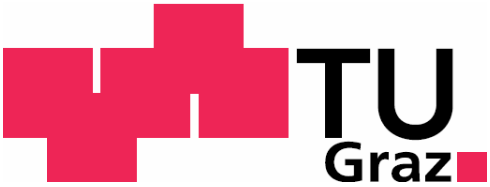
Gunnar Libiseller

Masterarbeit

DA Metabolomics: Aufbau einer Datenbank zur Identifizierung von Metaboliten aus LC/FTMS-Daten – MS-MetaboliteDB



JOANNEUM RESEARCH
Institut für Medizinische Systemtechnik und Gesundheitsmanagement
Stiftingtalstrasse 24
8010 Graz

Betreuer: Dr. Christoph Magnes

Begutachter: Dr. Zlatko Trajanoski

Graz, September 2010

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Ort, Datum	Libiseller Gunnar

Für meine Eltern

Abstract

Metabolomics, the most recent of all “-omics“-technologies, is devoted to the in-depth analysis of small molecular substances in biological specimen. The investigation of biomarkers for medical diagnosis and the development of pharmaceutical agents, in particular, is expected to profit significantly from this new technology. Currently, hardly any technological standards for metabolomics are available – neither in the field of analysis systems nor regarding bioinformatical tools, such as databases and raw data preparation.

The underlying aim of this master’s thesis is to build up a database system for the identification of metabolites from LC/FTMS-data. For this purpose, a MySQL-database was developed which stores data on metabolites on the one hand, while additionally recording data from the analysis on the other hand. Preprocessing of the data is done using the package XCMS, which is available in R. A graphical user interface was implemented using SWING-Components from Java, which serves as interface between the database and the statistical computer language R. Data derived thereof is stored in the analysis section of the database. Whereas current public databases identify metabolites only by mass, the database of Joanneum-Research takes the retention time into consideration as well, which allows a more precise identification of metabolites. In addition to the tabular representation of the data within the GUI, the creation of MSMS-diagrams and EIC-diagrams is furthermore possible.

Keywords: Metabolomics, mass spectrometry, Java, MySQL, R, XCMS, retention time, database

Kurzfassung

Metabolomics, die neueste aller „-omics“-Technologien beschäftigt sich mit der breiten Analyse kleinmolekularer Substanzen in biologischen Proben. Speziell die Biomarkerforschung für medizinische Diagnostik und pharamazeutische Wirkstoffentwicklung erhofft sich mit Hilfe dieser neuen Technologie signifikante Fortschritte. Derzeit gibt es noch kaum technologische Standards für Metabolomics – weder im Bereich der Analysensysteme noch im Bereich der Bioinformatik-Werkzeuge wie Datenbanken und Rohdatenaufbereitung.

Ziel dieser Diplomarbeit ist es ein Datenbanksystem aufzubauen das Metaboliten aus LC/FTMS-Daten identifiziert. Daher wurde eine MySQL-Datenbank entwickelt die zum einen Informationen über Metaboliten enthält und zum anderen Analysedaten speichert. Die Vorverarbeitung erfolgt mittels des R-Paketes XCMS. Eine grafische Benutzeroberfläche, mittels den in Java enthaltenen SWING-Komponenten implementiert, dient als Schnittstelle zur Datenbank sowie der statistischen Programmiersprache R. Die daraus gewonnen Daten werden im Analysenteil der Datenbank gespeichert. Bei öffentlichen Datenbanken ist lediglich eine Identifikation bezüglich der Masse möglich. Die Datenbank für Joanneum Research berücksichtigt zudem die Retentionszeit was zu einer sichereren Identifikation der Metaboliten führt. Zur Datenrepräsentation ist, zusätzlich zur tabellarischen Darstellung in der GUI, das Erzeugen von MSMS-Diagramme und EIC-Diagramme möglich.

Stichwörter: Metabolomics, Massenspektroskopie, Java, MySQL, R, XCMS, Retentionszeit, Datenbank

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

APCI	Atmospheric Pressure Chemical Ionization (Chemische Ionisation bei Atmosphärendruck)
API	Application Programming Interface
Brite	Biomolecular Relations in Information Transmission and Expression
CAS	Chemical Abstract Service
ChEBI	Chemical Entities of Biological Interest
CML	Chemical Marup Language
DNP	Dictionary of Natural Products
EC	Enzyme Commission
EIC	Extraced Ion Chromatogram
FTMS	Fourier-Transform-Massenspektrometer
HMDB	Human Metabolome Database
HMP	Human Metabolome Project
HTTP	Hyper Text Transfer Protocol
GPL	General Public License
GUI	Graphical User Interface
InChI	International Chemical Identifier
IUPAC	International Union of Pure and Applied Chemistry (Internationale Union für reine und angewandte Chemie)
KEGG	Kyoto Encylopedia of Genes and Genomes
KO	KEGG Ontology
LCMS	Liquid Chromatographie-Mass Spectrometry (Flüssigkeitschromatographie mit Massenspektrometrie-Koppelung)
LGPL	GNU Lesser Gerneral Public License
Metlin	Metabolite Link
MFE	Molecular Feature Extraction
MFG	Molecular Formel Generator
MoTo DB	Metabolome Tomato Database
MS	Massenspektrometer
NIST	National Institute of Standards and Technology
NMR	Nuclear Magnetic Resonance (Kernspinnresonanz)
OBO	Open Biomedical Ontologies

PHP	Hypertext Preprocessor oder Personal Home Page Tools
SMILES	Simplified Molecular Input Line Entry Specification
SNR	Signal-to-Noise-Ratio
SOAP	Simple Object Access Protocol
SQL	Structured Query Language (Strukturierte Abfragensprache)
SSDB	Sequence Similarity DataBase
WSDL	Web Service Description Language
XCMS	verschiedene Arten (X) chromatografischer Massenspektronomie
XML	Extensible Markup Language

1 Einführung

1.1 Metaboliten, Metabolomics und Metabonomics

Die Enzyklopedie der Universität von Columbia (The Columbia Encyclopedia, 2009) definiert einen Metaboliten als eine organische Verbindung die Ausgangs, Zwischen- oder Endprodukt des Stoffwechsels ist. Ausgangsprodukte sind Substanzen die für gewöhnlich klein sind, eine einfache Struktur besitzen und von einem Organismus als Nahrung aufgenommen werden. Hierzu zählen auch Vitamine und essentielle Aminosäuren. Ausgangsprodukte können zu komplexere Molekülen zusammengefügt werden oder sie können in noch einfachere aufgespalten werden. Am häufigsten kommen Zwischenprodukte vor. Sie können ebenfalls zu komplexeren Substanzen zusammengefügt oder in einfachere zerbrochen werden wobei oft chemische Energie freigesetzt wird. Hierbei ist vor allem Glukose zu erwähnen. Endprodukte sind das Ergebnis der Aufspaltung und können normalerweise keine anderen Metaboliten aufbauen. Daher werden sie aus dem Organsimus ausgeschieden wie etwa im Urin.

Man unterscheidet zwischen endogenen und exogenen Metaboliten. Endogene werden vom Organismus selbst hergestellt wie zum Beispiel Hormone. Metaboliten die dem Körper zugeführt werden nennt man exogene Metaboliten, beispielsweise Vitamine. Alle Metaboliten zusammen bilden das Metabolom das auch manchmal Metabonom genannt wird.

Die Begriffe Metabolomics und Metabonomics werden oft als Synonym für einander verwendet (Ryan, D. und Robards, K., 2006). Beide haben das Ziel das Metabolom zu analyisieren.
Metabolomics ist die umfassende Identifikation und Quantifizierung aller Metaboliten eines biologischen Systems. Die analyitschen Methoden müssen sehr selektiv und empfindlich sein. Zur Zeit gibt es keine Technik oder Kombination aus Techniken die es erlaubt alle mikrobischen, pflanzlichen oder die Metaboliten von Säugetieren parallel zu bestimmen. (Dunn, W. B., Bailey, N. J. und Johnson, H. E., 2005)

Metabonomics ist die quantitative Messung der dynamischen, multiparametrischen, metabolischen Reaktionen von lebenden Systemen auf pathophysiologische Stimulation oder genetische Modifikation. (J.K.Nicholson, J.C.Lindon und E.Holmes, 1999)

Metabolismus oder einfach Stoffwechsel wird laut (Villas-Bôas, S. G., Nielsen, J. und Smedsgaard, J., 2007) durch folgende Punkte definiert:

(i) Alle chemischen Reaktionen in Lebewesen sind organisiert und verknüpft in einem Netzwerk aus Stoffwechselwegen
(ii) Der Metabolismus versorgt die Zellen mit Ressourcen (Energie, Baustoffe) und ist essentiell für das Überleben der Zellen. Dieser Vorgang wird stark durch die Wechselwirkung mit der Umwelt beeinflusst.
(iii) Metaboliten speichern freie Energie in den Zellen oder sind in Strukturen der Zellen gebunden.
(iv) Spezifische Kontrollmechanismen beeinflussen Stoffwechselreaktionen.
(v) Man kann zwischen zentralen oder primären und sekundären Stoffwechsel unterscheiden. Der primäre Metabolismus nimmt hauptsächlich Bezug auf Energie und Aufbau der Hauptstrukturen einer Zelle. Viele primäre Metaboliten sind weitverbreitet in der Natur und sind in Funktion und Struktur in vielen Organsimen ident. Sekundärer Metabolismus bezieht sich auf viel spezialisierterer Metaboliten, die oft einzigartig für eine Spezies sind und oft eine unbekannte Funktion haben.
(vi) Desweiteren kann man zwischen anabolischen und katabolischen Stoffwechselreaktionen unterscheiden. Anabolische Reaktion dienen dem Aufbau von Zellen und der Speicherung von Energie. Katabolische hingegen sind für das Zerlegen komplexer Moleküle und der Freisetzung von Energie verantwortlich.

Das Interesse am Forschungsbereich Metabolomics wird immer stärker. Ein Indiz dafür ist die steigende Anzahl an Publikationen zu diesem Thema über die letzten Jahre. 2002 gab es ungefähr 100 Publikationen. Im Jahre 2006 hat sich die Zahl schon mehr als vervierfacht. Treibende Kraft hinter der Forschung ist die Möglichkeit, neues Wissen zu generieren. Die Nutzbarkeit dieses Wissen für die

Allgemeinheit kann nur über gemeinsame Standards führen (Sansone, S. A. und andere, 2007).

1.2 Ziel

Das Ziel der Diplomarbeit ist die Identifikation von Metaboliten aus LC/FTMS-Analysen. Dazu wird eine Datenbank benötigt die aus zwei Hauptbereichen besteht, dem Analyseteil und dem Metabolitenteil. Einzelne Analysen enthalten Informationen über gefundene Peaks, MSMS-Spektren sowie Metadaten zum Versuchsaufbau. Peaks aus mehreren Analysen können zu Features zusammengefasst werden. Diese Daten sollen im Analyseteil der Datenbank gespeichert werden. Der zweite Bereich der Datenbank enthält Informationen über Metaboliten wie etwa Namen, Masse, Formel, Retentionszeiten oder Links zu anderen Datenbanken. Beim Einfügen neuer Analysedaten sollen automatisch möglichst wahrscheinliche Metaboliten vorgeschlagen und Verknüpfungen zwischen diesen mitgespeichert werden können. Die Zuordnung soll sowohl über die Masse als auch die Retentionszeit erfolgen. Desweiteren soll eine grafische Benutzeroberfläche implementiert werden, die es ermöglicht Daten in die Datenbank zu spielen, von dort abzurufen und darzustellen, zu löschen oder Daten zu verändern.

1.3 Instrumente für die Analyse von Metaboliten

Die Wahl der richtigen Technologie für die Analyse von Metaboliten spielt eine entscheidenede Rolle, da es kein Gerät gibt, dass alle Metaboliten parallel detektieren kann. Der Grund dafür ist, dass das Ionisieren aller Metaboliten mit einer MS-Methode nicht funktioniert oder die Konzentration zu niedrig ist. Die zwei verbreitetsten Technologien sind das MS und NMR.

1.3.1 Massenspektrometer

Es ist möglich verschiedene Geräte zur Separation, Ionisierung und Detektion mitteinander zu koppeln (Scalbert, A. und andere, 2009). LCMS ist eine leistungsfähige Kombination das die hohe Auflösung der

Flüssigkeitschromatographie mit dem Identifikationsvermögen eines Massenspektrometers verbindet. Die Empfindlichkeit liegt dabei im Picogrammbereich (10^{-12}g). In den 90er Jahren konnten Fortschritte beim Interface mittels Elektrospray Ionisierung und APCI (atmospheric pressure chemical ionization) erzielt werden.
Die Analysen für die Diplomarbeit stammen aus einem LTQ Orbitrap XL Gerät das eine Kombination aus LC und FTMS ist.

1.3.2 Kernspinresonanzspektroskopie

Ebenfalls weit verbreitet ist die NMR-Spektroskopie. Sie hat eine höhere Durchsatzrate als MS und kann einige Hundert Proben pro Tag analysieren, hat jedoch nicht die selbe Sensitivität, welche etwa im Mikrogrammbereich (10^{-6}g) liegt. Aus den einzelnen Resonanzen in einem NMR-Spektrum kann man an Informationen über spezifische Verbindungen gelangen. So gibt die Kombination von chemischen Verschiebungen Aufschluss über das chemische Umfeld in denen ein Atomkern vorkommt, die Spin-Spin Koppelung zeigt die benachtbarten Atomkerne an und somit gleichzeitig die Bindungen. Rückschlüsse auf die Größe der Moleküle gibt die Diffusion.(Dunn, W. B., Bailey, N. J. und Johnson, H. E., 2005)

1.4 Onlinedatenbanken

Es gibt zahlreiche, öffentlich zugängliche Metabolitendatenbanken. Zugriffsmöglichkeiten, Architektur und beinhaltete Daten wurden recherchiert und interessante Lösungen in das Konzept der institutseigenen Datenbank übernommen. Die folgenden Kapitel geben einen Überblick über die wichtigsten öffentlichen Datenbanken und deren Funktion.

1.4.1 KEGG – Kyoto Encyklopedia of Genes and Genomes

KEGG ist der wichtigeste Teil des GenomeNet Projektes, das 1991 gestarted wurde, und soll die folgenden 4 Ziele erreichen (Ogata, H. und andere, 1999)(KEGG, 2009):

- Das gegenwärtige Wissen biochemischer Stoffwechselwege und andere Arten molekularer Interaktionen zu digitalisieren und darzustellen wie Moleküle oder Gene miteinander vernetzt sind.
- KEGG beinhaltet alle Gene einiger bisher komplett entschlüsselter Genome und einige noch nicht vollständige Genome. Es wird versucht eine konsistente und standardisierte Notation zu gewährleisten indem Gene auf Komponenten des biochemischen Stoffwechselweges verlinkt sind.
- Die LIGAND Datenbank enthält chemische Elemente, Verbindungen und andere Substanzen die ebenfalls mit dem Stoffwechselweg verknüpft sind. Das geschieht aufgrund der Annahme, dass sowohl genetische Informationen aus dem Genome, als auch chemische Information aus der Zelle benötigt werden um Zellfunktionen zu verstehen.
- Neue Technologien im Bereich der Informatik werden zu Verfügung gestellt um biologische Systeme simulieren und zukünftige Experimente designen zu können.

Tabelle 1 enthält einige statistische Information über die Anzahl der Datensätze in den einzelnen KEGG Datenbanken.

Datenbank	Datensatz	Anzahl
GENES	Genes in high-quality genomes (112 eukaryotes + 927 bacteria + 68 archaea)	5.051.680
SSDB	Best-Hit	31.266.386.284
	Best-Best-Hit	539.527.977
PATHWAY	Pathway maps, reference (total)	341 (96.674)
COMPOUND	Metabolites and other small molecules	16.034
REACTION	Biochemical reactions	8.044
BRITE	Functional hierarchies, reference (total)	84 (26.462)
MODULE	Pathway Modules	704
DISEASE	Human diseases	114
DRUGS	Drugs	9.148
ORTHOLOGY	KEGG Orthology (KO) Groups	12.772
GENOME	Organisms	1.203
DGENES	Genes in draft genomes (10 eukaryotes)	146.715
EGENES	Genes as EST contigs (85 eukaryotes)	3.350.468
GLYCAN	Glycans	10.969
RPAIR	Reactant pair chemical transformations	11.782
ENZYMES	Enzyme nomenclature	5.075

Tabelle 1: einige Statistiken von KEGG Datenbanken bezogen von http://www.genome.jp/kegg/docs/statistics.html im November 2009

KEGG besteht aus den folgenden Datenbanken.

Gene Database

Die Gendatenbank enthält komplette und einige unvollständige Genome. KEGG versucht alle Datensätze aktuell zu halten, da viele öffentliche Datenbanken das nicht tun.

Der Zugriff auf die Gendatenbank ist auf drei Arten möglich:

- Suche mittels DBGET/LinkDB
- Die Information über den Stoffwechselweg kann mittels eines hierarchischen Textbrowser ausgelesen werden

- Die Position im Chromosom kann mittels eines Java-Applet ermittelt werden

SSDB - Sequence Similarity DataBase

Die SSDB Datenbank enthält Informationen über alle Gemeinsamkeiten der Gene. Hierzu wurden alle möglichen, paarweisen Vergleiche mittels eines eigenem Programmes erstellt. Zusätzlich sind alle best hits und best-best hits gespeichert. Ein best hit ist jenes Gene, dass einem anderem am ähnlichsten ist. Das sind mittlerweile schon über 31 Millionen. Ein best-best hit tritt dann auf wenn beide Gene von einander die best hits sind. Die Datenbank liegt auf einem eigenen Server. Man kann alle ähnlichen Gene suchen und mittels eines Treshholds bestimmen, wie ähnlich sie sein müssen.

Pathway Database

Um den Stoffwechsel in jedem Organismus verstehen zu können, ist die Rekonstruktion und Analyse von Stoffwechselwegen essentiell. Die Pathway-Datenbank spiegelt die Interaktionen von Proteinen wider. Dabei wird zwischen Enzym-Enzym und Protein-Protein Interaktionen sowie Verbindungen von Gendarstellungen unterschieden. Die Informationen über die Stoffwechselwege sind zurzeit in sechs Gruppen eingeteilt die wiederum hierarchisch aufgebaut sind. Diese sind, Metabolismus, Verarbeitung genetischer Informationen, Verarbeitung umweltbezogener Informationen, zelluläre Prozesse, Humankrankheiten und Medikamentenentwicklung. Abbildung 1 zeigt den Schwefel-Stoffwechsel. Die Kästchen sind mit Enzymen, Metaboliten oder anderen Pathways aus KEGG verlinkt.

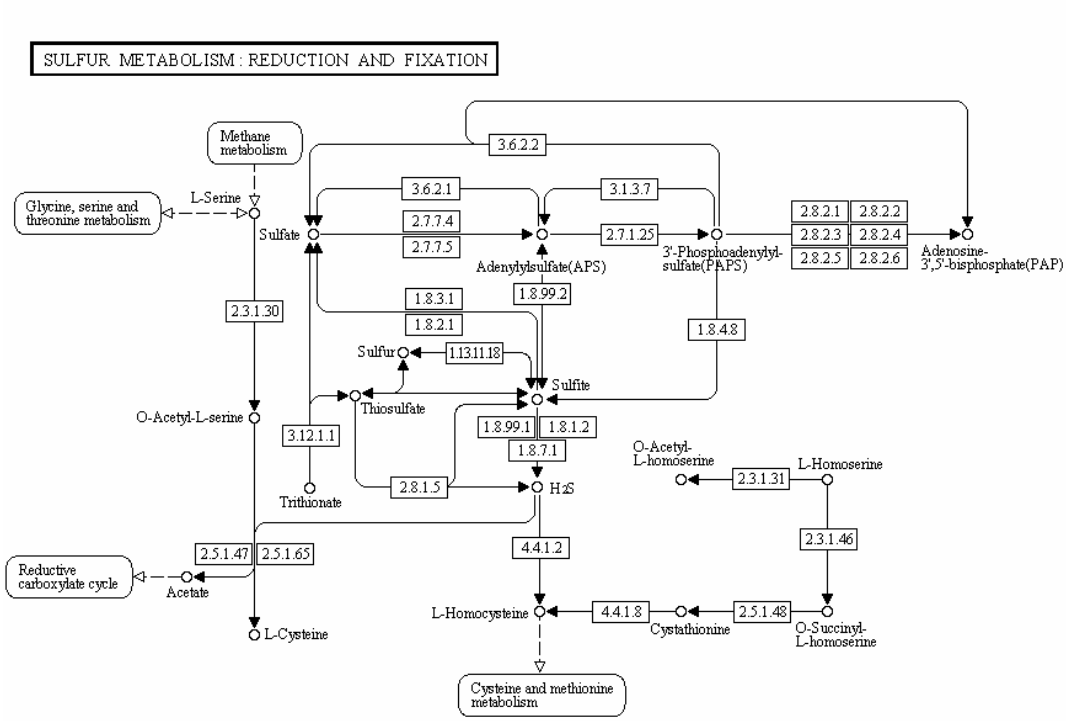


Abbildung 1: Schwefel-Stoffwechselweg

LIGAND Database

LIGAND besteht im Prinzip aus den Teilen COMPOUND, ENZYME und REACTION. ENZYME sammelt Informationen über alle bekannten Enzyme und Reaktionen, die mittels EC-Nummern klassifiziert werden. Ein Eintrag in der ENZYME-Datenbank besteht aus drei Teilen:

- Beschreibung des Enzymes und der Reaktion die es katalysiert
- Alle chemischen Verbindungen die eine Zugehörigkeit zum Enzym besitzen
- Links zu anderen Datenbanken

Das Hauptaugenmerk liegt auf dem chemischen Aspekt der enzymatischen Reaktion wobei durch die Verlinkung zu anderen Datenbanken der biologische Aspekt nicht außer acht gelassen wird.

Die COMPOUND-Datenbank beinhaltet alle chemischen Verbindungen, die in der ENZYME-Datenbank vorkommen. Jeder Eintrag bekommt eine eindeutige Nummer und es werden Namen, Synonyme und die Formel abgespeichert. Desweiteren wird die chemische Strukture als Bilddatei und im MDL MOL

Dateiformat archiviert. Das letztere Format steht mittels DBGET/LinkDB zur Verfügung und kann mit Programmen wie ISIS/Draw in ein 3d-Bild umgewandelt und verändert werden. Abbildung 2 zeigt einen Eintrag aus der COMPOUND-Datenbank.

Der REACTION-Teil enthält für jede Reaktion eine eindeutige Nummer sowie Verknüpfungen zum betroffenem Enzym und zum Stoffwechselweg in der die Reaktion stattfindet. Desweiteren werden die Reaktionen in drei Gruppen unterteilt. Die erste enthält jene Enzyme, die in der ENZYME-Datenbank mit EC-Nummer gespeichert sind, die zweite jene die dort nicht sind und die letzte Gruppe beinhaltet nicht-enzymatische Reaktionen. Die gespeicherten Informationen können zur Berechnung verschiedener Stoffwechselwege verwendet werden.

