1 Einleitung

1.1 Was Sie über Bioinformatik wissen – sollten

Bioinformatik ist ein dehnbarer Begriff. Er spannt einen Bogen von der Entwicklung von Algorithmen – was nicht Thema dieses Buches ist – bis hin zur reinen Anwendung von Software zur Datenanalyse – was dem Thema dieses Buches schon näher kommt. Lassen Sie mich dies anhand zweier Begegnungen verdeutlichen:

Neulich traf ich einen Krebsforscher. Mit ihm sprach ich unter anderem über die Fortschritte bei der Genomanalyse und welchen Beitrag sie für seine Arbeit leistet. Pfiffig wie er war, enthüllte er erst im Verlauf unseres Gespräches, dass er Krebse erforscht – nicht Tumore. Die Genomanalysen, die seine Arbeit betreffen, gehören in den Bereich des DNA-Barcodings, also der Identifizierung von biologischen Arten anhand eines eindeutigen genetischen Markers. Solche Marker zu finden, ist eine typische bioinformatische Aufgabe und zeigt, wie tief die Bioinformatik in klassische Disziplinen der Lebenswissenschaften eingedrungen ist.

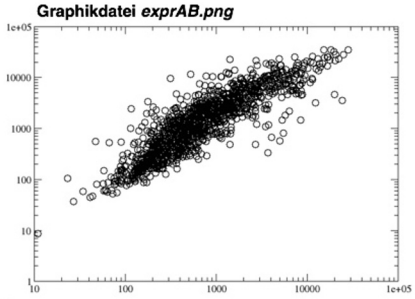
Eine weitere Begegnung: Mit einem Professor der Genetik hatte ich das Vergnügen, eine zweiwöchige Sommerakademie zur Synthetischen Biologie zu leiten. Privat sprachen wir auch über unsere Forschung. Er arbeitet unter anderem am Ribosomal Readthrough, einem Phänomen, bei dem Ribosomen während der Translation der mRNA in ein Protein gelegentlich das Stopcodon überlesen. Er hatte eine riesige Excel-Datei mit RNA-Sequenzen von Transkripten, in denen er nach bestimmten Sequenzmotiven suchen wollte. Ich zeigte ihm, wie einfach das mit den Linux-Kommandozeilenprogrammen **Sed** und **AWK** geht. Innerhalb weniger Stunden konnte er seine Daten selbstständig analysieren und war begeistert.

Bei beiden Begegnungen war Bioinformatik ein Thema, aber kein zentrales, sondern das Mittel zum Zweck. Und genau in diese Bioinformatik, also die Bioinformatik als Mittel zum Zweck, möchte ich Sie einführen. Dabei greife ich auf meine persönliche 15-jährige Erfahrung als forschender und lehrender Biologe zurück, der experimentell arbeitet und dabei große Datenmengen erzeugt. Zunächst waren dies überwiegend Genexpressionsdaten, die mit

Datenbank *compbiol* mit Tabelle *exprA_exprB*

gene	exprA	exprB	len	gc
all0001	1109	2202	1230	508
all0002	1094	1373	738	298
all0004	1969	7154	948	454
all0005	4893	7769	1521	708
all0006	10175	15090	552	246
all0007	4971	6861	564	264
all0008	6278	9638	492	223
all0010	15529	24388	756	325
...	...	...	...	...

Graphikdatei *exprAB.png*



Kommandozeilenbefehl

```
1 $mysql -u awkologist -pawkology compbiol -B -e 'SELECT * FROM exprA_exprB' |
2   sed '1d' |
3   awk '{print $2"\t"$3}' |
4   xmgrace -hdevice PNG -hardcopy -printfile "exprAB,png" -pipe -log xy -pexec
     "S0 line type 0, S0 SYMBOL 1"
```

Abbildung 1.1 Visualisierung von Genexpressionsdaten. Über die Kommandozeile kann mit einem zugegebenermaßen langen Befehl eine Datenbankabfrage gestellt (Zeile 1), die Daten formatiert (Zeilen 2+3) und anschließend geplottet (Zeile 4) werden.

DNA-Mikroarrays erzeugt wurden. Diese Daten mussten geplottet (Abb. 1.1) und annotiert (Abb. 1.2) werden. Eine hohe Zahl steht für eine hohe Aktivität des Gens. Später kamen Sequenzdaten aus NG (*next generation*) DNA- und RNA-Sequenzierungen dazu. Eine Sequenzierung erzeugt dabei rund 40 Millionen Sequenzen, die in einer etwa 2-3 GB großen Datei im FastQ-Format gespeichert sind. Die Verarbeitung dieser Daten stößt an die Grenzen von MS Excel und des Notepads.

Wichtiger Hinweis

Die Bioinformatik, die ich Ihnen zeigen möchte, ist eine angewandte Bioinformatik. Sie ist Mittel zum Zweck. Es geht mir nicht um die Herleitung oder gar Entwicklung von Algorithmen, sondern um die Verarbeitung und Analyse von Daten aus den Lebenswissenschaften. Und – dies ist kein Theoriebuch. Sie müssen die Tasten schon selbst tanzen lassen.

Tab.: *annotation.tab*

gene	function	metabolism
alr2938	iron superoxide dismutase	Detoxification
alr4392	nitrogen-responsive regulator	Nitrogen assimilation
alr4851	preprotein translocase subunit	Protein and peptide secretion
alr3395	adenylosuccinate lyase	Purine biosynthesis
alr1207	uridylate kinase	Pyrimidine biosynthesis
alr5000	CTP synthetase	Pyrimidine biosynthesis
all3556	succinate-dehydrogenase	TCA cycle
....	....	....

Tab.: *expression.tab*

gene	expr_level
alr1207	8303
alr2938	10323
alr3395	1432
all3556	8043
alr4392	729
alr4851	633
alr5000	5732
....	....

