine Korrelationsanalyse wird anschließend im Programm MOE, wie unter 7.4.2.1 beschrieben, durchgeführt.
Zusätzlich wird das Modell mit der *.fit Datei evaluiert. Dafür wird in MOE unter *Compute – Model – Evaluate* die *.fit Datei ausgewählt. Anschließend wird eine Korrelationsanalyse durchgeführt. Dieser Schritt wird als Validierung durchgeführt um zu Überprüfen, dass bei beiden Durchführungen das gleiche Ergebnis berechnet wird.

7.5.3.2 Datensatzsplit nach Deskriptor Methode

Es wird wie unter 7.5.3.1 beschrieben vorgegangen mit dem einzigen Unterschied, dass der Datensatz nach der Deskriptor Methode eingeteilt wird (siehe 3.6).

7.5.3.3 Datensatzsplit nach Random sample Methode

Es wird wie unter 7.5.3.1 beschrieben vorgegangen mit dem einzigen Unterschied, dass der Datensatz nach der *Random Sampling* Methode mit dem *X-Partioner* Knoten eingeteilt wird. Dies wurde drei Mal wiederholt.

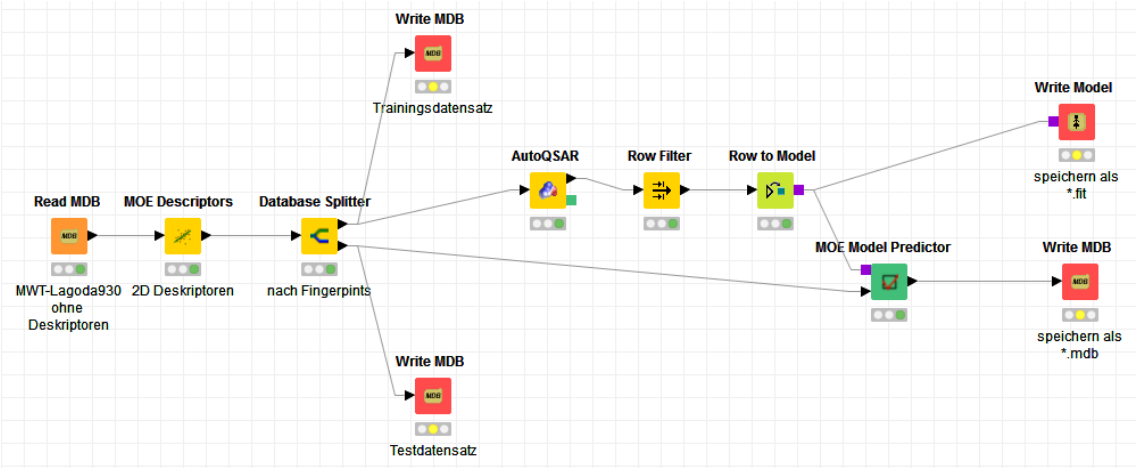

Abbildung 12 Schema für die Modellierung nach der Versuchsvorschrift von 7.5.3.1 In diesem Beispiel dient die *PLS* Methode als Trainingsmethode.

7.5.3.4 Datensatzsplit mit Normalisierung nach Fingerprint Methode

Es wird wie unter 7.5.3.1 beschrieben vorgegangen mit dem einzigen Unterschied, dass bevor der Datensatz geteilt wird, eine Normalisierung von allen Dateneinträgen stattfindet.

8 Literaturverzeichnis

(1) Snodin, D. J. An EU perspective on the use of in vitro methods in regulatory pharmaceutical toxicology. *Toxicology Letters* **2002**, *127*, 161–168.

(2) Kraljevic, S.; Stambrook, P. J.; Pavelic, K. Accelerating drug discovery. *EMBO reports* **2004**, *5*, 837–842.

(3) Adams, C. P.; van Brantner, V. Estimating the cost of new drug development: is it really 802 million dollars? *Health affairs (Project Hope)* **2006**, *25*, 420–428.

