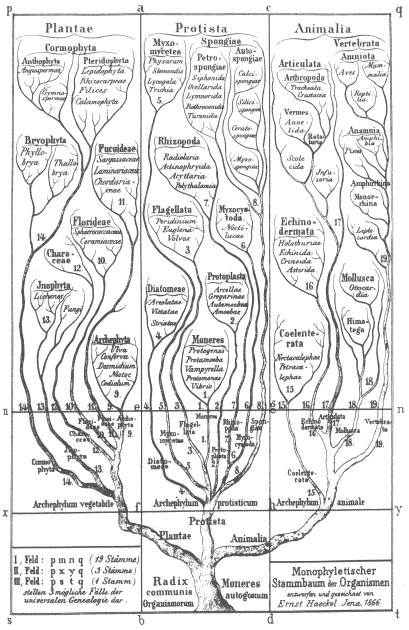
Institut für Genetik der
Universität zu Köln

Parallelisierung phylogenetischer Methoden zur Untersuchung der Crown Group Radiation



Diplomarbeit
vorgelegt von
Heiko A. Schmidt
aus Bonn

Köln, im Mai 1996

meiner Mutter

Alle wissenschaftlichen Erkenntnisse haben und werden auch weiterhin irgendwo ihre Grenzen finden, denn soweit wir auch gehen, der Horizont bleibt immer gleich weit entfernt.

Immanuel Kant

Die vorliegende Arbeit wurde im Zeitraum von Mai 1995 bis Mai 1996 am Institut für Informatik der Universität zu Köln durchgeführt.

Hiermit versichere ich, daß die hier vorgelegte Arbeit von mir selbständig ausgeführt und verfaßt wurde. Es wurden keine anderen Quellen und Hilfsmittel außer den angegebenen verwendet.

Köln, den 8. Mai 1996

Vorwort

Es handelt sich bei der hier vorliegenden Arbeit um eine interdisziplinäre Arbeit zwischen den Fachbereichen Biologie, speziell Genetik und Evolutionsbiologie, und Informatik. Da sich diese Arbeit an Leser aus beiden Bereichen wendet, kann es sein, daß im Folgenden Dinge beschrieben sind, die entweder Biologen oder Informatikern als selbstverständlich bekannt sind. Um dem jeweils anderen Teil der Leserschaft ein verständliches Bild zu ermöglichen, wurden solche „Selbstverständlichkeiten“ teilweise dennoch beschrieben.

Um einige Fragen, die bei der Lektüre auftreten können, leichter klären zu können, ist der Arbeit im Anhang sowohl ein kleines Glossar als auch eine Liste mit den gebrauchten Abkürzungen nachgestellt.

Bei der Untersuchung der Entwicklung der Organismen sollte man immer einen Satz von Ernst Haeckel[1] aus dem Jahre 1894 bedenken:

> „Natürlich bleiben aber auch diese Schemata (‚Stammbäume‘) ... immer nur Versuche, tiefer in die Geheimnisse der Stammesgeschichte einzudringen; sie sollen nur den Weg andeuten, auf welchem nach dem jetzigen beschränkten Zustand unserer Kenntnisse die weitere phylogenetische Forschung am besten vorzudringen hat. Ich brauche daher hier wohl nicht zu wiederholen, daß ich meinen Entwürfen von Stammbäumen und systematischen Tabellen keinen dogmatischen Wert beimesse“.

Es darf bei der Rekonstruktion phylogenetischer Stammbäume nicht vergessen werden, daß unsere Analysen auf gewissen Annahmen und Vereinfachungen beruhen, die nicht unbedingt mit der tatsächlichen Entwicklung übereinstimmen müssen. Somit ist alles, was wir rekonstruieren, auch nur ein Versuch, das Vergangene anhand von Indizien zu verstehen.

Teile dieser Arbeit sollen in dem Buch *The Origin of Algae and their Plastids* (D. Bhattacharya, Hrsg.) im Kapitel „Phylogeny of the Glaucocystophyta “(D. Bhattacharya und H.A. Schmidt) beim Springer–Verlag in Wien veröffentlicht werden (voraussichtliches Erscheinungsdatum Anfang 1997).

[1] nach Weberling und Stützel (1993)

Inhaltsverzeichnis

1 Einleitung **1**

1.1 Allgemeines 1
1.2 Datenmaterial für phylogenetische Untersuchungen 3
1.3 Erbgut und Evolution 5
1.4 Phylogenie 8
1.4.1 Phylogenetische Stammbäume 8
1.4.2 Evolutionsmodelle 11
1.4.3 Phylogenetische Methoden 12
1.5 Parallele Computerplattformen und Parallelrechnen 15
1.6 Stammbaum des Lebens, Crown Group Radiation und Plastidenentstehung 16
1.7 Mutationsraten und Molekulare Uhren 19
1.8 Ribosomale RNA und Stammbaumrekonstruktion 21
1.9 Entwicklung der Aktingene 22

2 Zielsetzung **24**

3 Material und Geräte **26**

3.1 Datenmaterial 26
3.1.1 Datensätze zur Laufzeituntersuchung 26
3.1.2 Aktinsequenzen 27
3.1.3 Eukaryotische 18S rRNA–Sequenzen 29
3.1.4 16S rRNA–Sequenzen aus Plastiden/Bakterien 31
3.2 Benutzte Software 32
3.2.1 fastDNAml 32
3.2.2 PHYLIP 33
3.2.3 treetool 33
3.2.4 MacClade 33
3.2.5 PARMACS 34
3.3 Geräte 34
3.3.1 Workstations und Workstationcluster 34

