Dissertation zur Erlangung des Doktorgrades
der Fakultät für Chemie und Pharmazie
der Ludwig-Maximilians-Universität München

BayTree: Ein Werkzeug zur gerüstbasierten Visualisierung und Aktivitätsanalyse von Screeningergebnissen chemischer Strukturdatenbanken

von
Dipl.-Chem. Stefan Seidler
aus
München

2004

Erklärung

Diese Dissertation wurde gemäß § 13 Abs. 3 bzw. 4 der Promotionsordnung der Ludwig-Maximilians-Universität München vom 29.01.1998 von Prof. Dr. Hubert Ebert betreut.

Ehrenwörtliche Versicherung

Diese Dissertation wurde selbstständig und ohne unerlaubte Hilfe erarbeitet.

München, den 20. Januar 2004

Dissertation eingereicht am 22. Januar 2004

1. Gutachter: Prof. Dr. Hubert Ebert
2. Gutachter: Prof. Dr. Christian Böhm

Mündliche Prüfung am 19. Mai 2004

Für Margarethe, Rebecca und Aurelia

Danksagung

Ich möchte allen danken, die mich während der vergangenen Jahre bei der Erstellung der Doktorarbeit unterstützt haben, auch wenn sie nicht namentlich erwähnt sind.

Besonderen Dank verdienen:

Dr. Axel Jensen, mein Betreuer in der Computational Chemistry Gruppe der Pharmaforschung der Bayer AG, für die Themenstellung, die vielfältigen Hinweise zum Finden von Antworten und die fortwährende Bereitschaft zur wissenschaftlichen Diskussion.

Prof. Dr. Hubert Ebert, mein Doktorvater, für die Möglichkeit, eine externe Promotion durchführen zu können, und für die Unterstützung während der Arbeit.

Dr. Christian Wünsche (Institutsleitung ehem. Strukturforschung) und Dr. Thomas Krämer (Abteilungsleiter MC VII), stellvertretend für die Bayer AG, für die Bereitstellung des Arbeitsplatzes und die Möglichkeit, die Infrastruktur der Firma nutzen zu können.

Dr. Markus Hauswald, Dr. Andreas Göller und die Postdocs Dr. Hongming Chen und Dr. Gertjan Boks, meine Abteilungskollegen in der Bayer AG, für das angenehme Arbeitsklima, die interessanten Diskussionen und die damit verbundenen Möglichkeiten, weitere Spezialgebiete kennenzulernen.

Dr. Til Huhne, Dr. Harald Freyer, Michal Kosuth und Jan Minar, meine Kollegen am Institut für Physikalische Chemie (Department Chemie der LMU), die den Kontakt zur universitären Basis ermöglichten.

Dr. Stefan Mundt (PH-R MST) als Ansprechpartner in der Bayer Pharma-Screeningabteilung.

Dr. Roger Brunne und Dr. Hans-Georg Rohbeck (beide PH-R SID) für die Hilfestellung bei Fragen rund um die Bayer Inhouse-Datenbanken.

Dr. Jens Ergüden, Dr. Wolfgang Thielemann und Dr. Henning Steinhagen, medizinische Chemiker in PH-R CWL, für das Interesse und die konstruktiven Rückmeldungen zu BayTree.

Dr. Ingo Mügge (Pharma Research Center Westhaven) und Dr. John Lohrenz (Pflanzenschutzzentrum Monheim) für die Testinstallationen von BayTree am entsprechenden Standort.

Dieter Thiemann (Zentrale Informatik) und Robert Kaminski (PH-R CWL) für die Administration der SGI-Rechner bzw. die Betreuung der PCs.

Dr. Wolf-Dietrich Ihlenfeld (Xemistry), Dr. Vincent Vivien (Bioreason), Dr. Chris Williams (CCG), Dr. Ulrike Uhrig (Tripos), und Markus Düringer (Spotfire) als Ansprechpartner zu den Produkten ihrer Firmen.

Wolfgang Kosten, stellvertretend für das Team der Wissenschaftlichen Bibliothek Elberfeld, für die Literaturbereitstellung.

Inhaltsverzeichnis

1 Einführung 1

1.1 Aufgabenstellung und Zielsetzung der Arbeit 1

1.2 Phasen der pharmazeutischen Forschung (Drug Development) 1

1.3 Technologiewandel und Paradigmenwechsel: Effizienzsteigerung durch Automatisierung 2

1.4 Hochdurchsatz-Screening (HTS) 3

1.5 HTS-Analyse: Selektion und Bewertung der Leitstruktur 4

1.6 Probleme der Informationstechnologie in der Wirkstoffforschung 5

2 Methoden zur Wirkstoffanalyse 7

2.1 Räumliche Analyse der Ligand-Target-Interaktion 7

2.2 Ligand bzw. Eigenschaftsähnlichkeit (Similarity) 8

2.3 Molekküldeskriptoren 9

2.4 Abstandsmaße und Metriken 13

2.5 Techniken zur Datengruppierung (Clusterverfahren) 16

2.6 Anwendungsgebiete für Strukturvergleiche 21

3 Vorhandene Systeme und kommerzielle Software 23

3.1 Molecular Spreadsheet 23

3.2 Deskriptorbasierte Clusterung von Strukturen 23

3.3 Spotfire 24

3.4 SCA: Scaffold-based Classification Approach 26

3.5 Distill 28

3.6 LeadScope 29

3.7 Bioreason 31

4 Bewertung und Planung 35

4.1 Nachteile der bekannten Strukturvergleichsverfahren 35

4.2 Idee und Realisierung des BayTree-Konzeptes 36

5 BayTree 39

5.1 Beschreibung der BayTree-Methode 39

5.1.1 Auswahl der Eingabestrukturen durch Eigenschaftsfilter 43

5.1.2 Graphentheoretische Beschreibung der Molekültopologie 43

5.1.3 Topologische Fragmentierung des Molekülgraphen 43

5.1.4 Idee der topologischen Referenzstruktur 46

5.1.5 Priorisierungsrational des Regelwerks 47

5.1.6 Priorisierung der topologischen Komponenten 49

5.1.7 Priorisierung innerhalb einer topologischen Kategorie 50

5.1.8 Erstellung des MolCode 52

5.1.9 Generierung der topologischen Referenzstruktur 54
5.1.10 Aufbau des topologischen Strukturbaums 55
5.1.11 Datenanalyse basierend auf den TST-Knoten 56
5.1.12 Priorisierung der chemischen Dekoration 57
5.1.13 Vergleich von Aktiv-TST und Inaktiv-TST 57

5.2 Verwendete Verfahren und Algorithmen 58
5.2.1 Ring Perception 58
5.2.2 Repräsentative molekulare Ringsets 60
5.2.3 Generierung der molekularen Linker 61
5.2.4 Berechnung von 2D-Koordinaten 63
5.2.5 2D-Alignment 64
5.2.6 Molecular Identification Numbers/Hash-Werte 65
5.2.7 Tree Layout/Dynamic Tree Drawing 69
5.2.8 Colorscales zur Knoten-Kolorierung 73
5.2.9 Estradas spektrale Momente 74
5.2.10 Lineare Diskriminanzanalyse 77
5.2.11 Modellbewertung per Kreuzvalidierung (leave-one-out) 81
5.2.12 Lückenanalyse im Strukturbaum der Eingabedaten 82

5.3 Implementierungsdetails 83
5.3.1 Die Scriptsprache Tcl/Tk 83
5.3.2 Das Cactvs-System 84
5.3.3 Erforderliche Kommandoerweiterungen und Zusatzprogramme 85
5.3.4 Anmerkungen zu Java 86

5.4 Erweiterungsmöglichkeiten 87

6 Anwendung auf den NCI Aids-Datensatz 89

6.1 Beschreibung des NCI Aids-Datensatzes 89

6.2 Topological Structure Tree 90

6.3 Generic Topological Structure Tree 96

6.4 Analyse des Belegungsgrades der Knoten 97

6.5 Aktivitätsanalyse der Knoten 101

6.6 Detailanalyse der aktiven Template 111

6.7 Festlegung der Anzahl der spektralen Momente zur Klassifizierung 115

6.8 Vergleich der Klassifikationsverfahren 116

6.9 Details zur LDA-Klassifikation 117

6.10 Verwendung lokaler Klassifikationsmodelle 120

6.11 Diskussion des MolCode-basierten Verfahrens 130

7 BayTree-Bedienungsanleitung 133

7.1 Grundlagen 133
7.1.1 Konfiguration und Programmstart 133
7.1.2 Switches beim Programmaufruf 133
7.1.3 Konfigurations-Datei .baytreerc 134
7.1.4 GUI-Komponenten 134
7.1.5 Command-Steuerung 136
7.1.6 Struktur-Eingabedaten: Formate und Erzeugung 137

7.1.7 Import von Aktivitätsdaten 139
7.1.8 Aktivitäts-Schwellenwert zur Klassenzuordnung 140

7.2 Templatbetrachtung 141
7.2.1 Navigation im Strukturbaum 141
7.2.2 Navigationshilfen für den Gesamtstrukturbaum 141
7.2.3 Baumlayout-Manipulationen mit dem Kontextmenü 144
7.2.4 Ein- und Ausblenden von Strukturen 145
7.2.5 Selektionslisten-Generierung und -Visualisierung 145
7.2.6 Struktur-Eigenschaftsanalyse durch Farbkodierung der Knoten 146
7.2.7 Zusammensetzung der Knoten 147
7.2.8 Belegungsgrad der Knoten/MolCodes 147

7.3 Strukturbetrachtung 149
7.3.1 Prädiktive Aktivitätsklassifizierung durch LDA 149
7.3.2 Kommandos zur Erstellung der Multimodell-Klassifikationen 152
7.3.3 XR-Table-Generierung: Dekonvolution der Templatdekoration 154
7.3.4 Exportmöglichkeiten für Bäume 155

8 Zusammenfassung 156

9 Abkürzungsverzeichnis 158

10 Glossar 159

11 Literatur 163

1 Einführung

1.1 Aufgabenstellung und Zielsetzung der Arbeit

In der folgenden Arbeit soll aufbauend auf den für die Pharmaforschung grundlegenden Prinzipien der Ähnlichkeit[1] und des Pharmakophors[2, 3] ein computergestütztes Verfahren zur Visualisierung und Aktivitätsanalyse von chemischen Strukturdatenbanken entwickelt werden, das interaktives Navigieren in beliebig großen informationsverdichteten Strukturbäumen erlaubt.

Die Notwendigkeit derartiger Analysen entspringt dem Einsatz automatisierter kombinatorischer Chemie[4], der Existenz von kommerziellen Synthonanbietern und der Etablierung der automatisierten Hochdurchsatz-Testung* (HTS High Throughput Screening) in der industriellen Pharmaforschung seit Beginn der 90er Jahre. Mit deren Hilfe wird eine große Anzahl von Verbindungen auf ihre biologische Aktivität zu Zielproteinen (Targets) geprüft. Die erzeugten Datenmengen können manuell nicht mehr effizient analysiert und ausgewertet werden, was die schnelle Identifizierung der Leitstruktur und ihre nachfolgende chemische Optimierung hinsichtlich Wirkstärke und pharmakologischem Profil erschwert. Es entstand ein Bedarf an neuen Methoden zur computergestützten Auswertung.

1.2 Phasen der pharmazeutischen Forschung (Drug Development)

Neue Arzneimittel sollen bei der kausalen Behandlung von Krankheiten und der Linderung von Schmerzen oder Altersbeschwerden helfen. Am Anfang der pharmazeutischen Forschung steht deshalb oft der Wunsch, ein neues oder wirksameres Medikament in einem definierten Indikationsbereich zu entwickeln. Dafür muss zuerst der Mechanismus der Krankheit verstanden werden oder zumindest ein möglicher Zielort (biologisches Target) für einen Wirkstoff identifiziert werden. Gibt es keine ausreichenden Informationen über das relevante (Ziel-)Protein und evtl. vorhandene endogene Liganden oder Targetmodulatoren, müssen geeignete Kandidaten durch „Probieren“ gefunden werden. Dies geschieht, indem eine große Zahl von Substanzen auf ihre Wirksamkeit getestet wird. Da die Hochdurchsatz-Testung (HTS) eine enorme Menge an Daten liefert, liegt das Problem in der schnellen und sicheren Leitstrukturidentifizierung. Der nächste Schritt ist die chemische Optimierung dieser Leitstruktur bezüglich biologischer Aktivität, Spezifität und pharmakokinetischer und toxikologischer Eigenschaften. Ist eine aussichtsreiche Substanz gefunden, ist das Forschungsziel erreicht. Dann muß die Wirksamkeit in aufwendigen und teuren klinischen Kontrollstudien an Probanden und Patienten bestätigt werden. Nach dem erfolgreichen Durchlaufen mehrerer klinischer Phasen in der Pharma-Entwicklung erhält das Medikament die Zulassung und erreicht nach vielen Jahren kostspieliger Forschung Marktreife. Nur ein sehr kleiner Prozentsatz der ursprünglichen Entwicklungskandidaten erreicht tatsächlich den Markt. Die meisten scheitern an unzureichenden Ergebnissen in den klinischen Tests (mangelnde Wirkung, toxische Nebenwirkungen, Unverträglichkeiten etc.). Je später eine Entwicklung gestoppt wird, desto höher sind die bereits angefallenen Kosten. Deshalb ist ein frühzeitiges Aussortieren („fail early, fail cheap“) von aussichtslosen Projekten wünschenswert. Andererseits ist es noch teurer, einen möglicherweise vorhandenen

* Eine Zusammenstellung der mehrfach benutzten Akronyme und Begriffe findet sich im Glossar in Kapitel 10 ab Seite 159.

Kandidaten für ein „Blockbuster-Medikament“ als singuläres Ergebnis zu übersehen („Nadel im Heuhaufen“). Ein Medikament kommt derzeit nach frühestens acht bis zwölf Jahren Forschungs- und Entwicklungsaufwand auf den Markt.

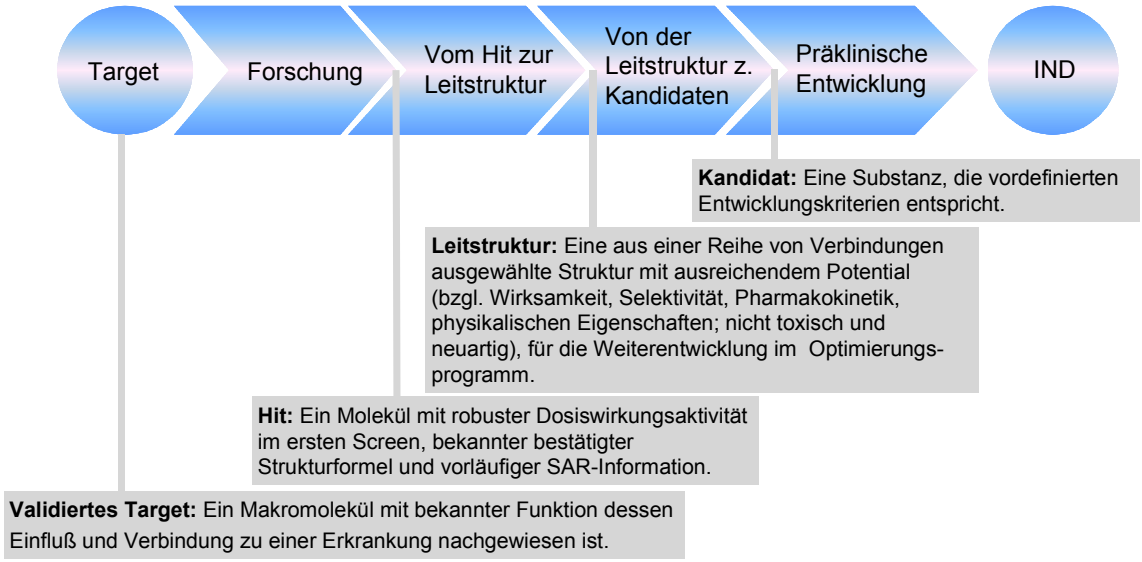


Abb. 1.1: Drug Discovery Vom Target zum IND (Investigational New Drug).

1.3 Technologiewandel und Paradigmenwechsel: Effizienzsteigerung durch Automatisierung

Um die Anzahl ihrer innovativen Medikamente auf dem Markt zu erhöhen, ist jede Firma bestrebt, möglichst viele Verbindungen durch Hochdurchsatz-Screening auf ihre pharmakologische Wirksamkeit und ihre Verwendbarkeit aufgrund ihrer Stoffeigenschaften zu prüfen. Dazu sind neue Technologien nötig, die es ermöglichen, große Testdatenmengen schnell zu erzeugen, hinsichtlich der wesentlichen Erfolgsparameter (chemische Merkmale, biologisches proof of principle, in vitro-Wirksamkeit, SAR, ADME, Toxizität etc.) zu bewerten und dann vorhandene Schwachstellen durch chemische Derivatisierung wegzuoptimieren.

Am Anfang steht die Identifizierung von neuen Targets durch Genomics. Seit dem ersten Abschluss des humanen Genom-Projektes[5] sind im Prinzip alle Gene und damit die Nukleotidsequenzen aller molekularen Targets[6] bekannt. Firmen wie z.B. Millennium[7] haben sich darauf spezialisiert, aus der großen Zahl der im menschlichen Organismus vorhandenen Gene die krankheitsrelevanten Proteine und ihre Funktion zu identifizieren und eine tragfähige Disease-Hypothese zu entwickeln. Dies geschieht mit Hilfe von Expressions-Profilen (microarray data) und Verfahren der Bioinformatik[8]. Anschließend ist das Problem zu lösen, die geeignetsten Targets gezielt mit niedermolekularen chemischen Liganden zu modulieren, d.h. in der Regel zu inhibieren oder zu aktivieren.

Dies gelingt zur Zeit nur durch massive Testung umfangreicher Substanzdatenbanken mit bekannten Strukturen, wobei neuerdings der Einsatz automatisierter Parallelsynthesen im

Rahmen der Kombinatorischen Chemie die Synthese mehrerer zehntausend Derivate eines geeigneten Strukturtemplates pro Laboreinheit und Jahr ermöglicht. Die Substanzbanken potenzieller Liganden werden in firmeneigenen Prüfpräparatelagern aufbewahrt. Wenn zu einem Target ein biochemisches Testsystem (Assay) vorliegt, werden die Prüfpräparate von Robotern automatisch entnommen und zum Testen auf geeigneten Mikrotiterplatten in Standardformaten konfektioniert. In der Hochdurchsatz-Testung (HTS) werden ebenfalls durch Roboter hochautomatisiert mehrere zehntausend Verbindungen pro Tag in enzymatischen Radio-Ligand-Assays oder Ganzzell-Assays[9] unter Verwendung von Reportergenen auf ihre biologische Aktivität in vitro getestet.

Da die Technologien etabliert sind, hat sich der Engpaß von der Datenerzeugung zur Datenauswertung verschoben[10]. Zur Entscheidungsfindung (decision support) ist erforderlich, die korrekte Durchführung des Screens zu gewährleisten, d.h. Kontrollen zur Detektion von Ausreißern oder systematischen Fehlern durchzuführen, und mögliche Kandidaten für falsch positive/negative Messungen zu identifizieren. Um die tatsächlich relevanten Daten zu extrahieren (data mining) und zur Grundlage neuer Erkenntnisse zu machen (turning data into knowledge), müssen gezielte Nachtestungen und analytische Kontrollen für die Reinheit (Beimischungen, Zersetzungen) der Testsubstanzen durchgeführt werden[11]. Erst dann kann die strukturbezogene Auswertung aller Daten tatsächlich beginnen. Dafür sind lediglich ansatzweise Programme vorhanden. Sowohl bei den zugrundeliegenden Algorithmen als auch bei der Anwendbarkeit sind Verbesserungen zwingend erforderlich.

1.4 Hochdurchsatz-Screening (HTS)

Bei der Hochdurchsatz-Testung wird durch Screening-Roboter hochgradig automatisiert eine große Anzahl an Substanzen in einem biologischen Assay getestet[12]. Damit erhält man zu jeder Verbindung einen Datenpunkt, der angibt, wie das biologische Testsystem beeinflußt, d.h. inhibiert oder aktiviert wurde. Die Substanzen mit der gewünschten Aktivität werden als Primärhits bezeichnet und nach einer Plausibilitätsanalyse in einem zweiten Test validiert (Nachtestung, confirmed hit).

Aufgrund der enormen Testgeschwindigkeit und der sinkenden Kosten pro Datenpunkt kann auf diese Weise in kürzester Zeit der gesamte Firmen-Substanzpool gescreent werden. Dieser enthält vor allem proprietäre Strukturen aus früheren Projekten, deren Umfang und Art von Firma zu Firma verschieden sind. Die chemische und strukturelle Diversität wird meist durch Einkauf von Substanzen oder Naturstoffextrakten externer Anbieter erhöht, wobei allerdings die Gefahr besteht, dass diese zur gleichen Zeit auch anderen Firmen zur Verfügung gestellt worden sind.

HTS ist die bevorzugte Methode, um zu geeigneten Liganden zu kommen, wenn es sich um Genomics-Targets handelt, für die es keine bekannten Liganden gibt. Zu bekannten Targets können andersartige Leitstrukturen bzw. Liganden gefunden werden, die noch nicht durch Patente von Konkurrenzunternehmen geschützt sind.

Mit der (gewünschten) hohen Geschwindigkeit wird in Kauf genommen, dass es einen nicht unerheblichen Anteil an Falschmessungen gibt. Von den validierten Hits werden einfache Dosis-Wirkungs-Kurven erstellt, aus denen ersichtlich ist, ob eine höhere Konzentration auch tatsächlich eine höhere Wirkung aufweist oder ob es sich um eine falsch positive Messung gehandelt hat. Auf diese Art können bis zu einige Tausend Verbindungen gemessen werden, aus denen dann eine oder mehrere Leitstrukturen selektiert werden.

Für alle wesentlichen Targetklassen besteht die Möglichkeit, ein molekularbiologisches Testsystem zu entwickeln. Vorzugsweise werden gereinigte Proteine verwendet. Falls dies nicht möglich ist, weil z.B. das spezielle Zielprotein (noch) nicht bekannt ist, kann eine gesamte Signalkaskade in whole cell assays getestet werden. Die wesentliche Aufgabe ist, die Interaktion zwischen Ligand und Target in ein makroskopisch messbares Antwortsignal zu verwandeln. Beim secondary screen zur Ermittlung von Dosis-Wirkungs-Kurven ist zusätzlich erforderlich, dass die Auswertung semiquantitativ erfolgen kann.

Führend auf dem Gebiet der Screening-Technologie ist die Firma EVOTEC OAI. Die neueste Entwicklung ist EVOscreen® NanoCarrier™ (2080-well plates) in Verbindung mit Fluorescence Correlation Spectroscopy (FCS), basierend auf Einzelmoleküldetektion mit konfokaler Laseroptik[13]. Aus dem Diffusionsverhalten der Substanzen im Testsystem wird abgeleitet, ob eine Bindung stattgefunden hat (langsame Diffusionbewegung) oder nicht (schnelle Diffusionsbewegung).

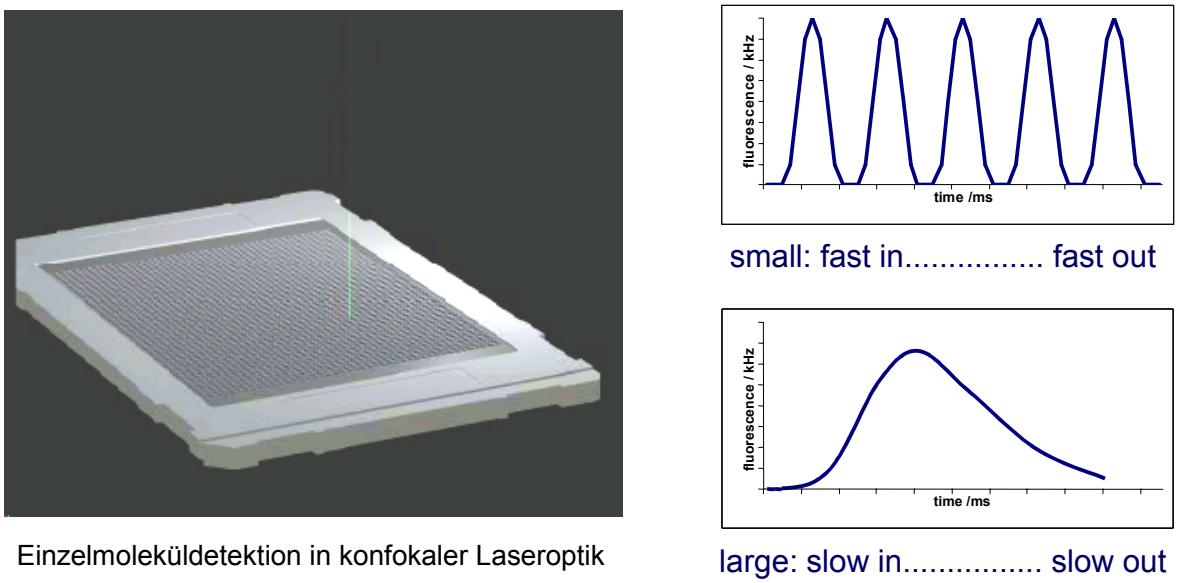


Abb. 1.2: EVOscreen® NanoCarrier™ (2080-well plates) & Fluorescence Correlation Spectroscopy.

1.5 HTS-Analyse: Selektion und Bewertung der Leitstruktur

Die tatsächliche Wirkstoffsuche beginnt mit der Auswertung der HTS-Ergebnisse. Der erste Schritt der Hitlistenbewertung besteht in der gesicherten Identifizierung der chemischen Grundkörper, die an dem betrachteten Target Wirkung auslösen, und der Art und Weise, wie diese durch chemische Derivatisierung in mitgetesteten Analoga modifiziert wurde. Auf diesem Weg will man herausfinden, in welchem Umfang sich diese Template zur chemischen Weiterbearbeitung eignen. Bei der Leitstrukturselektion (Lead-Picking) werden je nach Zusammensetzung und Qualität der Datenbasis eine oder mehrere Strukturklassen als Leitstrukturkandidaten nach chemischen Gesichtspunkten und Erfahrungen bzw. Ergebnissen der bisherigen Testungen oder Optimierungen ausgewählt. Die wichtigsten Selektionskriterien sind:

- Synthetische Zugänglichkeit (chemische Machbarkeit, KombiChem-Tauglichkeit)
- Modifikations- und Optimierungsmöglichkeiten (nicht zu schwere bzw. lipophile Ausgangsverbindungen[14])
- Auffälligkeit der Substanzklasse in bisherigen Testungen oder Optimierungsprogrammen
- Drug-likeness (Grundsätzliche Eignung eines Moleküls als Drug mit einem aus bekannten Drugs abgeleiteten typischen Eigenschaftenprofil)
- Hitstatistik des Templates (minimale Anzahl an Vertretern, hohes Risiko bei Singletons)
- Pharmakologische (ADME-Parameter) und physikochemische Eigenschaften (Löslichkeit und chemische Stabilität)
- Patentsituation

Die Auslegung der Kriterien hängt stark von der Erfahrung des jeweiligen medizinischen Chemikers ab und enthält immer ein gewisses Maß an Intuition.

Der Prozeß der Leitstrukturselektion wird in den seltensten Fällen dokumentiert, da der Fokus auf dem frühzeitigen und schnellen Beginn der Chemieaktivität liegt.

Die gewählten Leitstrukturen bzw. Leitstrukturklassen werden durch Substruktur- bzw. Ähnlichkeitssuchen (sog. Umfeldstudien) ergänzt, die Verbindungen mit gleichem Grundgerüst sowie chemisch entferntere Verwandte für die biologische Testung heraussuchen. Aufgrund der weiten Verbreitung und einfachen und schnellen Anwendbarkeit werden bevorzugt Fingerprints (siehe Abschnitt 2.3.1.2) in Verbindung mit dem Tanimoto-Koeffizienten (siehe Abschnitt 2.4) angewendet. Der Schwellenwert für die Ähnlichkeit wird iterativ so lange variiert, bis die gewünschte Anzahl von ähnlichen Verbindungen gefunden worden ist. Sind in der Firmendatenbank sehr viele ähnliche Strukturen enthalten, kann ein großer Schwellenwert verwendet werden, sind nur sehr wenige vorhanden, wird ein entsprechend kleinerer gewählt. Erfahrungsgemäß erzeugen Ähnlichkeitsabfragen mit einem Tanimoto-Schwellenwert unterhalb 0,6 Ergebnisse, die keinerlei Bezug mehr zur verwendeten Ausgangstruktur haben.

Die derart ermittelte Auswahl an Strukturen wird erneut im HTS gemessen. Aus den Ergebnissen wird eine erste vorläufige Struktur-Aktivitäts-Beziehung (Structure Activity Relationship, SAR) abgeleitet, die die ersten chemischen Modifikationen vorgibt.

Die SAR ermöglicht es, aus strukturellen oder strukturbezogenen Größen einen Trend der biologischen Aktivität abzuleiten. Im Gegensatz zur Quantitative Structure Activity Relationship (QSAR) sind keine quantitativen Vorhersagen möglich, es können aber erste Vermutungen über die chemische Interaktion zwischen Ligand und Target angestellt und mit Pharmakophoranalysen und 3D-Alignments untermauert werden.

1.6 Probleme der Informationstechnologie in der Wirkstoffforschung

Selbst in dem sehr eng gefaßten Bereich der Wirkstoffforschung gibt es zahlreiche Schwachstellen bzw. unvollständig gelöste Probleme der Informationsverarbeitung und Wissensbewertung sowie ihrer Nutzung zur Wirkstoffoptimierung. Dies hat mehrere Gründe:

- Es fallen sehr umfangreiche biologische Testdaten unterschiedlicher Qualität an (Radio-Ligand-Assays, Ganzzell-Tests, bakterielle Hemmkonzentrationen etc.)
- Die Bewertung der Testdaten hängt von der Kenntnis zusätzlicher Stoffeigenschaften wie Löslichkeit, Stabilität, Reinheit, Toxizität, Pharmakokinetik ab. Die Berücksichtigung all dieser Parameter für die Nützlichkeit der getesteten Verbindungsklassen ist nur durch entsprechende Experten möglich und zieht chemische Schlußfolgerungen für den weiteren Projektverlauf nach sich.
- Die biologischen Wirkdaten müssen jeweils kontextbezogen (für jeden Assay bzw. jedes Target) neu ausgewertet werden. Dabei muß die „Qualität" von Leitstrukturen sowie das chemische „Potential" (chemische Zugänglichkeit, IP-Wert, therapeutisches Eigenschaftsprofil etc.) der Verbindungsklassen beurteilt werden. Dies erfolgt über die Auswirkungen von chemischen Modifikationen in den Templaten.
- Alle Meßdaten sind auf komplexe Weise von der chemischen Struktur, der Konstitution des Templates der Testverbindung und der räumlichen Konformation in Lösung unter den gegebenen Meßbedingungen (pH-Wert, Lösungsmittel, Salzgehalt, Target- und Ligand-Reinheit etc.) abhängig.
- Über die Wirkstoffkonstitution müssen sehr viele unterschiedliche Target- und Ligand-Informationen kausal verknüpft und daraus neue chemische Optimierungsvorschläge für ausgewählte Template unter besonderer Berücksichtigung der chemischen Zugänglichkeit abgeleitet werden.
- Die Notwendigkeit, ein vordefiniertes Wirkstoff-Eigenschaftsprofil für die klinische Verwendbarkeit erreichen zu müssen, erfordert die simultane Optimierung der Wirkstärke, der physikalischen Stoffeigenschaften (v. a. Löslichkeit) und der chemischen Eigenschaften (Zugänglichkeit, Stabilität, Verfügbarkeit im Zielorganismus, Abwesenheit toxischer Eigenschaften, Spezifität der Wirkung etc.), die alle auf unterschiedliche Weise selbst strukturabhängig sind.

Manche der erforderlichen Bewertungen ergeben sich erst aus der vergleichenden Analyse des Verhaltens der Verbindungen in mehreren unterschiedlichen biologischen Tests. Zu den Bewertungen gehören:

- mangelnde Spezifität der Wirkung
- ungünstige chemische oder physikalische Eigenschaften
- toxische Nebenwirkungen
- mangelnde Optimierbarkeit bestimmter Eigenschaftsparameter (metabolische Stabilität, pharmakokinetische oder toxische Eigenschaften etc.)
- universelle Nutzbarkeit „privilegierter" Template[15, 16] (Diese treten entweder in vielen biologischen Tests als aktiv auf oder sind spezifisch für eine spezielle Gruppe von Targetmolekülen.)

Deshalb ist für die Substanzdatenbank-Bewertung auch der strukturbezogene Vergleich von Screening-Ergebnissen an unterschiedlichen Targets sehr aufschlußreich. Das Fehlen von Tools zur strukturbezogenen Informationsverdichtung großer Datenbestände sowie der Mangel einer standardisierten strukturbezogenen Informationszusammenführung limitieren die Leitstrukturbewertung und die Aufdeckung von Meßfehlern, d.h. falsch positiver/negativer Befunde. Derartige Tools sind auch für die Identifizierung systematischer oder strategischer Lücken im Substanzpool und der IP-Bewertung von Projekten erforderlich.

2 Methoden zur Wirkstoffanalyse

In der Praxis kommt eine umfangreiche Hierarchie von Methoden bei der Wirkstoffanalyse zum Einsatz. Die einzelnen Verfahren unterscheiden sich in der Vorgehensweise, dem Bearbeitungsaufwand, der Qualität der Ergebnisse und dem Grad der Automatisierung bzw. der Aussagesicherheit. Nachfolgend werden einige Techniken exemplarisch beschrieben.

Ein rechnergestützter Vergleich umfangreicher Strukturdatensätze kann prinzipiell unter verschiedenen Aspekten durchgeführt werden:

- räumliche Analyse der chemischen Ligand-Target-Interaktion
- in einem Deskriptor- bzw. Eigenschaftsraum
- in einem Ähnlichkeitsraum („chemische“ Distanzen)
- topologisch auf der Basis (maximaler) gemeinsamer Substrukturen.

Bislang ist kein Verfahren entwickelt worden, das eine vollautomatische Anwendung gestattet. Bisher bekannte teilautomatisierte Verfahren zielen darauf ab, eine Homogenisierung der Daten, z.B. durch Clusterung, zu erreichen, um so repräsentative Wirkstrukturen und versteckte SARs zu identifizieren oder Unterschiede bzw. Gemeinsamkeiten im physikalischen Eigenschaftsraum der aktiven/inaktiven Testverbindungen aufzudecken.

Zu den bekanntesten expertenunterstützenden Softwarepaketen dieser Art gehören unter anderem LeadScope (siehe Abschnitt 3.6) und Bioreason (siehe Abschnitt 3.7).

In Rahmen dieser Arbeit ist ein neues Programm für die Bayer-Pharmaforschung entwickelt worden, das eine standardisierte automatisierte Visualisierung und Analyse des Templatraums von Datenbanken erlaubt, um z.B. bei Anwendung auf HTS-Datensätze den Effekt chemischer Modifikationen (im folgenden auch als Dekoration bezeichnet) auf die Bioaktivität für alle Leitstrukturkandidaten gleichzeitig und einheitlich bewerten zu können.

2.1 Räumliche Analyse der Ligand-Target-Interaktion

Um den Einsatz kostspieliger Experimente durch Fokussierung auf essentielle Fragestellungen zu minimieren, wird versucht, am Computer ein Modell zu erzeugen, das zusätzliche Einblicke in den Bindungsmechanismus der Moleküle erlaubt und Vorhersagen über die Bindung von neuen Strukturen ermöglicht (Modelling). Den größten Informationsgehalt haben 3D-Strukturen, die in der Regel in Verbindung mit Kraftfeld-methoden[17, 18] verwendet werden. Der ideale Ausgangspunkt für das Design von neuen Drugs ist das Vorliegen einer 3D-Proteinstruktur im Komplex mit einem Liganden, die durch Protein-Kristallographie oder NMR-Messungen charakterisiert wurde. Dadurch ist die active site identifiziert und der Bindungsmodus bekannt. Dies ermöglicht das direkte Ableiten neuer Synthesevorschläge durch bioisosteren Ersatz[19], das Mapping von Bindungstaschen oder die Bewertung von ungetesteten Datenpools durch Docking-Rechnungen[20, 21] und Auswertung der Resultate.

Da für innovative Targets oft keine räumliche Proteinstruktur bekannt ist, müssen Pharmakophormodelle im 3D-Raum durch das Überlagern[22, 23, 24] bekannter, möglichst strukturdiverser Liganden abgeleitet werden (indirektes oder Ligand-basiertes Drug Design).

Mit einem solchen Modell können Datenbanken durch eine flexible 3D-Suche[25] ebenso virtuell gescreent werden wie dies beim Docking beschrieben wird[26].

Die 3D-Rechnungen sind sehr rechenzeitintensiv und oft für große Datenmengen, wie sie im HTS anfallen, nicht geeignet. Zusätzlich ist die automatisierte Bewertung der sehr umfangreichen Rechenergebnisse vor allem durch multiple Bindungskonformationen, problematisch, da gleichzeitig auch schwierige Target-Konformationsänderungen, Protonierungsänderungen und Solvenseffekte, die die Bewertungen der Ergebnisse beeinflussen, berücksichtigt werden müssen.

Deshalb gibt es vereinfachte konstruktive Inkrementansätze zum Hochdurchsatz-Docking wie beispielsweise das FlexX-Programm[27]. Der Nachteil dieser Verfahren liegt nach wie vor in der relativen Bewertung der vielen Lösungen, wobei strukturelle Targetvariationen, multiple Bindungsmodi und gegebenfalls tautomere Formen und unterschiedliche Ligand/Target-Protonierungszustände meist unberücksichtigt bleiben.

Eine noch größere Vereinfachung stellt die zweidimensionale Bearbeitung (2D) der Strukturen dar. Ist in 2D eine Vorauswahl getroffen, lassen sich präzisere Betrachtungen zum räumlichen Strukturvergleich und Mechanismus der Ligand/Target-Wechselwirkung wieder in 3D durchführen.

2.2 Ligand bzw. Eigenschaftsähnlichkeit (Similarity)

Die Verwendung des Ähnlichkeitsprinzips beruht auf dem Paradignum, dass „genügend" ähnliche Verbindungen auch ähnliche biologische Effekte auslösen müssen[28]. Im asymptotischen Grenzfall, d.h. im Bereich chemischer Identität (isotopenmarkierter Verbindungen), ist dies nachweisbar gültig. Tatsächlich umfaßt jede einzelne chemische Änderung aber ein ganzes Spektrum möglicher Änderungen in den Bereichen chemische Struktur (Geometrie), Topologie (Konstitutionsformel), Isomerie (Stereochemie), Tautomerie, physikalische Eigenschaften (Polarität, Löslichkeit, Azidität, Basizität etc.) und chemische Reaktivität. Damit stellt sich das Problem der Quantifizierung der Ähnlichkeit und ihrer Bewertung für jedes betrachtete Target neu und kann prinzipiell nicht absolut beantwortet werden.

Der Begriff Similarity wird zwar häufig verwendet, ist aber nicht klar definiert und wird daher je nach Anwendungszweck und Kontext von den Benutzern unterschiedlich ausgelegt. Zum Beispiel werden Objekte als ähnlich oder unähnlich klassifiziert, oder es wird ein Maß definiert, das den Grad an Ähnlichkeit bzw. Unähnlichkeit quantifiziert. Zwei DNA-Sequenzen werden als ähnlich betrachtet, wenn das Sequenzalignment einen gewissen Grad an Identität aufweist. Proteinstrukturen werden nach einer 3D-Überlagerung der rigiden Objekte als ähnlich betrachtet, wenn die mittlere Abstandsabweichung (root mean square distance) ihrer Backbone-Atome unter einen gewissen Wert fällt.

Die Ähnlichkeit von 3D-Strukturen kleinerer Wirkstoffmoleküle, die an das gleiche Target binden, muß mehrere unterschiedliche Faktoren einschließen, um einen gemeinsamen chemischen Mechanismus am Target auslösen zu können: funktionale und chemische als Basis für ein gemeinsames Pharmakophormodell ebenso wie physikalische Eigenschaften, z.B. Elektrostatik und Liphophilie. Rechnerisch lassen sich diese Faktoren z.B. mit Hilfe des Hodgkin- oder Carbó-Index quantifizieren[29]. Diese 3D-Ähnlichkeitsmaße integrieren den Überlapp von molekularen elektrostatischen Potentialen sowie die Überlagerung der

Atomvolumina im Raum. Die Potentiale werden von atomzentrierten Eigenschaften bestimmt. Unter Berücksichtigung der Flexibilität der Strukturen (rotierbare Bindungen) läßt sich dieses Ähnlichkeitsmaß zwischen zwei Molekülen maximieren (siehe Abb. 2.1).

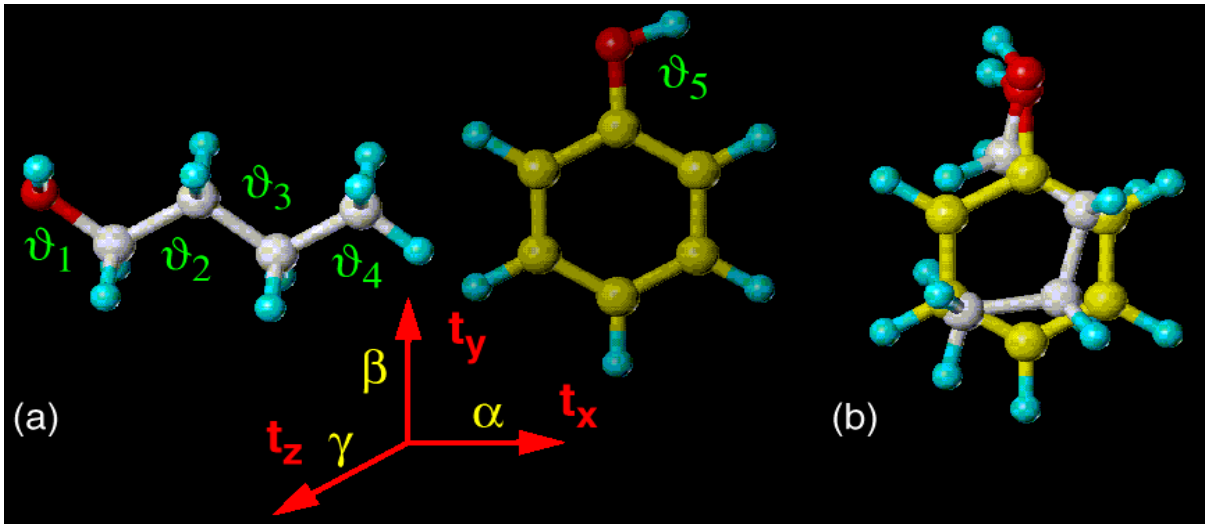


Abb. 2.1: Similarity-Alignment der Moleküle Phenol und Butanol. (a) Butanol (graues Kohlenstoffgerüst) und Phenol (gelbes Kohlenstoffgerüst). Das System besitzt 11 Freiheitsgrade: die rotierbaren Bindungen ϑ_1 bis ϑ_5, die Translationen t_x, t_y, t_z und die Rotationswinkel α, β, γ. (b) Die Überlagerung (Similarity-Alignment) der beiden Moleküle mit maximalem Hodgkin-Index von 0.8899 nach erfolgter Optimierung durch FAME[30].

Der Wertebereich liegt üblicherweise zwischen 0 (maximal unähnlich) und 1 (maximal ähnlich). Die Faktoren der 3D-Ähnlichkeitsmaße sind, wie erwähnt, so gewählt, dass eine hohe Ähnlichkeit auch eine ähnliche Wirkung an einem molekularen Wirkort bedeutet („Ähnliches wirkt ähnlich“). Dies ist eine für das drug design angestrebte und sehr hilfreiche Eigenschaft. Die Berechnung der 3D-Ähnlichkeitsmaße ist allerdings rechenintensiv und daher nicht für sehr große Datenmengen anwendbar. Dies hat zur Entwicklung von vereinfachten Ansätzen in 2D geführt.

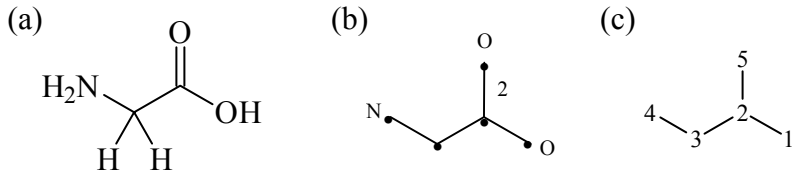
Diversität (auch Dissimilarity) ist der komplementäre Begriff zu Similarity. Er ist ebensowenig eindeutig definierbar wie Similarity und wird vor allem bei der Betrachtung von Gruppen von Molekülen in Trainingsdatensätzen, Kombinatorischen Bibliotheken oder Substanzpools verwendet. Er beschreibt, wie unterschiedlich die enthaltenen Strukturen zueinander sind. Je nach Fragestellung ist eine hohe Diversität wünschenswert: Substanzpools sollen einen möglichst großen Teil des chemischen Raums abdecken. Trainingsdatensätze müssen diverse Strukturen enthalten, um eine gewisse Vorhersagekraft eines Modells zu ermöglichen.

2.3 Molekülдeskriptoren

Aus der Idee der Molekülähnlichkeit und ihrer Bedeutung für die Wirkstoffanalyse entsprang der Versuch, die Ähnlichkeit statt aus der Gesamtstruktur aus einigen wenigen, aber wichtigen Teileigenschaften der Moleküle abzuleiten. Man glaubte, dadurch Moleküle schneller und quantitativ einfacher in ihrer Wirkung beschreiben und die großen Datenmengen schneller und effizienter auszuwerten zu können. Molekülдeskriptoren können in zwei Gruppen unterteilt werden. Zum einem gibt es die 3D-Deskriptoren, die von der Konformation der Strukturen abhängig sind, wie z.B. surface area, polar surface area und

Pharmakophor-Multipletts. Zum anderen gibt es die große Gruppe der konformationsunabhängigen 2D-Deskriptoren, die auf den Informationen der chemischen Struktur basieren. Die Graphentheorie definiert die chemische Struktur als einen kanten- und knotengefärbten bzw. gewichteten Graphen. Er enthält neben der Information über die Konnektivitäten der Knoten auch deren Atomtyp als Farbe bzw. Gewicht. Die Kantenfärbung entspricht der Bindungsordnung. Wasserstoffatome werden im allgemeinen nicht berücksichtigt und daher ignoriert. Alle Informationen können in Graphenalgorithmen (Isomorphismus, Clique-Detektion, MCS) einbezogen werden.

Molekül	↔	ungerichteter Graph G = (V, E)
Atom	↔	Knoten V (vertex, node); Knotenfarbe: Atomtyp
Bindung	↔	Kante oder Ecke E (edge); Bindungsfarbe: Bindungsordnung



$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \qquad \mathbf{D} = \begin{pmatrix} 0 & 1 & 2 & 3 & 2 \\ 1 & 0 & 1 & 2 & 1 \\ 2 & 1 & 0 & 1 & 2 \\ 3 & 2 & 1 & 0 & 3 \\ 2 & 1 & 2 & 3 & 0 \end{pmatrix}$$

adjacency matrix — distance matrix (shortest paths)

Abb. 2.2: Graphentheoretische Betrachtung des Moleküls Glycin (a). Im zugehörigen Graphen (b) werden Atomtyp und Bindungsordnung berücksichtigt, alle Wasserstoffatome werden ignoriert. Die Verknüpfung der Knoten ist in der adjacency matrix A kodiert, die verwendete Nummerierung der Knoten (sie entspricht der Zeilen- und Spaltennummer) ist (c) zu entnehmen. Die distance matrix D enthält die kürzesten Pfade der Knoten zueinander.

Da die komplette Grapheninformation von mathematischen Chemometriemethoden nicht berücksichtigt werden kann, muss ein möglichst großer Teil davon numerisch kodiert werden. Die Transformation in einen Deskriptorraum geht meist mit einem Informationsverlust einher, immer aber unter Verlust der „chemischen Struktur“. Dies erschwert die Rückübersetzung (inverses QSAR/QSPR-Problem) der gewonnenen Informationen in „neue Chemie“, die eigentlich zu lösende praktische Aufgabe der Datenanalyse.

Wegen der ständig wachsenden Anzahl berechenbarer Deskriptoren wurde versucht, universell anwendbare „optimale“ Deskriptoren zu finden bzw. zu definieren[31]. Da bekanntermaßen nicht einmal die Strukturformel in allen Fällen die strukturellen und elektronischen Eigenschaften komplexer Moleküle beschreiben kann (Säure/Base-Gleichgewicht, Tautomerie, Resonanzformel, Valenzisomere etc.), muß die Existenz bzw.

Aussagekraft universeller Deskriptoren bezweifelt werden. Da es keinen Satz solcher universell verwendbarer Deskriptoren gibt, müssen für jeden Anwendungsfall die bestmöglichen ausgewählt werden. Dafür stehen Selektionsverfahren zur Verfügung[32, 33]. Zur qualitativen Bewertung der Permeabilität von Molekülen im Organismus hat sich die von Lipinski aus der Praxis abgeleitete „rule of five“[34] (Lipophilie, Molekulargewicht, logP, Anzahl HB-Donatoren, Anzahl der HB-Akzeptoren) bewährt. Für die Vorhersage der biologischen Aktivität sind andere Deskriptoren zur Beschreibung der strukturellen Ähnlichkeit erforderlich. Sie müssen jedes Mal individuell festgelegt werden.

2.3.1.1 Graphentheoretische 2D-Deskriptoren

Mittlerweile gibt es eine kaum noch überblickbare Flut von Moleküldeskriptoren[35], die unterschiedlichste Moleküleigenschaften (Topologie, Größe, Molekülform, Ladungsverteilung, Oberfläche, Volumen, chemische und elektronische Eigenschaften etc.) beschreiben, um eine breite Palette von Aufgabenstellungen im Bereich der Struktur-Wirkungsforschung, der Syntheseplanung oder der Materialeigenschaften bearbeiten zu können.

Es gibt Programme, die eine sehr große Zahl an unterschiedlichen Deskriptoren berechnen können, z.B. Molconn-Z[36], QuaSAR-Descriptor[37], VolSurf[38] und Dragon[39].

Dragon V3.1 kann 1497 molekulare Deskriptoren berechnen, die in folgende 18 Gruppen unterteilt sind. Die genaue Anzahl der Deskriptoren ist in der Klammer dahinter angegeben.

- constitutional descriptors (47)
- topological descriptors (266)
- molecular walk counts (21)
- BCUT descriptors (64)
- Galvez topological charge indices (21)
- 2D autocorrelations (96)
- charge descriptors (14)
- aromaticity indices (4)
- Randic molecular profiles (41)
- geometrical descriptors (70)
- RDF descriptors (150)
- 3D-MoRSE descriptors (160)
- WHIM descriptors (99)
- GETAWAY descriptors (197)
- functional groups (121)
- atom-centred fragments (120)
- empirical descriptors (3)
- properties (3)

Besonders häufig eingesetzte Deskriptoren sind:

- BCUT-Values[40] aus Diverse Solutions[41]
- Electrotopological State Indices[42]
- Topological Torsions[43]
- Atom Pairs[44]

2.3.1.2 Fingerprints

Weite Verbreitung und Anwendung findet die Ähnlichkeitsberechnung basierend auf Fingerprints und Tanimoto-Koeffizient. Bis zu einem gewissen Maß entspricht sie dem intuitiven Verständnis von Ähnlichkeit[45]. Diese strukturelle Ähnlichkeit ist in weitaus geringerem Maße mit der biologischen Ähnlichkeit verknüpft, als lange angenommen wurde. Verbindungen mit einer Ähnlichkeit von größer 0,85 zu einer aktiven Verbindung sind nur mit einer Wahrscheinlichkeit von 30% und nicht - wie ursprünglich publiziert - von 80% aktiv[46]. Fingerprints sind Bitvektoren zu einer Struktur, die das Vorhandensein und die Häufigkeit von Substrukturfragmenten in vordefinierten Bitbereichen kodiert haben. Je nach Definition des Fingerprint haben einzelne Atome oder Fragmente einen besonders großen Einfluß auf die Ähnlichkeit und führen zu unerwarteten Resultaten. Ursprünglich wurden Fingerprints entwickelt, um für Abfragen in Datenbanken einen schnellen Vorfilter zu haben. Sie werden daher eigentlich bestimmungsfremd verwendet.

Erweiterungen im Hinblick auf eine Anwendung als Deskriptoren für QSAR oder Chemieraumdefinitionen (chemspace) führten zu „hashed fingerprints“ und „hologram fingerprints“[47]. Erstere kodieren zusätzlich die Häufigkeit des Auftretens von Fragmenten vorgegebener Größe bis zu einem vorgegebenen Maximum. Letztere erfassen alle Fragmente aller Größen einer Struktur und kodieren dann die Summe binär.

Werden sehr lange Fingerprints definiert, in denen voraussichtlich nur sehr wenige Bits gesetzt sind, werden sie zur Speicherersparnis gefaltet oder gehashed. Dies kann dazu führen, dass die gesetzten Bits aufgrund von Kollisionen nicht mehr eindeutig zugeordnet werden können.

Weit verbreitet sind die Definitionen von MDL MACCS/ISIS[48] (166 keys für funktionelle Gruppen), von Daylight[49] (1024, 2048, 4096 bits, die alle Substrukturen zwischen null und sieben Bindungen kodieren) und von Tripos Unity[50] (988 bits, eine Kombination aus keys für funktionelle Gruppen und Elemente und Substrukturen bis zu einer vorgegebenen Bindungsanzahl).

Die Kombination aus Fingerprints und Tanimoto-Koeffizient besitzt die folgenden nachteiligen Eigenschaften:

- sie beschreiben keine saubere Metrik
- sie verletzen das Pharmakophor-Konzept
- sie korrelieren nicht mit der Bioaktivität
- sie lassen sich nicht einfach in Chemie „zurückübersetzen“
- sie erfordern eine Vorabprozessierung des gesamten Datenbestandes
- sie erschweren templatbezogene Auswertungen.

Wegen des vorteilhaften schnellen Handlings der Fingerprints bei der Suche und der Berechnung von Ähnlichkeiten werden sie auch zur Kodierung anderer Informationen verwendet. Die 3D-Pharmakophor-Fingerprints enthalten in den Bits kodiert Information über die möglichen Distanzen von Pharmakophor-Features im Raum unter Berücksichtigung verschiedener Konformationen des zugehörigen Moleküls[51]. Über einen Vergleich dieser Fingerprints läßt sich zu einer Anfrage schnell eine Auswahl an in Frage kommenden Strukturen ermitteln.

2.4 Abstandsmaße und Metriken

Um den abstrakten Begriff der Ähnlichkeit bzw. Diversität möglichst einfach und anschaulich handhaben zu können, verwendet man oft Abstandsmaße zwischen den Deskriptor-Merkmalen der zu vergleichenden Verbindungen. Mit allen Zahlenvektoren beliebiger Dimensionalität können Distanzen zur Quantifizierung der Eigenschafts- und Ähnlichkeitsräume berechnet werden. Dafür gibt es unterschiedliche Metriken. Die Wahl der Metrik hat offensichtlich einen Einfluß auf das Ergebnis. Im Fall einer Diversitätsanalyse ist vor allem die erforderliche Rechenzeit von Bedeutung, was zur Folge hat, dass einfache, schnell zu berechnende Metriken wie die Manhattan-Distanz bevorzugt werden. Die bekannteste und am häufigsten verwendete Distanzfunktion ist die Euklidische Distanz. Eine statistisch verbesserte Form ist die Mahalanobis-Distanz.

Unter einer Distanzfunktion d versteht man ganz allgemein eine Funktion, die den Abstand zweier n-dimensionaler Vektoren **x** und **y** mißt. Die Funktion muß folgende Eigenschaften haben:

- $d(\mathbf{x}, \mathbf{x}) = 0$
- $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$
- $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$

Dies bedeutet, dass der kürzeste Abstand zweier Punkte die direkte Verbindung ist.

Die *Euklidische Distanz* ist als Wurzel aus der Summe aller quadrierten Komponenten-differenzen definiert:

$$d(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} = \sqrt{(\mathbf{x}-\mathbf{y})^T(\mathbf{x}-\mathbf{y})} = \|\mathbf{x}-\mathbf{y}\|$$

Bei drei bzw. zwei Koordinaten entspricht sie anschaulich dem Luftlinienabstand zweier Punkte im Raum bzw. in der Ebene.

Die *City Block-Distanz oder Manhattan-Distanz*[52] ist die Summe der Koordinatendifferenzen in allen Dimensionen:

$$d(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n}|x_i - y_i|$$

Der City Block-Abstand ist von Vorteil, wenn die verwendeten Attribute nicht kontinuierlich, sondern diskret, d.h. diskontinuierlich sind.

Der *Mahalanobis-Abstand*[53] berücksichtigt Verzerrungen, die durch korrelierende Variablen entstehen. Der Abstand ist bezüglich der Verteilung bzw. Form der zur Berechnung der Kovarianzmatrix verwendeten Datenpunkte korrigiert. Für nicht korrelierte Variablen entspricht die Kovarianzmatrix der Einheitsmatrix und der Mahalanobis-Abstand entspricht der Euklidischen Distanz.

$$d(\mathbf{x},\mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^T S^{-1}(\mathbf{x}-\mathbf{y})}$$

S^{-1} ist die Inverse der Kovarianzmatrix.

Zur graphischen Veranschaulichung enthält die Abb. 2.3 die geometrischen Orte, die vom Koordinatenursprung den Euklidischen, Manhatten- und Mahalanobis-Abstand eins besitzen (Kreis, Quadrat und Ellipse)[52]. Dabei sind Lage und Exzentrizität der Ellipse von der Gestalt der Kovarianzmatrix abhängig.

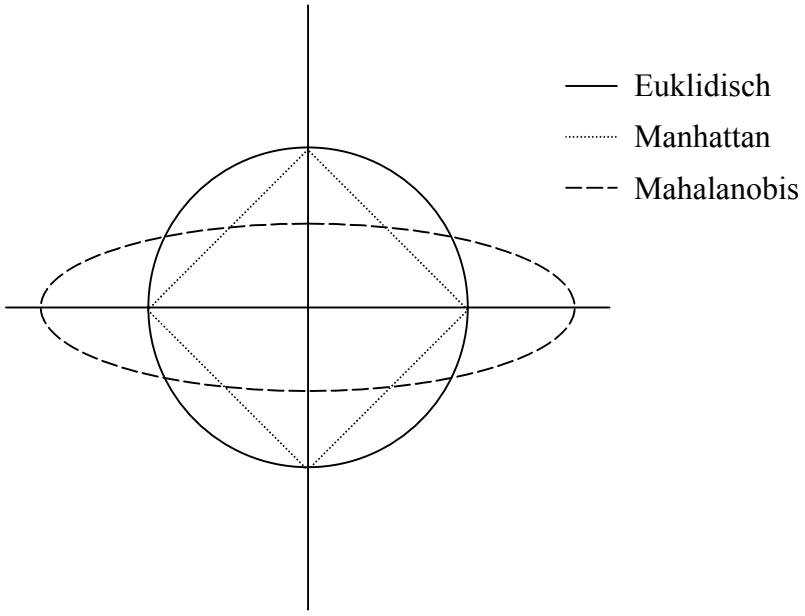


Abb. 2.3: Veranschaulichung der Punktmengen, die bei verschiedenen Metriken (Euklidischer, Manhattan- und Mahalanobis-Abstand) den Abstand eins zum Koordinatenursprung realisieren.

Der *Tanimoto-Koeffizient*[54] ist ein Maß für die Ähnlichkeit zwischen zwei Vektoren der Länge n. Der Tanimoto-Koeffizient s(**x**, **y**) zwischen zwei Vektoren **x** und **y** ist wie folgt definiert:

$$s(\mathbf{x},\mathbf{y}) = \frac{\sum_{i=1}^{n} x_i \cdot y_i}{\sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} y_i^2 - \sum_{i=1}^{n} x_i \cdot y_i}$$

Da der Tanimoto-Koeffizient der Dreiecksungleichung nicht genügt, ist er kein Distanzmaß. Dies hat zur Folge, dass „Clusterzentren“ im Tanimoto-Raum sehr schlecht definiert sind.

Ein Maß für die Unähnlichkeit (dissimilarity) ist die *Soergel-Distanz*[55], die als 1 − s(**x**, **y**) definiert ist.

Handelt es sich bei den Vektoren **x** und **y** um binäre Zeichenfolgen A und B, so gibt es für die Berechnung der Euklidischen Distanz d(A, B) und des Tanimotokoeffizienten s(A, B) vereinfachte äquivalente Rechenvorschriften:

$d(A,B) = \sqrt{a+b-2c}$ = A XOR B

$$s(A,B) = \frac{c}{a+b-c} = \frac{|A \cap B|}{|A \cup B|}$$

a Anzahl der Bits, die im Binärvektor A gesetzt sind
b Anzahl der Bits, die in Binärvektor B gesetzt sind
c Anzahl der Bits, die in beiden Fingerprints gemeinsam gesetzt sind.

Die Betragsstriche stehen für die Mächtigkeit (cardinality) der Schnittmenge (binäres AND) und der Vereinigungsmenge (binäres OR) der entsprechenden Binärvektoren A und B.

Die folgenden Eigenschaften des Tanimoto-Koeffizienten ermöglichen es, zu einer gegebenen Verbindung mit dem Fingerprint A sehr schnell und effizient den nächsten Nachbarn (mit maximalem s) in großen Datensätzen zu finden[56].

$$s(A,B) = \frac{\min(a,b)}{\max(a,b)}$$

$$a \cdot s_{ref} \le b \le \frac{a}{s_{ref}}$$

Beim Durchsuchen einer Liste von Strukturen wird s_{ref} jeweils auf den größten bislang aufgetretenen Wert gesetzt. Für alle Verbindungen, die unähnlicher als die gefundene Referenzverbindung zu A sind, deren b die Ungleichung also nicht erfüllt, kann auf die eigentliche Berechnung des Tanimoto-Koeffizienten gänzlich verzichtet werden.

Eine Reihe von Studien beschäftigt sich mit der Wahl der geeignetsten Metrik bei unterschiedlichen Anwendungen. Willet und Winterman[57] untersuchen die Leistungsfähigkeit unterschiedlicher Kombinationen von sechs Ähnlichkeitskoeffizienten mit zwei strukturellen Deskriptoren und sechs Gewichtungsschemata unter dem Aspekt einer biologischen Eigenschaftsvorhersage basierend auf dem Wert des nächsten Nachbarn. Brown und Martin[58] beschreiben den Einfluß auf verschiedene Clusterverfahren zur Strukturauswahl (compound selection), die auf der Berechnung der Similarity zwischen chemischen Strukturen basieren. Chen und Reynolds[59] beschreiben den Einfluß der Metrik auf die Effektivität von 2D-fragmentbasierten Ähnlichkeitssuchen von aktiven Strukturanaloga. Sie empfehlen, anstelle der bislang verfügbaren Suchverfahren in kommerziellen Programmen eine Kombination aus Atompfaden und des Tanimoto-Koeffizienten für Mengen zu verwenden.

Die Metriken können, wie Dixon et al.[60] vorgeschlagen haben, auch kombiniert verwendet werden. Bei der Selektion von diversen Subsets bioaktiver Verbindungen auf der Basis von 2D-Substruktur-Fingerprints hat die Wahl der Metrik einen Einfluß auf die Größe der ausgewählten Verbindungen. Das Komplement der Tanimoto-Metrik (1-Tanimoto) führt bevorzugt zu kleinen Verbindungen und die (quadrierte) Euklidische Distanz bevorzugt große Verbindungen. Bei der Verwendung eines Produkts aus beiden Metriken heben sich die gegenläufigen Effekte wieder auf und es kommt zu einer ausgeglichenen Größenzusammensetzung.

Die *Levenshtein-Distanz*[61], auch bekannt unter dem Namen Edit Distance, wird verwendet, um die Ähnlichkeit zweier Zeichenketten zu bewerten. Sie entspricht der minimalen Anzahl an Einfügungen, Löschungen und Ersetzungen, die erforderlich ist, um die eine Zeichenkette

in die andere zu überführen. Sie wird vorteilhafterweise unter Verwendung von dynamischer Programmierung berechnet[62]. Der Algorithmus ist im folgenden skizziert.

```
x, y := zu vergleichende Zeichenketten

m := |x|  Länge des Strings x
n := |y|  Länge des Strings y
cost := Matrix der Dimension m×n

FOR i := 1 TO m
     cost[i, 0] := i

FOR j := 1 TO n
      cost[0, j] := j

FOR i := 1 TO m
     FOR j := 1 TO n
          cost[i, j] := min( cost[i-1, j]+1, cost[i, j-1]+1,
                             cost[i-1, j-1]+δ(x[i], y[j]) )

RETURN cost[m, n]
```

Listing 2.1: Pseudocode zur Berechnung der Levenshtein-Distanz.

Der Levenshtein-Distanzwert d(x, y) entspricht dem Matrixwert cost[m, n], der die kleinstmögliche Summe der Werte durch die Matrix enthält. Die Funktion δ bewertet die Ungleichheit zweier Zeichen a und b, wobei $\delta(a,b)=0$ für $a=b$ und anderenfalls $\delta(a,b)=1$.

Um einen Ähnlichkeitswert im Bereich 0 und 1 zu erhalten, wird d(x, y) gemäß folgender Beziehung umskaliert:

$$s(x,y)=\max\left(0.0\,,\frac{|x|+|y|-2d(x,y)}{|x|+|y|}\right)$$

Das Verfahren von Needleman und Wunsch[63] zum paarweisen globalen Alignment von Sequenzen basiert auf dem gleichen Algorithmus. Allerdings werden dort in der Funktion δ Substitutionsmatrizen verwendet, die je nach Aminosäure- bzw. Nukleotidpaar unterschiedliche Werte verwenden.

Auch die Ähnlichkeit von MolCodes kann unter Verwendung der Levenshteindistanz berechnet werden.

2.5 Techniken zur Datengruppierung (Clusterverfahren)

Die Strukturmerkmale und die geschilderten „chemischen“ Distanzen (basierend auf Deskriptoren und Ähnlichkeitsmaßen) werden in der Praxis auf unterschiedliche Weise zur Datengruppierung und Datenverdichtung in umfangreichen Substanzdatenbanken herangezogen[64]. Dabei wird aufgrund einer allgemeinen Ähnlichkeitsbeziehung eine Cluster-Bildung für verwandte Verbindungen angestrebt. Dies bedeutet, dass für zwei Verbindungen ein berechneter Ähnlichkeitsindex (vide supra) einen bestimmten Schwellenwert (in der Regel

größer oder gleich 80 %) überschreitet, die chemische Distanz zwischen den Verbindungen einen kritischen Wert nicht übersteigt oder die Verbindungen in allen Deskriptorwerten einen gemeinsamen engen Deskriptor-Raumbereich abdecken. Damit soll erreicht werden, dass strukturell verwandte Verbindungen oder zumindest solche, die ein gemeinsames Eigenschaftsprofil haben, zusammengruppiert werden, um unterschiedliche Leitstrukturen über ihre besten repräsentativen Vertreter und ihre Eigenschaften besser vergleichen und bewerten zu können. Die praktische Durchführung der Clusteranalyse kann entweder interaktiv durch Inspektion der grafisch visualisierten Eigenschaftsräume oder aber numerisch nach geeigneten Algorithmen durchgeführt werden.

Clustern ist eine Technik zur Gruppierung von Datenpunkten, so dass der Grad der Ähnlichkeit zwischen den Datenpunkten innerhalb einer Gruppe hoch ist und zwischen Datenpunkten verschiedener Gruppen gering ist. Es gibt keine allgemein akzeptierte Definition eines Clusters. Clustern wird angewendet, um große Mengen von multidimensionalen Daten zu vereinfachen, zu interpretieren und zu verstehen[65]. In Abb. 2.4 ist eine allgemeine Hierarchie der verschiedenen Clusterverfahren aufgeführt[66].

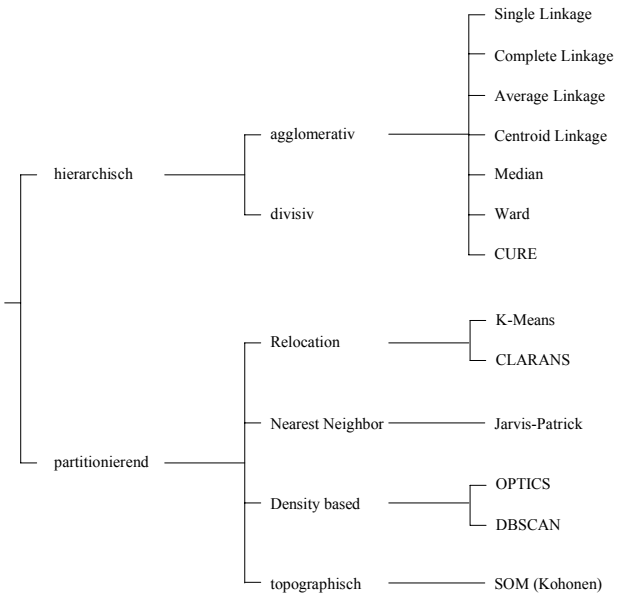


Abb. 2.4: Hierarchie der im folgenden aufgeführten Clusterverfahren.

Hierarchische Verfahren erstellen eine Hierarchie der Datenobjekte, die in einem Dendrogramm dargestellt werden kann. Am unteren Ende sind alle Datenobjekte in einem eigenen Cluster dargestellt und am oberen Ende bildet der gesamte Datensatz einen einzelnen Cluster. Durch Wahl eines horizontalen Schnitts im Dendrogram wird die Anzahl der Cluster ausgewählt. Wild et al.[67] haben verschiedene Verfahren zur Level-Selektion verglichen. Das am häufigsten eingesetzte ist das Verfahren nach Kelley[68]. Es wird für jeden Level einer Hierarchie ein Penalty-Wert berechnet, der die mittlere Verteilung in nicht-singleton Clustern und die Gesamtzahl an Clustern berücksichtigt. Das Minimum im Kelley plot, der Auftragung des Penalty-Werts gegen die Clusteranzahl, ergibt die optimale Anzahl an Clustern.

In agglomerativen Verfahren werden die Cluster der vorhergehenden Ebene der Reihe nach verknüpft, das Dendrogramm wird quasi von unten nach oben aufgebaut. In divisiven Verfahren wird der Gesamtdatensatz so lange zerteilt, bis jeder Cluster nur aus einem Objekt besteht oder eine vorgegebene Anzahl von Clustern erreicht ist, das Dendrogramm also von oben nach unten erstellt[69].

Nichthierarchische Verfahren erzeugen eine vorgegebene Anzahl von Clustern, ohne sie in Beziehung zueinander zu setzen. Die Anzahl der Cluster kann, je nach Verfahren, direkt vorgegeben werden oder indirekt über einen Ähnlichkeitsschwellenwert erreicht werden.

Breite Anwendung in der Chemoinformatik finden in den Implementierungen von Tripos[70], Daylight[71] und BCI[72] v.a. folgende Verfahren:

- Wards
- Jarvis-Patrick
- k-Means

Zur Verbesserung des Laufzeitverhaltens, das vor allem durch das wiederholte Suchen des jeweils nächsten Nachbarn bestimmt wird, können RNN (Reciprocal Nearest Neighbours)[73, 74] und k-d-Trees[75, 76] verwendet werden. Im ersten Fall werden sortierte Abstandslisten zu drei Referenzobjekten verwaltet, in denen so lange ein sequentieller Lookup erfolgt, bis zwei Objekte gefunden sind, die zueinander („reciprocal“) die nächsten Nachbarn sind. Im zweiten Fall wird ein mehrdimensionaler binärer Suchbaum durch wiederholte Unterteilung des durch die Deskriptoren aufgespannten Raums erzeugt. Bei einer geringen Anzahl an Dimensionen (bis 10) ist der dadurch erzeugte Overhead vernachlässigbar. Die Suche nach dem NN erfolgt dann durch eine Traversierung des Baums von oben nach unten, bis zusammengehörige Datenpunkte gefunden sind. In beiden Fällen ist die Anzahl an erforderlichen Distanzberechnungen deutlich vermindert.

Wards[77]

1. Berechne und Speichere die Abstandsmatrix **D**. Sie enthält die Distanz zwischen jedem Paar von Datenpunkten.
2. Suche in der Abstandmatrix das kleinste Element d_{ij} mit $i \neq j$.
3. Gruppiere die Punkte i und j zu einem einzigen neuen Datenobjekt zusammen.
4. Aktualisiere die Abstände (vide infra) des neuen Datenobjekts zu allen anderen.
5. Gehe (n-1 mal) zu Schritt 2. bis ein einzelnes Datenobjekt übrig bleibt.

Bei der Fusionierung zweier Cluster A und B, d.h. im wesentlichen der Zuordnung der gleichen Clusternummer, bestehend aus n_a bzw n_b Einzelobjekten, wird der neue Cluster [AB] mit $n_a + n_b$ Einzelobjekten gebildet. Der Abstand eines nicht an der Fusionierung beteiligten Clusters C mit n_c Einzelobjekten zum neuen Cluster [AB] wird aus den bekannten Abständen *d*(A, C), *d*(B, C) und *d*(A, B) nach folgender Formel hergeleitet:

$$d([AB],C) = \frac{(n_c + n_a)\cdot d(A,C) + (n_c + n_b)\cdot d(B,C) - n_c \cdot d(A,B)}{n_a + n_b + n_c}$$

Alle klassischen Verfahren zur hierarchischen Clusterung basieren auf dem gleichen Algorithmus, die Aktualisierung des Abstands zwischen zwei Clustern erfolgt allerdings nach anderen Rechenvorschriften[78].

Methode	$d([AB],C)$
Single Linkage	min [d(A, C), d(B, C)]
Complete Linkage	max [d(A, C), d(B, C)]
Average Linkage	[d(A, C) + d(B, C)]/2

Das Verfahren skaliert mit $O(N^3)$ und ist daher nicht für sehr große Datensätze geeignet. Unter Verwendung von Murtaghs Reciprocal Nearest Neighbors skaliert das Verfahren mit $O(N^2)$.

Jarvis-Patrick[79]

Im Jarvis-Patrick Verfahren gibt es zwei Eingabeparameter: K bezeichnet die Länge der Liste mit den nächsten Nachbarn (NN-Liste) und J die Anzahl der Datenpunkte, die für eine Zusammengruppierung gemeinsam in den NN-Listen vorhanden sein müssen.

1. Erstelle zu jedem Datenobjekt die NN-Liste mit der Länge K.
2. Gruppiere die Datenpunkte i und j zusammen, wenn
   a) i in der NN-Liste von j, und
   b) j in der NN-Liste von i, und
   c) NN-Liste von i und NN-Liste von j mindestens J gleiche Nachbarn enthalten.
3. Gehe zu 2 für alle (i, j) mit i < j.

Da die Anzahl der Cluster nicht spezifiziert werden kann und das Verfahren meist wenige große Cluster und viele Singletons erzeugt, gibt es einige Verfahren, das Ergebnis zu verbessern. Menard et al. beschreiben ein kaskadiertes Clustering, in dem Singeltons erneut geclustert werden[80].

Das Verfahren hat eine Laufzeitordnung von $O(N^2)$.

k-Means [81, 82]

k-Means gehört zum Typ der Relocation-Verfahren. Als Eingabeparameter ist die Anzahl der Cluster erforderlich. Das Verfahren hat eine Laufzeitordnung von O(N).

1. Selektiere k Datenpunkte als Prototypen.
2. Ordne alle Datenpunkte dem nächsten Prototypen zu.
3. Berechne die Clusterzentroide.
4. Ordne alle Datenpunkte dem nächsten Clusterzentroid zu.
5. Gehe zu 3, bis keine Veränderungen der Cluster mehr erfolgen.

In der k-Medoid-Variante wird als Cluster-Zentrum das nächste tatsächlich im Datensatz vorhandene Objekt verwendet.

Die Auswahl der Prototypen in Schritt 1 hat wesentlichen Einfluß auf das Ergebnis. Das Growing K-means-Verfahren ist diesbezüglich verbessert[83]. Ausgehend von nur zwei zufällig gewählten Prototypen werden alle weiteren Clusterzentren iterativ zwischen dem Zentrum des größten Clusters und dem am weitesten entfernten Datenpunkt ergänzt.

Unsupervised nonhierarchical clustering algorithm[84]

Taylor und Butina[85] beschreiben ein Verfahren zum nichthierarchischen Clustern von Strukturen. Die Strukturen werden durch Fingerprints repräsentiert. Der einzige Eingabeparameter ist ein Grenzwert (meist 0,85), bis zu dem zwei Strukturen unter Verwendung des Tanimoto-Koeffizienten als ähnlich betrachtet werden. Dies ist insbesondere für exploratorische Clusterungen von Vorteil, wenn die Anzahl der Cluster nicht a priori bekannt ist. Der Algorithmus ist im folgenden Flussdiagramm beschrieben. Der Autor hat eine Implementierung in Perl[86] vorgenommen.