Finde alle Gene, die am Purin- oder Pyrimidinmetabolismus beteiligt sind und deren Expressionswert unter 5000 liegt, und gebe deren Funktion und Expressionswert aus.

```
mysql> SELECT a.gene, a.function, e.expr_value FROM annotation AS a, expression AS e
    -> WHERE a.gene = e.gene AND a.metabolism REGEXP "(Purine|Pyrimidine) biosynthesis" AND
    -> e.expr_value < 5000;

+---------+-----------------------+------------+
| gene    | function              | expr_value |
+---------+-----------------------+------------+
| alr3395 | adenylosuccinate lyase |       1432 |

....

mysql>
```

Abbildung 1.2 Mit dem relationalen Dankenbanksystem MariaDB (oder MySQL, siehe Abschnitt 10.2) können Daten aus verschieden Tabellen kombiniert werden.

1.1.1 Bioinformatik heißt mit Computern »sprechen«

Was haben »Montagsmaler« und »Activity« mit Bioinformatik zu tun? Bis 1996 gab es die Sendung Montagsmaler. Ein Begriff musste gemalt und von den Gruppenpartnern geraten werden. Bei dem Gesellschaftsspiel Activity muss ein pantomimisch vorgestellter Begriff erraten werden. Beide Vorgehensweisen sind im weiteren Sinn graphisch, ebenso wie Powerpoint. Aber wie viele Powerpoint-Präsentation sind zu textlastig? Zu viele – und es gibt einen Stapel Ratgeber darüber, wie man dies vermeidet. Und warum sind diese Präsentationen zu textlastig? Weil es häufig leichter fällt, einen Sachverhalten zu beschreiben, anstatt ihn zu skizzieren.

In der Bioinformatik wird dem Computer über Programmiersprachen beschrieben, wie er Daten verarbeitet soll. Zum Beispiel: nehme aus der Datei *seqs.fasta* alle Sequenzen, die mit einem Startcodon (ATG) beginnen, im Leseraster mindestens ein Cystein codieren und mit einem Stopcodon (TAA, TGA, TAG) enden → markiere diese offenen Leseraster → markiere die Cysteincodons → zeige das Ergebnis an. Dies ist in Abb. 1.3 dargestellt. Der eigentliche Befehl steht in der ersten Zeile des Terminals. Wenn Sie das Kapitel 3 durchgearbeitet haben, werden Sie diesen Befehl auch verstehen.

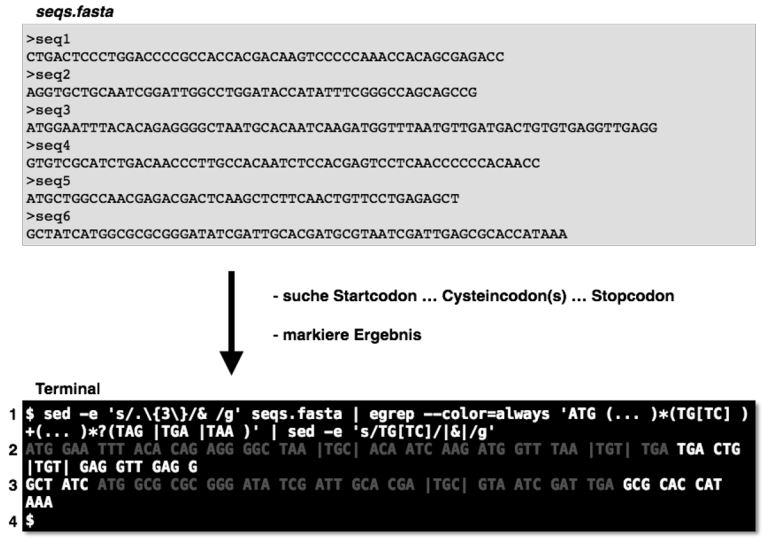


Abbildung 1.3 Der Terminal ist ein wichtiges Element der Bioinformatik, da sich komplexe Arbeitsanweisungen leichter als Text denn per »zeige-und-klicke« formulieren lassen.

Sie sehen also, dass Sie mit dem Terminal ein Powertool zu Verfügung haben, mit dem Sie sehr effizient komplexe Prozessierungen von Daten vornehmen können.

1.1.2 Kleine Geschichte der Bioinformatik

Historisch betrachtet sind die Entstehung der Chemo- und Bioinformatik die konsequente Antwort auf die Entwicklungen der Molekularbiologie. Die Sequenzierung des ersten Proteins durch Frederick Sanger 1953 und die erste Kristallstrukturanalyse eines Proteins durch Max Perutz und John Kendrew 1960 legten die Datengrundlage. John Kendrew nutzte zur Berechnung der Myoglobinstruktur aus Röntgenbeugungsdaten den ersten europäischen Computer namens ESDAC (*electronic delay storage automatic calculator*), der 1949 in Cambridge/UK seinen Betrieb aufnahm. Er war damit einer der ersten Chemoinformatiker.
Unix, als Vorläufer von Linux, war in gewisser Weise von Anfang an auch ein akademisches Betriebssystem, insbesondere seit der Weiterentwicklung von Unix an der Universität von Berkeley in Kalifornien/USA. Die Unix-Variante

