Bioinformatik

Überblick

Von einer Nischendisziplin zu einer zentralen Rolle in der Biologie

Die Bioinformatik hat sich in den letzten 20 Jahren von einer kleinen separaten Randdisziplin der Biologie zu einem integralen Bestandteil praktisch aller Bereiche der biologischen Forschung entwickelt. Der Haupttreiber hinter dieser Entwicklung ist der rasante Zuwachs der Datenmengen, die durch diverse „Omics"-Techniken (genomics, proteomics, transcriptomics, ...) und digitale Abbildungstechniken erzeugt werden. Selbst kleinere Projekte generieren inzwischen Datensätze, die so gross sind, dass sie sich weder „per Hand" noch mit standard Datenverarbeitungsprogrammen wie z.B. Excel bearbeiten lassen. Ausserdem werden die statistischen Methoden, die bei der Datenanalyse verwendet werden, immer komplexer.

Biologische Forschung ohne Informatik ist also kaum noch möglich und so gehören (Bio)informatikkenntnisse heute ganz selbstverständlich zum Rüstzeug eines jeden Biowissenschaftlers, Biotechnologen oder Pharmazeuten.

Die spezifischen Methoden und Ansätze, die in den verschiedenen biologischen Disziplinen angewendet werden, unterscheiden sich dabei sehr deutlich voneinander. Die mathematischen und physikalischen Grundlagen von molekular-dynamischen Untersuchungen zu Proteinstruktur und Funktion haben z.B. praktisch nichts gemein mit den Methoden, die zur Untersuchung von metabolischen Netzwerken verwendet werden. Diese wiederum unterscheiden sich ganz deutlich von den Ansätzen, die zur Analyse von DNA-Sequenzen verwendet werden.

Fokus auf bioinformatische Methoden in der Genetik

Anstatt einen Überblick über die gesamte Bioinformatik zu versuchen wird diese Lerneinheit den Fokus auf diejenigen Ideen, Datensätze und Methoden legen, die für den Rest des Kurses besonders relevant sind und an diesen konkreten Beispielen generelle Prinzipien der Bioinformatik besprechen. Konkret heisst dies, dass wir uns damit beschäftigen werden, wie man in Datenbanken Informationen über Gen- und Genomsequenzen, sowie genetische Variation findet, wie man DNA Sequenzen miteinander vergleicht (Sequenz-Alignment), wie man durch die Erstellung phylogenetischer Bäume evolutionäre Prozesse rekonstruieren kann und wie man Sequenzmuster erkennen und diese nutzen kann, um biologische Funktionen von DNA- und Proteinsequenzen zu interpretieren.

Später im Kurs werden Sie dann auf das hier Gelernte zurückgreifen, um z.B. Informationen über Gene nachzuschauen oder Sequenzen miteinander zu vergleichen. Darüber werden in den verschiedenen Lerneinheiten noch weitere, für diese speziellen Themen relevante, bioinformatische Methoden vorgestellt. Im Abschnitt zu genomweiten Assoziationsstudien (GWAS) werden wir z.B. die für solche Studien notwendigen statistischen Techniken besprechen.

Muss ich als Biologe programmieren können?

In der Physik und in vielen Bereichen der Chemie sind relativ hoch entwickelte Computerkenntnisse generell und Programmierkenntnisse im Besonderen eine absolute Selbstverständlichkeit. Dementsprechend sind viele Ressourcen und Services nur mit recht fundierten Computerkenntnissen nutzbar.

Unter Biowissenschaftlern gehören Computerkenntnisse traditionell nicht zum Grundwissen. Um die einsetzende Datenflut dennoch zu bewältigen, hat die biowissenschaftliche Community daher viel Energie in den Aufbau von intuitiv nutzbaren, webbasierten Tools investiert. Diese Tools sind allgemein zugänglich und kostenlos. Viele dieser Tools sind inzwischen ganz hervorragend und man kann auch ohne Programmierkenntnisse viele dieser Ressourcen nutzen.

Die Vorteile und Möglichkeiten, die sich durch selbst minimale Programmierkenntnisse erschliessen, sind aber so tiefgreifend, dass jedem zu empfehlen ist mindestens eine Programmiersprache wie z.B. Python, Perl, Matlab oder R zu lernen. Die Online-Ressourcen für das Erlernen dieser Programmiersprachen sind extrem umfangreich und reichen von gut gemachten Tutorialvideos für absolute Anfänger bis zu Plattformen wie z.B. Stackoverflow in denen man Antworten auf die vertracktesten Fragen findet.

Die hier genannten Programmiersprachen verfügen über spezielle Biologie-Module in denen eine sehr grosse Vielfalt von Funktionen (z.B. Sequenzalignment etc.) implementiert sind. Man kann also mit einigen wenigen Zeilen Code relativ komplexe Probleme lösen.

Bioinformatische Analyse Methoden basieren auf Modellen

George P. Box hat einmal gesagt „all models are wrong, but some are useful". Es ist wichtig, diesen Satz im Hinterkopf zu behalten.

Fast alle bioinformatische Analysemethoden, selbst so einfache wie der Vergleich zweier DNA Sequenzen, beruhen auf einem bestimmten Modell. Jedes Modell ist ein mehr oder weniger unzulänglicher Versuch, die sehr hohe Komplexität eines biologischen Sachverhalts in einen mehr oder weniger vereinfachten Rahmen zu pressen.

Die in der Biologie verwendeten Analysemethoden sind inzwischen so komplex, dass es uns als Nutzer oft nicht möglich ist, die in einem Programm zum Einsatz kommenden Algorithmen vollständig nachzuvollziehen. Wir werden im Anschluss sehen, dass selbst so scheinbar einfache Aufgaben wie der paarweise Vergleich von DNA-Sequenzen bereits relativ komplizierte Algorithmen benötigen, die einige relativ „gewagte" Annahmen machen.

Wenn wir diese Annahmen und das einem Algorithmus zugrundeliegende Modell nicht kennen, kann es leicht zu Problemen und Missinterpretationen kommen.

Erfahrungsgemäss entstehen Schwierigkeiten weitaus seltener dadurch, dass ein Algorithmus selbst fehlerhaft ist, als dadurch, dass ein perfekt funktionierender Algorithmus auf ein unpassendes Problem angewandt wird.

Wichtige Datensätze, online Datenbanken und Tools für die Genetik

Die Biologie hat sich in relativ kurzer Zeit von einer datenarmen zu einer sehr datenreichen Disziplin entwickelt. Einen entscheidenden Beitrag dazu geleistet hatten eine Reihe von Grossprojekten, deren Aufgabe einzig und alleine darin bestand, massiv Daten zu sammeln. Die aus diesen Projekten hervorgegangenen Datensätze stehen der Allgemeinheit kostenfrei über öffentliche Datenbanken und deren Webportale zur Verfügung.

Inzwischen sind diese Webinterfaces sehr benutzerfreundlich geworden und Suchen, sowohl mit Google-ähnlichen, unstrukturierten Kombinationen von Suchbegriffen, als auch mit strukturierten Suchanfragen, sind möglich.

Hier werden einige der wichtigsten Datensätze und Webressourcen besprochen, ohne welche die Forschung in den Biowissenschaften kaum noch vorstellbar wäre.

Human Genome Project und andere Genom-Projekte

Vor nicht einmal 20 Jahren wurde das Sequenzieren von gesamten tierischen und pflanzlichen Genomen als nahezu unmöglich betrachtet. Im Human Genome Project wurde dann zum ersten Mal ein gesamtes menschliches Genom sequenziert. Eine der Hauptherausforderungen dieses Projekts war es, aus den experimentell erhaltenen Sequenzen, die nur einige hundert bis tausend Basenpaare lang sind, die kontinuierlichen Sequenzen der einzelnen Chromosomen in einem so genannten *build* zusammenzusetzen (manchmal wird auch der Begriff *assembly* anstatt *build* verwendet). Besonders schwierig dabei ist das Zusammensetzen von homologen Sequenzen, die mehrfach in einem Genom auftreten. So gibt es weiterhin einige (wenige) Bereiche im menschlichen Genom bei denen die genaue Abfolge der Sequenzen noch nicht klar ist. Man spricht daher weiterhin von *draft assemblies*, also „Genom-Entwürfen". Neue Builds erscheinen etwa alle 1-2 Jahre. Der neuste Build wird als GRCh38 bezeichnet, aber viele Forscher verwenden weiterhin ältere Versionen, was zu einigen Verwirrungen führen kann. Die Unterschiede zwischen den neueren Builds sind inzwischen recht gering. Es kommt aber immer noch vor, dass sich z.B. herausstellt, dass ein Gen auf einem falschen Chromosom angesiedelt war oder dass einige Basen in der Sequenz gefehlt haben. Wenn man also den Ort eines Gens im Genom durch dessen Chromosomennummer und Basenpaarposition beschreiben will, muss man jeweils den Build angeben, auf den man sich bezieht.