3.3.2 Parallelrechner 34

4 Methoden, Modelle und Algorithmen 35
4.1 Modellierung evolutionärer Prozesse 35
4.1.1 Markov–Prozesse 35
4.1.2 Generalisiertes 2–Parameter–Modell 36
4.2 Die Maximum–Likelihood–Methode zur Rekonstruktion von Stammbäumen 37
4.2.1 Berechnung des Likelihood–Wertes eines Baumes . . . 37
4.2.2 Das Pulley–Prinzip 40
4.2.3 Finden der optimalen Kantenlängen 40
4.2.4 Suche nach dem optimalen Baum 41
4.3 Erstellen von gewichteten Datensätzen 42
4.4 Bootstrap–Verfahren 43

5 Parallelisierung und Laufzeituntersuchung 45
5.1 Ergebnisse 45
5.1.1 Implementierung 45
5.1.2 Untersuchungen des Laufzeitverhaltens 54
5.2 Diskussion 56
5.2.1 Speedup und Skalierbarkeit 56
5.2.2 Parallelisierungskonzept 59

6 Untersuchung der Crown Group Radiation 60
6.1 Ergebnisse 60
6.1.1 Aktinentwicklung 60
6.1.2 Plastidenentwicklung 64
6.1.3 Eukaryotenentwicklung 74
6.2 Diskussion 76
6.2.1 Bewertung von Bootstrap–Analysen 76
6.2.2 Gewichtung von Alignments 79
6.2.3 Maximum–Likelihood–Methode 82
6.2.4 Entwicklung der Aktingene 83
6.2.5 Plastidenentwicklung 84
6.2.6 Entwicklung der plastidentragenden Eukaryoten 86

7 Zusammenfassung und Ausblick 88
7.1 Zusammenfassung 88
7.2 Ausblick 89

A pfastDNAml (Bedienungsanleitung und Beschreibung) 91
A.1 Allgemeines 91
A.2 Programmaufruf 91
A.3 Format der Eingabedaten 92
A.4 Ausgabedateien 97
A.5 Installation 99
A.6 Quellcodestruktur 101
A.7 Hilfsprogramme für verschiedene Parallelplattformen 102

B Glossar 104

C Abkürzungen 109

Literaturverzeichnis 111

Danksagung 119

Kapitel 1

Einleitung

1.1 Allgemeines

Biologische Forschung läßt sich in zwei verschiedene methodische Ansätze unterteilen. Der „funktionelle“ Ansatz beschäftigt sich mit der Frage, wie ein Organismus funktioniert und wie er sich entwickelt. Der historische Ansatz fragt dagegen, warum das Leben, wie wir es heute erleben und beobachten können, so ist, wie es ist, und wie es entstanden ist. (Futuyma, 1986)

Die Frage nach dem Ursprung des Lebens beschäftigt die Menschen seit jeher. Aber erst nachdem Darvin im 19. Jahrhundert seine Theorie der Evolution durch natürliche Selektion entwickelt hatte, war es möglich, sich diesem Thema wissenschaftlich zu nähern. (Watson *et al.*, 1992)

Die Evolutionsbiologie, und so auch diese Arbeit, baut auf der Annahme auf, daß ein evolutionärer Prozeß existiert und sich die Organismen phylogenetisch[1] entwickelt haben, d.h. in Form eines Stammbaums, ausgehend von einem gemeinsamen Vorfahren. Ernst Haeckel[2] drückte dies 1884 so aus:

> „*Entweder* giebt es eine Phylogenie, oder es giebt *keine!*
> Entweder entwickelt sich die organische Welt phylogenetisch oder *nicht!*
> Zwischen dieser Alternative giebt es keine *ehrliche* Vermittlung!“

Es ist bis heute nicht bewiesen, daß sich die Organismen, durch Evolution gesteuert, in Form eines Stammbaumes entwickelt haben. Viele Indizien

[1] $\varphi\tilde{\upsilon}\lambda o\nu$ (griech.) – Geschlecht; $\gamma\acute{\varepsilon}\nu\varepsilon\sigma\iota\varsigma$ (griech.) – Werden, Entstehen

[2] nach Weberling und Stützel (1993)

