ZSL
Zentrum für Schulqualität
und Lehrerbildung
Baden-Württemberg

Bioinformatik

Handreichung zur Einführung des Bildungsplans im Beruflichen Gymnasium ab Schuljahr 2021/2022

ZSL – 2021

Redaktionelle Bearbeitung

Redaktion	Dr. Hauke Holtorf, Zentrum für Schulqualität und Lehrerbildung (ZSL)
Autor/in	Thomas Dürr, Augusta-Bender-Schule Mosbach Dr. Hauke Holtorf, Albert-Schweitzer-Schule Villingen-Schwenningen Klaus Steiner, Mildred-Scheel-Schule Böblingen Katharina Voigt, Bertha-von-Suttner-Schule Ettlingen
Erscheinungsjahr	2021

Impressum

Herausgeber

Land Baden-Württemberg
vertreten durch das Zentrum für Schulqualität und Lehrerbildung (ZSL)
Heilbronner Straße 314, 70469 Stuttgart
Telefon: 0711 21859-0
Telefax: 0711 21859-701
E-Mail: poststelle@zsl.kv.bwl.de
Internet: www.zsl-bw.de

Inhaltsverzeichnis

1 Allgemeine Vorbemerkungen zum neuen Bildungsplan

Bei der Erstellung des Bildungsplans Bioinformatik wurde berücksichtigt diesen inhaltlich vom Fach Informatik absetzen, um den speziellen interdisziplinären Charakter der Bioinformatik mit Querverweisen zu den Bildungsplänen der Fächer Biotechnologie, Biologie und Chemie deutlich zu machen.

Es kann nicht ausgeschlossen werden, dass sich die Zusammensetzung der Schülerschaft im Fach Bioinformatik je nach Schulstandort aus unterschiedlichen Richtungen des Beruflichen Gymnasiums zusammensetzen wird. Die speziellen Fachkenntnisse aus dem Profilfach Biotechnologie des Biotechnologischen Gymnasiums können daher nicht vorausgesetzt werden. Vor diesem Hintergrund wurden die einzelnen Bildungsplaneinheiten (BPE) entsprechend gestaltet. Auch wurde darauf geachtet, dass der Unterricht in den Jahrgangsstufen 1 und 2 auch ohne Vorkenntnisse aus der Eingangsklasse erfolgen kann. Außerdem wurde bei der Erstellung des Bildungsplans berücksichtigt, dass die Inhalte der Jahrgangsstufe 2 auch ohne Vorkenntnisse aus der Jahrgangsstufe 1 und umgekehrt unterrichtet werden können.

Im Gegensatz zum bisher gültigen Lehrplan wurden neue Themen in diesen Bildungsplan integriert, die einerseits die verstärkte Nutzung digitaler Medien abbilden und andererseits deren Verwendung zur Strukturierung und Vernetzung von Daten darstellen.

So wurde beispielsweise in der Eingangsklasse das neue Thema Digitales Laborjournal aufgenommen, um den Trend zur Digitalisierung im Laborbereich abzubilden als auch den strukturierten Austausch von Daten über eine digitale Plattform beispielhaft zu demonstrieren. Zusätzlich ist das Erstellen von Animationen und Lernvideos zur Erläuterung von zell- oder molekularbiologischen Vorgängen neu hinzugekommen, um den Schülerinnen und Schülern die modellhafte Darstellung von komplexen zellulären oder molekularen Vorgängen zu ermöglichen.

Die neue Bildungsplaneinheit 10 (Stammbaumrekonstruktion: Evolution, Taxonomie und molekulare Phylogenie) bietet den Schülerinnen und Schülern die Basis für die Erkenntnis, dass molekulare Daten in Form von biologischen Sequenzen zur Untersuchung der Phylogenie von Organismengruppen herangezogen werden können.

Ein weiteres durchgängiges Unterscheidungsmerkmal zum bisherigen Lehrplan ist die verstärkte Fokussierung auf bestimmte Themen und deren detailliertere, präzisierte Ausarbeitung der zu behandelnden Inhalte.

Das Fach Bioinformatik unterstützt die Entwicklung von digitalen Kompetenzen der Schülerinnen und Schüler und bereitet sie auf eine zunehmend digitalisierte Lebens- und Arbeitswelt vor. Die Auswahl der Themengebiete und die inhaltliche Gestaltung der Bildungsplaneinheiten berücksichtigt die biologischen und informationstechnischen Vorkenntnisse der Schülerinnen und Schüler und ermöglicht eine Weiterentwicklung ihrer fachlichen und personalen Kompetenzen. Die Schülerinnen und Schüler wer-

den außerdem durch die Ausbildung von inhalts- und prozessbezogenen Kompetenzen über eine kompetenzorientierte Gestaltung des Unterrichts im Fach Bioinformatik befähigt, sich aktiv an gesellschaftlichen und wissenschaftlichen Diskussionen zu beteiligen.

Diese Handreichung beschränkt sich auf die beispielhafte Darstellung der drei Bildungsplaneinheiten (BPE) 4, 10 und 11.

In der BPE 4 werden biologische Moleküle, speziell die Nukleinsäuren und Proteine, als zentrale Bestandteile des Informationsspeicher- und Informationsumsetzungssystems von Zellen besprochen. Aus der BPE 4 werden die BPE 4.1 und 4.2 vorgestellt. Die Schülerinnen und Schüler benennen hier zelluläre Biomakromoleküle und deren prinzipielle Funktion. Sie identifizieren die Nukleinsäuren und Proteine als informationstragende und funktionelle Moleküle und beschreiben die fachspezifische Schreibweise zur Darstellung biologischer Sequenzen. Sie beschreiben die Struktur und die Funktionselemente von pro- und eukaryotischen Genen und erklären deren Bedeutung für die Steuerung der Genexpression.

Die Konstruktion molekularer Stammbäume steht in BPE 10 im Vordergrund. Aus dieser Einheit werden die BPE 10.2 bis 10.5 anhand geeigneter Beispiele vorgestellt. Auf Basis der Kenntnisse zur Evolutionstheorie und Taxonomie lernen die Schülerinnen und Schüler die Anwendung molekularer Daten in Form von biologischen Sequenzen zur Aufstellung von Modellen zur Evolution von Organismengruppen kennen. Sie benennen Strukturelemente von Stammbäumen und erläutern die Unterschiede zwischen verschiedenen Stammbaumtypen. Das Konzept der molekularen Uhr ist ebenso Bestandteil der BPE wie die grundlegenden Prinzipien von verschiedenen Verfahren zur Rekonstruktion von Stammbäumen. Anhand geeigneter Programme und verschiedener Verfahren werden unterschiedliche Stammbäume erstellt und Hypothesen über die Validität der einzelnen Ergebnisse entwickelt. Das Prinzip des Bootstrapping wird erläutert.

In der BPE 11 liegt der Schwerpunkt auf der eigenhändigen Erstellung geeigneter Programme, die eine zielgerichtete Sequenzanalyse ermöglichen. Dabei wird beispielhaft gezeigt, dass die Programmierung Möglichkeiten schafft, um neue, bedarfsgerechte Werkzeuge für die bioinformatische Analyse biologischer Sequenzen zu entwickeln. Aus dieser Einheit werden die BPE 11.1 bis 11.3 vorgestellt. Aus dem Fach Informatik werden die Grundkenntnisse der Programmiersprache vorausgesetzt, sodass das Wissen lediglich auf neue bioinformatische Anwendungen übertragen werden muss.

Mit dem Ziel einen Algorithmus zur Analyse von Open Reading Frames (ORFs) als bioinformatisches Werkzeug der Sequenzanalyse zu programmieren, übertragen die Schülerinnen und Schüler zu Beginn der BPE 11 die vorhandenen Grundkenntnisse der Programmierung auf die Anwendung einer Programmiersprache (hier die Programmiersprache Python), die bevorzugt zur spezifischen Bearbeitung von Fragestellungen der Sequenzanalyse eingesetzt wird. Durch die Erweiterung der Grundkenntnisse werden spezifische Elemente nutzbar gemacht, die für die textuelle Mustererkennung als Grundlage einer Sequenzanalyse erforderlich sind. Das Entwickeln einfacher Algorithmen führt am Ende der BPE 11 schließlich zur Implementierung eines Algorithmus zur ORF-Analyse einer gegebenen DNA-Sequenz.

2 Einsatzmöglichkeiten von digitalen Medien im Unterricht

Grundsätzlich findet der Unterricht im Fach Bioinformatik gestützt durch digitale Medien wie PC, Tablet etc. statt. Neben dem standardmäßigen Unterricht mit digitalen Medien können bestimmte Unterrichtseinheiten, z. B. zur Einführung in eine bestimmte Thematik, auch mit „Stift und Papier“ gestaltet werden. Der Unterricht der Bildungsplaneinheit 3 ist insbesondere geeignet, die Inhalte teilweise mithilfe von Tablets umzusetzen. Die für diese Handreichung gewählten Umsetzungsbeispiele finden i. d. R. am PC statt.

3 Umsetzungsbeispiele

3.1 Biologische Moleküle als Informationsträger (BPE 4.1 bis 4.2)

3.1.1 VERLAUFSPLAN/STOFFVERTEILUNG

DAUER	UNTERRICHTSPHASE, INHALT	MATERIAL, MEDIEN	ANGESTREBTES ERGEBNIS, ERWARTETES SCHÜLERVERHALTEN
40	BPE 4.1 Einstieg in das Thema „Biologische Moleküle als Informationsträger“	Informationstext – Polysaccharide, Lipide, Proteinen und Nukleinsäuren als Biopolymere	Die SuS benennen zelluläre Biomakromoleküle und deren prinzipielle Funktion. Sie identifizieren Nukleinsäuren und Proteine als die wesentlichen informationstragenden sowie funktionellen Moleküle und beschreiben deren chemischen Aufbau.
25	BPE 4.1 Strangpolaritäten	Excel-Datei „DNA_Struktur_AB.xlsx“	Die SuS erstellen eigenhändig einen kurzen DNA-Doppelstrang und erkennen die Regeln von komplementärer Basenpaarung und Strangpolarität. Sie nutzen ihr Vorwissen zur Verwendung von Excel-Funktionen (BPE 2.1).
15	BPE 4.1 Strangpolaritäten	WORD-Datei „Aufbau_Struktur_Proteine_AB_u_LÖ.docx“	
10	BPE 4.1 QUIZ zum Informationstext	Excel-Datei „QUIZ_4.1.xlsx“.	Die SuS überprüfen ihr Wissen zu Struktur, Aufbau und Funktion von Nukleinsäuren und Proteinen.

45	BPE 4.2 Einstieg in das Thema „Dogma der Molekularbiologie“	Informationstext – Das Dogma der Molekularbiologie	Die SuS beschreiben die Struktur und die prinzipiellen Funktionselemente von pro- und eukaryotischen Genen, sowie den Fluss der genetischen Information von der DNA über die RNA zum Protein.
20	BPE 4.2 Überschrift A Information zum Thema XY	WORD-Datei „S_D_Konsensus_AB_u_LÖ.docx“	Die SuS bestimmen mithilfe eines Textverarbeitungsdokuments die Konsensus-Sequenzen der Shine-Dalgarno-Sequenz.
25		WORD-Dateien „TATA_Pribnow_Box_AB_u_LÖ.docx“„Pribnow_Konsensus_von_Hand_AB_u_LÖ.docx“	Die SuS bestimmen mithilfe eines Textverarbeitungsdokuments die Konsensus-Sequenzen von TATA- und Pribnow-Box.
45		Wahlweise: Excel-Dateien „Pribnow_Konsensus_AB.xlsx“ und „Pribnow_Konsensus_LÖ.xlsx“. Alternativ: „AATAAA_Konsensus_AB.xlsx“ und „AATAAA_Konsensus_LÖ.xlsx“	Die SuS bestimmen mithilfe eines Rechenblatts die Konsensus-Sequenz der Pribnow-Box (alternativ: AATAAA-Konsensus).
45		Excel-Dateien „Transkription_AB.xlsx“ und „Transkription_LÖ.xlsx“	Die SuS führen eine konzeptionelle Transkription mithilfe eines Rechenblattes durch.
45		Excel-Dateien „Code_und_Dogma_AB.xlsx“ und „Code_und_Dogma_LÖ.xlsx“	Die SuS führen eine konzeptionelle Translation mithilfe eines Rechenblattes durch.

3.1.2 FACHLICHE HINWEISE

Die Analyse von Sequenzdaten ist ein zentrales Thema der Bioinformatik. Die BPE 4 verdeutlicht den Schülerinnen und Schülern die Bedeutung der Proteine und Nukleinsäuren als Bestandteile des Informationsspeicher- und Informationsumsetzungssystems von Zellen. Durch die vergleichende Betrachtung von Genstrukturen lassen sich essenzielle Elemente innerhalb von Sequenzen identifizieren, denn der Informationsgehalt von DNA beschränkt sich nicht nur auf die Codierung von Bauplänen, sondern reguliert die Genexpression der Zelle. Das Speichermedium DNA hat ein enormes Potenzial hinsichtlich technologischer Anwendungen.

3.1.3 DIDAKTISCHE HINWEISE

Die Schülerinnen und Schüler benennen zelluläre Biomakromoleküle und deren prinzipielle Funktion. Sie identifizieren Nukleinsäuren und Proteine als die wesentlichen informationstragenden bzw. funktionellen Moleküle und beschreiben den Aufbau aus den verschiedenen Monomeren.

Die Struktur und die prinzipiellen Funktionselemente von pro- und eukaryotischen Genen sowie die Funktionsweise von DNA als biologischer Informationsspeicher werden anhand von geeigneten Beispielen erarbeitet.

Die Codierungsmöglichkeiten von DNA und anderen informationstragenden Medien werden verglichen und geeignete Strategien entwickelt, die sich zur Umwandlung und Speicherung von Texten in ein DNA-Molekül eignen.

Das Potenzial der DNA als modernes Speicher- und Kennzeichnungsmedium wird exemplarisch dargestellt.

3.1.4 ARBEITSMATERIALIEN/AUFGABEN

3.1.4.1 INFORMATIONSTEXT – PROTEINE UND NUKLEINSÄUREN ALS BIOPOLYMERE (BPE 4.1)

Ein Biopolymer ist ein Polymer, das in der Zelle eines Lebewesens synthetisiert wird. In Form z. B. von Polysacchariden, Proteinen, Nukleinsäuren etc. dienen Biopolymere als Energiespeicher (Glykogen, Stärke), haben strukturelle Funktionen für die Zelle (Lipidmembranen) oder den gesamten Organismus (Cellulose in Fasern), wirken im Stoffwechsel (Stoffaufnahme, -transport, Enzym, Exkretion), erkennen Zustände (Sensorik), bewirken Veränderungen (Muskulatur, Farbwechsel), wehren schädigende Einflüsse aktiv ab (Toxine, Immunologie), speichern oder vermitteln Information (genetisch, hormonell, im Zentralnervensystem).

Proteine und Nukleinsäuren sind die zentralen Bestandteile des Informationsspeicher- und Informationsumsetzungssystems.

In der nachfolgenden Tabelle sind wichtige, in der Natur vorkommende (biogene) Biopolymere mit dem jeweiligen Monomer, der Art der chemischen Bindung und Beispielen zu ihrem Vorkommen aufgeführt.

Biopolymer	Monomer(e)	Bindungstyp	Vorkommen
Proteine und Peptide	Aminosäuren	Amid- bzw. Peptidbindung	in allen Lebewesen als Enzyme, Gerüstsubstanzen, Stofftransportsysteme, Chemorezeptoren usw.
Nukleinsäuren	Nukleotide	Phosphodiesterbindung	DNA in allen Lebewesen und vielen Viren als Informationsträger der Erbsubstanz, RNA z. B. als mobiler Informationsträger (mRNA)

TABELLE 1 WICHTIGE BIOPOLYMERE FÜR DIE BIOINFORMATIK

Nukleinsäuren sind aus einzelnen Bausteinen, den Nukleotiden, aufgebaute Makromoleküle, die bei allen Organismen die genetische Information enthalten. Abwechselnde Einfachzucker und Phosphorsäureester bilden eine Kette, wobei an jedem Zucker eine Nukleinbase hängt. Ihr bekanntester Vertreter ist die Desoxyribonukleinsäure (DNS bzw. DNA), die aus dem Zucker Desoxyribose und den vier Basen: Adenin, Cytosin, Guanin und Thymin besteht. Sie ist der Speicher der Erbinformation. Die Ribonukleinsäure (RNA) besteht aus dem Zucker Ribose und den Basen: Adenin, Cytosin, Guanin und Uracil. Eine wesentliche Funktion der RNA in der Zelle ist die Umsetzung der genetischen Information der DNA in die Proteine bei der Proteinbiosynthese. In Form der mRNA (messenger RNA oder Boten-RNA) fungiert sie hierbei als Informationsüberträger. Daneben erfüllen spezielle RNA-Typen (rRNA, tRNA, u. a.) weitere Aufgaben in der Zelle. DNA und RNA unterschieden sich auch in ihrer Strangarchitektur: DNA-Moleküle treten meist als Doppelstrang auf, wohingegen die RNA nur in Form eines Einzelstrangs vorkommt (Abb. 1).

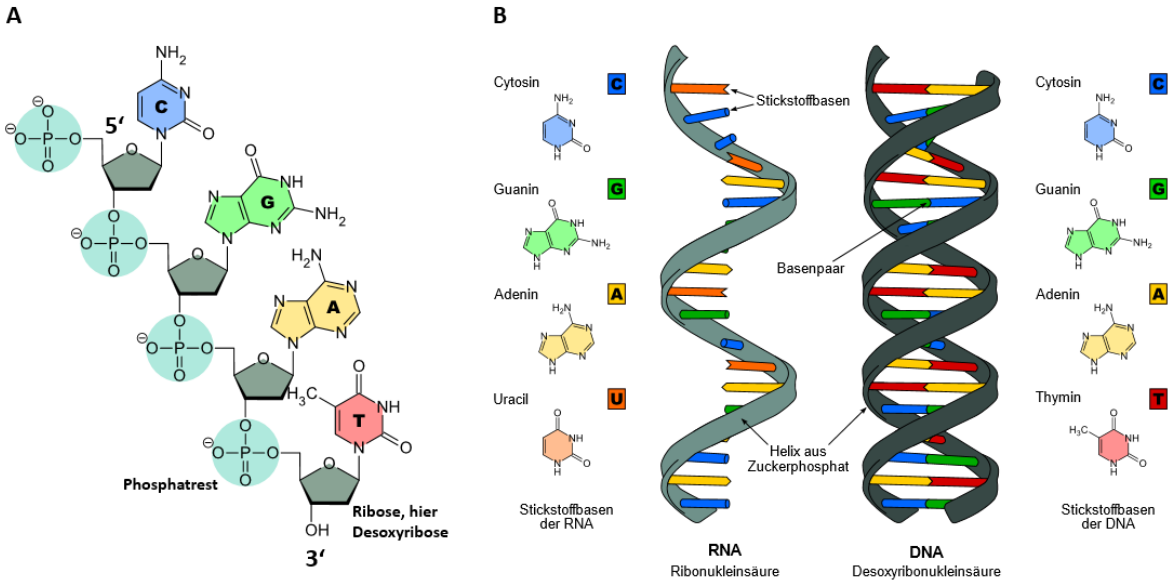


Abb. 1 Struktur von Nukleinsäuren. Nukleinsäurestrang einer DNA mit vier Kettengliedern (A). RNA- und DNA-Aufbau im Vergleich (B).(https://de.wikipedia.org/wiki/Datei:Difference_DNA_RNA-EN.svg, CC BY-SA 3.0, Sponk)

Der Informationsgehalt eines Nukleinsäuremoleküls hängt von der Reihenfolge der Kettenbausteine ab. Dabei sind nur die Basen A, C, G, T und U informationstragend. Der molekulare Aufbau verleiht einer Nukleinsäure immer eine eindeutige Strangpolarität. Sie hat ein 5′-Ende (sprich: 5-Strich-Ende), benannt nach dem C5-Atom des Zuckers, an dem ein Phosphatrest gebunden ist, und ein 3′-Ende, an dem die freie OH-Gruppe am C3-Atom die Kette abschließt. In der Bioinformatik werden einzelsträngige Nukleinsäuresequenzen der Konvention gemäß immer vom 5′ zum 3'-Ende geschrieben bzw. dokumentiert (Abb. 2). In der Zelle spielt die Polarität eine sehr wichtige Rolle. DNA- und RNA-Polymerasen beispielsweise synthetisieren einen neuen DNA-Strang nur in 5′ → 3′-Richtung. Die in einem Nukleinsäuremolekül gespeicherte Information wird also immer in 5′ → 3′-Richtung gelesen.

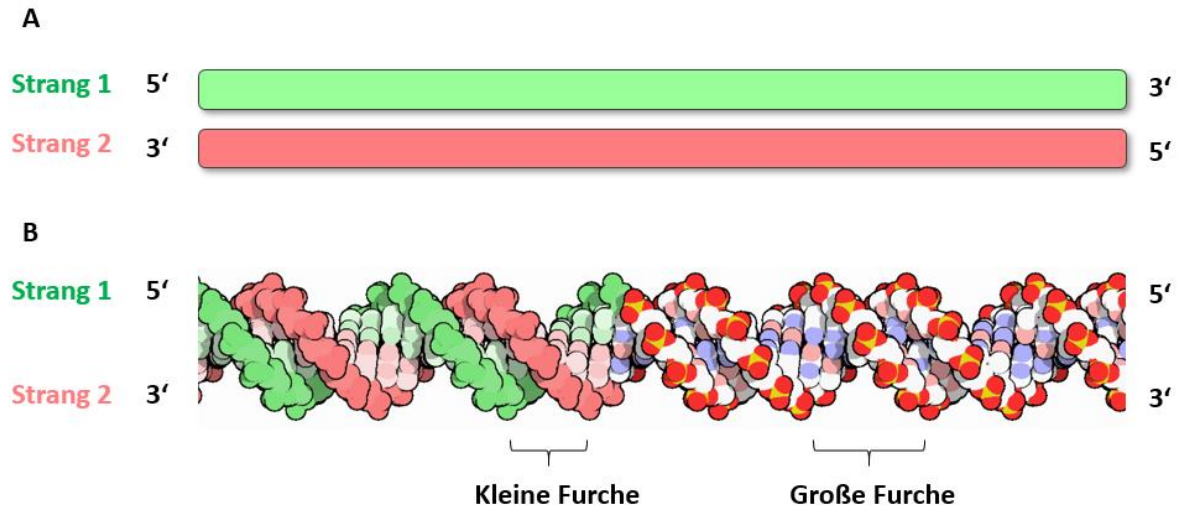


Abb. 2 DNA-Doppelstrang in stark schematisierter Darstellung (A) und als dreidimensional dargestelltes Kalottenmodell (B). Die 5′ → 3′-Strangpolaritäten sind jeweils hervorgehoben.
Aus: The Protein Data Bank H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne (2000) Nucleic Acids Research, 28: 235-242.

DNA-Struktur - Regeln der komplementären Basenpaarung

DNA-Basen	Stoffgruppe
Adenin	Purin
Cytosin	Pyrimidin
Guanin	Purin
Thymin	Pyrimidin

Komplementäre Basenpaarung				
Adenin	A	2	T	Thymin
Cytosin	C	3	G	Guanin
Guanin	G	3	C	Cytosin
Thymin	T	2	A	Adenin

Anzahl der H-Brücken

Übung 1

5'	A	T	G	C	A	G	T	C	T	A	T	C	T	A	G	3'		
	II	II	III	III	II	III	II	III	II	II	II	III	II	II	III		←	Anzahl
3'	T	A	C	G	T	C	A	G	A	T	A	G	A	T	C	5'		H-Brücken

Abb. 3 Screenshot aus der Musterlösung einer Übung zur Doppelstrangstruktur der DNA. Dateinamen: DNA_Struktur_AB.xlsx bzw. DNA_Struktur_LÖ.xlsx, mit Genehmigung von Microsoft.

Ein Protein, auch „Eiweiß“, ist ein biologisches Makromolekül, das aus Aminosäuren aufgebaut ist, die durch Peptidbindungen miteinander verknüpft sind. Kurze Aminosäureketten werden als Peptide bezeichnet.

Proteine finden sich in jeder Zelle – aber auch in Viren. Sie dienen ihr als molekulare „Werkzeuge“ und erfüllen je nach der besonderen Struktur unterschiedliche Aufgaben, indem sie beispielsweise Zellbewegungen ermöglichen, Metabolite transportieren, Ionenpumpen, chemische Reaktionen katalysieren oder Signalstoffe erkennen können. Dementsprechend gibt es Motor- und Gerüstproteine, Transportproteine, Ionenpumpen, Enzyme und Rezeptoren. Abbildung 4 zeigt Beispiele einiger solcher Proteine und deren Kenndaten (Tabelle 2).