BSD (Berkeley Software Distribution) ist übrigens die Basis für MacOSX und iOS. Dementsprechend wurde Unix von Anfang an insbesondere in den Naturwissenschaften angewendet. So gibt es seit den sechziger Jahren Publikationen aus den Bereichen Chemoinformatik und Bioinformatik. Voraussetzung für diese Entwicklung war zum einen der Zugang zu einem leistungsfähigen Computer, zum Beispiel dem IBM 7090 Computer, der auch die amerikanischen Mercury and Gemini Raumflüge unterstützte. Mit UNIVAC I (*universal automatic computer*) von John William Mauchly und John Presper Eckert war 1951 in den USA der erste kommerziell vertriebene Computer verfügbar. Zum anderen bedurfte es einer verständlichen Programmiersprache. John Backus von IBM entwickelte 1956 mit Fortran (*formula translation*) die erste Compiler-Programmiersprache. So wurden chemische beziehungsweise biologische Daten informatisch »fassbar«.
Fortran war auch das Werkzeug von Margaret Oakley Dayhoff aus den USA, einer Pionierin der Bioinformatik. Eines ihrer ersten Projekte war Anfang der 1960er Jahre die Erstellung eines Fortran-Programms, um aus Sequenzfragmenten eines Proteins seine vollständige Sequenz zu bestimmen (Dayhoff, 1964). Dieselbe Aufgabe, allerdings in unvergleichbar größerem Ausmaß und auf DNA-Sequenzen bezogen, musste das Team um Craig Venter bei der Etablierung des *whole-genome shutgun sequencing* lösen. Diese Methode zur Sequenzierung vollständiger Genome aus geschredderter genomischer DNA fand sowohl bei der Sequenzierung des ersten Genoms eines freilebenden Bakteriums (Fleischmann *et al.*, 1995) als auch bei der Sequenzierung des humanen Genoms durch Celera Genomics (Venter *et al.*, 2001) Anwendung.
In den 1960er Jahren entstanden auch die ersten Algorithmen zur Analyse von Proteinsequenzen (Dayhoff und Ledley, 1962), der Modellierung von Proteinstrukturdaten (Levinthal, 1966) und der Rekonstruktion der Evolution von Organismen aus Proteinsequenzen (Fitch und Margoliash, 1967).
In den 1970er Jahren untersuchte Margaret Oakley Dayhoff ausgehend von einigen hundert Proteinsequenzen die Häufigkeit des Aminosäureaustausches bei verwandten Proteinen mit statistischen Methoden. Daraus entwickelten sich die sogenannten Substitutionsmatrizen, die bis heute von großer praktischer Bedeutung in der Sequenzanalyse und -suche in Datenbanken sind. Im Jahre 1977 publizierten Alan Maxam und Walter Gilbert sowie Frederick Sanger unabhängig voneinander verschiedene Methoden zur DNA-Sequenzierung. Auf der deutlich eleganteren Methode von Sanger basieren die heutigen Verfahren zur automatisierten DNA-Sequenzanalyse. Zwei Jahre später brachte die Firma Oracle die erste kommerzielle Datenbanksoftware auf

den Markt, und 1979 entstand mit dem von Walter Goad entwickelten Prototyp von GenBank die erste öffentliche Gendatenbank. Um in dieser Datenbank Sequenzen zu finden, die ähnlich zu einer vorliegenden Sequenz sind, wurde ein zuvor von Saul Needleman und Christian Wunsch (Needleman und Wunsch, 1970) entwickelter und von Temple Smith und Michael Waterman (Smith und Waterman, 1981) modifizierter Alignment-Algorithmus verwendet, der in die BLAST (*basic local alignment software tool*) Software mündete (Altschul *et al.*, 1990). BLAST wird heute von annähernd jedem Molekularbiologen – und von uns in den Kapiteln 5, 6 und 7 – verwendet. Da ähnliche Proteinsequenzen oft auf Proteine mit ähnlichen Funktionen hinweisen, kommt der BLAST-Suche in Datenbanken eine große praktische Bedeutung zu.
Molecular Design Ltd. (chemische Datenbanken, gegründet 1978), Health Design Inc. (toxikologische Vorhersagen, gegründet 1978), Tripos Inc. (*Molecular Modeling* und *Drug Design*, gegründet 1979) und IntelliGENETICS (DNA- und Proteinsequenz-Analyse, gegründet 1980) waren Pioniere bei der marktwirtschaftlichen Anwendung von Computern (mit Unix) im Bereich der Chemie und Biochemie. Es ist schon erstaunlich: die Grundlage zu dem, was heute als bioinformatische Revolution gefeiert wird, ist schon vor über 30 Jahren gelegt worden.
Allein dieser kurze und unvollständige Rückblick auf die ersten Stunden der Chemo- und Bioinformatik zeigt, dass epochale Fortschritte sowohl in der Molekularbiologie als auch in der Computer- und Informationstechnologie zu einer überaus fruchtbaren Synthese führten.

1.1.3 Bioinformatik heute

Für viele Lebenswissenschaftler ist der Computer (scheinbar) wichtiger geworden als das Experiment. Egal, ob man Wald- und Wiesenbiologe ist, in der medizinischen Forschung arbeitet oder die Stammesgeschichte einer Tiergruppe nachvollziehen möchte: immer werden Daten mit dem Computer erfasst und ausgewertet. Die Entwicklung computergesteuerter Messgeräte hat diese Entwicklung massiv beschleunigt. Die Auswertung erfolgt häufig mit spezieller Software, die entweder mit dem Messinstrument mitgeliefert oder aber von Kollegen aus aller Welt entwickelt und bereitgestellt wird. Software von den Herstellern ist meistens mit einer benutzerfreundlichen graphischen Oberfläche ausgestattet. Das ist schön, wenn man nach Schema F verfährt. Häufig berücksichtigt diese Software aber weder die neuesten Erkenntnisse bei der Datenanlyse, noch können diese eigenhändig integriert werden. Die

aktuellsten Umsetzungen von neuen Algorithmen und Verfahren sind wiederum oft kommandozeilenbasiert – das bedeutet, diese laufen nur im Terminal. Wenn Sie mit dem Terminal umgehen können, öffnet sich eine neue Welt an Möglichkeiten. Und dies gilt nicht nur für den Linux/MacOSX-Terminal, sondern auch für das kommandozeilenbasierte R, das für die numerische Datenanalyse und -visualisierung besonders geeignet ist (siehe Kapitel 11).