(4) van de Waterbeemd, H.; Gifford, E. ADMET in silico modelling: towards prediction paradise? *Nature reviews. Drug discovery* **2003**, *2*, 192–204.

(5) Kola, I.; Landis, J. Can the pharmaceutical industry reduce attrition rates? *Nature Reviews Drug Discovery* **2004**, *3*, 711–717.

(6) Dehmer, M.; Varmuza, K.; Bonchev, D.; Emmert-Streib, F. *Statistical Modelling of Molecular Descriptors in QSAR/QSPR*, 2. Aufl.; Quantitative and Network Biology (VCH); Wiley-Blackwell: s.l., 2012.

(7) Lagoda, F. Analyse von Literaturdaten von pharmakokinetischen Daten zur späteren Modellgenerierung, Hochschule Mittweida, Mittweida, 2017.

(8) Hänsel, R.; Hölzl, J. *Lehrbuch der pharmazeutischen Biologie*; Springer Berlin Heidelberg: Berlin, Heidelberg, 1996.

(9) Knollman, B.; Chabner, B.; Brunton, L. *Goodman & Gilman's the pharmacological basis of therapeutics*, 12th ed.; New York: McGraw-Hill Medical, 2011.

(10) Sakai, J. B. *Practical pharmacology for the pharmacy technician*; LWW pharmacy technician education series; Lippincott Williams & Wilkins: Philadelphia, 2009.

(11) Leach, A. R.; Gillet, V. J. *An introduction to chemoinformatics*, Rev. ed.; Springer: Dordrecht, 2007.

(12) Bunin, B. A. *Chemoinformatics: Theory, practice, & products*; Springer: Dordrecht, 2007.

(13) Labute, P. A widely applicable set of descriptors. *Journal of Molecular Graphics and Modelling* **2000**, *18*, 464–477.

(14) *Handbook of the Molecular Operating Enviroment [2015.10];* Chemical Computing Group Inc., Ed.; Montreal, 2015.

(15) Ertl, P.; Rohde, B.; Selzer, P. Fast Calculation of Molecular Polar Surface Area as a Sum of Fragment-Based Contributions and Its Application to the Prediction of Drug Transport Properties. *J. Med. Chem.* **2000**, *43*, 3714–3717.

(16) Oprea, T. I. Property distribution of drug-related chemical databases*. *Journal of Computer-Aided Molecular Design* **2000**, *14*, 251–264.

(17) Willett, P.; Winterman, V.; Bawden, D. Implementation of nearest-neighbor searching in an online chemical structure search system. *J. Chem. Inf. Model.* **1986**, *26*, 36–41.

(18) Pearlman, R. S.; Smith, K. M. Novel software tools for chemical diversity. *Perspectives in Drug Discovery and Design* **1998**, *9*, 339–353.

(19) Brown, A. C.; Fraser, T. R. On the Connection between Chemical Constitution and Physiological Action; with special reference to the Physiological Action of the Salts of the Ammonium Bases derived from Strychnia, Brucia, Thebaia, Codeia, Morphia, and Nicotia. *Journal of anatomy and physiology* **1868**, *2*, 224–242.

(20) Hansch, C.; Maloney, P. P.; Fujita, T.; Muir, R. M. Correlation of Biological Activity of Phenoxyacetic Acids with Hammett Substituent Constants and Partition Coefficients. *Nature* **1962**, *194*, 178–180.

(21) Hansch, C.; Muir, R. M.; Fujita, T.; Maloney, P. P.; Geiger, F.; Streich, M. The Correlation of Biological Activity of Plant Growth Regulators and Chloromycetin Derivatives with Hammett Constants and Partition Coefficients. *J. Am. Chem. Soc.* **1963**, *85*, 2817–2824.

(22) Turner, J. V.; Glass, B. D.; Agatonovic-Kustrin, S. Prediction of drug bioavailability based on molecular structure. *Analytica Chimica Acta* **2003**, *485*, 89–102.