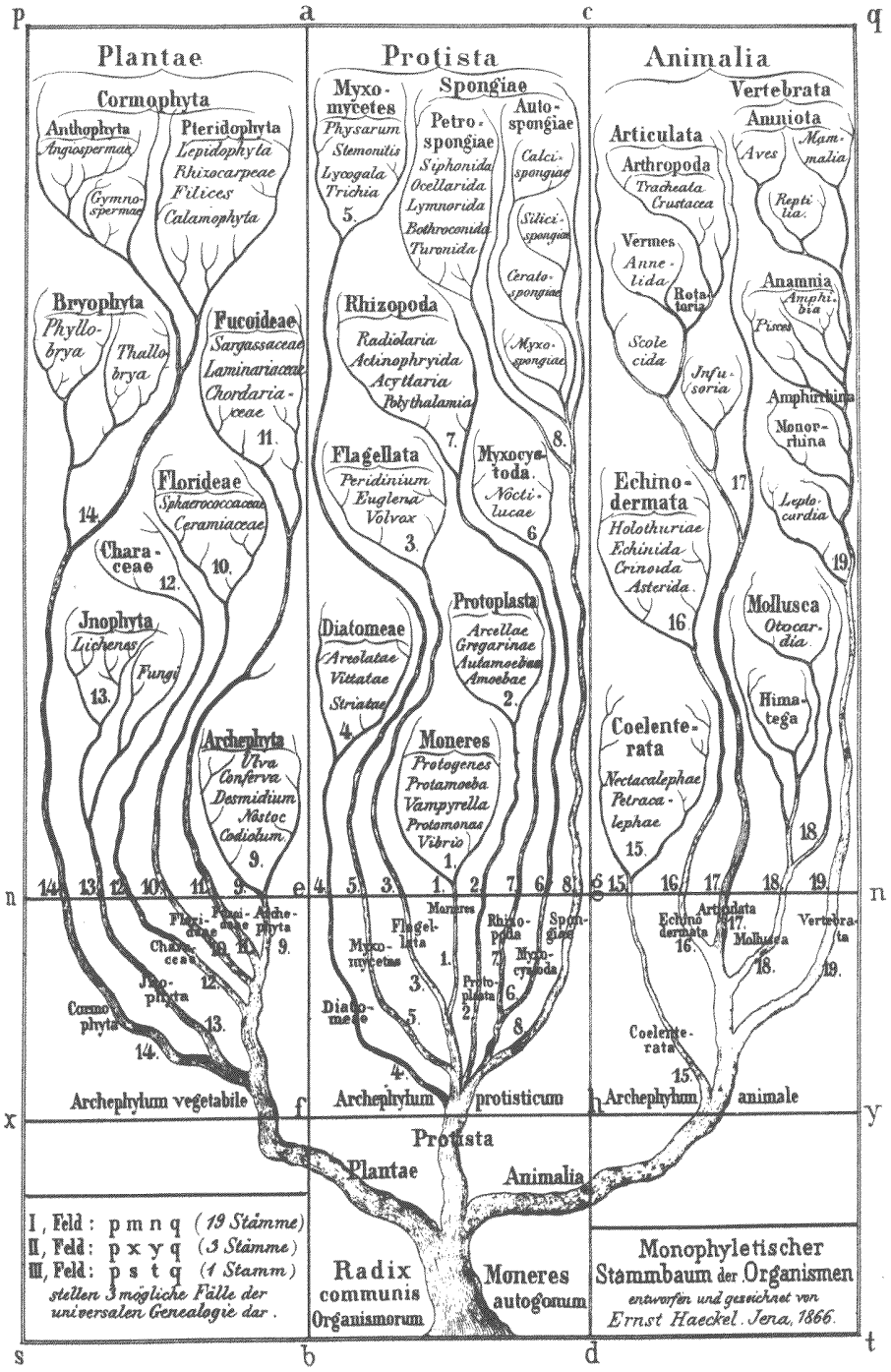


Abbildung 1.1: Monophyletischer Stammbaum der Organismen von Haeckel (1866) aus seinem Buch *Generelle Morphologie der Organismen*

sprechen jedoch dafür, so daß die Existenz von Evolution heute nur selten in Frage gestellt wird. (Futuyma, 1986)

Seit dem letzten Jahrhundert wird versucht, die Entstehung des Lebens anhand von phylogenetischen Stammbäumen immer detaillierter zu rekonstuieren, so z.B. von Haeckel in seinem monophyletischen Stammbaum der Organismen von 1866 (Abb. 1.1).

Auch diese Arbeit hat sich zum Ziel gesetzt, Teile des Stammbaumes des Lebens zu rekonstruieren und damit einen Einblick in die Entstehung heute bestehender Lebensformen und ihrer Phänomene zu gewinnen.

1.2 Datenmaterial für phylogenetische Untersuchungen

Um phylogenetische Bäume rekonstruieren zu können, sind wir auf Datenmaterial angewiesen, das die Entwicklungsgeschichte der Organismen widerspiegelt, d.h. in diesen Daten müssen die Spuren des Entwicklungsprozesses erkennbar sein. Früher wurden zur Rekonstruktion von Stammbäumen in erster Linie morphologische Daten herangezogen.

Eine andere Art von Daten, die vor allem in den letzten 15 Jahren vermehrt für Stammbaumanalysen herangezogen wird, sind molekulargenetische Sequenzdaten von Makromolekülen, wie DNA und Proteinen. (Futuyma, 1986)

Die genetische Information liegt in der Zelle in Form langer DNA–Ketten (Desoxyribonucleinsäure–Ketten) vor. Diese Ketten setzen sich aus Nucleotiden zusammen, die mit unterschiedlichen organischen Basen beladen und über eine Phosphodiesterbrücke zwischen der 3′–OH–Gruppe und der 5′–OH–Gruppe ihrer Desoxyribose miteinander verbunden sind. Die gebundenen Nucleotidbasen gliedern sich in zwei Typen: die Purine Adenin und Guanin sowie die Pyrimidine Cytosin und Thymin.

Die Nucleinsäuren haben die Eigenschaft, miteinander Basenpaarungen eingehen zu können, indem sie untereinander Wasserstoffbrückenbindungen ausbilden. Hierbei können sich Adenin mit Thymin und Guanin mit Cytosin paaren (Abb. 1.2 und 1.3). Daher nennt man die Basen Adenin und Thymin bzw. Guanin und Cytosin *komplementär* zueinander. In der Zelle liegt DNA immer in zwei komplementären Ketten vor, die antiparallel in Form einer Doppelhelix umeinander herumlaufen. Hierbei bilden immer die zwei

