Integrierte bioinformatische Methoden zur reproduzierbaren und transparenten Hochdurchsatz-Analyse von Life Science Big Data

INAUGURALDISSERTATION

zur Erlangung des Doktorgrades

der Fakultät für Chemie und Pharmazie

der Albert-Ludwigs-Universität Freiburg i. Br.

vorgelegt von

Björn Andreas Grüning

2015

Vorsitzender des Promotionsausschusses:	Prof. Dr. Stefan Weber Universität Freiburg Institut für Physikalische Chemie Physikalische Chemie Albertstraße 21 79104 Freiburg
Referent:	Jun.-Prof. Dr. Stefan Günther Universität Freiburg Institut für Pharmazeutische Wissenschaften Pharmazeutische Bioinformatik Hermann-Herder-Strasse 9 79104 Freiburg
Korreferent:	Prof. Dr. Michael Müller Universität Freiburg Institut für Pharmazeutische Wissenschaften Pharmazeutische und Medizinische Chemie Albertstr. 25 79104 Freiburg
Drittprüfer:	Prof. Dr. Rolf Backofen Universität Freiburg Institut für Informatik Lehrstuhl für Bioinformatik Georges-Köhler-Allee 106 79110 Freiburg
Prüfungsdatum:	26.10.2015

Danksagung

Diese Arbeit ist für und mit einer starken Community entstanden, der ich an dieser Stelle danken und meine Wertschätzung aussprechen möchte.

Danke an die Open Source Community, insbesondere alle Personen, die mit und an Galaxy arbeiten, die mich mit ihren moralischen und ethischen Vorstellungen jeden Tag aufs neue motivieren. Mein besonderer Dank gilt hierbei den IUC-Leuten, die unzählige Stunden ihrer Freizeit damit zubringen Tools besser zu machen, Fragen zu beantworten und best-practice guides zu erstellen. Danke!

Dem Team der Freiburger Galaxy-Instanz möchte ich danken für all die Arbeit, die es im Hintergrund leistet, obwohl sie nicht mit dem üblichen Reputationssystem der Wissenschaft entlohnt werden können. Für diese idealistische Arbeit danke ich Euch: Stefan Jankowski, Pavankumar Videm, Cameron Smith, Dr. Torsten Howaart, Dr. Anika Erxleben und Prof. Dr. Rolf Backofen!

Mein besonderer Dank gilt auch den Galaxy-Usern und Kooperationspartnern, die mich stets mit neuen Ideen fordern und so unglaublich inspirierend sind. Stellvertretend möchte ich mich bei PD Dr. Ralf Gilsbach, Dr. Sebastian Preissl, Dr. Andreas Präg, Dr. Julia Wunsch-Palasis, Dr. Sabrina Haßler und Dr. Alexander Fries, Dr. Deborah Roidl, Patrick Bovio, Dr. Thomas Wecker, Steffen Lott und Dan Miller bedanken.

Für die Unterstützung bei der Organisation von nationalen und internationalen Galaxy-Workshops, möchte ich mich recht herzlich bei Dr. Hans-Rudolf Hotz und der Bioinformatics Facility des MPI-IE in Freiburg bedanken. Dr. Thomas Manke, Dr. Fidel Ramirez, Sarah Diehl, Dr. Friederike Dündar, Dr. Devon Ryan, Dr. Fabian Kilpert, Dr. Hans-Rudolf Hotz - es ist grandios mit Euch zu arbeiten und die bioinformatische Lehre zu verbessern!

Für die angenehme und stets abwechslungsreiche Arbeitsatmosphäre möchte ich mich bei meinen Kollegen und ehemaligen Kollegen der Pharmazeutischen Bioinformatik bedanken: Kersten Döring, Dr. Christian Senger, Dr. Xavier Lucas, Dr. Hitesh Patel, Stephan Flemming und insbesondere bei Dr. Anika Erxleben für ihr Unterstützung in jeglicher Hinsicht.

Bedanken möchte ich mich auch bei meinem Betreuer Jun.-Prof. Dr. Stefan Günther. Es war eine sehr schöne Erfahrung ihn von Anfang an beim Aufbau der Pharmazeutischen Bioinformatik, Uni-Freiburg, zu begleiten und ich kann die Freiheiten, die er mir gewährt hat nicht hoch genug würdigen: Danke Stefan!

Herzlichen Dank auch an Prof. Dr. Michael Müller für die Übernahme des Korreferats, der Offenheit gegenüber dem Thema und der Problemstellung und so manch fruchtvoller Diskussion. Danke auch für Deine Unterstützung im außerwissenschaftlichen universitären Betrieb!

Prof. Dr. Rolf Backofen möchte ich zum einen danken, dass er die Rolle des Drittprüfers übernommen hat, zum anderen aber auch für das Vertrauen und die immense Unterstützung, die er mir und der Galaxy-Freiburg Idee entgegen gebracht hat.

Den letzten Absatz dieser Danksagung möchte ich allen widmen, die mich seit jeher unterstützt haben und die letzten Jahre zu einem wundervollen Erlebnis gemacht haben. Meinen Freunden, meinen Eltern, meiner Schwester Anja mit ihrer wachsenden Familie, ja meiner ganzen Familie! Meli und Ella für die wunderschöne Zeit, den Spaß und für eine kostbare Freundschaft; Alexander Burghardt, fürs 'Alex' sein!

Inhaltsverzeichnis

Abbildungsverzeichnis **vi**

Tabellenverzeichnis **vii**

Abkürzungsverzeichnis **ix**

Zusammenfassung **xi**

I Einleitung **1**

1 Anforderungen moderner Bioinformatik **2**

1.1 Reproduzierbarkeit wissenschaftlicher Ergebnisse 4

1.2 Transparenz bei der Auswertung wissenschaftlicher Daten 5

1.3 Zugänglichkeit bio- und cheminformatischer Methoden 7

2 Galaxy - ein modulares, integratives Framework **10**

2.1 Galaxy für Lebenswissenschaftler 11

2.2 Galaxy für Bioinformatiker 13

II Ergebnisse **15**

1 Genomanalyse **16**

1.1 Automatisierte Genomanalyse 19

1.1.1 Methoden 19

1.1.1.1 Aufbereitung der Rohdaten 19

1.1.1.2 RNA- und Genvorhersage 19

1.1.1.3 Annotationstransfer zwischen ähnlichen Sequenzen 21
1.1.1.4 Identifikation von Sequenzsignaturen 23
1.1.1.5 Genclustervorhersage 23
1.1.1.6 Veröffentlichung der Sequenzdaten in frei zugänglichen Datenbanken 23
1.1.2 Ergebnisse 29
1.1.2.1 *Streptomyces sp.* Tü6071 31
1.1.2.2 *Streptomyces viridochromogenes* Tü57 32
1.1.2.3 *Streptomyces aurantiacus* JA 4570 33
1.1.2.4 *Streptomyces afghaniensis* NC 5228T 33
1.1.2.5 *Streptomyces spectabilis* DSM 40779 33
1.1.2.6 *Glarea lozoyensis* ATCC 74030 34
1.1.3 Diskussion 34
1.2 Proteine in wissenschaftlichen Abhandlungen 36
1.2.1 Daten und Methoden 38
1.2.1.1 PubMed-Parser 38
1.2.1.2 Datenbanken 39
1.2.2 Ergebnisse 41
1.2.2.1 Beziehung zwischen Abstracts, Kleinstruktur- und Proteinsynonymen 43
1.2.2.2 Validierung der erhobenen Daten 43
1.2.2.3 Visualisierung 44
1.2.2.4 Interaktion zwischen Prolific und CIL 46
1.2.3 Diskussion 47

2 Analyse des chemischen Raums **48**
2.1 ChemicalToolBox 49
2.1.1 Methoden 51
2.1.1.1 Galaxy-Integration 51
2.1.1.2 Einbindung in die Galaxy Tool Shed 55
2.1.2 Ergebnisse 58
2.1.2.1 Tools in der ChemicalToolBox 58
2.1.2.2 ChemicalBox und PurchaseableBox 66

2.1.3 Diskussion 67
2.2 Kleinmoleküle in wissenschaftlichen Abhandlungen 70
2.2.1 Daten und Methoden 72
2.2.2 Ergebnisse 73
2.2.2.1 Beziehung zwischen Abstracts, Protein- und Kleinstruktursynonymen 75
2.2.2.2 Interaktion zwischen CIL und Prolific 76
2.2.3 Diskussion 76
2.3 StreptomeDB: Eine Datenbank von Naturstoffen aus Streptomyceten 78
2.3.1 Daten und Methoden 79
2.3.2 Ergebnisse 82
2.3.3 Diskussion 84

III Abschlussbetrachtung 86
3.1 Zugänglichkeit bio- und cheminformatischer Methoden 87
3.2 Verbesserung der Lehre 87
3.3 Nachhaltigkeit durch den Aufbau von Communitys 87
3.4 Reproduzierbarkeit wissenschaftlicher Ergebnisse 88
3.5 Text Mining und Datenbanken 89
3.6 Ausblick 90

Referenzen 96

Abbildungsverzeichnis

1 Speicherplatzkosten im Verhältnis zu DNA-Sequenzierkosten 3

2 Anstieg der öffentlich zugänglichen Sequenzen 17
3 Preisbestimmende Faktoren eines Sequenzierungsprojektes 18
4 Flussdiagramm der Genomanalyse-Pipeline 20
5 *Glimmer*-Genomannotation für Prokaryoten 22
6 Genclustervorhersage basierend auf antiSMASH 25
7 Genomannotations-Workflow mit Genclustervorhersage 26
8 Genomanalyse-Pipeline als Galaxy-Workflow 27
9 Genom-Browser-Integration in Galaxy 29
10 Auffinden zweier benachbarter Gene 30
11 Zirkulärer Genomplot von *Streptomyces sp.* Tü6071 31
12 Zirkulärer Genomplot von *Streptomyces viridochromogenes* Tü57 32
13 Prolific-Flussdiagramm 42
14 Prolific-Heatmap 45
15 Prolific-Detailansicht 46

16 Abhängigkeitsgraph der ChemicalToolBox 56
17 Automatische Konvertierung von Molekülformaten 60
18 Semiautomatische Konvertierung von Molekülformaten 61
19 Filtern von chemischen Bibliotheken 62
20 Filter-Tool mit verschiedenen benutzerdefinierten Regeln 63
21 Workflow zur Erstellung der ChemicalBox 69
22 CIL-Workflow 73
23 ChemicalToolBox-Workflow zur Identifikation von MCSS 81

24 MCSS der StreptomeDB 82
25 Exemplarische Naturstoffe in der StreptomeDB 83
26 Modulare virtualisierte Galaxy-Installationen 91

Tabellenverzeichnis

1.1 MeSH-Terme der *Streptomyces sp.* Tü6071-Publikation 37
1.2 Identifizierte Wirkstoff-Target-Relationen der DrugBank 43

2.1 Exemplarische Liste von Programmen in der ChemicalToolBox 59
2.2 PubMed-Statistik der CIL-Datenbank 75
2.3 PubChem-Statistik der CIL-Datenbank 75
2.4 UniProt-Statistik der CIL-Datenbank 76
2.5 StreptomeDB: Anzahl der unterschiedlichen Synthesewege 80
2.6 StreptomeDB: Entitäten und ihre Häufigkeit 84

Auflistungsverzeichnis

2.1 XML-basierte Softwarebeschreibung für Galaxy 50
2.2 Dynamische Adaption des Ausgabeformates 54
2.3 Installationsbeschreibung von Open Babel 57

Abkürzungsverzeichnis

CIL	Compounds in Literature
CLI	Command-line interface steht für die Kommandozeile in Unix-artigen Systemen; ein textbasiertes Interface zur Steuerung von Programmen.
CML	Chemical Markup Language; XML-Repräsentation von molekularen Informationen
DBMS	Datenbankmanagementsystem
ELIXIR	European life-sciences infrastructure for biological information
FDA	Food and Drug Administration; Arzneimittelzulassungsbehörde der Vereinigten Staaten von Amerika
FTP	File Transfer Protocol; Netzwerkprotokoll zur Übertragung von Dateien
GOA	Gene Ontology-Annotation
GOBLET	Global Organisation for Bioinformatics Learning, Education and Training
HPC	High-Performance-Computing; Hochleistungsrechnen
HTS	High-throughput screening oder High-throughput sequencing
ID	Identifikator
InChI	IUPAC International Chemical Identifier; eindeutige Repräsentation von chemischen Strukturen
IQR	Interquartilsabstand
IUC	Intergalactic Utilities Com-mission
IUPAC	International Union of Pure and Applied Chemistry; Internationale Union für reine und angewandte Chemie
KEGG	Kyoto Encyclopedia of Genes and Genomes; Datenbank mit Informationen über Stoffwechselwege, Gene und Biomoleküle
MCSS	Most Common Substructure Search
MeSH	Medical Subject Headings; kontrolliertes Vokabular zur Sacherschließung von wissenschaftlichen Abhandlungen
NIH	National Institutes of Health
NLM	National Library of Medicine®
NRPS	Nichtribosomale Peptidsynthetase
ORDBMS	Objektrelationales Datenbankmanagementsystem

ORM Object-Relational Mapping; Objektabbildung einer Programmiersprache in eine relationale Datenbank

PKS Polyketidsynthase

QED Quantitative estimate of drug-likeness

Ro5 Lipinski Rule of Five; Regeln zur Abschätzung der oralen Bioverfügbarkeit einer chemischen Verbindung

RPS Ribosomale Peptidsynthetase

SAX Simple API for XML; ereignisorientierte Programmierschnittstelle für das Parsen von XML-Dateien; Mehrschritt-Push-Parser

SMARTS Smiles Arbitrary Target Specification; System zur Beschreibung von regulären Ausdrücken für chemische Strukturen

SMILES Simplified Molecular Input Line Entry System; lesbare String-Repräsentation von chemischen Strukturen

SOAP Simple Object Access Protocol; Protokoll zum Austausch von Daten zwischen Systemen

SRA Sequence Read Archive; Datenbank des National Center for Biotechnology Information

XML Extensible Markup Language

Zusammenfassung

High-throughput-Techniken haben die Lebenswissenschaften in eine neue Ära geführt. Die Genomsequenzierung von Patienten ist in der alltäglichen Praxis der ersten Universitätskliniken angekommen, ebenso wie das systematische *in vitro* Screening von hunderttausenden Kleinstrukturen, welches eine Standardmethode für die Identifikation neuer Wirkstoffe in der Pharmaindustrie geworden ist. Das damit einhergehende Anwachsen von resultierenden Daten und deren hohe Komplexität stellt die Forschung vor neue Herausforderungen und bringt sie unweigerlich näher an die Informationstechnik.

Die vorliegende Arbeit beschäftigt sich mit der Analyse von Life Science Big Data mit dem Fokus auf Transparenz und Reproduzierbarkeit der eingesetzten Methoden. Aufbauend auf einem Framework zur Datenprozessierung (Galaxy) wurde unter anderem eine Analyseplattform für die Untersuchung von Genomen und chemischen Kleinstrukturen entwickelt.

Für die Genomanalyse wurde eine Vielzahl an Funktionen, angefangen von der funktionellen Annotation von pro- und eukaryotischen Genomen bis hin zur Aufbereitung der Sequenzen für ihre Veröffentlichung in frei verfügbaren Datenbanken entwickelt. Populäre Tools wie BLAST wurden dabei ebenso in Galaxy integriert wie spezialisierte Tools z. B. zur Genclustervorhersage. Die Flexibilität und Stärke dieser Analyseumgebung wurde anhand von sechs annotierten Organismen und einer Reihe pharmazeutisch relevanter Galaxy-Workflows gezeigt.

Zur Analyse und Prozessierung von Kleinstrukturen wurden cheminformatische Methoden und Workflows entwickelt, die in der Lage sind, Millionen von chemischen Strukturen zu charakterisieren. Die dadurch mögliche Klassifizierung in z. B. Wirkstoff- oder Naturstoffähnlichkeit konnte genutzt werden, um optimierte Molekülbibliotheken für das *in silico* Wirkstoffdesign zu erstellen.

Im Rahmen dieser Arbeit konnten die Wissenschaftsfelder Genomik und Cheminformatik in einer universellen Open Source-Analyseplattform vereinigt werden. Diese wurde zudem als Portal für die interdisziplinäre Datenanalyse an der Universität Freiburg ausgebaut und erfolgreich in der bioinformatischen Lehre eingesetzt. Die vorgestellten Resultate belegen, dass reproduzierbares und transparentes wissenschaftliches Arbeiten auch mit Big Data realisierbar ist und zeigen zudem, dass es Galaxy den Naturwissenschaftlern ermöglicht die Datenauswertung eigenständig durchzuführen.

Teil I

Einleitung

1

Anforderungen moderner Bioinformatik

Das Fehlen von benutzerfreundlicher Software, einhergehend mit unzureichenden Dokumentations- und Schulungsmaterialien, verhindert oft die Partizipation des Naturwissenschaftlers an der Auswertung der eigenen Daten.

Mit wachsender Komplexität der Daten und der damit immer größeren Notwendigkeit, hochkomplexe Analysemethoden zu verwenden, gerät der Wissenschaftler immer öfter in ein Abhängigkeitsverhältnis zu Personen, die diese Methoden bzw. Programme implementieren und bedienen können. Viele dieser Programme sind nicht auf Windows verfügbar, und die wenigsten haben eine graphische Benutzeroberfläche. Beides sind häufig Ausschlusskriterien für viele Lebenswissenschaftler.

Doch nicht nur die Komplexität wächst, auch die Menge der produzierten Daten steigt exponentiell an und ein Ende ist nicht in Sicht (siehe Abbildung 1). Allein die Menge an Daten hat die Bioinformatik in den letzten Jahren näher an die traditionelle IT herangebracht. Neue Speichersysteme mussten entwickelt und integriert werden, globale Partner wie Google™, Amazon™ und SAP™ investieren und helfen diese biologische Datenflut adäquat zu speichern. Auf dieser Ebene geht es noch gar nicht um die Auswertung, sondern um die pure Speicherung, sowie um den Transfer der Daten. Die eigene Fachbezeichnung *Data Scientist* ist in den letzten Jahren immer populärer geworden und mittlerweile ein fester Bestandteil der Bioinformatik, ebenso der Begriff *Big Data*, der Datenmengen beschreibt, die zu groß oder zu komplex sind, um sie mit herkömmlichen Methoden zu analysieren.

Die Auswertung dieser Daten stellt das nächste Problem dar. Algorithmen, die vor ein paar Jahren noch zur Analyse ausreichten, sind heutzutage schlichtweg zu langsam. Immer öfter muss ein Kompromiss zwischen Exaktheit und Geschwindigkeit der Berechnungen getroffen werden. BLAST (1, 2), das wohl bekannteste bioinformatische Programm, wurde in den letzten zwei Jahrzehnten so oft benutzt, dass das Verb *blasten* schon in unseren Sprachgebrauch übergegangen ist. BLAST ist bekannt für eine hohe Genauigkeit bei der Sequenzsuche und ist performant genug, um Genome zu durchsuchen. Mit dem Aufkommen der Metagenomics, bei denen mehrere 100 bis 1000 Genome gegen alle anderen bekannten Genome verglichen werden, ist BLAST aber zu langsam und muss zwangsläufig durch neuere Werkzeuge ersetzt werden (3, 4).

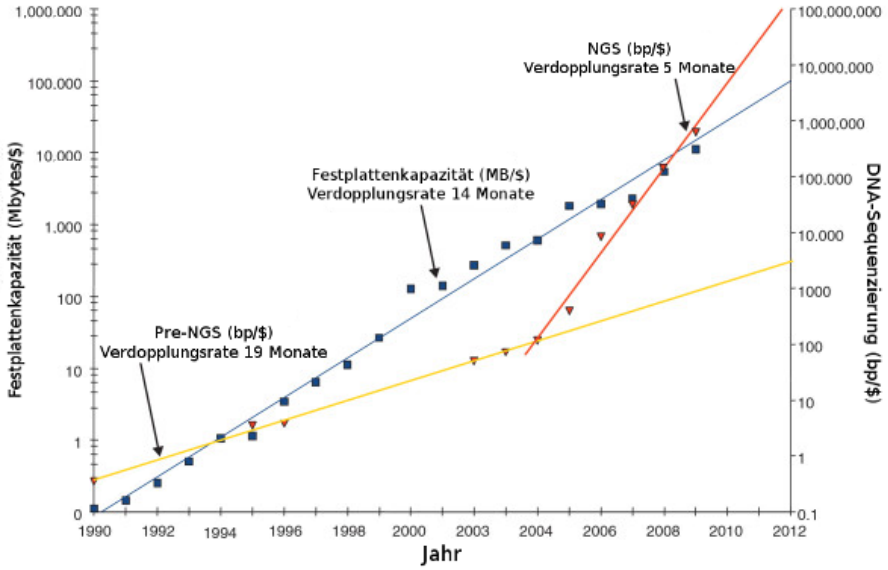

Abbildung 1: Speicherplatzkosten im Verhältnis zu DNA-Sequenzierkosten - In blauen Kästchen ist der Festplattenpreis in Megabytes pro Dollar angegeben. Die approximierte blaue Linie zeigt das exponentielle Wachstum mit einer Verdopplungsrate von ca. 1,5 Jahren. Die Sequenzierkosten (rote Linie) sind in Basenpaare pro Dollar angegeben und zeigen in der gelben Linie ebenfalls einen exponentiellen Verlauf. Allerdings ist dieser geringer in der Verdopplungsrate als die des Festplattenpreises. Mit dem Aufkommen der *High-throughput*-Sequenzierung ab 2004 änderte sich der Anstieg drastisch und die Verdopplungsrate fiel auf fünf Monate (5).

Dies bedeutet im Umkehrschluss, dass die datenerhebenden Wissenschaftler einen Zugang zu großen Datenspeichern und viel Rechenleistung brauchen - Rechenzentren. Eine Analyse auf einem Standardcomputer ist zumindest mit *High-throughput*-Daten nicht mehr möglich. Rechenzentren wiederum bieten sehr spezielle Zugangsmöglichkeiten zu

ihren Services an. Meistens sind Unix-Kenntnisse Voraussetzung und man muss mit einem *Scheduler* vertraut sein. Dies alles ist nicht benutzerfreundlich und verlangt von den Wissenschaftlern ein Verständnis von *High-Performance-Computing*, das die meisten scheuen. Die Einstiegshürden sind zu hoch, was viele in eine ähnliche Abhängigkeit treibt wie schon erwähnt. Abhängigkeiten, oder freundlicher formuliert Kooperationen, müssen nichts Negatives sein. Im Gegenteil: Kooperationen sind notwendig und müssen in einer Wissenschaft, die immer interdisziplinärer und spezialisierter wird, gefördert und ausgebaut werden. Eine Kooperation aber indirekt zu erzwingen, weil man den Zugang zu Technologien erschwert, ist unwissenschaftlich und sollte verhindert werden.

Die Bioinformatik sollte es daher schaffen, die verschiedenen Analysemethoden und verfügbaren Daten soweit zugänglich zu machen, dass jeder Wissenschaftler den Großteil seiner Ergebnisse selbstständig auswerten kann, ohne in alternativlose Abhängigkeiten zu geraten.

Reproduzierbarkeit, Transparenz der Analyse und der Zugang zu Daten und Software sind Schlüsselprobleme der Bioinformatik, die im Folgenden näher beleuchtet werden.

1.1 Reproduzierbarkeit wissenschaftlicher Ergebnisse

"The separation of software development from clinical research is costing lives."[1] Mit diesem Satz hat *The scientist* die Diskussionen der letzten Jahre über Reproduzierbarkeit und Verlässlichkeit von modernen Lebenswissenschaften auf den Punkt gebracht.

Es geht im Allgemeinen um mehr als Publikationen. Die gewonnenen publizierten Ergebnisse sind die Basis weiterer Untersuchungen. Sie tragen zum generellen Verständnis unserer Umwelt bei und haben optimalerweise einen positiven Effekt auf unsere Gesellschaft. Umso erschreckender ist das Ergebnis einer Studie von Prinz *et al.*, die 67 veröffentlichte Projekte auf ihre Reproduzierbarkeit hin untersuchte und feststellte, dass dies nur bei 25 % der Fall war (6). Darauf aufbauende Sekundärpublikationen manifestieren nicht nur falsche Erkenntnisse, sie verschwenden auch Zeit und Steuergelder. Nach einer Studie von Begley und Ellis haben nicht reproduzierbare Ergebnisse aus Publikationen eine nicht unerhebliche Anzahl an Zitationen, im Durchschnitt sogar mehr als die reproduzierbaren (7).

[1] `http://www.the-scientist.com/?articles.view/articleNo/35249/title/Researchers--Hire-Hackers`

Dabei spielt es keine Rolle, wo im wissenschaftlichem Spektrum sich der Wissenschaftler bewegt. In der *Target Discovery*, im *Drug Screening* oder in der Grundlagenforschung. Am Ende des Spektrums steht sehr häufig die klinische Studie bzw. der Mehrwert für unsere Gesellschaft. Gerade hier konnte aber gezeigt werden, dass die Erfolgsrate von klinischen Studien in der Phase II von 28 % auf 18 % gefallen ist (8). Ein besorgniserregender Prozess mit mannigfaltigen Ursachen, denen es gilt Einhalt zu gebieten. Schon 2005 konnte gezeigt werden, dass wissenschaftliche Journals dazu tendieren "hippe", publikumswirksame, positive Resultate zu publizieren (9) - auf Kosten der Reproduzierbarkeit. Die großen Verlage scheinen sich dem Problem auch bewusst zu werden und starten Initiativen ihrerseits, um zumindest den Review-Prozess zu verbessern (10).

Der gesamte ökonomische Schaden wird auf 28.000.000.000 $ geschätzt, und das nur in den Vereinigten Staaten von Amerika (11). 26 % davon sind allein der inadäquaten Datenanalyse geschuldet. Nicht reproduzierbare wissenschaftliche Veröffentlichungen sind daher nicht nur teuer für den Steuerzahler, sie kosten auch Zeit, die anderweitig eingesetzt, Leben retten könnte.

Die Deutsche Forschungsgemeinschaft, als Mitunterzeichnerin der "Berlin Declaration" für Open Access[1] und als Herausgeberin der "Vorschläge zur Sicherung guter wissenschaftlicher Praxis"[1] sowie die National Institutes of Health (NIH) (12) und eine Gruppe namhafter Journals (13) haben die Probleme erkannt und klare Positionen bezogen.

1.2 Transparenz bei der Auswertung wissenschaftlicher Daten

Transparenz ist eng mit der Reproduzierbarkeit von Analysen verknüpft. Ist eine Analyse nicht transparent, kann sie nicht oder nur unzureichend reproduziert werden. Das Ziel einer reproduzierbaren bioinformatischen Analyse sollte es daher sein, ein Experiment in ausreichend detailliertem Umfang zu beschreiben. Dieser Umfang ist jedoch nicht näher definiert und Journals bzw. Reviewer haben ihre eigene Vorstellung davon. In den meisten Fällen ist diese nicht ausreichend und führt zu Resultaten wie in Kapitel 1.1 beschrieben.

Projekte wie Galaxy oder das neugegründete *Software Sustainable Institute*[1] versuchen dies weitaus engmaschiger zu definieren. So müssen nicht nur die Programmversion, son-

[1] `http://openaccess.mpg.de/Berlin-Declaration`
[1] `http://bit.ly/1U0PSA8`
[1] `http://www.software.ac.uk`

dern auch alle Parameter eines Programms detailliert protokolliert werden. Alle Eingaben, wie Annotationsdaten oder Datenbanken, müssen genauso archiviert werden wie die Abfolge der einzelnen Analyseschritte. Dabei stellt das Archivieren von Eingabe-Datensätzen das erste größere Problem dar.

Zum einen können die Rohdaten von *High-throughput*-Techniken mehrere 100 GB groß sein, zum anderen verändern sich die Annotationsdaten fast täglich und müssen archiviert werden. Im Falle des BLAST-Beispiels bedeutet dies, dass der Wissenschaftler jede Datenbank gegen die er gesucht hat, prinzipiell archivieren muss. Andernfalls ist die Reproduzierbarkeit der Ergebnisse nicht gewährleistet. Das wiederum bedeutet, dass für eine reproduzierbare und transparente Analyse keine Webservices genutzt werden dürfen, da diese einem ständigen Update unterliegen.

Selbst wenn ein Experiment komplett beschrieben und archiviert werden kann, müssen all diese Daten auch der Community zur Verfügung gestellt werden. Bei solch großen Datenmengen steht die Wissenschaftscommunity aber vor infrastrukturellen Problemen. Auch hier wird oft ein Kompromiss zwischen der Archivierung von allen Daten und daraus entstehenden Kosten getroffen. So hat Chad Nusbaum vom Broad Institute[1] schon 2010 festgestellt, dass die Archivierung der Rohdaten teurer sei als eine Neusequenzierung[1].

Aber nicht nur die Daten müssen archiviert werden, sondern auch die Programmversionen. Software ist nicht nachhaltig. Es ist nicht gewährleistet, dass eine Software, die auf einem heutigen Computer installiert werden kann, auch noch in 5 Jahren installierbar ist. Es kann nicht einmal garantiert werden, dass man diese Software in einer speziellen Version zu einem beliebigen Zeitpunkt in der Zukunft noch beziehen kann. Das heißt wiederum, dass bei einer Analyse alle Softwarekomponenten genauso archiviert werden müssen wie die Rohdaten. Im Extremfall bis auf die Ebene des Betriebssystems. Das proprietäre Software oder Dateiformate dies erschweren bzw. unmöglich machen, liegt auf der Hand.

Eine Archivierung des kompletten *Softwarestacks* ist mit heutigen Techniken prinzipiell möglich, allerdings setzt es voraus, dass alle Komponenten frei verfügbar sind und man Rechte zur Speicherung, Verteilung und Modifizierung besitzt. *Open Source* ist noch immer kein Standard in vielen wissenschaftlichen Einrichtungen und macht daher ein transparentes und reproduzierbares Arbeiten nahezu unmöglich. In “The case for open

[1] `http://www.broadinstitute.org`
[1] `http://www.bio-itworld.com/2010/issues/sept-oct/broad.html`

computer programs“ wird daher gefordert, dass für jegliche Ergebnisse, die abhängig von computergestützen Analysen sind, eine Herausgabe des Quelltextes unerlässlich sei (14).

Ein anderer Aspekt von transparenter Wissenschaft ist, dass es Wissenschaftlern und Reviewern ermöglicht werden soll, auf einfachste Art und Weise die Analyse zu begutachten. Das Ziel ist es, möglichst viele Menschen in eine Analyse einzubeziehen und ihnen schon vor der Publikation die Möglichkeit zu geben, Verbesserungsvorschläge zu machen und Fehler frühzeitig zu erkennen. Wissenschaftler sollen ihre Analysen untereinander begutachten, aber auch mit ihren Vorgesetzten teilen. Dies verbessert die Kommunikation und nach “Linus's Law“ auch die Fehler einer Analyse (15).

1.3 Zugänglichkeit bio- und cheminformatischer Methoden

Zugänglichkeit ist eine integrale Forderung an ein System, welches dem Wissenschaftler wieder mehr Kontrolle über die Analyse seiner Daten geben soll.

Sollte es gelingen ein System zu erschaffen, das die Forderungen an Reproduzierbarkeit und Transparenz erfüllt, so bleibt immer noch das Problem, dieses System so einfach zu gestalten, dass es von jedem bedient werden kann. Streben wir ein uniformes System an, welches verschiedene Lebenswissenschaften abdeckt und eventuell sogar darüber hinaus? Programme zur Analyse biologischer Daten müssen mit denen zur Auswertung von chemischen Daten kompatibel sein. Die Benutzerschnittstelle muss einheitlich und einfach verständlich sein.

Für computergestützte Analysen ist das Mittel der Wahl vieler Biologen eine Tabellenkalkulation. Diese ist aber für die großen Datenmengen, die immer häufiger anfallen, nur sehr eingeschränkt nutzbar. Komplexe Analysen sind, wie bereits ausführlich geschildert, nicht mehr auf dem Standardcomputer zu bewerkstelligen, sondern müssen in Rechenzentren getätigt werden.

Tabellenkalkulationen werden nicht benutzt, weil sie Vorteile in der Bedienung oder Installation gegenüber anderen Alternativen hätten, es ist vielmehr so, dass die Dokumentation meist sehr gut ist, es gibt Trainingskurse und sie sind auf allen Computern vorinstalliert. Dokumentation und Installation sind immanent für den Erfolg eines Programms. In der Tat ist immer öfter zu beobachten, dass nicht das “beste” Programm die meiste Verbreitung findet, sondern das Programm mit der besten Dokumention bzw. mit der einfachsten Benutzerschnittstelle. Programme müssen sich dem Benutzer und den

Laufzeitumgebungen flexibel anpassen. Damit verhält sich Software nach der Darwin'schen Evolutionstheorie *Survival of the Fittest*.

Dies erklärt auch die Popularität von Webservices. Das Journal *Nucleic Acid Research* veröffentlicht jährlich in einem *Special Issue* mehr als 100 davon (16). Webservices können mit einem Browser benutzt werden und erfordern daher in der Regel keine Installation. Die Dokumentation ist hinreichend gut, was zu einer im Allgemeinen guten Akzeptanz der Webservices bei Lebenswissenschaftlern führt. Diese internetbasierten Services sind gut zugänglich, aber in der klassischen Form nicht kompatibel mit den Forderungen nach Reproduzierbarkeit und Transparenz. Im Gegenteil, es gibt Fälle bei denen Programme als online Service publiziert werden, aber auch auf Anfrage nicht ohne Restriktionen erhältlich sind, wie z.B. bei Proteomaps (17) geschehen.

In einer Studie zur Langzeitverfügbarkeit von 927 Webservices, die zwischen 2003 und 2009 in *Nucleic Acid Research* publiziert wurden, konnte gezeigt werden, dass nur noch 72 % aller Services im Jahre 2011 verfügbar waren (18). Die Funktionstüchtigkeit konnte nur bei 45 % bestätigt werden. Anzumerken ist hier, dass weder die Korrektheit der Ergebnisse untersucht wurde, noch ob sie mit denen der Erstveröffentlichung übereinstimmen. Der Fairness halber sollte aber dazu gesagt werden, dass Reproduzierbarkeit bei Webservices nicht im Vordergrund steht. Benutzer sollen einen einfachen Zugang zu Daten und Software bekommen, das Updaten der jeweiligen Daten und der Software wird dem Webservicebetreiber überlassen und damit ein Teil der Verantwortung übertragen - mit allen negativen Implikationen für die Transparenz und Reproduzierbarkeit.

Ein weiteres Problem von Webservices sind die unzureichenden Interoperabilitäten der verschiedenen Anwendungen. Die Ein- und Ausgaben sind nicht standardisiert und lassen sich nicht ohne weiteres ineinander überführen. Es bleibt dem Benutzer überlassen, die Dateien von einem Server herunterzuladen, zu modifizieren und/oder konvertieren und in eine andere Anwendung zu importieren. Gerade diese Schritte sind aber mitunter die aufwendigsten. Die Handhabung von verschiedenen Dateiformaten ist nicht trivial. Für das Überführen des einen Formates in ein anderes erfordert es nicht selten Programmierkenntnisse, zumindest aber externe Programme, die sich der Webservice-Benutzer installieren muss. Das *Semantic Web* (19), welches diese Probleme lösen und Daten zwischen zwei Instanzen einfach austauschbar und wiederverwendbar machen sollte, hat sich bis jetzt nicht durchgesetzt (20).

Weiterhin sind klassische Webservices für große Datenmengen in der Regel nicht ausgelegt und haben meist Restriktionen auf eine maximale Eingabegröße oder eine Anzahl an parallelen Berechnungen. Dies sind verständliche Maßnahmen, wenn man berücksichtigt, wie aufwendig und teuer es ist, einen Webservice zu konzipieren und zu betreiben, der mit der Masse an Benutzern und Daten skalierbar ist.

Eine erstrebenswerte Lösung würde die Probleme von Webservices lösen, dabei aber die Vorteile der einfachen Zugänglichkeit bewahren und ausbauen. In Kapitel 2 wird ein solcher Lösungsansatz anhand des Galaxy-Frameworks vorgestellt.

Wir haben heutzutage kein Technologieproblem, wir haben ein Zugangsproblem zu Technologien. Die Entwicklung eines weiteren Programms nützt nur noch einem kleinen Teil der Wissenschaftsgemeinschaft, der Großteil hat keinen Zugang mehr dazu. Die vorliegende Arbeit zeigt Wege und Technologien auf, dies zu ändern.

2

Galaxy - ein modulares, integratives Framework als Lösungsansatz

Die vorliegende Arbeit stellt ein modulares, integratives Framework als Lösungsansatz für die beschriebenen bioinformatischen Probleme vor und die erfolgreiche praktische Anwendung auf verschiedenste Bereiche der genomischen Analyse und der Cheminformatik.

Bei diesem Framework handelt es sich um eine offene, webbasierte Plattform für transparente und reproduzierbare computergestützte Analysen, genannt Galaxy. Dabei ermöglicht es Galaxy, Benutzern ohne Programmier- oder weiterführende IT-Kenntnisse komplexe Programme zu bedienen und Arbeitsabläufe zu steuern. Galaxy speichert während einer Analyse alle Informationen, die nötig sind um diese zu einem beliebigem Zeitpunkt von einem beliebigen Benutzer zu reproduzieren. Alle Daten und Arbeitsschritte können mit beliebigen Nutzern geteilt und im Internet publiziert werden.