(23) Moda, T. L.; Montanari, C. A.; Andricopulo, A. D. Hologram QSAR model for the prediction of human oral bioavailability. *Bioorganic & medicinal chemistry* **2007**, *15*, 7738–7745.

(24) Tian, S.; Li, Y.; Wang, J.; Zhang, J.; Hou, T. ADME evaluation in drug discovery. 9. Prediction of oral bioavailability in humans based on molecular properties and structural fingerprints. *Molecular pharmaceutics* **2011**, *8*, 841–851.

(25) Dugas, M.; Schmidt, K. *Medizinische Informatik und Bioinformatik: Ein Kompendium für Studium und Praxis*; Springer-Verlag, 2013.

(26) Handels, H. *Medizinische Bildverarbeitung: Bildanalyse, Mustererkennung und Visualisierung für die computergestützte ärztliche Diagnostik und Therapie*, 2., überarbeitete und erweiterte Auflage; Vieweg+Teubner Verlag / GWV Fachverlage GmbH Wiesbaden: Wiesbaden, 2009.

(27) Helland, I. S. Some theoretical aspects of partial least squares regression. *Chemometrics and Intelligent Laboratory Systems* **2001**, *58*, 97–107.

(28) Roy, K. *Advances in QSAR Modeling: Applications in Pharmaceutical, Chemical, Food, Agricultural and Environmental Sciences*; Challenges and Advances in Computational Chemistry and Physics; Springer, 2017.

(29) Kurgan, L.; Razib, A. A.; Aghakhani, S.; Dick, S.; Mizianty, M.; Jahandideh, S. CRYSTALP2: sequence-based protein crystallization propensity prediction. *BMC structural biology* **2009**, *9*, 50.

(30) URL-1. Abbildung lineare support vector machine. https://upload.wikimedia.org/wikipedia/commons/2/2a/Svm_max_sep_hyperplane_with_margin.png (accessed August 24, 2017).

(31) URL-2. Abbildung zu linear und nicht linear trennbareren Scatter Plot. https://upload.wikimedia.org/wikipedia/de/a/a0/Diskriminanzfunktion.png (accessed August 25, 2017).

(32) URL-3. Abbildung zur Anwendung des Kernel Tricks. https://upload.wikimedia.org/wikipedia/commons/thumb/c/cc/Kernel_trick_idea.svg/2000px-Kernel_trick_idea.svg.png (accessed August 22, 2017).

(33) Vapnik, V. N. *Statistical learning theory*; A Wiley-Interscience publication; Wiley: New York, 1998.

(34) Fischer, J. Support Vector Machines (SVM). http://www.mathematik.uni-ulm.de/stochastik/lehre/ss07/seminar_sl/fischer.pdf (accessed August 15, 2017).

(35) Wetson, J. Support Vector Machine: Tutorial. http://www.cs.columbia.edu/~kathy/cs4701/documents/jason_svm_tutorial.pdf (accessed August 15, 2017).

(36) Zeinali, Y.; Story, B. A. Competitive probabilistic neural network. *ICA* **2017**, *24*, 105–118.

(37) Specht, D. F. Probabilistic neural networks. *Neural networks* **1990**, *3*, 109–118.

(38) Cheung, V.; Cannons, K. An Introduction to Probalistic Neural Networks. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.182.4592&rep=rep1&type=pdf (accessed August 16, 2017).

(39) Goto, J. *AutoQuaSAR*; Ryoka Systems Inc, 2008.

(40) Buchholz, M. Inhibitoren der Glutaminyl Cyclase: Synthese, Charakterisierung und in-silico Untersuchungen. Dissertation, Martin-Luther-Universität Halle-Wittenberg, 2007.

(41) Chemical Computing Group Inc. *MOE®*; Chemical Computing Group Inc.: 1010 Sherbooke St. West, Suite #910, Montreal, QC, Canada, H3A 2R7, 2015.