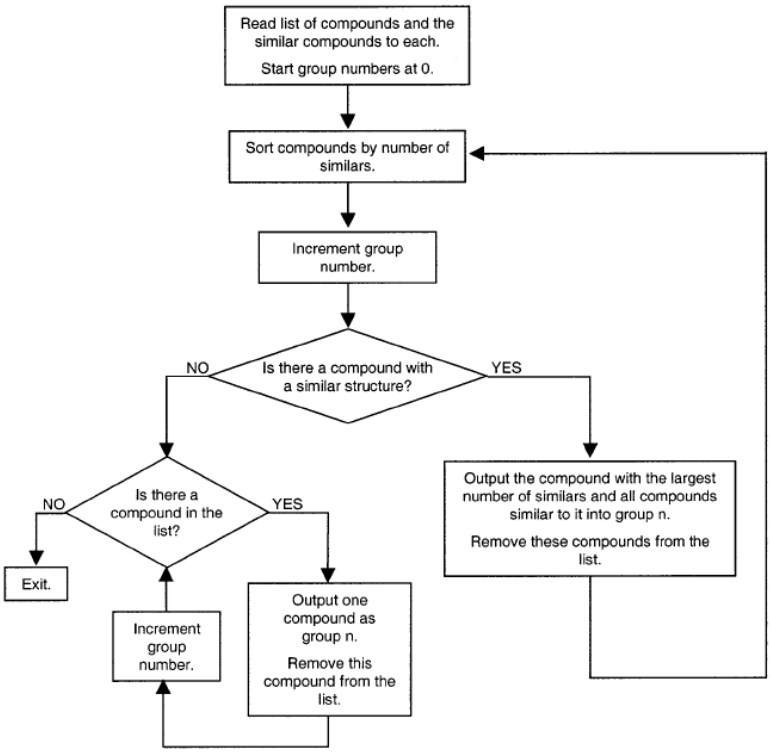


Abb. 2.5: Flowchart zum unsupervised nonhierarchical clustering algorithm[87].

Kohonen-Netzwerk[88]

Das Kohonen-Netzwerk (SOM, Self Organising Map) ist im Kern ein Relocation-Verfahren mit der Zusatzbedingung, dass die Clusterzentren auf einem regulären Gitter oder einer anderen topographischen Struktur angeordnet sind. Die Positionierung der Datenpunkte in diesem Gitter erfolgt im Rahmen des nichtüberwachten (unsupervised) Trainings eines künstlichen neuronalen Netzwerks und erfordert damit die Festlegung einer großen Anzahl von Zusatzparametern (Gittertopographie, Gittergröße, Lernfunktion, Lernrate, Nachbarschaftsfunktion, Nachbarschaftsradius, Anzahl der Lernschritte)[89, 90]. Die erzeugte SOM enthält durch nichtlineare Projektion (Dimensionsreduktion) auf 2D die Nachbarschaftbeziehung der Cluster zueinander und kann direkt zur Visualisierung in einem GUI verwendet werden[91, 92].

Von den vielen existierenden Clusterverfahren sind noch folgende erwähnenswert:

- DBScan/OPTICS[93, 94]
- CURE[95]

Bei der Verwendung von Clusterverfahren sind folgende Entscheidungen zu treffen:

- Wahl des Verfahrens, z.B. Wards, Jarvis-Patrick, k-Means
- Wahl der Deskriptoren, z.B. Fingerprints
- Wahl der Standardisierung (wenn erforderlich), z.B. Z-Standardisierung jeder Variablen auf einen Mittelwert von 0 und eine Standardabweichung von 1
- Wahl der Metrik, z.B. Tanimoto, Euklidisch
- Auswahl weiterer Parameter, mindestens die gewünschte Anzahl an Clustern

Das Limitierende an dieser Methode ist nicht so sehr die Nichtverfügbarkeit von guten Cluster-Algorithmen als vielmehr die Definition eines geeigneten Chemieraumes, in dem die Verbindungen in der gewünschten Art und Weise positioniert sind, so dass eine chemisch akzeptierte Clusterung grundsätzlich möglich ist.

Nachteile der Clusterverfahren:

- keine absolute strukturchemische Charakterisierung der Cluster bzw. der Datenbank
- ungünstiges Laufzeitverhalten (worst case: Paarvergleiche)
- uneinheitliche Clusterung bei Veränderungen im Datenbestand
- optimale Anzahl der Cluster a priori nicht bekannt
- für Anwender schwer einsichtiges und nachvollziehbares Ergebnis
- problematischer Umgang mit Singletons bzw. chemischen Prototypen
- Veränderung des Ergebnisses durch die Integration unterschiedlicher Datenquellen
- schwierige Beurteilung des abgedeckten Chemieraums und seiner Lücken

2.6 Anwendungsgebiete für Strukturvergleiche

Verfahren zum computerunterstützten Vergleich von Strukturen kommen zum Einsatz bei der

- Auswertung biologischer Screeningergebnisse (Clusteranalyse)
- Leitstrukturoptimierung (SAR, Eigenschaftsprofile, Selektivität)
- Selektion (Leitstrukturauswahl) und Ranking ähnlicher Verbindungen (virtuelles screening)
- Auswahl strukturell unterschiedlicher bzw. chemisch diverser Verbindungen (library design)
- linearen Anordnung für die visuelle Analyse (Intervall-Suche)
- Diversitätsanalyse und -optimierung von Substanzpools einschließlich Lückenanalyse (compound pool shaping)

Bei der Ähnlichkeitssuche (similarity search) oder Substruktursuche (substructure search) werden zu einer gegebenen Verbindung weitere ähnliche bzw. andere Derivate gesucht. Soll aus einer größeren Anzahl von Verbindungen eine strukturell bzw. chemisch diverse Untermenge ausgewählt werden (subset selection), so werden zusätzlich Selektionsalgorithmen benötigt. Bei der Intervall-Suche werden die zu ordnenden Strukturen sequentiell angeordnet, damit sie leichter zu finden sind.

Zum computergestützten 2D-Vergleich großer Mengen von Verbindungen gibt es prinzipiell drei Möglichkeiten:

- in einem höherdimensionalen Deskriptor- bzw. Eigenschaftsraum
- in einem Ähnlichkeitsraum („chemische“ Distanzen)
- topologisch auf der Basis (maximaler) gemeinsamer Substrukturen

Das Fingerprint-Verfahren mit Tanimoto-Metrik ist als Hybridverfahren anzusehen. Es liefert einen Ähnlichkeitskoeffizienten basierend auf der Anzahl gemeinsamer Substrukturfragmente der Verbindungen. Die Variante Atompairs bzw. Topological Torsions mit (Set-) Tanimoto-Metrik wird seltener verwendet. Dennoch ist die Verwendung strukturnaher Deskriptoren wie Fragmente, Atom Pairs oder Topological Torsions vorteilhafter, da sie als Teil der Struktur augenfällig sind, und unmittelbar vom Anwender interpretiert werden können.

Nach der Transformation der Strukturen in geeignete Deskriptoren ist das Problem zu lösen, welche der zahllosen berechneten Deskriptoren die geeignetsten sind und welcher Zusammenhang zwischen diesen Deskriptoren und der gewünschten Wirkung besteht. Diese Vorgehensweise ist rechnerisch leicht umzusetzen (Variablenselektions- und Regressionsverfahren), führt aber oft zu chemisch oder mechanistisch bedeutungslosen Zufallskorrelationen. Leider geht dabei oft der Anwendungsbezug – die Auswahl des "besten" Templates und die Umsetzung in neue wirksamere Synthesevorschläge – verloren bzw. wird erschwert. Deshalb ist man gezwungen, die strukturchemischen Aspekte der Wirkung neuer Liganden über den Umweg der Eigenschaftsanalyse in virtuellen Chemieräumen zu bearbeiten.

Im Interesse der Anwender und der zu lösenden praktischen Probleme der Wirkstofffindung sollten die zum Strukturvergleich verwendeten Verfahren

- plausibel, verständlich und chemisch (also mit Struktur- und Funktionalitätsbezug) interpretierbar sein
- der modularen Denkweise des synthetischen Chemikers bei der Synthese der Verbindungen und der Beschreibung der Wirkhypothese entgegenkommen
- trotz umfangreichen Datenmaterials sowohl eine praktikable strukturelle Übersicht (topologischen Anordnung und Belegungsdichte) über alle Strukturprototypen als auch eine Bewertung ihrer Molekülfragmente bezüglich der beobachteten Targetwirkung ermöglichen.
- eine standardisierte Form der Analyse und der Ergebnisaufbereitung erlauben, um unterschiedliche Datenquellen bei der Bearbeitung verwenden zu können
- gegenüber inhaltlichen und mengenmäßigen Veränderungen im Datenbestand robust sein, da ständig neue Substanzen synthetisiert werden
- für die Lückenanalyse (Identifizierung „fehlender“ Verbindungen) geeignet sein

3 Vorhandene Systeme und kommerzielle Software

In der Praxis werden oft verschiedene Verfahren kombiniert, die sich auf zentrale Techniken wie interaktive, höherdimensionale Datenvisualisierung, Nutzung eines chemisch orientierten Spreadsheets oder Berechnung von Eigenschaftsclustern mit Deskriptoren stützen.

3.1 Molecular Spreadsheet

Molecular Spreadsheets sind erweiterte Tabellenkalkulationen, mit denen man Molekülstrukturen anzeigen kann. Kommerziell erhältliche Produkte für Microsoft Excel sind z.B. ISIS for Excel[96] oder Accord for Excel[97]. Die Tabellen haben üblicherweise eine Zeile pro Verbindung und eine Spalte mit einem Strukturdiagramm. Die anderen Spalten enthalten einen Identifizierungscode (Registrierungsnummer) der Verbindung, biologische Daten und experimentell gemessene und berechnete Daten. Der medizinische Chemiker hat Zugriff auf verschiedene Firmen-Datenbanken und Projektfiles, aus denen er die gewünschten Informationen in das Spreadsheet lädt. Wenn die chemischen Strukturen, die biologische Aktivität und zusätzlich gewünschten Daten geladen sind, sortiert er die Zeilen der Tabelle üblicherweise nach der Aktivitätsspalte, um die aktivsten Verbindungen als erste angezeigt zu bekommen. Anschließend betrachtet er der Reihe nach die Strukturen mit fallender Aktivität. Bei diesem ersten Überblick werden gelegentlich neue Spalten manuell hinzugefügt, die zusätzliche Bewertungen für die Strukturen enthalten, z.B. sehr interessant (2), interessant (1), uninteressant (0), oder es wird manuell eine grobe Gruppierung in verschiedene Strukturtypen vorgenommen. Nachdem der medizinische Chemiker auf diese Weise 50-100 Strukturen betrachtet hat, bemerkt er wahrscheinlich einige häufiger auftretende Substrukturen. Das veranlasst ihn in der Regel zu einer ersten Hypothese, dass diese teilweise für die Aktivität der Verbindungen verantwortlich sein könnten.

Zur Untermauerung der Hypothese wird eine Substrukturrecherche in der HTS- bzw. Firmen-Datenbank vorgenommen und die Anzahl der aktiven und inaktiven Verbindungen (im weiteren kurz: Aktive und Inaktive) ermittelt. Da die Inaktiven nicht berücksichtigt werden, bleiben mehr als 90% der im HTS erzeugten Informationen unausgewertet. Statistische Testverfahren, die die Aktivität einer chemischen Familie, d.h. Verbindungen, die dieses spezielle strukturelle Element enthalten, mit der Aktivität des gesamten Datensatzes vergleichen, werden nur selten herangezogen. Mögliche Testverfahren[98] sind unter anderem Kolmogorov-Smirnov und χ^2 [99].

3.2 Deskriptorbasierte Clusterung von Strukturen

Zur Visualisierung der Ergebnisse von Strukturclusterungen werden im einfachsten Fall die Zeilen eines Spreadsheets entsprechend sortiert, die sequentiell von Anfang bis Ende durchgesehen werden. Eine komfortablere Auswertung ist mittels für diesen Zweck entwickelter Clusterviewer von Xemistry[100] oder Daylight[101] möglich. Abb. 3.1 enthält einen exemplarischen Screenshot des Xemistry Clusterviewer. Im linken Teil ist zu jedem Cluster die Größe und ein wählbarer Repräsentant aufgelistet. Dieser kann zufällig ausgewählt oder der größte oder kleinste sein. Im rechten Teil sind alle Mitglieder des selektierten Clusters

eingezeichnet. Dies ermöglicht dem Anwender, schnell einen Überblick über die vorhandenen Cluster und deren Zusammensetzung zu bekommen.

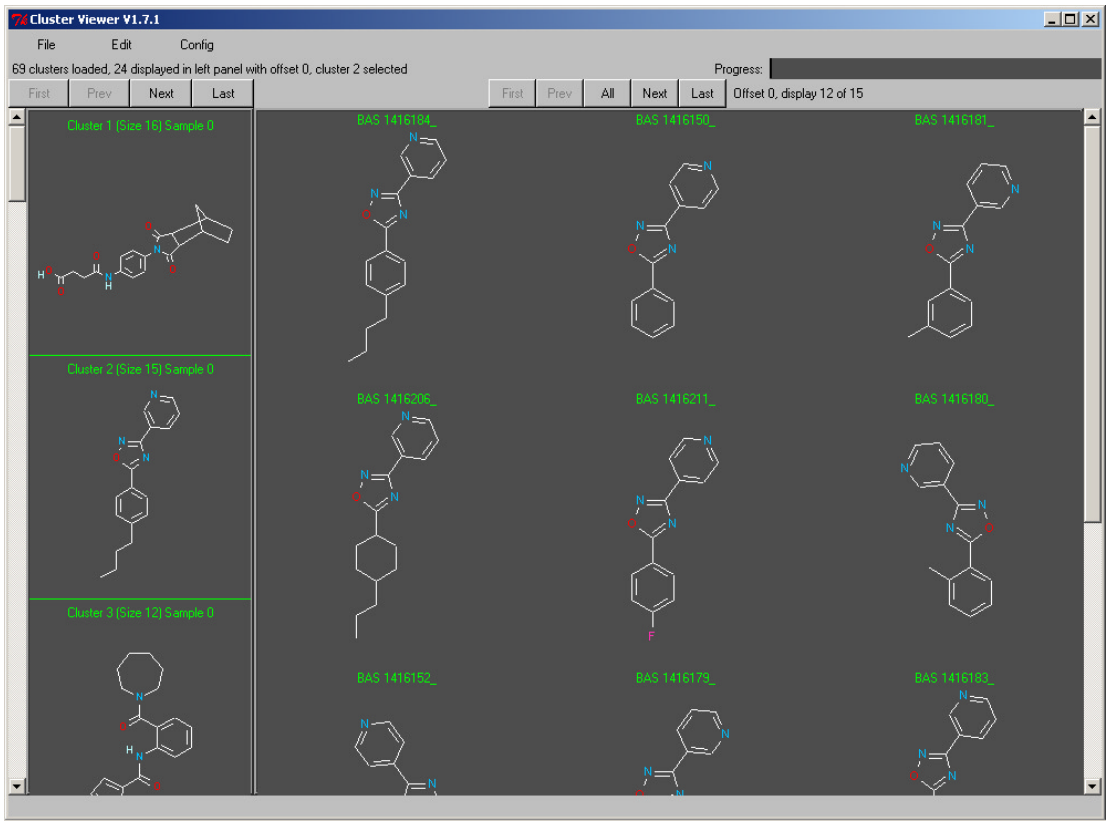


Abb. 3.1: Screenshot des Xemistry Clusterviewer.

3.3 Spotfire

Spotfire.net (früher Spotfire Pro), vertrieben von Spotfire Inc.[102], ist das Referenzprogramm zur visuellen interaktiven grafischen Analyse großer Datenmengen[103]. Um Zusammenhänge und Trends in Daten erkennen zu können, stehen verschiedene Diagrammtypen wie 2D/3D-Scatter Plots, Pie Charts und Histogramme zur Verfügung, in denen die Datenpunkte zusätzliche Attribute in Form, Größe und Farbe codiert enthalten. Die Auswahl der Achsenvariablen erfolgt aus Listenfeldern direkt in den Diagrammen; sie können daher interaktiv schnell verändert werden, um verschiedene Auftragungen auszuprobieren. Die Skalierung geschieht automatisch und kann durch den Achsen zugeordnete range sliders jederzeit verändert werden Der exemplarische Screenshot in Abb. 3.2 enthält im linken und mittleren Teil zwei Scatterplots. Am linken und unteren Rand der Diagramme sind jeweils die ausgewählten Variablen und die range sliders der Achsen erkennbar.

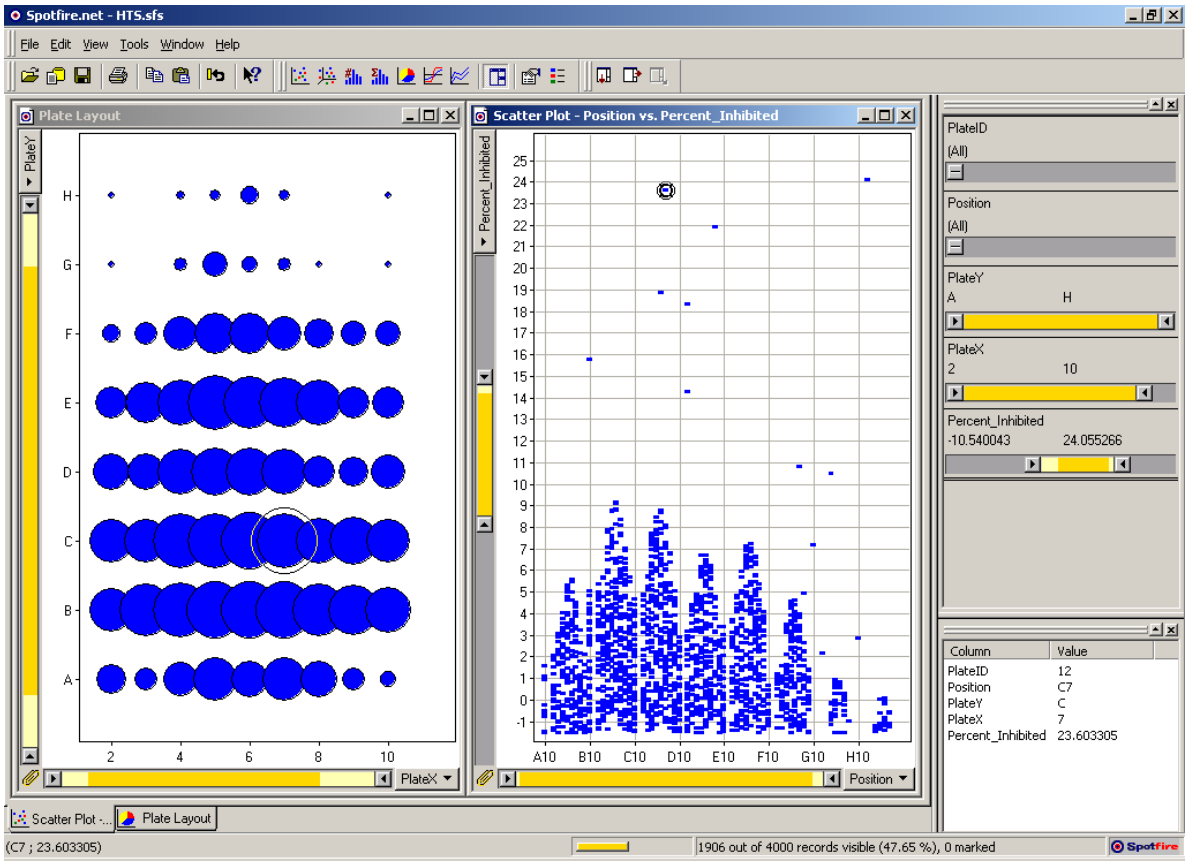


Abb. 3.2: Screenshot des Spotfire GUI.

Der range slider[104] ist eine Neuentwicklung des Spotfire-Erfinders Ch. Ahlberg (Chalmers University) und von Ben Shneiderman (Human Computer Interaction Laboratory, University of Maryland). Dabei handelt es sich um eine Kombination aus einem Rollbalken (scrollbar) mit zwei Scale-Widgets, mit dem zusätzlich sowohl eine Ober- als auch eine Untergrenze eingestellt werden kann. Durch Verschieben des mittleren Bereichs werden Ober- und Untergrenze gleichzeitig verändert, wobei das eingestellte Intervall unverändert bleibt. Dieses Prinzip ist inzwischen mit kleineren Abwandlungen in verschiedenen Programmen adaptiert worden. Einige experimentelle Implementierungen des Widgets stehen für Programmierer zur Verfügung[105, 106].

Spotfire ermöglicht die simultane visuelle Auswertung großer Datenmengen aus verschiedenen Datenquellen. Sind dem Programm alle Daten zu einem Projekt bekannt, werden diese zu Beginn einer Analyse bezüglich des Datentyps und des Wertebereichs unterzogen. Mit diesen Informationen wird jeder Variable ein geeigneter Filter (query device) zugeordnet. Zusätzlich werden Korrelationen der vorhandenen Attribute ermittelt und dadurch die Reihenfolge der „Wichtigkeiten“ bestimmt, die die Reihenfolge der query devices und Variablen in Listenfeldern beeinflusst. Das beim Start des Programms erzeugte Diagramm basiert ebenfalls auf diesen Informationen.

Für die Filter (query devices) gibt es folgende Varianten: die oben angesprochenen „range sliders“ für kontinuierliche Wertebereiche, „item sliders“ für diskontinuierliche Einzelwerte, „full text search“-Felder für Texte und „check boxes“ bzw. „radio buttons“ für Binärwerte oder Variable mit categorial data. Einige davon sind im rechten Teil der Abb. 3.2 abgebildet. Mit Hilfe dieser Filter können dynamische Datenbankabfragen erstellt werden. In den

Diagrammen werden nur die Datenpunkte angezeigt, die allen Filterkriterien entsprechen. Die erstellte Abfrage kann als SQL-Kommando exportiert werden.

Das Besondere an Spotfire ist, dass jeder Schritt bei der Änderung einer Abfrage kontinuierlich und simultan (< 150ms) in allen Visualisierungen angezeigt wird. Das unterstützt den Benutzer bei der interaktiven Formulierung geeigneter Datenbankabfragen, die interessante neue Zusammenhänge in den Daten aufzeigen.

Durch die brushing and linking-Technik werden in einem Diagramm selektierte Datenpunkte ebenfalls in allen anderen Diagrammen selektiert. Das selektierte Objekt in Abb. 3.2 ist in beiden Diagrammen durch einen Kreis gekennzeichnet. Dadurch können beispielsweise in vier Diagrammen bis zu 12 Dimensionen gleichzeitig überblickt werden.

Insgesamt unterstützt das Programm Spotfire das visuelle Datamining durch die folgenden drei Besonderheiten:

- Datenintegration (Einbinden mehrerer Datenbanken)
- Interaktive Formulierung von Abfragen durch den Benutzer
- Echtzeitaktualisierung aller sichtbaren Diagramme

Für chemische Fragestellungen gibt es die Möglichkeit, eine Strukturdatenbank einzubinden und von den Molekülen Strukturabbildungen zu erzeugen[107].

Spotfire gibt dem Benutzer bei der interaktiven Auswertung großer Datenmengen nahezu uneingeschränkte Freiheiten, liefert aber keine strategisch orientierte, standardisierte automatische Auswertung oder Vorgehensweise bei der Analyse. Eine Unterstützung zur einheitlichen Projektbearbeitung sowie eine Bewertung der Qualität oder der Erfolgsaussichten unterschiedlicher Auswertungsarten relativ zueinander ist nicht vorgesehen.

3.4 SCA: Scaffold-based Classification Approach

Unter Verwendung der Spotfire-Plattform hat Jun Xu von Boehringer Ingelheim Pharmaceuticals ein Verfahren mit dem Namen SCA vorgestellt. Die Abkürzung stand ursprünglich für scaffold-based clustering algorithm[108], neuerdings für scaffold-based classification approach[109]. Es handelt sich m. E. dennoch eher um ein Verfahren zur Clusterung als zur Klassifizierung. Basierend auf zwei Abstandskoeffizienten werden ohne zusätzliche Parameter natürliche strukturelle Familien identifiziert. Zu deren Berechnung werden zu jeder Struktur zwei Feature-Vektoren $\mathbf{V}_i$ und $\mathbf{S}_i$ definiert. Der Vektor $\mathbf{V}_i$ des Scaffolds (das Grundgerüst ohne Seitenketten und Substituenten) der Struktur i ohne Berücksichtigung der Wasserstoffatome besteht aus:

- Anzahl der Ringe (Mächtigkeit der Ringbasis)
- Anzahl der Atome
- Anzahl der Bindungen
- Summe der Ordnungszahlen der Atome

Der Vektor $\mathbf{S}_i$ enthält die folgenden strukturellen Deskriptoren:

- Summe der Ordnungszahlen der Nichtwasserstoffatome
- Anzahl der rotierbaren Bindungen

- Anzahl der Atome mit der Bindungsordnung 1
- Anzahl der Doppelbindungen
- Anzahl der Dreifachbindungen
- Anzahl der Atome mit der Bindungsordnung 2

$\mathbf{V}_i$ dient zur Berechnung der Complexity als Abstandskoeffizient des Scaffolds vom Referenzvektor $\mathbf{V}$, der die Maximalwerte der vier Komponenten enthält:

$$\text{Complexity(i)} = \frac{\|\mathbf{V}_i + \mathbf{V}\| - \|\mathbf{V}_i - \mathbf{V}\|}{\|\mathbf{V}_i + \mathbf{V}\|}$$

$\mathbf{S}_i$ wird zur Berechnung der Cyclicity als Abstandskoeffizient der Struktur i vom zugehörigen Scaffold verwendet.

$$\text{Cyclicity(i)} = \frac{\|\mathbf{S}_i + \mathbf{S}'_i\| - \|\mathbf{S}_i - \mathbf{S}'_i\|}{\|\mathbf{S}_i + \mathbf{S}'_i\|}$$

Der Featurevektor $\mathbf{S}'_i$ enthält die gleichen strukturellen Deskriptoren wie $\mathbf{S}_i$, allerdings für das der Struktur i zugehörige Scaffold.

Durch Auftragung von Cyclicity gegen Complexity in Spotfire können SCA-Karten (Abb. 3.3) verschiedener Datenbanken zur Visualisierung und zum Vergleich erzeugt werden.

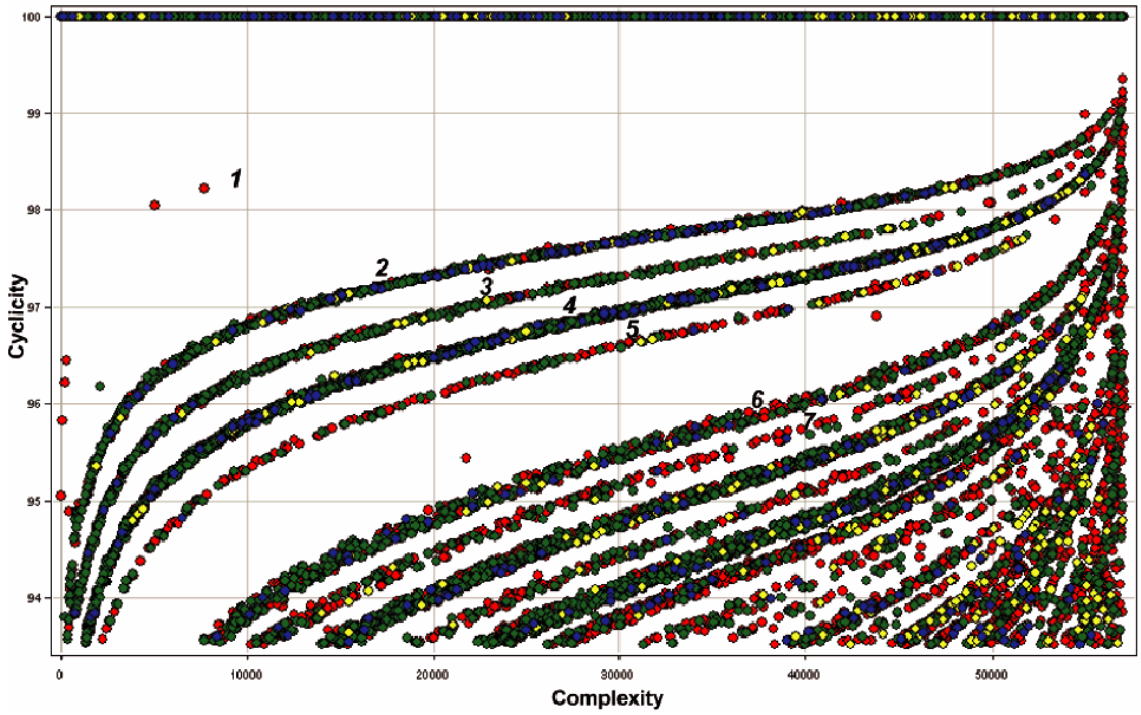


Abb. 3.3: Die SCA-Karte der Datenbanken ACD (rot), NCI (grün), CMC (blau) und MDDR (gelb).

Der SCA-Kartenausschnitt enthält Strukturen der ACD (rot), NCI (grün), CMC (blau) und MDDR (gelb) Datenbanken. Die Ecken der Karte repräsentieren vier Klassen von Verbindungen:

- Die linke obere Ecke enthält Verbindungen mit einfachen Ringen und wenigen Seitenketten.
- Die linke untere Ecke enthält Verbindungen mit einfachen Ringen und längeren oder komplizierteren Seitenketten.
- Die rechte obere Ecke enthält Verbindungen mit komplizierten Ringsystemen und weniger komplizierten Seitenketten.
- Die rechte untere Ecke enthält Verbindungen mit komplizierten Ringsystemen und komplizierten Seitenketten.

Die rechte untere Ecke ist in der Regel nicht populiert, da derartig komplexe Verbindungen schwer zu synthetisieren sind. Die Parallele zur Abszisse im Diagramm (cyclicity 100%) steht für reine Ringsysteme ohne Seitenketten. Jede Kurve (Diversitätsmuster) repräsentiert Verbindungen mit mindestens einer gemeinsamen Seitenkette, aber mit verschiedenen Scaffolds. Die in der Abbildung mit Zahlen als Beispiel gekennzeichneten Kurven enthalten folgende Verbindungen:

Kennzeichung	Diversitätsmuster
1	Verbindungen mit Lithium
2	Verbindungen mit einer Methylgruppe
3	Verbindungen mit einer primären Aminogruppe
4	Verbindungen mit einer Carbonyl- oder Hydroxygruppe
5	Verbindungen mit einem Fluor-Substituenten
6	Verbindungen mit zwei Methylgruppen oder einer Ethylgruppe
7	Verbindungen mit einer Methyl und einer primären Aminogruppe

3.5 Distill

Distill (früher Charisma)[110] ist ein Programm, das Verbindungen über die größte gemeinsame Substruktur (Maximum Common Substructure, MCS)[111] gruppiert. Es erstellt daraus ein hierarchisches Dendrogramm mit verschiedenen Levels. Gibt es keine übergeordnete gemeinsame Substruktur, werden die Strukturen verschiedener Knoten über den sog. MCS-Score zusammengefasst und die identische Struktur mehrfach eingetragen.

Die MCS-Scoringfunktion enthält drei Gewichtungsfaktoren für Ringbindungen, Heteroatome und Verzweigungen (Atome mit der Bindungordnung drei oder größer). Dadurch kann gesteuert werden, wo der Schwerpunkt bei der MCS-Scoreberechnung liegen soll.

$$MCS\_Score = n_{Atome} + n_{Bindungen} + w_{Ringbindungen} * n_{Ringbindungen} + w_{Heteroatome} * n_{Heteroatome} + w_{Verzweigungen} * n_{Verzweigungen}$$

Die vom Programm vorgegebenen Defaultwerte sind: $w_{Ringbindungen} = 3$, $w_{Heteroatome} = 4$, $w_{Verzweigungen} = 2$

Die Auswahl des MCS kann durch verschiedenen Parameter beeinflusst werden. Beispielsweise kann die Unterscheidung bezüglich des Atomtyps, des Bindungstyps oder der Ring- bzw. Nicht-Ring-Zugehörigkeit wahlweise deaktiviert und damit bei der Berechnung ignoriert werden.

Details zu dem verwendeten Algorithmus sind bislang nicht publiziert.

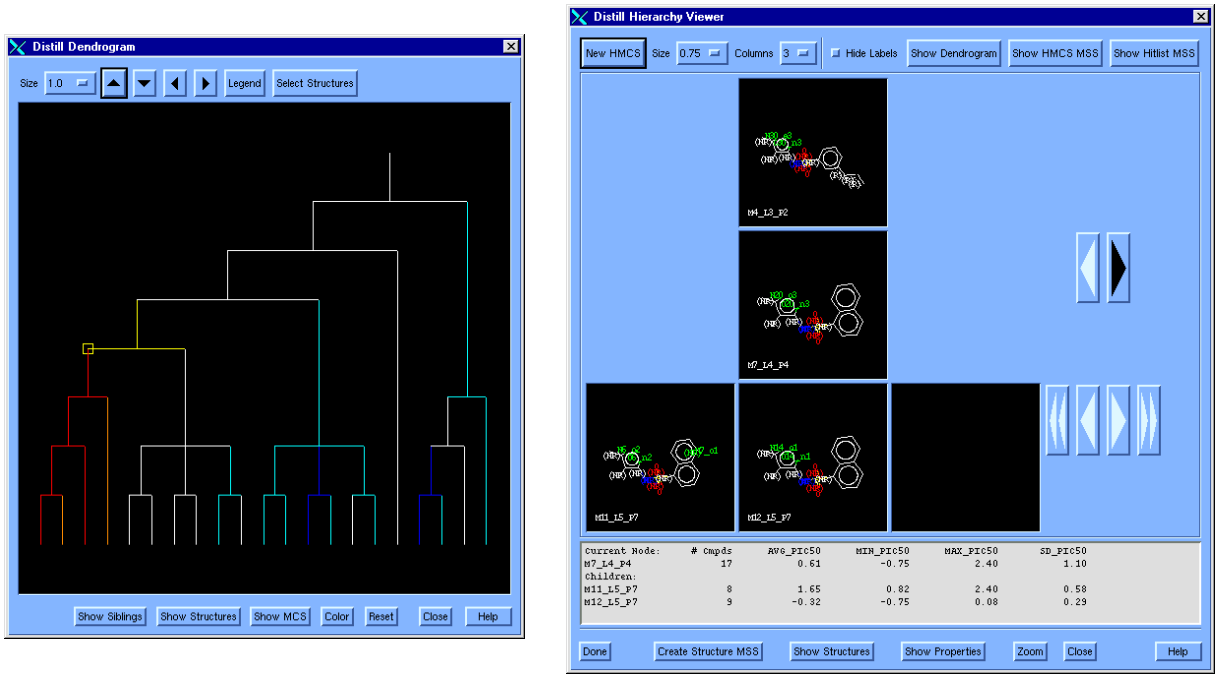


Abb. 3.4: Screenshot des Distill GUI. Das kleine Quadrat im Distill Dendrogram (links) kennzeichet den aktuellen Knoten, dessen zugehörige Struktur im Hierarchy-Viewer (rechts) in der mittleren Zeichenfläche abgebildet ist. Darüber ist die Struktur des Elternknotens und darunter die Strukturen in der nachfolgenden Hierarchieebene eingezeichnet.

Die Berechnung der Strukturen für das Dendrogramm ist aus Rechenzeitgründen (mindestens $O(N^3)$) nicht für sehr große Datensätze möglich.

Zur Betrachtung eines schematischen Dendrogramms gibt es den Dendrogramm-Viewer, mit dem die Strukturen und statistischen Informationen zu den Knoten angezeigt werden. Die Navigation ist durch direktes Anwählen der Knoten im Dendrogramm oder durch Anwahl der Navigationspfeile im Hierachy-Viewer möglich. Die Navigation und die sehr eingeschränkte Strukturvisualisierung sind bis dato ungenügend.

3.6 LeadScope

LeadScope Inc. (Columbus, USA) vertreibt das Programm LeadScope, eine „chemistry-based decision support software for discovery and development scientists"[112]. Dem Programm liegt eine vordefinierte Feature-Hierarchie aus ca. 27 000 Substrukturen zugrunde. Zu jedem Templat wird in der Vorbereitungsphase ein entsprechendes substructure-matching vorgenommen. Die interessantesten Haupstrukturklassen der obersten Hierarchieebene sind:

- Amino Acids
- Bases & Nucleosides
- Benzenes

- Heterocycles
- Functional Groups
- Peptidomimetics

Zusätzlich gibt es noch Klassen für chemische Elemente und Pharmakophore (generalisierte Atompairs). Dazu kommen weitere spezifische Level mit einem an die CAS-Nomenklatur angelehnten Namen, z.B. bei Benzenen ein Level für die Anzahl der Substituenten und Substitutionsmuster, und einen dritten für detailierter spezifizierte Substituenten[113].

benzene, 1-,2-subst	(level 1)
benzene, 1-R-, 2-alkoxy-	(level 2)
benzene, 1-acetamido-, 2-alkoxy-	(level 3)
benzene, 1-acetoxy-, 2-alkoxy-	
benzene, 1-acetyl-, 2-alkoxy-	
benzene, 1-alkenyl-, 2-alkoxy-	
benzene, 1-alkoxy-, 2-alkyl-	
benzene, 1-alkoxy-, 2-amino-	

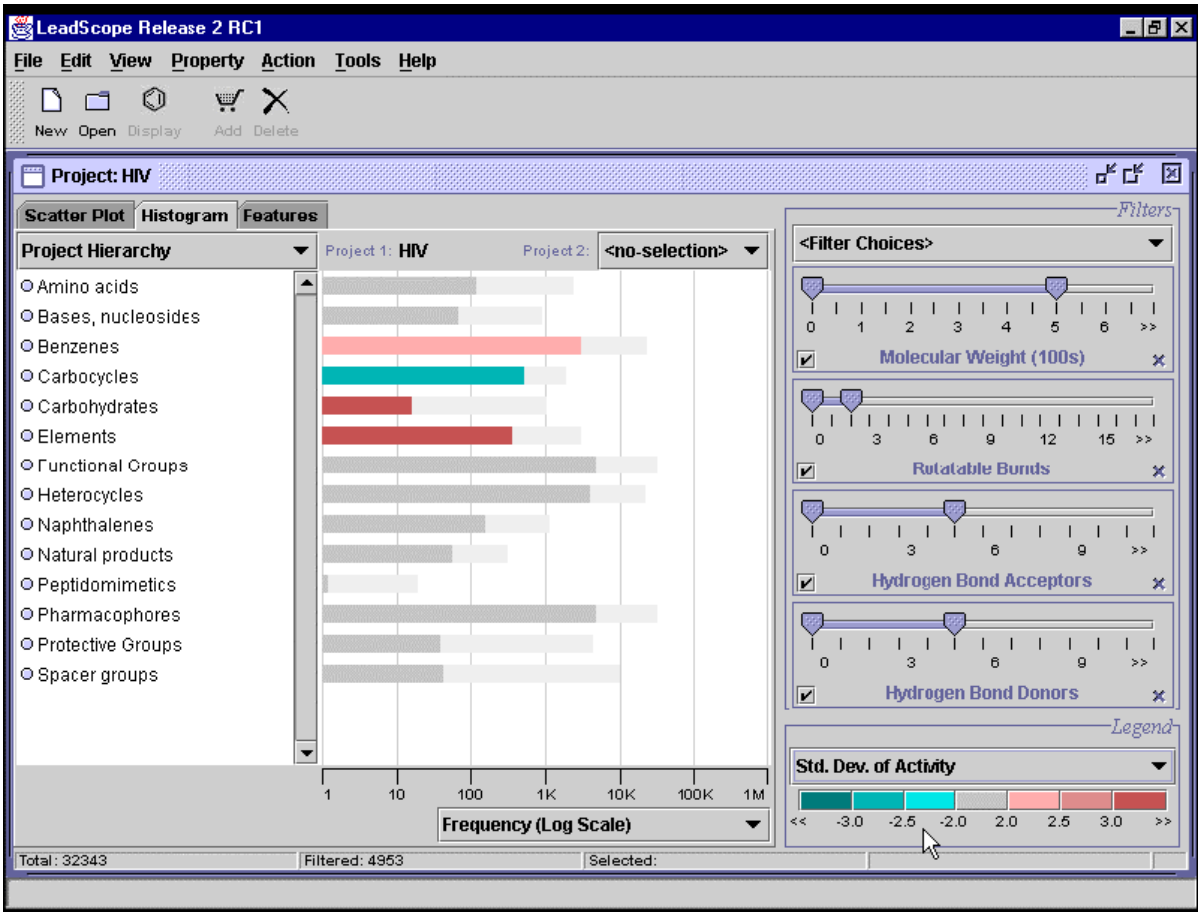


Abb. 3.5: Screenshot des Leadscope GUI.

Jedem Eintrag in der Strukturklassenhierachie ist ein Histogrammbalken zugeordnet, dessen Länge die Häufigkeit des Auftretens repräsentiert. Der Balken kann entsprechend einer auswählbaren Eigenschaft der Daten farbkodiert werden, z.B. entsprechend der Abweichung der Aktivitätshäufigkeit der Strukturklasse im Verhältnis zum Gesamtdatensatz. Auf diese Art

können Klassen mit vielen Repräsentanten mittels einfachem Browsen durch die Hierarchie identifiziert werden, indem auf die entsprechend gefärbten langen Balken geachtet wird.

Ein weiterer Bereich der Benutzeroberfläche besteht aus den an Spotfire angelehnten range sliders zur Wahl einer Ober- und Untergrenze für mehrere berechnete (Lipinski-Parameter) oder dazugeladene Attribute. Die Werte der Attribute werden in (veränderbare) Bereiche eingeteilt, damit sie in einem Bitvektor kodiert werden können. Die Filterung des Ergebnisses erfolgt, indem Vektoren der Attributbereiche mit den Strukturfeature-Bitvektoren über schnelle Boolsche Operationen (AND/OR) verknüpft werden[114].

3.7 Bioreason

Innerhalb der ClassPharmer Suite verwendet die Firma Bioreason Inc. (Santa Fe, USA) dynamisch berechnete MCS als zentrales Ordungskriterium für die identifizierten Klassen[115, 116]. Die MCS müssen nicht zwingend zusammenhängende Substrukturen sein; ein isofunktioneller Ersatz wird nicht direkt berücksichtigt. Obwohl intern eine baumartig rekursive Berechnung vorgenommen wird, werden die ermittelten Klassen ohne Beziehung zueinander als Zeilen einer Tabelle ausgegeben. Sie können nur nach einer Datenspalte, z.B. Molekulargewicht, sortiert werden.

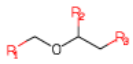
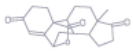
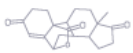
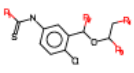
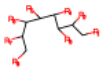
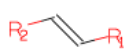
Class ID	Class Name	Scaffold	Actives	Median Activity	Average Activity	Actives SD	Max Activity	Filtered	% Actives
1	Class 1		37	6.70	6.90	0.97	9.20	506	6.81
2	Class 2		33	6.80	6.65	0.71	7.90	120	21.57
14	Class 14		30	6.80	6.85	0.58	7.90	18	62.50
11	Class 11		30	6.10	6.53	1.07	8.70	698	4.12
95	Class 95		29	6.70	7.02	1.01	9.20	333	8.01
3	Class 3		28	5.70	5.81	0.31	6.70	1624	1.70
5	Class 5		19	5.80	5.98	0.47	7.20	977	1.91
175	Class 175		19	6.30	6.59	1.15	8.70	148	11.38
178	Class 178		19	6.80	6.84	0.66	7.90	1	95.00
24	Class 24		18	5.70	5.66	0.12	5.90	222	7.50
4	Class 4		18	7.20	7.13	0.31	7.20	101	15.13

Abb. 3.6: Screenshot des ClassPharmer GUI.

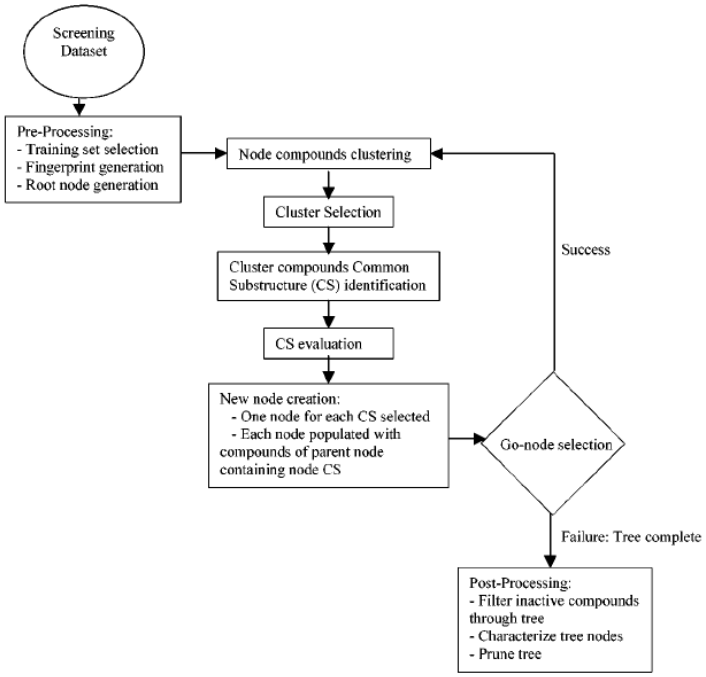


Abb. 3.7: Flowchart des von Bioreason verwendeten Algorithmus.

Zur Erstellung des phylogenetic-like tree (PGLT) wird ein Hybridalgorithmus verwendet, der verschiedene Techniken wie Neuronale Netzwerke, Genetische Algorithmen, Expertenregeln und Chemische Substruktursuche enthält[117, 118, 119]. Er besteht aus einer Initialisierungs- und Postprocessing-Phase und einer rekursiv wiederholten Hauptroutine. Der PGLT-Algorithmus verwendet keine Aktivitätsinformation der Verbindungen und ist daher unsupervised.

In der Initialisierungsphase wird als erstes der Wurzelknoten des PGLT erzeugt und mit allen aktiven Verbindungen des untersuchten HTS-Datensatzes populiert. Der Wurzelknoten wird in der Hauptroutine des Algorithmus prozessiert, die aus folgenden Teilschritten besteht:

1. Um die Moleküle basierend auf der Ähnlichkeit ihrer chemischen Deskriptoren (ISIS/MDL-Fingerprints) zu gruppieren, wird ein Clustering-Algorithmus verwendet. Zum Einsatz kommt die auf neuronalem Netzwerk basierende SOM-Methode (self-organizing map).
2. Aus dem Ergebnis der Clusterung wird eine Menge von „natürlichen“ Clustern selektiert (sog. hotspots, das sind SOM-Knoten, die eine vorgegebene Mindestanzahl an aktiven Verbindungen enthalten) .
3. Die MCS (maximum common substructure) für die Verbindungen jedes Clusters wird mittels eines genetischen Algorithmus bestimmt.
4. Die gemeinsamen Substrukturen werden durch Regeln eines Expertensystems bewertet und, falls sie keinen signifikanten Anteil an neuen Informationen repräsentieren bzw. identisch zum parent node oder einem anderen Subknoten des parent nodes sind, entfernt.
5. Für jede neu identifizierte gemeinsame Substruktur wird ein neuer Knoten erzeugt, dem alle Verbindungen des parent node, die die jeweilige Substruktur enthalten, hinzugefügt werden.
6. Die neuen Knoten werden mit dem parent node verknüpft.
7. Knoten, die weiterverarbeitet werden (sog. go nodes), werden nach Regeln bestimmt, die das Baumwachstum definieren (Tiefensuche, Breitensuche oder Knoten mit abnehmender Diversität).

Dieser Prozeß wird mit den go nodes als Input rekursiv wiederholt. Vorgesehene Abbruchkriterien sind:

- Die terminalen Knoten haben einen zu kleinen Diversitätskoeffizienten.
- Die terminalen Knoten sind unterhalb einer vorgegebenen Mindestgröße.
- Der Baum ist bis zur vorgegebenen Tiefe erzeugt worden.

Eine wichtige Eigenschaft des Algorithmus ist die Fähigkeit, die „multidomain nature“ der chemischen Verbindungen zu berücksichtigen, d.h. sie werden verschiedenen Knoten gleichzeitig zugeordnet, wenn sie mehrere relevante Substrukturen enthalten.

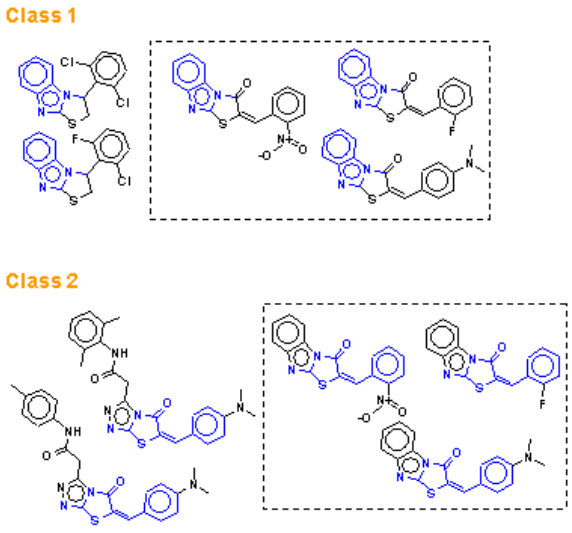


Abb. 3.8: Veranschaulichung der „multidomain nature“ von chemischen Verbindungen.

Der Benutzer kann den Gesamtprozess nur durch einen einzelnen abstrahierten Parameter beeinflussen. Ist der Grad der Homogenität auf hoch gesetzt, wird eine große Zahl an geringpopulierten Klassen und viele Singletons generiert[120]. Wird die Einstellung auf niedrig gesetzt, werden wenige große Klassen erzeugt, deren gemeinsames Scaffold nur sehr klein ist. Alle weiteren für die verwendeten Verfahren relevanten Parameter sind, für den Benutzer verborgen, programmintern abgeschätzt oder festgelegt.

4 Bewertung und Planung

4.1 Nachteile der bekannten Strukturvergleichsverfahren

Alle bisher bekannten Verfahren haben gravierende Mängel bei einem oder mehreren der genannten Kriterien zur Informationsverdichtung bzw. Strukturbewertung. Zu den wichtigsten Schwachstellen gehören:

- Verwendung von Ähnlichkeitsvergleichen im Ligand-Strukturraum (Clusteranalyse)
- Anfälligkeit der Ergebnisse gegenüber Änderungen des Datenbestandes (veränderte Clusterbildung)
- Erschwerte Chemiebeurteilung und Optimierung neuer Chemie auf der Ebene von Deskriptoren („inverses Designproblem“)
- Ignorieren mechanistischer Aspekte der Wirkstoff-Target-Interaktion, vor allem bei Fingerprints
- fehlende automatisierte Templaterkennung bzw. Identifizierung einer Referenzstruktur oder Notwendigkeit der Definition einer Templatbibliothek
- mangelnde Berücksichtigung des modularen Charakters von Wirkstoffen bei der Synthese und der Interaktion mit dem Target
- problematisches Laufzeitverhalten

Die Entwicklung eines Verfahrens, das allen Bewertungskriterien gerecht wird, böte die Möglichkeit, die Bewertung biologischer Hits und damit die nachgeschalteten Arbeitsabläufe der Wirkstoffoptimierung effizienter und damit auch erfolgreicher zu gestalten.

Bei der konzeptionellen Planung des BayTree-Programms, das im folgenden vorgestellt werden soll, wurde deshalb Wert auf folgende Eigenschaften gelegt:

- Die Bewertung und Visualisierung biologischer Testergebnisse erfolgt ausschließlich anhand der chemischen Struktur (nicht mittels skalarer Deskriptoren, Fingerprints etc.)
- Es sollen keine Paarvergleiche durchgeführt oder Substruktur-Gemeinsamkeiten algorithmisch bestimmt werden. Da jede Struktur nur einmal prozessiert wird, wird die kombinatorische Explosion der Vergleiche und die Sensitivität gegenüber Änderungen des Datenbestandes vermieden.
- Im Prinzip sollen alle Ergebnisse sichtbar sein – es soll aber ein strukturbasiertes „Verdichtungsprinzip“ existieren, das strukturverwandte Verbindungen durch repräsentative Vertreter exemplarisch beschreibt.
- Es soll eine benutzerkontrollierte, dynamisch handhabbare hierarchische Anordnungsform (Baumhierarchie) der repräsentativen Vertreter und aller zugeordneten „Derivate“ verwendet werden, in der die Anwender zu chemisch „interessanten“ Hits navigieren können.
- Die Kriterien zur hierarchischen Anordnung sollen regelbasiert und computer-automatisiert sein und chemische Gesichtspunkte, d.h. den modularen Aufbau der Wirkstoffe entsprechend der Wichtigkeit für die Interaktion mit dem Target berücksichtigen.
- Durch die Priorisierung der Molekülfragmente, z.B. nach topologischen und funktionalen Kriterien, soll die Darstellung unterschiedlicher chemischer Strukturdaten standardisiert werden. Dies ermöglicht, targetbezogene Informationen

aus unterschiedlichen Quellen (Patente, Publikationen, Vorträge etc.) inhaltlich nach gleichen Kriterien zusammenführen bzw. auszuwerten. Andererseits können verschiedene biologische Targets in ihrer Reaktion auf dieselbe chemische Substanzbank vergleichend dargestellt werden.
- Die Struktur- und Fragmentbewertung erfolgt nach einheitlichen Regeln und berücksichtigt den modularen Aufbau der Verbindungen.
- Der Bewertungsprozeß muß anschaulich sein und manuell (mit Bleistift und Papier) am Einzelbeispiel nachvollziehbar sein.

Die bisher bekannten Verfahren werden diesen Anforderungen nicht gerecht. Nachfolgend werden die Ausgangsüberlegungen zur Verbesserung der strukturbasierten Auswertung zusammengefaßt.

4.2 Idee und Realisierung des BayTree-Konzeptes

Das angestrebte Eigenschaftsprofil erfordert die Lösung folgender Teilprobleme:

- die Identifizierung einer geeigneten Referenzstruktur für jede aktuell untersuchte Verbindung (siehe Abschnitt 5.1.4)
- die Charakterisierung und Klassifizierung der Referenzstruktur durch einen MolCode (TSC) (siehe Abschnitt 5.1.8)
- die Bewertung unterschiedlicher chemischer Modifikationen und Dekorationen der Referenzstruktur (Derivatisierung, Pharmakophor) (siehe Abschnitt 5.1.12)
- die Festlegung von Priorisierungskriterien für die Strukturkomponenten jeder Verbindung, die im Rahmen eines hierarchischen Ordnungsprinzips „on the fly“ anwendbar sind (siehe Abschnitt 5.1.6)
- die dynamische Visualisierung der Strukturhierarchien in einer baumartigen Struktur (siehe Abschnitt 5.1.10)
- die anwendergesteuerte Navigation in der Informationshierarchie (siehe Abschnitt 5.1.11)

Die biologische Wirkung einer niedermolekularen Verbindung auf ein biologisches Target wird durch die räumliche Struktur, die Topologie, die chemische Dekoration sowie die physikalisch-chemischen Eigenschaften ihrer Fragmente und ihre Abstimmung auf das Target bestimmt. Daher liegt es nahe, eine „chemisch inerte“ aber topologisch und geometrisch möglichst gleichartige „virtuelle“ Referenzstruktur zu jedem gefundenen Konstitutions-prototypen zu definieren: das All-Kohlenstoff-Analogon gleicher Konstitution. Dieses ähnelt dem erstmals von Bemis und Murcko[121] als Framework bezeichneten Grundgerüst einer Verbindung. In ihrer Publikation haben sie die Frameworks der Strukturen der Comprehensive Medicinal Chemistry (CMC) Datenbank[122] mit 2D connectivity triangle shape Deskriptoren[123] codiert und eine Statistik erstellt. In den 5120 untersucheten Drugs sind 306 azyklische Verbindungen (ohne Framework) und 1179 verschiedene Frameworks enthalten. Davon sind 783 Framework-Singletons, die nur in einer einzelnen Verbindung vertreten sind. Alle Frameworks, die mindestens 20mal auftreten, sind in Abb. 4.1 aufgeführt. Diese 32 verschiedenen Frameworks, mit dem Sechsring an erster Position, repräsentieren ca. 50 % der untersuchten Drugs.

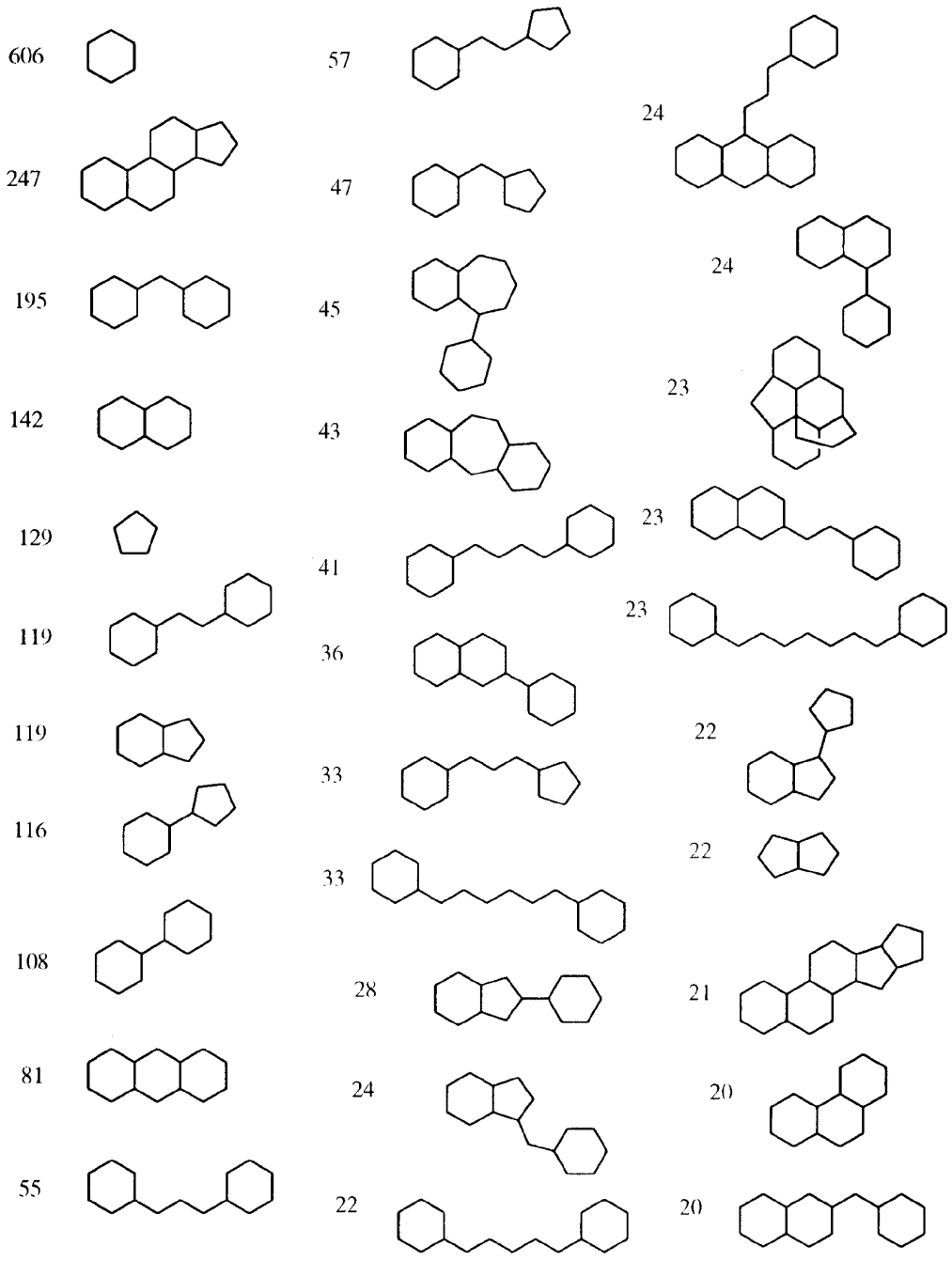


Abb. 4.1: Die 32 häufigsten Frameworks von Drugs der CMC-Datenbank. Die Zahlen stehen für die Anzahl an Vertretern.

Das Framework übernimmt die Rolle der Referenzstruktur und wird im Folgenden als Topologisches Cluster-Zentrum TCC (Topological Cluster Center) bezeichnet. Ohne jegliche Funktionalität bzw. Pharmakophore ist es per definitionem nicht in der Lage, mit dem Target chemisch zu interagieren und biologische Signale auszulösen. Als Nebeneffekt dieser Wahl der Referenzstruktur ergibt sich die Möglichkeit, die in der realen Verbindung identifizierte chemische Dekoration als Modulator der chemischen Wirkung (Pharmakophor-Elemente), synthetische Auxiliare oder die biologische Wirkung störende chemische Zusatzfunktionalität zu entlarven. Dies ist durch den Wirkvergleich topologisch ähnlicher, aber unterschiedlich funktionalisierter Verbindungen am selben Target möglich.

Berücksichtigt man den modularen Aufbau jeder Verbindung aus topologischen Komponenten (Ringe, Linker, Ketten etc.), legitimiert die Pharmakophor-Hypothese die Priorisierung der topologischen Fragmente über ihren chemischen Funktionalisierungsgrad. Da an den modularen Aufbau der Verbindung aus den topologischen Komponenten auch die Zusammensetzung des Gesamtpharmakophors geknüpft ist, können Hinweise darauf aus den Testergebnissen evtl. vorhandener gleichartig funktionalisierter Teilkomponenten-verbindungen abgeleitet werden. Der topologische Aufbau der getesteten Verbindungen läßt sich, ähnlich einer chemischen Summenformel, durch die Bruttozusammensetzung seiner Topologieelemente charakterisieren und deshalb zur Templat-Klassifizierung der Referenzstruktur im Topologiebaum ebenso wie zur elektronischen Speicherung und zur Vereinigung unterschiedlicher Datenquellen oder vorgefilterter Datenbank-Subsets verwenden. Chemische Derivate können danach in einer hierarchischen, topologischen Baumstruktur unter der Referenzstruktur als Topologisches Cluster Zentrum (TCC) gebündelt werden.

Die relative Priorisierung der Topologie-Komponenten jedes Moleküls erfolgt nach einfachen empirischen Regeln. Ausgehend von der Überlegung, dass eine betrachtete chemische Modifikation und die Entschlüsselung der dadurch am Target ausgelösten Wirkung um so eindeutiger möglich ist, je besser ihre geometrische Position in der Konstitutionsformel definiert ist. Dies ist in Ringen besser als in Ketten gleicher Länge, mit zunehmender Ketten-/Ring-Länge geringer, bei aromatischen (also achiralen, planaren) Ringen besser als bei gesättigten zyklischen Kohlenwasserstoffen und bei Heteroatomen besser als bei chemischen Substituenten ähnlichen Typs. Die Priorisierung der Fragmente folgt damit der Sicherheit und Eindeutigkeit der strukturchemischen Interpretierbarkeit des biologischen Ergebnisses.

Die Generierung des Topologischen Cluster Zentrums läßt sich algorithmisch als chemisches Morphing der Heteroatome zu Kohlenstoff durchführen und entspricht mathematisch einer surjektiven Abbildung aller chemischen Derivate einer gegebenen Topologie auf eine gemeinsame (repräsentative) Referenzstruktur. Dabei wird die Selbstähnlichkeitsanalyse umfangreicher Substanzdatenbanken zur Ermittlung struktureller Ähnlichkeiten konzeptionell vermieden und gleichzeitig sichergestellt, dass zu jedem Verbindungstyp unabhängig von der Häufigkeit ein repräsentativer topologischer Vertreter existiert.

Da alle Verbindungen gleichzeitig nach denselben Regeln in ihre topologischen Fragmente (Module) „on the fly“ zerlegt und diese nach einheitlichen Kriterien bewertet (priorisiert) werden, ergibt sich eine sehr effiziente hierarchische Informationsverdichtung und gleichzeitig eine Positionierung aller Substrukturen in einem hierarchisch organisierten Topologiebaum (im Folgenden als TST Topological Structure Tree bezeichnet) der Strukturtemplate, die dabei selbst wiederum als Referenzstrukturen ihrer Derivate in der Datenbank dienen.

Durch geeignete Kolorierung der Knotenstrukturen nach einer wählbaren Eigenschaft, z.B. Aktivität/Inaktivität, läßt sich der Einfluß von Templat und chemischer Modifikation auf das gewünschte Eigenschaftsprofil visualisieren und analysieren. Es bietet sich deshalb an, die zu analysierenden Daten bezüglich der gewünschten Eigenschaften vorzufiltern.

5 BayTree

5.1 Beschreibung der BayTree-Methode

Es wird ein regelbasiertes Bewertungsschema angewendet, um zu jeder betrachteten Verbindung das entsprechende TCC zu erzeugen und die individuellen topologischen Unterklassen zu bewerten. In einem TST (Topological Structure Tree) mit zunehmendem topologischem Detailgrad werden Pfadsegmente erzeugt, die als eine Abfolge von Verknüpfungen (Beziehungen, Kanten im TST) verwendet werden können, und zwar vom TST-Wurzelknoten, der das am höchsten priorisierte topologische Unterklassenelement enthält, zum abschließenden TCC-Knoten, an den alle Repräsentanten mit chemischen Modifikationen angehängt werden. Jede neue Verbindung wird analysiert und jeder Knoten in seinem TSP (Topological Sequence Path) wird zwischen der TST-Wurzel und dem TCC der Verbindung erzeugt. Der TSP kann gemeinsame Merkmale mit anderen Verbindungen aufweisen. Falls zum Zeitpunkt der Prozessierung ein Wurzelknoten noch nicht existiert, wird - wie zuvor beschrieben - ein neuer TST zu dem vollständigen topologischen Pfad erzeugt. Anderenfalls werden die überlappenden Abschnitte vorhandener TSTs zur Anknüpfung der neuen, nichtüberlappenden Strukturelemente verwendet. Der Templatbaum (TST), der letztendlich aus den Eingabedaten erzeugt wird, ermöglicht es, große Datenmengen nach topologischen Kriterien zu analysieren, die in dem zugrundeliegenden regelbasierten System zur Erzeugung verschiedener Ebenen von Detailgraden berücksichtigt werden. Diese zeigen die hierarchisch strukturelle Entwicklung der topologischen Merkmale, die in dem TSF (Topological Structure Forest) der Eingabedaten visualisiert werden.

Da die Anordnung und die Bewertung innerhalb des TST sowohl eindeutig als auch veränderbar bzgl. Reihenfolge und Inhalt der angewendeten Regeln ist, wird ein flexibles strukturbasiertes Auswertesystem erzeugt, das den Anforderungen des Benutzers so angepasst werden kann, dass er durch die visualisierten Baumstrukturen navigieren kann, um diejenigen Daten zu finden, die am geeignetsten für die favorisierten Syntheserouten oder die vorhandenen Synthone sind.

Die Anwendung des BayTree-Verfahrens läßt sich in folgende Teilschritte zerlegen:

- Festlegung der Reihenfolge der Teilaufgaben des Gesamtvorgangs
- Identifizierung der topologischen Komponenten eines Moleküls
- Bewertung der topologischen Komponenten durch Regeln
- Bewertung der Elemente innerhalb einer Topologieklasse mittles Regeln
- Erzeugung der Topologischen Referenzstruktur (TCC)
- Erzeugung, Verknüpfung, Kennzeichnung und Visualisierung der Knoten und der (Unter)Strukturen in einem Topologischen Baum (TST)
- strukturelle, statistische und biologische Analyse der TST-Knoten
- Speicherung und Laden von topologisch analysierten Datensätzen
- Subtree-Bewertung und Strukturierung unterhalb der TCC-Knotenebene.

Die strukturbasierte Analyse der molekularen Graphen chemischer Verbindungen in großen Datensätzen geht in folgenden Schritten vonstatten (die lateinischen Nummern beziehen sich auf Abb. 5.1):

1. Sequenzielles Einlesen der Struktur I und Erzeugung des molekularen Graphen II ohne Wasserstoffatome zur weiteren Analyse (siehe 5.1.2 Graphentheoretische Beschreibung der Molekültopologie)
2. Klassifizierung der Knoten des molekularen Graphen in topologische Unterklassen (III für Ringe und IV für Linker) (siehe 5.1.3 Topologische Fragmentierung des Molekülgraphen)
3. Priorisierung der topologischen Klassen VI und Erzeugung des Topological Sequence Path (TSP) zwischen dem höchstpriorisierten topologischen Unterklassenelement und dem TCC (siehe 5.1.6 Priorisierung der topologischen Komponenten und 5.1.7 Priorisierung innerhalb einer topologischen Kategorie)
4. Erweiterung des TSP zum MolCode (=TSC) durch Ergänzung von Zusatzinformationen V über Substituenten und Heteroatome (siehe 5.1.8 Erstellung des MolCode)
5. Erzeugung des Framework des topologischen Cluster-Zentrums (TCC) VII, der als Teil eines globalen TST für die Eingabedaten betrachtet wird, und Beschriftung mit dem dazugehörigen TSP oder TSC (siehe 5.1.9 Generierung der topologischen Referenzstruktur)
6. Erzeugung oder Verknüpfung des ganzen Pfades (siehe Abb. 5.2) oder von Teilen des TSP im vorhandenen TST (siehe 5.1.10 Aufbau des topologischen Strukturbaums)
7. Verknüpfung des eigentlichen molekularen Graphen der Eingabestruktur mit dem TCC (siehe Abb. 5.2)
8. Aktualisierung spezieller Speicherfelder für die Repräsentanten unterhalb des TCC oder aller Knoten im TST für Screening-Statistik, Subtree-Belegungsgrad oder Statistik der angehängten Tochterknoten etc. (siehe 5.1.11 Datenanalyse basierend auf den TST-Knoten)

Der Gesamtvorgang wird sequentiell für alle Verbindungen wiederholt. Zu jeder Struktur muss die Konnektivitätsinformation des molekularen Graphen gegeben sein.

Falls die Anzahl der Repräsentanten unterhalb eines TCC eine vorgegebene kritische Anzahl überschreitet, ist eine horizontale Sortierung auf dieser Ebene erforderlich. Dies kann erreicht werden, indem für jede Verbindung geeignete Grapheninvarianten berechnet werden, die unter Verwendung einer präzisen Metrik, z.B. der Mahalanobis-Distanz, zum Sortieren und Bewerten der Strukturen benutzt werden können (siehe 5.1.12 Priorisierung der chemischen Dekoration).

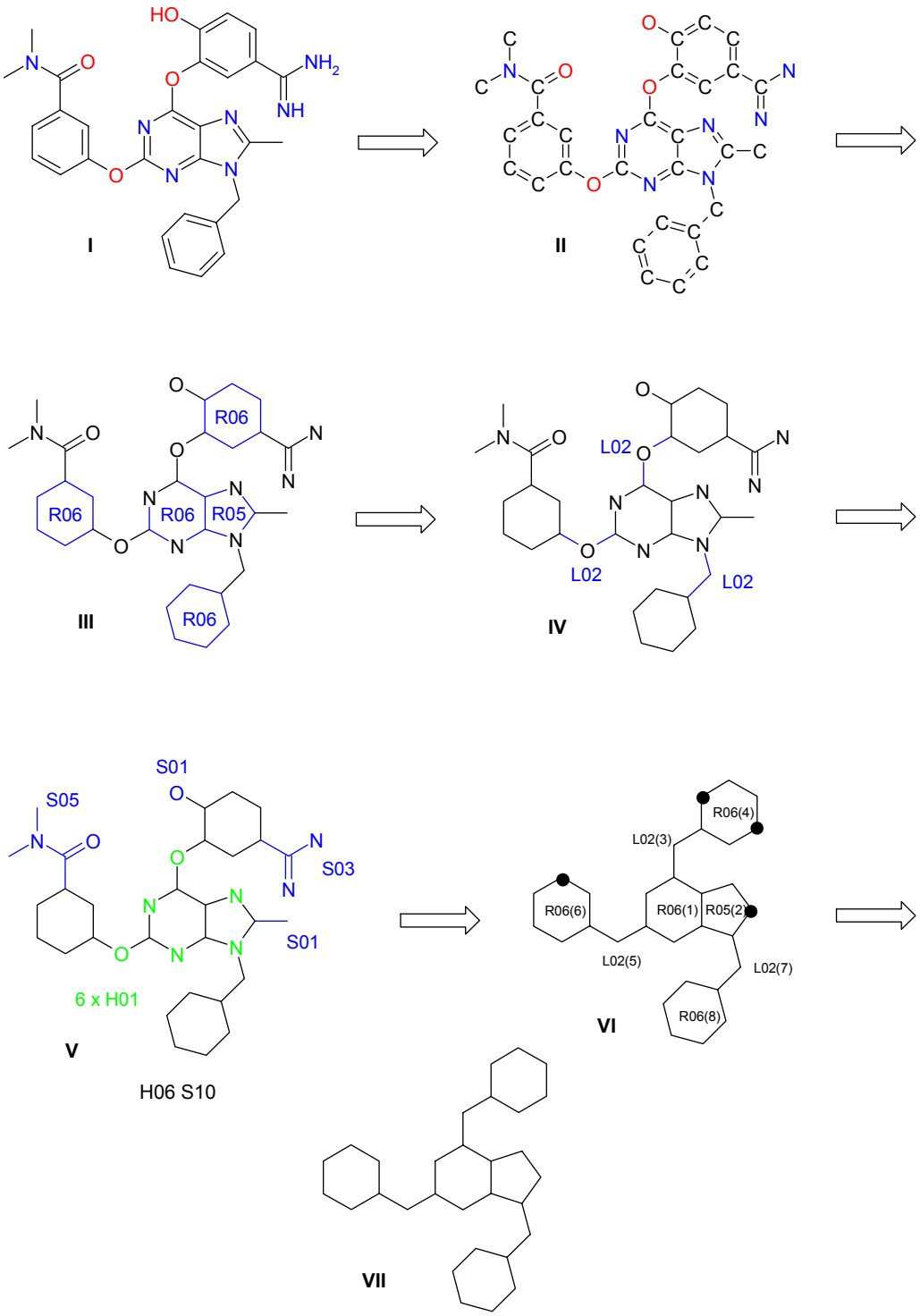


Abb. 5.1: Ausgewählte Schritte während der Erstellung des topologischen Cluster-Zentrums (TCC) und des MolCode (TSC). Ausgehend von der Eingabestruktur (I) wird der molekulare Graph (II) erzeugt, die Klassifizierung der Knoten (III-V) vorgenommen und nach Priorisierung der Unterklassenelemente (VI) das zugehörige TCC (VII) erzeugt.

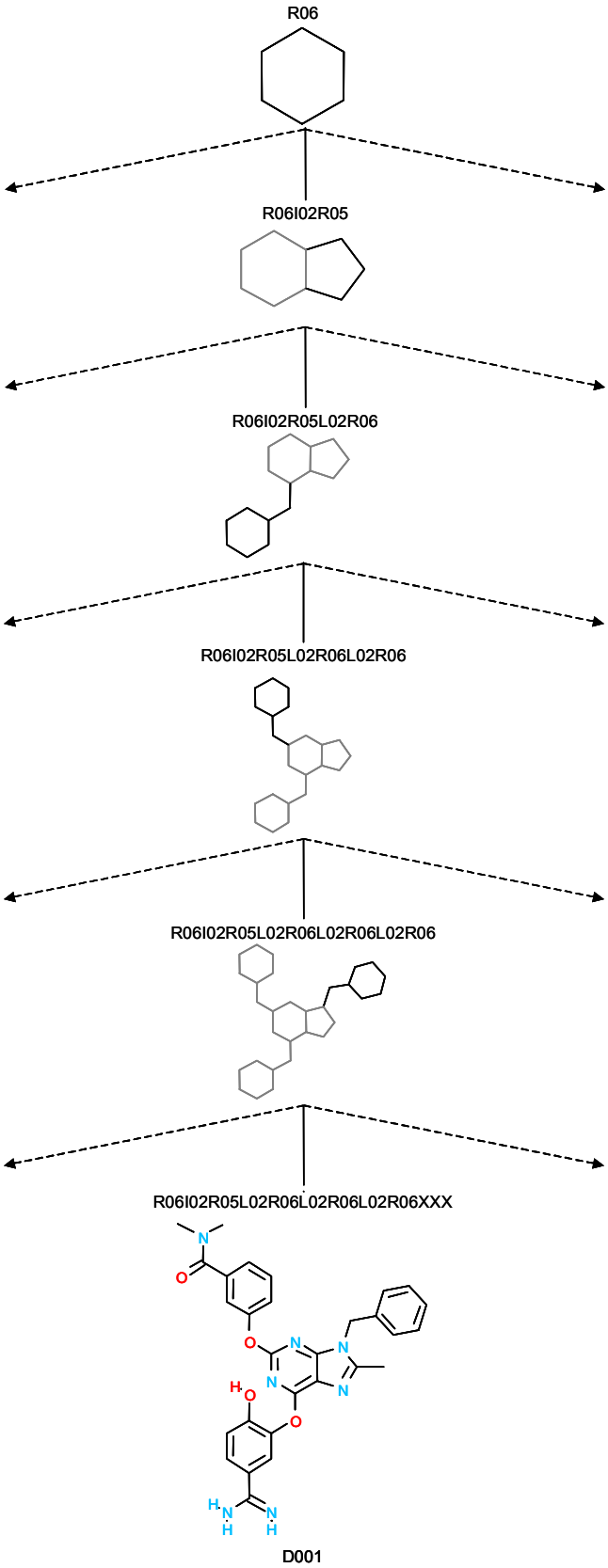


Abb. 5.2: Ausschnitt eines hierarchischen topologischen Strukturbaums (TST) entlang des topologischen Sequenzpfades (TSP) einer Verbindung und Bezeichnung der Frameworks jeder Hierarchieebene.