(Goto, S., Nishioka, T. und Kanehisa, M., 1998)(Goto, S., Nishioka, T. und Kanehisa, M., 1999)

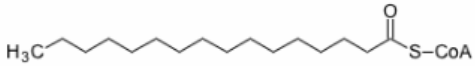
Entry	C00154 Compound
Name	Palmitoyl-CoA; Hexadecanoyl-CoA
Formula	C37H66N7O17P3S
Mass	1005.3449
Structure	 C00154 Mol file KCF file DB search Jmol KegDraw
Remark	BRITE hierarchy
Reaction	R01274 R01277 R01278 R01279 R01280 R01281 R01282 R01923 R01952 R02138 R02367 R02375 R04021 R04022 R05918 R07758 R08381
Pathway	PATH: ko00062 Fatty acid elongation in mitochondria PATH: ko00071 Fatty acid metabolism PATH: ko00563 Glycosylphosphatidylinositol(GPI)-anchor biosynthesis PATH: ko00600 Sphingolipid metabolism PATH: ko01040 Biosynthesis of unsaturated fatty acids PATH: ko01070 Biosynthesis of plant hormones PATH: ko01100 Metabolic pathways
Enzyme	1.2.1.42 1.3.1.8 1.3.1.38 1.3.3.6 1.3.99.3 1.3.99.13 1.3.99.- 2.3.1.21 2.3.1.50 2.3.1.76 2.3.1.96 2.3.1.100 2.3.1.123 2.3.1.139 2.3.1.142 2.3.1.- 2.3.-.- 3.1.2.2 3.1.2.22 6.2.1.3
Other DBs	CAS: 1763-10-6 PubChem: 3454 ChEBI: 15525 LIPIDMAPS: LMFA07050002 KNApSAcK: C00007462 3DMET: B04641 NIKKAJI: J207.891K
LinkDB	All DBs
KCF data	Show

Abbildung 2: Eintrag aus der COMPOUND-Datenbank

Brite – Biomolecular Relations in Information Transmission and Expression

Brite ist eine Datenbank die verschiedene, biologische Systeme hierarchisch sowie deren Verbindung untereinander darstellt. Somit kann man sie auch als eine ontologische Datenbank bezeichnen. Eine Ontologie besteht aus Daten und Regeln wie die Daten zusammenhängen. Sie ist somit ein Netzwerk aus Informationen die mittels Relationen verbunden sind.

Brite soll beim Entdecken empirischer Regeln bezüglich der Wechselbeziehung im Bereich des Genomes helfen sowie die funktionale Auswertung bezüglich der Pathway-Rekonstruktion automatisieren. Abbildung 3 zeigt beispielsweise eine Ontologie des Enzymes mit der EC-Nummer 3.1.3.7. In der Abbildung ist sowohl die Ontologie zum Auffinden des Enzymes im Schwefelstoffwechsel als auch die Ontologie für die hierarchische Einteilung des Enzymes ersichtlich.

```
KEGG Orthology (KO) [BR:ko00001]
  01100 Metabolism
    01102 Energy Metabolism
      00920 Sulfur metabolism [PATH:ko00920]
        K01082  E3.1.3.7; 3'(2'), 5'-bisphosphate nucleotidase [EC:3.1.3.7]

Enzymes [BR:ko01000]
  3. Hydrolases
    3.1  Acting on ester bonds
      3.1.3  Phosphoric-monoester hydrolases
        3.1.3.7  3'(2'),5'-bisphosphate nucleotidase
          K01082  E3.1.3.7; 3'(2'), 5'-bisphosphate nucleotidase [EC:3.1.3.7]
```

Abbildung 3: Brite Ontologie des Enzymes mit der EC-Nummer 3.1.3.7

DISEASE

Krankheiten sind oft von einer Kombination aus Umweltfaktoren und verschiedenen genetischen Faktoren abhängig. Zwischen diesen Faktoren und der Krankheitsursache besteht eine Verbindung. Einträge in der Datenbank bestehen aus einer eindeutigen Nummer und Listen bekannter, verursachender Gene, Umweltfaktoren, Symptome und Medikamenten. Die DISEASE-Datenbank soll helfen die molekularen Zusammenhänge besser zu verstehen indem es ein Netzwerk dieser Faktoren mit zur Verfügung stellt. Zusätzlich würde dieses Verständnis der Zusammenhänge bei der Entwicklung neuer Medikamente und Behandlungen unterstützend wirken.

DRUG

Die Drug Datenbank enthält chemischen Strukturen und Komponenten aller in Japan verschreibungspflichtigen und rezeptfreien Medikamente. Desweiteren ist ein Großteil der in den USA verschreibungspflichtigen Medikamente enthalten. Die Einbettung in die funktionale Hierarchie der BRITE-Datenbank gibt Aufschluss über die Wirksamkeit und den therapeutischen Nutzen der Medikamente. Da Medikamente in verschiedenen Ländern unterschiedliche Namen haben können oder derselbe Name für ein anderes Medikament verwendet wird, sind zusätzlich zu den Namen Codes gespeichert welche die Herkunft widerspiegeln. Zukünfig ist geplant auch Wechselwirkungen von Medikamenten ersichtlich zu machen. (Kanehisa, M. und andere, 2008)

DBGET/LinkDB

Dieses System ermöglicht es Daten aus allen GenomeNet-Datenbanken und einigen Onlinedatenbanken zu verwenden. Eine Suche mittels DBGET gibt dann alle Informationen, unterteilt in die Datenbanken in der sie gefunden wurden, zurück. LinkDB speichert Informationen über Verlinkungen zu über 100 anderen Datenbanken ab. Zurzeit beinhaltet die Datenbank über 570 Millionen Einträge. (Kanehisa, M. und andere, 2002)

KEGG API

Die KEGG API kann von anderen Programmen benutzt werden um den Zugriff auf KEGG zu automatisieren. Dabei wird die SOAP-Technologie (Simple Object Access Protocoll) verwendet, welche mittels HTTP Daten austauscht. Um unabhängig von Plattformen und Programmiersprachen zu bleiben, wird SOAP in Verwendung mit WSDL zur Verfügung gestellt welches die Struktur der versendeten Daten mittels XML beschreibt.

Es werden eine Vielzahl von Methoden angeboten. Man kann etwa Metaboliten nach Masse inklusive einer Abweichung oder einfach nach Namen suchen. Es ist auch möglich jene Stoffwechselwege zu finden, die alle Metaboliten einer Liste enthalten.

1.4.2 HMDB – Human Metabolome Database

Die HMDB ist derzeit wohl die weltgrößte Datenbank über menschlichen Metabolismus. Sie ist das Ergebnis des Human Metabolome Project (HMP) das 2004 gestartet wurde. Das Ziel war und ist es alle Metaboliten des menschlichen Körpers zu identifizieren und diese Information frei zugänglich zu machen. Grundgedanke war die Identifikation von Metaboliten aus den Daten von Massenspektrometern (MS) und Kernspinresonanzspektroskopie (NMR vom englischen nuclear magnetic resonance). Die 2.5 Version der HMDB beinhaltet 8.000 Metaboliten wie aus Tabelle 2 ersichtlich. Die Informationen über die Metaboliten werden in den sogenannten MetaboCards abgespeichert welche inzwischen 111 verschiedene Felder umfassen. Eine MetaboCard enthält Informationen wie Namen, Synonyme, Beschreibung, chemische Formel, Molekularmasse, Links zu anderen Onlinedatenbanken und noch viele mehr.

2009 wurde die zweite Version der HMDB vorgestellt. Die größte Veränderung zur Vorgängerversion betrifft die Anzahl der abgespeicherten Metaboliten. Desweiteren wurde das Suchen von Spektren verbessert. Hierzu wurde der Suchalgorithmus sowie die Datenbank selbst weiterentwickelt. Um die Anfragen an die Datenbank zu beschleunigen wurde der KinoSearch Algorithmus implementiert (KinoSearch, 2009). Das ist ein Suchmaschine die in Pearl implementiert und fünfmal so schnell wie die vorige Suche ist. Die hohe Geschwindigkeit wird erreicht indem der Text zunächst normalisiert wird. Dabei wird aus Wörter wie „horses“, „horse“, „horsing“ oder „horsed“ das Wort „hors“ gemacht. Häufig vorkommende Wörter wie „the“ oder „of“ werden komplett weggelassen. Zusätzlich unterstützt KinoSearch Korrekturhilfe bei Schreibfehlern und hebt gefundenen Text hervor.
Ebenso verändert wurde das Frontend von HMDB um die Darstellung der Suchergebnisse zu verbessern. HMDB liegt eine relationale MySQL Datenbank zu Grunde die ein webfreundliches Front-End besitzt. Sie läuft auf einem Linux Server mit 2 GHz Rechenleistung und 1 GB RAM. HMDB bietet keinen WebService an, der es erlaubt direkt auf die Datenbank zuzugreifen, dafür steht aber die komplette Datenbank im ZIP-Format zum Download zur Verfügung. (Wishart, D. S. und andere, 2009)(Wishart, D. S. und andere, 2007)(HMDB, 2009)

Datensatz	Anzahl
Metaboliten	7.982
Verbindungen mit MS/MS Spektren	2.654
Humane Stoffwechselwege	71

Tabelle 2: Datenbankstatistik der Version 2.5 der HMDB bezogen von www.hmdb.ca/release_notes im November 2009

1.4.3 MZedDB

Die manuelle Suche von Metaboliten in öffentlichen Datenbanken mittels der akkuraten Masse kann zeitaufwändig sein. Aufgrund der kleinen Anzahl gespeicherter Metaboliten beinhalten einzelne Datenbanken oft nicht das gesuchte Molekül und somit muss in mehreren Datenbanken gesucht werden.

Desweiteren stehen meist nicht genügend Mittel zur Pflege bereit, was zu redundanten Daten innerhalb einer Datenbank führen kann. MZedDB wurde entwickelt um diese Probleme zu vermeiden und eine Datenbank ohne Redundanz und zur automatischen Suche zur Verfügung zu stellen. Das wird erreicht indem Daten aus mehreren öffentlichen Datenbanken zusammengetragen und in ein einheitliches Format übertragen werden. Als Quellen werden

- HMDB
- Moto
- KEGG
- PubChem
- ChEBI
- MetaCyc
- Massbank
- MetaCrop
- Metlin
- ChemSpider
- KNApSAcK

verwendet wobei für die unterschiedlichen Datenfelder auch unterschiedliche Quellen verwendet werden. Für das Feld „Drugs“ etwa wurden die Datenbanken KEGG, PubChem und ChEBI befragt. In allen verwendeten Datenbanken ist aber eine Molekularformel vorhanden. Das sogenannte „Simplified Molecular Input Line Entry System“ (SMILES) ist die einfachste Methode nicht nur die Formel sondern auch die dahinterliegende Struktur zu speichern (Weininger, David, 2002). Die exakte Masse ist aber nicht Bestandteil jeder Datenbank und falls vorhanden reicht die Genauigkeit von 4-7 Dezimalstellen. Ein weiteres Problem entsteht dadurch, dass Metaboliten in verschiedenen chemischen Formen dargestellt werden können wie etwa geladen, neutral oder als Addukt. Um eine einheitliche Darstellung zu gewährleisten werden alle Informationen aus den Onlinedatenbanken geladen und die Salze entfernt. Metaboliten mit weniger als sechs Atomen und exotische Elementen werden ebenfalls nicht in die Datenbank aufgenommen. Falls möglich werden ionisierte Moleküle durch Hinzufügen oder Entfernen eines Wasserstoffatoms in eine neutrale Form gebracht. Die so

gewonnen Informationen werden in der Datenbank mittels SMILES, einer Identifikationsnummer und einem Link zur ursprünglichen Datenbank repräsentiert. Desweiteren wurden Regeln definiert, die aufgrund der Struktur und physikalischer Eigenschaften eine Liste wahrscheinlicher Kandidaten erstellt. Von den Kanditaten wird die akkurate Masse berechnet, die als Grundlage der Suche dient. Zusätzlich kann bestimmt werden in welchen anderen, öffentlichen Datenbanken mitgesucht werden soll. MZedDB wurde ebenfalls als MySQL Datenbank implementiert und läuft auf einem PowerPC mit 1.8 GHz Rechenleistung und 2 GB RAM. Das Webinterface ist öffentlich zugänglich und Datenbankanfragen werden mittels PHP übermittelt und beantwortet. MZedDB bietet desweiteren ein R-Skript an, welches das Suchen mittels selbst entwickelter Programmen in der Datenbank unterstützt. Außerdem kann man sich alle möglichen Summenformeln für eine bestimmte Masse berechnen lassen. (Draper, J. und andere, 2009)(MZedDB, 2009)

1.4.4 Metlin – Metabolite Link

Metlin ist eine weitere, öffentlich zugängliche Metabolitendatenbank. Sie enthält zur Zeit über 23.000 Metaboliten sowie teilweise deren FTMS, MS/MS oder LC/MS Daten wie in Tabelle 3 aufgeschüsselt. Jeder Metabolitendatensatz besteht zumindest aus der chemischen Formel und Struktur sowie Masse, Name und Links zu KEGG (Ogata, H. und andere, 1999) und CAS. Um besser mit den großen Datenmengen aus dem Massenspektrometer umgehen zu können, entwickelte Metlin eine eigene Datenmanagementstrategie. Um den Rechenaufwand zu vermindern, werden in der Datenbank nur die Peaks gespeichert, die zuvor noch von Rauschen bereinigt und durch Treshholding reduziert werden. Jeder Peak ist dreidimensional und besteht aus den Werten Retentionszeit, m/z und Intensität. Somit vermindert sich die Datenmenge laut Metlin um das tausendfache. Aber selbst nach Bearbeiten der Daten ist die Identifikation, durch Retentionszeitverschiebungen und weiterhin vorhandenem Rauschen, noch immer anspruchsvoll. Metlin ist ebenfalls eine relationale Datenbank und mittels MySQL realisiert. Das Webinterface und der Zugriff auf die Datenbank erfolgen mittels PHP. Die Rohdaten aus dem Massenspektrometer werden als Binärdaten gespeichert. Diese Daten werden

mittels einer eigenen Engine geparst um Peaks zu ermitteln und Grafiken auszugeben. Die interaktiven Grafiken werden mit Hilfe von GNUPLOT generiert. Metlin stellt keine API zur Verfügung die es ermöglicht mit einem selbstentwickelten Programm direkt auf die Datenbank zuzugreifen. (Smith, C. A. und andere, 2005)(Metlin, 2009)

Datensatz		Anzahl
LC/MS Analysen		216
Metaboliten		23.536
Metaboliten MS/MS Spektren	Positiver Modus	720
	Negativer Modus	711
Metaboliten mit MS/MS Spektren		1.069
Gesamte MS/MS Spektren		5.724

Tabelle 3: Statistik der Metlin Datenbank bezogen von http://metlin.scripps.edu/about.php im November 2009

1.4.5 Metlin Personal Database

Die Metlin Personal Database gehört nicht zu den Datenbanken die Online verfügbar sind, bietet aber einen interessanten Ansatz der Datenaufbereitung. Sie wurde eigens erstellt und mit Daten aus der Metlin Datenbank gefüllt. Zusätzlich wurde selbstentwickelte Funktionalität hinzugefügt, die es ermöglichen soll, unbekannte Metaboliten besser identifizieren zu können. Die korrekte Summenformel ist Vorraussetzung für diese Identifizierung. Ein sehr guter Ansatz die Formel zu finden sind die „Seven Golden Rules“ (Kind, T. und Fiehn, O., 2007). Das Suchen in Datenbanken verwendet normalerweise nicht alle Daten aus MS-Spektren. Die Software Agilent Masshunter Workstation wurde entwickelt, um auch diese Daten zum Finden der richtigen Formel zu berücksichtigen. Es wird eine Sortierung vorgenommen je nachdem welche der generierten Formeln am wahrscheinlichsten zutrifft. Diese Vorgehensweise reduziert die Zeit, die nötig ist, um die Daten zu interpretieren.

Ablauf

Nach analysieren der Proben im LCMS wurde MFE (Molecular Feature Extraction) verwendet um die vorhandenen Komponenten zu extrahieren. MFE bezeichnet Komponenten als Features. Da das gesamte Spektrum verwendet wird, ist es möglich mehrere Komponenten in nur einem Peak zu finden. Danach wird eine Datei erzeugt, die eine Liste aller gefundenen Features enthält. Die Features werden zum einen in der Datenbank gesucht und zum anderen wird mittels MFG (Molecular Formula Generator) eine Formel mit Wahrscheinlichkeit erstellt. Danach werden beide Ergebnisse kombiniert und somit eine Reihung vorgenommen. (Sana, T. R. und andere, 2008)

1.4.6 ChEBI – Chemical Entity of Biological Interest

Das Projekt ChEBI wurde 2002 vom Europäischen Bioinformatik Institut (EBI), mit dem Ziel eine freie Plattform für chemische Einheiten zu werden und eine Standardisierung chemischer Terminologie zu sein, ins Leben gerufen. Seit der ersten Veröffentlichung im Juli 2004 wuchs ChEBI auf über 19.000 Datensätze an. Abbildung 4 gibt einen Überblick über die Datenbankstatistik.
Das Projekt richtet sich nach vier Grundprinzipien:

- Die Terminologie die ChEBI verwendet ist eindeutig und wird von internationalen Gremien, wie etwa IUPAC, empfohlen.
- Keine Daten in der Datenbank sind proprietär oder von einer proprietären Quelle
- Jeder Datensatz ist auf die Originalquelle rückführbar
- Die komplette Datenbank ist jedem, ohne Vorbehalt mittels MySQL table dumps oder Flatfiles im Open Biomedical Ontologies (OBO) Format, verfügbar.

ChEBI strebte vom Beginn an eine 2-dimensionale Darstellung der chemischen Einheiten an. Um die genannten Prinzipien nicht zu verletzten, sind IUPAC Internation Chemical Identifier (InChI) und Chemical Markup Language (CML) ebenfalls Bestandteil der Datenbank. Die Datensätze für ChEBI stammen großteils aus drei Quellen:

- IntEnz: Eine relationale Enzymdatenbank von EBI
- KEGG COMPOUND: Ein Teil der LIGAND Datenbank die sich mit biochemischen Verbindungen befasst
- MSDchem: Die Makromolekülstruktudatenbank wurde im Jänner 2009 in Protein Datenbank in Europa umbenannt. Ebenfalls von EBI entwickelt bietet die Datenbank Onlinezugriff zu Liganden und kleinen Molekülen.

Außer den genannten Quellen wurden noch Daten aus anderen, öffentlich zugänglichen Datenbanken bezogen.

ChEBI basiert auf einer Oracle Datenbank. Die Funktionalität, wie etwa das Laden von Informationen aus externen Quellen, wurde mittels Java und Unix Skripten realisiert. Die Datenbank enthält die Felder: ChEBI ID, ChEBI Names, Definition, Structural diagrams, IUPAC InChI, SMILES, Formula, Ontology, IUPAC name(s), Synonyms, Database cross-references, Registry Number(s) und Comments. Die Ontologie ist eine Baumstruktur, die helfen soll die komplexe Klassifizierung chemischer Verbindungen zu veranschaulichen. Die komplette Datenbank steht als Flat-file table dumps, Oracle binary table dumps, SQL table dumps oder OBO ontology Format zum Download zur Verfügung. ChEBI bietet auch einen Webservice an, der, ähnlich wie die KEGG API, es erlaubt mittels SOAP-Technologie und WSDL direkt auf die Datenbank zuzugreifen. Es werden aber nicht soviele Suchmöglichkeiten wie bei KEGG angeboten. Die wichtige Suche nach Masse fehlt zur Zeit noch. (Degtyarenko, K. und andere, 2008)(ChEBI, 2009)

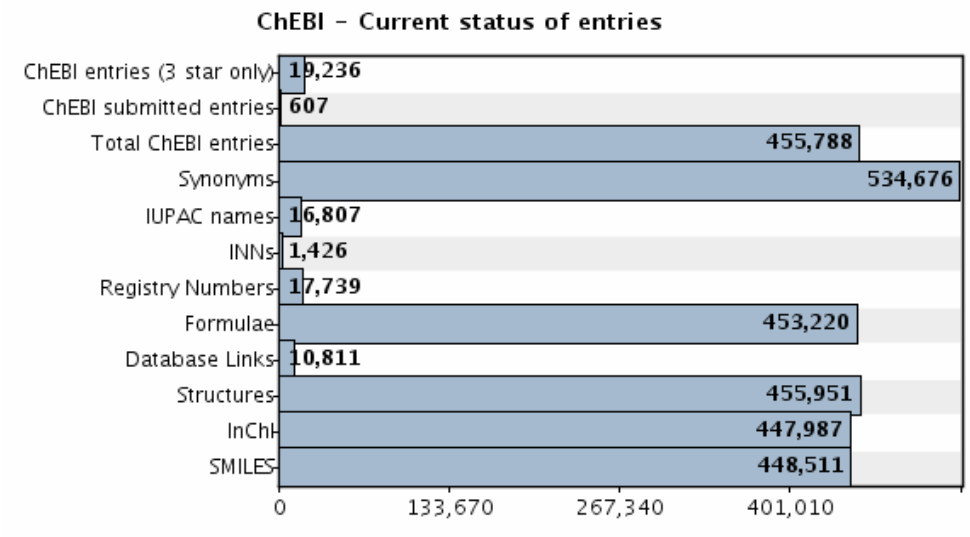


Abbildung 4: Datensätze in der ChEBI-Datenbank bezogen von http://www.ebi.ac.uk/chebi/statisticsForward.do im November 2009

1.4.7 MoTo DB – Metabolome Tomato Database

Die Metabolome Tomato Database enthält wie der Namen schon sagt nur Metaboliten die in Tomaten vorkommen. Bei einer Suche zwischen den Massen 0 und 2.000 wurden 110 Ergebnisse ausgegeben was wohl auch den gesamten Stand an Daten widerspiegelt. Die Suche gibt Informationen über akkurate Masse, Δmass und Δppm, Formel, PubChemId und IUPAC Namen. Folgt man dem Link „show more“ kommt man zur Ansicht die in Abbildung 5 dargestellt ist. Implementiert wurde die Datenbank mittels MySQL. (Moco, S. und andere, 2006)(MoTo DB, 2009)

Formula	CAS	PubChem
C27H30O16	153-18-4	517297

Names

Putative	Previously found	IUPAC
Quercetin-glucose-rhamnose	quercetin 3-O-rutinoside (rutin)	2-(3,4-dihydroxyphenyl)-5,7-dihydroxy-3-[3,4,5-trihydroxy-6-[(3,4,5-trihydroxy-6-methyl-oxan-2-yl)oxymethyl]oxan-2-yl]oxy-chromen-4-one

RT		Masses		
Av Ret (min)	StDev Ret (min)	MM	(M+H)+	(M-H)-
23.43	0.04	610.1534	611.1607	609.1461

MS/MS fragments	UV/Vis
301 271 255	256 299sh 355

Other

Standards	Non-hydrolized	Hydrolized
yes	found	not found

References

Fleuriet A, Macheix JJ (1977) Effect des blessures sur les composés phénoliques des fruits de tomates «cerise» (Lycopersicum esculentum var. cerasiforme). Physiol Veg 15: 239-250

Abbildung 5: Datensatz einer Verbindung in der MoTo Datenbank

1.4.8 MetaCyc

MetaCyc beinhaltet Informationen über Stoffwechselwege, enzymatische Reaktionen, Enzyme, chemische Verbindungen und Gene von verschiedensten Organismen. Anzumerken dabei ist, dass MetaCyc nur Daten enthält, die durch Experimente verifiziert sind. Die Datenbank hat das Ziel beim Verstehen von existenten biochemischen Netzwerken zu helfen. Desweiteren soll es möglich sein bewusst Modifikationen dieser Netzwerke zu erzeugen, um zelluläre Eigenschaften in gewünschte Bahnen zu lenken. Unter Modifikation wird das Hinzufügen, Ersetzen oder Entfernen eines Enzyms oder eines Pathways aus einem Organismus verstanden.