(42) Berthold, M. R.; Cebron, N.; Dill, F.; Gabriel, T. R.; Kötter, T.; Meinl, T.; Ohl, P.; Thiel, K.; Wiswedel, B. KNIME - the Konstanz information miner. *SIGKDD Explor. Newsl.* **2009**, *11*, 26.

(43) Goto, J. *QuaSAR-Evolution*; Ryoka Systems Inc, 2010.

(44) Landrum, G. *RDKit: Open-Source Cheminformatics Software*, 2016.

(45) Willighagen, E. L.; Mayfield, J. W.; Alvarsson, J.; Berg, A.; Carlsson, L.; Jeliazkova, N.; Kuhn, S.; Pluskal, T.; Rojas-Chertó, M.; Spjuth, O. *et al.* The Chemistry Development Kit (CDK) v2.0: Atom typing, depiction, molecular formulas, and substructure searching. *J Cheminform* **2017**, *9*, 37.

(46) URL-4. SVL code exchange site for the MOE user community. https:// svl.chemcomp.com/ (accessed September 2, 2017).

(47) Halgren, T. A. Merck molecular force field. I. Basis, form, scope, parameterization, and performance of MMFF94. *Journal of Computational Chemistry* **1996**, *17*, 490–519.

9 Anhang

Tabelle 12 Übersicht über die Leistungsfähigkeit aller erstellten Modelle.

Modellnummer	Trainingsmethode	Deskriptoranzahl	R	R^2
1	PLS	206	0,32	0,10
2	PCR	206	0,32	0,10
3	GA	206	0,30	0,09
4	PNN	204	0,27	0,08
5	SVM	204	0,00	0,00
6	PLS	204	0,33	0,11
7	PCR	204	0,30	0,09
8	PLS-Autoqsar	204	0,37	0,00
9	PCR-Autoqsar	204	0,26	0,19
10	PNN	321	0,32	0,10
11	SVM	308	0,20	0,04
12	PLS	321	0,33	0,11
13	PCR	321	0,30	0,09
14	PLS-Autoqsar	321	0,02	0,02
15	PCR-Autoqsar	321	0,02	0,02
16	PNN	484	0,35	0,12
17	SVM	470	0,17	0,03
18	PLS	484	0,14	0,02
19	PCR	484	0,14	0,02
20	PLS-Autoqsar	484	0,03	0,02
21	PCR-Autoqsar	484	0,03	0,02
22	PNN	499	0,06	0,00
23	SVM			
24	PLS	499	0,37	0,14
25	PCR	499	0,37	0,14
26	PLS-Autoqsar	Fingerprint	0,62	0,38
27	PLS-Autoqsar	Fingerprint	0,62	0,38
28	PLS-Autoqsar	Fingerprint	0,52	0,27
29	PLS-Autoqsar	Fingerprint	0,62	0,38
30	PLS-Autoqsar	Deskriptoren	0,36	0,13
31	PLS-Autoqsar	Deskriptoren	0,53	0,28
32	PLS-Autoqsar	Random Sample1	0,28	0,08
33	PLS-Autoqsar	Random Sample1	0,53	0,28
34	PLS-Autoqsar	Random Sample2	0,38	0,14
35	PLS-Autoqsar	Random Sample2	0,41	0,17
36	PLS-Autoqsar	Random Sample3	0,43	0,19
37	PLS-Autoqsar	Random Sample3	0,61	0,37
38	PLS-Autoqsar	Fingerprint normalisiert	0,12	0,01
39	PLS-Autoqsar	Fingerprint normalisiert	0,66	0,44