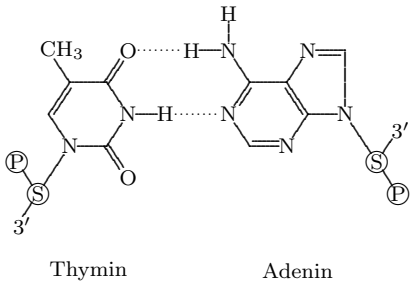


Abbildung 1.2: DNA–Basenpaarung zwischen Adenin und Thymin mittels Wasserstoffbrückenbindungen (gepunktet); innerhalb einer Nucleinsäurekette bildet der Phosphatrest P eine Phosphodiesterbindung zwischen dem 5′–Kohlenstoffatom des Zuckers S (Desoxyribose bei DNA bzw. Ribose bei RNA), an dem es gebunden ist und dem 3′–Kohlenstoff des nächsten Nucleotids

einander gegenüberliegenden komplementären Basen Wasserstoffbrückenbindungen aus. (Stryer, 1988; Li und Graur, 1991)

Seit der Erfindung von Sequenziermethoden, wie der Sequenzierung nach Sanger *et al.* (1977), ist es möglich, die Reihenfolgen der Basen auf molekularer Ebene zu entschlüsseln (zu sequenzieren). Diese Information liegt auf der DNA gerichtet vor. Daher kann der Inhalt, der durch die oben genannten Sequenziermethoden entschlüsselt wurde, als Zeichenketten der Buchstaben A, C, G und T (Adenin, Cytosin, Guanin und Thymin) vom 5′– zum 3′–Ende der DNA–Sequenz betrachtet werden. Informatisch gesehen, können Gensequenzen als Worte über einem Alphabet, bestehend aus den vier Symbolen A, C, G und T, dargestellt werden. (Li und Graur, 1991; Klaeren, 1991)

Seit der Erfindung der oben genannten Sequenziermethoden und deren Automatisierung durch Sequenziermaschinen, wie ALF von Pharmacia (Ansorge *et al.*, 1992, 1993) und ABI 373A (de Bellis *et al.*, 1994), wird eine immer größere Anzahl von Sequenzdaten immer schneller verfügbar. Dieses Anwachsen wird durch die große Anzahl an Genom–Projekten, wie z.B. Drosophila Genome Project, Yeast Genome Projekt und Human Genome Project, noch zusätzlich forciert (Ajioka *et al.*, 1991; Goffeau und Vassarotti, 1991; Watson, 1990). Da diese Daten im allgemeinen in großen Gendatenbanken, z.B. der EMBL/EBI–Datenbank (Rice *et al.*, 1993; Emmert *et al.*, 1994) und GenBank (Benson *et al.*, 1994) frei verfügbar sind, deren Inhalte exponentiell wachsen, erhalten wir die Möglichkeit, die Evolution auf molekularbiologischer Ebene zu betrachten.

Der Vorteil molekulargenetischer Sequenzdaten gegenüber morphologi-

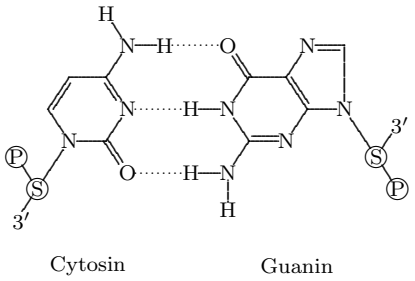


Abbildung 1.3: DNA–Basenpaarung zwischen Guanin und Cytosin mittels Wasserstoffbrückenbindungen(gepunktet); innerhalb einer Nucleinsäurekette bildet der Phosphatrest P eine Phosphodiesterbindung zwischen dem 5′–Kohlenstoffatom des Zuckers S (Desoxyribose bei DNA bzw. Ribose bei RNA), an dem es gebunden ist und dem 3′–Kohlenstoff des nächsten Nucleotids

schen Merkmalen liegt darin, daß jede Base in einer Gensequenz oder jede Aminosäure in einem Protein als einzeln untersuchbares Merkmal betrachtet werden kann. Somit können die vorliegenden Sequenzdaten im Vergleich zu den morphologischen Daten einen größeren Merkmalsumfang und so eine feinere Auflösung der Untersuchungsergebnisse liefern. (Futuyma, 1986)

1.3 Erbgut und Evolution

Das Erbgut oder Genom wird bei jeder Zellteilung semikonservativ verdoppelt. D.h. an jedem der beiden Stränge einer Doppelhelix wird durch DNA–Polymerasen ein neuer komplementärer DNA–Strang synthetisiert, so daß nach der Verdoppelung in jedem der beiden dann vorliegenden Helizes ein alter und ein neuer DNA–Strang enthalten sind. Die beiden so entstandenen Kopien des Genoms werden dann während der Zellteilung auf die beiden Tochterzellen verteilt. (Lewin, 1994; Knippers, 1995)

Das Genom ist aus unterschiedlichen Gründen verschiedenen Veränderungen (Mutationen) unterworfen. Bei Genom–Mutationen finden drastische Veränderungen des gesamten Genoms statt, z.B. Veränderungen der Chromosomenzahl wie etwa der Trisomie. Weiterhin treten Chromosomen–Mutationen auf, die die Struktur eines Chromosoms nachhaltig verändern, wie

- Translokationen (Verlagerung von Chromosomenabschnitten an eine andere Stelle im selben oder in einem anderen Chromosom)

- Deletionen (Verlust von Abschnitten des Chromosoms)
- Insertionen (Einfügen eines neuen Abschnitts in ein Chromosom)
- Inversionen (Umkehrung eines Chromosomen–Abschnitts).

Auf Sequenzebene treten, vor allem durch Fehler während der DNA–Replikation, Punktmutationen auf, die die DNA–Sequenzen in ihrer Basenfolge punktuell verändern. Die bekannten Mutationstypen hierbei sind:

- Substitution einzelner Basen durch andere
- Deletion einer oder mehrerer Basen
- Insertion einer oder mehrerer Basen.