5.1.1 Auswahl der Eingabestrukturen durch Eigenschaftsfilter

Je nach Zielsetzung der Analyse empfiehlt sich eine Strukturvorauswahl durch Anwendung geeigneter Eingabefilter. Die in Frage kommenden Filter lassen nur Strukturen mit dem gewünschten Eigenschaftsprofil durch:

- aktive Substanzen eines bestimmten Screening Assays für die Hitanalyse
- inaktive Substanzen eines bestimmten Screening Assays für die Wahrscheinlichkeits-abschätzung für falsch Positive/Negative
- alle aktiven Substanzen der Screening-Historie zum Bio-Profiling der Substanz-datenbank
- alle Substanzen des Prüfpräparatelagers oder Teile davon zum Profiling und zur Lückenanalyse als Entscheidungsbasis für Substanzsynthese und -einkauf
- Struktur-Aktivitätsdaten aus Konkurrenzpatenten zur Suche von Patentlücken und zur Wissenserweiterung
- aktive endogene Verbindungen (Bioeffektoren) oder aktive Metaboliten für indirekte Target-Klassifizierungen
- aktive Naturprodukte mit ungewöhnlichen Gerüststrukturen für Struktur-Aktivitätsanalysen und zur Templatauswahl

Die Auswahl der Eingabedaten bestimmt offensichtlich den Anwendungsbereich und die strukturellen und statistischen Ergebnisse der Analyse.

5.1.2 Graphentheoretische Beschreibung der Molekültopologie

Jede Verbindung wird als ein ungerichteter molekularer Graph **G**(**V**, **E**) betrachtet, dessen Wasserstoffatome entfernt worden sind, wobei **V**(v_1, v_2, ...) die Menge der Knoten (Atome) und **E**(e_1, e_2, ...) die Menge der Kanten (chemische Bindungen) sind. Für die Struktur i der Eingabedatei wird dieser Graph mit **G**(i) abgekürzt. Knoten (Atome) in diesem Graphen kann jedes Nicht-Wasserstoffatom sein, wobei Kohlenstoff als die virtuelle Referenz für drug-like-Verbindungen betrachtet wird. Kanten können Bindungen vom Typ einfach, doppelt, dreifach oder partiell doppelt, d.h. aromatisch, sein. Jede Verbindung bzw. ihr Graph kann in Subgraphenelemente unterteilt werden. Dies ist entweder eine topologische Unterklasse **T**={**R**, **L**, **C**, **S**} je nach Konnektivität als Ring (R), Linker (L), Chain (C), Substituent (S) oder ein Modulator für atomare Eigenschaften, z.B. Heteroatome H = $\{v_i \neq$ Kohlenstoff$\}$, die die physikalischen oder chemischen Eigenschaften beeinflussen. Für die Anwendung in späteren Phasen der Datenanalyse, z.B. der Pharmakophoranalyse, ist es erforderlich, dass einige dieser Mengen (S, H) weiter unterteilt werden, um die Funktionalität für ein Target und/oder Lösungsmittel zu charakterisieren, z.B. Wasserstoffbrücken Donoren, Akzeptoren oder ionisierbare Gruppen. Für quantitative Struktur-Aktivitäts-Beziehungen (QSAR) oder quantitative Struktur-Eigenschafts-Beziehungen (QSPR) oder Signifikanzanalysen der Verbindungen wird der Graph in einen äquivalenten Line-Graphen transformiert.

5.1.3 Topologische Fragmentierung des Molekülgraphen

Für jede Verbindung bzw. den entsprechenden molekularen Graphen **G** können die Elemente der topologischen Klassen ermittelt werden. In einem Graphen sind nur Ringelemente Start-

und Endpunkt für self-returning walks. Formal werden ausgehend von jedem Atom alle Pfade des molekularen Graphen erschöpfend analysiert. Alle Pfade, die nicht in Ringen enden oder Teil eines Ringes sind, werden abgeschnitten. Für die Klassenelemente R und L wird die Anzahl der Substituenten gezählt und für den Priorisierungsprozess gespeichert.

Jeder zyklische Subgraph innerhalb von **G** bildet einen Ring, der durch die Länge des Hamilton-Pfades dieser Substruktur, d.h. die Ringgröße (r=3, 4, 5, ...), charakterisiert wird. Alle Ringe der Verbindung bilden die Unterklasse **R**. Jeder Ring mit einer definierten Größe ist ein topologisches Klassenelement in der Subklasse aller Ringe.

Ein Linker ist definiert als eine azyklische Kette der Länge l (l=1, 2, 3, ...), deren Anfangs- und Endknoten zu unterschiedlichen Ringen gehören. Alle Linker werden in der Linkermenge der Unterklasse **L** zusammengefasst.

Alle weiteren vorhandenen linearen oder verzweigten Subgraphen, die entweder mit einem Ring oder einem Linker verknüpft sind, werden als Substituenten mit der Gesamtgröße s (s ist die Anzahl der Atome) definiert. Alle Substituenten werden in der Menge der Substituenten der Unterklasse **S** zusammengefasst.

In der folgenden Beschreibung wird zusätzlich die Klasse Chain **C** berücksichtigt. Sie basiert auf einer engeren Definition der Klasse der Substituenten: Die längste Kohlenstoffkette wird als Element der Klasse Chain **C** definiert und nur die bekannten chemischen funktionellen Gruppen, z.B. Halogen, Amin, Carboxyl, Hydroxyl, Sulfonamid, als Substituenten **S**. Im Programm wird die Klasse **C** derzeit nicht berücksichtigt.

Die Subklasse **H** wird durch die Menge der Heteroatome gebildet, die als Kohlenstoffersatz in Ringen, Linkern (oder Chains) des Moleküls auftreten. Sie sind verantwortlich für die Änderungen des entsprechenden All-Kohlenstoff-Framework, welches als virtuelles Topologisches Cluster Center (TCC) des Scaffold angenommen wird.

Heteroatome unterscheiden sich von Kohlenstoffatomen in der Anzahl der Valenzen (Bindungen, Lonepairs oder Elektronenlücken), die Einfluß haben auf sterische und elektronische Eigenschaften wie Basizität oder Acidität, chemische Reaktivität, Wasserstoffbrückenakzeptor-Eigenschaften oder Wasserstoffbrückendonor-Eigenschaften und physikalische Eigenschaften wie Löslichkeit und Bioaktivität (in vitro-Aktivität, pharmakologische Eigenschaften, Toxizität etc.). Durch diese Eigenschaften beeinflussen die Heteroatome die topologischen Subklassen und sind daher geeignet, die relative Wichtigkeit der Ringe, Linker und Substituenten untereinander in der topologischen Darstellung der Daten zu priorisieren.

Mit diesen Definitionen kann jedes strukturelle Element des Graphen einer chemischen Struktur eindeutig klassifiziert werden.

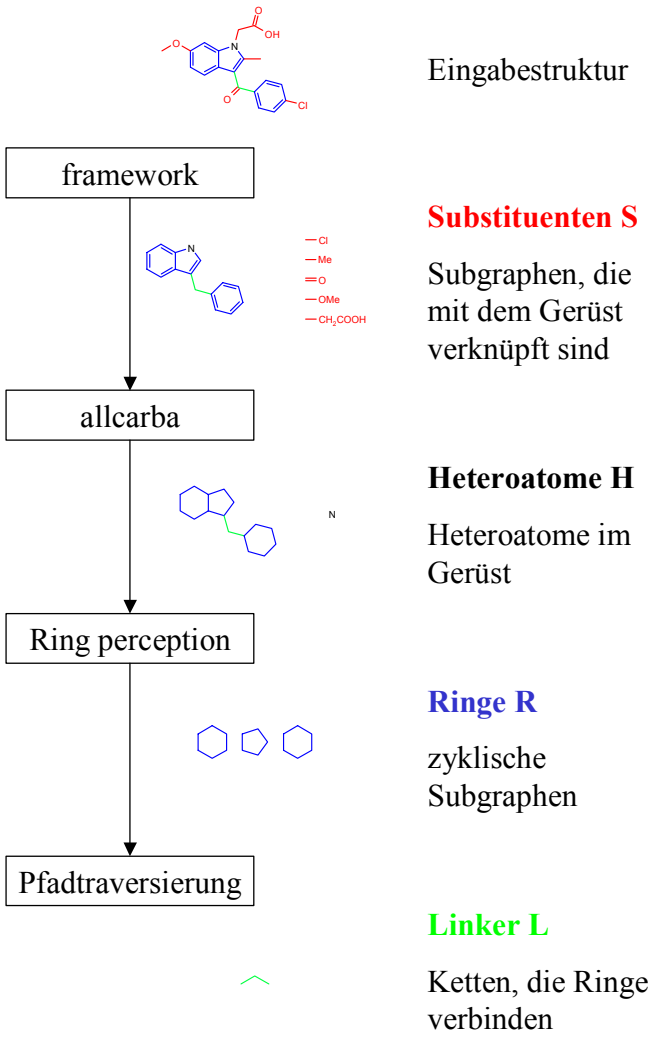


Abb. 5.3: Flowchart der Zerlegung einer Struktur in ihre topologischen Komponenten und Zuordnung dieser zu den Unterklassen.

Die Zuordnungen der Atome zu den Gruppen erfolgen programmintern in folgender Reihenfolge (Abb. 5.3):

1. Die Substituentenatome werden mittels des Framework-Algorithmus von Bemis/Murcko[121] erkannt. Dazu werden rekursiv alle Atome mit nur einem Nachbaratom, d.h. mit der Bindungsordnung eins, entfernt und der Menge S zugeordnet.
2. Die Struktur ist nun auf das Framework reduziert, in dem alle Nicht-Kohlenstoffatome als Heteroatome der Menge H zugeordnet werden. Damit sind alle Atomtypen berücksichtigt und das Gerüst enthält als All-Carba-Analogon nur noch die Topologieinformation.
3. Die Ringe werden über den in Abschnitt 5.2.1 beschriebenen graphentheoretischen Algorithmus detektiert und alle Ringkanten bzw. -atome der Menge R zugeordnet.
4. In der Struktur verbleiben nur noch die Atome der Linker, die die Menge L bilden. Diese werden aus den bei der Ringdetektion gespeicherten Verbindungspfaden zwischen Ankeratomen entsprechend dem Verfahren in Abschnitt 5.2.3 extrahiert.

Zur formal-mathematischen Beschreibung der Funktion eines geeigneten Computerprogramms, das die topologischen Unterklassen des molekularen Graphen ermittelt, wird ein allgemeiner topologischer Operator $\hat{\mathrm{T}}$ definiert. Er repräsentiert eine Menge topologischer Operatoren $\{\hat{R}, \hat{L}, \hat{S}, \hat{C}\}$, die bei k-facher rekursiver Anwendung auf einen molekularen Graphen **G** eine Untermenge von Atomen erzeugt, die der topologischen Unterklasse der Priorität k entspricht und allgemein als $\mathbf{T}_k$ bezeichnet wird. Zu einer gegebenen Verbindung mit r Ringen und l Linkern erzeugt die r-fache Anwendung von $\hat{\mathrm{R}}$ ($\hat{\mathrm{R}}^{\mathrm{r}}$) und die l-fache Anwendung von $\hat{\mathrm{L}}$ ($\hat{\mathrm{L}}^{\mathrm{l}}$) die vollständigen Sätze der Ringe **R** und der Linker **L**.

Im einzelnen gilt:

$$\mathbf{G}(i) = \hat{\mathrm{T}}^0\mathbf{G}(i) \qquad \mathbf{T}_k(i) = \hat{\mathrm{T}}^k\mathbf{G}(i)$$

$$\mathbf{R}(i) = \bigcup_{k=1}^{r} \hat{\mathrm{R}}^k\mathbf{G}(i) \qquad \mathbf{L}(i) = \bigcup_{k=1}^{l} \hat{\mathrm{L}}^k\mathbf{G}(i)$$

$$\mathbf{S}(i) = \bigcup_{k=1}^{s} \hat{\mathrm{S}}^k\mathbf{G}(i) \qquad \mathbf{C}(i) = \bigcup_{k=1}^{c} \hat{\mathrm{C}}^k\mathbf{G}(i)$$

$$\mathbf{G}(i) = \left\{ \mathrm{v}_k \mid \mathrm{v}_k \in \mathbf{V}, \mathrm{v}_k \in \mathbf{R}(i) \vee \mathrm{v}_k \in \mathbf{L}(i) \vee \mathrm{v}_k \in \mathbf{S}(i) \vee \mathrm{v}_k \in \mathbf{C}(i) \right\}$$

Die wiederholte und erschöpfende Anwendung der topologischen Operatoren zerlegt den molekularen Graphen (ohne Wasserstoffatome) in seine Komponenten für alle vorhandenen topologischen Elemente.

5.1.4 Idee der topologischen Referenzstruktur

Jede Verbindung kann durch die gemeinsame charakteristische Anordnung ihrer topologischen Komponenten in der Form einer einfach interpretierbaren topologischen Zeilennotation, dem MolCode, für ihre konstitutionell äquivalente, „virtuelle“ All-Kohlenstoff-Referenzstruktur beschrieben werden. Deren Topologiekomponenten sind aufgrund ihrer chemischen Dekoration mit möglichen Pharmakophorelementen, synthetischen Auxiliaren oder störenden Funktionalisierungen bezüglich ihrer Targetwirkung nicht gleichwertig. Jede chemische Funktionalisierung eines Topologieelements in einer wirksamen Verbindungsklasse beschreibt mit einer gewissen Wahrscheinlichkeit eine aktivitätssteigernde chemische Modifikation (Pharmakophorelement) im Vergleich mit der inaktiven Referenzverbindung. Der modulare Aufbau der Verbindungen aus topologischen Elementen kann deshalb zur Priorisierung der Einzelkomponenten über ihren chemischen Funktionalisierungsgrad bei den aktiven Vertretern ausgenutzt werden. Da der Minimalpharmakophor zur Auslösung der biologischen Wirkung am Target nicht a priori bekannt ist, muß man davon ausgehen, dass auch bereits die funktionalisierten (Teil-) Komponenten Targetwirkung besitzen können. Somit erweist sich die prioritätsgesteuerte Aufsplittung des Gesamttemplats in schalenförmige Erweiterungen des Kerntemplates als eine mögliche sinnvolle Zerlegung. Diese können als topologische Substrukturen interpretiert werden, die einen zusammenhängenden topologischen Pfad im Topologie-Baum bilden, der alle gleich priorisierten chemischen Derivatisierungen bündelt. Falls die entsprechenden

funktionalisierten Teilkomponenten als eigenständige Derivate im Testdatensatz enthalten sind, lassen sie entsprechende Rückschlüsse auf die biologische Bedeutung der topologischen Molekülkomponenten, die tatsächliche Pharmakophorgröße und die kleinste noch aktive Verbindung zu. Falls sie fehlen, fallen sie als Templatlücken im Topologie-Baum auf. Dies beruht darauf, dass alle Frameworks (TCCs, Template), die aus den Eingabedaten generiert werden können, als Bestandteile eines gemeinsamen hierarchischen topologischen Strukturbaums für den Gesamteingabedatensatz betrachtet werden, der den chemischen Templatraum und den Grad ihrer jeweiligen chemischen Modifikationen simultan beschreibt. Aufgrund seiner standardisierten Erzeugung ist die Darstellung und Bewertung des Strukturbaums unabhängig vom Belegungsgrad und der Einheitlichkeit der verwendeten Datenquellen und beschreibt damit auch den komplementären Ligandraum der konstitutionellen und funktionellen Strukturlücken. Zur gezielten Handhabung der Lücken muß der Strukturraum aber durch geeignete Ordnungskriterien partitioniert werden.

5.1.5 Priorisierungsrational des Regelwerks

Die Regeln, die zur Priorisierung der Templat-Klassen relativ zueinander verwendet werden, basieren auf folgenden Überlegungen:

Bei der Leitstruktur-Optimierung wird aus synthetischen Gründen meist ein „zentraler“ Core als Templat verwendet. Er wird unverändert gehalten und nur an möglichen Verknüpfungspunkten modifiziert. Bei einer solchen „homologen“ oder kongeneren (lat. con genus = mit gleichem Stamm) Reihe eines Projektdatensatzes wird zuerst dieser bekannte Strukturbestandteil erkannt. Daher ist die Verwendung der Templat-Struktur als Hauptordnungskriterium wünschenswert. Das Identifizieren gemeinsamer Substrukturelemente der Verbindungen kann algorithmisch über einen Maximum Common Substructure (MCS)-Algorithmus[124, 125] erfolgen. Dieser für den Computer relativ aufwendige Vergleich mehrerer Strukturen kann von Experten, mit einer ausgeprägten Fähigkeit zur Mustererkennung visuell leicht vorgenommen werden, allerdings nur solange die Gesamtzahl der Strukturen überschaubar bleibt. Anderenfalls ist der Computereinsatz zwingend erforderlich.

Eine von der Bewertung anderer Strukturen unabhängige Core-Identifizierung läßt sich wie folgt vornehmen: Da der Core als zentrales Element jeder Struktur die meisten Verknüpfungen im Gerüst hat, d.h. entweder mit anderen Ringen direkt anelliert oder über Linker verbunden ist, werden von diesem zentralsten und damit höchstpriorisierten Ring ausgehend alle weiteren Gerüstkomponenten (Ringe, Linker) entsprechend der Verknüpfungen mit abnehmender Priorität als topologische Core-Erweiterung ergänzt. Gibt es zwei oder mehr Ringe mit der gleichen Anzahl an Gerüstverknüpfungen, wird der kleinere bevorzugt (siehe Priorisierungsregeln in Abschnitt 5.1.7). Als drittes Kriterium wird die Summe der Atome der abgehenden Linker verwendet. Der Ring, der die kürzeren Linker trägt, wird priorisiert, da er weniger Freiheitsgrade zur optimalen Präsentation seiner chemischen Dekoration am Target besitzt[126]. Als letzte Möglichkeit wird die Anzahl der Modifikatoren des Rings zur Unterscheidung verwendet. Dies ist die Anzahl der Heteroatome und Substituenten des Rings. Der Ring mit der größeren Zahl an Modifikationen ist priorisiert, da er die größte Chance hat, mit dem Target chemisch zu interagieren.

Die Priorisierung der Gerüstkomponenten kann dazu führen, dass das selbe Ring-Ring-Framework zwei unterschiedlichen Baumknoten zugeordnet wird, je nachdem, an welchem der beiden Ringe die nächste Verknüpfung stattfindet.

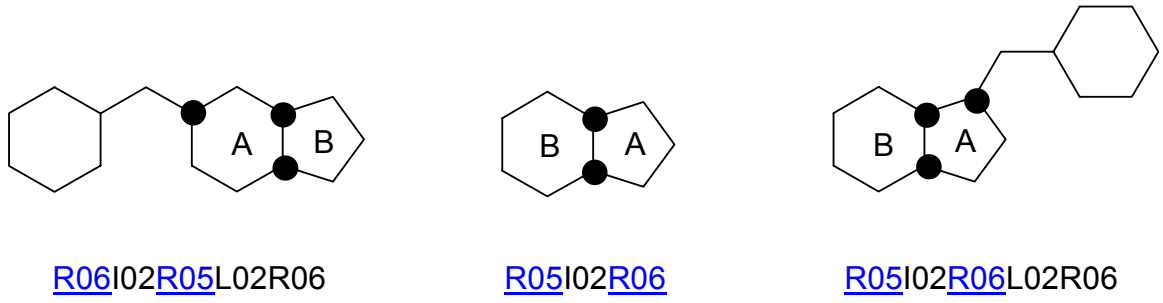


Abb. 5.4: Prioritätswechsel durch Gerüsterweiterung am Beispiel des Indol-Framework R05I02R06 (Mitte), in dem der kleinere Fünfring die höhere Priorität hat. Im Falle einer Gerüsterweiterung am Sechsring (links) erhält dieser die Priorität A und das Indol-Framework den MolCode R06I02R05. Die schwarzen Kreise verdeutlichen die relevanten Gerüstverknüpfungen.

Aus der Position des Framework im Baum ist erkennbar, an welcher Seite die nächste Vergrößerung erfolgen wird. Auf diese Weise wird erreicht, dass immer nur Verbindungen mit vergleichbarem Substitutionsgrad zusammengefasst und verglichen werden.

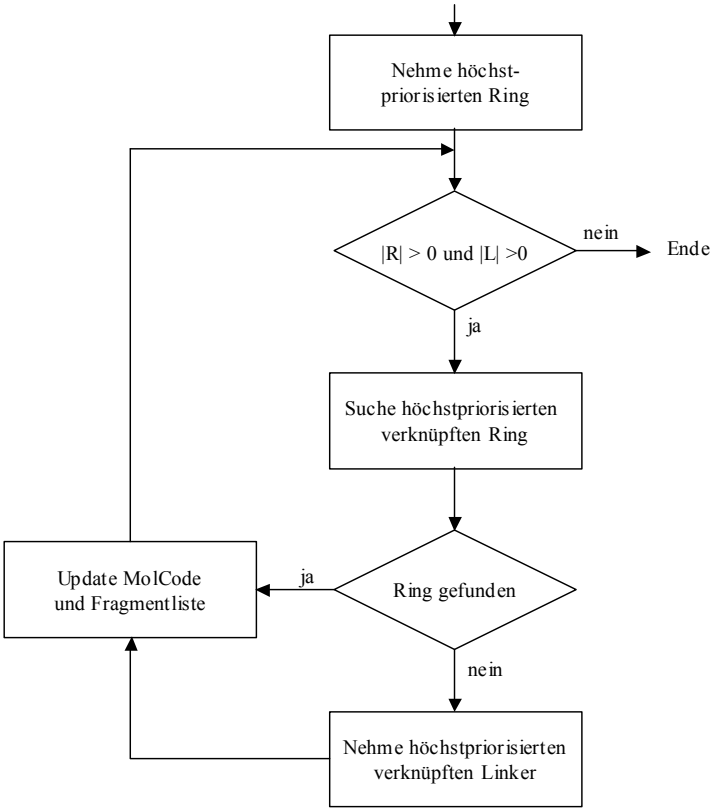


Abb. 5.5: Flowchart zum Aufbau des MolCode aus den Ringen und Linkern mit abnehmender Priorität. |R| und |L| steht für die Mächtigkeit der Menge der Ringe bzw. Linker.

Erfahrungsgemäß werden bei der Leitstrukturauswahl verschiedene Kriterien angewandt. Z.B. sollen die Strukturen nicht allzu komplex sein, d.h. keine sehr großen und schwierig zu

synthetisierenden Ringsysteme enthalten. Favorisiert werden daher kleinere Ringe gegenüber größeren; Dreiringe und Vierringe treten selten auf.

Pharmakologisch werden rigide Strukturen bevorzugt, da sie meist spezifischer wirken und entropisch begünstigt sind. Naturstoffe stellen in dieser Hinsicht ein auf der Evolution basierendes Optimum dar. Sie sind hoch spezifisch und sehr wirksam. Für die Vorhersage der aktiven Konformation von Wirkstoffen liefern sie im Modelling wertvolle Informationen. Dort werden sie als rigides Templat verwendet und die flexibleren Liganden über geeignete Verfahren (SEAL, FAME) angepasst. Ihr großer Nachteil für die pharmazeutische Industrie liegt darin, dass zu ihrer Optimierung zeitaufwendige vielstufige Einzelsynthesen erforderlich sind, die den Optimierungsprozess erschweren und verlangsamen. Nur in den seltensten Fällen lassen sich die gewünschten kombinatorischen Bibliotheken erstellen[127, 128] . Auch die nach der Optimierung erforderliche großtechnische Synthese kann die Verwendung als Medikament aus wirtschaftlichen Gründen verhindern.

5.1.6 Priorisierung der topologischen Komponenten

Für jede topologische Komponente wird die Rangfolge im heuristischen Priorisierungschema wie folgt festgelegt:

(1) Ringe
(2) Linker
(3) Heteroatome
(4) Substituenten
(5) Ketten

Diese Bewertungsschema wird nach Bedarf von oben nach unten für jeden einzelnen molekularen Graphen angewandt.

Diese Reihenfolge versucht, die unterbewusst stattfindende Zerlegung einer Struktur bei der Betrachtung durch einen Chemiker nachzuahmen, der bei neuen und unbekannten Strukturen zuerst die Größe und grobe Form (shape) der Struktur wahrnimmt, die vor allem durch die Ringsysteme vorgegeben wird, dann die Verknüpfungen der Ringsysteme und zuletzt die Heteroatome und Substituenten. Bei nicht unvoreingenommener Betrachtung, z.B. im Hinblick auf ein Projekt, wird möglichweise ein in Erinnerung behaltenes Pharmakophormodell und die zu deren Realisierung erforderlichen Heteroatome oder Substituenten als Interaktionspunkte mit dem Target an die erste Stelle treten.

Aus der Definition der topologischen Hauptmerkmale ergibt sich, dass der Wurzelknoten, d.h. die höchstpriorisierte topologische Klasse, für jedes Molekül entweder ein Ring oder eine Kette in einer azyklischen Verbindung ist. Da die Definition für Linker an die Existenz von terminalen Ringen gebunden ist, ist die Priorisierung der Linker gleichfalls mit den Ringprioritäten verknüpft.

Die Bevorzugung von Ringen gegenüber Linkern und Ketten gleicher Länge beruht darauf, dass die als Targetinformation beobachtete biologische Wirkung einer Verbindung aufgrund einer gleichen chemischen Modifikation in ihrem Informationsgehalt in dieser Reihenfolge genauer determiniert und auswertbar ist. Die Zahl der konformativen Alternativen (Freiheitsgrade) zur bioaktiven Konformation ist bei gegebener Fragmentgröße in den Ringen am kleinsten. Sie schränken somit die geometrische Pharmakophor-Realisierung am stärksten

ein und sollten deshalb besonders klare Signale für die Bewertung ihrer chemischen Dekoration und ihre Sensitivität gegenüber geometrischen Effekten geben. Ein ähnlicher Effekt liegt der Unterscheidung zwischen Heteroatompriorisierung und Substituenten-priorisierung zugrunde.

5.1.7 Priorisierung innerhalb einer topologischen Kategorie

Innerhalb jeder Kategorie erfolgt die Priorisierung durch Anwendung der folgenden Bewertungsregeln in der angegebenen Abfolge:

a) Anzahl der Verknüpfungen in Framework. Linker haben je einen Verknüpfungspunkt an beiden Seiten. Anellierte Ringe haben zwei Verknüpfungspunkte. Ringe mit mehr Frameworkverknüpfungen haben höhere Priorität.

b) Anzahl der Atome (Knoten) in der topologischen Unterklasse bzw. entsprechende Anzahl an Kanten. Bei verzweigten Linkern wird für alle möglichen Kombinationen die Priorität streng nach der Pfadlänge zugewiesen. Kürzeste Pfade und kleinste Ringe haben höchste Priorität.

c) Atomsumme der verknüpften Linker bei gleichgroßen Ringen. Der Ring mit den kürzeren Linkern wird höher priorisiert. Bei Linkern identischer Länge wird der mit dem höherpriorisierten Ring verbundene favorisiert.

d) Anzahl der Modifikatoren, d.h. Anzahl der Heteroatome bzw. Substituenten und Ketten in einer topologischen Unterklasse. Ein höherer Substitutionsgrad führt zu einer höheren Priorisierung.

Die Bewertung der topologischen Gerüste erfolgt durch eine allgemeine Funktion, die die Regeln (1) bis (5) (von Seite 49) und a) bis d) auf einen molekularen Graphen und auf die darin identifizierten topologischen Elemente anwendet. Ist - im äußersten Fall - nach Anwendung aller Regeln keine endgültige Priorisierung möglich, wird eine lokale chemische Identität oder ein Konstitutionsisomer gefunden.

Die Flowcharts in den folgenden Abb. 5.6 und Abb. 5.7. zeigen die Abfolge der Entscheidungen zur relativen Priorisierung zweier Ringe bzw. zweier Linker.

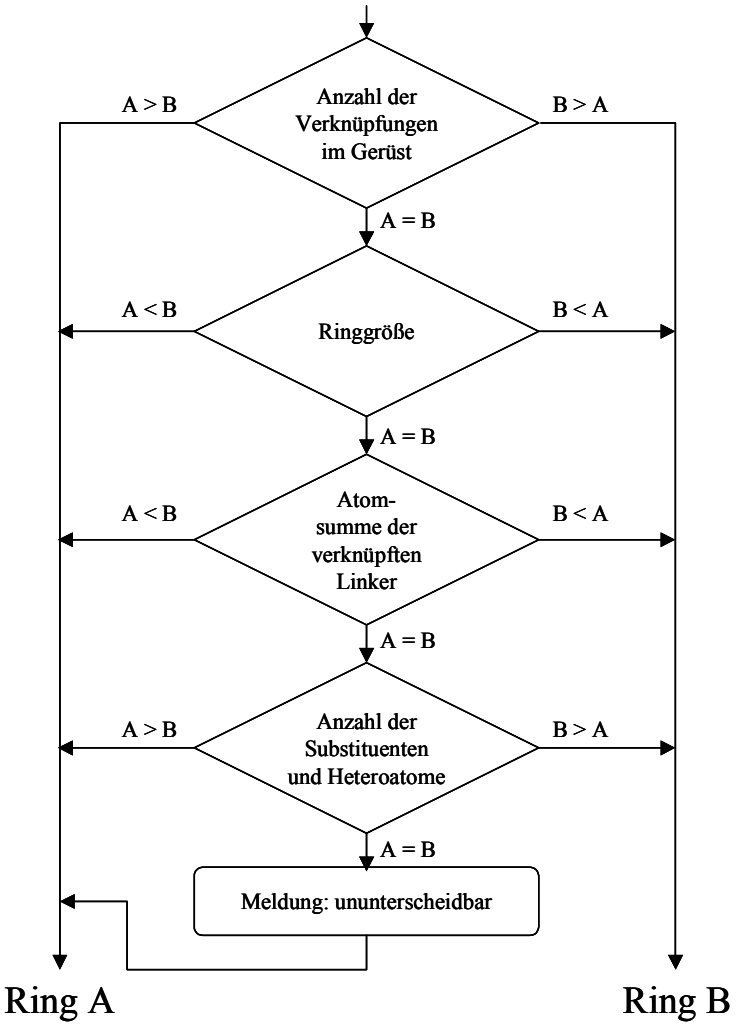


Abb. 5.6: Flowchart zur Priorisierung der Ringe A und B.

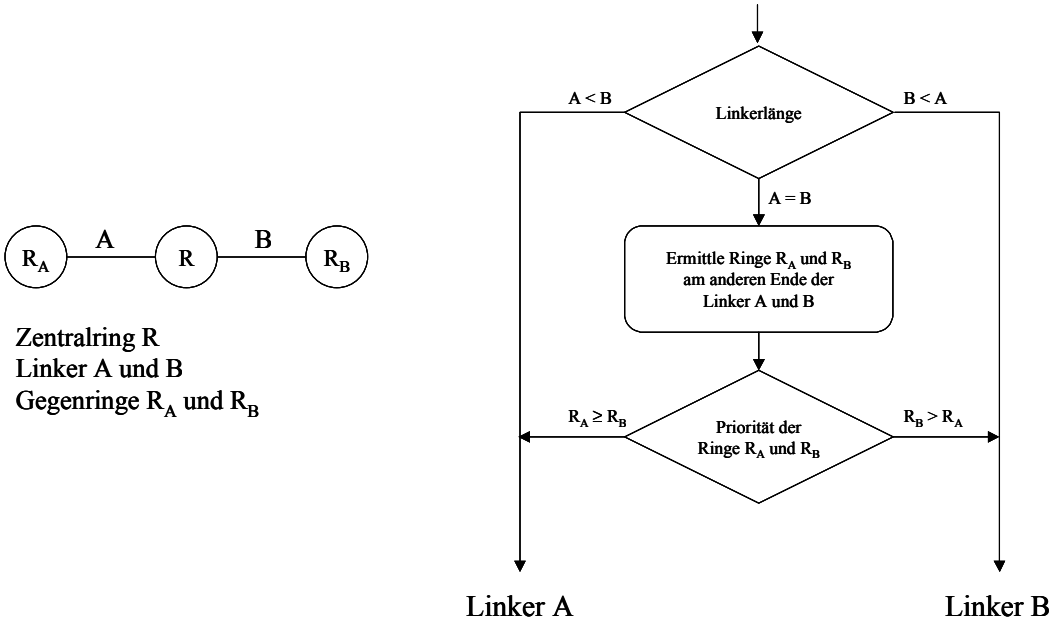


Abb. 5.7: Flowchart zur Priorisierung der Linker A und B.

5.1.8 Erstellung des MolCode

Der MolCode oder Topological Sequence Code (TSC) beschreibt die Verknüpfung und den Typ der vorhandenen Unterklassen des TCC-Graphen (Framework).

Nach der Partitionierung der Struktur werden die Fragmente dem Regelwerk und ihrer Priorisierung entsprechend wieder zusammengesetzt und dabei der MolCode als priorisierter topologischer Sequenzstring erstellt. Das Startfragment ist der Ring mit der höchsten Priorität. Systeme aus mehreren Ringen werden zuerst vervollständigt, dann wird von allen in Frage kommenden Linkern der höchstpriorisierte bearbeitet. Dadurch werden alle weiteren Fragmente rekursiv von innen nach außen wieder verknüpft und die Reihenfolge im MolCode festgehalten. Im Konnektivitäts-Modul K wird zusätzlich gespeichert, zu welchem Fragment die Verknüpfung tatsächlich stattgefunden hat. Dies ist vor allem in Hinblick auf eine Enumerierung von Strukturvarianten, d.h. die Retransformation eines MolCode in eine Struktur, bei der Lückenanalyse von Bedeutung (siehe Abschnitt 5.2.12). Jedes Modul des MolCode, das einem Fragment entspricht, erhält in abnehmender Priorität einen Kleinbuchstaben zugeordnet. Anders formuliert: alle Ringe und Linker erhalten von links nach rechts im MolCode einem ihrer Position entsprechenden Kleinbuchstaben. Jeder Buchstabe im K-Modul gibt der Position entsprechend an, von welchem Fragment die Verknüpfung ausgeht. Für das erste Modul a ist diese Information nicht erforderlich und wird daher weggelassen. Der Buchstabe an erster Position entspricht daher dem Modul b und dieser ist immer mit Modul a verknüpft.

Der MolCode besteht aus mehreren Modulen, die jeweils mit einem Grossbuchstaben beginnen und daran anschliessend in der Regel zwei Stellen für eine genauere Spezifizierung enthalten. Die verwendeten Module und der entsprechende Grossbuchstaben sind in folgender Tabelle beschrieben:

Modul	**Beschreibung**
Rrr	Ring mit der Größe rr (Atomanzahl)
Lll	Linker mit der Länge ll (Bindungsanzahl)
Iii	Schnittmenge (Intersection) ii zweier Ringe
Hhh	Gesamtanzahl hh der Heteroatome
Sss	Summe ss aller Substituentenatome
Kk...k	Konnektivitäten k...k der Gerüstfragmente (siehe Text)

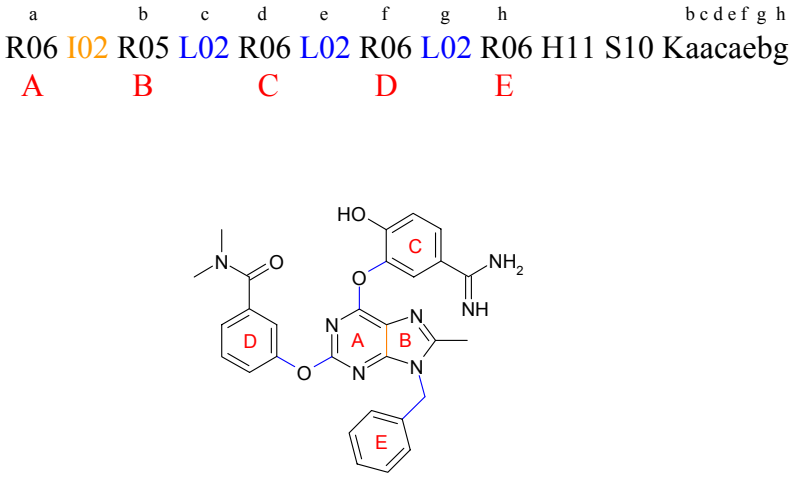


Abb. 5.8: Exemplarische Beschreibung der MolCode-Komponenten.

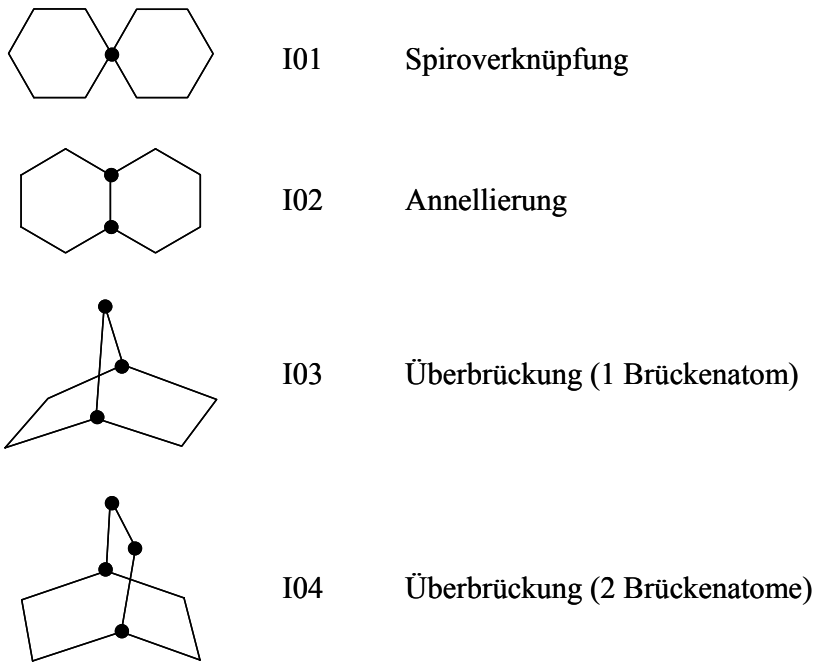


Abb. 5.9: Erläuterungen zum Intersection-Modul I.

Die klassifizierte Verbindung wird mit dem TCC verknüpft und stellt einen speziell dekorierten Repräsentanten, d.h. ein Beispiel für chemische Derivatisierung dar. Unterhalb einer TCC-Struktur werden alle existierenden bzw. im Datensatz vorhandenen chemischen Derivate des Framework zusammengefasst, solange sie mit den Priorisierungsregeln der Fragmente im Einklang sind. Damit ist sichergestellt, dass zwischen Core-Fragment und TCC eine erwünschte hierarchische Informationsverdichtung stattfindet, die als Baumstruktur darstellbar ist und gleichzeitig als Ordnungskriterium für darunter zusammengefasste chemische Derivate dient.

Kleine rigide Strukturen, d.h. Ringe mittlerer Größe mit Fünfringen oder Sechsringen und möglichst kurzen Linkern, werden in der Forschung (siehe Abschnitt 5.1.5) bevorzugt. Da

sich diese Bevorzugung aufgrund des Regelwerks im MolCode widerspiegelt, kann für die Reihenfolge der Tochterknoten (subnodes) die Sortierreihenfolge des MolCode verwendet werden. Daraus folgt, dass die Sortierreihenfolge als Ordnungskriterium bei der Anordung der Strukturen im topologischen Baum berücksichtigt werden kann. Ringgröße und Linkerlänge nehmen deshalb im Baum von links nach recht zu.

5.1.9 Generierung der topologischen Referenzstruktur

Sobald alle vorhandenen topologischen Klassen identifiziert worden sind, wird das oben erwähnte Priorisierungsschema so lange rekursiv auf sie angewendet, bis alle Molekülkomponenten priorisiert sind. Die Priorisierung der topologischen Fragmente definiert einerseits den topologischen Namen (MolCode, TSC) der daraus abgeleiteten Referenzstruktur (TCC) und läßt sich andererseits als hierarchisch geordnete Sequenz topologischer Strukturschalen um das topologische Core-Element herum interpretieren, die später als zusammenhängender Pfad im Topologiebaum sichtbar werden (TSP).

Da das topologische Scaffold (der molekulare Graph ohne Substituenten) zu diesem Zeitpunkt noch die Heteroatome in den Ringen und Linkern enthält, werden diese in Kohlenstoffatome verwandelt, um das Referenzframework des TCC-Graphen zu erhalten. Zu diesem Zweck wird ein globaler Morphingoperator $\hat{M}$ als ein Sonderfall für einen allgemeinen chemischen Transformationsoperator $\hat{T}_p(\mathbf{V}_p)$ definiert. Dieser erzeugt ein topologisch äquivalentes Kohlenstoffanalogon $\mathbf{T}_i^C$, wenn er auf einen molekularen Graphen **G** oder eine topologische Substruktur $\mathbf{T}_i$ angewendet wird. Jede topologische Unterklasse T_i des TCC kann gezielt modifiziert werden, indem der Operator $\hat{T}_p(\mathbf{V}_p)$ zur Umwandlung einer Position p in eine neue Gruppe $\mathbf{V}_p$ (funktionelle Gruppe oder Heteroatom) angewendet wird.

Eine solche allgemeine Transformation kann eine der folgenden vier Basisoperatoren enthalten:

- Identitätsoperator ($\hat{I}$), der das Fragment unverändert lässt
- Morphingoperator ($\hat{M}$), der ein Atom p verändert
- Gruppenoperator ($\hat{O}_+$), der eine Menge von Atomen $\mathbf{V}_p$ hinzufügt
- Gruppenoperator ($\hat{O}_-$), der eine Menge von Atomen $\mathbf{V}_p$ entfernt

Zur Erzeugung des TCC-Framework kommt der Morphingoperator $\hat{M}$ zum Einsatz, der einzelne Kohlenstoffatome mit geeigneter Wertigkeit erzeugt.

$$\hat{T}_p(\mathbf{V}_p) \in \left\{\hat{I}, \hat{M}, \hat{O}_+, \hat{O}_-\right\}$$

$$\hat{M} := \hat{M}_p(\{C\}) = \hat{T}_p(\mathbf{V}_p) \quad \text{für alle Positionen p}$$

$$\mathbf{T}(i) \xrightarrow{\hat{M}} \mathbf{T}^C(i) \quad \text{bzw.} \quad \mathbf{T}^C(i) = \hat{M} \cdot \mathbf{T}(i)$$

$$\mathbf{G}(i) \xrightarrow{\hat{M}} \mathbf{TCC}(i) \quad \text{bzw.} \quad \mathbf{TCC}(i) = \mathbf{G}^C(i) = \hat{M} \cdot \mathbf{G}(i)$$

$\mathbf{T}_i$ und $\mathbf{T}_i^C$ sind die Mengen aller topologischen Klassen oder deren Kohlenstoffanaloga. Das Framework **TCC**(i) des Graphen **G**(i) ist das Ergebnis des Morphingvorgangs, angewendet auf das Scaffold **SF**(i), das durch Entfernen der Mengen **S**(i) und **C**(i) im Graphen **G**(i) entsteht.

$$\mathbf{SF}(i) = (\mathbf{G}(i) \setminus \mathbf{S}(i)) \setminus \mathbf{C}(i)$$
$$\mathbf{TCC}(i) := \hat{M}(\mathbf{SF}(i))$$

5.1.10 Aufbau des topologischen Strukturbaums

Die TCC-Unterbäume aller analysierten Verbindungen werden in einem hierarchischen Topological Structure Tree (TST) zusammengefasst, der von oben nach unten einen zunehmenden Detailgrad bzgl. der topologischen Klassifizierung der Unterstrukturelemente aufweist. Das höchstpriorisierte Unterstrukturelement (ein Ring) stellt den Wurzelknoten eines Topological Sequence Path (TSP) dar, der ein gültiger Pfad im TST ist.

$$\mathbf{T}_m(i) = \max\left[score(\mathbf{R}_1(i)), score(\mathbf{L}_1(i))\right]$$
$$\mathbf{T}_m(i) \in \left\{\mathbf{R}_1(i), \mathbf{C}_1(i)\right\}$$
$$\mathbf{TSP-root}(i) := \hat{M}(\mathbf{T}_m(i))$$

Die Funktion max(score(), score()) ermittelt die topologische Root-Komponente, die den höchsten Rang entsprechend den Regeln (1) bis (5) und a) bis d) hat. Ausgehend von dem oben liegenden Wurzelknoten des TST, der den höchstpriorisierten, d.h. am höchsten funktionalisierten und kleinsten Ring der Verbindung enthält, werden in den darauffolgenden Schalen der topologischen Ordnung die folgenden Elemente der gleichen oder der nächsten Unterklasse zur Charakterisierung verwendet. Ausgehend vom obersten Knoten mit der Bezeichnung R6 ergibt zum Beispiel im nächsten Wirkungskreis der topologischen Verknüpfung (sphere of topological linkage) R6L1 den neuen TST-Knoten R6L1R6, der mit dem darüberliegenden R6-Knoten verknüpft wird. Jeder neue Knoten weist einen höheren Detailgrad auf. Dieser Vorgang wird so oft wiederholt, bis alle topologischen Klassen abgearbeitet und zugewiesen worden sind und damit der TCC-Knoten des jeweiligen Moleküls erreicht ist; dann ist die Verbindung lokalisiert.

$$\mathbf{TSP}_{j+2} = \mathbf{TSP-root} \bigcup \hat{M}(\mathbf{T}_{j+1}(i))$$
$$j = 1,...,(t-2), j \neq m, t = r + l$$
$$\mathbf{T}_{j+1} \in (\left\{\mathbf{R}, \mathbf{R}x\mathbf{L}\right\} \setminus \mathbf{TSP-root}(i)) \setminus \mathbf{T}_j$$
$$score(\mathbf{T}_{j+1}) \leq score(\mathbf{T}_j)$$

Die Elemente der topologischen Menge **T** jeder Verbindung werden dazu verwendet, den ursprünglichen molekularen Graphen **G**(i) durch topologische Partitionierung auf einen topologischen Sequenzpfad TSP im Strukturbaum des Gesamtdatensatzes abzubilden. Dabei beschreibt die Rangordnung für die topologischen Komponenten des TCC das Templatwachstum durch Kantenverknüpfung der entsprechenden Substrukturen im Strukturbaum, beginnend mit dem Core-Templat und endend im TCC. Der hierarchische Aufbau der Baumstruktur spiegelt sich in der analogen sequentiellen Folge der priorisierten

Topologie-Komponenten wider, wobei die topologischen Klassenbezeichnungen der verknüpften Strukturelemente zu einem topologischen Strukturcode vereinigt werden. So wird ein eindeutiges lexigraphisches Identifizierungsmerkmal für jeden Knoten im TST erzeugt. Diese Merkmale können für verschiedene Eingabedatensätze verwendet werden, um die Schnittmengen der TSPs bzw. des gesamten TSF zu überprüfen. Zwei Moleküle *i* und *o* können z.B. nur eine nichtleere Schnittmenge haben, wenn sie mindestens eine gemeinsame TSP-Wurzel haben.

$$\mathbf{I} := \mathbf{TSP}(i) \cap \mathbf{TSP}(o)$$

Die Schnittmenge zweier Knoten kann durch einfachen lexikalischen Vergleich ihrer TSPs gefunden werden. Beispielsweise haben R6L2R6 und R6L2R6L1R6 einen gemeinsamen Wurzelknoten R6 bzw. sogar die gemeinsame topologische Sequenz R6L2R6 und daher gemeinsame Abschnitte im TST. Zusätzliche Verbindungen des zu analysierenden Datensatzes werden auf die gleiche Weise bearbeitet. Dadurch werden neue Knoten erzeugt oder es werden Knoten von bereits analysierten Molekülen verwendet. Die Verknüpfungen zu zusätzlichen Unterknoten im TST erscheinen auf der Ebene, auf der erste Unterschiede in der Priorisierung oder in den strukturellen Modifikationen auftreten. In Ausnahmefällen passiert dies erst auf der Ebene der TCC. Dies bedeutet, dass verschiedene Funktionalisierungen des gleichen Framework identifiziert worden sind. Dies ist ein für die SAR-Analyse von Aktiv- und Inaktivlisten erwünschtes Verhalten.

Für jedes Modul bzw. jede Komponente des MolCode wird ein Knoten erzeugt. Dadurch wird über dem topologischen Alphabet ein dynamischer Strukturraum durch die Sequenz der Knoten definiert. Es ist kein chemischer Raum mit einer limitierten Anzahl an Dimensionen vorgegeben.

Auf der Suche nach Schnittmengen könnten anstelle des lexikalischen Vergleichs auch andere Techniken wie edit distance, clique-detection, maximum common substructure search oder fingerprint screening eingesetzt werden.

Enthalten die in den Eingabedaten gefundenen Frameworks keine gemeinsamen Subklassenelemente, werden die einzelnen (Teil-)Bäume durch einen nicht sichtbaren leeren Hauptknoten zu einem einzigen Gesamtbaum zusammengeführt.

5.1.11 Datenanalyse basierend auf den TST-Knoten

Beim Bioprofiling enthalten zusätzliche Datenfelder Aktivitätsdaten zu allen Testsystemen, in denen dieses Templat sich als aktiv und damit als privilegiert erwiesen hat. Um die Anreicherungsfaktoren (hit enrichment) zu überwachen, werden die Datenfelder mit dem eigentlichen molekularen Graphen verknüpft, der als Blatt dem TCC-Knoten im TST zugeordnet ist.

Basierend auf diesen Informationen, können die folgenden Aufgaben effizient durchgeführt werden:

- SAR-Profiling für topologische Scaffolds durch R-Gruppen-Dekonvolution für die Aktiven und Inaktiven
- Framework-basierte Wahrscheinlichkeitsanalyse für Aktivität

- Überprüfung falsch Positiver und falsch Negativer durch Boolsche Operationen von TSTs, die durch unterschiedliche Eingabefilter erzeugt worden sind
- Lückenanalyse in aktiven Templatklassen, Screening-Bibliotheken, Prüfpräparatelagern und privilegierten Scaffolds über die HTS-Historie bzw. beim Substanzeinkauf
- Diskriminanzanalyse für Aktivität oder physikalische Eigenschaften basierend auf Grapheninvarianten der chemischen Struktur
- Berechnung der chemischen Distanz zwischen TST-Knoten mittels der Mahalanobis-Distanzmetrik
- Einbeziehung von Patentstrukturen und SARs
- Auswahl von targetspezifischen und strukturdiversen topologischen und funktionalen Prototypen für 3D-Alignment und mechanistische Analyse der Drug-Target-Interaktion
- vergleichende Analyse von Bioeffektor-Datenbanken und inhouse-Frameworks von screening hits (indirekte Targetanalyse).

5.1.12 Priorisierung der chemischen Dekoration

Die Strukturen unterhalb eines TCC-Knotens im TST können auch durch von der Struktur abgeleitete, d.h. berechnete Deskriptoren, charakterisiert werden. Diese können benutzt werden, um

- die „Chemische Distanz“ zu dem TCC-Framework, dem (virtuellen) Cluster-Zentrum, zu messen und die Derivate, basierend auf diesem Abstand, zu sortieren
- die chemischen Modifikationen eines TCC im Hinblick auf Aktivität zu unterscheiden
- die Deskriptoren mit physikalischen Eigenschaften oder Aktivität zu korrelieren.

Als geeignete Deskriptoren zur Berechnung der „Chemischen Distanz“ innerhalb eines Clusters oder zwischen TST-Knoten sind die spektralen Momente (siehe Abschnitt 5.2.9) des sog. Line-Graphen in der Literatur als geeignet beschrieben worden. Dieses Verfahren erzeugt nicht nur linear unabhängige strukturbasierte Deskriptoren, sondern unterscheidet innerhalb einer Diskriminanzanalyse auch zwischen strukturellen Modifikationen, die die Aktivität oder Inaktivität beeinflussen.

5.1.13 Vergleich von Aktiv-TST und Inaktiv-TST

In den topologischen Bäumen der Aktiven und Inaktiven eines gegebenen Tests können aufgrund der strukturchemischen Bedeutung des Topological Sequence Code (TSC) einander entsprechende topologische Gruppen leicht über ihre identischen Knotenbezeichnungen identifiziert werden. Dadurch kann der Effekt, den chemische Modifikationen auf Aktivität bzw. Inaktivität in einem Assay haben, für vergleichbare topologische Frameworks beobachtet und interpretiert werden. Zusätzlich können weitere Analysen durchgeführt werden, indem die berechneten Deskriptoren der Verbindungen dieser Gruppen verglichen oder die vorhandenen Substituenten und Heteroatome weiter nach chemischen Gesichtspunkten, z.B. durch Klassifizierung in HB-Donatoren oder -Akzeptoren bzw. ionisierbare saure oder basische Gruppen, unterteilt werden. Auf diese Weise werden in den Gruppen der Aktiven und Inaktiven diejenigen Partner gefunden, die zusätzlich zu dem gleichen topologischen Framework auch die meisten funktionellen chemischen Merkmale gemeinsam haben (Pharmakophoranalyse). Hingegen sind in diesen Gruppen bei

gegensätzlicher Wirkklassifizierung der Mitglieder je nach tatsächlicher Wahrscheinlichkeitsverteilung der Aktiven und Inaktiven voraussichtlich Kandidaten für falsch Positive oder falsch Negative in einem Test. Durch die Analyse der sich entsprechenden TCCs beider Gruppen werden die Verbindungen, die für eine Nachtestung vorgesehen werden sollten, festgelegt. Die Ergebnisse der Nachtestung bestätigen entweder die Zuordnung zu Aktiv und Inaktiv oder sie korrigieren die Klassifizierung. Wird diese Ähnlichkeitsanalyse wiederholt, erhält man Informationen über die wahrscheinlichen Pharmakophorelemente sowie eine R-Gruppen-Dekonvolution zur Beschreibung der SAR der Verbindungen eines TCC.

Die abschließende Analyse der aktiven Drugs und ihrer Pharmakophore erfolgt durch vertiefende numerische Untersuchung, z.B. Lineare Diskriminanzanalyse mit den spektralen Momenten, klassisches QSAR oder 3D-Strukturalignment diverser Liganden mit Pharmakophoranalyse. Dazu werden die Verbindungen und ihre Fragmentierungen relativ zu den Aktiv/Inaktiv-Kategorien des Trainingsdatensatzes bewertet.

5.2 Verwendete Verfahren und Algorithmen

5.2.1 Ring Perception

Zur Zerlegung jeder Verbindung in ihre topologischen Komponenten ist im ersten Schritt die Detektion der Ringe erforderlich[129]. Die beiden graphentheoretischen Grundalgorithmen, Ringe in Graphen zu detektieren, sind Tiefensuche (depth-first search) und Breitensuche (breadth-first search)[130, 131]. Die Tiefensuche erfordert weniger Speicher und ist im allgemeinen besser geeignet, alle Ringe zu finden, als die Breitensuche. Die Breitensuche ist schneller im Finden von kleinen Ringen. Da die Pfade während der Suche gespeichert werden, ist die korrekte Abfolge der Knoten bekannt, wenn ein Ring gefunden ist. Den aktuellsten Algorithmus auf der Basis von Graphentraversierung beschreibt Figueras[132].

Einen anderen Ansatz verfolgen Graphenreduktionsalgorithmen. Sie können in einem Vorbereitungsschritt zur Vereinfachung des Graphen (homeomorphic reduction)[133] für die Suchalgorithmen oder direkt zur Ringdetektion verwendet werden. Der aktuellste Algorithmus dieser Art ist von Hanser et al. beschrieben worden, die den Graphenreduktionsalgorithmus auf einen aus dem molekularen Graphen (M-Graphen) abgeleiteten Pfadgraphen (P-Graphen) anwenden[134]. Das Verfahren ist einfach zu implementieren und für chemische Graphen mit einer maximalen Bindungsordnung von vier sehr schnell. Der M-Graph wird in einen Pfadgraphen konvertiert, der anfangs die gleichen Kanten und Knoten wie der originale Graph besitzt, aber jede Kante trägt ein eindeutiges Label, das aus den Namen der verknüpften Knoten besteht. Aus dem Pfadgraphen werden iterativ alle Knoten entfernt, bis keiner mehr übrigbleibt, und bei jeder Iteration werden die Pfade der relevanten Labels aktualisiert. Beim Entfernen des Knotens b werden die beiden Kanten [a-b] und [b-c] gelöscht und eine neue Kante [a-b-c] erstellt (siehe Abb. 5.10). Das Label der neuen Kante [a-b-c] ist die Verknüpfung der Label von [a-b] und [b-c]. Jeder Knoten, der letztendlich mit sich selber verknüpft wird, entspricht einem Ring im ursprünglichen M-Graphen, wobei das Label der selbstverknüpften Kante (self-loop) die Knoten des Rings enthält. Aus Effizienzgründen werden die Knoten mit der geringsten Konnektivität zuerst aus dem Pfadgraphen entfernt.

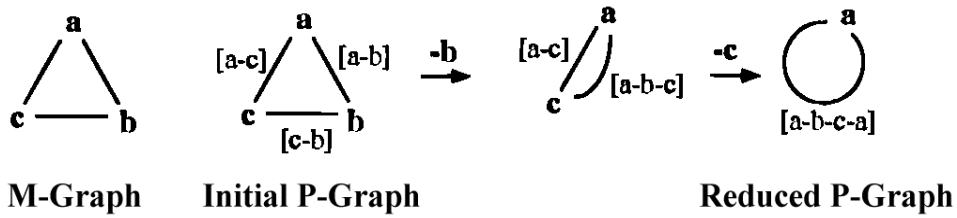


Abb. 5.10: Erzeugung des reduzierten P-Graphen.

Die BayTree-Implementierung des Algorithmus benutzt – abweichend von der Beschreibung von Hanser et al. – als Datenstruktur keine unsortierten Mengen (sets), sondern Listen, die die Reihenfolge der Knoten beibehalten. Dies ist für die Identifizierung der Ankeratome der Linker ebenso erforderlich wie für zukünftige Erweiterungen auf der Basis der Pfadabstände von Heteroatomen.

```
Paths := Kanten (Edges) des Graphen
Nodes := Knoten des Graphen mit steigender Konnektivität

Rings := ∅
Paths3 := ∅

FOR EACH n ∈ Nodes
     Paths2 := ∅
     FOR EACH p ∈ Paths
          IF p enthält Knoten n
               Ergänze p in Paths2
               Lösche p aus Paths
               IF Konnektivität des Anfangs- und Endknoten von p ≥ 3
                    Ergänze p in Paths3

     FOR i = 1 TO |Paths2|
          pi := Pfad i aus Paths2
          FOR j = i+1 TO |Paths2|
               pj := Pfad j aus Paths2
               p := Verknüpfung von pi und pj an Knoten n
               isect := Anzahl gemeinsamer Knoten in pi und pj
               IF isect = 1
                    Ergänze p in Paths
               ELSE IF isect =2 und Anfangs- und Endknoten von p identisch
                    Ergänze p in Rings


Rings: Enthält alle Ringe des Graphen
Paths3: Enthält alle Pfade des Graphen, die Knoten mit der Konnektivität >=
3 verknüpfen
```

Listing 5.1: Pseudocode der P-Graph-Ring Perception.

5.2.2 Repräsentative molekulare Ringsets

Die Menge aller möglichen einfachen Ringe ist einzigartig für jede gegebene Struktur[135], enthält aber üblicherweise weitaus mehr Ringe, als zur Beschreibung des Ringsystems erforderlich sind.

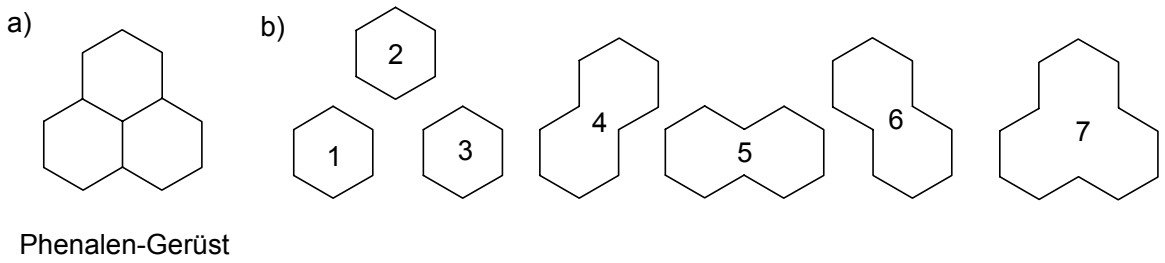


Abb. 5.11: Menge aller einfachen Ringe b), die im Phenalen-Gerüst a) enthalten sind.

Daher gibt es unterschiedliche Definitionen für Ringuntermengen. Die wichtigste und am häufigsten verwendete ist das SSSR (Smallest Set of Smallest Rings). Es enthält die Grundmenge (basis set) der kleinsten Ringe der Struktur. Die Frerejacque-Zahl N_R, d.h. die Anzahl der kleinsten Ringe im SSSR, auch als cyclomatic complexity bezeichnet, kann über die Cauchy-Formel[136] berechnet werden:

$$N_R = N_E - N_V + N_C$$

N_E ist die Anzahl der Kanten, N_V die Anzahl der Knoten und N_C die Anzahl der nicht zusammenhängenden Komponenten des Graphen. Alle größeren Ringe, die durch „Linearkombination“[137] der kleineren beschrieben werden können, werden nicht aufgeführt. Das Phenalen-Gerüst in Abb. 5.11 enthält insgesamt sieben Ringe (siehe Nummerierung), das SSSR wird nur aus den Ringen 1-3 gebildet.

```
Rings := Menge aller einfachen Ringe eines Graphen

FOR EACH Ring_i ∈ Rings
     Menge A := ∅
     FOR EACH Ring_j ∈ Rings
          IF (|Ring_j| ≤ |Ring_i|) und (Ring_j ≠ Ring_i)
               Erweitere Menge A um Atome von Ring_j
     IF alle Atome von Ring_i in Menge A enthalten
          Lösche Ring_i aus der Menge Rings

Rings: enthält nur noch die Ringe des SSSR
```

Listing 5.2: Pseudocode der SSSR-Reduktion.

Da das erste Kriterium beim Erstellen des SSSR die Größe der Ringe ist, gibt es einige Fälle, in denen es zu unerwarteten Resultaten kommt. In dem Graphen in Abb. 5.12 wird beispielsweise statt des erforderlichen Neunrings der hervorgehobene Achtring gefunden. Zusätzlich ist die Menge der Ringe nicht eindeutig bestimmt, wenn es in speziellen komplexen Graphen mehrere symmetrieäquivalente Ringe gibt.

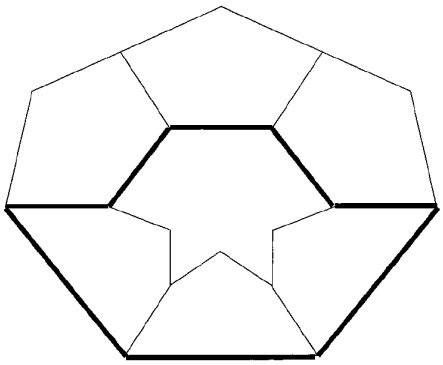
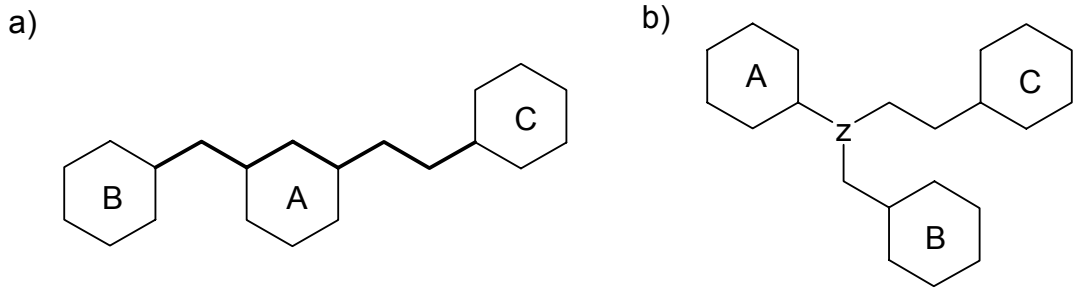


Abb. 5.12: Der SSSR des Graphen enthält statt des erwarteten zentralen Achtrings den hervorgehobenen Neunring.

Andere Ringuntermengen sind Essential Set of Essential Rings (ESER)[138], eine eindeutige Erweiterung des SSSR-Konzepts und Extended Set of Smallest Rings (ESSR)[139].

5.2.3 Generierung der molekularen Linker

Bei der Ringdetektion durch den P-Graph-Algorithmus werden zusätzlich alle Pfade zwischen Knoten mit einer Konnektivität größer als zwei erzeugt und abgespeichert. Basierend auf diesen Daten, wird in der Prozedur zur Linker-Perception die Menge der Linker erzeugt. Im ersten Schritt werden alle Pfade aussortiert, die mehr als ein Atom eines einzelnen Rings enthalten. Der in Abb. 5.13 a) hervorgehobene Pfad enthält drei Atome von Ring A und ist daher kein Linker. Gibt es in dem Graphen den Sonderfall von verzweigten Linkern, so müssen die vorhandenen Teilpfade zwischen dem Linkerzentrum Z und den Ringen zusammengesetzt werden. Dazu wird das Linkerzentrum eliminiert und es werden alle möglichen Kombinationen der Teilpfade erzeugt. Für den Graphen in Abb. 5.13 b) sind die vorhandenen Ausgangspfade und die daraus erzeugten Linker in Abb. 5.13 c) aufgeführt. Abschliessend werden alle mehrfach vorhandenen Linker auf einen einzelnen reduziert. Damit ist die gewünschte Menge der Linker des Graphen ermittelt.



c)

Teilpfade	Pfadlänge
A-Z	1
B-Z	2
C-Z	3

→

Linker	Linkerlänge
A-B	L03
A-C	L04
B-C	L05

Abb. 5.13: Veranschaulichung der Sonderfälle bei der Linker-Perception.

```
Ringnodes := Menge aller Knoten die Bestandteil eines Rings sind

Paths3 := alle Pfade des Graphen, die Knoten mit der Konnektivität >=3
verknüpfen
Rings := Ringe des Graphen

Linker := ∅
Linker_lz := ∅
Linkerzentren := ∅

FOR EACH p ∈ Paths3

     IF p mit keinem r ∈ Rings mehr als einen gemeinsamen Knoten besitzt

          IF Anfangsatom a := p[0] von p kein Ringatom ist
               Ergänze a in Linkerzentren
               Ergänze p in Linker_lz

          ELSE IF Endatom a := p[end] von p kein Ringatom ist
               Ergänze a in Linkerzentren
               Ergänze p in Linker_lz

          ELSE
               Ergänze p in Linker

FOR EACH lz ∈ unique(Linkerzentren)

     FOR EACH Tupel li und lj aus Linker_lz
          IF lz sowohl in li als auch lj enthalten
               Verknüpfe li und lj an Knoten lz zu p
               Ergänze erzeugten Gesamtpfad p in Linker

FOR EACH Tupel li und lj aus Linker
     IF lj vollständig in li enthalten
          Entferne lj aus Linker

Linker: Menge aller Linker des Graphen
```

Listing 5.3: Pseudocode der Linker Perception.

Bei Verwendung eines anderen Algorithmus zur Ringdetektion, welcher die paths3-Daten nicht erzeugt, müssen die Linkerpfade durch Traversierung gefunden werden. Ist die Kantenliste des Framework gegeben, sind folgende vier Schritte erforderlich:

1. Aussortieren oder Kennzeichnen aller Ringbindungen, d.h. aller Kanten, deren Anfangs- und Endknoten Ringatome sind.
2. Kennzeichnen aller Ankeratome, d.h. aller Ringatome, deren Konnektivität größer oder gleich 3 ist und die ein Nicht-Ringatom als Bindungspartner haben.
3. Ausgehend von jedem Ankeratom, rekursive Verknüpfung aller gegebenen Kanten zu Pfaden, bis ein zweites Ankeratom erreicht ist.
4. Aussortieren aller mehrfach vorhandenen Linker.

5.2.4 Berechnung von 2D-Koordinaten

Die Erstellung von Strukturdiagrammen (SDG, Structure Diagram Generation) ist der Vorgang, bei dem aus den Konnektivitäten eines Moleküls die 2D-Koordinaten der Atome erzeugt werden. Diese sind erforderlich, um eine konventionelle Strukturzeichnung zur Visualisierung erstellen zu können. Die Berechnung von 3D-Koordinaten zu einer Struktur für die Weiterverarbeitung durch den Computer ist eine andere Fragestellung. Dafür stehen die beiden kommerziellen Programme CORINA[140] und CONCORD[141, 142] zur Verfügung.

SDG tritt in verschiedenen Situationen der Datenauswertung auf:

- bei der Übersetzung von Linearnotationen (SMILES, SLN)
- beim Retrieval aus Datenbanken (Tripos Unity bietet erst in der neuesten Version die Möglichkeit, 2D-Koordinaten abzuspeichern)
- bei der Nachbearbeitung (clean-up) von manuell skizzierten Strukturen in einem Strukturzeichenprogramm, z.B. MDL ISIS/Draw oder CambridgeSoft ChemDraw
- bei der Atomlabel-Expansion, z.B. ausgehend von $PhO(CH_2)_3iPr$
- bei der Strukturenumerierung, z.B. bei der Erstellung einer kombinatorischen Bibliothek, wobei die R-Gruppen einer Markushformel durch konkrete Substituenten ersetzt werden
- zur übersichtlicheren Darstellung von 3D-Strukturen.

Grundsätzlich werden überlappende Atome und überkreuzte Bindungen vermieden. Darüber hinaus gibt es einige informelle historische Konventionen, die von Chemikern beim manuellen Erstellen von Strukturzeichnungen intuitiv beachtet werden:

- Bindungslängen und Bindungswinkel werden möglichst einheitlich gehalten.
- Kleine bis mittelgroße Ringe werden als reguläre Polygone gezeichnet.
- Große Ringe und Naturprodukte wie Steroide, Zucker und Alkaloide haben eine spezielle Form und Orientierung.
- Ringsysteme werden bevorzugt horizontal ausgerichtet.
- Ketten werden im Zickzack und bevorzugt horizontal gezeichnet.
- Möglichst viele Bindungen werden entlang der Koordinatenachsen ausgerichtet (Vielfache von 30° und 45°).
- Heteroatome in Heterozyklen werden in der Regel in der Reihenfolge oben, unten, rechts eingezeichnet.
- Ringsubstituenten zeigen vornehmlich nach oben oder nach rechts.

Bei automatisch generierten Abbildungen wird die Berücksichtigung dieser Regeln erwartet. Für die Nutzungsbereitschaft der Anwender und zur Vermeidung von Mißverständnissen spielt es eine große Rolle, dass die Abbildungen als „ansprechend und richtig“ akzeptiert werden.

Grundsätzlich gibt es zwei Ansätze für SDG-Algorithmen. Entweder werden die gewünschten Strukturen aus vorgegebenen abgespeicherten Templaten (Ringe, Ringsysteme, Ketten) zusammengesetzt oder sie werden mit Hilfe einer geeigneten empirischen Zielfunktion und eines Strafterms für die Distanzmaximierung der Bindungen durch ein Optimierungsverfahren berechnet, wie z.B.

$$\min(F) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_{ij}^{-2} + k \sum_{i=1}^{n'-1} \sum_{j=i+1}^{n'} (d_{ij} - 1.0)$$

d_{ij} ist der kartesische Abstand zwischen den Atomen i und j. Der zweite Term ist ein hochgewichteter Strafterm für verlängerte Bindungen in mehrzyklischen Systemen, die von der Standardbindungslänge abweichen. In den Verfahren müssen zusätzlich verschiedene Spezialfälle berücksichtigt werden, z.B. Strukturen, die aus mehreren Komponenten (Gegenionen) bestehen. Meistens werden in einem Nachbearbeitungschritt die Molekülorientierung und die Bindungausrichtung korrigiert. Weitergehende Informationen sind dem Review von Harold E. Helson zu entnehmen[143].

In den meisten kommerziellen Programmen (ChemDraw, MDL ISIS/Base, Unity dbtranslate) sind entsprechende Routinen vorhanden. Sie sind allerdings weder unabhängig verwendbar, noch ist der verwendete Algorithmus dokumentiert.

Das Programm dbtranslate von Tripos ist ein Kommandozeilen-Tool zur Konvertierung verschiedener Dateiformate. Bei der Umwandlung von SMILES oder SLN in Formate, die 2D-Koordinaten enthalten müssen, wie MOL-File, werden diese automatisch berechnet. Das Programm kann daher als Subprozess gestartet und zur SDG verwendet werden.

Das Cactvs-Programmsystem enthält die Eigenschaft A_XY, die automatisch berechnet wird, wenn sie nicht eingelesen worden ist. Sie kodiert die x- und y-Koordinaten zu den Atomen.

Weitere Alternativen sind DEPICT[144] aus dem kommerziellen Daylight-Toolkit[145] oder MDRAW[146] bzw. JMDRAW[147], die im Quellcode vorliegen.

In BayTree können die Cactvs-Routine und Tripos dbtranslate alternativ verwendet werden. Letzteres muss entsprechend von Tripos lizensiert werden.

Liegen für die Atome der zu zeichnenden Struktur die x- und y-Koordinaten vor, werden sie entsprechend skaliert und die Atomsymbole zentriert eingezeichnet. Je nach Bindungsordnung werden die Verbindungslinien einfach, doppelt oder dreifach eingezeichnet; bei Vorliegen entsprechender Stereoinformation wird ein Keil verwendet. Durch einen Clipping-Algorithmus (Cohen-Sutherland Clipping)[148] wird sichergestellt, dass die Bindungslinien verkürzt werden und die Atomsymbole nicht überlappen.

5.2.5 2D-Alignment

Liegen mehrere Strukturen mit einer gemeinsamen Substruktur vor, ist es zur Vereinfachung eines Vergleichs bzw. zum Erkennen der Unterschiede erwünscht, dass die Strukturzeichnungen identisch ausgerichtet sind. Das Programm dbreport[149] ermöglicht eine derartige Ausrichtung an einem Templat (meist der Datenbankabfrage). Wild D.J. von Parke-Davis hat beim Daylight User Group Meeting 1999 einen möglichen Algorithmus dazu beschrieben[150].

Für die Strukturen in BayTree ist ein alternatives Verfahren entwickelt worden, das unabhängig von einem vorgegebenem Templat eine unabhängige einheitliche Ausrichtung, basierend auf dem MolCode, vornimmt.

Dazu wird wie folgt vorgegangen: Zu den Atomen des höchstpriorisierten Rings A und des nächstpriorisierten Rings B wird jeweils der Koordinatenschwerpunkt und damit der Ringzentroid ausgerechnet. Das Molekül wird so gedreht, dass die Verbindungslinie zwischen Ringzentroid A und Ringzentroid B horizontal ausgerichtet wird und A links von B liegt. Im nachfolgenden Schritt wird versucht, den Ring A möglichst weit nach links oben zu verschieben. Indem das Molekül horizontal, vertikal und horizontal und vertikal gespiegelt wird, werden drei weitere Koordinatensätze erzeugt. Verwendet wird der Koordinatensatz, in dem der Abstand der Atome von Ring A und eines Hilfsatoms aus Ring B zur linken oberen bounding box-Ecke am kleinsten ist.

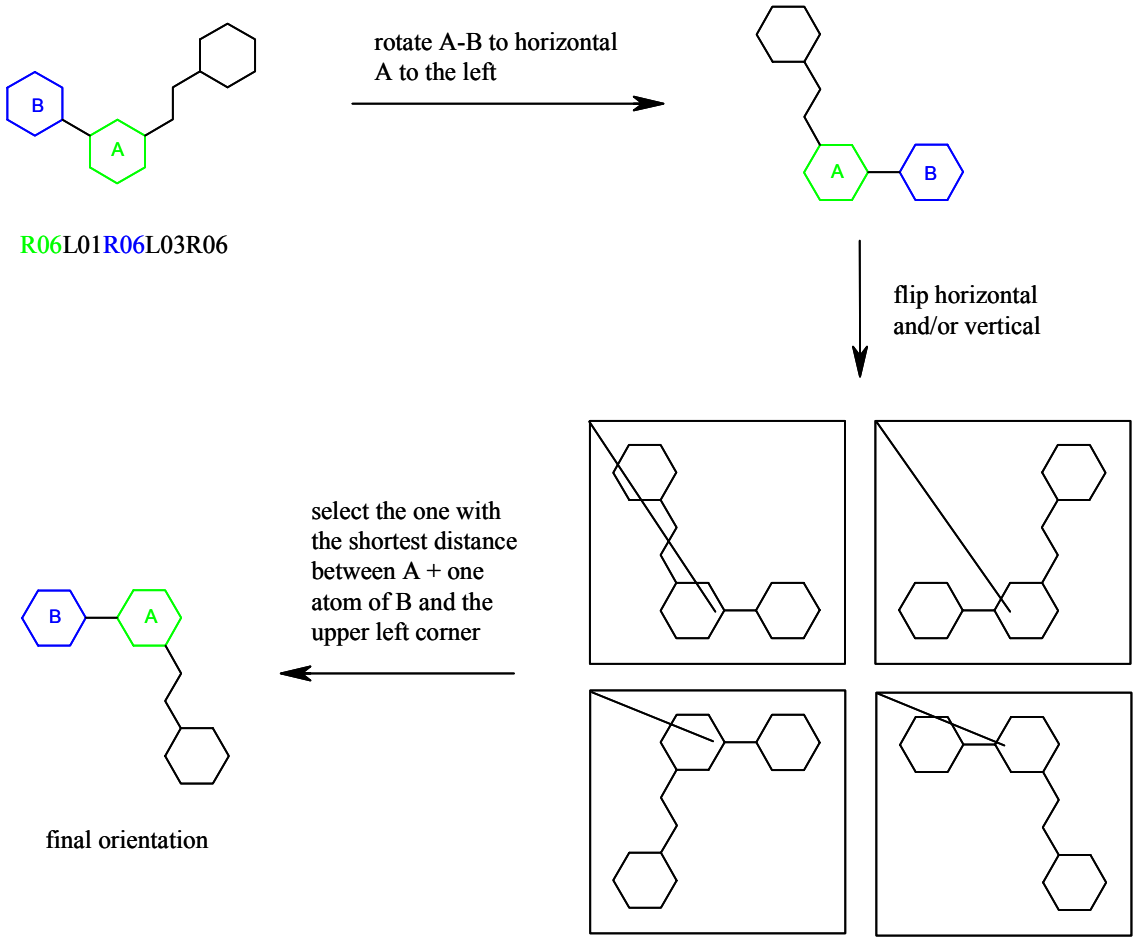


Abb. 5.14: Veranschaulichung der Alignment-Heuristik für Strukturabbildungen basierend auf dem MolCode.