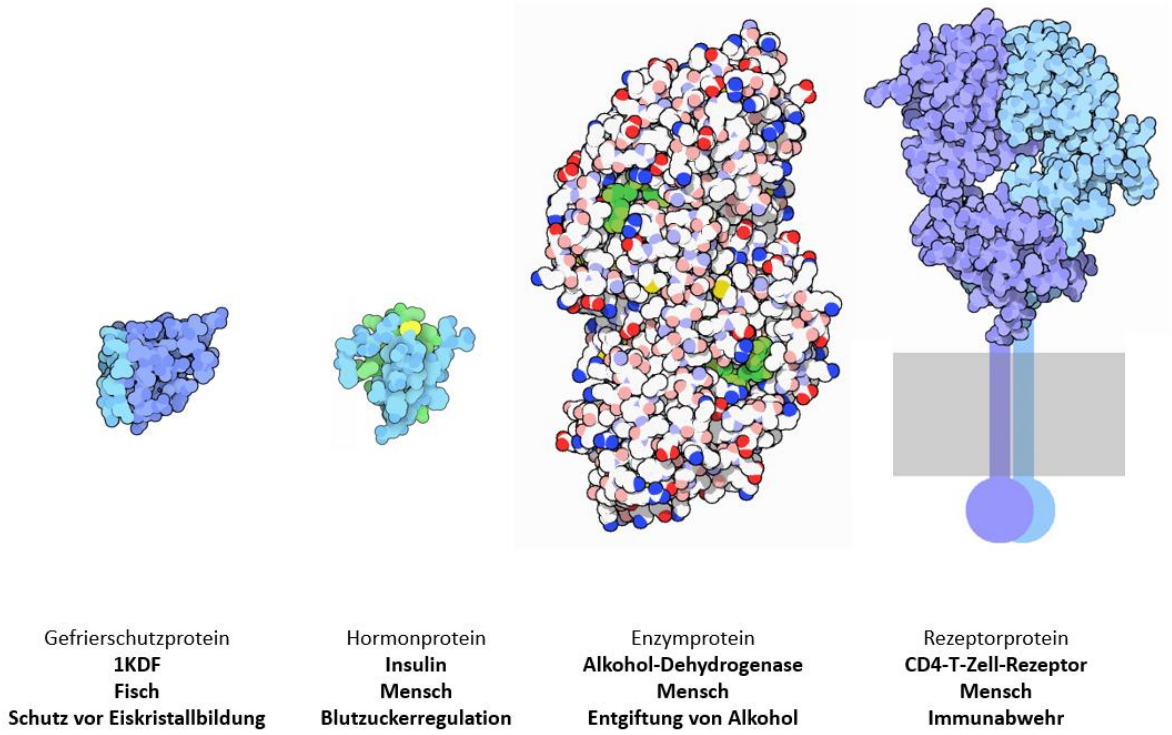


Abb. 4 Beispiele für Proteine mit unterschiedlichen Strukturen und Funktionen.
Aus: The Protein Data Bank H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne (2000) Nucleic Acids Research, 28: 235-242

Tabelle 2 Länge, Molekulargewicht und Struktur unterschiedlicher Proteine aus Abbildung 4

	Gefrierschutz-protein 1KDF	Insulin	Alkohol-Dehyd-rogenase ADH	CD4-T-Cell-Re-zeptor
Länge des Proteins (Anzahl der Aminosäuren)	70	63	374	458
Molekulargewicht des Proteins (in Kilo-Dalton)	7,5	7,2	40,0	51,1
Struktur	Tertiärstruktur	Tertiärstruktur	Quartärstruktur aus zwei Proteinen	Quartärstruktur aus zwei Proteinen
Accession number	1KDF_A	AAA59172.1	AAA82165.1	NP_000607

Kleine Peptide werden als Oligopeptide bezeichnet. Größere Peptide mit mehr als zehn Aminosäuren werden Polypeptide genannt. Die meisten Proteine sind Ketten von 100 bis 300 Aminosäuren. Das größte bekannte Protein des Menschen heißt Titin und besteht aus einer Kette von über 30.000 Aminosäuren.

Proteine bestehen aus einem Polypeptidgerüst mit angehängten Seitenketten. Jedes Protein unterscheidet sich in seiner Aminosäuresequenz von anderen Proteinen. Unterschiedliche Aminosäuren weisen unterschiedliche Seitenketten (Reste) auf. Damit verleihen die aufeinander folgenden chemisch verschiedenen Seitenketten jedem Protein seine individuellen Eigenschaften. Die beiden Enden jeder Polypeptidkette sind chemisch verschieden. Der N-Terminus weist eine Aminogruppe auf, wohingegen der C-Terminus eine Carboxylgruppe aufweist. Die Aminosäuresequenz eines Proteins wird immer vom N- zum C-Terminus angegeben (Abb. 5).

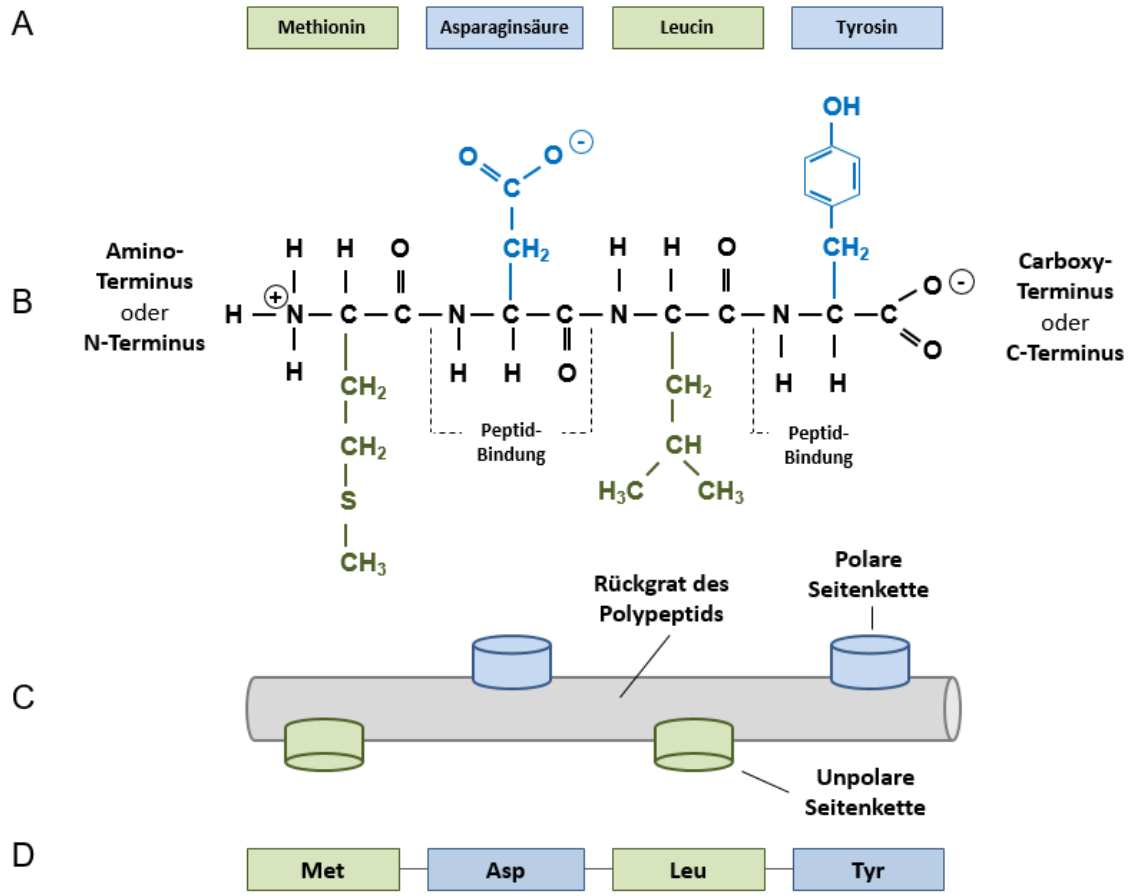


Abb. 5 Ausschnitt aus einer kurzen Aminosäuresequenz (Polypeptid). A: Diese vier Aminosäuren bilden das Polypeptid, B: Strukturformel des Polypeptids, Seitenketten der einzelnen Aminosäuren sind blau bzw. grün hervorgehoben. C: Darstellung von Rückgrat und Seitenketten des Polypeptids. D: Polypeptidsequenz als Drei-Buchstaben-Code. Übungsdatei: Aufbau_Struktur_Proteine_AB_u_LÖ.docx.

Proteine brauchen für ihre Funktion eine gewisse Größe. So können Oligopeptide als Signalstoffe – etwa als Hormon oder als Neurotransmitter – eingesetzt werden, für eine Enzymfunktion sind aber meist mehr als 50 Aminosäuren nötig.

Die räumliche Struktur bedingt die Wirkungsweise der Proteine. Die Proteinstruktur lässt sich auf vier Betrachtungsebenen beschreiben:

Als *Primärstruktur* eines Proteins wird die Abfolge (Sequenz) der einzelnen Aminosäuren einer Polypeptidkette bezeichnet. Die Primärstruktur beschreibt lediglich eine Aminosäurensequenz.

Als *Sekundärstruktur* wird die Zusammensetzung des Proteins aus besonders häufig auftretenden Strukturen für die räumliche Anordnung der Aminosäuren bezeichnet. Man unterscheidet dabei zwischen den Strukturtypen α-Helix und β-Faltblatt.

Die *Tertiärstruktur* ist die der Sekundärstruktur übergeordnete räumliche Anordnung der Polypeptidkette. Sie wird v. a. von den Kräften und Bindungen zwischen den Aminosäureresten (d. h. den Seitenketten) der Aminosäuren bestimmt.

Viele Proteine müssen sich, um funktionsfähig sein zu können, mit anderen gleichartigen oder unterschiedlichen Proteinmolekülen zu einem Proteinkomplex zusammenlagern, der sogenannten *Quartärstruktur*.

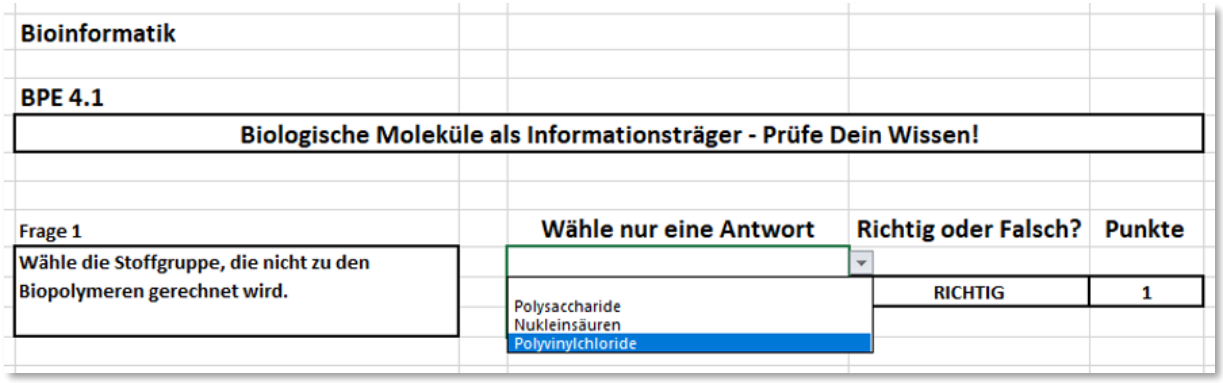


Abb. 6 Ausschnitt aus einem Quiz zur Überprüfung der Inhalte des „Informationstext - Polysaccharide, Lipide, Proteinen und Nukleinsäuren als Biopolymere“. Dateiname: QUIZ_4.1.xlsx, mit Genehmigung von Microsoft.

3.1.4.2 INFORMATIONSTEXT – DAS DOGMA DER MOLEKULARBIOLOGIE (BPE 4.2)

Der Fluss der genetischen Information in einer lebenden Zelle geht von der DNA über die RNA zum Protein. Ein Protein selbst übt immer eine bestimmte Funktion in der Zelle aus. Jedes Protein trägt damit direkt zur Merkmalsbildung bei.

Alle Zellen, vom Bakterium bis zum Menschen, exprimieren (lat. *exprimiere*, ausdrücken) ihre genetische Information auf diese Weise. Dieses Prinzip ist so fundamental, dass es 1958 von dem Mitentdecker der DNA-Doppelhelix Francis Crick als das zentrale Dogma der Molekularbiologie bezeichnet wurde. Heute weiß man, dass es in bestimmten RNA-Viren Ausnahmen zu dieser Regel gibt (Stichwort: Reverse Transkription).

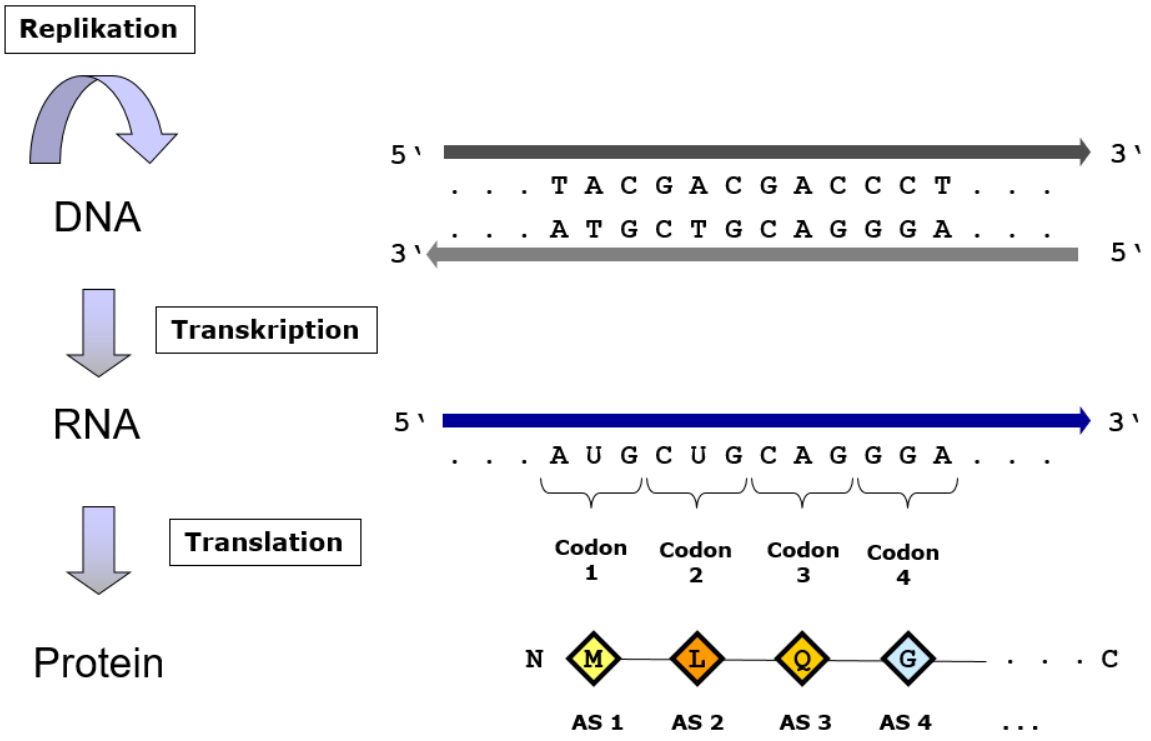


Abb. 7 Das Dogma der Molekularbiologie: „DNA makes RNA makes Protein“.

Als genetischer Code wird die Weise bezeichnet, mit der die Nukleotidsequenz eines RNA-Einzelstrangs in die Aminosäurensequenz der Polypeptidkette eines Proteins übersetzt wird. In der Zelle geschieht dies, nachdem zuvor die in der Abfolge von Basenpaaren des DNA-Doppelstrangs niedergelegte Erbinformation in die Sequenz des mRNA-Einzelstrangs umgeschrieben wurde (Transkription, Abb. 8).

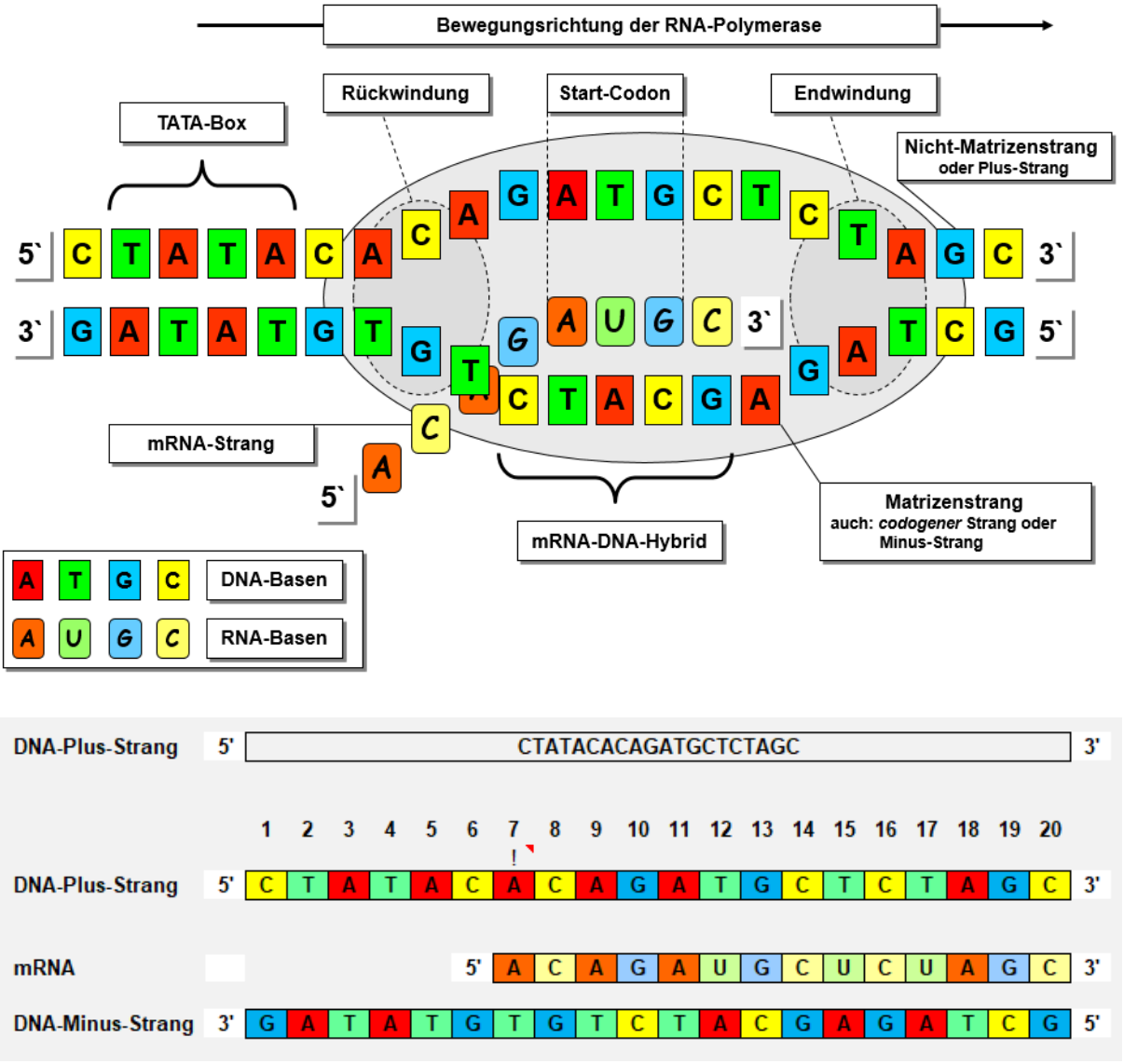


Abb. 8: Schematische Darstellung der Transkription (oben) und Nachvollziehen des Konzepts mithilfe eines Excel-Rechenblatts (unten). Dateinamen: Transkription_AB.xlsx und Transkription_LÖ.xlsx.

Dieser genetische Code ist bei allen bekannten Arten von Lebewesen in den Grundzügen gleich. Er ordnet einem Triplett von drei aufeinanderfolgenden Nukleobasen der Nukleinsäuren – dem sogenannten Codon – jeweils eine bestimmte proteinogene Aminosäure zu. Die Übersetzung, Translation genannt (Abb. 7), findet an den Ribosomen im Zytosol einer Zelle statt. Sie bilden nach Vorgabe der Sequenz von Nukleotiden einer mRNA die Sequenz von Aminosäuren eines Peptids, indem jedem Codon über das Anticodon einer Transfer-Ribonukleinsäure (tRNA) eine bestimmte Aminosäure zugewiesen und diese mit der vorherigen verbunden wird. Auf diese Weise wird eine bestimmte vorgegebene Information in die Form einer Peptidkette überführt, die sich dann zur besonderen Form eines Proteins faltet.

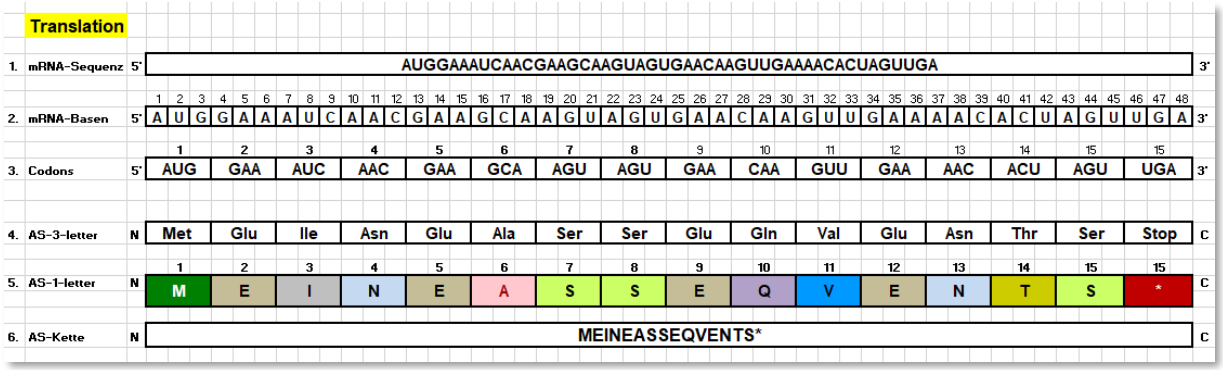


Abb. 9 Durchführung einer Translation mithilfe eines Excel-Rechenblatts. Auf Basis des genetischen Codes werden den Codons einer mRNA-Sequenz die passenden Aminosäuren zugeordnet. Dateien: Code_und_Dogma _AB.xlsx, Code_und_Dogma_LÖ.xlsx, mit Genehmigung von Microsoft.

Als Codon bezeichnet man das Variationsmuster einer Abfolge von drei Nukleobasen der mRNA, eines Basentripletts, das für eine Aminosäure codieren kann. Insgesamt existieren 43 = 64 mögliche Codons, davon codieren 61 für die insgesamt 20 proteinogenen Aminosäuren; die restlichen drei sind sogenannte Stoppcodons zur Termination der Translation.

Der Vorgang, mit dem die Information eines Gens in der Form eines Proteins ausgedrückt wird (Genexpression), ergibt sich aus einer Folge definierter Schritte: (1) Transkription – ein Genabschnitt der DNA des Genoms wird durch RNA-Polymerase in mRNA umgeschrieben, (2) posttranskriptionale Modifikation der präRNA, Reifung zur mRNA, (3) Transport der mRNA aus dem Zellkern in das Cytoplasma, (4) Translation der mRNA am Ribosom, Übersetzung der Information in ein Polypeptid/Protein. Die Translation ist also der Schritt, bei dem die genetische Information der Basentriplett-Abfolge in eine Aminosäure-Abfolge umgesetzt wird.

Alle Lebewesen benutzen in Grundzügen denselben genetischen Code. Bei der Transkription eines DNA-Abschnitts dient einer RNA-Polymerase der codogene Strang (auch: Minus-Strang) als Matrize für die Herstellung des Transkripts: die DNA-Basensequenz wird basenpaarend in die komplementäre RNA-Basensequenz umgeschrieben (Abb. 8). Die entstehende Sequenz des mRNA-Strangs entspricht der DNA-Sequenz des Plus-Strangs. So wird durch die Transkription auf die in der DNA abgelegte genetische Information zugegriffen, die dann in Form der mRNA für die Proteinbiosynthese zur Verfügung steht. Die Initiation der Transkription setzt mit der Bildung eines Komplexes aus unterschiedlichen Proteinen in der Promotorregion ein. Die Synthese des Transkripts (Elongation) beginnt in der Nähe des Promotorbereichs an Position +1, mit dem ersten Nukleotid der mRNA. Sie endet hinter dem Stopp-Codon, nachdem der 3'UTR fertiggestellt wurde. Entscheidend für die Beendigung der Transkription (Termination) ist die Terminator-Region im 3'-Bereich des Gens.

Im Jahr 1977 wurde entdeckt, dass die Gene von Eukaryoten oft nicht aus einer einzigen durchgehend kodierenden Nukleotidsequenz bestehen. Die genetische Information eines solchen Mosaikgens ist von DNA-Sequenzen unterbrochen, die für die Kodierung des Genprodukts nicht erforderlich sind. Derartige nichtkodierende Segmente der DNA werden Introns genannt, kodierende dagegen Exons. Sowohl Introns als auch Exons eines Mosaikgens werden in eine vorläufige prämRNA transkribiert. Sie verlässt

aber nicht den Zellkern, sondern wird noch im Zellkern im Verlauf der post-transkriptionellen Modifikation spezifisch enzymatisch verändert: Die Introns werden aus der prä-mRNA unter Ausbildung von „Lasso-Strukturen“ herausgeschnitten. Die jeweils zu beiden Seiten gelegenen Exons werden zu einer zusammenhängenden mRNA verknüpft. Dieser Vorgang wird Spleißen (engl. splicing, Zusammenschneiden) genannt. Am 5′-Ende wird eine cap-Struktur (engl. cap, Mütze, Kappe) aus einem methylierten Guanosin-Triphosphat angeheftet. Sie erleichtert die Anlagerung der mRNA an das Ribosom und schützt das 5′-Ende vor enzymatischem Abbau. An das 3′-Ende wird eine Sequenz von bis zu 250 Adenin-Nucleotiden angefügt. Dieser Poly(A)-Schwanz erleichtert den Export der mRNA ins Cytoplasma und schützt das 3′-Ende vor enzymatischem Abbau. Die Veränderungen im Zellkern, die von der prä-mRNA zur reifen mRNA führen, werden als mRNA-Reifung oder mRNA-Prozessierung zusammengefasst.

Die reife mRNA enthält noch nichtkodierende Bereiche: Die 5′-cap-Struktur und die sich anschließende Leader-Sequenz (auch 5′-UTR für 5‘ untranslated region), der 3′-UTR-Abschnitt, der dem Stopp-Codon folgt und der Poly(A)-Schwanz werden nicht in Aminosäuren übersetzt.