Inzwischen gibt es Genomsequenzen nicht nur vom Menschen, sondern von vielen verschiedenen Spezies. Diese Genomsequenzen stehen z.B. auf der NCBI Webseite[1] zum Download bereit.

HapMap, 1000 Genomes und dbSNP

Die oben beschriebenen Genomsequenzen repräsentieren jeweils ein Genom für eine bestimmte Spezies. Unter den Individuen innerhalb einer Spezies bestehen aber immer genetische Unterschiede. Es gibt also nicht ein einziges menschliches Genom, sondern genauso viele menschliche Genome wie es Individuen gibt. Die Projekte Hapmap und 1000 Genomes haben diese Variationen beim Menschen systematisch untersucht. Dank dieser Projekte haben wir inzwischen ein sehr gutes Verständnis davon, wo im Genom besonders viele Variationen auftreten und welche genetischen Marker oft gemeinsam vererbt werden (Haplotypen). Aus diesen Informationen lassen sich z.B. Regionen im Genom erkennen, die evolutionärem Druck ausgesetzt waren, oder es lassen sich prähistorische Völkerwanderungen rekonstruieren. Darüber

[1] http://www.ncbi.nlm.nih.gov/genome/browse/

hinaus spielt diese Information bei der Entwicklung und Interpretation von diagnostischen Gentests eine entscheidende Rolle.

Die Daten aus den HapMap und 1000 Genomes Projekten sind auf der 1000 Genomes Webseite[2] zugänglich.

Die dbSNP-Datenbank[3] sammelt alle bekannten, im menschlichen Genom auftretenden Einzelbasenvariationen (SNPs, Insertions und Deletions). Man kann dort z.B. nachschauen, wie oft eine bestimmte Variation in einer bestimmten Bevölkerungsgruppe auftritt.

KEGG, ein Verzeichnis von Stoffwechselwegen und Enzymen

KEGG[4] steht für Kyoto Encyclopedia of Genes and Genomes und ist ein grosses Japanisches Bioinformatikportal. KEGG ist vor allem für seine Informationen zu Stoffwechselprodukten und -wegen bekannt. Man kann in dieser Datenbank z.B. nach allen Stoffwechselwegen suchen, die zu einem bestimmten Stoffwechselprodukt führen. KEGG bietet auch ein recht angenehmes Interface zu Geninformationen, wobei vor allem die Resultatseiten sehr übersichtlich gestaltet sind.

GenBank die zentrale Datenbank für DNA-Sequenzen

GenBank enthält inzwischen DNA-Sequenzen mit einer Gesamtlänge von fast einer Trillion (1'000'000'000'000) Basenpaare. Dabei handelt es sich nicht einfach um Rohdaten, sondern um annotierte Sequenzen, also Sequenzen zu denen Informationen wie Spezies, Ursprung, potentielle Gene, das Ursprungslabor etc. vorhanden sind. Wichtig: jede der in GenBank enthaltenen Sequenzen hat eine individuelle Identifikationsnummer (engl. *accession number*). Dies erlaubt es Forschenden sich eindeutig darüber zu verständigen auf genau welche Sequenz sie sich beziehen.

Der zentrale Vorteil von GenBank, die ungeheure zur Verfügung stehende Datenmenge, ist zugleich das Hauptproblem. In der Vielfalt der Daten ist es oft schwer die relevanten Daten zu finden.

GenBank bietet daher die Möglichkeit, Suchen auf Sub-Datensätzen zu beschränken. Besonders nützlich ist dabei der nicht-redundante RefSeq-Datensatz. Nicht-redundant bedeutet hier, dass für jedes Chromosomensegment, jede RNA und jedes Protein jeweils nur ein Eintrag besteht. Diese Sequenz dient dann als Referenz relativ zu der man Abweichungen beschreiben kann. Z.B. könnte man eine neugefundene Variante eines Gens dadurch beschreiben, dass sie sich an Basenposition X, Y und Z in dieser und jener Weise von der RefSeq-Referenzsequenz unterscheidet. RefSeq-Einträge sind auch besonders gut annotiert, die zu der Sequenz erhältlichen Informationen sind also besonders umfangreich und mit besonderer Sorgfalt erstellt.

ENCODE

Encode steht für Encyclopedia of DNA Elements. Ziel dieses noch laufenden Langzeitprojekts ist die komplette Annotation des gesamten menschlichen Genoms mittels diverser *high throughput* Technologien. Dazu wird z.B. die Expression aller menschlichen Gene in verschiedenen Zelltypen gemessen oder die Bindungstellen für die wichtigsten Transkriptionsfaktoren im gesamten Genom bestimmt. Mehr Informationen zum ENCODE Projekt finden Sie auf der Webseite[5].

Bioinformatik-Tools im Web

Das EBI (European Bioinformatics Institute) und das NCBI (National Center for BioInformatics in den USA) bieten eine breite Palette von Webtools an, mit denen viele einfache Aufgaben (z.B. Übersetzung von DNA Sequenzen in Protein Sequenzen), aber auch zunehmend komplexe Bioinformatik-Operationen durchgeführt werden können. Eine Liste solcher Webservices finden Sie auf der EBI Webseite[6].

In diesem Kursabschnitt werden Sie diese Services benutzen, um nach Sequenzen zu suchen, solche Sequenzen untereinander zu alignen und Phylogramme zu erstellen. Sehen Sie sich dazu die entsprechenden Tutorial-Videos und Aufgaben auf Moodle an.

Pubmed

Die Literatur zur biologisch-medizinischen Forschung wächst rasant. Pubmed[7] bietet Ihnen ein intuitiv nutzbares Interface, um diese Literatur zu durchsuchen. In vielen Fällen können Sie direkt von dort auf die gefundenen Forschungsartikel zurückgreifen.

Genome-Browser

Eine besonders interaktive Möglichkeit die Unmenge von existierenden Daten aus den vielen Datenbanken zusammenzubringen und zu visualisieren sind sogenannte Genome-Browser.

[2] http://www.internationalgenome.org/
[3] http://www.ncbi.nlm.nih.gov/SNP
[4] http://www.genome.jp/kegg/
[5] http://genome.ucsc.edu/ENCODE/
[6] http://www.ebi.ac.uk/services/all
[7] http://www.ncbi.nlm.nih.gov/pubmed

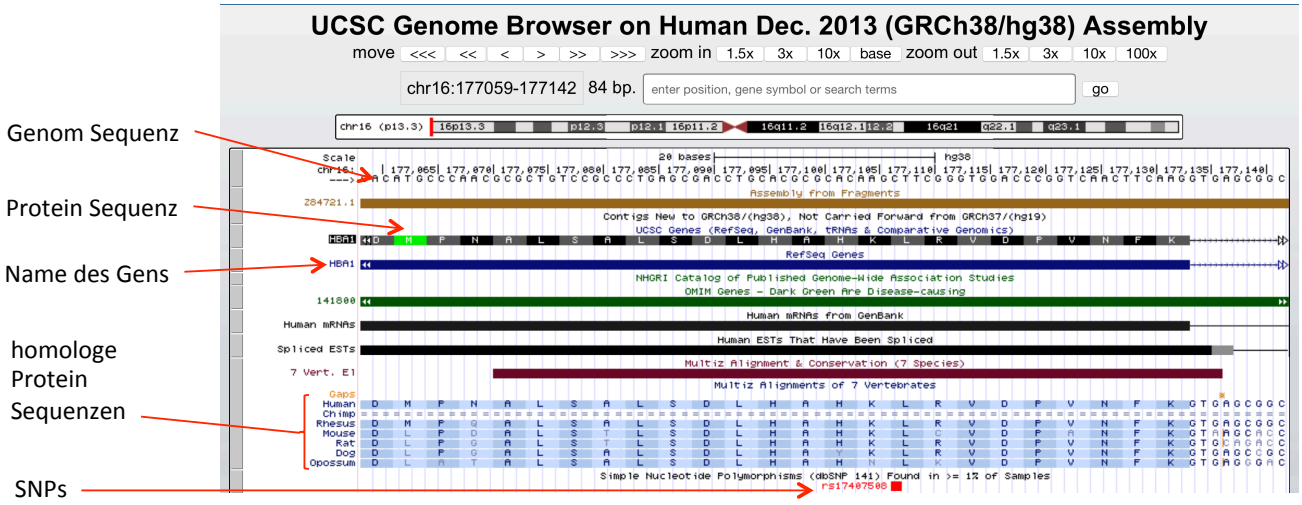


Abbildung 1 *Screenshot des UCSC Genome Browsers mit mehreren Tracks, welche Informationen zur DNA- und Proteinsequenz, zu homologen Protein in anderen Organismen und zu genetischen Variationen (SNPs) zeigen.*

Genome-Browser sind in der Regel als interaktive Webseiten implementiert, die es erlauben, sich einen Überblick über lange Sequenzabschnitte zu verschaffen oder in bestimmte Bereiche der Genomsequenz hereinzuzoomen.

Einer der meistbenutzten Browser dieser Art ist der UCSC Genome Browser[8] der von der University of California Santa Cruz betrieben wird.