Warnung

Bioinformatik und Excel passen nicht gut zusammen. Die meisten Lebenswissenschaftler arbeiten mit Excel. Das ist okay, solange die Datenmengen nicht zu groß sind und man weiß, was man tut, beziehungsweise Excel weiß, was man will. Ein Beispiel: das Protein Sept7 (ein Septin) codiert für ein Protein, das u.a. am Aufbau des Cytoskeletts beteiligt ist. Öffnen Sie einmal Excel und geben Sie den Gennamen *sept7* ein. Ja, versuchen Sie es einfach einmal. – Und? Mein Excel (MS Excel für Mac 2011, Version 14.4.8) macht daraus das Protein Sept-07, womit allerdings der 1. September 2007 gemeint ist (Abb. 1.4). – Man muss wissen, das Gennamen im Allgemeinen kursiv und klein geschrieben werden, Proteine dagegen normal gesetzt sind und mit Großbuchstben beginnen. – Okay, dann eben die Zelle mit Sept-07 rechtsklicken → Zellen formatieren ... → Text → OK. Huch: jetzt steht da 39326!? Dies ist die Anzahl der Tage zwischen dem 1. Januar 1900 und dem 1. September 2007. Nur wenn man die Zellenformatierung vor der Eingabe auf *Text* einstellt, wird der Genname korrekt eingetragen. Diese Besonderheit von Excel führte dazu, dass Sept-07 als Genname in die renommierte GenBank Einzug hielt (Zeeberg *et al.*, 2004) und niemand das Gen fand. Erst später wurde der Fehler bemerkt und korrigiert.

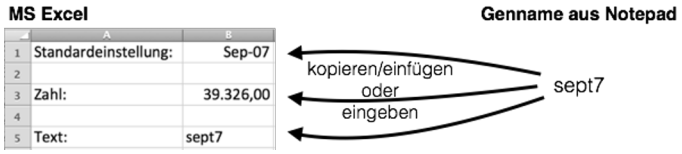


Abbildung 1.4 Vorsicht mit Excel: mit seinen Standardeinstellungen wird aus manchem Gennamen ein Datum.

1.1.4 Experimente und Bioinformatik

Bei meiner eigenen Arbeit spreche ich gerne davon, dass ich *experimental and computational biology* betreibe. Leider lässt sich das nicht stolperfrei ins Deutsche übertragen. Im Englischen wird die Einheit aus Experiment und computergestützter Datenanalyse unmittelbar deutlich.

Ich hatte schon angesprochen, dass bei Experimenten durch die moderne Messmethodik massenhaft Daten anfallen. Abb. 1.5 gibt einen Eindruck aus meiner eigenen Arbeit. Um die Dynamik der Biologie und ihrer regulativen Prozesse während der Biogasfermentation zu analysieren, verfolgen wir die abiotischen Faktoren (linker Plot) und entnehmen zu verschieden Zeitpunkten Proben für die Sequenzierung. Dabei sequenzieren wir alle zum jeweiligen Zeitpunkt in einer Gemeinschaft von Lebewesen aktiven Transkripte (RNA-Seq), das sogenannte Metatranskriptom. Jede Sequezierung resultiert in einer Datei, die im komprimierten Zustand rund 2 GB groß ist. Meine Doktorandin hat berechnet, dass alle in ihrer Arbeit ausgewerteten RNASeq-Daten ausgedruckt einen Papierberg mit dem Gewicht von acht Eiffeltürmen ergeben würde (Standardpapier und -textgröße vorausgesetzt). Ohne grundlegende bioinformatische Kenntnisse ist eine Auswertung unmöglich. In den Kapiteln 6 und 7 werden Sie den Umgang mit solchen Daten lernen.

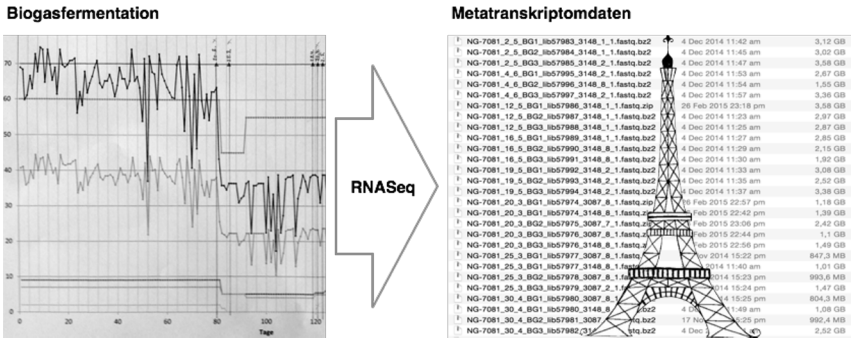


Abbildung 1.5 Bei der Analyse der Dynamik des Metatrankriptoms bei der Biogasfermentation fallen massenweise Daten an – die ausgedruckt dem mehrfachen Gewicht des Eiffelturms entsprechen würden.

Ich möchte aber noch auf einen weiteren wichtigen Aspekt bei der bioinformatischen Auswertung experimenteller Daten eingehen: den Weg der Daten (Abb. 1.6). Wenn Sie mit dem Computer experimentelle Daten auswerten, müssen Sie auf die verwendete Software vertrauen können. Je komplexer die Software und je weniger Rohdaten Sie in den Fingern halten, desto schwieriger ist es, Fehler zu entdecken. So wurde die Bildung des Ozonlochs über der Antarktis lange übersehen, weil für viele Jahre die Daten von TOMS (*Total Ozone Mapping Spectrometer*) Satelliten der NASA falsch ausgewertet wurden – und zwar von einem Computer. Der Algorithmus – von einem Menschen erstellt –

markierte zu geringe Werte als fehlerhaft, und diese wurden somit von der Gesamtanalyse ausgeschlossen (Pearce, 2008). Eine Reanalyse der Daten in den 80er Jahren aufgrund der Veröffentlichung eines britischen Wissenschaftlers (Farman *et al.*, 1985) zeigte, dass das Programm zu viele Daten markiert hat. Das Ozonloch hätte schon Jahre früher erkannt werden können.