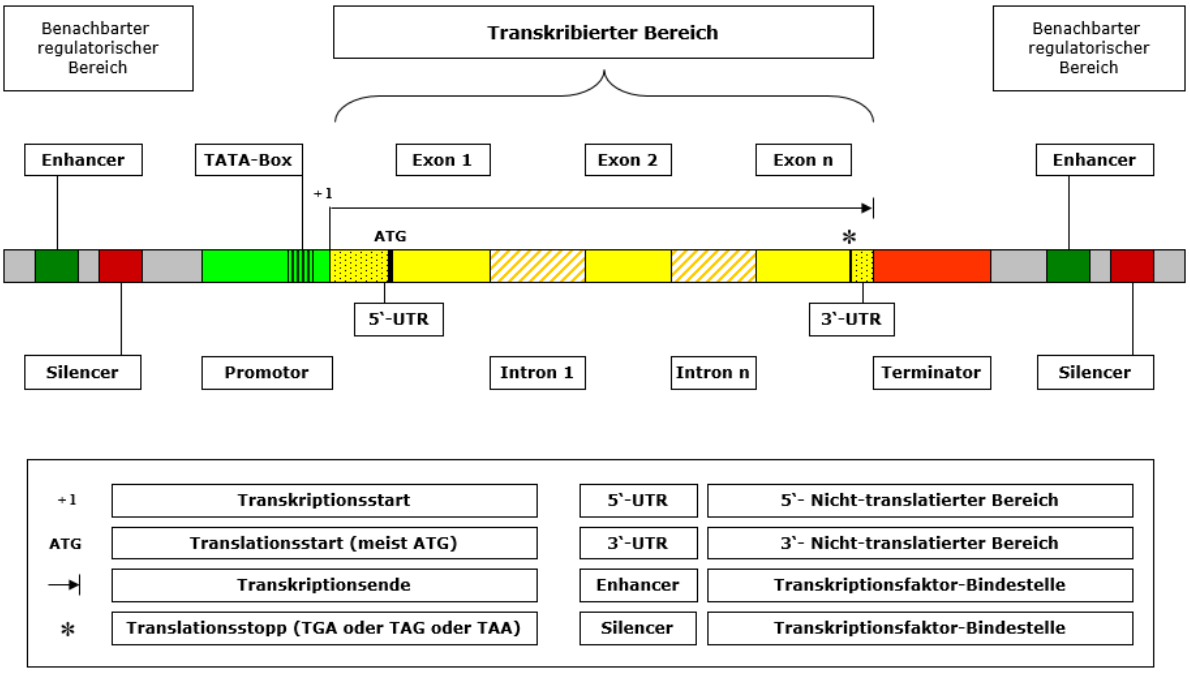


Abb. 10 Die Strukturelemente eines typischen eukaryotischen Gens. Dargestellt ist der Plus-Strang. Enhancer und Silencer sind bestimmte Regionen, die die Transkription unter Vermittlung von spezifischern Proteinfaktoren positiv bzw. negativ beeinflussen können.

Ein Operon ist eine Funktionseinheit der DNA von Prokaryoten. Die funktionelle Einheit eines Operons besteht aus Promotor, Operator und mehreren Strukturgenen, die für Proteine codieren, die zwar unterschiedliche Funktionen übernehmen, aber im Dienste eines übergeordneten Ziels stehen, z. B. dienen die Strukturgene des lac-Operons dem Transport und Abbau von Lactose in Bakterien. Je nach Operon

können verschiedene regulatorische Proteine als Repressoren oder als Aktivatoren mit den Operatoren in Wechselwirkung treten und dadurch die Transkription der Gene im Operon an- oder abschalten. Dadurch kann die Synthese der betreffenden mRNA (messenger-RNA) aktiviert oder gehemmt werden, indirekt also auch die Translation der codierten Proteine.

Alle Gene eines Operons werden koordiniert gesteuert. Das Operon-Modell der Genregulation wurde anhand des lac-Operons des Darmbakteriums Escherichia coli entwickelt und später erweitert.

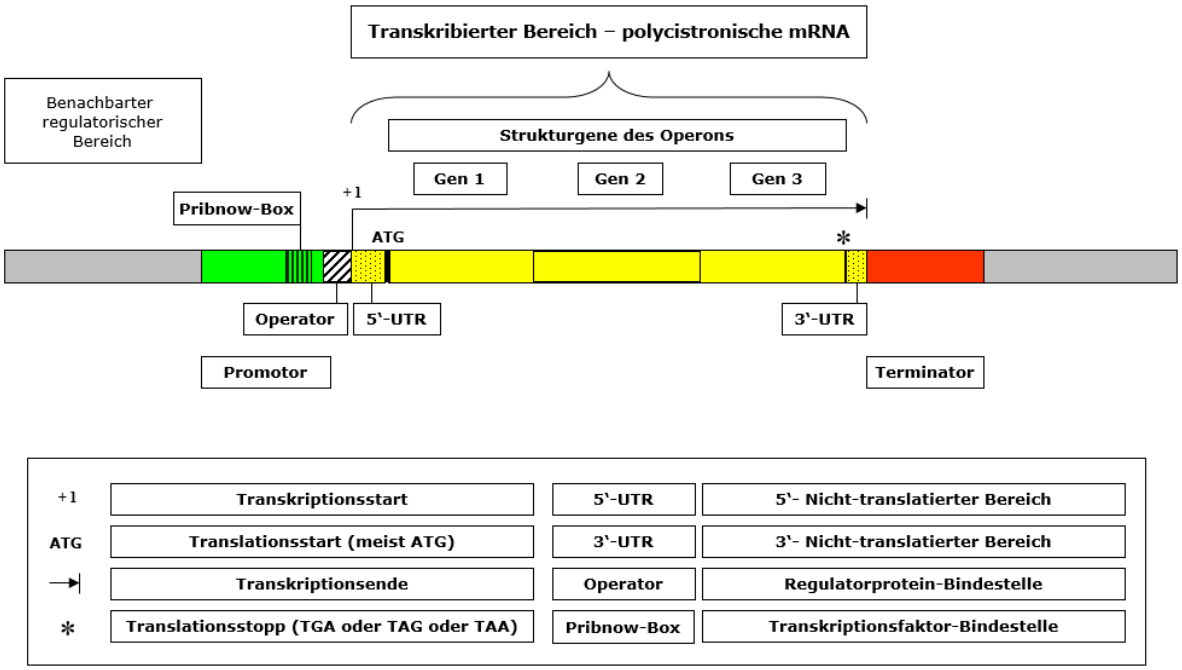


Abb. 11 Die Strukturelemente eines typischen prokaryotischen Operons. Dargestellt ist der Plus-Strang.

Als Konsensussequenz (auch: Consensusmotiv) wird diejenige Sequenz von Nukleotiden oder Aminosäuren bezeichnet, welche in der Summe am wenigsten von einer gegebenen Menge von entsprechenden Mustersequenzen abweicht.

Meist liegt der Erstellung einer Konsensussequenz die Annahme zugrunde, dass die gegebenen Sequenzen einen gemeinsamen evolutionären Ursprung haben oder ein Sequenzmotiv mit einer bestimmten biologischen Aufgabe repräsentieren (z. B. Ribosomenbindestelle bei mRNAs, Transkriptionsfaktorbindestelle, Startpunkt der Polyadenylierung von mRNAs im 3'UTR u. v. a.), wobei es oft auch sinnvoll sein kann, mehrdeutige Konsensussequenzen zu formulieren.

Bei Nukleinsäuren können hierfür die Basensymbole der Nukleinsäure-Nomenklatur verwendet werden, also neben den eindeutigen Basensymbolen A, C, G, T, U auch beispielsweise R für eine beliebige Purinbase, Y für eine beliebige Pyrimidinbase oder N für ein beliebiges Nukleotid schlechthin.

In der Regel werden Konsensussequenzen aus einem multiplen Sequenzalignment (BPE 8) erstellt. Im einfachsten Fall wird dasjenige Element in die Konsensussequenz aufgenommen, welches in der entsprechenden Spalte des MSA am häufigsten vorkommt.

Das multiple Sequenzalignment (MSA), also das gleichzeitige Analysieren mehrerer Sequenzen, liefert im Vergleich zum paarweisen Sequenzvergleich genauere Informationen über Aminosäureverteilungen an einzelnen Positionen ähnlicher Gene oder Proteine. Solche Verteilungen können nicht nur Aufschluss über konservierte Bereiche geben, sondern sie sind auch die Grundlage für profilbasierte Datenbanksuchen und phylogenetische Analysen (BPE 10).

Darstellung des multiplen Alignments

Um die konservierten Positionen in dem multiplen Alignment hervorzuheben, wird oft eine sogenannte Konsensus-Sequenz unter das Alignment geschrieben. Sie besteht aus den Zeichen, die innerhalb des Alignments am häufigsten vertreten sind. In der unten gezeigten Abbildung sind drei mögliche Darstellungsformen der Konsensussequenz gezeigt.

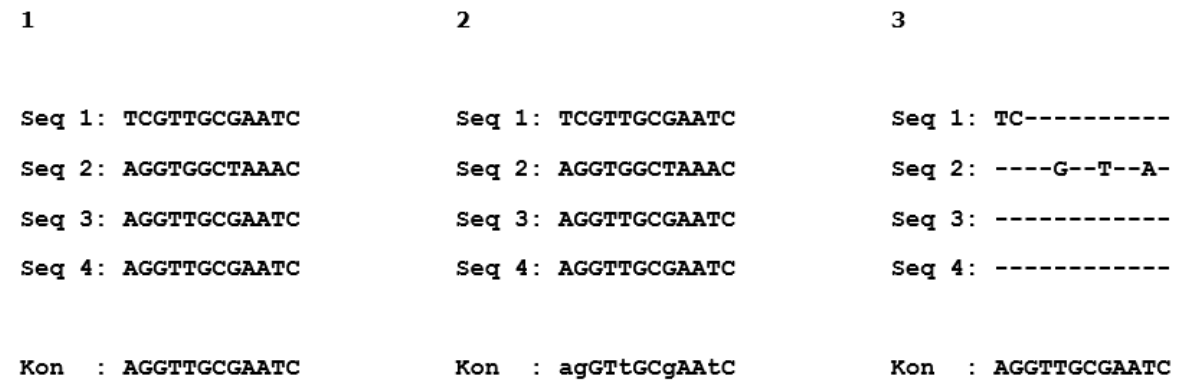


Abbildung 12: Drei Möglichkeiten der Darstellung einer Konsensussequenz (Kon).

Die einfachste Art einer Konsensussequenz zeigt Abbildung 12.1. Es werden die Zeichen in die Zeile der Konsensussequenz geschrieben, die am häufigsten an den Positionen in den untersuchten Sequenzen auftreten. In Abbildung 12.2 wird zwischen den Zeichen unterschieden, die in allen Sequenzen gleich sind (Großbuchstaben), und denen, die in der Mehrzahl auftreten (Kleinbuchstaben). Die dritte Form (Abbildung 12.3) hebt in den Sequenzen nur die Positionen hervor, die sich von der Konsensussequenz unterscheiden.

Konsensus-Beispiel 1: Innerhalb des 5'-UTR der mRNA von Prokaryoten befindet sich die nach Ihren Entdeckern benannte Shine-Dalgarno-Sequenz, die die Ribosomenbindestelle (RBS, auch: Ribosomenerkennungsstelle) darstellt.

Die Shine-Dalgarno-Sequenz besteht bei vielen prokaryotischen mRNAs aus der Konsensus-Sequenz AGGAGGU. Dieser Konsensus befindet sich meistens 4 bis 7 Nukleotide stromaufwärts (in Richtung 5'-Ende) vom Initiations-Codons AUG aus gesehen (Abb. 13).

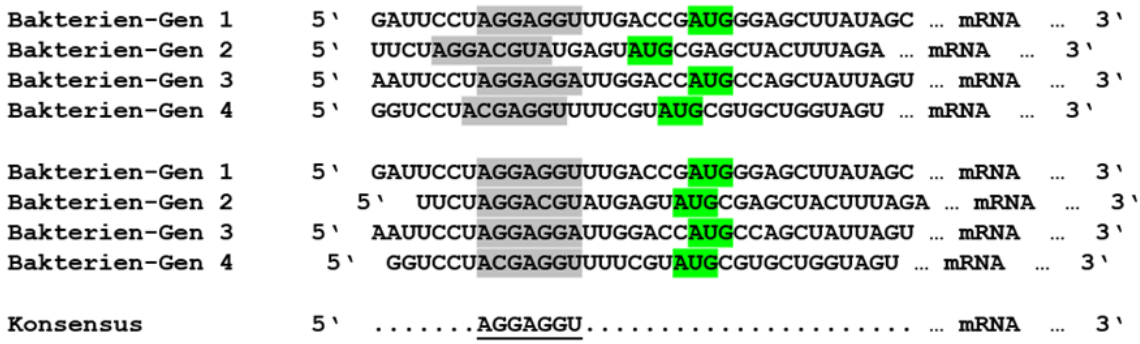
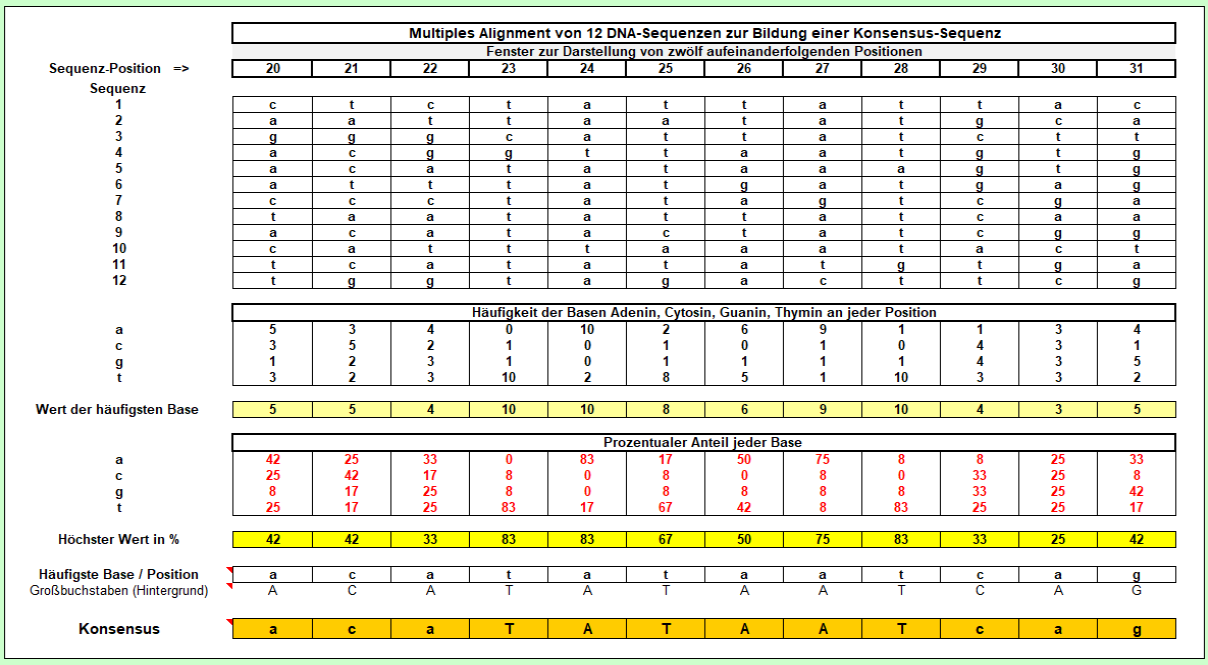


Abb. 13 Screenshot aus der Übungsdatei zur Ermittlung der Shine-Dalgarno-Konsensussequenz. Dateiname: S_D_Konsensus_AB_u_LÖ.docx.

Konsensus-Beispiel 2: Gene enthalten Regionen, die man als Promotorstellen bezeichnet. Diese binden spezifisch die RNA-Polymerase und bestimmen, wo die Transkription beginnt. In Bakterien sind zwei Sequenzen auf der 5′-Seite des ersten transkribierten Nucleotids (stromaufwärts, upstream) von Bedeutung (siehe Abb. 14). Die sogenannte PribnowBox besitzt die Konsensussequenz TATAAT und liegt bei -10 (das heißt, zehn Nucleotide von der 5′-Seite des ersten transkribierten Nucleotids entfernt, das man mit +1 bezeichnet). Die zweite Sequenz wird als -35-Region bezeichnet und besitzt die Konsensus-Sequenz TTGACA. Das erste transkribierte Nucleotid ist gewöhnlich ein Purin. Diese funktionalen Abschnitte eines Genpromotors weisen bestimmte Basenabfolgen auf, die es bestimmten Proteinen ermöglichen die Struktur dieser Basenzeichenfolgen zu erkennen und an diese zu binden.

Position	.	1	.	.	.	.	6	.	.	.	.	.	12	.
Seq_1	.	c	t	c	t	a	t	t	a	t	t	a	c	.
Seq_2	.	a	a	t	t	a	a	t	a	t	g	c	a	.
Seq_3	.	g	g	g	c	a	t	t	a	t	c	t	t	.
Seq_4	.	a	c	g	g	t	t	a	a	t	g	t	g	.
Seq_5	.	a	c	a	t	a	t	a	a	a	g	t	g	.
Seq_6	.	a	t	t	t	a	t	g	a	t	g	a	g	.
Seq_7	.	c	c	c	t	a	t	a	g	t	c	g	a	.
Seq_8	.	t	a	a	t	a	t	t	a	t	c	a	a	.
Seq_9	.	a	c	a	t	a	c	t	a	t	c	g	g	.
Seq_10	.	c	a	t	t	t	a	a	a	t	a	c	t	.
Seq_11	.	t	c	a	t	a	t	a	t	g	t	g	a	.
Seq_12	.	t	g	g	t	a	g	a	c	t	t	c	g	.
Anzahl a	.	5	3	4	0	10	2	6	9	1	1	4	4	.
Anzahl c	.	3	5	2	1	0	1	0	1	0	5	2	1	.
Anzahl g	.	1	2	3	1	0	1	1	1	1	3	3	5	.
Anzahl t	.	3	2	3	10	2	8	5	1	10	3	3	2	.
KONSENSUS	.	a	c	a	T	A	T	A	A	T	c	a	g	.

Abb. 14 Screenshot aus der WORD-Übungsdatei zur Ermittlung der Pribnow-Konsensussequenz TATAAT. Dateiname: Pribnow_Konsensus_von_Hand_AB_u_LÖ.docx, mit Genehmigung von Microsoft.



Multiples Alignment von 12 DNA-Sequenzen zur Bildung einer Konsensus-Sequenz												
Fenster zur Darstellung von zwölf aufeinanderfolgenden Positionen												
Sequenz-Position =>	20	21	22	23	24	25	26	27	28	29	30	31
Sequenz												
1	c	t	c	t	a	t	t	a	t	t	a	c
2	a	a	t	t	a	a	t	a	t	g	c	a
3	g	g	g	c	a	t	t	a	t	c	t	t
4	a	c	g	g	t	t	a	a	t	g	t	g
5	a	c	a	t	a	t	a	a	a	g	t	g
6	a	t	t	t	a	t	g	a	t	g	a	g
7	c	c	c	t	a	t	a	g	t	c	g	a
8	t	a	a	t	a	t	t	a	t	c	a	a
9	a	c	a	t	a	c	t	a	t	c	g	g
10	c	a	t	t	t	a	a	a	t	a	c	t
11	t	c	a	t	a	t	a	t	g	t	g	a
12	t	g	g	t	a	g	a	c	t	t	c	g
Häufigkeit der Basen Adenin, Cytosin, Guanin, Thymin an jeder Position												
a	5	3	4	0	10	2	6	9	1	1	3	4
c	3	5	2	1	0	1	0	1	0	4	3	1
g	1	2	3	1	0	1	1	1	1	4	3	5
t	3	2	3	10	2	8	5	1	10	3	3	2
Wert der häufigsten Base	5	5	4	10	10	8	6	9	10	4	3	5
Prozentualer Anteil jeder Base												
a	42	25	33	0	83	17	50	75	8	8	25	33
c	25	42	17	8	0	8	0	8	0	33	25	8
g	8	17	25	8	0	8	8	8	8	33	25	42
t	25	17	25	83	17	67	42	8	83	25	25	17
Höchster Wert in %	42	42	33	83	83	67	50	75	83	33	25	42
Häufigste Base / Position	a	c	a	t	a	t	a	a	t	c	a	g
Großbuchstaben (Hintergrund)	A	C	A	T	A	T	A	A	T	C	A	G
Konsensus	a	c	a	T	A	T	A	A	T	c	a	g

Abb. 15 Screenshot aus der Excel-Übungsdatei zur Ermittlung der Pribnow-Konsensussequenz. Dateiname: Pribnow_Konsensus_AB.xslx und Pribnow_Konsensus_LÖ.xlsx, mit Genehmigung von Microsoft.

Konsensus-Beispiel 3: Alternativ zum Beispiel 2 kann durch Verwendung von 12 Sequenzen aus dem 3‘-UTR-Bereich unterschiedlicher menschlicher Gene der AATAAA-Konsensus erstellt werden. Als Transkriptions-Terminations-Signal fungiert bei Eukaryonten die Sequenz AATAAA im 3’-Bereich der transkribierten Gensequenz, wenn danach ein G- und T-reicher Sequenzabschnitt (der Terminator) folgt. Das resultierende AAUAAA-Sequenzmotiv (Abb. 16) der mRNA dient als Polyadenylierungssignal bei der posttranskriptionellen Modifikation der prä-mRNA.

Alle eukaryontischen mRNAs tragen einen sogenannten Poly(A)-Schwanz an ihrem 3’-Ende. Dieser „Schwanz“ besteht aus 150 bis 250 Adenosin-Einheiten. Der Poly(A)-Schwanz wird direkt nach der Transkription kurz hinter dem Signal AAUAAA angeheftet. Er erleichtert den Transport aus dem Zellkern in das Zytoplasma und schützt das 3’-Ende der mRNA vor dem enzymatischen Abbau. Ein Polyadenylierungssignal ist bei allen eukaryontischen mRNAs vorhanden. Es besteht in den meisten

Fällen aus der Sequenz AAUAAA – die aber auch leicht abgewandelt sein kann. Erstellt man ein Alignment der 3’UTRs verschiedener eukaryontischer mRNAs so erhält man die Konsensussequenz A-AUAAA. Basis ist dieselbe Excel-Rechenblatt-Struktur wie für das Konsensus-Beispiel 2. Dateinamen: AATAAA_Konsensus_AB.xlsx und AATAAA_Konsensus_LÖ.xlsx.

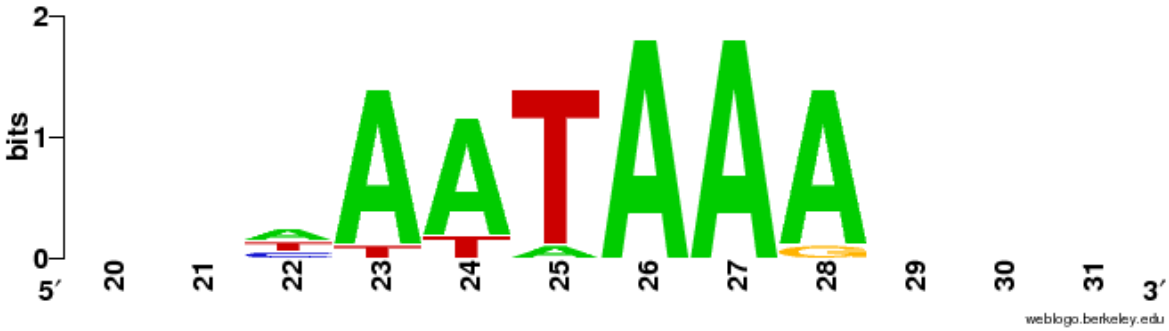


Abb. 16 Screenshot der AATAAA-Konsensussequenz, die mithilfe des Online-Programms WEBLOGO (https://weblogo.berkeley.edu/logo.cgi) erstellt wurde. Dateinamen: AATAAA_Konsensus_AB.xlsx und AATAAA_Konsensus_LÖ.xlsx

3.2 Stammbaumrekonstruktion (BPE 10.2 bis 10.5)

3.2.1 VERLAUFSPLAN/STOFFVERTEILUNG

DAUER	UNTERRICHTSPHASE, INHALT	MATERIAL, MEDIEN	ANGESTREBTES ERGEBNIS, ERWARTETES SCHÜLERVERHALTEN
45	BPE 10.2 Einstieg ins Thema Zusammensetzung und Bedeutung von Stammbäumen, Gruppen innerhalb von Stammbäumen	Arbeitsblatt: „Grundlagen der Phylogenie“	Die SuS sind mit den wesentlichen Begriffen der molekularen Phylogenie vertraut und bereiten sich damit auf die konkrete Stammbaumanalyse vor.
45	BPE 10.2 Anfertigen eines Glossars mit den wesentlichen Fachbegriffen zur Phylogenie entsprechend den Informationen aus dem Arbeitsblatt (s. o.).	Arbeitsblatt: „Grundlagen der Phylogenie“, PC	Ergebnissicherung zum vorliegenden Arbeitsblatt.
45	BPE 10.3 Bearbeiten von Aufgaben zum Thema „Molekulare Uhr“.	Textarbeit: „Molekulare Uhr“	Die SuS kennen das Prinzip der molekularen Uhr und seine Bedeutung für die Erstellung von molekularen Stammbäumen.
45	BPE 10.3 Erarbeiten von Methoden der Stammbaumrekonstruktion.	Arbeitsblatt: „Rekonstruktionsmöglichkeiten von Stammbäumen“	Die SuS erarbeiten ausgewählte Möglichkeiten der Stammbaumrekonstruktion.
90	BPE 10.4	Arbeitsblatt Mammut	Anwendung von 2 Baumberechnungsverfahren.

	Anwendung früherer Kenntnisse zur Arbeit mit NCBI, Aufbereiten von Datenmaterial. Anwenden der Kenntnisse über Neighbour-Joining und Maximum Parsimony im PC-Programm SeaView.	Einzelarbeit am PC.	Prüfen von einer Hypothese zur Verwandtschaft von Mammut und Elefant.
90	BPE 10.5 Bootstrapping erstellt durch wiederholtes Ziehen mit Zurücklegen von Spalten aus dem Alignment und anschließender Baumrekonstruktion eine Statistik, die Aussagen darüber macht, wie wahrscheinlich jede Verzweigung im Baum ist. Anwendung der Kenntnisse zur Arbeit mit NCBI, Aufbereiten von Datenmaterial, Anwenden der Kenntnisse über Neighbour-Joining im PC-Programm SeaView. Anwendung des Bootstrapping am PC.	Informationstext zum Bootstrapping. Arbeitsblatt zur Anwendung des Bootstrapping.	Bootstrapping als statistische Absicherung kennenlernen. Anwenden des Bootstrapping am Beispiel. Überprüfen einer Hypothese zur nächsten Verwandtschaft beim Elefanten in Gruppenarbeit.