5.2.6 Molecular Identification Numbers/Hash-Werte

Die Identität zweier chemischer Strukturen muß normalerweise mittels Graphen-isomorphismen-Algorithmen[151] ermittelt werden (siehe Abb. 5.15). Dies ist allerdings ein vergleichsweise aufwendiger Prozess und daher für große Datenmengen zu langwierig. Ein direkter Vergleich der Strukturen ist wegen unterschiedlicher Nummerierungsmöglichkeiten der Atome und Bindungen nicht möglich.

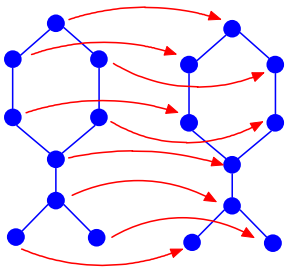


Abb. 5.15: Zuordnung gleicher Knoten/Atome in zwei Graphen/Molekülen.

Als Alternativen bleiben topologische Indizes (sog. Grapheninvarianten). Dies sind Zahlen, die die molekulare Struktur charakterisieren. Es werden zwei entgegengesetzte Ziele bei der Entwicklung von topologischen Indizes (TI) verfolgt: einerseits in Hinblick auf Struktureigenschaftsbeziehungen (QSPR/QSAR), wo gewünscht ist, dass die Deskriptoren mit physikalischen Eigenschaften oder biologischer Aktivität korrelieren, oder aber in Hinblick auf die chemische Dokumentation für Speicherung und Zugriff. Im letzteren Fall soll der TI maximal zwischen Strukturen unterscheiden. Es ist wünschenswert, dass eine einzelne mathematische Invariante, unabhängig von der Nummerierung der Atome, jedes Molekül eindeutig repräsentiert.

Es sind verschiedene TIs mit hohem Unterscheidungsvermögen entwickelt worden, die auf all-paths-Methoden oder erweiterten Adjazenzmatrizen basieren und Fließkommazahlen sind[152, 153, 154]. Die Verfahren sind aus Rechenzeitgründen wegen exponentiellem Laufzeitverhalten nicht für große Moleküle anwendbar. Die Tatsache, dass sie zu Gleitkommazahlen führen, macht sie anfällig für Rundungsfehler bei der Berechnung und beim Vergleich. Es ist bisher trotz intensiver Suche nicht gelungen, absolut eindeutige TIs zu finden.

Ein Hash-Wert ist eine hochkomprimierte Kodierung einer Datenstruktur mit festem Wertebereich und fester Länge. Es ist nicht möglich, die zugrundeliegende Datenstruktur zu rekonstruieren. In der Informatik werden Hash-Werte als Indizes für Arrays oder Dateien benutzt. In der Chemoinformatik werden sie zur Ermittlung der topologischen Identität von Atomen oder Strukturen verwendet. Der Algorithmus ist so gewählt, dass eine Kollision (Entartung) sehr unwahrscheinlich ist. Dies ermöglicht es, die Identität zweier Strukturen einfach über einen Vergleich der Hash-Werte zu bestimmen und mehrfach vorhandene Strukturen in einem Datensatz zu erkennen. Ein aufwendiger Test auf Graphen-isomorphismus ist nicht erforderlich. Der Hash-Wert ist als Grapheninvariante unabhängig von der Reihenfolge der Atomnummerierung. Im Programmpaket Cactvs ist die Eigenschaft E_HASHY verfügbar, die zu einer gegebenen Struktur einen 64-Bit-Wert (16 Zeichen Hexadezimal-String) berechnet[155].

Der Algorithmus kombiniert die Grundidee des Morgan-Algorithmus[156, 157]mit einem Pseudo-Zufallszahlengenerator[158], um eine gleichmäßige Verteilung der Bitwerte zu erhalten.

Skizze des Basisalgorithmus:

1. Initialisiere die Hash-Werte aller Atome mit einem Produkt aus Primzahlen. Jedem Parameter ist ein Abschnitt einer Primzahlentabelle zugeordnet und der Wert des Parameters wird als Index verwendet. Auf diese Art ist jede Primzahl individuell einem Wert, der für die Initialisierung verwendet werden kann, zugeordnet. Dadurch

ist gewährleistet, dass in dieser Phase keine ungewünschten Kollisionen auftreten, d.h. unterschiedliche Kombinationen von Parametern den gleichen Zahlenwert ergeben. Folgende Parameter werden verwendet:

- Anzahl der Nachbarn
- Anzahl der H-Atome
- Ordnungszahl des Atoms
- Gesamtzahl der Atome im Molekül (modulo 257)

Optional können je nach Anwendungszweck weitere Parameter wie Stereoinformation, Isotopen und Ladungen berücksichtigt werden. Zur Unterscheidung der Frameworks in BayTree ist dies nicht erforderlich.

Die Zuordnung der ersten beiden Parameter ist im folgendem Schema verdeutlicht:

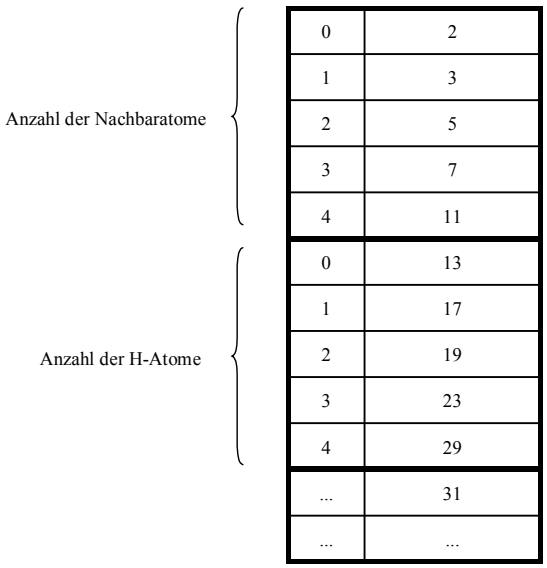


Abb. 5.16: Zuordnung der Primzahlen zu den Initialisierungsparametern.

2. Kombiniere die Werte der direkten Nachbaratome. Anders als beim Morgan-Algorithmus werden die Werte nicht addiert, sondern binär XOR verknüpft. Dieser Vorgang der Informationsverteilung wird in 32 Zyklen für alle Atome der Reihe nach vorgenommen. Dadurch werden die 32 Atom-Sphären um ein Atom herum berücksichtigt. Vor der Verknüpfung erfolgt jeweils ein feedback shift des ursprünglichen Werts; die Nachbarwerte erhalten gegebenenfalls einen occurence count shift. Bei einem feedback shift wird die Zahl um ein Bit nach links verschoben und das 0. Bit durch die XOR-Verknüpfung zweier höherwertiger Bits ersetzt (siehe Abb. 5.17).

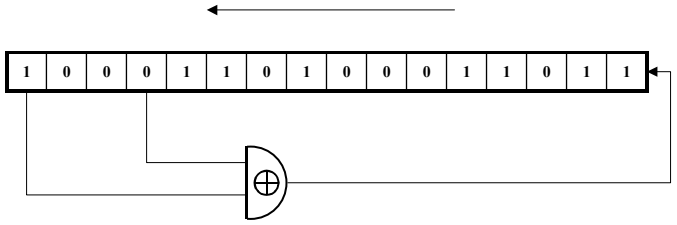


Abb. 5.17: Veranschaulichung des feedback shift.

Der occurence count shift ist erforderlich, da sich der Einfluß einer geraden Anzahl identischer Werte bei der XOR-Verknüpfung wieder aufhebt. Um dies zu umgehen, wird nur der erste Wert direkt verknüpft, alle weiteren identische Werte werden entsprechend der Häufigkeit des Auftretens nach links rotiert, d.h. der zweite Wert wird um eine Bitposition rotiert, der dritte Wert um zwei Bitpositionen usw.

3. Der Hash-Wert der gesamten Struktur setzt sich aus den XOR-verknüpften Atom-Hashwerten zusammen. Dabei werden bei mehrfach vorkommenden Werten wieder occurence count shifts vorgenommen.

Strukturen, die ausschließlich einheitliche Atomumgebungen, gleiche Größe und gleiche Elementzusammensetzung haben (Konstitutionsisomere), können mit dem Basisalgorithmus nicht unterschieden werden. Dies ist beispielsweise bei Decalin und Bicyclopentan der Fall (siehe Abb. 5.18).

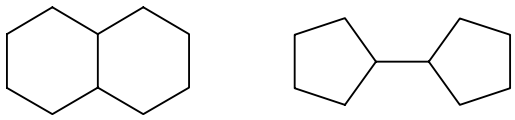


Abb. 5.18: Decalin und Bicyclopentan enthalten identische Atomumgebungen und können daher mit dem Basisalgorithmus nicht anhand ihrer HASH-Werte unterschieden werden. Die MolCodes R06I02R06 und R05L01R05 der Strukturen sind unterschiedlich.

Zur Unterscheidung ist folgender Schritt erforderlich: Nach der Berechnung der Atom-Hashwerte wird der kleinste Satz an mehrfach auftretenden identischen Hash-Werten der Scaffold-Atome (nicht-terminale Atome) ermittelt. Gibt es mehrere identischer Größe, wird der Satz mit dem kleinsten Hash-Wert verwendet. Ausgehend von den ursprünglichen Atom-Hashwerten wird der Reihe nach jeweils einer mittels eines feedback shift perturbiert und gemäß Schritt 2 über die Struktur verteilt. Das Ergebnis ist dann die Kombination (XOR mit occurence count shifts) aller perturbierten Struktur-Hashwerte.

Durch die Verwendung von maschinennahen shift- und Boolschen Operationen ist eine hoch-performante Implementierung des Algorithmus möglich.

Voigt et al. haben unter Verwendung des Cactvs-Hashwertes acht große chemische Datenbanken aus verschiedenen Quellen miteinander verglichen[159]. Die untersuchten Eigenschaften waren u.a.: Rate an internen Duplikaten und Überlapp identischer Strukturen zwischen zwei Datenbanken. Die Rate an Duplikaten innerhalb einer Datenbank bewegte sich

zwischen 0,4% und 13,4%, der Überlapp zwischen den Datenbanken zwischen 0,3% und 87,4%.

5.2.7 Tree Layout/Dynamic Tree Drawing

Die im TSP des MolCode kodierten Verknüpfungspfade der Knoten werden in BayTree verwendet, um einen hierarchischen Strukturbaum zu erzeugen. Die zugrundeliegende Aufgabenstellung, das Erstellen von Diagrammen zu Graphen, ist ein umfangreiches eigenes Forschungsgebiet, zu dem jährliche Graph Drawing-Konferenzen abgehalten werden[160]. Je nach Art der interessierenden Graphenklasse gibt es spezielle Algorithmen und Verfahren und entsprechende Software[161]. Erwähnenswert sind die universellen Programme DaVinci[162] und graphlet[163, 164] und die speziellen Programme für Phylogenetische Bäume (drawtree aus PHYLIP)[165], Metabolische Netzwerke (metabolic pathways)[166] oder Binäre Bäume[167].

Welchen praktischen Nutzen die Zeichnung eines Graphen hat, hängt von ihrer Lesbarkeit bzw. Interpretierbarkeit ab. Die Bedeutung des Diagramms muß schnell und klar erkannt werden können und ästhetisch ansprechend sein. Die Ästhetik gibt die Rahmenkriterien vor, z.B. Planarität, minimale Anzahl an Überlappungen (Kanten, die sich mit anderen Kanten oder Knoten schneiden), kompakte Darstellung (kleine Zeichenfläche), maximale Symmetrie.

Zum Zeichnen von Bäumen, die ein Sonderfall eines Graphen ohne Loops darstellen, gibt es vereinfachte Algorithmen, bei denen der Test auf Graphenplanarität entfallen kann. Zusätzlich gibt es im wesentlichen nur eine Art, um Bäume mit einer Wurzel (rooted trees) zu zeichnen.

Bei sogenannten one shot-Algorithmen wird zu einem gegebenen Input nur einmal von Grund auf eine neue statische Zeichnung erstellt. Im Gegensatz dazu muss bei Anwendungen, in denen der Anwender mit dem angezeigten Graphen interagiert, d.h. Kanten und Knoten hinzufügt oder entfernt, die Zeichnung nach jeder Änderung entsprechend aktualisiert werden. Bei solchen interaktiven bzw. dynamischen Algorithmen sollte die Aktualisierung schnell erfolgen und das Diagramm nicht grundlegend verändert werden, damit der Benutzer das neue Diagramm mit dem ursprünglichen in Beziehung setzen kann. Eine kleine Änderung im Graphen darf nur zu einer kleinen Änderung im Diagramm führen.

Das für BayTree einsetzbare Verfahren muss folgenden Kriterien genügen:

- Es soll ein ansprechendes Diagramm erstellt werden.
- Es muss ein dynamischer Algorithmus sein.
- Die Anzahl der Kindknoten muss beliebig sein dürfen.
- Die Größenausdehnung der Knoten soll variabel sein.
- Die Subtrees sollen so nahe wie möglich zueinander plaziert werden.

Einen geeigneten Algorithmus hat Sven Moen beschrieben[168]. Das Besondere daran ist, dass die Kontur zu jedem Knoten explizit als Polygon verwaltet wird. Dieses besteht aus zwei polylines für den linken und den rechten Rand. Die untere Begrenzung ist als zusätzliches Segment im linken Polygon enthalten. Da die polylines monoton sind, nehmen sie immer von unten nach oben und niemals von oben nach unten zu. Formal ist eine solche monotone polyline definiert als eine Abfolge von Punkten $P_i = (x_i, y_i)$ mit $1 \leq i \leq n$ wobei $y_{i+1} \geq y_i$ für $i < n$. Diese Eigenschaft ermöglicht es, die Berechnung des relativen Abstands zweier Knoten und die Verknüpfung zweier Konturen schnell durchzuführen.

Bei den Blattknoten des Baums entspricht die Kontur der Umhüllenden (Bounding Box) der Knotengraphik mit einem zusätzlichen Rand als Zwischenraum zu den Nachbarknoten. Die Kontur aller anderen Knoten enthält den Knoten mit dem jeweiligen Subtree. Die Kontur des Wurzelknotens enthält daher den vollständigen Baum.

```
class Node {
     string   tag  // Bezeichnung
     Node *parent  // Zeiger auf Elternknoten
     Node *child   // Zeiger auf ersten Kindknoten
     Node *sibling // Zeiger auf Schwesterknoten
     int  width    // Breite
     int  height   // Höhe
     int  border   // Abstand
     Point    pos  // absolute Position
     Point    prev_pos  // vorhergehende absolute Position
     Point    offset    // relative Position zum vorhergehenden Knoten
     Polygon  contour   // Kontur
}

class Polygon {
     Polyline     left    // linke Seite und untere Begrenzung
     Polyline  right     // rechte Seite
}
```

Listing 5.4: Variablen der Klassen Node und Polygon.

Der Grundalgorithmus läßt sich wie folgt beschreiben:

Ausgehend vom Wurzelknoten wird die Prozedur **layout** rekursiv für alle weiteren Knoten des Baums aufgerufen. Handelt es sich bei node um einen Leafnode, wird die Kontur, wie in Abb. 5.19 links, beschrieben initialisiert. Für alle anderen Knoten, die Subnodes enthalten, wird die Prozedur **layout_subnodes** aufgerufen. In dieser wird die Kontur des übergebenen Knotens node mit der Kontur des ersten Kindknotens initialisiert. Die Position des nächsten Kindknotens wird (relativ dazu) in der Prozedur **offset** berechnet. Dazu werden alle Punkte der rechten polyline von node (Knoten A) mit der linken polyline des Kindknotens (Knoten B) verglichen und der offset (ausgehend von 0) so lange vergrößert, bis sich die Konturen nur noch berühren und nicht mehr überlappen (distance > 0) (siehe Abb. 5.19 rechts).

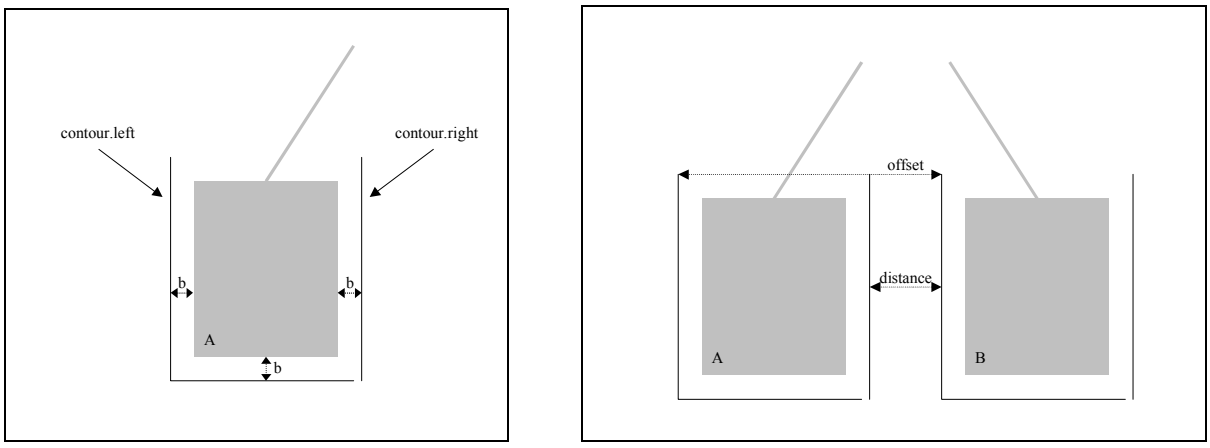


Abb. 5.19: links: Kontour eines Leafnodes; rechts: Veranschaulichung des offset zwischen zwei Knoten.

Ist der Knoten positioniert, wird die Kontur des bearbeiteten Kindknotens in der Prozedur **merge_contour** zur Kontur von node hinzugefügt. Die letzten beiden Schritte werden für alle weiteren Kindknoten wiederholt. Als letzter Schritt wird in der Prozedur **attach_parent** die Ausdehnung des eigentlichen Elternknotens C angefügt (siehe gestrichelte Linien in Abb. 5.20 rechts).

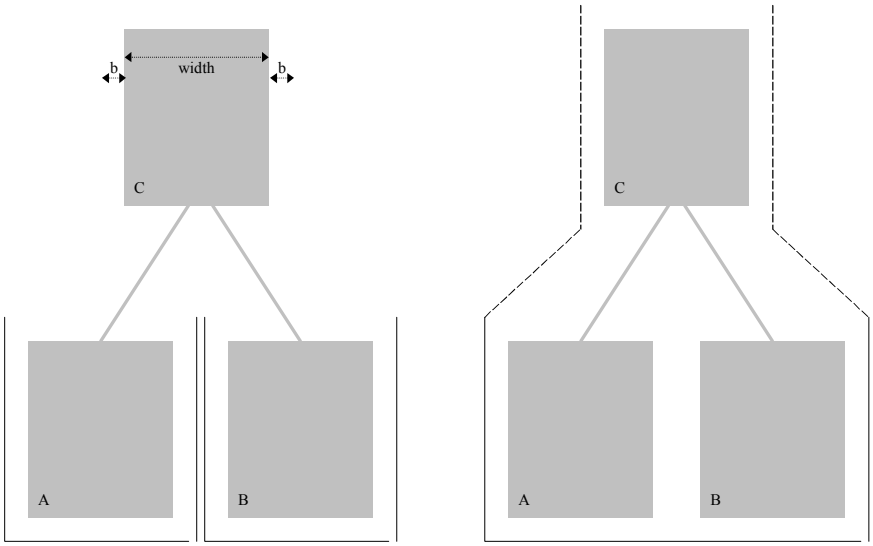


Abb. 5.20: Veranschaulichung der Prozeduren merge_contour **und** attach_parent.

Die so entstandene Gesamtkontur des Knotens C, der den gesamten Subtree bestehend aus den Knoten A, B und C enthält, wird verwendet, wenn die Position des Elternknotens von C berechnet wird. Der zusätzliche Knoten D kann platzsparend direkt neben dem Knoten C oberhalb von Knoten B positioniert werden (siehe Abb. 5.21).

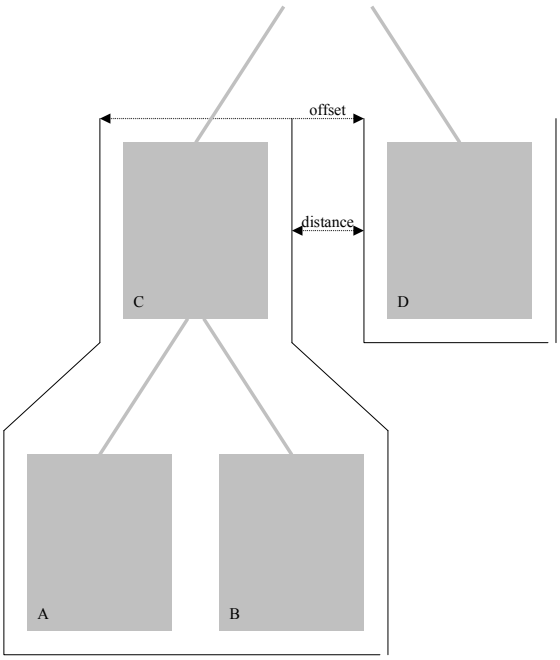


Abb. 5.21: Relative Positionierung des Knotens D zu den Knoten A bis C.

Die folgenden beiden Listings enthalten den Pseudocode der Prozeduren **layout** und **layout_subnodes** zur Berechnung des Treelayouts.

```
layout(node)
     FOR EACH subnode ∈ children(node)
          layout(subnode)

     IF isleaf(node)
          layout_leaf(node)
     ELSE
          layout_subnodes(node)
```

Listing 5.5: Pseudocode der Prozedur layout.

```
layout_subnodes(node)

     first = TRUE

     FOR EACH subnode ∈ children(node)
          IF first
               first = FALSE
               node.offset = (0, 0)
               node.contour = subnode.contour
          ELSE
               offset(node, subnode)
               merge_contour(node, subnode)

     attach_parent(node)
```

Listing 5.6: Pseudocode der Prozedur layout_subnodes.

In einem zweiten Durchlauf wird ausgehend von absolut positionierten Wurzelknoten die Position aller anderen Knoten berechnet und diese in der Zeichenfläche entsprechend verschoben. Abschließend werden zwischen den Knoten die Verbindungslinien eingezeichnet bzw. entsprechend verschoben und skaliert. Sie können wahlweise als direkte Verbindunglinie oder als Polygon, bestehend aus drei Linien im rechten Winkel, eingezeichnet werden.

Zur Anwendung eines derartigen Tree Layout-Algorithmus für Graphikelemente eines Tk-Canvas steht mit TkTree[169] von Allan Brighton eine Sammlung von Tcl-Kommandos zur Verfügung. Diese sind durch Verwendung der Canvas-Tags sehr elegant in das Tk-System integriert und ermöglichen es, Gruppen von Graphikobjekten als Baum anzuordnen.

5.2.8 Colorscales zur Knoten-Kolorierung

Farbskalen werden üblicherweise verwendet, um eine Abfolge von Werten erkennbar zu machen. Zu diesem Zweck werden diese linear auf einer Farbskala abgebildet. Weit verbreitet sind folgende Farbskalen: Regenbogenskala, Grün-Gelb-Rot, Blau und Grau[170].

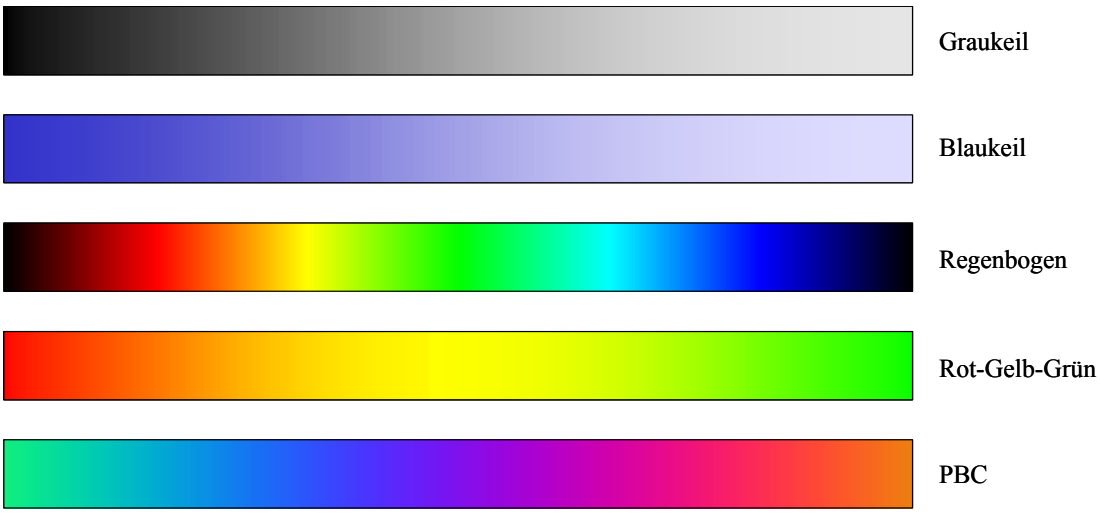


Abb. 5.22: Farbskalen zur Kolorierung.

Nach absoluten Maßstäben kann das menschliche Auge Farbtöne vergleichsweise schlecht erkennen, relative Unterschiede von Farbtönen allerdings sehr gut[171, 172]. Wenn die unterschiedlichen Farben gleichzeitig auftreten, können auch kleine Farbnuancen erkannt werden. Abbildungen, z.B. Falschfarbenfotos, die auf engem Raum unterschiedliche Farbflächen aufweisen, erleichtern durch die Färbung das Erkennen gleicher Bereiche. Da die Mehrzahl der Knoten der BayTree-Zeichenfläche ausserhalb des sichtbaren Ausschnitts liegt und nur durch ein Verschieben des Ausschnitts sichtbar gemacht werden kann, können ähnliche Farben einander nur schwer zugeordnet werden. Als Alternative kommt eine vordefinierte Anzahl von unterscheidbaren Farben in Frage.

Für das Abbilden von Klassen in Farben, um unterschiedliche Klassen als solche zu erkennen, sind andere Farbmodelle erforderlich. M. Ankerst hat zu diesem Zweck die PBC-Farbskala entwickelt[173]. Sie basiert auf dem HSI-Farbmodell, das eine Variante des HSV-Modells ist[174]. Im HSI-Modell wird jede Farbe durch ein Tripel aus Farbton (hue), Sättigung (saturation) und Intensität (intensity) repräsentiert. Die Farbskalen werden durch lineare Interpolation zwischen einem Minimum und einem Maximum der hue-, saturation- und intensity-Werte berechnet und dann in das RGB-Farbmodell (Rot-Grün-Blau) umgerechnet. Gut unterscheidbare klare Farben, die nicht ähnlich zu schwarz und weiß wahrgenommen werden, erhält man mit folgenden Parametern: Für das Minimum wird hue = 2.5 und intensity = saturation = 1.0 verwendet und für das Maximum hue = 0.5 und intensity = saturation = 1.0. Letztendlich wird in der PBC-Farbskala nur der Farbton-Parameter variiert; in den meisten anderen Fällen wird ein großer Anteil der Farben zu ähnlich und zu dunkel.

5.2.9 Estradas spektrale Momente

Estrada et al. verwenden in TOSS-MODE (Topological Substructure Molecular Design) die spektralen Momente als 2D-Deskriptoren[175, 176, 177]. Diese werden aus der Bindungs-Adjazenzmatrix der Moleküle wie folgt berechnet:

$$\mu_j(\hat{L}(\mathbf{G})) = tr(\mathrm{A}(\hat{L}(\mathbf{G}))^j)$$

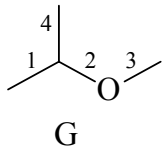
Die Bindungs-Adjazenzmatrix **B** enthält analog zur üblicheren Atom-Adjazenzmatrix die Verknüpfungsmuster der Bindungen. Sie kann daher auch als Atom-Adjazenzmatrix **A** des ersten Linegraphen des Moleküls betrachtet werden. Der Linegraph L(G) wird erzeugt, indem die Kanten von G zu Knoten in L(G) transformiert werden. Zwei Knoten in L(G) sind benachbart, wenn die zugehörigen Kanten in G benachbart sind, d.h. am gleichen Knoten zusammentreffen. Die Adjazenzen der Kanten in G entsprechen den Adjazenzen der Knoten in L(G). Um zusätzlich zur Topologie des Moleküls auch die Heteroatome und Bindungsordnungen zu berücksichtigen, werden auf den Diagonalelementen Gewichte eingesetzt. Mögliche Gewichtungsfaktoren sind Bindungsdipolmomente, Bindungslängen, Bindungspolarisationen etc. Für QSAR werden bevorzugt die Bindungsdipolmomente verwendet, diese sind in folgender Tabelle aufgeführt.

Bindung	Gewicht	Bindung	Gewicht	Bindung	Gewicht
C-C	0.00	C=C	0.00	C≡C	0.00
C-N	0.40	C=N	0.90	C≡N	3.60
C-O	0.86	C=O	2.40		
C-S	2.95	C=S	2.80		
N-O	0.30	N=O	2.00		
C-F	1.51				
C-Cl	1.56				
C-Br	1.48				
C-I	1.29				

Tabelle 5.1: Chemische Bindungsdipolmomente[178].

Nicht tabellierte Werte können problemlos über eine semiempirische AM1 MOPAC[179]-Rechnung unter Verwendung der Bindungslänge und der Atomladungen bereitgestellt werden. Für den überwiegenden Teil der gängigerweise untersuchten organischen Wirkstoffmoleküle ist das allerdings nicht erforderlich.

Das j-te spektrale Moment entspricht der Spur der j-ten Potenz der Bindungs-Adjazenzmatrix. Die Spur („trace“) einer Matrix ist die Summe aller Diagonalelemente. Das nullte spektrale Moment μ_0 enthält stets die Anzahl der Atome des Graphen G.

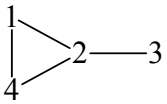


G

C-C: 0.0
C-O: 0.7

$$\mathbf{B} = \begin{bmatrix} 0.0 & 1 & 0 & 1 \\ 1 & 0.7 & 1 & 1 \\ 0 & 1 & 0.7 & 0 \\ 1 & 1 & 0 & 0.0 \end{bmatrix}$$

$$\mu_k = tr(\mathbf{B}^k)$$



L(G)

k	0	1	2	3	...	10	...	15
μ_k	4.0	1.4	6.98	6.99		1433.49		52876.74

Abb. 5.23: Berechnung der spektralen Momente für Isopropylmethylether.

In Abb. 5.23 sind am Beispiel des Moleküls Isopropylmethylether alle Schritte zusammengefasst. Der Ether wird als Graph G betrachtet. Die Bindungen werden nummeriert und als Knoten des Linegraphen L(G) verwendet und damit die Außerdiagonalelemente der Bindungs-Adjazenzmatrix B besetzt. Es gibt nur Bindungen vom Typ C-C und C-O, die mit 0,0 und 0,7 gewichtet werden[180]. Diese Werte werden auf den zugehörigen Diagonalelementen eingesetzt. Ausgehend von der derart initialisierten Matrix werden durch Potenzierung und Spurberechnung die in der Tabelle aufgelisteten spektralen Momente erzeugt.

Die ungewichteten spektralen Momente können mit Hilfe von Substruktursuchen anschaulich über embedded frequencies berechnet werden. Die Abb. 5.24 enthält die Terme und die verwendeten Subtrukturen. Enthalten die Strukturen allerdings Heteroatome, wird die erforderliche Anzahl an unterschiedlichen Substrukturen sehr schnell unüberschaubar und die Berechnung der spektralen Momente ist daher auf diesem Weg nicht mehr durchführbar.

$$\mu_0(L^1(G)) = |F_1| \quad (1)$$

$$\mu_2(L^1(G)) = 2|F_2| \quad (2)$$

$$\mu_3(L^1(G)) = 6|F_3| \quad (3)$$

$$\mu_4(L^1(G)) = 2|F_2| + 12|F_3| + 24|F_4| + 4|F_5| \quad (4)$$

$$\mu_5(L^1(G)) = 30|F_3| + 120|F_4| + 10|F_6| \quad (5)$$

$$\mu_6(L^1(G)) = 2|F_2| + 60|F_3| + 480|F_4| + 12|F_5| + 24|F_6| + 6|F_7| + 36|F_8| + 24|F_9| \quad (6)$$

$$\mu_7(L^1(G)) = 126|F_3| + 1680|F_4| + 84|F_6| + 210|F_8| + 112|F_9| + 14|F_{10}| + 14|F_{11}| + 84|F_{12}| \quad (7)$$

$$\mu_8(L^1(G)) = 2|F_2| + 252|F_3| + 5544|F_4| + 28|F_5| + 200|F_6| + 32|F_7| + 1008|F_8| + 464|F_9| + 32|F_{10}| + 40|F_{11}| + 672|F_{12}| + 8|F_{13}| + 48|F_{14}| + 48|F_{15}| + 32|F_{16}| + 32|F_{17}| + 288|F_{18}| \quad (8)$$

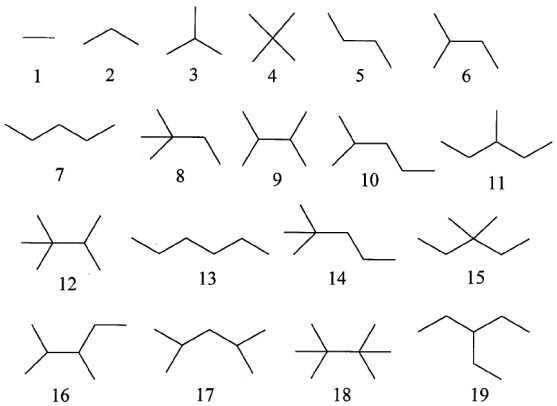


Abb. 5.24: Rechenterme und Subtrukturen zur Berechnung der ungewichteten spektralen Momente über embedded frequencies[181].

Erste Anwendung fanden diese Deskriptoren für einfache QSPRs, z.B. Siedepunkte von Alkanen[182], magnetische Suszeptibilitäten[183] bis hin zu Dipolmomenten von substituierten Benzolen[184]. Nach den vielversprechenden Ergebnissen, wurde das TOSS-MODE-Verfahren auch für QSAR-Fragestellungen verwendet. Die folgende Tabelle enthält die Ergebnisse für eine Auswahl von Aktiv/Inaktiv-Klassifikationen:

	richtig klassifizierte Verbindungen in Prozent (Anzahl der richtig klassifiziert Verbindungen / Gesamtzahl der Verbindungen im Datensatz)	
	Trainingsdatensatz	Testdatensatz (Vorhersage)
Sedativa[185]	76.47% (143/187)	81.82% (63/77)
Anticancer[186]	88.39% (198/224)	91.43% (64/70)
Anticonvulsiva[187]	89.35% (151/169)	87.88% (58/66)

Die Ergebnisse ließen sich leider nicht eindeutig reproduzieren, da die Datensätze entweder nur teilweise in den Publikationen aufgeführt waren oder eine eindeutige Strukturzuordnung aufgrund der von E. Estrada angegebenen Trivialnamen nicht möglich war. E. Estrada wollte die Strukturen der vollständigen Datensätze nicht zur Verfügung stellen.

In Verbindung mit der Linearen Diskriminanzanalyse beschreibt Estrada die Berechnung der Beiträge beliebiger Fragmente (fragment contribution) zur Aktivität. Dazu werden zu dem ausgewählten Fragment alle möglichen Subgraphen enumeriert[188, 189] und von den spektralen Momenten des Fragments die spektralen Momente aller Subgraphen subtrahiert. Durch Einsetzen der derart berechneten Differenz der spektralen Momente in das Diskriminanzmodell kann ihr Beitrag zur biologischen Aktivität quantifiziert werden. Die biologische Aktivität eines Moleküls ist das Resultat der Beiträge aller ihrer Fragmente und nicht notwendigerweise durch den Einfluss eines einzelnen speziellen Fragments bestimmt. Nichtsdestotrotz ist die Identifikation struktureller Merkmale interessant, um sich bei der Synthese neuer Derivate daran zu orientieren und Gruppen, die negativ zur Aktivität beitragen, zu minimieren und solche mit positivem Beitrag zu maximieren. Zur Vereinfachung der Interpretation des Beitrags ist von Vorteil, die Aktiven des Trainingsdatensatzes mit +1 und die Inaktiven mit –1 bei der Erzeugung des Modells zu berücksichtigen. Positive Werte des discriminant score stehen für Fragmente, die positiv zur Aktivität beitragen, und solche mit negativen Werten führen zu Aktivitätsverlust. Zur Ableitung von Regeln werden die Beiträge von Gruppen zusammengehöriger möglicher Fragmente berechnet und ihre jeweiligen Beiträge betrachtet. Eine solche Gruppe kann beispielsweise ein Ringsystem mit unterschiedlichen Heteroatomen sein: Tetrahydrofuran, Oxathiolan und Dioxolan.

Die spektralen Momente kodieren leider keinerlei Chiralitätsinformation. Die einfachste Möglichkeit, diese in der Klassifizierung zu berücksichtigen, ist die Ergänzung einer zusätzlichen Indikatorvariablen für jedes relevante Chiralitätszentrum. Deren Wert wird für R auf 1, für S auf –1 und im achiralen Fall auf 0 gesetzt.

Zur Berechnung der spektralen Momente unabhängig von BayTree steht das Programm **specmom** zur Verfügung. Es wird von der Kommandozeile mit specmom <*descriptorfile.csv*> <*infile*> <*infile*> aufgerufen. Es werden beliebig viele Eingabefiles *infile* in den Formaten SLN, SDF, MOL2 und MDB unterstützt. Die berechneten Deskriptoren werden zu jeder Struktur – verknüpft durch die RegId – im CSV-Format in *descriptorfile.csv* geschrieben.

Desweiteren steht vom Autor eine Implementierung in SVL[190] zur nahtlosen Integration in das QuaSAR-Descriptor Modul von MOE[191] zur Verfügung.

5.2.10 Lineare Diskriminanzanalyse

Die Lineare Diskriminanzanalyse (LDA)[192, 193] ist ein mathematisches Verfahren zur Klassifizierung von Objekten in vorgegebene Gruppen (in userem Fall den Klassen aktiv/inaktiv). Dabei wird zu einem Trainingsdatensatz eine Transformationsmatrix erstellt. Sie enthält die Gewichte der Deskriptoren der Objekte, aus denen die Diskriminanzvektoren so berechnet werden, dass deren Streuung innerhalb der Gruppen minimal und zwischen den Gruppen maximal ist. Nach der Transformation sind die gegebenen Gruppen eines Trainingsdatensatzes optimal linear separiert, was die Zuordnung neuer Testobjekte zu den Objektklassen über die Diskriminanzfunktion ermöglicht. Das Verfahren beruht auf der Annahme einer multivariaten Normalverteilung für die verwendeten Deskriptoren in den Datensätzen, was nicht erfüllt sein muß. Die Qualität der Ergebnisse wird um so besser sein, je besser die Auswahl der Trainings- und Testsatz-Verbindungen für die LDA und je aussagekräftiger die Deskriptoren für die Beschreibung des Wirkmechanismus sind. Das Verfahren steht innerhalb von Statistikprogrammen wie SPSS[194] oder R[195] zur Verfügung.

In den beiden Varianten wird die Datenmatrix **X** mit den Elementen x_{ik} mit $i = 1 \ldots p$ (Deskriptoren) und $k = 1 \ldots n$ (Objekte) verwendet.

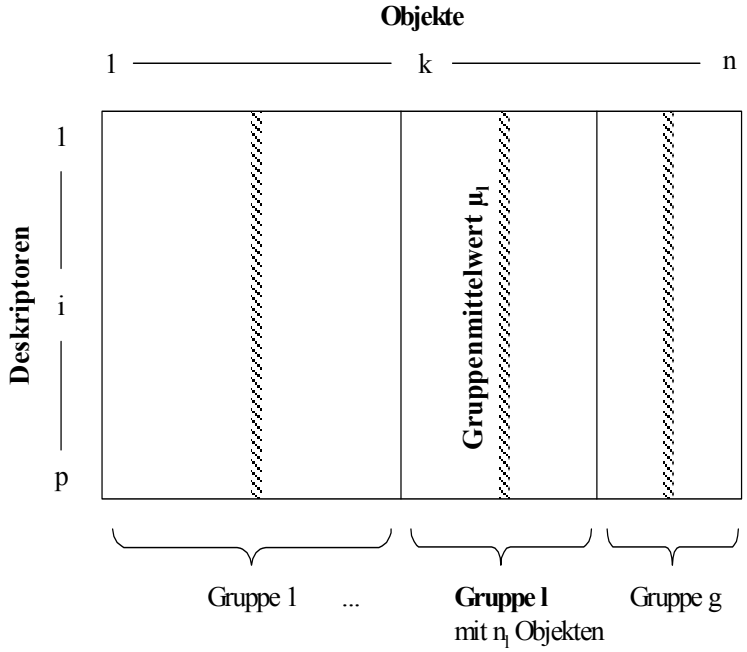


Abb. 5.25: Die Datenmatrix für die LDA. Jedes der n Datenobjekte ist durch einen Spaltenvektor mit p Deskriptoren repräsentiert. Eingezeichnet sind drei Gruppen und der zugehörige Gruppenmittelwertvektor μ_l für die l-te Gruppe.

Die Bedeutung der anderen Variablen ist folgender Übersicht zu entnehmen:

g: Anzahl der Objektgruppen/Klassen
p: Anzahl der Variablen/Deskriptoren
n: Gesamtzahl der Objekte
n_l: Anzahl der Objekte in Gruppe l

Im einfachsten Fall der Bayesschen Klassifikation[196], wird der Mahalanobis-Abstand d_l eines Testobjekts **x** vom Gruppenmittelwert $\boldsymbol{\mu}_l$ (l = 1 ... g) unter Verwendung der Gruppenkovarianzmatrix $\mathbf{S}_l$ berechnet. Dies wird für alle g Gruppen vorgenommen und das Objekt **x** wird der Klasse zugeordnet, zu der der Abstand d_l am kleinsten ist.

Der Mahalanobis-Abstand wird nach folgender Vorschrift berechnet:

$$d_l^2(\mathbf{x}) = (\mathbf{x} - \boldsymbol{\mu}_l)^T \mathbf{S}_l^{-1} (\mathbf{x} - \boldsymbol{\mu}_l)$$

In vielen Fällen ist die Korrelation zwischen den Variablen innerhalb jeder Gruppe gleich. Wenn die Korrelation unabhängig von der Gruppe ist, haben alle Gruppen dieselbe Kovarianzmatrix und, da die Datenmatrix unverändert bleibt, kann der Ausdruck so vereinfacht werden, dass der Gewichtsvektor $\mathbf{v}_l$ für jede Gruppe nur einmal berechnet zu werden braucht.

$$\mathbf{v}^l = \boldsymbol{\mu}_l^T \mathbf{S}^{-1}$$

$$f_l(\mathbf{x}) = \sum_{i=1}^{p} v_i^l x_i + v_0^l$$

Bei dieser linearen Form der Bayesschen Klassifikation werden die Objekte **x** der Gruppe mit dem größten discriminant score f(**x**) zugeordnet. Die Wahrscheinlichkeit (posterior probability) für die Gruppenzugehörigkeit berechnet sich entsprechend dem Bayesschen Theorem:

$$P(G_l \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid G_l) \cdot P(G_l)}{\sum_{i=1}^{g} P(\mathbf{x} \mid G_i) \cdot P(G_i)}$$

$P(G_l)$ ist die prior probability, d.h. die Wahrscheinlichkeit für ein Datenobjekt, ohne zusätzliche Informationen aus der Gruppe l zu kommen. Sie ist dem Anteil der Datenpunkte der Gruppe l im Verhältnis zur Gesamtzahl der Objekte n_l/n proportional. Die prior probabilities für alle Gruppen werden im Weiteren als identisch angenommen.

Für den größten Diskriminanzfunktionswert berechnet sich die posterior probability

$$P(G_l \mid \mathbf{x}) = \frac{1}{\sum_{i=1}^{g} \exp[f_i - \max(f_l)]}$$

mit $P(\mathbf{x} \mid G_i) = \exp[\, f_i(\mathbf{x}) \,]$ als bedingter Wahrscheinlichkeit (conditional probability).

Der Mittelwertsvektor $\boldsymbol{\mu}_l$ und die Gesamtkovarianz-Matrix **S** (p×p) werden nach folgenden Formeln berechnet:

$$\mu_i = \frac{1}{n}\sum_{k=1}^{n} x_{ik} \quad \text{Mittelwertsvektor für die l-te Gruppe}$$

$$S_{ij} = \sum_{k=1}^{n_l}(x_{ik} - \mu_i)(x_{jk} - \mu_j) \qquad i,j = 1 \ldots p$$

Der letzte Ausdruck wird für eine effizientere Berechnung durch den Computer wie folgt umgestellt.

$$S_{ij} = \sum_{k=1}^{n} x_{ik}x_{jk} - \frac{1}{n}\sum_{k=1}^{n} x_{ik}\sum_{k=1}^{n} x_{jk}$$

Um die Klassifikation von unbekannten Objekten zu verbessern, ist es bei der sog. Kanonischen Analyse möglich, die zur Berechnung der Linearkombinationen der Deskriptoren verwendeten Gewichtsvektoren zu optimieren. Die optimalen Gewichte v_i sind die, für die die Daten der gegebenen g Gruppen nach der Transformation maximal trennen. Ein geeignetes Maß der Gruppenseparation muss sowohl die Differenzen zwischen den Mittelwerten als auch die Streuung der Gruppen um ihre Mittelpunkte berücksichtigen. Der Zweiklassenfall in zwei Dimensionen ist in Abb. 5.26 veranschaulicht. Die beiden Gruppen sind durch Ellipsoide repräsentiert und dazwischen ist eine wilkürliche Trennlinie gezogen. Wenn die Streuungsellipsoide und die Mittelpunkte auf die Linie projiziert werden, ist erkennbar, dass deren Ausdehnung und Abstand vom Winkel der Linie abhängt. Die optimale Orientierung der Trennlinie erhält man durch die Kanonische Analyse[197].

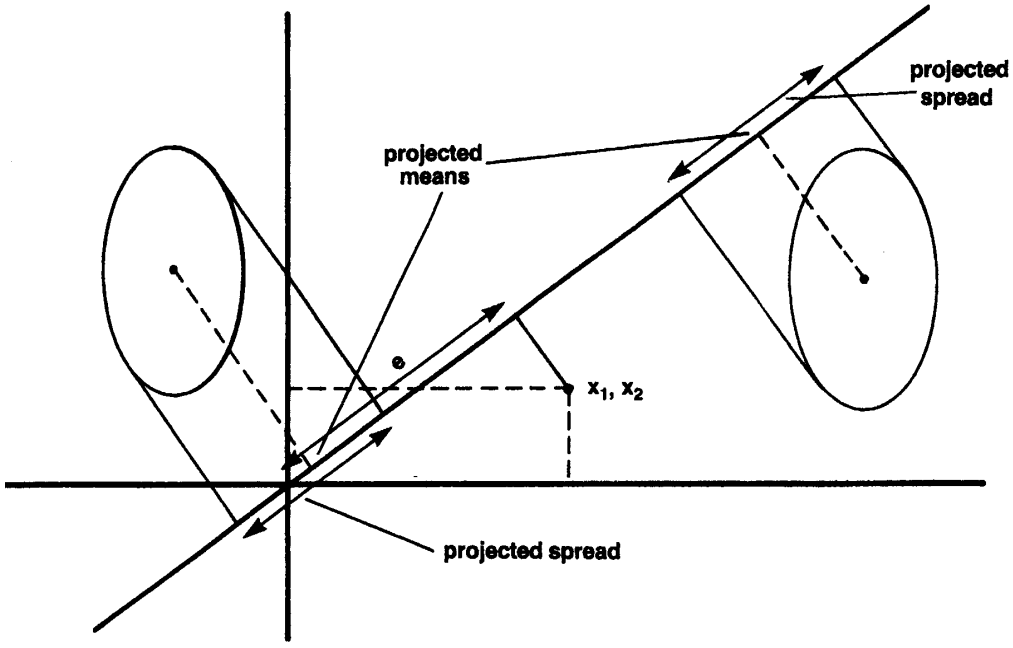


Abb. 5.26: Der Zweiklassenfall in zwei Dimensionen. Auf die Trennlinie sind die Mittelpunkte und die Ausdehnung der Gruppenellipsoide projiziert.

Anders formuliert ist die Separation um so besser, je homogener die Gruppen sind, je kleiner also die Streuung der Objekte innerhalb der Gruppen verglichen mit der Streuung der Gruppenmittelpunkte („zwischen den Gruppen“) ist. Dies entspricht dem Maximum des Separationsquotienten.

Die mittlere Streuungsmatrix (gemittelte Kovarianzmatrix) **W** (within) innerhalb der Gruppen entspricht bis auf einen Vorfaktor der Gesamtkovarianzmatrix **S**.

$$W_{ij} = \frac{1}{n-g} S_{ij}$$

Die Streuung **B** (Kovarianzmatrix) zwischen (between) den Gruppenmittelpunkten, d.h. die Abweichung der Gruppenschwerpunkte vom Gesamtschwerpunkt, ist wie folgt definiert:

$$B_{ij} = \sum_{l=1}^{g} (\mu_i^l - \mu_i)(\mu_j^l - \mu_j) \quad \text{bzw.} \quad B_{ij} = \sum_{l=1}^{g} \mu_i^l \mu_j^l - \frac{1}{g} \sum_{l=1}^{g} \mu_i^l \sum_{l=1}^{g} \mu_j^l$$

$$\mu_i^l = \frac{1}{n_l} \sum_{k=1}^{n_l} x_{ik}^l$$

Die Berechnung des optimalen Gewichtsvektors **v** zur Transformation von **x** in optimierte Diskriminanzkoordinaten **y** erfolgt in mehreren Schritten:

$$\mathbf{y} = \mathbf{v}^T \mathbf{x}$$

$\lambda = \frac{\mathbf{v}^T \mathbf{B} \mathbf{v}}{\mathbf{v}^T \mathbf{W} \mathbf{v}}$	Maximierung des Separations-Quotienten
$\frac{\partial \lambda}{\partial \mathbf{v}} = 0$	erste partielle Ableitung gleich Null setzen
$\mathbf{B}\mathbf{v} = \lambda \mathbf{W}\mathbf{v}$	allgemeines Eigenwertproblem
$\mathbf{W}^{-1}\mathbf{B}\mathbf{v} = \lambda \mathbf{v}$	gewöhnliches Eigenwertproblem (mögliche Vorgehensweise)
$\mathbf{W} = \mathbf{D}^T \mathbf{D}$	Cholesky-Zerlegung zur Sicherstellung der Symmetrie
$\mathbf{C} = (\mathbf{D}^{-1})^T \mathbf{B} \mathbf{D}^{-1}$	
$\mathbf{C}\mathbf{v}_i^* = \lambda_i \mathbf{v}_i^*$	Lösung des Eigenwertproblems (Jacobi-Verfahren[198]) liefert g-1 nicht-negative Eigenwerte λ_i und Eigenvektoren $\mathbf{v}_i^*$
bzw. $\mathbf{C}\mathbf{V}^* = \mathbf{\Lambda}\mathbf{V}^*$	$\mathbf{V}^*$: Eigenvektormatrix, $\mathbf{\Lambda}$: Eigenwertdiagonalmatrix
$\mathbf{v}_i = \mathbf{D}^{-1}\mathbf{v}_i^*$	Rücktransformation liefert den gesuchten Gewichtsvektor $\mathbf{v}_i$

Die so erhaltenen Diskriminanzkoordinaten **y** werden in die Formel zur Berechnung des discriminant score f(**y**) eingesetzt und zur Klassifikation verwendet. Alternativ kann die Zuordnung auch zu der Klasse mit dem nächstliegenden Mittelpunkt erfolgen. Dazu werden die Klassenmittelpunkte ebenfalls in Diskriminanzkoordinaten transformiert und die Euklidischen Abstände berechnet.

Im Gegensatz zur verwandten Hauptkomponentenanalyse (Principal Component Analysis, PCA)[199] die zur Variablenorthogonalisierung bzw. Dimensionalitätsreduktion verwendet wird, liefert die Lösung des Eigenwertproblems bei der Kanonischen Analyse nur g-1 nicht-negative Eigenwerte und Eigenvektoren, da der Rang der Matrix **B** ebenfalls nur g-1 ist. Anschaulich bedeutet dies, dass zur Trennung von g Gruppen g-1 Trennlinien bzw. Hyperebenen erforderlich sind, die durch die kanonische Vektoren beschrieben werden.

5.2.11 Modellbewertung per Kreuzvalidierung (leave-one-out)

Wenn für die Erstellung von Klassifikations- oder Regressionsmodellen nur wenige Daten zur Verfügung stehen, ist es ungünstig, eine weitere Aufteilung in Trainings- und Testdaten (hold out sample) vorzunehmen. Zum einen fehlen die Daten bei der Modellerstellung und zum anderen ist die Abschätzung der Qualität des Modells, basierend auf einem kleinen Testdatensatz, unzureichend[200]. In diesem Fall sollte das leave-one-out-Verfahren herangezogen werden, bei dem der Reihe nach jede Verbindung mit einem separaten Modell vorhergesagt wird, das aus allen übrigen Verbindungen erstellt worden ist. Das leave-one-out-Verfahren kann also als Grenzfall der Aufteilung in Trainings- und Testdatensatz betrachtet werden, in dem letzterer aus nur einer einzelnen Verbindung besteht. Für n Verbindungen müssen n Modelle erzeugt werden.

Für große Datensätze ist die Selbstklassifizierung möglich, d.h. die Vorhersage aller Verbindungen mittels eines Modells, welches aus allen Verbindungen erstellt worden ist, oder die Aufteilung in einen großen Trainingsdatensatz und einen großen Testdatensatz. Ob ein Datensatz als groß oder klein zu betrachten ist, ist abhängig von der Anzahl der verwendeten Deskriptoren. In James M.[201] ist als Chemometrie-Faustregel beschrieben, dass ein Datensatz als groß gilt, wenn er mindestens den Faktor 10 mehr Verbindungen als Deskriptoren enthält. Zur Überprüfung der Regel wurden die Vorhersagegenauigkeiten für die Selbstklassifizierung mit denen aus leave-one-out-Rechnungen verglichen. Dazu wurden exemplarisch aus dem R05-Subtree zufällig ausgewählte Teilmengen verschiedener Grösse zur Modellerstellung und -bewertung verwendet.

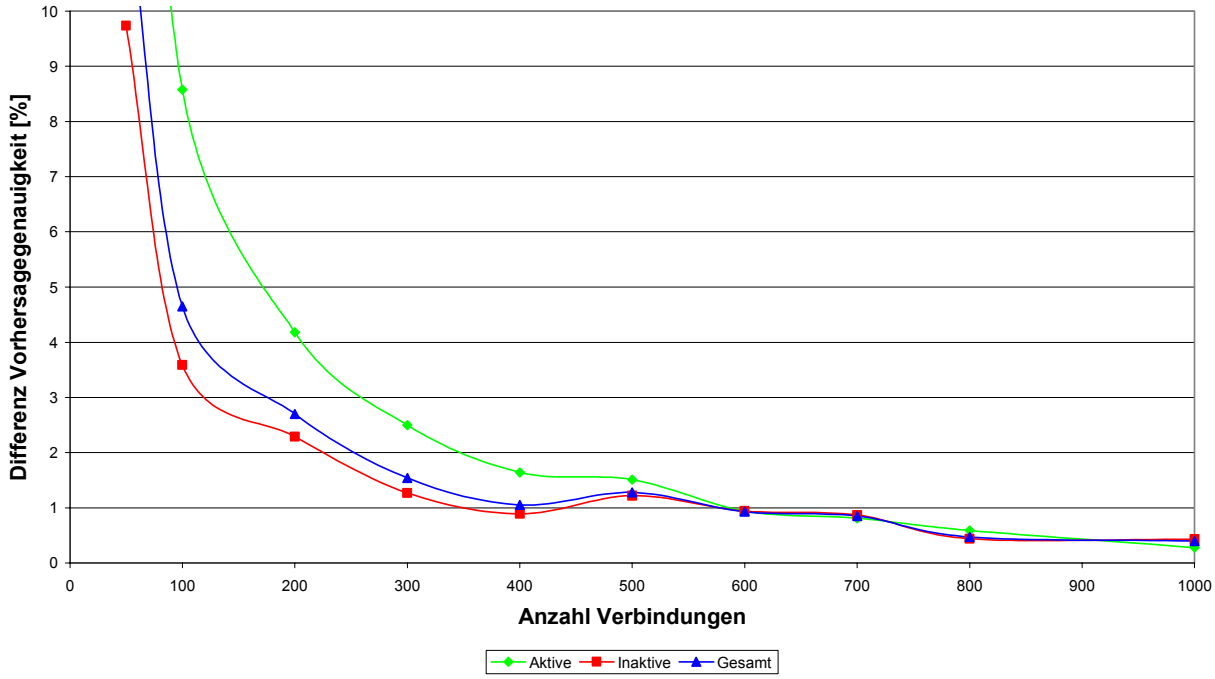


Abb. 5.27: Differenz der Vorhersagegenauigkeiten bei der Selbstklassifizierung und bei Verwendung von leave-one-out in Abhängigkeit von der Anzahl der im Modell enthaltenen Verbindungen.

Die in Abb. 5.27 veranschaulichten Werte sind jeweils Mittelwerte aus fünf unabhängigen LDA-Rechnungen mit 13 Deskriptoren. Entsprechend der Faustregel ist zu erwarten, dass die Differenzen der Vorhersagegenauigkeiten ab einer Datensatzgröße von 130 Verbindungen (= Faktor 10 * 13 Deskriptoren) vernachlässigbar sind. Im Durchschnitt liegt die Abweichung allerdings noch bei ca. 5%. Um diesen Wert auf unter 1% zu verringern und damit auf der sicheren Seite zu sein, wird in BayTree ein erhöhter Faktor von 50 verwendet. Bei Verwendung von 13 Deskriptoren werden also bis zu einer Datensatzgröße von 650 Verbindungen die Vorhersagen auf der Basis von leave-one-out-Modellen erstellt und bei mehr Verbindungen unter Verwendung nur eines einzelnen Modells.*

5.2.12 Lückenanalyse im Strukturbaum der Eingabedaten

Auf der Basis des MolCode können im Prinzip Lücken im vorhandenen Strukturraum der Eingabedaten leicht identifiziert werden. Der Strukturraum wird durch die Gesamtheit aller MolCodes definiert. Durch eine Auflistung der Werte rr bei Rrr bzw. ll bei Lll an einer bestimmten Position des MolCode zeigen fehlende Werte der Reihe die strukturellen Lücken gleich priorisierter Fragmente auf.

* Durch Eingabe des Befehls `set ::baytree::expertconfig(leaving_one_out) 1` in der BayTree-Kommandozeile wird die Verwendung von leave-one-out-Modellen unabhängig von der Datensatzgröße erzwungen.

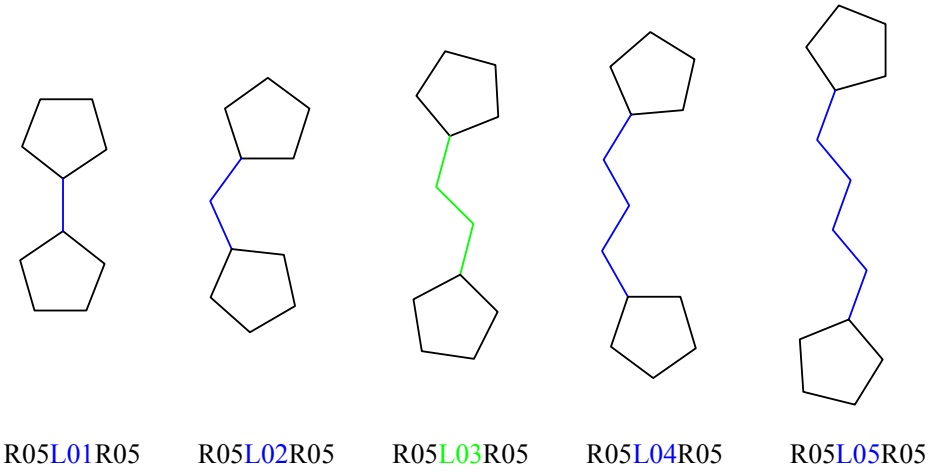


Abb. 5.28: Lückenanalyse anhand der MolCodes R05L01R05 ..... R05L05R05.

Bei der praktischen Durchführung der Lückenanalyse ist allerdings darauf zu achten, dass niederpriorisierte Fragmente durch die chemische Funktionalisierung ihre (relative) Priorisierung ändern können und deshalb im Topologie-Baum auch an anderer Stelle positioniert sein können.

Mathematisch vollständiges Enumerieren von Strukturen zu einer gegebenen Summenformel ist z.B. mit den Programmen SMOG[202] oder MOLGEN[203] möglich. Der große Nachteil dieser Ansätze ist, dass der Großteil der Strukturen zwar graphentheoretisch korrekt, aber chemisch uninteressant ist. Dies macht nachgeschaltete Filterverfahren erforderlich, die den überwiegenden Teil der erzeugten Strukturen wieder löschen. Bei neueren Versionen der beiden Programme wird deshalb intern mit goodlists und badlists zur Verbesserung der ausgegebenen Strukturen gearbeitet. Ausgehend von einem existierenden Algorithmus zur Erzeugung von polyzyklischen Ketten[204]wird im Rahmen einer Diplomarbeit an der Uni Bielefeld im Arbeitskreis Brinkmann derzeit eine effiziente Lösung gesucht[205].

5.3 Implementierungsdetails

5.3.1 Die Scriptsprache Tcl/Tk

BayTree ist für IRIX 6.5 und Windows entwickelt worden und in wesentlichen Teilen in Tcl/Tk (V8.2)[206] geschrieben. Zeitkritische Routinen sind als dynamisch ladbare Extensions[207] in C/C++[208, 209] kodiert. Die universell einsetzbare Scriptsprache Tcl (Tool Command Language) ist Ende der 80er Jahre von John Ousterhout[210] an der Berkeley Universität konzipiert worden*. Der Tcl-Interpreter besteht aus einer Library von C-Funktionen, die neben der eigentlichen Implementierung der Sprache eine Reihe von

* Einige grundlegende interne Veränderungen haben in den Versionen ab 8.0 http://www.tcl.tk/software/tcltk/relnotes/tcl8.0a1.txt (Dezember 1996) zu einer deutlichen Beschleunigung (bis zu Faktor 10) der Laufzeit geführt: die Einführung des Bytecode-Compilers und eines alternativen Objekt-Mechanismus in das C API zur Übergabe von Argumenten, der das wiederholte Konvertieren der Zeichenketten in andere Datentypen überflüssig macht.

Funktionen enthält, mit der der Befehlsschatz von Tcl beliebig erweitert werden kann. Die wichtigste Erweiterung stellt Tk (Toolkit für das X-Windows System) dar. Sie ermöglicht es, Applikationen mit graphischen Benutzerschnittstellen auf einem hohen Abstraktionsniveau zu schreiben. Ein einfaches GUI besteht nur aus wenigen Zeilen Tcl/Tk-Code und im Vergleich zu C-basierten GUI-Toolkits[211] muss man sich um weniger Details kümmern. Der aus Tcl und Tk bestehende Sprachkern liegt als open source im Quellcode vor (http://tcl.sourceforge.net/), läßt sich für alle relevanten Plattformen und Betriebsysteme (Unix/X11, Microsoft Windows, MacOS) kompilieren und ist daher universell verfügbar.

Die Sprache Tcl hat eine feste Benutzergemeinschaft, die den Sprachkern in Form von TIP Tcl Improvement Proposals (http://www.purl.org/tcl/tip/) und eines Tcl Core Teams kontinuierlich weiterentwickelt. Zusätzlich verwaltet sie Internetressourcen wie die Website www.tcl.tk, die Tcl Developer Xchange Website http://tcl.ActiveState.com/software/tcltk/, die Newsgruppe comp.lang.tcl und das Wiki-Board http://mini.net/tcl. Tcl wurde in den letzten Jahren in ihrem Beliebtheitsgrad etwas von Perl, Python und PHP verdrängt[212]. Nichtsdestotrotz werden alle modernen Technologien wie XML, CORBA durch entsprechende Erweiterungen TclXML[213], TclXSLT, TclDOM[214] und Combat[215, 216] unterstützt. Seit dem Aufkommen von SourceForge.net als universelle Entwicklungsumgebung und Host für Open-Source Projekte sind fast alle Extensions dort zu finden. Davor mussten die aktuellsten Versionen von den über das Internet verstreuten Websites der jeweiligen Autoren zeitaufwendig zusammengesucht werden. Eine komplette binäre BI-Distribution (batteries-included distribution), in der viele Extensions integriert sind, ist seit April 2001 für die Plattformen Windows, Linux und Solaris kostenlos von ActiveState erhältlich[217]. Für die MIPS-Plattform von SGI mit Irix 6.5 steht eine derartig umfangreiche und gepflegte Distribution nicht zur Verfügung, so dass die Erweiterungen individuell zusammengestellt und kompiliert werden müssen.

5.3.2 Das Cactvs-System

Eine Ausnahme stellt die Cactvs-Distribution von Xemistry[218] dar, die neben Linux, Solaris und Windows ebenfalls für Irix zur Verfügung steht. Sie enthält zusätzlich zur eigentlichen Cactvs-Extension einen Tcl/Tk-Interpreter, die entsprechenden Bibliotheken, Anwendungsscripte (z.B. Browser csbr und Sketcher csed für chemische Strukturen) und zusätzlich eine Auswahl der wichtigsten Erweiterungen. Im Kern ist die Cactvs-Extension ein Toolkit für das Handling von chemischen Strukturen[219, 220].

Die in BayTree verwendeten Kommandos und Funktionalitäten des Cactvs-Toolkits sind in Tabelle 5.2 und Tabelle 5.3 mit einer kurzen Beschreibung aufgelistet. Äquivalente Alternativen finden sich im Daylight Toolkit[221] in Verbindung mit dem DayTcl-Wrapper[+] von Dalke Scientific Software[222] oder in der OpenEye Library OELib[223] von Eyesopen in Verbindung mit einem universellen Tcl-Wrapper wie SWIG[224] für C/C++-Bibliotheken.

Property	**Beschreibung**
A_XY	atom xy coordinates (on the fly-calculation if needed)
B_ATOMS	atoms pairs connected by bonds

[+] Ein Wrapper erzeugt zu einer bestehenden Bibliothek von C/C++-Funktionen Tcl-Kommandos und kümmert sich um die erforderliche Anpassung der Argumente. Dadurch kann die Funktionalität der Bibliothek auf der Scripting-Ebene verwendet werden.

B_ORDER	order of bonds
E_HASH	hash value
E_NATOMS	number of atoms
E_NBONDS	number of bonds
E_NMOLECULES	number of unconnected molecule parts (e. g. counterions)
E_NRINGS	number of rings
E_SMILES	smiles string for molecule (not vital, only for testing!)
E_XYEXTENT	molecule coordinate extension
E_GIF	gif-file for molecule depiction
A_ELEMENT	element of atom
A_NEIGHBORS	number of neighbors
A_FORMAL_CHARGE	formal charge of atom

Tabelle 5.2: Verwendete Properties des Cactvs-Toolkits.

Kommando	**Beschreibung**
molfile open $slnfile molfile read $mfhandle	Read molecule from SD-File to internal representation
molfile string $ehandle format sln	Create SLN-string for molecule
ens create $smiles	Create molecule from SMILES-string (only for testing purposes)
ens copy $ehandle_neu $ehandle ens dup $ehandle ens delete $ehandle	Copy/Duplicate/Delete a molecule
ens atoms $ehandle ens bonds $ehandle	List atom and bond numbers
ens hadd $ehandle ens hstrip $ehandle	Add or Remove all Hydrogens
ens get $ehandle PROPERTY	Get molecule-related information
atom delete $ehandle $a	Delete atom from molecule
atom get $ehandle $a PROPERTY atom set $ehandle $a PROPERTY	Get/Set atom-related information
plotatom plotbond	Draw lines for bonds and symbols for atoms
match ss	Substructure matching (subgraph isomorphism)

Tabelle 5.3: Verwendete Kommandos des Cactvs-Toolkits.

Alle IRIX 6.5 Versionen ab 3.112 (getestet bis 3.206) sind verwendbar. In BayTree sind für einige Bugs früherer Versionen Workarounds integriert, diese sind in aktuelleren Versionen teilweise nicht mehr erforderlich, da die Fehler in Cactvs beseitigt worden sind. Für Windows kommt die Win32 V3.195 zum Einsatz.

5.3.3 Erforderliche Kommandoerweiterungen und Zusatzprogramme

Zur Ausführung des BayTree-Programms sind zusätzlich folgende Tcl-Extensions erforderlich:

Paketname	Version	Funktionalität	Quelle und Dokumentation
TclX	8.2	Unix Signalhandling, Listenmanipulation, Keyed List	Lehenbauer K., Diekhans M., TclX: Tcl Extension, http://www.neosoft.com/TclX/
BLT	2.4u	Datenvektoren, 2D-Graph Widget, Event-Handling, tabset-Widget	Howlett, G. A., http://www.tcl.tk/blt/ http://sourceforge.net/projects/blt/
Tix	4.1.0	zusätzliche Widgets wie ComboBox und Baloon-Help	Lam I., Tix: Tk Interface eXtension, http://tix.sourceforge.net
TkTable	2.6	2D Tabellen-Widget	Hobbs J., http://tktable.sourceforge.net
Oratcl (+)	3.3	Zugriff auf ORACLE-Datenbankserver	Poindexter T., Helfter Todd, Oratcl: Oracle Database Server access commands for Tcl http://oratcl.sourceforge.net
mclistbox (+)	1.02	Listenfeld-Widget für mehrere Spalten	Oakley B., enthalten in TclLib http://tcllib.sourceforge.net/
wmf, gdi, hdc (+)	0.4.0.2, 0.9.9.11, 0.2.0.1	Erzeugung von EMF-Files (nur für Windows Version)	Schwart M., Tcl Extensions http://www.du.edu/~mschwart/tcl-tk.htm

(+) nicht in Cactvs-Distribution enthalten

Als Workaround für fehlerhaftes SLN-Parsing im Cactvs-Toolkit und zur Erzeugung von XY-Koordinaten entsprechend der Tripos-Regeln wird das Unity-Programm dbtranslate verwendet[225]. Dafür müssen die Environment-Variablen TA_3DB oder TA_ROOT und TA_3DB_TABLES definiert sein.

5.3.4 Anmerkungen zu Java

Zur Einbindung von Java-Klassen und Bibliotheken ist die TclBlend-Extension[226, 227] entwickelt worden. Mit ihrer Hilfe kann Javacode auf der Tcl-Scripting-Ebene verwendet werden. Interessante Bibliotheken aus dem Bereich Chemie stellen JCHEM[228] und CDK (Chemistry Development Kit)[229] dar. Eine vollständige Implementierung vieler Machine Learning-Algorithmen ist in Weka (Waikato Environment for Knowledge Analysis)[230] enthalten.

Java als Entwicklungssprache für BayTree kam nicht in Frage, da

- das Look & Feel von Java-GUIs auf der hauptsächlich verwendeten Zielplattform IRIX bislang inakzeptabel ist[231]
- die Geschwindigkeit rechenintensiver Routinen mit der unter IRIX verfügbaren JVM im Vergleich zu C/C++ zu langsam ist
- keine dem Tk-Canvas äquivalente Funktionaliät mit vergleichbarer Geschwindigkeit zur Verfügung steht[232]
- keine schnellen Entwicklungs-Test-Zyklen und einfache Erweiterungen wie bei einer Skriptsprache möglich sind.

5.4 Erweiterungsmöglichkeiten

BayTree kann wie jedes Programm noch weiterentwickelt werden. Mögliche zukünftige Ergänzungen sind sowohl im technologischen als auch im methodischen Bereich denkbar:

- Direkte Anbindung an die BAYTREE-Datenbank zum Wiedereinlesen der Ergebnisse unter Berücksichtigung von Filterbedingungen ohne Umweg über ein zu erzeugendes Hitlistfile(+)
- Verfahren zur Variablenselektion (forward stepwise) bei der Linearen Diskriminanzanalyse(+)
- Berechnung und Verwendung zusätzlicher alternativer Deskriptoren(+)
- Berücksichtigung physikochemischer Eigenschaften bei der Erstellung der XR-Table
- Einsatz modifizierter oder alternativer Regeln zur Priorisierung, z.B. indem die pharmakophoren Dekorationen bei der Priorisierung an erster Stelle berücksichtigt werden
- Nachbearbeitung des Baums unter Verletzung der Hierarchie und Anordnung durch den MolCode: Repositionierung von Singletons, Berücksichtigung von Bioisosterien oder vorhandenem Zusatzwissen, das nicht im Regelwerk berücksichtigt worden ist oder berücksichtigt werden kann
- Einbindung der Verfahren aus TOSS-MODE zur Bewertung von Substrukturfragmenten

(+) Es ist geplant, diese Punkte durch die Integration in die PIX-Plattform[233] der Business Unit Pharma der Bayer AG abzudecken.

6 Anwendung auf den NCI Aids-Datensatz

6.1 Beschreibung des NCI Aids-Datensatzes

Chemische und biologische Screeningdaten wurden vom National Cancer Institute (NCI) Bethesda, MD, USA, im Rahmen des Developmental Therapeutics Program (DTP)[234] erzeugt und sind von dort zu erhalten. Der Hauptzweck des DTP ist es, die Entdeckung neuer Therapeutika für die Behandlung von Krebs und AIDS (acquired immuno deficiency syndrome) zu erleichtern, indem sie Screeningkapazität für User zur Verfügung stellt, die über interessante chemische Bibliotheken oder Naturstoffextrakte verfügen. Der öffentlich verfügbare Gesamtdatensatz enthält 249 081 Moleküle und damit verknüpfte Screening-ergebnisse für Krebs (Stand August 1999) und/oder AIDS (Stand Oktober 1999)[235]. Aus den Strukturen wurde eine Unity-Datenbank[236] erstellt und die AIDS-relevante Teilmenge extrahiert, was zu einem reduzierten Datensatz der Größe 42 550 führt. Die biologischen Werte für diese Moleküle wurden einem zellbasierten Assay entnommen, der den Schutz vor HIV-1 Infektion misst. Zum Einsatz kam ein löslicher Formazan-Assay, durch den kalorimetrisch bestimmt werden kann, wie stark die jeweilige Verbindung menschliche CEM Zellen vor dem HIV-1 induzierten Zelltod schützt[237]. Da es sich um einen funktionalen, also mechanistisch unselektiven Assay handelt, kann nicht zwischen nukleosidischen (NRTI) bzw. nicht-nukleosidischen Reverse-Transcriptase-Inhibitoren (NNRTI), Protease-Hemmern (HIVPR) und HIV-Integrase-Inhibitoren unterschieden werden[238]. Dies war auch der Grund dafür, dass Screeningaktivitäten größeren Umfangs unter Verwendung dieses Assays eingestellt wurden.

Jedes Molekül ist einer von drei möglichen Kategorien zugeordnet:

Kategorie	Kenn-zeichnung	Beschreibung	Numerische Bezeichnung	Anzahl Vertreter
aktiv	CA	confirmed active	2	423
schwach aktiv	CM	confirmed moderately active	1	1078
inaktiv	CI	confirmed inactive	0	41049

Die mit CA gekennzeichneten Verbindungen bieten 100% Schutz, die mit CM gekennzeichenten Verbindungen bieten mindestens 50% Schutz und mit CI sind die übrigen Verbindungen gekennzeichnet, die entweder keinen Schutz bieten oder sogar toxisch für die verwendeten CEM-Zellen sind. Die Kategorien schwach aktiv und inaktiv wurde in einer Klasse kombiniert und werden im folgenden als Inaktive bezeichnet. Es liegen damit 423 aktive und 42127 inaktive Verbindungen vor. Die Strukturinformation lag als 0D SMILES vor und wurde ins SLN-Format konvertiert.

Aus diesen zur Verfügung stehenden Rohdaten wurde ein Beispieldatensatz von 1983 Verbindungen zusammengestellt, der alle 423 Aktiven und 1560 zufällig ausgewählte Verbindungen der Inaktiven enthält. Dazu wurde ein Hitlistfile erzeugt, das als RegId die NSC (die interne ID-Nummer des NCI) verwendet und das Attribut **act** für die Aktivitätskategorie des AIDS-Screens enthält. Das Einlesen in BayTree benötigt auf einem Pentium4 1,8GHz mit 512 MB Hauptspeicher 4:05 Minuten.