Galaxy ist aber auch eine Kommunikationsplattform, die Bioinformatiker und Lebenswissenschaftler näher zusammenbringt. In den nächsten zwei Abschnitten werden die Vorteile von Galaxy für Lebenswissenschaftler und Entwickler getrennt voneinander betrachtet. Äquivalent dazu wird sich Teil 1 tiefer mit der Benutzerinteraktion von Galaxy und Workflows auseinandersetzen und Teil 2 die technischen Hintergründe bei der Implementierung von Tools beleuchten.

2.1 Galaxy für Lebenswissenschaftler

Gibt man Kindern Legobausteine, bauen sie damit erstaunliche Sachen. Sie kombinieren die Steine nach Farbe und Form und erschaffen so interessante Gebilde. Diese Bausteine ermöglichen es, in einfachster Weise die Kreativität eines jeden zu entfesseln. Die Resultate sind in ihrer Komplexität frei skalierbar und nur abhängig von dem initialen Satz an Steinen und der eigenen Kreativität. Erhöht man die Diversität der Steine und damit die Entropie des ganzen Systems, lassen sich komplexere Gebilde erschaffen, ohne jedoch die Komplexität des elementaren Schrittes, das Kombinieren zweier Steine, zu erhöhen.

In der Software-Entwicklung ist dieser Gedanke ein Bestandteil der *Unix-Philosophie* (21), die unter anderem besagt, dass Computerprogramme nur eine Aufgabe erledigen sollen, diese aber besonders gut. Des Weiteren sollen Computerprogramme so geschrieben werden, dass sie einfach miteinander zusammenarbeiten sollen, als universelle Schnittstelle sollen Textströme dienen (22). Mit Hilfe der *Unix Pipes* lassen sich verschiedenste Programme frei kombinieren und komplizierte Operationen vornehmen. In der Softwareentwicklung hat dies noch den Vorteil, dass die einzelnen Programme, die Bausteine, nicht an Komplexität gewinnen, daher leichter zu warten sind und theoretisch weniger Fehler beinhalten. Wird eine neue Funktion benötigt, wird diese nicht in ein bestehendes Programm integriert und damit die Komplexität dieses Programms erweitert, sondern es wird ein neues Programm erstellt, welches mit allen anderen kommunizieren kann und auf diese eine Funktion spezialisiert ist.

Galaxy integriert diese Idee in ein Web-Framework, welches im Gegensatz zu den Unix-Programmen einfach zu benutzen und zu erlernen ist. Galaxy bietet hierzu eine Reihe von Abstraktionsebenen an, die die Komplexität moderner bioinformatischer Methoden vor dem Benutzer verbergen. Damit erhöht sich die Benutzerfreundlichkeit drastisch. Galaxy kann mit jedem Browser auf jedem Betriebssystem von jedem netzwerkfähigen Computer aus genutzt werden. Nutzt man eine von den etlichen frei zugänglichen Galaxy-Instanzen[1] oder eine Galaxy-Instanz der jeweiligen Universität, wie die der Universität Freiburg[1], so ist eine Installation nicht notwendig. Der Benutzer kann sofort mit der Analyse beginnen. Dies hat auch den Vorteil, dass eine Analyse unabhängig von dem Rechner ist, von

[1] `https://wiki.galaxyproject.org/PublicGalaxyServers`
[1] `http://galaxy.uni-freiburg.de`

dem sie gestartet wurde. Prozesse laufen weiter, auch wenn der Benutzer seinen Rechner ausschaltet oder ihn wechselt.

Eine weitere Abstraktionsebene stellt das Management von Daten in Galaxy dar. Dies nimmt, wie schon in Kapitel 1 erwähnt, einen immer größeren Platz in den Lebenswissenschaften ein. Galaxy abstrahiert Daten komplett von einem Dateisystem. Der Benutzer muss sich zu keinem Zeitpunkt darum kümmern, wie und wo seine Daten auf einen Dateisystem gespeichert werden. Daten werden in Galaxy in *Libraries* und in *Histories* organisiert und können optional mit Notizen oder Tags annotiert werden. Der Benutzer kann Daten von *Libraries*, der eigenen Festplatte oder externen Datenbanken in Galaxy einspeisen. Bei letzterem werden die Daten direkt ins Galaxy-System kopiert, eine Kopie auf dem Benutzerrechner entfällt.

Die in Kapitel 1.3 angesprochene Problematik der Dateiformate wird in Galaxy dahingehend gelöst, dass für jedes Format Konverter, Sniffer und Metadaten definiert werden können. Jede Galaxy-Instanz kann Datentypen installieren und damit die Funktionalität erweitern. Konverter können automatisch, ohne Benutzerinteraktion, verschiedene Dateitypen ineinander überführen, sollte dies von einem Programm gefordert werden. Sniffer können Datentypen automatisch beim Fileupload erkennen. Metadaten werden automatisch zu Datentypen hinzugefügt und können Indizes oder Zusatzinformationen wie Spaltenanzahl beinhalten. In Kapitel 2.1.2.1 wird dies anhand eines Beispiels für Datentypen der Cheminformatik näher erklärt.

Galaxy *Pages*, *Interactive Environments* oder die Visualisierung von Daten sind weitere Themen, die für Anwender von großer Bedeutung sind, an dieser Stelle aber nicht näher behandelt werden können. Ein weiterer Punkt, der für die Anwenderfreundlichkeit von Galaxy spricht, ist die starke Nachfrage in der Lehre. Als Mitglied des Galaxy Trainingsnetzwerkes[1] und der *Global Organisation for Bioinformatics Learning, Education and Training*[1] (GOBLET) wurde während des Zeitraumes dieser Dissertation an der Verbesserung der bioinformatischen Lehre gearbeitet. Diese basiert in weiten Teilen auf dem Konstruktionismus (23) und ermöglicht es Wissenschaftlern, sich Wissen selbstständig durch aktives Analysieren von Daten anzueignen.

[1] `https://wiki.galaxyproject.org/News/GalaxyTrainingNetwork`
[1] `http://www.mygoblet.org/`

2.2 Galaxy für Bioinformatiker

Entwickler und Bioinformatiker können in aller Regel Software installieren und kompilieren und sind es gewöhnt, mit verschiedenen Systemen, einschließlich Computercluster in Rechenzentren, zu arbeiten. Sie sind daher nicht so abhängig von einer Verbesserung des *Status quo* wie Lebenswissenschaftler. Trotz alledem, die zuvor besprochenen Probleme sollten Argument genug sein, um ein Paradigmenwechsel einzufordern.

Darüber hinaus gibt es eine ganze Reihe anderer attraktiver Gründe für ein Framework wie Galaxy zu entwickeln. Eines der wohl wichtigsten Argumente ist die Abstraktion des Prozesssteuerungssystems, welches für eine nahtlose Integration von bioinformatischen Anwendungen in *High-Performance-Computing* (HPC)-Umgebungen sorgt. Ein in Galaxy integriertes Programm kann auf einer Vielzahl von unterschiedlichen Systemen laufen und damit in Rechenzentren ebenso eingesetzt werden wie auf einem Standardcomputer oder kommerziellen Cloud-Angeboten. Dabei werden Techniken unterstützt, die das automatische Aufteilen der Eingabedaten ermöglichen und die Ergebnisse wieder automatisch zusammenführen. In der in Kapitel 2.1.1.1 beschriebenen ChemicalToolBox wurden diese Techniken implementiert und erfolgreich zur Beschreibung des kommerziellen chemischen Raums angewendet (24).

Einen weiteren Grund stellt das in Galaxy integrierte Deployment von Programmen und Datentypen dar. Bei der Integration von Tools besteht die Möglichkeit, Abhängigkeiten zu definieren und diese automatisch in Galaxy zu installieren. Hierbei können hochkomplexe, versionsgenaue Abhängigkeitsgraphen abgebildet werden (siehe Teil 2 16). Diese können anschließend in eine beliebige Galaxy-Instanz integriert werden. Hat der Entwickler dies einmal definiert, können seine Tools über die Galaxy Tool Shed (25) verteilt werden. Die Galaxy Tool Shed fungiert hierbei als Sammlung von Galaxy-Tools, die ein Galaxy-Administrator auf einfachste Weise installieren kann. Damit einhergehend ergibt sich eine Reihe von Synergieeffekten. Entwickler können auf bereits verfügbare Bibliotheken, Programme, Datentypen und Konvertierungstools zurückgreifen, ohne diese neu implementieren zu müssen. Dies erhöht die Produktivität der gesamten Community und verringert die potentiellen Fehler nach “Linus’s Law“.

Dieser einheitliche Weg Programme zu verteilen und Benutzern zugänglich zu machen, überzeugt immer mehr Anwendungsentwickler, Galaxy-Integrationen schon zum Zeitpunkt der Publikation anzubieten (26, 27, 28, 29).

Galaxy ermöglicht es einem Anwendungsentwickler somit, die eigenen Programme mit den bereits verfügbaren zu kombinieren, um transparente, reproduzierbare Workflows für Benutzer zu erstellen. Über Dateimanagement und eine gute Benutzerschnittstelle muss sich der Entwickler keine Gedanken machen, und für zeitintensive Rechenschritte kann Galaxy mit einem Computercluster verbunden werden.

Teil II

Ergebnisse

1

Genomanalyse

Mit dem Fortschritt der Sequenziertechniken der letzten Jahrzehnte und die damit einhergehenden fallenden Preise stieg die Zahl der Sequenzierung kompletter Genome exponentiell an (siehe Abbildung **??**). Sequenzdaten können so günstig und so schnell wie noch nie erhoben werden, dabei wird die Qualität immer besser. Die Abdeckung des Genoms (engl. coverage) ist so hoch, dass man statistisch signifikante Änderungen an einer einzelnen Base zwischen zwei Proben detektieren kann und so Rückschlüsse auf Mutationen möglich sind. Der eigentliche Schritt der Sequenzierung ist nicht mehr limitierend, es ist vielmehr die Probenaufbereitung und in einem hohen Maße die nachfolgende Auswertung der Daten.

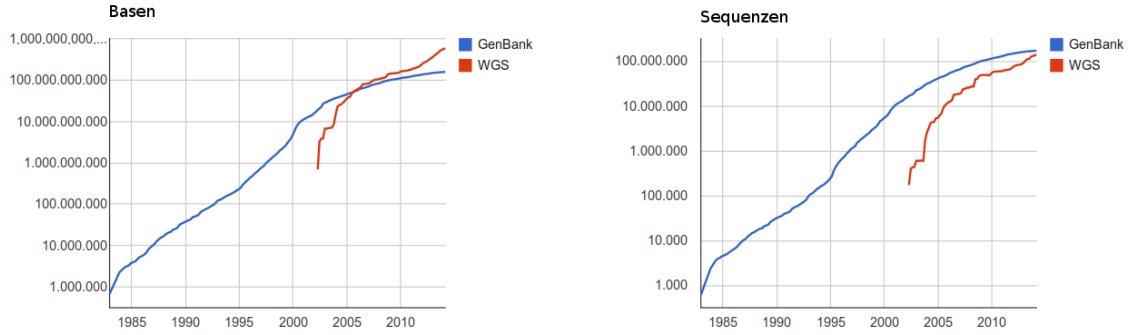

Abbildung 2: Anstieg der öffentlich zugänglichen Sequenzen - Die GenBank und die WGS-Datenbank des NCBI zeigen seit Jahren einen exponentiellen Anstieg sowohl in der Anzahl der Sequenzen als auch in der Anzahl der gespeicherten Basenpaare.[1]

War die Finanzierung der Sequenzierung vor 10 Jahren noch eines der größten Probleme eines Genom-Projektes, sind es heute die bioinformatischen Auswertungen, das Datenmanagement und das experimentelle Design eines Experiments (siehe Abbildung **??**). Die wahren Kosten eines Sequenzierungsprojektes haben sich seit der Einführung des *High-throughput-Screenings* extrem verschoben und dieser Trend wird nach Ansicht von Andrea Sboner und Kollegen auch die nächsten Jahre anhalten (30). Der Fokus geht immer mehr in Richtung Auswertung, und die größte Herausforderung der nächsten Jahre wird es sein, diese so intuitiv und zugänglich wie möglich zu gestalten.

Am Anfang einer jeden Sequenzanalyse müssen die Rohdaten, Reads genannt, in die richtige Reihenfolge gebracht werden. Ist ein Referenzgenom bekannt, kann dies durch ein

[1] `http://www.ncbi.nlm.nih.gov/genbank/statistics`

mapping gegen dieses geschehen. Ist keines bekannt, müssen die Rohdaten durch Assemblierung in die richtige Reihenfolge gebracht werden. Letzteres ist nicht nur aufwendiger, sondern bedeutet auch, dass keine Annotationen von einem Referenzgenom übernommen werden können.

Das Assemblieren des Genoms wird, gegen ein Entgelt, auch von Sequenzierfirmen angeboten und wird in dieser Arbeit nur angerissen. Der folgende Teil der Arbeit behandelt die Problematik der Annotation von *de novo* assemblierten Genomen, einem in Galaxy integrierten Workflow (Kapitel **??**) zur Genomanalyse und einem Webservice zur Proteinannotation durch vergleichende Suchen in Primärliteratur (Kapitel 1.2).

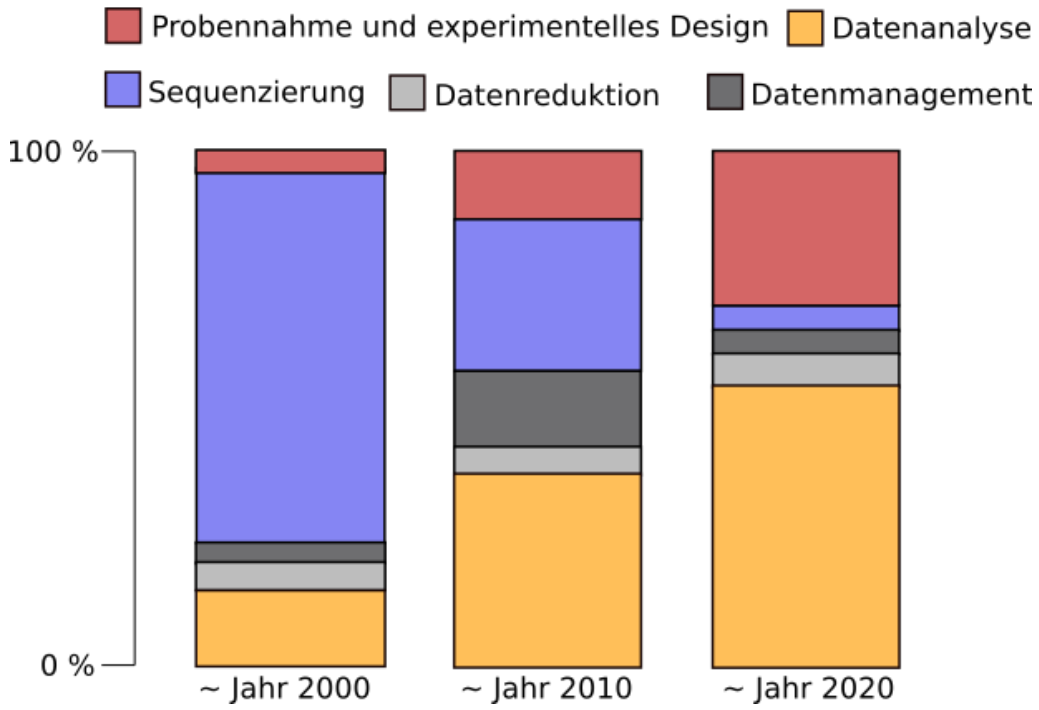

Abbildung 3: Preisbestimmende Faktoren eines Sequenzierungsprojektes - Die Kosten eines Sequenzierungsprojektes werden seit der Einführung von HTS-Techniken nicht mehr von der Sequenzierung selbst, sondern von den nachfolgenden Kosten der Datenanalyse dominiert (30).

1.1 Automatisierte Genomanalyse

Teile des folgenden Abschnittes wurden in peer-reviewed Journals veröffentlicht.

Peter J A Cock, Björn A Grüning, Konrad Paszkiewicz, und Leighton Pritchard. Galaxy tools and workflows for sequence analysis with applications in molecular plant pathology. *PeerJ*, 1:e167, January 2013. ISSN 2167-8359. doi: 10.7717/peerj.167

Peter J. A. Cock, John M. Chilton, Björn Grüning, James E. Johnson, und Nicola Soranzo. NCBI BLAST+ integrated into Galaxy. January 2015. doi: 10.1101/014043

1.1.1 Methoden

1.1.1.1 Aufbereitung der Rohdaten

Für die Verarbeitung der exponentiell wachsenden Sequenzdaten werden effiziente aber auch spezialisierte Programme in einer leistungsfähigen Infrastruktur benötigt. Im Rahmen des Genomanalyse-Pipeline-Projektes wurden zahlreiche Programme in das Galaxy-Framework integriert und fehlende Programme entwickelt, um eine umfassende und zugleich flexible Genomanalyse zu gewährleisten.

Die isolierte und aufgereinigte DNA wird in der Regel von einer Firma sequenziert, die auch eine Assemblierung des Genoms anbietet. Ist dies der Fall, bekommt der Wissenschaftler neben den Rohdaten ein bereits assembliertes Genom zurück und kann dieses direkt als Ausgangspunkt für die Annotation in Galaxy verwenden. Für den Fall, dass die Sequenzdaten nicht assembliert vorliegen oder die vorliegende Assemblierung nicht zufriedenstellend ist, wurden verschiedene Assembler in Galaxy integriert, die der Genomanalyse vorgeschaltet werden können. Die erhaltene genomische Sequenz sollte im ersten Schritt auf Kontaminationen getestet und bei Bedarf bereinigt werden. Das Programm *Remove contamination* identifiziert dabei Vektorsequenzen, Adapter, Linker oder Primer aus der UniVec-Datenbank, die üblicherweise bei der Klonierung von cDNA oder genomischer DNA Verwendung finden, und entfernt diese aus den Rohdaten.

1.1.1.2 RNA- und Genvorhersage

Ausgehend von der bereinigten genomischen Sequenz können verschiedene Programme herangezogen werden, um Gene oder RNAs vorherzusagen. *Aragorn* und *tRNAscan* sind

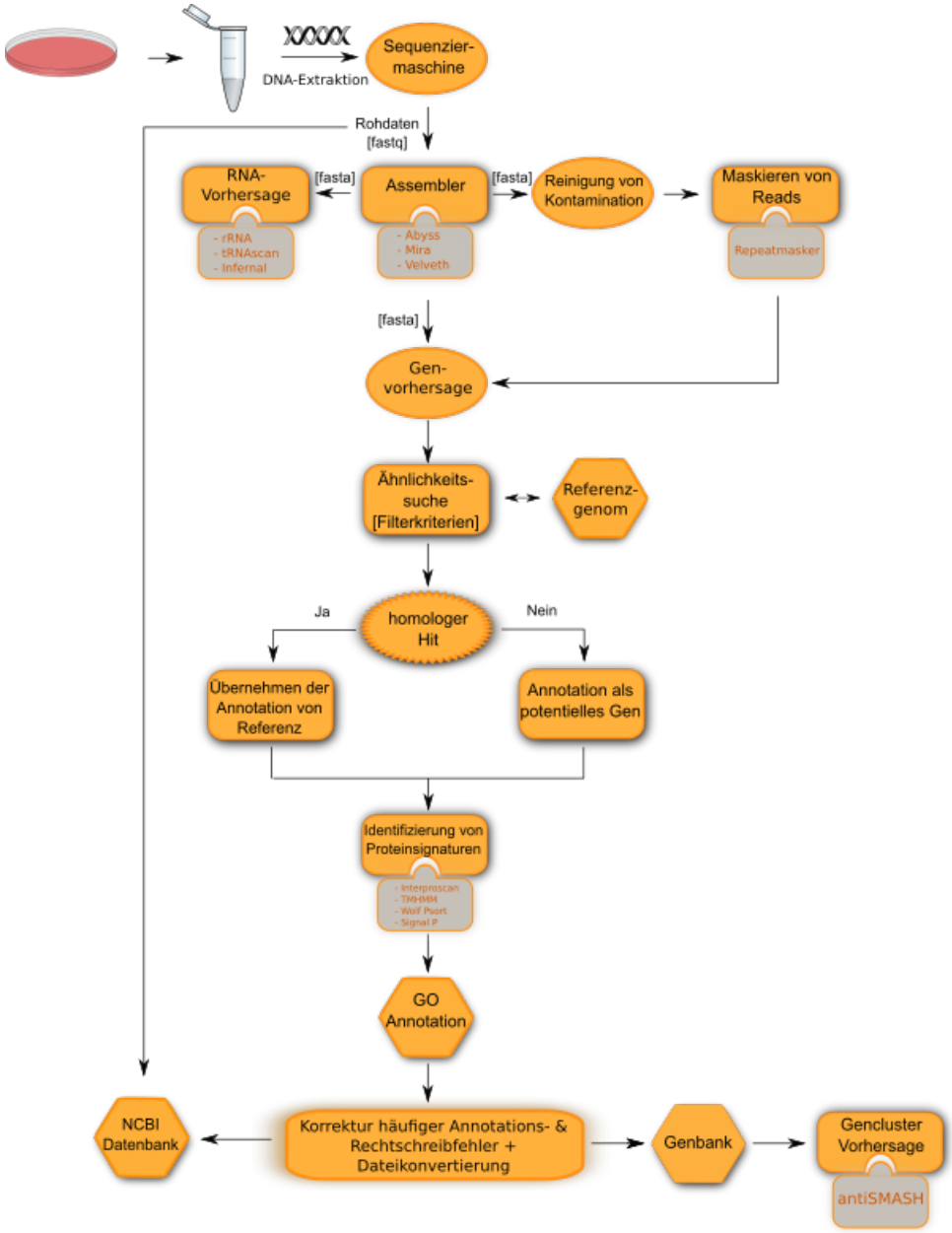

Abbildung 4: Flussdiagramm der Genomanalyse-Pipeline - Die Resultate der Sequenzierung werden entweder von der Sequenzierfirma oder als Teil der Genomanalyse-Pipeline assembliert. Die Assemblierung kann dann optional von Kontaminationen bereinigt und maskiert werden. Im Anschluss erfolgt eine Genvorhersage und eine Ähnlichkeitssuche. Die Suche nach homologen Sequenzen kann entweder gegen ein Referenzgenom erfolgen oder gegen andere Sequenzdatenbanken. Homologe Hits werden automatisch bewertet und als Annotation übernommen. Anschließend können Proteinsignaturen identifiziert werden oder eine Annotation mithilfe der Gene Ontology erfolgen. Am Ende der Pipeline können alle Annotation auf häufige Annotations- & Rechtschreibfehler untersucht und in verschiedene Sequenzformate überführt werden. Eine Genclustervorhersage ist mit annotierten Genen im GenBank-Format möglich, ebenso ein Upload der Sequenz in eine NCBI-Datenbank. Alle Schritte sind optional und können, sofern die Ein- & und Ausgaben übereinstimmen, frei kombiniert werden.

in der Lage, rRNAs, tmRNAs und tRNAs vorherzusagen. Nicht-kodierende RNAs, cis-regulatorische Elemente oder selbst-spleißende RNAs können mithilfe von *Infernal* (33) und der *Rfam*-Datenbank (34, 35) identifiziert werden. Für die Genvorhersage wurde *Glimmer* (36) für prokaryotische und *Augustus* (37) für eukaryotische Organismen integriert. Die Genvorhersage basiert meist auf einem gelernten Modell von bekannten Genstrukturen, die möglichst nahe verwandt mit denen des zu untersuchenden Organismus sein sollten. *Augustus* bietet hierfür eine Reihe von vorgefertigten Modellen an. In Galaxy kann der Benutzer ein Modell auswählen, welches dann die Grundlage der Genvorhersage bietet. *Glimmer* ist ebenso abhängig von einem trainierten Genmodell, um eine gute Genvorhersage zu gewährleisten, bietet jedoch keine vorberechneten Modelle an. Daher wurde die Genomanalyse-Pipeline dahingehend erweitert, *Glimmer*-Genmodelle bei Bedarf selbst erstellen zu können (siehe Abbildung **??**). Der Wissenschaftler muss lediglich eine Liste mit bereits bekannten Genen angeben, z.B. von einem verwandten bereits annotierten Organismus, auf deren Grundlage dann ein Modell für potentielle neue Gene erstellt wird. In Bezug auf prokaryotische Organismen, konzentrierte sich die Forschung des Instituts für Pharmazeutische Wissenschaften der Universität Freiburg hauptsächlich auf die Erforschung und Charakterisierung einzelner Arten der Gattung *Streptomyces*. Im Rahmen dieser Arbeit wurde speziell für Streptomyceten ein Genmodell aus den sehr gut untersuchten Modellorganismen *Streptomyces coelicolor* und *Streptomyces avermitilis* erstellt und in Galaxy jedem Wissenschaftler zur Verfügung gestellt. Dies ermöglicht unter anderem eine standardisierte und vergleichbare Genvorhersage aller Streptomyceten. Durch das Erkennen und Maskieren von langen repetitiven Sequenzen kann die Genvorhersage zum Teil verbessert werden. Zu diesem Zweck wurde das Programm *Repeatmasker* (38) integriert.

1.1.1.3 Annotationstransfer zwischen ähnlichen Sequenzen

Wurden die potentiellen Gene identifiziert, stehen eine Vielzahl von Programmen zur näheren Charakterisierung in der Genomanalyse-Pipeline zur Verfügung. Für Sequenz-Ähnlichkeitssuchen wurden die NCBI BLAST+ Tools in Galaxy integriert (32). Mit diesen ist es möglich, eine Query-Sequenz gegen die großen Sequenzdatenbanken, wie z.B.

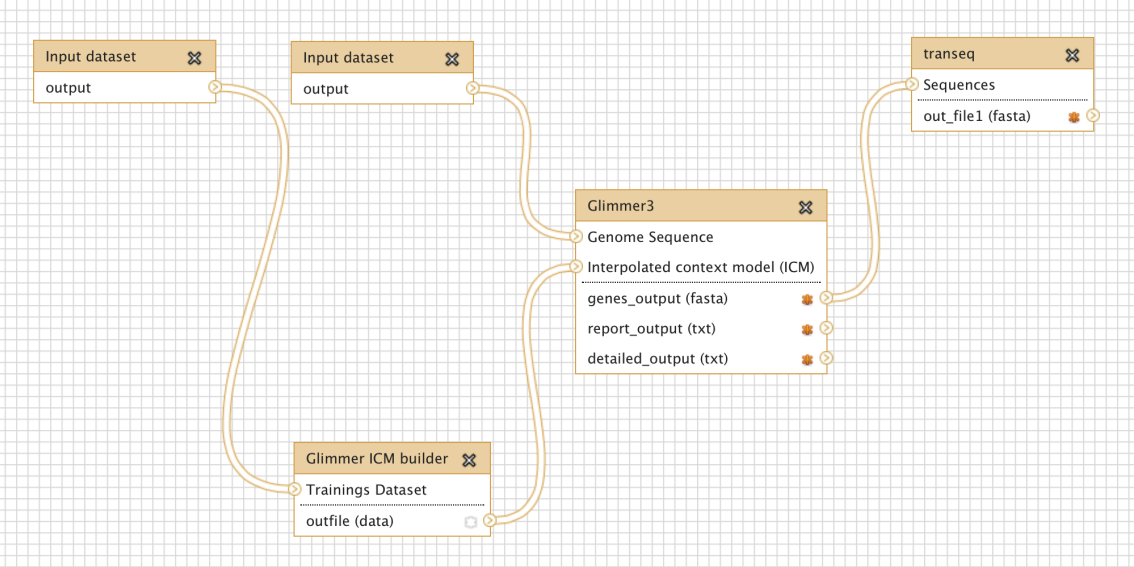

Abbildung 5: *Glimmer*-Genomannotation für Prokaryoten - *Glimmer* kann dazu benutzt werden prokaryotische Gene vorherzusagen. Dazu kann ein Genmodell erstellt werden, welches sich von bereits bekannten Genstrukturen ableitet. Das Modell und das zu annotierende Genom dienen als Eingabe für *Glimmer*, welches eine FASTA-Datei mit allen vorhergesagten Genen produziert. Im letzten Schritt werden diese mit dem *transeq* Programm in Proteinsequenzen übersetzt (31).

Refseq[2], NR/NT[1] oder UniProt[2], zu suchen und aufgrund der gefundenen ähnlichen Sequenzen Rückschlüsse auf die Funktion der Query-Sequenz zu ziehen. Dieser Rückschluss ist jedoch nicht trivial und mit etlichen Problemen, wie der Paralogie zwischen Genen und dem *Moonlighting problem*, gekennzeichnet. Es sei an dieser Stelle auf die Veröffentlichung von Punta und Ofran verwiesen (39), die sich mit dem Annotationstransfer zwischen homologen Sequenzen näher befasst haben und zeigen, dass Techniken zur Vermeidung dieser Probleme, wie die reziproke BLAST-Suche, sehr leicht in der Genomanalyse-Pipeline umsetzbar sind. Auch die Arbeit von Moreno-Hagelsieb und Latimer, die sich mit den Parametern einer BLAST-Suche zur Identifizierung von orthologen Gene beschäftigt (40), ist einfach in Galaxy abbildbar.

[2]`http://www.ncbi.nlm.nih.gov/refseq`
[1]`ftp://ftp.ncbi.nih.gov/blast/db`
[2]`http://www.uniprot.org`

1.1.1.4 Identifikation von Sequenzsignaturen

Ähnlichkeitssuchen sind aber nur dann sinnvoll, wenn die Grundannahme, dass eine ähnliche Sequenz auch eine ähnliche Funktion des Proteins zur Folge hat, auch stimmt. Viele funktionelle und evolutionär relevante Beziehungen von Proteinen sind aber nur durch einen Vergleich der dreidimensionalen Struktur zu erkennen und lassen sich nicht auf Sequenzebene finden (41, 42). Liegen diese 3D-Strukturen nicht vor oder soll die funktionelle Annotation im größeren Maßstab durchgeführt werden, können die Sequenzen mit zuvor identifizierten konservierten Motiven verglichen werden. Das BLAST+ Programm *psiblast/rpsblast* ermöglicht solche Suchen und damit auch die Identifizierung von entfernt verwandten Proteinen. Bekommt der Wissenschaftler mit den BLAST+ Programmen keinerlei zufriedenstellendes Ergebnis, besteht die Möglichkeit, funktionell wichtige Domänen bzw. Signaturen in der Sequenz zu identifizieren. Zu diesem Zweck wurde *InterProScan* (43) in die Genomanalyse-Pipeline integriert. *InterProScan* ist in der Lage, Sequenzen zu Proteinfamilien zuzuordnen und Domänen oder andere funktionell wichtige Positionen in der Sequenz zu identifizieren. *InterProScan* sucht in einer Vielzahl unterschiedlicher Datenbanken mit unterschiedlichen Schwerpunkten[3].

1.1.1.5 Genclustervorhersage

Gencluster sind Ansammlungen von Genen in unmittelbarer Nachbarschaft, deren Genprodukte für die Synthese vieler Sekundärmetabolite, wie Antibiotika oder anderen pharmazeutischen Stoffen, essentiell sind. In der Genomanalyse-Pipeline können Gencluster mithilfe von antiSMASH identifiziert werden (44). Hierbei kann der Benutzer zwischen 24 verschiedenen Genclustertypen wählen und bekommt eine interaktive Ausgabe (siehe Abbildung **??**) zum Inspizieren der identifizierten Cluster.

1.1.1.6 Veröffentlichung der Sequenzdaten in frei zugänglichen Datenbanken

Das NCBI ermöglicht es Wissenschaftlern, sequenzierte Genome in eine öffentliche Datenbank zu integrieren und damit für alle zugänglich zu machen. Dies können sowohl die Rohdaten eines Sequenzierungsprojektes sein als auch die bereits assemblierten und annotierten Sequenzen. Die Rohdaten gelangen unmodifiziert in das Sequenz Read Archive (SRA) des NCBI, die annotierten Sequenzen hingegen werden prozessiert und in

[3] `http://bit.ly/1UkD6OZ`

verschiedenen Datenbanken, wie der GenBank (45) oder der UniProtKB/TrEMBL (46), hinterlegt. Für die Einreichung der annotierten Sequenzen müssen die Daten in das NCBI *feature table*-Format überführt werden, aus dem in einem zusätzlichen Schritt eine Datei im *Sequin*-Format erzeugt wird. Diese kann zusammen mit anderen Metadaten in das Reviewverfahren des NCBI eingebracht werden.

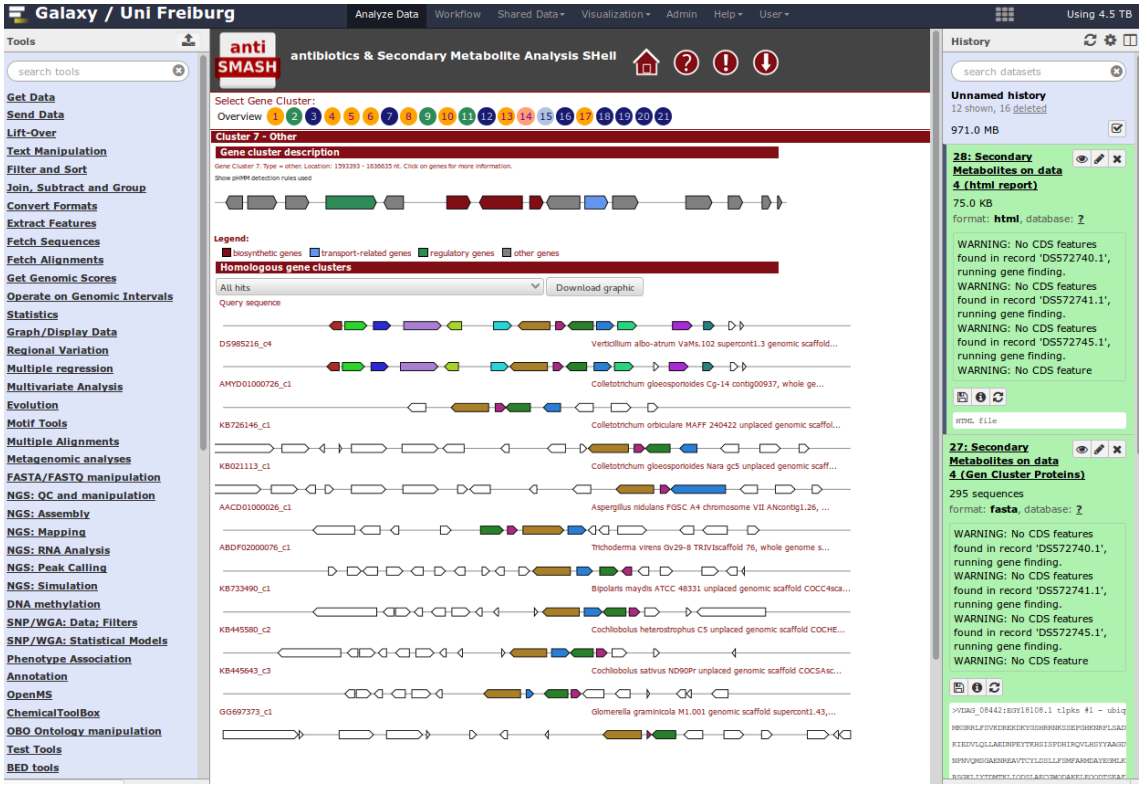

Abbildung 6: Genclustervorhersage basierend auf antiSMASH - Gencluster können in der Genomanalyse-Pipeline mihilfe von antiSMASH vorhergesagt werden. Der Benutzer bekommt eine interaktive Übersichtsseite als Ausgabe, sowie Datensätze die sich zur weiteren Verarbeitung in der Genomanalyse-Pipeline eignen. In diesem Beispiel wurden 21 Gencluster identifiziert. Zu sehen ist eine Detailansicht von Gencluster 7 mit ähnlichen Genclustern in anderen Organismen.

Die Genomanalyse-Pipeline beinhaltet alle nötigen Programme und Konverter, um ausgehend von Sequenzdaten alle benötigten Dateien für einen NCBI-Upload zur Verfügung zu stellen. In Abbildung **??** ist der Arbeitsablauf skizziert, der für die annotierten Organismen (Kapitel **??** - **??**) verwendet wurde.

Zuerst werden etwaige Kontaminationen aus den Sequenzen entfernt (siehe **??**). Anschließend werden mit dem Programm *Split Scaffolds* alle Sequenzen nach längeren nicht eindeutig identifizierten Nucleotiden durchsucht. Diese werden meist durch multiple *N* in der Sequenz gekennzeichnet und können durch ein uneindeutiges Assemblieren der Reads entstehen oder durch das Zusammensetzen von Contigs zu Scaffolds. Contigs entstehen durch das Assemblieren von überlappenden Reads. Scaffolds werden durch die lineare Anordnung von Contigs erstellt, bei denen die Reihenfolge der Contigs zwar bekannt ist,

aber nicht zwangsläufig der Abstand oder die komplette Sequenz zwischen zwei Contigs. Das NCBI erlaubt solche undefinierten Sequenzabschnitte (*N-runs*) nicht, daher wird bei jedem $len(N) > 2$ die Sequenz geteilt. Die erste Sequenz am Ende und die zweite Sequenz am Anfang werden von nicht definierten Basen (N) bereinigt. Die Sequenzreihenfolge, die damit verloren geht, wird in einer zusätzlichen Datei gespeichert und kann an das NCBI als Metadaten übermittelt werden. Nach dem Bereinigen von *N-runs* entfernt das Programm *Split Scaffolds* auch noch alle Sequenzen, die eine kürzere Sequenzlänge als 200 Nucleotide aufweisen. Diese Mindestlänge von Sequenzen ist ein weiteres Kriterium des NCBI.