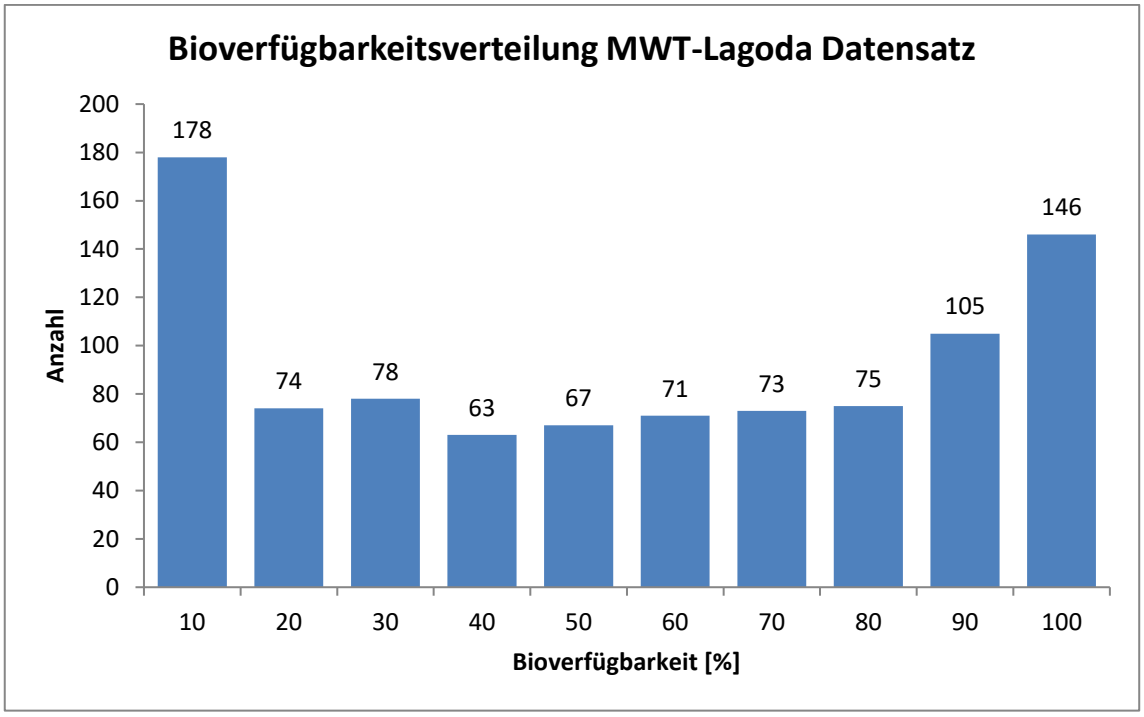

Abbildung 13 Bioverfügbarkeitsverteilung des MWT-Lagoda Datensatzes mit Mittelwerten. Die Bioverfügbarkeits Mittelwerte wurden nach der in 7.3 beschriebenen Vorschrift ermittelt.