Hinweis: Trennen Sie die Stunden bzw. Doppelstunden durch eine leere Zeile voneinander ab.

3.2.2 FACHLICHE HINWEISE

Molekulare Daten in Form von biologischen Sequenzen können zur Untersuchung der Phylogenie von Organismengruppen herangezogen werden. Verschiedene Verfahren zur Rekonstruktion von Stammbäumen führen zu Modellen der Phylogenie und liefern damit Erklärungsmöglichkeiten zur Evolution von Organismengruppen.

3.2.3 DIDAKTISCHE HINWEISE

Die Schülerinnen und Schüler definieren den Begriff „Phylogenetik“ und benennen Strukturelemente von Stammbäumen. Verschiedene Stammbaumtypen werden gegeneinander abgegrenzt.

Das Konzept einer molekularen Uhr und deren Anwendungsmöglichkeit werden dargestellt. Die prinzipiellen Grundlagen von verschiedenen Verfahren zur Rekonstruktion von Stammbäumen werden erarbeitet.

Molekulare Stammbäume werden mithilfe von geeigneten Programmen auf der Basis eines Alignments modelliert. Die Schülerinnen und Schüler formulieren das grundsätzliche Prinzip des Bootstrapping. Sie diskutieren und entwickeln Hypothesen über die Validität der einzelnen Ergebnisse.

3.2.4 ARBEITSMATERIALIEN UND AUFGABEN ZU DEN BPE 10.2 BIS 10.5

3.2.4.1 ARBEITSBLATT: GRUNDLAGEN DER PHYLOGENETIK (BPE 10.2)

WAS BEDEUTET „PHYLOGENIE“?

Unter Phylogenie versteht man die Abstammungsgeschichte einer Art. Die phylogenetische Forschung verfolgt dabei das Ziel, die Verwandtschaftsbeziehungen zwischen biologischen Arten, Populationen, Individuen oder einzelnen Genen aufzuklären. Unter Verwandtschaft versteht man hierbei die Abstammung einer gemeinsamen Stammart.

WAS VERSTEHT MAN UNTER EINEM STAMMBAUM?

Die Ergebnisse phylogenetischer Forschungen werden in Stammbäumen dargestellt. Ein Stammbaum, der alle Nachkommen einer Art aufführt, besitzt einen Ursprung (Wurzel) und wird daher als rooted tree bezeichnet. Solche Bäume sind gut dazu geeignet, Abstammungen darzustellen. Es ist ebenso möglich einen Stammbaum ohne Wurzel (unrooted tree) zu erstellen. In einem solchen unrooted tree lassen sich Verwandtschaftsbeziehungen ohne Angabe eines gemeinsamen Vorfahrens darstellen. In Abbildung 1 sind die beiden Stammbaumarten dargestellt.

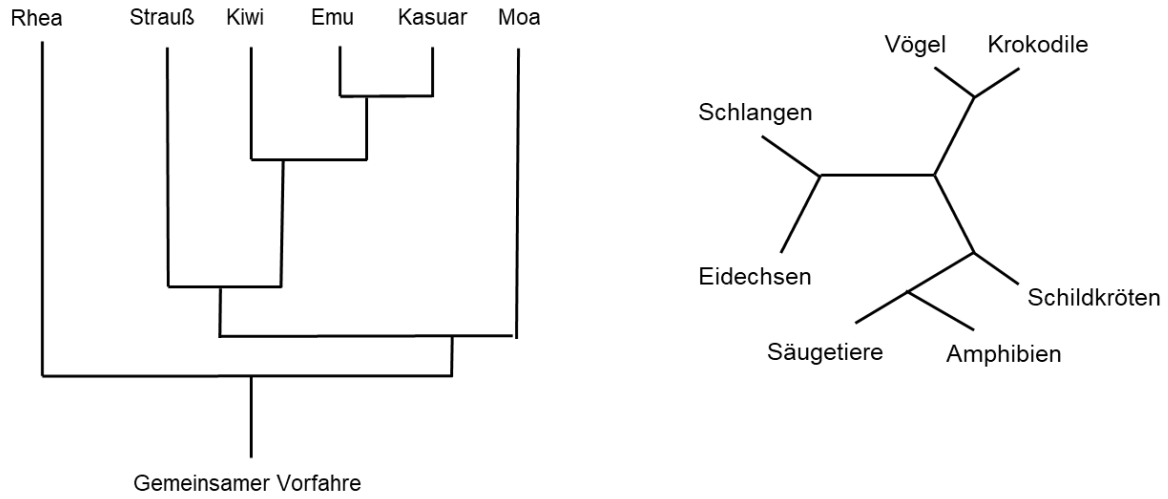


Abb. 1: Möglichkeiten der Stammbaumdarstellung (links: rooted tree am Beispiel flugunfähiger Vögel, den Ratiten; rechts: unrooted tree am Beispiel der Wirbeltiergruppen).

In den meisten Fällen wird der rooted tree bevorzugt, da er mehr Informationen enthält. Die Taxa, die man in einem Stammbaum unterscheiden möchte, werden als Innengruppe bezeichnet. Demgegenüber gibt man eine Außengruppe (outgroup) an, die nicht näher mit den zu untersuchenden Taxa verwandt sein soll. Die Darstellung der Außengruppe hilft dabei, abgeleitete Aspekte von ursprünglichen Merkmalen zu unterscheiden, und ist eine gute Bewertungsgrundlage erstellter Stammbäume.

In einem Stammbaum bezeichnet man Verzweigungspunkte zwischen den abgebildeten Taxa als „Knoten“. Ein Knoten stellt oftmals eine ausgestorbene Stammart dar. Der „Ast“ bildet die Verzweigungen zwischen den Taxa. Man unterscheidet interne Äste, welche zwei Knoten miteinander verknüpfen. Die Endpunkte der Verzweigung (also die eigentlichen Taxa oder auch „terminalen Äste“) werden als „Blätter“ bezeichnet.

GRUNDLAGE FÜR DIE ERSTELLUNG VON STAMMBÄUMEN

Im Prinzip sortiert man Arten in einem Stammbaum nach ihrer Ähnlichkeit. Man geht davon aus, dass Ähnlichkeit ihre Ursache in gemeinsamer Abstammung besitzt. Diese Ähnlichkeit kann sich auf verschiedene Beobachtungen stützen. Bei den „klassischen“ Möglichkeiten bezieht man Befunde aus Fossilfunden (z. B. morphologische Homologien) oder Erkenntnisse aus der Embryologie der Lebewesen mit ein.

Eine moderne Möglichkeit, Ähnlichkeiten in einem evolutionsbiologischen Kontext festzustellen, besteht in der Analyse von DNA-Sequenzen. Der Vorteil bei der Erstellung phylogenetischer Stammbäume mithilfe von DNA-Sequenzen liegt darin, dass die Daten digital zur Verfügung stehen und sie sich somit einfach bearbeiten lassen. Gene haben die Fähigkeit, sich unterschiedlich schnell zu verändern. Dies ermöglicht den Vergleich eng miteinander verwandter Arten oder die Unterscheidung großer Gruppen der Lebewesen. Verwendet man Gene, die sich schnell verändern, wie zum Beispiel die Gene des mitochondrialen Genoms der Säugetiere, so kann man eng miteinander verwandte Arten miteinander vergleichen. Untersucht man hingegen Gene, die sich nicht sehr schnell verändern, wie z. B. Gene für ribosomale RNA, kann man einen Stammbaum erstellen, welcher die Beziehungen zwischen den Großgruppen Archaea, Bacteria und Eukarya veranschaulicht.

WIE ERSTELLT MAN PHYLOGENETISCHE STAMMBÄUME?

Der deutsche Insektenforscher Willi Hennig (1913–1976) legte die Grundlagen zur Erstellung von Stammbäumen, die im Prinzip auch heute noch gültig sind. Die von ihm begründete „Kladistik“ betrachtet Arten als verwandt, wenn sie von einem einzigen gemeinsamen Vorfahren abstammen. Spaltet sich eine Art (die Stammart) in zwei neue auf, so bezeichnet man die beiden neu gebildeten Arten als Schwesterarten. Auf diese Art und Weise entstehen auch die gegabelten (dichotomen) Stammbäume.

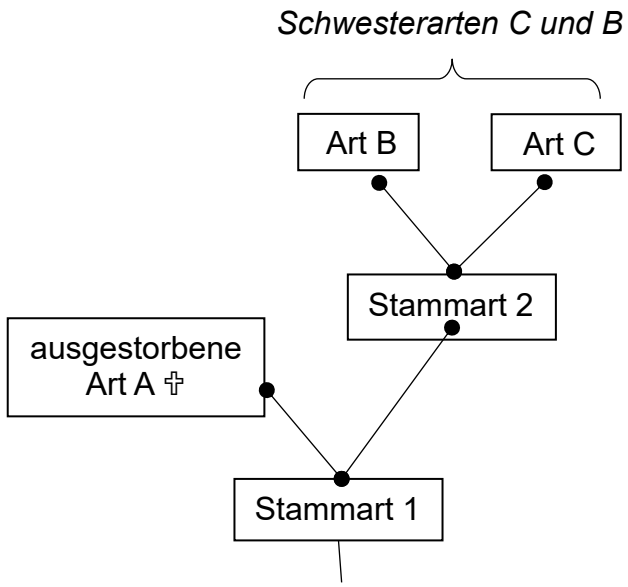


Abb. 2: Ein dichotomer Stammbaum mit Erläuterungen.

In gewisser Weise ist es das Ziel der Kladistik, sogenannte monophyletische Gruppen ausfindig zu machen. Darunter versteht man die Ermittlung von Arten, die aus einer gemeinsamen Stammart entstanden sind, die also eindeutige Abstammungsgemeinschaften abbilden.

Innerhalb von Stammbäumen kann man dabei zwischen mono-, para- und polyphyletischen Gruppen/Taxa unterscheiden. Diese Begriffe sind im Folgenden an den Stammbäumen der „Eulenartigen“ erläutert:

1. MONOPHYLETISCHE GRUPPEN

In einer monophyletischen Gruppe (Klade), findet man alle und vor allem nur die Abkömmlinge eines gemeinsamen Vorfahren (Stammart). Für die Entscheidung, ob eine solche monophyletische Gruppe vorliegt, werden nur die gemeinsamen Merkmale betrachtet. Im Falle der Uhus ist das gemeinsame Merkmal die Ohrbüschel, die bei „gewöhnlichen“ Eulen nicht vorkommen. Da der Uhu und der Virginia-Uhu einen gemeinsamen Vorfahren besitzen, bezeichnet man die beiden Arten als monophyletisch.

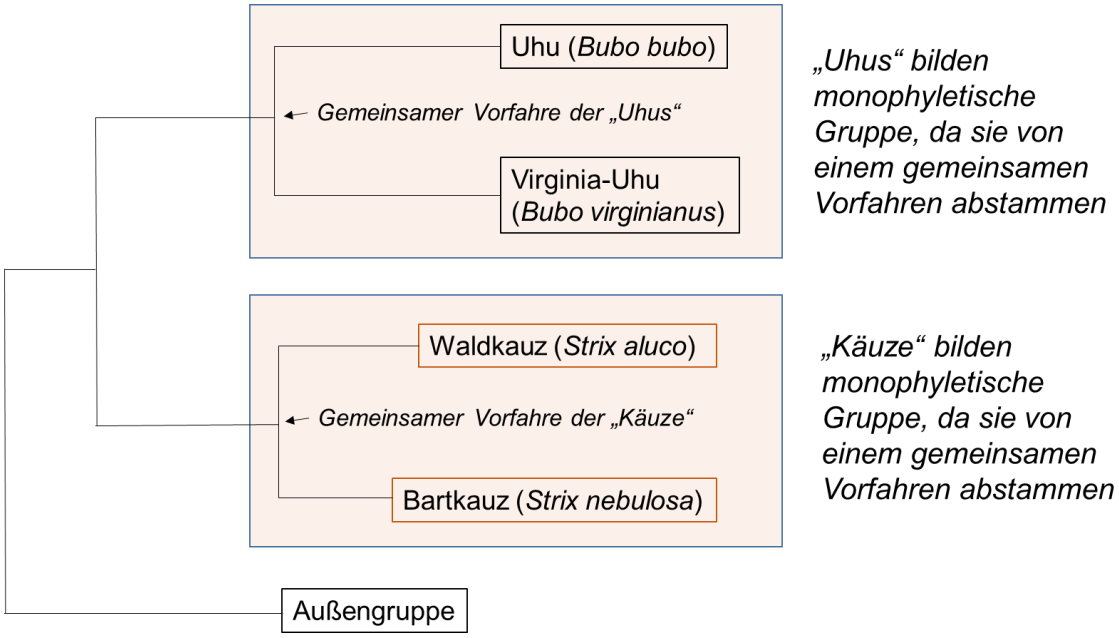


Abb. 3: Monophyletische Gruppen.

2. PARAPHYLETISCHE GRUPPEN

Angehörige paraphyletischer Gruppen haben zwar eine gemeinsame Stammform, die sie aber auch mit Arten einer anderen Gruppierung teilen. Viele klassische Taxa sind nach neueren Erkenntnissen tatsächlich paraphyletisch. Ein typisches Beispiel hierfür sind die Reptilien, zu denen die Eidechsen, Krokodile, Schildkröten und Schlangen gehören. Allerdings ist es heute unumstritten, dass aus dieser Gruppe auch die Vögel hervorgegangen sind. Die Reptilien bilden somit eine paraphyletische Gruppe, die aber auch Merkmale mit Arten anderer Gruppen (hier z. B. den Vögeln) teilen. Paraphyletische Gruppen bilden somit zwar eine Abstammungsgemeinschaft, aber keine geschlossene Abstammungsgemeinschaft.

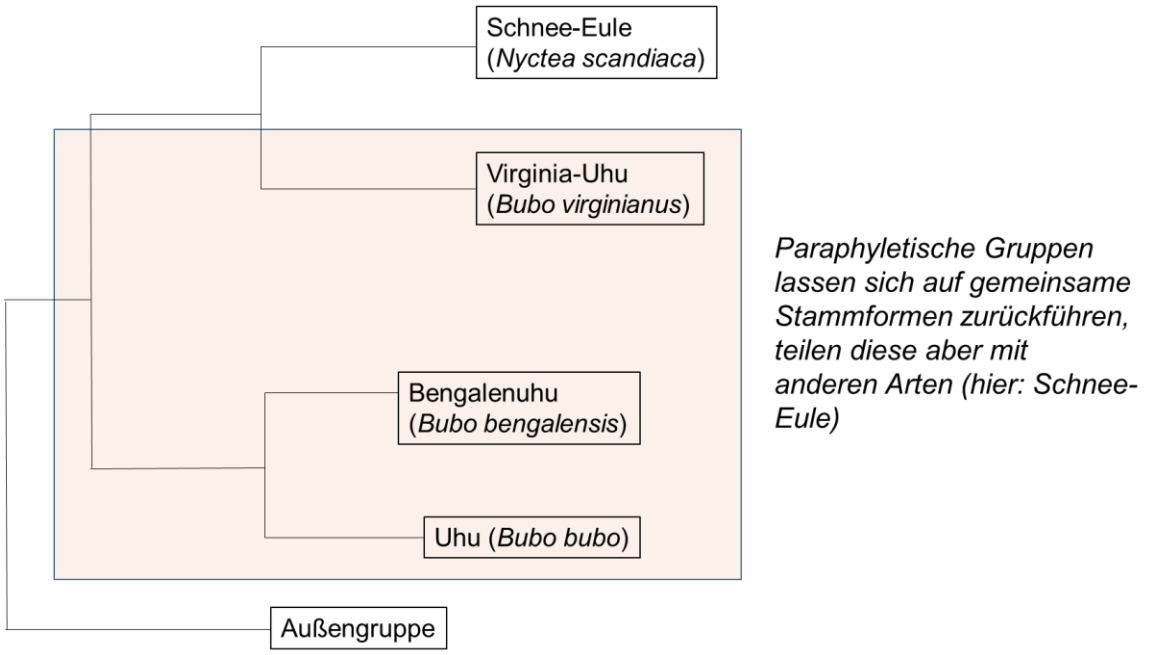


Abb. 4: Paraphyletische Gruppe.

3. POLYPHYLETISCHE GRUPPEN

Polyphyletische Gruppen gehen auf mehr als eine Stammart zurück, sind also nicht unbedingt näher miteinander verwandt.

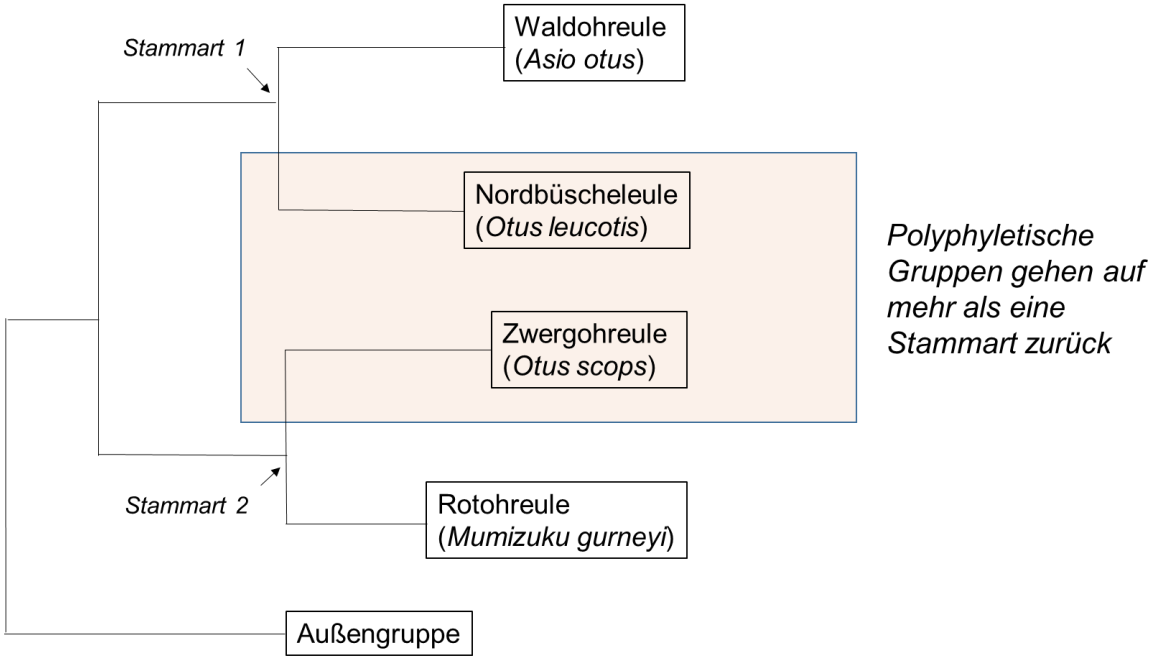


Abb. 5: Polyphyletische Gruppe.

Polyphyletische Gruppen kommen vor allem dadurch zustande, dass Merkmale auch unabhängig voneinander mehrmals in der Evolution entstehen können. Solche Merkmale werden auch als Analogien

bezeichnet. Die Flügel von Vögeln und Fledermäusen sind ein Beispiel für eine solche Analogie, da die beiden Tiergruppen nicht näher miteinander verwandt sind. Eine typische paraphyletische Gruppe bilden die „Würmer“ zu denen die verschiedensten Tiergruppen gehören. Die Nordbüscheleule und die Zwergohreule stellen ein weiteres Beispiel für eine solche paraphyletische Gruppe dar, da sie nicht aus einer gemeinsamen Stammart hervorgegangen sind.

WELCHE TYPEN VON STAMMBÄUMEN GIBT ES?

Man unterscheidet Kladogramme von Phylogrammen. Bei einem Kladogramm ist die Länge der Äste von keinerlei Bedeutung. Es dient lediglich zur Veranschaulichung von Verwandtschaftsverhältnissen über sein Verzweigungsmuster.

In einem Phylogramm spiegelt sich in der Astlänge die Distanz zwischen gewählten Taxa wider. So kann man zum Beispiel die Unterschiede in den DNA-Sequenzen zwischen verschiedenen Taxa über die Astlänge abbilden. Je länger ein Ast, desto mehr Substitutionen haben im Laufe der Zeit stattgefunden.

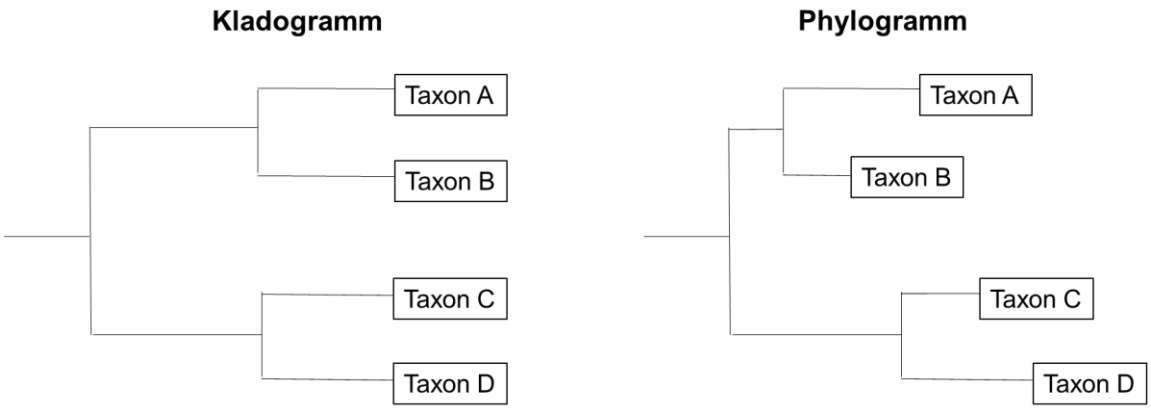


Abb. 6: Veranschaulichung des Unterschieds zwischen Klado- und Phylogramm.

WEITERFÜHRENDE LITERATUR:

Knoop, Müller: Gene und Stammbäume. Spektrum-Verlag, 2009.

Lesk, Arthur, Bioinformatik: Eine Einführung. Berlin: Spektrum, Akad. Verl., 2003.

Sadava, D., Hillis, D., Heller, H.C., Hacker, S. Herausgeber: Markl, Jürgen (Hrsg.), Biologie, Spektrum-Verlag, 2006.

Storch et al.: Evolutionsbiologie. Springer-Verlag, 2001.

ARBEITSAUFTRAG: VERVOLLSTÄNDIGEN SIE IN EINZELARBEIT DAS GLOSSAR ZUM THEMA „PHYLOGENIE UND STAMMBÄUME“. ZEIT: 30 MINUTEN.

Fachbegriff	*Erläuterung*
dichotom	
Kladistik	
Kladogramm	
monophyletisch	
paraphyletisch	
Phylogenie	
Phylogramm	
polyphyletisch	
Schwesterart	
Stammbaum	

LÖSUNGSVORSCHLAG:

Fachbegriff	*Erläuterung*
dichotom	gabelartig verzweigt
Kladistik	Konzept, bei dem die Erstellung von Stammbäumen nach dem Prinzip der Ähnlichkeit im Vordergrund steht
Kladogramm	Stammbaum, bei welchem die Astlängen keine Aussagekraft besitzen
monophyletisch	Taxa, die aus einer gemeinsamen Stammart hervorgegangen sind
paraphyletisch	Abstammungsgemeinschaft, die aber nicht geschlossen ist
Phylogenie	Stammesgeschichte, Lehre von der Abstammung der Lebewesen
Phylogramm	Stammbaum, bei dem die Astlängen (evolutionäre) Distanzen wiedergeben
polyphyletisch	Gruppierungen nicht näher miteinander verwandter Organismen, die aber in Folge von Analogien gemeinsame Merkmale besitzen
Schwesterart	zwei Arten, die aus einer gemeinsamen Stammart hervorgegangen sind
Stammbaum	grafische Darstellung der Abstammung von Taxa

3.2.4.2 TEXTARBEIT: MOLEKULARE UHREN (BPE 10.3)

WAS VERSTEHT MAN UNTER EINER „MOLEKULAREN UHR“?

Im Jahre 1961 schlugen Linus Pauling (1901–1994) und Emile Zuckerkandl (1922–2013) das Konzept der molekularen Uhr vor und berechneten damit evolutionäre Distanzen anhand der Anzahl von Nucleotidsubstitutionen in der DNA bzw. Aminosäuresubstitutionen in Proteinen. Je mehr Unterschiede zwischen zwei oder mehreren Sequenzen zu finden sind, desto weiter sind die beiden Arten aus evolutionärer Sicht voneinander entfernt. Man versucht somit, die Rate der molekularen Evolution zu ermitteln.

Eine molekulare Uhr bezieht sich auf Mutationsraten in einzelnen Genen oder Proteinen und kann nur dann zuverlässig sein, wenn sie mit entsprechenden Fossilfunden etc. abgeglichen werden kann.

EVOLVIEREN ALLE GENE BZW. PROTEINE MIT DER GLEICHEN GESCHWINDIGKEIT?

Man hat festgestellt, dass die Evolutionsrate nicht konstant ist. Von Protein zu Protein variiert diese. Dies hängt davon ab, wie wichtig die Beibehaltung der Raumstruktur für die Funktion eines Proteins ist. Aktive Zentren von Proteinen zum Beispiel evolvieren sehr langsam, da Substitutionen in diesen Bereichen fatale Folgen für die Funktion des Enzyms haben können, die letztlich zur Selektion aus dem Organismus führen können.