In diesem Browser werden zusammen mit der DNA-Sequenz in sogenannten Tracks eine breite Vielfalt von Informationen aus anderen Datenbanken angezeigt (z.B. regulatorische Sequenzen, bekannte genetische Variationen oder Hinweise auf mit den Genen verbundene Krankheiten). Durch entsprechende Buttons im Oberteil des Browsers ist es möglich, an der Sequenz entlang zu scrollen oder in die Sequenz herein oder herauszuzoomen. Viele der angezeigten Elemente sind anklickbar und verlinken den Nutzer zu den entsprechenden Datenbanken, aus denen die entsprechenden Informationen gesammelt wurden.

Man kann sich so sehr schnell einen Überblick über die Umgebung eines Gens machen. Der UCSC Genome Browser ermöglicht es dem Nutzer darüberhinaus zusätzliche, vom Nutzer bereitgestellte, Informationen in sogenannten *custom tracks* darzustellen.

APIs eine effiziente Alternative zu Webinterfaces

Webinterfaces zu den oben beschriebenen Datenbanken sind sehr praktisch, wenn man nach einem oder zwei Einträgen sucht. Für umfangreichere Suchen oder Suchen die Resultate aus mehreren Datenbanken verknüpfen, stossen diese Interfaces aber schnell an ihre Grenzen.

Deshalb gibt es für viele online Datenbanken auch sogenannte *application programming interfaces (APIs)* über die man ohne Browser (z.B. direkt aus einem Python-, Perl- oder R-Programm) wesentlich direkter und strukturierter mit diesen Datenbanken interagieren kann.

Man greift dabei nicht direkt auf die Datenbank zu, sondern die eingegebenen Suchanfragen werden von der *API* in die von der Datenbank intern verwendete Suchsyntax übersetzt. Diese vielleicht kompliziert anmutende indirekte Struktur hat aber den entscheidenden Vorteil, dass die interne Struktur der Datenbank angepasst werden kann, ohne dass sich alle Nutzer und Programme auf eine neue Suchsyntax umstellen müssen.

Für weiterführende Informationen über die APIs zu den von der NCBI betriebenen Datenbanken sind computergewandte Kursteilnehmer auf das folgenden E-Book verwiesen: Entrez Programming Utilities Help[9].

Online Kurse für Bioinformatik

Inzwischen gibt es eine sehr grosse Anzahl von exzellenten Online-Kursen zum Thema Bioinformatik, die allgemein und kostenlos zugänglich sind. Die Qualität dieser Kurse ist oft ausgezeichnet.

David Searls von der University of Pennsylvania hat in einem Übersichtsartikel in PLOS Computational Biology die besten Online-Kurse im Bereich Bioinformatik beschrieben. Für Studenten die sich in diesem Bereich selbst weiterbilden wollen, steht eine Kopie dieses Artikels auf Moodle zum Download bereit.

[8] http://genome.ucsc.edu

[9] http://www.ncbi.nlm.nih.gov/books/NBK25501/

Sequenzalignment

Sequenzalignments spielen in der bioinformatischen Behandlung von genetischen Daten eine ganz zentrale Rolle. Wichtige Anwendungsbeispiele sind die Konstruktion von Stammbäumen, die Zusammensetzung von Teilsequenzen in Sequenzierungsprojekten oder die Annotation von Genomsequenzen, also die Zuordnung von Funktionen zu bestimmten Sequenzabschnitten.

Wenn wir Sequenzen durch ein Alignment miteinander vergleichen, stellen sich die folgenden Fragen:

- Wie können die Nucleotide oder Aminosäuren in den beiden Sequenzen einander zugeordnet werden?
- Wie ähnlich sind die zwei Sequenzen zueinander?
- Sind Muster der Konservierung und der Variabilität zu erkennen?
- Deuten die beobachteten Muster auf eine biologische (z.B. evolutionäre) Beziehung zwischen den beiden Sequenzen hin und wenn ja, welche?

Was ist ein Alignment?

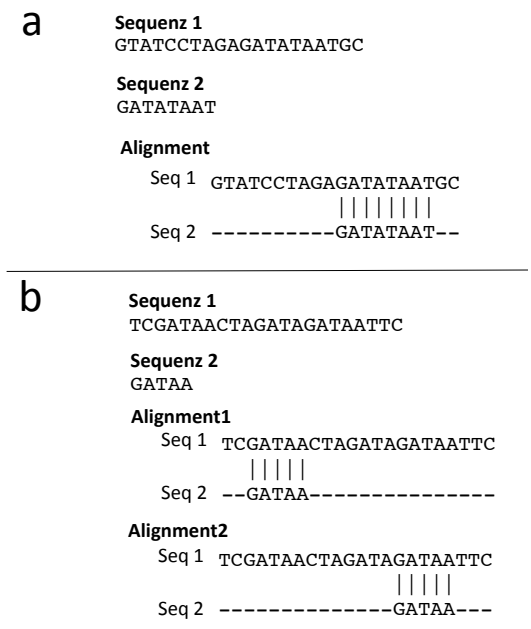


Abbildung 2 *Panel **a** zeigt ein einfaches Beispiel eines Sequenzalignments. In diesem Fall lässt sich jeder Base in Sequenz 1 eine entsprechende Base in Sequenz 2 zuordnen. Wir erhalten einen perfekten Match. In Panel **b** sind zwei andere Sequenzen gezeigt. Alignment 1 zeigt, dass wir auch hier einen perfekten Match erhalten. Wie aber Alignment 2 zeigt, ist die Zuordnung der Basen nicht eindeutig, denn es gibt ein weiteres, ebenfalls perfektes Alignment.*

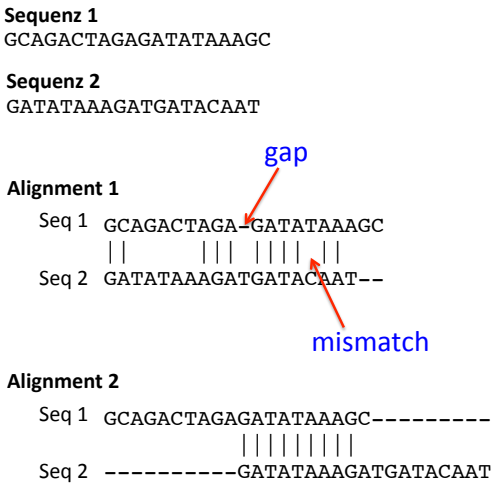


Abbildung 3 *Alignments sind selten so perfekt wie jene in Abbildung 2. Um mit diesen „Imperfektionen“ umzugehen, muss man gegebenenfalls mismatches akzeptieren oder gaps einfügen. Dabei ist es oft möglich mehrere unterschiedliche Alignments zu generieren, die durchaus plausibel erscheinen. Hier zeigt Alignment 1 z.B. 11 matches, benötigt dafür aber eine gap. Alignment 2 erzeugt nur 9 matches aber benötigt keine gap.*

Ein Alignment ist die eins-zu-eins-Zuordnung von Nucleotiden (bzw. Aminosäuren) zwischen verschiedenen Sequenzen, wobei die Reihenfolge der Nucleotide erhalten bleibt. Es gibt dabei sogenannte paarweise Alignments, bei denen zwei Sequenzen miteinander verglichen werden und multiple Alignments, bei denen drei oder mehr Sequenzen gleichzeitig miteinander verglichen werden. Dieser Abschnitt beschäftigt sich mit paarweisen Alignments. Multiple Alignments werden im nächsten Abschnitt behandelt.

Abbildung 2 bis Abbildung 4 zeigen einige Alignments und erklären die Begriffe und Konzepte, die wir benutzen, um die Qualität von Alignments miteinander zu vergleichen.

Ein Alignment gilt dabei als umso besser:

- je mehr Übereinstimmungen (engl. *matches*) es zwischen den Sequenzen erzeugt bzw. je weniger *mismatches* es generiert.
- je weniger Lücken bzw. Lehrstellen (engl. *gaps*) es enthält.
- je kürzer die *gaps* sind.