Insertionen und Deletionen werden oft unter dem Oberbegriff Indel zusammengefaßt. Substitutionen sind der am häufigsten auftretende Mutationstyp. Bei Untersuchungen an Teilen des *lac I*–Gens wurde gefunden, daß es sich bei ca. 70% der spontanen Mutationen in diesem Gen um Basensubstitutionen handelt. Substitutionen gliedern sich in zwei verschiedene Typen, je nachdem ob eine Base durch eine andere desselben oder unterschiedlichen Typs ersetzt wird. Wird eine Purinbase durch eine Purinbase ($A \leftrightarrow G$) oder eine Pyrimidinbase durch eine Pyrimidinbase ($C \leftrightarrow T$) ersetzt, so spricht man von einer Transition. Bei der Ersetzung einer Purinbase durch eine Pyrimidinbase oder umgekehrt ($\{A, G\} \leftrightarrow \{C, T\}$) handelt es sich um eine Transversion. (Knippers, 1995; Stryer, 1988)

Mutationen können z.B. bei der Replikation durch Polymerasen spontan durch Fehleinbau organischer Basen entstehen. Dies kann durch Tautomerien der organischen Basen ausgelöst werden, bei diesen können durch Protonenverschiebungen z.B. Ketogruppen in die Enolform oder Aminogruppen in eine Iminoform übergehen. Durch diese Tautomerien sind die Basen dazu in der Lage, Wasserstoffbrücken mit einer anderen als der normalerweise bevorzugten Base auszubilden. Diese Tautomere liegen im Gleichgewichtszustand zu einem sehr geringen Anteil (ca. 10^{-4}) in der DNA vor. (Stryer, 1988)

Der Vorgang der Replikation verläuft, durch verschiedene Reparaturmechanismen unterstützt, mit erstaunlich wenigen Fehlern ab. Die Mutationen können neben dem schlichten Fehleinbau auch durch äußere Einflüsse ausgelöst werden. Sogenannte Mutagene können Veränderungen an der DNA verursachen, die zu einem veränderten Bindungsverhalten der Basen führen

können. Dadurch entstehen in den DNA–Sequenzen Fehlpaarungen, sogenannte Mismatches. Für solche Mismatches existieren Reparaturmechanismen, die ungepaarte Bereiche auf der DNA erkennen und anschließend korrigieren. Hierbei kann es allerdings passieren, daß die falsche der beiden Basen korrigiert wird und so die Mutation erhalten bleibt. (Knippers, 1995; Lodish *et al.*, 1995)

Weitere Möglichkeiten zur Entstehung von Mutationen sind Depurinierungen und Depyrimidinierungen, bei denen eine organische Base von der DNA abdissoziiert, oder die Bildung von Dimeren zwischen Basen durch Einwirkung von UV–Licht. Am häufigsten sind hierbei Thymidindimere zu beobachten. Auch für diese Mutation haben sich in der Zelle Reparaturmechanismen entwickelt. (Knippers, 1995)

Viele Mutationen werden korrigiert. Die Mutationen aber, die falsch oder nicht rechtzeitig korrigiert werden, manifestieren sich spätestens bei der nächsten DNA–Replikation, sofern sich die Zelle noch teilt, und werden an die Nachfolgezellen weitergegeben. Bei einzelligen Organismen, werden solche Mutationen unmittelbar an die Nachkommen oder die Geschlechtszellen weitergegeben. Bei mehrzelligen Organismen dagegen müssen Mutationen in der Keimbahn auftreten, um an die Nachkommen weitergegeben zu werden. Keimbahnzellen sind solche Zellen, in deren Nachkommenschaft sich Geschlechtszellen befinden. Mutationen die in somatischen Zellen auftreten wirken sich zwar eventuell auf den Organismus aus, werden aber nicht weitervererbt. (Alberts *et al.*, 1994; Knippers, 1995)

Die grundlegenden Annahmen der Genetik sind, daß Mutationen zufällig und ungerichtet auftreten und daß Mutanten durch natürliche Selektion begünstigt werden, wenn sie dem betroffenen Organismus einen Vorteil gegenüber dem nichtmutierten Organismus bringen. Außerdem setzen sich Mutationen, die dem Organismus Nachteile bringen, nicht durch, wenn sie die Lebensfähigkeit des betroffenen Organismus herabsetzen. Dieser Vorgang nennt sich Selektion. Ein drastisches Beispiel sind Letalmutanten, die mit dieser Mutation nicht überlebensfähig sind. (Knippers, 1995)

Viele Mutationen manifestieren sich in Bereichen der DNA, auf die kein oder nur geringer Selektionsdruck wirkt. Solche Sequenzen, wie z.B. nichtkodierende Bereiche, die auch sonst keine erkennbare Funktion haben, sind im allgemeinen für Stammbaumanalysen ungeeignet, da sie zu schnell mutieren können. Dagegen werden Sequenzen, die für Funktionen kodieren, durch den Selektionsdruck stabilisiert. Die Veränderungen solcher Sequenzen können genutzt werden, um Stammbaumanalysen durchzuführen.

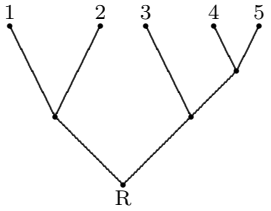


Abbildung 1.4: Gewurzelter phylogenetischer Baum

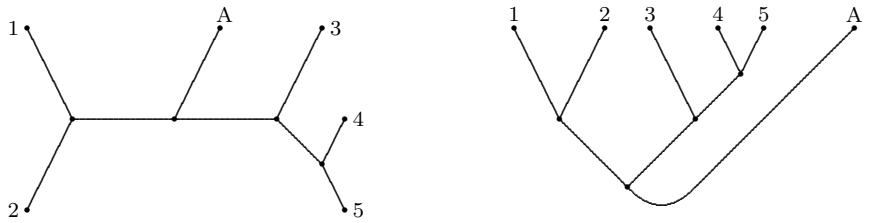
1.4 Phylogenie

Auf der im letzten Kapitel beschriebenen Mutabilität und Vererbung genetischer Sequenzen beruht die Möglichkeit, anhand von Sequenzdaten biologischer Makromoleküle die Entstehung des Lebens mittels Rekonstruktion von Stammbäumen zu untersuchen.