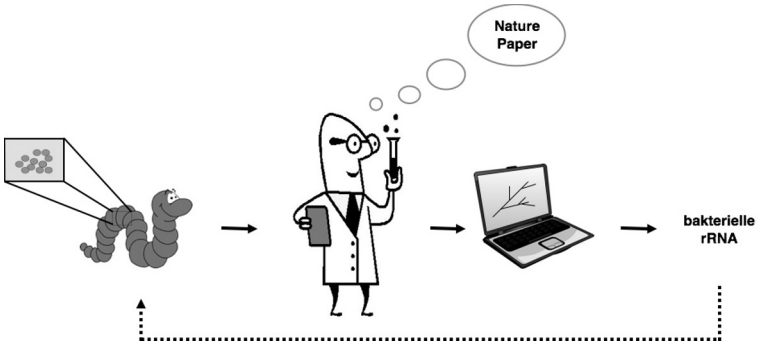


Abbildung 1.6 Ob mit oder ohne Computer, man muss den Weg der experimentellen Daten immer genau prüfen. Dieser Wissenschaftler glaubt, der Wurm sei ein Bakterium. Dabei hat er nur übersehen, dass der Wurm von Bakterien besiedelt ist, deren ribosomale RNA (rRNA) letztendlich zu einem falschen Ergebnis führte.

1.2 Meine Leser

Wenn Sie Bioinformatik als ein **Werkzeug** verstehen, um Ihre biologischen Daten zu analysieren und zu visualisieren, dann sind Sie meine Zielgruppe. Ich werde nicht über Algorithmen sprechen. Sie werden sich Linux und seine Tools sowie die bioinformatische Software anhand vieler Beispiele erarbeiten. Mit diesen Werkzeugen sollen Sie lernen und können Sie lehren. In jedem Fall steht die **praktische Anwendung** im Vordergrund, wie insbesondere Teil II zeigt.

Warnung

Bevor Sie loslegen: beachten Sie bitte unbedingt die Box zur Wahl des Betriebssystems auf Seite 43.

Tipp

»Und mit Geistesstärke tu' ich Wunder auch.« – Sie, der Leser oder die Leserin, halten mit diesem Buch einen Schnelleinstieg in den Händen. Nach der Bearbeitung der Beispiele sollten Sie in der Lage sein, die Methoden auf Ihre Daten anzuwenden und tiefergehende Quellen hinzuzuziehen. Goethes Zauberlehrling hatte die Möglichkeit nicht, aber Ihnen lege ich, unabhängig von diesem Buch, ans Herz: machen Sie immer Backups von Ihren Daten. Sonst kann auch Ihnen schnell das Wasser bis zum Halse stehen!

1.3 Notwendiges Vorwissen

Optimaler Weise haben Sie schon ein Basiswissen in Biologie (siehe Abb. 1.7) und keine Berührungsängste mit dem Computer.

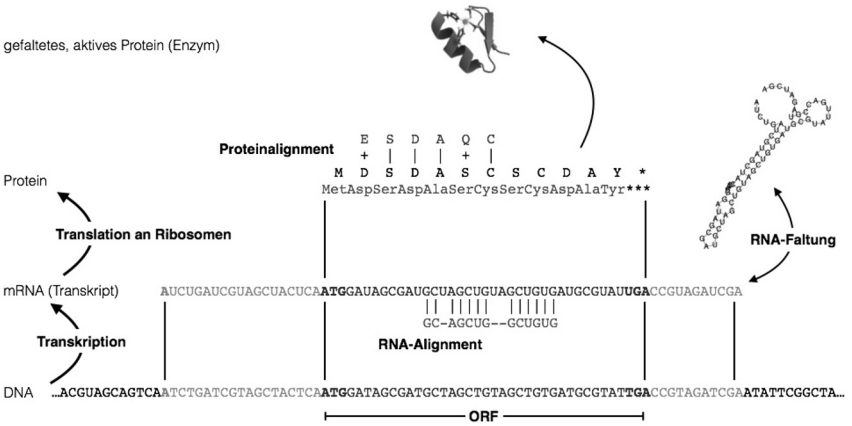


Abbildung 1.7 Grundwissen Biologie. Wenn Ihnen die gezeigten Begriffe bekannt vorkommen, dann haben Sie eine gute Basis. ORF: *open reading frame* (offenes Leseraster); im Proteinalignment bezeichnen die Pluszeichen ähnliche, die vertikalen Striche identische Aminosäuren von zwei Proteinsequenzen.

Ansonsten ist kein Vorwissen notwendig. Sie sollten aber Zugang zu einem Computer haben, auf dem Sie Software installieren dürfen und mit dem Sie in das Internet kommen. – Eine Sache wäre aber doch hilfreich, wenngleich nicht notwendig: wenn Sie ein Problem mitbringen – ein Datenverarbeitungsproblem. Denn die größte Motivation zum Lernen ist die Notwendigkeit (und Freude), ein Problem zu lösen.

1.4 Ziel des Buches

Der Weg ist das Ziel! Ich möchte Ihnen zeigen, wie Sie mit Daten spielen und aus Daten anschauliche Ergebnisse gewinnen. Ich werde keine Ergebnisse diskutieren und keine Algorithmen vorstellen. Mein primäres Ziel ist, dass Sie durch praktische Beispiele einen Einblick in die Prozessierung, Analyse und Visualisierung von biologischen sequenzbasierten Daten (DNA, RNA, Protein) unter Linux bekommen. Ich möchte Sie damit für die Datenflut in Praktika, Abschlussarbeiten und Forschungsprojekten rüsten (Abb. 1.8).

Abbildung 1.8 Mein Ziel ist es, Ihnen Grundlagen für die Verarbeitung und Analyse von Daten aus den Lebenswissenschaften zu vermitteln.

Wichtiger Hinweis

»Es gibt nichts Gutes, außer man tut es« – das sagte einst Erich Kästner. Lernen durch Handeln, das sagen die Pädagogen. Lassen Sie die Tasten schwingen, das sage ich. Ohne Übung geht nichts, und daher ist dieses Buch durchsetzt von sogenannten Terminals mit Beispielen, die Sie durcharbeiten und – das ist mir wichtig – nicht schnöde abtippen, sondern mit denen Sie spielen sollen. Verändern Sie Befehle, schauen Sie was passiert. *Learning by doing!* Der ganze Teil II dient der Übung.