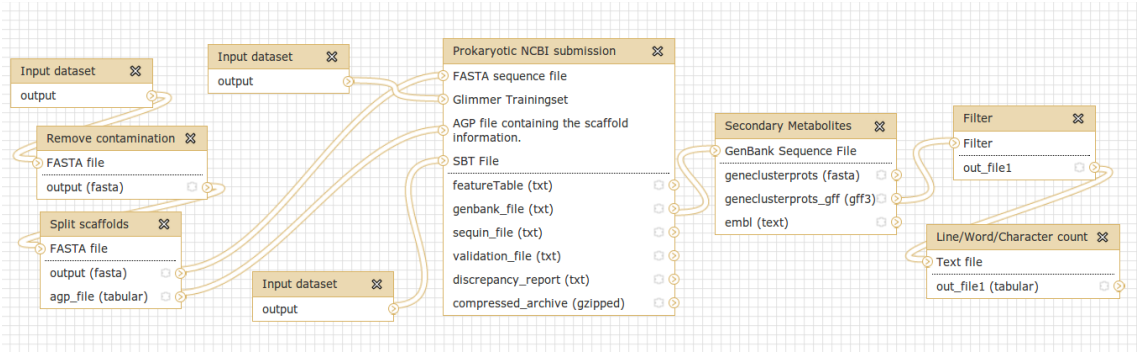

Abbildung 7: Genomannotations-Workflow mit Genclustervorhersage - Abgebildet ist der Workflow, der zur prokaryotischen Genomannotation verwendet wurde. Ausgehend von drei Eingabedateien (genomische Sequenz, Trainingsdatensatz für die *Glimmer*-Genvorhersage, Autoren für die Einreichung bei dem NCBI) wird zuerst die Sequenz von Kontaminationen befreit und in Scaffolds gegliedert. Die anschließende Genomannotation generiert eine Reihe von Ausgabedateien, die für die NCBI-Einreichung wichtig sind. Die generierte GenBank-Datei dient darauffolgend als Eingabedatei der Genclustervorhersage mit anschließender Bestimmung der Anzahl detektierter Cluster.

Im darauf folgenden Schritt wird jede einzelne der bereinigten und gefilterten Sequenzen prozessiert. Ist das Genom eukaryotisch, wird *Augustus* für die Genvorhersage verwendet, ist das Genom prokaryotisch, wird *Glimmer* herangezogen. Nach der Genvorhersage wird jedes Gen gegen eine zuvor vom Benutzer ausgewählte Sequenzdatenbank gesucht. Hier wird für eine NCBI-Submission die Ähnlichkeitssuche gegen die SwissProt-Datenbank empfohlen, da diese die manuell annotierte und geprüfte Version der UniProtKB ist. Wird zu einem Gen ein entsprechendes Homolog gefunden, wird die Annotation des homologen Gens übernommen. Bleibt die Ähnlichkeitssuche ohne Erfolg, wird das Gen als potentiell neues Gen eingestuft und dementsprechend annotiert.

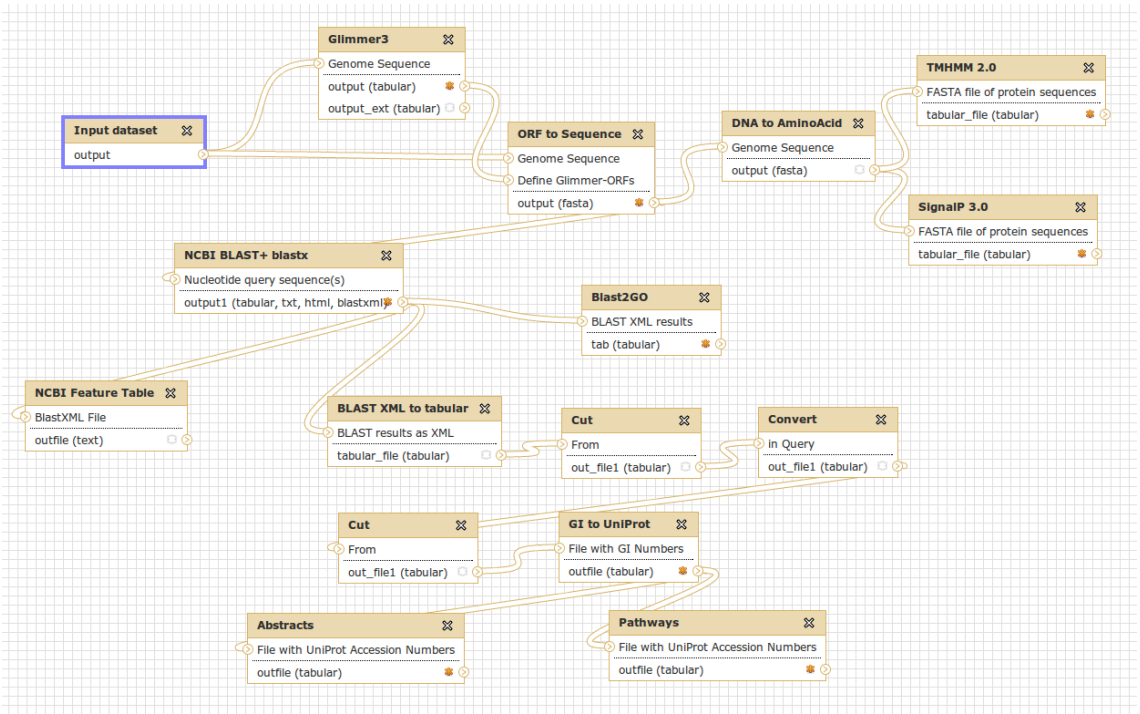

Abbildung 8: Genomanalyse-Pipeline als Galaxy-Workflow - Ausgehend von einer Eingabedatei (blau markiert) werden Gene mit *Glimmer* vorhergesagt. Die Genprodukte werden zum einen auf Signalpeptide als auch auf Transmembranhelices hin untersucht. Eine Ähnlichkeitssuche mittels BLAST+ sucht nach homologen Sequenzen in anderen Organismen. Diese homologen Hits dienen zum einen der weiteren Annotation durch Gene Ontology-Terme, aber auch zum Erstellen einer *NCBI Feature Table*, die für den NCBI-Upload notwendig ist. Des Weiteren können die Identifikationsnummern der BLAST+ Suche auch dazu verwendet werden, nach Stoffwechselwegen oder weiterführenden Literaturreferenzen zu suchen.

Neben den bereits erwähnten Arbeiten von *Punta et al.* und *Moreno-Hagelsieb et al.* zur *in silico* Funktionsannotation von Genen und Proteinen wurden die Probleme der automatischen Genomannotation von Emily J. Richardson und Mick Watson ausführlich diskutiert (39, 40, 47). Um dessen gerecht zu werden, verfolgt die Genomanalyse-Pipeline zwei Strategien. Einerseits wird versucht, so viele Annotationen wie möglich in besonders hoher Qualität zuzuordnen, andererseits werden dem Benutzer sämtliche Optionen, Zwischenschritte und Zwischenergebnisse transparent präsentiert. So bieten die Programme *Prokaryotic-* und *Eukaryotic NCBI submission* bis zu sechs verschiedene Ergebnis-Dateien an, die unter anderem auch Dateien enthalten, in denen Unstimmigkeiten in den Annotationen aufgelistet werden. Diese Funktion stammt aus dem vom NCBI entwickelten Programm *tbl2asn*, welches die Qualitätskontrolle übernimmt und vor häufigen Rechtschreib-

fehlern oder Identifikationsnummern warnt, die fälschlicherweise übernommen wurden. Soweit bekannt, wird versucht diese Fehler automatisch zu korrigieren, dies ist jedoch nicht immer möglich und im uneindeutigen Fall ist eine manuelle Änderung der Annotation des Benutzers erforderlich. Mit der gleichen Intention wird auch versucht, den Benutzer zu einer strengen Auswahl der Ähnlichkeitssuchparameter zu bewegen. Die Standardeinstellungen von 50 % Sequenzabdeckung und 50 % Sequenzidentität sind sehr restriktiv gesetzt und gewährleisten eine bessere Annotationsqualität zugunsten der Quantität aller annotierten Gene. Es sei hier aber erwähnt, dass alle Parameter im Benutzerinterface verändert werden können und dass es Anwendungsfälle gibt, bei denen nicht-restriktive Einstellungen erwünscht sind.

In Abbildung **??** ist ein Arbeitsablauf skizziert, der eine Vielzahl an Programmen beinhaltet und mit einer beliebigen genomischen Sequenz als Eingabe ausgeführt werden kann. Die gegebene Flexibilität und der Zugang zu etlichen Datenbanken ermöglichen den Genomannotationsmodulen in Galaxy auch die Beantwortung von komplexeren Fragestellungen. Gene und Proteine können ihren Signalwegen zugeordnet oder in Ontologien wie die Gene Ontology gruppiert werden. Assoziationen mit PubMed-Artikeln und mit potentiell interagierenden Kleinmolekülen wie in Abschnitt 2.2 beschrieben, erlauben eine breite Palette an Anwendungen nach der primären Annotation.

Zur Visualisierung der funktionell annotierten Genome wurde der Genom-Browser *GBrowse* in Galaxy integriert (Abbildung **??**). *Trackster* (48), der interne Galaxy Genom-Browser oder IGV (49) können ebenso verwendet werden. Zur weiteren Visualisierung wurde *Mauve* (50) zur Berechnung multipler genomischer Alignments sowie ein Programm zur Erstellung von Genplots integriert.

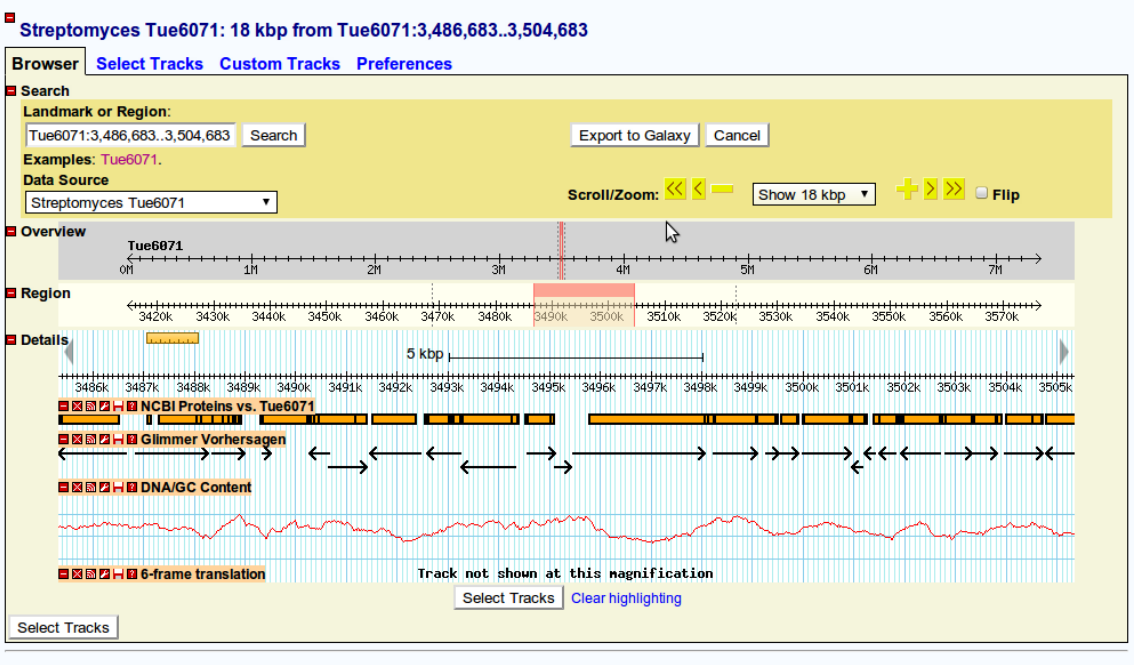

Abbildung 9: Genom-Browser-Integration in Galaxy - Die Genomanalyse-Pipeline kann mit einer Reihe von Genom-Visualisierungen erweitert werden. Zum Beispiel mit *GBrowse* zum anschaulichen Inspizieren der automatisch erstellten Annotationen. Abgebildet ist das *Streptomyces sp.* Tü6071-Genom mit vorhergesagten Genen (schwarze Pfeile), die Protein-Annotation mittels NCBI BLAST+ (orange Kästchen) und die Verteilung des GC-Gehalts (rote Linie).

1.1.2 Ergebnisse

Bakterien aus der Gattung der *Streptomyces* sind sehr wichtige Naturstoffproduzenten. Ihre Metabolite haben ein breites therapeutisches Spektrum und weisen unter anderem antibiotische, antitumorale und immunosuppressive Eigenschaften auf. Von den bekannten, natürlich vorkommenden Antibiotika werden ca. zwei Drittel von Streptomyceten produziert (51).

Die Identifizierung und Charakterisierung von Genclustern des Sekundärstoffwechsels ist grundlegend für das Verständnis der speziellen Biosynthese von Naturstoffen eines Organismus. Dies bildet die Grundlage für gezielte Veränderungen des Genoms und des Stoffwechsels, mit dem Ziel, die Metabolitenproduktion im Organismus zu beeinflussen.

Sechs Organismen wurden als Teil der vorliegenden Arbeit näher charakterisiert und ihre Gencluster identifiziert. Darunter sind nicht nur Prokaryoten aus der Gattung der

Streptomyces, sondern auch der eukaryotische Pilz *Glarea lozoyensis*, welches ein Beleg für die hohe Flexibilität der Genomannotations-Pipeline ist. Des Weiteren wurden verschiedene Workflows für das Institut für Pharmazeutische Wissenschaften der Universität Freiburg entwickelt, von denen einer exemplarisch in Abbildung **??** dargestellt ist. Der abgebildete Workflow wurde von Dr. Andreas Präg (Arbeitskreis Prof. Michael Müller, Institut für Pharmazeutische Wissenschaften, Universität Freiburg) angefragt und bei der Aufklärung der regio- und stereoselektiven intermolekularen oxidativen Phenolkopplung verwendet (52).

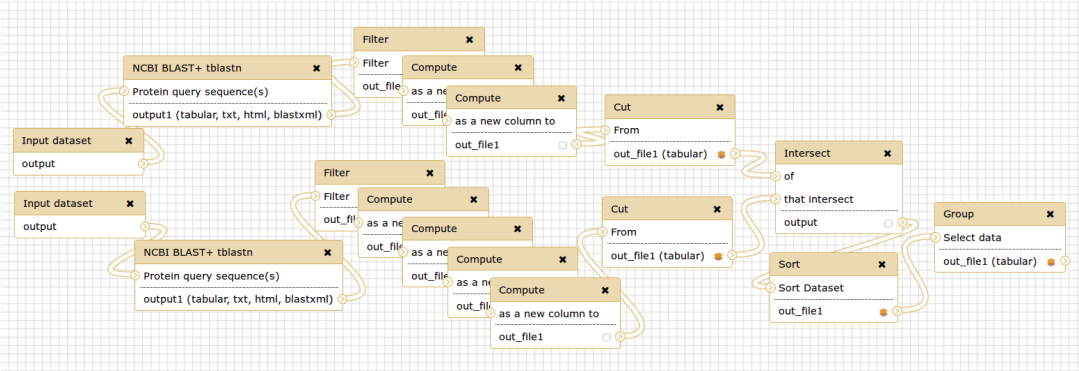

Abbildung 10: Auffinden zweier benachbarter Gene - Zwei Proteinsequenzen dienen als Eingabe in diesen Workflow. Sie werden gegen eine Sequenzdatenbank (voreingestellt ist die *NCBI-WGS*-Datenbank) gesucht und das Ergebnis anschließend in verschiedenen Schritten bearbeitet. Die genomischen Koordinaten eines Hits werden von einem der zwei Eingaben um 10.000 bp in beide Richtungen erweitert. Damit vergrößert sich die homologe Region von einem der Eingabeproteine. Beide Äste des Workflows werden in ein BED-Format überführt, welches drei Spalten hat (Sequenzidentifikator, Start-Koordinate und Stop-Koordinate). *Intersection* zweier BED-Dateien ist definiert als die Überlappung zweier genomischer Regionen, welches in diesem Anwendungsfall bedeutet, dass die zwei Eingabeproteine auf einer zusammenhängenden Sequenz gefunden wurden und nicht weiter als 10.000 bp voneinander entfernt liegen. Die Ausgabe des Workflows sind die sortierten und gruppierten Sequenz-IDs, die Aufschluss darüber geben, wie oft die zwei initialen Proteinsequenzen in unterschiedlichen Organismen in unmittelbarer Nachbarschaft vorliegen. (32)

Im Folgenden ist eine Auflistung der sechs Organismen gegeben, die als Teil dieser Arbeit für das Institut für Pharmazeutische Wissenschaften mithilfe der Genomannotations-Pipeline annotiert wurden.

1.1.2.1 *Streptomyces sp.* Tü6071

Teile des folgenden Abschnittes wurden veröffentlicht.

Anika Erxleben, J. Wunsch-Palasis, B. A. Grüning, M. Luzhetska, A. Bechthold, und S. Gunther. Genome Sequence of Streptomyces sp. Strain Tu6071. *Journal of Bacteriology*, 193(16):4278–4279, June 2011. ISSN 0021-9193. doi: 10.1128/JB.00377-11

Das annotierte Genom wurde unter der Accession-Nummer AFHJ00000000[4] in der DDBJ/EMBL/GenBank hinterlegt.

Streptomyces sp. Tü6071 ist ein Bakterium mit einem sehr aktiven Isoprenoid-Stoffwechsel und produziert das industriell wichtige Terpen Phenalinolacton, welches eine antibakterielle Wirkung gegen Gram-postive Bakterien hat.

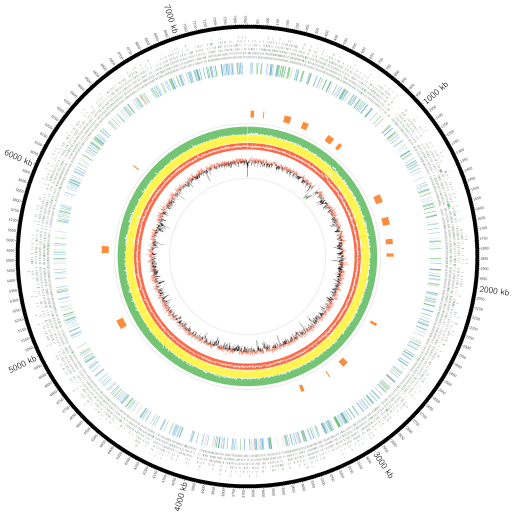

Abbildung 11: Zirkulärer Genomplot von *Streptomyces sp.* Tü6071 - Von außen nach innen; Kreis 1: vorhergesagten Gene; Kreis 2: tRNA-kodierende Gene (rot), Membranprotein kodierende Gene (blau), Transport-assoziierte Gene (grün); Kreis 3: mittels antiSMASH vorhergesagte sekundäre Biosynthesegencluster (orange); Kreis 4-7: Nucleotidverteilung: G (grün), C (gelb), A (rot-1) T (rot-2); Kreis 8: GC-Gehalt.

Das Genom besteht aus einem linearen Chromosom mit ca. 7.359.000 Basenpaaren und einem GC-Gehalt von 73,1 %, sowie einem linearem Plasmid (ca. 147.347 bp, 70,9 % GC-Gehalt). Die Analyse des Chromosoms identifizierte 6466 potentiell Protein-kodierende Gene, von denen 4887 eine putative Funktion zugeordnet werden konnte. Auf dem Plasmid konnten von 176 vorhergesagten offenen Leserahmen 73 funktionell annotiert werden. Des Weiteren wurden 6 rRNA Operons und 74 tRNAs auf dem linearen Chromosom identifiziert. Das Biosynthese-Gencluster von Phenalinolacton (54) wurde auf dem Chromosom an Position 1,69-1,73 Mb gefunden. Neben diesem Gencluster konnten 16 weitere Naturstoff-Gencluster identifiziert werden. Darunter Gencluster

für fünf Terpene, fünf NRPS, zwei PKS-NRPS-Hybride, ein PKS I, ein Siderophor, ein Gamma-Butyrolacton, ein Ectoin, sowie eines für Melanin.

1.1.2.2 *Streptomyces viridochromogenes* Tü57

Teile des folgenden Abschnittes wurden veröffentlicht.

Björn A Grüning, Anika Erxleben, Anna Hähnlein, und Stefan Günther. Draft Genome Sequence of Streptomyces viridochromogenes Strain Tu57, Producer of Avilamycin. *Genome announcements*, 1(3):e00384–13, January 2013. ISSN 2169-8287. doi: 10.1128/genomeA.00384-13

Das annotierte Genom wurde unter der Accession-Nummer AMLP01000000[1] in der DDBJ/EMBL/GenBank hinterlegt.

Streptomyces viridochromogenes Tü57 produziert unter bestimmten Bedingungen Avilamycin A, ein Antibiotikum gegen Gram-positive Bakterien. Avilamycin ist ein antibiotisch wirksames Oligosaccharid der Orthosomycin Gruppe (56). Vertreter dieser Gruppe zeigen eine sehr hohe antibakterielle Aktivität gegen Glycopeptid-resistente Enterokokken, Penicillin-resistente Streptokokken und Methicilin-resistente Staphylokokken (57, 58, 59). Avilamycin A besteht aus einer Heptasaccharid-Seitenkette und einer aus dem Polyketitsyntheseweg abgeleiteten Dichloroisoeverninsäure als Aglycon (60). Das Genom von *S. viridochromogenes Tü57* hat eine Größe von 9,7 Mbp mit einem durchschnittlichen GC-Gehalt von

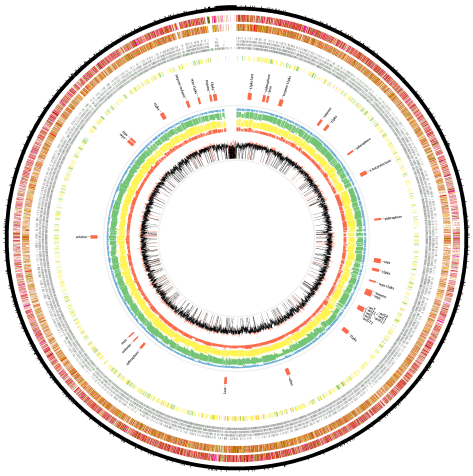

Abbildung 12: Zirkulärer Genomplot von *Streptomyces viridochromogenes* Tü57 - Von außen nach innen; Kreis 1-3: vorhergesagten Gene (verschiedene Darstellung); Kreis 4: tRNA-kodierende Gene (rot), Membranprotein-kodierende Gene (blau), Transport-assoziierte Gene (grün); Kreis 5: mittels antiSMASH vorhergesagte sekundäre Biosynthesegencluster (orange); Kreis 6-9: Nucleotidverteilung: G (grün), C (gelb), A (blau) T (rot); Kreis 10: GC-Gehalt.

71 %. Es wurden 33 Gencluster für Sekundärmetabolite vorhergesagt, darunter zwei PKSI, ein PKSII, zwei PKSIII, neun NRPS, zwei NRPS-PKSI-Hybride, drei Terpene und 14 andere Gencluster. Weiterhin wurden 68 tRNAs und eine tmRNA in dem Genom identifiziert. Einige Gene des an der Biosynthese von Avilamycin A beteiligten Sekundärmetabolitengenclusters wurden bereits charakterisiert (60, 61, 62) und konnten im Rahmen der Genomanalyse lokalisiert werden.

1.1.2.3 *Streptomyces aurantiacus* JA 4570

Die Ergebnisse dieser Genomsequenzierung wurden veröffentlicht und die erhaltenen Sequenzdaten bei DDBJ/EMBL/GenBank unter der Accession-Nummer AOPZ00000000[2] und der SRA Nummer SRP018101[3] hinterlegt.

Streptomyces aurantiacus JA 4570 ist der Produzent von Setomimycin, einem Antibiotikum, das gegen Gram-positive Bakterien wie z.B. Mycobakterien aktiv ist, sowie gegen solide Sarcoma-180 Tumoren in Mäusen (63). Das Genom von *S. aurantiacus* JA 4670 hat eine Größe von 8,76 Mbp und besitzt einen GC-Gehalt von 72 %. Es konnten 44 Biosynthesegencluster für Sekundärmetabolite detektiert werden.

1.1.2.4 *Streptomyces afghaniensis* NC 5228T

Die Ergebnisse dieser Genomsequenzierung wurden veröffentlicht und die erhaltenen Sequenzdaten bei DDBJ/EMBL/GenBank unter der Accession-Nummer AOPY00000000[1] und der SRA-Nummer SRP018099[1] hinterlegt.

Streptomyces afghaniensis NC 5228T produziert das antivirale Antibiotikum Julimycin B (64), welches unter anderem eine antitumorale Wirkung hat (65). Das Genom von *S. afghaniensis* NC 5228T hat eine Größe von 9,84 Mb und besitzt einen GC-Gehalt von 71 %. Es konnten 46 Biosynthesegencluster für Sekundärmetabolite detektiert werden.

1.1.2.5 *Streptomyces spectabilis* DSM 40779

Die Sequenzendaten sind bislang nicht veröffentlicht, können aber auf Anfrage von Jun.-Prof. Dr. Stefan Günther, Pharmazeutische Bioinformatik, Institut für Pharmazeutische Wissenschaften, Universität Freiburg, zur Verfügung gestellt werden.

Streptomyces spectabilis DSM 40779 ist ein Produzent der Tetrahydroanthracen-Antibiotika Stectomycin A1, A2 und B1 (66). Die Genomgröße von *S. spectabilis* DSM 40779 beträgt 10,6 Mbp mit einem GC-Gehalt von 72 %. Es wurden 53 Biosynthesegencluster für Sekundärmetabolite detektiert.

1.1.2.6 *Glarea lozoyensis* ATCC 74030

Teile des folgenden Abschnittes wurden in peer-reviewed Journals veröffentlicht.
Loubna Youssar, Björn Andreas Grüning, Anika Erxleben, Stefan Günther, und Wolfgang Hüttel. Genome sequence of the fungus Glarea lozoyensis: the first genome sequence of a species from the Helotiaceae family. *Eukaryotic cell*, 11(2):250, February 2012. ISSN 1535-9786. doi: 10.1128/EC.05302-11

Das annotierte Genom wurde unter der Accession-Nummer AMLP01000000[1] in der DDBJ/EMBL/GenBank hinterlegt.

Der anamorphe Pilz *Glarea lozoyensis* ATCC 74030 ist ein Überproduzent von Pneumocandin B0, welches chemisch in Cancidas konvertiert werden kann, einem potenten Antibiotikum gegen klinisch wichtige Pilz-Pathogene (68). Pneumocandine gehören zur Gruppe der Echinocandin-Antibiotika. Das Genom von *Glarea lozoyensis* ATCC 74030 hat eine Größe von ca. 38,6 Mbp mit 7904 Protein-kodierenden Genen. Auf dem Genom konnten drei PKS-NRPS-Hybrid- und sechs NRPS-Cluster vorhergesagt werden.

1.1.3 Diskussion

Es konnte in diesem Kapitel gezeigt werden, dass mithilfe der Genomanalyse-Pipeline, erfolgreich Gene und Gencluster in genomischen Sequenzen identifiziert werden können. Weiterhin war es möglich, zu vielen dieser Gene das Genprodukt näher zu charakterisieren und Rückschlüsse auf eine potentielle Funktion zu ziehen. Speziell auf die Pharmazie der Universität Freiburg zugeschnittene Workflows wurden erfolgreich zur Annotation von pharmazeutisch relevanten Organismen angewendet und führten zu erheblichem Erkenntnisgewinn mit daraus resultierender Veröffentlichung der Ergebnisse. Die hohe Flexibilität durch das Kombinieren der verschiedenen Tools, einhergehend mit der einfachen Zugänglichkeit und der hohen Skalierbarkeit grenzt die Genomanalyse-Pipeline von bisherigen Lösungen, wie Prokka (69), RAST (70) oder DIYA (71) ab. Das Galaxy-Framework

erleichtert zudem die Zusammenarbeit und den Austausch zwischen Arbeitsgruppen erheblich. Der direkte Erkenntnisaustausch wird somit gefördert, und hat unter anderem zu vergleichenden Genomanalysen und einem speziellen für die Pharmazie entwickelten Genomannotationsworkshop geführt. Ein entscheidender Vorteil des hier beschriebenen Ansatzes ist die einfache Integration von Ergebnissen in andere Wissenschaftsfelder. Die identifizierten Gene können zum Beispiel zur Erschließung des Proteoms in eine Galaxy-basierte Massenspektrometrie-Analyse einfließen oder bei der Aufklärung des Metaboloms helfen. Am Ende einer Analyse stehen verschiedene Visualisierungsoptionen zur Darstellung von komplexen Resultaten und Plots ganzer Genome, die ihre Nützlichkeit schon bewiesen haben und vielfach in Publikationen verwendet wurden.

Die in der Genomanalyse-Pipeline nicht annotierten Gene stellen insofern ein Problem dar, dass sie entweder bisher komplett unbeschrieben sind oder dass ihre funktionelle Beschreibung nicht in strukturierter Form in einer der uns zur Verfügung stehenden Datenbanken erhältlich ist. Letzteres ist Thema des nächsten Kapitels, in dem Primärliteratur automatisch aufgearbeitet wird und Informationen über Biomoleküle strukturiert und durchsuchbar in einer Datenbank hinterlegt werden.

1.2 Proteine in wissenschaftlichen Abhandlungen

Teile des folgenden Abschnittes wurden in einem peer-reviewed Journal veröffentlicht.

Christian Senger, Björn A Grüning, Anika Erxleben, Kersten Döring, Hitesh Patel, Stephan Flemming, Irmgard Merfort, und Stefan Günther. Mining and evaluation of molecular relationships in literature. *Bioinformatics (Oxford, England)*, 28(5):709–14, March 2012. ISSN 1367-4811. doi: 10.1093/bioinformatics/bts026

Prolific steht unter `http://prolific.pharmaceutical-bioinformatics.de` als Webservice zur Verfügung.

Im vorherigen Kapitel konnte gezeigt werden, dass ein Genom mithilfe der Genomannotations-Pipeline strukturell sowie funktionell annotiert werden kann. Aber nicht für alle Gene oder Proteine ist es möglich, über Ähnlichkeitssuchen (z.B. BLAST+, Diamond) oder Motivsuchen (z.B. InterProScan, Pfam) eine funktionelle Beschreibung zu erhalten. Gründe dafür können vielfältig sein. Zum Beispiel kann das zu suchende Protein mehrere nahe Verwandte mit unterschiedlichen Funktionen haben, in diesem Fall ist eine eindeutige Identifikation schwierig. Es kann aber auch vorkommen, dass keine Informationen in den Annotationsdatenbanken hinterlegt wurden.

In diesen Fällen kann der Wissenschaftler auf Primärliteratur zurückgreifen. Mit jährlich 500.000 neuen Publikationen auf PubMed[1] und ca. 20 Millionen insgesamt, ist dies jedoch sehr aufwendig und es besteht die Möglichkeit, Publikationen zu übersehen. Das Hauptproblem hierbei ist, dass Informationen im Text nicht in strukturierter Form vorliegen. Biomoleküle werden zum Beispiel mit etlichen verschiedenen Synonymen referenziert und selten mit Identifikationsnummern. Dies macht es schwer, Biomoleküle in der Primärliteratur zu finden.

Die *Medical Subject Headings* (MeSH[2]) (73) stellen hier einen interessanten Ansatz zur Lösung des Problems dar. Seit 1960 stellen die MeSH-Terme ein kontrolliertes Vokabular zur Sacherschließung einer Publikation zur Verfügung und machen eine Schlagwortsuche möglich. In der Praxis stellen sich MeSH-Suchen als vorteilhaft, doch leider nicht als Lösung des Problems heraus. MeSH-Terme werden vom NCBI in einem semiautomatischen Prozess vergeben und nicht alle Publikationen sind vollständig annotiert.

[1] `http://www.ncbi.nlm.nih.gov/pubmed`
[2] `http://www.nlm.nih.gov/mesh/`

Die Publikation von Erxleben *et al.* (53) enthält zum Beispiel nur die MeSH-Terme in Tabelle 1.1 und die Publikation von Grüning *et al.* (55) ist noch mit keinem Schlagwort versehen. Biomoleküle, die in diesen Abhandlungen erwähnt wurden, sind demnach nicht über eine MeSH-Suche zu finden. Das Ausmaß des Problems wird bewusst, wenn die Zahlen der Neuveröffentlichungen betrachtet werden, die sich mit der Gattung der *Streptomyces* beschäftigen. Über 100 Veröffentlichungen entstehen jeden Monat zu diesem Thema, mit neuen Informationen, die weder in Annotationsdatenbanken zu finden noch mit MeSH-Terme versehen sind.

Tabelle 1.1: MeSH-Terme der Publikation "Genome Sequence of *Streptomyces sp.* Tü6071" von Erxleben *et al.*. Stand Juni 2015.

MeSH-Term
Genome, Bacterial
Molecular Sequence Data
Streptomyces/classification
Streptomyces/genetics

In diesem Kapitel wird ein alternativer Ansatz beschrieben, der es dem Wissenschaftler ermöglicht, Proteine und Gene in wissenschaftlichen Abhandlungen zu identifizieren. Darüber hinaus werden die Kookkurrenzen zwischen gefundenen Proteinen und Genen mit potentiellen Metaboliten aufgeschlüsselt und dem Benutzer anschaulich präsentiert.

Kookkurrenzen ermöglichen es, potentielle Leitstrukturen für ein gegebenes Target, bzw. potentielle Nebenwirkungen von Wirkstoffen, zu erschließen. Damit ist der in dieser Arbeit entwickelte Prolific-Service (*protein-literature investigation for interacting compounds*) nicht nur für die Annotation von Genen interessant, sondern auch für die Wirkstoffentwicklung und Cheminformatik, die in Teil 2 der Arbeit beschrieben werden.

1.2.1 Daten und Methoden

1.2.1.1 PubMed-Parser

Teile des folgenden Abschnittes wurden in einem peer-reviewed Journal eingereicht.

Kersten Döring, Björn Grüning, Kiran K Telukunta, und Stefan Günther. PubMed2Go: A Framework for Developing Text Mining Applications. *BMC bioinformatics*, (submitted), 2015

PubMed2Go steht im Quellcode unter https://github.com/KerstenDoering/PubMed2Go zur Verfügung.

Eine der größten Datenbanken für wissenschaftliche Publikationen ist die von der *U.S. National Library of Medicine®(NLM)* Bibliografische Datenbank MEDLINE®. Der Inhalt der MEDLINE-Datenbank, exklusive Volltext-Publikationen, steht in einem XML-basierten Format zum Download zur Verfügung und hat eine Größe von über 110 GB. Ein speziell entwickelter Parser prozessiert alle XML-Entitäten und speichert diese in entsprechenden Tabellen eines objektrelationalen Datenbankmanagementsystems (ORDBMS). Das zugrundeliegende Datenbankschema wurde in Anlehnung an das MEDLINE-Schema der BioText-Initiative der Berkeley University[1] entworfen, optimiert sowie erweitert und in SQLAlchemy[2], eine Python-Bibliothek für objektrelationale Abbildungen (ORM), implementiert. Die Nutzung eines ORM hat den Vorteil, unabhängig vom verwendeten Datenbankmanagementsystem (DBMS) zu sein. Dem Benutzer steht es damit frei, ein DBMS nach seinen Anforderungen zu wählen. Aufgrund der großen Menge an MEDLINE-Daten wurde ein ereignisorientierter SAX-XML-Parser implementiert, der von Beginn an für paralleles Rechnen auf multiplen Prozessoren konzipiert wurde.

Nach der Extraktion aller Daten in die Datenbank wurde ein Xapian[3]-basierter Suchindex über die speziellen Felder “titles”, “abstracts”, “MeSH terms”, “keywords” und “chemical substances” erstellt. Xapian ist eine Volltext-Suchmaschine basierend auf einem probabilistische Modell, die sich sehr einfach in Webseiten integrieren lässt und sehr performant ist. Im Folgenden wurde der Volltextindex vor allem für das Erstellen der Datenbank benutzt, aber auch als Volltextsuche in die Webseite integriert.

[1] `http://bit.ly/1KQdQyv`
[2] `http://www.sqlalchemy.org`
[3] `http://xapian.org`

1.2.1.2 Datenbanken

Die drei Entitätsklassen Protein, Kleinstruktur und wissenschaftlicher Artikel wurden in einem objektrelationalen Datenbankmanagementsystem gespeichert.

Titel, Abstracts, Schlagwörter, etwaige Substanznamen und etliche weitere Metadaten wurden aus wissenschaftliche Publikationen der PubMed extrahiert, aufbereitet und indiziert gespeichert. Der Zugriff auf Volltext-Publikationen war aus rechtlichen Gründen nicht möglich[1]. Insgesamt wurden 20,6 Millionen Artikel prozessiert, davon beinhalteten 11,7 Millionen durchsuchbare Abstracts.

Auf Seiten der Kleinstrukturen wurde die PubChem (75), eine der größten frei verfügbaren chemischen Datenbanken, verwendet. Ihre Strukturen wurden hierarchisch und indiziert gespeichert. Hierarchisch bezieht sich auf die *Parents*-Klassifikation der PubChem. Ein *Parent* ist eine Struktur, die laut Definition der PubChem den wichtigen Teil einer Struktur ausmacht, wenn diese eine oder mehrere kovalente Komponenten besitzt[2]. Insgesamt wurden 28,3 Millionen *Parents* mit 35,4 Millionen Strukturen und ca. 606,3 Millionen Synonymen in Prolific gespeichert.

Etwa 2 Millionen Proteinsynonyme wurden aus der UniProtKB/Swiss-Prot (76) extrahiert und mit den zugehörigen Gene Ontology-Annotationen (GOA) (77, 78) sowie deren *Gene symbols* gespeichert. *Gene symbols*, genauso wie PubChem-*Parents*, dienten der Gruppierung von sehr ähnlichen Entitäten. Im Falle von Proteinen sind dies zum Beispiel Homologe aus unterschiedlichen Organismen.