6.2 Topological Structure Tree

Der TST zum Datensatz enthält 1170 unterschiedliche TSPs, von denen der größere Teil der 633 Knoten (54 %) populiert ist. Die nichtpopulierten Knoten repräsentieren nicht enthaltene Frameworks, die entlang der MolCode/TSP-Pfade erzeugt wurden. 84% der Knoten (984) werden durch ein einzelnes Framework repräsentiert. In den übrigen Knoten sind bis maximal acht unterschiedliche Konstitutionsisomere zusammengefasst. Diese werden erst mittels zusätzlicher Module im MolCode weiter aufgesplittet. Ein Extrembeispiel stellt der Knoten R06I02R06I02R06I02R06 (siehe Abb. 6.1) dar. Er enthält acht Möglichkeiten, vier Sechsringe zu anellieren.

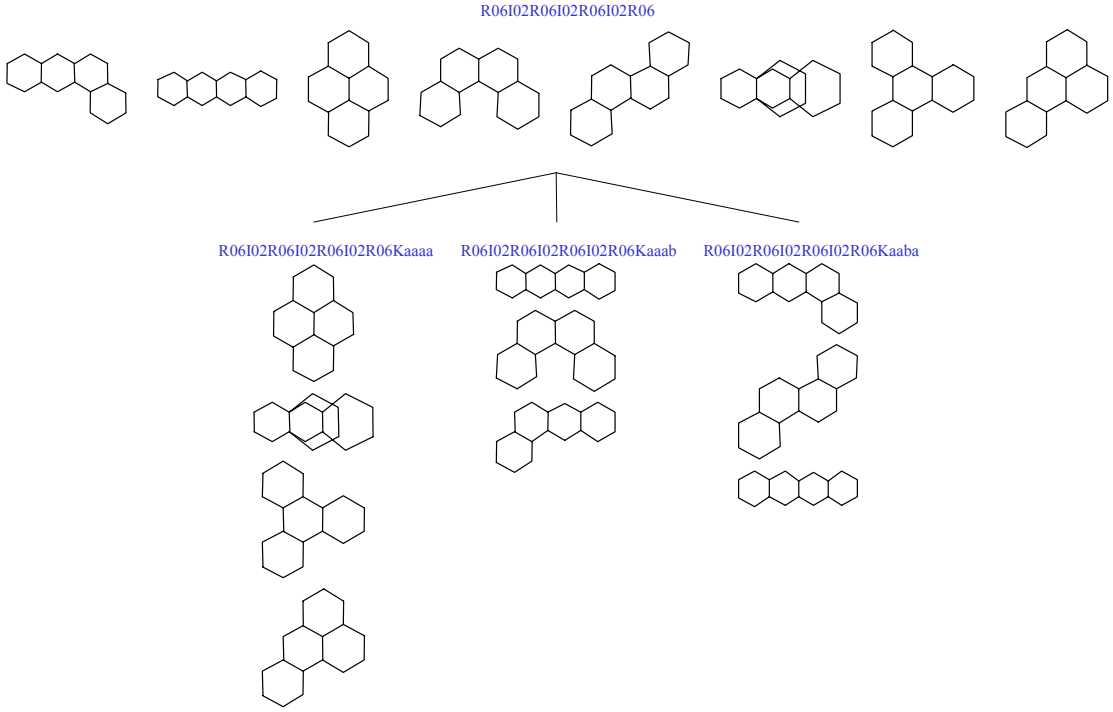


Abb. 6.1: Konstitutionsisomere des Knotens R06I02R06I02R06I02R06. Das Naphthacen- und Benzo[a]anthracen-Gerüst ist aufgrund von dekorationsbedingtem Prioritätswechsel der Ringe sowohl unter ...Kaaab als auch ...Kaaba aufgeführt.

Die Häufigkeit des Auftretens von Konstitutionsisomeren stellt sich wie folgt dar:

Anzahl Konstitutionsisomere	Anzahl der Knoten
1	984
2	129
3	32
4	14
5	7
6	3
8	1

Der schematische Überblick über den erzeugten Gesamtbaum des Datensatzes ist in Abb. 6.2 abgebildet. Die Wurzelknoten der Unterbäume sind gekennzeichnet.

Die Aufteilung der Verbindungen auf die einzelnen Unterbäume ist in Tabelle 6.1 aufgeführt. Als Zeilenbezeichner werden die Wurzelknoten der Unterbäume verwendet. 92,4 % (1827) der Daten sind in den Unterbäumen R05, R06 und R07 zu finden. Bei den übrigen 151 Verbindungen handelt es sich vor allem um „ungewöhnliche“ Verbindungen, die für die Datenauswertung keine größere Rolle spielen. In der Tabelle 6.1 sind Verweise auf die Positionen im Baum und exemplarische RegIds aufgeführt.

Unterbaum	Gesamtzahl Verbdg.	Position im Baum	Beispielverbindung
R00	71		Abb. 6.3
R03	25		Abb. 6.4
R04	29		Abb. 6.5
R05	686	---	---
R06	1099	---	---
R07	42	---	---
R08	6	11	628551
R09	4	12	669146
R10	1	13	281604
R11	1	14	636589
R12	3		
R14	1		
R16	4	15	681456
R17	3	16	636592
R24	3	17	7229
R29	1		
R33	1	18	290193
R35	2		
R36	1		

Tabelle 6.1: Aufteilung der Verbindungen auf die Unterbäume. Die Beispielverbindungen zu den Unterbäumen R08 bis R36 sind in Abb. 6.6 zusammengefasst.

R00 ist der Sammelknoten für alle aliphatischen Verbindungen des Datensatzes (siehe Abb. 6.3).

Die meisten der Verbindungen mit Metallatomen sind aufgrund von konkreten Bindungen in den Koordinationspolyedern im Unterbaum R03 positioniert.

Ab Ringgröße R09 sind die Naturstoffe wie Makrolide und Polypeptide aufgelistet. Beispielverbindungen sind in Abb. 6.6 aufgeführt.

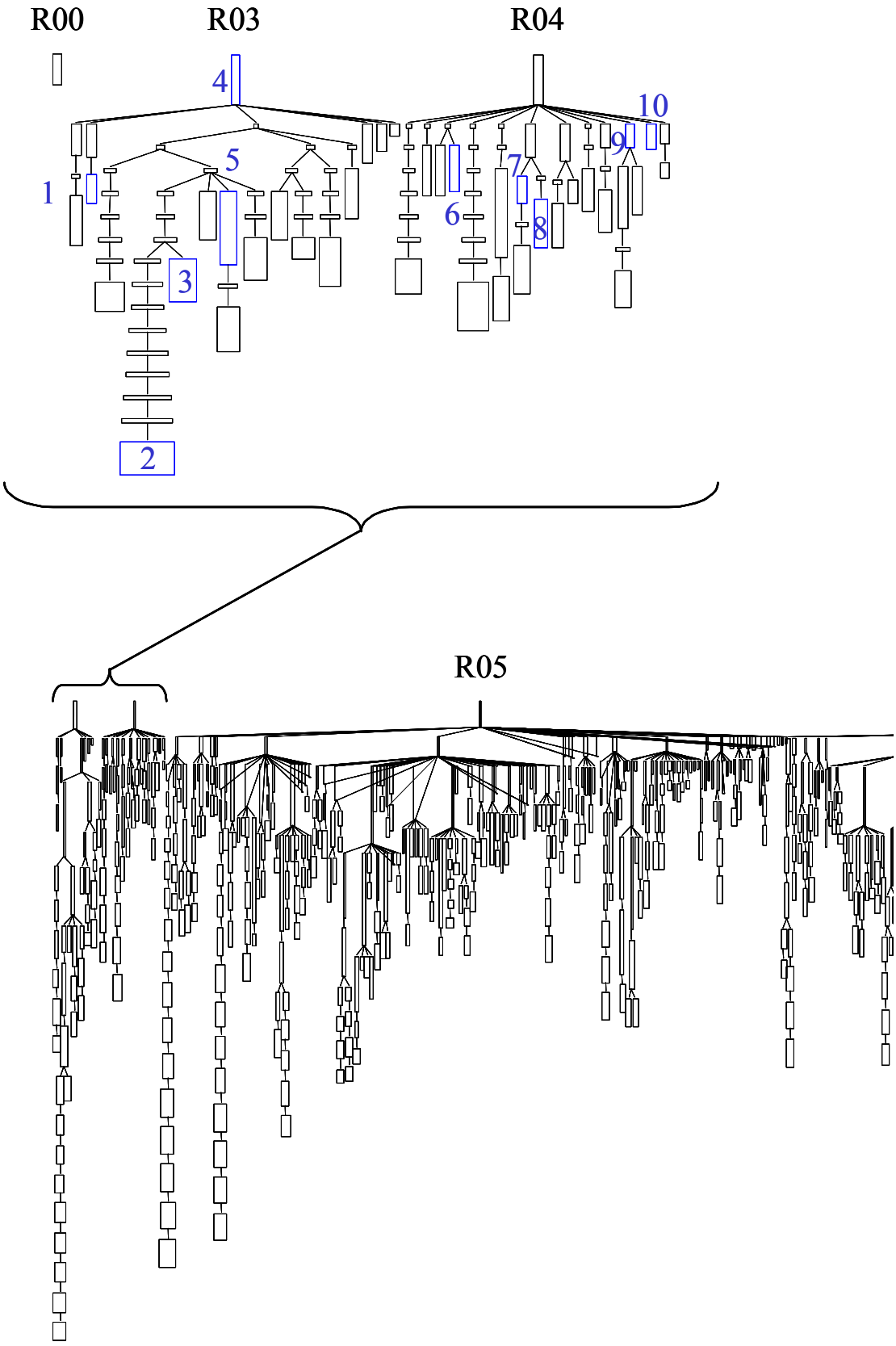
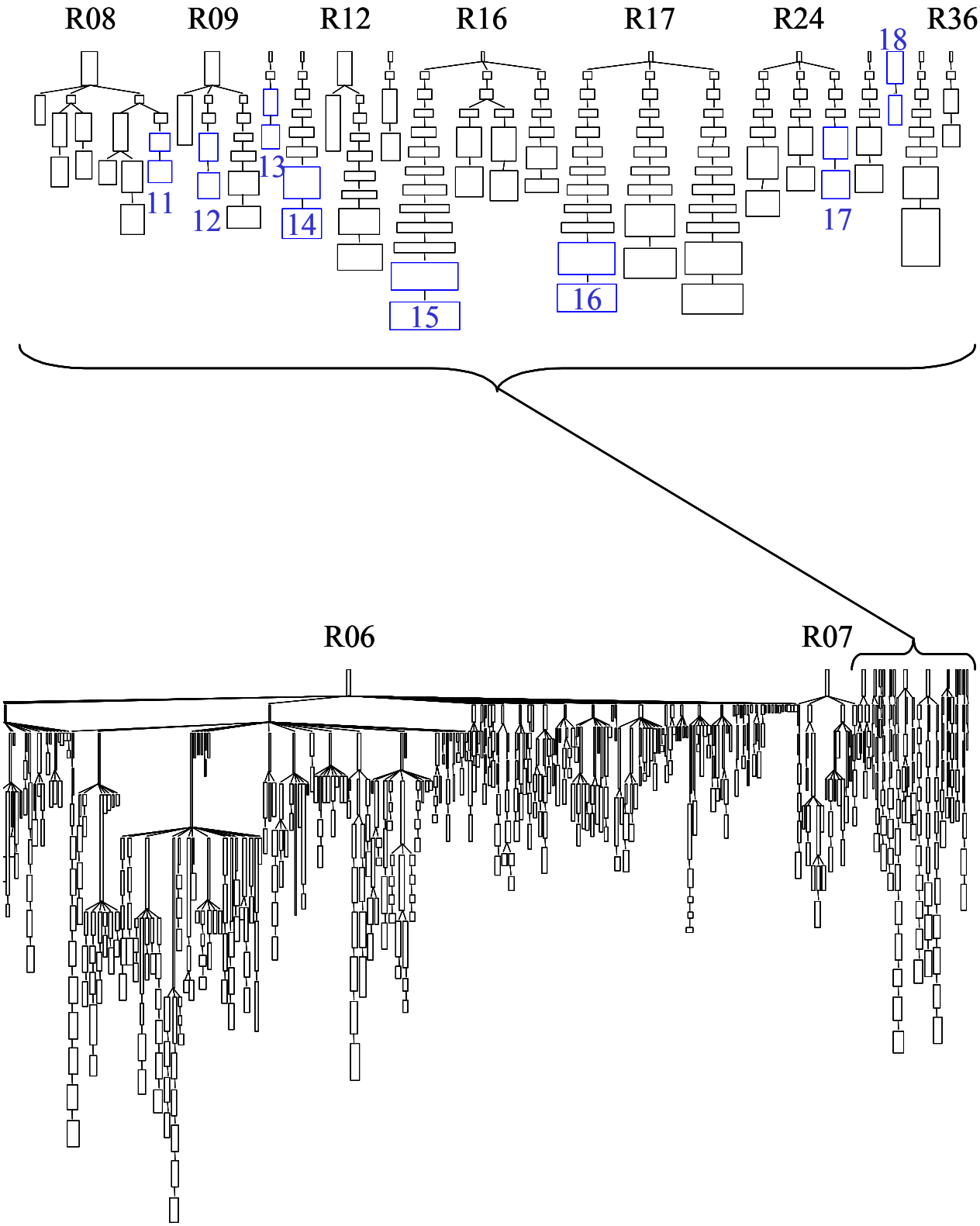


Abb. 6.2: Globalview des Gesamtbaums.


R08
R09
R12
R16
R17
R24
R36
11
12
13
14
15
16
17
18
R06
R07

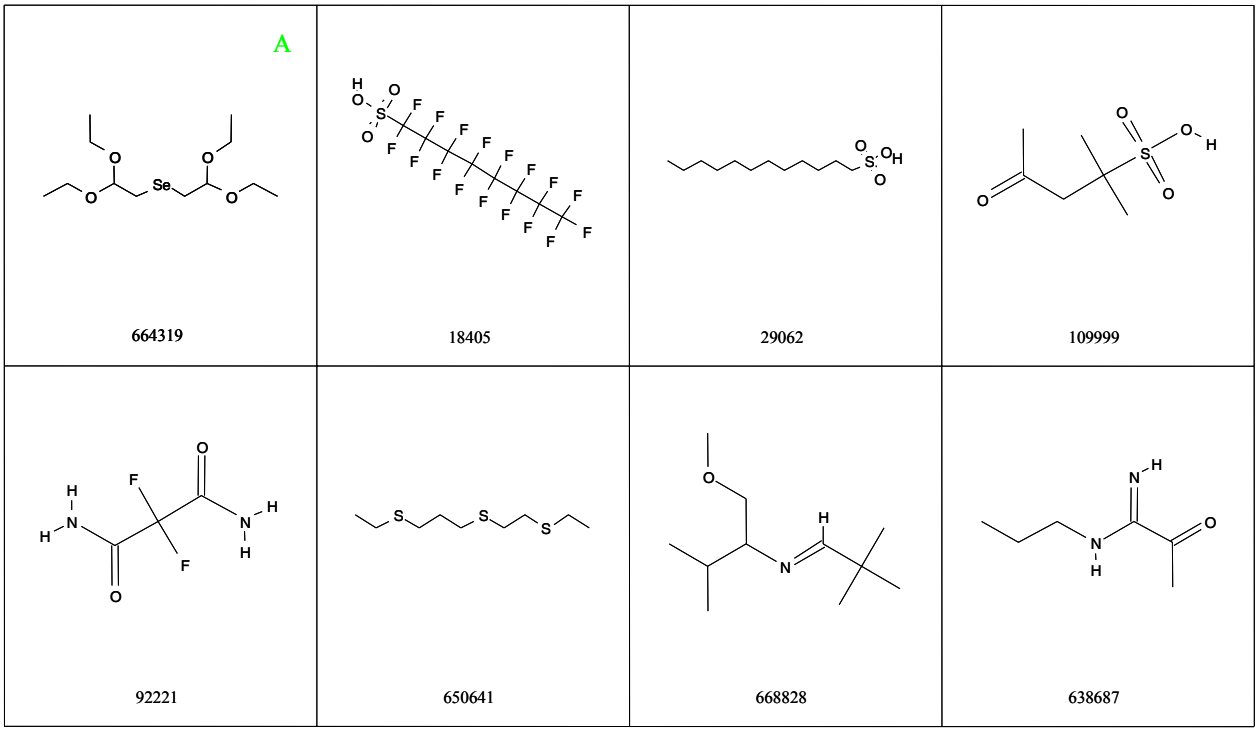


Abb. 6.3: Aliphatische Verbindungen im Datensatz; Beispiele aus dem Subtree R00.

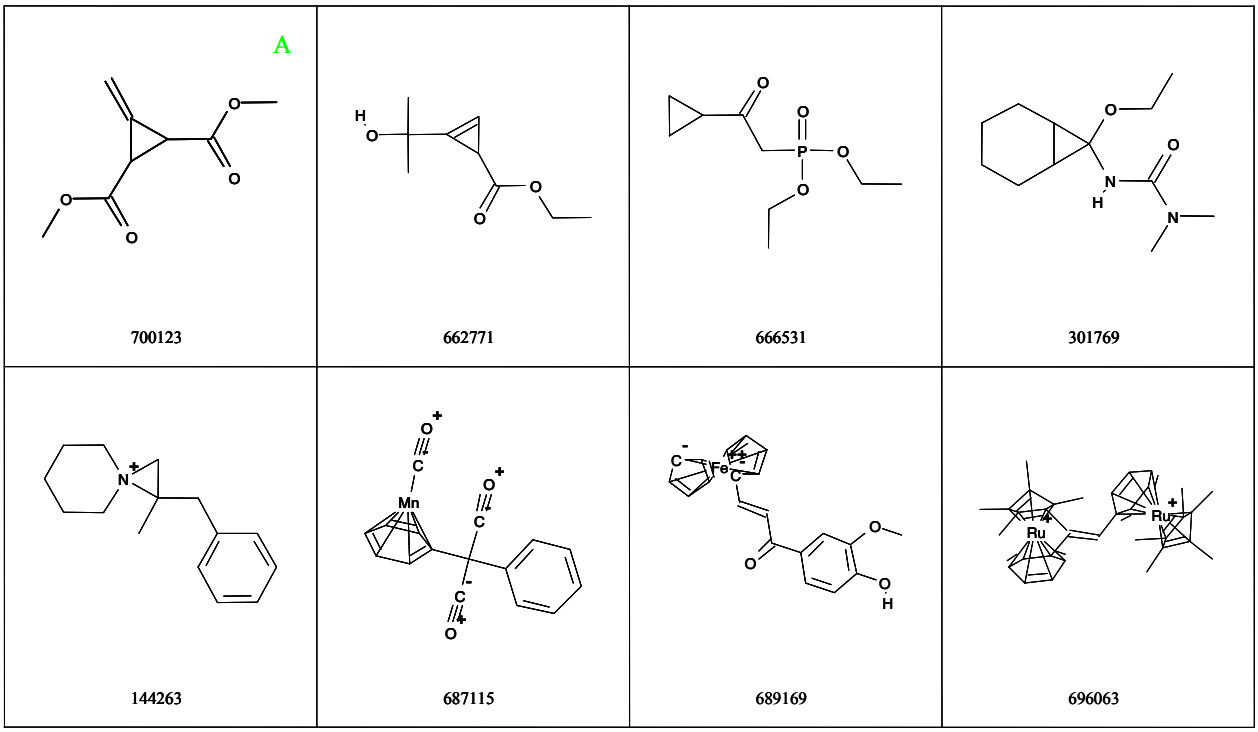


Abb. 6.4: Beispielverbindungen aus dem Subtree R03.

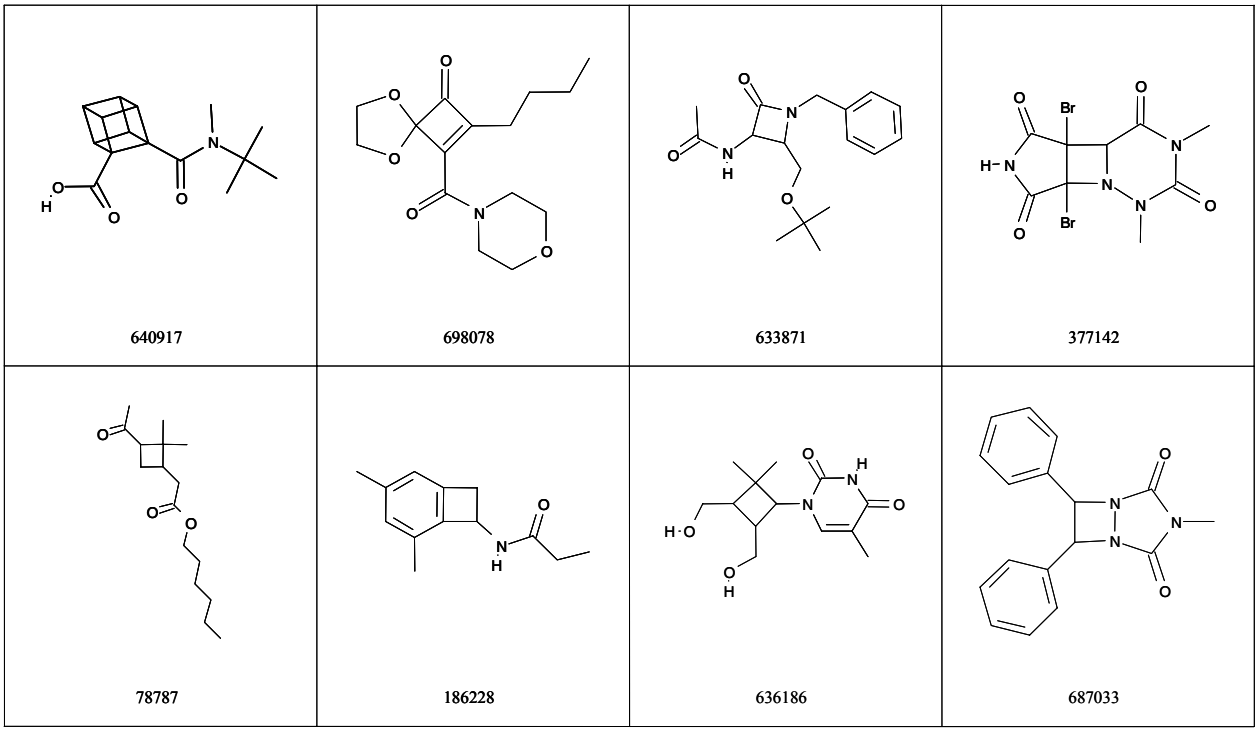


Abb. 6.5: Beispielverbindungen aus dem Subtree R04.

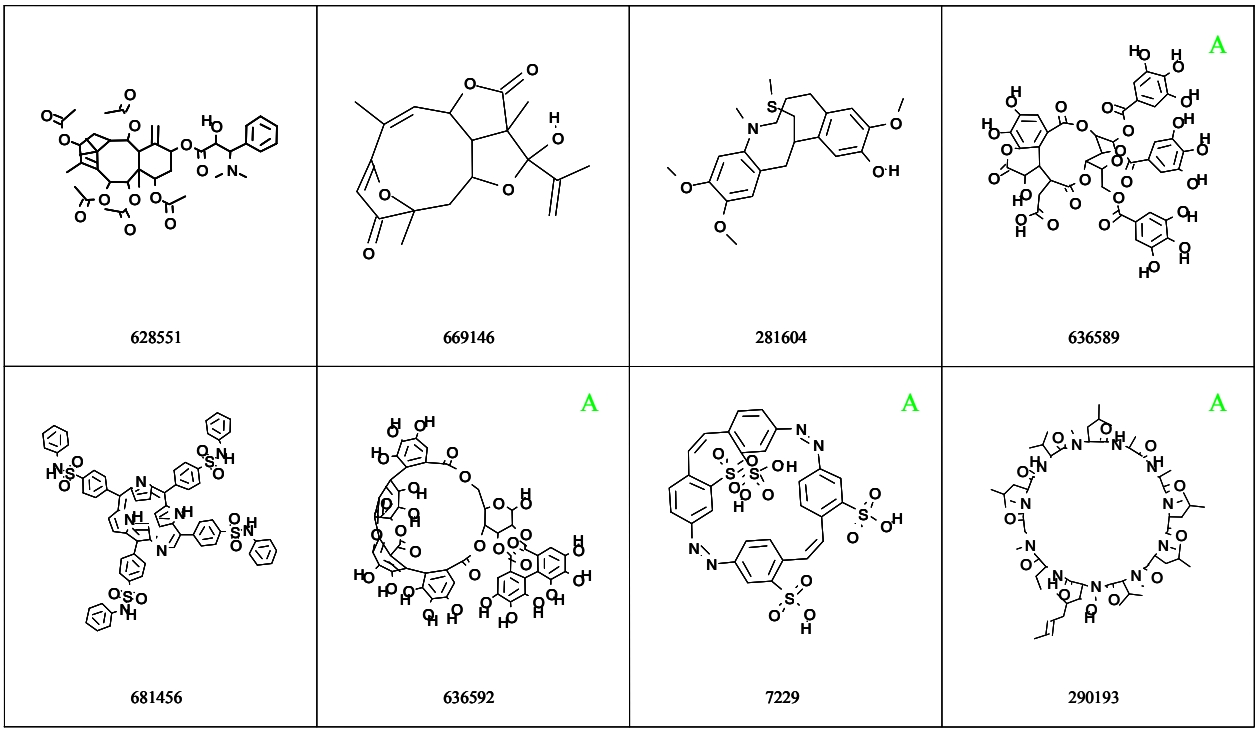


Abb. 6.6: Beispielverbindungen aus den Subtrees R08-R36, die Zuordnung ist Tabelle 6.1 zu entnehmen.

Die Subtrees R05-R07 enthalten 97,2% der aktiven Verbindungen und nur der Rest von 2,8% (12) Aktiven findet sich in den übrigen Subtrees. Letztere sind in den Abb. 6.3 bis Abb. 6.6 mit „A“ gekennzeichnet. Verbindungen, die nur geringfügig modifizierte Derivate darstellen oder deren Strukturbilder in der Tabellenzelle nicht erkennbar sind, werden nicht gesondert aufgeführt.

In allen Fällen ist eine unspezifische Wirkung auf des Testsystem anzunehmen. Deshalb finden die Verbindungen im Weiteren keine Berücksichtigung.

6.3 Generic Topological Structure Tree

Die maximale Abstaktionsstufe der Frameworks wird durch den generic MolCode beschrieben (siehe Seite 133). Dieser enthält die Abfolge der ggf. anellierten Ringsysteme und Linker unabhängig von deren spezifischer Größe bzw. Länge.

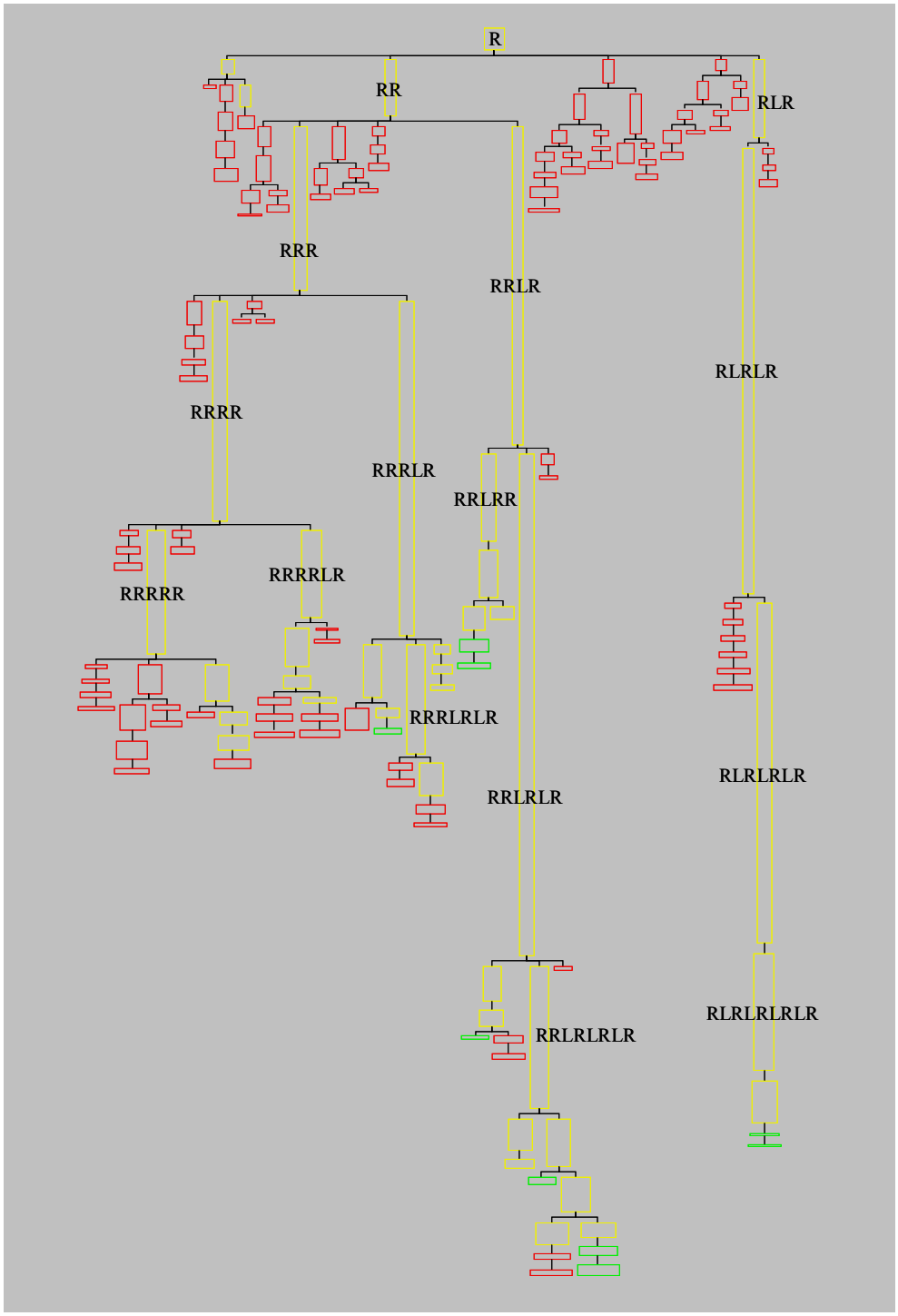


Abb. 6.7: Globalview des TST, basierend auf 104 generic MolCodes.

Die Abb. 6.7 enthält den Globalview des TST. Die Länge der Rechtecke ist proportional zur Anzahl der vorhandenen Frameworks. Bestimmend sind die stark populierten Mischknoten (dargestellt in Gelb), die sowohl aktive als auch inaktive Frameworks bzw. Verbindungen enthalten. Vereinzelt vertretene Frameworks zu ungewöhnlichen generic MolCodes (I01, I03, I04) enthalten in der Regel ausschliesslich inaktive Verbindungen. Der generic Molcode eignet sich daher je nach Fragestellung als Filter, um auf besondere Template aufmerksam zu werden oder um sie auszusortieren.

Die Tabelle 6.2 enthält die Knoten mit mindestens 10 aktiven Verbindungen. Neben den einfachen Systemen wie R und RLR fallen auch hier die häufig vorhandenen langezogenen Moleküle RRLRLRLRR, RLRLRLR und RLRLR auf. Mögliche Strukturen für die MolCodes R, RLR RLRLR und RRLR sind in Abb. 6.8 skizziert.

MolCode (generic)	Anzahl Aktive	Anzahl Inaktive	Knotenbelegung	Anteil Aktiver [%]
RLR	140	236	376	37.23
RRLRLRLRR	39	1	40	97.50
RLRLRLR	32	57	89	35.96
RLRLR	30	125	155	19.35
RRLR	20	152	172	11.63
RRRRLRLR	14	6	20	70.00
RRRLR	14	51	65	21.54
RRRR	13	51	64	20.31
RRLRLRLRLRLRR	12	0	12	100.00
RRRLRRLRR	12	1	13	92.31
RLRLRLRLR	11	14	25	44.00
RRR	11	75	86	12.79
R	11	191	202	5.45

Tabelle 6.2: generic MolCode-Knoten mit mindestens 10 aktiven Verbindungen.

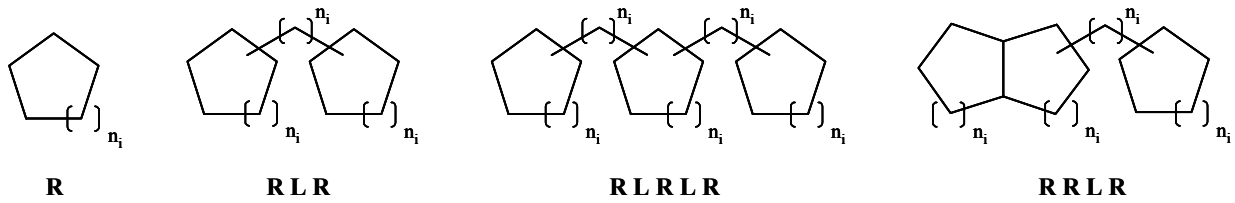


Abb. 6.8: Markush-Formeln mit möglichen Strukturen zu den generic MolCodes R, RLR, RLRLR und RRLR

6.4 Analyse des Belegungsgrades der Knoten

Für eine erste Analyse des Datensatzes eignet sich die Betrachtung der Belegungsgrade der Knoten (siehe Abb. 6.9). Die Häufigkeitsverteilung der Framework Occupancy ist Tabelle 6.3 zu entnehmen.

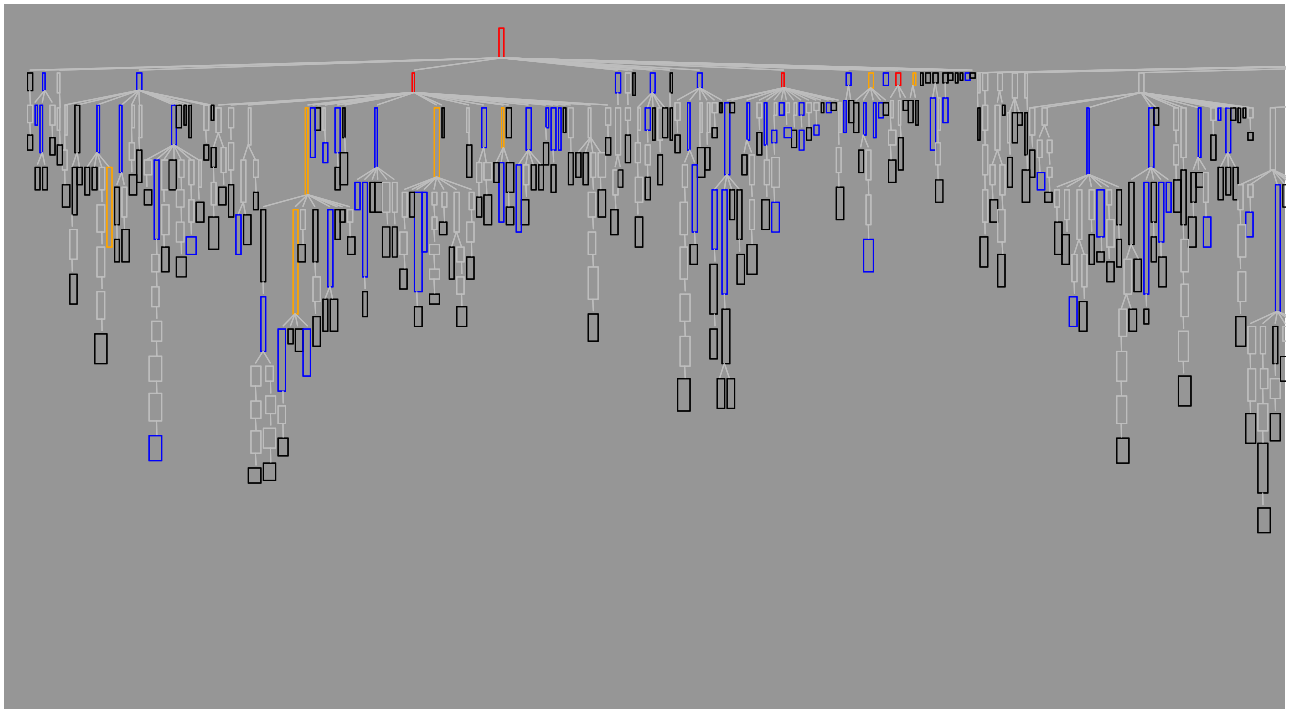
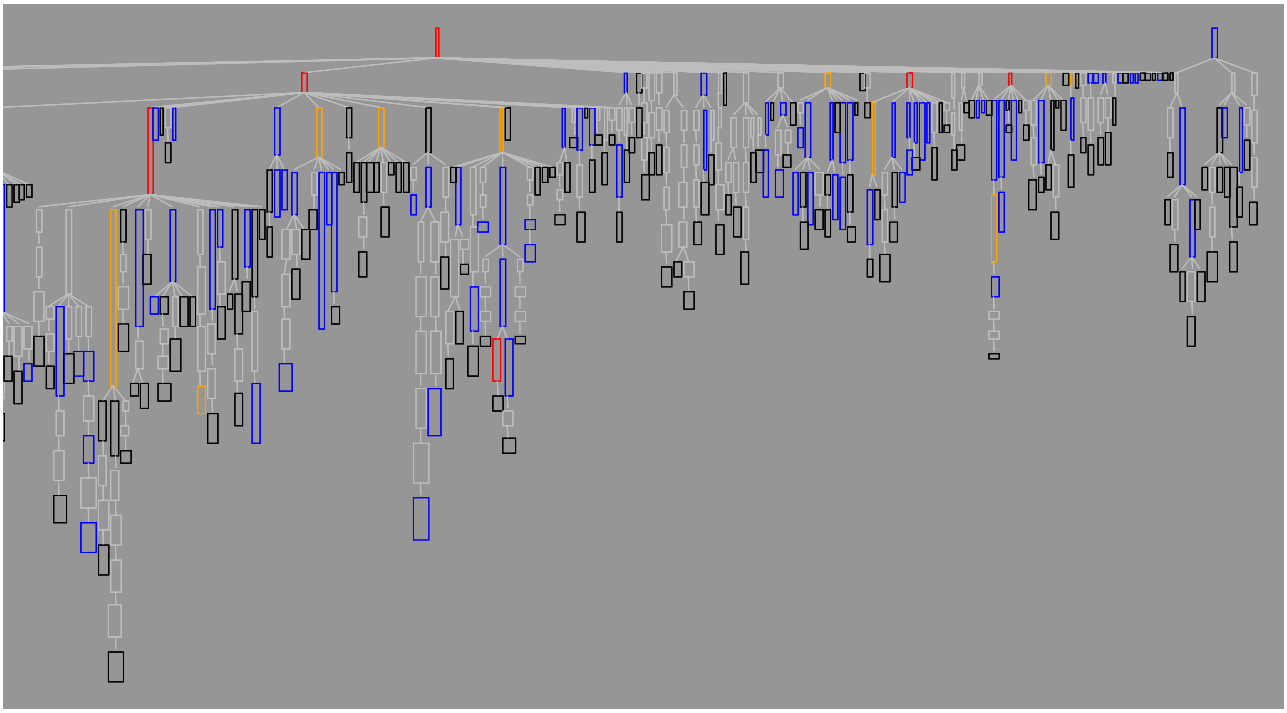


Abb. 6.9: Globalview der Unterbäume R05 bis R07 mit Belegungsgradkolorierung. Frameworks ohne Repräsentanten sind grau eingezeichnet.

Belegungsgradbereich	Farb-kodierung	Anzahl Knoten in R05-R07 Subtrees	Anzahl Knoten im Gesamtdatensatz
1		359	435
2		64	75
3 ... 10		93	96
11 ... 25		17	17
>25		10	11

Tabelle 6.3: Häufigkeitsverteilung der Framework Occupancy der Unterbäume R05 bis R07.



Alle Frameworks mit mehr als 25 Repräsentanten sind mit ihren MolCodes in folgender Tabelle 6.4 wiedergegeben.

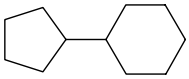
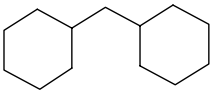
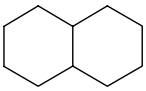
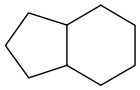
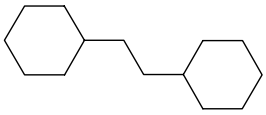
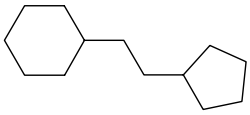
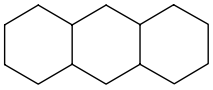
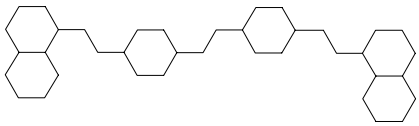
Rang	Molcode	Anzahl Vertreter	Depiction
1	R06	163	
2	R05L01R06	95	
3	R00	71	aliphatische Verbindungen
4	R06L02R06	64	
5	R06I02R06	53	
6	R05I02R06	44	
7	R05	41	
8	R06L03R06	37	
9	R05L03R06	34	
10	R06I02R06I02R06	29	
11	R06I02R06L03R06L03R06L03R06I02R06	27	

Tabelle 6.4: Auflistung der Frameworks und MolCodes mit mehr als 25 Repräsentanten.

Alle Frameworks (ohne 3 und 11) sind ebenfalls in den Spitzenplätzen (unter den ersten 12) in der von Bemis/Murcko beschriebenen Tabelle (siehe Abb. 4.1 auf Seite 37) zu finden, sogar die Abfolge stimmt im Wesentlichen überein. Der Datensatz enthält also einen Querschnitt an typischen drug-like Verbindungen. Das Framework an Position 10 ist aufgrund der beschriebenen Konstitutionsisomere häufig vertreten. Position 11 entspricht einer Serie von aktiven Verbindungen, die vermutlich im Rahmen einer Optimierung erzeugt worden sind.

6.5 Aktivitätsanalyse der Knoten

Als sogenannte Aktivitätshotspots werden Knoten bezeichnet, die eine Mindestanzahl an biologisch aktiven Verbindungen enthalten. Diese wird sinnvollerweise je nach Größe des Datensatzes gewählt. Die vereinfachte Häufigkeitsverteilung der Anzahl der Aktiven für die Teilbäume R05-R07 sieht wie folgt aus:

Anzahl/Bereich Aktiver	Anzahl Knoten
1	67
2	17
3 ... 10	26
11 ... 25	2
>25	4

Die Knoten in der schematischen Darstellung des TST für die Subtrees R05 bis R07 in Abb. 6.10 sind entsprechend der Aktivität der Verbindungen ihrer Subtrees farbig kodiert. Unterbäume, die ausschliesslich aktive Verbindungen enthalten, sind grün. Unterbäume, die ausschliesslich inaktive Verbindungen enthalten, sind rot. Gelbe Unterbäume enthalten sowohl Aktive als auch Inaktive. Blau hervorgehoben sind interessante Unterbäume, die den überwiegenden Teil der Aktiven enthalten. Im Detail sind sie in den folgenden Abb. 6.11 bis Abb. 6.14 dargestellt.

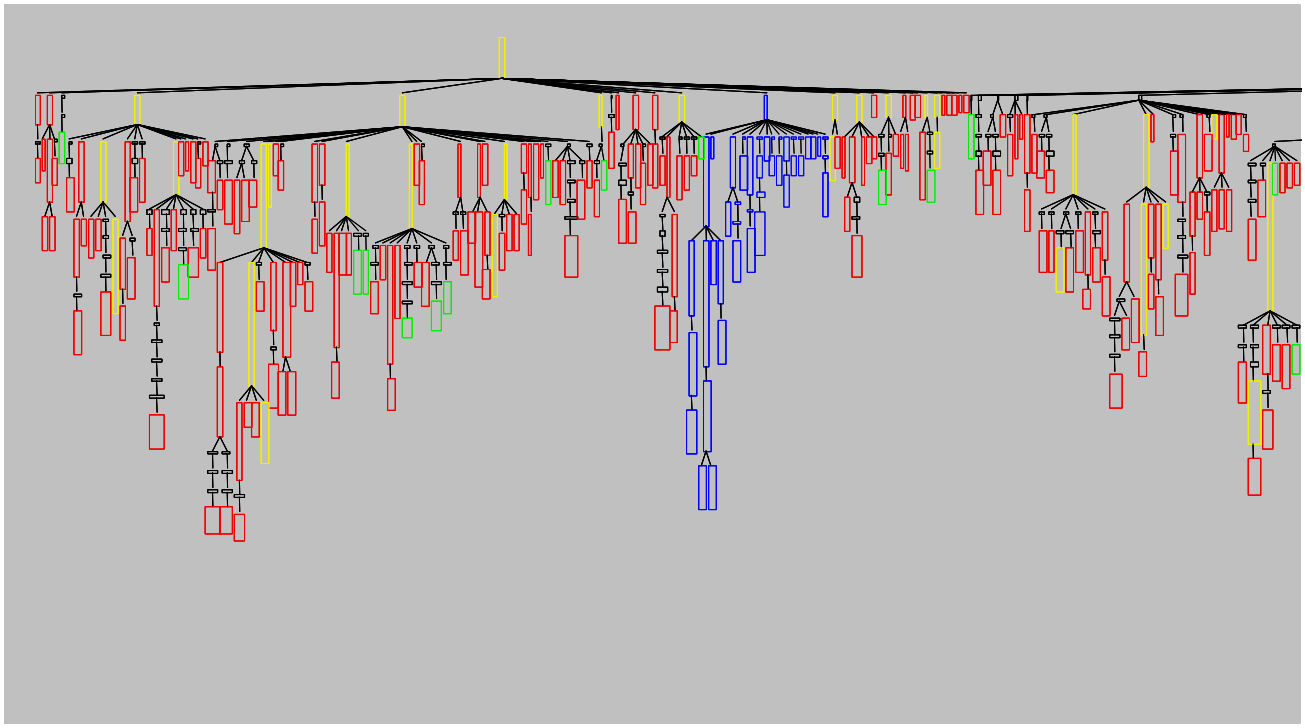
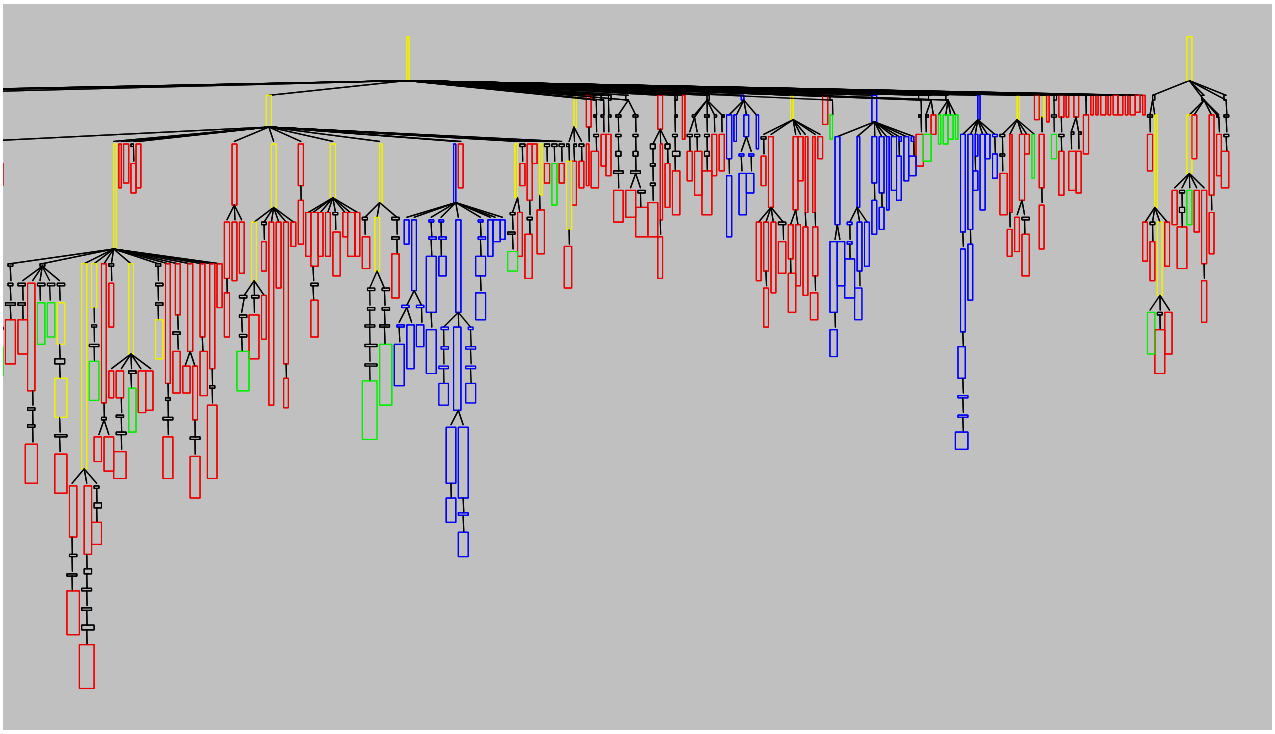


Abb. 6.10: Schematische Darstellung des TST für die Subtrees R05 bis R07.

Die Knoten sind entsprechend der Aktivität der im Unterbaum enthaltenen Verbindungen farbig gekennzeichnet. Unterbäume, die ausschliesslich aktive Verbindungen enthalten, sind grün. Unterbäume, die ausschliesslich inaktive Verbindungen enthalten, sind rot. Gelbe Unterbäume enthalten sowohl Aktive als auch Inaktive. Blau hervorgehoben sind interessante Unterbäume, die den überwiegenden Teil der Aktiven enthalten.



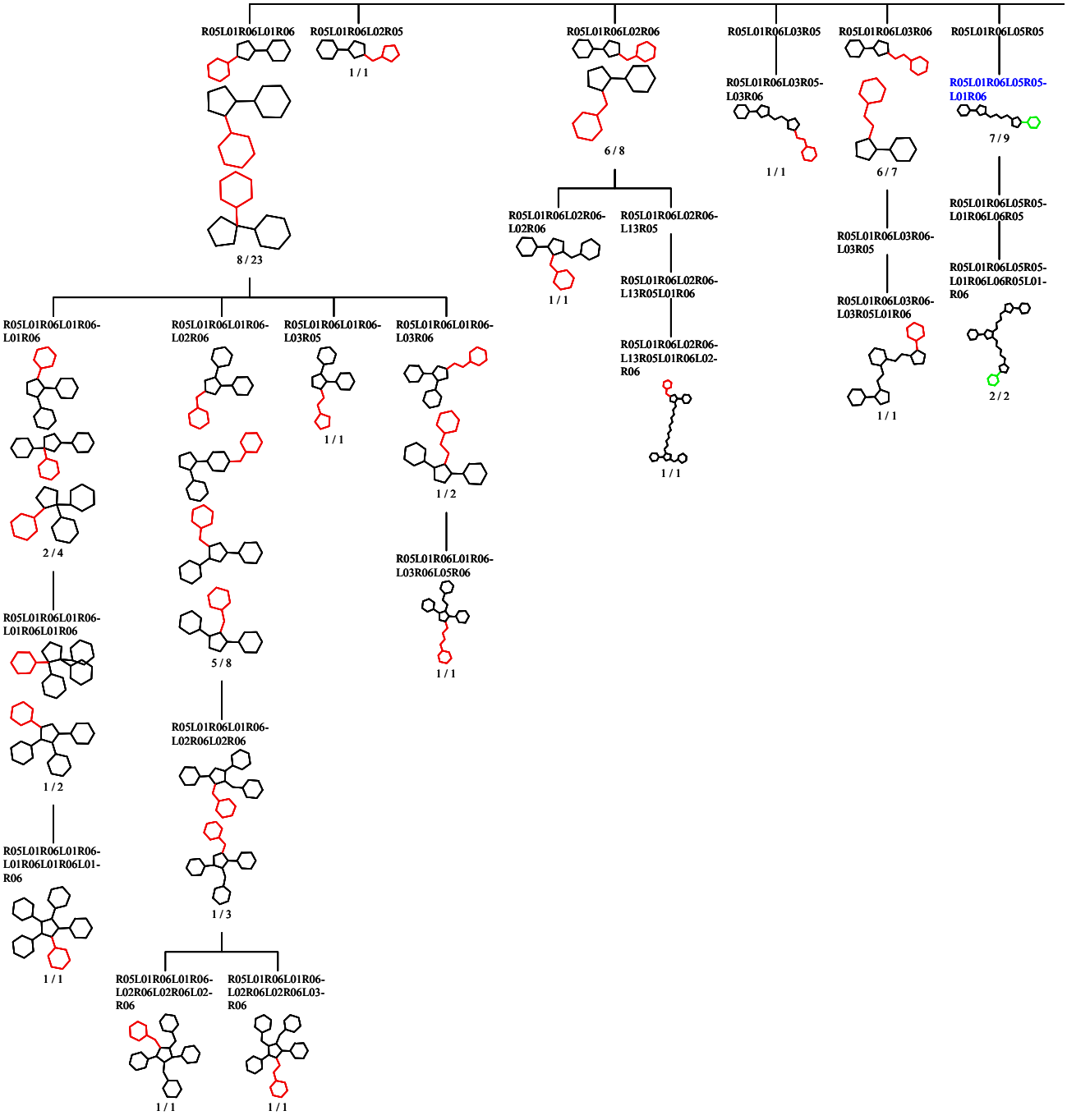
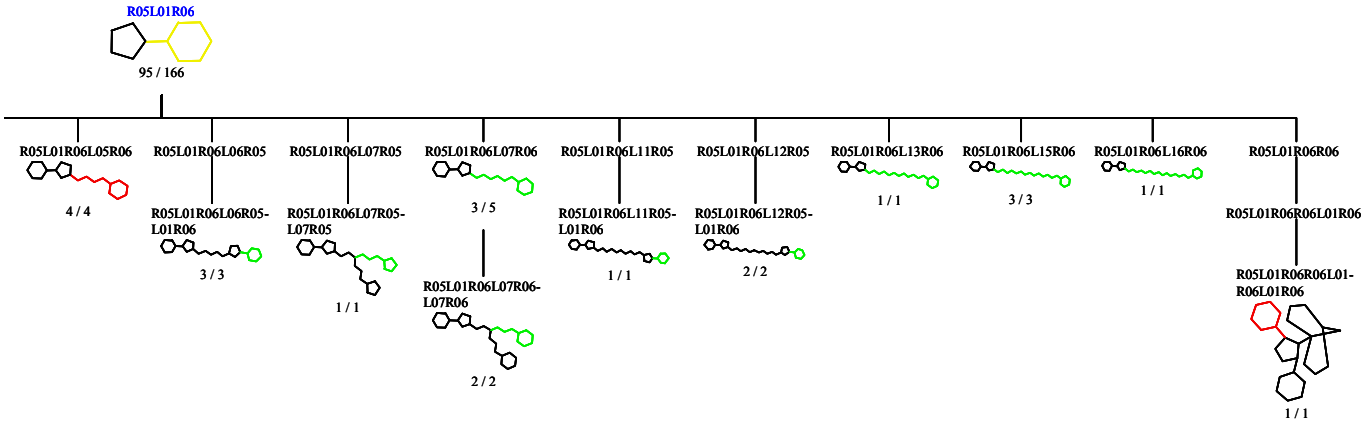
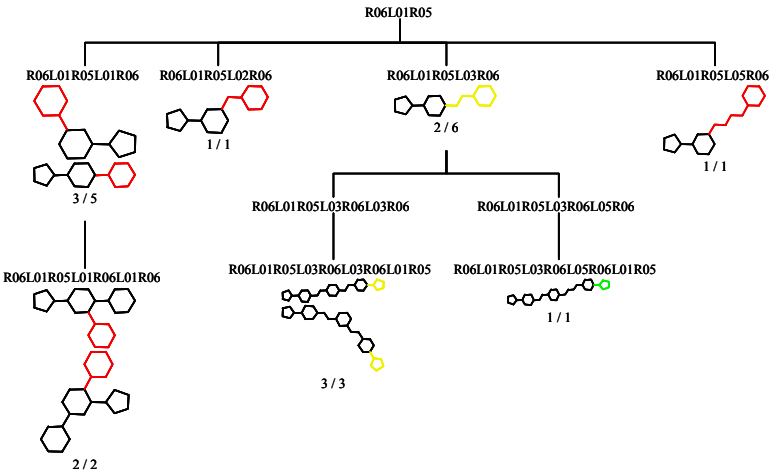


Abb. 6.11: Unterbaum R05L01R06 im Detail.





Die Frameworks der aktiven Unterbäume sind grün, die der inaktiven Unterbäume sind rot und die der Unterbäume mit Aktiven und Inaktiven sind gelb. Blau eingezeichnet sind die Frameworks aus der nachfolgenden Tabelle. Es ist zu erkennen, dass die Aktiven eine Framework-Erweiterung mit einer mindestens sechs Bindungen langen Kette besitzen, die zu einem langgestreckten Molekül mit zwei entfernten Ringsystemen führt. Erweiterungen, die zu einem „knödeligen“ Molekül mit hohem Raumbedarf in mehrere Richtungen führen, sind, wie durch ihre Rotfärbung erkennbar, inaktiv.
Der kleinere Ausschnitt rechts unten enthält den R06L01R05-Teilbaum. Der Prioritätswechsel der Ringe ist dadurch bedingt, dass die Gerüsterweiterung am R06-Ring stattfindet. Die vier in dem Teilbaum enthaltenen aktiven Verbindungen sind in den Knoten nach der dritten linearen Linker-Ring-Erweiterung zu finden.

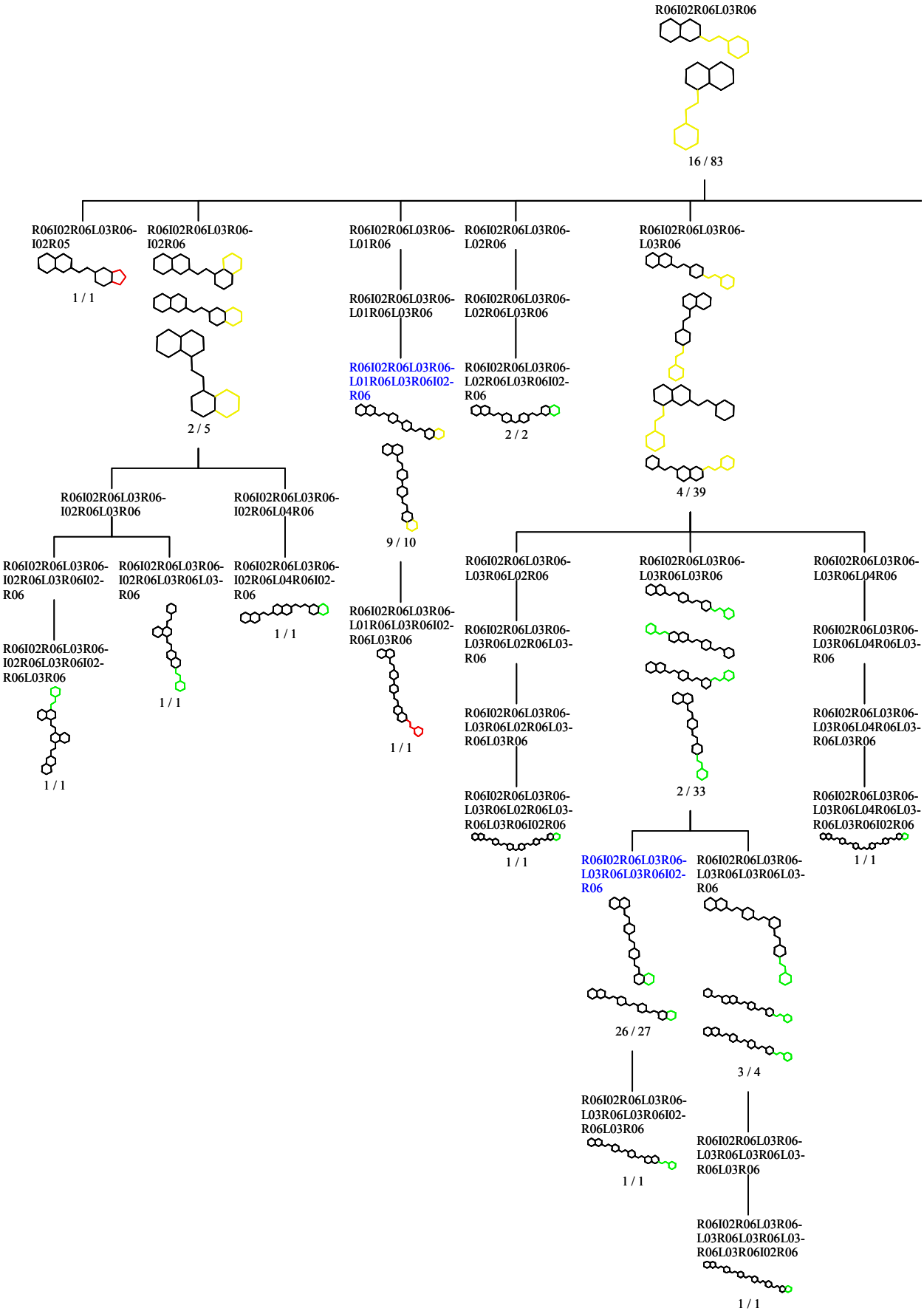
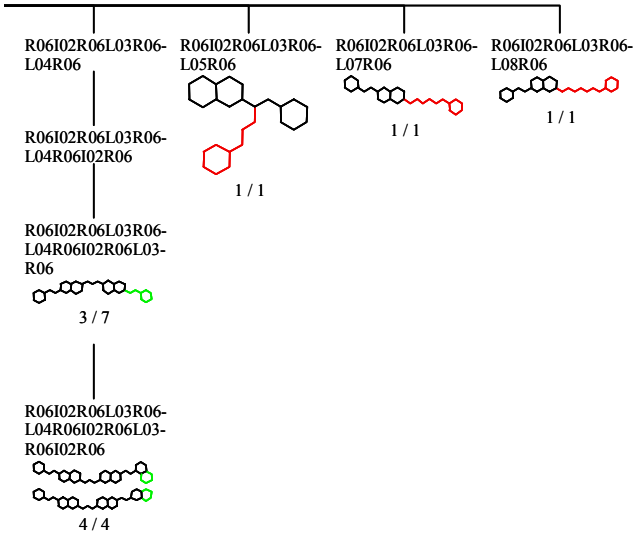


Abb. 6.12: Unterbaum R06I02R06L03R06 im Detail.



Unterhalb des R06I02R06L03R06-Knotens sind fast ausschließlich aktive Verbindungen zusammengefasst. Hier ist insbesondere das Framework n aus Abb. 6.15 enthalten, das bei der Analyse der Frameworks auf Rang 11 als häufig vertreten aufgefallen ist. Da die Größe der Frameworks der Aktiven zu denen in Abb. 6.11 ähnlich ist, kann auf einen entsprechenden Wirkmechanismus geschlossen werden. Die Farbkodierung entspricht der in der vorhergehenden Abbildung verwendeten.

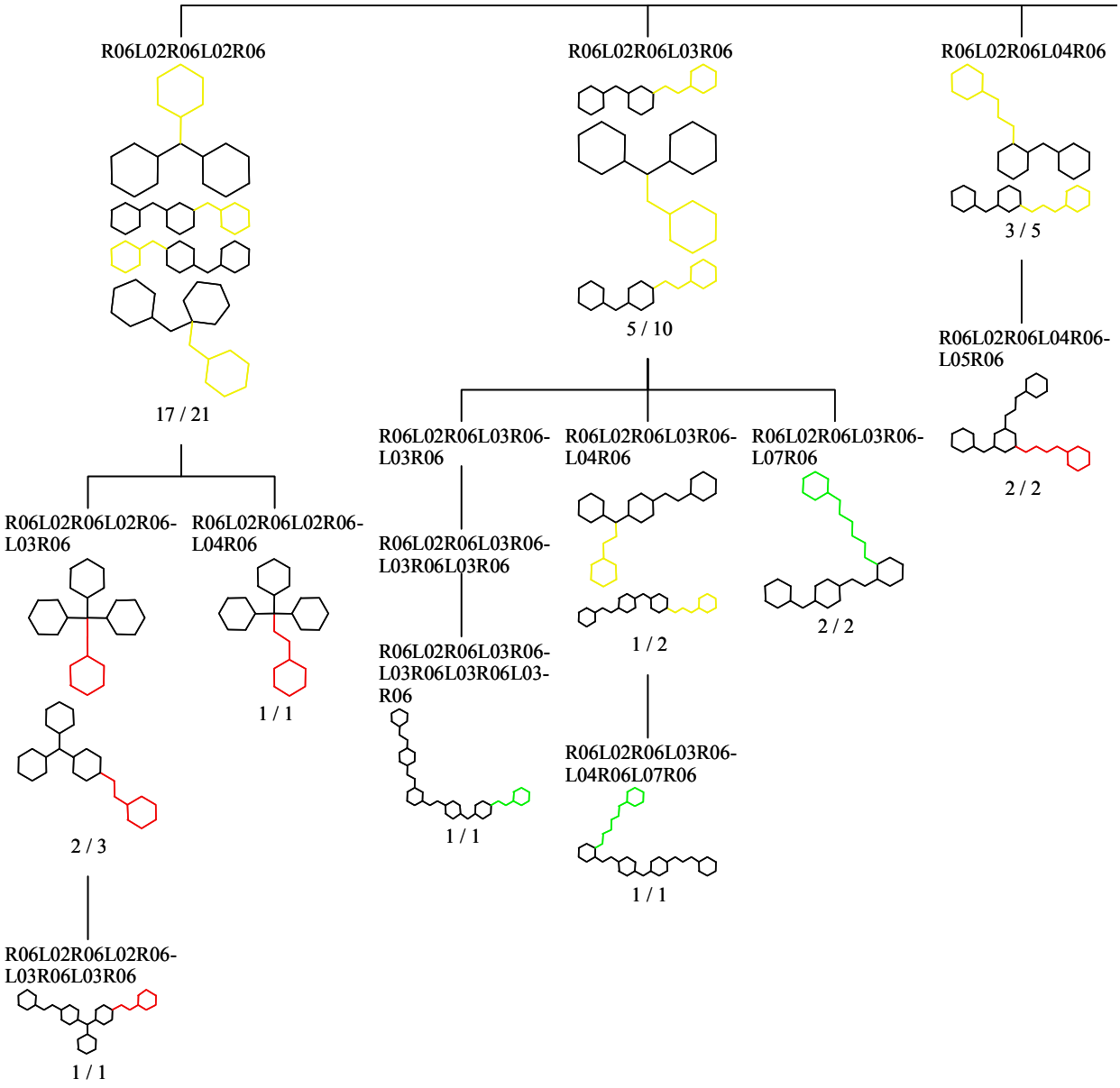
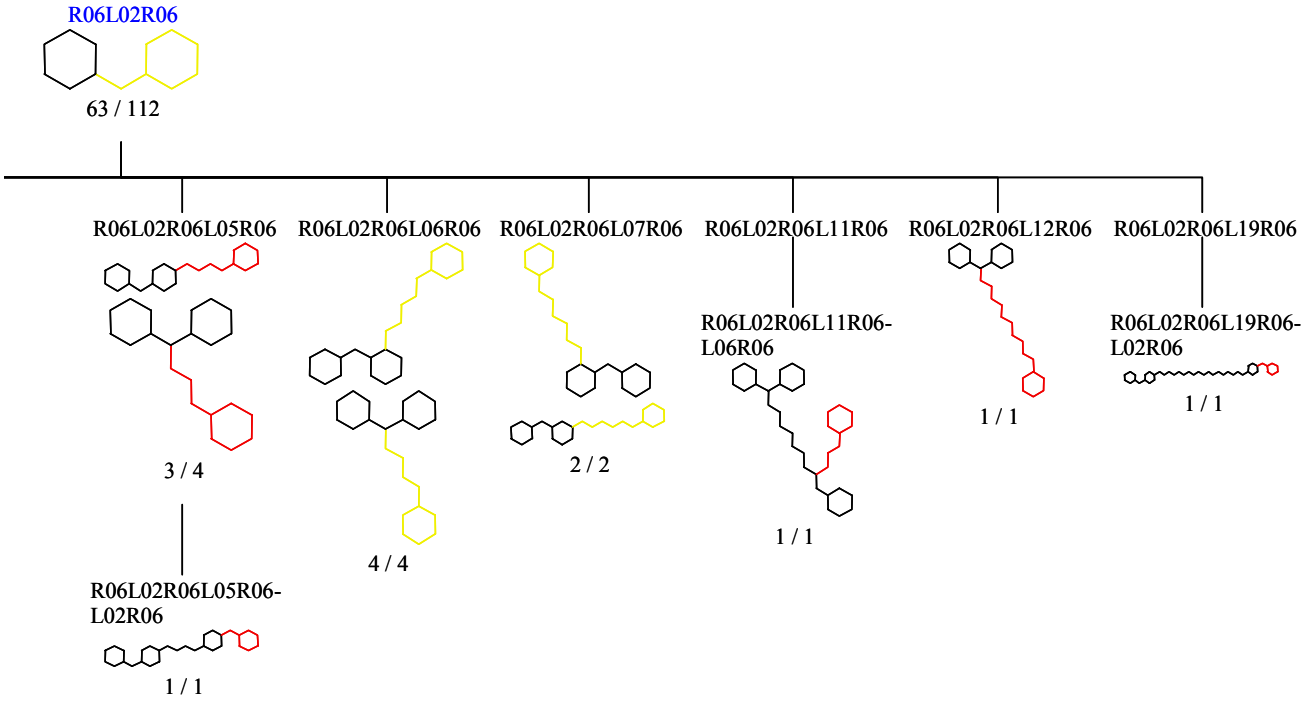


Abb. 6.13: Unterbaum R06L02R06 im Detail.



43% der Verbindungen mit dem kleinen R06L02R06-Grundgerüst sind Aktive. Dieser Größenwechsel deutet im Vergleich zu den Verbindungen aus Abb. 6.11 und Abb. 6.12 einen alternativen Wirkmechanismus an. Erweiterungen an dem Grundgerüst werden, wie durch die Farbcodierung angezeigt, in einigen Fällen toleriert (grün) und führen in anderen Fällen zu Inaktivität (rot). Die Farbkodierung ist die gleiche wie in Abb. 6.11.

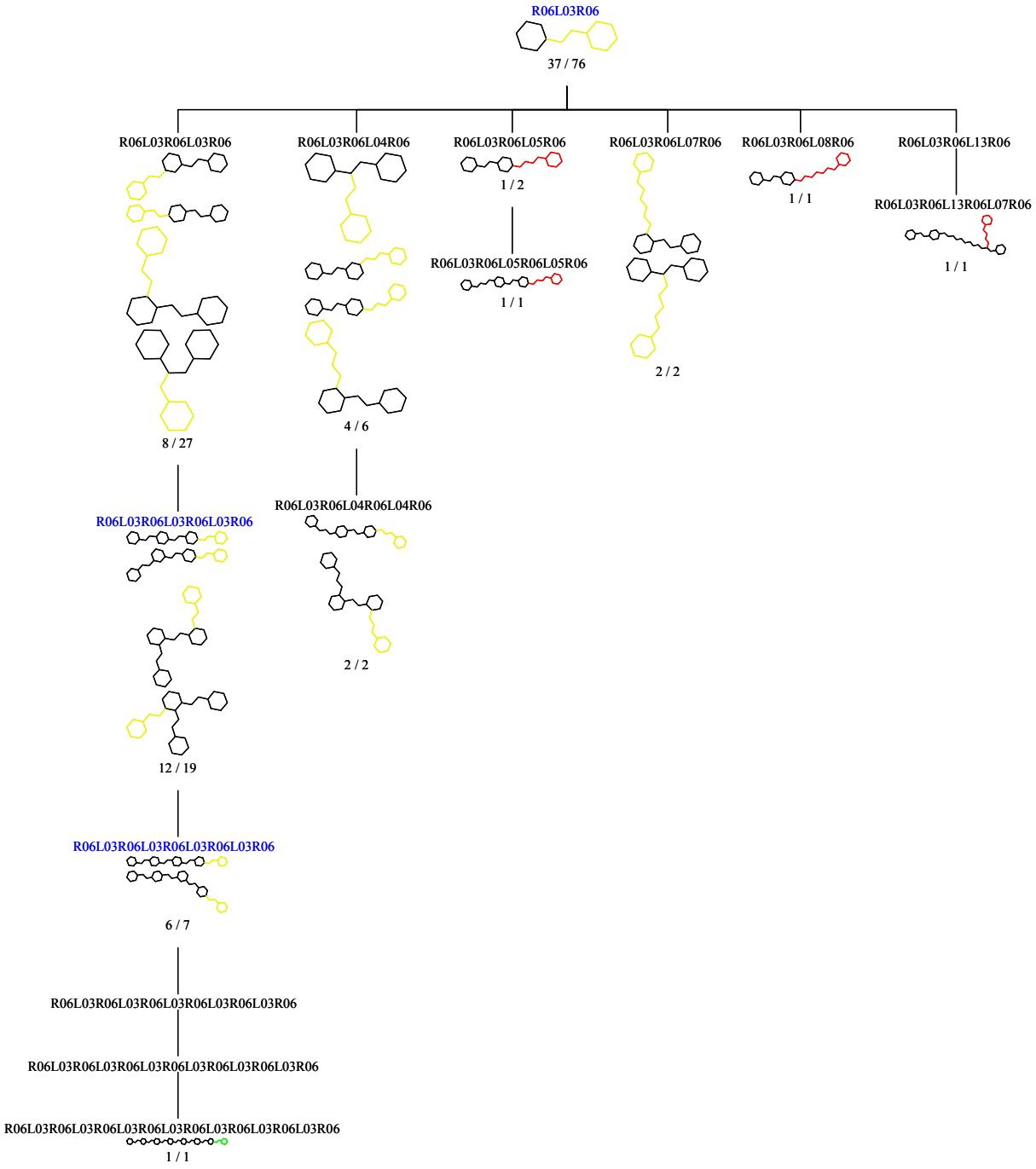


Abb. 6.14: Unterbaum R06L03R06 im Detail.

Die Abbildung mit der gleichen Farbkodierung enthält zwei der Knoten mit mehr als fünf Aktiven, die das R06L03R06-Framework als Basis besitzen. Das R06L03R06-Framework ist dem aus Abb. 6.13 bekannten R06L02R06-Framework ähnlich. Die großen, langgezogenen und linearen Frameworks entsprechen den in Abb. 6.11 und Abb. 6.12 blau hervorgehobenen. Eine entsprechende Zuordnung ist bei den jeweiligen Wirkmechanismen zu erwarten.

6.6 Detailanalyse der aktiven Template

17 Teilbäume enthalten mindestens 5 Aktive, die nach abnehmender Häufigkeit in folgender Tabelle aufgeführt sind. Die am häufigsten vertretenen Framework-Isomere dieser Knoten sind in Abb. 6.15 dargestellt. Das triviale Cyclohexangerüst für das Framework R06 ist nicht enthalten.

Anzahl Aktive	Gesamt-zahl Verbind.	Anteil Aktiver in Prozent	MolCode-Wurzel des Subtree
74	95	77,89	R05L01R06
27	63	42,86	R06L02R06
26	26	100,00	R06I02R06L03R06L03R06L03R06I02R06
25	34	73,53	R05L03R06
12	13	92,31	R06I02R06I02R06L03R05R05L03R05R05
11	12	91,67	R06L03R06L03R06L03R06
10	25	40,00	R06I02R06I02R06I02R06
10	159	6,29	R06
9	9	100,00	R06I02R06L03R06L01R06L03R06I02R06
9	37	24,32	R06L03R06
8	8	100,00	R06I02R06L03R05L03R05L04R05L03R05L03R06I02R06
8	10	80,0	R05I02R06I02R07
7	7	100,00	R05L01R06L05R05L01R06
6	10	60,00	R06I02R06I02R06I02R05L05R06L02R06
6	19	31,58	R05I02R06L01R06
5	6	83,33	R06L03R06L03R06L03R06L03R06
5	12	41,67	R05I02R05I02R06L01R06

Tabelle 6.5: Subtrees mit mindestens fünf Aktiven.