Pathways

Ebenso wie KEGG enthält MetaCyc Pathways. Jene von KEGG sind jedoch größer da auch Kombinationen von Pathways vorkommen wodurch die Übersichtlichkeit leidet. MetaCyc umgeht das Problem indem es verschiedene Kombinationen in sogenannten Superpathways abspeichert. Ein Datensatz zum

Stoffwechselweg enthält Namen, Synonyme, Superklasse, Spezien in denen der Pathway vorkommt, eine Zusammenfassung mit einer generellen Beschreibung, Beziehung zu anderen Pathways und Experimentelle Beweise. Desweiteren werden Links anderen Pathways und Datenbanken sowie Zitate gespeichert. Abbildung 6 zeigt die Grafik zu Cholindegeneration. Weitere Informationen werden als Text unter dem Pathway angezeigt.

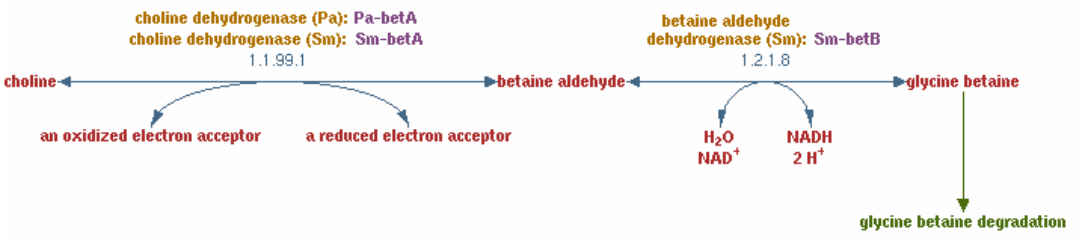


Abbildung 6: Pathway der Cholindegeneration bezogen von http://biocyc.org/META/NEW-IMAGE?type=PATHWAY&object=CHOLINE-BETAINE-ANA-PWY im November 2009

Compounds

Metaboliten können auf mehrere Arten gesucht werden. Einerseits mittels Namen oder ID, andererseits nach Masse wobei man eine Unter- und Obergrenze angeben kann. Desweiteren wird die Suche mittels Formeln, oder Teilen von Formeln, und InChI unterstützt. Sollten diese Suchkriterien nicht ausreichen, kann eine eigenes Suchschema in der erweiterten Suche erstellt werden. Eine Auswahl direkt via Ontologie ist ebenfalls möglich. Die Metaboliten sind aus mehreren Gründen in eine Klassenhierarchie eingeteilt. Zum einen kann eine Einteilung nach gegenseitig austauschbaren Enzymsubstrat getroffen werden, zum anderen nach funktionalen Gruppen oder der metabolischen Bestimmung. Datensätze bezüglich Metaboliten enthalten die Felder Namen und Synonyme, Superklasse, chemische Struktur, chemische Formel und Masse welche beide von der Struktur berechnet werden, Gibbs Bildungsenergie, Links zu anderen Datenbanken, Zusammenfassung und Zitate.

Reactions

Die Datensätze zu den Reaktionen beinhalten EC-Nummer, EC-Namen, ob die Reaktion spontan ist und Änderung in der Gibbsenergie wobei darauf geachtet werden muss, dass die Reaktionsrichtung stimmt. Unter dem Punkt

Zusammenfassung wird eine kurze Beschreibung gespeichert sowie ob die Reaktion neuartig ist und warum, ob sie theoretisch ist inklusive einer Beweisführung oder ob sie spontan ist und unter welchen Umständen. Zitate sind auch hier Teil des Datensatzes. Das Suchen mittels der Ontologie ist ebenfalls möglich. Es kann aber auch nach EC-Nummer und EC-Namen sowie Edukten und Produkten gesucht werden. Abbildung 7 zeigt eine Beispielgrafik zu einer Reaktion.

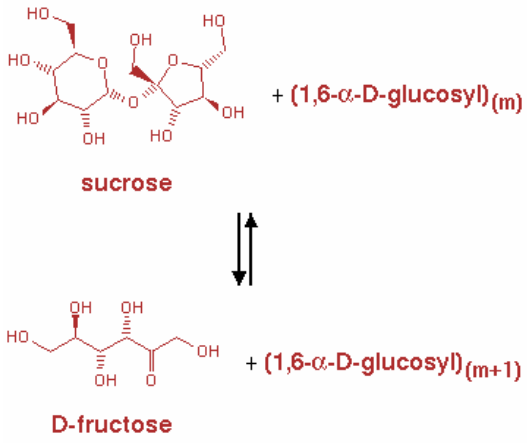


Abbildung 7: Reaktion mit der EC-Nummer: 2.4.1.5 bezogen von http://biocyc.org/META/NEW-IMAGE?type=REACTION&object=DEXTRANSUCRASE-RXN im November 2009

Proteine

Für Proteine werden zum einen generelle Informationen über das Protein zum anderen enzymatische Aktivitäten gespeichert. Zu den generellen Informationen gehören Spezies, Namen und Synonyme, Lokalisierung in der Zelle, Molekularmasse, Zusammenfassung, Zitate und wann der Datensatz zuletzt verändert wurde. Die enzymatischen Aktivitäten beinhalten Felder für Namen und Synonyme, Zusammenfassung, Hemmstoffe, Aktivatoren, Kofaktoren, alternative Substrate und Zitate.

Zur richtigen Verwaltung und Pflege der Datensätze wurde eigens ein Paper mit Richtlinien erstellt (Caspi, R., Fulcher, C. A., Ingraham, J., Keseler, I, Krummenacker, M., and Paley, S., 2009). Die Datenbank kann direkt auf dem

Computer gespeichert werden indem eine Reihe von Flatfiles heruntergeladen werden. Der Download wird viermal jährlich aktualisiert. MetaCyc stellt das Paket JavaCyc zur Verfügung mit dem es möglich ist Daten aus der heruntergeladenen Datenbank auszulesen. Tabelle 4 gibt einen Überblick über den Datenstand.

Datensatz	Anzahl
Pathways	1.399
Reaktionen	8.094
Proteine	2.432
Enzyme	5.857
Genes	5.508
Chemische Verbindungen	8.221

Tabelle 4: Statistik der MetaCyc Datenbanken bezogen von http://metacyc.org/release-notes.shtml im November 2009

(Caspi, R. und andere, 2008)(Caspi, R. und andere, 2006)(Karp, P. D. und andere, 2002)(MetaCyC, 2009)

1.4.9 Zusammenfassung Onlinedatenbanken

Die beschriebenen Datenbanken sind natürlich alle etwas anders aufgebaut und unterstützen verschiedene Funktionalitäten. Einen groben Überblick bietet Tabelle 5. Die Anzahl der Metaboliten in MZedDB ist in keiner Statistik enthalten. Datenfelder >X bedeutet, dass einige Felder mehr als einen Wert enthalten können. Das Minuszeichen bei dem MSMS-Spektren der MoTo DB bedeutet, dass nur die Anzahl der MSMS-Fragmente angegeben wird, die Spektren ansich sind jedoch nicht verfügbar. MetaCyc bietet zwar keine API an, die es erlaubt direkt auf die Onlinedatenbank zuzugreifen, dafür aber ein Java-Tool, das es erlaubt auf die heruntergeladenen Datenbank zuzugreifen.

Datenbank	#Metaboliten	Datenfelder	MSMS-Spektren	API	Download-bar
KEGG	16.034	> 8	Nein	Ja	Nein
HMDB	7.982	ca. 111	Ja	Nein	Ja
MZedDB	?	> 12	Nein	Ja	Nein
Metlin	23.536	13	Ja	XCMS	Nein
ChEBI	19.236	> 24	Nein	Ja	Ja
MoTo DB	110	> 10	-	Nein	Nein
MetaCyc	8.221	> 10	Nein	Ja	Ja

Tabelle 5: Überblick über die besprochenen Onlinedatenbanken

1.5 Datenverarbeitung

Um die Rohdaten aus dem Massenspektrometer verarbeiten zu können müssen diese erst in ein standardisiertes Format gebracht werden, damit Programme wie XCMS damit arbeiten können.

1.5.1 Formate

Rohdaten aus Massenspektrometern sind nicht standardisiert was dazu führt, dass sie erst in eines der folgenden Formate konvertiert werden müssen.

MzXML

MzXML ist ein frei verfügbares Datenformat welches zum Austausch von Massenspektrometerdaten dient. Zusätzlich zu MS-Daten kann es auch MS^n-Daten speichern. Um mit etwaigen Innovationen Schritt halten zu können, muss ein Format flexibel genug sein um diesen Veränderungen stand zu halten aber auch robust genug um sicher verwendet werden zu können. Unter Flexibilität ist gemeint, dass man einfach neue Felder definieren kann ohne dass sich beim Zugriff auf ältere Versionen was ändert. Das kommt daher, dass MzXML, wie der Namen schon sagt, mit XML definiert ist. XML für seinen Teil ist definiert als erweiterbare Sprache. Da aber diese Flexibilität leicht zu Dialektbildung führen

kann und die Standardisierung dadurch nicht mehr gegeben wäre, wurde ein Forum eingerichtet in dem Feedback gegeben werden kann und eine Fachgruppe den Standard vorgibt. XML hat aber auch Nachteile. So kann aufgrund der großen Datenmenge die ein Massenspektrometer erzeugt, der Zugriff auf die Information in der mzXML-Datei lange dauern. XML kann keine Binärdaten speichern und die Umwandlung in menschenlesbaren Text würde eine nochmalige Vergrößerung der Datenmenge mit sich bringen. Daher wird eine Base64-Codierung verwendet. Diese ist zwar ein bißchen größer als die ursprünglichen Binärdaten, können aber in XML eingebunden werden. Um die Datenmenge noch weiter zu reduzieren, hilft es alle Peaks mit einer Intensität von Null einfach zu eleminieren. Ein weiterer Nachteil von XML ist, das Informationen sequentiell ausgelesen werden müssen. Will man also den 500 Scan auslesen muss man zuerst alle 499 vorangehende lesen. MzXML umgeht das Problem indem es für jeden Scan einen Index mit der Position in der Datei gibt. Damit können Programme einen Offset angeben, der angibt von wo weg gelesen werden soll. Abbildung 8 zeigt den schematischen Aufbau einer MzXML-Datei. Das Root-Element ist das mzXML-Element das vier Kinder besitzt. In msRun werden die eigentlich Daten gespeichert. Index hält alle Offsets der Scans. IndexOffset zeigt an, wo der Index beginnt. Sha1 ist eine Prüfsumme die sicherstellt, dass alle Daten im Dokument enthalten sind. Die Überprüfung erfolgt einfach mittels Neuberechnung der Prüfsumme und Vergleich mit der im Dokument gespeicherten Summe.

MsRun hat für sich selbst wieder ein eigenes XML-Schema definiert. Es selbst besitzt noch die Attribute scanCount, startTime und endTime und sollte daher so aussehn:

<msRun scanCount="4388" startTime="PT0.3738S" endTime="PT3000.16S">

ParentFile gibt eine chronologische Liste aller Dateien wieder, die verwendet wurden um diese Datei zu erzeugen. msInstrument enthält Informationen über das verwendete Gerät. Informationen über die Konvertierung der Rohdaten zu mzXML-Format und jegliche andere vorangehende Verarbeitung wird im Feld dataProcessing gespeichert. Da manche MS-Techniken sehr eng mit der Seperationsart gekoppelt sind, wurde das Feld separation eingeführt. Scan enthält alle Informationen über einen Scan wie etwa die Retentionszeit. Darin

enthalten sind auch die Base64-codierten Peaks. Sha1 gibt wieder eine Prüfsumme, diesmal pro Lauf, an.
(Pedrioli, P. G. und andere, 2004)

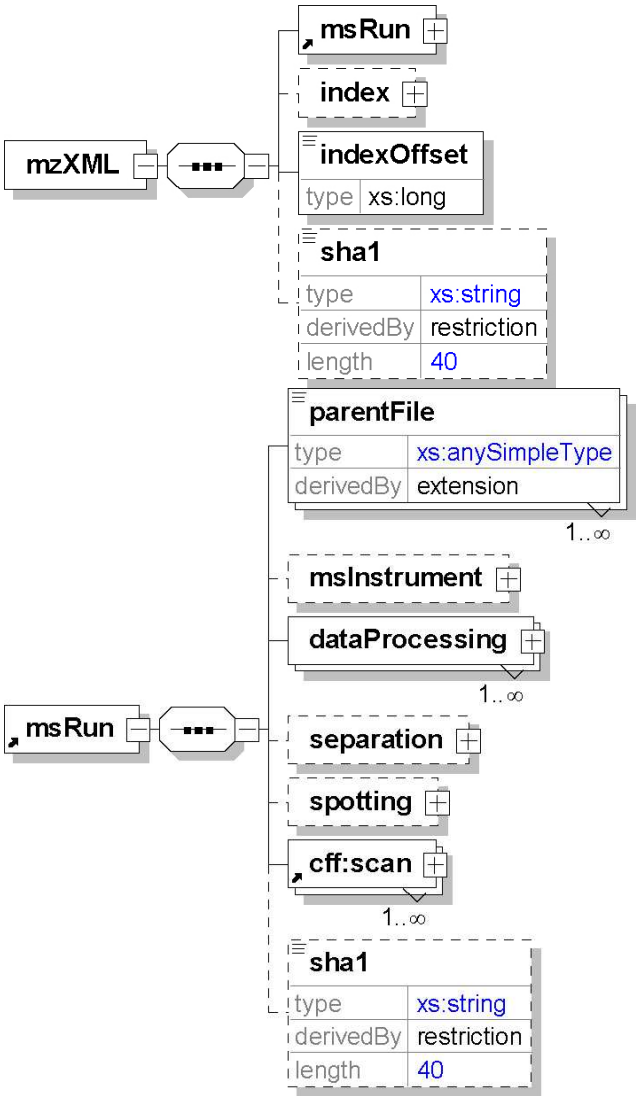


Abbildung 8: Schematischer Aufbau des mzXML-Formates bezogen von http://sashimi.sourceforge.net/schema_revision/mzXML_2.1/Doc/mzXML_2.1_tutorial.pdf im November 2009

MzML

MzML wurde als Nachfolger von mzXML und mzData entwickelt da ein einheitlicher Standard für alle geschaffen werden sollte. MzData ist weitaus flexibler als mzXML was jedoch zur Bildung von Dialekten führt was Probleme für die Entwickler von Parsern und anderer Softwareentwicklern bringt, die mit den verschiedenen Dialekten arbeiten sollten. MzML löst dieses Problem indem es einen semantischen Validierer zusätzlich zu den kontrollierten Vokabular veröffentlicht. Diese Vorgehensweise erzwingt die korrekte Verwendung des Vokabulars an der richtigen Stelle. Abbildung 9 gibt ein Schema wieder, das nicht die volle Tiefe des mzML-Formats wiederspiegelt, jedoch den grunsätzlichen Aufbau gut veranschlicht.

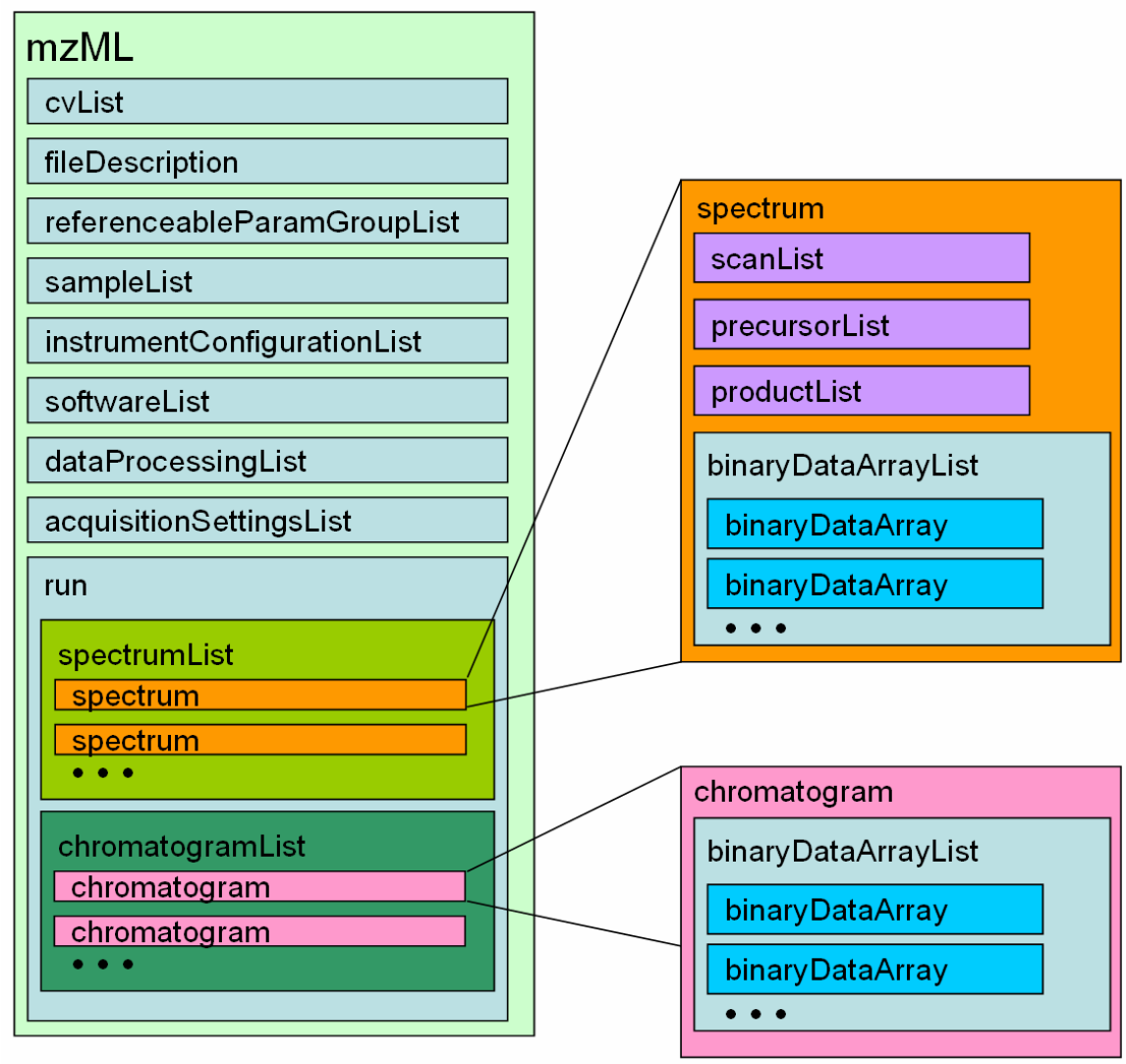


Abbildung 9: Highlevel-mzML-Format bezogen von Seite 13 von http://psidev.cvs.sourceforge.net/*checkout*/psidev/psi/psi-ms/mzML/document/mzML1.1.0_specificationDocument.doc im November 2009

1.5.2 XCMS

XCMS ist ein Akronym und steht für verschiedene Formen (X) von chromatografischer Massenspektronomie. Es ist ein frei erhältliches Tool das zur Vorverarbeitung und Identifikation von Metaboliten genutzt wird. Die folgende Abbildung illustriert den Programmablauf.