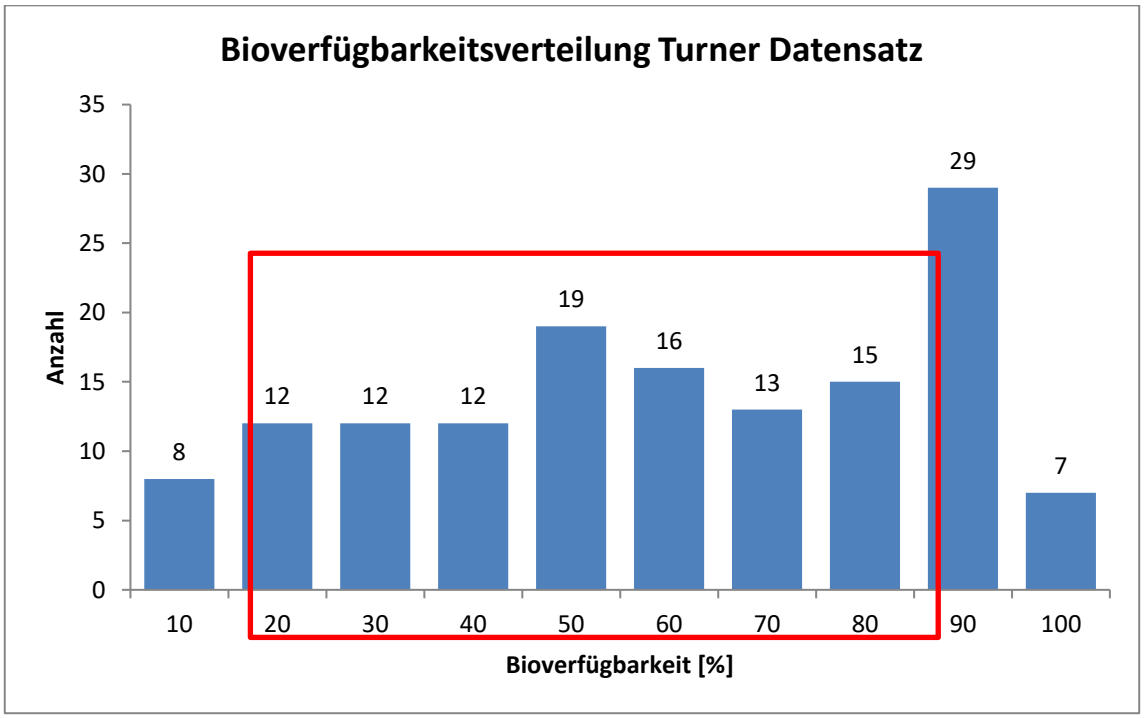

Abbildung 14 Bioverfügbarkeitsverteilung des Turner Datensatzes. Im rot umrandeten Bereich befinden sich die 10 Moleküle die vom Turner QSPR Modell vorhergesagt wurden. Verändert nach [7]

Liste der 499 Deskriptoren die in 7.5.2 verwendet wurden:

Standard fingerprints for mol_washed, ALogP, Mannhold LogP, Moreau-Broto Autocorrelation (mass) descriptors, Atomic Polarizabilities, Aromatic Atoms Count,Aromatic Bonds Count, Element Count, BCUT, Bond Polarizabilities, Bond Count, Moreau-Broto Autocorrelation (charge) descriptors, Moreau-Broto Autocorrelation (polarizability) descriptors, Charged Partial Surface Areas (3D), Eccentric Connectivity Index, Fragment Complexity, VABC Volume Descriptor,
Hydrogen Bond Acceptors, Hydrogen Bond Donors, Largest Chain, Largest Pi Chain, Petitjean Number, Rotatable Bonds Count, Lipinski's Rule of Five, Topological Polar Surface Area, Vertex adjacency information magnitude, Molecular Weight, XLogP, Zagreb Index, Molecular Formula, Formal Charge, Formal Charge (pos), Formal Charge (neg), Heavy Atoms Count, Molar Mass, SP3 Character, Rotatable Bonds Count (non terminal), XLogP (#1), SlogP, SMR, LabuteASA, TPSA, AMW, ExactMW, NumLipinskiHBA, NumLipinskiHBD, NumRotatableBonds, NumHBD, NumHBA, NumAmideBonds, NumHeteroAtoms, NumHeavyAtoms, NumAtoms, NumRings, NumAromaticRings, NumSaturatedRings, NumAliphaticRings, NumAromaticHeterocycles, NumSaturatedHeterocycles, NumAliphaticHeterocycles, NumAromaticCarbocycles, NumSaturatedCarbocycles, NumAliphaticCarbocycles, FractionCSP3, Chi0v, Chi1v, Chi2v, Chi3v, Chi4v, Chi1n, Chi2n, Chi3n, Chi4n, HallKierAlpha, kappa1, kappa2, kappa3, slogp_VSA1, slogp_VSA2, slogp_VSA3, slogp_VSA4, slogp_VSA5, slogp_VSA6, slogp_VSA7, slogp_VSA8, slogp_VSA9, slogp_VSA10, slogp_VSA11, slogp_VSA12, smr_VSA1, smr_VSA2, smr_VSA3,
smr_VSA4, smr_VSA5, smr_VSA6, smr_VSA7, smr_VSA8, smr_VSA9, smr_VSA10, peoe_VSA1, peoe_VSA2, peoe_VSA3, peoe_VSA4, peoe_VSA5, peoe_VSA6, peoe_VSA7, peoe_VSA8, peoe_VSA9, peoe_VSA10, peoe_VSA11, peoe_VSA12, peoe_VSA13, peoe_VSA14, MQN1, MQN2, MQN3, MQN4, MQN5, MQN6, MQN7, MQN8, MQN9, MQN10, MQN11, MQN12, MQN13, MQN14, MQN15, MQN16, MQN17, MQN18, MQN19, MQN20, MQN21, MQN22, MQN23, MQN24, MQN25, MQN26, MQN27, MQN28, MQN29, MQN30, MQN31, MQN32, MQN33, MQN34, MQN35, MQN36, MQN37, MQN38, MQN39, MQN40, MQN41, MQN42, mol_washed (Fingerprint), mol_washed (Charges), a_acc, a_acid, a_aro, a_base, a_count, a_don, a_donacc, a_heavy, a_hyd, a_IC, a_ICM, a_nB, a_nBr, a_nC, a_nCl, a_nF, a_nH, a_nI,