1.4.1 Phylogenetische Stammbäume

In Stammbaumanalysen konstruierte Stammbäume werden graphentheoretisch als binäre Bäume dargestellt (Abb. 1.4). Solche Bäume sind azyklisch verbundene, in diesem Fall ungerichtete Graphen aus einer Menge von Knoten, die mittels Kanten miteinander verbunden sind. In der Menge von Knoten hat jeder Knoten einen Grad $d \leq 3$, d.h. er ist nie mit mehr als drei weiteren Knoten verbunden. Ein Knoten repräsentiert die Wurzel ($d = 2$) und jeder interne Knoten hat genau zwei Nachfolger und einen Vorgänger ($d = 3$). Die internen Knoten repräsentieren gemeinsame Vorfahren an den Verzweigungspunkten der Artbildung. Bei den Blättern ($d = 1$) handelt es sich um rezente Organismen. Die Kantenlängen geben eine Form von Zeit an, die zwischen den einzelnen Knoten liegt, wie z.B. in der Einheit Mutationen pro Generation. (Brandstädt, 1994; Waterman, 1995)

Es ist natürlich nicht klar, daß an einem Verzweigungspunkt immer genau zwei Arten entstanden sind. Diese Ungenauigkeit wird dadurch umgangen, daß man Kanten mit einer Länge null zuläßt. Dies bedeutet, daß die beiden adjazenten Knoten der „Nullkante“ denselben Vorfahren repräsentieren. Beide Knoten haben dann dieselben Merkmale, da die Evolution *keine* Zeit hatte, eines der Merkmale zu verändern. (Felsenstein, 1981)



(a) (b)

Abbildung 1.5: Zwei typische rekonstruierte phylogenetische Bäume. (a) ungewurzelt; (b) wurde mittels einer Außengruppe A gewurzelt

Ein zweites Problem ist, daß wir keine Möglichkeit haben, die Sequenzen des frühesten gemeinsamen Vorfahren, der Wurzel, zu bestimmen, um den Sitz des Ursprungs des Stammbaums zu bestimmen. Daher nimmt man einen Organismus oder eine Gruppe von Organismen, von denen bekannt ist, daß sie im Stammbaum einen gemeinsamen Vorfahren mit den anderen Organismen besitzen, der zeitlich vor dem Aufspalten der zu untersuchenden Gruppe existiert hat. Ein solcher Organismus bzw. eine solche Gruppe heißt Außengruppe, da sie außerhalb des eigentlich zu untersuchenden Bereichs liegt, und fungiert dann als Wurzel (Abb. 1.5b). Es handelt sich bei den gefundenen Bäumen eigentlich um sogenannte *ungewurzelte* Bäume (Abb. 1.5a). Auch die Sequenzen der anderen internen Knoten können heute im Normalfall nicht mehr bestimmt werden. Diese werden in Abhängigkeit von der benutzten Methode meist in irgendeiner Form für die Berechnung rekonstruiert. (Felsenstein, 1981; Swofford und Olsen, 1990)

Bei der Rekonstruktion phylogenetischer Stammbäume werden im Idealfall alle möglichen Bäume anhand eines vorgegebenen Kriteriums (z.B. Maximum Likelihood) untersucht, um sicher zu gehen, daß man auch den nach diesem Kriterium besten Baum findet. (Felsenstein, 1981, 1982; Swofford und Olsen, 1990)

Hierzu muß man allerdings wissen, daß für n Spezies

$$\prod_{k=3}^{n}(2k-5) = \frac{(2n-5)!}{2^{n-3}(n-3)!}$$

verschiedene ungewurzelte binäre Stammbäume existieren. Betrachtet man gewurzelte Stammbäume, so erhöht sich n um eins, da sich die Wurzel wie

ein zusätzliches Blatt im Baum verhält. Dies bedeutet für die Analysen, daß für nur 10 Spezies schon über zwei Millionen ungewurzelter Stammbäume existieren. (Cavalli-Sforza und Edwards, 1967; Felsenstein, 1978)

Aufgrund dieses immensen Wachstums der Anzahl der Bäume, ist es im allgemeinen nicht möglich, die Gesamtheit aller möglichen Bäume zu betrachten, um den optimalen Baum zu finden. Aus diesem Grund werden heuristische Verfahren verwendet, um die Anzahl der zu betrachtenden Bäume einzuschränken. Zwei häufig verwendete Verfahren sind das Clustering–Verfahren[3] und das schrittweise Einfügen (Swofford und Olsen, 1990).