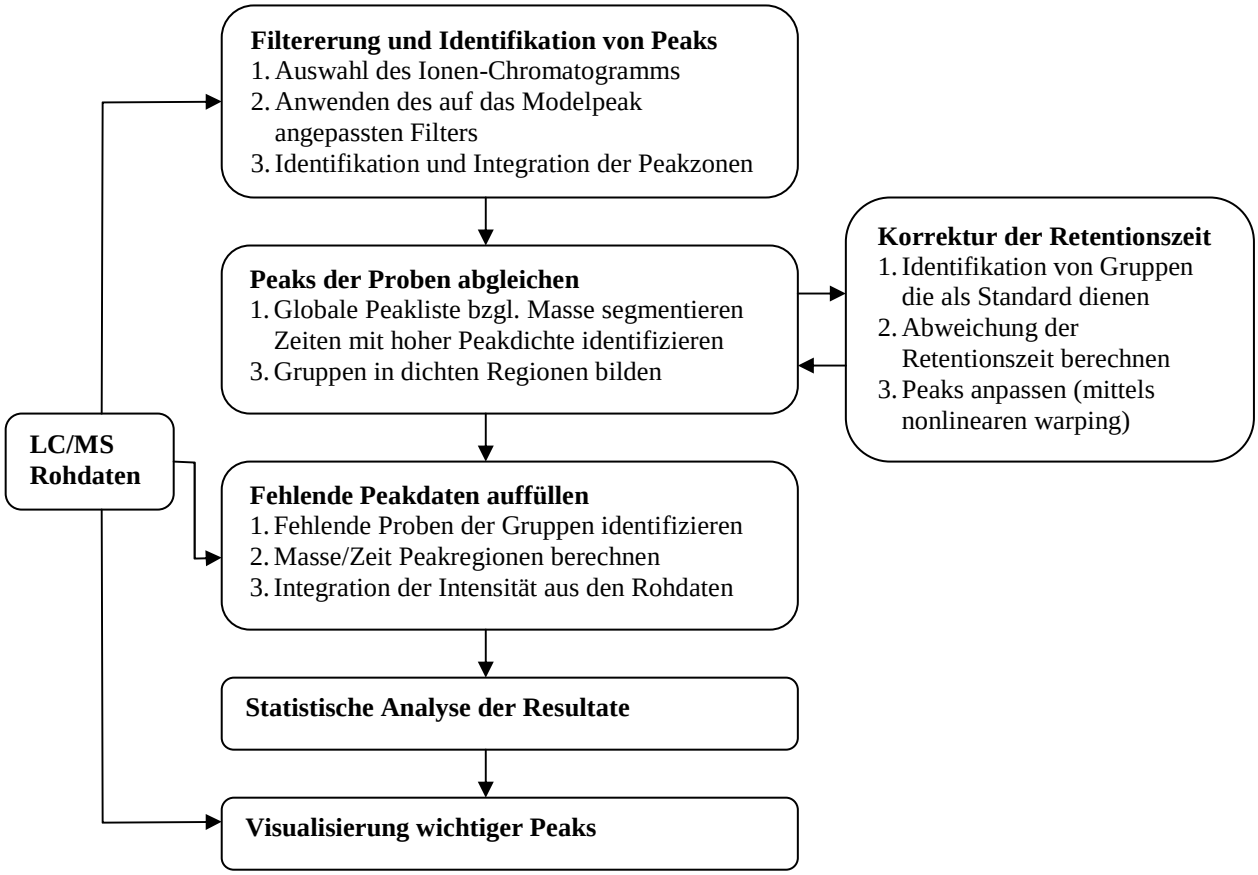


Abbildung 10: XCMS Programmablauf

Finden der Peaks

Zunächst wird das Spektrum in Teile aufgeschnitten die so groß sind wie eine Masseneinheit (0.1 m/z) ist. In diesen Teilen oder auch Slices genannt wird in jedem Zeitschritt das Signal mit der höchsten Intensität ermittelt. Danach wird auf jeden Slice ein matched Filter angewandt. Dieser Filter findet Templates in Signalen wieder. Das von XCMS verwendete Template ist eine Gaussfunktion zweiter Ordnung. Matched Filtering hat dabei den Vorteil, dass es die Signal-to-Noise-Ratio (SNR) maximiert. Die Peaks werden mittels einer Begrenzung des SNR bestimmt. Dabei hat sich bei Versuchen gezeigt, dass eine Grenze von zehn am effizientesten ist. Dem Benutzer steht aber frei, diese anders zu wählen. Zusätzlich ist pro Slice ein Limit von fünf Peaks definiert. Um die Weite der Peaks zu berechnen werden die Nullschwellenwerte verwendet die durch die

Verwendung der Gaussfunktion zweiter Ordnung entstehen. Wenn die Peakweite größer als die Slicegröße ist, kann ein Peak über mehrere Slices reichen. XCMS geht in einem Postprocessing-Schritt die Peakliste nach Intensität durch und löscht jene Peaks die in der Umgebung von 0.7 m/z von Peaks mit höherer Intensität liegen. Es kann auch sein, dass die Peakweite viel kleiner als die Sliceweite ist. In diesem Fall werden die Peaks von mehreren benachtbarten Slices einfach kombiniert um gleichmäßiger Peaks zu erhalten.

Peak Anpassung

Nachdem die Peaks in einzelnen Proben identifiziert wurden, müssen diese mit anderen verglichen werden um Abweichungen der Retentionszeit und relative Intensitätsvergleiche berechnen zu können. Danach werden Bins definiert die ein Intervall von 0.25 m/z abdecken. Diese überlagern sich jeweils um die Hälfte damit es nicht dazu kommen kann, dass Gruppen getrennt werden. Somit werden alle Peaks zweimal gezählt. Die doppelten Peaks werden aber in einem Postprocessing-Schritt wieder eliminiert. Danach wird die Verteilung aller Peaks über die gesamte Retentionszeit untersucht und Regionen gefunden in denen Peaks die selbe Retentionszeit haben. Mittels der Kernel-Density-Estimation ist es möglich Verteilungen abzuschätzen. Diese Methode erlaubt es Histogramme mittels anderer Funktionen zu glätten. Im Fall von XCMS wird eine Gaussfunktion verwendet. Aus dieser Verteilung werden sogenannte „Meta-Peaks“ extrahiert, welche mehrere Peaks mit der selben Retentionszeit darstellen. Diese Meta-Peaks, startend beim höchsten, werden dann links und rechts entlang gegangen, bis die Werte wieder steigen. Innerhalb dieser Grenzen werden alle Peaks einer Gruppe zugeordnet. Gruppen mit weniger Peaks als die Hälfte der Probenanzahl werden mangels Wichtigkeit eliminiert. Für den Intensitätsvergleich wird dann das Peak, welches am nähesten zum Mittelwert kommt, verwendet. Für die Anpassung der Retentionszeit jenes mit dem höchsten Wert.

Anpassung der Retentionszeit

Aufgrund der vorangegangenen Einteilung in Gruppen kann die Retentionszeit aller Proben gleichzeitig angepasst werden. Es werden sogenannte „well-behaved“ Gruppen ermittelt. Das sind jene Gruppen in denen sehr wenige Proben keine Peaks und auch sehr wenige mehr als ein Peak zuweisen. Diese

Gruppen haben eine hohe Wahrscheinlichkeit zusammenpassend zu sein und dienen als zeitweiliger Standard. Für jede Gruppe wird die durchschnittliche Retentionszeit berechnet und für jede Probe darin die Abweichung davon. Aufgrund der Normalverteilung der Gruppen im maßgeblichen Teil des Chromatogramms kann für jede Probe eine genaue Abweichungskontur der Retentionszeit erstellt werden. An jenen Stellen im Chromatogramm wo keine well-behaved Gruppen sind, wird interpoliert. Zu Beginn und zum Ende des Chromatogramms, sollten keine solche Gruppen sein, wird einfach eine Konstante verwendet. Mithilfe der so erstellten Abweichungskonturen werden die Retentionszeiten korrigiert. Danach werden dann nochmals Gruppen gebildet. Der ganze Vorgang der Retentionszeitkorrektur kann iterativ wiederholt werden. Vorteil dieser Methode ist, dass sie ohne vordefinierte Standards auskommt, da diese selbst erstellt werden. Jeder Iterationsschritt kann visualisiert werden und falls notwendig manuell nachgebessert. Desweiteren ist von Vorteil, dass nur Peakdaten notwendig sind. (Smith, C. A. und andere, 2006)

1.5.3 XCMS²

XCMS² ist eine Erweiterung zum ursprünglichen XCMS. Diese Version unterstützt daher alles was schon die vorige Version konnte. Zusätzlich bietet sie Funktionalität zur Suche von MSMS-Spektren in der Metlin-Datenbank. Desweiteren wurde eine „Similarity-Search“ entwickelt, welche eine strukturelle Klassifikation noch unbekannter Metaboliten aus MSMS-Daten vornehmen kann.

Spektrenvergleich

MS Spezialisten diskutieren noch wie man der Sicherheit, mit der man sagt, dass zwei Spektren identisch sind, einen Wert zuweisen kann. Die geläufigsten Methoden sind zur Zeit „shared peak count“ und „spectral convolution“. Beim ersten wird die Anzahl der Massen der Referenzspektren und der des Experimentes berechnet. Bei der zweiten wird durch Faltung der selben Spektren eine Differenzmatrix erstellt. Eine Differenz von Null zeigt Gleichheit an. XCMS² verwendet eine modifizierte Version der „shared peak count“-Methode. Zunächst werden aus einer indizierten Metlin-Datei alle Massen des Ausgangsstoffes gelöscht die nicht passen. Die Masse des Ausgangsstoffes wird precursor mass

genannt. Nach dieser Annäherung wird die Kollisionsenergie des MSMS-Spektrums verglichen. Falls keine gefunden wird, wird eine höhere verwendet. Dann werden Fragmentdaten mit einem Referenzspektrum verglichen. Dazu kann vom Benutzer ein Fehlerfenster in ppm (parts per million) für jedes Fragment definiert werden. Wenn die Massen außerhalb dieses Fenster liegen, wird das nächste Fragment verglichen. Für diese Operation werden eine Gleichheits- und eine Distanzmatrix berechnet. Jede Zelle in der Gleichheitsmatrix wird mittels (1) und jede in der Distanzmatrix mit (2) berechnet.

$$S_{ij} = \max(M_{i-1\,j}, M_{ij-1}, M_{i-1\,j-1}) - C \quad \textbf{(1)}$$

$$D_{ij} = C + \min(M_{i-ij}, M_{ij-1}, M_{i-1\,j-1}) \quad \textbf{(2)}$$

Der Wert, der mittels der Gleichheitsmatrix ermittelt wird, gibt an wie ähnlich sich zwei MSMS-Spektren sind. Sind die zwei Massen, die verglichen werden, gleich, ist C 0 ansonsten 1. Die Startwerte sind zu Beginn die maximale Fragmentanzahl beider Spektren welche auch die größte Gleichheit wäre. Die Gleichheit steht zum Schluss in der Zelle rechts unten. Die Distanzmatrix startet mit dem Wert 0 und gibt zum Schluss die Ähnlichkeit beider Spektren aus.
Sind beide Spektren exakt gleich gibt (3) an, wieviel Punkte erzielt werden würden. Sind beide komplett unterschiedlich ist die Punktanzahl laut Definition gleich 0 was (4) ausdrückt. Die prozentuelle Ähnlichkeit wird dann mittels (5) berechnet. Die Formel ist normalisiert was den Vergleich verschiedener Verbindungen erlaubt.

$$H = \min_{Length}(Exp, \mathrm{Re}\ f) - \underset{Length}{diff}\,(Exp, \mathrm{Re}\ f) \quad \textbf{(3)}$$

$$L = -\max_{Length}(Exp, \mathrm{Re}\ f) \quad \textbf{(4)}$$

$$S_{\%} = 100 \times \frac{(S - D) + L}{L - H} \quad \textbf{(5)}$$

Falls kein passendes Molekül gefunden wurde, ist es entweder nicht in der Datenbank oder das Fehlerfenster ist nicht groß genug gewählt. Die zusätzliche Verwendung der akkuraten Masse beschleunigt den Vorgang und bringt eine zusätzliche Bestätigung.

Similarity Search

Es wurde beobachtet, dass, auch wenn ein akkurater Treffer beim Spektrenvergleich auftritt, andere Verbindungen ebenfalls eine hohe Punktezahl erreichen. Diese haben Verwandtschaften in der Struktur was zu dieser Art der Suche inspiriert. Mit dem vorher beschriebenen Bewertungsschema für Verbindungen werden nun Spektren mittels Neutralverlusten oder Fragmentionen verglichen. Ein Unterschied besteht nur in der Berechnung mit Neutralverlusten. Hier wird die Differenz der Massen mit jedem benachtbarten Peak berechnet. Wenn eine hohe Gleichheit besteht, werden die Verbindungen in eine Tabelle geschrieben. Der Benutzer kann selbst bestimmen wie gleich die Verbindungen sein müssen um in die Tabelle aufgenommen zu werden. Die Entwickler empfehlen aber eine Gleichheit von mindestens 70%. (Benton, H. P. und andere, 2008)

1.5.4 Seven Golden Rules

Diese Regeln werden angewandt um die wahrscheinlichste chemische Formel für eine akkurate Masse zu finden. Dies bringt eine große Zeitersparnis, da bei größerer Masse auch die Anzahl der möglichen Massenformeln steigt. Zum Beispiel werden alle möglichen 8 Milliarden C, H, N, S, O, P Formeln auf 623 Millionen reduziert. (Kind, T. und Fiehn, O., 2007)

Regel 1 – Beschränkung der Anzahl der Elemente

Diese Regel spart Rechenzeit und Speicherplatz. Durch Betrachtung von Formeln die aus DNP (Dictionary of Natural Products) sind, entstehen die folgenden Zahlen.

Masse	C	H	N	O	P	S	F	Cl	Br
< 500	29	72	10	18	4	7	15	8	5
< 1000	66	126	25	27	6	8	16	11	8
< 2000	115	236	32	63	6	8	16	11	8
< 3000	162	208	48	78	6	9	16	11	8

Tabelle 6: Maximale Anzahl der jeweiligen Elemente bezüglich der Masse. Diese Werte wurden mittels Formeln aus DNP ermittelt.

Regel 2 – LEWIS und SENIOR Überprüfung

Die LEWIS Überprüfung verlangt, dass Moleküle die aus Elementen der Hauptgruppe, wie etwa Kohlenstoff, Stickstoff und Sauerstoff, bestehen volle s,p-Valenzschalen haben. Freie Radikale und hypervalente Moleküle würden dann aber nicht erfasst werden. Daher sollte die Möglichkeit bestehen diese Regel auszulassen. Die SENIOR Überprüfung besteht aus drei Regeln:

1. Die Summe der Valenzen oder die gesamte Anzahl der Atome mit ungeraden Valenzen ist gerade.
2. Die Summe der Valenzen ist größergleich der doppelten maximalen Valenz.
3. Die Summe der Valenzen ist größergleich der doppelten Anzahl der Atome weniger 1.

Regel 3 – Filterung isotopischer Muster

Mittels der Massenformel können isotopische Muster berechnet werden. Diese müssen dann verglichen werden. Die Isotopenhäufigkeit wird auf 100 normalisiert. Die Differenzen der berechneten und Zielintensitäten der Peaks werden berechnet und die Summe der Differenzen mit den Zielintensitäten verglichen. (Kind, T. und Fiehn, O., 2006)

Regel 4 – Überprüfung des Wasserstoff/Kohlenstoff-Verhältnisses

Das Überprüfen der Verhältnisse chemischer Elemente trägt ebenfalls zur Reduktion von Formeln bei. Sehr wichtig dabei ist das Wasserstoff/Kohlenstoff-Verhältnis. In den meisten Fällen übersteigt dieses nicht H/C > 3. Im Großteil der

Fälle liegt das Verhältnis sogar zwischen 2 und 0.125. Natürlich gibt es auch Ausnahmen wie etwa Methylhydrazin mit CH_6N_2.

Regel 5 – Überprüfung des Verhältnisses der Heteroatome

Heteroatome sind jene Atome die weder Kohlenstoff noch Wasserstoff sind. Diese Verhältnisse sind sogar noch hilfreicher als jene von H/C da viele Formeln gar kein Heteroatom besitzen.

Elemente	Übliches Verhältnis
F/C	0 - 1.5
Cl/C	0 - 0.8
Br/C	0 - 0.8
N/C	0 - 1.3
O/C	0 - 1.2
P/C	0 – 0.3
S/C	0 - 0.8
Si/C	0 - 0.5

Tabelle 7: Zeigt die üblichen Verhältnisses verschiedener Elemente zu Kohlenstoff an. Die Daten stammen aus 45.000 Formeln aus der Wiley mass spectral database für die Masse von 30 bis 1500 Da.

Regel 6 – Überprüfung der Wahrscheinlichkeit von Elementen

Diese Regel überprüft übermäßig große Elementanzahlen in Molekülen. Die Zahlen sind abgeleitet aus Daten von DNP, NIST08 und Wiley.

Elementenanzahl	Regel
NOPS alle > 1	N < 10, O < 20, P < 4, S < 3
NOP alle > 3	N < 11, O < 22, P < 6
OPS alle > 1	O < 14, P < 3, S < 3
PSN alle > 1	P < 3, S < 3, N < 4
NOS alle > 6	N < 19, O < 14, S < 8

Tabelle 8: Beschränkung von Elementen für Verbindungen bis 2000 Da.

Regel 7 – TMS Überprüfung

Dieser Schritt ist nur notwendig, wenn ein GC/MS-Gerät verwendet wurde. Da der Derivatisierungsprozesses oft Trimethylsilylierung (TMS) verwendet, werden säurehaltige Protonen mit TMS ausgetauscht. Um das nicht derivatisierte Molekül zu berechnen muss TMS wieder abgezogen werden.

1.6 Darstellung chemischer Strukturen

Die Darstellung chemischer Strukturen muss eindeutig sein. Es gibt mehrere Ansätze diese Strukturen darzustellen wie etwa mittels Diagrammen, Verbindungstabellen, aussprechbaren Namen oder Indexen. Grafische Darstellungen sind für den Benutzer leichter zu interpretieren, beinhalten aber überflüssige Information. Aussprechbare Namen vereinfachen die orale Kommunikation, die chemische Struktur dahinter wieder herzustellten ist jedoch schwierig beziehungsweise gar nicht möglich. Im folgenden Teil wird auf die Notationen SMILES und InChI eingegangen welche direkt aus der Struktur abgeleitet werden können. Ein Nachteil daran ist jedoch, dass die Länge mit der Komplexität der Struktur zunimmt.

1.6.1 SMILES – Simplified Molecular Input Line Entry Specification

SMILES ist ein chemisches Notationssystem welches von Daylight Chemical Information Systems entwickelt wurde (SMILES, 2009) und es erlaubt chemische Strukturen mit Hilfe einer sehr einfachen Grammatik auszudrücken. Zusätzlich ist es für eine schnelle, maschinelle Verarbeitung geeignet. Darunter fällt etwa die Erzeugung eindeutiger Notationen, Datenbankabfragen mit konstanter Geschwindigkeit oder die Suche nach Substrukturen. SMILES hat folgende, grundlegende Ziele:

1. Der Graph einer chemischen Struktur soll eindeutig beschrieben werden.
2. Eine benutzerfreundliche Struktur soll bereitgestellt werden, deren Regeln schnell erlernt werden können.

3. Ein maschinenfreundliches und –unabhängiges System zur Interpretation und Erzeugung eindeutiger Notationen soll konzipiert werden.

Es ist zu beachten, dass es für eine Struktur mehrere mögliche SMILES-Notationen geben kann, jedoch eine Notation genau eine Struktur beschreibt. SMILES geht grundsätzlich von einer zweidimensionalen Darstellung aus und übersetzt diese mittels folgender Regeln:

Atome

Atome werden durch ihre üblichen Symbole dargestellt. Etwaige zweite Buchstaben müssen dabei klein geschrieben werden. Elemente aus der organischen Untergruppe (B, C, N, O, P, S, F, Cl, Br, I) können ohne eckige Klammern geschrieben werden. Ein Beispiel wäre etwa die Notation C für Methan (CH_4). Elemente die nicht in der organischen Untergruppe enthalten sind, müssen in eckigen Klammern geschrieben werden wie zum Beispiel Gold [Au]. Angehängte Wasserstoffe werden mittels H, Ladungen mittels + oder - gefolgt von einer optionalen Zahl angegeben. Alles erfolgt innerhalb von eckigen Klammern wie bei folgenden:

[H+]	Proton
[OH3+]	Hydronium Kation
[Fe+2] oder [Fe++]	Eisen(II) Kation

Bindungen

Einfache, zweifache, dreifache und aromatische Bindungen werden mittels -, =, # und : angegeben. Einfache und aromatische können ganz weggelassen werden. Beispiele:

CC	Ethan(CH_3CH_3)
C=C	Ethylen ($CH_2=CH_2$)
C#N	Blausäure (HCN)

Abzweigungen

Abzweigungen werden mittels runden Klammern angezeigt wie Abbildung 11 zeigt. Abbildung 12 zeigt eine tiefere Verschachtelung der Klammern was ebenfalls möglich ist.

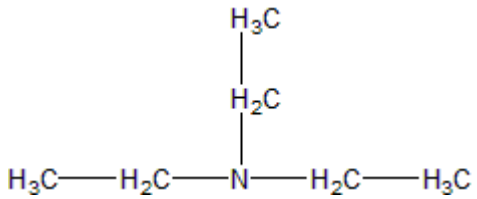


Abbildung 11: Triethylamin mit dem SMILES: CCN(CC)CC

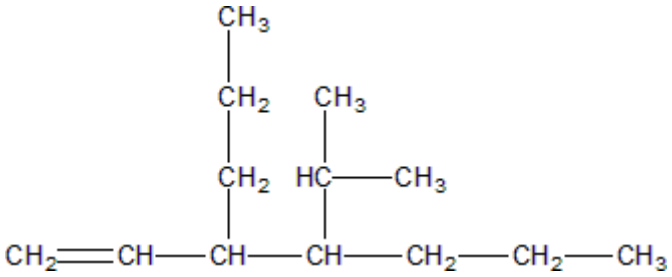


Abbildung 12: 3-Propyl-4-Isopropyl-1-Hepten mit dem SMILES: C=CC(CCC)C(C(C)C)CCC

Kreisförmige Strukturen

Diese werden dargestellt indem der Ring auf einer Stelle gebrochen wird und die gelöschte Bindung eine Zahl erhält. Die betroffenen Randstücke erhalten ebenfalls diese Zahl. Das ergibt wieder einen nichtzyklischen Graphen der mit den vorigen Regeln beschrieben werden kann.

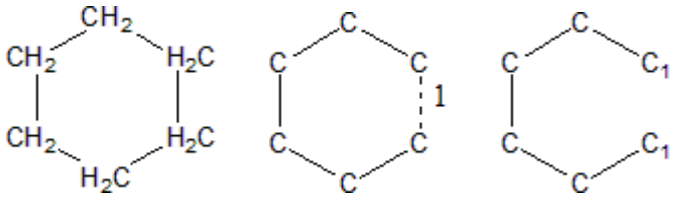


Abbildung 13: Schritte der Umwandlung von Cyclohexan . Von der kompletten Darstellung, über das Weglassen der Wasserstoffe bis hin zum Öffnen der Kreisstruktur. Der entstehende SMILES lautet C1CCCC1.

Einzelne Atome können auch mehr als eine Ringverbindung haben. Wie etwa beim Beispiel von Kuban gezeigt werden kann.

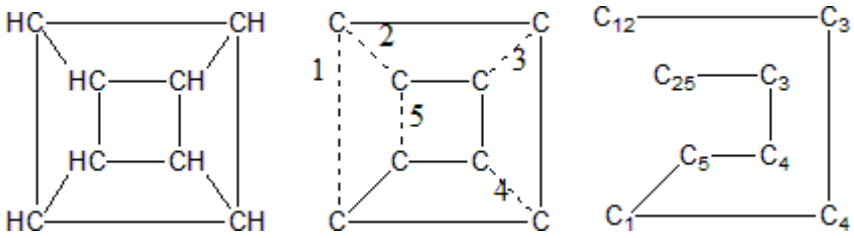


Abbildung 14: Umwandlung von Kuban in einen SMILES C12C3C4C1C5C4C3C25. Im zweiten Schritt sieht man welche Bindungen gekappt werden. Die dritte Abbildung zeigt die Bezifferung der Atome.

Falls gewünscht können die Zahlen bereits geschlossener Ringe für andere Ringe wieder verwendet werden.

Nicht verbundene Strukturen

Strukturen die nicht verbunden sind werden separat geschrieben und durch einen Punkt getrennt. Dabei muss bei der Reihenfolge keine Rücksicht auf etwaige Ladungen genommen werden. Die Gesamtladungen muss ebenfalls nicht neutral sein. Man kann auch den SMILES einer Verbindung beliebig in einen anderen einbauen wie die Abbildung 15 zeigt.

Aromatische Verbindungen

Die Atome in aromatischen Verbindungen werden mit Kleinbuchstaben geschrieben. Ein Beispiel dafür ist in der folgenden Abbildung zu sehen.