Suche nach ähnlichen Proteinen Eine Suche in Prolific kann mit einer Sequenz oder einem Synonym gestartet werden. Im Falle einer Sequenz hat der Benutzer die Möglichkeit, zwischen einer Protein- und einer Nucleotidsequenz zu wählen. Im Falle einer Nucleotidsequenz wird diese in die korrespondierende Aminosäuresequenz übersetzt. Die Sequenzähnlichkeit wird mithilfe der *BLAST*-Programme (1, 2) berechnet. Als Datenbank dient die UniProtKB/Swiss-Prot (76).

Sollte mit einem Synonym gesucht werden, wird standardmäßig eine exakte Suche gegen die Prolific-Datenbank durchgeführt. Dies geschieht unter der Annahme, dass die meisten Benutzer mit einem UniProt-Identifikator die Suche starten. Ist die Textsuche

[1] `http://www.nlm.nih.gov/databases/license/license.html`
[2] `https://pubchem.ncbi.nlm.nih.gov/help.html`

nicht erfolgreich, kann eine Ähnlichkeitssuche mit beliebigen Suffixen oder Präfixen gestartet werden. Um häufige Tippfehler bei der Eingabe der Synonyme zu vermeiden, wurde die Bibliothek hunspell[1] mit einem auf Proteinnamen angepassten Wörterbuch, integriert.

Suche nach biologischen Entitäten in Artikeln Abstracts wurden aus der lokalen Datenbank extrahiert und automatisiert dem Whatizit-Webservice (79) zur Identifikation von Proteinsynonymen übergeben. Whatizit annotiert Proteine im Titel und im Abstract mit eindeutigen Protein-Identifikationsnummern. Die annotierten Texte wurden im Anschluss mit einer Protein-Stoppwortliste gefiltert und in einer Datenbank mit der dazugehörigen Artikel-Protein-Relation hinterlegt. Diese Liste besteht aus unerwünschten Wörten bzw. Synonymen der natürlichen englischen Sprache, sowie Wörter mit einer unnatürlich hohen Häufigkeit in Abstracts. Das Wort "And" zum Beispiel, welches für das Protein "Calmodulin-related protein 97A" steht, oder "ANOVA", eine Abkürzung für das "RNA-binding protein Nova-2", kommt in statistischen Analysen oft vor. Die Stoppwortliste wurde in einem zweistufigem Prozess generiert. Zuerst wurden alle gefundenen Proteinsynonyme der Häufigkeit ihrer Nennung in PubMed-Abstracts sortiert und auf ihre Korrektheit analysiert. Die restlichen Synonyme wurden nach den Regeln von Hettne *et al.* bearbeitet (80). Zum Beispiel wurden Synonyme entfernt, die nach Tokenisierung und Entfernen von Stoppwörtern nur noch arabische oder romanische Zahlen beinhalteten oder nur noch aus einem Buchstaben bestanden (*short token filter rule*).

Die Kleinstrukturen wurden mit Synonymen der PubChem gegen die indizierten Abstracts und MeSH-Terme gesucht. Die gefundenen Abstract-Kleinstruktur-Beziehungen wurden nach dem Filtern gegen eine Kleinstruktur-Stoppwortliste, analog zu der Protein-Stoppwortliste in einer relationalen Datenbank gespeichert. Aus Kleinstruktur-Artikel- und Protein-Artikel-Relationen wurden Protein-Kleinstruktur-Artikel-Relationen zusammengefasst und in vier verschiedene Klassen eingestuft:

[1] `http://hunspell.sourceforge.net`

1. Kookkurrenz von Proteinen und Kleinstruktur in einem Abstract
2. Kookkurrenz von Proteinen und Kleinstruktur in einem Satz
3. Kookkurrenz in einem Satz, einschließlich einem *"functional process"* oder *"molecular function"* annotiertem Begriff aus der *Gene ontology* (78)
4. Kookkurrenz in einem Satz, einschließlich einem Verb, welches Beziehungen zwischen Protein und Kleinstruktur ausdrückt. Diese Verben wurden aus ca. 2 Millionen Abstracts und deren Sätzen extrahiert, in denen jeweils Protein und Kleinstruktur zusammen Nennung fanden.

Die Korrektheit der gefundenen Kookkurrenzen wurde mit einem Testdatensatz aus der Datenbank SuperTarget evaluiert (81). Hierzu wurden zufällig 120 Kleinstruktur-Protein-Interaktionen aus der manuell kuratierten SuperTarget mit ihren dazugehörigen Publikationen extrahiert. Vier Biologen und Bioinformatiker annotierten diese Abstracts erneut, um *non-targets* und *non-drugs* in den Testdatensatz zu integrieren. Im Anschluss wurden alle Abstracts in der Prolific-Datenbank gesucht und ein *precision* und *recall* von 94 % respektive 72 % ermittelt (*F-Score*: 82 %).

1.2.2 Ergebnisse

Der Prolific-Webservice versteckt den komplexen Workflow und den Suchprozess vor dem Benutzer, lässt ihm aber die Möglichkeit, Suchparameter nach Belieben zu verändern und damit Einfluss auf die Suchergebnisse zu nehmen (Abbildung 13). Die Suche kann mit einem Proteinnamen, einem Identifikator (ID) oder einer Sequenz initiiert werden. Sollte der exakte Proteinname oder der Identifikator nicht in der Datenbank gefunden werden, so wird die Datenbank nach (lexikographisch) ähnlichen Namen durchsucht und dem Nutzer eine Liste mit potentiellen Proteinen präsentiert. Durch diese Funktion werden auch Schreibfehler in der Sucheingabe vorgebeugt. Sollten mehrere Treffer zu einem Suchterm gefunden werden, wird dem Benutzer eine Liste mit allen gefundenen Proteinen präsentiert. In beiden Fällen werden nur Proteine angezeigt, zu denen auch Sequenzinformationen in der Datenbank hinterlegt wurden. Im Falle eines exakten, eindeutigen Treffers wird die dazugehörige Sequenz in der Datenbank gesucht.

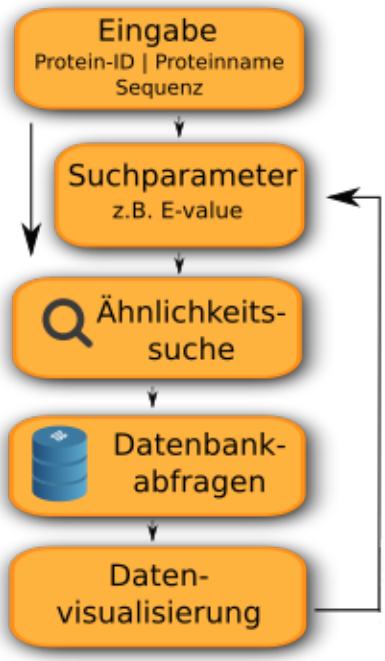

Abbildung 13: Prolific-Flussdiagramm - Der Prolific-Webservice benötigt ein Proteinnamen bzw. Identifikator oder eine Protein- bzw. Nucleotidsequenz als Eingabe. Anschließend werden sequenzähnliche Proteine in der Prolific-Datenbank gesucht und Artikel, sowie assoziierte Kleinstrukturen extrahiert. Das Triple aus Protein-Artikel-Kleinstruktur wird mit zahlreichen Zusatzinformationen visualisiert.

Unabhängig von der Benutzereingabe steht am Ende eine Proteinsequenz, mit der eine Ähnlichkeitssuche, basierend auf dem BLAST-Algorithmus (1, 2), gegen die komplette Prolific-Datenbank durchgeführt wird. Die erhaltenen ähnlichen Proteine werden im nachfolgenden Schritt in einer vorberechneten Datenbank gesucht und PubMed-Publikationen zugeordnet. Kleinstrukturen, die in den von Prolific identifizierten Artikeln erwähnt wurden, werden extrahiert. Hierbei wird auf die CIL-Datenbank (siehe Kapitel 2.2) zurückgegriffen. Die gesammelten Informationen werden gefiltert, gruppiert, visuell aufbereitet und dem Benutzer in einer anschaulichen Form präsentiert. Publikationszusammenfassungen werden mit farblich markierten Entitäten dargestellt und Hyperlinks zu weiterführenden Informationen angeboten. Sollten die Suchergebnisse zu vielzählig oder wenig aussagekräftig sein, besteht die Möglichkeit in verschiedenen Stellen des Workflows einzugreifen und Parameter, z.B. den Schwellenwert für die Ähnlichkeitssuche, zu ändern. Auch können Suchergebnisse auf Kookkurrenzen in einem Satz eingegrenzt werden, anstatt auf einen ganzen Abstract, wie es die Standardeinstellung ist. Weiterhin kann der Benutzer funk-

tionelle Gene Ontology-Terme oder Verben zur Beschreibung von Relationen zwischen Kleinstruktur und Protein angeben, die neben den eigentlichen Entitäten in einem Satz vorkommen müssen.

1.2.2.1 Beziehung zwischen Abstracts, Kleinstruktur- und Proteinsynonymen

In allen wissenschaftlichen Artikeln konnten ca. 8,8 Millionen Proteine und 12,5 Millionen Kleinstrukturen identifiziert werden. Betrachtet man diese zusammen, ergeben sich ca. 309,5 Millionen Relationen zwischen Proteinen und Kleinstrukturen in allen Publikationen. All diese Informationen sind über ein Web-Interface einsehbar und vor allem suchbar. Für einen automatisierten Zugriff auf die Daten wird eine auf SOAP (Simple Object Access Protocol) basierende API angeboten. Diese kann zum Beispiel dazu genutzt werden, um Prolific in das Galaxy-System zu integrieren.

1.2.2.2 Validierung der erhobenen Daten

Für die Validierung der gesammelten Daten wurden die in der DrugBank annotierten Drug-Target-Beziehungen herangezogen. Die Proteine (DrugBank-Targets) wurden in Prolific gesucht und die Ergebnisse mit den bekannten DrugBank-Drugs und deren annotierten Artikeln verglichen. Es konnte gezeigt werden, dass die Prolific-Datenbank 56 % aller Relationen der DrugBank enthält (siehe Tabelle 1.2). Zieht man in Betracht, dass DrugBank auch Biopharmaka wie Antikörper enthält, die nicht Teil der Prolific-Datenbank sind und das einige Proteine und Kleinstrukturen in keinen Artikeln gefunden werden konnten, so konnten 69 % aller DrugBank-Relationen identifiziert werden.

Tabelle 1.2: Identifizierte Wirkstoff-Target-Relationen der DrugBank in Prolific mit dem Target als Suchprotein (72).

Kategorie	#Gefunden (%)
gefundene Relationen	795 (56)
nicht gefundene Protein-Kleinstruktur Relation	357 (25)
Wirkstoff der Relation ist ein Biomolekül	157 (11)
Protein der Relation wurde in keinem Abstract gefunden	65 (5)
Kleinstruktur der Relation wurde in keinem Abstract gefunden	37 (3)
Alle Wirkstoff-Target-Relationen der DrugBank	1411 (100)

Von den 357 nicht identifizierten Relationen wurden stichprobenartig 50 einer qualitativen Untersuchung unterzogen. Dabei wurden 132 Volltexte untersucht. In 78 Abstracts wurden Proteinsynonyme gefunden, die abweichend von den UniProt-Synonymen in der Prolific-Datenbank waren. 48 Publikationen enthielten den gesuchten Proteinnamen nur im Volltext und Prolific enthält aus rechtlichen Gründen nur Abstract-Informationen. Daher wurden die Proteine nicht gefunden. Sechs von 50 Targets waren DNA oder RNA und sind daher ebenfalls nicht Teil der Datenbank.

Im Allgemeinen kann geschlussfolgert werden, dass die Treffergenauigkeit sehr von der Qualität der Synonyme abhängt. Von den 102 Proteinen und Kleinstrukturen, die in keinem Artikel identifiziert wurden, waren die meisten Synonyme entweder zu exakt, z.B. "gamma-aminobutyric acid receptor subunit rho-1" anstatt "GABA", oder sie enthielten nur Identifikationsnummern, die in den UniProt-Synonymen fehlten.

Wurde ein DrugBank-Target gefunden, so konnte gezeigt werden, dass die dazugehörige Kleinstruktur in 70 % aller Fälle unter den ersten 100 Treffern der Prolific-Ergebnisse zu finden ist. Von allen Kleinstrukturen, die eine Relation zu einem Protein aus der DrugBank-Target-Liste aufweisen, sind im Mittel 23 % unter den "FDA approved drugs" (Median 23 %, IQR: 17-27) und 32 % unter den "FDA approved and experimental drugs" (Median: 23 %, IQR: 27-40) der DrugBank.

1.2.2.3 Visualisierung

Die Ergebnisse einer Prolific-Suche werden visuell anschaulich in einer Heatmap dargestellt (siehe Abbildung 14). Eine Zelle der Heatmap gibt Aufschluss über die Anzahl des gemeinsamen Vorkommens der jeweiligen Kleinstruktur in der Zeile, sowie des Proteins in der dazugehörigen Spalte in einem Artikel. Für eine bessere Übersicht sind die Zellen farbkodiert.

In den Spalten sind alle zum Suchprotein ähnlichen Proteine aufgelistet. Neben dem Proteinnamen und der Identifikationsnummer sind auch Informationen zur Ähnlichkeit und weiterführende Hyperlinks zu jedem Protein vorhanden. Die Proteine wurden nach ihrem *gene symbol* gruppiert. Dies geschah hauptsächlich, um homologe Proteine aus unterschiedlichen Organismen zusammenzufassen und die Ergebnisse besser zu strukturieren. Die Proteine sind nach ihrer Ähnlichkeit zur Suchstruktur sortiert.

Die Zeilen repräsentieren Kleinstrukturen und weisen etliche Informationen, wie Synonyme oder PubChem-ID, auf. Die Struktur des Moleküls wird automatisch aus dem

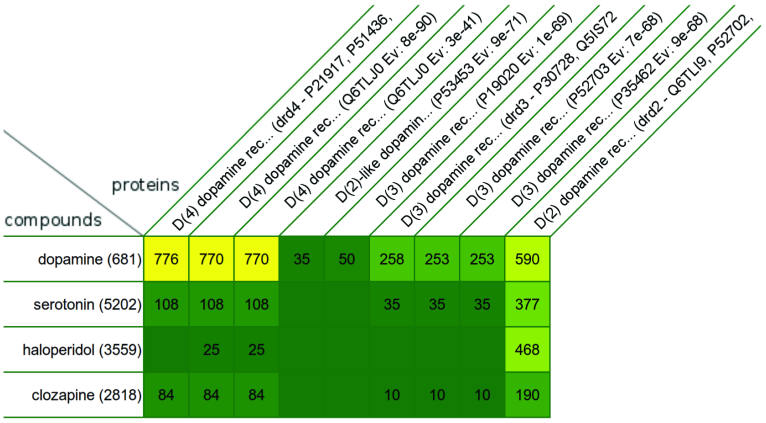

Abbildung 14: Prolific-Heatmap - Zeilen repräsentieren Kleinstrukturen und Spalten Proteine. Die Zahl, sowie der Farbcode der Zellen, geben an wie viele Artikel mit dem jeweiligen Struktur-Protein-Paar in der Prolific-Datenbank gefunden wurden. Sortiert ist die Tabelle nach der Zeilensumme, so dass das häufigste Triple Kleinstruktur-Artikel-Protein am Kopf der Tabelle steht.

dazugehörigen SMILES berechnet und dem Benutzer auf Wunsch angezeigt. Kleinstrukturen mit Synonymen wie *diclofenac sodium* oder *diclofenac potassium* werden mithilfe der PubChem *Parents* gruppiert. Die Zeilen werden anhand ihrer Zeilensumme sortiert, so dass die Struktur mit den meisten simultanen Nennungen über alle Proteine am weitesten oben steht.

Standardmäßig wird die Heatmap gefiltert, so dass nur Relationen angezeigt werden, die über eine minimale Anzahl von gemeinsamen Struktur-Protein-Nennungen liegen. Dies soll vor allem dazu dienen, das natürliche Rauschen durch schlechte Synonyme zu verhindern und dadurch die Heatmap übersichtlicher zu gestalten. Dieser Grenzwert ist vom Benutzer in den Einstellungen frei adjustierbar. Jede Zelle ist mit einer Artikelliste verlinkt, in der der Benutzer alle Informationen zu den gefundenen Artikeln abrufen kann. In Abbildung 15 ist ein exemplarischer Artikel in der Detailansicht gezeigt.

Die Synonyme der identifizierten Kleinstruktur und des Proteins werden visuell hervorgehoben und sind Hyperlinks zu den Datenbanken PubChem bzw. UniProt. Jeder Artikel ist mit PubMed verlinkt und ermöglicht es so, bequem zur Originalpublikation mit Volltext zu gelangen. Um eine große Anzahl an Artikeln betrachten zu können, wurde die Navigation mithilfe der Tastatur implementiert. Mit den Pfeiltasten ist es möglich, sich alle Artikel zu einem Protein-Kleinstruktur-Paar effizient anzuschauen, vergleichbar einer Bildergalerie.

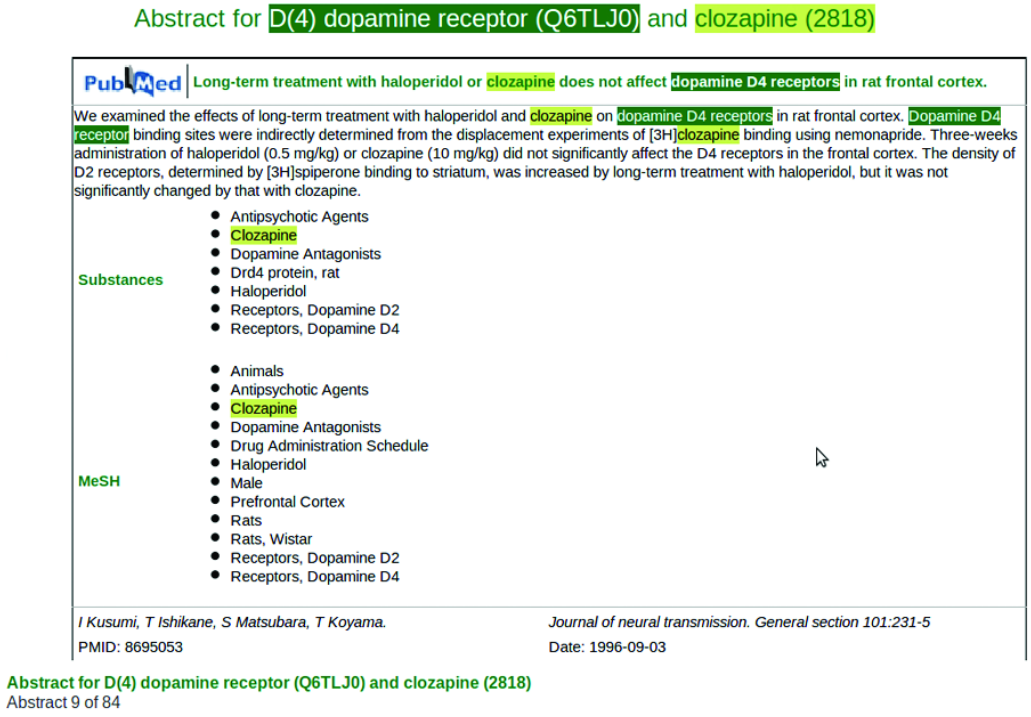

Abbildung 15: Prolific-Detailansicht - Detailansicht eines Abstracts in Prolific mit den gefundenen Entitäten Clozapin und Dopamin-D4-Rezeptor. Beide Kookkurenzen werden farblich hervorgehoben und haben Hyperlinks zu den Datenbanken PubChem und UniProt.

1.2.2.4 Interaktion zwischen Prolific und CIL

Die Benutzerfreundlichkeit von Prolific spielte nicht nur beim Design der Benutzeroberfläche eine wichtige Rolle, sondern auch bei der Anbindung an andere Webservices, wie dem in Kapitel 2.2 beschriebene Webservice CIL. Kleinstrukturen, die in Prolific als potentielle Interaktionspartner identifiziert wurden, können als Ausgangspunkt für eine Suche in CIL dienen. Dazu muss nur der entsprechende Hyperlink neben jeder Struktur angeklickt werden. Der Benutzer wird dann auf den CIL-Webservice weitergeleitet, in dem automatisch eine Suche gestartet wird.

Unter Umständen möchte man gefundene ähnliche Proteine mit interessanten Kleinstrukturinteraktionen näher untersuchen und mit diesen eine erneute Prolific-Suche starten. Hierzu wurden Hyperlinks neben den Proteinnamen platziert.

1.2.3 Diskussion

Wie unter 1.2.2.2 beschrieben, sind die Ergebnisse von Prolific stark von der Qualität der PubChem- und UniProt-Synonyme, die in der Datenbank gespeichert werden, abhängig. Daher muss die Prolific-Datenbank regelmäßig aktualisiert und andere Synonymquellen integriert werden. Zur Zeit werden große Anstrengungen unternommen, das Vokabular auf Seiten der NCBI und auch der UniProt zu vereinheitlichen und zu säubern. Dies sollte sich positiv auf die Qualität der Synonymnamen auswirken.

Auf absehbare Zeit wird es nicht möglich sein, alle Volltexte von Publikationen zu durchsuchen und damit die Suchqualität -und quantität zu erhöhen. Alle frei zugänglichen Artikel aus PubMed Central[1] sollten aber in einer zukünftigen Version von Prolific indiziert und damit ein Teil der Datenbank werden. Prinzipiell stehen auch Metadaten, wie z.B. das Erscheinungsdatum der Publikation, in der erstellten Datenbank zur Verfügung. Aus diesen könnten Trends zu bestimmten Protein-Kleinstruktur-Interaktionen abgeleitet werden.

Wünschenswert wäre auch eine Anbindung an KEGG (82, 83) oder Wikipathways (84) bzw. eine komplette Migration des Prolific-Services in einen Galaxy-Workflow mit Text Mining-Modulen. Kooperationen und Ideen sind hierzu vorhanden.

[1] `http://www.ncbi.nlm.nih.gov/pmc`

2

Analyse des chemischen Raums

Frank K. Brown definierte 1998 den Begriff der Cheminformatik[1] als das Zusammenführen von Informationsressourcen, um Daten in Informationen und Informationen in Wissen zu transferieren, mit dem Ziel, den Entscheidungsprozess in der Wirkstoffidentifikation und Wirkstoffoptimierung schneller und besser zu gestalten (85, 86).

Für die Identifizierung neuer Wirkstoffe ist die systematische Untersuchung des chemischen Raumes, der eine nahezu unendliche Anzahl an möglichen organischen Verbindungen enthält (87, 88), ein entscheidender Schritt, und stellt die Cheminformatik vor große Herausforderungen. Das Ziel ist es, diesen chemischen Raum mit einer reduzierten, jedoch diversen Sammlung an vielversprechenden Molekülen abzubilden, um diese in *High-throughput-Screenings* (89) zu testen. Naturstoffe und wirkstoffähnliche Substanzen werden ebenfalls in die Entwicklung adäquater chemischer Bibliotheken einbezogen, sowie verschiedenste semiempirische Verfahren zur Bewertung von Kleinstrukturen (90, 91).

Dieses Kapitel beschäftigt sich mit cheminformatischen Grundlagen und Methoden, die zum Design einer Strukturbibliothek und zur Annotation von Molekülen notwendig sind. Ein System, welches diese Methoden modular, reproduzierbar und skalierbar bis zu mehreren 100 Millionen Molekülen implementiert, wird in Kapitel 2.1 vorgestellt. In Kapitel 2.3 wird ein Informationsportal für Metabolite, die von Streptomyceten synthetisiert werden, beschrieben. Die Integration von Primärliteratur und Cheminformatik wird in Kapitel 2.2 ausführlich behandelt.

2.1 ChemicalToolBox

Eine immer größer werdende Menge an frei verfügbaren Daten stehen den Wissenschaftlern zur Verfügung und können zur Beschleunigung und Verbesserung von Vorhersagen und Entscheidungen beitragen. Gleichermaßen existiert eine Vielzahl an Programmen und Techniken, die diese Daten in Informationen überführen und aufbereiten können. Beides wird jedoch viel zu wenig in Laboren genutzt. Das Problem, wie in Abschnitt 1.3 dargestellt, ist, dass die meisten Programme nicht benutzerfreundlich genug sind. Dies beginnt bei der oft komplizierten Installation der einzelnen Programme, der unzureichenden Dokumentation, dem Fehlen einer intuitiven Benutzeroberfläche oder der fehlenden Skalierbarkeit auf die wachsende Datenmenge.

[1]Cheminformatik und Chemoinformatik können äquivalent benutzt werden; Cheminformatik wird jedoch häufiger verwendet.

Auflistung 2.1: XML-basierte Softwarebeschreibung für Galaxy

```
<tool id="ctb_np-likeness-calculator" name="Natural Product" version="0.2.1">
    <description>likeness calculator</description>
    <parallelism method="multi" split_inputs="query" split_mode="to_size"
        split_size="1000" merge_outputs="output1"></parallelism>
    <requirements>
        <requirement type="package" version="2.0">np-likeness-scorer</requirement>
    </requirements>
    <stdio>
        <exit_code range="1:" />
        <exit_code range=":-1" />
    </stdio>
    <command>
    <![CDATA[
        ## NPLC is really picky with file extensions.
        ## A workaround is to create a symlink with a proper file-extension.
        #set $temp_link = "temp_file.%s" % ($infile.ext)
        ln -s $infile $temp_link;

        java -jar \$NPLS_JAR_PATH/NP-Likeness-2.0.jar
            -in "${temp_link}"
            -out "${outfile}"
            #if $reconstruct_fragments:
                -reconstructFragments true
                -outFragments $outfragments
            #end if
        &&
        ## replace space with tabular to get it into a proper SMILES format
        sed -i 's/ /\t/g' "${outfile}"
    ]]>
    </command>
    <inputs>
        <param format="smi,sdf" name="infile" type="data"
            label="Molecule file" help="Dataset missing? See TIP below"/>
        <param name='reconstruct_fragments' type='boolean'
            truevalue='' falsevalue=''
            label='Fragments and scores are written out in SMILES format' />
    </inputs>
    <outputs>
        <data format_source="infile" name="outfile" />
        <data format="tabular" name="outfragments">
            <filter>reconstruct_fragments is True</filter>
        </data>
    </outputs>
    <help>
<![CDATA[

**What this tool does**

The 'Natural-Product-Likeness Scorer'_ calculates the Natural Product(NP)-likeness of
a molecule, i.e. the similarity of the molecule to the structure space covered by known
natural products. The more positive the score, the higher is the NP-likeness and vice versa.

.. _Natural-Product-Likeness Scorer: http://sourceforge.net/projects/np-likeness/
.. image:: $PATH_TO_IMAGES/score_distribution.png

]]>
    </help>
    <citations>
        <citation type="doi">10.1186/1471-2105-13-106</citation>
        <citation type="doi">10.1021/ci700286x</citation>
    </citations>
</tool>
```

Im Zuge dieser Arbeit wurde eine, auf dem Galaxy-Framework (siehe Kapitel 2) aufbauende Sammlung an cheminformatischen Tools erstellt, die die bereits besprochenen Unzulänglichkeiten behebt und eine Brücke zwischen enormen Datenmengen, spezialisierten Tools und dem Wissenschaftler schlägt.

2.1.1 Methoden

2.1.1.1 Galaxy-Integration

Prinzipiell ist jedes Programm, welches über die Kommandozeile (command-line interface, CLI) aufrufbar und mit Parametern steuerbar ist, in Galaxy integrierbar. Dabei ist die verwendete Programmiersprache des jeweiligen Programms irrelevant, da nur ein kleiner, dynamischer Adapter zwischen Kommandozeilenaufruf und Galaxy erstellt werden muss. Dieser Adapter, auch Wrapper genannt, kommuniziert zwischen dem eigentlichen Programm und Galaxy, um z.B. die Eingabe- und Ausgabe-Dateien zu verwalten. Ein Wrapper wird in XML definiert und bietet ein hohes Maß an Flexibilität, um auch komplexe Programme mit einem intuitiven Benutzerinterface in Galaxy zu integrieren. Im Folgenden (Auflistung 2.1) wird beispielhaft ein Wrapper für die Vorhersage von Naturstoffen (92, 93) näher beschrieben und auf die wichtigsten sechs Bestandteile eingegangen.

Wrapper-Definition und -Abhängigkeiten [Zeile 1-6]

Als erstes werden für jeden Wrapper ein eindeutiger Identifikator (ID), eine Versionsnummer und ein Name bzw. eine kurze Beschreibung definiert. Der Identifikator wird zusammen mit der Versionsnummer genutzt, um das Tool eindeutig zu referenzieren, um es zum Beispiel mit allen Einstellungen in der History abzuspeichern. Der Tool-Name und die Beschreibung sollten kurz aber aussagekräftig sein. Sie werden in der linken Tool-Leiste in Galaxy angezeigt und erlauben es dem Benutzer, danach zu suchen und einen ersten Eindruck über die Funktionalität des Tools zu bekommen. Der optionale *parallelism*-Eintrag in Zeile 3 ermöglicht es Galaxy, automatisch multiple Prozessoren zu benutzen, auch wenn das zugrunde liegende Programm, hier *Natural product likeness calculator* (92, 93), dies nicht unterstützt. Die Eingabe-Datei wird dabei in kleinere Dateien aufgeteilt und das Programm auf jeder dieser einzelnen Dateien parallel aufgerufen.

Ein weitere wichtige Funktion verbirgt sich hinter dem optionalem *requirement*-Eintrag. Dieser spezifiziert eine Abhängigkeit zu einem oder mehreren Programmen, die für die

Funktionalität des Wrappers essentiell sind. Diese Abhängigkeiten können in einer weiteren XML-Datei definiert und vom Galaxy-Administrator installiert werden. In Verbindung mit dem Identifikator und der Versionsnummer erlaubt die Definition des *requirement*-Eintrags, verschiedene Versionen des Programms und des Wrappers parallel zu installieren, was somit ein hohes Maß an Reproduzierbarkeit wissenschaftlicher Ergebnisse ermöglicht.

Definition des Kommandozeileninterfaces [Zeile 7-19]

In diesem Teil des Wrappers wird die Kommunikation mit dem Programm definiert. Der auszuführende Befehl wird dabei mithilfe der Cheetah-*template-engine*[1] definiert. Cheetah weist einen an Python angelegten Syntax auf und ermöglicht ein hohes Maß an Flexibilität, um auch komplexe Programme in Galaxy zu integrieren. Dabei kann auf Einstellungen des Benutzers dynamisch reagiert werden und z.B. die Ein- und Ausgabe von Dateien geändert werden. Galaxy versteht es mit Standard-Datenströmen wie Standardausgabe (stdout) oder Standardfehlerausgabe (stderr) umzugehen und speichert diese zu jedem Aufruf in der History. Sollte die Standardfehlerausgabe Informationen enthalten, wird der Programmaufruf als fehlerhaft erachtet und entsprechend markiert. Einige Programme schreiben aber, fälschlicherweise, Zusatzinformationen in die Standardfehlerausgabe, was wiederum zu einem falsch-negativen Programmende führt. Um diesem zu begegnen, steht dem Entwickler der optionale *stdio*-Eintrag zur Verfügung. Dieser ermöglicht es, Standardausgabe und Standardfehlerausgabe mithilfe von regulären Ausdrücken zu untersuchen, und entsprechend darauf zu reagieren, um falsch-negative Programmaufrufe zu verhindern. Darüber hinaus gibt es einige sehr spezialisierte Optionen, auf die hier nicht weiter eingegangen werden soll.

Definition der Eingabe-Datei/Dateien und Parameter [Zeile 20-23]

Der Eingabe-Teil (Input) eines jeden Wrappers dient zur Definition des Benutzerinterfaces und legt fest, welche Parameter dem Benutzer angeboten werden. Die XML-Definitionen werden hierbei in HTML-Eingabefelder umgewandelt und eine typische Eingabemaske, wie sie von anderen Webseiten bekannt ist, generiert. Galaxy stellt hierfür verschiedene Standard-Eingabefelder, wie die zur Dateiauswahl oder Zahlen- und Textfelder zur Verfügung. Für jeden Parameter können voreingestellte Werte definiert werden und es

[1] `http://www.cheetahtemplate.org`

besteht die Möglichkeit, jede Einstellung ausführlich zu dokumentieren. Jeder Parameter und der vom Benutzer spezifizierte Wert werden zurück an das Template übergeben und stehen in der Kommandozeilen- und Ausgabesektion, der XML-Datei, zur Verfügung. Dies ermöglicht es, dynamisch auf Benutzereinstellungen zu reagieren und z.B. bestimmte Kommandozeilenparameter nur zu setzen, wenn eine bestimmte Option ausgewählt ist, wie in Auflistung 2.1 beschrieben.

Beim Design der verschiedenen Eingabemasken aller unterschiedlichen Programme wurde in der ChemicalToolBox besonders auf die Benutzerfreundlichkeit geachtet. Komplexe Programme wie *obabel* bieten die unterschiedlichsten Funktionen und Parameter an, einige sind nur valide in einem bestimmten Kontext, zum Beispiel bei einem bestimmtem Eingabe-Dateiformat, andere sind sehr speziell und bieten sich in der Standardmaske nicht an. Mittels sogenannter Bedingungen (engl. conditionals) kann auf bestimmte Eingaben des Benutzers dynamisch reagiert und z.B. zusätzliche Parameter ein- oder ausgeblendet werden. Dies wurde unter anderem dazu genutzt, selten verwendete Parameter nur unter den erweiterten Einstellungen anzubieten und somit das Standardinterface nicht zu überladen. Eine weitere Funktion, die Galaxy zur Erstellung der Eingabemasken anbietet, ist die beliebige Wiederholung von Teilbereichen einer Oberfläche. Dies wurde unter anderem im Filter-Tool der ChemicalToolBox verwendet. Der Benutzer hat so die Freiheit, nach einer Eigenschaft zu filtern oder nach beliebig vielen Eigenschaften zeitgleich. Die Oberfläche passt sich automatisch an und zeigt nur so viele Filtereinstellungen wie gewünscht sind (als Beispiel siehe Abbildung 20). Ein weiteres Anwendungsbeispiel für das Wiederholungselement sind Tools, die eine variable Anzahl an Eingabedateien erlauben.

Definition der Ausgabe-Datei/Dateien [Zeile 24-29]

In der Ausgabe-Beschreibung (output) eines Wrappers kann für jede generierte Ausgabedatei ein Eintrag definiert werden. Hierbei können Dateityp und Dateibezeichnung in der History spezifiziert werden. Beides kann dynamisch erzeugt werden. Zum Beispiel kann der Historyname abhängig vom Tool- oder vom Eingabenamen definiert werden. Der Dateityp kann abhängig von einer Benutzereingabe vergeben werden. Ein besonders anschauliches Beispiel ist im Convert-Tool der ChemicalToolBox gegeben (Auflistung 2.2). Die Auswahl des Benutzers, in welches Format seine Moleküle konvertiert werden sollen, wird genutzt, um das Ausgabeformat anzupassen. Einige Tools bieten zudem die Möglichkeit, zusätzliche Daten zu generieren, und haben damit eine variable Anzahl an Ausgabedateien. Dies ist in

Abbildung 2.1 am Beispiel des *natuaral product likeness scorers* veranschaulicht. Wenn die Option *reconstruct fragments* erwünscht ist, wird eine zusätzliche Ausgabedatei erzeugt, da die Bedingung im Filter erfüllt ist.

Auflistung 2.2: Dynamische Adaption des Ausgabeformates

```
<tool id="ctb_compound_convert" name="Compound Convert" version="0.1">
2
    ...
4
    <outputs>
6       <data name="outfile" type="data" format="text"
            label="Convert to ${oformat.oformat_opts_selector} from ${on_string}">
8           <change_format>
                <when input="oformat.oformat_opts_selector" value="sdf" format="sdf"/>
10              <when input="oformat.oformat_opts_selector" value="can" format="smi"/>
                <when input="oformat.oformat_opts_selector" value="smi" format="smi"/>
12              <when input="oformat.oformat_opts_selector" value="mol2" format="mol2"/>
                <when input="oformat.oformat_opts_selector" value="inchi" format="inchi"/>
14              <when input="oformat.oformat_opts_selector" value="cml" format="cml"/>
                <when input="oformat.oformat_opts_selector" value="mol" format="mol"/>
16              <when input="oformat.oformat_opts_selector" value="pdb" format="pdb"/>
                <when input="oformat.oformat_opts_selector" value="fs" format="obfs"/>
18          </change_format>
        </data>
20
    ...
```

Unit-Tests [Zeile 30-31]

In diesem Abschnitt können alle Parameter inklusive einer Test-Eingabedatei und einer Ausgabedatei definiert werden. Galaxy testet den Wrapper, indem es das Programm mit spezifizierten Testparametern auf der Eingabedatei ausführt und das Ergebnis mit der gegebenen Ausgabedatei des Testabschnittes vergleicht. Sollten sich keine Unterschiede ergeben, ist der Test positiv verlaufen. Unit-Tests können und sollten vom Galaxy-Administrator regelmäßig ausgeführt werden, um etwaige Fehler frühzeitig zu erkennen und zu beheben. Vor allem aber sind diese Tests für den Toolentwickler ein nützliches Werkzeug, um die Funktionalität des Wrappers zu gewährleisten.