Gute Alignments entstehen dabei durch eine ausgewogene Balance zwischen den oben genannten Qualitätskriterien. Die ausschliessliche Optimierung eines einzelnen Kriteriums resultiert hingegen häufig in unsinnigen Alignments. Z.B. kann man durch Einführen entsprechend vieler *gaps* praktisch immer ein Alignment mit perfekten *matches* erzeugen (vorausgesetzt eine der Sequenzen ist

a

Sequenz 1
GTATCCTAGAGATATAATGC

Sequenz 2
GACTGTG

Bad Alignment

```
Seq 1  GTATCCTAGAGATATAATGC
       | | | | |   |     |
Seq 2  G-A-C-T-G---T-----G-
```

b

Sequenz 1
TCGATAACTAGTATATATA

Sequenz 2
CGATAACTATATATATA

Bad Alignment

```
Seq 1  TCGATAACTAGTATATATA
         |||||||||
Seq 2  -CGATAACTATATATATA
```

Abbildung 4 *Durch Übergewichtung eines einzelnen Qualitätskriteriums entstehen oft unsinnige Alignments, bei denen es unwahrscheinlich ist, dass sie ein biologisch relevantes Verwandtschaftsverhältnis der beiden Sequenzen reflektieren. Panel a zeigt ein unsinniges Alignment bei dem die Zahl der matches dadurch optimiert wurde, dass eine Vielzahl von gaps eingeführt wurde. Panel b zeigt ein Gegenbeispiel: hier wurde eine grosse Anzahl potentieller matches (blau) nicht realisiert um das Einfügen einer gap zu vermeiden.*

wesentlich länger als die andere). Abbildung 4 zeigt dieses und ein weiteres Beispiel für die unsinnigen Alignments, die aus der Übergewichtung eines einzelnen Qualitätskriteriums resultieren.

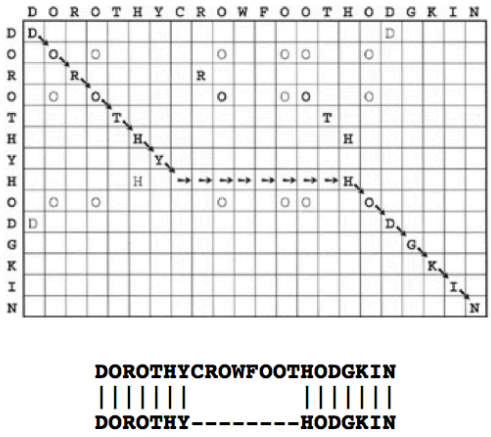
Dot-Plots zeigen alle potentiell möglichen Alignments zwischen zwei Sequenzen

Wie wir bereits gesehen haben, gibt es meist eine Vielzahl von möglichen Alignments und wenn die verglichenen Sequenzen länger werden, wird es zunehmend schwerer, alle potentiell wichtigen Alignments zu erkennen. Dot-Plots (Abbildung 5) bieten eine konzeptionell simple und graphisch intuitive Methode, alle möglichen Alignments zu erfassen.

In einem Dot-Plot wird eine der beiden „Sequenzen“ horizontal und die andere vertikal aufgetragen. In der so geformten Matrize werden dann die Felder, bei denen die Buchstaben in den entsprechenden Zeilen und Spaltenpositionen übereinstimmen, mit eben diesem Buchstaben markiert (in Dot-Plots von längeren Sequenzen markiert man solche Übereinstimmungen nur noch durch einen Punkt, engl. dot, wodurch sich der Name dieser Plots ableitet.)

Dabei entspricht jeder Pfad, der von der oberen linken Ecke zur unteren rechten Ecke des Plots führt und dabei

a




```
DOROTHYCROWFOOTHODGKIN
|||||||         |||||||
DOROTHY--------HODGKIN
```

b

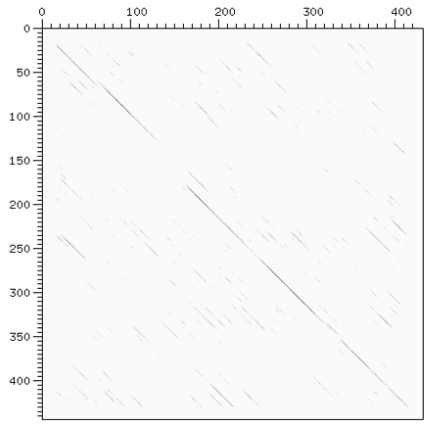


Abbildung 5 *Panel **a** zeigt ein Beispiel eines Dot-Plots. Zu Demonstrationszwecken wird an Stelle einer DNA-Sequenz der Name der berühmten Strukturbiologin Dorothy Crowfoot Hodgkin verwendet (einmal mit und einmal ohne den Mittelnamen) (Abbildung aus Introduction to Bioinformatics, fourth edition by Arthur Lesk).*

*Panel **b** zeigt den Dot-Plot für das menschliche Hämoglobin-Gen im Vergleich zum entsprechenden Gen vom Huhn. Man erkennt das relevante Alignment der Sequenzen als diagonale Reihe von Punkten, erkennt aber auch einige gaps und eine ganze Reihe von alternativen lokalen Alignments.*

nur Schritte nach rechts, nach unten oder nach diagonal rechtsunten benutzt, einem möglichen Alignment und alle möglichen Alignments sind durch Pfade im Dot-Plot darstellbar. Ein Dot-Plot repräsentiert also einen globalen Vergleich der beiden Sequenzen und die Gesamtheit aller möglichen Alignments zwischen den beiden Sequenzen.

In Abbildung 5a ist einer dieser möglichen Pfade durch Pfeile markiert und das diesem Pfad entsprechende Alignment ist unter dem Plot angezeigt.

Wie findet man unter den vielen möglichen Alignments das beste?

Wie wir uns anhand der Dot-Plots verdeutlicht haben, besteht immer eine Vielzahl von möglichen Alignments zwischen zwei Sequenzen. Es stellt sich also die Frage, wie wir sicherstellen können, dass wir in unserer Suche nach dem besten Alignment alle möglichen Alignments in Betracht ziehen und wie wir entscheiden, welches von zwei Alignments wir als besser bewerten.

Um also aus allen möglichen das absolut „beste“ Alignment zu finden, benötigen wir zwei Dinge.

1) Eine **Bewertungsfunktion**, die aus einem Alignment, basierend auf der Anzahl und Art der *mismatches* und *gaps* etc., einen Zahlenwert berechnet, der die relative Qualität dieses Alignments gegenüber anderen Alignments ausdrückt.
2) Einen **Suchalgorithmus**, der aus allen theoretisch möglichen Alignments dasjenige mit der besten Bewertung heraussucht.

Mit der Methode des *dynamic programming* existiert eine Möglichkeit, um mathematisch beweisbar aus allen möglichen Alignments immer dasjenige Alignment zu finden, das den besten *score* der Bewertungsfunktion ergibt.

Die Bewertungsfunktion selber hängt aber von der jeweiligen biologischen Fragestellung ab. Für manche Anwendungen würde man vielleicht *gaps* eher tolerieren als *mismatches* und müsste dann die Bewertungsfunktion entsprechend anpassen.

Lokale Alignments sind oft nützlicher im Auffinden biologischer Zusammenhänge

Bisher haben wir nur Szenarien besprochen, in denen zwei relativ ähnliche, in etwa gleichlange und eher kurze Sequenzen mit einander aligniert wurden. In solchen Fällen sind globale, also solche die die Gesamtheit zweier Sequenzen alignieren, Methoden ideal. Das wohl berühmteste Beispiel für einen solchen *dynamic programming* Algorithmus ist der Needleman-Wunsch-Algorithmus.

In der Forschung gibt es aber auch viele Fragestellungen, bei denen wir erwarten, dass nur Teilabschnitte der beiden verglichenen Sequenzen ähnlich sind. In solchen Fällen macht ein globales Alignment weniger Sinn und man verwendet lokale Alignment-Methoden, bei denen nur die Teile der beiden Sequenzen aligniert werden, die einander auch tatsächlich entsprechen. Der Smith-Waterman-Algorithmus ist ein Beispiel für einen auf lokale Alignments spezialisierter *dynamic programming* Algorithmus.

Heuristische Algorithmen liefern schnelle „pragmatische“ Problemlösungen

Trotz immer schneller werdender Computern sind die oben beschriebenen, formal als optimal beweisbaren, *dynamic programming* Algorithmen für grössere Alignment Probleme (z.B. das Alignment einer Suchsequenz gegen alle Sequenzen in einer Datenbank) zu langsam. Für solch umfangreiche Alignment Anwendungen weicht man daher auf sogenannte heuristische Algorithmen aus.

Heuristische Algorithmen nutzen bei der Lösung eines Problems „Abkürzungen“ und Annahmen, die formal vielleicht nicht ganz korrekt sind und von denen man nicht mit mathematischer Sicherheit weiss, dass sie immer die optimale Lösung finden. Aber aus Erfahrung weiss man, dass diese Algorithmen unter bestimmten Rahmenbedingungen meist sehr gute Lösungen finden und dies sehr schnell tun. Man erkauft sich also die sehr viel höhere Geschwindigkeit durch eine gewisse Unsicherheit, ob eine gefundene Lösung auch wirklich die optimale Lösung ist.

In der Bioinformatik sind heuristische Algorithmen sehr weit verbreitet. Dies ist sicherlich zum Teil dadurch erklärt, dass viele biologische Probleme besonders komplex sind und mit exakten Algorithmen nicht lösbar wären. Vielleicht spielt hier aber auch die Forschungskultur der Biologie eine Rolle, die traditionell pragmatische und effiziente Lösungen gegenüber formal korrekten aber umständlichen Lösungen bevorzugt.