Das Kriterium einer molekularen Uhr erfüllen daher vor allem Proteine, deren Evolutionsraten über längere Zeit konstant bleiben. Beispiele hierfür sind der Blutfarbstoff Hämoglobin und das in der Atmungskette wichtige Protein Cytochrom c. Durch den Abgleich mit Fossilfunden kann man ermitteln, wie lange es zurückliegt, dass sich bestimmte Organismengruppen auseinanderentwickelt haben. Diese Zeitspanne trägt man dann gegen die Anzahl der Aminosäuresubstitutionen im Cytochrom c auf. Auf diese Weise kann man erkennen, dass sich die Unterschiede in den Cytochrom c-Sequenzen mit relativ konstanter Rate entwickelt haben. Die Produkte von housekeeping genes, wie zum Beispiel die Histone, eignen sich ebenfalls zur Erstellung molekularer Uhren.

ARBEITSAUFTRAG: BEARBEITEN SIE DIE FOLGENDEN AUFGABEN IN EINZELARBEIT. ZEIT: 20 MINUTEN.

1. Nennen Sie mindestens zwei Argumente, die für die Idee der molekularen Uhr sprechen.
2. Nennen Sie mindestens ein Argument, welches gegen die Idee einer molekularen Uhr spricht.
3. Nennen Sie Gene bzw. Genprodukte, die sich für die Erstellung von molekularen Uhren eignen.
4. Erläutern Sie, an welche Bedingung die Gültigkeit der molekularen Uhr geknüpft ist.

WEITERFÜHRENDE LITERATUR:

Sadava, D., Hillis, D., Heller, H.C., Hacker, S. Herausgeber: Markl, Jürgen (Hrsg.), Biologie, Spektrum-Verlag, 2006.

Storch et al.: Evolutionsbiologie. Springer-Verlag, 2001.

LÖSUNGSVORSCHLAG

1. Nennen Sie mindestens zwei Argumente, die für die Idee der molekularen Uhr sprechen.

- Sie hilft, den Ursprung taxonomischer Gruppen festzulegen.
- Die molekulare Uhr hilft bei der Rekonstruktion der Vergangenheit.
- Man kann die Entstehung der Arten vor Millionen Jahren zeitlich abschätzen.

2. Nennen Sie mindestens ein Argument, welches gegen die Idee einer molekularen Uhr spricht.

- Die Mutationsrate variiert bei verschiedenen Genen. Es eignen sich nur solche Gene/Proteine als molekulare Uhren, deren Evolution sehr konstant verläuft.

3. Nennen Sie Gene bzw. Genprodukte, die sich für die Erstellung von molekularen Uhren eignen.

- Geeignete Gene sind Cytochrom c oder Hämoglobin oder Histone.

4. Erläutern Sie, an welche Bedingung die Gültigkeit der molekularen Uhr geknüpft ist.

- Durchschnittliche Evolutionsgeschwindigkeit aller verwendeten Gene.

3.2.4.3 INFORMATIONSBLATT: REKONSTRUKTIONSMÖGLICHKEITEN VON STAMMBÄUMEN

Rekonstruktionsmöglichkeiten von Stammbäumen

Prinzipiell gibt es zwei Möglichkeiten, einen Stammbaum zu erstellen:

1. Nach dem Distanzverfahren und
2. Nach dem Merkmalsverfahren.

1. DISTANZVERFAHREN

Bei distanzbasierten Methoden zur Stammbaumrekonstruktion werden ausschließlich Gesamtähnlichkeiten zwischen Taxa ermittelt und miteinander verglichen. Bei der Erstellung phylogenetischer Stammbäume nach dem Distanzverfahren nimmt man biologische Arten und stellt die Unterschiede oder Gemeinsamkeiten, wie z. B. molekulare Unterschiede in den Protein- oder Gensequenzen, fest. Anschließend bedient man sich der Clusterbildung (clustering). Hierzu erstellt man eine Matrix, in welcher nur die Unterschiede in den Sequenzen dargestellt werden. Um einen Stammbaum zu konstruieren, wählt man zuerst die beiden Taxa mit den geringsten Unterschieden aus und fügt einen Knoten ein, der ihren gemeinsamen Vorfahren präsentiert. So reduziert man schrittweise die Matrix, bis man alle Taxa in einem Stammbaum zusammengefasst hat.

Ein konkretes Distanzverfahren bildet die Neighbour-Joining-Methode. Mit ihrer Hilfe lassen sich sehr schnell große Datenmengen analysieren. Ein weiterer Vorteil (und gleichzeitig auch Nachteil) ist, dass man als Ergebnis nur einen Stammbaum erhält.

2. MERKMALSVERFAHREN

Bei Merkmalsverfahren der Stammbaumrekonstruktion ordnet man Taxa so an, dass die Anzahl der Merkmalsänderungen möglichst klein ist. Eine solche Methode ist die Maximum-Parsimony-Methode („Methode der größten Sparsamkeit oder Geizigkeit“). Hierbei versucht man denjenigen Stammbaum zu finden, der die wenigsten Mutationen postuliert. Hat man beispielsweise vier Arten, die durch die homologen Sequenzen ATCG, ATGG, TCCA und TTCA charakterisiert sind, so lassen sich mehrere verschiedene Bäume über deren gemeinsamen Ursprung annehmen. Veranschaulicht wird dies exemplarisch in der in Abbildung 1 gezeigten zwei möglichen Stammbäume. Der linke Baum weist nur vier Mutationen auf, während der rechte Baum acht Mutationen enthält. Somit ist der linke Baum nach der Maximum-Parsimony-Methode optimal, da hier das Prinzip der Sparsamkeit verwirklicht ist. Viele Belege und Abgleiche mit Fossilfunden sprechen zudem dafür, dass einfache Lösungswege wahrscheinlicher sind als komplexe.

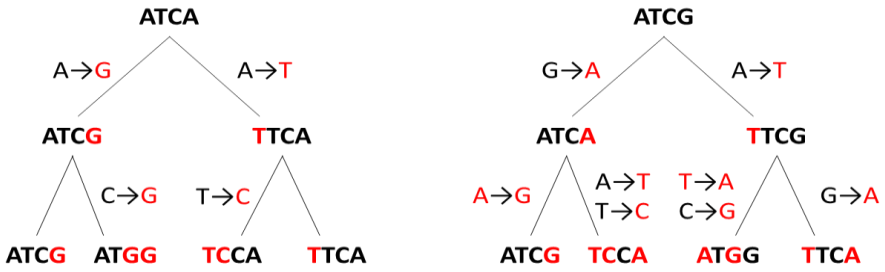


Abb. 1: Prinzip der maximalen Sparsamkeit.

Die Maximum-Parsimony-Methode kann man sowohl auf morphologische und molekulare Merkmale anwenden.

WEITERFÜHRENDE LITERATUR:

Lesk: Bioinformatik – Eine Einführung, Spektrum-Verlag, 2003.

Storch et al.: Evolutionsbiologie. Springer-Verlag, 2001.

3.2.4.4 SEQUENZANALYSE MIT DEM PROGRAMM SEAVIEW (BPE 10.4)

INFORMATIONSBLATT: EINFÜHRUNG IN DES PROGRAMMS SEAVIEW

Link zum SeaView-Programm: http://doua.prabi.fr/software/seaview

ZIEL FÜR DEN EINSATZ VON SEAVIEW

SeaView ist eine plattformübergreifende grafische Benutzeroberfläche für die Ausrichtung mehrerer Sequenzen im Fachbereich der molekularen Phylogenie.

SeaView liest verschiedene Dateiformate, so z. B. FASTA von DNA- und Proteinsequenzen.

Es soll in diesem Einführungsbeispiel aus einem Datensatz von 6 Sequenzen ein multiples Alignment berechnet werden. Also die gleichzeitige Analyse mehrerer Sequenzen. Anschließend werden aus dem Alignment die phylogenetischen Stammbäume berechnet. Dabei kommen die Methode Neighbour-Joining (geringste evolutionäre Entfernung) und Maximum Parsimony (wenigste Substitutionen) zum Einsatz. Bei Bedarf wird der berechnete Stammbaum umgewurzelt.

VERWENDETES MATERIAL UND VORBEREITUNG DER SEQUENZEN

Das verwendete Material (DNA-Sequenzen von Bakterien, siehe Tabelle 1 wurde aus der Datenbank des NCBI heruntergeladen. Von den hier verwendeten Bakterien ist im Vorfeld bekannt, ob sie gram-positiv oder gramnegativ sind.

Es werden folgende Sequenzen der 16S ribosomal RNA mit der Länge von 1343 bp bis 1509 bp verwendet (Tabelle 1). Dabei bedeutet GP grampositiv, GN gramnegativ, AG Außengruppe:

TABELLE 1 VERTRETER GRAMPOSITIVER UND GRAMNEGATIVER BAKTERIEN MIT ACCESSION-NUMBER ZUR AUFFINDUNG DER SEQUENZEN AM NCBI

Bakterien der Gruppe *Firmicutes* :	Bakterien der Gruppe Proteobacteria :
JF833086| *Bacillus subtilis* (GP)	M59156| *Flavobacterium lutescens* (GN)
JF938537| *Bacillus cereus* (GP)	FJ543058| *Legionella* (GN)
JF903803| *Lactobacillus fermentum strain* (GP)	JF703356| *Thiobacillus* (GN)
	Außengruppe *Cyanobacteria* (AG)
	AY218833| *Nostoc commune* (AG)

Die Proteobacteria stellen eine der größeren als Stämme (im Sinne von Phylum) angesehenen Gruppen der Domäne der Bakterien dar. Zu ihnen gehören viele wichtige stickstofffixierende Bakterien und Krankheitserreger. Alle Proteobacteria sind gramnegativ.

Die Firmicutes bilden einen artenreichen Stamm innerhalb der Domäne der Bakterien und bilden zusammen mit den Actinobacteria die große Gruppe der meist grampositiven Bakterien. Viele spielen als Krankheitserreger eine wichtige Rolle.

Die Cyanobakterien bilden eine Abteilung der Domäne der Bakterien. Sie zeichnen sich vor allen anderen Bakterien durch ihre meist, aber nicht in jeder Art vorhandene Fähigkeit zur oxygenen Photosynthese aus. Früher wurden sie zu den Phycophyta (Algen) gerechnet und als Klasse Cyanophyceae (Blaualgen) geführt. Sie zählen zu den ältesten Lebensformen überhaupt. Die Cyanobacteria sind gramnegativ, sie treten hier in der Funktion der Außengruppe auf.

Die Sequenzen werden aus der GenBank (NCBI) in ein FASTA-Dokument geladen. Der Speichername ist **Einführung.fasta**. Diese Datei und alle weiteren, im Verlauf des Beispiels entstehenden Dateien, sollen in einem Ordner **Einführung** gespeichert werden.

AUFBAU EINES FASTA-SEQENZEINTRAGS

Jeder Eintrag einer Sequenz besteht aus der Kommentarzeile und der darunter stehenden Sequenz. Die Informationen sind derart aufbereitet, dass nach dem „>„-**Zeichen** diejenigen Informationen stehen, die später im Baum zu lesen sein sollen. Nach einem Leerzeichen in der Information wird der Text abgebrochen. Daher sind **Unterstriche** einzusetzen. 20 Zeichen nach dem „>„**Zeichen** wird der Text ebenfalls abgebrochen. Es dürfen keine Sonderzeichen verwendet werden.

HIER WERDEN VERWENDET:

>GP_Name für grampositive Bakterien,
>GN_Name für gramnegative Bakterien,
>AG_Name für Außengruppe

Zur Veranschaulichung sieht man hier die ersten 4 Zeilen der verwendeten Sequenz **Einführung.fasta** im FASTA-Format (Abbildung 1). Dabei wird wegen des Leerzeichens vor der Accession number M59156.1 nur der Text >GN_Flavobacterium am Ast des phylogenetischen Baumes stehen.

>GN_Flavobacterium M59156.1 Flavobacterium lutescens 16S ribosomal RNA
NAACTGAAGAGTTTGATCATGGCTCAGATTGAACGCTGGCGGCAGGCCTAACACATGCAAGTCGAGCGGA
TGAGTGGAGCTTGCTCCATGATTCAGCGGCGGACGGGTGAGTAATGCCTAGGAATCTGCCTGGTAGTGGG
GGACAACGTTTCGAAAGGAACGCTAATACCGCATACGTCCTACGGGAGAAAGTGGGGGATCTTCGGACCT
…

Abb. 1: Beginn einer der verwendeten Sequenzen im FASTA-Format (Einführung.fasta)

ARBEITEN MIT DEM PROGRAMM SEAVIEW/DATEN LADEN

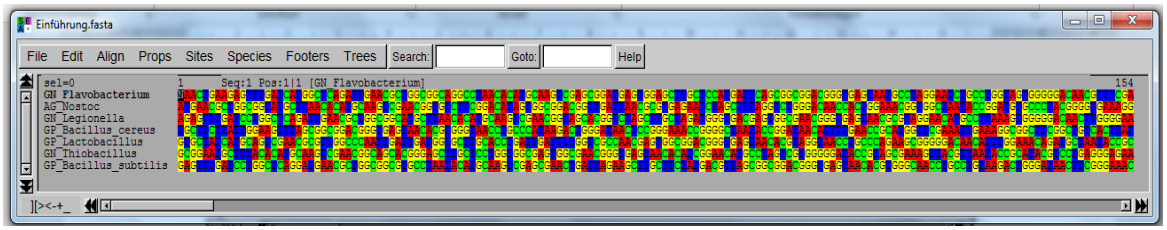


Abb. 2: Die Abbildung zeigt die Ansicht des Datensatzes vor dem Alignen.(Einführung.fasta)

Der Datensatz **Einführung.fasta** wird nun in SeaView mit File/open geladen (Abbildung 2). Die Basen sind in verschiedenen Farben markiert.

Vor dem Alignen wird nun das Programm **muscle** über **Align/Alignment options/muscle** eingestellt (Abbildung 3). Diese Einstellung ist geeignet bei besonders langen Sequenzen oder bei einem Datensatz von 10 oder mehr Sequenzen.

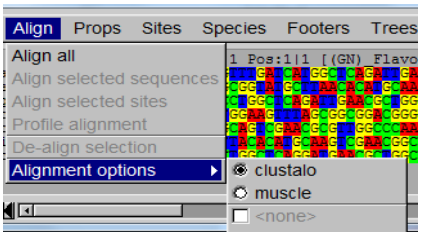


Abb. 3: Dialogbox mit Align-Optionen

DAS ALIGNMENT ERSTELLEN

Dann erfolgt das Ausrichten der Sequenzen über den Befehl **Align/Align all**. Je länger die Sequenzen sind bzw. je mehr Organismen im Datensatz sind, desto größer ist der Rechenaufwand, desto länger muss man auf das Ergebnis warten. Sobald das **OK** von Grau nach Schwarz wechselt, ist das Alignment berechnet. **OK klicken** (Abbildung 4).

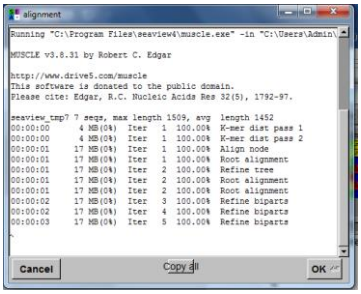


Abb. 4: Dialogbox für den Ablauf des Alignens

Fehlermöglichkeiten: Hat man irgendwann das >Zeichen vergessen oder steht in der 1. Zeile keine Information, dann kann das Programm nicht alignen und man bekommt einen Negativ-Bescheid.

Falls die Sequenzen zu sehr differieren, wird das Alignen abgebrochen und das Programm liefert nach einiger Wartezeit einen Negativ-Bescheid.

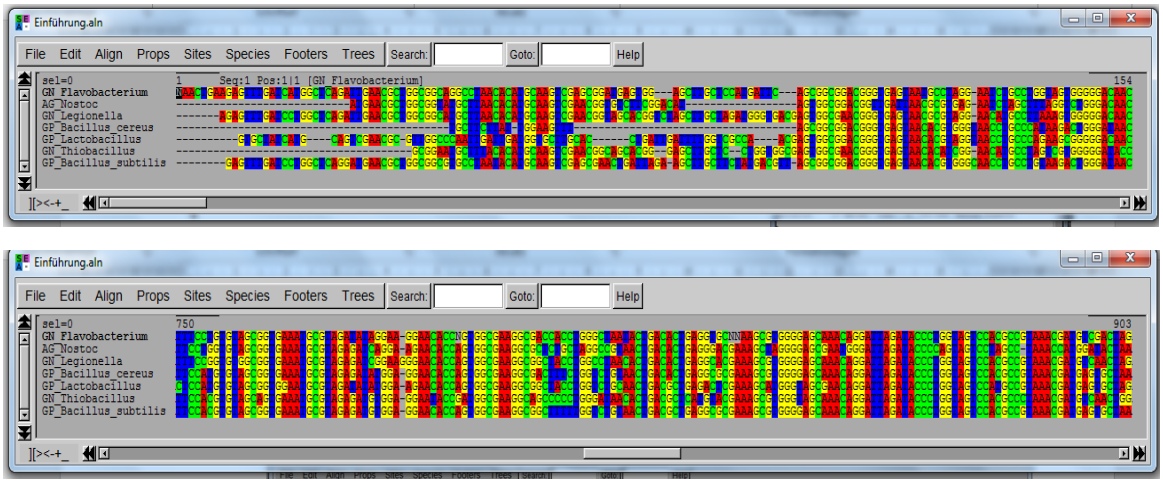


Abb. 5: Hier dargestellt sind die Position 1 bis 154, der Balken ist daher ganz links am Anschlag im Alignment. Zum Vergleich wird auch die Position 750 bis 903 gezeigt, der Balken ist in der Mitte (Einführung.aln).

Das Alignment sollte man kritisch begutachten. Das Ergebnis ist eine Ausrichtung der Sequenzen unter dem Aspekt, so viele übereinstimmende Matches zu erreichen und so wenig Gaps wie möglich einzufügen (Abbildung 5). Das Alignment unterscheidet sich (fast) immer durch das Auftreten zahlreicher Gaps von dem zuerst geladenen Datensatz. Abschließend wird das Alignment unter der Endung **.aln** im Ordner gespeichert. Neben Einführung.fasta liegt nun auch die Datei Einführung.aln im Ordner.

DIE BERECHNUNGSVERFAHREN FÜR DEN STAMMBAUM

Von diesem Alignment **Einführung.aln** ausgehend kann ein Distanzbaum nach einer distanzbasierten Methode (Distance methods, Abbildung 6), z. B. dem Neighbour-Joining-Verfahren (NJ) berechnet werden, nach der motivbasierten Methode (Maximum-Parsimony, MP), oder nach der wahrscheinlichkeitsbasierten Methode (Phyllis-Maximum-Likelihood, PhyML, ML) ein Maximum-Likelihood-Baum berechnet werden.

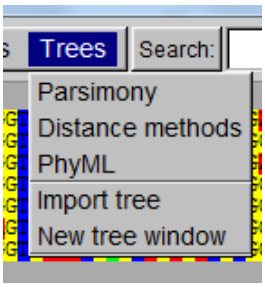


Abb. 6: Pulldown-Menü der Rekonstruktionsmethoden

Fehlermöglichkeit: Es ist möglich, einen Baum aus einem nicht-alignten Datensatz zu berechnen. Das Ergebnis zeigt dann aber eine unerwartete Struktur und die Gruppierungen sind auf den ersten Blick als falsch zu erkennen.

BERECHNUNG EINES PHYLOGENETISCHEN BAUMES NACH VERSCHIEDENEN METHODEN:

Die Ergebnisse der einzelnen Baumberechnungsmethoden (NJ und MP) sind unten dargestellt. Beide Methoden sind bei SeaView leicht durchzuführen.

BAUMBERECHNUNG NACH NEIGHBOUR-JOINING

Sobald man mit der Wahl von Trees bei SeaView in die Baumrekonstruktion einsteigt, öffnet sich ein Dialogfenster. Hier wurde mit **Trees/Distance methods** die Distanzmethode gewählt, da NJ ein distanzbasierter Rechen-Algorithmus ist. Nach Wahl der Distanzmethode öffnet sich wieder eine Dialogbox (Abbildung 7).

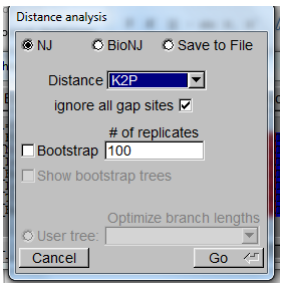


Abb. 7: Dialogbox zur Einstellung der Distanzkorrektur

Die Einstellung in der Dialogbox für die Baumberechnung, ist NJ und K2P (Kimura2Parameter). Es kann prinzipiell noch zusätzlich das Bootstrapping gewählt werden.

NJ ist bekannt für die kurze Rechenzeit. Es wird ein phylogenetischer Baum dargestellt, der links die Wurzel besitzt (Abbildung. 8). Die Darstellung der ausgewählten 7 Sequenzen zeigt, dass die grampositiven und die gramnegativen Bakterien jeweils eine eigene Gruppe bilden.

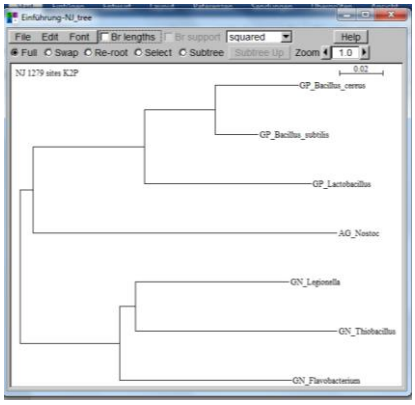


Abb. 8: Darstellung des berechneten phylogenetischen Baums mit NJ

Die Außengruppe Nostoc bildet eine Grenze zwischen den beiden Gruppen. Leider liegt die Außengruppe mitten im Baum und nicht außen, wie es wünschenswert ist.

Damit die Außengruppe *Nostoc* an den oberen oder unteren Rand gestellt wird, wählt man *Re-root*. Sofort erhalten alle Knoten ein schwarzes Quadrat. Durch Klicken auf dieses Quadrat nahe bei *Nostoc*, wird nun *Nostoc* zur neuen Wurzel (outgroup) des Baumes erklärt. Der Baum wird „umgewurzelt“. Der Eintrag *Nostoc* hat dann kein Quadrat mehr (Abbildung. 9). Anschließend werden durch Klicken auf „*Full*“ die schwarzen Quadrate wieder entfernt. Für die Änderung der Ansicht ist es möglich, die einzelnen Äste oder Astgruppen an den Knoten zu drehen („*Swap*“). Man klickt dazu einfach auf das Quadrat des letzten gemeinsamen Vorfahrens.

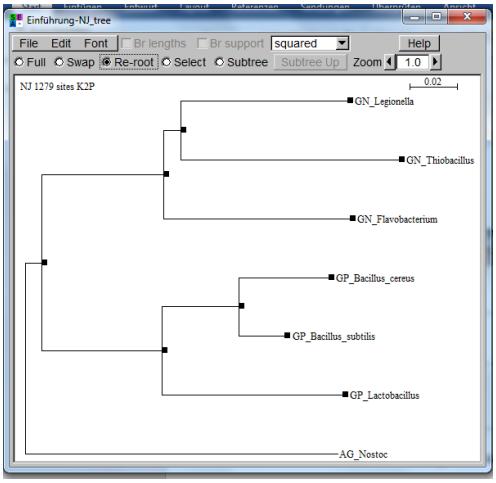


Abb. 9: Phylogenetischer Baum mit NJ und squared (Einführung-NJ_tree.trees)

Das fertige Ergebnis wird mit der Endung .trees als rooted tree oder unrooted tree oder auch als PDF-Datei gespeichert. Hier **Einführung-NJ_tree.trees**

Die Darstellung als Phylogramm, also mit Ästen unterschiedlicher Länge, heißt hier squared.

ANDERE BAUMDARSTELLUNG DES BERECHNETEN STAMMBAUMES:

Die Darstellung circular zeigt den Baum als **unrooted tree.** Es ist die Darstellung des Baums in ungewurzelter Weise. Dadurch wird die reine Struktur des Baumes dargestellt. Man sieht gut, dass *Nostoc* die 6 Arten in 2 Gruppen trennt. Diese Darstellung des radialen Baumes ist günstig, wenn es keine Außengruppe gibt, sondern nur die Struktur als Ergebnis gesucht war (Abbildung 10).

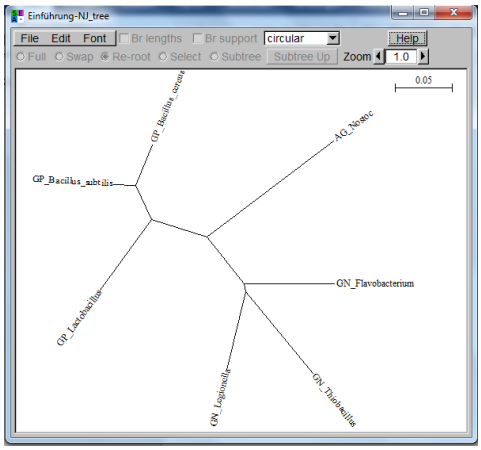


Abb. 10: Phylogenetischer Baum mit NJ und circular (Einführung-NJ_tree.trees)

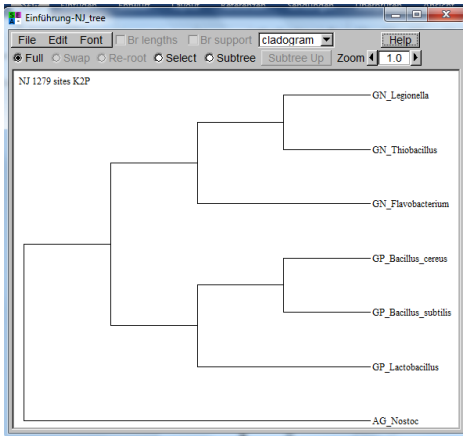


Abb. 11: Phylogenetischer Baum mit NJ und cladogramm (Einführung-NJ_tree.trees)

In diesem **Kladogramm** (Abb. 11) sind alle Astlängen angeglichen. Somit sind alle Namen- und Verzweigungsstrukturen sehr deutlich zu erkennen. Diese Darstellung wird verwendet, wenn man davon ausgeht, dass alle verwendeten Organismen jetzt leben und somit auf einer Ebene liegen. Die Astlängen geben keine Informationen darüber, wie groß die Unterschiede in den verwendeten Sequenzen waren. Auch über die Zeit, die nach Aufspaltung in einzelne Arten verstrichen ist, wird nichts gesagt.