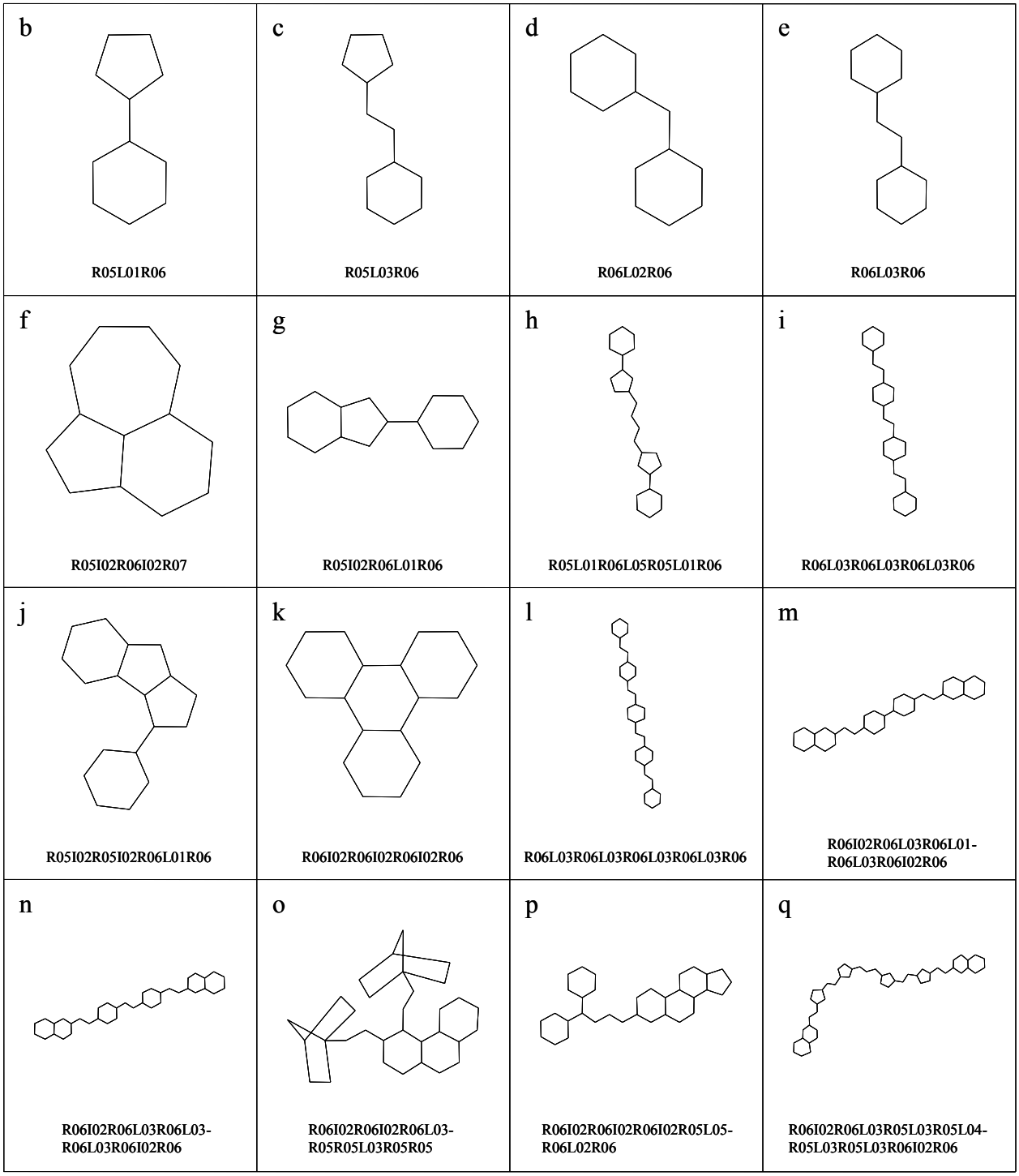


Abb. 6.15: Das am häufigsten vertretene Framework-Isomer der Knoten, die mehr als fünf aktive Repräsentanten enthalten. Nicht eingezeichnet ist das Cyclohexangerüst a des Frameworks R06.

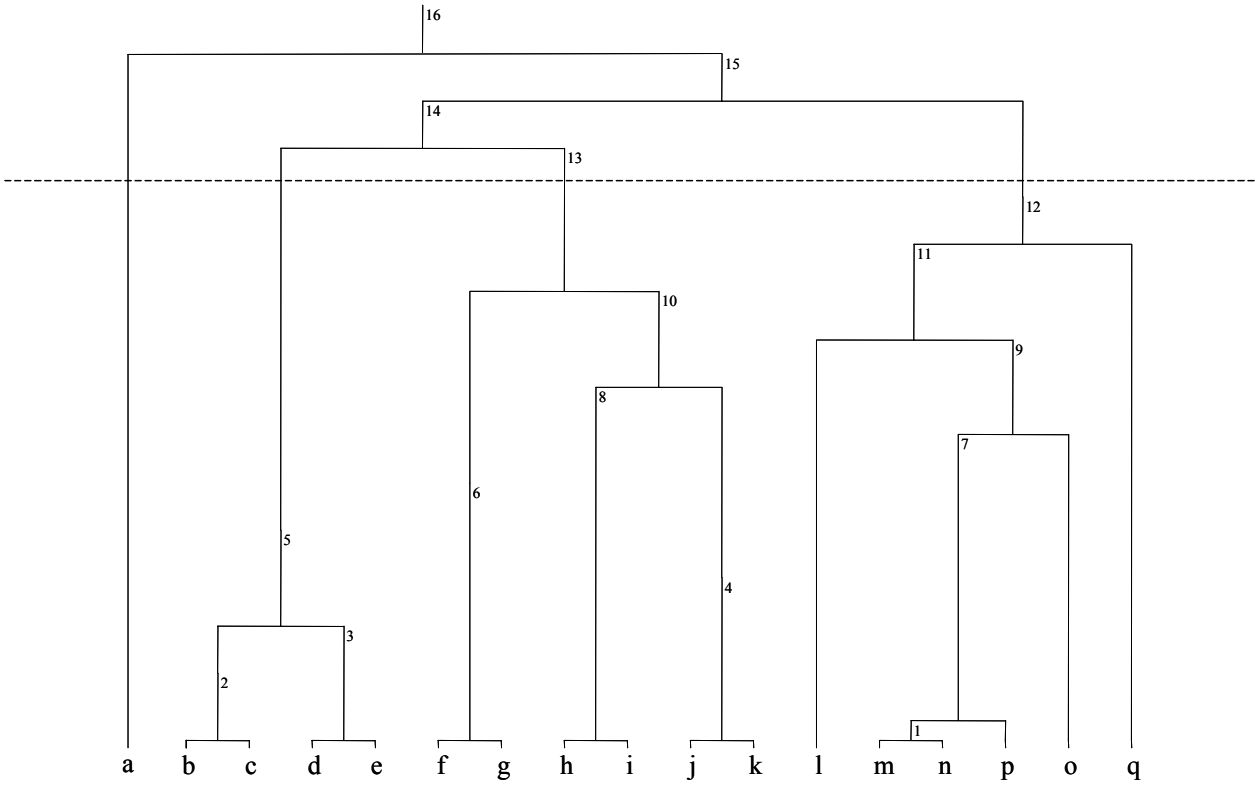
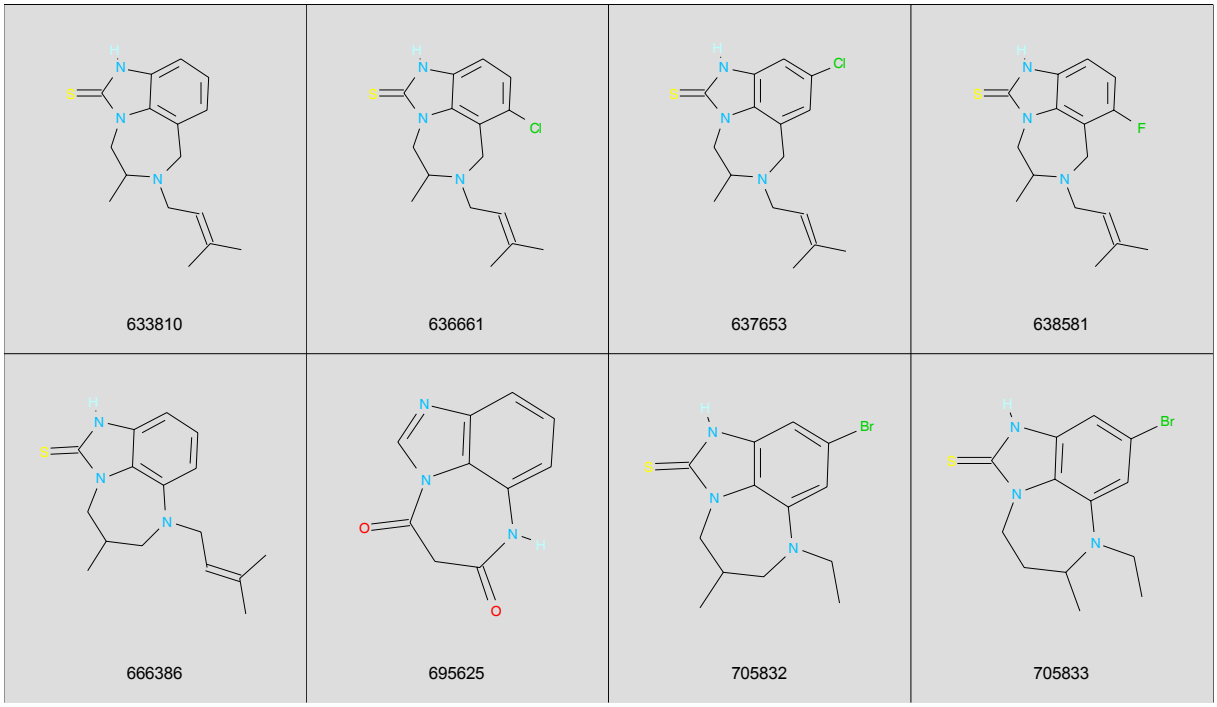


Abb. 6.16: Dendrogramm der hierachischen Clusterung (Average Linkage) unter Verwendung der Levenshtein-Distanz der MolCodes.

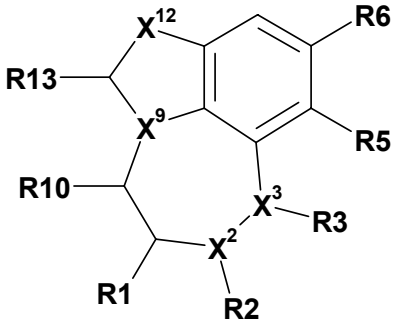
Das Dendrogramm der hierarchischen Clusterung der häufig vertretenen aktiven Template ist in Abb. 6.16 veranschaulicht. Die MolCodes wurden unter Verwendung der Levenshtein-Distanz und des Average Linkage-Verfahrens geclustert. Die horizontale Trennlinie zwischen dem Agglomerationsschritt 12 und 13 zeigt den ausgewählten Level an, der die Frameworks in drei Cluster und ein Singleton (a) teilt. Die drei Cluster bestehen aus den Frameworks b bis e, f bis k und l bis q.

Exemplarisch sind in Abb. 6.17 alle im Datensatz vorhandenen TIBO-Derivate (Tetrahydroimidazobenzodiazepinone) mit dem Framework f aufgeführt. TIBO-Derivate gehören zur Gruppe der Reverse-Transkriptase-Inhibitoren[239].

(a)



(b)



(c)

regid	X2	X3	X9	X12	R1	R2	R3	R5	R6	R10	R13
633810	N		N	N	Me	CH2CH=C(CH3)CH3					=S
636661	N		N	N	Me	CH2CH=C(CH3)CH3		Cl			=S
637653	N		N	N	Me	CH2CH=C(CH3)CH3			Cl		=S
638581	N		N	N	Me	CH2CH=C(CH3)CH3		F			=S
666386		N	N	N	Me		CH2CH=C(CH3)CH3				=S
695625		N	N	N		=O				=O	
705832		N	N	N	Me		Et		Br		=S
705833		N	N	N		Me	Et		Br		=S

Abb. 6.17: TIBO-Derivate (Tetrahydroimidazobenzodiazepinone) mit dem MolCode R05I02R06I02R07. Die Substituenten bzw. Heteroatomgruppen der Verbindungen in (a) sind in der XR-Tabelle (c) farbig hervorgehobenen. Die Zuordnungen der Substituenten R1 bis R13 und Heteroatome X2 bis X12 zu den Positionen in Framework sind in (b) veranschaulicht.

6.7 Festlegung der Anzahl der spektralen Momente zur Klassifizierung

Da das Klassifikationsverfahren alle vorhandenen Deskriptoren verwendet und keine Selektion vornimmt, wurde die optimale Anzahl der zu verwendenden spektralen Momente durch Testrechnungen festgelegt. Dazu wurden alle 1789 Verbindungen aus den R05/R06/R07-Subtrees zur Modellbildung verwendet und dann mit diesem Modell vorhergesagt. Das Resultat dieser Selbstklassifikation des gesamten Datensatzes mit 5 bis 16 verwendeten Deskriptoren ist in Abb. 6.19 wiedergegeben. Die Anzahl der korrekt klassifizierten Verbindungen liegt relativ konstant knapp über 85%. Der Anteil der korrekt klassifizierten Aktiven dagegen erreicht ein Maximum von 56,2% bei 12 Deskriptoren. Einen insgesamt größeren Einfluß hat die Anzahl der Deskriptoren bei der Klassifikation der Daten des R05I02R06-Unterbaums (siehe Abb. 6.19). Das Maximum der Vorhersagegenauigkeit liegt bei 87% und wird bei Verwendung von 13 Deskriptoren erreicht. Im folgenden werden daher stets 13 Deskriptoren (spektrale Momente bis zur Ordnung 12) verwendet.

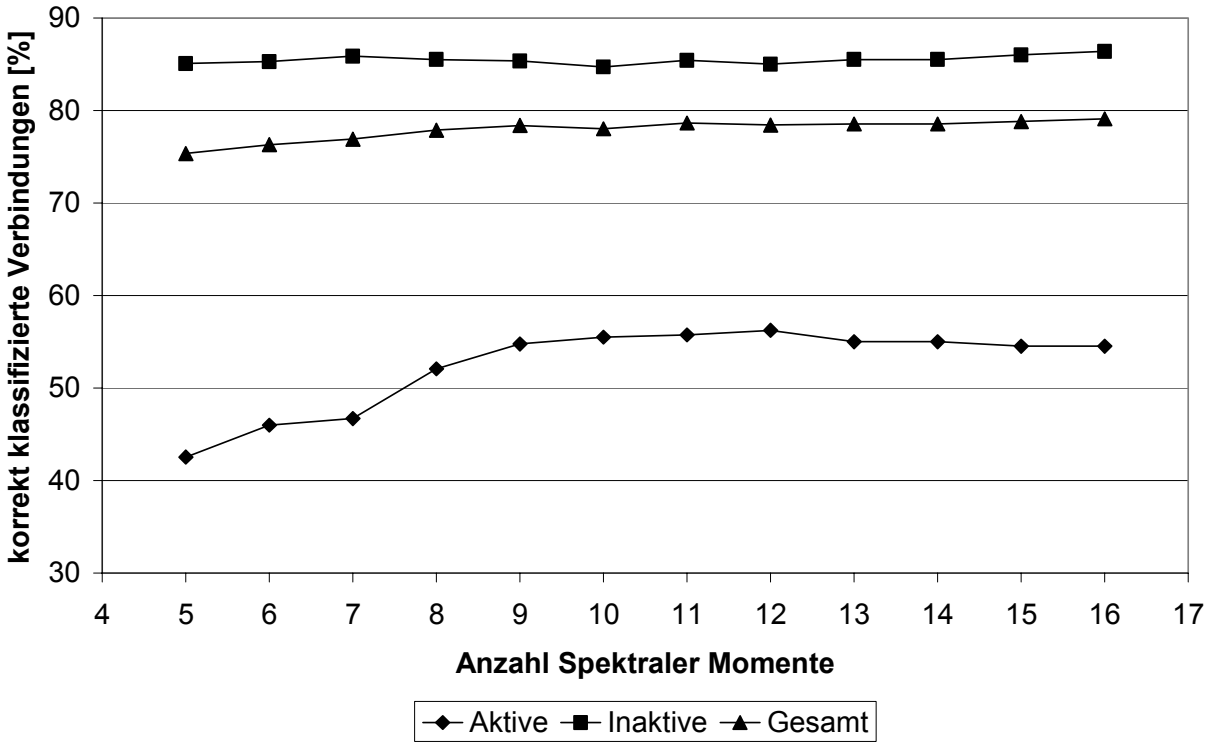


Abb. 6.18: Einfluss der Anzahl der verwendeten spektralen Momente auf den Anteil korrekt klassifizierter Verbindungen für den Datensatz bestehend aus den R05, R06 und R07-Unterbäumen (Aktive: 409, Inaktive: 1380).

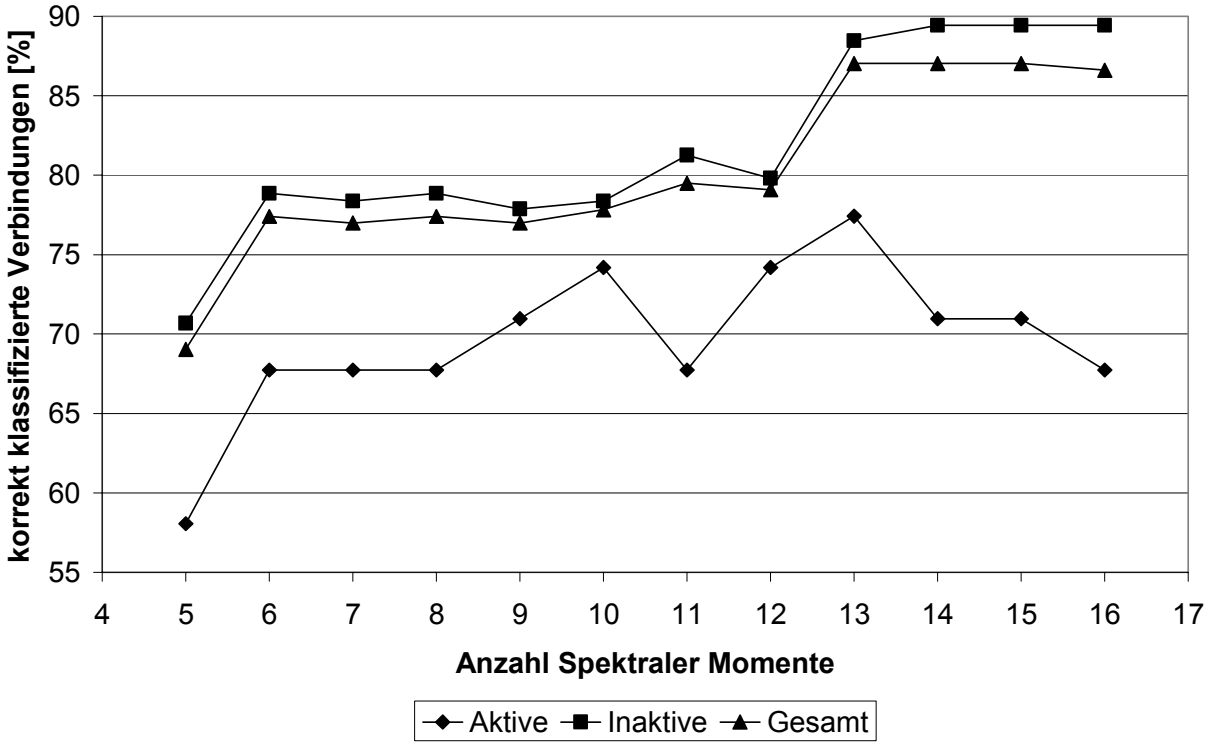


Abb. 6.19: Einfluss der Anzahl der verwendeten spektralen Momente auf den Anteil korrekt klassifizierter Verbindungen für den Datensatz bestehend aus dem R05I02R06-Unterbaum (Aktive: 31, Inaktive: 208).

6.8 Vergleich der Klassifikationsverfahren

Auf den Datensatz R05 bis R07 wurden die in folgender Tabelle beschriebenen Kombinationen von Klassifikationsverfahren und Deskriptoren verwendet. Zum Einsatz kamen zusätzlich zur Linearen Diskriminanzanalyse (LDA), die Logistic Regression (LOGREG)[240, 241] und das BinaryQSAR-Verfahren (BINQSAR)[242, 243] von Paul Labute.

Deskriptoren	Anzahl Deskriptoren	Verfahren	Vorhersagegenauigkeit [%]		
			über alle Daten	der Aktiven	der Inaktiven
Spektrale Momente μ_0 bis μ_{10}	11	LDA	77,47	53,44	85,58
		LOGREG	79,54	31,93	95,59
		BINQSAR	77,92	29,71	94,17
8 Hauptkomponenten (Varianz 100%) nach PCA von μ_0 - μ_{10}	8	LDA	76,69	49,67	85,80
		LOGREG	80,27	32,82	96,26
		BINQSAR	80,60	39,25	94,54
MACCS Fingerprint (Länge 166 Bit)	153[(a)]	LDA	89,32	79,38	92,68
		LOGREG[(b)]	90,83	74,94	96,19
		BINQSAR	86,14	73,61	90,36

80 Hauptkomponenten (Varianz 92.3 % [c]) nach PCA des MACCS Fingerprint	80	LDA	86,25	77,61	89,16
		LOGREG	88,93	69,40	95,52
		BINQSAR	89,6	66,7	97,3

[a] Bits, die für alle Verbindungen konstant sind (einheitlich 0 oder 1), wurden nicht berücksichtigt.

[b] Die Modellerstellung benötigt auf einem Pentium 4@1,8MHz ca. 2,5 h. Alle anderen Modelle sind in einigen Sekunden oder wenigen Minuten erstellbar.

[c] Aufgrund der Limitierung der maximalen Feldanzahl (auf 256) im MOE Database Viewer wurden keine weiteren Komponenten berechnet.

Die Vorhersagegenauigkeit der Aktiven ist aus statistischen Gründen generell geringer als die der Inaktiven. Die LDA hat immer den größten Wert der drei verglichenen Verfahren. Am schlechtesten schneidet das BINQSAR Verfahren bei der Vorhersage der Aktiven ab. Eine Dekorrelation durch PCA der spektralen Momente hat keinen positiven Effekt und ist daher nicht erforderlich. Erstaunlich gut schneiden die MACCS-Fingerprints ab; auch bei diesen Deskriptoren bringt eine Verwendung der ersten 80 Hauptkomponenten keinen Gewinn für der Vorhersagegenauigkeit.

6.9 Details zur LDA-Klassifikation

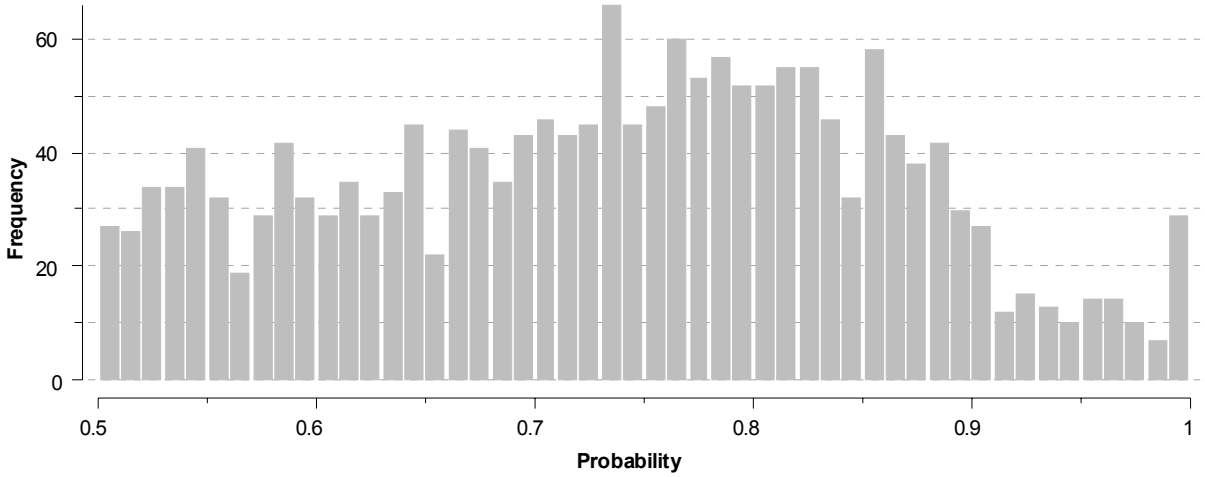
Die Detailinformationen der Selbstklassifikation der Unterbäume R05-R07 – unter Verwendung von 13 spektralen Momenten als Deskriptoren – mittels LDA sind in Abb. 6.20 veranschaulicht.

Das erzeugte Modell hat eine Gesamtvorhersagegenauigkeit von 78,5%. Von den 409 Aktiven werden allerdings nur 225 (55%) als solche erkannt. Das Histogramm der Klassifikationswahrscheinlichkeiten a) enthält alle Werte zwischen 0,5 und 1,0, mit einer geringfügigen Anhäufung zwischen 0,7 und 0,9. Der Bereich zwischen 0,9 und 1,0 enthält 141 (7,9%) Verbindungen, diese werden mit einer sehr hohen Wahrscheinlichkeit vom Modell vorhergesagt. Die Säulen im Bereich kleiner Werte zwischen 0,50 bis 0,55 enthalten die Werte der 172 (9,6%) Verbindungen, die mit einer sehr geringen Wahrscheinlichkeit einer der beiden Gruppen zugeordnet wurden. Diese sollten als „nicht-klassifizierbar“ betrachtet werden.

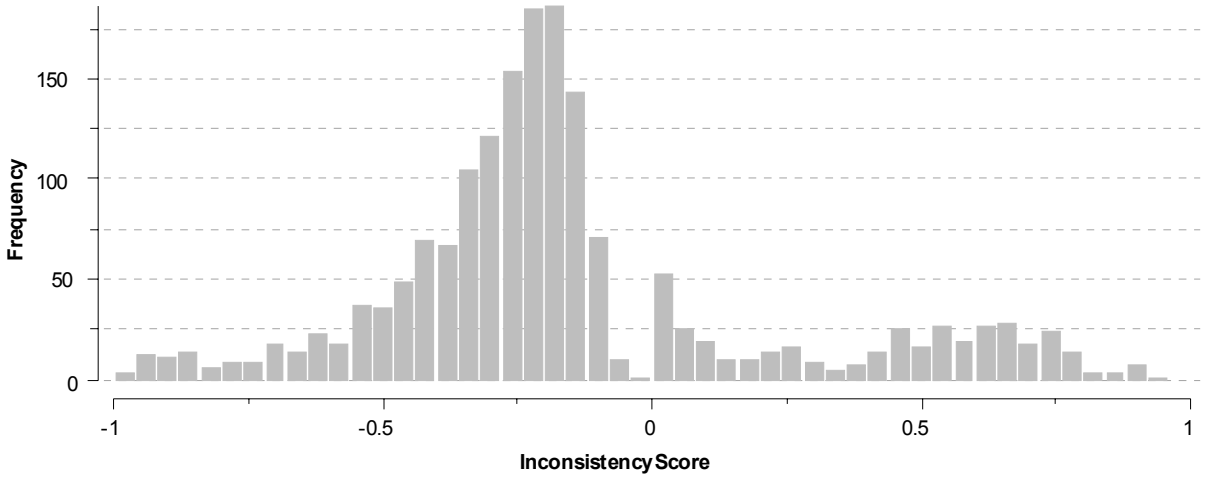
Die im Histogramm der Inconsistency Scores b) häufig vertretenen Werte zwischen –0,1 und –0,4 enthalten die Verbindungen, die mit einer geringen Wahrscheinlichkeit als falsch negativ klassifiziert wurden. Das häufigere Auftreten von negativen Inconsistency Score-Werten (siehe Abschnitt 7.3.1) ist darauf zurückzuführen, dass zur Modellbildung mehr inaktive als aktive Verbindungen im Datensatz zur Verfügung standen und die Vorhersage daher in diese Richtung verzerrt ist. Diese Verzerrung ist unerwünscht, da es problematischer ist, unerkannt gebliebene aktive Verbindungen (falsch negative) zu früh auszusortieren, als falsch postive in einem späteren Stadium als solche zu identifizieren.

Jeweils acht der im Datensatz enthaltenen Verbindungen, denen ein minimaler bzw. maximaler Inconsistency Score zugeordnet ist, sind in den Abb. 6.21 bzw. Abb. 6.22 aufgeführt.

a)



b)



c)

C O N F U S I O N M A T R I X
actual group=col
assigned group=row

	A C T U A L				
	active	inactive	noclass		
active	225	200	0	425	(55.01% correct classified)
inactive	184	1180	0	1364	(85.51% correct classified)
noclass	0	0	0	0	(0.00% correct classified)
	409	1380	0	1789	(78.54% correct classified)

Abb. 6.20: Detailinformationen zur Selbstklassifikation der Unterbäume R05-R07. Histogramme der Vorhersagewahrscheinlichkeiten (a) und der Inconsistency Scores (b) und Confusion Matrix (c).

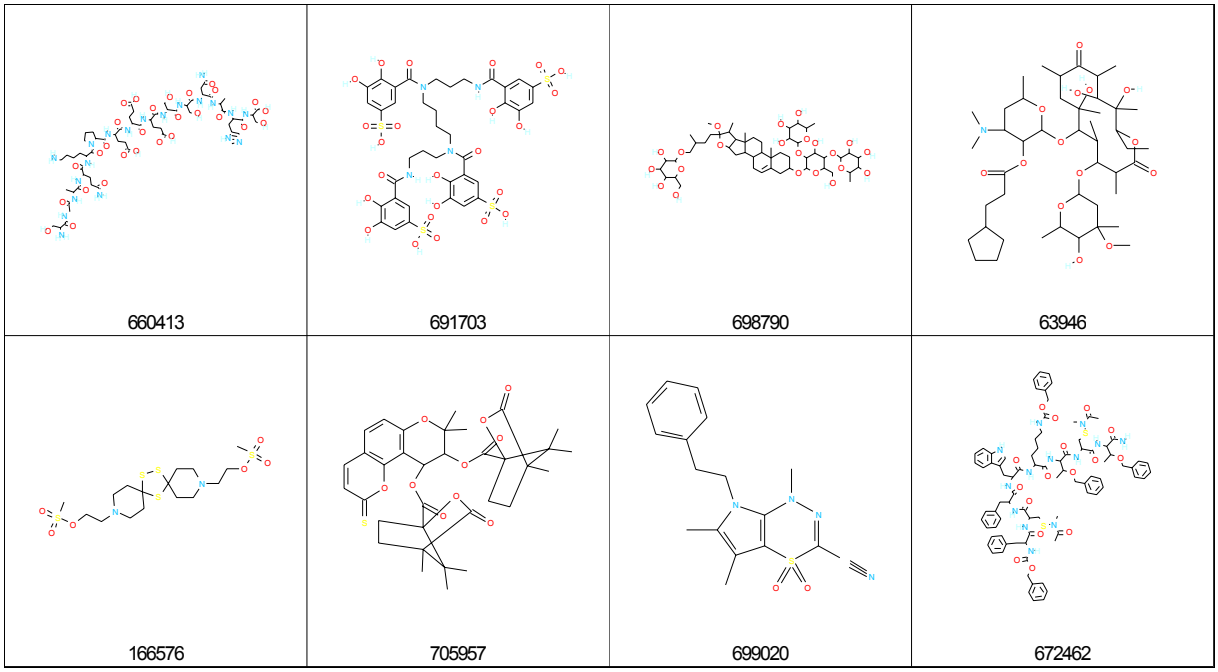


Abb. 6.21: Die acht Verbindungen mit minimalem Inconsistency Score (-0,98 bis –0,94). Diese sind falsch negativ klassifizierte Verbindungen.

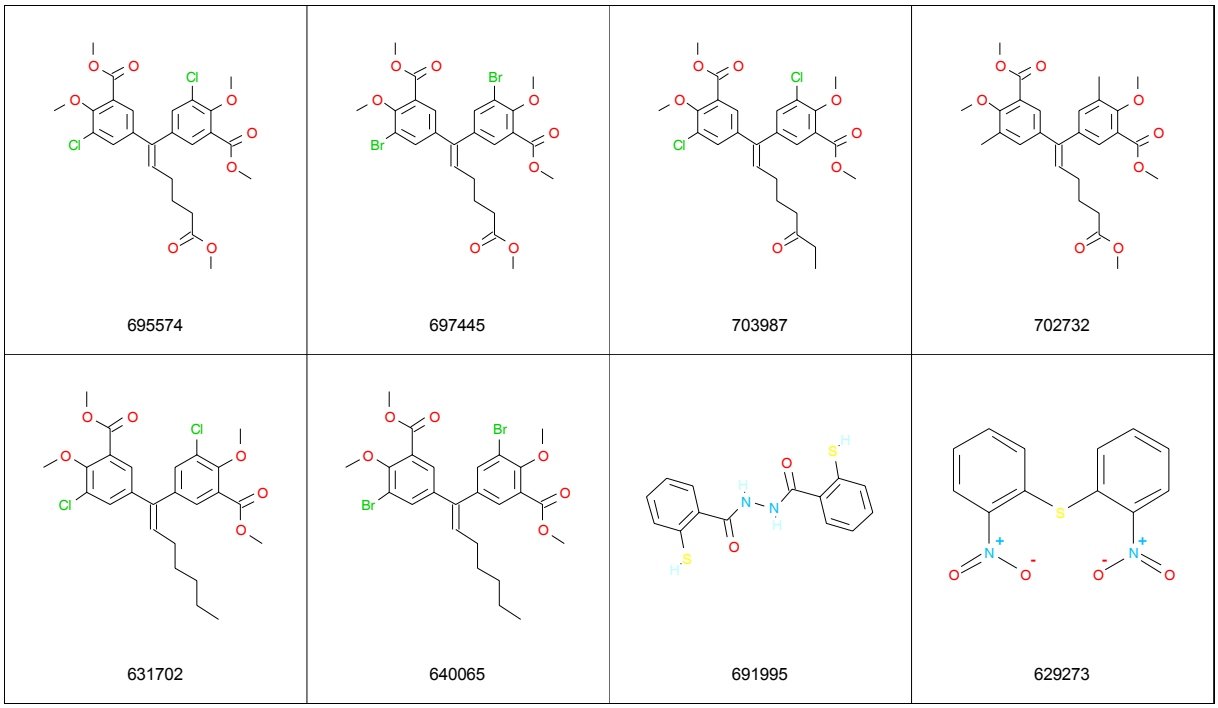


Abb. 6.22: Die acht Verbindungen mit maximalem Inconsistency Score (0,94 bis 0,89). Diese sind falsch positiv klassifizierte Verbindungen.

6.10 Verwendung lokaler Klassifikationsmodelle

Mittels geeigneter Deskriptoren und Klassifikationsverfahren können – wie oben gezeigt – entsprechende Modelle zu einem Datensatz für die mathematische Aktivitätsvalidierung und zur Vorhersage erstellt werden. Das Identifizieren der Aktiven ist in der Regel deutlich fehlerbehafteter als die Vorhersage der Inaktiven. Dies ist vor allem darauf zurückzuführen, daß erstere im Datensatz deutlich unterrepräsentiert sind. Zusätzlich sind Modelle, die alle zur Verfügung stehenden Daten verwenden, aufgrund der hohen Diversität im Datenpool zwar in der Lage, auch sehr „exotische“ Verbindungen zu klassifizieren, aber sie enthalten bei der Prädiktion von interessierenden Verbindungsgruppen einen hohen Anteil an Rauschen.

Eine Möglichkeit zur Beseitigung dieser Limitierungen bietet die kombinierte Verwendung von unterschiedlichen Verfahren zur Vorgruppierung der Daten und zur anschliessenden Klassifikationsmodellerstellung aus den erzeugten kleineren Gruppen[244, 245]. Diese enthalten nur Verbindungen mit einem hohen Grad an struktureller Homogenität, die zusätzlich vermuten läßt, dass ihre Wirkung auf dem gleichen oder zumindest einem ähnlichen biologischen Mechanismus beruht. Auch die Ableitung einer SAR zu den einzelnen Gruppen ist im Vergleich zum Gesamtdatensatz vereinfacht und intuitiv verständlich.

In BayTree werden lokale LDA-Modelle auf der Basis von MolCodes zu Subtrees generiert. Geeignete Gruppen für solche lokalen Modelle können manuell vorgegeben werden oder automatisch durch Festlegung von drei Parametern identifiziert werden. Diese sind, jeweils bezogen auf das lokale Modell:

1. die Mindestanzahl an Verbindungen
2. der Mindestanteil aktiver Verbindungen in Prozent
3. die minimale Vorhersagegenauigkeit in Prozent.

Im Unterschied zur Erhöhung der Anzahl von verwendeten Deskriptoren hat die Verwendung von mehreren Modellen nicht automatisch eine verbesserte Vorhersagegenauigkeit zur Folge, außer es handelt sich tatsächlich um bessere lokale Modelle.

In Abb. 6.23 sind zu dem aus den R05-R07-Unterbäumen bestehenden Datensatz verschiedene Multimodell-Klassifikationen durchgeführt worden. Dazu wurde der Datensatz unter Beibehaltung des Aktiv/Inaktiv-Verhältnisses nach dem Zufallsprinzip in mehrere gleiche Teile aufgeteilt, die dann als Basis für die Erzeugung einer entsprechenden Anzahl von lokalen Modellen dienten. In keinem Fall stieg die Vorhersagegenauigkeit deutlich an; sie blieb im Vergleich zum Einzelmodell praktisch unverändert.

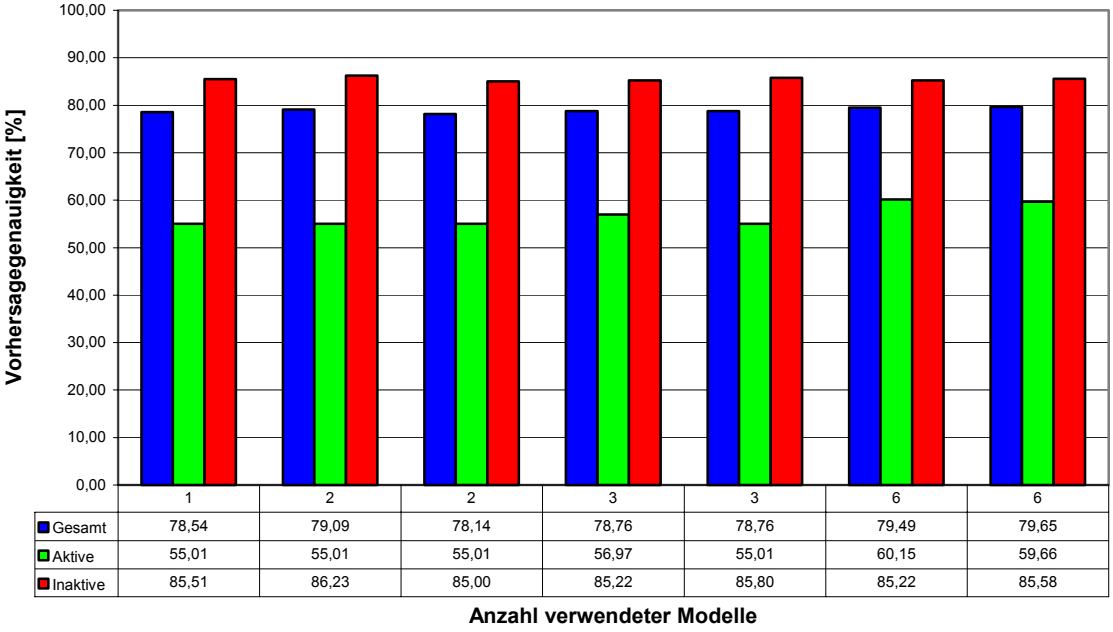


Abb. 6.23: Vorhersagegenauigkeit von zufällig erzeugten Multimodell-Klassifikationen im Vergleich zur Einmodell-Klassifikation.

In Tabelle 6.6 sind alle 33 möglichen Untermodelle aufgelistet. Die restlichen 829 Unterbäume kamen als Untermodell nicht in Betracht, da kein lokales Modell erstellt werden konnte.

MolCode Subtreeroot	Anzahl Verbindungen			Anteil in Prozent		Vorhersage genauigkeit	Enthalten in Modell
	Gesamt	Aktive	Inakt.	Aktive	Inakt.		
R05	666	171	495	25,68	74,32	79,88	B, C, F, G
R05I02R05	58	6	52	10,34	89,66	91,38	G
R05I02R05I02R06	18	5	13	27,78	72,22	61,11	G
R05I02R05I02R06L01R06	12	5	7	41,67	58,33	58,33	---
R05I02R06	239	31	208	12,97	87,03	83,68	G
R05I02R06L01R05	21	5	16	23,81	76,19	57,14	---
R05I02R06L01R06	33	9	24	27,27	72,73	69,70	F, G
R05I02R06L03R06	13	4	9	30,77	69,23	69,23	E, G
R05L01R06	166	100	66	60,24	39,76	95,78	D, E, F, G
R05L03R06	37	26	11	70,27	29,73	86,49	D, E, F, G
R06	1082	230	852	21,26	78,74	85,40	C, F, G
R06I02R05	78	4	74	5,13	94,87	76,92	G
R06I02R06	436	127	309	29,13	70,87	88,30	F, G
R06I02R06I02R05	24	5	19	20,83	79,17	87,50	F, G
R06I02R06I02R05I02R06	17	4	13	23,53	76,47	64,71	G
R06I02R06I02R06	136	40	96	29,41	70,59	88,24	F, G
R06I02R06I02R06I02R05	27	15	12	55,56	44,44	81,48	E, F, G
R06I02R06I02R06I02R05-L05R06	17	10	7	58,82	41,18	41,18	---
R06I02R06I02R06I02R05-L05R06L02R06L03R06	7	4	3	57,14	42,86	42,86	---
R06I02R06I02R06I02R06	30	10	20	33,33	66,67	73,33	D, E, F, G

R06I02R06L01R06	48	3	45	6,25	93,75	95,83	G
R06I02R06L03R05	19	14	5	73,68	26,32	26,32	---
R06I02R06L03R06	83	59	24	71,08	28,92	89,16	D, E, F, G
R06I02R06L03R06L03R06	39	35	4	89,74	10,26	89,74	D, E, F, G
R06L01R05	13	4	9	30,77	69,23	69,23	E, G
R06L02R06	112	39	73	34,82	65,18	77,68	D, E, F, G
R06L02R06L03R06	10	5	5	50,00	50,00	50,00	---
R06L03R06	76	29	47	38,16	61,84	65,79	D, E, F, G
R06L03R06L03R06	27	18	9	66,67	33,33	74,07	E, F, G
R07	41	8	33	19,51	80,49	65,85	C, G
R07I02R05	21	6	15	28,57	71,43	80,95	F, G
R07I02R05I02R06	20	6	14	30,00	70,00	75,00	E, F, G
R07I02R05I02R06L01R06	10	5	5	50,00	50,00	50,00	---

Tabelle 6.6: Auflistung der 33 möglichen Untermodelle für den R05-R07-Teildatensatz.

Die Zuordnung der lokalen Modelle zu den im folgenden verwendeten Multi-Modellen B bis G ist aus der letzten Spalte ersichtlich.

In Modell A wurden die Subtrees R05-R07 zur Erstellung eines einzigen Klassfikationsmodells verwendet. Modell B enthält ein lokales Modell für den R05 Subtree. In Modell C wurden für die drei Subtrees R05, R06 und R07 drei separate Untermodelle erzeugt. Die Kriterien zur Auswahl der Untermodelle für die Multimodelle D-G können folgender Tabelle entnommen werden.

Multi-Modell	Mindestanzahl Verbindungen	Mindestanteil Aktiver	Mindest-vorhersage genauigkeit	Selektierte Untermodelle	Anzahl der Untermodelle
A				0	0
B				1	1
C				3	3
D	30	30%	60%		7
E		30%	60%		12
F	20	20%	60%		17
G			60%		26

Zum Zwecke der Validierung wurden alle Multimodelle nach ihrer Erstellung zur Selbstklassifikation des Gesamtdatensatzes verwendet. Die Gesamtvorhersagegenauigkeit ist in Abb. 6.25 veranschaulicht. Die Vorhersagegenauigkeit für die Gruppe der Aktiven und der Inaktiven ist gesondert aufgeführt. Es wird deutlich, dass besonders die Vorhersagegenauigkeit der Aktiven mit einer erhöhten Anzahl an Untermodellen zunimmt, ohne dass die der Inaktiven sinkt. Bei der klassischen Verwendung nur eines einzigen Modells wie bei A liegt die Vorhersagegenauigkeit nur bei 55,01%, während sie schon bei der Verwendung von zwei oder drei Untermodellen um über 12% ansteigt. Werden zusätzliche Untermodelle einbezogen, können wie bei Modell F und G bis über 80% erreicht werden.

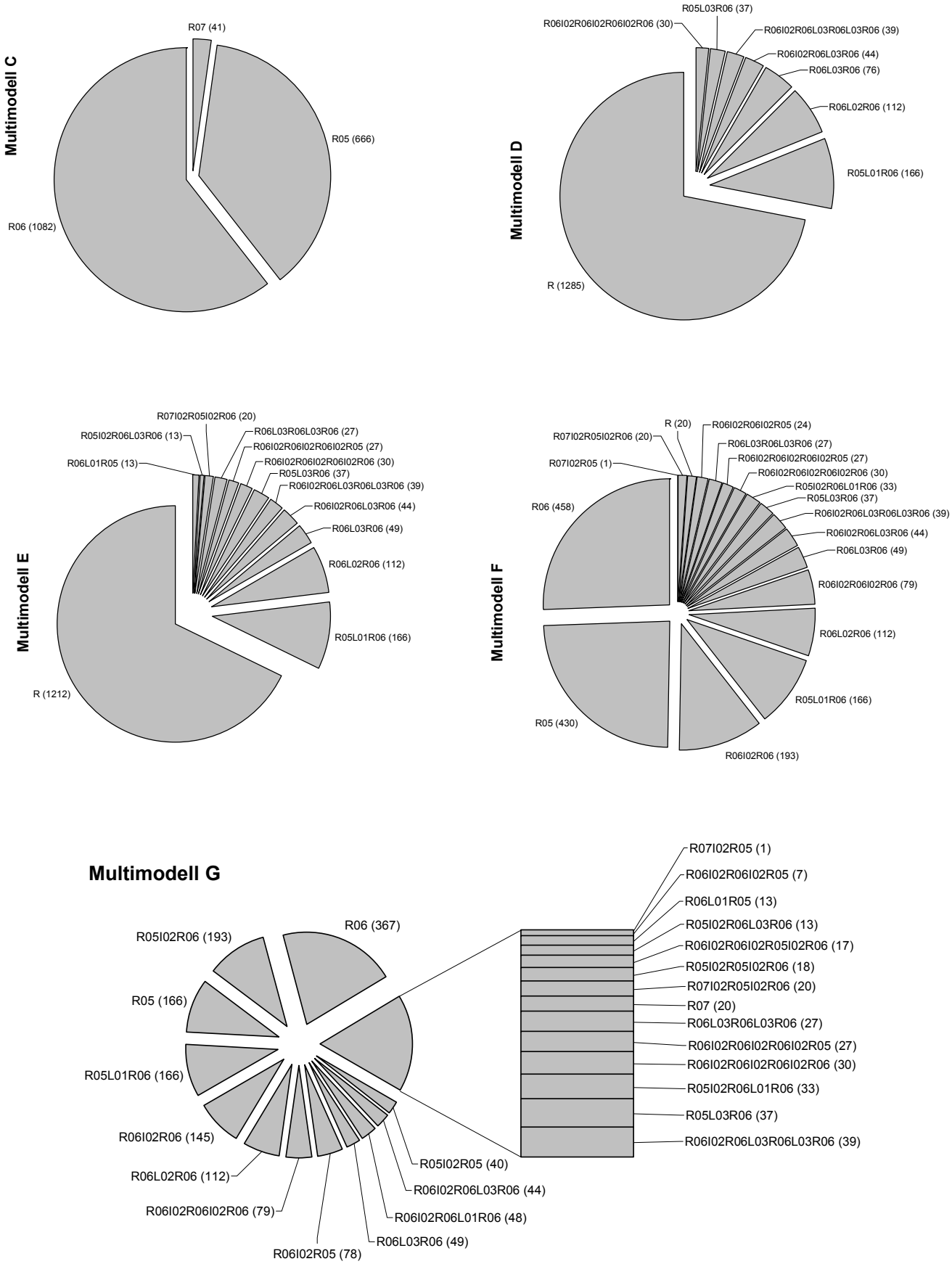
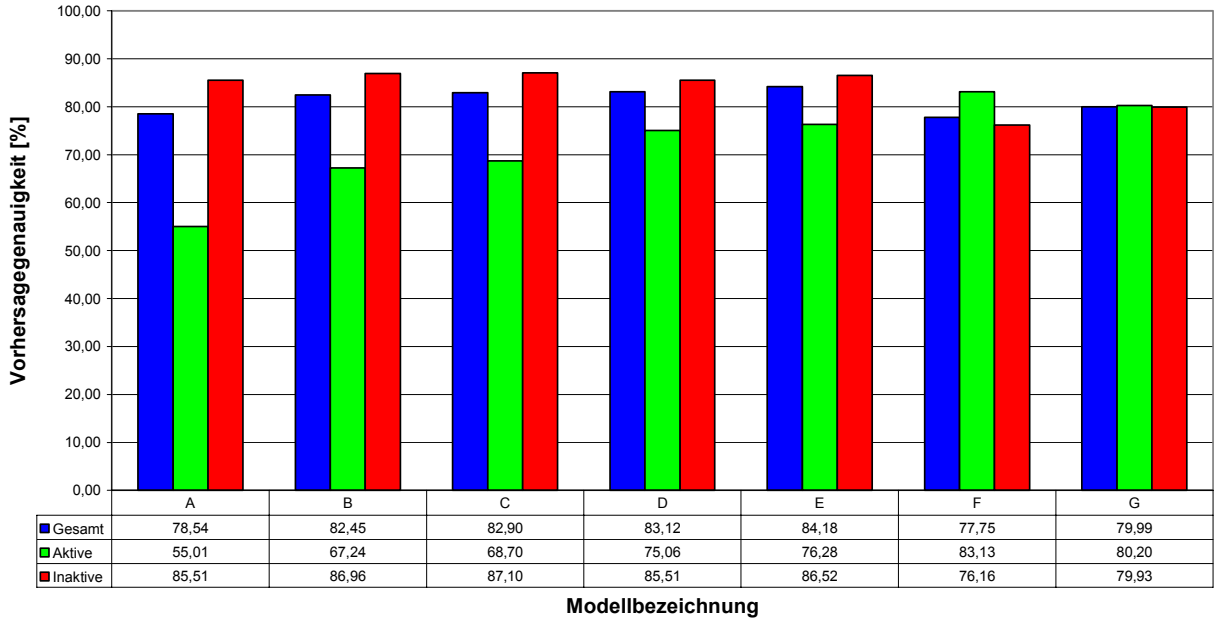


Abb. 6.24: Die Aufteilung der Verbindungen des Datensatzes in den Multimodellen C bis G auf ihre jeweiligen Untermodelle.



	A	B	C	D	E	F	G
Gesamt	78,54	82,45	82,90	83,12	84,18	77,75	79,99
Aktive	55,01	67,24	68,70	75,06	76,28	83,13	80,20
Inaktive	85,51	86,96	87,10	85,51	86,52	76,16	79,93

Abb. 6.25: Vorhersagegenauigkeit der Klassifikationsmodelle A bis G (Validierung).

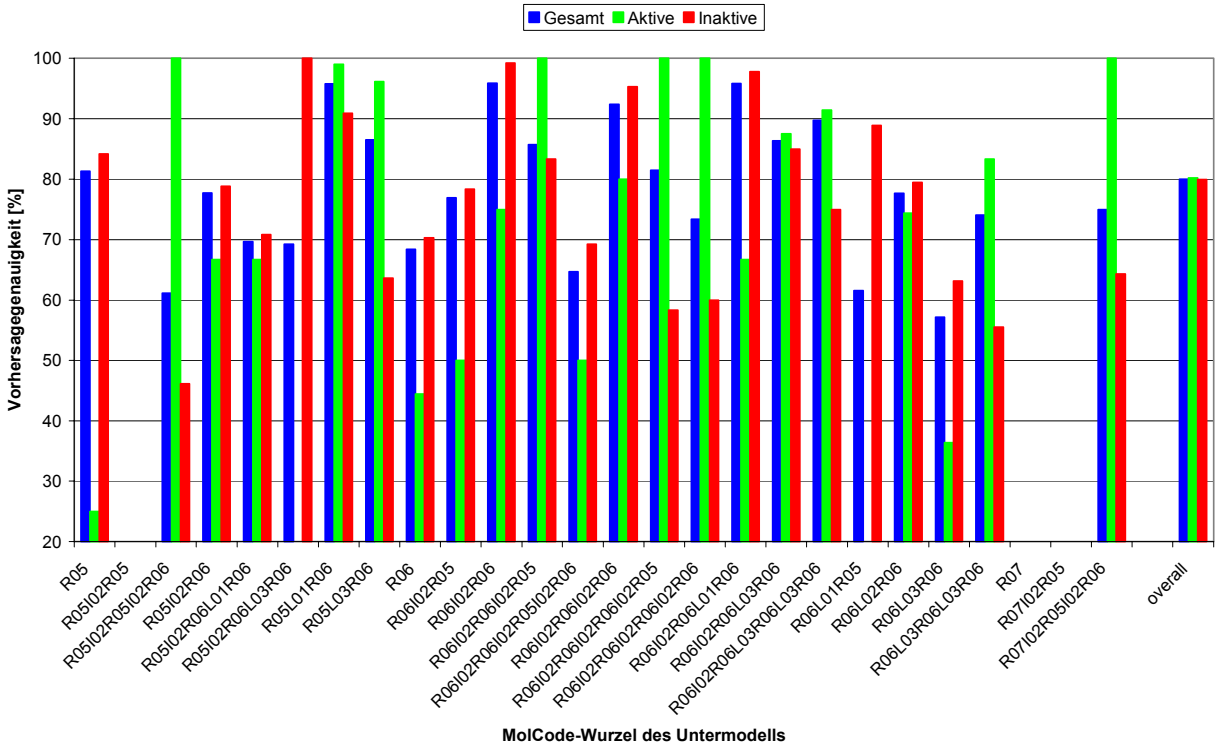
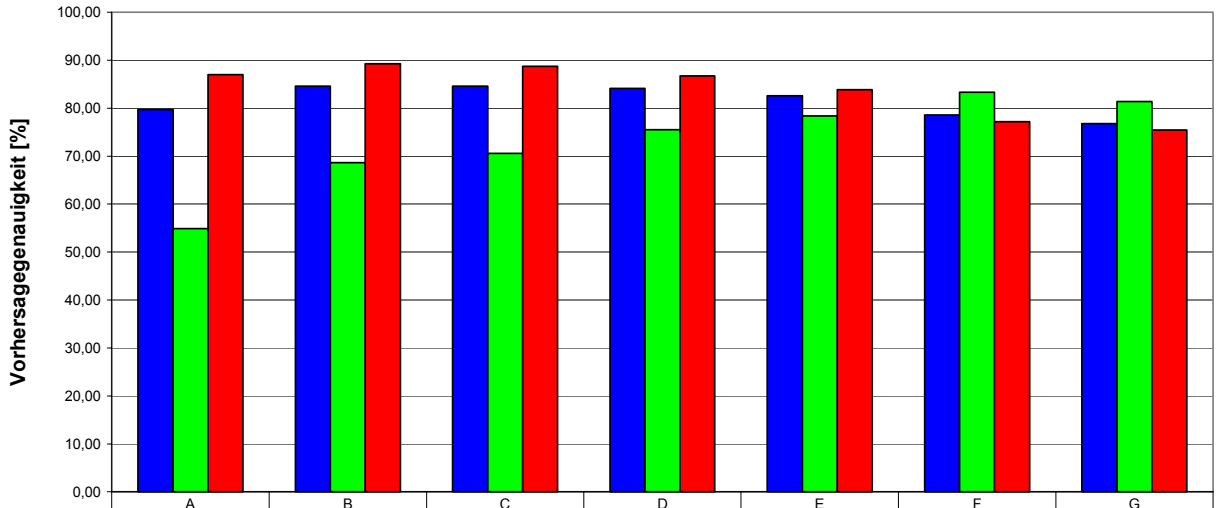


Abb. 6.26: Vorhersagegenauigkeit der 26 Untermodelle in Multimodell G. Sechs von ihnen haben eine Klassifizierungsgenauigkeit von über 99%.

In Multimodell G werden alle zur Verfügung stehenden Untermodelle mit einer Vorhersagegenauigkeit von mindestens 60% verwendet. Damit wird eine Gesamtvorhersagegenauigkeit von 80% und eine Vorhersagegenauigkeit für die Aktiven von 80,2 %

erreicht. Dies ist möglich, da 6 der Untermodelle eine Klassifizierunggenauigkeit von über 99% aufweisen (siehe Abb. 6.26). Drei der 26 Untermodelle haben noch ein spezifischeres Untermodell, dem der überwiegende Teil der Verbindungen zugewiesen wird, die in beiden Modellen enthalten sind. Daher steht für die Erzeugung ersterer keine ausreichende Datenbasis mehr zur Verfügung und sie blieben unberücksichtigt.

Zusätzlich zur eben beschriebenen Selbstklassifizierung zur Validierung der Modelle wurde der Datensatz in einen Trainingsdatensatz und Testdatensatz aufgeteilt. 25% der Daten wurden nach dem Zufall selektiert und exklusiv dem Testdatensatz zugeordnet, dies sind 448 Verbindungen, von denen 102 aktiv sind. Der relative Anteil an Aktiven und Inaktiven von 29% wurde beibehalten. Die Vorhersagegenauigkeit der Modelle A bis G ist in Abb. 6.27 verdeutlicht. Ausgehend vom Einzelmodell A nimmt die Vorhersagegenauigkeit der Aktiven bei Verwendung zusätzlicher Untermodelle bis Modell zu F zu. Das Modell G mit einer sehr hohen Zahl an Untermodellen, zeigt wie im Validierungsfall einen etwas geringeren Zugewinn an Genauigkeit. Die eine Ursache ist, dass die lokalen Modelle unter Verwendung der Verteilung der Aktiven im Trainingsdatensatz ausgewählt wurden. Diese Verteilung entspricht nicht der Verteilung im Testdatensatz. Zusätzlich blieben in Modell G aufgrund der schon beschriebenen Zuordnung von Verbindungen zum speziellsten Untermodell drei Modelle unberücksichtigt. Daher wurden 21 Verbindungen von der Klassifikation ausgeschlossen. Der große Vorteil der automatischen Verwendung von Untermodellen kommt besonders dann zum Tragen, wenn die zu klassifizierenden Verbindungen keinen Querschnitt aus dem Gesamtdatensatz darstellen, sondern tatsächlich den von den Untermodellen abgedeckten Bereichen des TST entsprechen.



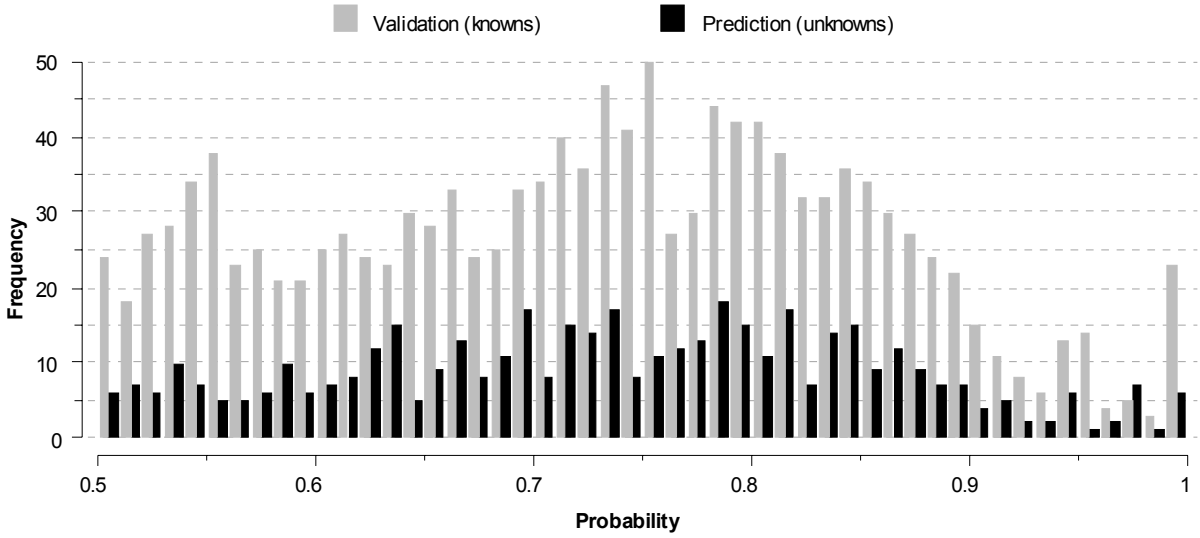
	A	B	C	D	E	F	G
Gesamt	79,69	84,60	84,60	84,15	82,59	78,57	76,79
Aktive	54,90	68,63	70,59	75,49	78,43	83,33	81,37
Inaktive	86,99	89,31	88,73	86,71	83,82	77,17	75,43

Abb. 6.27: Vorhersagegenauigkeit der Klassifikationsmodelle A bis G für den Testset.

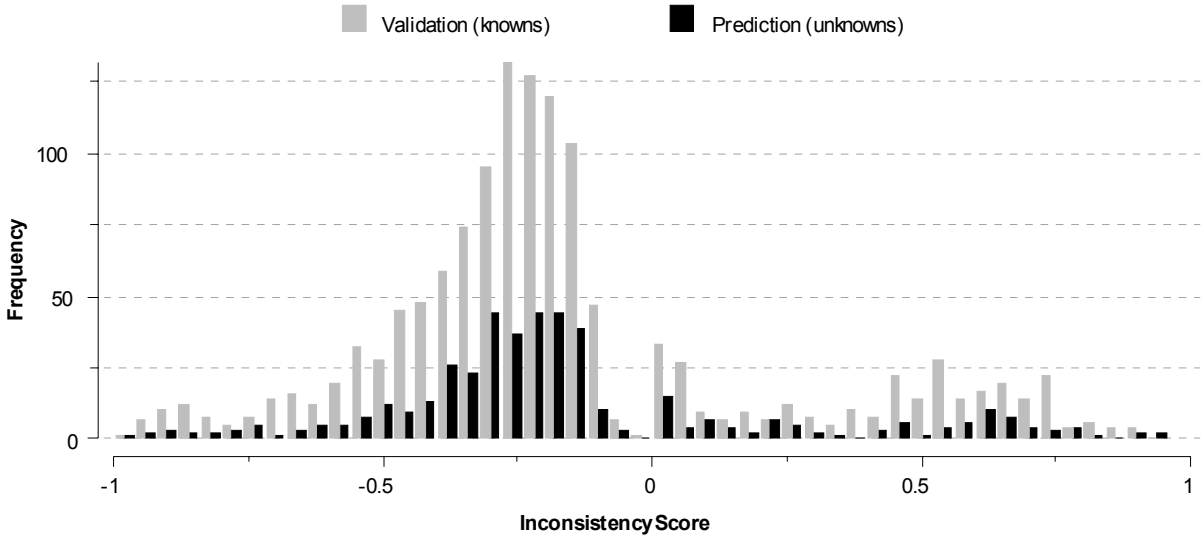
Detailinformationen zum Einzelmodell A und dem Multimodell F sind auf den folgenden Seiten in Abb. 6.28 und Abb. 6.29 aufgeführt.

Zum Multimodell F ist zusätzlich das Untermodell R05L01R06 in Abb. 6.30 abgebildet. Es wurde aus 117 Verbindungen erstellt und dient zur Klassifikation von 49 Verbindungen, die zu 95,92% korrekt klassifiziert werden. Die Histogramme spiegeln diese hohe Rate entsprechend wider: Der überwiegende Teil der Verbindungen wird mit einer sehr großen Wahrscheinlichkeit nahe beim Maximum von 1.0 vorhergesagt. Da die Vorhersagen korrekt sind, liegt deren Inconsistency Score um das Optimum von 0.0.

a)



b)



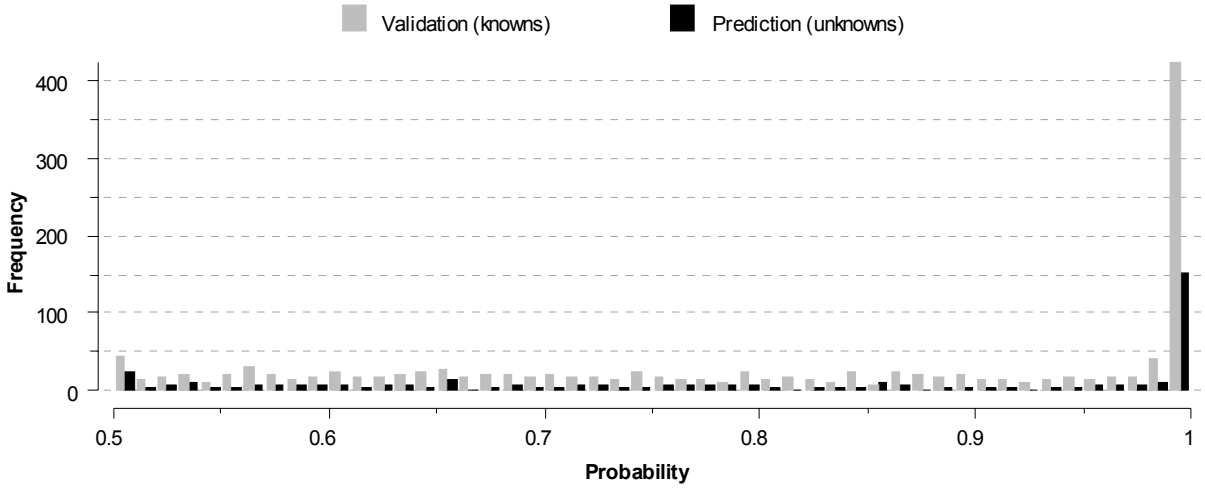
c)

C O N F U S I O N M A T R I X
actual group=col
assigned group=row

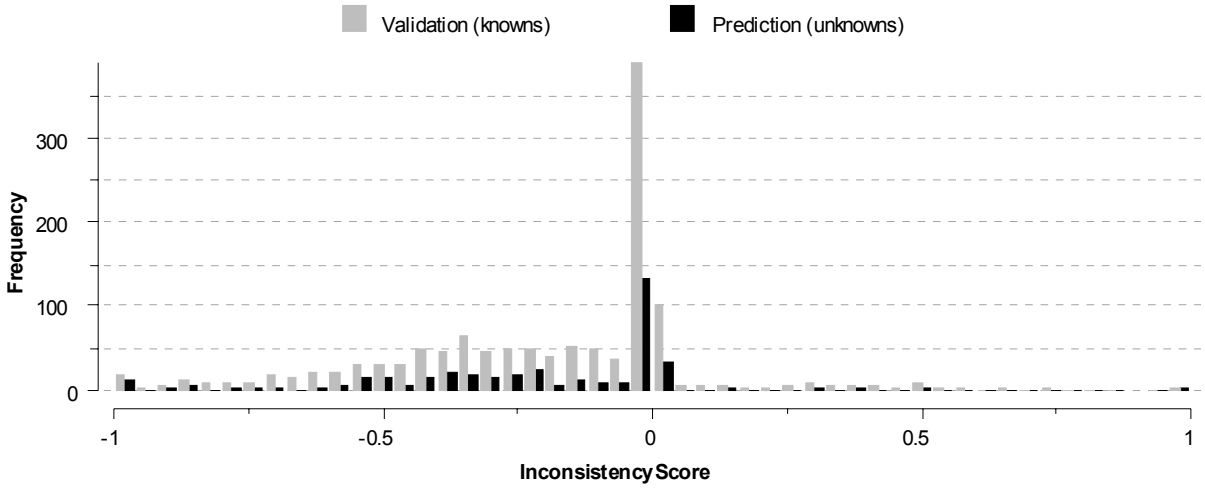
	A C T U A L				
	active	inactive	noclass		
active	56	45	0	101	(54.90% correct classified)
inactive	46	301	0	347	(86.99% correct classified)
noclass	0	0	0	0	(0.00% correct classified)
	102	346	0	448	(79.69% correct classified)

Abb. 6.28: Detailinformationen zu Modell A. Histogramme der Vorhersagewahrscheinlichkeiten (a) und der Inconsistency Scores (b) und Confusion Matrix für den Testset (c).

a)



b)



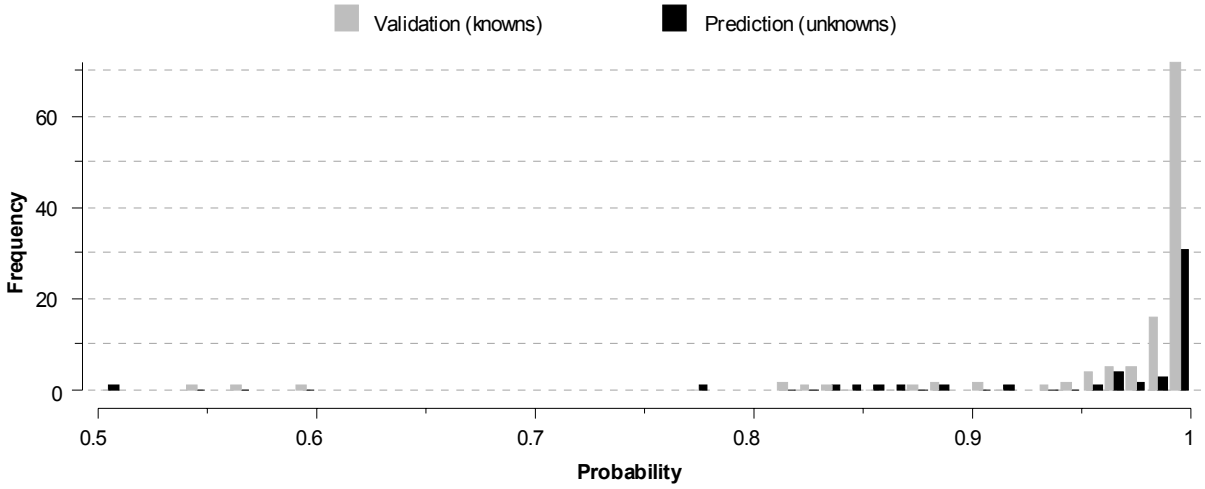
c)

C O N F U S I O N M A T R I X
actual group=col
assigned group=row

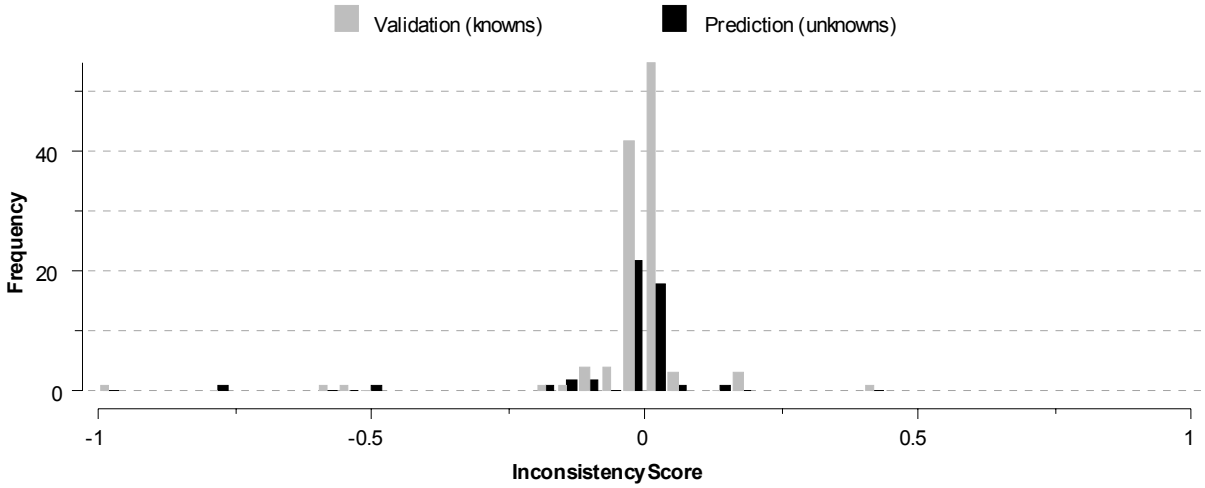
	A C T U A L				
	active	inactive	noclass		
active	85	75	0	160	(83.33% correct classified)
inactive	16	267	0	283	(77.17% correct classified)
noclass	1	4	0	5	(0.00% correct classified)
	102	346	0	448	(78.57% correct classified)

Abb. 6.29: Detailinformationen zu Modell F. Histogramme der Vorhersagewahrscheinlichkeiten (a) und der Inconsistency Scores (b) und Confusion Matrix für den Testset (c).

a)



b)



c)

C O N F U S I O N M A T R I X
actual group=col
assigned group=row

	A C T U A L				
	active	inactive	noclass		
active	25	2	0	27	(100.00% correct classified)
inactive	0	22	0	22	(91.67% correct classified)
noclass	0	0	0	0	(0.00% correct classified)
	25	24	0	49	(95.92% correct classified)

Abb. 6.30: Detailinformationen zum Untermodell R05L01R06 aus Multimodell F. Histogramme der Vorhersagewahrscheinlichkeiten (a) und der Inconsistency Scores (b) und Confusion Matrix für den Testset (c).

6.11 Diskussion des MolCode-basierten Verfahrens

Ausgehend von den lange praktizierten Verfahren der Leitstrukturselektion aus einem Stapel von auf Papier ausgedruckten Molekülstrukturen bzw. der Durchsicht von Spreadsheets mit als Abbildung eingebundenen Strukturdiagrammen, bietet BayTree umfangreiche Möglichkeiten, einen kompletten Datensatz zu präsentieren und damit das Kennenlernen bzw. die Durchsicht der Strukturen zu unterstützen. Im ersten Schritt werden Strukturen, die auf einem gemeinsamen Framework basieren, zusammengefasst und zusätzlich farblich kodiert, um einen Anhaltspunkt zu haben, ob diese Gruppierung schon eine Klassifizierung in „aktive“ und „inaktive“ Gerüste erlaubt. Der Aufbau des Framework erlaubt dem Anwender in jedem Fall schon in dieser Phase, eine schnelle Selektion in synthetisch interessante und uninteressante Gerüste vorzunehmen und letztere auszusortieren. Diese Aufbereitung der Daten ist intuitiv verständlich und ermöglicht es vor allem auch präparativen Medizinischen Chemikern, entsprechende Datensätze zu handhaben, ohne sich mit den Parametern von Chemometrie- oder Datamining-Verfahren beschäftigen zu müssen. Durch die eindeutige und vorhersagbare Zuordnung der Strukturen zu den Knoten im Baum sind keine unerwarteten Ergebnisse bei der Gruppenzusammensetzung zu erwarten. Die alphanumerische Sortierung auf Basis des MolCode spiegelt eine logische Abfolge der Gerüste wider und ermöglicht es, Abschnitte von nicht favorisierten Strukturtypen von der weiteren Betrachtung auszuklammern. Die gilt insbesondere für Strukturen mit sehr kleinen oder sehr grossen Ringen. Im Fall des NCI Aids-Datensatzes wurden durch die Beschränkung auf die R05-R07-Teilbäume 97,2% der aktiven Verbindungen beibehalten. Dabei konnten 151 Verbindungen für die weiteren Betrachtungen ausgeklammert werden.

Die Erstellung einer Hierarchie, basierend auf den Frameworks von Verbindungen, hebt die von BayTree vorgenommene Aufbereitung der Daten von allen anderen gängigen Vorgehensweisen ab. In den üblichen Programmen wie Distill, ClassPharmer und Leadscope liegt der Schwerpunkt auf der direkten Identifizierung von größten gemeinsamen Substrukturen, die in der Regel v.a. durch die Heteroatome bestimmt werden und nur zusätzlich verbindende Anteile des Gerüsts enthalten. Zur Ableitung von SARs ist der fließende Übergang von der Verwendung als Ordnungskriterium zur Verwendung als Parameter für eine Regressionsgleichung von Vorteil, für eine Gruppierung von großen Datenmengen aber nicht die beste Wahl. Zusätzlich ist Distill von vorneherein konzeptionell durch die Laufzeit limitiert, Leadscope durch die vorgegebene Strukturhierarchie und ClassPharmer durch die fehlende Hierachie der Resultate.

Bei kleineren Datensätzen werden je nach struktureller Diversität der Verbindungen sehr viele nicht oder nur einfach populierte Knoten erzeugt. Dann macht es Sinn, diese nach dem gewünschten Detailgrad zusammenzufassen und auf der Stufe von Unterbäumen zu betrachten. Je größer der Datensatz ist, desto höher sind die einzelnen Knoten populiert und desto weniger neue Knoten müssen im TST ergänzt werden. In Abb. 6.31 sind die Gesamtzahl der erzeugten Knoten und die Anzahl der darin belegten Knoten für den NCI Aids-Gesamtdatensatz und die Teilbäume R05 bis R07 (jeweils ohne Verbindungen mit Metallatomen) gegenübergestellt. In jedem Fall ist deutlich erkennbar, dass die Kurven immer flacher werden. Auch im Teildatensatz ist der überwiegende Anteil an Knoten repräsentiert. Die Zahl der Verbindungen, die zu überwiegend geringpopulierten Knoten bzw. TSP führen, ist deutlich geringer.

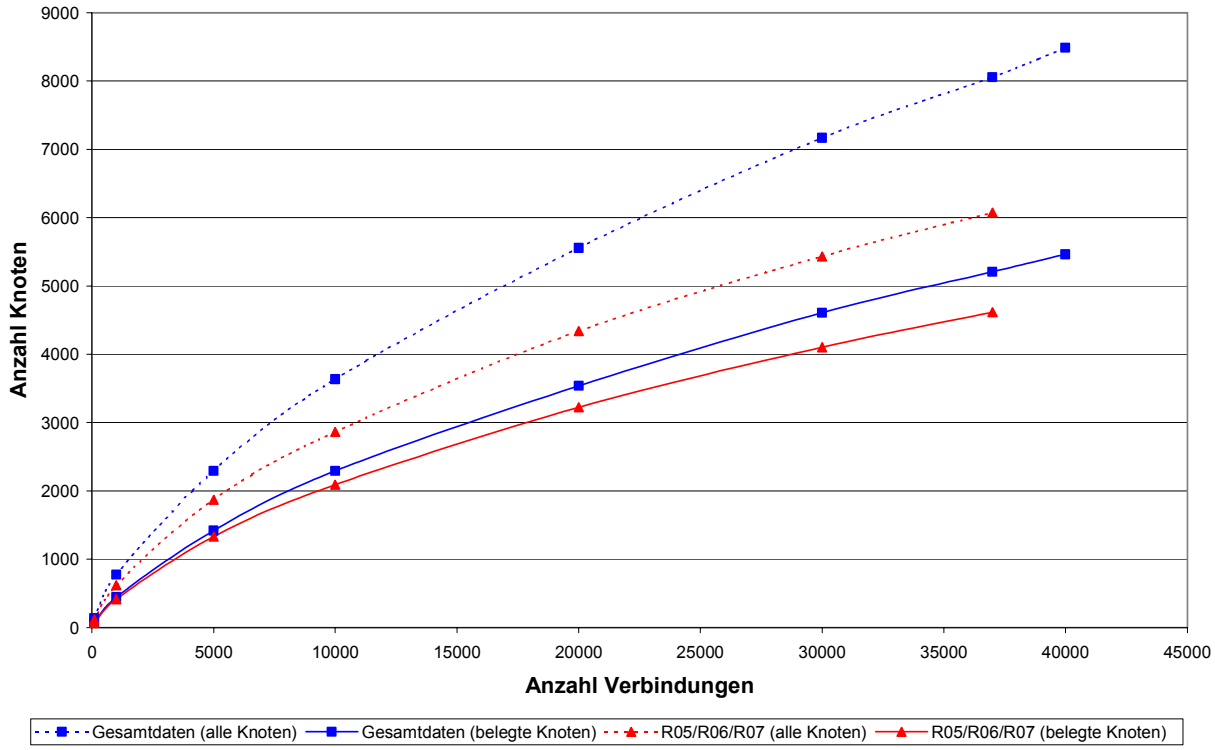


Abb. 6.31: Gegenüberstellung der Gesamtzahl der erzeugten Knoten und die Anzahl der darin belegten Knoten für den NCI Aids Gesamtdatensatz (blau) und die Teilbäume R05 bis R07 (rot).