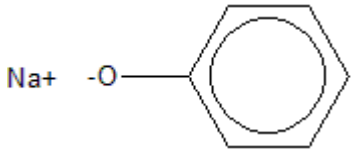


Abbildung 15: Sodiumphenoxyd als Beispiel für eine nichtverbundene Struktur. Es sind die SMILES [Na+].[O-]c1ccccc1 und c1cc([O-].[Na+])ccc1 möglich

Die Beispiele sowie die Beschreibung von SMILES stammen aus (Weininger, David, 1988).

1.6.2 InChI - International Chemical Identifier

InChI wurde von IUPAC in Kooperation mit NIST mit den folgenden Zielen entwickelt:

- Der Identifikator kann mittels eines Algorithmus aus der Struktur abgeleitet werden.
- Er akzeptiert gebräuchliche Abbildungskonventionen.
- Es gibt genau einen Identifikator pro Struktur.
- Er ist für jeden frei verfügbar.

Um verschiedene Detailierungsgrade zu erreichen und erweiterbar zu sein, ist InChI mit Hilfe von bis zu fünf verschiedener Ebenen aufgebaut die jeweils eigene Startsymbole besitzen:

1. Main Layer /C
2. Charge Layer /p
3. Stereochemical Layer /s
4. Isotopic Layer /i
5. Fixed-H Layer /f

Der Main Layer besteht wiederum aus den Ebenen Massenformel, Atomverbindungen (/c) und Wasserstoffatome (/h). Die Massenformel ist der einzige Layer der kein eigenes Startsymbol hat, da diese Ebene sowieso am Anfang des Main Layers steht. Im Charge Layer steht eine etwaige Ladung der Verbindung. Der Stereochemical Layer besteht aus zwei Sublayern. Eine ist zuständig für Doppelbindungen und die andere für tetraedrische Stereochemie. Im Isotopic Layer werden als isotopisch gekennzeichnete Atome festgehalten. Der Fixed-H Layer beinhaltet die Position mobiler Wasserstoffatome. Je nach Information die am Start des Algorithmus vorliegen, können Layer im InChI vorhanden sein oder nicht. Einzig der Main Layer muss enthalten sein. Um aus einer Struktur einen InChI generieren zu können sind einige Schritte notwendig. (InChI, 2009)(IUPAC InChI, 2009)

Normalisierung

Bei der Normalisierung werden Bindungen und Ladungen ignoriert. Falls die Struktur komplexer ist, können noch die folgenden 5 Normalisierungsschritte hinzu kommen:

1. Salze abtrennen
2. Metalle abtrennen
3. Wenn möglich Radikale eliminieren
4. Variable Protonierung berechnen
5. Ladungen und mobile Wasserstoffatome berechnen

Kanonisierung

Bei der Kanonisierung werden den gleichartigen Atomen in der Struktur die gleichen Nummern zugewiesen und danach nochmal nach diesen Nummern aufsteigend jedes Atom einzeln durchnummeriert.

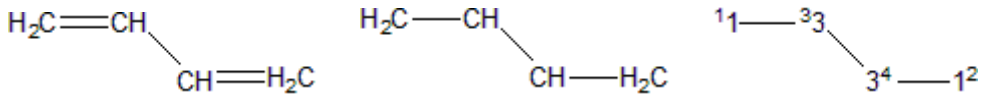


Abbildung 16: 1-3, Butadien von der Ausgangsstruktur über Normalisierung hin zur kanonisch nummerierten Darstellung mit dem InChI: InChI=1/C4H6/c1-3-4-2/h3-4H,1-2H2.

Da InChI eindeutig ist, es aber mehrere SMILES für eine Struktur geben kann, ist InChI als allgemeiner Identifikator besser geeignet.

2 Methoden

Dieses Kapitel beschreibt verwendete Programmiersprachen und Tools die zur Erstellung der Diplomarbeit benützt wurden.

2.1 Java

Zur Entwicklung der GUI (Graphical User Interface) und der Verarbeitung der Daten bzw. Einbindung von MySQL und R wird Java Version 1.6.0_17 (Java, 2010) verwendet, als Programmierumgebung dient Eclipse Version 3.4.1 (Eclipse, 2010). Eclipse verfügt über Standardfunktionalität wie Syntaxhighligthing und gibt Vorschläge für Klassenimport. Von Vorteil ist ebenfalls, dass Makefiles nicht vom Benutzer erstellt werden müssen. Standardmethoden wie Getter- und Setter-Methoden können auch automatisch generiert werden.

2.2 SWING

SWING (Java-Swing, 2010) wurde von Sun Microsystems für Java entwickelt und ist eine Programminterface sowie eine Grafikbibliothek zum erstellen grafischer Oberflächen von Anwendungen. Ein großer Vorteil von SWING-Grafikkomponenten ist, dass sie direkt von Java gerendert werden und somit komplett plattformunabhängig sind.
Grundlage jeder grafischen Oberfläche ist normalerweise ein JFrame in das dann weitere Komponenten eingebettet werden können. Die Position dieser wird mittels einem der zahlreichen LayoutManagern definiert.

2.3 MySQL

MySQL (MySQL, 2010) ist eine relationales Datenbanksystem das unter der GNU General Public License (GPL) verfügbar ist. Mittels MySQL ist es möglich mehrere Datenbanken zu verwalten die wiederum mehrere Tabellen beinhalten können. Die Tabellen wiederum besitzen Spalten mit verschiedenen Namen und

Datentypen. Die einzelnen Zeilen in der Tabelle entsprechen den gespeicherten Datensätzen. Beim Erstellen einer relationalen Datenbank ist auf die Einhaltung der Normalformen zu achten. Eine relative einfache Erklärung ist unter (SQL-Normalformen, 2010) zu finden.
Die Datenbank für die Diplomarbeit wurde mit MySQL Version 14.14 Distribution 5.1.41 erstellt welches im Xampp (Xampp, 2010) Paket enthalten ist. Als Tabellenformat wurde InnoDb verwendet da es bei diesem Typen durch die Hilfe von Fremdschlüssel zu keiner Verletzungen der referenziellen Integrität kommen kann.

2.4 R

R (R, 2010) ist eine frei verfügbare Programmiersprache mit der vor allem statistische Berechnungen durchgeführt werden können. Für die Auswertung von Daten aus dem Massenspektrometer ist das XCMS-Paket sehr gut geeignet. Die Diplomarbeitet verwendet die R-Version 2.10.1.

2.5 ReadW

ReadW ist eine Software die Rohdaten aus Massenspektrometern in mzXML-Dateien konvertiert. Damit ReadW funktioniert muss die Xcalibur-Software von ThermoFinnigan installiert sein, da die XRawfile.dll-Bibliothek verwendet wird. Aufgerufen wird das Programm in der Eingabeaufforderung mit beispielsweise dem Befehl:

ReadW –c Testmix_01.RAW C_Stock_1c.mzXML

2.6 JFreeChart

JFreeChart (JFreeChart, 2010) ist eine OpenSource Software die es erlaubt Diagramme mit einfachen Mitteln zu generieren. Sie wird unter der GNU Lesser General Public Licence (LGPL) (LGPL, 2010) verbreitet welche auch den Einsatz in proprietärer Software gestattet.

Zusätzlich zur grafischen Generierung der Diagramme, besitzt jedes Diagramm ein Popup-Menü, das es erlaubt Einstellungen zu ändern, zu kopieren, zu speichern, zu drucken oder zu zoomen. Das Zoomen kann aber auch mit der Maus erfolgen indem man den zu vergrößernden Bereich mit gedrückter Maustaste von links oben nach rechts unten markiert. Herauszoomen erfolgt in die andere Richtung.

3 Design

In diesem Teil wird das Grundgerüst der MySQL-Datenbank für die Diplomarbeit beschrieben sowie der Aufbau des Programmes zur Verwaltung dieser und Finden der Metaboliten.

3.1 Programmaufbau

Das Programm ist in die Pakete GUI, JoanneumDatabase, OtherDatabases, Testing und XcmsInterface aufgeteilt. Einstiegspunkt bei Programmstart ist die Klasse StartGUI die mit dem Parameter -Xmx512m ausgeführt werden sollte damit genügend Speicher vorhanden ist.

3.1.1 GUI-Paket

Dieses Paket enthält alle grafischen Oberflächen. Desweiteren ist das Paket Actions enthalten in dem wiederum alle Oberflächen sind, die bei Aktionen aus einem Popup-Menü erscheinen. Zusätzlich besitzt dieses Paket noch das Pakete MenuDatabase mit den Oberflächen für Aktionen aus dem Menü sowie das Paket MouseListeners welches Klassen enthält, die MouseEvents verarbeiten.

3.1.2 JoanneumDatabase-Paket

Im JoanneumDatabase-Paket sind Klassen enthälten, die für die Ver- und Bearbeitung von Daten erforderlich sind aber nicht wirklich unter die Actions im GUI-Paket fallen. Wichtig in diesem Paket sind vorallem die Pakete zur Datenbankeinbindung. Diese wären:

- DatabaseElement
  Enthält Klassen zu allen Tabellen in der Datenbank. Diese bestehen meist nur aus Variablen die den Spalten der Tabellen entsprechen und entsprechenden Getter- und Setter-Methoden.

- Deletes
  Enthält Klassen, die Einträge aus den Datenbanktabellen löschen.
- Inserts
  Enthält Klassen, die Einträge in die Datenbanktabellen hinzufügen.
- Selects
  Enthält Klassen, die Einträge aus den Datenbanktabellen auslesen.
- Updates
  Enthält Klassen, die Einträge in den Datenbanktabellen verändern.

3.1.3 OtherDatabases-Paket

In diesem Paket wird der Zugriff auf andere Datenbanken geregelt. Zurzeit werden nur die offline verfügbar gemachten Datenbanken MetaCyc und ChEBI verwendet.

3.1.4 Testing-Paket

Ein Paket in dem Testfälle für einzelne Klassen enthälten sind. Verfügt zusätzlich über Pakete InsertTesting, SelectTesting und UpdateTesting das einige Testfälle für den Datenbankzugriff auf die Joanneum-Datenbank enthält.

3.1.5 XcmsInterface-Paket

Beinhaltet die Klassen RserveStarter, welches einen R-Process startet, RserveStopper welches einen R-Process wieder beendet sowie XcmsStarter. Die letzte Klasse enthält Methoden zum Steuern von R sowie zum Beziehen von Daten aus R.

3.2 MySQL-*Datenbank*

Die Datenbankstruktur ist in Abbildung 17 zu sehen. Unterteilt werden kann die Datenbank in die Gruppen CompoundData, SpectraData und Hilfstabellen.

CompoundData

CompoundData enthält direkte Informationen über Metaboliten. Hier wurden jene Felder verwendet, die durch die bereits beschriebenen, schon existierenden Datenbanken, als wichtig empfunden wurden. Wie die meisten anderen, speichert auch die Joanneum Datenbank die Masse ab und enthält, InCHI, Smiles, Synonyme und Verlinkungen zu anderen Datenbanken. Zusätzlich zur Masse werden Retentionszeiten abgespeichert. Diese Zeiten sind abhängig von der verwendeten LC-Methode und ermöglichen eine exaktere Identifikation als dies bei bisherigen Datenbanken möglich war.

SpectraData

Im Teil SpectraData werden Informationen gespeichert die aus dem Massenspektrometer gewonnen werden. Es werden Metadaten zu dem Gerät gespeichert und Informationen über die einzelnen Analysen. Desweiteren werden natürlich alle Peakinformationen und die resultierenden Featuredaten abgelegt. Ebenso gespeichert werden eventuelle MSMS-Spektren. Diese werden zunächst nur dem jeweiligen Parentpeak zugeordnet. Da Features aus mehreren Peaks berechnet werden und somit auch mehrere MSMS-Spektrum möglich sind, wird jenes verwendet, dass die höchste Summe der Intensitäten aufweist. FeaturePeaklist enthält die Zuordnung der Peaks zu den Features. Dies ist notwendig um die erste Normalform zu gewährleisten, da es im voraus nicht absehbar ist aus wievielen Peaks ein Feature berechnet wird. Die Tabelle ExperimentData enthält eine Id als Primärschlüssel und das verwendete Arbeitsverzeichnis aus dem die mzXML-Dateien stammen. In dem Arbeitsverzeichnis werden im Zuge der Berechnung der Peaks und der Features auch die R-Datei mit den verwendeten XCMS-Einstellungen und der R-workspace abgelegt um die spätere Nachvollziehbarkeit zu gewährleisten. Zum Dateinamen wird die ExperimentId hinzugefügt, da es möglich sein kann mit verschiedenen XCMS-Einstellungen ein und das selbe Experiment aufzunehmen.

Hilfstabellen

Zusätzlich zu den Teilen CompoundData und SpectraData sind die Tabellen User, XcmsSettings, Adducts, CompoundFeatureList und CompoundMsmsList

vorhanden, welche die Hilfstabellen darstellen. Die Usertabelle enthält Informationen über Größe und Position der GUI sowie die Abweichungsgrenze (Ppm) die bei der Berechnung für die Zuordnung von Features und Metaboliten verwendet wird. XcmsSettings enthält den Pfad zur akutell verwendeten Xcms-Konfigurationsdatei. Ebenfalls essentiell für die Berechnung der Zuordnung ist die Addukttabelle. Die Zuordnung selbst wird in der CompoundFeatureList festgehalten. Die Verbindung von Metaboliten zu MSMS-Daten erfolgt über CompoundMsmsList.

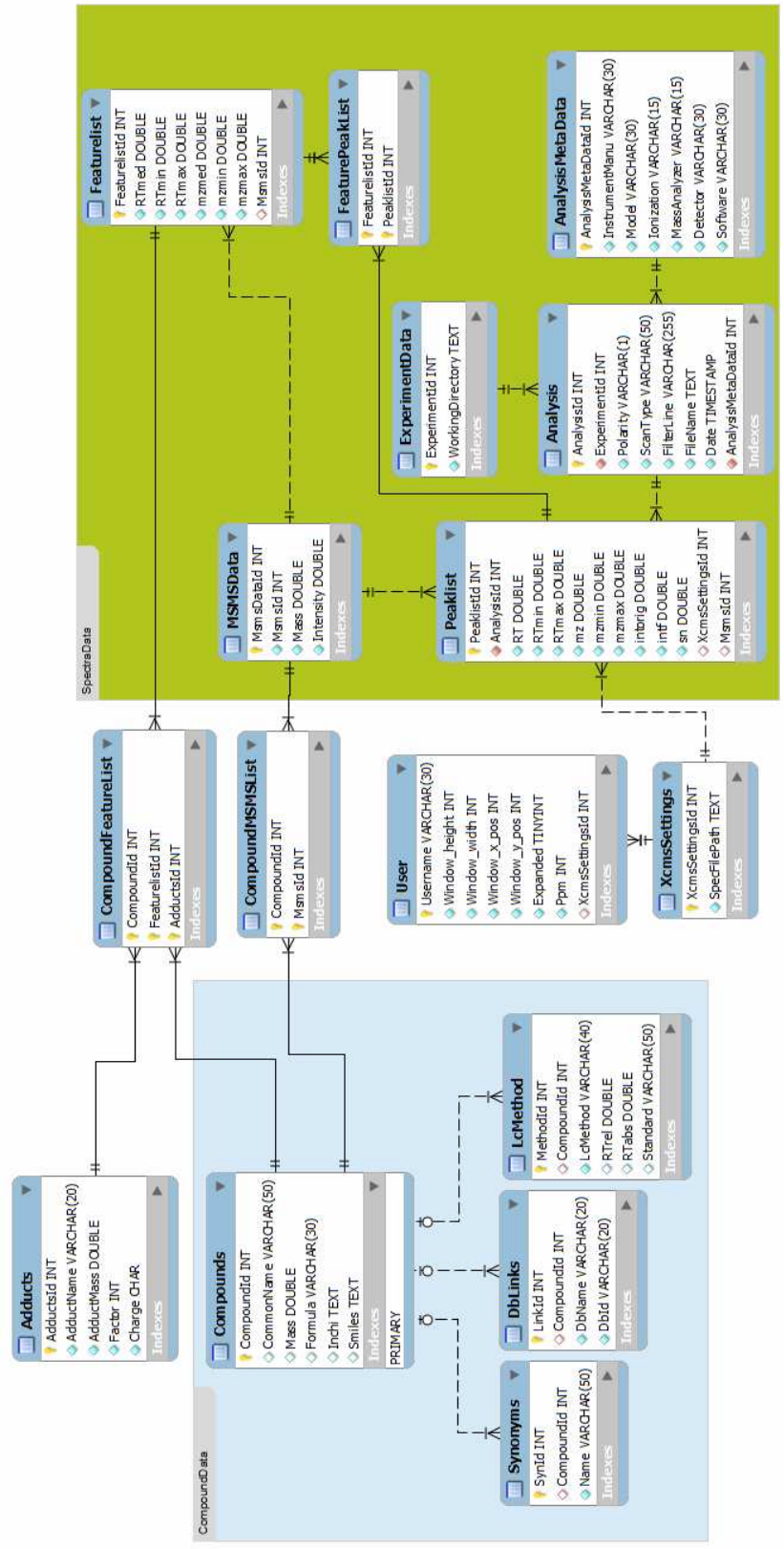


Abbildung 17: Datenbankstruktur

4 Implementierung

Der Implementierungsteil beschreibt die programmtechnische Umsetzung der im Designkapitel beschriebenen Pakete. Wie erwähnt dient als Backend eine MySQL Datenbank, das Frontend ist mit Java erstellt. Die Anpassung der Retentionszeitverschiebungen und das Peakalignment werden mit dem R Paket XCMS durchgeführt wobei R mit Hilfe von R-Serve in Java eingebunden ist.

4.1 GUI – Graphical User Interface

Die grafische Oberfläche wurde mittels SWING programmiert. Abbildung 18 zeigt die GUI welche in diesem Fall Experimente anzeigt. Die Oberfläche ist in drei Abschnitte Menü, DisplayPanel und InfoPanel eingeteilt. DisplayPanel und InfoPanel sind in ein JSplitPane eingebettet. Dieses ist dem ContentPane der MainWindow-Klasse hinzugefügt.

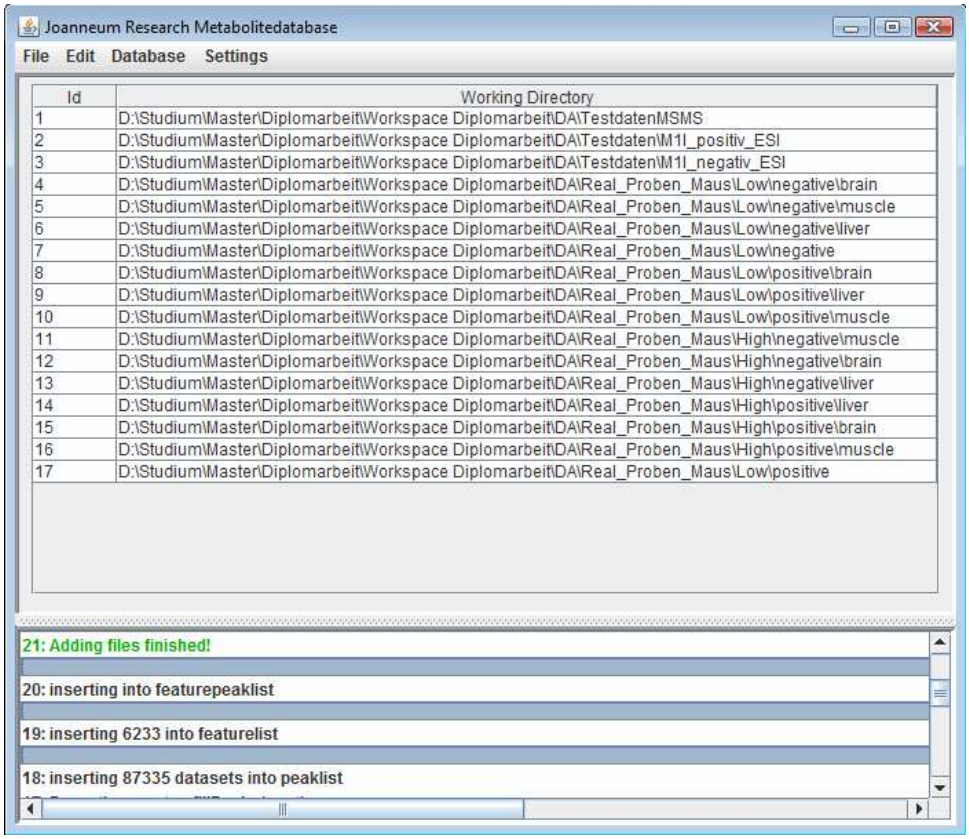


Abbildung 18: Anzeige von Experimenten in der GUI

4.1.1 Menü

Die Klasse Menu ist abgeleitet von JMenuBar und enthält die Menüs File, Edit, Database und Settings welche ansich JMenus sind. Die Menüpunkte sind wiederum mit JItems gefüllt welche durch anklicken eine bestimmte Funktionalität aufrufen.

File:

- Clear:
  Löscht alle Komponenten aus dem DisplayPanel.
- Close:
  Schließt die Anwendung

Edit:

- Add Experiment:
  Fügt Daten eines Experimentes zur Datenbank hinzu. Dazu muss mittels eines JFileChoosers das Arbeitsverzeichnis ausgewählt werden. Falls das darin enthaltene Experiment mit den gleichen XCMS-Einstellungen bereits in die Datenbank aufgenommen wurde, wird eine Fehlermeldung angezeigt. Falls das nicht der Fall ist, werden alle mzXML-Dateien aus dem Arbeitsverzeichnis und allen Unterordnern in die Datenbank aufgenommen. Als erstes wird in der ExperimentData-Tabelle ein neues Experiment angelegt. Danach werden Metadaten und Daten zur Analyse angelegt. Diese werden direkt aus den mzXML-Dateien bezogen. Zum Parsen der Informationen aus den Dateien wird der SAX-Parser (SAX, 2010) verwendet. Anschließend werden Peak-, Feature-, und MSMS-Informationen verarbeitet was mit Hilfe von R geschieht und dem R-Paket XCMS. Wie die Einbindung von R funktioniert, wird später erklärt. Mittels eines R-Skriptes, dass man über XCMS-Settings auswählen kann, wird XCMS gesagt, wie die Verarbeitung der Daten ablaufen soll. Falls das xcmsSet in R den Namen xset hat, erfolgt der Zugriff auf die darin enthaltenen Peaks in R mittels xset@peaks. Rserve ermöglicht den Zugriff mittels
  connection.eval(„xset@peaks“).asDoubleMatrix();

das auch gleich eine einfach zu verarbeitende Matrix mit Double-Werten liefert. Featuredaten können equivalent mittels xset@groups bezogen werden. Die Zuteilung der Peaks zu den Features ist in xset@groupidx gespeichert welches aber eine R-Liste zurück gibt und keine Matrix. Jedes Element dieser Liste besteht aus einem Integerarray welche die Ids der Peaks enthält. Der Index des Elementes ist gleichzeitig die FeatureId. Es werden zuerst die Peakinformationen, dann die Featuredaten und zum Schluss die Zuteilung gespeichert, da es sonst zu Problemen mit den Fremdschlüsseln kommen kann. Die Retentionszeiten bei Peak und Featuredaten werden durch 60 dividiert, also in Minuten umgerechnet. Dies geschieht, da diese Darstellung für den Benutzer angenehmer ist.