Mit dem Einzug von Hochtechnologie und Hochdurchsatzverfahren in den Laboralltag nimmt die digitale Datenverarbeitung einen immensen Stellenwert ein. Es ist heute selbstverständlich, dass ein Naturwissenschaftler seine Forschungsdaten selbstständig graphisch aufarbeitet und präsentiert. Früher hat dies der Institutszeichner oder ein Graphiker gemacht. Ein Großteil der Zeit wird dabei am Computer mit der Formatierung der Daten verbracht. Häufig müssen Daten umformatiert werden, um den Formatansprüchen einer bestimmten Analysesoftware zu entsprechen: Kommata in Punkte, Tabulatoren in Semikolons, Leerzeichen in Unterstriche, Spalten in Zeilen umwandeln, Information aus zwei Dateien in eine Datei zusammenfügen etc. Hier setzt

dieses Buch an. Mit diesem Buch möchte ich Ihnen eine digitale Pipette in die Hand geben; dabei steht die Pipette für ein universelles Hilfsmittel im Labor. Während mit der Pipette Flüssigkeiten »prozessiert« werden, möchte ich Ihnen zeigen, wie Sie experimentelle Daten prozessieren können (Abb. 1.9).

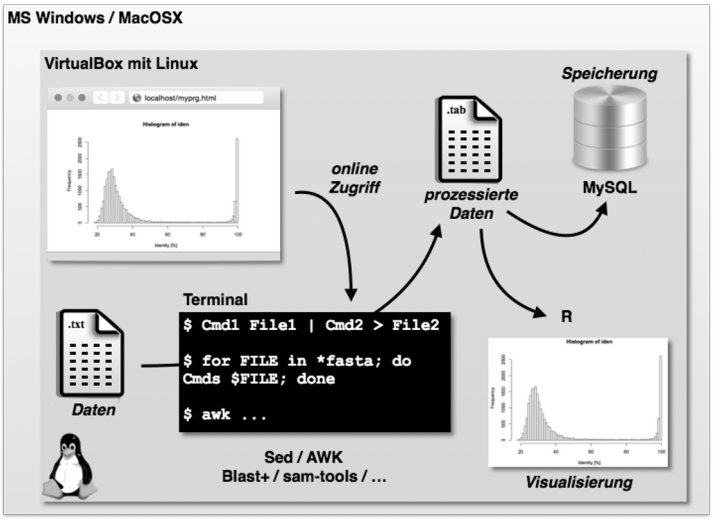


Abbildung 1.9 Datenprozessierung mit Linux und dem Terminal. Egal, welches Betriebssystem auf Ihrem Computer installiert ist, mit der freien VirtualBox-Software von Oracle installieren wir gemeinsam eine virtuelle Linux-Maschine (siehe Abschnitt 3.2).

1.4.1 Teil I: Vorbereiten

Im Zentrum dieses Buches steht der Terminal (Abb. 1.9). Mit verschiedenen Programmen werden wir experimentelle Daten prozessieren, speichern und visualisieren. Unabhängig davon, welches Betriebssystem auf Ihrem Computer installiert ist, können Sie mit der VirtualBox-Software von Oracle ein komplettes Linuxsystem als virtuelle Maschine laufen lassen (siehe Abschnitt 3.2). Nach der Bearbeitung des ersten Teils haben Sie also Ihr eigenes Linuxsystem am Start und beherrschen das Basiswerkzeug für die Verarbeitung von Daten.

Linux Linux ist ein kostenfreies, codeoffenes, aber leistungsfähiges Betriebssystem, das in der Bioinformatik breite Verwendung hat und mit welchem man

effizient Datenströme lenken und verarbeiten kann. Mithilfe der VirtualBox von Oracle steht eine kostenfreie und komfortable Möglichkeit zur Verfügung, Linux unter jedem anderen Betriebssystem laufen zu lassen. Wie das geht, zeige ich in Abschnitt 3.2. Für die Faulen unter Ihnen stelle ich »meine« virtuelle Maschine, die ich bei der Erstellung dieses Schnellkurses verwendet habe, auf der Webseite dieses Buches zum Download (*datenmassen.de*) zur Verfügung. Deren Installation stelle ich in Abschnitt 3.2.3 vor.

Terminal Der Terminal (auch Konsole oder Shell genannt) bezeichnet die Schnittstelle zum Linuxsystem (Abb. 1.9). Hier sind wir nahe an unseren Daten und erstellen Pipelines zu deren Verarbeitung. Im Terminal stehen eine große Zahl von kleinen Programmen (Linux-Tools) zur Verfügung. Die wichtigsten Tools für die Bioinformatik stelle ich Ihnen vor.

AWK AWK ist eine einfache, in jedem Linux-, Unix- und MacOSX-Betriebssystem verfügbare Skriptsprache. Sie eignet sich hervorragend für den Anfänger und reicht vollkommen aus, um effizient Daten zu prozessieren (Abb. 1.10).

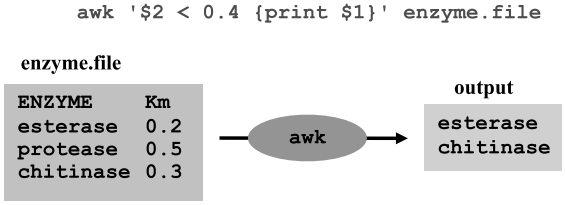


Abbildung 1.10 AWK. Ein Powertool aus der digitalen Steinzeit.

1.4.2 Teil II: Arbeiten

Der zweite Teil führt in die Bearbeitung von DNA-, RNA- und Protein-Sequenzen ein. Dazu arbeiten Sie einige Beispiele durch und werden so ein Gefühl für die Anwendung von bioinformatischer Software bekommen. Die Beispiele sind sowohl für das Selbststudium als auch den Einsatz in der Lehre geeignet. Ganz konkret werden wir mit bioinformatischen Methoden Fragen nachgehen wie:

- **Forensische Mikrobiologie – Was ist so schlimm an EHEC?** → Mit einem lokalen, *BLAST+* basierten Genomvergleich begeben wir uns im Kapitel 5 auf die Suche nach Pathogenitätsfaktoren in *E. coli*-Stämmen.