BAUMBERECHNUNG NACH DEM MAXIMUM-PARSIMONY-VERFAHREN

Sobald man mit der Wahl von Trees bei SeaView in die Baumrekonstruktion einsteigt, öffnet sich ein Dialogfenster. Hier wurde mit **Trees/Parsimony** diejenige Methode gewählt, die mit informativen Spalten im Alignment arbeitet. Rechts wird die Dialogbox für die Berechnung des Maximum-Parsimony-Baumes gezeigt. Es kann die vorgegebene Einstellung übernommen werden (Abbildung 12). Zusätzlich kann die statistische Absicherung Bootstrapping gewählt werden.

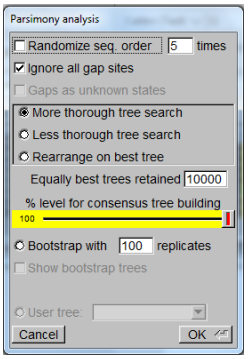


Abb. 12: Dialogbox zur Baumkonstruktion
Nach der Maximum-Parsimony Methode-Baumes

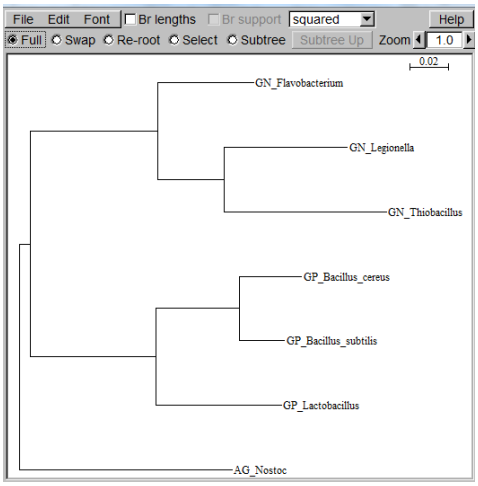


Abb. 13: phylogenetischer Baum mit MP und squared.
(Einführung-Dnapars MP.trees)

Nach dem Re-root mit Nostoc, zeigt das Programm die Darstellung des phylogenetischen Baums, wie er sich mit der Berechnung durch das Maximum Parsimony-Verfahren darstellt (Abbildung 13).

Nostoc ist wie zu erwarten die outgroup und die beiden Bakteriengruppen GP und GN sind deutlich getrennt.

Im Einführungsbeispiel wurde auf das Boostrapping verzichtet. Es gibt für die statistische Absicherung ein separates Infoblatt.

Diese Einführung kann zum Selbststudium verwendet werden. Sie kann auch als Ergänzung zum Unterrichtsmaterial eingesetzt werden.

Abb. 2 – 13 mit Genehmigung von Microsoft

ARBEITSBLATT MAMMUT

HOFFNUNG

Jedes Mal, wenn ein tiefgefrorenes Mammut (Abbildung 1) aus der Eiszeit gefunden wird, flammt die Hoffnung auf, man könne aus dem alten Gewebe ein lebendes Mammut klonen. Bisher sind alle Versuche gescheitert. Forscher in den USA versuchen jetzt, Elefantengenome so zu verändern, dass sich statt Elefanten Mammuts entwickeln.

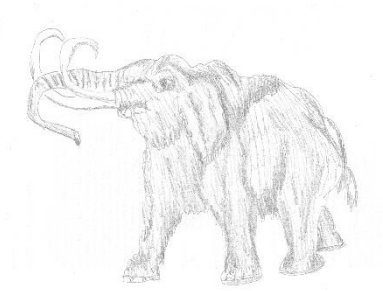


Abb. 1: Mammut

BESONDERHEIT IM FALLE DES MAMMUTS

So erläutert **George Church**, Genetiker von der Harvard Medical School in Boston, wie das bewerkstelligt werden soll: „Wir besitzen bereits die vollständige Erbinformation von Mammuts aus zwei verschiedenen Regionen. Die Qualität der Daten ist sehr gut. Wir wissen über das Mammut-Genom fast schon so viel wie über das menschliche Genom – und mehr als über das des Elefanten."

Das Wollhaarmammut „*Mammuthus primigenius*" und der Asiatische Elefant „*Elephas maximus*" sind zum Beispiel zwei verschiedene Arten; aber sie sind eng miteinander verwandt. Enger als der asiatische mit dem Afrikanischen Elefanten.

EIN NEUER ELEFANT ENTSTEHT

Alexei Tikhonov von der Russischen Akademie der Wissenschaften in St. Petersburg hatte schon 2015 eine Idee: „Wir nehmen eine lebende Zelle eines Elefanten und vergleichen die Erbinformation mit der eines Mammuts. Dann verändern wir mit den Methoden der Gentechnik viele Tausend Gene. Theoretisch lässt sich so in der Elefantenzelle eine Erbinformation herstellen, die genau dem Erbgut eines Mammuts entspricht."

Das Ergebnis wäre kein Mammut, wie in der Eiszeit, sondern ein Mammut-ähnlicher Elefant, räumt George Church ein. Aber diese Mammut-Elefanten könnten in der Steppe Sibiriens oder in anderen Regionen am Polarkreis die ökologische Funktion des Mammuts ausüben, spekuliert er.

Diese Überzeugung drückte er mit den folgenden Worten aus: „Als das Mammut ausstarb, hatte das dramatische Folgen für das ganze Ökosystem. Die Steppe und ihre ökologische Vielfalt verschwanden. Denn das Mammut ist eine Schlüssel-Spezies. Nur Mammuts hauen mal eben einen Baum um."

So könnte ein Mammut-ähnlicher Elefant ein untergegangenes Ökosystem wiederbeleben, hofft der Genetiker. Im heutigen Sibirien könnte wieder eine urzeitliche Eiszeitsteppe entstehen, inklusive all der Arten, die bis heute überdauert haben. Was fehlt, wird einfach mit Methoden der Gentechnik wieder neu erschaffen.

AUFGABE:

Sie können dem Mammut-Spezialisten Alexei Tikhonov leider nicht helfen, einen Mammut-ähnlichen Elefanten zu erschaffen. Aber Sie können überprüfen, ob das Wollhaarmammut tatsächlich eher mit dem Asiatischen Elefanten, als mit dem afrikanischen, verwandt ist. Anhand von Mitochondrien-DNA (mtDNA) soll dieser Sachverhalt nun belegt werden.

SEQUENZ-MATERIAL

Auf dem Tauschordner liegt eine FASTA-Datei Mammut.fasta, die folgende Sequenzen der jeweils kompletten mtDNA enthält (Tabelle 1).

TABELLE 1 WICHTIGE INFORMATIONEN ZU DEN SEQUENZEN (MAMMUT.FASTA)

.Accession-Nr	Deutscher Name	Systematischer Name		Seq.-Länge
KX027559.1	Präriemammut	*Mammuthus jeffersonii*		16458 bp
NC_005129.2	Asiatischer Elefant Sri Lanka	*Elephas maximus*		16902 bp
NC_000934.1	Afrikanischer Elefant	*Loxodonta africana*		16866 bp
KY616981.1	Waldelefant	*Loxodonta cyclotis*		16007 bp

Es wurde die Sequenz eines weiteren Mammuts (Präriemammut) hinzugefügt, so kann man beobachten, ob sich das Wollhaarmammut näher bei Elefanten von heute oder doch näher bei seinen ausgestorbenen Verwandten wiederfindet. Der Waldelefant ist ein Afrikanischer Elefant. So kann die Trennung von Afrikanischen und Asiatischen Elefanten besser dargestellt werden. Nun kommt also noch das Wollhaarmammut *Mammuthus primigenius* dazu, welches beim NCBI unter der Accession-Nr. LC136999.1 abgelegt ist. Aber wo im phylogenetischen Baum wird es zu finden sein?

ARBEITSAUFTRAG: BEARBEITEN SIE DIE AUFGABEN IN EINZELARBEIT (ZEITVORGABE: 90 MIN.).

1. Kopieren Sie die Datei **Mammut.fasta** in einen Ordner **Mammut Projekt**.
2. Ergänzen Sie in der FASTA-Datei die Nukleotidsequenz des Wollhaarmammuts, das Sie im NCBI finden. Achten Sie auf die korrekte Darstellung in der Kommentarzeile gleich nach dem „>„**-Zeichen**. Speichern Sie die Datei unter dem gleichen Namen.
3. Alignen Sie die Sequenzen mit dem Programm SeaView/Align/muscle und speichern Sie das multiple Alignment unter **Mammut.aln**.
4. Rekonstruieren Sie einen Neighbour-Joining-Baum und einen Maximum-Parsimony-Baum. Speichern Sie das Ergebnis unter **Mammut NJ.trees** und **Mammut MP.trees**,
5. Nehmen Sie nun begründet Stellung zur Behauptung:
   „Anhand von Mitochondrien-DNA soll nun belegt werden: Das Wollhaarmammut „*Mammuthus primigenius*“ und der Asiatische Elefant „*Elephas maximus*“ sind eng miteinander verwandt. Zudem soll die Behauptung belegt werden, dass die Verwandtschaft enger ist als die zwischen dem Asiatischen mit dem Afrikanischen Elefanten.“

ARBEITSBLATT MAMMUT LÖSUNG

ARBEITSAUFTRAG: BEARBEITEN SIE DIE AUFGABE IN EINZELARBEIT (ZEITVORGABE: 20 MIN.).

Ergebnis zu 1 und 2:

Die Datei ergänzt mit dem Wollhaarmammut ist in den Anlagen unter **Mammut mit Wollhaarmammut.fasta** gespeichert.

Die FASTA-Datei soll folgende Kommentarzeilen zeigen, besonders der Beginn mit dem Aufbau **>Name_auf_Deutsch** ist wichtig (Abbildung 2). Dann stehen deutsche Namen an den Ästen im Baum.

>Präriemammut_jefferson KX027559.1 Mammuthus jeffersonii isolate Wyanet_ISM07 mitochondrion, partial genome
GTTAATGTAGCTTAAAACAAAAGCAAGGTACTGAAAATACCTAGACGAGTATATCCAACT
CCATAAACAACAAAGGTTTGGTCCCGGCCTTCTTATTGGTTACTAGGAAACTTATACATG

….

>asiatischer_Elefant_Sri_Lanka NC_005129.2 Elephas maximus mitochondrion, complete genome
GTTAATGTAGCTTAAAACAAAAGCAAGGTACTGAAAATACCTAGACGAGTATATCCAACT
CCATAAACAACAAAGGTTTGGTCCCGGCCTTCTTATTGGTTACTAGGAAACTTATACATG

….

>afrikanischer_Elefant NC_000934.1 Loxodonta africana mitochondrion, complete genome
GTTAATGTAGCTTAAAACAAAAGCAAGGTACTGAAAATACCTAGACGAGTATATCCAACT
CCATAAACAACAAAGGTTTGGTCCCGGCCTTCTTATTGGTTACTAGGAAACTTATACATG

….

>afrikanischer_Waldelefant KY616981.1 Loxodonta cyclotis isolate GR0039 mitochondrion, complete genome
GTTAATGTAGCTTAAAACAAAAGCAAGGTACTGAAAATACCTAGACGAGTACATCCAACT
CCATAAACAACAAAGGTTTGGTCCCGGCCTTCTTATTGGTTACTAGGAAACTTATACATG

….

>Wollhaarmammut LC136999.1 Mammuthus primigenius mitochondrial DNA, complete genome, isolate: Yuka
GTTAATGTAGCTTAAAACAAAAGCAAGGTACTGAAAATACCTAGACGAGTATATCCAACT
CCATAAACAACAAAGGTTTGGTCCCGGCCTTCTTATTGGTTACTAGGAAACTTATACATG

…..

Abb. 2: hier sind jeweils nur die Kommentarzeile und die ersten 120 Basen des FASTA-Formates gezeigt (Mammut_Projekt nur Kommentarzeilen.fasta).

ERGEBNIS ZU 3:

Deutlich sind die zwei Darstellungen zu unterscheiden (Abbildung 3, Abbildung 4), denn erst nach dem Alignen sind Gaps eingefügt.

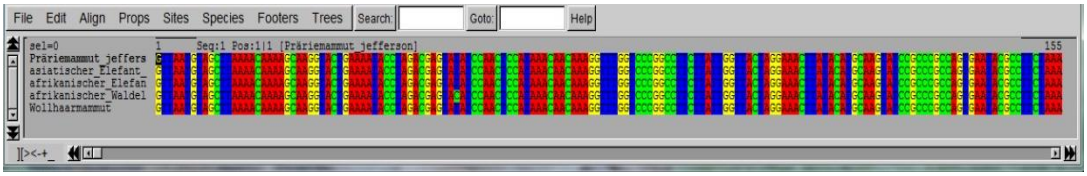


Abb. 3: Die Abbildung zeigt die Ansicht des Datensatzes vor dem Alignen.

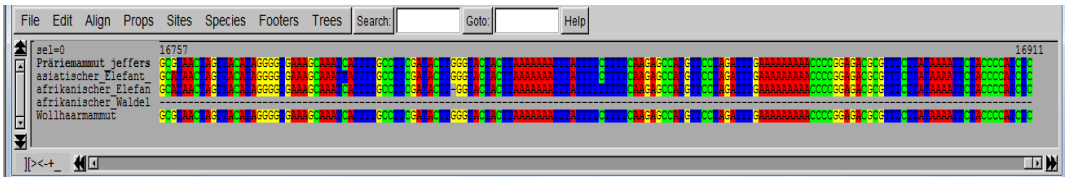


Abb. 4: Diese Abbildung zeigt die Ansicht nach dem erfolgreichen Alignen (Mammut.aln).

ERGEBNIS ZU 4:

Nur an der Beschriftung ist zu erkennen, dass die zwei Stammbäume (Abbildung 5, Abbildung 6) mit unterschiedlichen Berechnungsverfahren erstellt wurden. Das Ergebnis beider Rekonstruktionsmethoden stimmt überein und zeigt daher mit großer Wahrscheinlichkeit eine stammesgeschichtliche Verwandtschaft zwischen dem Wollhaarmammut und dem Asiatischen Elefanten an.

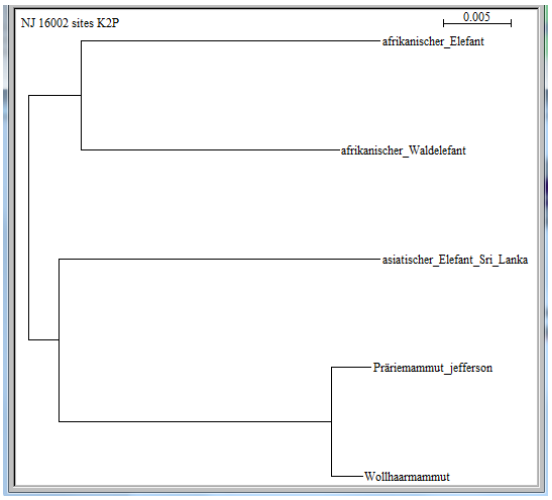


Abb. 5: Phylogenetischer Baum mit NJ und squared (Mammut NJ.trees)

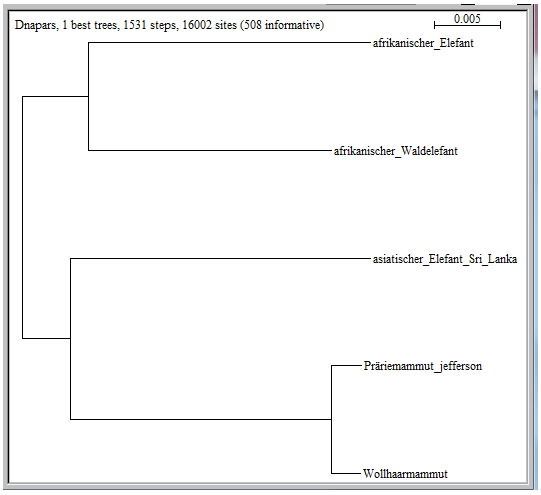


Abb. 6: Phylogenetischer Baum mit MP und squared (Mammut MP.trees)

Abb. 3 – 6 mit Genehmigung von Microsoft

ERGEBNIS ZU 5:

Die Stellungnahme könnte folgende Argumente enthalten:

Das Wollhaarmammut ist am nächsten mit dem Präriemammut verwandt, wie beide Rekonstruktionsmethoden nahelegen. Der nächste Verwandte unter den Elefanten ist, wie angenommen, der Asiatische Elefant. Deutlich sichtbar ist auch, dass der Asiatische Elefant wirklich näher mit dem ausgestorbenen Wollhaarmammut verwandt ist, als mit dem Afrikanischen Elefanten. Daher ist die Idee, einen Mammut-ähnlichen Elefanten auf Basis des Genoms des Asiatischen Elefanten zu entwickeln, gut nachvollziehbar.

Die geografische Nähe zwischen Sibirien und dem jetzigen Verbreitungsgebiet des Asiatischen Elefanten untermauert die nähere Verwandtschaft von Wollhaarmammut und asiatischem Elefanten.

Es ist nicht völlig auszuschließen, dass zwischen den Vorfahren des Wollhaarmammuts und des Asiatischen Elefanten ein horizontaler Transfer von Genen oder Genomabschnitten stattgefunden hat. Dadurch könnte eine gewisse Sequenzähnlichkeit entstanden sein, die sich über die folgende lange Entwicklungszeit immer mehr festigte.

Es wäre möglich, dass das Mammut eigentlich mit dem Afrikanischen Elefanten näher verwandt ist, aber die verwendete Sequenz einige Gene enthält, die sich beim Afrikanischen Elefanten schneller entwickelt haben, sodass sich bei diesen mit der Zeit mehr Mutationen angesammelt haben.

WEITERFÜHRENDE LITERATUR:

https://www.deutschlandfunk.de/genforschung-erbgut-vom-mammut-zum-elefanten.676.de.html?dram:article_id=327439 (letzter Aufruf 02.01.2020).

https://www.welt.de/wissenschaft/article709980/Das-Erbe-der-Urzeit-kann-entraetselt-werden.html (letzter Aufruf 27.1.2020).

Lesk, Arthur, Bioinformatik: Eine Einführung. Berlin: Spektrum, Akad. Verl., 2003.

3.2.4.5 STATISTISCHE ABSICHERUNG: BOOTSTRAPPING (BPE 10.5)

INFORMATIONSTEXT ZUM BOOTSTRAPPING

FRAGESTELLUNG: IST MEIN BERECHNETER STAMMBAUM ZUVERLÄSSIG?

Wenn man schließlich mit einem geeigneten Verfahren einen phylogenetischen Baum berechnet hat, stellt sich die berechtigte Frage, wie „sicher“ ist denn nun der von uns berechnete Baum? Kann das Verkleinern des Datensatzes durch Weglassen einer einzelnen Art, die gesamte Topologie des Baumes ändern? Oder gibt es einen Effekt nur begrenzt auf das Umfeld der weggelassenen Art? Andererseits könnte man den Datensatz durch Hinzufügen einer neuen Art vergrößern und prüfen, wie sich das auf die Topologie des Baumes auswirkt.

Eine Antwort auf die Frage: „Wie sicher ist mein Ergebnis?“, erhält man dadurch jedoch nicht.

DIE METHODE

Eine gute Methode, um die Robustheit der Rechenergebnisse zu überprüfen, ist das sogenannte „bootstrapping“. Dazu bestimmen wir die statistische Wahrscheinlichkeit dafür, wie sicher die einzelnen Knoten in unserem Baum sind.

Nur wenn wir von Sequenzen ausgegangen sind, die als unsere Sequenzen eindeutig zu analysieren waren, wird es durch das Rechenverfahren einen eindeutigen Baum ergeben. Dieser eindeutige Baum bestätigt mit großer Wahrscheinlichkeit alle Verzweigungen, die wir mit dem Rechenverfahren bestimmt haben. Diese Eindeutigkeit basiert dann allerdings auf einer statischen Wahrscheinlichkeit und ist keine 100%ige Sicherheit. Aber das ist im Vergleich zu vorher, immens viel.

DAS VERFAHREN

Die Grundlage der Methode des Boostrapping nach Joe Felsenstein (1985) lautet: Wir gehen davon aus, dass unser Datensatz von z. B. fünf Organismen und einer jeweiligen Sequenzlänge von z. B. 500 Basen eine repräsentative Auswahl von einer größeren vorhandenen Datenmenge darstellt.

Stellen wir uns den Datensatz als multiples Alignment vor. Daraus wird ein erster Baum – der Referenzbaum – berechnet. Wir erzeugen nicht aus dem größeren Datensatz mit noch mehr Arten, sondern aus der vorhandenen Datenmenge mit unseren fünf Organismen einen neuen Datensatz. Dies passiert, indem 500-mal eine Spalte zufällig aus dem vorhandenen Datensatz gezogen wird (Abbildung 1). Dabei kann jedoch jede Spalte mehrfach gezogen werden. Also wir „ziehen mit Zurücklegen“.

Aus diesem zufälligen neuen Datensatz wird nun ein Baum berechnet. Dieses Bootstrapping-Verfahren wird 100-mal oder 1000-mal, je nach verfügbarer Zeit, wiederholt und jedes Mal ein Baum berechnet.

Original-Alignment

		1	2	3	4	5	…
Seq	1	C	G	C	C	T	…
Seq	2	G	G	T	C	C	…
Seq	3	G	G	C	T	G	…
Seq	4	G	G	A	T	A	…
Seq5	5	A	A	T	G	A	…

1. Neues Alignment

		3	1	4	3	5	….
Seq	1	C	C	C	C	T	…
Seq	2	T	G	C	T	C	…
Seq	3	C	G	T	C	G	…
Seq	4	A	G	T	A	A	…
Seq5	5	T	A	G	T	A	…

2. Neues Alignment

		2	2	1	4	5	…..
Seq	1	G	G	C	C	T	…
Seq	2	G	G	G	C	C	…
Seq	3	G	G	G	T	G	…
Seq	4	G	G	G	T	A	…
Seq5	5	A	A	A	G	A	…

3. … usw.

Abb. 1: Zufälliges Ziehen von Spalte für ein neues Alignment aus dem Original-Alignment.

DIE AUSWERTUNG

Je häufiger nun in diesen 100 bzw. 1000 Bäumen ein Knoten zwischen zwei bestimmten Gruppen zu finden ist, umso wahrscheinlicher und sicherer ist diese Verzweigung zwischen diesen zwei bestimmten Gruppen.

Aus der relativen Häufigkeit (Anzahl der Bäume, in denen diese Verzweigung auftritt, geteilt durch die Gesamtzahl der generierten Bäume) kann man eine Wahrscheinlichkeit als Prozentzahl errechnen. Diese Prozentzahlen werden dann im ersten berechneten Baum, dem Ausgangsbaum, an die jeweiligen Knoten notiert. Diese Bootstrapping-Prozentzahlen geben nun an, wie sicher jede einzelne Verzweigung in unserem berechneten Baum ist.

Die Aussagequalität von Bootstrapwerten ist umstritten, weil man mit zufällig veränderten Merkmalen (zufälliges Ziehen) arbeitet.

Nach Joe Felsenstein bedeutet ein Wert von über 95 %, dass eine Verzweigung so gut abgesichert ist, dass eine monophyletische Entstehung (siehe BPE 10.3) angenommen werden kann.

Hillis & Bull (1993) weisen mit ihren Vergleichen von computersimulierten und echten DNA-Sequenzen darauf hin, dass „sogar mehr als 95 % aller Verzweigungen mit einer Bootstrapunterstützung von mehr als 70 % als statistisch sehr gut abgesichert gelten können“.

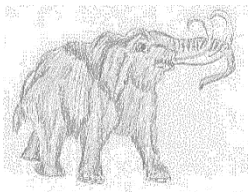
Die Berechnung des Bootstraps von großen Datensätzen ist bei heuristischen Verfahren auch mit modernen Computern sehr zeitaufwendig. In manchen Fällen kann sie deshalb sogar nicht durchgeführt werden.

Wir befinden uns hier bei Werten von Wahrscheinlichkeiten und es bedarf einer vielseitigen Analyse und einer umsichtigen Arbeitsweise, damit man sich auf das Gebiet „statistisch sehr gut abgesichert“ oder „sicher eine monophyletische Entstehung“ wagen kann. Dies kann der Unterricht an der Schule natürlich nicht leisten und das sollte auch den Schülerinnen und Schülern klar vermittelt werden.

ARBEITSBLATT ZUR ANWENDUNG DES BOOTSTRAPPING:

DIE NÄCHSTEN VERWANDTEN DES ELEFANTEN BIS GESTERN

Noch immer ist rätselhaft, mit welcher Tiergattung der Elefant nun als Nächstes verwandt ist. Bis vor kurzem waren da zwei Kandidaten heiß im Rennen, die Seekuh und der Klippschliefer (Abbildung 1). Drei Tierarten, die phänotypisch unterschiedlicher nicht sein könnten, sollen trotzdem so eng miteinander verwandt sein?



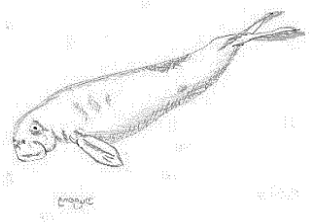
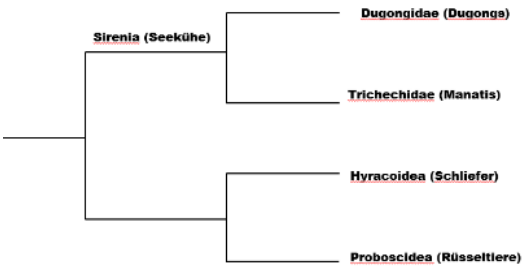


Abb. 1: Elefant, Klippschliefer und Seekuh

Der Elefant trampelt tonnenschwer und trompetend durch die afrikanische Savanne.