Zum Schluss werden noch die MSMS-Daten bezogen. In der aktuellen XCMS-Version sind diese leider noch nicht direkt im xcmsSet enthalten. Daher muss man den Umweg über *frags <- xcmsFragments(xset)* gehen, was auch schon im R-Skript enthalten sein muss. Die Übertragung der Daten nach Java erfolgt dann mit

connection.eval(„frags@peaks“).asDoubleMatrix();

Danach werden allerdings nur jene Peaks verwendet, die in der MS-Level-Spalte eine Zwei stehen haben. Die Zuteilung zu den Peaks erfolgt über m/z und Retentionszeit des Elternpeaks. Sind die MSMS-Spektren den Peaks zugeteilt, erfolgt die Zuordnung zu den Features. Um dem Benutzer über den Fortschritt des Hinzufügens auf dem laufenden zu halten, wird dem InfoPanel eine JProgressBar hinzugefügt, die den Fortschritt darstellt.

Zum Abschluss wird ein JFrame angezeigt, in welchem mögliche Zuteilungen von Metaboliten, Features und Addukten widergibt. Die Funktionsweise wird später beschrieben.

- Search:

  Search dient dazu, nach Metaboliten in der Joanneum Datenbank zu suchen. Abbildung 19 zeigt die möglichen Suchparameter. In allen Felder exklusive Mass und Tolerance ist es möglich die MySQL Wildcards % und _ zu verwenden. Wildcards sind Platzhalter wobei _ in MySQL für genau einen Charakter steht und % für beliebig viele. So würde etwa To_ die Wörter Tor oder Tom matchen, nicht jedoch Tore weil _ nur für einen Charakter steht. To% würde auch Tor aber auch Tore matchen oder

jegliches andere Wort das mit To anfängt. Gibt man in der Suchmaske bei Name nur % ein, werden also alle Metaboliten ausgegeben. Um Wildcards mit MySQL verwenden zu können, ist darauf zu achten den Vergleich mittels LIKE und nicht = zu verwenden. Weiß man die Masse eines Metaboliten nicht genau, ist es möglich auch noch eine Toleranz anzugeben. Ein Beispielhaftes Suchergebnis wird in Abbildung 20 gezeicht. Das Feld für DbLinks ist dabei erweitert, alle anderen sind nicht erweitert. Die Standardeinstellung ist der Punkt expanded unter den Benutzereinstellungen. Synonyme werden einfach mittels Strichpunkt getrennt. Die Lupe bei den Features berechnet mögliche Zuordnungen von Features und Addukten zu dem jeweiligen Metaboliten. InCHI und SMILES werden nur angezeigt, falls in der Datenbank auch vorhanden. Dann erscheinen sie rechts von Mass bzw. Formula. Das gesamte Suchergebnis wird im DisplayPanel angezeigt.

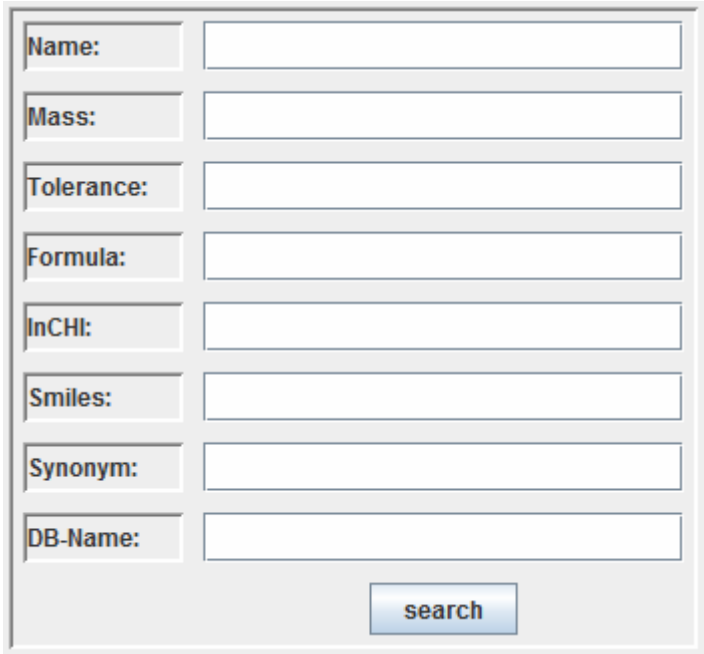


Abbildung 19: Suche nach Metaboliten

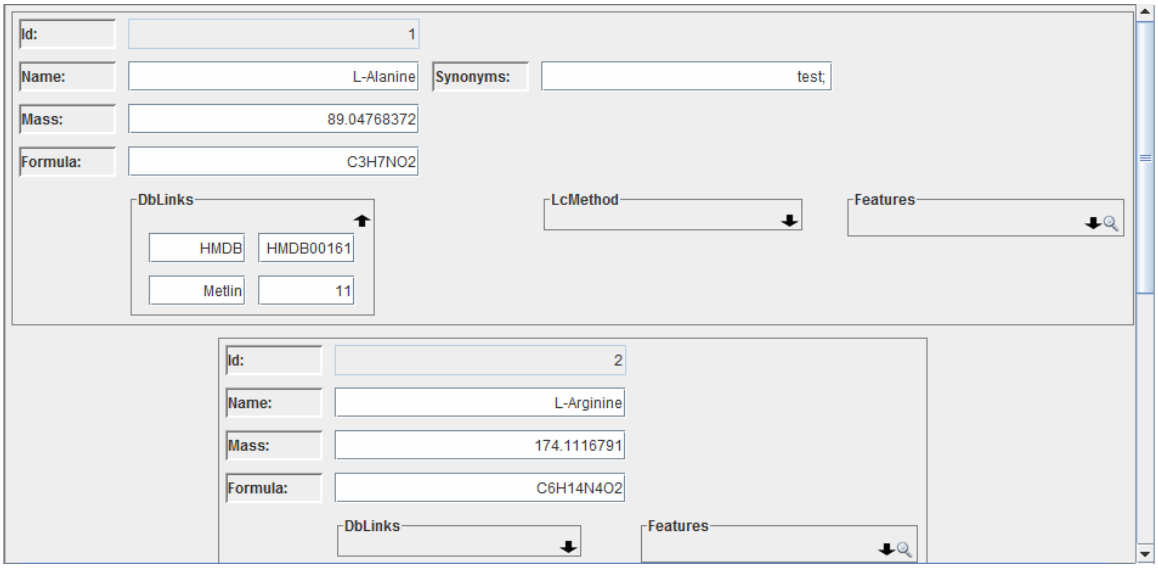


Abbildung 20: Suchergebnis

Database:

Einige der Klassen die hier zur Anzeige von Daten verwendet werden können die Daten, die sie darstellen sollen selbst aus der Datenbank holen oder sie können mittels einer Liste von Elementen konstruiert werden und dann diese anzeigen. Desweiteren enthalten alle dargestellten Tabellen eigene Integer- und Double-Comparator-Klassen. Wird nur der Standard-Sorter verwendet würde 10 vor 2 kommen da nach Strings sortiert werden würde.

- Show Analysis:
  Zeigt eine Tabelle mit allen Analysedaten an. Die zugehörigen Metadaten des Instrumentes können ebenfalls angezeigt werden.
- Show Compounds:
  Zeigt eine Tabelle mit Id, Name, Masse, Formel, InCHI und SMILES an. Zusätzliche Informationen wie Synonyme, Links zu anderen Datenbanken und LC-Methode können mittels eines Popup-Menüs angezeigt werden. Ist eine Zuteilung zu einem Feature vorhanden, wird die Reihe blau eingefärbt und man kann die Featureinformationen anzeigen lassen. Sind auch MSMS-Spektren kann auch davon ein Diagramm erstellt werden. Das Popup-Menü enthält außerdem ein Edit-Menü in welchen man alle Felder exklusive der Id ändern kann.

- Show Experiments:
  Dieses Menü dient dazu alle in der Datenbank enthaltenen Experimente anzeigen zu lassen. Gezeigt wird die Id des Experimentes und das verwendete Arbeitsverzeichnis. Mittels eines Popup-Menüs kann man einzelne Experimente löschen oder nochmals die Berechnung für die Zuordnung von Features und Metaboliten durchführen lassen. Wie diese genau funktionieren, wird unter dem Punkt Funktionalitäten präziser beschrieben.
- Show Features:
  Auch hier wird eine Tabelle angezeigt. Sie enthält die Spalten Id, RtMed, RtMin, RtMax, MzMed, MzMin, MzMax, AvgInt und MSMS-Id. AvgInt wird extra berechnet und enthält die durchschnittliche Intensität der Peakintensitäten des jeweiligen Features. Es werden zu Beginn jedoch nur die Daten des letzten Experimentes angezeigt. Features die eine Zuteilung in der CompoundFeatureList haben, werden blau eingefärbt. Zu diesen Features können dann auch die Metabolitinformationen ausgegeben werden. Falls man sich sicher ist, dass ein Feature einen Metabolite zuzurechnen ist, über diesen aber noch keine dementsprechende Forschung vorhanden ist, kann man einen unbekannten Metabolite hinzufügen. Als Addukt für die Zuteilung wird übergangsweise eine undefiniertes mit der Id 8 verwendet. Falls mehr über diesen Metaboliten bekannt wird, kann man dann später Namen usw. ändern. Falls MSMS-Spektren vorhanden sind, kann auch davon ein Diagramm erstellt werden.
  Um die Übersichtlichkeit zu verbessern, wird in den Spalten mit Double-Werten nicht die komplette Genauigkeit ausgenutzt. Vielmehr werden die Spalten mit Retentionszeiten auf zwei Kommastellen und jene mit m/z-Werten auf vier Kommastellen beschränkt. Bei der durchschnittlichen Intensität ist gar keine Kommastelle notwendig.
  Über das Popup-Menü kann außerdem ein Filter, der in Abbildung 21 zu sehn ist, angezeigt werden. Dieser ermöglicht es nach Experiment zu filtern oder RtMed und MzMed einzuschränken. Der Filter bietet auch die Möglichkeit, nach mehreren Experimenten auszusortieren. Dazu müssen die einzelnen Ids der Experimente mit einem Strichpunkt getrennt werden,

wie zB. „1;2;3“. Die Filterung wird beim Klicken auf den filter-Button ausgelöst aber auch bei Bestätigung einer Eingabe in einem Textfeld mittels Enter. Die Felder werden auf eine korrekte Zahleneingabe überprüft und bei nicht korrekten Eingaben wird eine Fehlermeldung mittels eines JOptionPane ausgegeben.

- Show Peaks:
  Diese Tabelle ist der Show Features sehr ähnlich. Anstelle der durchschnittlichen Intensität wird jedoch die originale Intensität angezeigt. Zusätlich gibt es Spalten für AnalyseId, Settings und Signal-To-Noise-Ratio. Es ist möglich die zugrundeliegende Analyse, Einstellung oder das resultierende Feature anzeigen zu lassen. Desweiteren hat Show Peaks die gleiche Filterfunktionalität wie Show Features.

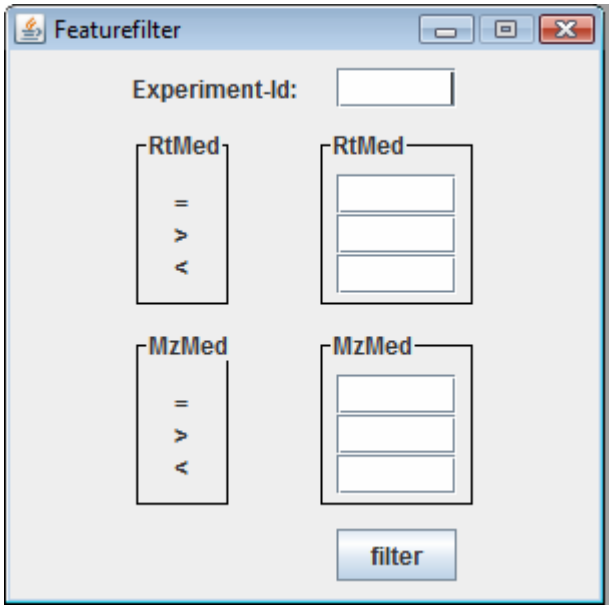


Abbildung 21: Filter

- Show Settings:
  Show Settings zeigt alle in der Datenbank gespeicherten Einstellungen an. Diese bestehen aus einer Id und dem Pfad zu einer XCMS-Spezifikationsdatei. Die Einstellung, die vom Benutzer gerade in Verwendung ist, wird grün angezeigt. Es ist nicht möglich diese zu löschen. Jede andere kann jedoch gelöscht werden. Außerdem ist es

möglich eine andere Einstellung aus der Liste auszuwählen. Falls keine passend ist, keine eine neue hinzugefügt werden. Dafür muss man lediglich ein R-Skript auswählen, was wiederum mittels eines JFileChoosers bewerkstelligt wird.

Settings:

- User:
  Unter diesem Punkt können Einstellungen bezüglich des Benutzers geändert werden. Es ist möglich die Startgröße und Position der MainWindow-Klasse zu ändern, die am Start angezeigt wird. Zusätzlich kann man die aktuelle Position speichern. Ppm bezieht sich auf die maximale Abweichung, die bei der Berechnung zur Zuordnung von Features auf Metaboliten zulässig ist. Expanded gibt an, ob die Felder DbLinks, LcMethod und Features erweitert dargestellt werden sollen oder nicht.
- Info:
  Info zeigt eine JTextArea an, die kurz auf die verschiedenen Menüpunkte eingeht und auch kurz der Verwendung von Wildcards in der Suche beschreibt.

4.1.2 DisplayPanel

Falls kein neues JFrame erstellt wird, werden im DisplayPanel Informationen in beispielsweise Tabellen oder Masken wie etwa die Suchmaske widergegeben. Da das DisplayPanel von JPanel abgeleitet ist, kann auch diesem mittels add ein Swing-Komponent leicht hinzugefügt werden. Das Aufrufen der revalidate()-Methode nach dem Hinzufügen stellt sicher, dass der Inhalt auch dargestellt wird.

4.1.3 InfoPanel

Dieses Panel besteht aus einer Liste, welche die bisherigen Tätigkeiten seit Programmstart wiederspiegelt. Außerdem werden Fehlermeldungen, falls nicht akut, darin ausgegeben. Unter akut wird verstanden, dass etwa in ein Feld, das einen Zahlenwert verlangt, ein Buchstabe geschrieben wird. In diesem Fall wird

das nicht im InfoPanel angezeigt sondern direkt mittels einer JOptionPane. Die Liste wird mit JLabels gefüllt damit man die jeweiligen Zeilen mit Farbe versehen kann.

4.2 MySQL-Anbindung

Der Zugriff auf die MySQL-Datenbank erfolgt mit Hilfe des J-Connectors von MySQL (J-Connector, 2010) Version 5.1.10. Diese erlaubt es sehr einfach eine Verbindung zu der Datenbank mittels

```
Class.forName("com.mysql.jdbc.Driver").newInstance();
Connection connection =
DriverManager.getConnection("jdbc:mysql:///Joanneum", username,
passwort);
```

aufzubauen. Die Klasse Connection stellt die Methode prepareStatement(String statement) zur Verfügung welches eine Instanz der Klasse PreparedStatement zurück gibt. Der Parameter statement ist ein String der einen SQL-Aufruf wie etwa

```
SELECT * FROM Compounds WHERE CompoundId = ?
```

darstellt. Das ? in diesem Statement ist eine Variable, die erst mit der richtigen Methode aufgefüllt werden muss. Die Methode ist abhängig vom Datentyp. In unserem Beispiel müsste PreparedStatement.setInt(1, 2) verwendet werden. Das bedeutet, dass das erste ? im Statement mit einer 2 gefüllt wird. Dadurch ist die Möglichkeit gegeben mehrere Parameter in einem Statement verwenden zu können. Allerdings muss man auch jedem ? des Statements einen Wert zuweisen. Ist das PreparedStatement vorbereiten kann mittels der Methode executeQuery() dieses ausgeführt werden. Als Rückgabewert erhält man ein ResultSet. In diesem stehen nun die Informationen die etwa ein SELECT-Statement zurückgeben soll. Es ist aber darauf zu achten, dass bei INSERT-, UPDATE- oder DELETE-Statements nicht executeQuery() sondern executeUpdate() verwendet werden muss, was als Rückgabewert einen Integer hat. Ein ResulSet kann mittels Aufruf der Methode next() zeilenweise durchgegangen werden. Zugriff auf die Werte hat man indem man eine der Getter-Methoden verwendet die wieder datentypspezifisch sind. Dabei muss man

einfach die Spalte mittels Integer oder mittels deren Namen als Parameter mitgeben.

Da in Zukunft von einer großen Datenmenge in der Joanneum Datenbank ausgegangen werden kann, verwenden alle SQL-Statements LEFT JOIN da dieses erstens weitaus speicherschonender ist als etwa ein INNER JOIN und zweitens auch meistens schneller ausgeführt wird.

Prepared Statements (PreparedStatements, 2010) verhinden effektiv Angriffe auf die Datenbank durch sogenannte SQL-Injections (SQL-Injection, 2010). Bei solch einem Angriff versucht jemand SQL-Befehle in die Parameter einer Abfrage einzuschleusen um Informationen zu erlangen, die er nicht haben sollte oder sonstigen Schaden anzurichten. Prepared Statements verhindern dass, indem das Statement schon übersetzt im System ist und die Parameter somit nicht mitübersetzt werden. Der selber Umstand führt zu einer Erhöhung der Geschwindigkeit falls das gleiche Statement wieder und wieder verwendet wird und lediglich mit neuen Parametern gefüllt wird. Das könnte zum Beispiel bei mehreren Inserts der Fall sein.

4.3 R-Anbindung

Um R in Java verwenden zu können, wird R-Serve von Rosunda verwendet (R-Serve, 2010). Dieses bietet ein einfach zu bedienendes Interface um Daten von Java nach R zu schicken, dort zu verarbeiten und wieder aus R zurück zu Java zu holen.
Damit R-Serve läuft, muss allerdings auch ein dementsprechender R-Prozess gestartet werden was unter Windows mittels der Library ProcessBuilder bewerkstelligt wird. Die dabei aufgerufene Batch-Datei enthält lediglich die Zeile:

"C:\Program Files\R\R-2.10.1\bin"\R.exe CMD BATCH startRServe.r

In der Datei startRserve.r wird wiederum nur die Library Rserve geladen und die Methode Rserve() aufgerufen:

```
library(Rserve)
Rserve()
```

Der Prozess wird gleich am Anfang des Programmes gestartet. Somit hat dieser genug Zeit zu starten bevor der erste Zugriff auf R geschieht, was wiederum eine Exception zur Folge hätte.

Unter Unix-Systemen benötigt der Process keine Batch-Datei sondern lediglich

Process proc = Runtime.getRuntime().exec(new String[] {"/bin/sh", "-c", "R CMD Rserve --vanilla"});

Der Prozess wird beim Schließen des Programmes automatisch wieder beendet um den Speicher nicht mit Prozessen zu belasten, die ohnehin nicht mehr verwendet werden. Das notwendige Event wird mittels eines WindowListener abgefangen, welcher die MainWindow-Klasse überwacht.

R wird eingebunden damit das Packet XCMS verwendet werden kann, welches Peaks, Features und MSMS-Daten aus mzXML-Dateien ausliest und Retentionszeit- sowie Peakanpassungen vornimmt.

Da nur ein R-Prozess läuft, muss darauf geachtet werden, dass nicht mehrere Threads gleichzeitig diesen verwenden wollen, da es dabei zu Fehlern kommen kann. Um das zu verhindern besitzt die MainWindow-Klasse die Variable r_lock_. Diese kann nur über die synchronisierten Methoden lockR() beziehungsweise unlockR() geändert werden. Der Zugriff auf den Wert der Variablen erfolgt ebenfalls synchronisiert durch die Methode getRLock(). Diese wird aufgerufen bevor neue Experimente zur Datenbank hinzugefügt werden oder auch bevor EIC-Diagramme erstellt werden. Ist r_lock_ auf „true“ gesetzt wird eine Mitteilung angezeigt, dass R gerade beschäftigt ist. Ansonsten wird lockR() aufgerufen. Nach Beenden der Benutzung des R-Prozesses wird dann unlockR() aufgerufen um anderen den Zugriff auf den R-Prozess wieder zu ermöglichen.

4.4 Zuordnung von Features zu Metaboliten

Diese Funktionalität ist wohl die wichtigste, welche die GUI besitzt. Dabei geht es darum Vorschläge für die Zuordnung von Features zu Metaboliten zu berechnen.

4.4.1 Joanneum-Datenbank

Die Berechnung für die Joanneum-Datenbank geschieht automatisch nachdem ein neues Experiment hinzugefügt wurde. Sie kann auch später durchgeführt

werden indem man die Experimentenliste anzeigen lässt und im Popup-Menü „show adduct/feature hits“ auswählt. Das Ergebnis so einer Berechnung zeigt Abbildung 22. Blaue Zeilen sind bereits Metaboliten zugeordnet. Das kann über den Menüpunkt „assign“ im Popup-Menü durchgeführt werden. Sollte die Zuordnung nicht stimmen, kann sie über „unassign“ wieder gelöscht werden. Grüne Zeilen sind noch nicht zugeordnet, die Retentionszeit der LC-Methode des Metaboliten liegt aber in einem vom Benutzer zu definierendem Rahmen zu Rtmed. Da die Tabelle benutzerdefiniert ist und die Methode prepareRenderer überschrieben wird, werden die Farben der Zellen nach jedem repaint() der Tabelle überprüft. Daher werden auch spätere Änderungen der Retentionszeitabweichung des Benutzers unmittelbar sichtbar ohne dass die ganze Tabelle neu berechnet werden muss. Weiße Zeilen sind keinem Metaboliten zugeordnet und der Retentionszeitrahmen wurde auch nicht eingehalten bzw ist keine LC-Methode für den Metaboliten definiert. Desweiteren kann man Informationen zum Metaboliten und Feature anzeigen lassen. Es ist auch möglich EIC-Diagramme der Featuredaten ausgeben zu lassen. Dies wäre zwar auch über den Umweg über das Anzeigen des Features möglich, wäre aber unpraktisch. Um die Daten aus der Tabelle auch in anderen Programmen benutzen zu können gibt es auch eine Export-Funktion. Diese speichert in eine Datei, die vorher mittels eines JFileChoosers abgefragt wurde. Die einzelnen Spalten werden dabei einfach mittels eines Tabulators getrennt, die Reihen durch Zeilenumbrüche. Bei den Double-Werten wird der Punkt als Kommazeichen durch einen Beistrich ersetzt um Kompatibiltät mit etwa Excel zu gewährleisten.