- **RNASeq und Biogas – Identifikation Biogas-produzierender Bakterien** → Im Kapitel 6 werden wir Sequenzdaten im FastQ-Format aus einer Metatranskriptomsequenzierung analysieren. Wir führen eine Qualitätskontrolle der Sequenzdaten mit *Trimmomatic*, *FastQC* und dem *FASTX-Toolkit* durch. Mit *BLAST+* analysieren wir die Zugehörigkeit ribosomaler RNA zu bekannten Bakterien und Archaeen.
- **Vom Gen zum Methan – Expressionsdaten auf Stoffwechselkarten projizieren** → Welche Enzyme sind in einem Biogasfermenter aktiv? Im Kapitel 7 ordnen wir Sequenzen aus einer Metatranskriptomsequenzierung über einen *BLAST+* basierten Sequenzvergleich Enzymen zu. Diese Enzyme markieren wir in Stoffwechselkarten von KEGG.
- **Bio(t)error – Gefährliche Mutationen des H1N1-Schweinegrippevirus** → Die Spanische Grippe war die große Seuche des letzten Jahrhunderts. Welche Mutationen sind beim Grippevirus vom Typ H1N1 seitdem aufgetreten? Wie wirken sich diese auf die Wirksamkeit der wichtigsten Medikamente Tamiflu und Relenza aus? Um das zu untersuchen, werden wir in Kapitel 8 Neuraminidasesequenzen verschiedener Influenzaviren mit der Neuraminidase des Spanischen Grippevirus vergleichen. Dazu nutzen wir *Clustal-Omega* und schauen uns die Verwandtschaft der Proteinsequenzen mit *TreeView* an. Die Daten laden wir mit dem Tool *Entrez Direct* aus der GenBank in den Terminal. Die Visualisierung der 3D-Proteinstruktur mit Hervorhebung der Mutationen übernimmt *Jmol*.
- **Ebola – Detektion von Variation durch Resequenzierung** → Die Suche nach SNPs (*single nucleotide polymorphisms*) als genetische Marker beschäftigt uns in Kapitel 9. Mit *EMBOSS*, *Bowtie2* und den *SAMtools* untersuchen wir dazu Genomresequenzierungsdaten im FastQ-Format. Für die Visualisierung stelle ich Ihnen das SAM-Tool *tview*, *IGV* und *Genome Savant* vor.

Wichtiger Hinweis

Daten aus erster Hand – Sie werden u.a. mit aktuellen Daten aus meiner Forschung arbeiten. So untersuchen wir zum Beispiel, welche Gene in den Mikroorganismen einer Biogasanlage aktiv sind. Dazu sequenzieren wir alle Transkripte (mRNAs) in einer biologischen Probe. Sie sollen dann herausfinden, welche Organismen und welche Gene an der Methanproduktion beteiligt sind.

1.4.3 Teil III: Präsentieren und Veröffentlichen

Datenbanken Bevor wir Daten darstellen oder veröffentlichen, sollten sie in vernünftiger und nachhaltiger Weise gespeichert werden. Dafür sind Datenbanksysteme wie MySQL oder MariaDB (die beide zu 100 % kompatibel sind) hervorragend geeignet. Ich werde Ihnen die Verwendung von MariaDB vorstellen und darüber hinaus zeigen, wie Sie Datenbankanfragen erstellen.

R und Latex Einen wichtigen Anteil an der wissenschaftlichen Arbeit nimmt das Präsentieren und Publizieren der Ergebnisse ein, sei es das Schreiben eines Versuchsprotokolls, ein Arbeitsgruppenvortrag, eine Abschlussarbeit oder eine Veröffentlichung. Wichtig ist in allen Fällen eine ordentliche Auswertung und Darstellung der Daten. Mit der Programmierumgebung R (*r-project.org*) und der Satzsprache Latex (*latex-project.org*) zeige ich Ihnen Werkzeuge zur Datenanalyse, zur graphischen Darstellung von Daten und der Formatierung von Texten, die weit über die Möglichkeiten von MS Excel und MS Word hinausgehen.

Apache und HTML Wenn Sie eine Datenverarbeitungspipeline etabliert haben, dann möchten Sie diese vielleicht auch mit Kollegen teilen. Damit diese nicht die Kommandozeile lernen müssen (wahrscheinlich haben Ihre Kollegen sich nicht durch dieses Buch gearbeitet), bietet sich der Webbrowser an. Daher zeige ich Ihnen, wie Sie einen eigenen Apache Webserver starten, eine Webseite aufsetzen und über das Internet Daten austauschen.

1.4.4 »Kann ich das?«

Von verschiedenen Seiten habe ich folgende Bedenken gehört: a) Programmieren können doch nur Informatiker oder Freaks. Dem halte ich entgegen: Und kochen können nur Köche? b) Wenn schon, dann möchte ich JAVA oder C++ oder wenigstens Perl oder Python lernen. Dem halte ich entgegen: Schön – wenn die Lösung Ihrer Probleme so lange warten kann. Vorher können Sie mit AWK eine gute Grundlage zum Erlernen dieser Sprachen legen, denn AWK ist aus C hervorgegangen und in Perl und Python gemündet. Als in der Industrie und Hochschule tätiger Wissenschaftler kann ich berichten, dass ich den überwiegenden Teil der Datenverarbeitungsprobleme mit Linux, seinen Tools und AWK lösen kann. Aber sehen Sie selbst ...

1.4.5 Wie geht es weiter?

Wenn Sie dieses Buch durchgearbeitet haben, dann sollten fit sein, um a) Ihre eigenen Daten zu bearbeiten und b) selbständig neue Methoden anzuwenden. Es lohnt sich beispielsweise ein Blick auf das BioLinux-Projekt von Tracey Timms-Wilson vom Centre for Ecology & Hydrology in England (*environmentalomics.org/bio-linux/*). Zahlreiche BioLinux-Projekte sind in dem Paper von Field *et al.* (2006) zusammengefasst.
Es gibt hilfreiche Bücher und Onlinekurse, welche die hier eingeführten Werkzeuge wie Python, R, MySQL (MariaDB) und Latex vertiefen. Mit dem hier Gelernten im Rücken sollte deren Bearbeitung für Sie kein Problem darstellen. Eine gute Adresse, um Fragen zur Programmierung zu stellen und Antworten zu finden, ist *stackoverflow.com*. Für biologische und bioinformatische Fragestellungen bieten sich *biostars.org*, *http://seqanswers.com* und *biology.stackexchange.com* an. Meistens landen Sie über Google ohnehin bei einer dieser Seiten.