Erläuterungen und Hilfe [Zeile 32-63]

Der letzte Abschnitt eines Wrappers bietet Platz für die Dokumentation der Funktionalität des Wrappers und des zugrundeliegenden Programms. Die schon in der Eingabesektion gegebene Erklärung der einzelnen Parameter kann hier aufgegriffen und näher erläutert werden. Generell gibt es aber keine Vorgaben und die Qualität der Hilfe liegt einzig und allein beim Entwickler. Gute Hilfen beinhalten sowohl theoretisches Grundwissen für die Abschätzung der richtigen Parameter als auch praktische Tipps, wie nach und vor der

Benutzung des Tools mit den Daten verfahren werden kann. Referenzen zu externen Dokumentationen und Publikationen zählen genauso wie Beispiele zu Ein- und Ausgaben, und Hinweise zur *best practice*. Die Hilfe wird in der *markup* Sprache, *restructuredText* geschrieben und ermöglicht es, relativ einfach eine anschauliche Dokumentation zu erstellen, die von Galaxy in HTML umgewandelt und dem Benutzer präsentiert wird.

Eine ausführliche und verständliche Dokumentation ist elementar für ein Programm. Ist die Dokumentation unzulänglich, wird das Programm meistens nicht oder falsch benutzt. Daher sollte bei der Tool-Integration viel Wert auf die Gestaltung des Benutzerinterfaces und der Hilfe gelegt werden, idealerweise mit Beispielen von Ein- und Ausgaben.

2.1.1.2 Einbindung in die Galaxy Tool Shed

Neben einer guten Dokumentation ist die einfache Installierbarkeit ein weiterer entscheidender Faktor für die Zugänglichkeit von Programmen. Tools, die einfach in ein bestehendes System zu integrieren sind, finden eine große Verbreitung und fördern dadurch die Qualität und Langzeitverfügbarkeit. Dies ergibt sich daraus, dass Förderorganisationen erfolgreiche Projekte mit nachweisbaren Nutzerzahlen längerfristig fördern, aber auch weil eine größere Gemeinschaft zur Verfügung steht, deren Interesse eine hohe Qualität des Tools ist.

Um die Installation von zusätzlichen Tools in die Galaxy-Plattform so einfach wie möglich zu gestalten, wurde die Galaxy Tool Shed von Blankenberg *et al.* entwickelt (94). Entwickler können ihre Tools in eine zentrale Tool Shed speichern und somit für alle Galaxy-Benutzer zugänglich machen. Dieses Konzept ist vergleichbar mit dem *App Store* von Apple Inc. oder dem *Google Play* von Google Inc.. Tools können durchsucht und bewertet werden und es gibt Vorschauoptionen, sowie detaillierte Toolbeschreibungen. Ein Galaxy-Administrator kann, wie ein Smartphonebenutzer, die Galaxy Tool Shed durchsuchen und Tools installieren. Dabei werden nicht nur die in Abschnitt 2.1.1.1 beschriebenen XML-Dateien heruntergeladen, sondern auch die definierten Abhängigkeiten. Hierzu unterstützt Galaxy eine Beschreibungssprache zur Installation von Programmen, mit der es zum Beispiel möglich ist, *Open Babel*, eine Abhängigkeit der ChemicalToolBox, zu installieren. Als Beispiel sei hierzu auf Auflistung 2.3 verwiesen.

Die ChemicalToolBox hat durch ihre sehr umfassende Sammlung an Tools eine sehr große Zahl an Abhängigkeiten, von denen selbst wiederum einige Abhängigkeiten aufweisen. Eine Übersicht ist in Abbildung 16 gegeben. Der Galaxy-Administrator hat die Ent-

scheidung, das Paket ChemicalToolBox zu installieren, welches alle anderen Abhängigkeiten automatisch installiert, oder individuell nur spezifische Tools zu installieren.

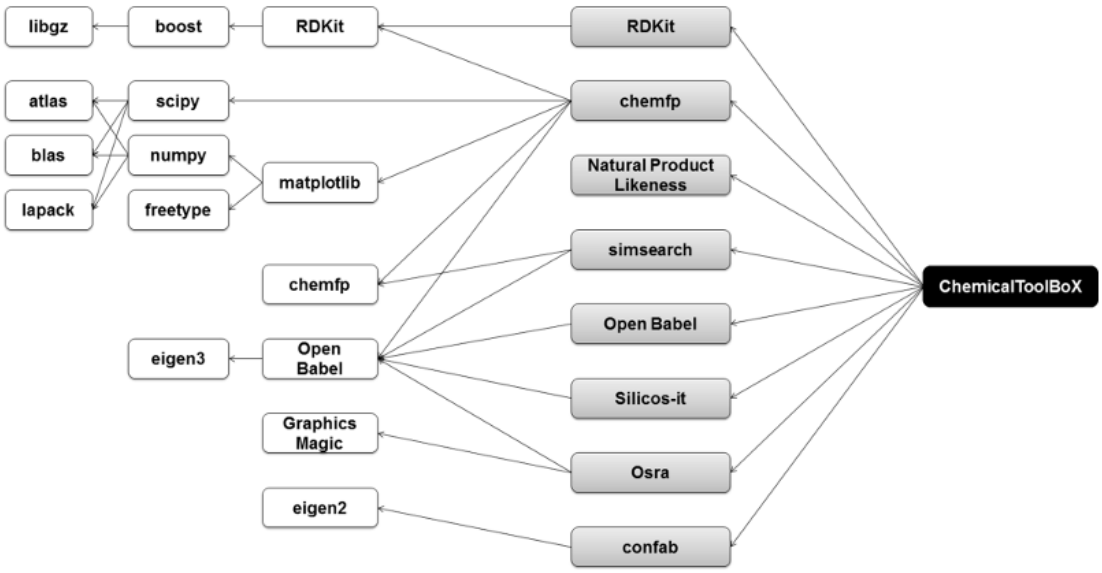

Abbildung 16: Abhängigkeitsgraph der ChemicalToolBox - Die ChemicalToolBox (in schwarz) ist als *repository suite* implementiert und definiert eine Reihe von Abhängigkeiten zu diversen Programmen (in grau). Jedes dieser Programme hat wiederum Abhängigkeiten zur Laufzeit oder zur Kompilierzeit (in weiß). Für eine erfolgreiche Installation der kompletten ChemicalToolBox müssen alle Abhängigkeiten entgegengesetzt des gerichteten Abhängigkeitsgraphen in der korrekten Reihenfolge installiert werden.

Neben Tools und Abhängigkeiten, können in der Galaxy Tool Shed auch Definitionen zu Datenformaten und Workflows gespeichert werden. Die ChemicalToolBox erweitert Galaxy um 10 neue Datenformate, darunter SDF, SMILES und das PDB-Format[1]. Workflows basierend auf der ChemicalToolBox sind auch in der Tool Shed zu finden[2],[1] und können in beliebige Galaxy-Instanzen installiert werden.

[1]`https://toolshed.g2.bx.psu.edu/view/iuc/molecule_datatypes`
[2]`https://toolshed.g2.bx.psu.edu/view/bgruening/chemicaltoolbox_library_hole_filling_workflow`
[1]`https://toolshed.g2.bx.psu.edu/view/bgruening/chemicaltoolbox_merging_chemical_databases_workflow`

Auflistung 2.3: Installationsbeschreibung von Open Babel

```
<tool_dependency>
    <package name="eigen3" version="3.1.3">
        <repository name="package_eigen_3_1" owner="iuc" prior_installation_required="True" />
    </package>
    <package name="openbabel" version="2.3.2">
        <install version="1.0">
            <actions>
                <action type="download_by_url">
                    https://github.com/openbabel/openbabel/archive/openbabel-2-3-2.tar.gz
                </action>
                <!-- populate the environment variables from the dependend repos -->
                <action type="set_environment_for_install">
                    <repository name="package_eigen_3_1" owner="iuc">
                        <package name="eigen3" version="3.1.3" />
                    </repository>
                </action>
                <action type="shell_command">
                    cmake . -DPYTHON_BINDINGS=ON -DCMAKE_INSTALL_PREFIX=$INSTALL_DIR
                   -DEIGEN3_INCLUDE_DIR=$EIGEN3_SOURCE_PATH
                   -DPYTHON_LIBRARY=`python -c 'import distutils.sysconfig;
                    print "%s/libpython%s.so" % (distutils.sysconfig.get_config_var("LIBPL"),
                    distutils.sysconfig.get_python_version())'`
                </action>
                <action type="shell_command">make</action>
                <action type="shell_command">make install</action>
                <action type="set_environment">
                    <environment_variable name="PATH" action="prepend_to">
                        $INSTALL_DIR/bin
                    </environment_variable>
                    <environment_variable name="PYTHONPATH" action="prepend_to">
                        $INSTALL_DIR/lib
                    </environment_variable>
                    <!-- internal variables for OpenBabel -->
                    <environment_variable name="BABEL_DATADIR" action="set_to">
                        $INSTALL_DIR/share/openbabel
                    </environment_variable>
                    <environment_variable name="BABEL_LIBDIR" action="set_to">
                        $INSTALL_DIR/lib/openbabel/2.3.2
                    </environment_variable>
                    <!-- galaxy variables for other tool wrappers -->
                    <environment_variable name="OPENBABEL_LIB_DIR" action="set_to">
                        $INSTALL_DIR/lib
                    </environment_variable>
                    <environment_variable name="OPENBABEL_INCLUDE_DIR" action="set_to">
                        $INSTALL_DIR/include
                    </environment_variable>
                    <environment_variable action="prepend_to" name="LD_LIBRARY_PATH">
                        $INSTALL_DIR/lib/
                    </environment_variable>
                    <environment_variable action="prepend_to" name="CPLUS_INCLUDE_PATH">
                        $INSTALL_DIR/include
                    </environment_variable>
                    <environment_variable action="prepend_to" name="C_INCLUDE_PATH">
                        $INSTALL_DIR/include
                    </environment_variable>
                </action>
            </actions>
        </install>
        <readme>
        Compiling OpenBabel requires g++ and CMake 2.4+. Optional but required for a few
        features are the cairo development libraries.
        OPENBABEL_INCLUDE_DIR and OPENBABEL_LIB_DIR can be accessed from other tool wrappers.
        </readme>
    </package>
</tool_dependency>
```

2.1.2 Ergebnisse

2.1.2.1 Tools in der ChemicalToolBox

In die initiale Version der ChemicalToolBox wurden 43 Tools integriert. Der Fokus lag in dieser Version darauf, ein breites funktionales Spektrum abzubilden und die Diversität der verschiedenen Programme und verwendeten Techniken, sowie der Programmiersprachen, hervorzuheben. Die Angaben zur ChemicalToolBox beziehen sich auf die erste offizielle Version (0:087550f392d0[1]) und unterliegen ständigen Änderungen, vor allem in den mitgelieferten Tools. Eine ausführliche Liste der Tools befindet sich in Tabelle 2.1. Im folgenden wird auf die Schwerpunkte der ChemicalToolBox eingegangen, und in Kapitel 2.1.2.2 werden einige Anwendungsbeispiele näher erläutert.

Importieren von Molekülen und Molekülbibliotheken

Das Arbeiten mit Molekülen in der ChemicalToolBox setzt voraus, dass die Moleküle in das System geladen werden. Abhängig von der Größe der zu bearbeitenden Bibliothek kann schon dieser erste Schritt zu einem Flaschenhals im Arbeitsablauf werden. Die ChemicalToolBox bietet hierfür eine Reihe an unterschiedlichen Tools an. Einzelne Moleküle können manuell im Webbrowser gezeichnet werden und werden dann als SMILES oder im SD-Format in der Benutzerhistory abgespeichert. Kleinere Datenmengen (unter 2 GB) können problemlos via Webbrowser in die ChemicalToolBox geladen werden. Größere Datenmengen können Probleme bereiten und der Upload mithilfe eines FTP (File Transfer Protocol)-Programms wird empfohlen. Sollten die Bibliotheken bereits im Internet verfügbar sein, kann die jeweilige URL in das Upload-Tool kopiert werden und diese werden dann direkt auf den Server geladen. Dies geschieht ohne die Daten auf den Benutzerrechner zu laden, welcher eventuell selbst sehr Ressourcen-limitiert ist. Das Upload-Tool kann komprimierte Archive entpacken und deren Inhalt, sollten es mehrere Dateien sein, zusammenfügen. Nicht-Moleküldateien können gegebenenfalls ignoriert werden.

Eine der größten öffentlichen Moleküldatenbanken ist die von der NCBI entwickelte PubChem (75). Diese bietet ihre Moleküle im SD-Format an, verteilt auf mehrere 100 komprimierte Dateien. Dies birgt zwei Probleme: Zum einen die Verteilung auf mehrere Dateien, die alle separat in die ChemicalToolBox geladen werden müssten und ständigen Änderungen unterworfen sind, zum anderen die Größe des Datensatzes im SD-Format. Das

[1] `https://toolshed.g2.bx.psu.edu/view/bgruening/chemicaltoolbox/087550f392d0`

Tabelle 2.1: Exemplarische Liste von Programmen in der ChemicalToolBox. Jede Zeile beschreibt ein Tool der ChemicalToolBox. Alle Tools sind auf `https://github.com/bgruening/galaxytools` verfügbar und in der Galaxy Tool Shed hinterlegt.

Galaxy-Tool-Name	Kurzbeschreibung	Referenz(en)
Molecule editor	Eingabemodul	(95)
PubChem downloader	Eingabemodul	(95)
Molecule recognition	optische Strukturerkennung	(96)
Compound convert	Konvertierung von Dateiformaten	(97, 98)
Compound search	Suche eines SMILES/SMARTS	(97)
Multi-Compound search	Suche einer Liste von SMILES oder SMARTS	(97)
Remove counterions/fragments	Entfernen von Gegenionen & Fragmenten	(98)
Remove duplicated molecules	Entfernen von strukturell identischen Strukturen	(97)
Molecule filter	Filtern von Molkülen nach Eigenschaften	(98)
Remove small molecules	Entfernen von kleinen Molekülen	(97)
Spectrophore search	Spectrophore-Suche	(99, 100)
Similarity search	Molekülähnlichkeitssuche	(97, 101)
Substructure search	Substruktursuche	(97)
Physicochemical properties	Berechnung von physikochemischen Eigenschaften	(95)
Add hydrogen atoms	Hinzufügen von Wasserstoffatomen	(95)
Remove protonation state	Entfernung des Protonierungszustandes	(95)
Change title	Änderung der Molekülbeschreibung	(95)
Strip-it	Extraktion von vordefinierten Scaffolds	Silicos-it
Shape-it	3D-Strukturalignment	Silicos-it
Align-it	Pharmakophor-Alignment	Silicos-it
Conformer calculation	Konformerberechnung	(102)
Molecules to fingerprints	Fingerprint-Berechnung	(97, 101)
Drug likeness	Klassifikator für Wirkstoffähnlichkeit	(103)
NxN clustering	Ähnlichkeitsclustering von Molekülen	(101)
Taylor-Butina clustering	Clustering nach Taylor-Butina	(101)
MDS scatter plot	Multidimensionale Skalierung & Visualisierung	(95)
Opsin	Konvertiert IUPAC-Namen zu Strukturen	(104)
Natural product likeness scorer	Klassifikator für Naturstoffähnlichkeit	(92, 93)
Compound Visualization	Graphische Darstellung von Strukturen	(95)

SD- sowie auch das CML-Format haben ein sehr schlechtes Größen-Information-Verhältnis, welches die PubChem auf über 100 GB in unkomprimierter Form anwachsen lässt. Für viele Anwendungsfälle sind die Informationen, die in einer SD-Datei kodiert werden können, wie z.B. 2D- oder 3D-Koordinaten, nicht relevant und die gleichen Strukturen können in einer SMILES-Datei in ca. 4 GB gespeichert werden. Die ChemicalToolBox beinhaltet daher ein spezielles Tool, welches jede einzelne PubChem-Datei herunterlädt, entpackt, in SMILES konvertiert und in einen Datensatz zusammenfügt. Dem Benutzer wird eine Ausgabedatei offeriert, welche die komplette PubChem im SMILES-Format beinhaltet. Abhängig vom Server können mehrere Dateien parallel konvertiert werden. Dabei werden zu keinem Zeitpunkt 100 GB an Speicher belegt, was einen großen Vorteil für die Administration bedeutet.

Für die Zukunft sind spezielle Import-Optionen in den frei zugänglichen Datenbanken, wie z.B. PubChem (75), ChEMBL (105), DrugBank (106), geplant. Diese bieten dann einen direkten Import von der jeweiligen Webseite in die ChemicalToolBox an.

Konvertierung

Eine sehr wichtige Funktion für die Benutzerfreundlichkeit der ChemicalToolBox ist das einfache Konvertieren zwischen den verschiedenen Formaten, wie z.B. SDF, SMILES, InChI (107), mol2. Das *Convert*-Tool bietet diese Möglichkeit mit einer Vielzahl an unterschiedlichen Optionen für jedes Format. Darüber hinaus wurden für alle Konvertierungsmöglichkeiten insgesamt 31 Konverter erstellt, die die Formate ineinander überführen. Diese Konverter stellen weniger Optionen zur Verfügung, als das universelle *Convert*-Tool, sind aber durch die tiefere Integration in das Galaxy-Framework fest assoziiert mit einem Datensatz. Dies bedeutet, das Konvertierungen als *post-processing*-Schritt in einem Workflow initiiert werden können oder über das Eigenschaftsmenü eines Datensatzes in der History erreichbar sind (siehe Abbildung 18).

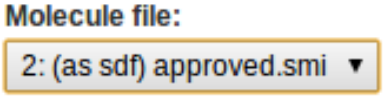

Abbildung 17: Automatische Konvertierung von Molekülformaten - Sollte ein Programm nur mit SDF arbeiten können, konvertiert Galaxy automatisch SMILES in SDF. Dies ist angezeigt als "(as sdf)" in einem Dateieingabefeld und erfolgt für alle Dateiformate, die Konvertierungsprogramme definiert haben.

Sollte ein Tool als Eingabe nur Dateien im SD-Format erlauben, bieten diese Konverter die Möglichkeit, andere Formate automatisch ohne Benutzerinteraktion in das SD-Format zu überführen (siehe Abbildung 17). Dies geschieht dann als *pre-processing* vor dem eigentlichen Programmaufruf.

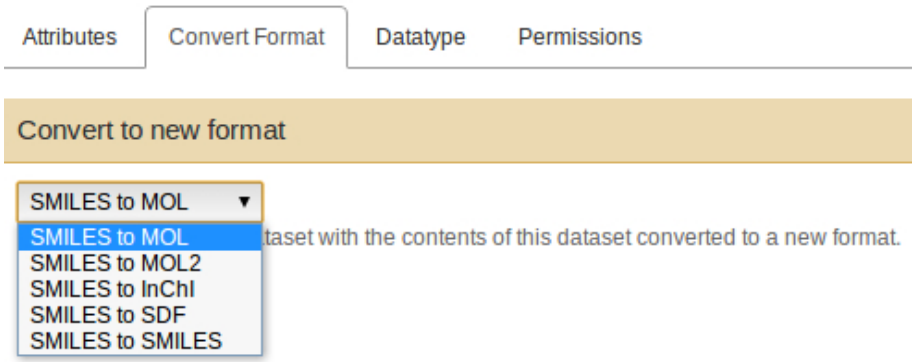

Abbildung 18: Semiautomatische Konvertierung von Molekülformaten - Neben dem automatischen Konvertieren von Formaten, bietet Galaxy auch eine semiautomatische Möglichkeit an. Diese ist von jedem Datensatz aus der History zu erreichen und listet alle definierten Konvertierungsoptionen auf. Eine SMILES-Datei kann in MOL, MOL2, InChI, SDF und SMILES konvertiert werden. Letzteres ist sinnvoll, um sicherzustellen, dass alle SMILES mit dem selben Algorithmus erstellt wurden.

Zusammengefasst kann die ChemicalToolBox mit allen Standardformaten der Cheminformatik umgehen und beherrscht das Konvertieren der verschiedenen Formate. Sollte der Benutzer keines dieser Formate besitzen, sondern nur Abbildungen von Strukturen, offeriert die ChemicalToolBox auch fortgeschrittene Transformationstools. Diese können zum Beispiel IUPAC-Namen (108) oder Abbildungen in chemische Standardformate überführen.

Filter

Die ChemicalToolBox kann mehrere Dutzend Millionen Moleküle in kürzester Zeit filtern. Dabei werden meist mehrere Tools in Kombination verwendet. Ein Beispiel ist der in Abbildung 19 gezeigte Workflow zur Filterung nach *Drug-likeness* unter der Verwendung des QED-Scores (103). Nach der Berechnung einer Eigenschaft, hier der QED Wert, wird das Resultat anhand des berechneten Wertes gefiltert. Exemplarisch neben einer Reihe

weiterer Filtertools sei an dieser Stelle ein Tool zur Filterung nach physikochemischen Eigenschaften näher beschrieben (siehe Abbildung 20).

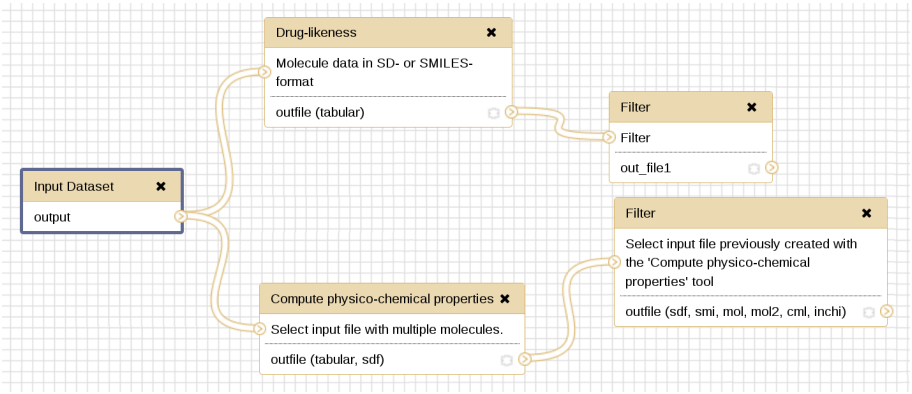

Abbildung 19: Filtern von chemischen Bibliotheken - Berechnung und Filtern von physikochemischen Eigenschaften sowie der *Drug-likeness* einer Kleinstrukturbibliothek.

Mit einer Molekülbibliothek als Eingabe, filtert das Tool alle Moleküle heraus, die bestimmten Kriterien entsprechen. Diese können vom Anwender spezifiziert werden oder es kann aus vordefinierten Filterregeln gewählt werden. Folgende stehen zur Verfügung:

- Lipinski's Rule of Five (109)
- Lead Like properties (110)
- Drug Like properties (91)
- Fragment Like properties (111)
- User-defined properties

Diese sind in der Hilfe näher beschrieben und ermöglichen das einfache Filtern nach sehr häufig wiederkehrenden Anforderungen. Die benötigten physikochemischen Eigenschaften werden standardmäßig für jedes Molekül berechnet und gegen die Filterregeln evaluiert. Möchte man eine Bibliothek jedoch mehrfach filtern oder Regeln sukzessive erweitern oder anpassen, bietet es sich an, zuvor alle Eigenschaften mit einem weiteren Tool zu berechnen und als Metadaten in eine SD-Datei zu speichern. Im nachfolgenden Filterschritt werden die Eigenschaften nicht mehr berechnet, sondern nur gelesen und evaluiert. Dies bedeutet vor allem bei großen Bibliotheken oder mehreren Filterschritten einen großen Geschwindigkeitsvorteil.

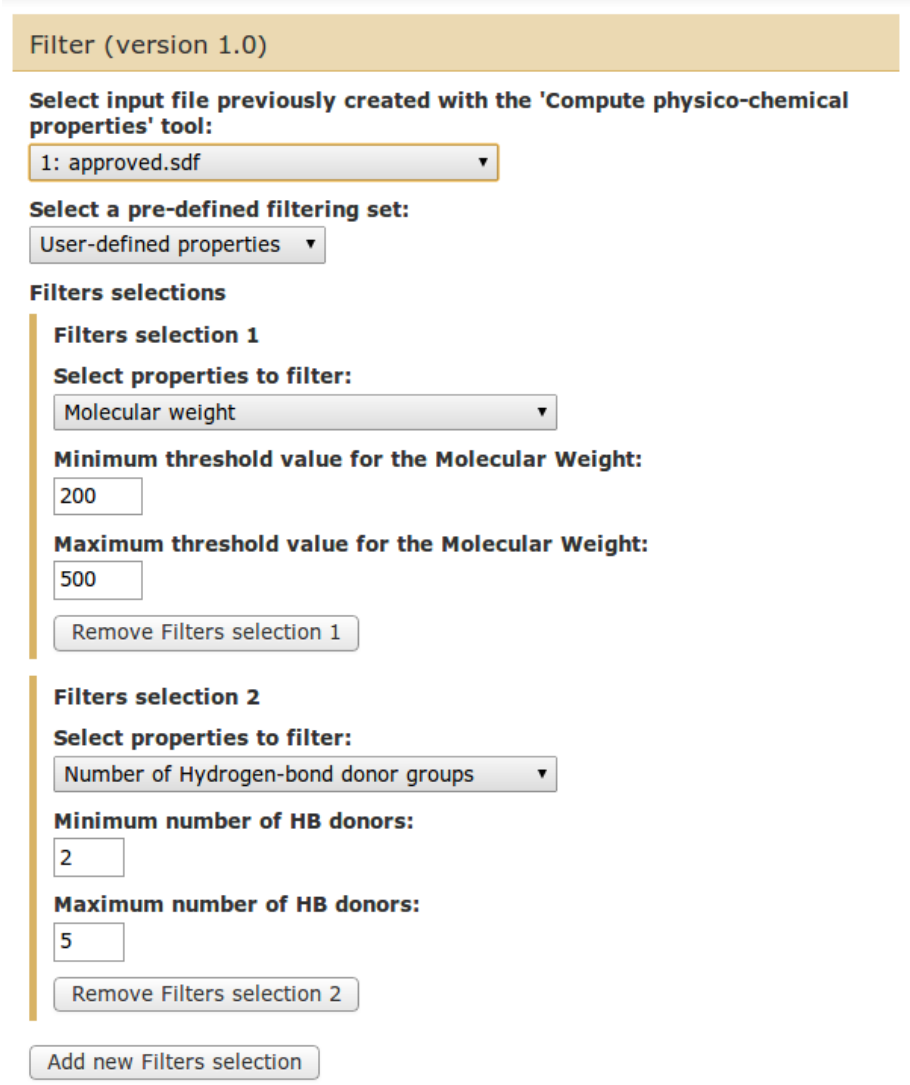

Abbildung 20: Filter-Tool mit verschiedenen benutzerdefinierten Regeln - Das Filter-Tool der ChemicalToolBox kann beliebige chemische Bibliotheken nach physikochemischen Eigenschaften filtern. Benutzerdefinierte Filterkriterien können beliebig kombiniert werden, z.B. das Filtern nach dem Molekulargewicht zwischen 200 und 300 Da und der Anzahl der Wasserstoffbrückendonatoren zwischen 2 und 5.

Neben dem Filtern nach Moleküleigenschaften bietet die ChemicalToolBox auch das Filtern nach Substrukturen oder speziellen Mustern in Molekülen an. Diese Muster können als SMARTS (SMiles ARbitray Target Specification)[1] definiert werden und ermöglichen komplexe strukturbezogene Filterkriterien. Werden mehrere SMARTS oder SMILES, die eine Untermenge der SMARTS darstellen, als Filterkriterium einer Bibliothek herangezogen, werden diese parallel gesucht und die Ergebnisse zu einer Datei zusammengefügt.

Ähnlichkeitssuchen

Ähnlichkeitssuchen sind schon seit Jahren essentieller Bestandteil des *in silico* Drugdesign (112). Dabei werden klassischerweise Moleküle in einen Bitstring überführt, dem Fingerprint. Diese lassen sich anschließend effizienter miteinander vergleichen als komplette Molekülgraphen, erreichen dies jedoch zu Lasten der Genauigkeit. Es gibt unterschiedliche Arten von Fingerprints, je nachdem, wie sie die strukturellen Information eines Moleküls kodieren. Die Wahl des richtigen Fingerprints für eine spezielle Molekülbibliothek kann entscheidend für die Ähnlichkeitssuche sein. Die ChemicalToolBox bietet zu diesem Zweck 10 verschiedene Fingerprints aus zwei verschiedenen Toolkits (97, 113) an:

- Open Babel FP2 fingerprints
- Open Babel FP3 fingerprints
- Open Babel FP4 fingerprints
- Open Babel MACCS fingerprints
- RDKit topological fingerprint
- RDKit topological Torsion fingerprints
- RDKit Morgan fingerprints
- RDKit Atom Pair fingerprints
- RDKit MACCS fingerprints
- RDKit substructure fingerprints

[1] `http://www.daylight.com/dayhtml/doc/theory/theory.smarts.html`

Fingerprints können in Kombination mit anderen Tools, wie z.B. dem Clustering, leicht getestet und für eine bestimmte Aufgabenstellung evaluiert werden.

Die klassischen Fingerprints kodieren nur für die 2D-Struktur eines Moleküls, sind aber sehr schnell zu berechnen und können auf Millionen von Strukturen angewendet werden, um strukturell Ähnliche zu identifizieren. Ist jedoch das Ziel, neuartige Grundgerüste ausgehend von bekannten aktiven Substanzen zu erschließen, sogenannte *Scaffold Hopper*, sind Deskriptoren notwendig, die die 3D-Struktur eines Moleküls beschreiben (114, 115). Die ChemicalToolBox stellt hierzu Tools bereit, die ein Pharmakophor bzw. ein Spectrophores™ von gegebenen Strukturen berechnen können, um diese im nachfolgenden Schritt für die Identifizierung 3D-ähnlichen Molekülen in einer Datenbank zu verwenden.

Molekül-Modifikationen

Etliche Tools zur Strukturmodifikation wurden integriert. So ist es zum Beispiel möglich, den Protonierungszustand pH-abhängig zu ändern, Ionen oder kleine Fragmente aus Strukturen zu entfernen, Konformere zu berechnen oder doppelte Moleküle aus einer größeren Sammlung zu entfernen. Letzteres ist abhängig von einer Definition der Molekülidentität und kann der Aufgabenstellung entsprechend adaptiert werden.

Visualisierung

Die Visualisierung von Ergebnissen und Molekülbibliotheken war kein primäres Ziel der ersten ChemicalToolBox-Version. Unter anderem auch, weil zum Zeitpunkt des Entstehens der Arbeit ein neues Visualisierungsframework in Galaxy integriert wurde, auf dem auch die ChemicalToolBox aufsetzen sollte. In Zukunft wird die Darstellung von chemischen Strukturen und Ergebnissen einen zentralen Punkt in der Entwicklung der ChemicalToolBox einnehmen. Die Zusammenarbeit mit der BioJS (116) Community, spezialisiert auf interaktive Visualisierungen, erbrachte bereits erste Ergebnisse (117). Momentan können kleinere Bibliotheken oder einzelne Moleküle in Bilder konvertiert werden und die Ähnlichkeit aller Moleküle zueinander mit dem Plotten der Ähnlichkeitsmatrix, nach dem *Multi-Dimensional-Scaling*, visualisiert werden. Barplots und Scatterplots sind Teil von Galaxy und können zusätzlich genutzt werden.

Clustering

Das Gruppieren von ähnlichen Strukturen in sogenannte Cluster bietet sich zum Beispiel an, um die Diversität einer Molekülbibliothek zu bestimmen. Die ChemicalToolBox stellt zwei Clustermethoden zur Auswahl. Das NxN-Clustering-Tool vergleicht jede Struktur mit allen anderen und bietet sich daher nur für kleine Datensätze an. Für große Bibliotheken mit mehreren Millionen Strukturen wurde der von Taylor und Butina ersonnene Cluster-Algorithmus (118, 119) integriert.

Galaxy-Tools

Eine Vielzahl an zusätzlichen Tools aus der Galaxy Standardinstallation oder der Tool Shed können den Funktionsumfang der ChemicalToolBox noch erweitern. Zum Beispiel bieten sich die *Text manipulation* Tools, wie *unique*, *merge columns*, *compare* oder *add column*, an, SMILES- oder InChI-Dateien zu manipulieren.

2.1.2.2 ChemicalBox und PurchaseableBox

Teile des folgenden Abschnittes wurden in einem peer-reviewed Journal veröffentlicht.

Xavier Lucas, Björn A Grüning, Stefan Bleher, und Stefan Günther. The Purchasable Chemical Space: A Detailed Picture. *Journal of chemical information and modeling*, April 2015. ISSN 1549-960X. doi: 10.1021/acs.jcim.5b00116

Die Analyse von medizinisch wirksamen Strukturen und kommerziellen Kleinstrukturbibliotheken, welche in *High-throughput-Screenings* Verwendung finden, deutet darauf hin, dass der medizinisch relevante chemische Raum weitaus größer ist als bisher angenommen (120). López-Vallejo *et al.* argumentieren, dass durch die Hinzunahme von kombinatorischen Kleinstrukturbibliotheken und Naturstoffen der traditionelle medizinisch relevante chemische Raum erweitert werden sollte, um neue Leitstrukturen mit neuem cheminformatischen Profil zu erschließen.

Diese Motivation als Grundlage nehmend, wurden die Molekülbibliotheken ChemicalBox und die PurchaseableBox erstellt. Die ChemicalBox vereint PubChem, ZINC, ChEMBL und die DrugBank in eine einheitlich prozessierte Datenbank aus über 81 Millionen frei zugänglichen Substanzen. Die PurchaseableBox ist eine Sammlung von 115 Bibliotheken, deren über 68 Millionen Substanzen käuflich erwerbbar und damit auch

synthetisierbar sind. Sie ist aber leider nicht frei zugänglich, da einige Anbieter ihre Bibliothek nur unter Restriktionen zur Verfügung stellten. Die ChemicalBox ist hingegen frei und kann nicht nur heruntergeladen werden[1], sondern steht auch als Galaxy-Workflow[2] zur Verfügung.

Beide Bibliotheken wurden mithilfe der ChemicalToolBox entwickelt und können daher leicht reproduziert werden[3]. Der in Abbildung 21 dargestellte Workflow erstellt die ChemicalBox auf dem Freiburger Galaxy-Server, mithilfe eines Computerclusters, in einem Tag. Diese hohe Geschwindigkeit ist der Fähigkeit der ChemicalToolBox geschuldet, viele Teilschritte parallel auszuführen und die gegebene Computerinfrastruktur optimal auszunutzen.

2.1.3 Diskussion

Das ChemicalToolBox-Projekt hat es geschafft, *High-Performance-Computing* (HPC) in die Cheminformatik zu integrieren. Ein Schritt, der aufgrund der enormen Datenmengen in der Biologie schon zum Standard gehört, hat mit dieser Arbeit nun auch in der Chemie und Pharmazie Einzug gehalten. Bisher war es nur mit hohem finanziellem und personellem Aufwand möglich, Millionen von Kleinstrukturen cheminformatisch zu charakterisieren und zu untersuchen. Mit der ChemicalBox und der PurchaseableBox konnte gezeigt werden, dass es möglich ist, 100 Millionen Substanzen zu prozessieren und in pharmazeutisch relevante Klassen einzuteilen.

Pipeline Pilot, ein kommerzielles Produkt der Firma Accelrys, und KNIME (121) waren laut Wendy A. Warr 2012 Marktführer der Workflowmanagment-Systeme in der Cheminformatik (122). *Pipeline Pilot* ist aufgrund seiner proprietären Lizenz keine Alternative für ein reproduzierbares und transparentes wissenschaftliches Arbeiten. Die KNIME-Plattform ist hingegen Open Source, hat jedoch keine frei verfügbare HPC-Integration und ist nicht für multi-Benutzer-Szenarien entwickelt worden. KNIME hat eine große Auswahl an *components* (vergleichbar mit Galaxy-Tools), die es zu einer sehr vielfältigen Plattform in der Cheminformatik macht.

ChemicalToolBox ist ein Open Source-Projekt, basierend auf Galaxy, komplett frei und von der Community entwickelt. Jeder kann die ChemicalToolBox erweitern, Ideen

[1] `ftp://pharmaceutical-bioinformatics.org/chemicalbox/`
[2] `http://bit.ly/1KYGDB1`
[3] Eine Liste aller Anbieter ist in Tabelle S1 des PurchaseableBox Manuskriptes zu finden (24).

einbringen und die Zukunft des Projektes beeinflussen. Einige Ideen und Wünsche, inklusive besserer Visualisierungsmöglichkeiten mittels ChemVis und BioJS, sind im Projekt Bugtracker[1] aufgelistet und geben einen Ausblick auf die Zukunft des Projektes.

Die ChemicalToolBox kann auf jedem Standardcomputer installiert werden, ist aber bei Bedarf skalierbar und kann mehreren 1000 Wissenschaftlern gleichzeitig zur Verfügung gestellt werden. *Cloud Computing* von großen kommerziellen Anbietern, wie die *Amazon Elastic Compute Cloud*, werden ebenso von der ChemicalToolBox unterstützt wie nationale und internationale Wissenschaftsclouds.

Ein nicht zu vernachlässigender Vorteil der ChemicalToolBox ist die nahtlose Integration in das Galaxy-Framework und der damit einhergehende Zugang zu anderen Wissenschaftsfeldern. Cheminformatik verbunden mit *Image Analysis*, der Genomik bzw. Transkriptomik lassen viele neue interessante Analysen und Ergebnisse erwarten, die zuvor so in einer freien, skalierbaren Plattform nicht möglich waren.

Das von Frank K. Brown definierte Ziel, Informationen in Wissen zu transferieren, kann durch die Integration verschiedener Informationsressourcen und die einfache Möglichkeit der Informationsaufarbeitung und Analyse mit der ChemicalToolBox erreicht werden. Mit der Integration von Primärdaten, dem sich das nächste Kapitel (2.2) widmet, und den erhobenen Sekundärdaten aus Kapitel 2.3, steht mit der ChemicalToolBox ein umfassendes cheminformatisches Analysesystem für die pharmazeutische Forschung zur Verfügung.