Man muss sich aber bewusst sein, dass die Verwendung von heuristischen Algorithmen dem Nutzer eine gewisse Verantwortung abverlangt. Während man exakte Algorithmen als *black box,* also ohne Verständnis derer Funktionsweisen verwenden kann (man weiss ja, dass sie immer die ideale Lösung finden), empfiehlt sich bei der Verwendung eines heuristischen Algorithmus ein gewisses Verständnis derer Funktionsweise. Nur so kann man sicher sein, dass die gefundene Lösung auch sinnvoll ist.

Der BLAST Algorithmus: eine heuristische Lösung des Alignment Problems

Unter den heuristischen Algorithmen für Sequenzalignments ist der sogenannte BLAST (*basic local alignment search tool*) Algorithmus besonders effizient, findet sehr oft das optimale oder ein nahezu optimales Alignment und ist daher aussergewöhnlich populär.

Wie oben beschrieben, sollten wir heuristische Algorithmen nicht als *black box* verwenden. Deshalb ist die Funktionsweise des BLAST Algorithmus hier etwas eingehender beschrieben.

Eine BLAST-Suche beginnt mit dem Herausfiltern sogenannter *low complexity* Regionen (z.B. AAAAA... oder *GTGTGTGT...)* aus der Suchsequenz.

Die verbleibende Suchsequenz wird dann in kurze sogenannte „Wörter“ zerlegt (für DNA Alignments ist die Standard Wortlänge 11 Nukleotide). Das erste Wort reicht also von Nucleotid 1-11, das zweite von 2-12 das dritte von 3-13 usw. Dann wird jedes dieser Wörter daraufhin untersucht wie wahrscheinlich es per Zufall auftreten würde und nur die relativ unwahrscheinlichen Wörter werden ausgewählt. BLAST sucht dann in der zu durchsuchenden Datenbank nach exakten Übereinstimmungen zu diesen „Wörtern“ – keine *gaps* keine *mismatches.* Wird ein solches Wort in der Datenbank gefunden sucht BLAST von diesem Match aus in beiden Richtungen entlang der Datenbank Sequenz nach Übereinstimmungen zwischen der Datenbank-und der Suchsequenz. *Mismatches* sind jetzt erlaubt aber keine *gaps*. Eine Bewertungsfunktion bestimmt dabei, ob eine weitere Verlängerung dieses lokale Alignment insgesamt verbessert oder nicht. Im letzten Schritt verknüpft der BLAST Algorithmus dann die so gefundenen lokalen Alignments miteinander – aber nur wenn diese Abschnitte nahe beieinanderliegen. Bei diesem letzten Schritt, der dem *dynamic programming* Algorithmus verwandt ist, sind dann auch *gaps* erlaubt. Die Einfügung und Erweiterung dieser *gaps* wird dabei wie gehabt von einer Bewertungsfunktion gesteuert.

Wie wir sehen, vertraut der BLAST Algorithmus darauf, dass zwei biologisch verwandte Sequenzen relativ lange Abschnitte enthalten, in denen diese Sequenzen perfekt übereinstimmen und dass diese Abschnitte durch weniger konservierte Sequenzabschnitte verknüpft sind.

Durch diese Annahme führen wir die Suche nach möglichen Alignments sehr viel schneller aus. Zum einen sind exakte Übereinstimmungen wesentlich schneller zu finden als Ähnlichkeiten zwischen Sequenzen. Zum anderen können weite Bereiche der Suchmatrix, die keine exakten Übereinstimmungen enthalten, verworfen werden wodurch der *search space* wesentlich kleiner gehalten wird. Durch diese beiden „Abkürzungen“ ist der BLAST-Algorithmus um ein Vielfaches schneller als ein *dynamic programming* Algorithmus, verpasst aber auch potentielle Alignments, falls diese Alignments nicht den Erwartungen entsprechen. Wenn z.B. die potenzielle Zielsequenz nicht mindestens einen Abschnitt hat, in dem die geforderte Anzahl der exakten Übereinstimmung vorliegt, findet BLAST dieses Alignment nicht, selbst wenn die Sequenzen insgesamt sehr gut miteinander übereinstimmen.

Der zentrale Kompromiss beim BLAST-Algorithmus ist also zwischen der gewählten „Wort“-Länge und der Geschwindigkeit mit der die Suche stattfindet. BLAST-Suchen mit langen „Wörter“ machen die Suche schnell, verpassen aber Alignments die nicht mindestens einen perfekt übereinstimmenden Sequenzabschnitt enthalten, der länger ist, als die ausgewählte „Wort“-Länge.

Als Nutzer ist es also an uns, die entsprechenden Parameter der Suche so zu wählen, dass die Suche an unsere Fragestellung angepasst ist.

Next Generation Sequencing benötigt extrem effiziente Sequenzalignment-Algorithmen: Burrows-Wheeler-Alignment

Um die genetischen Veränderungen zu finden, die einen bestimmten Phänotyp verursachen, sequenziert man inzwischen häufig das gesamte Genom (*whole genome sequencing* (WGS)) der Individuen (z.B. Fliegen, Menschen, Bakterien, ...), die diesen Phänotyp aufzeigen und vergleicht dieses Genom dann zum Wildtyp- bzw. Referenzgenom. Die dabei verwendeten *next generation sequencing* (NGS) Techniken generieren eine sehr grosse Anzahl (bis zu 1 Milliarde) von relativ kurzen, als *reads* bezeichnete, Sequenzen (~100 bp Länge) die dann anhand eines Referenzgenoms wieder zusammengesetzt werden. Praktisch bedeutet dies, dass bis zu einer Milliarde dieser 100 bp langen Sequenzen mit dem Referenzgenom (einige Milliarden Basenpaare lang) aligniert werden müssen – also ein phänomenales Sequenzalignment-Problem.

Traditionelle Alignment-Methoden, die jeden einzelnen Read gegen das gesamte Genom alignen, sind für diese Aufgabe viel zu langsam.

Man ist daher dazu übergegangen, den ersten und geschwindigkeitsbestimmenden Schritt im Sequenzalignment, d.h. die Suche nach den in den Reads vorkommenden Sequenzwörten (siehe den Abschnitt zu BLAST), nicht im Genom selber, sondern in einem Index des Genoms durchzuführen. Ein solcher Index ist konzeptionell vergleichbar mit einem Einwohnerregister, das die Namen aller Einwohner in alphabetischer Reihenfolge speichert und zu jedem Namen die dazugehörige Adresse angibt. Aufgrund der alphabetischen Reihenfolge kann man bei einer Suche in einem Index sehr schnell zu dem gesuchten Namen springen und so die Adresse finden. Das ist viel schneller als die BLAST-äquivalente Suche bei der man alle Adressen durchgeht, um zu testen ob die gesuchte Person dort wohnt.

Natürlich muss ein solcher Index zunächst generiert werden was auch Rechenaufwand benötigt, aber diese Investition wird um ein Vielfaches wieder amortisiert, wenn man dann Millionen oder gar Milliarden von Suchen gegen diesen Index durchführt.

Der sogenannte Burrows-Wheeler-Index ist ein besonders effizienter Index für lange DNA-Sequenzen (Genome) und sehr viele Sequenzalignment-Algorithmen für NGS-Anwendungen benutzen Burrows-Wheeler-Indizes für die Suche nach exakten Übereinstimmungen zu Sequenzwörten, mit denen der Alignmentprozess beginnt.

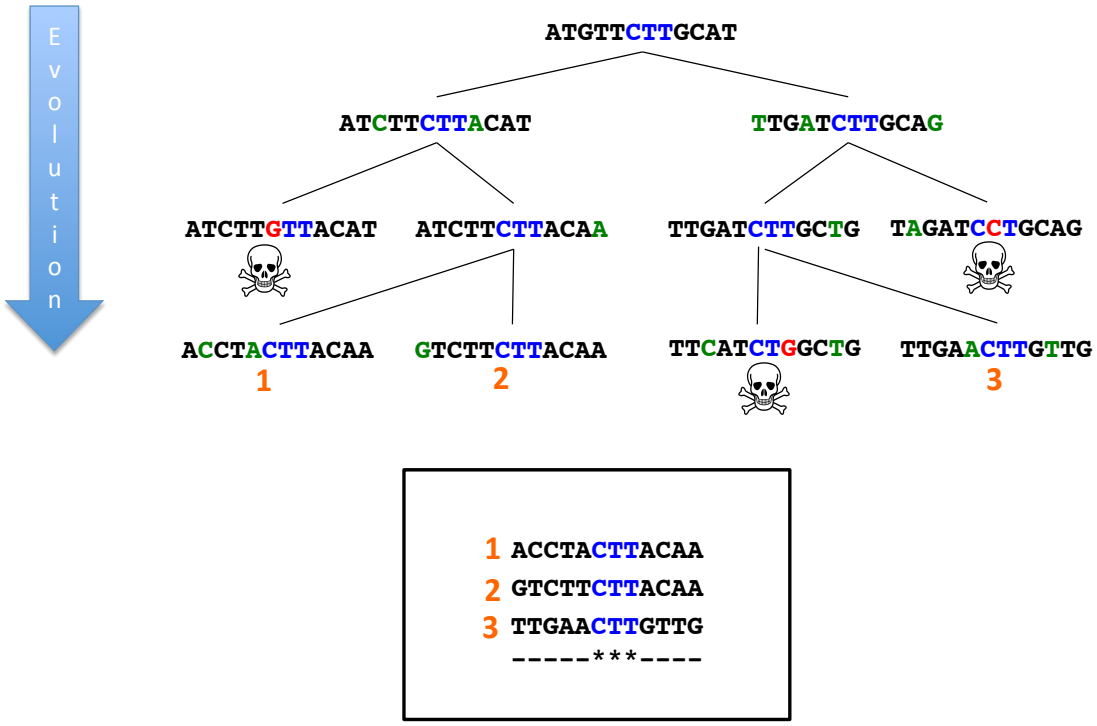


Abbildung 6 *Schematische Darstellung des Selektionsprozesses, der zur Konservierung von funktional wichtigen Sequenzelementen führt. Mutationen im funktional wichtigen Sequenzabschnitt (blau) zerstören dessen Funktion, reduzieren die Fitness des Organismus und verschwinden daher aus dem Genpool. Das MSA (Box) der überlebenden Sequenzen zeigt die Konservierung des funktional wichtigen Sequenzabschnitts.*