1.5 README

In der Tradition von Software- und Datenarchiven, die in aller Regel eine *readme*-Textdatei mit wichtigen Informationen enthalten, möchte ich hier das Wichtigste zu diesem Schnellkurs zusammenfassen.
Informationen und Daten zu diesem Buch finden Sie unter *datenmassen.de*. Hier gibt es ...

- **Dateien** mit den im Schnellkurs verwendeten Daten.
- »meine« virtuelle Maschine mit **Ubuntu-Linux**, auf der alle verwendeten Daten vorhanden sind und die angewandte Software installiert ist.
- angepasste Anleitungen, wenn sich Programme, Daten oder Internetlinks ändern.
- alle **Abbildungen** aus dem Buch in Farbe für Freude und Lehre.

Für alle, die **Facebook** mögen: Auf der Seite *facebook.com/awkologist* poste ich gelegentlich Informationen aus der Welt der Bioinformatik.
Die **Beispiele** im Buch sollten mit allen gängigen Linux-Versionen und auch Apples OSX funktionieren. Getestet habe ich alles mit der virtuellen Maschine, deren Installation ich im Abschnitt 3.2 beschreibe.
Antworten zu den durch die Kapitel gestreuten Fragen finden sich in Kapitel 13.

Ich erkläre immer, was in den **Terminals** passiert – manchmal aber erst nach dem Terminal.
Dies ist ein Schnellkurs. Zögern Sie nicht, **offene Fragen** an Onkel Google, Tante Yahoo oder Baby Bing zu stellen.
Und zu guter Letzt: Am Ende des Buches, im Abschnitt Whitespace, finden Sie Platz für Notizen zu eigenen Befehlen, Passwörtern oder Dateipfaden.

Have a lot of fun.

1.6 Was bedeutet was

Der wichtigste Bestandteil dieses Buches ist der Terminal (Terminal 1.1). Der Terminal im Computer wird auch Shell oder Bash genannt. **Bash** ist die »Sprache« des Terminals. In den Terminals in diesem Buch sind die Befehle angegeben, die Sie auch auf Ihrem Computer ausführen. Das vorne stehende Dollarzeichen ($) ist der sogenannte Eingabeprompt. Diesen müssen Sie nicht eingeben – ich zeige es nur, damit Sie Eingaben von Ausgaben der Bash unterscheiden können. Hinter den Befehlen schreibe ich teilweise Kommentare, die auf ein Hashzeichen (#) folgen (etwa in den Zeilen 1-2 im Terminal 1.1). Auch diese können Sie beim Abarbeiten der Befehle ignorieren. Sie dienen nur dem besseren Verständnis. Zur besseren Gliederung habe ich Befehlsworte fett gedruckt. Beachten Sie auch, dass eine Zeile im Terminal dieses Buches umgebrochen sein kann wie Zeile 3 im Terminal 1.1. Sie erkennen das an der Zeilennummerierung und müssen den Befehl als eine Zeile in Ihrem Terminal eintippen. Am Ende einer jeden Zeile drücken Sie die Return Taste, um den Befehl auszuführen.

Terminal 1.1 Beispiel für einen Terminal

```
1 $ date | awk '{print $4}'     # führe den Befehl date aus
2 16:40:07                      # und extrahiere das 4. Feld
3 $ ./wenn ein Befehl sehr lang ist und hier nicht in eine Zeile passt dann
    ↪ wird sie umgebrochen
4 $
```

Der Inhalt einer Datei wird wie in Datei 1.1 dargestellt.

Datei 1.1 *id-exprA-exprB-len-gc.tab*

```
all0001     1109      2202      1230    508
all0002     1094      1373.25   738     298
all0004     1968.5    7153.5    948     454
all0005     4892.75   7768.75   1521    708
all0006     10175.2   15089.5   552     246
all0007     4971      6861      564     264
all0008     6277.5    9637.75   492     223
all0010     15528.5   24387.5   756     325
```

Tipp

Sie finden alle Dateien und zusätzliche Informationen zu diesem Buch auf meiner Seite im Internet unter *datenmassen.de*

Auch Programme haben ihre eigene Box wie in Programm 1.1. Zur besseren Übersicht habe ich wichtige Befehle fett gesetzt. Das ist ansonsten ohne Funktion.

Programm 1.1 *pubmed2.sh*

```
#!/bin/bash
# Get PubMed Abstract for QUERY_TERM from the past number of DAYS
if [ $# -ne 2 ]
  then
    echo "No arguments supplied / Usage: pubmed QUERY_TERM DAYS"
    exit 1
fi
wget -qO- "http://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=
   ↪ pubmed&rettype=uilist&retmode=xml&datetype=pdat&reldate=$2&field=
   ↪ title&term=$1" |
egrep -o '(<Id>.*</Id>|<Count>[0-9]+</Count>)' |
sed 's/<[/A-Za-z]\+>//g' |
head -n -2 |
awk '{
  if($0 == 0){print "No hit in titles"; exit}
  else{print "Only the first 20 Abstracts are shown"};
  rec=$0;
  print "Hits for Query Term: "rec;
  if(rec>20){rec=20};
  for(i; i<rec; i++){
    getline; hit=$0","hit};
  getline;
  print "Total Hits: "$0
  }
  END{
  print hit;
  system("wget -qO-
     ↪ \"http://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=
     ↪ pubmed&id="hit"&retmode=text&rettype=abstract\"")
}' | less
```

Code im Text wird so `awk -f test.awk` dargestellt.