Die ChemicalToolBox kann über die Galaxy Tool Shed[2] installiert werden. Entwickelt wird das Projekt auf GitHub unter `https://github.com/bgruening/galaxytools`.

[1] `https://trello.com/b/t9Wr8lSY`
[2] `https://toolshed.g2.bx.psu.edu/view/bgruening/chemicaltoolbox`

Abbildung 21: Workflow zur Erstellung der ChemicalBox - Kleinstrukturen werden aus verschiedenen Datenquellen heruntergeladen und mit dem Programm "Compound convert" in canonische SMILES konvertiert. Zwecks Rückverfolgung einzelner Moleküle zu den Ursprungsdatenbanken wird ein Identifikator an jede Moleküldatei angehängt und im Anschluss werden alle Dateien konkateniert. Die nachfolgenden Schritte bereinigen die erstellte Molekülbibiothek und entfernen zum Beispiel Gegenionen und alle redundanten Strukturen.

2.2 Kleinmoleküle in wissenschaftlichen Abhandlungen

Teile des folgenden Abschnittes wurden in einem peer-reviewed Journal veröffentlicht.

Björn A Grüning, Christian Senger, Anika Erxleben, Stephan Flemming, und Stefan Günther. Compounds In Literature (CIL): screening for compounds and relatives in PubMed. *Bioinformatics (Oxford, England)*, 27(9):1341–2, May 2011. ISSN 1367-4811. doi: 10.1093/bioinformatics/btr130

CIL steht als Webservice unter `http://cil.pharmaceutical-bioinformatics.de` zur Verfügung.

Die Suche nach Kleinstrukturen in wissenschaftlichen Abhandlungen ist in vielerlei Hinsicht ein nicht triviales Problem:

1. Eindeutige Bezeichnungen eines Moleküls wie der IUPAC-Name oder der für die Websuche entwickelte InChI-Key werden selten in wissenschaftlichen Abhandlungen genutzt und sind daher auch nicht mit einer Suche auffindbar.

2. Für Suchen werden meist Trivialnamen verwendet. Kleinstrukturen haben aber in aller Regel keinen eindeutigen Namen, sondern weisen eine Reihe an Synonymen oder Identifikationsnummern auf. Von den verwendeten PubChem-Molekülen hatten 6,5 % aller Strukturen mehr als vier Synonyme und im Durchschnitt hatte jedes Molekül 1,7 verschiedene Namen. Für eine ausführliche Literaturrecherche müssten alle Synonyme gleichermaßen gesucht werden. Dies ist jedoch nicht trivial und setzt voraus, dass dem Wissenschaftler alle Synonyme bekannt sind.

3. Sollten keine Synonyme bekannt sein, sondern nur die Struktur des Moleküls, ist eine anfängliche Struktursuche notwendig, um die Synonyme zu identifizieren.

4. Neu entdeckte Kleinstrukturen, wie z.B. Metabolite aus Streptomyceten (siehe Kapitel **??**), wurden mit hoher Wahrscheinlichkeit noch nicht näher beschrieben. Suchanfragen mit Namen oder Struktur würden in diesem Fall keine Ergebnisse bringen. Informationen von strukturell ähnlichen Molekülen zu studieren und aufbauend auf diesen Erkenntnissen Schlussfolgerungen für das zu untersuchende Molekül zu ziehen, ist vielversprechend, aber sehr zeitaufwendig und rechenintensiv.

5. Schlagwörter oder MeSH-Terme[1] sind in vielen Fällen nicht vorhanden oder unzureichend annotiert. "Versteckte" Informationen, die nicht im Hauptfokus der jeweiligen Publikation lagen, werden nicht erfasst und werden daher in einer MeSH- und Schlagwortsuche nicht gefunden.

Compounds in Literature (123), kurz CIL, ist ein Webservice, der diese Probleme adressiert und es ermöglicht, wissenschaftliche Abhandlungen aus der Datenbank PubMed[2] nach Kleinstrukturen zu durchsuchen und darüber hinaus potentiell interagierende Proteine zu identifizieren. Das Projekt wurde mit Dr. Christian Senger bearbeitet und 2011 im Journal *Bioinformatics* publiziert. Es ist unter `http://cil.pharmaceutical-bioinformatics.de` zu erreichen.

Das CIL-Projekt besteht aus zwei Kernbereichen: Der erste annotiert Artikel aus der PubMed mit Kleinstrukturen aus der PubChem und Proteinen aus der UniProt. Damit werden bereits existierende Systeme wie MeSH[3] oder das durch Autoren spezifizierte Schlagwortverzeichnis erweitert. Der zweite Bereich stellt sicher, dass die neu gewonnenen Informationen dem Benutzer übersichtlich, schnell und flexibel präsentiert werden können. Zu diesem Zweck wurde eine SOAP-Schnittstelle entwickelt, die auch in Galaxy integriert wurde, und es ermöglicht, mit Synonymen oder Strukturen Listen von relevanten PubMed-Artikeln zu erhalten. Das Hauptaugenmerk lag auf einem intuitiven Webauftritt mit einer übersichtlichen und kompakten Darstellung der Suchergebnisse.

CIL ist in der Lage, mit einem Namen oder der Struktur eines Moleküls nahezu alle Artikel der PubMed zu identifizieren, in denen es erwähnt wird. Dabei wird nicht nur das Suchmolekül gesucht und in den Ergebnissen angezeigt, sondern auch strukturell verwandte Moleküle. Dies ermöglicht auch dann eine Abschätzung der Funktion und Interaktionen des Suchmoleküls, wenn zu diesem bisher nur sehr wenig oder nichts bekannt ist. Durch die Integration und Identifikation von Proteinen in Abhandlungen kann das gemeinsame Auftreten von Kleinstruktur und Protein gezählt werden und als Indiz für eine Interaktionsbeschreibung in jenem Artikel gelten.

In den folgenden Kapiteln werden zugrunde liegende Daten, Algorithmen und Techniken eingehender beschrieben und auf die übersichtliche Heatmap-Darstellung der Suchergebnisse näher eingegangen.

[1] `http://www.nlm.nih.gov/mesh`
[2] `http://www.ncbi.nlm.nih.gov/pubmed`
[3] `http://www.nlm.nih.gov/pubs/factsheets/mesh.html`

2.2.1 Daten und Methoden

CIL verknüpft drei große Entitäten: Literatur (PubMed), Kleinstrukturen (PubChem) und Proteine (UniProt). Um die Informationen aus PubMed zu extrahieren und für einen Service wie CIL zu Verfügung zu stellen, wurden alle Artikelbeschreibungen, *citations* genannt, heruntergeladen. Mit einem selbst entwickeltem Parser wurden alle Informationen in eine relationale Datenbank übertragen. Verwendet wurde die Open Source-Datenbank PostgreSQL (siehe dazu auch Kapitel 1.2.1.1). Indizes für eine schnelle Suche wurden nach den Bedürfnissen der CIL-SQL-Abfragen hinzugefügt bzw. erweitert. Insgesamt wurden 18 Millionen *citations* und 10 Millionen Artikel, von 1975 bis 2010, in die Datenbank integriert. Zusätzlich wurden alle Abstracts, Publikationstitel, Schlagwörter und MeSH-Terme in einen Xapian-Volltextindex gespeichert.

Für die Ähnlichkeitssuche mit Kleinstrukturen und deren Annotation in wissenschaftlichen Texten wurden alle 28 Millionen PubChem-Strukturen und deren 55 Millionen Synonyme ebenfalls in die relationale Datenbank PostgreSQL geladen. PostgreSQL wurde hierfür mit dem cheminformatischen Modul pgchem[1] erweitert. Pgchem stellt Datenstrukturen und Funktionen zur Bearbeitung und Suche von chemischen Molekülen bereit.

Um die Ergebnisse einer CIL-Suche möglichst schnell und effizient zu gestalten, wurden die Relationen aller Kleinstrukturen vorberechnet und in einer Datenbank abgelegt. Um dem Problem einer automatischen Annotation wie in Kapitel **??** beschrieben zu entgegnen, wurden die Kleinstruktursynonyme vor ihrer Verwendung gefiltert und modifiziert. Als erstes wurden nur die besten fünf Synonyme einer jeden Kleinstruktur verwendet. Hierzu wurde das Synonymranking von PubChem verwendet. Dieses sortiert die Synonyme absteigend nach ihrer Relevanz. Als zweiten Schritt wurden alle Synonyme nach den Regeln von Hettne *et al.* modifiziert, um sie im Anschluss gegen eine sogenannte Stoppwortliste zu vergleichen und zu filtern (80). Diese Liste wurde iterativ durch Begutachtung der Suchergebnisse manuell erstellt und ist auf der CIL-Webseite[2] publiziert. Sie beinhaltet 1162 Wörter, welche nicht nur Kleinstruktursynonyme repräsentieren, sondern auch in normalen Texten vorkommen können oder anderweitig zu falsch-positiven Ergebnissen führen können. Die Möglichkeit, mit einer benutzerspezifischen Synonymliste zu suchen und gänzlich auf die vorberechneten Werte zu verzichten, ist mit dem Volltextindex aber zu jeder Zeit gegeben und wird dem Benutzer auch offeriert.

[1] `https://github.com/ergo70/pgchem_tigress`
[2] `http://cil.pharmazeutische-bioinformatik.de`

Für die Identifikation der Proteine in den Abstracts wurde der Webservice Whatizit verwendet (79). Die Ergebnisse wurden, wie auch zuvor bei den Kleinstrukturen beschrieben, gespeichert und mit einer Stoppwortliste für Proteine mit 160 Wörtern gefiltert, um die Spezifität der Suchergebnisse zu erhöhen.

2.2.2 Ergebnisse

Aufbauend auf der CIL-Datenbank, wurde eine auf dem Django-Framework[1] basierende Webseite erstellt, die den in Abbildung 22 gezeigten Arbeitsablauf umsetzt. Die Suche in CIL kann dabei mit einem SMILES, InChI, einem beliebigen Synonym oder einer gezeichneten Struktur begonnen werden (Abbildung 22a). Der Benutzer kann im Anschluss zwischen zwei Such-Modi wählen. Standardmäßig wird auf den vorberechneten Relationen von Struktur, Protein und Abstract gesucht. Hinzu kommen ausgewählte Standardsuchparameter, wie der Tanimotokoeffizient für die Ähnlichkeitssuche, die so gewählt wurden, dass eine Suche qualitativ hochwertige Ergebnisse bei einer geringen Suchzeit liefert. Dem Benutzer sollte jedoch auch die Freiheit eingeräumt werden, alle Parameter nach eigenen Wünschen anzupassen.

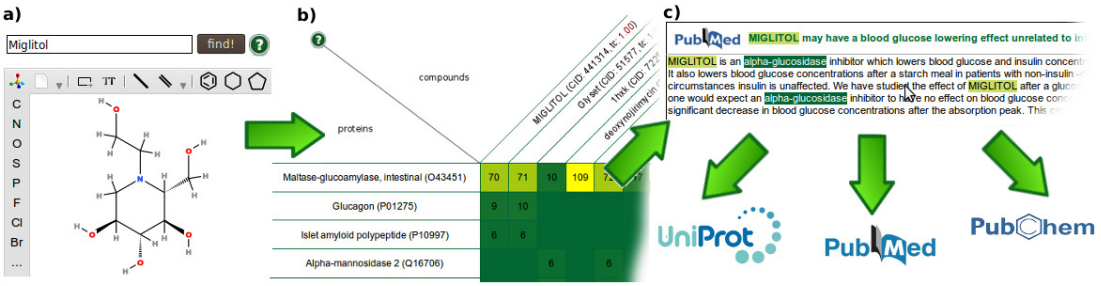

Abbildung 22: CIL-Workflow - a) Suche nach Molekülen mittels Namen, SMILES, InChI oder gezeichneter Struktur; b) Heatmap Übersicht: ähnliche Kleinstrukturen in den Spalten, Anzahl der Publikationen in den Zellen und identifizierte Proteine in den Zeilen; c) Abstract mit hervorgehobenen, identifizierten Strukturen und Proteinen. Verweise auf externe Quellen wie PubMed, PubChem und UniProt, sowie Hilfe für die Benutzerführung sind gegeben.

Das Resultat dieser Bemühungen ist ein spezieller Modus, bei dem Parameter wie Ähnlichkeitskoeffizient, Organismus und die maximale sowie minimale Anzahl an Relationen veränderbar sind. Dies betrifft insbesondere die zur Suche benutzten Synonyme. Die von CIL benutzten Standardsynonyme können ergänzt oder limitiert werden. In diesem

[1] `https://www.djangoproject.com/`

Fall kann nicht mehr auf die vorberechneten Relationen zurückgegriffen werden, sondern es wird direkt auf dem Volltextindex gesucht. Die Suche dauert in aller Regel länger als die Standardsuche, bietet aber ein Höchstmaß an Flexibilität. Die Suchzeit hängt zusätzlich in einem hohem Maße von dem Tanimotokoeffizienten, dies bedeutet die Anzahl der zu suchenden ähnlichen Strukturen, und dem Rauschfilter (max/min Anzahl an Relationen) ab. Letzteres limitiert die Suchergebnisse in der Hinsicht, dass die Mindestanzahl (bzw. Maximalanzahl) von Struktur-Abstract-Protein-Relationen festgelegt werden können. Dieser Parameter trägt der Tatsache Rechnung, dass Buchstabengruppen, wie Synonyme, mit steigender Textgröße alleine durch Zufall gefunden werden können. Schlechte Synonyme, die nicht nur in einem strukturellem Kontext gefunden werden, wie z.B. *AIR* (PubChem CID: 161.500), können mit dem Parameter für die maximale Anzahl an Hits kontrolliert werden und verhindern falsch positive Ergebnisse.

Ist die Suche abgeschlossen, wird dem Benutzer eine farbkodierte Tabelle mit allen gefundenen Relationen zur Suchstruktur und strukturell Verwandten präsentiert (Abbildung 22b). In den Spalten befinden sich alle zur Suchstruktur ähnlichen Moleküle, und im Spaltenkopf die jeweiligen Synonyme, Tanimotokoeffizienten und Abbildungen, geordnet absteigend nach Ähnlichkeit zur Suchstruktur. In den Tabellenzeilen befinden sich Proteine, die zusammen mit den Strukturen in einem Abstract annotiert wurden. Der Zeilenkopf enthält zusätzlich alle Synonyme zum Protein und einen Verweis zum Eintrag in der UniProt-Datenbank. Die Anzahl an Abstracts mit dem jeweiligem annotierten Struktur-Proteinpaar ist als Zahl und als Farbgradient in den Tabellenzellen abgelegt. Die Zeilensumme bestimmt das Ranking der einzelnen Proteine. Wird das Protein im hohen Maße zusammen mit den zur Suchstruktur ähnlichen Molekülen gefunden, wird es weiter oben in der Tabelle angezeigt und ist ein stärkeres Indiz für eine Interaktion zwischen der jeweiligen Struktur und Protein.

Die Tabellenzellen sind verlinkt zu einer Übersicht aller assoziierten Abstracts. In diesen sind die identifizierten Proteine und Strukturen farblich hervorgehoben und mit einem Verweis auf PubChem, respektive UniProt versehen (Abbildung 22c). Des Weiteren ist jeder Abstract auf den Originaleintrag in PubChem referenziert.

2.2.2.1 Beziehung zwischen Abstracts, Protein- und Kleinstruktursynonymen

Die CIL-Datenbank enthält mehr als 133 Millionen Protein- und mehr als 685 Millionen Kleinstrukturverweise auf wissenschaftlichen Abhandlungen in PubMed. Die lokal gespeicherte Version der PubMed hat mehr als 11 Millionen Abstracts, zu denen im Mittel 1,4 Kleinstrukturen und 0,5 Proteine annotiert werden konnten (siehe Tabelle 2.2). Die lokale Version der PubChem beinhaltet über 35 Millionen Strukturen und über 606 Millionen Synonyme. Jede Struktur kam dabei im Mittel in vier Abstracts vor (siehe Tabelle 2.3). Die Datenbank der Proteine ist erwartungsgemäß kleiner und beinhaltet knapp 530.000 Sequenzen mit 2 Millionen Synonymen (siehe Tabelle 2.4).

Tabelle 2.2: PubMed-Statistik der CIL-Datenbank (Stand: Juni 2015).

Anzahl der Artikel:	20.573.755
Anzahl der Abstracts:	11.655.448
Anzahl der Kleinstrukturen pro Artikel	
min	0
mean	1,4
max	867
Anzahl der humanen Proteine pro Abstract	
min	0
mean	0,5
max	89

Tabelle 2.3: PubChem-Statistik der CIL-Datenbank (Stand: Juni 2015).

Anzahl der Kleinstrukturen:	35.354.203
Anzahl der gefundenen Kleinstrukturen in Abstracts:	92.851
Anzahl der Synonyme aller Kleinstrukturen:	606.303.734
Anzahl der Abstracts pro Kleinstruktur	
min	1
mean	4
max	224.153
Anzahl der Synonyme pro Kleinstruktur	
min	1
mean	1,7
max	777

Tabelle 2.4: UniProt-Statistik der CIL-Datenbank (Stand: Juni 2015).

Anzahl der Proteine:	529.056
Anzahl der Proteinsynonyme:	2.015.012
Anzahl der Abstracts pro Protein	
min	1
mean	49
max	194.160
Anzahl der Synonyme pro Protein	
min	3
mean	6
max	132

2.2.2.2 Interaktion zwischen CIL und Prolific

Bei dem Design des Benutzerinterfaces wurden auch zwei Szenarien berücksichtigt, die sowohl eine Interaktion mit dem in Kapitel 1.2 beschriebenen Webservice Prolific ermöglichen, als auch einen iterativen Suchansatz in CIL.

Identifizierte potentielle Interaktionsproteine können so zum Beispiel direkt als Ausgangspunkt für eine Suche in Prolific dienen. Strukturell ähnliche Wirkstoffe können wiederum in eine CIL-Suche gespeist werden, ohne sich durch die Eingabemasken der initialen Suche zu klicken. Entsprechende Hyperlinks wurden neben den Kleinstrukturen, respektive den Proteinen, platziert.

Es konnte gezeigt werden, dass die CIL-Suche mit den modifizierten und gefilterten Synonymen eine bessere Qualität aufweist als die von Hettne *et al.* (80). Es wurde auf dem gleichen Corpus[1] (124) eine *precision* von 52 %, ein *recall* von 72 % und ein *F-score* von 60 % erreicht.

2.2.3 Diskussion

CIL kombiniert die drei großen Datenbanken PubMed, PubChem und UniProt und ermöglicht es, sonst sehr aufwendige und umständliche Suchen schnell und effizient durchzuführen. Abschätzungen der biologischen Funktion einer unbekannten Struktur sind damit ebenso umsetzbar, wie das Suchen nach potentiellen Cross-Targets, sei es für das *Drug-Repositioning* oder die Aufklärung von Nebenwirkungen. Beginnend mit einem Kleinstruk-

[1] `http://www.scai.fraunhofer.de/chemcorpora.html`

turnamen, einem SMILES, InChI oder einer gezeichneten Struktur, wird die Suchstruktur sowie auch zu dieser ähnliche Strukturen in Abstracts identifiziert und mit Proteinen, die im gleichen Kontext erwähnt werden, in Beziehung gesetzt. Die zu Grunde liegenden Datenbanken und verwendeten Techniken bilden darüber hinaus die Grundlage für Prolific (siehe Kapitel 1.2) und die StreptomeDB (siehe Kapitel 2.3).

Einige ähnliche Ansätze zum Auffinden von Proteinen (79, 125), Kleinstrukturen (80) oder einer Kombination aus Beidem (126, 127, 128) wurden schon früher veröffentlicht. Diese bieten jedoch keine intuitive und kompakte Darstellung der Suchergebnisse, weswegen ihre Zugänglichkeit sehr eingeschränkt ist. Der *WENDI* Service hat zudem Probleme mit der Verfügbarkeit.

2.3 StreptomeDB: Eine Datenbank von Naturstoffen aus Streptomyceten

Teile des folgenden Abschnittes wurden im Journal *Nucleic Acids Research* veröffentlicht.

Xavier Lucas, Christian Senger, Anika Erxleben, Björn A Grüning, Kersten Döring, Johannes Mosch, Stephan Flemming, und Stefan Günther. StreptomeDB: a resource for natural compounds isolated from Streptomyces species. *Nucleic acids research*, 41 (Database issue):D1130–6, January 2013. ISSN 1362-4962. doi: 10.1093/nar/gks1253

StreptomeDB steht unter http://streptomedb.pharmaceutical-bioinformatics.de zur Verfügung.

Die bedeutende Rolle der Gattung der *Streptomyces* bei der Produktion von natürlich vorkommenden bioaktiven Kleinstrukturen wurde bereits in Kapitel **??** besprochen, ebenso die Relevanz der entsprechenden Gencluster bei der Synthese dieser Kleinstrukturen. Streptomyceten sind ubiquitäre Bakterien, relativ leicht zu züchten und sind seit Jahrzehnten Teil der Naturstoffforschung. Die Meta-Datenbank PubMed der nationalen medizinischen Bibliothek der Vereinigten Staaten listet 18.612 Artikel (Stand: Juli 2015) mit einem *Streptomyces*-assoziierten MeSH-Schlagwort. Der älteste Artikel mit dem Titel *Two Antibiotics Produced by a Streptomyces* ist von 1947 (130). Aber auch schon früher wurden Stämme wie *Streptomyces antibioticus*, *Streptomyces lavendulae* (131, 132) und *Streptomyces griseus* (133) beschrieben. Über die Jahrzehnte hat sich eine erhebliche Menge an unstrukturierter Primärliteratur gesammelt, die schwer zu sichten ist.

Im Rahmen der vorliegenden Arbeit wurde das StreptomeDB-Projekt bearbeitet, welches das Ziel verfolgt, die Informationen der Streptomyceten-relevanten Primärliteratur zu extrahieren und strukturiert in einer Datenbank zugänglich zu machen. Diese kann dann als Ausgangspunkt für spezielle Suchanfragen oder als Datenquelle für *High-throughput*-Cheminformatik dienen, wie in Kapitel 2.1 beschrieben.

StreptomeDB ist eine strukturzentrierte Datenbank mit mehr als 2.300 unterschiedliche Kleinstrukturen aus mehr als 800 verschiedenen Streptomyceten-Stämmen. Zu jeder Struktur wurden, soweit möglich, zusätzliche Informationen gesammelt bzw. berechnet.

Das Suchinterface wurde gewollt einfach aber flexibel gehalten, um verschiedenste Fragestellungen beantworten zu können. Es kann nach produzierenden Streptomyceten ge-

sucht werden, um alle zu einem Organismus bekannten Strukturen zu identifizieren, oder nach einer biologischen Aktivität, wenn sich die Fragestellung auf Strukturen mit z.B. einer antitumoralen Wirkung beschränkt. Ist der Benutzer an Genclustern und Synthesewegen interessiert, ist es auch möglich, von diesen ausgehend eine Liste von allen Strukturen zu bekommen. Neben der Vielzahl an unterschiedlichen Suchoptionen bietet das Benutzerinterface der StreptomeDB auch Optionen für Ähnlichkeitssuchen und Substruktursuchen an. Die StreptomeDB enthält eine Reihe von Informationen, wie zum Beispiel Molekülstruktur in verschiedenen Formaten, Molekülname und Synonyme, physikochemische Eigenschaften, produzierender Organismus, Literaturreferenz, biologische Aktivität (z.B. antibiotisch, antitumoral) und Synthesewege (z.B. PKS, NRPS). Die StreptomeDB ist somit ein geeignetes Werkzeug zur Untersuchung von Sekundärmetaboliten und der Identifikation neuer therapeutisch relevanter Naturstoffe.

2.3.1 Daten und Methoden

Text Mining

Die Extraktion von Kleinstrukturinformationen aus wissenschaftlichen Abhandlungen wurde schon in Kapitel 2.2 beschrieben und diente als Grundlage der StreptomeDB. Zuerst wurde das Wort "streptomyces" in allen Abstracts, MeSH-Termen und Schlüsselwörtern gesucht, um diese im Anschluss nach auftretenden Kleinstrukturen zu filtern. Von den resultierenden Abstracts wurden ca. 8.400 manuell von der AG Günther gelesen und annotiert. Dabei wurden Informationen über Kleinstrukturen, produzierende Organismen, Synthesewege und die Aktivität der Struktur extrahiert und in einer Datenbank gesammelt. Kleinstrukturen wurden, wenn vorhanden, den zugehörigen PubChem-Einträgen zugeordnet oder manuell gezeichnet. Organismennamen wurden mithilfe der *NCBI Taxonomy* gegliedert und eingeordnet. Es wurde dabei vor allem darauf geachtet, *Strains* und Mutanten unter den jeweiligen Stammorganismus einzugliedern. Synthesewege wurden nach Schlagwörtern kategorisiert (siehe Tabelle 2.5), ebenso wie Aktivitäten, z.B. *antiaging*, *antibiotic* oder *suppressor*.

Tabelle 2.5: StreptomeDB: Anzahl der unterschiedlichen Synthesewege

NRPS	45
Shikimate-Polyketide	1
CDA-Gencluster	1
PKS	253
Terpene	26
hybrid PKS / NRPS	8
RPS	2
hybrid PKS / RPS	1
Shikimate	6

Daten-Integration

Mit der KNApSAcK-Datenbank (134), der Novel Antibiotics-Datenbank[1] und dem strukturierten Wörterbuch der Medical Subject Headings (MeSH) standen neben dem Text Mining drei frei zugängliche Datenquellen zur Verfügung, um den Datensatz der StreptomeDB zu erweitern bzw. den verwendeten Text Mining-Ansatz zu evaluieren. Extrahierten Kleinstruktur-Organismen-Pärchen wurden eindeutige Organismen- bzw. Strukturnamen zugeordnet und in die StreptomeDB integriert. Ein großes Problem waren hierbei die nicht eindeutigen Synonyme, Schreibfehler, nicht zurückverfolgbare Referenzen oder Kleinstruktursynonyme, denen keine eindeutige Struktur zugeordnet werden konnte. Insgesamt konnten in den MeSH-Beschreibungen 83 Kleinstruktur-Organismus-Relationen identifiziert werden. Aus der *Novel Antibiotics Datenbank*, welche 2.557 für diese Arbeit relevante Artikel aus dem *Journal of Antibiotics*[2] enthält, wurden 1.225 Relationen in die StreptomeDB übernommen. Zu guter Letzt konnte von den 1.988 annotierten Metaboliten aus Streptomyceten in der der Datenbank KNApSAcK 1.245 bestätigt und integriert werden.

Daten-Verwaltung und Präsentation

Zur Speicherung, Visualisierung und der Erstellung des Webauftritts wurden auf die in CIL (siehe Kapitel 2.2) entwickelten und erfolgreich angewendeten Techniken zurückgegriffen. Alle Daten wurden in einer PostgreSQL-Datenbank (Version 8.4), mit pgchem-Erweiterung, gespeichert. Open Babel (97) wurde zur Berechnung physikochemischer Eigenschaften und

[1] `http://www0.nih.go.jp/~jun/NADB/search.html`
[2] `http://www.antibiotics.or.jp/journal/ja-top.htm`

chemischer Abbildungen herangezogen und das Django-Framework diente als Grundlage der Webseite.

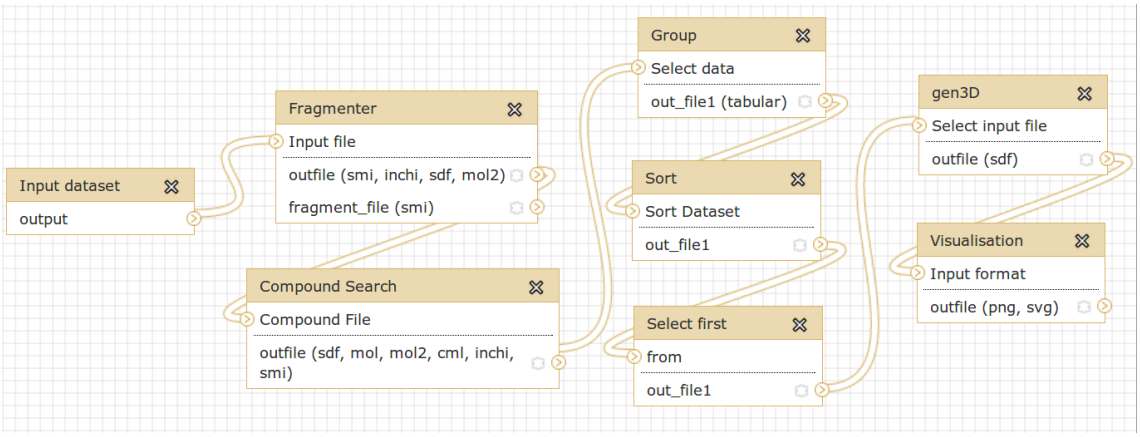

Abbildung 23: ChemicalToolBox-Workflow zur Identifikation von MCSS - ChemicalToolBox-Workflow zum Berechnen und Plotten der am häufigsten auftretenden Substrukturen. Nach dem Fragmentieren können die Strukturen mittels SMARTS-Pattern gefiltert werden. Im Anschluss werden identische Substrukturen zusammengefasst und ihr Vorkommen gezählt. Sortiert nach der Häufigkeit werden die ersten 120 Strukturen ausgewählt und ihre 3D-Struktur berechnet, um im finalen Schritt alle Strukturen zu plotten.

Most Common Substructure Search

Most Common Substructure Search, kurz MCSS, ermöglicht dem Benutzer, nach sehr häufig auftretenden Strukturbausteinen zu suchen. Hierzu wurden alle Strukturen der StreptomeDB nach den RECAP-Regeln (135) fragmentiert und nach der Häufigkeit ihres Auftretens sortiert. Der verwendete Workflow wurde mit Tools der ChemicalToolBox (siehe Kapitel 2.1) umgesetzt und ist in Abbildung 23 zu sehen. Die 120 häufigsten zyklischen Strukturelemente werden dem Benutzer als Miniaturbilder auf einer speziellen Suchseite angeboten und sind frei kombinierbar (siehe Abbildung 24). Dies ermöglicht komplexe Substruktursuchen mit mehreren Strukturelementen.

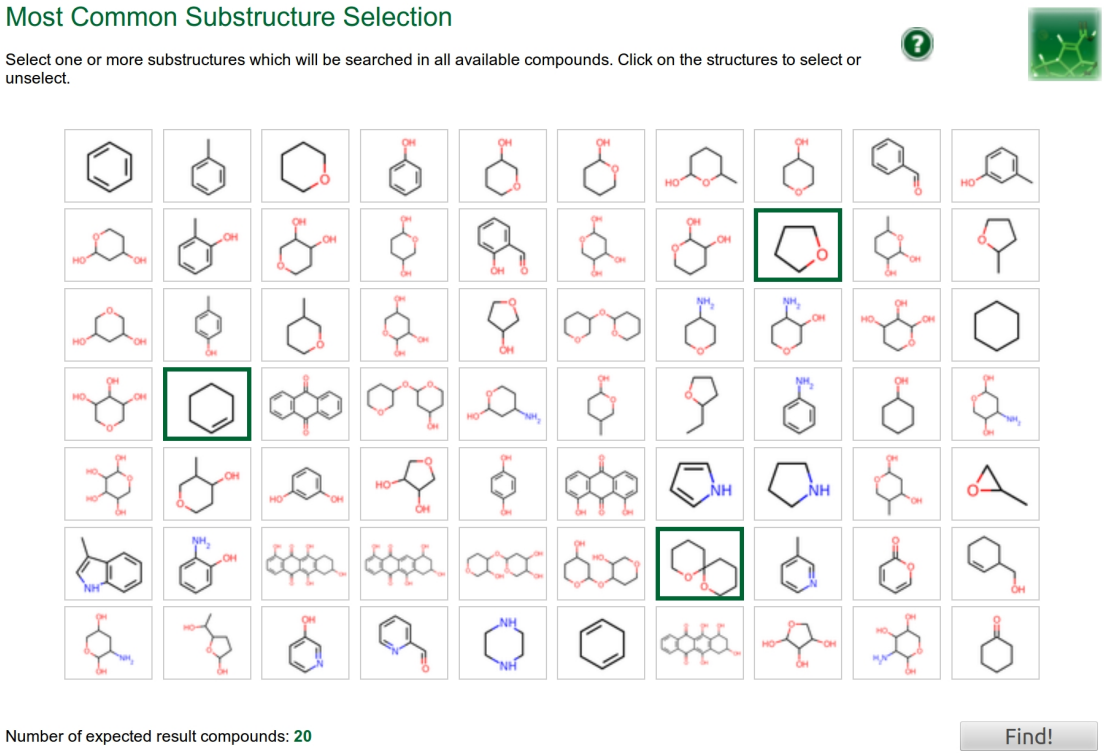

Abbildung 24: MCSS der StreptomeDB - Die am häufigsten auftretenden zyklischen Strukturelemente der StreptomeDB können frei kombinierbar gesucht werden.

2.3.2 Ergebnisse

Diversität der Kleinstrukturen

Die Datenbank besteht aus 2332 verschiedenen Sekundärmetaboliten, alle produziert von Organismen der Gattung *Streptomyces*. Tabelle 2.6 gibt eine Übersicht über den Inhalt der StreptomeDB. Eine kleine Auswahl an Strukturen ist in Abbildung 25 gegeben und gibt einen Eindruck über die Diversität der Strukturklassen innerhalb der StreptomeDB. Der hohe Anteil an komplexen Strukturen resultiert unter anderem in einem stark erhöhtem mittleren Molekulargewicht (Median: 454 g/mol), verglichen zu denen typischer Arzneistoffe (Median: 310 g/mol (136)).

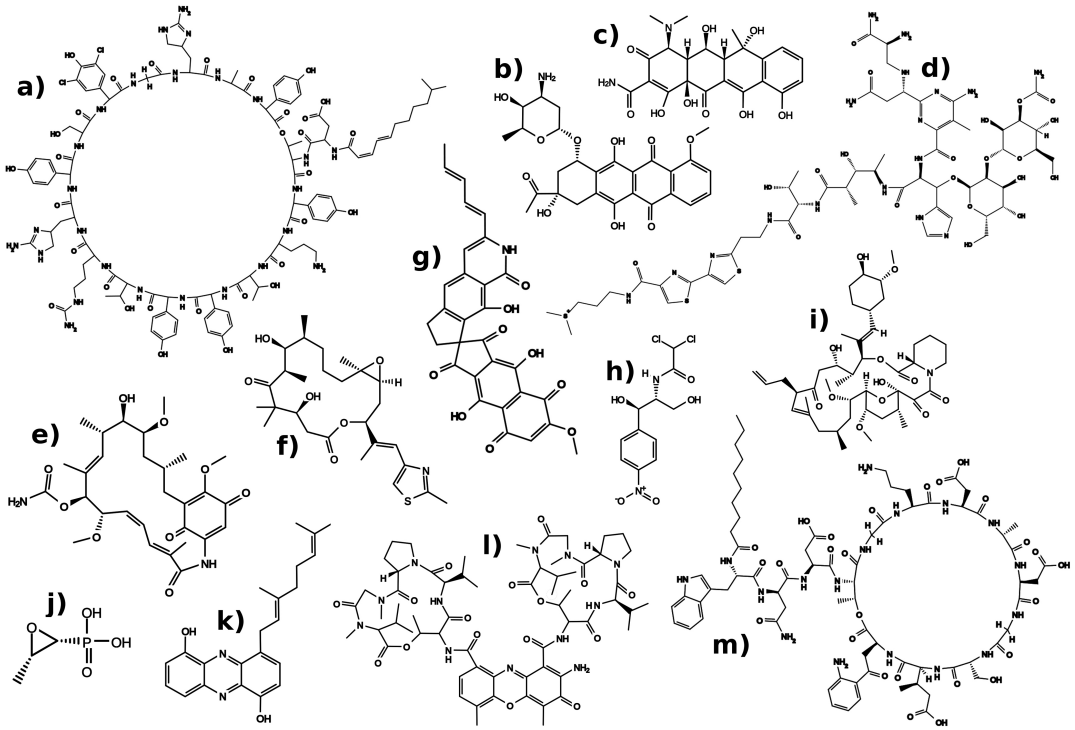

Abbildung 25: Exemplarische Naturstoffe in der StreptomeDB - (a) Enduracidin A, NRPS-abgeleitetes Antibiotikum; (b) Daunorubicin, antitumor Anthracyclin; (c) Oxytetracycline, Tetracyclin-Antibiotikum; (d) Bleomycin, NRPS/PKS-abgeleitetes Antibiotikum; (e) Geldanamycin, makrozyklischer Hsp90-Inhibitor; (f) Epothilone B, antitumor Makrozyklin; (g) Fredericamycin A, C-5 spirocyclischer DNA-Polymerase-Inhibitor; (h) Chloramphenicol, Breitspektrumantibiotikum; (i) Tacrolimus, makrozyklisches Immunosuppressivum; (j) Fosfomycin, Breitspektrumantibiotikum; (k) Geranylphenazinediol, Acetyl-CoA-Inhibitor; (l) Actinomycin, NRPS/PKS-abgeleiteter antineoplastischer Wirkstoff und (m) Daptomycin, NRPS-abgeleitetes Antibiotikum.

Des Weiteren wird die hohe Diversität der Strukturen durch die Ergebnisse des MCSS-Identifikationsworkflows unterstrichen. Viele und strukturell sehr unterschiedliche *Most Common Substructures* sind ein sehr gutes Indiz für die Diversität einer Kleinstrukturbibliothek.