Beim Clustering werden die Sequenzen oder Sequenzgruppen gesucht, die sich nach einem bestimmten Maß, meist der evolutionären Distanz zwischen den Sequenzen, am ähnlichsten sind. Diese bilden zusammen eine neue Gruppe und gehen als solche in das weitere Clustering ein. (Swofford und Olsen, 1990)

Beim schrittweisen Einfügen wird, angefangen mit einer kleinen Teilmenge der zu untersuchenden Taxa, der optimale Baum für diese Taxa bestimmt. Anschließend werden Schritt für Schritt weitere Taxa eingefügt und der, basierend auf dem vorher gefundenen Baum, nächste optimale Baum gesucht. (Felsenstein, 1981; Swofford und Olsen, 1990)

Bei beiden oben genannten Verfahren handelt es sich um Greedy–Verfahren[4], die sich Schritt für Schritt von einer lokal besten Entscheidung zur nächsten bewegen. Diese Verfahren sich vergleichbar mit den in der Informatik verwendeten Verfahren von Kruskal bzw. Prim zum Berechnen Minimaler Spannbäume (MSB) in Graphen (Vingron, 1995, persönliche Kommunikation).

Clustering entspricht hierbei der Methode von Kruskal, bei der die kürzeste verbindende Kante zwischen zwei Zusammenhangskomponenten im Graphen gesucht wird, die dann durch die gemeinsame Zusammenhangskomponente, hier Organismen–Gruppen, ersetzt werden. Dieses wird solange fortgeführt, bis alle Knoten des Graphen zu einer Zusammenhangskomponente verbunden sind (Cormen *et al.*, 1990; Ottmann und Widmayer, 1993).

Prim's Algorithmus entspricht dem schrittweisen Einfügen. Hierbei wird die kürzeste Kante gesucht, die einen neuen Knoten mit einer bestehenden Zusammenhangskomponente verbindet. Dabei wird nur eine Zusammenhangskomponente, hier Bäume der bisher betrachteten Organismen, schrittweise vergrößert, bis alle Knoten enthalten sind (Cormen *et al.*, 1990; Ottmann und Widmayer, 1993).

[3] Cluster (engl.) – Gruppe
[4] greedy (engl.) – gefräßig

1.4.2 Evolutionsmodelle

Da der Evolutionsprozeß ein hochkomplexer und nicht vollständig verstandener Prozeß ist, müssen die angestellten Betrachtungen immer eine Vereinfachung der Wirklichkeit sein.

Daher bedient man sich expliziter Evolutionsmodelle, die versuchen, das Wirken der Evolution zu modellieren. Man nimmt hierbei im allgemeinen an, daß es sich bei der Evolution um einen stochastischen (zufälligen) Prozeß handelt, der mit einer gewissen Wahrscheinlichkeit Basen einer DNA–Sequenz mutiert. Hierfür sind schon viele Modelle entwickelt worden. (Goldman, 1990; Swofford und Olsen, 1990; Yang *et al.*, 1994)

Um die Entwicklung der Modelle zu demonstrieren, seien hier die bekanntesten vorgestellt. Um zu zeigen, daß die in die Modelle eingehenden Parameter immer umfangreicher geworden sind, wurden die Ratenmatrizen mitangegeben. Diese enthalten die Substitutionsraten von einer Basen durch eine andere pro Zeiteinheit (normalerweise pro Zellgeneration). Die Matrizen enthalten an folgenden Stellen die entsprechenden Substitutionsraten:

$$\begin{bmatrix} & (A\to C) & (A\to G) & (A\to T) \\ (C\to A) & & (C\to G) & (C\to T) \\ (G\to A) & (G\to C) & & (G\to T) \\ (T\to A) & (T\to C) & (T\to G) & \end{bmatrix}$$

Diese Ratenmatrizen werden bei der Modellierung von Evolutionsprozessen benutzt, wie in Kapitel 4.1.2 beschrieben.

- Das Jukes–Cantor–Modell oder 1–Parametermodell (Jukes und Cantor, 1969) ist ein sehr einfaches Modell, das annimmt, Transitionen und Transversionen kämen mit gleicher Häufigkeit vor. (α gibt die allgemeine Substitutionsrate an.)

$$\begin{bmatrix} -3\alpha & \alpha & \alpha & \alpha \\ \alpha & -3\alpha & \alpha & \alpha \\ \alpha & \alpha & -3\alpha & \alpha \\ \alpha & \alpha & \alpha & -3\alpha \end{bmatrix}$$

- Das 2–Parameter–Modell von Kimura (1980) unterscheidet in der Häufigkeit zwischen Transitionen (α) und Transversionen (β), berücksichtigt aber die Basenfrequenzen nicht.

$$\begin{bmatrix} -\alpha-2\beta & \beta & \alpha & \beta \\ \beta & -\alpha-2\beta & \beta & \alpha \\ \alpha & \beta & -\alpha-2\beta & \beta \\ \beta & \alpha & \beta & -\alpha-2\beta \end{bmatrix}$$

- Das Modell von Felsenstein (1981), das früher in `dnaml` verwendet wurde, geht zwar auf die Basenfrequenzen (f_i) ein, betrachtet aber nur eine einheitliche Substitutionsrate α.

$$\begin{bmatrix} -\alpha(f_C+f_G+f_T) & \alpha f_C & \alpha f_G & \alpha f_T \\ \alpha f_A & -\alpha(f_A+f_G+f_T) & \alpha f_G & \alpha f_T \\ \alpha f_A & \alpha f_C & -\alpha(f_A+f_C+f_T) & \alpha f_T \\ \alpha f_A & \alpha f_C & \alpha f_G & -\alpha(f_A+f_C+f_G) \end{bmatrix}$$

- Ein erstes Modell, das beide Parameter berücksichtigte, wurde veröffentlicht von Hasegawa, Kishino und Yano (1985)

$$\begin{bmatrix} .-(\beta f_C+\alpha f_G+\beta f_T) & \beta f_C & \alpha f_G & \beta f_T \\ \beta f_A & -(\beta f_A+\beta f_G+\alpha f_T) & \beta f_G & \alpha f_T \\ \alpha f_A & \beta f_C & -(\alpha f_A+\beta f_C+\beta f_T) & \beta f_T \\ \beta f_A & \alpha f_C & \beta f_G & -(\beta f_A+\alpha f_C+\beta f_G \end{bmatrix}$$

- Das von Kishino und Hasegawa (1989) veröffentlichte generalisierte 2–Parameter–Modell betrachtet ebenfalls beide Parameter und berücksichtigt ebenfalls explizit die Frequenzen von Purinen und Pyrimidinen. Dieses Modell wird bei den in dieser Arbeit angestellten Analysen benutzt und ist in Kapitel 4.1.2 beschrieben.