Der Klippschliefer wird gerade mal vier Kilogramm schwer, lebt in Felsspalten und liebt Sonnenbaden.

Die Seekuh hat statt der Vorder- und Hinterbeine Flossen und schwimmt trotz ihres massigen Körpers elegant durch die Meere.



Nach der Literatur und der vorherrschenden Lehrmeinung sind bis vor ein paar Jahren die Schliefer die nächsten Verwandten der Elefanten, die ja Rüsseltiere sind (Abbildung 2).

Abb. 2: Innere Systematik der Afrotheria

Eine ältere Publikation aus dem Jahr 2006 zeigt noch Schliefer und Seekuh als gleich weit entfernt in der Verwandtschaft zu den Elefanten.

EIN NEUER VERWANDTER BETRITT DIE BÜHNE:

Doch nun soll ein weiterer Kandidat im Rennen sein: Das erst kürzlich entdeckte Graugesichtige Rüsselhündchen (Abbildung 3). Es lebt in den Wäldern Tansanias und wiegt mit rund 700 Gramm etwa ein Viertel mehr als seine bisher bekannten Verwandten unter den Elefantenspitzmäusen. Das berichten Forscher um Galen Rathbun von der Kalifornischen Akademie der Wissenschaften in San Francisco in der Zeitschrift „Journal of Zoology“, der Zoologischen Gesellschaft Londons. Das Pelztier erhielt den lateinischen Namen *Rhynchocyon udzungwensis*.

Abb. 3: Rüsselhündchen

Auch wenn das jetzt nicht so aussieht: Dies ist ein Verwandter des Elefanten!

Wo unter den bisherigen Kandidaten ist das Rüsselhündchen wohl einzuordnen?

Wird das Rüsselhündchen dem Klippschliefer oder der Seekuh den Rang ablaufen?

AUFGABE:

Das Ziel ist ein Neighbour-Joining-Baum aller Kandidaten, die für den „nächsten Verwandten des Elefanten“ infrage kommen, zu erstellen und eine statistische Absicherung anzuschließen. Danach sollen Sie ein begründetes Urteil zur nächsten Verwandtschaft des Elefanten abgeben. Ist es wirklich immer noch der Klippschliefer? Bestätigt die Analyse des genetischen Materials die frühere Lehrmeinung?

SEQUENZ-MATERIAL:

Auf dem Tauschordner liegt eine FASTA-Datei **Elefant und Verwandte.fasta**, welche die folgenden Sequenzen für das komplette mtDNA-Genom enthält (Tabelle 1).

TABELLE 1: TABELLE MIT WICHTIGEN INFORMATIONEN ZU DEN VERWENDETEN SEQUENZEN (ELEFANT UND VERWANDTE.FASTA)

Accession-Nr	Deutscher Name	Systematischer Name	Seq.-Länge
NC_005129.2	Asiatischer Elefant Sri Lanka	*Elephas maximus*	16902 bp
NC_000934.1	Afrikanischer Elefant	*Loxodonta africana*	16866 bp
NC_004919.1	Klippschliefer	*Procavia capensis*	16743 bp
NC_003314.1	Dugong_Gabelschwanzseekuh	*Dugong dugon*	16850 bp

ARBEITSAUFTRAG: BEARBEITEN SIE DEN ARBEITSAUFTRAG IN EINZELARBEIT

(ZEITVORGABE: 90 MIN.)

1. Kopieren Sie die Datei **Elefant und Verwandte.fasta** in einen Ordner **Elefant Verwandte Projekt**.
2. Ergänzen Sie in der FASTA-Datei die Nukleotidsequenz des Graugesichtigen Rüsselhündchens, das Sie im NCBI finden:
   MH056211.1 *Rhynchocyon udzungwensis*
   Achten Sie auf die korrekte Darstellung in der Kommentarzeile gleich nach dem >-Zeichen.
   Speichern Sie die Datei unter **Elefant und Verwandte mR.fasta**
3. Alignen Sie die Sequenzen mit dem Programm SeaView/Align/muscle und speichern Sie das multiple Alignment unter **Elefant und Verwandte mR.aln**, das bildet die Grundlage für die Rekonstruktion eines ungewurzelten Baumes.
4. Rekonstruieren Sie einen NJ-Baum. Führen Sie zur Ergänzung ein Bootstrapping mit 1000 Wiederholungen durch. Speichern Sie das Ergebnis unter **Elefant und Verwandte mR NJ.trees** bzw. unter **Elefant und Verwandte mR NJ 1000bt.trees**
5. Drucken Sie einen Baum aus, ergänzen Sie, falls notwendig, die Bootstrapping-Werte größer 50.
6. Nehmen Sie nun begründet Stellung zu: „Der nächste Verwandte des Elefanten ist ……?“ und äußern Sie sich kritisch zu Ihrem Ergebnis.

LÖSUNGSBLATT ZUR AUFGABE: ELEFANT UND VERWANDTE

ERGEBNIS ZU 1 UND 2:

Hier sind die Kommentarzeilen und die ersten 2 Zeilen der Sequenz gezeigt, so kann der korrekte Zeilenbeginn überprüft werden (Abbildung 1).

>asiatischer_Elefant_Sri_Lanka NC_005129.2 Elephas maximus mitochondrion, complete genome
GTTAATGTAGCTTAAAACAAAAGCAAGGTACTGAAAATACCTAGACGAGTATATCCAACTCCATAAACAA
CAAAGGTTTGGTCCCGGCCTTCTTATTGGTTACTAGGAAACTTATACATGCAAGTATCCGCCCGCCAGTG

….

>afrikanischer_Elefant NC_000934.1 Loxodonta africana mitochondrion, complete genome
GTTAATGTAGCTTAAAACAAAAGCAAGGTACTGAAAATACCTAGACGAGTATATCCAACTCCATAAACAA
CAAAGGTTTGGTCCCGGCCTTCTTATTGGTTACTAGGAAACTTATACATGCAAGTATCCGCCCGCCAGTG

….

>Dugong_Gabelschwanzseekuh NC_003314.1 Dugong dugon mitochondrion, complete genome
GTTAATGTAGCTTAAACCCCAAAGCAAGGCACTGAAAATGCCTAGATGAGTTCTCCCACTCCATAGACAT
AAAGGTTTGGTCCTGGCCTTCTTATTGGTTCTTAGCGAGCTTACACATGCAAGTATCCGCGCGCCAGTGA

….

>Klippschliefer_NC_004919.1 Procavia capensis mitochondrion, complete genome
GTTAATGTAGCTTAAACAATTATAAAGCAGGGCACTGAAAATGCCCAGATGAGCTTCCAAAGCTCCATAA
ACATTCTAAAGGTTTGGTCCTGGCCTTCTTATTAGTTATCAGTAAGCTTATACATGCAAGCATCCGCACC

….

>Rüsselhund_MH056211.1 Rhynchocyon udzungwensis mitochondrion, complete genome
GTTAATGTAGCTTAAATTTTAAAGCAAAGCACTGAAAATGCTAAGATGAGTCTTTAAGACTCCATAAACA
CAAAGGTTTGGTCCTAGCCTTGCTATTAGTTATTAGCAAAATTATACATGCAAGTATCTGCCCCCCGGTG

….

Abb. 1: Von den 5 Sequenzen sind je 3 Zeilen des FASTA-Formates dargestellt. (Elefant und Verwandte mR nur Kommentarzeilen.fasta)

Hier ist die FASTA-Datei dargestellt, wenn sie im Analyseprogramm geöffnet wird (Abbildung 2). Die Sequenzen sind noch nicht aligned, es sind keine Gaps eingefügt:

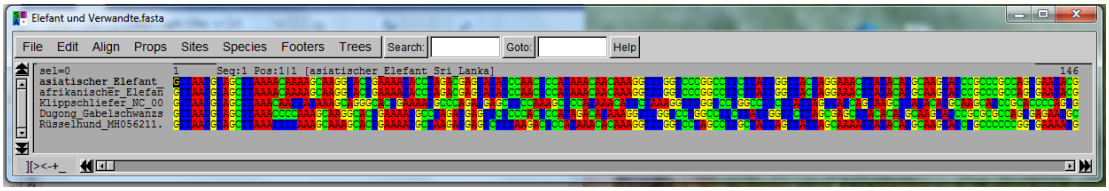


Abb. 2: Die Fasta-Datei im Analyseprogramm, sichtbar sind die Sequenznamen und der Beginn der Sequenzen

ERGEBNIS ZU 3:

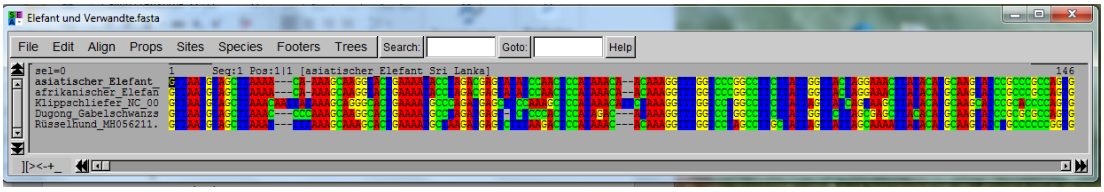


Abb. 3: Die FASTA-Datei als fertiges Alignment (Elefant und Verwandte mR.aln)

Nach dem Alignen sind die eingefügten Gaps gut zu erkennen (Abbildung 3). Mit dem Balken kann das komplette Alignment durchgeschaut werden.

ERGEBNIS ZU 4:

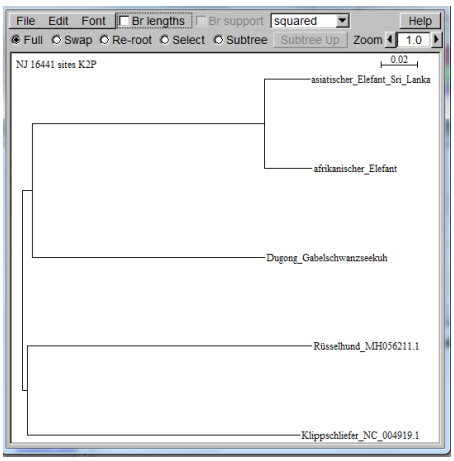


Abb. 4: Elefant und Verwandte mR NJ.trees

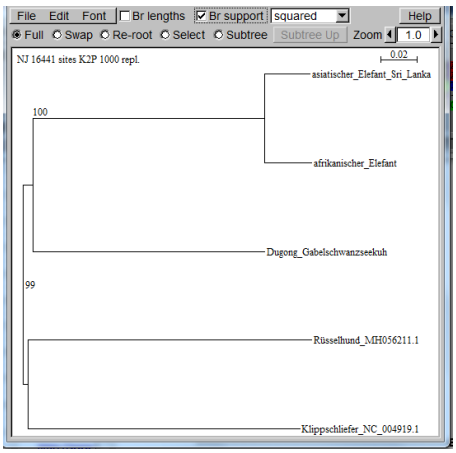


Abb. 5: Elefant und Verwandte mR NJ 1000bt.trees

Hier in der Abbildung haben beide Ergebnisse den gleichen Aufbau (Abbildung 4, Abbildung 5). Es sind also topologisch identische Bäume. Leider gehen durch das Ästetauschen die Zahlen für Bootstrapping oft verloren. Hier ist glücklicherweise kein Tauschen der Äste mit Swap notwendig. Der Baum der Neighbour-Joining-Analyse kann ausgedruckt werden und als Ergebnissicherung zur Aufgabe geheftet werden.

ERGEBNIS ZU 5 UND 6:

Die Stellungnahme der Schülerinnen und Schüler kann die nachfolgenden Aspekte enthalten:

- Nach der Analyse und der Absicherung mit Bootstrapping sagen die Werte über 70 % aus, dass es eine nahe Verwandtschaft zwischen den Elefanten und den Seekühen gibt. Der „neue“ Kandidat, das Rüsselhündchen, hat sich mit dem Klippschliefer als Gruppe schon länger abgespalten. Unsere Ergebnisse besagen, dass die vorherrschende Lehrmeinung, dass der Klippschliefer der nächste Verwandte der Elefanten ist, angezweifelt werden muss.
- Interessant ist es, nachzuforschen, ob es aktuelle Literaturbelege für unser Ergebnis gibt.
- Man sollte immer kritisch überprüfen, ob es auch sein könnte, dass einige DNA-Abschnitte in der mt-Sequenz unabhängig voneinander entstanden sind. Dann könnte ein unbekannter Selektionsdruck zu der Ähnlichkeit geführt haben. In diesem Fall würde uns eine stammesgeschichtliche Verwandtschaft nur vorgegaukelt.
- Oder es kann auch eine Analyse mit nicht-mitochondrialen Genen (z. B. ribosomale RNA) durchgeführt werden.

Abb. 2-5 mit Genehmigung von Microsoft

WEITERFÜHRENDE LITERATUR:

https://www.geo.de/geolino/natur-und-umwelt/9795-rtkl-warum-klippschliefer-elefant-und-seekuh-verwandt-sind (letzter Aufruf 02.01.2020).

https://www.welt.de/wissenschaft/article1613951/Skurriler-Verwandter-des-Elefanten-entdeckt.html
https://www.welt.de/wissenschaft/article1613951/Skurriler-Verwandter-des-Elefanten-entdeckt.html
(letzter Aufruf 02.01.2020).

Matjaz Kuntner, Laura J. May-Collado und Ingi Agnarsson: Phylogeny and conservation priorities of afrotherian mammals (Afrotheria, Mammalia). Zoologica Scripta 40 (1), 2011, S. 1–15,

(letzter Aufruf 02.01.2020).

https://dewiki.de/Lexikon/Schliefer#Systematik (letzter Aufruf 02.01.2020).

https://www.spektrum.de/news/verwandtschaftsverhaeltnisse-des-wollhaarmammuts-geklaert/799132
(letzter Aufruf 02.01.2020).

https://www.spektrum.de/news/ungewoehnliche-verwandte/349863

(letzter Aufruf 02.01.2020).

https://www.tanzania-experience.com/de/blog/klippschliefer-verwandten-des-elefanten/

(letzter Aufruf 02.01.2020).

3.3 Programmierung zur Sequenzanalyse (BPE 11.1 bis 11.3)

3.3.1 VERLAUFSPLAN/STOFFVERTEILUNG

Die Dauer der einzelnen Unterrichtsphasen ist Abhängigkeit vom Vorwissen der Schülerinnen und Schüler. Aus der Informatik sind die Grundkenntnisse der Programmiersprache bereits vorhanden, sodass das Wissen auf biologische Anwendungen übertragen werden kann. Das Material findet sich in der Handreichung, die passenden Python-Programme und Musterlösungen im digitalen Anhang.

DAUER	UNTERRICHTSPHASE, INHALT	MATERIAL, MEDIEN	ANGESTREBTES ERGEBNIS, ERWARTETES SCHÜLERVERHALTEN
45	Einstieg in die Programmierung in der Bioinformatik und in Python BPE 11.1/11.2	Arbeitsblatt: „Kurzer Steckbrief zu Python“	Die SuS können die Bedeutung der Bioinformatik darstellen und begründen den Einsatz von Python als ein relevantes Werkzeug.
45	Einführung in die Verwendete Arbeitsumgebung und deren Installation. BPE 11.1/11.2	Downloads aktueller Versionen z. B. von Visual Studio Code und Python, Arbeitsblatt: „Das erste Programm“	Die SuS können mit der eingeführten Programmierumgebung ein Programm erstellen und dieses ausführen. Die Installation auf einem privaten Rechner ist vorbereitet.
90	Variablen in Python. BPE 11.1/11.2	Arbeitsblatt: „Variablen In Python“	Die SuS nutzen passenden Variablentypen zur Anwendung in einfachen Python-Programmen.
90	Daten aus Dateien in Python auslesen und Dateien aus Python in eine Datei schreiben. BPE 11.1/11.2	Arbeitsblätter: „Daten aus einer Datei lesen“ und „Daten in eine Datei schreiben“	Die Schülerinnen und Schüler lesen eine Fasta-Datei in Python ein und erstellen eine Fasta-Datei mit Python.
90	Arbeit mit Listen in Python. BPE 11.1/11.2	Arbeitsblatt: „Listen in Python“	Die SuS kennen die Funktion und Anwendung von Listen in Python.
90	Dictionaries in Python. BPE 11.1/11.2	Arbeitsblatt „Dictionaries in Python“	Die SuS stellen eine elektronische Translation mithilfe eines Dictionaries in Python.
45	Bedingungen in Python. BPE 11.1/11.2	Arbeitsblatt: „If…Else Bedingung“,	Die SuS nutzen if…else und verknüpfen die Funktion mit der Erstellung einer Liste.

45	Schleifen in Python. BPE 11.1/11.2	Arbeitsblatt: „Schleifen in Python“	Die SuS wenden exemplarisch die While- und die For-Schleife an.
45	Funktionen in Python. BPE 11.1/11.2	Arbeitsblatt „Funktionen in Python“	Die SuS nutzen Funktionen zum Einlesen von Fasta-Dateien.
45	Projekteinführung BPE 11.3	Arbeitsblatt „ORF-Finder“	Die SuS beginnen ein Projekt zum Auffinden von ORFs unter Einbeziehung weiterer Tutorials.
90	Projekt ORF-Finder		

3.3.2 FACHLICHE HINWEISE

Nachdem die Schülerinnen und Schüler einige Programme und Werkzeuge zur Sequenzanalyse kennengelernt und damit deren Bedeutung für die Genomik erkannt haben, soll ihnen nun die Programmierung als prinzipielle Möglichkeit der Entwicklung neuer, bedarfsgerechter Werkzeuge aufgezeigt werden. Hierzu können sie auf bereits vorhandene Grundkenntnisse der Programmierung zurückgreifen, trainieren aber in besonderer Weise die Entwicklung von spezifischen Algorithmen im Kontext der Mustererkennung als Basis der Sequenzanalyse.

3.3.3 DIDAKTISCHE HINWEISE

Die Schülerinnen und Schüler übertragen vorhandene Grundkenntnisse der Programmierung auf die Anwendung einer Programmiersprache, die bevorzugt zur spezifischen Bearbeitung von Fragestellungen der Sequenzanalyse eingesetzt wird. Sie erweitern die Grundkenntnisse um spezifische Elemente, die für die textuelle Mustererkennung als Grundlage einer Sequenzanalyse erforderlich sind.

Die Schülerinnen und Schüler entwickeln einfache Algorithmen, die es ermöglichen, die spezifischen Anforderungen der Sequenzanalyse zu bearbeiten. Sie dokumentieren, kommentieren und implementieren die erarbeiteten Algorithmen. Die in der Regel zu Stundenanfang vorgestellten Befehle werden durch die Übertragung auf fachspezifische Inhalte geübt.

3.3.4 KURZER STECKBRIEF ZU PYTHON

Programmiersprache, in den 1990ern entwickelt von Guido van Rossum (ein Niederländer).

TABELLE 1: EIGENSCHAFTEN VON PYTHON

Der Name Python:	Hommage an die britische Komikertruppe Monty Python
Typ:	Interpretierte Programmiersprache, die Elemente von Skriptsprachen enthält. Objektorientierte Programmierung ist möglich.
Ziel der Erfindung:	Entwicklung einer Programmiersprache, die mächtig und auch leicht zu erlernen ist.
Ausführung von Programmen:	Speicherung als einfache Textdatei (.py als Dateinamenerweiterung), die von einem Compiler in Bytecode übersetzt und dann vom Python Interpreter ausgeführt wird.
Kompilierung:	unmittelbar vor Ausführung des Programms
Kosten:	keine, da Open Source
Aktuelle Version:	3.8.1 zum Download unter https://www.python.org/ (Stand 02/2020).
Vorteile:	Sehr flexibel, leicht auf andere Rechner übertragbar, da plattformunabhängig (läuft z. B. auf Windows, Linux und OS X ohne Quellcodeänderung), Erweiterbarkeit ist einfach.
Verbreitung:	Viele Biologen nutzen Python, um kleinere Programme selbst zu verfassen.

3.3.5 AUFGABE „DAS ERSTE PROGRAMM“

a) Schreiben Sie in Visual Studio Code gemeinsam mit Ihrem Lehrer Ihr erstes Programm

```
#!/usr/local/bin/python
#Das ist mein erstes Programm, welches „hallo Welt“ auf den Bildschirm schreibt


print („Hallo Welt“)
```

b) Speichern Sie ihr erstes Programm unter hallowelt.py ab.

c) Starte das Skript mit: python hallowelt.py

Und? Es antwortet: Hallo Welt!

ERKLÄRUNG

1. Mit der ersten Zeile *#!/usr/local/bin/python* wird der Python-Interpreter aufgerufen. Diese Zeile spielt auf Windowssystemen keine Rolle und wird ignoriert. In Windows wird der Interpreter direkt mit python … aufgerufen.
2. In der zweiten Zeile steht ein Kommentar. Ein Kommentar wird mit dem Zeichen # eingeleitet und gilt bis zum Zeilenende. Der Interpreter lässt Kommentare unbeachtet.
3. Mit dem Befehl *print* wird der Interpreter angewiesen, alles, was in der Klammer zwischen den Anführungszeichen steht, herauszugeben.

3.3.6 VARIABLEN IN PYTHON

1. WERTE IN VARIABLEN SPEICHERN

Variablen speichern Werte verschiedenster Art. Jede Variable besitzt einen einzigartigen Namen und einen Wert. Eine Variable kann nacheinander verschiedene Werte annehmen, speichert aber immer nur den Wert, der ihr als Letztes zugewiesen wurde. Technisch gesehen, stellt eine Variable eine Speicheradresse dar, die den Speicherplatz mit einem bestimmten Inhalt belegt.
Wir lernen in Python 5 Datentypen kennen, die in Variablen gespeichert werden können. Bei der Wertzuweisung steht die Variable immer links. Es folgt das Gleichheitszeichen und dann der zugewiesene Wert.

```
zahl = 5

# Ab dieser Zeile hat die Variable Zahl den Wert 5, bis dieser Wert überschrieben wird.
# Die Variablendefinition bzw. –einführung nennt man auch Variablendeklaration.

#Es ist sind auch folgende Zuweisungen möglich:
zahl = 5
Zahl = Zahl + 1
# Jetzt beinhaltet die Variable Zahl den Wert 6.
```

2. DATENTYPEN

TABELLE 2: DIE WESENTLICHEN DATENTYPEN IN PYTHON

Variablentyp	Beispiel für Zuweisung einer Variablen	Datentyp
String (Zeichenketten)	String=„Hallo Welt“	Zahl, Zeichen oder Zeichenkette (=string), mit der nicht gerechnet werden kann.
Integer (Ganze Zahlen)	X= 5	Ganze Zahl, mit der gerechnet werden kann.
Float (Gleitkommazahl)	X=5.3	Gleitkommazahl, mit der gerechnet werden kann. **Als Trennzeichen gilt der Punkt!!**
Listen	Liste1=[2,4,7,A]	Die Liste kann Strings oder Zahlen beinhalten. Listen können verschiedene Datentypen enthalten, auch Listen!
Dictionaries	d={„Schlüssel1“ : „Wert1“, „Schlüssel2“ : „Wert2“}	Zuordnung wie in einer Datenbanktabelle: jedem Schlüssel ist ein Wert zugeordnet. Ein Anwendungsbeispiel ist: jedes Codon entspricht einem Schlüssel, als Werte sind die entsprechenden Aminosäuren zugeordnet.

3. BENENNUNG VON VARIABLEN

Variablennamen müssen aus diesen Zeichen bestehen:

Groß- und Kleinbuchstaben von a bis Z, Unterstriche und Ziffern

Variablen dürfen nicht mit einer Ziffer beginnen. Sie müssen mit einem Buchstaben oder dem Unterstrich beginnen.

3.3.7 DATEN VON DER STANDARDEINGABE (=TASTATUR) EINLESEN

Als Standardeingabemöglichkeit ist die Eingabe über die Tastatur eingerichtet.

Dies ist für Zeichenketten über *input()* möglich. Seit Python 3 ist für Zahlen *eval(input())* notwendig, damit ausgewertet wird, ob die Eingabe eine Zahl oder eine Zeichenkette ist. Anstatt *eval* kann auch direkt der gewünschte Variablentyp angegeben werden, z. B. *float(input())*.

Beispielskript:

#Der Text in den Klammern wird ausgegeben. Dann wird so lange der Programmablauf gestoppt, bis der Nutzer seine Eingabe mit der Return Taste abgeschlossen hat. Das eingegebene wird in der angegebenen Variablen (hier „Name“) abgelegt.

Name = input(„Geben Sie bitte Ihren Namen ein! „)

#Es folgt die Ausgabe des eingegebenen Strings. Variablen werden außerhalb der Anführungszeichen geschrieben und durch ein Komma von weiteren Ausgaben getrennt.

print („Oh, ich bin nicht allein! Hallo“,Name,“!“)

Die Tastatureingabe muss mit dem Tippen der Return-Taste abgeschlossen werden, dann erst läuft das Skript weiter und gibt bei einer Eingabe von „Klaus“ Folgendes aus:

Oh, ich bin nicht allein! Hallo Klaus!

AUFGABE 1:

Führen Sie das obige Skript auf dem Computer aus.

AUFGABE 2:

Schreiben Sie ein Programm Restriktion.pl, das einen Namen für ein Restriktionsenzym (z. B. HindIII) und die entsprechende Erkennungssequenz (AAGCTT) in je einer Variablen speichert. Dies soll über die Zuweisung einer Eingabe aus der Kommandozeile geschehen. Die Ausgabe auf die Kommandozeile soll so aussehen:

Name des Restriktionsenzyms: HindIII

Erkennungssequenz für HindIII: AAGCTT

AUFGABE 3:

Schreiben Sie ein Programm, das zunächst zwei Summanden vom Nutzer abfragt und dann die Rechnung und die Summe am Bildschirm ausgibt. (Achtung, nutzen Sie *eval(input())* für die Eingabe!