Die nachfolgenden Schritte sind dann vergleichbar mit denen im BLAST-Algorithmus.

Multiples Sequenzalignment (MSA)

> „One sequence plays coy, a pair of homologous sequences whispers but many aligned sequences shout out loud.“
>
> - Arthur Lesk

Sequenzkonservierung deutet auf funktionale Wichtigkeit hin

Wie das Zitat von Arthur Lesk andeutet, ist der gleichzeitige Vergleich mehrerer Sequenzen besonders dann hilfreich, wenn es darum geht Sequenzelemente zu finden, die funktional wichtig sind. Die dem zugrundeliegende Idee ist, dass Zufallsmutationen über den Lauf der Evolution dazu führen sollten, dass die Sequenzen eines Gens immer weiter von der Sequenz des Vorgängergens divergieren. Diesem Prozess sind aber dadurch Grenzen gesetzt, dass diese Sequenz weiterhin seine Funktion ausüben muss. Eine Mutation, die eine (lebens-) wichtige Funktion dieser Sequenz zerstört, ist dabei ein Nachteil für den Organismus der diese Mutation trägt und führt dazu, dass solche Mutationen aus dem Genpool verschwinden.

Wenn man homologe Sequenzabschnitte (also Sequenzabschnitte, die von der selben Vorgängersequenz abstammen und weiterhin die selbe molekulare Funktion ausüben) aus mehreren Organismen vergleicht, sind darin die funktional wichtigen Sequenzelemente stärker konserviert als die funktional unwichtigen. Dieser Prozess ist in stark schematisierter Form in Abbildung 6 gezeigt.

Die Fähigkeit solche Konservierungsmuster zu erkennen wächst dabei sowohl mit der Anzahl der gemeinsam untersuchten Sequenzen, als auch mit der evolutionären Entfernung zwischen diesen Sequenzen. Mit der riesigen Menge der in den Datenbanken zur Verfügung stehenden Sequenzdaten von evolutionär sehr weitverzweigten Spezies haben wir also potentiell eine sehr hohe statistische Aussagekraft diese Muster zu erkennen.

Je weiter der evolutionäre Abstand zwischen zwei Sequenzen ist, desto schwerer ist es aber auch zu erkennen, ob diese Sequenzen zu einander homolog sind. Bei der Zusammenstellung eines MSAs brauchen wir also Methoden, die hohe Sensitivität mit hoher Selektivität verknüpfen. Sensitivität ist dabei die Fähigkeit, homologe Sequenzen selbst dann zu erkennen, wenn sie nur sehr ent-

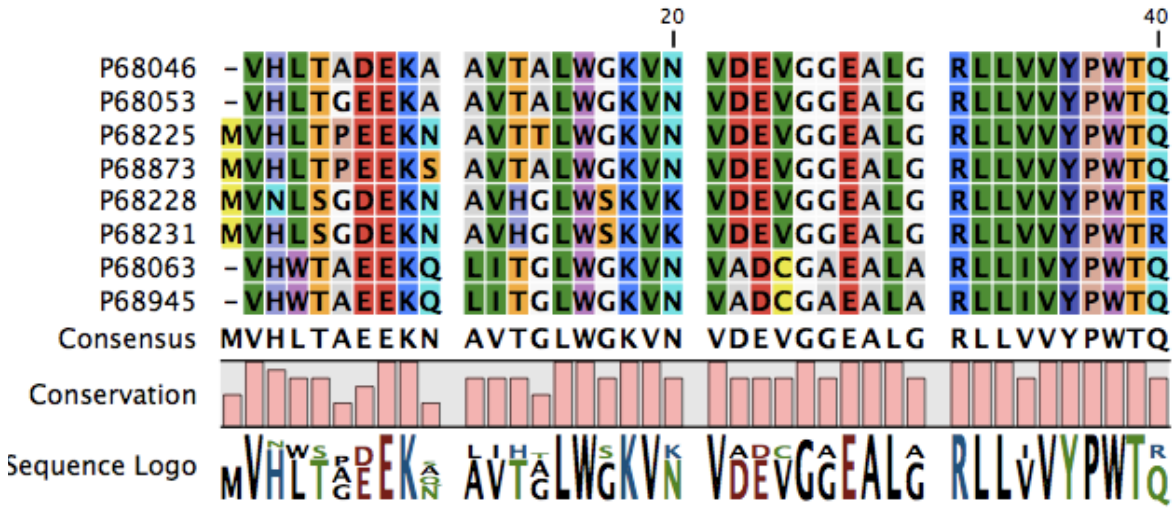


Abbildung 7 *Beispiel eines multiplen Sequenz Alignments von acht homologen Proteinsequenzen. Einige der Aminosäurepositionen sind perfekt konserviert während andere eine gewisse Variabilität aufzeigen. Die Logo-Darstellung (unterste Zeile) verdeutlicht die vorhandene Variabilität bzw. Konservierung in besonders anschaulicher Weise.*

fernt verwandt sind. Dies liefert die Diversität der Sequenzen, die es ermöglicht, die wirklich essentiellen Sequenzelemente zu bestimmen. Selektivität, also die Fähigkeit nicht miteinander verwandte Sequenzen auszuschliessen, vermeidet dabei, dass ein Sequenzmuster durch nicht homologe Sequenzen verwässert wird.

MSAs werden besonders häufig und erfolgreich auf Proteinsequenzen angewendet. Die grössere Anzahl von 20 möglichen Aminosäuren gegenüber 4 möglichen Basen gibt den Sequenzen einen höheren Informationsgehalt und erleichtert dadurch das Alignment von entfernt verwandten Sequenzen. Für proteinkodierende DNA-Sequenzen verwendet man für die Alignments daher in der Regel nicht die DNA-Sequenz selber, sondern deren Übersetzung in die entsprechende Proteinsequenz.

Formal beweisbare Algorithmen für MSA sind rechnerisch zu aufwendig

Im Prinzip könnten dieselben *dynamic programming* Algorithmen, die wir bei paarweisen Alignments kennengelernt haben, auch auf multiple Alignments angewendet werden. Der rechnerische Aufwand nimmt aber mit jeder zusätzlichen Sequenz derart dramatisch zu, dass diese formal beweisbaren Algorithmen für das Alignment mehrerer Sequenzen nicht geeignet sind.

MSAs basieren also generell auf heuristischen Methoden, die sich in ihrer Komplexität und rechnerischem Aufwand erheblich unterscheiden und unterschiedlich gut für unterschiedliche Probleme geeignet sind. Im Folgenden sind einige der populären Ansätze besprochen.

Progressive Alignment Algorithmen

Eine konzeptionell einfache und rechnerisch wenig aufwendige Methode zur Erstellung von MSAs ist das progressive Alignment, wie es z.B. im weiterhin populären Programm CLUSTALW implementiert ist.

Zwei Sequenzen können immer dann besonders gut mit einander aligniert werden, wenn sie sich besonders ähnlich sind. Dies ist die Basis von progressiven Alignment Algorithmen. Man bestimmt dabei durch paarweise Alignments die zwei einander ähnlichsten Sequenzen und aligniert diese mit einander. Weitere Sequenzen werden dann progressive gegen dieses Alignment aligniert, so dass das MSA mit jedem Zyklus wächst.

Das zentrale Problem von progressiven Alignments ist, dass „Fehler“ in den ersten Alignmentzyklen später nicht mehr korrigiert werden können. Dies bedeutet, dass man in der Lage sein muss, die ersten Sequenzen schon perfekt zu alignieren, obwohl man noch nicht auf die in den anderen Sequenzen enthaltene Information zugreifen kann. Progressive Alignment-Algorithmen wie CLUSTALW funktionieren in der Regel dann besonders gut, wenn die zu alignierenden Sequenzen untereinander recht ähnlich sind.