a_nN, a_nO, a_nP, a_nS, b_1rotN, b_1rotR, b_ar, b_count, b_double, b_heavy, b_max1len, b_rotN, b_rotR, b_single, b_triple, lip_acc, lip_don, lip_druglike, lip_violation, chi0, chi0v, chi0v_C, chi0_C, chi1, chi1v, chi1v_C, chi1_C, Kier1, Kier2, Kier3, KierA1, KierA2, KierA3, KierFlex, BCUT_PEOE_0, BCUT_PEOE_1, BCUT_PEOE_2, BCUT_PEOE_3, BCUT_SLOGP_0, BCUT_SLOGP_1, BCUT_SLOGP_2, BCUT_SLOGP_3, BCUT_SMR_0, BCUT_SMR_1, BCUT_SMR_2, BCUT_SMR_3, GCUT_PEOE_0, GCUT_PEOE_1, GCUT_PEOE_2, GCUT_PEOE_3, GCUT_SLOGP_0, GCUT_SLOGP_1, GCUT_SLOGP_2, GCUT_SLOGP_3, GCUT_SMR_0, GCUT_SMR_1, GCUT_SMR_2, GCUT_SMR_3, PEOE_VSA+0, PEOE_VSA+1, PEOE_VSA+2, PEOE_VSA+3, PEOE_VSA+4, PEOE_VSA+5, PEOE_VSA+6, PEOE_VSA-0, PEOE_VSA-1, PEOE_VSA-2, PEOE_VSA-3, PEOE_VSA-4, PEOE_VSA-5, PEOE_VSA-6, SlogP_VSA0, SlogP_VSA1, SlogP_VSA2, SlogP_VSA3, SlogP_VSA4, SlogP_VSA5, SlogP_VSA6, SlogP_VSA7, SlogP_VSA8, SlogP_VSA9, SMR_VSA0, SMR_VSA1, SMR_VSA2, SMR_VSA3, SMR_VSA4, SMR_VSA5, SMR_VSA6, SMR_VSA7, PEOE_PC+, PEOE_PC-, PEOE_RPC+, PEOE_RPC-, PEOE_VSA_FHYD, PEOE_VSA_FNEG, PEOE_VSA_FPNEG, PEOE_VSA_FPOL, PEOE_VSA_FPOS, PEOE_VSA_FPPOS, PEOE_VSA_HYD, PEOE_VSA_NEG, PEOE_VSA_PNEG, PEOE_VSA_POL, PEOE_VSA_POS, PEOE_VSA_PPOS, Q_PC+, Q_PC-, Q_RPC+, Q_RPC-, Q_VSA_FHYD, Q_VSA_FNEG, Q_VSA_FPNEG, Q_VSA_FPOL, Q_VSA_FPOS, Q_VSA_FPPOS, Q_VSA_HYD, Q_VSA_NEG, Q_VSA_PNEG, Q_VSA_POL, Q_VSA_POS, Q_VSA_PPOS, SlogP(0), SMR(0), vsa_acc, vsa_acid, vsa_base, vsa_don, vsa_hyd, vsa_other, vsa_pol, apol, ast_fraglike, ast_fraglike_ext, ast_violation, ast_violation_ext, balabanJ, bpol, chiral, chiral_u, density, diameter, FCharge, h_ema, h_emd, h_emd_C, h_logD, h_logP, h_logS, h_log_dbo, h_log_pbo, h_mr, h_pavgQ, h_pKa, h_pKb, h_pstates, h_pstrain, logP(o/w), logS, mr, mutagenic, nmol, opr_brigid, opr_leadlike, opr_nring, opr_nrot, opr_violation, PC+, PC-, petitjean, petitjeanSC, radius, reactive, rings, RPC+, RPC-, TPSA(0), VAdjEq, VAdjMa, VDistEq, VDistMa, vdw_area, vdw_vol, weinerPath, weinerPol, zagreb, E (MMFF94x/R-Field), E_ang (MMFF94x/R-Field), E_ele (MMFF94x/R-Field), E_nb (MMFF94x/R-Field), E_oop (MMFF94x/R-Field), E_sol (MMFF94x/R-Field), E_stb (MMFF94x/R-Field), E_str (MMFF94x/R-Field), E_strain (MMFF94x/R-Field), E_tor (MMFF94x/R-Field), E_vdw