Cid	Name	AdductId	AdductName	FeatureId	Variance	Mzmed	Mzmin	Mzmax	RT-Variance	Rtmed	Rtmin	Rtmax
1	L-Alanine	5	H-	4,841	2.0367	88.0407	88.0406	88.0407	1.02	9.11	9.1	9.11
1	L-Alanine	5	H-	4,842	2.8363	88.0407	88.0407	88.0408	4.03	14.16	14.12	14.18
1	L-Alanine	5	H-	5,030	1.1145	88.0406	88.0406	88.0406	35.39	45.52	45.45	45.52
1	L-Alanine	5	H-	5,460	1.2825	88.0406	88.0406	88.0406	35.71	45.83	45.79	45.93
2	L-Arginine	5	H-	4,577	0.2479	173.1044	173.1044	173.1045	0.08	11.03	11.02	11.06
3	L-Asparagine	5	H-	4,524	0.2051	131.0463	131.0463	131.0463	0	10.18	10.15	10.19
4	L-Aspartic acid	5	H-	4,706	0.3017	132.0303	132.0303	132.0303	0.67	12.49	12.49	12.5
6	L-Glutamic acid	5	H-	4,768	0.082	146.0459	146.0459	146.046	0.64	12.52	12.5	12.53
7	L-Glutamine	5	H-	4,652	0.859	145.0621	145.0621	145.0621	0.22	9.89	9.88	9.9
7	L-Glutamine	5	H-	4,653	1.6835	145.0617	145	617	3.13	13.24	13.24	13.24
7	L-Glutamine	5	H-	4,654	1.098	145.0618	145	618	1.62	11.73	11.73	11.73
7	L-Glutamine	5	H-	5,414	1.3765	145.0617	145	617	3.45	13.56	13.56	13.56
8	Glycine	5	H-	5,402	3.1423	74.0251	74.0	51	0.2	9.99	9.99	9.99
9	L-Histidine	5	H-	4,549	0.2464	154.0623	154				38	10.41
9	L-Histidine	5	H-	5,437	0.3213	154.0622	154.0622	154.0			05	43.09
10	L-Leucine	5	H-	4,678	0.6516	130.0874	130.0873	130.0			3	8.16
11	L-Isoleucine	5	H-	4,678	0.6516	130.0874	130.0873	130.0			3	8.16
12	L-Lysine	5	H-	4,611	0.8195	145.0982	145.0982	145.0			61	11.64
13	L-Serine	5	H-	4,538	1.2758	104.0355	104.0355	104.0356	0.19	10.22	10.21	10.22
14	L-Threonine	5	H-	4,850	1.2189	118.0512	118.0512	118.0512	0.14	9.84	9.82	9.86
15	L-Tryptophan	5	H-	4,189	1.5545	203.0824	203.0823	203.0825	0.17	8.29	8.26	8.29
16	L-Tyrosine	5	H-	4,323	0.6969	180.0666	180.0664	180.0666	0.27	9.27	9.25	9.27
17	L-Valine	5	H-	4,670	0.2555	116.0718	116.0718	116.0718	0.12	8.79	8.76	8.81
18	Phosphoserine	5	H-	4,639	0.5211	184.0016	184.0016	184.0017	4.51	12.49	12.43	12.49
20	L-Homoserine	5	H-	4,850	1.2189	118.0512	118.0512	118.0512	0.14	9.84	9.82	9.86

Abbildung 22: Resultat der Berechnung für die Zuordnung von Metaboliten und Features

Für die Berechnung werden alle Metaboliten mit allen Features des jeweiligen Experimentes verglichen. Daher werden zuerst alle Metaboliten aus der Datenbank geholt. Zur Berechnung werden desweiteren Feature und Adduktdaten benötigt die mittels des folgendem SQL-Statements bezogen werden:

> *SELECT featurelist.featurelistid, featurelist.rtmed, featurelist.rtmin, featurelist.rtmax, featurelist.mzmed, featurelist.mzmin, featurelist.mzmax, featurelist.msmsid, polarity, ionization, adductsid, adductname, adductmass, factor, charge FROM adducts, featurelist LEFT JOIN featurepeaklist ON (featurelist.featurelistid = featurepeaklist.featurelistid) LEFT JOIN peaklist ON (featurepeaklist.peaklistid = peaklist.peaklistid) LEFT JOIN analysis ON (peaklist.analysisid = analysis.analysisid) LEFT JOIN analysismetadata ON (analysis.analysismetadataid = analysismetadata.analysismetadataid) WHERE experimentid = ? AND (polarity = charge OR adductname = 'nothing') GROUP BY adductsid, featurelist.featurelistid*

Dabei werden mittels einer Abfrage alle benötigten Daten bezogen und auch gleich eine Einschränkung dieser vorgenommen. So werden nur jene Addukte

mit in die Liste genommen, deren Ladung gleich der Ionisierungsladung des Massenspektrometers ist, welche aus der Analysetabelle ersichtlich ist. Das Hilfsaddukt „nothing“ mit Masse Null und Faktor eins, wird jedesmal mitgeliefert. Dieses wird benötigt um Moleküle zu identifizieren die ansich eine Ladung aufweisen. Desweiteren wird die Ionisierungsmethode aus der AnalyseMetaData-Tabelle mitgeliefert. Dies ist notwendig um später eine weitere Einschränkung der Berechnung zu gewährleisten was sich wiederum positiv auf die Laufzeit auswirkt. So müssen bei der Ionisierungsmethode APCI nur H+ oder H-, je nach Polarität, und das Addukt „nothing“ verwendet werden. Bei der Methode ESI kommt nur die Einschränkung bezüglich Polarität zu tragen.

Für jeden Metaboliten werden dann die richtigen Massen mittels (6) berechnet. Metabolitenmasse ist dabei die Masse die aus der Tabelle Compounds kommt. Adduktmasse und Faktor sind in den Daten, die mittels des zuvor beschriebenen SQL-Statements bezogen wurden, enthalten. In dem Resultat werden jene Daten angezeigt, die (7) entsprechen:

$$\frac{Metabolitenmasse + Adduktmasse}{Faktor} = Masse \tag{6}$$

$$\left|\frac{Mzmed - Masse}{Masse}\right| \cdot 1000000 < ppm \tag{7}$$

Mzmed ist wiederum in den zuvor bezogenen Daten enthalten und kommt ursprünglich aus der Featurelist-Tabelle. Ppm kommt von den aktuellen Benutzereinstellungen.

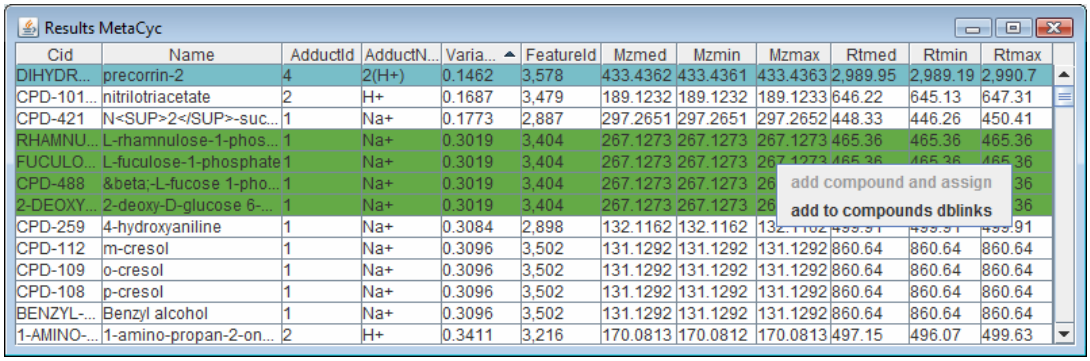
4.4.2 MetaCyc-Datenbank

Die MetaCyc-Datenbank kann als gepackt heruntergeladen werden. Dieses ist weiterunterteilt in verschiedene Spezies. Die für uns interessanten Informationen liegen in den jeweils darin enthaltenen „compounds.dat“-Dateien. Mittels eines

Parsers werden die Unique-Id, Typ, Name, Formel, Links zu anderen Datenbanken, InCHI, SMILES, Masse, Synonyme und Spezien geparst. Nach dem Parsen werden die Informationen in die offline-erstellte MetaCyc-Datenbank geschrieben. Sollte ein Metabolite bereits für eine andere Spezies aufgenommen worden sein, wird nur der Namen der neuen Spezies im Feld Spezies mitgespeichert. Zur Zeit befinden sich in dieser Offline-Version der MetaCyc-Datenbank 2.277 Einträge.

Von der Experimentenansicht aus, kann man auch in dieser nach Treffern bezüglich der zuvor beschriebenen Art suchen. Die Anzeige dieser wird in Abbildung 23 veranschaulicht. Die Spalten sind die gleichen wie bei der Suche in der Joanneum-Datenbank. Blaue Zeilen, zeigen an, dass der Namen, die AdductId und die FeatureId bereits in der Datenbank gespeichert sind. Grüne Zeilen zeigen eine Zuordnung in der der Namen nicht identisch ist. In diesem Fall ist es möglich den Eintrag aus dem Resultat zu den Datenbanklinks hinzuzufügen. Sollte eine Zeile Orange dargestellt sein, sind mehrere Metaboliten mit dem selben Addukt und dem selben Feature bereits verknüpft. In diesem Fall ist ein Hinzufügen von Datenbanklinks nicht möglich. Weiße Zeile haben noch keine Zuteilung. Deshalb ist es hier möglich die Informationen aus der MetaCyc-Datenbank zu verwenden um einen neuen Metaboliten in der Joanneum-Datenbank anzulegen. Gleichzeitig erfolgt die Zuordung von Metaboliten, Feature und Addukt.

Die Einschränkung bezüglich Retentionszeit ist nicht möglich, da keine Zeiten in der MetaCyc-Datenbank enthalten sind.



Cid	Name	AdductId	AdductN...	Varia...	FeatureId	Mzmed	Mzmin	Mzmax	Rtmed	Rtmin	Rtmax
DIHYDR...	precorrin-2	4	2(H+)	0.1462	3,578	433.4362	433.4361	433.4363	2,989.95	2,989.19	2,990.7
CPD-101...	nitrilotriacetate	2	H+	0.1687	3,479	189.1232	189.1232	189.1233	646.22	645.13	647.31
CPD-421	N²-suc...	1	Na+	0.1773	2,887	297.2651	297.2651	297.2652	448.33	446.26	450.41
RHAMNU...	L-rhamnulose-1-phos...	1	Na+	0.3019	3,404	267.1273	267.1273	267.1273	465.36	465.36	465.36
FUCULO...	L-fuculose-1-phosphate	1	Na+	0.3019	3,404	267.1273	267.1273	267.1273	465.36	465.36	465.36
CPD-488	β-L-fucose 1-pho...	1	Na+	0.3019	3,404	267.1273	267.1273	26			36
2-DEOXY...	2-deoxy-D-glucose 6-...	1	Na+	0.3019	3,404	267.1273	267.1273	26			36
CPD-259	4-hydroxyaniline	1	Na+	0.3084	2,898	132.1162	132.1162	132.1162	499.91	499.91	499.91
CPD-112	m-cresol	1	Na+	0.3096	3,502	131.1292	131.1292	131.1292	860.64	860.64	860.64
CPD-109	o-cresol	1	Na+	0.3096	3,502	131.1292	131.1292	131.1292	860.64	860.64	860.64
CPD-108	p-cresol	1	Na+	0.3096	3,502	131.1292	131.1292	131.1292	860.64	860.64	860.64
BENZYL-...	Benzyl alcohol	1	Na+	0.3096	3,502	131.1292	131.1292	131.1292	860.64	860.64	860.64
1-AMINO-...	1-amino-propan-2-on...	2	H+	0.3411	3,216	170.0813	170.0812	170.0813	497.15	496.07	499.63

Abbildung 23: Zuordnung von Features zur MetaCyc-Datenbank

4.4.3 ChEBI-Datenbank

Diese Datenbank kann auf mehrere Methoden Offline verfügbar gemacht werden. Da die ChEBI-Datenbank ca. 18 Millionen Einträge besitzt, ist ein Befüllen mittels SQL-Insert-Reihen, wie sie in den SQL-Dumps vorkommen wenig praktikabel, da diese Methode viel zu lange dauert. Schneller geht es mittels Flatfile-TableDumps. Hier sind die Informationen der Tabellen dargestellt indem die einzelnen Spalten durch Tabulatoren, die einzelnen Reihen durch Zeilenumbrüche getrennt sind. Manche Reihen enthalten aber überflüssige Zeilenumbrüche die mittels eines Skriptes bereinigt wurden. Desweiteren enthalten manche Reihen zuwenige Spalten. Zur Beseitigung dieses Problems genügt etwa der folgende sed-Befehl in der Eingabeaufforderung:

sed 's/null\tnull\tnull/null\tnull\tnull\tnull/g' compounds_3star.tsv > compounds_3star_test.tsv

Aus diesen Dateien kann mit Hilfe des LOAD DATA-Befehls die Datenbank relative schnell befüllt werden. Hier ein Beispiel zur Verwendung des Befehles für die Reference-Tabelle:

load data infile 'D:\\Studium\\Master\\Diplomarbeit\\Workspace Diplomarbeit\\DA\\databases\\chebi\\flatfiles\\reference.tsv' into table chebi.reference fields terminated by '\t' lines terminated by '\n' (id,compound_id,reference_id,reference_db_name,location_in_ref,reference_name)

Ist die ChEBI-Datenbank hergestellt, kann wiederum über die Experimentansicht die Berechnung von Treffern durchgeführt werden. Das Ergebnis sieht gleich aus wie Abbildung 23 dargestellt und verfügt über die gleiche Funktionalität wie das Ergebnis für die MetaCyc-Datenbank. Eine Einschränkung mittels der Retentionszeit ist auch hier nicht möglich, da diese Informationen auch in der ChEBI-Datenbank nicht enthalten sind. Aufgrund der großen Anzahl von Metaboliten, die in der Datenbank enthalten sind, dauert eine Berechnung für etwa 3.000 Features ca. 10 Minuten.

4.5 Diagrammerzeugung

Zur Veranschaulichung von Daten sind Diagramme die beste Lösung. Daher können Diagramme für MSMS-Daten und EIC-Diagramme erzeugt werden.

4.5.1 MSMS-Diagramm

MSMS-Diagramme können über die Popup-Menus in den Ansichten „Show Compounds“, „Show Features“ und „Show Peaks“ generiert werden.
Die Klasse ShowMsmsDiagram ist von JFrame abgeleitet damit darin das Diagramm eingebettet werden kann. Der Konstruktor bekommt die altbekannt MainWindow-Klasse, die MsmsId und die Masse des Parentpeaks. Die MainWindow-Klasse wird einerseits benötigt um das Diagramm in der Mitte der Anwendung anzuzeigen, andererseits dient die SQL-Verbindung der Klasse zum Beziehen der MSMS-Daten. Genau dafür wird ebenfalls die MsmsId benötigt. Die Masse des Parentpeaks wird benötigt um die Länge der X-Achse festzulegen.
Die Daten für das Diagramm können einfach mit der MsmsId aus der Joanneum-Datenbank bezogen werden und in der von JFreeChart bereitgestellten Klasse IntervalXYDataset gespeichert werden. XYBarChart wird mit der ChartFactory erstellt und repräsentiert die visuelle Darstellung des Diagrammes. Der Konstruktor bekommt unter anderem Titel, Achsenbeschriftungen und das Datenset übergeben. Mit dem erstellten Diagramm kann ein ChartPanel erzeugt werden, welches dann in das Frame eingebunden wird. Abbildung 24 zeigt ein MSMS-Diagramm.

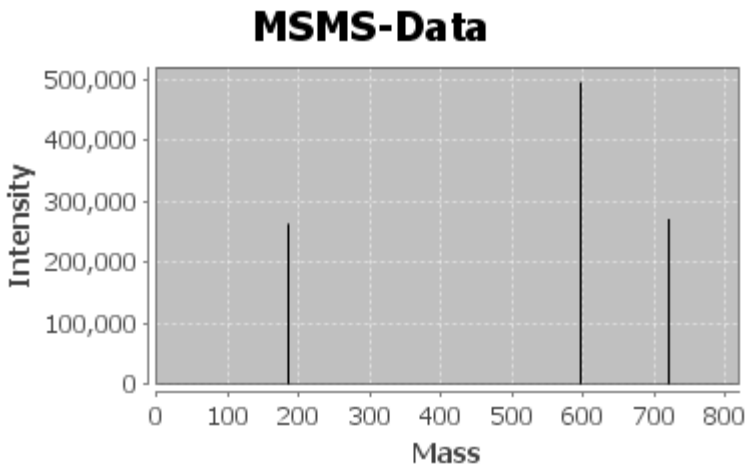


Abbildung 24: MSMS-Diagramm

4.5.2 EIC-Diagramm

EIC-Diagramme können für Features erstellt werden wobei EIC Extraced Ion Chromatogram heißt. Dabei kann man wählen ob man die Daten vor oder nach der Retentionszeitkorrektur verwenden will. Abbildung 25 zeigt ein Beispiel vor der Retentionszeitkorrektur, Abbildung 26 zeigt das gleiche Feature mit korrigierten Retentionszeiten.

Anders als beim MSMS-Diagramm stammen die Daten für die EIC-Diagramme nicht aus der Joanneum-Datenbank sondern aus dem R-Workspace, der beim Hinzufügen des Experimentes mitabgespeichert wurde. Das Erstellen der EIC-Diagramme war mit ein Grund das zu tun, da sonst das komplette xcmsSet für jedes Diagramm erneut berechnet werden müsste viel Zeit an Anspruch nehmen würde. Mit der jetzigen Methode kann der Workspace leicht mittels des Arbeitsverzeichnis aus der Experimententabelle in R geladen werden. Mittels RServe wird der Befehl

connection_.eval("eic <- getEIC(xset, rtrange = 500, groupidx = " + feature_id + ", rt = \"raw\")");

für Daten ohne Retentionszeitkorrektur beziehungsweise

connection_.eval("eic <- getEIC(xset, rtrange = 500, groupidx = " + feature_id + ", rt = \"corrected\")");

mit Retentionszeitkorrektur ausgeführt. Wie man sieht, wird ein Retentionszeitbereich von 500 Sekunden um das ausgewählte Feature verwendet. Von R nach Java bekommt man die Daten mit dem Aufruf

connection_.eval("eic@eic$" + filename).asList()

Die zurückgelieferte Liste beinhaltet die Intensitäten und Retentionszeiten als Double-Werte. Es ist aber zu berücksichtigen, dass in der Liste zuerst alle Retentionszeiten stehen und danach alle Intensitäten. Es ist also leider keine Liste von Double-Paaren. Somit müssen die Daten vor Verwendung in eine sinnvollere Form, also ein Doublearray welches auf jeder Position ein Double-Paar hält, gebracht werden. Da XCMS die Retentionszeiten in Sekunden speichert wird dieser Wert mit 60 dividiert um mit den Retentionszeiten in der Joanneum-Datenbank konform zu gehen. Dieser Vorgang, inklusive des obigen Befehles muss für alle Dateien, die das Experiment beinhaltet, durchgeführt werden, da diese die einzelnen Linien im Diagramm darstellen.

Die Daten werden dann der Klasse XYSeriesCollection hinzugefügt. Mittels der ChartFactory wird diesmal ein XYLineChart erzeugt. Um die wesentlichen Peaks im Diagramm besser hervorzuheben, wurde ein sogenannter Marker gesetzt, in den folgenden Abbildungen als grauer Balken ersichtich. Die Werte davon sind die kleinsten Rtmin beziehungsweise die größten Rtmax Werte der Peaks, aus denen das Feature berechnet wurde. In der Legende sieht man die Namen der Dateien, aus denen die Daten stammen.