3.3.8 DATEN AUS EINER DATEI LESEN

Um eine Datei lesen zu können, muss man diese öffnen. Dabei wird ihr ein FileObject (beliebiger, frei gewählter Name, üblicherweise in Großbuchstaben) zugeordnet, der für die weiteren Zugriffe verwendet wird. Beim Öffnen der Datei wird bereits angegeben, ob die Datei gelesen oder beschrieben werden soll. Zum Lesen muss die Datei bereits existieren, sonst gibt es eine Fehlermeldung.

Zum Öffnen einer Datei dient der Befehl: (alternativ zum Dateinamen kann auch der gesamte Dateipfad angegeben werden).

```
fobj=open („Dateiname.txt“, „r“)
```

BEISPIEL

Es existiert eine Datei **name.txt** mit Textinhalt. Folgendes Skript öffnet die Datei und liest den Inhalt und gibt diesen am Bildschirm wieder:

```
#!usr/bin/env python

# Lesen der Datei name.txt
# Zuordnung des Inhalts von name.txt über den FileObject fobj
fobj=open („name.txt“, “r“)
# for Schleife zur zeilenweisen Ausgabe der Datei auf dem Bildschirm
for line in fobj:
    #line.strip zum entfernen von newLine zeichen (\n) am Zeilenende
    line =line.strip()
    print (line)
fobj.close()
```

AUFGABE

Führen Sie das obige Skript auf dem Computer aus. Legen Sie dazu zuerst die Datei name.txt an (mit einem Editor wie Visual Studio Code) und füllen Sie diese Datei mit Text. Die Datei name.txt sollte im selben Verzeichnis wie das Skript liegen.

3.3.9 DATEN IN EINE DATEI SCHREIBEN

Möchte man in eine Datei schreiben, sind folgende drei open-Befehle möglich:

TABELLE 3: OPEN-BEFEHL IN PYTHON

open („dateiname.txt“,“r“)	Zum Lesen öffnen
open („dateiname.txt“,“w“)	Zum Schreiben öffnen (Datei wird zunächst geleert und dann neu beschrieben)
open („dateiname.txt“,“a“)	Zum Schreiben öffnen. Eine bestehende Datei wird nicht überschrieben, sondern erweitert
(Liste unvollständig, für uns aber ausreichend)	

Zum Schreiben in einer Datei muss also zunächst ein open-Befehl und dann ein print-Befehl erfolgen.

Folgendes Beispiel schreibt den Text aus der Datei name.txt in eine schreiben.txt:

```
fobj = open („nameSchreiben.txt“,“w“)
ausgabe= „Klaus“
# die gelesenen Daten in die neue Datei schreiben
fobj.write (ausgabe)
fobj.close()
```

AUFGABE:

Schreiben Sie ein Programm, das den Inhalt einer Datei liest. Im Anschluss soll eine Datei erzeugt werden, die sowohl den Inhalt der gelesenen Datei als auch eine Zeile mit einem Text enthält, der während des Programmablaufs vom Benutzer eingegeben werden kann.

3.3.10 MUSTERERKENNUNGSFUNKTIONEN IN PYTHON

DIE FUNKTION *TRANSLATE* IN PYTHON

In der Bioinformatik ist es oft hilfreich, wenn der komplementäre DNA-Strang oder auch ein RNA-Strang zu einer gegebenen DNA-Sequenz erstellt werden kann. Dazu ist translate () aus Python geeignet:

#Ersetzen von Zeichen mit translate ()

```
codingsequence=„ATGCTGTGCAAATCC“
print(codingsequence)
codogenSeqence=codingsequence
table=codogenSeqence.maketrans(„ACGT“,“TGCA“)
codogenSeqence=codogenSeqence.translate(table)
print(codogenSeqence)
```

DIE FUNKTION REVERSE IN PYTHON

Um einen String in umgekehrter Reihenfolge auszugeben, kann man sich zu diesem Zweck beispielsweise eine Funktion reverse() erstellen und die Umkehrung durchführen lassen.

```
#String rückwärts ausgeben
# Function to reverse a string
def reverse(string):
    string = “„.join(reversed(string))
    return string
 Vorwaerts=„ABCDEFG“
 print („The original string is:“, Vorwaerts)
Rueckwaerts=reverse(Vorwaerts)
print („The reversed string(using reversed) is :“, Rueckwaerts)
```

SUCHEN VON TEILSTRINGS

Um bestimmte Codons wie, z. B. ATG in einer DNA-Sequenz zu finden, ist *find()* hilfreich.

#Teilstring mit find (Suchstring,Start,Stop) suchen. Start und Stop geben die indices im Suchstring an,
#zwischen denen gesucht werden soll.

```
Teststring=„AAACGTATGCATGGTCA“
Suchstring=„ATG“
Ergebnis=Teststring.find(Suchstring)+1
print(„Der gesuchte String“,Suchstring, “ist an Position“, Ergebnis,“ zu finden“)
```

LÄNGE VON STRINGS AUSLESEN

Die Länge eines Strings kann mit der Funktion *len()* gesucht werden.

```
Teststring=„AAACGTATGCATGGTCA“
Laenge=len(Teststring)
print („Die Laenge des Strings“,Teststring „ist“,Laenge,“.“)
```

AUFGABE:

Schreiben Sie ein Programm, dem Sie zwei DANN-Sequenzen (codierender Strang) eingeben können. Mit den Sequenzen soll das Programm dann folgende Aufgaben durchführen:

Die Länge der DNA-Sequenzen ausgeben.

Die Länge der beiden Teilsequenzen addieren und die Gesamtlänge ausgeben.

Die beiden Sequenzen aneinanderhängen.

Die Gesamtsequenz ausgeben.

Die komplementäre Sequenz ausgeben.

Die reverse komplementäre Sequenz ausgeben.

Die RNA-Sequenz ausgeben.

Die ersten 3 und die letzten 3 Basen der Sequenz ausgeben.

Die Position des ersten ATG in der Sequenz finden.

Musterlösung: 06_Variablenmanipulation

3.3.11 LISTEN IN PYTHON:

Bisher haben wir alle Programme mit skalaren Variablen ausgeführt. Viele Daten der Bioinformatik können aber auch als Listen, statt als Einzelwerte angesehen werden. Beispielsweise kann eine Sequenz von 7 Nukleotiden in einer Variablen X gespeichert werden, aber auch als Liste von 7 einzelnen Nukleotiden, z. B. [A,C,G,G,G,T,T] oder auch als eine Liste von Zahlen.

Eine Liste deklariert man analog zu einer skalaren Variablen; die Listenelemente werden dabei in eckige Klammern geschrieben.

INITIALE WERTE KANN MAN WIE FOLGT ZUWEISEN:

Sequenz=['A','C','G','G','G','T','T']

Die Werte der zugewiesenen Liste stehen in eckigen Klammern und werden durch Kommata separiert. Zahlen und Strings können gemischt werden.

Alternativ kann eine Liste auch mithilfe einer for-Anweisung gefüllt werden, z. B. das Erstellen einer Liste mit Quadratzahlen:

*Quadratzahlen=[i*i for i in range(10)]*

Oder aus einem String erstellen:

Mylist=list(Mystring)

ZUGRIFF AUF EIN LISTENELEMENT:

```
Sequenz=['A','C','G','G','G','T','T']
#Zugriff auf das erste Element:
Firstelement= Sequenz[0]
#Zugriff auf das zweite Element:
secondElement= Sequenz[1]
```

Beachten Sie, dass das Erste Element den Index 0 und nicht 1 besitzt!!

```
# Zugriff auf das letzte Element
lastElement=Sequenz[-1]
```

AUSGABE DER LISTE AUF DEM BILDSCHIRM:

```
print (Sequenz)
```

EXTRAHIEREN EINER TEILLISTE:

```
Teilliste= Sequenz[2:5]
Teillistebis5= Sequenz[:5]
print(Teilliste)
```

ELEMENT MIT INDEX X AUS LISTE ENTFERNEN;

ENTSPRECHEND WÜRDE AUCH FUNKTIONIEREN DIE INDICES ZWISCHEN X UND Y ZU LÖSCHEN: DEL LISTE[X:Y]

```
x=5
del Liste[x]   #Eckige Klammern!
print (liste)   #Die Liste ist nun um das Element mit Index x kürzer
```

ANHÄNGEN EINES ELEMENTS AN DAS ENDE DER LISTE

```
x=5
liste.append(x)          #am Ende der Liste liste wurde nun eine 5 angehängt. Das würde auch mit Buchstaben funktionieren.
```

ANHÄNGEN EINER ZWEITEN LISTE AN EINE BESTEHENDE LISTE

```
liste=[1,2,3,4,5,6,7,8,9]
langeListe=liste
langeListe.extend(liste)          #die lange Liste enthält nun eine vorhandene Liste zweimal
```

ENTFERNEN VON ELEMENTEN AUS DER LISTE. WIRD KEIN ARGUMENT ÜBERGEBEN, DANN WIRD DAS LETZTE ELEMENT ENTFERNT

```
liste=[1,2,3,4,5,6,7,8,9]
liste.pop()          #liste enthält jetzt nur noch 1,2,3,4,5,6,7,8
```

ENTFERNEN BESTIMMTER WERTE AUS EINER LISTE

```
liste=[4,1,2,3,4,5,6,7,8,9]
liste.remove(4)          #liste ohne den ersten enthaltenen 4ern
```

LISTE UMKEHREN

```
liste =[1,2,3,4,5,6,7,8,9,10]
print („Die Liste“,liste)
liste.reverse()
print („Umgekehrte Liste: “,liste)
```

LISTE SORTIEREN

```
liste=[10,9,8,7,5,4,4,3,5,6,2,1]
liste.sort()
print(„und wieder sortiert“,liste)
```

LÄNGE EINER LISTE AUSLESEN

```
Listenlaenge=len(liste)
```

AUFGABE:

Schreiben Sie ein Programm, das Folgendes leistet:
Der Programmnutzer soll eine DNA-Sequenz beliebiger Länge über die Eingabeaufforderung eingeben können.
Die Sequenz soll in einer Datei namens Sequenz.txt gespeichert werden.
Der Inhalt der entsprechenden Variablen soll in einer Liste gespeichert werden.
Die Sequenzlänge soll über die Ermittlung der Listenlänge erfolgen und auf dem Bildschirm ausgegeben werden.
Auf dem Bildschirm soll Folgendes ausgegeben werden:

Sequenzlänge
das erste Zeichen der Sequenz
das letzte Zeichen der Sequenz
die komplette Sequenz als kontinuierliche Sequenz

Musterlösung: 07_ListenInPythonMusterlsg.py

3.3.12 DICTIONARIES IN PYTHON

In einem solchen Dictionary ist wie in einer Datenbank einem bestimmten Schlüssel ein bestimmter Wert zugeordnet.
Die Reihenfolge innerhalb der Dictionaries ist nicht von Bedeutung (anders als z. B. in Listen)!

Beispiele für Anwendungen:

Name des Restriktionsenzyms und Erkennungssequenz:

Schlüssel	Wert
Acc16I	TGCGCA
BamHI	GGATCC
BciVI	GTATCC
BmrI	ACTGGG

Oder auch: Zuordnung von Codon zu Aminosäuren:

UUU	Phe
UUC	Phe
UUA	Leu
…	

Ein leeres Dictionary kann wie folgt erzeugt werden:

```
Enzyme={}
```

Soll das Dictionary direkt Daten enthalten, funktioniert das wie folgt:

```
Enzyme={„Acc16I“: „TGCGCA“, „ BamHI“: „GGATCC“}
```

Zugriff auf einem Wert zu einem bestimmten Schlüssel (Achtung: hier werden eckige Klammern genutzt):

```
ValueZuAcc16I=Enzyme[„Acc16I“]
```

Löschen eines Schlüssel-Wertepaares:

```
del Enzyme[„Acc16I“]
#Testen, ob ein Eintrag im Dictionary enthalten ist
Tester=„Acc16I“ in Enzyme
print (Tester)
#Neuen Eintrag in ein Dictionary erzeugen (oder falls schon vorhanden den alten
überschreiben) und direkt auf dem Bildschirm ausgeben:
Enzyme[„Acc16I“]=„TGCGCA“
print (Enzyme[„Acc16I“])
#Abfragen aller keys des Dictionarys mit der Methode keys
```

```
SchluesselDesDictionarys=Enzyme.keys()
print(SchluesselDesDictionarys)
#Ebenso geht das mit den values oder auch den Wertepaaren, die im folgenden Beispiel direkt am Bildschirm ausgegeben werden:

print(„Die Werte im Dictionary sind aktuell: „,Enzyme.values())
print(„Die Wertepaare im Dictionary sind derzeit: „,Enzyme.items())
#Prüfen, ob ein Schlüssel im Dictionary vorhanden ist
EnzymVorhanden=input(„Geben Sie einen Enzymnamen ein: „)
    if EnzymVorhanden in Enzyme:
    print(„Die Erkennungssequenz von „,EnzymVorhanden,“ist“,Enzyme[EnzymVorhanden],“.“)
else:
    print(„Dieses Enzym ist noch nicht im Dictionary“)
```

ÜBUNG:

Schreiben Sie ein Programm, das ein Dictionary mit allen Codons und den zugehörigen Aminosäuren enthält.
Das Dictionary darf zunächst unvollständig sein.
Das Programm soll den Nutzer nach einem Codon fragen.
Ist dieser im Dictionary soll die passende Aminosäure ausgegeben werden.
Ist das Codon nicht enthalten, soll zusätzlich die passende Aminosäure erfragt werden, das Dictionary ergänzt und am Bildschirm (und Zusatzübung: in einer Datei) ausgegeben werden.

Musterlösung: 09_DictionariesMusterlsg.py

3.3.13 IF... ELSE BEDINGUNG

```
if bedingung1:
    anweisungen1
elif bedindung2:
    anweisungen2
else:
    anweisungen3
```

Beachten Sie, dass der Anweisungsblock immer eingerückt sein muss. Dieser kann aus einem Befehl oder auch aus mehreren bestehen.
Mit elif können weitere Bedingungen mit den passenden Anweisungen eingebracht werden.
Else steht als letzte Bedingung und fängt alle nicht vorher abgefangen Bedingungen ab.
Beispiel:

```
if Abfrage == „Acc16I“:
        print „Die Erkennungssequenz fuer Abfrage ist Enzyme[„Acc16I“]“
# Wenn der Wert von Abfrage gleich der Zeichenkette „Acc16I“
# ist, dann gibt der Test den Wert „false“ zurueck und der zweite Block wird
# ausgefuehrt.
else:
        print („Keine Daten vorhanden fuer das Enzym“, Abfrage)
```

Die Bedingung wird direkt nach dem *if* eingefügt und mit einem Doppelpunkt beendet. Mögliche Bedingungen sind unten angegeben.
Ist sie wahr (true), so wird die Befehlsfolge innerhalb der folgenden eingerückten Befehlsfolge ausgeführt und die Befehlsfolge nach dem else übersprungen.
Ist sie unwahr (false), so wird nur die Befehlsfolge, die nach dem else: eingerückt ist, ausgeführt.

Bedingungen können wie folgt lauten:
If x == 'Hallo': ist wahr, wenn x die Zeichenkette Hallo enthält
If x ==0: ist wahr, wenn x die Zahl 0 enthält
If x <0: ist wahr, wenn x die eine Zahl kleiner als 0 enthält
If x >0: ist wahr, wenn x die eine Zahl größer als 0 enthält
If x <=0: ist wahr, wenn x die eine Zahl kleiner oder gleich 0 enthält
If x >=0: ist wahr, wenn x die eine Zahl größer oder gleich 0 enthält
If x !=0: ist wahr, wenn x die eine Zahl ungleich 0 enthält

```
if “Testenzym“ in Enzyme:
   print(„Die Erkennungssequenz von “,Testenzym,“ist“,Enzyme[EnzymVorhanden],“.“)
else:
```

print(„Dieses Enzym ist noch nicht im Dictionary“)

Es ist auch möglich, weitere Bedingungen einzufügen:

```
if Abfrage == 'Acc16I':
  print („Die Erkennungssequenz fuer Abfrage ist Enzyme[„Acc16I „)
# Wenn der Wert von $Abfrage gleich der Zeichenkette 'Acc16I'
# ist, dann gibt der Test den Wert „false“ zurueck und der zweite Block wird ausge-
fuehrt.
Elif Abfrage == 'Restriktionsenzym2':
  print („Die Erkennungssequenz fuer“, Abfrage,“ist“, Enzyme[Restriktionsenzym2]“)
else:
   print („Keine Daten vorhanden fuer das Enzym „, Abfrage,“)
exit;
}
```

AUFGABEN:

Erzeugen Sie ein Dictionary mit einer Liste von Codons.
Das Programm soll Folgendes leisten:
Es fragt den Nutzer nach der Eingabe eines Codons.
Ist dieses in der Liste, so soll der Name der zugehörigen Aminosäure ausgegeben werden.
Falls nicht, so soll das Programm fragen, ob ein neuer Eintrag erfolgen soll.
Falls ja, so wird das Dictionary um das neue Schlüssel-/Wertepaar erweitert.

Musterlösung: 10_BedingungenInPython.py

3.3.14 SCHLEIFEN IN PYTHON

1. WHILE-SCHLEIFE

Schleifen dienen dazu, eine bestimmte Befehlsfolge oft zu wiederholen. In Python gibt es zwei Typen von Schleifen:

1. Die While-Schleife, deren eingerückte Befehlsfolge so lange durchgeführt wird, bis die angegebene Bedingung nicht mehr erfüllt ist:

```
#While Schleife
#initialisieren der Zählvariable, (häufig nimmt man i)
i=1
Summe=0
#solange die Bedingung hinter der while-funktion wahr ist wird die eingerückte Befehlsfolge #immer wieder durchlaufen
#danach wird zur nicht eingerückten Befehlsfolge gesprungen.
while i<=100:
    Summe=Summe+i
    i=i+1
print („Die Summe der Zahlen von 1-100 ist“, Summe)
# mit break kann die Schleife direkt verlassen werden.
# mit continue wird die aktuelle Befehlsfolge verlassen und am Beginn der Schleife geprüft, #ob die Bedingung für die Schleife noch erfüllt ist.
# beide Bedingungen ergeben insbesondere in Kombination mit Bedingungen (if) Sinn.

# die folgende Schleife macht im Prinzip das gleiche wie die obere, lässt aber die #eingegebene Zahl aus.
AusgelasseneZahl =eval(input(„Welche Zahl soll ausgelassen werden?“))
j=0
Summe2=0
while j<=100:
    #wenn die auszulassende Zahl gleich der Zählvariablen j ist, dann wird hier dafür gesorgt, dass j nicht zur Summe2 dazugezählt wird
    if j==AusgelasseneZahl:
        print („folgende Zahl wird ausgelassen:“,AusgelasseneZahl)
        j=j+1
 #continue springt zum Schleifenanfang
        continue
#wenn j ungleich der Ausgelassenen Zahl ist, wird dieser Teil der Whileschleife ausgeführt       #und j zur Summe2 dazugezählt
        Summe2=Summe2+j
```

```
        j=j+1
print („Die Summe der Zahlen von 1-100 ohne“,AusgelasseneZahl,“ist“,Summe2)
```

2. FOR-SCHLEIFE

Oft muss in einem Programm eine bestimmte Befehlsfolge für z. B. alle Elemente einer Liste gleichermaßen durchgeführt werden. Ein Beispiel wäre eine Liste von Codons, deren zugehörige Aminosäure gefunden werden soll.

Eine for-Schleife geht alle Elemente z. B. einer Liste oder eines Dictionaries oder die Buchstaben eines Strings durch und wird nach Erreichen des Listenendes verlassen.

Mit if- kann für einzelne Elemente eine andere Befehlsfolge angegeben werden.

Beispiel:

```
#Definition eines Dictionary
codon={
„AAA“:“Lys“,
„AAC“:“Asn“,
„AAG“:“Lys“,
„ATG“:“Met“
}
#Ausgabe der Keys des Dictionarys; wenn ein startcodon gefunden wird, wird ein
#entsprechender Hinweis ausgegeben.
#Die eingerückten Bereiche enthalten die in der For-schleife abgehandelte
#Befehlsfolge

for code in codon:
        print(code)
        if code==„ATG“:
                print(code,“ist das Startcodon“)
```

AUFGABEN:

Schreiben Sie ein Programm, das ein Dictionary mit allen Codons enthält. Es sollen alle Codons ausgegeben werden (nutzen Sie die For-Schleife).

Wenn ein Start oder ein Stopcodon ausgegeben wird, soll dies entsprechend vermerkt werden (nutzen Sie If).

Zusatzaufgabe: Erstellen Sie ein Programm, welches das Dictionary in eine Datei schreibt bzw. dieses aus der Datei ausliest. Wenn Sie Befehle vermissen, schlagen Sie online nach. Z. B. da: https://www.python-kurs.eu/.

Musterlösung: 11_SchleifenInPython_Musterlsg.py

3.3.15 FUNKTIONEN IN PYTHON

Funktionen definieren: Wenn eine bestimmte Befehlsfolge öfter in einem Programm benötigt wird, kann man sich selbst eine passende Funktion schreiben und die dann an den benötigten Stellen einfach wieder aufrufen, ohne den gesamten Code neu zu schreiben.

Als Beispiel dient hier die Bestimmung des GC-Gehaltes einer DNA-Sequenz und eine Funktion, die dafür sorgt, dass ein als Dezimaltrennzeichen eingegebenes Komma als von Python geforderter Punkt ausgegeben wird.

Die Definition startet mit *def*, gefolgt vom Namen der Funktion und in Klammern den zu übergebenden Parametern und einem Doppelpunkt. Der Code der Funktion erfolgt eingerückt. Der Rückgabewert erfolgt mit *return*.

```
# Definieren der Funktionen mit def:
def Berechne_GC(DNA_Sequenz):
    DNA_Laenge=len(DNA_Sequenz)
    G_in_DNA=DNA_Sequenz.count('G')
    C_in_DNA=DNA_Sequenz.count('C')
    GC_Gehalt=(G_in_DNA+C_in_DNA)/DNA_Laenge
    return GC_Gehalt

def KommaErsetzen(Kommazahl):
    text=„„
    for i in (Kommazahl) :
        if i == ',':
            text = text + '.'
        else:
            text = text + i
    return float(text)

# Hier beginnt das eigentliche Programm mit
MySequence= input(„Geben Sie die DNA-Sequenz ein, deren GC-Gehalt berechnet
werden soll: “)
#Die definierte Funktion Berechne_GC wird hier innerhalb eines Printbefehls aufgerufen
print(„Der GC-Gehalt der eingegebenen Sequenz ist: “, Berechne_GC(MySequence))

Kommazahl= input(„geben Sie eine Kommazahl ein: “)
#Ausgabe der Kommazahl mit Punkt:
Kommazahl=KommaErsetzen(Kommazahl)
```

print(Kommazahl)

AUFGABEN

Erstellen Sie ein Programm, welches mithilfe einer selbst geschriebenen Funktion eine Fasta-Datei einliest und jeweils am Bildschirm ausgibt. Die Namen der Dateien sollen vom Benutzer eingegeben werden können.

Musterlösung: 12_Funktionen_Musterlsg.py

3.3.16 DIE SCHÜLERINNEN UND SCHÜLER ENTWICKELN, DOKUMENTIEREN, KOMMENTIEREN UND IMPLEMENTIEREN EINEN ALGORITHMUS ZUR ORF-ANALYSE ALS BIOINFORMATISCHES WERKZEUG DER SEQUENZANALYSE. ARBEITSMATERIALIEN/AUFGABEN

3.3.17 PROJEKT ORF FINDER

In der Bioinformatik wird ein ORF als die Sequenz vom Startcodon bis zum Stopcodon eines Gens definiert. Der Bereich dazwischen enthält keine Stoppcodons und ist in der Regel relativ lang (siehe auch Informationen in der Datei „Stichwort ORF.docx“). Nach einer Sequenzierung ist es oft nicht klar, ob das initiale Startcodon im sequenzierten Bereich enthalten ist. Aus diesem Grund kann es sinnvoll sein, nach langen Sequenzen ohne Stopcodon zu suchen. Dabei muss beachtet werden, dass man ohne das Startcodon nicht weiß, in welchem Leserahmen (Reading-Frame) man sucht. Man sucht dann in allen sechs Reading-Frames nach langen Codon-Sequenzen ohne Stopcodon. Ist man im falschen Leserahmen, folgt im Mittel nach 20 Aminosäuren ein Stoppcodon.

AUFGABE 1

Begründen Sie, warum bei einer zufälligen Sequenz, die ja in den falschen Reading-Frames vorliegt, im Mittel nach 20 Codons ein Stopcodon folgen sollte.

AUFGABE 2:

Welche sechs Reading-Frames sind gemeint?

AUFGABE 3 (PROJEKT):

Schreiben Sie ein Programm, welches das längste, nicht durch ein Stopcodon unterbrochene Stück DNA, dessen Länge und die zugehörige Aminosäuresequenz ausgibt. Nutzen Sie hierzu die vorherigen Arbeitsergebnisse und Onlineangebote wie z. B. https://www.w3schools.com/python/default.asp oder https://www.python-kurs.eu/ oder Erweiterungen von Python, wie z. B. Biopython (https://biopython.org/).

Musterlösung (Aufgabe 1–3): 13_Projekt_Orf_Musterlsg.py

4 Umsetzungsbeispiele für Vertiefung – individualisiertes Lernen – Projektunterricht (VIP)

4.1 BPE 7

Teil 5 - Die Genomorganisation des HIV

In: Handreichungen zur Bioinformatik, Handreichungen des Landesinstituts für Schulentwicklung (LS)

https://www.schule-bw.de/faecher-und-schularten/berufliche-bildung/bioinformatik/handreichungen_bioinf

5 Anhang

5.1 BPE 4.3 Alphabet als Sprach- und Textcode – Cäsarcode

https://www.sicherheit-macht-schule.de/Unterrichtsideen/Klasse_7_10/1262_Verschluesselung.htm
https://lehrerfortbildung-bw.de/u_matnatech/informatik/gym/bp2016/fb1/3_rechner_netze/2_kopier/5_caesar/

5.2 Handreichungen zur Bioinformatik auf dem Landesbildungsserver

https://www.schule-bw.de/faecher-und-schularten/berufliche-bildung/bioinformatik/handreichungen_bioinf