Für anspruchsvollere Alignment-Aufgaben, wenn die Sequenzen also stärker divergieren, gelten progressive Alignment-Algorithmen aber nicht mehr als zeitgemäss.

Iterative Alignment Algorithmen

Iterative Alignment-Algorithmen umgehen die Probleme der progressiven Algorithmen dadurch, dass das Alignment der zuerst alignierten Sequenzen nach der Zugabe von weiteren Sequenzen überprüft und wenn notwendig angepasst wird. Die dazu notwendigen iterativen Zyklen

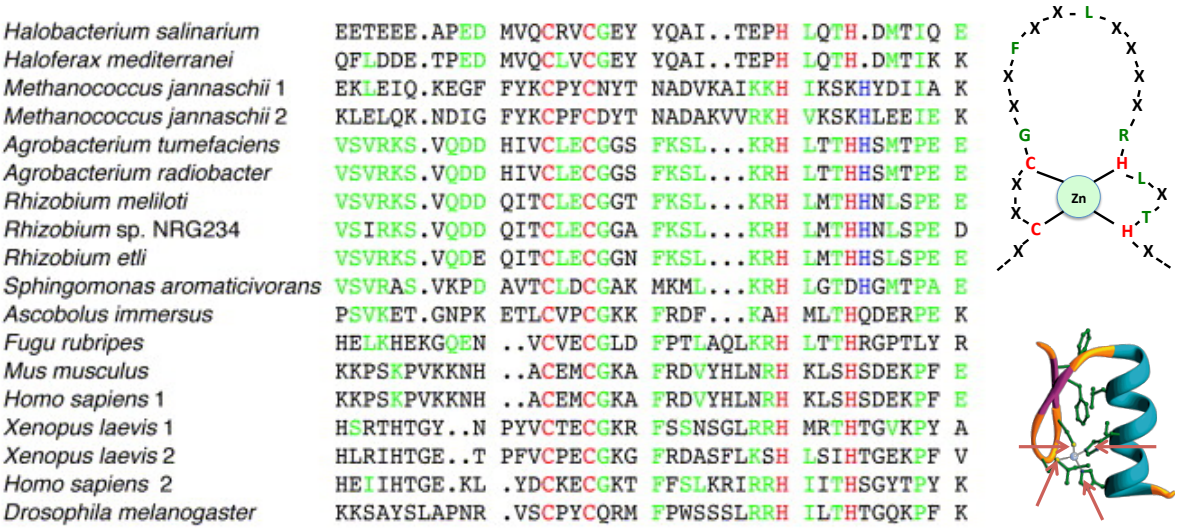


Abbildung 8 *Ausschnitt aus einem profil-basierten multiplen Sequenzalignment von evolutionär weit voneinander entfernten Proteinen aus der Familie der Zinkfinger-Transkriptionsfaktoren. Die für die Funktion absolut essentiellen Aminosäuren, zwei Cysteine (C) und zwei Histidine (H), sind in rot bzw. blau angezeigt. Häufig aber nicht immer konservierte Aminosäuren sind in grün angezeigt. Daneben ist die molekulare Struktur eines Mitglieds dieser Familie schematisch (oben) und als Ribbon-Diagramm (unten) gezeigt. Die Struktur zeigt, dass die in rot angezeigten Aminosäuren eine essentielle Funktion haben: Sie binden das Zink Ion, welches die Struktur des Zinkfingers stabilisiert. Eine Mutation dieser vier Aminosäuren würde die Zink Bindungsstelle und damit die Struktur und die Funktion dieses Proteins komplett zerstören.*

erhöhen den rechnerischen Aufwand, ermöglichen aber auch das Alignment von stärker divergierten Sequenzen. MUSCLE ist eine der erfolgreichsten Implementierungen des iterativen Alignment Ansatzes und ist auch als Webservice[10] verfügbar.

Hidden Markov Model (HMM) Algorithmen

Hidden Markov Models sind eine Klasse von statistischen Modellen, die ursprünglich aus der Informatik (Handschrifterkennung, Spracherkennung usw.) kommen, aber zunehmend in der Bioinformatik Anwendung finden. Die detaillierte Funktionsweise von HMM Algorithmen ist für Nicht-Mathematiker nur schwer nachvollziehbar, aber HMM-basierte MSA Programme, wie z.B. HMMER liefern derzeit die wohl besten Alignments. Dies gilt auch für selbst sehr stark divergierten Sequenzen. Allerdings benötigen sie für grössere Alingnmentprobleme um Grössenordnungen längere Rechenzeit. Für kleine Alignmentprobleme arbeiten diese Algorithmen aber inzwischen annähernd in Echtzeit. Ein populäres HHM-basiertes multiple sequence alignement Program ist ClustalOmega[11].

Profil-basiertes Alignment

Eine weitere sehr erfolgreiche Spielart von Alignment-Methoden sind die profil-basierten (auch *motif finding* genannten) Algorithmen. Anstatt das Alignment über die gesamte Sequenz zu optimieren, suchen diese Algorithmen nach relativ kurzen Motiven, die aber sehr stark konserviert sind. Profil-basierte Alignments sind besonders bei der Suche nach funktional und strukturell wichtigen Sequenzelementen in Proteinen erfolgreich und erlauben es oft extrem stark divergierte Sequenzen mit verblüffender Genauigkeit einer bestimmten Funktion zu zuordnen.

Phylogenetische Bäume

Phylogenetische Bäume rekapitulieren evolutionäre Prozesse

Wir vergleichen und alignieren Sequenzen von DNA und Proteinen miteinander, um deren funktionale und evolutionäre Beziehung untereinander zu verstehen.

Ein phylogenetischer Baum fasst diese evolutionären und funktionalen Verwandtschaftsverhältnisse zwischen mehreren Sequenzen grafisch zusammen und ermöglicht es dadurch selbst relativ komplexe Beziehungen zu verstehen, die aus einem Sequenzalignment nur schwer ersichtlich wären.

Hier besprechen wir molekulare phylogenetische Bäume, welche die evolutionäre Beziehung zwischen spezifischen DNA-, Protein- oder RNA Sequenzen darstellen. Insgesamt sollten diese molekularen Bäume die generellen phylogenetischen Bäume rekapitulieren, welche die

[10] http://www.ebi.ac.uk/Tools/msa/muscle/

[11] http://www.ebi.ac.uk/Tools/msa/clustalo/

a

```
cone_opsin_blau_Mensch          ATLRYKKLRQPLNYILVNVSFGGFLLCIFSVFPVFVASCNGYFVFGRHVCALEGFLGTVA
rhodopsin_Meerschweinchen       VTVQHKKLRTPLNYILLNLAVANLFMVLGGFTTTLYTSMNGYFVFGPTGCNLEGFFATLG
rhodopsin_Kaninchen             VTVQHKKLRTPLNYILLNLAVADLFMVLGGFTTTLYTSLHGYFVFGPTGCNVEGFFATLG
rhodopsin_Totenkopfaffe         VTVQHKKLRTPLNYILLNLAVADLFMVFGGFTTTLYTSLHGYFVFGPTGCNLEGFFATLG
rhodopsin_Mensch                VTVQHKKLRTPLNYILLNLAVADLFMVLGGFTSTLYTSLHGYFVFGPTGCNLEGFFATLG
rhodospin_Meerkatze             VTVQHKKLRTPLNYILLNLAVADLFMVFGGFTSTLYTSLHGYFVFGPTGCNAEGFFATLG
rhodopsin_Rhesusaffe            VTVQHKKLRTPLNYILLNLAVADLFMVFGGFTTTLYTSLHGYFVFGPTGCNAEGFFATLG
rhodopsin_Maus                  VTVQHKKLRTPLNYILLNLAVADLFMVFGGFTTTLYTSLHGYFVFGPTGCNLEGFFATLG
rhodsopsin_Schaf                VTVQHKKLRTPLNYILLNLAVADLFMVFGGFTTTLYTSLHGYFVFGPTGCNLEGFFATLG
rhodopsin_Katze                 VTVQHKKLRTPLNYILLNLAVADLFMVFGGFTTTLYTSLHGYFVFGPTGCNLEGFFATLG
rhodopsin_Flughund              VTVQHKKLRTPLNYILLNLAVADLFMVFGGFTTTLYTSLHGYFVFGPTGCNLEGFFATLG
                                .*:::**** ******:*::.. ::: : ..  .: :* .******   *  ***:.*:.
```