Biologische Aktivität und Syntheseweg

Viele pharmazeutisch relevante Strukturen werden durch die nichtribosomalen Peptidsynthetasen (NRPS) synthetisiert (138). Diese zeichnen sich durch eine Vielzahl an unterschiedlichen biologischen Aktivitäten und pharmakologischen Eigenschaften aus. NRPS-abgeleitete Moleküle in der StreptomeDB sind zum Beispiel das Zytostatikum Bleomycin

Tabelle 2.6: StreptomeDB: Entitäten der StreptomeDB und ihre Häufigkeit

enthaltene Kleinstrukturen	2332
Mittelwert des Molekulargewichtes	453,61
Ro5-konforme Kleinstrukturen (91, 137)	1343
Kleinstrukturen mit annotierter Aktivität	875
unterschiedliche Organismen (inklusive Stämme)	2043
referenzierte Primärliteratureinträge	4309
Kleinstruktur-Organismus-Relationen	4280

(139) oder die Antibiotika Daptomycin (140) oder Enduracidin (141).

Die 217 Strukturen, die als Polyketid-Synthase (PKS)-assoziiert beschrieben wurden, stellen die größte Klasse innerhalb der StreptomeDB dar. PKS, ein Multienzymkomplex, spielt analog zu NRPS eine sehr wichtige Rolle in der Produktion von pharmakologisch-relevanten Molekülen (142). Viele der therapeutisch genutzten Antibiotika, wie zum Beispiel Tetracycline, werden mithilfe des PKS-Komplexes synthetisiert. Wichtige Polyketide mit einer annotierten Aktivität in der StreptomeDB sind unter anderem das antibakterielle Oxytetrazyklin und das antitumorale Geldanamycin (81). Neben NRPS und PKS sind noch sieben weitere Synthesewege in der StreptomeDB annotiert, darunter *Ribosomal-Peptide-Synthesis* (RPS) oder *Terpene-Synthesis* (siehe Tabelle 2.5).

Zu 875 Naturstoffen konnte, mindestens eine Aktivitätsinformationen identifiziert werden. Ein Großteil dieser Annotationen sind Klassifikationen/Wirkspektren (z.B. *antifungal* oder *antiviral*), können jedoch zum Teil sehr spezifisch sein (z.B. *PGE2 production is blocked*). Insgesamt werden 73 verschiedene Aktivitäten in der Datenbank verwendet.

2.3.3 Diskussion

Die Größe des chemischen Raums aller durch Streptomyceten synthetisierter Naturstoffe ist bis heute unklar. Bérdy *et al.* beschrieb 2005 in einer Publikation im *Journal of Antibiotics* 3000 bioaktive Strukturen. Die referenzierte Datenbank ist aber nicht mehr erreichbar und es ist unklar, auf welche Strukturen Bezug genommen wurde (143). KNApSAcK und NADB beinhalten ca. 1500 von Streptomyceten synthetisierte Strukturen, die StreptomeDB ca. 2300. Die tatsächliche Anzahl dürfte jedoch um ein Vielfaches größer sein. Nicht nur, weil immer neue Streptomyceten entdeckt und beschrieben werden, die unter 2.3.1 beschriebene Methodik birgt auch den Nachteil, dass wichtige Informationen,

die nur im Volltext von Publikationen enthalten sind, bisher nicht berücksichtigt wurden. Unter anderem auch, weil es schwer ist, an die Volltextpublikationen zu gelangen.

Der komplette Datensatz kann, inklusive Strukturinformationen, heruntergeladen werden. Die Strukturinformationen ermöglichen eine Modellierung von molekularen Interaktionen und können als Ausgangspunkt für ein virtuelles Screening, z.B. mit der ChemicalToolBox (siehe Kapitel 2.1), dienen. Die assoziierten Metadaten ermöglichen eine weitere Analyse der Gencluster bzw. der involvierten Enzyme und können damit z.B. zur Aufklärung der Synthese verwandter Strukturen beitragen (siehe Kapitel **??**). Bis heute ist für viele Naturstoffe das komplette Aktivitätsspektrum nicht geklärt, und eine Kombination aus der ChemicalToolBox und der Genomanalyse-Pipeline könnte hier nähere Aufschlüsse liefern.

Mit den immer wichtiger werdenden *in silico*-Ansätzen zur Identifikation neuer Wirkstoffe (144, 145, 146) bietet die StreptomeDB einen wichtigen Sekundärdatensatz für alle, die im Bereich der pharmazeutischen Wissenschaften arbeiten. Insbesondere weil *Streptomyces* eine der wichtigsten Gattungen für die Produktion von therapeutisch relevanten Naturstoffen ist.

Die StreptomeDB wird von der AG Günther gepflegt und eine erweiterte Version ist für das dritte Quartal 2015 geplant.

Teil III

Abschlussbetrachtung

3.1 Zugänglichkeit bio- und cheminformatischer Methoden

Die Anwendung bioinformatischer Methoden wurde in der vorliegenden Arbeit nachweislich einer breiten Benutzerschicht zugänglich gemacht. Lebenswissenschaftler sind mithilfe der Galaxy-Plattform in der Lage, ihre eigenen Daten selbstständig auszuwerten. Dabei spielt die Größe der Daten keine Rolle, wie am Beispiel der ChemicalBox gezeigt wurde. Die neuen Tools, Workflows und Visualisierungen für die Genomannotation wurden publiziert (31, 32) und dienten als Grundlage für die Analyse von einer Vielzahl von Streptomyceten und Pilzen in dieser Arbeit und durch Anwender.

Cheminformatik im HPC-Bereich war ohne ein großes finanzielles Budget vor Existenz der ChemicalToolBox nur eingeschränkt möglich. Mit der vorliegenden Arbeit wurde eine Cheminformatik-Community etabliert, die es sich zur Aufgabe gesetzt hat, cheminformatische Methoden für Big Data zu etablieren und reproduzierbar zugänglich zu machen.

3.2 Verbesserung der Lehre

Die Arbeit hat dazu beigetragen, die Lehre und die Ausbildung von Nachwuchswissenschaftlern und Studenten zu fördern. Der Freiburger Galaxy-Server wird aktiv in der pharmazeutischen und biologischen Lehre verwendet und dient der Ausbildung von Doktoranden im Umgang mit Big Data und bioinformatischen Methoden. Mehrere Workshops im Jahr gewährleisten eine adäquate Einführung in die verschiedenen Themen der Bio- und Cheminformatik. Ein reger Austausch mit internationalen Trainingsnetzwerken wie dem *Galaxy Training Network*[1] oder der *Global Organisation for Bioinformatics Learning, Education and Training*[2] (GOBLET) gewährleisten eine hohe Qualität und Aktualität der Lehrmaterialien (147).

3.3 Nachhaltigkeit durch den Aufbau von Communitys

Ein regelmäßiger Austausch mit anderen Entwicklern und Benutzern war integraler Bestandteil dieser Arbeit. Viele der integrierten Tools und entwickelten Workflows sind aufgrund von Benutzeranfragen entstanden, welches sich auch in dem interdisziplinären Charakter der Arbeit niederschlägt. Das Pharmazeutische Institut ist sehr interdisziplinär auf-

[1] `https://wiki.galaxyproject.org/News/GalaxyTrainingNetwork`
[2] `http://www.mygoblet.org/`

gestellt und hatte bioinformatische Anforderungen im Bereich der Wirkstoffentwicklung und der Genomik. Beides konnte in dieser Plattform vereinigt werden. Dieser Tatsache ist es auch geschuldet, dass die Freiburger Galaxy-Instanz mittlerweile auch Workflows und Tools zur Analyse von *High-throughput*-Sequenzierdaten und Massenspektrometrie anbietet. Erste Publikationen aus Freiburg sind hierzu bereits veröffentlicht und belegen eindrucksvoll, wie man Galaxy als universelle *Omics-Plattform* einsetzen kann (148, 149).

Ein Ziel der Arbeit war es, die Nachhaltigkeit der entwickelten Tools und Dienstleistungen zu gewährleisten. Um dies zu erreichen, wurde von Anfang an die Community in den Entwicklungsprozess einbezogen. Der Quellcode wurde öffentlich zur Verfügung gestellt, dokumentiert und *best practice guidelines* etabliert. Das *galaxytools* GitHub Archiv, in dem alle Tools dieser Arbeit entwickelt wurden, zählt bisher 25 beitragende Personen. Die große Anzahl an Beitragenden gewährleistet eine nachhaltige Weiterentwicklung und Pflege der Genomanalyse Tools und der ChemicalToolBox. Eine *Intergalactic Utilities Commission* (IUC)[1] wurde gegründet, um einheitliche Standards[2] zu etablieren und qualitativ hochwertige Tools in der Galaxy Tool Shed zu forcieren. Die Publikation von Tang *et al.* verdeutlicht exemplarisch, dass die Idee der engen Zusammenarbeit in einer Community funktioniert (150). Sie implementiert eine RNA Strukturvorhersage und definiert acht Abhängigkeiten, die alle in dieser Arbeit bzw. von der IUC entwickelt wurden. Der Mehraufwand für Tang *et al.* konnte so erheblich reduziert werden. Ähnlich machte es Steffen Lott aus der Arbeitsgruppe Hess für experimentelle Bioinformatik in Freiburg. Sein Programm *CoVennTree* wurde noch vor der Publikation in die Galaxy Tool Shed und dem Freiburger Galaxy Server integriert(29).

3.4 Reproduzierbarkeit wissenschaftlicher Ergebnisse

Eine große Herausforderung ist und bleibt die Reproduzierbarkeit wissenschaftlicher Ergebnisse. In der vorliegenden Arbeit konnte gezeigt werden, dass Galaxy in der Lage ist, ein Experiment so detailliert zu protokollieren, dass eine Reproduzierbarkeit gewährleistet ist. Dies setzt jedoch voraus, dass alle Tools und Annotationsdatenbanken zu jeder Zeit verfügbar sind, eine Voraussetzung, die nicht immer gegeben ist. Es gab etliche Fälle,

[1] `https://wiki.galaxyproject.org/IUC`
[2] `https://github.com/galaxy-iuc/standards`

bei denen der Quellcode eines Programms verschwand und nicht mehr für die Galaxy-Installationsbeschreibung zur Verfügung stand. Darüber hinaus werden Sequenzdatenbanken immer noch nicht versioniert und machen eine Reproduzierbarkeit unmöglich. Die einzige Möglichkeit dem zu begegnen ist es, alle Quellen, deren Nachhaltigkeit nicht garantiert ist, selbst zu archivieren. Dies ist jedoch teuer und wird von bisherigen Förderstrukturen nicht unterstützt. Als temporärer Ausweg wurde daher ein freier Onlinespeicherservice genutzt um, die essentiellen Quellpakete zu sichern[1]. Sequenzdatenbanken können damit jedoch nicht gesichert werden (persönliche Korrespondenz GitHub Inc.).

3.5 Text Mining und Datenbanken

Die drei veröffentlichten Webservices CIL, Prolific und StreptomeDB basieren zum großen Teil auf der gleichen Datenbankstruktur. Die hierzu notwendige Infrastruktur ging zum einen im PubMed2Go Projekt auf, welches zur Zeit von Kersten Döring betreut wird, zum anderen in der ChemicalToolBox.

Sowohl die Genomanalyse-Pipeline als auch die ChemicalToolBox arbeiten hauptsächlich mit Datenbanken, die in den meisten Fällen Sekundärdaten beinhalten. Primärdaten sind zum größten Teil immer noch in Publikationen und damit in unstrukturierten Texten verborgen. Kommen Kleinstrukturen, Gene oder Proteine in keiner Datenbank vor, sollte in den Primärdaten gesucht werden. CIL und Prolific machen dies besonders einfach und ermöglichen dem Benutzer die Identifikation von Strukturen oder Genen in der Primärliteratur. Basierend auf CIL und manueller Kuration von Publikationen, konnte mit der StreptomeDB eine einzigartige Naturstoffdatenbank von Streptomyceten produzierten Metaboliten geschaffen werden, die ihrerseits wieder in die ChemicalBox einfließen kann und damit die Grundlage für das *in silico Screening* zur Identifikation neuer Wirkstoffe sein kann.

[1] `https://github.com/bgruening/download_store`

3.6 Ausblick

Zugänglichkeit, Reproduzierbarkeit und Transparenz in der Wissenschaft ist ein hohes Ziel! Dies weiter zu verbessern und zu priorisieren, wird auch in Zukunft notwendig sein. Mit Galaxy sind dafür gute Grundlagen gelegt, doch das Potential der Plattform ist noch lange nicht erschöpft. Im Rahmen des Galaxy Freiburg Projektes wird intensiv an der Integration weiterer Wissenschaftsfelder gearbeitet. Proteomik-Tools und -Workflows werden mit der Unterstützung der AG Schilling[1] (Institut für Molekulare Medizin und Zellforschung, Universität Freiburg) entwickelt, die Integration von bildgebenden Verfahren wird im Rahmen einer Kooperation mit dem ZBSA vorangetrieben und das Institut für Pharmazeutische Wissenschaften der Universität Freiburg hat ein stetes Interesse an der Weiterentwicklung der Genomanalyse-Pipeline. Mit internationalen Partnern werden Erweiterungen für die Ingenieurwissenschaften und Literaturwissenschaften entwickelt, die in naher Zukunft der Universität Freiburg zur Verfügung stehen sollen.

Die Virtualisierung von Analyseplattformen wird mit der wachsenden Menge an Daten immer wichtiger. Der Transfer der Daten wird ein zunehmend limitierender Faktor, der es immer aufwendiger macht die Daten zu den Auswertungstools zu befördern. Eine Lösungsstrategie zu diesem Problem ist es, die Software zu den Daten zu bewegen. Praktisch würde dies bedeuten, dass Galaxy-Instanzen mit allen nötigen Tools und Workflows virtualisiert zur Verfügung gestellt werden und ohne Installation, unabhängig von der zugrundeliegenden Hardware, überall betrieben werden können. Aus dieser Idee heraus wurde vom Autor das Galaxy-Docker Projekt[2] initiiert, welches in Abbildung 26 skizziert ist. Eine virtualisierte Galaxy-Instanz, die vorkonfiguriert für den produktiven Einsatz für mehrere 100 Benutzer ist, wird in Zukunft als Alternative zu heute gängigen Galaxy-Installationen angeboten. Dies soll auch kleineren Instituten die Möglichkeit geben, Galaxy als Service anzubieten.

Fehlende Flexibilität ist eines der Hauptgründe, weshalb einige Benutzer mit Programmier- und Unixkenntnissen sich scheuen, Galaxy für ihre Analyse zu benutzen. Das Konzept von Tools ist nicht so flexibel wie eine Programmiersprache oder die Kommandozeile in Unix-artigen Systemen, bietet jedoch die erwähnten Vorteile der Reproduzierbarkeit und Transparenz, die normalerweise sonst nicht gegeben sind. Um beide Welten näher

[1] `https://www.mol-med.uni-freiburg.de/mom/schilling`
[2] `https://github.com/bgruening/docker-galaxy-stable`

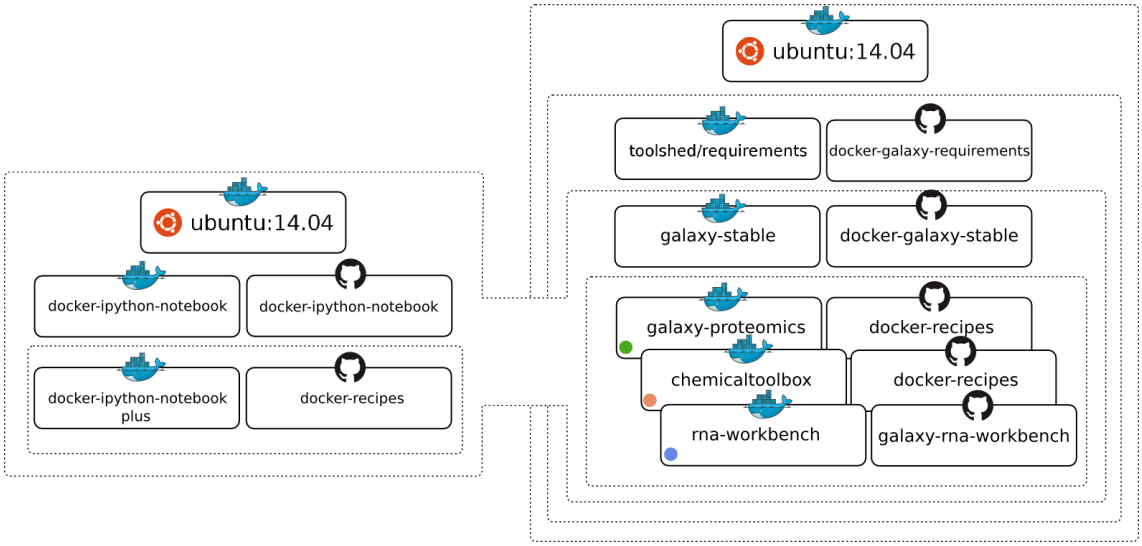

Abbildung 26: Modulare virtualisierte Galaxy-Installationen - Aufbauend auf einem Standard-Linux-System werden verschiedenen Komponenten installiert, die am Ende eine Galaxy-Instanz ergeben. Diese Basis-Galaxy-Installation (galaxy-stable), kann mit beliebigen Tools aus der Galaxy Tool Shed erweitert werden und ermöglicht es, individuell angepasste Analyseplattformen, sogenannte Galaxy-Flavors, zu kreieren. Andere Erweiterungen, wie die interaktive Programmierumgebung IPython, können ebenso in die virtualisierten Galaxy-Flavors integriert werden.

zusammenzubringen, wurde das Konzept der *Interactive Environments* entwickelt und IPython in Galaxy integriert. Mit IPython steht eine interaktive Entwicklungsumgebung in Galaxy zur Verfügung, mit der die Benutzer in einer Programmiersprache der Wahl mit ihren Daten interagieren können. Der Quellcode wird dabei in der Galaxy-History gespeichert und gewährleistet die Reproduzierbarkeit und Transparenz der Analyse.

Mit dem Interesse von ELIXIR (*European life-sciences Infrastructure for biological Information*[1]) an Galaxy und den Virtualisierungsansätzen sieht der Ausblick für die in dieser Arbeit entwickelten Tools und Workflows vielversprechend aus und die Nachhaltigkeit scheint auch in Europa gesichert.

[1] `https://www.elixir-europe.org/`

Eigenleistungen

Quellcode und Dokumentation

Die erstellten Programme, Datentypen, Visualisierungen und Dokumentationen sind in versionierter Form unter `https://github.com/bgruening` für jeden zugänglich und unter anderem in der Freiburger Galaxy-Instanz integriert. Darüber hinaus werden virtuelle Container mit einer vorkonfigurierten Galaxy-Instanz für den Produktiveinsatz angeboten. Allein im letzten Jahr der vorliegenden Dissertation wurden vom Autor 2.604 Beiträge erarbeitet und der Community auf GitHub zur Verfügung gestellt. Weitere Beiträge wurden zu zahlreichen anderen Projekten getätigt, im Folgenden eine Auflistung der wichtigsten für diese Arbeit:

- Galaxy-Tools-Archiv[1] (1.367 Quellcode-Beiträge)
- IUC-Tools-Archiv[2] (361 Quellcode-Beiträge)
- Galaxy-Projekt[3] (147 Quellcode-Beiträge)
- Galaxy-Docker-Projekt[4] (139 Quellcode-Beiträge)
- Galaxy-Proteomik-Projekt[5] (109 Quellcode-Beiträge)
- DevTeam-Tools[1] (39 Quellcode-Beiträge)

[1] `https://github.com/bgruening/galaxytools/commits?author=bgruening`
[2] `https://github.com/galaxyproject/tools-iuc/commits?author=bgruening`
[3] `https://github.com/galaxyproject/galaxy/commits?author=bgruening`
[4] `https://github.com/bgruening/docker-galaxy-stable/commits?author=bgruening`
[5] `https://github.com/galaxyproteomics/tools-galaxyp/commits?author=bgruening`
[1] `https://github.com/galaxyproject/tools-devteam/commits?author=bgruening`

Publikationen

Teile dieser Arbeit wurden publiziert. Insgesamt sind im Rahmen dieser Dissertation die folgenden 18 peer-reviewed Publikationen entstanden.

1. Guy Yachdav, Tatyana Goldberg, Sebastian Wilzbach, David Dao, Iris Shih, Saket Choudhary, Steve Crouch, Max Franz, Alexander García, Leyla J García, Björn A Grüning, Devasena Inupakutika, Ian Sillitoe, Anil S Thanki, Bruno Vieira, José M Villaveces, Maria V Schneider, Suzanna Lewis, Steve Pettifer, Burkhard Rost, und Manuel Corpas. Anatomy of BioJS, an open source community for the life sciences. *eLife*, 4, January 2015. ISSN 2050-084X. doi: 10.7554/eLife.07009

2. Sebastian Preissl, Martin Schwaderer, Alexandra Raulf, Michael Hesse, Björn A Grüning, Claudia Köbele, Rolf Backofen, Bernd K Fleischmann, Lutz Hein, und Ralf Gilsbach. Deciphering the Epigenetic Code of Cardiac Myocyte Transcription. *Circulation Research*, June 2015. ISSN 0009-7330. doi: 10.1161/CIRCRESAHA.115.306337

3. Deborah Roidl, Nicole Hellbach, Patrick Bovio, Alejandro Villarreal, Stefanie Heidrich, Sigrun Nestel, Björn A. Grüning, Ulrike Bönisch, und Tanja Vogel. DOT1L activity promotes proliferation and protects cortical neural stem cells from activation of ATF4-DDIT3-mediated ER stress in vitro. *STEM CELLS*, August 2015. ISSN 10665099. doi: 10.1002/stem.2187

4. Xavier Lucas, Björn A Grüning, Stefan Bleher, und Stefan Günther. The Purchasable Chemical Space: A Detailed Picture. *Journal of chemical information and modeling*, April 2015. ISSN 1549-960X. doi: 10.1021/acs.jcim.5b00116

5. Andreas Präg, Björn A Grüning, Matthias Häckh, Steffen Lüdeke, Marcel Wilde, Andriy Luzhetskyy, Michael Richter, Marta Luzhetska, Stefan Günther, und Michael Müller. Regio- and stereoselective intermolecular oxidative phenol coupling in Streptomyces. *Journal of the American Chemical Society*, 136(17):6195–8, April 2014. ISSN 1520-5126. doi: 10.1021/ja501630w

6. Hitesh Patel, Björn A Grüning, Stefan Günther, und Irmgard Merfort. PyWATER: a PyMOL plug-in to find conserved water molecules in proteins by clustering. *Bioinformatics (Oxford, England)*, July 2014. ISSN 1367-4811. doi: 10.1093/bioinformatics/btu424

7. Fidel Ramírez, Friederike Dündar, Sarah Diehl, Björn A Grüning, und Thomas Manke. deepTools: a flexible platform for exploring deep-sequencing data. *Nucleic acids research*, 42(Web Server issue):W187–91, July 2014. ISSN 1362-4962. doi: 10.1093/nar/gku365

8. Ralf Gilsbach, Sebastian Preissl, Björn A. Grüning, Tilman Schnick, Lukas Burger, Vladimir Benes, Andreas Würch, Ulrike Bönisch, Stefan Günther, Rolf Backofen, Bernd K. Fleischmann, Dirk Schübeler, und Lutz Hein. Dynamic DNA methylation orchestrates cardiomyocyte development, maturation and disease. *Nature communications*, 5:5288, January 2014. ISSN 2041-1723. doi: 10.1038/ncomms6288

9. Desirée Schubert, Claudia Bode, Rupert Kenefeck, Tie Zheng Hou, James B Wing, Alan Kennedy, Alla Bulashevska, Britt-Sabina Petersen, Alejandro A Schäffer, Björn A Grüning, Susanne Unger, Natalie Frede, Ulrich Baumann, Torsten Witte, Reinhold E Schmidt, Gregor Dueckers, Tim Niehues, Suranjith Seneviratne, Maria Kanariou, Carsten Speckmann, Stephan Ehl, Anne Rensing-Ehl, Klaus Warnatz, Mirzokhid Rakhmanov, Robert Thimme, Peter Hasselblatt, Florian Emmerich, Toni Cathomen, Rolf Backofen, Paul Fisch, Maximilian Seidl, Annette May, Annette Schmitt-Graeff, Shinji Ikemizu, Ulrich Salzer, Andre Franke, Shimon Sakaguchi, Lucy S K Walker, David M Sansom, und Bodo Grimbacher. Autosomal dominant immune dysregulation syndrome in humans with CTLA4 mutations. *Nature medicine*, 20(12):1410–6, December 2014. ISSN 1546-170X. doi: 10.1038/nm.3746

10. Xavier Lucas, Christian Senger, Anika Erxleben, Björn A Grüning, Kersten Döring, Johannes Mosch, Stephan Flemming, und Stefan Günther. StreptomeDB: a resource for natural compounds isolated from Streptomyces species. *Nucleic acids research*, 41(Database issue):D1130–6, January 2013. ISSN 1362-4962. doi: 10.1093/nar/gks1253

11. Peter J A Cock, Björn A Grüning, Konrad Paszkiewicz, und Leighton Pritchard. Galaxy tools and workflows for sequence analysis with applications in molecular plant pathology. *PeerJ*, 1:e167, January 2013. ISSN 2167-8359. doi: 10.7717/peerj.167

12. Björn A Grüning, Anika Erxleben, Anna Hähnlein, und Stefan Günther. Draft Genome Sequence of Streptomyces viridochromogenes Strain Tu57, Producer of Avilamycin. *Genome announcements*, 1(3):e00384–13, January 2013. ISSN 2169-8287. doi: 10.1128/genomeA.00384-13

13. Loubna Youssar, Björn Andreas Grüning, Stefan Günther, und Wolfgang Hüttel. Characterization and phylogenetic analysis of the mitochondrial genome of Glarea lozoyensis indicates high diversity within the order Helotiales. *PloS one*, 8(9):e74792, January 2013. ISSN 1932-6203. doi: 10.1371/journal.pone.0074792

14. Jan Bieschke, Martin Herbst, Thomas Wiglenda, Ralf P Friedrich, Annett Boeddrich, Franziska Schiele, Daniela Kleckers, Juan Miguel Lopez del Amo, Björn A Grüning, Qinwen Wang, Michael R Schmidt, Rudi Lurz, Roger Anwyl, Sigrid Schnoegl, Marcus Fändrich, Ronald F Frank, Bernd Reif, Stefan Günther, Dominic M Walsh, und Erich E Wanker. Small-molecule conversion of toxic oligomers to nontoxic β-sheet-rich amyloid fibrils. *Nature chemical biology*, 8(1):93–101, January 2012. ISSN 1552-4469. doi: 10.1038/nchembio.719

15. Loubna Youssar, Björn Andreas Grüning, Anika Erxleben, Stefan Günther, und Wolfgang Hüttel. Genome sequence of the fungus Glarea lozoyensis: the first genome sequence of a species from the Helotiaceae family. *Eukaryotic cell*, 11(2):250, February 2012. ISSN 1535-9786. doi: 10.1128/EC.05302-11

16. Christian Senger, Björn A Grüning, Anika Erxleben, Kersten Döring, Hitesh Patel, Stephan Flemming, Irmgard Merfort, und Stefan Günther. Mining and evaluation of molecular relationships in literature. *Bioinformatics (Oxford, England)*, 28(5):709–14, March 2012. ISSN 1367-4811. doi: 10.1093/bioinformatics/bts026

17. Anika Erxleben, J. Wunsch-Palasis, B. A. Grüning, M. Luzhetska, A. Bechthold, und S. Gunther. Genome Sequence of Streptomyces sp. Strain Tu6071. *Journal of Bacteriology*, 193(16):4278–4279, June 2011. ISSN 0021-9193. doi: 10.1128/JB.00377-11

18. Björn A Grüning, Christian Senger, Anika Erxleben, Stephan Flemming, und Stefan Günther. Compounds In Literature (CIL): screening for compounds and relatives in PubMed. *Bioinformatics (Oxford, England)*, 27(9):1341–2, May 2011. ISSN 1367-4811. doi: 10.1093/bioinformatics/btr130

Referenzen

[1] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, und D.J. Lipman. Basic Local Alignment Search Tool. *Journal of molecular biology*, 215(3):403–410, 1990. ISSN 0022-2836. doi: 10.1006/jmbi.1990.9999.

[2] Christiam Camacho, George Coulouris, Vahram Avagyan, Ning Ma, Jason Papadopoulos, Kevin Bealer, und Thomas L Madden. BLAST+: architecture and applications. *BMC bioinformatics*, 10:421, January 2009. ISSN 1471-2105. doi: 10.1186/1471-2105-10-421.

[3] Derrick E Wood und Steven L Salzberg. Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome biology*, 15:R46, 2014. ISSN 1465-6914. doi: 10.1186/gb-2014-15-3-r46.

[4] Benjamin Buchfink, Chao Xie, und Daniel H Huson. Fast and sensitive protein alignment using DIAMOND. *Nature methods*, 12(1):59–60, November 2014. ISSN 1548-7105. doi: 10.1038/nmeth.3176.

[5] Lincoln D Stein. The case for cloud computing in genome informatics. *Genome biology*, 11(5):207, January 2010. ISSN 1465-6914. doi: 10.1186/gb-2010-11-5-207.

[6] Florian Prinz, Thomas Schlange, und Khusru Asadullah. Believe it or not: how much can we rely on published data on potential drug targets? *Nature reviews. Drug discovery*, 10(9):712, September 2011. ISSN 1474-1784. doi: 10.1038/nrd3439-c1.

[7] C Glenn Begley und Lee M Ellis. Drug development: Raise standards for preclinical cancer research. *Nature*, 483(7391):531–3, March 2012. ISSN 1476-4687. doi: 10.1038/483531a.

[8] John Arrowsmith. Trial watch: Phase II failures: 2008-2010., 2011. ISSN 1474-1776.

[9] John P A Ioannidis. Why most published research findings are false. *PLoS medicine*, 2(8):e124, August 2005. ISSN 1549-1676. doi: 10.1371/journal.pmed.0020124.

[10] Rebooting review. *Nature Biotechnology*, 33(4):319–319, April 2015. ISSN 1087-0156. doi: 10.1038/nbt.3202.

[11] Leonard P. Freedman, Iain M. Cockburn, und Timothy S. Simcoe. The Economics of Reproducibility in Preclinical Research. *PLOS Biology*, 13(6):e1002165, June 2015. ISSN 1545-7885. doi: 10.1371/journal.pbio.1002165.

[12] Francis S. Collins und Lawrence A. Tabak. Policy: NIH plans to enhance reproducibility. *Nature*, 505(7485):612–613, January 2014. ISSN 0028-0836. doi: 10.1038/505612a.

[13] Journals unite for reproducibility. *Nature*, 515(7525):7–7, 2014. ISSN 0028-0836. doi: 10.1038/515007a.

[14] Darrel C Ince, Leslie Hatton, und John Graham-Cumming. The case for open computer programs. *Nature*, 482(7386):485–8, February 2012. ISSN 1476-4687. doi: 10.1038/nature10836.

[15] Eric S. Raymond. *The Cathedral & The Bazaar: Musings on Linux and Open Source*. O'Reilly Media, Cambridge, Mass., 1999. ISBN 1-56592-724-9.

[16] Xosé M. Fernández-Suárez, Daniel J. Rigden, und Michael Y. Galperin. The 2014 Nucleic Acids Research Database Issue and an updated NAR online Molecular Biology Database Collection. *Nucleic acids research*, 42 (Database issue):D1–6, January 2014. ISSN 1362-4962. doi: 10.1093/nar/gkt1282.

[17] Wolfram Liebermeister, Elad Noor, Avi Flamholz, Dan Davidi, Jörg Bernhardt, und Ron Milo. Visual account of protein investment in cellular functions. *Proceedings of the National Academy of Sciences of the United States of America*, 111(23):8488–93, June 2014. ISSN 1091-6490. doi: 10.1073/pnas.1314810111.

[18] Sebastian J. Schultheiss, Marc-Christian Münch, Gergana D. Andreeva, und Gunnar Rätsch. Persistence and availability of Web services in computational biology. *PloS one*, 6(9):e24914, January 2011. ISSN 1932-6203. doi: 10.1371/journal.pone.0024914.

[19] Ora Lassila Tim Berners-Lee, James Hendler. The Semantic Web: a new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, 284(5):34–43, 2001.

[20] Tim Berners-Lee Nigel Shadbolt, Wendy Hall. The Semantic Web Revisited. In *IEEE Intelligent Systems*, 2006.

[21] E.S. Raymond. *The art of unix programming*. Addison-Wesley, 2003. ISBN 0131429019.

[22] P. Naur, B. Randell, und NATO Science Committee. Software Engineering: Report of a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7-11 Oct. 1968. In *NATO Software Engineering Conference*, page 231, 1968. doi: 10.1093/bib/bbp050.

[23] Seymour Papert, Idit Harel, und By Seymour Papert. Situating Constructionism. *Constructionism*, 36:1–11, 1991. doi: 10.1111/1467-9752.00269.

[24] Xavier Lucas, Björn A Grüning, Stefan Bleher, und Stefan Günther. The Purchasable Chemical Space: A Detailed Picture. *Journal of chemical information and modeling*, April 2015. ISSN 1549-960X. doi: 10.1021/acs.jcim.5b00116.

[25] Daniel Blankenberg, James Taylor, und Anton Nekrutenko. Making whole genome multiple alignments usable for biologists. *Bioinformatics (Oxford, England)*, 27 (17):2426–8, September 2011. ISSN 1367-4811. doi: 10.1093/bioinformatics/btr398.

[26] A. Medina-Rivera, M. Defrance, O. Sand, C. Herrmann, J. A. Castro-Mondragon, J. Delerce, S. Jaeger, C. Blanchet, P. Vincens, C. Caron, D. M. Staines, B. Contreras-Moreira, M. Artufel, L. Charbonnier-Khamvongsa, C. Hernandez, D. Thieffry, M. Thomas-Chollier, und J. van Helden. RSAT 2015: Regulatory Sequence Analysis Tools. *Nucleic Acids Research*, 2015. ISSN 0305-1048. doi: 10.1093/nar/gkv362.

[27] Laurent Modolo und Emmanuelle Lerat. UrQt: an efficient software for the Unsupervised Quality trimming of NGS data. *BMC Bioinformatics*, 16(1):137, 2015. ISSN 1471-2105. doi: 10.1186/s12859-015-0546-8.

[28] Yassen Assenov, Fabian Müller, Pavlo Lutsik, Jörn Walter, Thomas Lengauer, und Christoph Bock. Comprehensive Analysis of DNA Methylation Data With RnBeads. http://rnbeads.mpi-inf.mpg.de. 2014. ISSN 15487105. doi: 10.1038/nmeth.3115.

[29] Steffen C. Lott, Björn Voss, Wolfgang R. Hess, und Claudia Steglich. CoVennTree: a new method for the comparative analysis of large datasets. *Frontiers in Genetics*, 6, February 2015. ISSN 1664-8021. doi: 10.3389/fgene.2015.00043.

[30] Andrea Sboner, Xinmeng Jasmine Mu, Dov Greenbaum, Raymond K Auerbach, und Mark B Gerstein. The real cost of sequencing: higher than you think! *Genome biology*, 12(8):125, January 2011. ISSN 1465-6914. doi: 10.1186/gb-2011-12-8-125.

[31] Peter J A Cock, Björn A Grüning, Konrad Paszkiewicz, und Leighton Pritchard. Galaxy tools and workflows for sequence analysis with applications in molecular plant pathology. *PeerJ*, 1:e167, January 2013. ISSN 2167-8359. doi: 10.7717/peerj.167.

[32] Peter J. A. Cock, John M. Chilton, Björn Grüning, James E. Johnson, und Nicola Soranzo. NCBI BLAST+ integrated into Galaxy. January 2015. doi: 10.1101/014043.

[33] Eric P. Nawrocki und Sean R. Eddy. Infernal 1.1: 100-fold faster RNA homology searches. *Bioinformatics*, 29 (22):2933–2935, 2013. ISSN 13674803. doi: 10.1093/bioinformatics/btt509.

[34] Sam Griffiths-Jones, Alex Bateman, Mhairi Marshall, Ajay Khanna, und Sean R. Eddy. Rfam: An RNA family database, 2003. ISSN 03051048.

[35] E. P. Nawrocki, S. W. Burge, A. Bateman, J. Daub, R. Y. Eberhardt, S. R. Eddy, E. W. Floden, P. P. Gardner, T. A. Jones, J. Tate, und R. D. Finn. Rfam 12.0: updates to the RNA families database. *Nucleic Acids Research*, 43(D1):D130–D137, 2014. ISSN 0305-1048. doi: 10.1093/nar/gku1063.

[36] Arthur L. Delcher, Kirsten A. Bratke, Edwin C. Powers, und Steven L. Salzberg. Identifying bacterial genes and endosymbiont DNA with Glimmer. *Bioinformatics*, 23(6):673–679, 2007. ISSN 13674803. doi: 10.1093/bioinformatics/btm009.

[37] Mario Stanke und Stephan Waack. Gene prediction with a hidden Markov model and a new intron submodel. In *Bioinformatics*, volume 19, 2003. ISBN 1367-4811 (Electronic)\n1367-4803 (Linking). doi: 10.1093/bioinformatics/btg1080.

[38] A. F. A. Smit, R. Hubley, und P. Green. RepeatMasker. URL `http://repeatmasker.org`.

[39] Marco Punta und Yanay Ofran. The rough guide to in silico function prediction, or how to use sequence and structure information to predict protein function. *PLoS computational biology*, 4(10):e1000160, October 2008. ISSN 1553734X. doi: 10.1371/journal.pcbi.1000160.