Für eine vollständige rein visuelle Inspektion ist der erzeugte TST in der Regel zu umfangreich. Werden hotspots farblich hervorgehoben, kann eine schnelle Fokussierung auf die interessanten Bereiche erfolgen. Solche hotspots können Knoten sein, die entweder sehr hoch populiert sind oder die sehr viele aktive Verbindungen enthalten. Im NCI-Aids Datensatz entspricht die Abfolge der Knoten, von einer indikationsspezifischen Ausnahme abgesehen, der publizierten typischen Reihenfolge in Wirkstoffdatenbanken.

Im NCI-Aids Datensatz konnte gezeigt werden, dass trotz seiner grossen Strukturvielfalt die aktiven Verbindungen im Wesentlichen auf nur 17 Frameworks beruhen. Alleine aufgrund der Form und Größe der Gerüste war es möglich, eine Zuordnung zum Zielprotein bzw. eine entsprechende Gruppierung vorzunehmen.

In der Praxis macht es Sinn, nach der Identifizierung der häufig vertretenen Gerüsttypen in deren durch die Hierarchie definiertem Umfeld nach ebenfalls aktiven, aber gering populierten alternativen Templaten zu suchen bzw. das kleinste mögliche Gerüst zu finden. Darüber hinaus können zusätzliche Informationen über die vom Zielprotein tolerierten Veränderungen am Gerüst abgeleitet werden, z.B. indirekt die Größe von hydrophoben Taschen oder Bereiche von Liganden, die aus dem Protein ins Lösungsmittel hineinragen.

Die Hierarchie basiert auf durch ein Regelwerk identifizierte und priorisierte größte gemeinsame Substrukturen, die umso größer sind, je tiefer sich die Knoten im Baum befinden. Besonders nützlich ist sie auch für die Selektion von Verbindungen, die wie im folgenden beschrieben, zur Erstellung von Untermodellen verwendet werden.

Da die aktiven Verbindungen zahlenmässig typischerweise deutlich unterrepräsentiert sind, ist deren Vorhersagegenauigkeit typischerweise, unabhängig vom eingesetzen Klassifikations-verfahren bzw. den verwendeten Deskriptoren, nicht zufriedenstellend. Es konnte gezeigt

werden, dass die Verwendung von Untermodellen zu einer deutlichen Verbesserung der Vorhersagegenauigkeit der Aktiven führt. Bei Verwendung von 13 Untermodellen betrug die Vorhersagegenauigkeit der Aktiven 82,4% im Vergleich zu 55,1% bei der Einzelmodell-klassifikation.

Die Untermodelle werden quasi lokal für eine geeignete homogene Teilmenge erzeugt und dann für die Prädiktion von strukturell dazu passenden Verbindungen herangezogen. Als homogene Teilmengen werden geeignete Subtrees im TST verwendet. Von Vorteil ist, dass die Untermodelle qualitativ hochwertig im Sinne einer Intrapolation verwendet werden und einfach zu interpretieren sind. Nachteilig ist, dass alle nicht von Untermodellen abgedeckten Strukturen im TST von einem einzelnen „Restmodell“ vorhergesagt werden, dessen Vorhersagegenauigkeit noch schlechter ist als das eines einzelnen Modells über den Gesamtdatensatz. Zu bedenken ist aber auch, dass vom mathematischen Standpunkt aus eine Klassifizierung aller aktiven Verbindungen als inaktiv rechnerisch zwar einen hohen „Vorhersagegenauigkeitswert“ erzeugt, der umso günstiger ist, je größer der Anteil an Inaktiven ist, der aber für die angestrebte Anwendung nutzlos ist. Andererseits fehlt für unterrepräsentierte Einzelverbindungen letztendlich eine geeignete Datenbasis zur Erzeugung eines adäquaten Modells.

Die BayTree zugrundeliegende Methodik der Anordnung über den MolCode ermöglicht es zusätzlich, mehrere Datensätze in effizienter Weise zusammenzuführen und damit dynamisch, aber dennoch determiniert, Strukturen zum topologischen Strukturbaum zu ergänzen.

7 BayTree-Bedienungsanleitung

7.1 Grundlagen

7.1.1 Konfiguration und Programmstart

Die IRIX-Version des BayTree-Programms ist in das Bayer Software-Konfigurationssystem eingebunden. Zur Aktivierung dient die Befehlsfolge: **cfg baytree on** in einer UNIX-Shell. Dadurch wird die jeweils aktuellste Version zur Verfügung gestellt. Aufruf des Programms erfolgt durch Eingabe von **baytree <*hitliste.hits*>**. Die Windows-Version wird durch Doppelklicken des BayTree-Icons gestartet oder indem die einzulesende Hitlist-Datei auf dieses gezogen und fallengelassen wird. Von der Eingabeaufforderung aus erfolgt der Start durch Aufruf von **baytree.bat <*hitliste.hits*>**.

7.1.2 Switches beim Programmaufruf

-nogui	BayTree wird ohne Benutzeroberfläche gestartet, um für große Datensätze ein Resultfile zu erstellen.
-oracle	Die Ergebnisdaten werden zusätzlich in die Oracle-Datenbank BAYTREE eingetragen.
-nometals	Strukturen, die mindestens ein Metallatom (Cu, Pd, Pt etc.) enthalten, werden nicht in den Baum eingetragen, sondern übersprungen. Im Textfenster wird nach dem Einlesen aller Strukturen eine entsprechende Meldung mit allen RegIds ausgegeben.
-splitdhs	Der MolCode wird um die Abschnitte Dxx (Anzahl der Doppelbindungen) Hxx (Anzahl der Heteroatome) und Sxx (Anzahl der Substituentenatome) erweitert und die Knoten im Baum werden entsprechend gesplittet.
-kmodule	Der MolCode wird um das Konnektivitäts-Modul erweitert und jeweils ein zusätzlicher Knoten im Baum eingezeichnet.
-genericrings	Die Größe xx aller Ringe Rxx wird einheitlich auf xx=99 gesetzt.
-genericlinker	Die Länge xx aller Linker Lxx wird einheitlich auf xx=99 gesetzt.

Die zur Übernahme der Daten verwendete Tabelle btreedata der BAYTREE-Datenbank (ORACLE_SID=BAYTREE) ist durch folgendes SQL-Kommando[246, 247] angelegt worden:

```
create table btreedata (
        regid           varchar(100) not null,
        molcode         varchar(1000),
        fragments       varchar(1000),
        dataset         varchar(100)
);
```

Das Feld regid enthält die Registrierungsschlüssel zur Identifizierung der Verbindung und dataset den Namen des bearbeiteten Datensatzes. Er entspricht dem unter BayTree verwendeten Dateinamen. molcode und fragments enthalten den erstellten MolCode und die Zuordnung der Atomnummern zu den Fragmenten.

7.1.3 Konfigurations-Datei .baytreerc

Beim Programmstart von BayTree wird die Datei **.baytreerc** aus dem Home-Verzeichnis des Benutzers eingelesen. In ihr sind verschiedene Konfigurationsinformationen hinterlegt, durch die die Defaulteinstellungen verändert werden können. Die Einträge bestehen aus einem Schlüsselwort und, durch ein Gleichheitszeichen getrennt, einem zugehörigen Wert. Leerzeichen und Tabulatoren können beliebig zur Ausrichtung verwendet werden und werden ignoriert. Zeilen, die mit einem Doppelkreuz „#“ beginnen, werden als Kommentarzeilen betrachtet und nicht ausgewertet. Folgende Schlüsselworte werden derzeit unterstützt.

Schlüsselwort	**Defaultwert**	**Beschreibung**
color_bg	darkgray	Hintergrundfarbe
color_fg	black	Vordergrundfarbe
color_mclabel	yellow	Farbe der MolCodes
color_cmpdlabel	black	Farbe der Verbindungsnamen
color_occlabel	blue	Farbe der Occupancy-Beschriftung
color_selected	blue	Farbe der selektierten Strukturen
color_commoncore	lightgray	Farbe des übergeordneten Framework-Fragments
depictsize_fw	100	Größe der Framework-Depiction (in Pixeln)
depictsize_cmpd	100	Größe der Verbindungs-Depiction (in Pixeln)
canvasfont	MS_SansSerif_8	Schriftart für Text
nodeborder	20	Abstand um jeden Knoten (in Pixeln)
leveldistance	30	Abstand zwischen den Baumebenen (in Pixeln)
last_used_build	20010101	Builddatum der zuletzt verwendeten Version von BayTree. Der Eintrag wird automatisch aktualisiert und wird verwendet, um beim erstmaligen Aufruf einer neueren Version den Benutzer über die zwischenzeitlich erfolgten Veränderungen zu informieren.

7.1.4 GUI-Komponenten

Zur Auswertung von Datensätzen bietet BayTree ein komfortables graphisches Benutzer-interface. Zu jeder Instanz gehört ein Terminal-Fenster (bzw. unter Windows ein Console-Fenster) und das Hauptfenster. Optional werden bei Aufruf der entsprechenden Funktionen zusätzliche Fenster für den GlobalView, die MC-Liste, die XR-Tabelle und Hitlist- oder Einzelverbindungs-Depiction geöffnet. Zur Wahrung des Überblicks enthalten alle Fenster den Namen des Datensatzes in der Titelzeile bzw. in den Icon-Bezeichnungen. Dies ist insbesondere von Vorteil, wenn zur vergleichenden Auswertung von Datensätzen mehrere Instanzen des BayTree-Programms nebeneinander gestartet werden, was problemlos möglich ist.

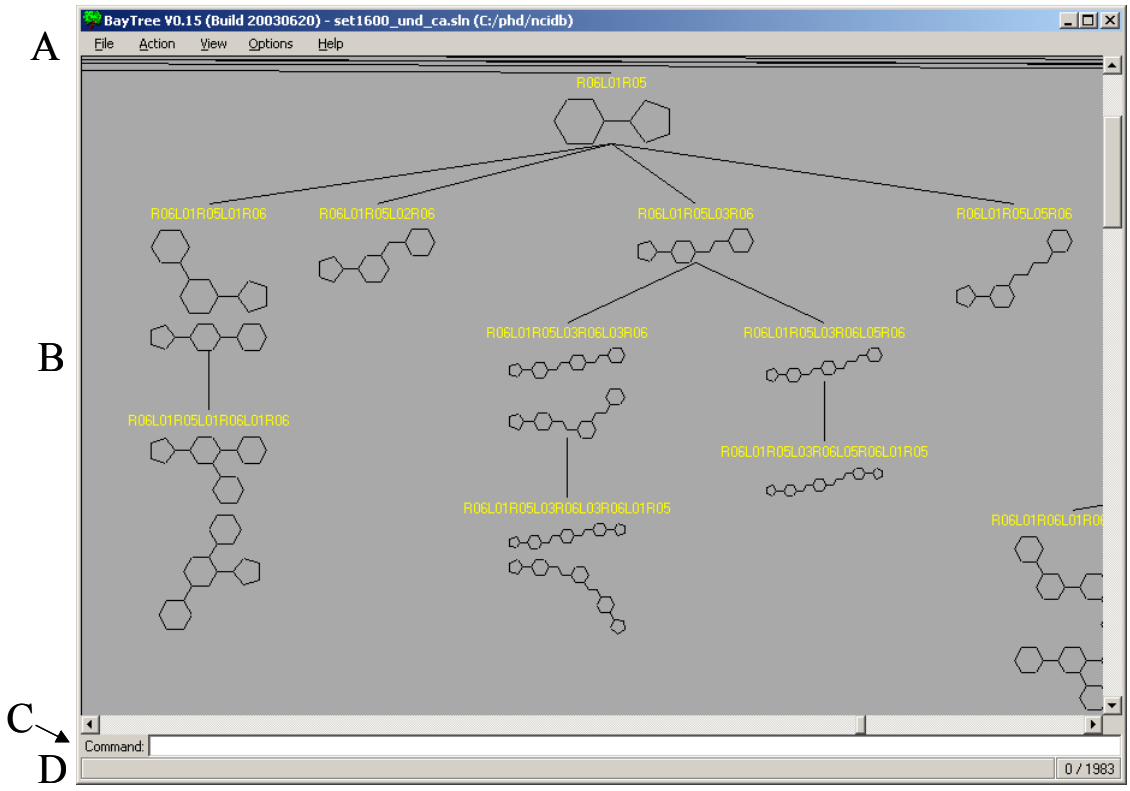


Abb. 7.1: Screenshot des BayTree GUI.

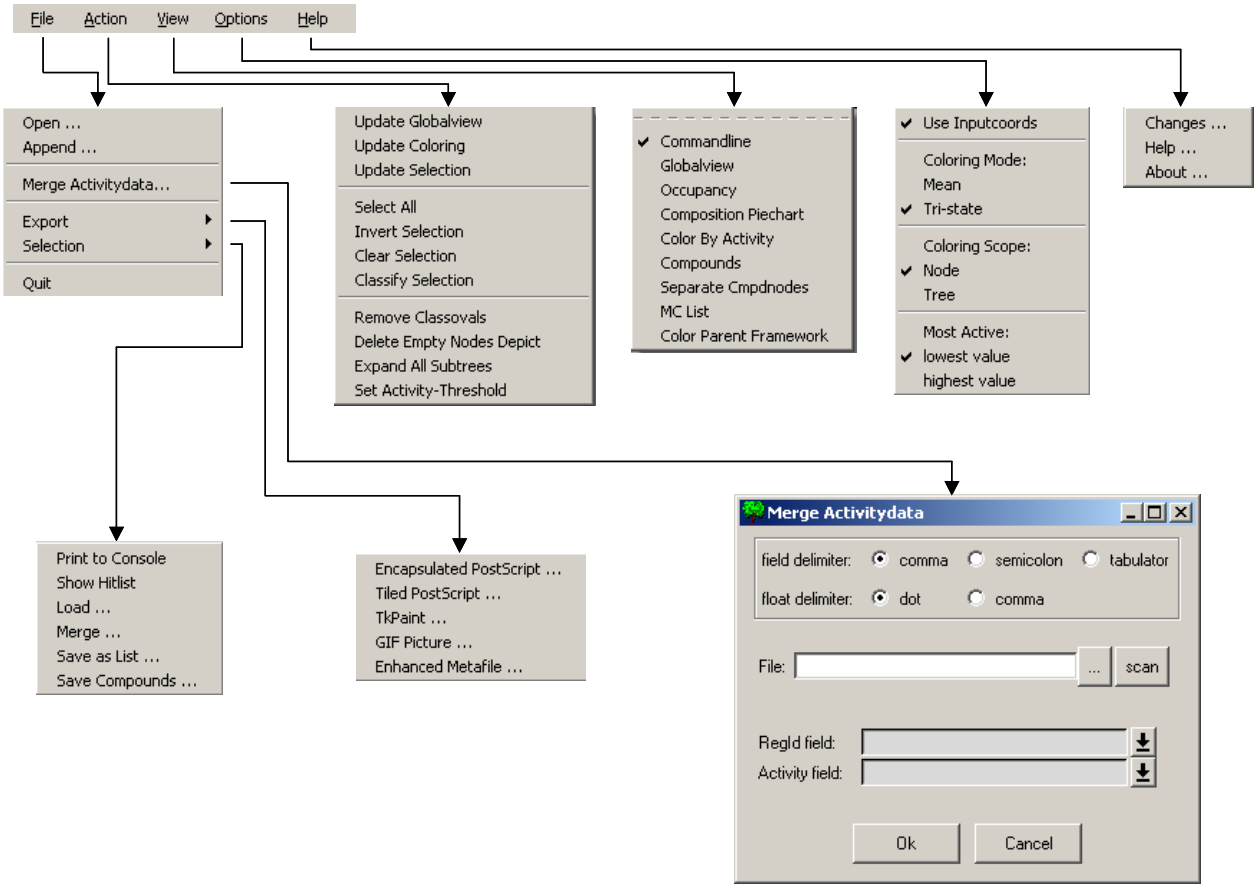


Abb. 7.2: BayTree-Menüstruktur.

Das BayTree-Hauptfenster enthält vier Bereiche: Am oberen Rand die Hauptmenüzeile A zum Aufruf der Programmfunktionen und am unteren Rand die Statuszeile D für Meldungen des Programms. Im rechten Teil enthält diese, durch einen Schrägstrich getrennt, die Anzahl der aktuell selektierten Verbindungen und die Gesamtzahl der eingelesenen Verbindungen. Darüber ist optional die Kommandozeile C zur Eingabe von Kommandos eingeblendet. Die Zeichenfläche B zur Darstellung des topologischen Strukturbaums nimmt den größten Raum ein.

7.1.5 Command-Steuerung

Alle Funktionen können auch unabhängig von der Mausbedienung per Kommando angewählt werden. Dazu dient unterhalb der Baumzeichenfläche das Eingabefeld oder das Konsolefenster. Die Kommandos können abgekürzt werden, solange sie eindeutig bleiben. Im folgenden sind, von einigen Ausnahmen abgesehen, nur Befehle aufgeführt, die nicht durch das GUI angewählt werden können.

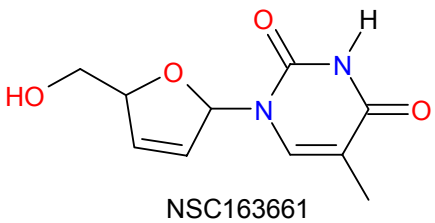
center_mc <*molcode*>	navigiert im Baum zu dem Knoten mit dem angegebenen MolCode. Dieser wird in der Zeichenfläche zentriert.
center_cmpd <*regid*>	zentriert die Depiction der <*regid*> oder den Knoten, der das Framework der Verbindung <*regid*> enthält
picker_mc <*molcode liste*>	öffnet Listenfenster zum Ansteuern durch Auswahl aus mehreren MolCodes
picker_cmpd <*regid liste*>	öffnet Listenfenster zum Ansteuern durch Auswahl aus mehreren RegIds
set_activity_from_field <*attrname*>	wählt in einer eingelesenen Hitliste ein alternatives SLN-Attribut aus, das fortan als Aktivitätsinformation verwendet wird
scan_for_activityfield	gibt zur eingelesenen Hitliste alle Attributbezeichnungen aus, die als Aktivitätsfelder in Frage kommen
depictcmpd <*regid*>	öffnet ein Fenster mit einer größenveränderbaren Struktur-Depiction zu <*regid*>
update_selectionstate <*color*>	färbt alle aktuell selektierten Verbindungen unabhängig von der Voreinstellung in der Farbe <color>
descriptors2cache <*csv file*>	importiert alternative Deskriptoren für die Klassifizierung (Format des <*csv file*>: 1. Spalte RegId; alle weiteren Spalten enthalten durch Kommata getrennt die Deskriptoren)
make_globalview	öffnet das GlobalView-Fenster
show_occupancy	blendet unterhalb der Framework-Knoten den Belegungsgrad ein
hide_occupancy	… und wieder aus
colorbyactivity	aktiviert die Aktivitätsfärbung der Knoten
colorbyocc	färbt die Knoten entsprechend ihrem Belegungsgrad
save_selection [*filename*]	schreibt die RegIds der selektierten Verbindungen in die Datei
classifiy_using_submodels <*molcode liste*> <*regid liste*>	klassifiziert die RegIds aus <*regid liste*> unter Verwendung von Untermodellen zu den MolCodes der <*molcode liste*>

Im Eingabefeld steht das Ausrufezeichen zum Zugriff auf die Befehlshistorie zur Verfügung: Das vorhergehende Kommando wird durch Eingabe von **!!** erneut angezeigt und durch **!***abc*

wird das zuletzt ausgeführte Kommando, das mit *abc* beginnt, angezeigt. Folgt auf das Ausrufezeichen eine Zahl (!nnn), wird auf das nnn-letzte Kommando zugegriffen. Wenn das Eingabefeld den Tastaturfokus besitzt, kann zusätzlich mit den Up/Down-Pfeiltasten durch die vorhergehenden Kommandos geblättert werden. Alle Zeichen im Eingabefeld werden durch Drücken der Taste **ESC** oder **Ctrl-U** gelöscht.

7.1.6 Struktur-Eingabedaten: Formate und Erzeugung

Eingabefile ist eine Unity SLN-Hitliste[248], die evtl. nach Eigenschaftsmerkmalen vorgefiltert ist. Sie enthält die Strukturinformationen der Moleküle codiert in einem SLN-String mit Attributen, die durch Semikolons getrennt in Klammern („<...>“) erscheinen. Obligatorisch ist ein Attribut, das der Struktur einen eindeutigen Namen (regId, Registrierungsschlüssel) zuordnet. Das Eingabefile wird entweder direkt beim Aufruf des Programms angegeben oder über **File | Open ...** eingelesen. Unter dem Menüpunkt **File | Append ...** können in einen bestehenden Baum weitere Strukturen eingelesen werden. Beim Einlesen werden zu jedem Eingabefile automatisch zwei Filterfiles verwaltet: *<hitliste>_select.sel* und *<hitliste>_ignore.sel*. Die Zeichenkette *<hitliste>* muss dabei dem Dateinamen des Hitlistfiles entsprechen. Existieren diese Dateien im gleichen Verzeichnis, werden alle RegIds, die in **_ignore.sel* enthalten sind, beim Einlesen der Hitliste übersprungen bzw. nur die RegIds aus **_select.sel* beibehalten. Das Format der Filterlisten entspricht den aus BayTree abspeicherbaren Selectionslisten.



NSC163661

Für die Verbindung NSC163661 aus dem NCI Datensatz sieht der Hitlisteintrag wie folgt aus:

```
N[1](CH=C(C(NHC@1=O)=O)CH3)C[15]HOCH(CH=CH@15)CH2OH\
<name="NSC163661";regId="NSC163661";act=2;ic50=-3.730>
```

Die Attribute name und regId enthalten beide den Registrierungsschlüssel NSC163661. Das Attribut act enthält den Wert 2 für CA (confirmed active) und das Attribut ic50 den numerischen IC_{50}-Wert der Verbindung aus dem HIV-Screen (siehe Abschnitt 6.1).

Wird das Programm mit einem SD-File (Extensions: *.mol, *.sd, *.sdf, *.maccs) aufgerufen, wird dieses zuerst mittels **dbtranslate** (und den Optionen –type maccs –translate sln –3d +stereo +chiral) in eine Hitliste konvertiert, welche bei folgenden Programmaufrufen direkt verwendet werden kann. In einem Prescan der ersten hundert Strukturen wird das regId-Attribut nach folgendem Verfahren ermittelt. In Frage kommen nur Attribute, die für alle Strukturen einen unterschiedlichen Wert haben. Gibt es mehrere solcher Attribute, werden sie in der folgenden Reihenfolge verwendet: BAY_NO, COS_NO, EXTREG, COP_NO. Verschiedene Groß-/Kleinschreibungsvarianten sind dabei äquivalent. Existiert kein Attribut, das für alle Einträge einen unterschiedlichen Wert hat, wird das verwendet, das die meisten

unterschiedlichen Werte besitzt. Diese Rückfalllösung kann z.B. erforderlich sein, wenn der Eingabedatensatz doppelte Einträge enthält. Existieren solche Duplikate, wird der jeweils zuerst auftretende Eintrag verwendet und alle weiteren werden übersprungen. Über die Gesamtzahl und die Namen der im Datensatz vorhandenen Mehrfacheinträge informiert die Meldung „skipped duplicates“ im Textfenster.

Die notwendigen Eingabe-Strukturfiles können (bei Bayer Pharma) auf folgenden Wegen erhalten werden:

- PIX-Client
- Isis for Excel (Nach Markieren der entsprechenden Spalten, Abspeichern als SD-File)
- ISIS/Base HVIEW
- Basis-Applikation (Bayer AG In-house-Software)
- Erzeugung aus BAY/COS-Nummernliste in einer Unix-Shell. *liste.txt* ist eine vom Benutzer erstellte Eingabeliste, z.B. eine exportierte Spalte aus einem Excel-Spreadsheet, die BAY-Nummern für Prüfpräparate bzw. COS-Nummern für CombiChem-Präparate enthält.
  grep ^BAY *liste.txt* | **dbexport** -database /db/unity/basis/stock/stock.db -type sln -query reg > *out_bay.hits*
  grep ^COS *liste.txt* | **dbexport** -database /db/unity/cosis/stock/stock.db -type sln -query reg > *out_cos.hits*
  Zum Zugriff auf das Datenbankverzeichnis muss der Benutzer der entsprechenden Unix-Gruppe zugeordnet sein.

Zeilen mit einem „#“ am Zeilenanfang werden als Kommentarzeilen betrachtet und übersprungen. Falls einzelne Strukturen/SLNs Probleme bereiten, können sie auf diesem Weg von der Bearbeitung ausgeschlossen werden.

Die Strukturen des Eingabefiles werden eingelesen und sequentiell abgearbeitet.

- Die Struktur wird in Substituenten und das Framework zerlegt; das Framework wird in die aufbauenden Ringe und Linker unterteilt.
- Die Ringe und Linker werden priorisiert und entsprechend dem Regelwerk in einem linearen MolCode kodiert.
- Der MolCode wird als Pfadbeschreibung im Baum verwendet. Noch nicht vorhandene Framework-Knoten werden automatisch erzeugt. Die Gesamtstruktur erscheint als Endknoten (Blatt, Leaf) im Baum.

Jeder Baumknoten enthält den MolCode als Textlabel, der per Cut und Paste kopiert werden kann, und die im Datensatz vorhandenen zugehörigen Frameworks. Zu den MolCodes existieren meist zusätzliche Verknüpfungsmuster, die im Rahmen einer Lückenanalyse enumeriert werden könnten.

Die zusätzlich erzeugten Informationen dieser Prozessierung (MolCode, Atomnummern der einzelnen Ringe und Linker) werden als zusätzliche Attributfelder zur Struktur gespeichert und in ein Ergebnisfile mit der Namensergänzung „_result“ geschrieben.

Der Result-Eintrag für das Molekül NSC163661 sieht wie folgt aus:

```
N[1](CH=C(C(NHC@1=O)=O)CH3)C[15]HOCH(CH=CH@15)CH2OH<coord2d=(4.157,0.456),\
(4.657,-0.410),(4.347,-0.947),(5.657,-0.410),(6.157,0.456),(5.657,1.322),\
(5.967,1.859),(4.657,1.322),(4.157,2.188),(7.157,0.456),(6.157,-1.276),\
```

```
(5.620,-1.586),(6.467,-1.813),(6.694,-0.966),(3.157,0.456),(3.439,-0.096),\
(2.570,1.265),(1.619,0.956),(1.716,1.569),(1.619,-0.044),(1.117,-0.408),\
(2.570,-0.353),(2.761,-0.942),(0.810,1.544),(1.256,1.975),(0.463,2.058),\
(-0.104,1.137),(-0.606,1.502);act=2;ic50=-3.730;f01:=15,17,18,20,22,15;\
f02:=1,15;f03:=1,2,4,5,6,8,1;mc="R05L01R06";mcx="R05L01R06Kaab";\
name="163661";regId="163661">
```

Die zusätzlichen Attribute sind:

coord2d	2D-Koordinaten zur Erstellung der Struktur-Depiction
f01:, f02:, f03	Zuordnung der Atomnummern zu den Fragmenten R05, L01 und R06. Der nachfolgende Doppelpunkt kennzeichnet die Attribute als „reorderable“, d.h. die Atomnummern beziehen sich auf die Abfolge der Atome im angegebenem SLN-String. Wird dieser umgestellt, müssen die Atomnummern dieser Attribute entsprechend angepasst werden.
mc	MolCode der Struktur
mcx	erweiterter (eXtended) MolCode

Der Prozessierungsvorgang der Verbindungen kann mit **Ctrl-C** abgebrochen werden. Es erscheint die Meldung „User interrupt request“ im Console-Fenster und nach Abarbeiten der aktuellen Verbindung wird der Vorgang abgebrochen. Diese Verzögerung stellt einen stabilen Systemzustand sicher. Um BayTree komplett abzubrechen: Ctrl-Z drücken, am Prompt **jobs** eingeben und dann **kill %X**, wobei X die Zahl ist, die in eckigen Klammern am Zeilenanfang des jobs-Output steht. Die Windows-Version wird im Task-Manager (Aufruf mittels der Tastenkombination **Ctrl-Alt-Delete**) nach Auswahl in der Prozessliste durch Anklicken von „Task beenden“ abgebrochen.

7.1.7 Import von Aktivitätsdaten

File | Merge Activitydata...

In den ersten beiden Zeilen der Dialogbox werden die Trennzeichen für die einzulesende Textdatei eingestellt. Zur Auswahl stehen Komma, Semikolon und Tabulator als Spaltentrennzeichen, und Punkt bzw. Komma als Dezimaltrennzeichen, die je nach Herkunft der Datei anzupassen sind.

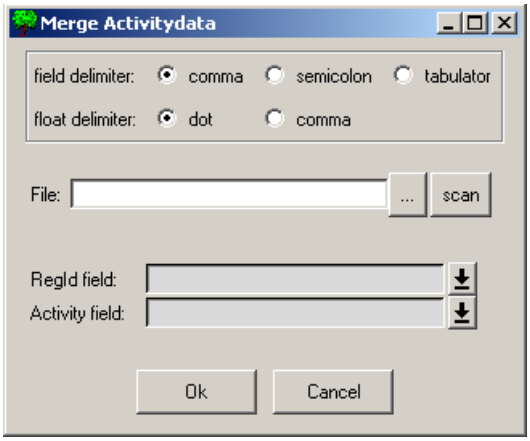


Abb. 7.3: Merge Activitydata Dialog.

Die Zeilenendezeichen NL (Unix) bzw. NL/CR (PC) werden automatisch richtig behandelt. Das deutsche Excel erzeugt standardmässig eine CSV-Datei mit Semikolon als Spaltentrennzeichen und Komma als Dezimaltrennzeichen.

Im Feld File wird der Dateiname direkt eingegeben oder mittels der **...**-Schaltfläche über die Dateiauswahlbox selektiert. Alle in der Datei vorhandenen Spaltennamen werden in die RegId- und Activity-Listenfelder übertragen und können darüber ausgewählt werden. Nach Anklicken von OK werden die Aktivitätsdaten eingelesen und den bereits geladenen Strukturen zugeordnet. Im Textfenster erscheint eine kurze Statusmeldung, die angibt, wieviele RegIds zugeordnet (field found) werden konnten, wieviele RegIds gefehlt (missing values) haben und wieviele nicht-numerische Werte ignoriert worden sind. Fehlende Werte werden intern durch –1 kodiert. Wenn eine diesbezügliche Warnung ausgegeben wird, sollten Aktivitätsdaten die negative Werte enthalten, in jedem Fall vor dem Einlesen vorteilhaft transformiert werden.

7.1.8 Aktivitäts-Schwellenwert zur Klassenzuordnung

Ist die Klassenzuordnung Aktiv/Inaktiv extern erfolgt, wird als Kodierung für Aktive der Wert 1 und für Inaktive der Wert 0 erwartet. Fehlende Werte werden durch die Zahl –1 repräsentiert. Liegt die Aktivitätsinformation als Wertebereich vor, muss eine Grenze zwischen Aktiv und Inaktiv festgelegt werden. Dies kann interaktiv unter **Action | Set Activity-Threshold** erfolgen. Es wird ein Histogramm der vorhandenen Werte erstellt. Die rote horizontale Trennlinie kann mit der Maus beliebig nach links oder rechts verschoben werden. Die Werte der drei Comboboxen **Threshold**, **Count left** und **Count right** werden entsprechend angepaßt. Das Feld **Threshold** enthält den aktuellen Schwellenwert und **Count left/right** gibt an, wie die Verbindungen zwischen den Gruppen Aktiv und Inaktiv aufgeteilt werden. Ein Klick auf die **Accept**-Schaltfläche setzt den gewählten Grenzwert und schließt die Dialogbox.

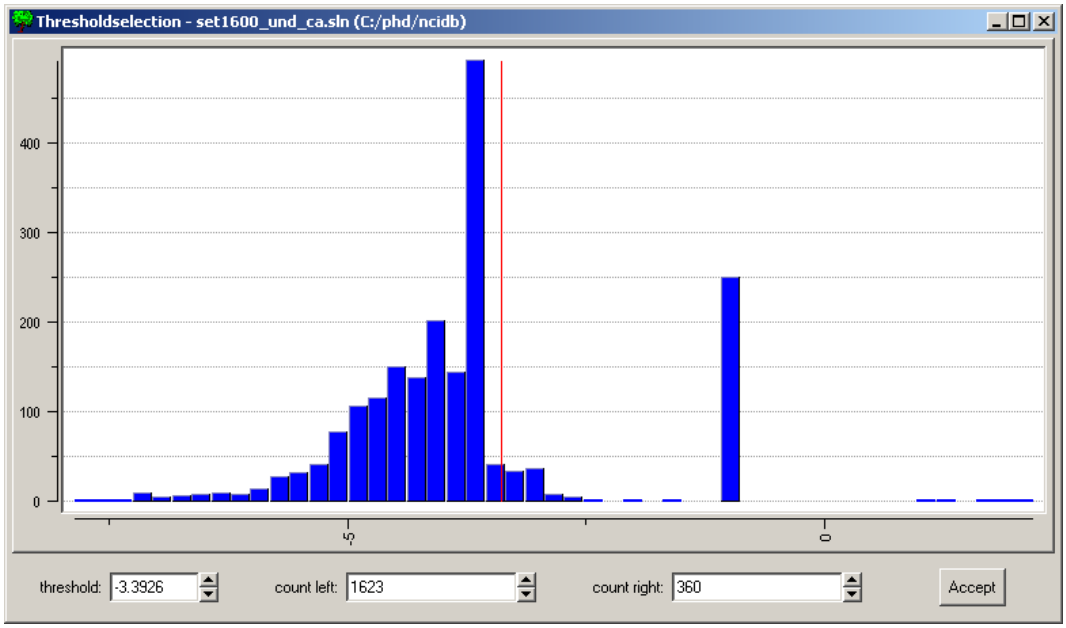


Abb. 7.4: Histogram zur Festlegung des Aktivitäts-Schwellenwertes.

Je nach Datensatz werden die kleinen Werte, z.B. IC_{50}, oder die großen Werte, z.B. Counts von HTS-Rohdaten, bevorzugt. Dazu wird unter **Options | Most Active lowest value** oder **highest value** ausgewählt.

7.2 Templatbetrachtung

7.2.1 Navigation im Strukturbaum

Um die gewünschte Ansicht der Daten bzw. eines Subsets zu erhalten, gibt es folgende Techniken zur Navigation im Baum:

- Der sichtbare Bildausschnitt wird mit den vertikalen und horizontalen Scrollbars oder mit dem Mausrad verschoben.
- In kleinen Schritten kann dieser mit den Pfeiltasten und seitenweise mit PageUp/PageDown und Ctrl-Left/Ctrl-Right bewegt werden.
- Die mittlere Maustaste (beide Maustasten bei einer Zweitastenmaus) kann verwendet werden, um den sichtbaren Bildausschnitt zu verändern. Maustaste drücken und festhalten bewirkt, dass die Zeichenfläche der Bewegung der Maus folgt.
- Die Pfeiltasten in Verbindung mit gedrückter Shift-Taste ermöglichen Bewegungen entsprechend der Baumhierarchie. Im Zentrum der Zeichenfläche wird bei Navigation nach oben der Eltern- (Parent-)Knoten positioniert und bei Navigation nach unten der erste Kind- (Child-)Knoten, bei Bewegung nach links/rechts der linke/rechte Geschwisterknoten. Gibt es keinen entsprechenden Knoten in der Hierarchie, erfolgt keine Neupositionierung.
- Im aktivierten Globalview-Fenster entspricht das grüne Rechteck dem sichtbaren Bereich des Baums. Dieser wird beim Verschieben des Rechtecks entsprechend angepasst. Modifikationen in Baum (Pruning, Collapsing, Expanding) werden zur Zeit aus Geschwindigkeitsgründen nicht automatisch in das Globalview-Fenster übertragen. Dazu ist ein manueller Update (**Edit | Update Globalview**) erforderlich.
- Durch Eingabe der Kommandos „**center_mc <***molcode***>**“ bzw. „**center_cmpd <***regid***>**“ wird der Knoten mit dem gewünschten MolCode bzw. die angegebene Verbindung auf der Zeichenfläche zentriert. Durch Verwendung der Befehle **picker_mc** <*molcode liste*> bzw. **picker_cmpd** <*regid liste*> wird ein zusätzliches Fenster geöffnet, aus dessen Listenfeld die übergebenen MolCodes bzw. RegIds ausgewählt und angesteuert werden können.

Zur Verwendung der Tastenkombinationen muss die Zeichenfläche den Bildschirmfokus (erkennbar an dem roten Rand) haben; dazu ggf. einmal mit der linken Maustaste hineinklicken.

7.2.2 Navigationshilfen für den Gesamtstrukturbaum

Durch Aktivierung von **View | Globalview** wird ein zusätzliches kleines Fenster geöffnet, das die Position des Hauptfensters im Datenraum des gesamten Strukturbaums in verdichteter Form zeigt. Alle Knoten werden durch ein proportional verkleinertes Rechteck repräsentiert. Der sichtbare Ausschnitt des Baums im Hauptfenster wird durch ein grünes Rechteck, den Viewframe, markiert. Wird dieser mit der linken oder mittleren Maustaste verschoben, so wird der sichtbare Bereich des Hauptfensters entsprechend angepasst. Wird der Mauszeiger in dieses Globalview-Fenster bewegt, blinkt der Viewframe mehrmals blau auf, damit er von

den Rechtecken der grünen aktiven Knoten unterschieden werden kann. Das Globalview-Fenster kann beliebig in der Größe verändert werden, der Inhalt wird proportional angepasst. Eine horizontale Vergrößerung entzerrt die Darstellung bei sehr breiten Bäumen und erhöht damit den Überblick.

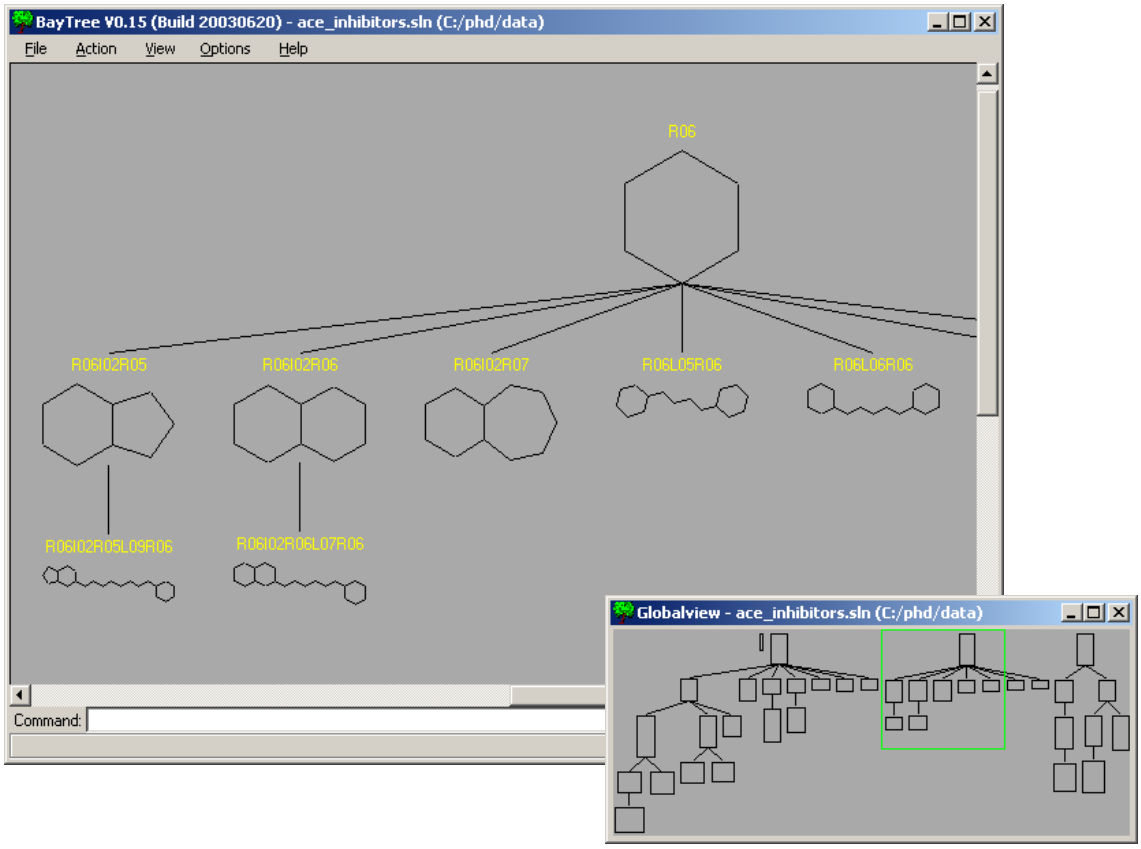


Abb. 7.5: Gegenüberstellung von View und Globalview. Der sichtbare Ausschnitt im großen Fenster entspricht der grünen Begrenzung im kleinen Fenster.

Im MC List-Fenster (**View | MC List**) werden alle MolCodes mit zusätzlichen Knoteninformationen in einer Listbox aufgeführt. Diese sind Belegungsgrad (Occupancy), Wert der aktivsten Verbindung (Highest Act), Mittelwert der Aktivitäten (Average Act) und Anteil der aktiven Verbindungen in Prozent (Percent Act). Die Darstellungsbreite der einzelnen Spalten kann durch Verschieben der Trennlinie mit der linken Maustaste verändert werden.

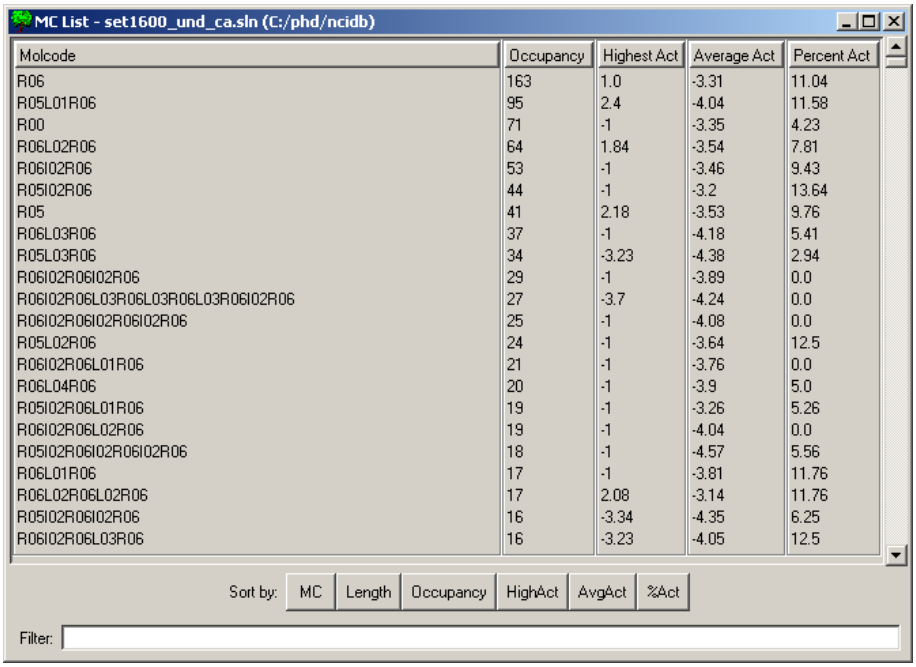


Abb. 7.6: Die Listbox des MC List-Fensters.

Beim Doppelklicken auf eine Zeile wird der Baumknoten mit dem betreffenden MolCode im Hauptfenster zentriert. Durch Anklicken der Spaltenlabel mit der linken Maustaste wird die Tabelle entsprechend der zugehörigen Spalte fallend sortiert; Anklicken mit der rechten Maustaste sortiert aufsteigend. Die Sortierung kann auch durch Anklicken der entsprechenden „**Sort by**"-Schaltfläche ausgelöst werden. Die Schaltfläche **Length** sortiert nach der Länge des MolCode, die ein Maß für die Größe des Framework ist.

Die MolCodes, d.h. der Inhalt der Listbox, können durch Eingabe eines entsprechenden Filterausdrucks vorgefiltert werden. Dazu wird ein Ausdruck mit Wildcard-Zeichen entsprechend der Globbing-Regeln in der Filter-Eingabezeile eingegeben und die Filterung durch Drücken der Return-Taste aktiviert. Ist kein Filterausdruck eingegeben, werden alle MolCodes angezeigt.

Nutzbare Globbing-Zeichen:

***** steht für beliebig viele Zeichen

? steht für ein beliebiges Zeichen

[...] steht für alle aufgeführten Zeichen; Bereichsangabe mit Minuszeichen ist möglich, z.B. [3-9] für alle ganzen Zahlen zwischen 3 und 9

Beispiele:

R0?L0?R0? Zwei Ringe beliebiger Größe, die durch einen Linker beliebiger Länge verknüpft sind.

I01 Alle Spiroverbindungen

7.2.3 Baumlayout-Manipulationen mit dem Kontextmenü

Für jeden Knoten des Baums kann mit der rechten Maustaste ein Kontextmenü aufgerufen werden, das die interaktive Bearbeitung von Baumknoten und Subtrees ermöglicht.

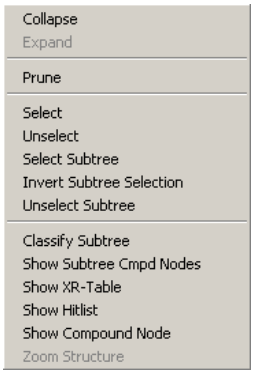


Abb. 7.7: Das Kontextmenü der Knoten.

Aus dem Kontextmenü können folgende Befehle aufgerufen werden:

Collapse	Der Subtree unter dem Knoten wird ausgeblendet und durch ein Plus-Symbol ersetzt.
Expand	Der unter dem Knoten ausgeblendete/kollabierte Subtree wird wieder eingeblendet.
Prune	Der Knoten wird einschließlich des gesamten Subtree gelöscht.
Select	Alle Verbindungen des Knotens werden selektiert, d.h. ins Selektionsset aufgenommen und farblich hervorgehoben.
Unselect	Alle Verbindungen des Knotens werden deselektiert.
Select Subtree	Alle Verbindungen des Subtree werden selektiert und farblich hervorgehoben.
Invert Subtree Selection	Alle Verbindungen, die selektiert waren, werden deselektiert und alle deselektierten werden selektiert.
Unselect Subtree	Alle Verbindungen des Subtree werden deselektiert.
Show Compound Node	Der Cmpdnode mit den Struktur-Depictions der Verbindungen des MolCode/Framework wird erstellt.
Show Subtree Cmpd Nodes	Zum gewählten Subtree werden alle Cmpdnodes eingezeichnet.
Classify Subtree	Alle Verbindungen des Subtree werden als Input für die Aktivitäts-Klassifizierung verwendet. Das Ergebnis wird als Färbung der Classovals an den Strukturen der Cmpdnodes zurückgegeben.
Show XR-Table	Zu allen Verbindungen mit dem Framework des Knotens wird die XR-Table erstellt. (X-Atome = Heteroatome des Framework; R-Gruppen = Substituenten des Framework)
Show Hitlist	Es wird ein Hitlistfenster mit allen Verbindungen des Knotens geöffnet.

Zoom Structure	Die gewählte Struktur wird in einem separaten Fenster vergrößert eingezeichnet und kann durch Verändern der Fenstergröße skaliert werden.

Bei Modifikationen wird das Layout des Baums den Veränderungen angepasst, d.h die Subtrees und Knoten werden neu positioniert.

Beim Doppelklick auf das MolCode-Label eines Framework erscheinen die Namen (RegIds) der zugeordneten Verbindungen im Console-Fenster.

Beim Doppelklick auf die Framework-Nodes werden alle Verbindungen des Knotens selektiert. Falls sie schon selektiert waren, werden sie deselektiert.

7.2.4 Ein- und Ausblenden von Strukturen

Zu jedem MolCode/Framework-Knoten kann als erster Subnode der Cmpd-Node eingeblendet werden. Dieser enthält alle Repräsentanten des Framework. Die Verbindungen werden mit Struktur-Depiction und RegId vertikal mit abnehmendem Aktivitätswert eingezeichnet.

Durch Aktivierung von **View | Compounds** werden alle Strukturknoten des Baums auf einmal eingeblendet bzw. bei Deaktivierung ausgeblendet. Strukturknoten zu einem einzelnen Framework bzw. einem Subtree können über das Node-Kontextmenü mit **Show Compound Node** bzw. **Show Subtree Cmpd Nodes** einblendet werden. Über die Kontext-Funktion **Prune** werden einzelne Strukturknoten wieder gelöscht.

7.2.5 Selektionslisten-Generierung und -Visualisierung

Zur Datenfokussierung ist es hilfreich, nur einen gewünschten Subset des Gesamtdatensatztes für die Detailauswertung zu erstellen. Hierzu wird in BayTree eine Selektionsliste verwaltet. Diese kann interaktiv mit den **Select/Unselect**-Funktionen aus dem Kontextmenü verändert werden oder mit **File | Selection | Load** eingelesen werden. Derartige Listen können von BayTree stammen oder durch ein anderes Programm erzeugt worden sein. Auf diese Art können z.B. Clusterzuordnungen visualisiert werden. Durch **File | Selection | Merge...** werden die Verbindungen der neuen Selektionsliste den aktuell selektierten Verbindungen beigefügt.

Alle weiteren Funktionen zur Handhabung der Selektionsliste finden sich ebenfalls unter **File | Selection**.

Die Namen (RegIds) aller selektierten Strukturen können im Textfenster ausgegeben (**Print to Console**) oder als Liste (ASCII-File) gespeichert werden (**Save as List**). Die Strukturen selber können in einem Hitlist-Fenster angezeigt (**Show Hitlist**) oder als Hitlistfiles (**Save Compounds...**) gespeichert werden. Die Reihenfolge der Entries entspricht jeweils der Selektionsreihenfolge.

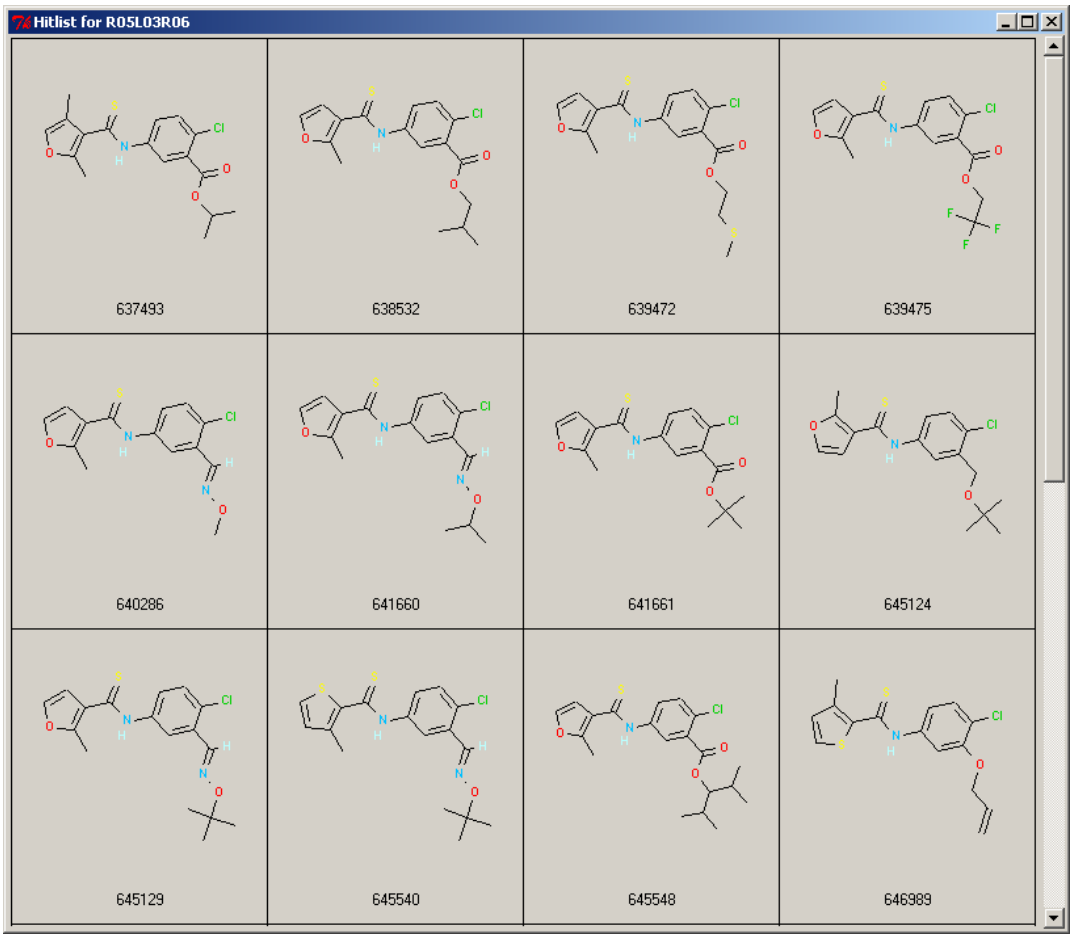


Abb. 7.8: Screenshot des Hitlist-Fensters.

Die abgespeicherten Selektionslisten können auch als Eingabefilefilter verwendet werden, wenn die Konvention in der Wahl des Dateinamens berücksicht wird (vide supra).

Das Dateiformat der Selektionslisten sieht jeweils eine RegId pro Zeile vor. Enthält die Zeile am Anfang ein Doppelkreuz (#), wird sie als Kommentarzeile betrachtet und ignoriert.

7.2.6 Struktur-Eigenschaftsanalyse durch Farbkodierung der Knoten

Aktivitätsfärbung (View | Activity Coloring)

Die Aktivitätsfärbung der Knoten kann durch zwei Einstellungen beeinflusst werden. Durch **Options** | **Coloring Scope** wird der Bereich vorgegeben, aus dem die Aktivitätswerte zur Berechnung der kumulativen Aktiviät verwendet werden. Bei der Einstellung **Node** werden nur die Verbindungen des Knotens selbst verwendet, bei der Einstellung **Tree** werden die Werte aller Subnodes berücksichtigt. Da nur die Leafnodes einen Aktivitätswert enthalten, können dadurch nicht-populierte Frameworknodes entsprechend gefärbt werden. Der Baum wird von unten nach oben eingefärbt, d.h. die Farbe des Elternknotens wird aus der der Kindknoten berechnet. Leafnodes mit unbekannter Aktivität bleiben in jedem Fall unberücksichtigt.

Der verwendete Farbbereich wird unter **Options | Coloring Mode** eingestellt. Es gibt zwei Modi:

- **Tri-state mode**: Wenn nur eine aktiv/inaktiv-Information zur Verfügung steht, gibt es die Farben rot = inaktiv, grün = aktiv, gelb = „neutral“. Die Knoten mit unbekannter Aktivität bleiben schwarz. Ein Knoten ist nur aktiv, wenn alle Knoten des Coloring Scope aktiv sind, bzw. inaktiv, wenn alle Knoten inaktiv sind. Andernfalls ist er „neutral“ und erhält die Farbe gelb, da dieses Gerüst keinen direkten Rückschluss auf Aktivität oder Inaktivität zulässt.
- **Colorscale mode**: Wenn eine quantitative Aktivitätsinformation zur Verfügung steht, wird die Aktivität auf einer Farbskala von rot = inaktiv über gelb nach grün = aktiv dargestellt. Der Aktivitätswert und damit der Farbwert der Knoten wird aus dem Mittelwert der Aktivitäten der Knoten des Coloring Scope berechnet.

7.2.7 Zusammensetzung der Knoten

Mittels **View | Composition Piechart** wird zu jedem Knoten, der ein Aktivitätsmischprofil besitzt, ein Tortendiagramm eingeblendet, dessen Segmentgröße den Anteilen der aktiven (grün), inaktiven (rot) und unbekannten (schwarz) Verbindungen des Knotens entspricht (siehe Abb. 7.9). Das simultane Auftreten aktiver und inaktiver Derivate eines Templates suggeriert, dass Pharmakophor-Verletzungen im Rahmen der SAR aufgedeckt wurden. Es können aber auch falsch positive/negative Ergebnisse oder grenzwertige Ergebnisse vorliegen.

7.2.8 Belegungsgrad der Knoten/MolCodes

View | Occupancy

Unterhalb eines jeden Knotens werden zwei durch „/“ getrennte Zahlen ausgegeben (siehe Abb. 7.9). Die erste Zahl entspricht der Anzahl der Verbindungen des Framework, die als konkrete Repräsentanten mit Heteroatomen und Substituenten den zugehörigen MolCode besitzen. Die zweite Zahl gibt die Gesamtzahl der unterhalb dieses Knotens vorhandenen Verbindungen an, d.h. sie enthalten dieses Framework als Teilgerüst.

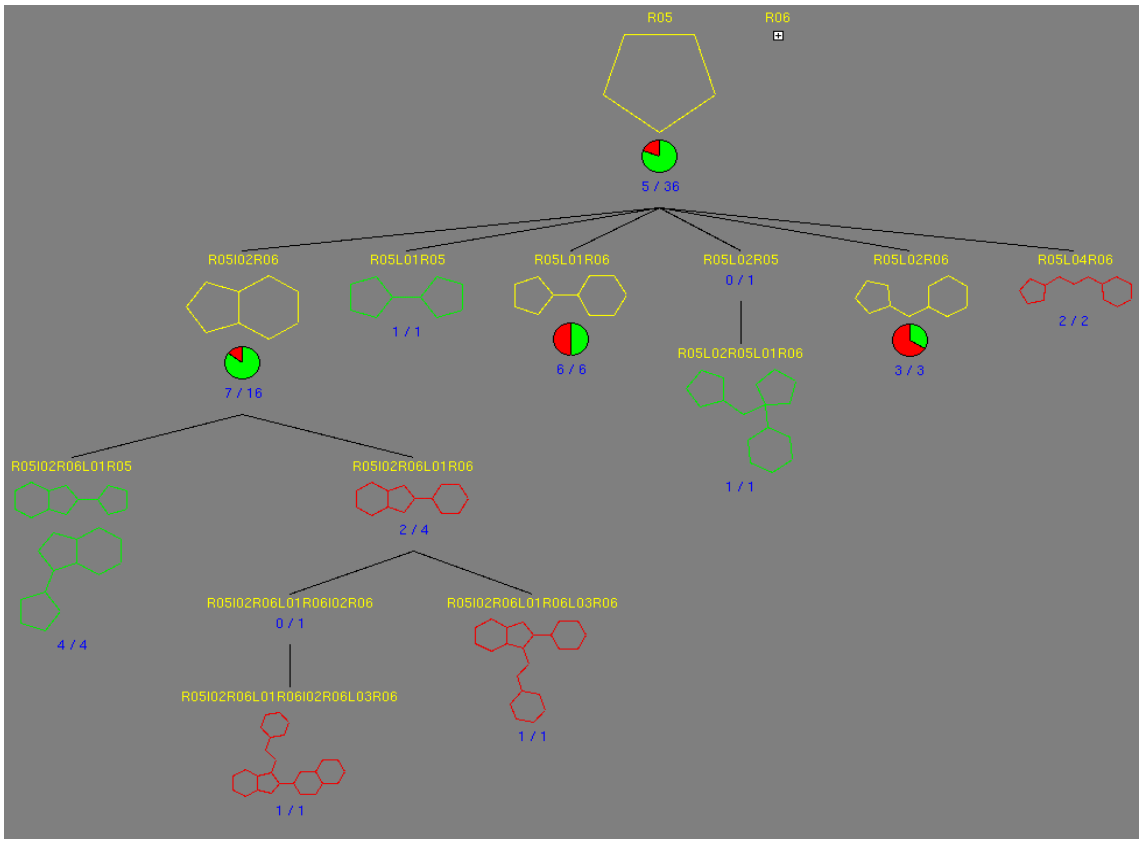


Abb. 7.9: Screenshot mit eingeblendeter Occupancy-Information und Aktivitätskolorierung am Beispiel des R05-Unterbaums des Sedativa-Datensatzes.

Belegungsgradfärbung

Der konkrete Zahlenwert der Belegung eines MolCode ist in der MC-Liste aufgeführt und das zugehörige Framework kann von dort durch Anklicken angesteuert werden. Zusätzlich können Hotspots im Belegungsgrad auch visuell durch Occupancy Coloring erkannt werden (siehe Abb. 6.9 auf Seite 98). Die Belegungsgradfärbung wird durch das Kommando **colorbyocc** aktiviert. Im Legendenwindow in Abb. 7.10 sind die vordefinierten Bereiche und die zugeordneten Farben aufgeführt.

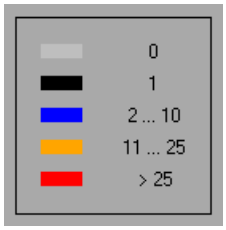


Abb. 7.10: Legende für die Farbcodes des Occupancy Coloring.

Es gelten die folgenden Farbcodes: grau für nichtbelegte Knoten, schwarz für singulär belegte Knoten, blau für Knoten mit 2 bis 10 Repräsentanten, orange für 11 bis 25 Repräsentanten und rot für Knoten, die mit mindestens 26 Strukturen belegt sind.

7.3 Strukturbetrachtung

7.3.1 Prädiktive Aktivitätsklassifizierung durch LDA

Die Aktivitätsklassifizierung ermöglicht es, für Verbindungen ohne bekannte Aktivitätsinformation eine Zuordnung als Vorhersage vorzunehmen oder die bekannte Aktivitätsinformation von Verbindungen mit dem Diskriminanzmodell zu überprüfen, um Ausreißer (falsch positive/falsch negative) zu identifizieren. Als mathematisches Klassifizierungsverfahren wird die Lineare Diskriminanzanalyse verwendet, als Deskriptoren werden die über Bindungsdipolmomente gewichteten spektralen Momente von Estrada benutzt. Alternative Deskriptoren können per Kommando (siehe Abschnitt 7.1.5) eingelesen werden. Vor der gewünschten Klassifizierung muss ein Diskriminanzmodell mit Hilfe eines Trainings-Datensatzes, für den die Klassifizierung bekannt ist, erstellt worden sein. Da die Qualität des Diskriminanzmodells wesentlich von der Auswahl der Trainingsverbindungen und der Eliminierung fehlerhafter Daten abhängt, bietet sich BayTree zur Bearbeitung dieser Teilfrage an.

Die Klassifizierung und die Auswahl der dafür verwendeten Daten können auf zwei verschiedene Arten vorgenommen werden: erstens durch Selektieren der gewünschten Verbindungen, Knoten oder Subtrees im gesamten Baum und durch Aufrufen von **Action | Classify Selection** oder zweitens durch **Classify | Subtree** aus dem Node-Kontextmenü, wodurch automatisch alle Verbindungen des Subtree verwendet werden.

Die Diskriminanzgleichung (das Modell) wird unter Verwendung der Verbindungen erstellt, für die eine Aktivitätsinformation vorliegt; alle Verbindungen, für die keine Aktivitätsinformation existiert, werden vorhergesagt, d.h. vom Modell klassifiziert.

Das Ergebnis der Klassifizierung, d.h. die Zuordnung der Verbindungen zu den Klassen aktiv bzw. inaktiv, wird durch einen entsprechend gefärbten Kreis rechts oben neben der Verbindung eingezeichnet. Dazu muss der Cmpd-Node vorher eingeblendet worden sein. Der Zahlenwert gibt die Klassifizierungswahrscheinlichkeit für die Zuordnung zu der jeweiligen Gruppe an.

Zu jedem Klassifizierungslauf wird ein Classification Results-Fenster (siehe Abb. 7.11) geöffnet. Diese sind in der Titelzeile fortlaufend durchnummeriert. Alle Resultate der Registerkarten können durch Eingabe von Ctrl-s in eine Datei abgespeichert werden; für die Histogramme wird ein PostScript-File erstellt.

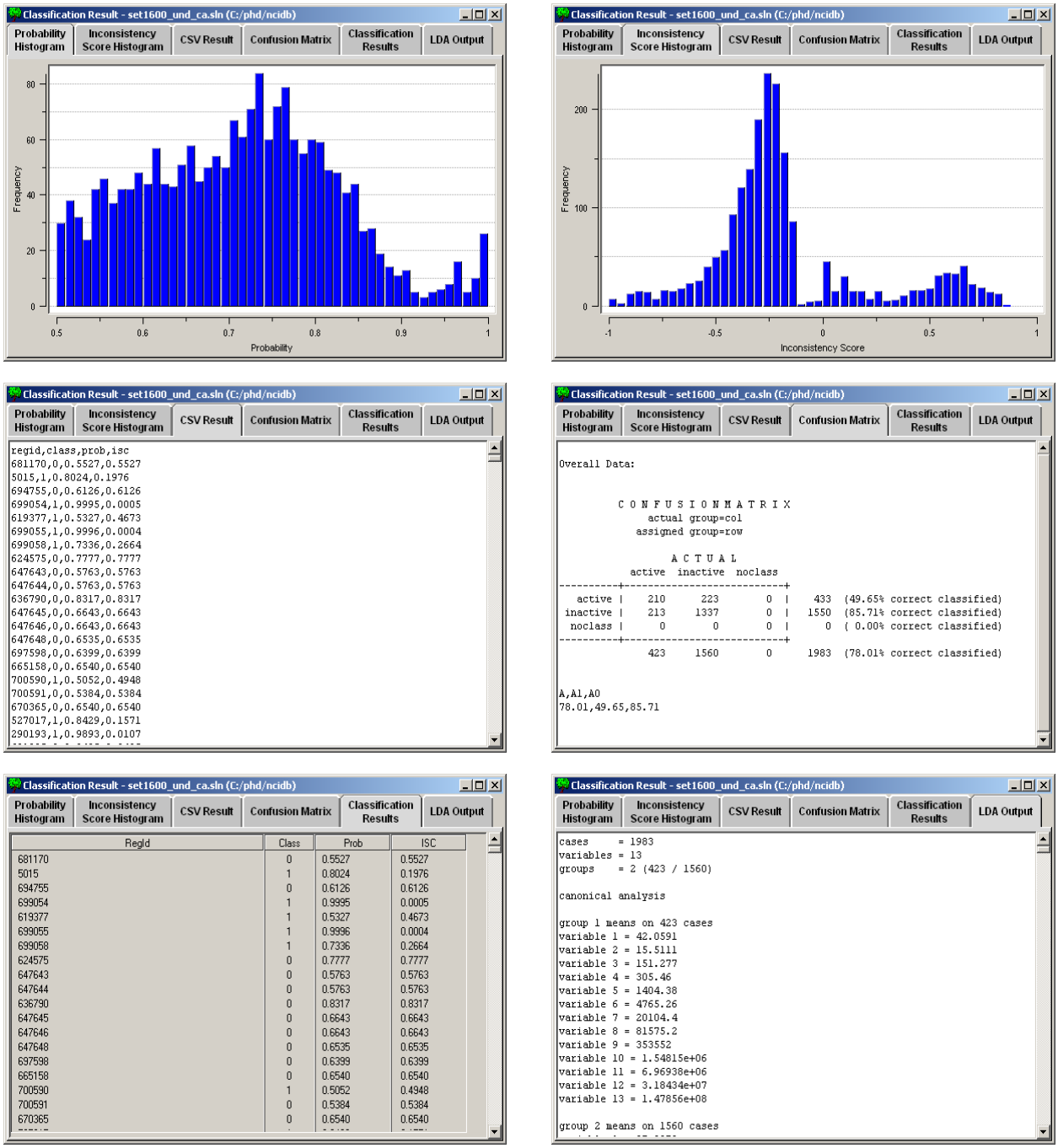


Abb. 7.11: Screenshots der Registerkarten des Classification Results-Fenster.

Folgende Registerkarten sind verfügbar:

1. Confusion Matrix

Diese Karteikarte enthält die Confusion Matrix der Klassifizierung aller verwendeten Verbindungen zu den Gruppen Aktiv (active), Inaktiv (inactive) und Unbekannt (noclass) . In den Spalten der Matrix sind die tatsächlichen Klassen und in den Zeilen der Matrix die vorhergesagten Klassen aufgeführt. Außerhalb der Matrix sind zusätzlich die Zeilen- und Spaltensummen ausgegeben. Die Diagonalelemente enthalten alle korrekt klassifizierten Verbindungen, die falsch klassifizierten sind auf den Außerdiagonalelementen zu finden.

Aus den Matrixelementen können verschiedene Bewertungsscores für die Leistungsfähigkeit eines Classifiers berechnet werden[249]. Sehr häufig wird das enrichment ratio verwendet, es wird definiert als der Quotient aus den Verbindungen, die aktiv vorhergesagt wurden und aktiv sind, und dem Anteil aktiver Verbindungen im Gesamtdatensatz. Je höher der Zahlenwert des enrichment ratio ist, umso besser ist das Modell.

$$\text{enrichment ratio} = \frac{N \cdot N_{TP}}{N_A \cdot (N_{TP} + N_{FP})}$$

N: Gesamtzahl der Verbindungen
N_A: Gesamtzahl der aktiven Verbindungen
N_{TP}: Anzahl der richtig positiv (true-positive) vorhergesagten Verbindungen (Aktive die korrekt klassifiziert wurden)
N_{FP}: Anzahl der falsch positiv (false-positive) vorhergesagten Verbindugen (Inaktive die falsch klassifiziert wurden)

2. Classification Results

Diese Registerkarte enthält die Ergebnisse der Klassifizierung in Form einer Tabelle. Die Zeilen der Tabelle können durch Anklicken der Spaltenüberschriften mit der linken bzw. rechten Maustaste aufsteigend bzw. fallend sortiert werden. Durch Doppelklicken mit der Maus auf eine Zeile wird die Struktur der entsprechenden Verbindung im TST zentriert. Die Tabelle enthält die vier Spalten RegId, Class, Prob. und ISC. RegId ist der Name der Verbindung, Die Class-Spalte enthält die vorhergesagte Klasse 1=aktiv, 0=inaktiv und –1 = nicht klassifiziert. Die posterior probability der Klassenzuordnung ist in der Spalte Prob aufgeführt. Die letzte Spalte (IS) enthält den Inconsistency Score der Verbindung (siehe Inconsistency Score Histogram in Abb. 7.11). Sind die Daten nach den Werten dieser Spalte sortiert, können die Ausreisser leicht am Anfang und am Ende identifiziert werden.

3. CSV Results

Um die Ergebnisse der Klassifizierung außerhalb von BayTree in anderen Programmen weiterverwenden zu können, werden die Resultate in dieser Registerkarte im CSV-Format aufgelistet. Sie können zusätzlich durch Ausschneiden und Einfügen über die Zwischenablage übertragen werden. Die Tabelle enthält vier Spalten: RegId der Verbindung, Binärwert für Aktivität (1=aktiv, 0=inaktiv), Wahrscheinlichkeitswert der Klassenzuordnung und IS für Inconsistency Score (vide infra). Ist letzterer nicht berechenbar, wird er auf 10 gesetzt.

4. Inconsistency Score Histogram

Diese Registerkarte enhält das Histogramm des Inconsistency Score (IS) der Klassifizierung.

Der Inconsistency Score (IS)[250] einer Verbindung misst die Übereinstimmung zwischen der Aktivitätsprädiktion durch ein Modell und der tatsächlichen Aktivitätsklasse. Er ist definiert als die Abweichung zwischen der tatsächlichen Klasse (aktiv/inaktiv) und der Wahrscheinlichkeit, aktiv zu sein:

$IS_i = C_i - p_i(\text{aktiv})$

C_i: tatsächliche Aktivitätsklasse der Verbindung i, wobei 1 für aktiv und 0 für inaktiv steht
p_i(aktiv): Prädiktionswahrscheinlichkeit für Verbindung i, aktiv zu sein

Der IS nimmt Werte zwischen –1 und +1 an. Je näher er an 0 ist, mit desto höherer Wahrscheinlichkeit ist die Verbindung korrekt klassifiziert. Werte nahe bei –1 weisen auf falsch negative Klassifizierungen und Werte nahe bei +1 weisen auf falsch positive Klassifizierungen hin.

5. Probability Histogram

Diese Registerkarte enthält das Histogramm der Klassifizierungswahrscheinlichkeiten (posterior probabilities). Ein Klassifizierungsmodell ist umso verlässlicher, je höher der Anteil an Vorhersagen mit hoher Wahrscheinlichkeit ist.

6. LDA-Output

Diese Registerkarte enthält zur Kontrolle die Ausgabe des Klassifizierungsprogramms. Die Koeffizienten der Diskriminanzgleichung sind dort unter den Überschriften „canonical vectors“ bzw. „discriminant functions“ zu finden. Die Gruppe 1 enthält die Aktiven, Gruppe 2 die Inaktiven und Gruppe 3 – wenn vorhanden – Verbindungen, die zwar nicht zur Modellerstellung verwendet, aber klassifiziert wurden.

7.3.2 Kommandos zur Erstellung der Multimodell-Klassifikationen

Mit dem Kommando **scan_all_submodels** <*mc_searchspec_or_lst*> werden zu allen Subtrees aller Ebenen folgende Informationen zusammengestellt und ausgegeben: Belegungsgrad, Anzahl Aktiver, Anzahl Inaktiver, Anteil Aktiver, Anteil Inaktiver und die Klassifizierungsgenauigkeit. Letztere ist 0, falls kein Modell erzeugt werden konnte. Diese Zusammenstellung ist eine vollständige Charakterisierung aller theoretisch möglichen Untermodelle des TST, ausgehend vom virtuellen Wurzelknoten, der den Gesamtdatensatz enthält, bis hin zu den Blättern des Baums, die jeweils alle Repräsentanten eines spezifischen Framework enthalten. Die Daten werden im CSV-Format im Textfenster ausgegeben und stehen damit für eine Analyse ausserhalb von BayTree zur Verfügung. Um den Suchbereich von vorneherein einzuschränken, kann der Befehl mit einem optionalen Parameter aufgerufen werden. Dabei kann es sich entweder um eine Liste mit beliebigen MolCodes handeln oder um eine MolCode-Suchspezifikation mit Stern und Fragezeichen als Wildcard-Symbole, z.B. R05*, wenn nur die Untermodelle des R05-Subtree geprüft werden sollen.

Innerhalb von BayTree werden geeignete Untermodelle mit dem Kommando **get_possible_submodels** <*mincmpds*> <*minact_percent*> <*minaccuracy_percent*> ausgewählt. Zur Steuerung stehen drei Kriterien zur Verfügung, die der Reihe nach überprüft werden: <*mincmpds*> ist die erforderliche Mindestanzahl an Verbindungen im Untermodell, <*minact_percent*> ist der erforderliche Mindestanteil von Aktiven (in Prozent) und <*minaccuracy_percent*> ist die mindeste geforderte Klassifizierungsgenauigkeit. Wird der letzte Parameter auf Null gesetzt, wird keine Klassifikationsrechnung gestartet und die

Ausgabe erfolgt ohne merkliche Verzögerung. Dadurch können die ersten beiden Parameter iterativ den Wünschen des Benutzers angepasst werden. Das Kommando liefert als Rückgabewert eine Liste mit den Root-MolCodes, deren Subtrees allen Kriterien entsprechen und die damit als Untermodell in Frage kommen.

Die Klassifikation unter Verwendung von Untermodellen wird mit dem Kommando **classify_using_submodels** <*submodel_mcs*> <*regids*> <*force_to_unknowns*> aufgerufen. Als erster Parameter <*submodel_mcs*> wird die Liste mit den Root-MolCodes für die zu erzeugenden Untermodelle angegeben. Wird der zweite Parameter <*regids*> nicht angegeben, wird der Gesamtdatensatz verwendet, anderenfalls nur die Verbindungen, deren RegIds in der <*regids*>-Liste aufgeführt wurden. Analog zum Aufruf per GUI werden die Modelle basierend auf den Verbindungen erzeugt, für die Aktiv/Inaktiv-Informationen vorliegen. Alle anderen Verbindungen werden mit den Modellen vorhergesagt. Zur Bewertung der Modelle, z.B. im Rahmen einer Trainigs/Testset-Aufteilung, können Verbindungen, für die die Aktiv/Inaktiv-Informationen zur Verfügung stehen, aus der Modellbildung herausgenommen werden und nur vorhergesagt werden. Zu diesem Zweck kann die Liste der RegIds der Verbindungen des Testset als dritter Parameter <*force_to_unknowns*> übergeben werden. Im Classification Results Window werden dann für den Testset zusätzliche Informationen ausgegeben.