(MMFF94x/R-Field), AM1_dipole, AM1_E, AM1_Eele, AM1_HF, AM1_HOMO, AM1_IP, AM1_LUMO, PM3_dipole, PM3_E, PM3_Eele, PM3_HF, PM3_HOMO, PM3_IP, PM3_LUMO, MNDO_dipole, MNDO_E, MNDO_Eele, MNDO_HF, MNDO_HOMO, MNDO_IP, MNDO_LUMO, vsurf_A, vsurf_CP, vsurf_CW1, vsurf_CW2, vsurf_CW3, vsurf_CW4, vsurf_CW5, vsurf_CW6, vsurf_CW7, vsurf_CW8, vsurf_D1, vsurf_D2, vsurf_D3, vsurf_D4, vsurf_D5, vsurf_D6, vsurf_D7, vsurf_D8, vsurf_DD12, vsurf_DD13, vsurf_DD23, vsurf_DW12, vsurf_DW13, vsurf_DW23, vsurf_EDmin1, vsurf_EDmin2, vsurf_EDmin3, vsurf_EWmin1, vsurf_EWmin2, vsurf_EWmin3, vsurf_G, vsurf_HB1, vsurf_HB2, vsurf_HB3, vsurf_HB4, vsurf_HB5, vsurf_HB6, vsurf_HB7, vsurf_HB8, vsurf_HL1, vsurf_HL2, vsurf_ID1, vsurf_ID2, vsurf_ID3, vsurf_ID4, vsurf_ID5, vsurf_ID6, vsurf_ID7, vsurf_ID8, vsurf_IW1, vsurf_IW2, vsurf_IW3, vsurf_IW4, vsurf_IW5, vsurf_IW6, vsurf_IW7, vsurf_IW8, vsurf_R, vsurf_S, vsurf_V, vsurf_W1, vsurf_W2, vsurf_W3, vsurf_W4, vsurf_W5, vsurf_W6, vsurf_W7, vsurf_W8, vsurf_Wp1, vsurf_Wp2, vsurf_Wp3, vsurf_Wp4, vsurf_Wp5, vsurf_Wp6, vsurf_Wp7, vsurf_Wp8, ASA, ASA+, ASA-, ASA_H, ASA_P, CASA+, CASA-, DASA, DCASA, dens, dipole, FASA+, FASA-, FASA_H, FASA_P, FCASA+, FCASA-, glob, npr1, npr2, pmi, pmi1, pmi2, pmi3, rgyr, std_dim1, std_dim2, std_dim3,vol, VSA

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Halle (Saale), den 11.10.2017

Florian Lagoda