b

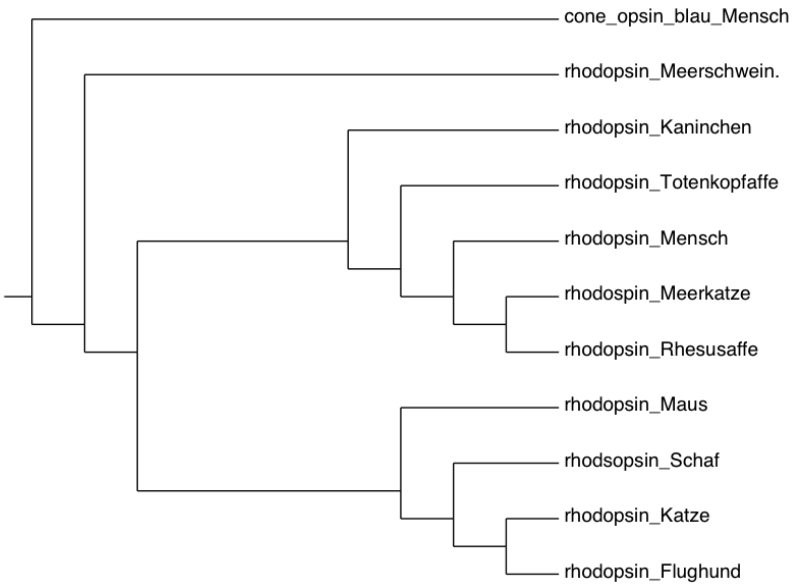


Abbildung 9 *(**a**) Ausschnitt aus einem multiplen Sequenzalignement von Rhodopsin- Sequenzen aus mehreren Säugetieren. Rhodopsine sind die am Sehprozess beteiligten Photorezeptor-Proteine. (**b**) Der aus diesem Alignment erstellte phylogenetische Baum. Die horizontale Achse repräsentiert dabei den Verlauf der Evolution. Sequenzen die enger miteinander verwandt sind, liegen näher zusammen. Jede Verzweigung repräsentiert eine gemeinsame Vorgängersequenz.*

Verwandtschaft zwischen unterschiedlichen Spezies beschreiben. Aber Prozesse wie *gene conversion,* Genduplikation, Gentransfer oder funktionale Selektion führen dazu, dass sich molekulare Bäume nicht eins-zu-eins aus den generellen phylogenetischen Bäumen herleiten lassen. Man muss molekulare phylogenetische Bäume also für jede Fragestellung wieder neu erstellen.

Unterschiedliche phylogenetische Bäume beantworten unterschiedliche biologische Fragestellungen.

Es gibt eine Vielzahl von Arten phylogenetische Bäume darzustellen. Einige der Unterschiede sind rein kosmetischer Natur (z.B. ob der Baum horizontal oder vertikal angezeigt ist oder ob die Zweige gerade oder gekrümmt dargestellt sind).

Andere Unterschiede sind aber funktionaler Art, so gibt es z.B. Unterschiede in der Information, die dargestellt wird. So kann z.B. die Distanz zwischen der Wurzel und den „Blättern“ willkürlich gleich lang gehalten werden (siehe z.B.Abbildung 9). Ein solcher Baum ist besonders gut geeignet um evolutionäre Prozesse darzustellen. Alternativ kann die Länge der Äste benutzt werden um explizit die Anzahl der Mutationen zwischen zwei Punkten darzustellen. Ein weiterer konzeptioneller Unterschied besteht zwischen Bäumen mit Wurzeln, die versuchen explizit einen evolutionären Prozess abzubilden, und „wurzellosen“ Bäumen (siehe z.B.Abbildung 10), die eher darauf abzielen, die Ähnlichkeit zweier Sequenzen zu vergleichen.

Algorithmen zur Erstellung phylogenetischer Bäume

Wie für alle anderen bisher besprochenen Anwendungen gibt es auch bei der Erstellung von phylogenetischen Bäumen mehrere unterschiedliche Algorithmen.

Genau wie beim Sequenzalignment gibt es auch hier wieder Methoden die aufwendiger und potentiell genauer

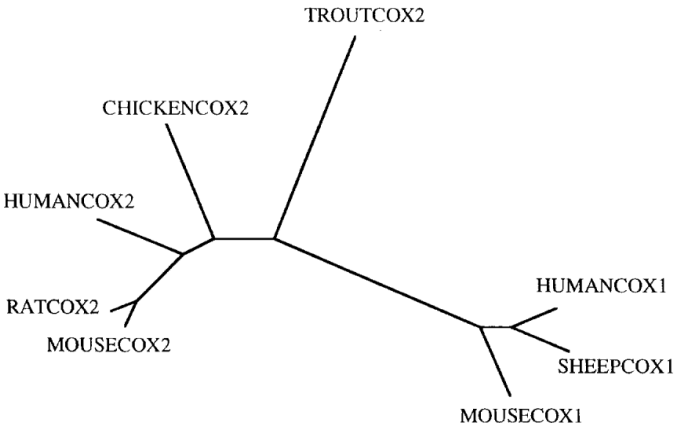


Abbildung 10 *Beispiel eines wurzellosen phylogenetischen Baums bei dem die relative Entfernung von zwei „Blättern" das Ausmass der Unterschiede zwischen den entsprechenden Sequenzen anzeigt. Die COX-Gene sind an der Produktion von Prostaglandinen beteiligt.*

sind und solche die rechnerisch einfacher sind, aber auf bestimmten vereinfachenden Annahmen beruhen.

Maximum-Likelihood-basierte Algorithmen bilden alle potentiell möglichen Bäume und berechnen für jeden dieser Bäume die Wahrscheinlichkeit, dass bei der von diesem Baum repräsentierten evolutionären Geschichte die beobachteten Sequenzen auftreten würden. Die Anzahl der potentiell möglichen Bäume wächst mit zunehmender Anzahl der Sequenzen sehr schnell, sodass die Maximum-Likelihood Methoden für Bäume aus einer grossen Anzahl von Sequenzen rechnerisch sehr aufwendig werden kann.

Parsimony-basierte Algorithmen suchen nach Bäumen und Ursprungssequenzen durch welche die in der Gegenwart beobachteten Sequenzen mit der kleinstmöglichen Anzahl von Mutationen erzeugt werden können. Dieser Ansatz eignet sich besonders gut für eng miteinander verwandte Sequenzen, trifft für weiter verwandte Sequenzen (z.B., wenn mehrfache Mutationen an derselben Position auftreten) aber auf Schwierigkeiten.

Entfernungs-basierte phylogenetische Bäume

Für den täglichen Gebrauch verwendet man daher häufig die konzeptionell einfachen Entfernungs-basierten Methoden (engl. *distance based methods*). Diese Methoden basieren zwar auf Annahmen von denen wir wissen, dass diese nicht wirklich korrekt sind, aber sie liefern mit geringem rechnerischen Aufwand relativ zuverlässige Resultate.

Diese Algorithmen erstellen eine Tabelle aller paarweisen „Entfernungen" zwischen den verwendeten Sequenzen (siehe Abbildung 10). Diese „Entfernungen" werden dabei häufig durch den *score* des Smith-Waterman-Algorithmus (genannt SW-Score; für eine Einführung des SW-Algorithmus siehe Abschnitt Sequenzalignment) für das Alignment der zwei Sequenzen bestimmt (genau genommen berechnet man den SW-Score eines Alignments der längeren Sequenz gegen sich selbst und subtrahiert davon den SW-Score des Alignments der längeren gegen die kürzeren Sequenzen).

Man versucht dann einen zweidimensionalen Baum zu finden, der alle diese Entfernungen reflektiert.

Für Bäume von mehr als drei Sequenzen wird oft eine Situation auftreten in der es keine zweidimensionale Anordnung von Punkten gibt, welche die berechneten Entfernungen exakt reflektiert. Die Aufgabe der entfernungs-basierten phylogenetischen Algorithmen ist es nun einen Kompromiss zu finden, der die gemessenen Distanzen so gut wie möglich in einer zweidimensionalen Ebene repräsentiert.

Eine der populärsten Lösungen für diese Aufgabe ist der UPGMA-Algorithmus (von engl. *unweighted pair-group method using arithmetric averages*). Dieser Algorithmus basiert auf der Annahme, dass die genetische „Entfernung" zwischen zwei Sequenzen, also vereinfacht ausgedrückt die Anzahl der Mutationen zwischen den Sequenzen, proportional zu der Zeit ist, die seit dem letzten gemeinsamen Vorgänger dieser zwei Sequenzen vergangen ist. Mit anderen Worten der Algorithmus nimmt eine konstante Mutationsrate an.

Verwendung einer Outgroup verbessert die Qualität von phylogenetischen Bäumen

Durch praktische Erfahrungen hat sich z.B. herausgestellt, dass viele Algorithmen für die Erstellung von phylogenetischen Bäumen wesentlich besser funktionieren, wenn neben eng miteinander verwandten Sequenzen auch eine etwas weiter entfernte Sequenz mit untersucht wird. Diese weiter entfernte Sequenz wird als Outgroup bezeichnet. Solche Finessen finden sich in fast allen Anwendungen der hier aufgezählten Tools und sind oft nicht auf den ersten Blick ersichtlich. Man kann und sollte jedoch von Online-Foren und erfahrenen Kollegen profitieren um die komplette Bandbreite der bioinformatischen Möglichkeiten ausnutzen zu können.