Zur zufälligen Aufteilung des Datensatzes, z.B. in einen Trainigsset und Testset, steht das Kommando **get_random_subset** <*fraction*> <*regids*> zur Verfügung. Es liefert als Rückgabeliste den zufällig ausgewählten Anteil <*fraction*> der RegIds <*regids*> zurück. Wird das Kommando ohne Parameter aufgerufen, werden die Defaultwerte 0.25 für <*fraction*> und Gesamtdatensatz für <*regids*> verwendet.

Zum Speichern und Laden der RegId-Listen im Selectionlist-Format stehen die beiden Kommandos **write_selectionlistfile** <*regids*> <*fname*> und **read_selectionlistfile** <*fname*> zur Verfügung. <*fname*> ist jeweils durch den gewünschten Dateinamen zu ersetzen.

Ein möglicher Aufruf für eine Multimodell-Klassifikation könnte beispielsweise kombiniert in folgender Form in der Kommandozeile erfolgen:

classify_using_submodels [**get_possible_submodels** 50 30 70] {} [**get_random_subset**]

Die Befehlsfolge erzeugt ein Multimodell unter Einbeziehung aller Untermodelle, die aus 50 oder mehr Verbindungen bestehen, von denen mindestens 30% aktiv sind und deren Klassifizierungsgenauigkeit oberhalb von 70% liegt. Das Gesamtmodell wird aus drei Viertel der Daten (Trainigsset) erzeugt und mit einem Viertel der Daten (Testset) validiert. Die leere Liste {} steht als Platzhalter für den zweiten Parameter und hat zur Folge, dass der Gesamtdatensatz verwendet wird. Bei dem Beispiel bleibt einfachheitshalber unberücksichtigt, dass ein Teil der Verbindungen, der für die Untermodellauswahl herangezogen worden ist, später im Trainingsset nicht mehr zur Verfügung steht.

7.3.3 XR-Table-Generierung: Dekonvolution der Templatdekoration

Die XR-Tabelle zu einem MolCode/Framework wird aus dem Node-Kontextmenü aufgerufen. Sie enthält eine Auflistung aller Modifikationen des Framework. Die X-Atome sind die Heteroatome des Framework, die R-Gruppen seine Substituenten.

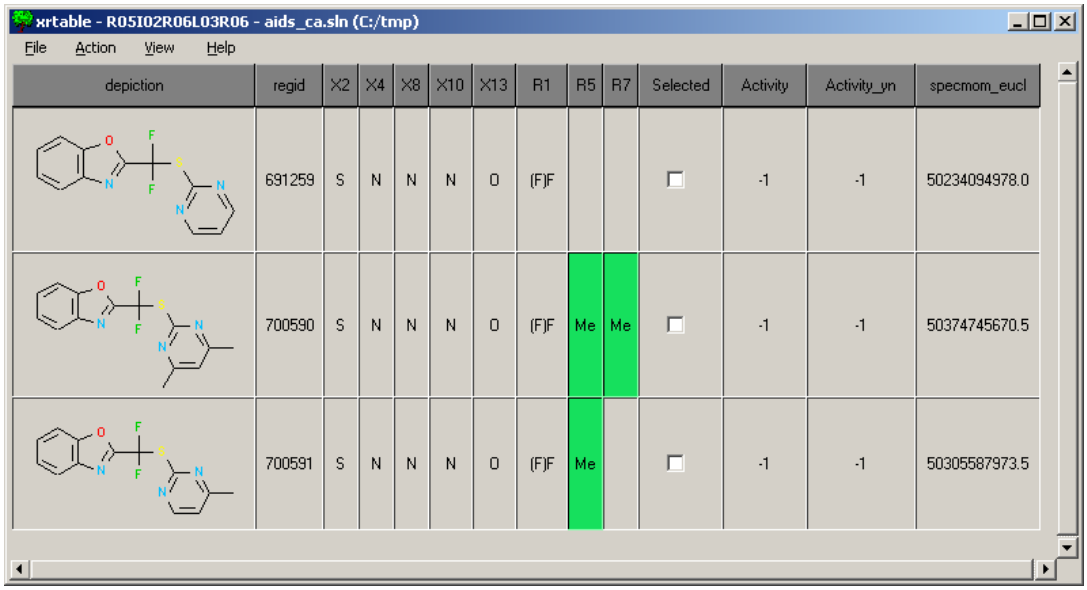


Abb. 7.12: Screenshot der XR-Table. Der Wert –1 in der Aktivitätsspalte steht für unbekannt (missing value).

Gibt es mehrere Verknüpfungsvarianten zu einem MolCode, so werden diese in verschiedenen Tabellen aufgelistet, welche dann über den eingeblendeten „**fw *X***"-Registerkartenreiter ausgewählt werden.

Durch eine Heuristik wird versucht, die Heteroatome bzw. Substituenten an äquivalenten Positionen des Gerüsts mit maximaler Übereinstimmung in den gleichen Spalten aufzuführen, wobei die Depictions nicht verändert werden. Als Referenz wird die Verbindung mit der größten Anzahl an Modifikationen, d.h. Heteroatomen und Substituenten, verwendet. Die Zuordnung der Positionen aller anderen Verbindungen erfolgt durch Minimierung eines Misfitscores, der Vorhandensein bzw. Größenunterschied der Modifikatoren bewertet[251].

Die Tabelle kann unter **File | Export | As HTML** im HTML-Format exportiert werden. Alle Depictions werden als separate Grafikfiles im GIF-Format geschrieben und eingebunden. Die reine Textinformation der Tabelle kann im CSV-Format exportiert werden. Die Strukturen der ersten Spalte werden in diesem Fall als SMILES-String ausgegeben. Diese können beispielsweise mit Isis for Excel wieder in eine Struktur-Depiction umgewandelt werden.

Zur Unterstützung der Analyse können Spalten, die nur identische Werte enthalten, mittels **Action | Hide One Value Columns** ausgeblendet werden. Die Zellen von Spalten, die mehrere Gruppen von identischen Werten enthalten, können mit **Show | Color By Values** gruppenweise farbig unterlegt werden.

Durch Doppelklicken der Spaltenüberschriften (Header) wird die Tabelle entsprechend der Werte der Spalte umsortiert. Erfolgt das Doppelklicken mit der linken Maustaste, wird aufsteigend sortiert; Doppelklicken mit der rechten Maustaste sortiert absteigend.

Alle Zellen können durch Verschieben der Begrenzungen mit der rechten Maustaste in der Größe verändert werden.

Neben den obligatorischen Spalten für Depiction und RegId sind die in folgender Tabelle aufgeführten Spalten grundsätzlich vorhanden:

Selected	Checkbox zum Selektieren bzw. Deselektieren der Verbindung
Activity	eingelesener Aktivitätswert (missing value = -1)
Activity_jn	Aktiv/Inaktiv-Zuordnung (active=100, inactive = 0, unknown = -1)
Specmom_eucl	„chemischer Abstand“ der Verbindung zu seinem Framework, berechnet als Euklidische Distanz unter Verwendung der spektralen Momente als Deskriptoren.

7.3.4 Exportmöglichkeiten für Bäume

Der Baum, wie er in der Zeichenfläche angezeigt ist, kann unter **File | Export** zu Dokumentations- und Reportingzwecken in verschiedenen Formaten exportiert und gespeichert werden: als GIF Picture, in dem die gesamte scrollbare Zeichenfläche als Bitmap im GIF-Format gespeichert wird, oder als Encapsulated PostScript, mit dem ein EPS-File für den Ausdruck erstellt wird. Bei großen Bäumen kann ein posterized Postscript erzeugt werden, d.h. der Baum wird auf mehrere DIN A4-Seiten verteilt, die nach dem Ausdruck zusammengeklebt werden können. Als weiteres Format steht TkPaint-PIC zur Verfügung. TkPaint[252] ist ein plattformunabhängiges Zeichenprogramm, mit dem die Baumgrafik nachbearbeitet werden kann. Es können z.B. Anmerkungen oder Pfeile ergänzt werden. Mit der vom Autor modifizierten TkPaint-Version ist unter Windows die Konvertierung ins WMF (Windows MetaFile) bzw. EMF (Enhanced MetaFile)-Format möglich. Dieser Zwischenschritt ist zur Erstellung von EMF-Dateien ausgehend von unter UNIX gespeicherten PIC-Files erforderlich. Mit der Windows-Version von BayTree kann direkt im EMF-Format exportiert werden. Der Baum als Vektorgrafik kann ohne Auflösungverlust und skalierbar in Windows-Applikationen (MS Word, MS PowerPoint) integriert werden.

Die aus BayTree gespeicherten PIC-Files enthalten zusätzlich eine Startup-Funktion, die es ermöglicht, den gespeicherten Baum nur mittels eines Tcl/Tk-Interpreters („wish“) anzuzeigen. Dies ist auf jeder Plattform (Windows, Unix) möglich. Aufruf unter Unix: **wish** <*file.pic*>, unter Window: Das File <*file.pic*> per Drag und Drop auf das bei der Tcl/Tk-Installation erzeugte Wish-Icon fallen lassen.

8 Zusammenfassung

Im Rahmen der Arbeit wurden das BayTree-Programm und die zugrundeliegenden neuen Verfahren zur Auswertung von biologischen Screeningdaten entwickelt. Unterstützte Betriebssysteme sind Irix und Windows. Screeningdaten werden in grossem Umfang in der industriellen Pharmaforschung generiert, wenn bei der automatisierten Hochdurchsatztestung chemische Verbindungen auf ihre Wirkung auf ein biologisches Testsystems untersucht werden. Bislang existiert keine einheitliche oder generell akzeptierte Lösung zur schnellen Beantwortung der bei der Auswertung auftretenden Fragen.

In BayTree wird, um einen Überblick über die vorhandenen Template zu erhalten, zu den Strukturen eines Screeningdatensatzes ein hierarchischer topologischer Baum der Frameworks erzeugt. Die Anordnung wird eindeutig und standardisiert durch einen MolCode vorgegeben, der, priorisiert durch ein Regelwerk, die das Gerüst aufbauenden Strukturmerkmale enthält. Der MolCode wird – zu einem Satz von Priorisierungsregeln – einmalig ermittelt und ermöglicht die Positionierung der Strukturen im Topological Structure Tree. Seine Erzeugung ist unabhängig von anderen Strukturen und ermöglicht daher – ohne performanceverringernde Paarvergleiche – ein lineares Laufzeitverhalten. Das Gerüst der Template ist für die räumliche Anordnung der pharmakophoren Elemente zuständig. Daher werden alle Strukturen eines MolCodes als konkret vorliegende Repräsentanten unterhalb des Frameworks angeordnet und dieses als virtuelle Referenzstruktur betrachtet. Daher sind im Prinzip alle Ergebnisse sichtbar, es existiert aber ein strukturbasiertes „Verdichtungsprinzip“, das strukturverwandte Verbindungen durch repräsentative Vertreter exemplarisch beschreibt. Da als Ordnungskriterium wesentliche Anteile der Struktur selber verwendet werden, wird eine „strukturnahe“ hohe Interpretierbarkeit erreicht. Das beschriebene Regelwerk erhält die in Optimierungsreihen und vielen kombinatorischen Bibliotheken innewohnende Systematik: Zentrale Strukturfragmente bleiben konstant und Veränderungen finden vor allem in der Peripherie statt. Die Darstellung unterschiedlicher chemischer Strukturdaten wird standardisiert, so dass targetbezogene Informationen aus unterschiedlichen Quellen inhaltlich nach gleichen Kriterien zusammengeführt und ausgewertet werden können.

Zur Systematisierung der pharmakophoren Elemente eines Frameworks steht die XR-Tabelle zur Verfügung, die unter Berücksichtigung von Automorphismen die Heteroatome und Substituenten der Verbindungen eines MolCodes in den Spalten einer Tabelle anordnet.

Desweiteren wurde zur Identifizierung von Ausreißern und falsch gemessenen Verbindungen ein Klassifizierungsverfahren implementiert. Dieses basiert auf der linearen Diskriminanzanalyse und verwendet als 2D-Strukturdeskriptoren Estradas bindungsdipolgewichtete spektralen Momente. Damit ist es wahlweise möglich, ein Gesamtmodell zu einem Datensatz zu erstellen oder interaktiv verschiedene lokale Modelle zu erzeugen. Diese auf einer vorhergehenden Gruppierung durch den MolCode basierenden lokalen Modelle sind im allgemeinen robuster und einfacher zu interpretieren. Zusätzlich besteht die Möglichkeit die Vorhersagegenauigkeit der aktiven Verbindungen zu erhöhen, indem mehrere lokale Untermodelle zu einem Multimodell zusammengefasst werden. Die Auswahl der zu verwendenden Untermodelle kann teilautomatisiert anhand von berechneten Kenngrößen erfolgen oder individuell gesteuert werden. Die Validierung aller erzeugten Modelle wird mittels Kreuzvalidierung (leave-one-out) oder anhand eines Testdatensatzes vorgenommen.

Das BayTree-Programm fasst die beschriebenen Verfahren unter einer komfortablen graphischen Benutzeroberfläche zusammen und bietet dem Anwender umfangreiche

Unterstützung bei der „Navigation“ durch den Datensatz. Zur Verfügung stehen: Schematisches Übersichtsfenster, Berechnungsverfahren für statistische Kenndaten und unterschiedliche Färbungen zur Identifizierung von Hotspots.

Zum Dokumentieren der Resultate gibt es umfangreiche Exportmöglichkeiten: für den TST, die Hitliste und Struktur-Depictions die Formate GIF, PostScript und EMF, für die XR-Table die Formate HTML und CSV und für die Klassifizierungsergebnisse die Formate CSV und PostScript.

Die Anwendung des Programms und der Verfahren wurde exemplarisch an einem Subset des öffentlichen NCI-Datensatzen gezeigt. Die Ergebnisse stimmen mit anderen bekannten Verfahren und publizierten Arbeiten überein.

Die Auswertung eines Screening-Datensatzes ist derzeit noch nicht vollständig automatisierbar. Sie bedarf eines Spezialisten auf dem Gebiet. BayTree unterstützt diese zeitaufwendigen Arbeit durch die nahtlose Integration aller für die Auswertung relevanter Arbeitschritte wie Strukturimport, Verknüpfung der Strukturen mit den biologischen Daten, Festlegung eines Aktivitätsgrenzwertes, Gruppierung und Visualisierung der Daten, Auswahl und Selektion von Teilmengen und Strukturprototypen und die sehr flexible Erstellung und Validierung von Klassifikationsmodellen.

Die Ergebnisse der Arbeit sind teilweise durch das Patent WO02074035 geschützt und werden in die proprietäre PIX-Plattform der Business Unit Pharma der Bayer AG integriert.

9 Abkürzungsverzeichnis

ACD	Available Chemicals Directory
AIDS	Acquired Immune Deficiency Syndrome
PBC	Perception-based Classification
CACTVS	Chemical Algorithm Construction, Threading and Verification System
CAS	Chemical Abstract Service
CMC	Comprehensive Medicinal Chemistry
CSV	Comma Separated Values
EMF	Enhanced MetaFile
FCS	Fluorescence Correlation Spectroscopy
GIF	Graphics Interchange Format
GUI	Graphical User Interface
HIV	Human Immunodeficiency Virus
HIS	Hue Saturation Intensity
HTS	High Troughput Screening
IND	Investigational New Drug
IP	Intellectual Property
IS	Inconsistency Score
LDA	Lineare Diskriminanzanalyse
MCS	Maximum Common Substructure
MDB	Molecular Database
MDDR	MDL Drug Data Report
MOE	Molecular Operating Environment
NCI	National Cancer Institute
NMR	Nuclear Magnetic Resonance
NN	Nearest Neighbour
PCA	Principal Component Analysis
QSAR	Quantitative Structure Activity Relationship
QSPR	Quantitative Structure Property Relationship
RGB	Red Green Blue
SAR	Structure Activity Relationship
SCA	Scaffold-based Classification Approach
SDF	Structure Definition File
SDG	Structure Diagram Generation
SLN	Sybyl Line Notation
SOM	Self-Organizing Map
SQL	Structured Query Language
SSSR	Smallest Set of Smallest Rings
SVL	Scientific Vector Language
TCC	Topological Cluster Center
TCL	Tool Command Language
TSC	Topological Sequence Code
TSF	Topological Structure Forest
TSP	Topological Sequence Path
TST	Topological Structure Tree
WMF	Windows MetaFile

10 Glossar

Adjazenzmatrix (Adjacency Matrix): Synonym zu Atom Connection Matrix. Sie enthält die Informationen über die Verknüpfungen der Knoten eines Graphen. Die adjacency matrix **A** = **A**(G) eines Graphen G mit n Knoten ist die quadratische symmetrische n×n Matrix, deren Elemente in der i-ten Zeile und j-ten Spalte wie folgt definiert sind:

$A_{ij} = 1$	wenn $i \neq j$ und $e_{ij} \in E(G)$, d.h. Knoten i ist mit Knoten j verknüpft
$A_{ij} = 0$	wenn $i = j$ oder $e_{ij} \notin E(G)$, d.h. Knoten i ist nicht mit Knoten j verknüpft

ADME (Absorption Distribution Metabolism Excretion). Die ADME-Eigenschaften beschreiben das pharmakokinetische und pharmakodynamische Profil einer Verbindung. ADMET schließt die Toxizität ein.

Aktive: Verbindungen, die die gewünschte Wirkung auf ein biologisches System zeigen, werden als aktive Verbindungen (vereinfacht Aktive oder Hits) bezeichnet.

Assay (Biologischer Test): Ein Assay ermöglicht es, die molekulare Wechselwirkung zwischen einem Target und einem Liganden sichtbar und messbar zu machen.

Cactvs Chemistry Toolkit: (Chemical Algorithm Construction, Threading and Verification System) Tcl/Tk-basiertes Programmier-Toolkit von W. D. Ihlenfeld/J. Gasteiger des CCC der Uni Erlangen (siehe Abschnitt 5.3.2).

Core: Zentrales chemisches Grundgerüst, vor allem für kombinatorische Chemie, das aus beispielhaften Verbindungen durch retrosynthetische Schnitte generiert werden kann.

CSV-File (Comma-Separated Values): Dateiextension und Beschreibung eines Formats zum Austausch von Tabellendaten. Die im Klartext (ASCII-Format) vorliegenden einzelnen Werte einer Zeile werden durch Kommata getrennt. Wenn als Trennzeichen Semikolons oder Tabulatoren zum Einsatz kommen, steht das Komma als Dezimaltrennzeichen zur Verfügung. Das Format wurde mit dem ersten Tabellenkalkulationsprogramm VisiCalc eingeführt und enthält zusätzlich einige Quotingregeln im Umgang mit Textfeldern, die bei numerischen Daten keine Rolle spielen. Eine offizielle Spezifikation existiert nicht.

dbtranslate: Kommandozeilentool von Tripos, das zur Konvertierung zwischen verschiedenen Strukturdateiformaten (SDF, SLN, Smiles und MOL2) dient. Es ist Bestandteil von Sybyl und Unity. Bei der Konvertierung von Strukturformaten, die keine 2D-Koordinaten enthalten, werden diese automatisch berechnet (siehe Abschnitt 5.2.4).

Dekoration: Substituenten und Heteroatome, die nicht Frameworkatome einer chemischen Verbindung sind.

Depiction: Strukturzeichnung bzw. Abbildung des molekularen Graphen einer chemischen Struktur.

Deskriptor: Zahlenvektor oder Komponente eines solchen, der je nach Definition spezielle Eigenschaften eines Moleküls numerisch kodiert und damit das Molekül charakterisiert (siehe Abschnitt 2.3).

Drug: Wirkstoff eines zugelassenen Medikaments, Synonym zu Wirkstoffmolekül und Ligand.

falsch positiv/falsch negativ: Bei einer Klassifizierung oder Messung wird das Ergebnis als falsch positiv bezeichnet, wenn das Verfahren sie als positiv klassifiziert („predicted"), sie aber tatsächlich („observed") negativ sind. Falsch negativ bedeutet, sie werden negativ bewertet, sind aber tatsächlich positiv.

observed	predicted	
	0	1
0	korrekt	falsch positiv
1	falsch negativ	korrekt

Fingerprint: Abfolge von Bits mit dem Wert 0 oder 1, die das Vorhandensein oder die Abwesenheit von Strukturbestandteilen kodieren, die in der Fingerprintdefinition festgelegt sind (siehe Abschnitt 2.3.1.2). Ein solcher Binärstring kann besonders schnell von Computern verarbeitet werden.

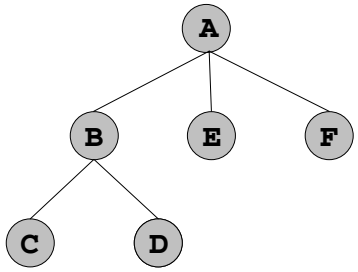
Framework: Von Bemis/Murcko[121] eingeführte Bezeichnung für das allcarba-Gerüst ohne Berücksichtigung der Bindungsordnung und der Substituenten einer Struktur (siehe Abbildung unter Scaffold).

Hit: Chemisch definierte und charakterisierte Verbindung, die beim HTS in einem biologischen Test als aktiv aufgetreten ist.

HTS (High Throughput Screening): Methode zum Testen von Substanzbibliotheken (üblicherweise mehrere 10 000 bis 100 000 Verbindungen) unter Verwendung von roboterisierten Assays (siehe Abschnitt 1.4).

Inaktive: Verbindungen, die keine oder eine zu geringe Wirkung auf ein biologisches System haben. Inaktive Verbindungen werden vereinfacht auch als Inaktive bezeichnet. Gegensatz zu Aktive.

Knotenbezeichnungen:



A	root node	Wurzelknoten des Baums
B	parent node	Elternknoten von C und D

C, D	subnodes, child nodes	Tochterknoten von B
C, D, E, F	leaves, leaf nodes	Blätter des Baums
ABC		Pfad von Knoten A zu C
B, E, F	siblings	Geschwisterknoten

MolCode: Synonym zu TSC (Topological Sequence Code). Er kodiert zusätzlich zum TSP weitere Informationen einer Verbindung, wie Konnektivitäten (K-Modul) und Substituenten (S-Modul). Siehe Abschnitt 5.1.8.

Modul: Durch einen Grossbuchstaben eingeleitete Abschnitte des MolCode. In der Regel entsprechen sie einer topologischen Komponente.

Occupancy/Belegungsgrad: Anzahl der Repräsentanten, die ein spezielles Framework enthalten.

Pharmakophor: Abstrahierte relative räumliche Anordnung von chemischen Interaktionspunkten (Akzeptorgruppen, Donorgruppen, hydrophobe oder ionische Gruppen), die für die chemische Wechselwirkung mit dem Target oder die räumliche Ligandform und damit indirekt für die Aktivität eines Liganden erforderlich sind.

Prüfpräparatelager: Synonym zu compound pool, Repository, Testsubstanzen. Umfangreiche Sammlung von Verbindungen, die als Feststoff oder Lösung zum Screening bereitgehalten werden

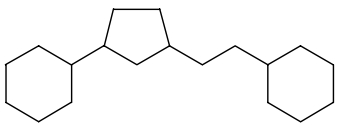
QSAR (Quantitative Structure Activity Relationship): Methode zu quantitativen Erklärung des Einflusses von Strukturbestandteilen (Substrukturen, Substituenten) auf die biologische Aktivität.

RegId: Identifizierungscode oder Namen, über den in einer Datenbank oder einem Datensatz eindeutig auf eine Struktur zugegriffen werden kann.

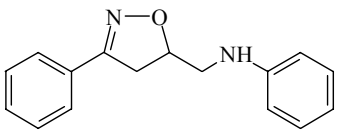
SAR (Structure Activity Relationship): Methode zur qualitativen Erklärung des Einflusses von Strukturbestandteilen (Substrukturen, Substituenten) auf die biologische Aktivität.

Scaffold: Chemisches Grundgerüst einer Struktur ohne Seitenketten bzw. Substituenten, aber mit evtl. vorhandenen Heteroatomen (siehe folgende Abbildung).

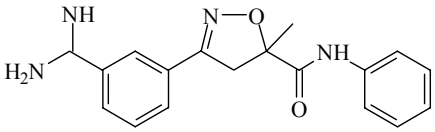
Framework



Scaffold



komplettes Molekül



Screening: Allgemeiner Begriff für die in vitro-Reihentestung chemischer Verbindungen.

SD-File (Structure Data File): Es enthält Strukturinformationen und zugehörige Daten für eine oder mehrere Verbindungen. Das Format wurde vom Datenbankhersteller MDL Information Systems Inc. entwickelt. Eine detailierte Beschreibung des Formats ist auf der MDL Website zu finden[253]. Es stellt den Standard zur Speicherung von chemischen Verbindungen und Daten dar und wird von fast jedem Programm unterstützt.

SLN (Sybyl Line Notation): Kompakte Zeilennotation, in erster Linie zur Beschreibung der molekularen Graphen (Atome und Bindungen) von chemischen Strukturen. SLN wurde ausgehend von SMILES[254] von der Firma Tripos entwickelt. Die Spezifikation sieht vor, dass beliebige zusätzliche Informationen (Strukturnamen, Koordinaten etc.) in Form von Attributen enthalten sein können. Diese bestehen aus einem Namen und dem zugehörigen Wert. Die genaue Beschreibung des Formats ist im SLN Manual von Tripos[255] bzw. in *J. Chem. Inf. Comput. Sci.*[256] zu finden.

SSSR (Smallest Set of Smallest Rings): Repräsentative Untermenge der Menge aller Ringe einer Verbindung, die kleinste Teilmenge aller kleinsten Ringe, die in einer chemischen Struktur beobachtbar sind. Sie entspricht in der Regel der „intuitiv“ gezählten Anzahl an Ringen (siehe Abschnitt 5.2.2).

Target: Makromolekulares Zielmolekül (in der Regel ein Protein), welches kausal mit der molekularen bzw. chemischen Ursache einer Krankheit verknüpft ist bzw. für die kausale Therapie genutzt werden kann, z.B. indem es durch ein kleineres Ligandmolekül in seiner Aktivität moduliert wird.

TCC (Topological Cluster Center): Referenzknoten für alle Strukturen mit identischem Framework (siehe Abschnitt 5.1.4).

Templat: Je nach Kontext das Framework, das Scaffold oder ein größeres Fragment einer Struktur.

topologisch: Geometrische Anordnung von Punkten (speziell Atomen) im Raum. Im Kontext der molekularen Graphen: Verknüpfung der Atome, die in der connection table einer speziellen Adjazenzmatrix kodiert werden.

TSC (Topological Sequence Code): Synonym zu MolCode, wegen der Konkordanz der Bezeichnungen beibehalten.

TSP (Topological Sequence Path): Zwingend vorhandener erster Teil des MolCode, der die Abfolge der priorisierten topologischen Unterklassen enthält und zur Erzeugung des entsprechenden Ausschnitts der Hierarchie des TST verwendet wird.

TST/TSF (Topological Structure Tree bzw. Topological Structure Forest): Bezeichnungen für den in BayTree erzeugten hierarischen topologischen Baum. Sie werden nicht klar getrennt, da der TSF nach Ergänzung eines nicht eingezeichneten virtuellen Wurzelknotens, der allen TST gemeinsam ist, selbst in einen TST übergeht.

11 Literatur

[1] Johnson, M. A.; Maggiora, G. M. (Eds.) *Concepts and Applications of Molecular Similarity*, Wiley: New York, **1990**.

[2] Ehrlich, P. Über den jetzigen Stand der Chemotherapie. *Ber. Dt. Chem. Ges.* **1909**, *22*, 17-47.

[3] Milne, G. W. Pharmacophore and Drug Discovery. *Encyclopedia of Computational Chemistry*, v. Ragué-Schleyer, P. (Editor-in-Chief), Wiley: New York, **1999**, 2046-2056.

[4] Frobel, K.; Krämer, Th. Kombinatorische Synthese. *Chemie in unserer Zeit* **1996**, *30*, 270-285. Terrett, N. K. *Combinatorial Chemistry*, Oxford University Press: Oxford, **1998**.

[5] Lander, E. S.; Linton, L. M.; Birren, B.; Nusbaum, C.; Zody, M. C.; Baldwin, J.; Devon, K.; Dewar, K. et al. Initial sequence and analysis of the human genome. *Nature* **2001**, *409*, 860-921. Venter, J. C.; Adams, M. D.; Myers, E. W.; Li, P. W.; Mural, R. J.; Sutton, G. G.; Smith, H. O.; Yandell, M.; Evans, C. A. et al. The sequence of the human genome. *Science* **2001**, *291*, 1304-1351.

[6] Hopkins, A. L.; Groom, C. R. The druggable genome. *Nature Reviews Drug Discovery* **2002**, *1*, 727-730.

[7] Millennium Pharmaceuticals Inc., Granta Park, Great Abington, Cambridge CB1 6ET, United Kingdom, http://www.mlnm.com/rd/engine/platform.asp.

[8] Lengauer, T. (Ed.) *Bioinformatics – From Genomes to Drugs* Methods and Principles in Medicinal Chemistry Volume 14, Mannhold, R.; Kubinyi, H.; Timmerman, H. (Eds.), Wiley-VCH: New York, **2002**.

[9] Amersham Biosciences, Amersham Place, Little Chalfont, Buckinghamshire HP7 9NA, England, *Scintillation Proximity Assay (SPA) Development* Technical Presentation Assay_development.pdf von http://www.amershambiosciences.com. Lutz, M.; Kenakin, T. *Quantitative Molecular Pharmacology and Informatics in Drug Discovery,* John Wiley & Sons: New York, **1999**.

[10] Ogenstad, S.; Hastizs, C. Drowning in Information But Thirsting for Knowledge. *Applied Clinical Trials* **2000***, 9*, 46 (und Folgende mit Werbung & Fortsetzung ab 81).

[11] Fayyad, U.; Grinstein, G.; Wierse, A. *Information Visualization in Data Mining and Knowledge Discovery*, Morgan Kaufmann Publishers, **2001**.

[12] Bolger, R. High-throughput screening: new frontiers for the 21st century. *Drug Discovery Today* **1999***, 4*, 251-253.

[13] Auer, M.; Moore, K. J.; Meyer-Almes, F. J.; Guenther, R.; Pope, A. J. et al. Fluorescence correlation spectroscopy: lead discovery by miniaturized HTS. *Drug Discovery Today* **1998***, 3*, 457-465.

[14] Teague, S. J.; Davis, A. M.; Leeson, P. D.; Oprea T. Design kombinatorischer Leitstruktur-Bibliotheken. *Angew. Chem.* **1999**, *111*, 3962-3967.

[15] Evans, B. E. et al. Method for Drug Discovery: Development of Potent, Selective, Orally Effective Cholecystokinin Antagonists. *J. Med. Chem.* **1988**, *31*, 2235-2246.

[16] Mason, J. S.; Morize, I.; Menard, P. R.; Cheney, D. L.; Hulme, C. et al. New 4-Point Pharmacophore Method for Molecular Similarity and Diversity Applications: Overview of the Method and Applications, Including a Novel Approach to the Design of Combinatorial Libraries Containing Privileged Substructures. *J. Med. Chem.* **1999**, *42*, 3251-3264.

[17] Kunz, R. W. *Molecular Modelling für Anwender*, Teubner: Stuttgart, **1997**, Kapitel 2.1: Kraftfeldprogramme, 86-133.

[18] Jensen, F. *Introduction to Computational Chemistry*, John Wiley & Sons: New York, **1999**, Chapter 2: Force Field Methods, 6-51.

[19] Patani, G. A.; LaVoie, E. J. Bioisosterism: A rational approach in drug design. *Chem. Rev.* **1996**, *96*, 3147-3176.

[20] Oshiro, C. M.; Kuntz, I. D.; Kuegtel R. M. Molecular Docking and Structure-based Design. *Encyclopedia of Computational Chemistry*, v. Ragué-Schleyer, P. (Editor-in-Chief), Wiley: New York, **1999**, 1606-1603.

[21] Muegge, I.; Rarey, M. Small Molecule Docking and Scoring. *Reviews in Computational Chemistry Volume 17* Edited by Lipkowitz, K. B.; Boyd, D. B. Wiley-VCH: New York, **2001**, 1-60.

[22] Lemmen, C.; Lengauer, T.; Klebe, G. FLEXS: A Method for Fast Flexible Ligand Superposition. *J. Med. Chem.* **1998**, *41*, 4502-4520.

[23] Miller, M. D.; Sheridan, R. P.; Kearsley, S. K. SQ: A Program for Rapidly Producing Pharmacophorically Relevent Molecular Superpositions. *J. Med. Chem.* **1999**, *42*, 1505-1514.

[24] Labute, P.; Williams, C.; Feher, M.; Sourial, E.; Schmidt, J. M. Flexible Alignment of Small Molecules. *J. Med. Chem.* **2001**, *44*, 1483-1490.

[25] Wang, T.; Zhou, J. 3DFS: 3D Flexible Searching System for Lead Discovery – New Version 1.2. *J. Mol. Model.* **1999**, *5*, 231-251.

[26] Walters, W. P.; Stahl, M. T.; Murcko, M. A. Virtual Screening – an overview. *Drug Discovery Today* **1998**, *3*, 160-178.

[27] Rarey, M.; Kramer, B.; Lengauer, T.; Klebe, G. A Fast Flexible Docking Method using an Incremental Construction Algorithm. *Journal of Molecular Biology* **1996**, *261*, 470-489. Rarey, M.; Wefing, S.; Lengauer, T. Placement of medium-sized molecular fragments into active sites of proteins. *Journal of Computer-Aided Molecular Design* **1996**, *10*, 41-54.

[28] Dean, P. M. (Ed.) *Molecular Similarity in Drug Design*, Chapman and Hall: Glasgow, **1994**.

[29] Hodgkin, E. E.; Richards, W. G. Molecular similarity based on electrostatic potential and electric field. *Int. J. Quantum Chem., Quantum Biol. Symp.* **1987***, 14*, 105-110.

[30] Hauswald, M. *Structural Alignment of Isofunctional Molecules*, Dissertation FU Berlin: Berlin, **1998**.

[31] Labute, P. A widely applicable set of descriptors. *J. Mol. Graph. Model.* **2000**, *18*, 464-477.

[32] Cho, S. J.; Hermsmeier, M. A. Genetic Algorithm Guided Selection: Variable Selection and Subset Selection. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 927-936.

[33] Whitley, D. C.; Ford, M. G.; Livingstone, D. J. Unsupervised Forward Selection: A Method for Eliminating Redundant Variables. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 1160-1168. UFS Programm ist erhältlich von http://www.cmd.port.ac.uk.

[34] Lipinski, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv. Drug. Deliv. Rev.* **1997**, *23*, 3-25.

[35] Todeschini, R.; Consonni, V. *Handbook of Molecular Descriptors*, Methods and Principles in Medicinal Chemistry Volume 11, Mannhold, R.; Kubinyi, H.; Timmerman, H. (Edts.), Wiley-VCH:New York, **2000**.

[36] Molconn-Z Software Package for Molecular Topology Analysis Version 4.00, eduSoft LC, PO Box 1811, Ashland, VA, 23005 USA, http://www.edusoft-lc.com/molconn/manuals/400/.

[37] Lin, A. QuaSAR-Descriptor. *J. Chem. Computing Group.* http://www.chemcomp.com/Journal_of_CCG-/Features/descr.htm. Bestandteil von MOE (Molecular Operating Environment), Version 2001.01, Chemical Computing Group Inc., 1255 University Street, Montreal, Quebec, Canada, H3B 3X3.

[38] Cruciani, G.; Pastor, M.; Guba, W. VolSurf: A New Tool for Pharmacokinetic Optimization of Lead Compounds. *Eur. J. Pharm. Sci.* **2000**, *11 (Suppl. 2)*, 29-39. Cruciani, G.; Crivori, P.; Carrupt, P.-A.; Testa, B. Molecular Fields in Quantitative Structure-Permeation Relationships: The VolSurf Approach. *J. Mol. Struct. (Theochem)* **2000**, *503*, 17-30.

[39] Todeschini, R.; Consonni, V.; Mauri, A.; Pavan, M. *DRAGON 2003 TALETE* srl, Milano Chemometrics and QSAR Research Group, http://www.disat.unimib.it/chm/Dragon.htm.

[40] Burden, F. R. A Chemically Intuitive Molecular Index Based on the Eigenvalues of a Modified Adjacency Matrix. *Quant. Struct.-Act. Relat.* **1997**, *16*, 309-314.

[41] *DiverseSolutions V4.0.5*, University of Texas, Austin; vertrieben von Tripos Inc., St. Louis, MO 63144.

[42] Hall, L. H.; Kier, L. B.; Brown, B. B. Molecular Similarity Based on Novel Atom-Type Electrotopological State Indices. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 1074-1080.

[43] Nilakantan, R.; Bauman, N.; Dixon, J. S.; Venkataraghavan, R. Topological Torsion: A New Molecular Descriptor for SAR Applications. Comparison with Other Descriptors. *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 82-85.

[44] Carhart, R. E.; Smith, D. H.; Venkataraghavan, R. Atom Pairs as Molecular Features in Structure-Activity Studies: Definition and Applications. *J. Chem. Inf. Comput. Sci.* **1985**, *25*, 64-73.

[45] Flower, D. R. On the Properties of Bit String-Based Measures of Chemical Similarity. *J. Chem. Inf. Comput. Sci.* **1998***, 38*, 379-386.

[46] Martin, Y. C.; Kofron, J. L.; Traphagen, L. M. Do Structurally Similar Molecules Have Similar Biological Activity? *J. Med. Chem.* **2002**, *45*, 4350-4358.

[47] *SYBYL 6.8 Ligand-Based Design Manual – HQSAR* October **2001**, Tripos Inc. St. Louis, MO 63144. Lowis, D. HQSAR – A New, Highly Predictive QSAR Technique. *Tripos Technical Notes* **1997**, *1*, No. 5.

[48] MDL Information Systems Inc., 14600 Catalina Street, San Leandro, CA 94577.

[49] James, C.; Weininger, D.; Delany J. *Daylight Theory Manual Version 4.82* Juni **2003**, Daylight Chemical Information Systems Inc., Santa Fe, NM 87501, 6. Fingerprints – Screening and Similarity. http://www.daylight.com/dayhtml/doc/theory/theory.finger.html.

[50] *UNITY Reference Guide Version 4.3* October **2001**, Tripos Inc., St. Louis, MO 63144, Chapter 7.5 Screen Files, 60-76.

[51] McGregor, M. J.; Muskal, S. M. Pharmacophore Fingerprinting. 1. Application to QSAR and Focused Library Design. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 569-574.

[52] Henrion, R.; Henrion, G. *Multivariate Datenanalyse*, Springer: Berlin, Heidelberg, **1995**, Kapitel 7.10 Alternative Abstandsmaße, 227-228.

[53] Mahalanobis, P. C. On the generalized distance in statistics. *Proc. Nat. Inst. Sci. India* **1936***, 2*, 49-55.

[54] Brown, R. D.; Martin, Y. C. Use of Structure-Activity Data To Compare Structure-Based Clustering Methods and Descriptors for Use in Compound Selection. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 572-584.

[55] Gower, J. C. Measures of similarity, dissimilarity and distance. *Encyclopedia of statistical sciences* Volume 5, Kotz, S.; Johnson, N. L. (Eds.) John Wiley & Sons: New York, **1985**, 397-405.

[56] *Molecular Diversity Manual Version 6.8 – Selector* October **2001**, Tripos Inc., St. Louis, MO 63144, Chapter 3.2.4 Similarity Coefficients, 53-55.

[57] Willett, P.; Winterman, V. A. Comparison of Some Measures for the Determination of Intermolecular Structural Similarity. Measures of Intermolecular Structural Similarity. *Quant. Struct.-Act. Relat.* **1986**, *5*, 18-25.

[58] Brown, R. D.; Martin, Y. C. Use of Structure-Activity Data To Compare Structure-Based Clustering Methods and Descriptors for Use in Compound Selection. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 572-584.

[59] Chen, X.; Reynolds, C. H. Performance of Similarity Measures in 2D Fragment-Based Similarity Searching: Comparison of Structural Descriptors and Similarity Coefficients. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 1407-1414.

[60] Dixon, S. L.; Koehler, R. T. The Hidden Component of Size in Two-Dimensional Fragment Descriptors: Side Effects on Sampling in Bioactive Libraries. *J. Med. Chem.* **1999**, *42*, 2887-2900.

[61] Levenshtein, V. I. Binary codes capable of correcting deletions, insertions and reversals. *Doklady Akademii Nauk SSS.* **1965**, *163(4)*, 845-848.

[62] Gilleland, M. *Levenshtein Distance in Three Flavors*, Merriam Park Software http://www.merriampark-.com/ld.htm.

[63] Needleman, S. B.; Wunsch, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* **1970**, *48*, 443-453.

[64] Barnard, J. M.; Downs, G. M. Clustering of Chemical Structures on the Basis of Two-Dimensional Similarity Measures. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 644-649.

[65] Murtagh, F. Clustering in Massive Data Sets. *Proceedings of Chemical Data Analysis in the Large*, Italy **2000**, 28-51.

[66] Downs, G. C571, Indiana University. http://www.indiana.edu/~cheminfo/C571/c571_Barnard7.ppt.

[67] Wild, D. J.; Blankley, C. J. Comparison of 2D Fingerprint Types and Hierarchy Level Selection Methods for Structural Grouping Using Ward's Clustering. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 155-162.

[68] Kelley, L. A.; Gardner, S. P.; Sutcliffe, M. J. An automated approach for clustering an ensemble of NMR-derived protein structures into conformationally-related subfamilies. *Protein Eng.* **1996**, *9*, 1063-1065.

[69] Guénoche-Verfahren aus dem BCI Clustering Package.

[70] *Ligand-Based Design Manual Version 6.8 – QSAR* October **2001**, Tripos Inc., St. Louis, MO 63144, Chapter 4.5.7 Hierarchical Cluster Analysis und 4.5.8 Non-Hierarchical Algorithms, 184-193.

[71] Weininger, D.; Delany J. *Daylight Clustering Manual Version 4.82* Juni **2003**, Daylight Chemical Information Systems Inc., Santa Fe, NM 87501, http://www.daylight.com/dayhtml/doc/cluster/index.html.

[72] BCI Clustering Package Versions 2.5 & 3.0, Barnard Chemical Information Ltd., 46 Uppegate Road, Sheffield, S6 6BX UK, barnard@bc1.demon.co.uk.

[73] Murtagh, F. Multidimensional clustering algorithms. *COMPSTAT Lectures* **1985**, *4*, Physica-Verlag: Vienna.

[74] *Molecular Diversity Manual Version 6.8 – Selector* October **2001**, Tripos Inc., St. Louis, MO 63144, Chapter 3. Selector Theory: Reciprocal Nearest Neighbor Clustering, 62-64.

[75] Agrafiotis, D. K.; Lobanov, V. S. An Efficient Implementation of Distance-Based Diversity Measures Based on k-d Trees. *J. Chem. Inf. Comput. Sci.* **1998**, *39*, 51-58.

[76] Murphy, M.; Skiena, S. *A study of data structures for orthogonal range and nearest neighbor queries in high dimensional spaces*, CSE 523/524 Masters's Project, Department of Computer Science, State University of New Your at Stony Brook.

[77] Wards, J. H. Hierarchical Grouping to Optimize an Objective Function. *J. Am. Statistical Assoc.* **1963**, *58*, 236-244.

[78] Henrion, R.; Henrion, G. *Multivariate Datenanalyse*, Springer: Berlin, Heidelberg, **1995**.

[79] Jarvis, R. A.; Patrick, E. A. Clustering Using a Similarity Measure Based on Shared Near Neighbors. *IEEE Transactions on Computers* **1973**, *C-22*, 1025-1034.

[80] Menard, P. R.; Lewis, R. A.; Mason, J. S. Rational Screening Set Design and Compound Selection: Cascaded Clustering. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 497-505.

[81] Hartigan, J. A.; Wong, M. A. A K-Means Clustering Algorithm. *Applied Statistics* **1979**, *28*, 100-108.

[82] MacQueen, J. Some Methods for Classification and Analysis of Multivariate Observations. *Proc. 5th Berkeley Symp. on Math., Stat. and Prob.* **1967**, *1*, 281-297.

[83] Daszykowski, M.; Walczak, B.; Massart, D. L. On the optimal partitioning of data with K-means, Growing K-means, Neural Gas and Growing Neural Gas. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 1378-1389.

[84] Taylor, R. Simulation Analysis of Experimental Design Strategies for Screening Random Compounds as Potential New Drugs and Agrochemicals. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 59-67.

[85] Butina, D. Unsupervised Data Base Clustering Based on Daylight's Fingerprint and Tanimoto Similarity: A Fast and Automated Way to Cluster Small and Large Data Sets. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 747-750.

[86] Wall, L.; Christansen, T.; Schwartz, R. L. *Programmieren mit Perl*, O'Reilly: Köln, **1997**.

[87] Martin, Y.; Kofron, J.; Traphagen, L. Do Structurally Similar Molecules Have Similar Biological Activity? *J. Med. Chem.* **2002**, *45*, 4350-4358.

[88] Kohonen, T. The self-organizing map. *Proceedings of the IEEE* **1990**, *78(9)*, 1464-1480.

[89] Kohonen, T.; Hynninen, J.; Kangas J.; Laaksonne J. *SOM_PAK: The Self-Organizing Map Program Package* Version 3.1 (7. April 1995) Technical Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science, FIN-02150 Espoo, Finland, **1996** http://www.cis.hut.fi/research-/som_pak/.

[90] Zell, A.; Mache, N. et al. *SNNS 4.2 Stuttgart Neural Network Simulator*, http://www-ra.informatik.uni-tuebingen.de/SNNS/.

[91] Zupan, J.; Gasteiger, J. *Neural Networks for Chemists – An Introdction*, VCH: Weinheim, **1993**.

[92] Bauknecht, H.; Zell, A.; Bayer, H.; Levi, P.; Wagener, M. et al. Locating Biologically Active Compounds in Medium-Sized Heterogeneous Datasets by Topological Autocorrelation Vectors: Dopamine and Benzodiazepine Agonists. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 1205-1213.

[93] Daszykowski, M.; Walczak, B.; Massart, D. L. Looking for Natural Patterns in Analytical Data. 2. Tracing Local Density with OPTICS. *J. Chem. Inf. Comput Sci.* **2002**, *42*, 500-507.

[94] Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining*, Portland, OR, AAA Press, **1996**, 226-231. Ankerst, M.; Breunig, M.; Kriegel, H.-P.; Sander J. OPTICS: Ordering Points To Identify the Clustering Structure. *Proc. ACM SIGMOD '99*, Int. Conf. on Management of Data, Philadelphia, PA, **1999**, 49-60.

[95] Guha, S.; Rastogi, R.; Shim, K. CURE: An Efficient Clustering Algorithm for Large Databases. *Information Systems* **2001**, *26*, 35-58.

[96] ISIS for Excel, MDL Information Systems Inc., 14600 Catalina Street; San Leandro, CA 94577.

[97] Accord for Excel, Accelrys Inc., San Diego, CA.

[98] Hecker, H. *Auswahl, Anwendung und Interpretation statistischer Tests – Eine kurze Einführung.* Medizinische Hochschule Hannover, Institut für Biometrie, November **1997**. http://www.mh-hannover.de-/institut/biometrie/Scripte/Tests/swtest7.pdf.

[99] Ertl, P. *Enhancement of hit rate in HTS by using fragment-based virtual screening techniques.* UK QSAR and Chemoinformatics Group Meeting, October **2001**.

[100] Xemistry Clusterviewer, clusterviewer.tk aus cactvstools-Windows_NT-3.23.tar.gz, November **1998**.

[101] Johnson, P. ClusterView, verfügbar über http://www.daylight.com/support/contrib/clusterview/.

[102] Spotfire Inc., Mölndal/Schweden, http://www.spotfire.com und http://www.visualdatamining.com.

[103] Shneiderman, B. Dynamic queries for visual information seeking. *IEEE Software* **1994**, *11*, 70-77.

[104] Ahlberg, C.; Shneiderman, B. Visual information seeking: Tight coupling of dynamic query filters with starfield displays. *Proceedings of the CHI'94 Conference on Human Factors in Computing Systems,* ACM: New York, **1994**, 313-317.

[105] Hesselberg, P. User Interface Programming: Range Slider. *Windows Developer Magazine* **2001**, *12*, http://www.wd-mag.com/wdm/articles/2001/0112/ und http://www.wd-mag.com/documents/wdj0112e/.

[106] *BioPSE: Problem solving environment for modeling, simulation, and visualization of bioelectric fields.* Scientific Computing and Imaging Institute (SCI), http://software.sci.utah.edu/biopse.html, **2002**.

[107] Ahlberg, C. Visual exploration of HTS databases: bridging the gap between chemistry and biology. *Drug Discovery Today* **1999**, *4*, 370-376.

[108] Xu, J. *SCA: New Cluster Algorithm for Structural Diversity Analysis and Applications.* Spotfire Webcast, 30.05.**2001**.

[109] Xu, J. A New Approach to Finding Natural Chemical Structure Classes. *J. Med. Chem.* **2002**, *45*, 5311-5320.

[110] *Molecular Diversity Manual Version 6.8 – Distill* October **2001**, Tripos Inc., St. Louis, MO 63144.

[111] Bron, C.; Kerbosch, J. *Finding All Cliques of an Undirected Graph*, Collected Algorithms from CACM, Algorithm 457, http://www.netlib.org/tomspdf/457.pdf.

[112] Technical Support *LeadScope 2.0 Whitepaper*, http://www.leadscope.com/products/whitepaper.html, 28. Juni **2001**.

[113] Roberts, G.; Myatt, G. J.; Johnson, W. P.; Cross, K. P.; Blower, P. E. LeadScope: Software for Exploring Large Sets of Screening Data. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 1302-1314.

[114] Blower, P. E.; Johnson, W. P.; Myatt, G. J. Method of analyzing, organizing and visualizing chemical data with feature hierarchy. Patente US6323852 (20011127) und EP1157325 (20011128).

[115] Nicolaou, Ch.; Kelley, B.; Nutt, R.; Bassett, S. Method and system for artificial intelligence directed lead discovery through multi-domain clustering. Patent WO 049539A1 vom 24. Aug. **2000**.

[116] Vivien, V.; Williams, R. *Analysis of HIV/NCI Data.* Bioreason Produktpräsentation, Pharmaforschungszentrum Bayer AG, 22. Mai **2001**.

[117] Nicolaou, Ch. Growing Phylogenetic-Type Trees Describing SAR Groups for Lead Discovery from HTS Data. *MUG 1999 Conference*, 25. Feb. **1999**, Santa Fe, New Mexico.

[118] Nicolaou, C. A.; Tamura, S. Y.; Kelley, B. P.; Bassett, S. I.; Nutt, R. F. Analysis of Large Screening Data Sets via Adaptively Grown Phylogenetic-Like Trees. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 1069-1079.

[119] Bacha, P. A.; Gruver, H. S.; Den Hartog, B. K.; Tamura, S. Y.; Nutt, R. F. Rule Extraction from a Mutagenicity Data Set Using Adaptively Grown Phylogenetic-like Trees. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 1104-1111.

[120] Tamura, S. Y.; Bacha, P. A.; Gruver, H. S.; Nutt, R. F. Data Analysis of High-Throughput Screening Results: Application of Multidomain Clustering to the NCI Anti-HIV Data Set. *J. Med. Chem.* **2002**, *45*, 3082-3093.

[121] Bemis, G. W.; Murcko, M. A. The Properties of Known Drugs. 1. Molecular Frameworks. *J. Med. Chem.* **1996**, *39*, 2887-2893.

[122] Datenbank Comprehensive Medicinal Chemistry (CMC) Release 94.1 erhältlich von MDL Information Systems Inc., San Leandro, CA.

[123] Bemis, G. W.; Kuntz, I. D. A fast and efficient method for 2D and 3D molecular shape description. *J. Comput.-Aided Mol. Des.* **1992**, *6*, 607-628.

[124] Xu, J. GMA: A Generic Match Algorithm for Structural Homomorphism, Isomorphism, and Maximal Common Substructure Match and Its Applications. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 25-34.

[125] Sheridan, R. P.; Miller, M. D. A Method for Visualizing Recurrent Topological Substructures in Sets of Active Molecules. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 915-924.

[126] Andrews, P. R.; Craik, D. J.; Martin, J. L. Functional group contributions to drug-receptor interactions. *J. Med. Chem.* **1984**, *27*, 1648-1657. Williams, D. H.; Westwell, M. S. Aspects of weak interactions. *Chem. Soc. Rev.* **1998**, *27*, 57-64.

[127] Nicolaou, K. C.; Pfefferkorn, J. A.; Roecker, A. J.; Cao, G.-Q.; Barluenga, S. et al. Natural Product-like Combinatorial Libraries Based on Privileged Structures. 1. General Principles and Solid-Phase Synthesis of Benzopyrans. *J. Am. Chem. Soc.* **2000**, *122*, 9939-9953.

[128] Breinbauer, R.; Vetter, I. R.; Waldmann H. From Protein Domains to Drug Candidates – Natural Products as Guiding Principles in Compound Library Design and Synthesis. *Angew. Chem. Int. Ed.* **2002**, *41*, 2878-2890 bzw. *Angew. Chem.* **2002**, *114*, 3002-3015.

[129] Downs, G. M.; Gillet, V.; Holliday, J.; Lynch, M. Review of ring perception algorithms for chemical graphs. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 172-187.

[130] Sedgewick, R. *Algorithmen*, Addison-Wesley: Bonn, München, **1991**, Kapitel 29: Elementare Algorithmen für Graphen.

[131] Weiss, M. A. *Data Structures and Algorithm Analysis in C*, Benjamin/Cummings: California, **1993**, Chapter 9: Graph Algorithms.

[132] Figueras, J. Ring Perception Using Breadth-First Search. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 986-991.

[133] Balaban, A. T.; Filip, P.; Balaban, T.-S. Computer Program for Finding All Possible Cycles in Graphs. *J. Comp. Chem.* **1985**, *6*, 316-329.

[134] Hanser, T.; Jauffret, P.; Kaufmann, G. A New Algorithm for Exhaustive Ring Perception in a Molecular Graph. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 1146-1152.

[135] Laet de, A.; Hehenkamp, J.; Wife, R. Finding Drug Candidates in Virtual and Lost/Emerging Chemistry. *J. Heterocyclic Chem.* **2000**, *37*, 669-674.

[136] Petitjean, M.; Fan, B. T.; Panaye, A.; Doucet, J. Ring Perception: Proof of a Formula Calculating the Number of Smalles Rings in Connected Graphs. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 1015-1017.

[137] Garcia, G. C.; Ruiz, I.; Gomez-Nieto, M. Cyclical Conjunction: An Efficient Operator for the Extraction of Cycles from a Graph. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 1415-1424.

[138] Fujita, S. A new algorithm for selection of synthetically important rings. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 22-26.

[139] Downs, G. M. Ring Perception. *Encyclopedia of Computational Chemistry*, v. Ragué-Schleyer, P. (Editor-in-Chief), Wiley: New York, **1999**, 2509-2515.

[140] Sadowski, J.; Gasteiger, J. From Atoms and Bonds to Three-Dimensional Atomic Coordinates: Automatic Model Builders. *Chem. Rev.* **1993**, *93*, 2567-2581. http://www2.chemie.uni-erlangen.de/software/corina/index.html.

[141] Pearlman, R. S. Rapid Generation of High Quality Approximate 3D Molecular Structures. *Chem. Des. Auto. News.* **1987**, *2*, 1-7.

[142] CONCORD: A Program for the Rapid Generation of High Quality Approximate 3-Dimensional Molecular Structures, University of Texas, Austin; Vertrieben von Tripos Inc., St. Louis, MO 63144.

[143] Helson, H. E. Structure Diagram Generation. *Reviews in Computational Chemistry Volume 13* Edited by Lipkowitz, K. B.; Boyd, D. B. Wiley-VCH: New York, **1999**, 313-398.

[144] Weininger, D. SMILES 3. Depict: Graphical Depiction of Chemical Structures. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 237-243.

[145] *Depict Toolkit V 4.8,* http://www.daylight.com/products/depict_kit.html. James, C.; Weininger, D.; Delany, J.; Kappler, M.; Scofield J. *Daylight Toolkit Programmer's Guide Version 4.82* Juni **2003**, Daylight Chemical Information Systems Inc., Santa Fe, NM 87501, Chapter 11. DEPICT TOOLKIT, http://www.-daylight.com/dayhtml/doc/prog/prog.depict.html.

[146] Bley, K.; Brandt, J.; Dengler, A.; Frank, R.; Ugi, I. Constitutional Formulae Generated from Connectivity Information: The Program MDRAW. *J. Chem. Res. (M)* **1991**, 2601-2689.

[147] Steinbeck, Ch. *JMDraw 0.9*, http://jmdraw.sourceforge.net/, August **2000**.

[148] Müller, O.; Kunze, R. Vorlesung Computergrafik, Universität Osnabrück, Fachbereich Mathematik/ Informatik, *Kapitel 5: Clipping*. http://www-lehre.informatik.uni-osnabrueck.de/~cg/1997/Skript/kap5.ps bzw. http://www-lehre.informatik.uni-osnabrueck.de/~cg/2002/Pdf/.

[149] *UNITY User Guide Version 4.3* October **2001**, Tripos Inc. St. Louis, MO 63144, Chapter 3. Unix Commands: dbreport, 196-199.

[150] Wild, D. J. *VisualiSAR: A Web-based SAR Tool.* Daylight User Group Meeting, Februar **1999**. O'Donnell, T. *MolAlign*. Daylight User Group Meeting, März **2001**.

[151] Ullmann, J. R. An algorithm for subgraph isomorphism. *J. Assoc. Comput. Mach.* **1976**, *23*, 31-42.

[152] Burden, F. R. Molecular Identification Number for Substructure Searches. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 225-227.

[153] Randic, M. On Molecular Identification Numbers. *J. Chem. Inf. Comput. Sci.* **1984**, *24*, 164-175.

[154] Randic, M. Molecular ID Numbers by Design. *J. Chem. Inf. Comput. Sci.* **1986**, *26*, 134-136.

[155] Ihlenfeldt, W.-D.; Gasteiger, J. Hash Codes for the Identification and Classification of Molecular Structure Elements. *J. Comput. Chem.* **1994**, *15*, 793-813.

[156] Morgan, H. L. The Generation of a Unique Machine Description for Chemical Structures - A Technique Developed at Chemical Abstracts Service. *J. Chem. Doc.* **1965**, *5*, 107-113.

[157] Xu, Y.; Johnson, M. Algorithm for Naming Molecular Equivalence Classes Represented by Labeled Pseudographs. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 181-185.

[158] Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. *Numerical Recipes in C*, Cambridge University Press, **1992**, Kapitel 7.4 Generation of Random Bits, 296-300.

[159] Voigt, J. H.; Bienfait, B.; Wang, S.; Nicklaus, M. C. Comparison of the NCI Open Database with Seven Large Chemical Structural Databases. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 702-712.

[160] The Tradition of Graph Drawing, http://www.gd2003.org/tradition.html.

[161] Battista, G.; Eades, P.; Tamassia, R.; Tollis, I. Algorithms for Drawing Graphs: an Annotated Bibliography. *Comput. Geom.* **1994**, *4*, 235-282. ftp://wilma.cs.brown.edu/pub/papers/compgeo/gdbiblio.ps.

[162] Fröhlich, M.; Werner, M. *The Graph Visualization System daVinci – A User Interface for Applications*, Technical Report No. 5/94, Department of Computer Science, University of Bremen, **1994**.

[163] Koutsofios, E.; North, S. Drawing Graphs with dot – dot User´s Manual, AT&T Bell Laboratories, Murray Hill, New Jersey, http://www.graphviz.org.

[164] Gansner, E.; Koutsofios, E.; North, S.; Vo, K. A Technique for Drawing Directed Graphs. *IEEE Trans. Software Eng.* **1993**, *19*, 214-230.

[165] Felsenstein, J. *PHYLIP (Phylogeny Inference Package) Version 3.5c.* Department of Genetics, University of Washington, Seattle, **1993**. http://evolution.genetics.washington.edu/phylip.html. Felsenstein, J. PHYLIP - Phylogeny Inference Package (Version 3.2). *Cladistics* **1989**, *5*, 164-166.

[166] Becker, M. Y.; Rojas, I. A graph layout algorithm for drawing metabolic pathways. *Bioinformatics* **2001**, *5*, 461-467.

[167] Shin, C.; Kim, S.; Kim, S.-H.; Chwa, K. Algorithms for drawing binary trees in the plane. *Information Processing Letters* **1998**, *66*, 133-139.

[168] Moen, S. Drawing Dynamic Trees. *IEEE Software* **1990**, *7*, 21-28.

[169] Brighton, A. *Tree-8.0.3 - A Tree Widget for Tk8.0.3 based on C++ and optionally [incr Tk-3.0]*. http://archive.eso.org/~abrighto/tree/tree.html.

[170] Foley, J. D.; van Dam, A.; Feiner, S. K.; Hughes, J. F. *Computer Graphics: Principles and Practice,* Addison-Wesley: Reading, 1990.

[171] MacAdam, D. L. Specification of small chromaticity differences. *Journal of the Optical Society of America* **1943***, 33*, 18-26.

[172] Levkowitz, H. Perceptual Steps Along Color Scales. *Intl. Journal of Imaging Systems and Technology* **1996***, 7*, 97-101.

[173] Ankerst, M. *Visual Data Mining*, Dissertation, Institut für Informatik der Universität München, Arbeitskreis Prof. Kriegel, **2000**.

[174] Keim, D. A. *Visual Support for Query Specification and Data Mining*, Institut für Informatik, Universität München, **1994**.

[175] Estrada, E. Spectral Moments of the Edge Adjacency Matrix in Molecular Graphs. 1. Definition and Applications to the Prediction of Physical Properties of Alkanes. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 844-849.

[176] Estrada, E. Spectral Moments of the Edge-Adjacency Matrix of Molecular Graphs. 2. Molecules Containing Heteroatoms and QSAR Applications. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 320-328.

[177] Estrada, E. Spectral Moments of the Edge Adjacency Matrix in Molecular Graphs. 3. Molecules Containing Cycles. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 23-27.

[178] Ferguson, L. N. *The Modern Structural Theory Of Organic Chemistry*, Englewoods Cliffs: Prentis-Hall, **1963,** 200.

[179] Steward, J. J. *MOPAC 93 Manual,* Fujitsu Limited, **1994**.

[180] Potapov, V. *Stereochemistry*, Mir: Moscow, **1978**.

[181] Estrada E. A computer-based approach to describe the ^{13}C NMR chemical shifts of alkanes by the generalized spectrall moments of the iterated line graphs. *Computers & Chemistry* **2000**, *24*, 193-201.

[182] Estrada, E. Generalized Spectral Moments of the Iterated Line Graphs Sequence. A Novel Approach to QSPR Studies. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 90-95.

[183] Estrada, E.; Gutierrez, Y.; Gonzalez, H. Modeling Diamagnetic and Magnetooptic Properties of Organic Compounds with the TOSS-MODE Approach. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 1386-1399.

[184] Estrada, E.; González, H. What Are the Limits of Applicability for Graph Theoretic Descriptors in QSPR/QSAR? Modeling Dipole Moments of Aromatic Compounds with TOPS-MODE Descriptors. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 75-84.

[185] Estrada, E.; Peña, A.; García-Domenech, R. Designing sedative/hypnotic compounds from a novel substructural graph-theoretical approach. *J. Comput.-Aided Mol. Design* **1998**, *12*, 583-595.

[186] Estrada, E.; Uriarte, E.; Montero, A.; Teijeira, M.; Santana, L. et al. A Novel Approach for the Virtual Screening and Rational Design of Anticancer Compounds. *J. Med. Chem.* **2000**, *43*, 1975-1985.

[187] Estrada, E.; Peña, A. In Silico Studies for the Rational Discovery of Anticonvulsant Compounds. *Bioorg. Med. Chem.* **2000**, *8*, 2755-2770.

[188] Bone, R. G. A.; Villar, H. O. Exhaustive Enumeration of Molecular Substructures. *J. Comp. Chem.* **1997**, *18*, 86-107.

[189] Rücker, G.; Rücker, Ch. Automatic Enumeration of All Connected Subgraphs. *match* **2000**, *41*, 145-149.

[190] Sanatvy, M.; Labute, P. SVL: The Scientific Vector Language. *J. Chem. Computing Group.* http://www.chemcomp.com/features/svl.html.

[191] MOE (Molecular Operating Environment), Version 2001.01, Chemical Computing Group Inc., 1255 University Street, Montreal, Quebec, Canada, H3B 3X3.

[192] Fisher, R.A. The use of multiple measurements in taxonomic problems. *Annals of Eugenics* **1936**, *7*, 179-188.

[193] Huberty, C. J. *Applied Discriminant Analysis*, John Wiley & Sons: New York, **1994**.

[194] SPSS Inc., 233 South Wacker Drive, Chicago, IL 60606-6412. SPSS 11.5 Syntax Reference Guide Base System, Advanced Models, Regression Models Command DISCRIMINANT, 301-320. SPSS Inc. http://www.spss.com/tech/stat/Algorithms/11.5/discriminant.pdf.

[195] R Development Core Team *An Introduction to R.* **1999**, http://www.r-project.org/. Venables, W. N.; Ripley, B. D. *Modern Applied Statistics with S-PLUS*, Springer: New York, **1999**. http://www.stats.ox.ac.uk-/pub/MASS3/.

[196] Henrion, R.; Henrion, G. *Multivariate Datenanalyse*, Springer: Berlin, Heidelberg, **1995**. Kapitel 4.5 Bayessche Klassifikation, 83-87.

[197] James, M. *Classification Algorithm,*. Collins Professional and Technical Books: London, **1985**.

[198] Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. *Numerical Recipes in C*, Cambridge University Press, **1992**, Kapitel 11.1 Jacobi Transformations of a Symmetric Matrix, 463-469.

[199] Jolliffe, I. T. *Principal Component Analysis*, Springer: New York, **1986**.

[200] Hawkins, D.; Basak, S.; Mills, D. Assessing Model Fit by Cross-Validation. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 579-586.

[201] James, M. *Classification Algorithm,* Collins Professional and Technical Books: London, **1985**, Chapter 6: Evaluating Rules – Estimating Error Rates, 77.

[202] Molchanova, M. S.; Shcherbukhin, V.V.; Zefirov, N. S. Computer Generation of Molecular Sctructures by the SMOG Program. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 888-899.

[203] Grund, R.; Kerber, A.; Laue, R. MOLGEN, ein Computeralgebra-System für die Konstruktion molekularer Graphen. *match* **1992**, *27*, 87-131.

[204] Brinkmann, G.; Dobrynin, A. A.; Krause, A. Fast Generation of Polycyclic Chains with Arbitrary Ring Sizes. *match* **2000**, *41*, 137-144.

[205] Scheibe, A. Diplomarbeit Arbeitskreis Brinkmann, Universität Bielefeld, in Bearbeitung.

[206] Suchenwirth, R. *Einfach Tcl* http://mini.net/tcl/2548.html; Tcl/Tk http://www.tcl.tk/.

[207] Welch, B.; Thomas, M. *The Tcl Extension Architecture*, http://www.tcl.tk/doc/tea/tea_tcl2k.pdf.

[208] Kernighan, B.; Ritchie, D. *Programmieren in C*, Hanser: München, Wien, **1990**.

[209] Josuttis, N. *The C++ standard library: a tutorial and reference*, Addison Wesley Longman: Reading Massachusetts, **1999.**

[210] Osterhout, J. *Tcl und Tk: Entwicklung grafischer Benutzerschnittstellen für das X Windows System*, Addison-Wesley: München, **1995**.

[211] Qt3: http://www.trolltech.com/products/qt/. GTK+: http://www.gtk.org/.

[212] Seeger, J. Leserumfrage 2000. *iX Magazin für professionelle Informationstechnik*, November **2000**, 10, http://www.heise.de/ix/artikel/2000/11/010/. Seeger, J. Leserumfrage 2002. *iX Magazin für professionelle Informationstechnik*, November **2002**, 12, http://www.heise.de/ix/artikel/2002/11/012/.

[213] Balls, S. *TclXML: Tcl interface to XML parser*, http://www.zveno.com/open_source/, http://tclxml-.sourceforge.net/.

[214] Ball, S. XML Support for Tcl. *Proceedings of the 6th Annual Tcl/Tk Workshop*, USENIX, September **1998**, 109.

[215] Pilhofer, F. A CORBA Language Mapping for Tcl. *Proceedings of the First European Tcl/Tk Meeting*, Hamburg, 15. Juni **2000**.

[216] Pilhofer, F. *Combat: CORBA scripting with Tcl*, http://www.fpx.de/Combat/.

[217] http://www.ActiveState.com/Tcl, http://www.ActiveState.com/ASPN/Tcl/Downloads/.

[218] Ihlenfeld, W.-D., http://www.xemistry.de/ und ftp://www2.ccc.uni-erlangen.de/pub/catvs/ bzw. http://www2.chemie.uni-erlangen.de/software/cactvs/index.html.

[219] Ihlenfeldt, W. D.; Takahashi, Y.; Abe, H.; Sasaki, S. Computation and Management of Chemical Properties in CACTVS: An Extensible Networked Approach toward Modularity and Flexibility. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 109-116.

[220] Ihlenfeldt, W.-D. *Preliminary Documentation for the CACTVS Chemistry Toolkit*. Computer-Chemie-Centrum, Universität Erlangen, Juli **2001**.

[221] James, C.; Weininger, D.; Delany, J.; Kappler, M.; Scofield J. *Daylight Toolkit Programmer's Guide Version 4.82* Juni **2003**, Daylight Chemical Information Systems Inc., Santa Fe, NM 87501, http://www.daylight.com/dayhtml/doc/prog/prog.toc.html.

[222] Dalke, A. http://www.dalkescientific.com/, info@dalkescientific.com, http://pydaylight.sourceforge.net/.

[223] Stahl, M. T. *OELib Primer: An Introduction to Programming with the OpenEye Library*, http://www.eyesopen.com/oelib.html.

[224] SWIG: Simplified Wrapper and Interface Generator, http://www.swig.org/.

[225] *UNITY User Guide Version 4.3* October **2001**, Tripos Inc., St. Louis, MO 63144, Chapter 3. Unix Commands: dbtranslate, 241-247.

[226] DeJong, M.; Redman, S. *Tcl and Java Integration*, http://www.tcl.tk/software/java/.

[227] Stanton, S. TclBlend: Blending Tcl and Java. *Dr. Dobb´s Journal* Februar **1998**, http://www.ddj.com-/documents/ddj9802d/.

[228] Csizmadia, F. JChem: Java Applets and Modules Supporting Chemical Database Handling from Web Browsers. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 323-324. http://www.chemaxon.com.

[229] Steinbeck, Ch.; Han, Y.; Kuhn, S.; Horlacher, O.; Luttmann, E., Willighagen, E. The Chemistry Development Kit (CDK): An Open-Source Java Library for Chemo- and Bioinformatics. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 493-500. http://cdk.sourceforge.net.

[230] Witten, I. H.; Eibe, F. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann, **1999**. Quellcode: http://www.cs.waikato.ac.nz/ml/weka oder www.mkp.com/datamining.

[231] *UNITY 4.3 – Java-Based Hitlist Manager* October **2001**, Tripos Inc., St. Louis, MO 63144.

[232] Rost, U.; Bornberg-Bauer, E. TreeWiz: interactive exploration of huge trees. *Bioinformatics* **2002**, *18*, 109-114.

[233] BayNews *Bayer erweitert die High-Tech-Plattform für Life-Science mit einzigartiger Informatik.* Pressemitteilung vom 16. Oktober **2000**.

[234] Sausville, E. A.; Shoemaker, R. H. Role of the National Cancer Institute in Acquired Immunodeficiency Syndrome-Related Drug Discovery. *J. Natl. Cancer Inst. Monographs* **2000,** *28*, 55-57.

[235] http://dtp.nci.nih.gov/docs/aids/aids_data.html.

[236] *UNITY User Guide Version 4.3* October **2001**, Tripos Inc., St. Louis, MO 63144.

[237] Weislow, O. S.; Kiser, R.; Fine, D. L.; Bader, J.; Shoemaker, R. H.; Boyd, M. R. New Soluble-formazan Assay for HIV-1 Cytopathic Effects: Application to High-flux Screening of Synthetic and Natural Products for AIDS-antiviral Activity. *J. Natl. Cancer Inst.* **1989**, *81*, 577-586. Erratum in *J. Natl. Cancer Inst.* **1989**, *81*, 963.

[238] Mitsuya, H.; Yarchoan, R.; Broder, S. Molecular Targets For Aids Therapy. *Science* **1990**, *249*, 1533-1544.

[239] Solov'ev, V. P.; Varnek, A. Anti-HIV Activity of HEPT, TIBO and Cyclic Urea Derivatives: Structure-Property Studies, Focused Combinatorial Library Generation and Hits Selection Using Substructural Molecular Fragment Method. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1703-1719.

[240] Chang, J. Classification Tool LogisticRegression.py *Biopython Package V1.00a4* **2001**, http://www.biopython.org.

[241] SPSS Inc. *Logistic Regression* http://www.spss.com/tech/stat/Algorithms/11.5/logistic_regression.pdf, **2001**. Schafer, J. L. *Introduction to Logistic Regression*, Statistics 544: Categorical Data Analysis I, Fall **2001**, Department of Statistics, Pennsylvania State University, PA, http://www.stat.psu.edu/~jls/stat544-/2001/lec8.pdf.

[242] Gao, H.; Williams, Ch.; Labute, P.; Bajorath, J. Binary Quantitative Structure-Activity Relationship (QSAR) Analysis of Estrogen Receptor Ligands. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 164-168.

[243] Labute, P. Binary QSAR: A New Method for the Determination of Quantitative Structure Activity Relationships. *Proceedings of the 1999 Pacific Biocomputing Symposium* **1999**, 444-455.

[244] Miller, D. W. Results of a New Classification Algorithm Combining K Nearest Neighbors and Recursive Partitioning. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 168-175.

[245] Miller, D. W. A Chemical Class-Based Approach to Predictive Model Generation. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 568-578.

[246] Allen, Ch. *Oracle PL/SQL für Einsteiger. Der Einsatz von SQL und PL/SQL in der Oracle-Datenbank*, Hanser: München, Wien, **2001**.

[247] Loney, K. *Oracle8i. Die umfassende Referenz*, Hanser: München, Wien **2001**.

[248] *UNITY Reference Guide Version 4.3* October **2001**, Tripos Inc., St. Louis, MO 63144, Chapter 6.1 UNITY Hitlist Format, 44-47.

[249] Baldi, P.; Brunak, S.; Chauvin, Y.; Andersen, C.; Nielsen, H. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics* **2000**, *16*, 412-424.

[250] Engels, M.; Wouters, L.; Verbeeck, R.; Vanhoff, G. Outlier Mining in High Troughput Screening Experiments. *J. Biomol. Screen.* **2002**, *7*, 341-351.

[251] Das Verfahren wird vom Autor an anderer Stelle veröffentlicht.

[252] Zafrany, S. TkPaint 1.5.4, http://www.netanya.ac.il/~samy/tkpaint.html.

[253] MDL Information Systems Inc., 14600 Catalina Street, San Leandro, CA 94577, CTfile Formats, August **2002**, http://www.mdli.com/downloads/literature/ctfile.pdf.

[254] Weininger, D. SMILES, A Chemical Language and Information-System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31-36.

[255] *SLN Manual* October **2001**, Tripos Inc., St. Louis, MO 63144.

[256] Ash, S.; Cline, M. A.; Homer, R. W.; Hurst, T.; Smith, G. B. SYBYL Line Notation (SLN): A Versatile Language for Chemical Structure Representation. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 71-79.

Lebenslauf

Persönliche Daten

Name	Dipl.-Chem. Stefan Seidler
Geburtsdatum	13. Juni 1971
Geburtsort	München
Nationalität	deutsch
Eltern	Prof. Dr. Franz W. Seidler; Renate E. Seidler, geb. Gütgemann
Familienstand	verheiratet, zwei Kinder

Schulausbildung und Zivildienst

1977 - 1981	Grundschule am Lehrer-Götz-Weg, München
1981 - 1983	Schulformunabhängige Orientierungsstufe München
1983 - 1991	Werner-von-Siemens-Gymnasium, München
1991 - 1992	Zivildienst

Studium

Nov. 1992 - Okt.1998	Diplomstudium Chemie mit Nebenfach Informatik an der Ludwig-Maximilians-Universität München
Februar 1998	Hauptdiplomprüfung
März - Oktober 1998	Diplomarbeit "Implementierung des Messy Simulated Annealing Verfahrens zur Optimierung von Pharmakophormodellen", Prof. H. Ebert, Institut für Physikalische Chemie, Universität München und Dr. A. Jensen, Pharmaforschung der Bayer AG
seit April 1999	Doktorand in der Pharmaforschung der Bayer AG und am Institut für Physikalische Chemie, AK Prof. Ebert, Promotionsstudium am Department Chemie der Universität München

Berufliche Tätigkeiten

April 1999 - April 2002	Mitarbeiter am Forschungszentrum der Bayer AG inWuppertal
Juli - November 2002	Wissenschaftlicher Mitarbeiter am Department Chemie der Ludwig-Maximilians-Universität München

Sonstige Tätigkeiten

Mai 1995 - März 1997	Studentischer Mitarbeiter am Institut für Physikalische Chemie, Prof. Bräuchle
Aug. 1996 - Jan 1997	Softwareentwicklung für das Prüfungssekretariat der Fakultät Chemie und Pharmazie
März - April 1997	Praktikum in der Pharmaforschung der Bayer AG