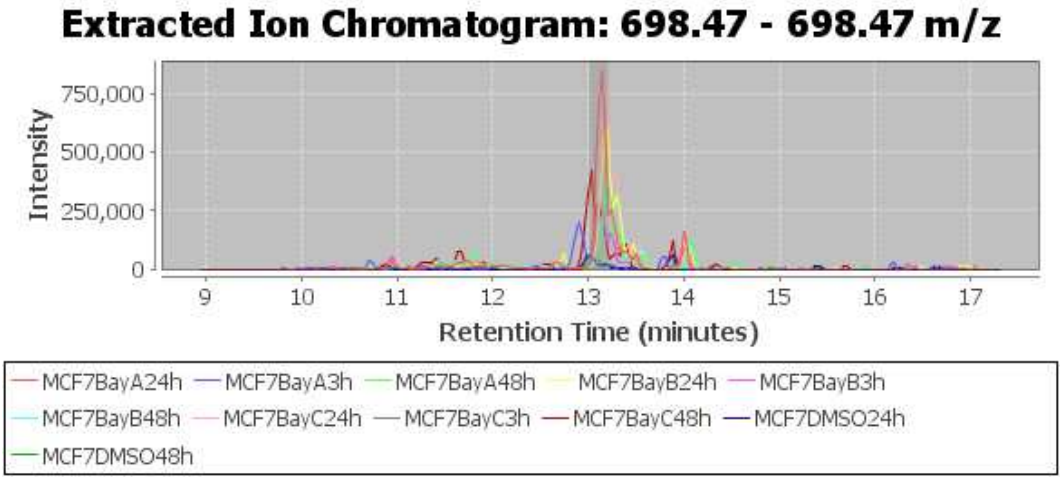


Abbildung 25: EIC-Diagramm vor Retentionszeitkorrektur

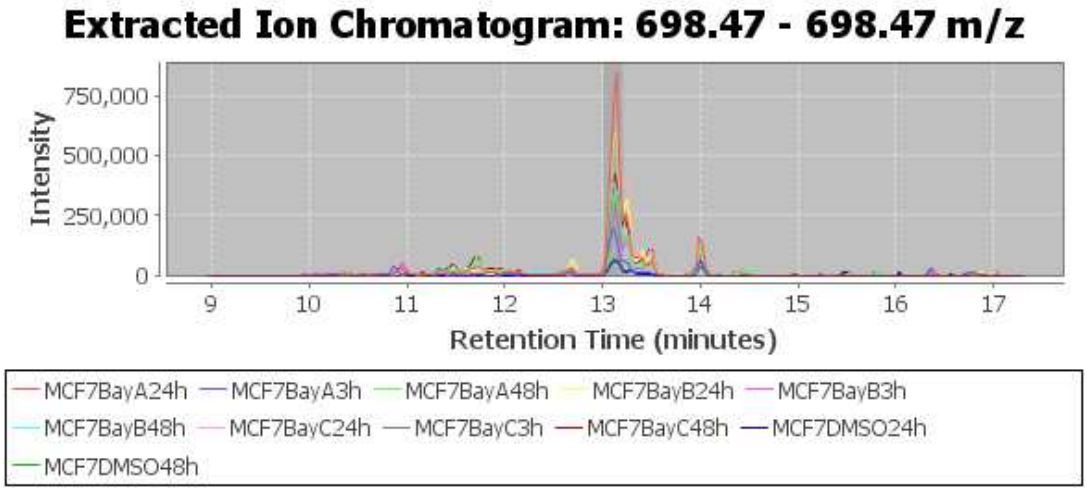


Abbildung 26: EIC-Diagramm nach Retentionszeitkorrektur

4.6 Testdaten

Zu Beginn wurden 142 Standardmetaboliten inklusive Datenbanklinks und LC-Methode in sowie 580 Phosphorlipide in die Datenbank gespielt. Desweiteren enthält die Addukttabelle die folgenden Einträge:

AdductsId	AdductName	AdductMass	Factor	Charge
1	Na+	22.9892	1	+
2	H+	1.0072	1	+
3	NH4+	18.0338	1	+
4	2(H+)	2.0145	2	+
5	H-	-1.0072	1	-
6	2H-	-2.0145	2	-
7	NOTHING	0	1	0
8	undefined	0	1	0

Tabelle 9: Addukttabelle

Für die Testdaten wurden Proben aus Gehirn, Muskel und Leber von Mäusen in einem LTQ Orbitrap XL der Firma Thermo Scientific mit den Einstellungen

- Ionisierung: ESI

- Massen-Analysierer: FTMS
- Polarität: positiv und negativ
- FilterLine: FTMS - c ESI SIM ms [70.00-600.00]
  FTMS - c ESI Full ms [500.00-2000.00]

Die aus dem Massenspektrometer entstanden RAW-Dateien wurden mittels ReadW in mzXML-Dateien konvertiert. Die Dateien wurden entsprechend der verwendeten Einstellungen in die Ordner High und Low eingeteilt. Diese enhalten die Ordner Positive und Negativ und diese enthalten wiederum Ordner für Gehirn, Leber und Muskel in welche dann die entsprechenden mzXML-Dateien verschoben wurden.
Mittels der für die Diplomarbeit erstellten Anwendung wurde dann jeder Gehirn-, Leber- und Muskel-Ordner in die Joanneum-Datenbank übernommen. Dafür wurde folgende XCMS-Einstellung verwendet:

```
library(xcms)
library(Rmpi)
library(CAMERA)
library(RANN)
xset <- xcmsSet(method='centWave', ppm=10, peakwidth=c(30,120),
   prefilter=c(3,10000), mzCenterFun="wMean", integrate=1, mzdiff=-
   0.001, fitgauss=TRUE, verbose.columns=TRUE, nSlaves=1)
xset <- group(xset,  method = "nearest", mzVsRTbalance=100,
   mzCheck=0.005, rtCheck=15, kNN=10)
xset <- retcor(xset, missing = 2, extra = 0, method = c("loess"), span = .2)
xset <- group(xset,  method = "nearest", mzVsRTbalance=100,
   mzCheck=0.005, rtCheck=15, kNN=10)
xset <- fillPeaks(xset)
xanno<-xsAnnotate(xset,sample=1)
xanno<-groupFWHM(xanno)
xanno<-groupCorr(xanno)
xanno<-findIsotopes(xanno)
raw <- xcmsRaw(xset@filepaths[1])
xanno<-findAdducts(xanno,polarity=as.character(raw@polarity[1]))
an<-getPeaklist(xanno)
```

Tabelle 10 zeigt eine Statistik zu Anzahl der Peaks, Features, Treffer, Treffer im Retentionszeitrahmen und Dauer der Berechnung der Treffer. Die Berechnung wurde mit einem Ppm < 10 durchgeführt. Der Rt-Rahmen liegt bei zwei Minuten.
In Summe wurden mit der Studie 142.349 Peaks und 28.828 Features zur Datenbank hinzugefügt. Es konnten insgesamt 2.446 Metaboliten identifiziert

werden wobei 568 davon innerhalb des Retentionszeitrahmen lagen. Die Berechnungen dauerten alles in allem etwa 47 Sekunden. Die geringe Anzahl der Treffer im Rt-Rahmen in Bezug auf die Anzahl der Treffer bei den Experimenten im hohen Massenbereich ist daruf zurückzuführen, dass hauptsächlich Phosphorlipide gefunden wurden und für diese noch keine Lc-Methoden in der Datenbank vorhanden sind.

Experiment	Peaks#	Features#	Treffer#	Treffer# in Rt-Rahmen	Zeit in Sek.
low-neg-brain	10.239	1.706	120	64	1,1731
low-neg-muscle	5.452	1.363	155	56	1,1809
low-neg-liver	9.668	1.611	137	78	1,1935
low-pos-brain	12.962	3.239	184	103	5,5226
low-pos-muscle	20.782	3.461	156	88	6,6256
low-pos-liver	10.976	2.744	138	84	4,4665
high-neg-brain	5.280	880	51	6	1,1045
high-neg-muscle	3.112	778	11	5	0,983
high-neg-liver	6.630	1.105	86	12	1,1279
high-pos-brain	14.516	3.629	453	21	6,6677
high-pos-muscle	28.448	4.741	485	24	8,8798
high-pos-liver	14.284	3.571	470	27	7,7238
Summe	**142.349**	**28.828**	**2446**	**568**	**46,6489**

Tabelle 10: Statistik zu Testdaten

5. Diskussion

Diese Kapitel behandelt Probleme die bei der Programmierung der Diplomarbeit auftraten aber auch Verbesserungen die vorgenommen wurden um die Performance zu steigern. Desweiteren wird ein Ausblick darüber gegeben welche Features zukünftig noch eingebaut werden sollen.

5.1 Performance-Steigerungen

Um beim Finden von Metaboliten in Analysedaten eine wirkliche Zeitersparnis für den Benutzer zu erreichen wurde darauf geachtet, dass Berechnungen und Datenbankoperationen schnell von Statten gehen. Da XCMS direkt in R verwendet wird, entsteht dort kein Zeitverlust und somit entstehen auch keine Verbesserungsmöglichkeiten. Jedoch konnte der Zeitaufwand in Java beim Einfügen von Daten in die Datenbank und bei den Berechnungen für die Zuordnung von Metaboliten zu Features im Laufe der Diplomarbeit verbessert werden.

5.1.1 MySQL

Für die INSERT-Operationen werden wie schon erwähnt PreparedStatements verwendet. Vor den jeweiligen Schleifen, die Peaks, Features, Zuordnung von Peaks zu Features und MSMS-Daten in die Datenbank eintragen, wird die AutoCommit-Variable der Connection auf „false“ gesetzt. Dadurch werden nicht bei jedem Aufruf der Methode PreparedStatement.executeUpdate() die Daten in die Datenbank geschrieben sondern erst beim nächsten Connection.Commit(). Dadurch kann eine enorme Verbesserung der Geschwindigkeit der INSERT-Statements erreicht werden. Nach Connection.Commit() wird der AutoCommit-Wert der Verbindung wieder auf „true“ gesetzt um nicht bei anderen Datenbankoperationen, die diese vorgehensweise nicht benötigen, Probleme hervor zurufen.

5.1.2 Zuordnung von Features zu Metaboliten

Die Berechnung für die Zuordnung von Features zu Metaboliten erfolgt, wie zu erwarten ist, innerhalb von zwei Schleifen, eine für die Metaboliten und eine für die Featuredaten. Damit wird bei etwa 700 Metaboliten und 3000 Features eine Laufzeit von ca. 6 Sekunden erreicht. Die zuvor verwendete Berechnung die nicht alle Addukte- und Featuredaten auf einmal bezog sondern in einer dritten Schleife für jedes Feature die passenden Addukte aus der Datenbank holte dauerte in etwa 10 Minuten. Somit konnte durch die neue Methode eine Verbesserung der Laufzeit um 10.000% erreicht werden. Dies ist vor allem für die Suche in der ChEBI-Datenbank wichtig, die mittels Flatfiles offline verfügbar gemacht werden kann und 557.000 Metaboliten beinhaltet. Ohne die Verbesserung würde die Suche darin etwa 16 Stunden dauern was keinem Benutzer zumutbar wäre.

5.2 Kompatibilität Windows/Unix

Die Diplomarbeit wurde unter Windows erstellt. Da sie aber auch unter Unix laufen soll, mussten einige Adaptionen vorgenommen werden. Java ist hierbei weniger das Problem, da dieses in der Java Virtual Machine ausgeführt wird und somit betriebssystemunabhängig ist. Anders verhält es sich mit MySQL und dem Start von Prozessen, im Fall der Diplomarbeit der Rserve-Prozess.

Bei MySQL ist zu beachten, dass sich das Verhalten bezügliche der Beachtung von Groß- und Kleinschreibung von Datenbank- und Tabellennamen auf das zugrunde liegende Betriebssystem bezieht. So ist MySQL unter Windows in Bezug auf Datenbank- und Tabellennamen nicht case sensitive, unter Unix sehr wohl. Aus diesem Grund mussten einige der SQL-Befehle in den Klassen des Paketes JoanneumDatabase angepasst werden.

Desweiteren ist die Methode getName() der Klasse java.io.File mit vorsicht zu behandeln. Hier würde Windows für die Datei gunnar/media/test.txt etwa test.txt retournieren, Linux jedoch den kompletten Pfad mit Namen. Wird darauf nicht geachtet gibt es Probleme beim Erstellen der EIC-Diagramme weil dort beim Auslesen von Daten aus R nur der Namen von Dateien verwendet werden darf.

Ebenfalls unterschiede gibt es beim Starten von neuen Prozessen was schon im Abschnitt zur Einbindung von R aufscheint.

5.3 Ausblick

Dieser Abschnitt befasst sich mit möglichen Verbesserungen die in Zukunft noch geplant sind um entweder die Identifikation von Metaboliten zu Verbessern oder weitere Arbeitserleichterungen zu schaffen.

5.3.1 Automatisierte Datenvorbereitung

Zur Zeit werden zum Hinzufügen von Experimenten MzXML-Dateien verwendet. Um nicht mehr abhängig von ReadW zu sein und die manuelle Konvertierung von RAW-Dateien in MzXML-Dateien zu vermeiden, ist angedacht direkt RAW-Dateien verwenden zu können. Dazu müsste ein eigener Parser für diese Rohdateien geschrieben werden. Ein weiterer Vorteil ergäbe sich daraus, dass man mit dem Massenspektrometer zusätzliche Informationen in den RAW-Dateien schreiben könnte und später dann auch automatisch in die Datenbank speichern könnte.

5.3.2 Retentionszeitreihenfolge

Metaboliten haben die Eigenschaft, dass die zeitliche Reihenfolge ihres Auftauchens immer gleich ist. Somit können Metaboliten trotz Abweichung zu ihrer normalen Retentionszeit identifiziert werden. Die Vorgehensweise dabei sieht so aus, dass man den vorangehenden und den nachfolgenden Metaboliten identifiziert und aus anderen Experimenten sieht, welches Stoffwechselprodukt zwischen den beiden kommen sollte.

5.3.3 Systempeakseliminierung

Hierbei geht es darum unnötige Daten, die durch Rauschen des Massen-spektrometers entstehen, gar nicht erst in die Datenbank aufzunehmen. Ein Ansatz wäre es, Leerlaufanalysen durchzuführen. Peaks die darin enthalten sind

und auch in anderen Analysen vorkommen, sollten am besten gar nicht erst in die Datenbank aufgenommen werden sonder gleich ausgefiltert werden. Für Features, die aus solchen Peaks berechnet werden, verhält es sich gleich.

5.3.4 QC-Filter

Zur Qualitätssicherung werden sogenannte QC-Messungen durchgeführt. Ein QC ist eine gleichmäßige Mischung aus allen Proben eines Experimentes. Um Schwankungen in der Sensitivität eines Massenspektrometers festzustellen, wird in periodischen Abständen eine Messung dieses QCs durchgeführt. Wird später festgestellt, dass die Standardabweichung der Messungen zu hoch ist, können dementsprechend Features ausgefiltert werden.

Referenzliste

Benton, H. P., Wong, D. M., Trauger, S. A. und Siuzdak, G. XCMS2: processing tandem mass spectrometry data for metabolite identification and structural characterization. Anal.Chem. 80[16], 6382-6389. 2008-08-15.

Caspi, R., Foerster, H., Fulcher, C. A., Hopkinson, R., Ingraham, J., Kaipa, P., Krummenacker, M., Paley, S., Pick, J., Rhee, S. Y., Tissier, C., Zhang, P. und Karp, P. D. MetaCyc: a multiorganism database of metabolic pathways and enzymes. Nucleic Acids Res. 34[Database issue], D511-D516. 2006-01-01.

Caspi, R., Foerster, H., Fulcher, C. A., Kaipa, P., Krummenacker, M., Latendresse, M., Paley, S., Rhee, S. Y., Shearer, A. G., Tissier, C., Walk, T. C., Zhang, P. und Karp, P. D. The MetaCyc Database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. Nucleic Acids Res. 36[Database issue], D623-D631. 2008-01.

Caspi, R., Fulcher, C. A., Ingraham, J., Keseler, I, Krummenacker, M., and Paley, S., http://bioinformatics.ai.sri.com/ptools/curatorsguide.pdf, zuletzt besucht: 2009-11-11

ChEBI, http://www.ebi.ac.uk/chebi/, zuletzt besucht: 2009-11-11

Degtyarenko, K., de Matos, P., Ennis, M., Hastings, J., Zbinden, M., McNaught, A., Alcantara, R., Darsow, M., Guedj, M. und Ashburner, M. ChEBI: a database and ontology for chemical entities of biological interest. Nucleic Acids Res. 36[Database issue], D344-D350. 2008-01.

Draper, J., Enot, D. P., Parker, D., Beckmann, M., Snowdon, S., Lin, W. und Zubair, H. Metabolite signal identification in accurate mass metabolomics data with MZedDB, an interactive m/z annotation tool utilising predicted ionisation behaviour 'rules'. BMC Bioinformatics. 10, 227. 2009.

Dunn, W. B., Bailey, N. J. und Johnson, H. E. Measuring the metabolome: current analytical technologies. Analyst 130[5], 606-625. 2005-05.

Eclipse, http://www.eclipse.org/platform, zuletzt besucht: 2010-03-03

Goto, S., Nishioka, T. und Kanehisa, M. LIGAND: chemical database for enzyme reactions. Bioinformatics. 14[7], 591-599. 1998.

Goto, S., Nishioka, T. und Kanehisa, M. LIGAND database for enzymes, compounds and reactions. Nucleic Acids Res. 27[1], 377-379. 1999-01-01.

HMDB, http://www.hmdb.ca/, zuletzt besucht: 2009-11-11

InChI, http://www.inchi.info/, zuletzt besucht: 2009-12-12

IUPAC InChI, http://www.iupac.org/inchi/, zuletzt besucht: 2009-12-12

J-Connector, http://dev.mysql.com/downloads/connector/j/, zuletzt besucht: 2010-03-03

J.K.Nicholson, J.C.Lindon und E.Holmes. 'Metabonomics': understanding the metabolic responses of living systems to pathophysiological stimuli via multivariate statistical analysis of biological NMR spectroscopic data. xenobiotica 29[11], 1181-1189. 1999.

Java, http://java.com/de/, zuletzt besucht: 2010-03-03

Java-Swing, http://java.sun.com/j2se/1.4.2/docs/api/javax/swing/package-summary.html, zuletzt besucht: 2010-03-03

JFreeChart, http://www.jfree.org/jfreechart/, zuletzt besucht: 2010-04-04

Kanehisa, M., Araki, M., Goto, S., Hattori, M., Hirakawa, M., Itoh, M., Katayama, T., Kawashima, S., Okuda, S., Tokimatsu, T. und Yamanishi, Y. KEGG for linking genomes to life and the environment. Nucleic Acids Res. 36[Database issue], D480-D484. 2008-01.

Kanehisa, M., Goto, S., Kawashima, S. und Nakaya, A. The KEGG databases at GenomeNet. Nucleic Acids Res. 30[1], 42-46. 2002-01-01.

Karp, P. D., Riley, M., Paley, S. M. und Pellegrini-Toole, A. The MetaCyc Metabolic Pathway Database. Nucleic Acids Res. 30[1], 59-61. 2002-01-01.

KEGG, http://www.genome.jp/, zuletzt besucht: 2009-11-11

Kind, T. und Fiehn, O. Metabolomic database annotations via query of elemental compositions: mass accuracy is insufficient even at less than 1 ppm. BMC Bioinformatics. 7, 234. 2006.

Kind, T. und Fiehn, O. Seven Golden Rules for heuristic filtering of molecular formulas obtained by accurate mass spectrometry. BMC Bioinformatics. 8, 105. 2007.

KinoSearch, http://www.rectangular.com/kinosearch/, zuletzt besucht: 2009-11-11

LGPL, http://www.gnu.org/licenses/lgpl.html, zuletzt besucht: 2010-04-04

MetaCyC, http://metacyc.org/, zuletzt besucht: 2009-11-11

Metlin, http://metlin.scripps.edu/, zuletzt besucht: 2009-11-11

Moco, S., Bino, R. J., Vorst, O., Verhoeven, H. A., de Groot, J., van Beek, T. A., Vervoort, J. und de Vos, C. H. A liquid chromatography-mass spectrometry-based metabolome database for tomato. Plant Physiol 141[4], 1205-1218. 2006-08.

MoTo DB, http://appliedbioinformatics.wur.nl/moto/, zuletzt besucht: 2009-11-11

MySQL, http://www.mysql.com/, zuletzt besucht: 2010-03-03

MZedDB, http://maltese.dbs.aber.ac.uk:8888/hrmet/index.html, zuletzt besucht: 2009-11-11

Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H. und Kanehisa, M. KEGG: Kyoto Encyclopedia of Genes and Genomes. Nucleic Acids Res. 27[1], 29-34. 1999-01-01.

Pedrioli, P. G., Eng, J. K., Hubley, R., Vogelzang, M., Deutsch, E. W., Raught, B., Pratt, B., Nilsson, E., Angeletti, R. H., Apweiler, R., Cheung, K., Costello, C. E., Hermjakob, H., Huang, S., Julian, R. K., Kapp, E., McComb, M. E., Oliver, S. G., Omenn, G., Paton, N. W., Simpson, R., Smith, R., Taylor, C. F., Zhu, W. und Aebersold, R. A common open representation of mass spectrometry data and its application to proteomics research. Nat.Biotechnol. 22[11], 1459-1466. 2004-11.

PreparedStatements, http://de.wikipedia.org/wiki/Prepared_Statement, zuletzt besucht: 2010-03-03

R, http://de.wikipedia.org/wiki/R_%28Programmiersprache%29, zuletzt besucht: 2010-04-04

R-Serve, http://rosuda.org/Rserve/, zuletzt besucht: 2010-03-03

Ryan, D. und Robards, K. Metabolomics: The greatest omics of them all? Anal.Chem. 78[23], 7954-7958. 2006-12-01.

Sana, T. R., Roark, J. C., Li, X., Waddell, K. und Fischer, S. M. Molecular formula and METLIN Personal Metabolite Database matching applied to the identification of compounds generated by LC/TOF-MS. J.Biomol.Tech. 19[4], 258-266. 2008-09.

Sansone, S. A., Fan, T., Goodacre, R., Griffin, J. L., Hardy, N. W., Kaddurah-Daouk, R., Kristal, B. S., Lindon, J., Mendes, P., Morrison, N., Nikolau, B., Robertson, D., Sumner, L. W., Taylor, C., van der, Werf M., van Ommen, B. und Fiehn, O. The metabolomics standards initiative. Nat.Biotechnol. 25[8], 846-848. 2007-08.

SAX, http://www.saxproject.org/, zuletzt besucht: 2010-03-03

Scalbert, A., Brennan, L., Fiehn, O., Hankemeier, T., Kristal, B. S., Ommen, B., Pujos-Guillot, E., Verheij, E. R., Wishart, D. S. und Wopereis, S. Mass-spectrometry-based metabolomics: limitations and recommendations for future progress with particular focus on nutrition research. Metabolomics . 2009. Springer Boston.

SMILES, http://www.daylight.com/smiles/, zuletzt besucht: 2009-12-12

Smith, C. A., O'Maille, G., Want, E. J., Qin, C., Trauger, S. A., Brandon, T. R., Custodio, D. E., Abagyan, R. und Siuzdak, G. METLIN: a metabolite mass spectral database. Ther.Drug Monit. 27[6], 747-751. 2005-12.

Smith, C. A., Want, E. J., O'Maille, G., Abagyan, R. und Siuzdak, G. XCMS: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. Anal.Chem. 78[3], 779-787. 2006-02-01.

SQL-Injection, http://de.wikipedia.org/wiki/SQL-Injection, zuletzt besucht: 2010-03-03

SQL-Normalformen, http://reeg.junetz.de/DSP/node7.html#SECTION033400000000000000000 , zuletzt besucht: 2010-04-04

The Columbia Encyclopedia, http://www.encyclopedia.com/topic/metabolism.aspx, zuletzt besucht: 2009-11-11

Villas-Bôas, S. G., Nielsen, J. und Smedsgaard, J. Metabolome Analysis: An Introduction. 17. 2007.

Weininger, David. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. Journal of Chemical Information and Computer Sciences 28[1], 31-36. 1988-01-01.

Weininger, David. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. Journal of Chemical Information and Computer Sciences 28[1], 31-36. 2002-05-01.

Wishart, D. S., Knox, C., Guo, A. C., Eisner, R., Young, N., Gautam, B., Hau, D. D., Psychogios, N., Dong, E., Bouatra, S., Mandal, R., Sinelnikov, I., Xia, J., Jia, L., Cruz, J. A., Lim, E., Sobsey, C. A., Shrivastava, S., Huang, P., Liu, P., Fang, L., Peng, J., Fradette, R., Cheng, D., Tzur, D., Clements,

M., Lewis, A., De Souza, A., Zuniga, A., Dawe, M., Xiong, Y., Clive, D., Greiner, R., Nazyrova, A., Shaykhutdinov, R., Li, L., Vogel, H. J. und Forsythe, I. HMDB: a knowledgebase for the human metabolome. Nucleic Acids Res. 37[Database issue], D603-D610. 2009-01.

Wishart, D. S., Tzur, D., Knox, C., Eisner, R., Guo, A. C., Young, N., Cheng, D., Jewell, K., Arndt, D., Sawhney, S., Fung, C., Nikolai, L., Lewis, M., Coutouly, M. A., Forsythe, I., Tang, P., Shrivastava, S., Jeroncic, K., Stothard, P., Amegbey, G., Block, D., Hau, D. D., Wagner, J., Miniaci, J., Clements, M., Gebremedhin, M., Guo, N., Zhang, Y., Duggan, G. E., Macinnis, G. D., Weljie, A. M., Dowlatabadi, R., Bamforth, F., Clive, D., Greiner, R., Li, L., Marrie, T., Sykes, B. D., Vogel, H. J. und Querengesser, L. HMDB: the Human Metabolome Database. Nucleic Acids Res. 35[Database issue], D521-D526. 2007-01.

Xampp, http://www.apachefriends.org/de/xampp-windows.html, zuletzt besucht: 2010-03-03