[40] Gabriel Moreno-Hagelsieb und Kristen Latimer. Choosing BLAST options for better detection of orthologs as reciprocal best hits. *Bioinformatics*, 24(3):319–324, 2008. ISSN 13674803. doi: 10.1093/bioinformatics/btm585.

[41] L Holm und C Sander. New structure–novel fold? *Structure (London, England : 1993)*, 5(2):165–71, February 1997. ISSN 0969-2126.

[42] S E Brenner, C Chothia, und T J Hubbard. Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. *Proceedings of the National Academy of Sciences of the United States of America*, 95(11):6073–8, May 1998. ISSN 0027-8424.

[43] Philip Jones, David Binns, Hsin-Yu Chang, Matthew Fraser, Weizhong Li, Craig McAnulla, Hamish McWilliam, John Maslen, Alex Mitchell, Gift Nuka, Sebastien Pesseat, Antony F Quinn, Amaia Sangrador-Vegas, Maxim Scheremetjew, Siew-Yit Yong, Rodrigo Lopez, und Sarah Hunter. InterProScan 5: genome-scale protein function classification. *Bioinformatics (Oxford, England)*, pages btu031–, January 2014. ISSN 1367-4811. doi: 10.1093/bioinformatics/btu031.

[44] Kai Blin, Marnix H. Medema, Daniyal Kazempour, Michael A. Fischbach, Rainer Breitling, Eriko Takano, und Tilmann Weber. antiSMASH 2.0–a versatile platform for genome mining of secondary metabolite producers. *Nucleic acids research*, 41(Web Server issue), 2013. ISSN 13624962. doi: 10.1093/nar/gkt449.

[45] Dennis A Benson, Mark Cavanaugh, Karen Clark, Ilene Karsch-Mizrachi, David J Lipman, James Ostell, und Eric W Sayers. GenBank. *Nucleic acids research*, 41 (Database issue):D36–42, January 2013. ISSN 1362-4962. doi: 10.1093/nar/gks1195.

[46] Rolf Apweiler, Amos Bairoch, Cathy H Wu, Winona C Barker, Brigitte Boeckmann, Serenella Ferro, Elisabeth Gasteiger, Hongzhan Huang, Rodrigo Lopez, Michele Magrane, Maria J Martin, Darren A Natale, Claire O'Donovan, Nicole Redaschi, und Lai-Su L Yeh. UniProt: the Universal Protein knowledgebase. *Nucleic acids research*, 32(Database issue):D115–9, January 2004. ISSN 1362-4962. doi: 10.1093/nar/gkh131.

[47] Emily J Richardson und Mick Watson. The automatic annotation of bacterial genomes. *Briefings in bioinformatics*, pages bbs007–, March 2012. ISSN 1477-4054. doi: 10.1093/bib/bbs007.

[48] Jeremy Goecks, Nate Coraor, The Galaxy Team, Anton Nekrutenko, und James Taylor. NGS analyses by visualization with Trackster. *Nature biotechnology*, 30 (11):1036–9, November 2012. ISSN 1546-1696. doi: 10.1038/nbt.2404.

[49] Helga Thorvaldsdóttir, James T Robinson, und Jill P Mesirov. Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. *Briefings in bioinformatics*, 14(2):178–92, April 2012. ISSN 1477-4054. doi: 10.1093/bib/bbs017.

[50] Aaron C E Darling, Bob Mau, Frederick R. Blattner, und Nicole T. Perna. Mauve: Multiple alignment of conserved genomic sequence with rearrangements. *Genome Research*, 14(7):1394–1403, 2004. ISSN 10889051. doi: 10.1101/gr.2289704.

[51] Maria Papagianni. Recent advances in engineering the central carbon metabolism of industrially important bacteria. *Microbial Cell Factories*, 11(1):50, 2012. ISSN 1475-2859. doi: 10.1186/1475-2859-11-50.

[52] Andreas Präg, Björn A Grüning, Matthias Häckh, Steffen Lüdeke, Marcel Wilde, Andriy Luzhetskyy, Michael Richter, Marta Luzhetska, Stefan Günther, und Michael Müller. Regio- and stereoselective intermolecular oxidative phenol coupling in Streptomyces. *Journal of the American Chemical Society*, 136(17):6195–8, April 2014. ISSN 1520-5126. doi: 10.1021/ja501630w.

[53] Anika Erxleben, J. Wunsch-Palasis, B. A. Grüning, M. Luzhetska, A. Bechthold, und S. Gunther. Genome Sequence of Streptomyces sp. Strain Tu6071. *Journal of Bacteriology*, 193(16):4278–4279, June 2011. ISSN 0021-9193. doi: 10.1128/JB.00377-11.

[54] Clemens Dürr, Hans-Jörg Schnell, Andriy Luzhetskyy, Renato Murillo, Monika Weber, Katrin Welzel, Andreas Vente, und Andreas Bechthold. Biosynthesis of the terpene phenalinolactone in Streptomyces sp. Tü6071: analysis of the gene cluster and generation of derivatives. *Chemistry & biology*, 13(4):365–77, April 2006. ISSN 1074-5521. doi: 10.1016/j.chembiol.2006.01.011.

[55] Björn A Grüning, Anika Erxleben, Anna Hähnlein, und Stefan Günther. Draft Genome Sequence of Streptomyces viridochromogenes Strain Tu57, Producer of Avilamycin. *Genome announcements*, 1(3):e00384–13, January 2013. ISSN 2169-8287. doi: 10.1128/genomeA.00384-13.

[56] F Buzzetti, F Eisenberg, H N Grant, W Keller-Schierlein, W Voser, und H Zähner. Avilamycin. *Experientia*, 24(4): 320–3, April 1968. ISSN 0014-4754.

[57] W S Champney und C L Tober. Evernimicin (SCH27899) Inhibits both Translation and 50S Ribosomal Subunit Formation in Staphylococcus aureus Cells. *Antimicrobial Agents and Chemotherapy*, 44(6):1413–1417, June 2000. ISSN 0066-4804. doi: 10.1128/AAC.44.6.1413-1417.2000.

[58] P C Fuchs, A L Barry, und S D Brown. In vitro activities of SCH27899 alone and in combination with 17 other antimicrobial agents. *Antimicrobial agents and chemotherapy*, 43(12):2996–7, December 1999. ISSN 0066-4804.

[59] D R Foster und M J Rybak. Pharmacologic and bacteriologic properties of SCH-27899 (Ziracin), an investigational antibiotic from the everninomicin family. *Pharmacotherapy*, 19(10):1111–7, October 1999. ISSN 0277-0008.

[60] S Gaisser, A Trefzer, S Stockert, A Kirschning, und A Bechthold. Cloning of an avilamycin biosynthetic gene cluster from Streptomyces viridochromogenes Tü57. *Journal of bacteriology*, 179(20):6271–8, October 1997. ISSN 0021-9193.

[61] G Weitnauer, A Mühlenweg, A Trefzer, D Hoffmeister, R D Süssmuth, G Jung, K Welzel, A Vente, U Girreser, und A Bechthold. Biosynthesis of the orthosomycin antibiotic avilamycin A: deductions from the molecular analysis of the avi biosynthetic gene cluster of Streptomyces viridochromogenes Tü57 and production of new antibiotics. *Chemistry & biology*, 8(6):569–81, June 2001. ISSN 1074-5521.

[62] G Weitnauer, S Gaisser, L Kellenberger, P F Leadlay, und A Bechthold. Analysis of a C-methyltransferase gene (aviG1) involved in avilamycin biosynthesis in Streptomyces viridochromogenes Tü57 and complementation of a Saccharopolyspora erythraea eryBIII mutant by aviG1. *Microbiology (Reading, England)*, 148(Pt 2):373–9, February 2002. ISSN 1350-0872.

[63] S Omura, H Tanaka, Y Iwai, K Nishigaki, J Awaya, Y Takahashi, und R Masuma. A new antibiotic, setomimycin, produced by a strain of Streptomyces. *The Journal of antibiotics*, 31(11):1091–8, November 1978. ISSN 0021-8820.

[64] N Tsuji. Studies on julimycins. I. The structure of julimycin B-II. *Tetrahedron*, 24(4):1765–76, February 1968. ISSN 0040-4020.

[65] S Matsuura, O Shiratori, Y Harada, und K Katagiri. Antitumor activity of julimycin B-II. *The Journal of antibiotics*, 20(5):282–6, September 1967. ISSN 0021-8820.

[66] A L Staley und K L Rinehart. Spectomycins, new antibacterial compounds produced by Streptomyces spectabilis: isolation, structures, and biosynthesis. *The Journal of antibiotics*, 47(12):1425–1433, 1994. ISSN 00218820.

[67] Loubna Youssar, Björn Andreas Grüning, Anika Erxleben, Stefan Günther, und Wolfgang Hüttel. Genome sequence of the fungus Glarea lozoyensis: the first genome sequence of a species from the Helotiaceae family. *Eukaryotic cell*, 11(2):250, February 2012. ISSN 1535-9786. doi: 10.1128/EC.05302-11.

[68] P S Masurekar, J M Fountoulakis, T C Hallada, M S Sosa, und L Kaplan. Pneumocandins from Zalerion arboricola. II. Modification of product spectrum by mutation and medium manipulation. *The Journal of antibiotics*, 45(12): 1867–74, December 1992. ISSN 0021-8820.

[69] Torsten Seemann. Prokka: Rapid prokaryotic genome annotation. *Bioinformatics*, 30(14):2068–2069, 2014. ISSN 14602059. doi: 10.1093/bioinformatics/btu153.

[70] Ramy K Aziz, Daniela Bartels, Aaron A Best, Matthew DeJongh, Terrence Disz, Robert A Edwards, Kevin Formsma, Svetlana Gerdes, Elizabeth M Glass, Michael Kubal, Folker Meyer, Gary J Olsen, Robert Olson, Andrei L Osterman, Ross A Overbeek, Leslie K McNeil, Daniel Paarmann, Tobias Paczian, Bruce Parrello, Gordon D Pusch, Claudia Reich, Rick Stevens, Olga Vassieva, Veronika Vonstein, Andreas Wilke, und Olga Zagnitko. The RAST Server: rapid annotations using subsystems technology. *BMC genomics*, 9:75, 2008. ISSN 1471-2164. doi: 10.1186/1471-2164-9-75.

[71] Andrew C. Stewart, Brian Osborne, und Timothy D. Read. DIYA: A bacterial annotation pipeline for any genomics lab. *Bioinformatics*, 25(7):962–963, 2009. ISSN 13674803. doi: 10.1093/bioinformatics/btp097.

[72] Christian Senger, Björn A Grüning, Anika Erxleben, Kersten Döring, Hitesh Patel, Stephan Flemming, Irmgard Merfort, und Stefan Günther. Mining and evaluation of molecular relationships in literature. *Bioinformatics (Oxford, England)*, 28(5):709–14, March 2012. ISSN 1367-4811. doi: 10.1093/bioinformatics/bts026.

[73] F B Rogers. Medical subject headings. *Bulletin of the Medical Library Association*, 51:114–6, January 1963. ISSN 0025-7338.

[74] Kersten Döring, Björn Grüning, Kiran K Telukunta, und Stefan Günther. PubMed2Go: A Framework for Developing Text Mining Applications. *BMC bioinformatics*, (submitted), 2015.

[75] Yanli Wang, Jewen Xiao, Tugba O Suzek, Jian Zhang, Jiyao Wang, und Stephen H Bryant. PubChem: a public information system for analyzing bioactivities of small molecules. *Nucleic acids research*, 37(Web Server issue): W623–33, July 2009. ISSN 1362-4962. doi: 10.1093/nar/gkp456.

[76] Activities at the Universal Protein Resource (UniProt). *Nucleic acids research*, 42(Database issue):D191–8, January 2014. ISSN 1362-4962. doi: 10.1093/nar/gkt1140.

[77] Rachael P Huntley, Midori A Harris, Yasmin Alam-Faruque, Judith A Blake, Seth Carbon, Heiko Dietze, Emily C Dimmer, Rebecca E Foulger, David P Hill, Varsha K Khodiyar, Antonia Lock, Jane Lomax, Ruth C Lovering, Prudence Mutowo-Meullenet, Tony Sawford, Kimberly Van Auken, Valerie Wood, und Christopher J Mungall. A method for increasing expressivity of Gene Ontology annotations using a compositional approach. *BMC bioinformatics*, 15(1):155, January 2014. ISSN 1471-2105. doi: 10.1186/1471-2105-15-155.

[78] M Ashburner, C A Ball, J A Blake, D Botstein, H Butler, J M Cherry, A P Davis, K Dolinski, S S Dwight, J T Eppig, M A Harris, D P Hill, L Issel-Tarver, A Kasarskis, S Lewis, J C Matese, J E Richardson, M Ringwald, G M Rubin, und G Sherlock. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature genetics*, 25(1):25–9, May 2000. ISSN 1061-4036. doi: 10.1038/75556.

[79] Dietrich Rebholz-Schuhmann, Miguel Arregui, Sylvain Gaudan, Harald Kirsch, und Antonio Jimeno. Text processing through Web services: calling Whatizit. *Bioinformatics (Oxford, England)*, 24(2):296–8, January 2008. ISSN 1367-4811. doi: 10.1093/bioinformatics/btm557.

[80] Kristina M Hettne, Rob H Stierum, Martijn J Schuemie, Peter J M Hendriksen, Bob J A Schijvenaars, Erik M van Mulligen, Jos Kleinjans, und Jan A Kors. A dictionary to identify small molecules and drugs in free text. *Bioinformatics (Oxford, England)*, 25(22):2983–91, November 2009. ISSN 1367-4811. doi: 10.1093/bioinformatics/btp535.

[81] Nikolai Hecker, Jessica Ahmed, Joachim Von Eichborn, Mathias Dunkel, Karel Macha, Andreas Eckert, Michael K. Gilson, Philip E. Bourne, und Robert Preissner. SuperTarget goes quantitative: Update on drug-target interactions. *Nucleic Acids Research*, 40(D1), 2012. ISSN 03051048. doi: 10.1093/nar/gkr912.

[82] M Kanehisa und S Goto. KEGG: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1):27–30, January 2000. ISSN 0305-1048.

[83] Minoru Kanehisa, Susumu Goto, Yoko Sato, Masayuki Kawashima, Miho Furumichi, und Mao Tanabe. Data, information, knowledge and principle: back to metabolism in KEGG. *Nucleic acids research*, 42(Database issue):D199–205, January 2014. ISSN 1362-4962. doi: 10.1093/nar/gkt1076.

[84] Thomas Kelder, Martijn P van Iersel, Kristina Hanspers, Martina Kutmon, Bruce R Conklin, Chris T Evelo, und Alexander R Pico. WikiPathways: building research communities on biological pathways. *Nucleic acids research*, 40(Database issue):D1301–7, January 2012. ISSN 1362-4962. doi: 10.1093/nar/gkr1074.

[85] Frank K. Brown. Chemoinformatics: What is it and How does it Impact Drug Discovery. pages 375–384. 1998. doi: 10.1016/S0065-7743(08)61100-8.

[86] Frank Brown. Editorial opinion: chemoinformatics - a ten year update., 2005. ISSN 1367-6733.

[87] Jean-Louis Reymond, Ruud van Deursen, Lorenz C. Blum, und Lars Ruddigkeit. Chemical space as a source for new drugs. *MedChemComm*, 1(1):30, 2010. ISSN 2040-2503. doi: 10.1039/c0md00020e.

[88] Lars Ruddigkeit, Ruud Van Deursen, Lorenz C. Blum, und Jean Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. *Journal of Chemical Information and Modeling*, 52(11):2864–2875, 2012. ISSN 15499596. doi: 10.1021/ci300415d.

[89] Ricardo Macarron, Martyn N Banks, Dejan Bojanic, David J Burns, Dragan A Cirovic, Tina Garyantes, Darren V S Green, Robert P Hertzberg, William P Janzen, Jeff W Paslay, Ulrich Schopfer, und G Sitta Sittampalam. Impact of high-throughput screening in biomedical research. *Nature reviews. Drug discovery*, 10(3):188–195, 2011. ISSN 1474-1776. doi: 10.1038/nrd3368.

[90] Alan L Harvey. Natural products in drug discovery. *Drug discovery today*, 13(19-20):894–901, October 2008. ISSN 1359-6446. doi: 10.1016/j.drudis.2008.07.004.

[91] Christopher a Lipinski. Drug-like properties and the causes of poor solubility and poor permeability. *Journal of Pharmacological and Toxicological Methods*, 44(1):235–249, 2000. ISSN 10568719. doi: 10.1016/S1056-8719(00)00107-6.

[92] Kalai Vanii Jayaseelan, Pablo Moreno, Andreas Truszkowski, Peter Ertl, und Christoph Steinbeck. Natural product-likeness score revisited: an open-source, open-data implementation. *BMC bioinformatics*, 13:106, January 2012. ISSN 1471-2105. doi: 10.1186/1471-2105-13-106.

[93] Peter Ertl, Silvio Roggo, und Ansgar Schuffenhauer. Natural product-likeness score and its application for prioritization of compound libraries. *Journal of chemical information and modeling*, 48(1):68–74, January 2008. ISSN 1549-9596. doi: 10.1021/ci700286x.

[94] Daniel Blankenberg, Gregory Von Kuster, Emil Bouvier, Dannon Baker, Enis Afgan, Nicholas Stoler, James Taylor, und Anton Nekrutenko. Dissemination of scientific software with Galaxy ToolShed. *Genome biology*, 15(2): 403, 2014. ISSN 1465-6914. doi: 10.1186/gb4161.

[95] Grüning A, Björn, Smith Cameron, Houwaart Torsten, Soranzo Nicola, Rasche Eric. URL `https://github.com/bgruening/galaxytools`.

[96] Igor V Filippov und Marc C Nicklaus. Optical structure recognition software to recover chemical information: OSRA, an open source solution. *Journal of chemical information and modeling*, 49(3):740–3, March 2009. ISSN 1549-9596. doi: 10.1021/ci800067r.

[97] Noel M O'Boyle, Michael Banck, Craig A James, Chris Morley, Tim Vandermeersch, und Geoffrey R Hutchison. Open Babel: An open chemical toolbox. *Journal of cheminformatics*, 3:33, January 2011. ISSN 1758-2946. doi: 10.1186/1758-2946-3-33.

[98] Noel M O'Boyle, Chris Morley, und Geoffrey R Hutchison. Pybel: a Python wrapper for the OpenBabel cheminformatics toolkit. *Chemistry Central journal*, 2:5, January 2008. ISSN 1752-153X. doi: 10.1186/1752-153X-2-5.

[99] Patrick Bultinck, W. Langenaeker, P. Lahorte, F. De Proft, P. Geerlings, C. Van Alsenoy, und J. P. Tollenaere. The electronegativity equalization method II: Applicability of different atomic charge schemes. *Journal of Physical Chemistry A*, 106(34):7895–7901, 2002. ISSN 10895639. doi: 10.1021/jp020547v.

[100] Patrick Bultinck, Wilfried Langenaeker, Ramon Carbó-Dorca, und Jan P. Tollenaere. Fast calculation of quantum chemical molecular descriptors from the Electronegativity Equalization Method. In *Journal of Chemical Information and Computer Sciences*, volume 43, pages 422–428, 2003. ISBN 0095-2338. doi: 10.1021/ci0255883.

[101] Andrew Dalke. ChemFP. URL `http://chemfp.com/`.

[102] Noel M. O'Boyle, Tim Vandermeersch, Christopher J. Flynn, Anita R. Maguire, und Geoffrey R. Hutchison. Confab - Systematic generation of diverse low-energy conformers. *Journal of Cheminformatics*, 3(1), 2011. ISSN 17582946. doi: 10.1186/1758-2946-3-8.

[103] G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, und Andrew L Hopkins. Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90–8, February 2012. ISSN 1755-4349. doi: 10.1038/nchem.1243.

[104] Daniel M Lowe, Peter T Corbett, Peter Murray-Rust, und Robert C Glen. Chemical name to structure: OPSIN, an open source solution. *Journal of chemical information and modeling*, 51(3):739–53, March 2011. ISSN 1549-960X. doi: 10.1021/ci100384d.

[105] Anna Gaulton, Louisa J Bellis, A Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, und John P Overington. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic acids research*, 40(Database issue):D1100–7, January 2012. ISSN 1362-4962. doi: 10.1093/nar/gkr777.

[106] David S Wishart, Craig Knox, An Chi Guo, Savita Shrivastava, Murtaza Hassanali, Paul Stothard, Zhan Chang, und Jennifer Woolsey. DrugBank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic acids research*, 34(Database issue):D668–72, January 2006. ISSN 1362-4962. doi: 10.1093/nar/gkj067.

[107] Stephen R Heller, Alan McNaught, Igor Pletnev, Stephen Stein, und Dmitrii Tchekhovskoi. InChI, the IUPAC International Chemical Identifier. *Journal of Cheminformatics*, 7(1):23, December 2015. ISSN 1758-2946. doi: 10.1186/s13321-015-0068-4.

[108] Henri A. Favre und Warren H. Powell. *Nomenclature of Organic Chemistry: IUPAC Recommendations and Preferred Names 2013*. 2013. ISBN 0854041826.

[109] Christopher A. Lipinski, Franco Lombardo, Beryl W. Dominy, und Paul J. Feeney. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced drug delivery reviews*, 46(1-3):3–26, March 2001. ISSN 0169-409X. doi: 10.1016/S0169-409X(00)00129-0.

[110] SJ Teague, AM Davis, PD Leeson, und T Oprea. The Design of Leadlike Combinatorial Libraries. *Angewandte Chemie (International ed. in English)*, 38(24):3743–3748, December 1999. ISSN 1521-3773.

[111] Robin A E Carr, Miles Congreve, Christopher W Murray, und David C Rees. Fragment-based lead discovery: leads by design. *Drug discovery today*, 10(14):987–92, July 2005. ISSN 1359-6446. doi: 10.1016/S1359-6446(05)03511-7.

[112] W.Patrick Walters, Matthew T Stahl, und Mark A Murcko. Virtual screening—an overview. *Drug Discovery Today*, 3(4):160–178, April 1998. ISSN 13596446. doi: 10.1016/S1359-6446(97)01163-X.

[113] RDKit: Open-source cheminformatics. URL `https://github.com/rdkit/rdkit`.

[114] Hans Joachim Böhm, Alexander Flohr, und Martin Stahl. Scaffold hopping. *Drug Discovery Today: Technologies*, 1 (3):217–224, 2004. ISSN 17406749. doi: 10.1016/j.ddtec.2004.10.009.

[115] Steffen Renner und Gisbert Schneider. Scaffold-hopping potential of ligand-based similarity concepts. *ChemMedChem*, 1(2):181–185, 2006. ISSN 18607179. doi: 10.1002/cmdc.200500005.

[116] John Gómez, Leyla J. García, Gustavo A. Salazar, Jose Villaveces, Swanand Gore, Alexander García, Maria J. Martín, Guillaume Launay, Rafael Alcántara, Noemi Del-Toro, Marine Dumousseau, Sandra Orchard, Sameer Velankar, Henning Hermjakob, Chenggong Zong, Peipei Ping, Manuel Corpas, und Rafael C. Jiménez. BioJS: An open source JavaScript framework for biological data visualization. *Bioinformatics*, 29(8):1103–1104, 2013. ISSN 13674803. doi: 10.1093/bioinformatics/btt100.

[117] Guy Yachdav, Tatyana Goldberg, Sebastian Wilzbach, David Dao, Iris Shih, Saket Choudhary, Steve Crouch, Max Franz, Alexander García, Leyla J García, Björn A Grüning, Devasena Inupakutika, Ian Sillitoe, Anil S Thanki, Bruno Vieira, José M Villaveces, Maria V Schneider, Suzanna Lewis, Steve Pettifer, Burkhard Rost, und Manuel Corpas. Anatomy of BioJS, an open source community for the life sciences. *eLife*, 4, January 2015. ISSN 2050-084X. doi: 10.7554/eLife.07009.

[118] Robin Taylor. Simulation Analysis of Experimental Design Strategies for Screening Random Compounds as Potential New Drugs and Agrochemicals. *Journal of Chemical Information and Modeling*, 35(1):59–67, January 1995. ISSN 1549-9596. doi: 10.1021/ci00023a009.

[119] D. Butina. Unsupervised Data Base Clustering Based on Daylight's Fingerprint and Tanimoto Similarity: A Fast and Automated Way To Cluster Small and Large Data Sets. *Journal of Chemical Information and Modeling*, 39(4):747–750, July 1999. ISSN 1549-9596. doi: 10.1021/ci9803381.

[120] Fabian López-Vallejo, Marc A Giulianotti, Richard A Houghten, und José L Medina-Franco. Expanding the medicinally relevant chemical space with compound libraries. *Drug discovery today*, 17(13-14):718–26, July 2012. ISSN 1878-5832. doi: 10.1016/j.drudis.2012.04.001.

[121] Michael R Berthold, Nicolas Cebron, Fabian Dill, Thomas R Gabriel, Tobias Kotter, Thorsten Meinl, Peter Ohl, Christoph Sieb, Kilian Thiel, und Bernd Wiswedel. Knime. *Web*, pages 1–8, 2007. ISSN 19310145. doi: 10.1007/978-3-540-78246-9.

[122] Wendy A. Warr. Scientific workflow systems: Pipeline Pilot and KNIME. *Journal of Computer-Aided Molecular Design*, 26(7):801–804, July 2012. ISSN 0920-654X. doi: 10.1007/s10822-012-9577-7.

[123] Björn A Grüning, Christian Senger, Anika Erxleben, Stephan Flemming, und Stefan Günther. Compounds In Literature (CIL): screening for compounds and relatives in PubMed. *Bioinformatics (Oxford, England)*, 27(9):1341–2, May 2011. ISSN 1367-4811. doi: 10.1093/bioinformatics/btr130.

[124] Corinna Kolářik, Roman Klinger, Christoph M. Friedrich, Martin Hofmann-Apitius, und Juliane Fluck. Chemical Names: Terminological Resources and Corpora Annotation. In *Workshop on Building and evaluating resources for biomedical text mining (6th edition of the Language Resources and Evaluation Conference)*, 2008.

[125] Junguk Hur, Adam D Schuyler, David J States, und Eva L Feldman. SciMiner: web-based literature mining tool for target identification and functional enrichment analysis. *Bioinformatics (Oxford, England)*, 25(6):838–40, March 2009. ISSN 1367-4811. doi: 10.1093/bioinformatics/btp049.

[126] Dietrich Rebholz-Schuhmann, Harald Kirsch, Miguel Arregui, Sylvain Gaudan, Mark Riethoven, und Peter Stoehr. EBIMed–text crunching to gather facts for proteins from Medline. *Bioinformatics (Oxford, England)*, 23(2):e237–44, January 2007. ISSN 1367-4811. doi: 10.1093/bioinformatics/btl302.

[127] Michael Kuhn, Damian Szklarczyk, Andrea Franceschini, Monica Campillos, Christian von Mering, Lars Juhl Jensen, Andreas Beyer, und Peer Bork. STITCH 2: an interaction network database for small molecules and proteins. *Nucleic acids research*, 38(Database issue):D552–6, January 2010. ISSN 1362-4962. doi: 10.1093/nar/gkp937.

[128] Qian Zhu, Michael S Lajiness, Ying Ding, und David J Wild. WENDI: A tool for finding non-obvious relationships between compounds and biological properties, genes, diseases and scholarly publications. *Journal of cheminformatics*, 2:6, January 2010. ISSN 1758-2946. doi: 10.1186/1758-2946-2-6.

[129] Xavier Lucas, Christian Senger, Anika Erxleben, Björn A Grüning, Kersten Döring, Johannes Mosch, Stephan Flemming, und Stefan Günther. StreptomeDB: a resource for natural compounds isolated from Streptomyces species. *Nucleic acids research*, 41(Database issue):D1130–6, January 2013. ISSN 1362-4962. doi: 10.1093/nar/gks1253.

[130] P C Trussell, C O Fulton, und G A Grant. Two Antibiotics Produced by a Streptomyces. *Journal of bacteriology*, 53(6):769–80, June 1947. ISSN 0021-9193.

[131] S A Waksman und H B Woodruff. Actinomyces antibioticus, a New Soil Organism Antagonistic to Pathogenic and Non-pathogenic Bacteria. *Journal of bacteriology*, 42 (2):231–49, August 1941. ISSN 0021-9193.

[132] S A Waksman und H B Woodruff. Selective Antibiotic Action of Various Substances of Microbial Origin. *Journal of bacteriology*, 44(3):373–84, September 1942. ISSN 0021-9193.

[133] A Schatz und S A Waksman. Strain Specificity and Production of Antibiotic Substances: IV. Variations Among Actionomycetes, with Special Reference to Actinomyces Griseus. *Proceedings of the National Academy of Sciences of the United States of America*, 31(5):129–37, May 1945. ISSN 0027-8424.

[134] Farit Mochamad Afendi, Taketo Okada, Mami Yamazaki, Aki Hirai-Morita, Yukiko Nakamura, Kensuke Nakamura, Shun Ikeda, Hiroki Takahashi, Md Altaf-Ul-Amin, Latifah K Darusman, Kazuki Saito, und Shigehiko Kanaya. KNApSAcK family databases: integrated metabolite-plant species databases for multifaceted plant research. *Plant & cell physiology*, 53(2):e1, February 2012. ISSN 1471-9053. doi: 10.1093/pcp/pcr165.

[135] X Q Lewell, D B Judd, S P Watson, und M M Hann. RECAP–retrosynthetic combinatorial analysis procedure: a powerful new technique for identifying privileged molecular fragments with useful applications in combinatorial chemistry. *Journal of chemical information and computer sciences*, 38(3):511–22, 1998. ISSN 0095-2338.

[136] Varun Khanna und Shoba Ranganathan. Structural diversity of biologically interesting datasets: a scaffold analysis approach. *Journal of cheminformatics*, 3(1):30, January 2011. ISSN 1758-2946. doi: 10.1186/1758-2946-3-30.

[137] Christopher Lipinski und Andrew Hopkins. Navigating chemical space for biology and medicine. *Nature*, 432 (7019):855–61, December 2004. ISSN 1476-4687. doi: 10.1038/nature03193.

[138] Gene H. Hur, Christopher R. Vickery, und Michael D. Burkart. Explorations of catalytic domains in non-ribosomal peptide synthetase enzymology, 2012. ISSN 0265-0568.

[139] Liangcheng Du, César Sánchez, Mei Chen, Daniel J. Edwards, und Ben Shen. The biosynthetic gene cluster for the antitumor drug bleomycin from Streptomyces verticillus ATCC15003 supporting functional interactions between nonribosomal peptide synthetases and a polyketide synthase. *Chemistry and Biology*, 7(8):623–642, 2000. ISSN 10745521. doi: 10.1016/S1074-5521(00)00011-9.

[140] Sascha Doekel, M.-F. Coeffet-Le Gal, J.-Q. Gu, Min Chu, Richard H. Baltz, und Paul Brian. Non-ribosomal peptide synthetase module fusions to produce derivatives of daptomycin in Streptomyces roseosporus. *Microbiology*, 154(9):2872–2880, September 2008. ISSN 1350-0872. doi: 10.1099/mic.0.2008/020685-0.

[141] Xihou Yin. The enduracidin biosynthetic gene cluster from Streptomyces fungicidicus. *Microbiology*, 152 (10):2969–2983, October 2006. ISSN 1350-0872. doi: 10.1099/mic.0.29043-0.

[142] Frank E Koehn und Guy T Carter. The evolving role of natural products in drug discovery. *Nature reviews. Drug discovery*, 4(3):206–220, 2005. ISSN 1474-1776. doi: 10.1038/nrd1657.

[143] János Bérdy. Bioactive microbial metabolites. *The Journal of antibiotics*, 58(1):1–26, January 2005. ISSN 0021-8820. doi: 10.1038/ja.2005.1.

[144] Olivier Taboureau, Jonathan B. Baell, Juan Fernández-Recio, und Bruno O. Villoutreix. Established and emerging trends in computational drug discovery in the structural genomics era, 2012. ISSN 10745521.

[145] Tiejun Cheng, Qingliang Li, Zhigang Zhou, Yanli Wang, und Stephen H. Bryant. Structure-Based Virtual Screening for Drug Discovery: a Problem-Centric Review, 2012. ISSN 1550-7416.

[146] William L. Jorgensen. Efficient drug lead discovery and optimization. *Accounts of Chemical Research*, 42(6):724–733, 2009. ISSN 00014842. doi: 10.1021/ar800236t.

[147] Teresa K. Atwood, Erik Bongcam-Rudloff, Michelle E. Brazas, Manuel Corpas, Pascale Gaudet, Fran Lewitter, Nicola Mulder, Patricia M. Palagi, Maria Victoria Schneider, und Celia W. G. van Gelder. GOBLET: The Global Organisation for Bioinformatics Learning, Education and Training. *PLOS Computational Biology*, 11(4):e1004143, April 2015. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1004143.

[148] Ralf Gilsbach, Sebastian Preissl, Björn A. Grüning, Tilman Schnick, Lukas Burger, Vladimir Benes, Andreas Würch, Ulrike Bönisch, Stefan Günther, Rolf Backofen, Bernd K. Fleischmann, Dirk Schübeler, und Lutz Hein. Dynamic DNA methylation orchestrates cardiomyocyte development, maturation and disease. *Nature communications*, 5:5288, January 2014. ISSN 2041-1723. doi: 10.1038/ncomms6288.

[149] Sebastian Preissl, Martin Schwaderer, Alexandra Raulf, Michael Hesse, Björn A Grüning, Claudia Köbele, Rolf Backofen, Bernd K Fleischmann, Lutz Hein, und Ralf Gilsbach. Deciphering the Epigenetic Code of Cardiac Myocyte Transcription. *Circulation Research*, June 2015. ISSN 0009-7330. doi: 10.1161/CIRCRESAHA.115.306337.

[150] Y. Tang, E. Bouvier, C. K. Kwok, Y. Ding, A. Nekrutenko, P. C. Bevilacqua, und S. M. Assmann. StructureFold: Genome-wide RNA secondary structure mapping and reconstruction in vivo. *Bioinformatics*, 2015. ISSN 1367-4803. doi: 10.1093/bioinformatics/btv213.

[151] Deborah Roidl, Nicole Hellbach, Patrick Bovio, Alejandro Villarreal, Stefanie Heidrich, Sigrun Nestel, Björn A. Grüning, Ulrike Bönisch, und Tanja Vogel. DOT1L activity promotes proliferation and protects cortical neural stem cells from activation of ATF4-DDIT3-mediated ER stress in vitro. *STEM CELLS*, August 2015. ISSN 10665099. doi: 10.1002/stem.2187.

[152] Hitesh Patel, Björn A Grüning, Stefan Günther, und Irmgard Merfort. PyWATER: a PyMOL plug-in to find conserved water molecules in proteins by clustering. *Bioinformatics (Oxford, England)*, July 2014. ISSN 1367-4811. doi: 10.1093/bioinformatics/btu424.

[153] Fidel Ramírez, Friederike Dündar, Sarah Diehl, Björn A Grüning, und Thomas Manke. deepTools: a flexible platform for exploring deep-sequencing data. *Nucleic acids research*, 42(Web Server issue):W187–91, July 2014. ISSN 1362-4962. doi: 10.1093/nar/gku365.

[154] Desirée Schubert, Claudia Bode, Rupert Kenefeck, Tie Zheng Hou, James B Wing, Alan Kennedy, Alla Bulashevska, Britt-Sabina Petersen, Alejandro A Schäffer, Björn A Grüning, Susanne Unger, Natalie Frede, Ulrich Baumann, Torsten Witte, Reinhold E Schmidt, Gregor Dueckers, Tim Niehues, Suranjith Seneviratne, Maria Kanariou, Carsten Speckmann, Stephan Ehl, Anne Rensing-Ehl, Klaus Warnatz, Mirzokhid Rakhmanov, Robert Thimme, Peter Hasselblatt, Florian Emmerich, Toni Cathomen, Rolf Backofen, Paul Fisch, Maximilian Seidl, Annette May, Annette Schmitt-Graeff, Shinji Ikemizu, Ulrich Salzer, Andre Franke, Shimon Sakaguchi, Lucy S K Walker, David M Sansom, und Bodo Grimbacher. Autosomal dominant immune dysregulation syndrome in humans with CTLA4 mutations. *Nature medicine*, 20(12):1410–6, December 2014. ISSN 1546-170X. doi: 10.1038/nm.3746.

[155] Loubna Youssar, Björn Andreas Grüning, Stefan Günther, und Wolfgang Hüttel. Characterization and phylogenetic analysis of the mitochondrial genome of Glarea lozoyensis indicates high diversity within the order Helotiales. *PloS one*, 8(9):e74792, January 2013. ISSN 1932-6203. doi: 10.1371/journal.pone.0074792.

[156] Jan Bieschke, Martin Herbst, Thomas Wiglenda, Ralf P Friedrich, Annett Boeddrich, Franziska Schiele, Daniela Kleckers, Juan Miguel Lopez del Amo, Björn A Grüning, Qinwen Wang, Michael R Schmidt, Rudi Lurz, Roger Anwyl, Sigrid Schnoegl, Marcus Fändrich, Ronald F Frank, Bernd Reif, Stefan Günther, Dominic M Walsh, und Erich E Wanker. Small-molecule conversion of toxic oligomers to nontoxic β-sheet-rich amyloid fibrils. *Nature chemical biology*, 8(1):93–101, January 2012. ISSN 1552-4469. doi: 10.1038/nchembio.719.

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Stellen der Arbeit, die wörtlich oder sinngemäß aus Veröffentlichungen oder aus anderweitigen fremden Äußerungen entnommen wurden, sind als solche kenntlich gemacht. Ferner erkläre ich, dass die Arbeit noch nicht in einem anderen Studiengang als Prüfungsleistung verwendet wurde.

Freiburg, 01.09.2015