Da die Gesetzmäßigkeiten beim Auftreten von Indels noch nicht sehr gut verstanden sind und somit noch keine Möglichkeit besteht, diese sinnvoll zu modellieren, werden sie bei allen gebräuchlichen Evolutionsmodellen außeracht gelassen und nur Basensubstitutionen betrachtet.

1.4.3 Phylogenetische Methoden

Zur Rekonstruktion phylogenetischer Bäume aus Sequenzdaten biologischer Makromoleküle haben sich verschiedene Methoden etabliert. Diese lassen sich allgemein in zwei Gruppen unterteilen, in distanzbasierte und merkmalsbasierte Methoden. (Felsenstein, 1982; Swofford und Olsen, 1990; Waterman, 1995; Weir, 1990)

Bei den distanzbasierten Methoden wird für alle Sequenzpaare eine evolutionäre Distanz berechnet. Im einfachsten Fall ist dies die Hammingdistanz (Cormen *et al.*, 1990), bei der die Anzahl der sich unterscheidenden Basen zweier Sequenzen aufsummiert wird. Die errechnete Distanzmatrix dient als Grundlage zur Konstruktion des Stammbaumes. Hierfür werden zumeist

Clustering–Verfahren verwendet (Kap. 4). An dieser Methode wird kritisiert, daß die vorhandenen Sequenzdaten zur Berechnung des Baumes auf die Distanzen reduziert werden. (Fitch und Margoliash, 1967; Saitou und Nei, 1987)

Wichtige Vertreter der merkmalsbasierten Methoden sind die Maximum–Parsimony–Methode[5] und der Maximum–Likelihood–Ansatz. Das Maximum–Parsimony–Verfahren konstruiert für alle internen Knoten eines vorgegebenen Stammbaumes Sequenzen, die die von diesen Knoten repräsentierten Organismen gehabt haben könnten. Diese Sequenzen werden so konstruiert, daß die Sequenzen entlang des Baumes, während der vom Stammbaum vorgegebenen evolutionären Entwicklung, möglichst wenigen Mutationen unterworfen sind. Die Gesamtsumme aller im Baum nötigen Mutationen ist dann das Maß für die Qualität des Baumes. Von der Maximum–Parsimony–Methode wird versucht, aus allen möglichen Stammbäumen denjenigen zu finden, für den die geringste Anzahl an Mutationen nötig ist. Diese Methode wurde ursprünglich für morphologische Daten entworfen und hat sich bewährt, wenn sich die beobachteten Merkmale nur selten ändern. Das gilt im allgemeinen für morphologische Daten. Diese Methode scheitert jedoch, wenn die beobachteten Merkmale hochvariabel sind oder sehr lange Kanten im gesuchten Baum vorkommen. (Swofford und Olsen, 1990; Waterman, 1995)

Der anerkannteste Ansatz basiert auf der von Fisher (1912) eingeführten Maximum–Likelihood–Methode[6] (Kreyszig, 1975). Die Maximum-Likelihood–Methode wird benutzt, um unbekannte Parameter zu schätzen, von denen eine bekannte Wahrscheinlichkeitsfunktion für einen stochastischen Prozeß abhängt. Mit dieser Methode werden dann anhand einer festen Stichprobe, in unserem Fall der Sequenzdaten, die unbekannten Parameter so geschätzt, daß der Wert der Wahrscheinlichkeitsfunktion, bei fester Stichprobe als Likelihood–Funktion bezeichnet, sein Maximum erreicht. Die in der Stammbaumanalyse zu schätzenden unbekannten Parameter sind die Kantenlängen in einem vorgegebenen Baum. (Kreyszig, 1975; Goldman, 1990)

Diese Methode wird in dieser Arbeit verwendet und ist in Kap. 4.2 ausführlich beschrieben.

Der Vorteil dieser Methode liegt darin, daß unter Berücksichtigung eines expliziten Evolutionsmodells bei der Berechnung der Stammbäume die vollständigen Daten in die Analyse mit eingehen. Der Hauptnachteil der ML–Methode ist, daß enorme Rechenzeiten nötig sind, um die große Anzahl der möglichen Stammbäume zu überprüfen, die auch bei heuristischen Methoden

[5] Maximum Parsimony (engl.) – maximale Sparsamkeit
[6] Maximum Likelihood (engl.) – maximale Wahrscheinlichkeit

Achlya bisexualis	U	A	G	U	C	A	U	A	C	G
Ochromonas danica	U	A	G	-	C	A	U	A	C	G
Guillardia theta	U	A	G	U	C	A	U	A	U	G

Abbildung 1.6: Ausschnitt aus dem Alignment von 18S rRNA–Sequenzen aus dem Zellkern

meist exponentiell mit der Anzahl der benutzten Spezies wächst. (Felsenstein, 1981; Swofford und Olsen, 1990)

Die meisten merkmalsbasierten Methoden, die Sequenzen biologischer Makromoleküle benutzen, benötigten von diesen Sequenzen ein Alignment[7]. In einem solchen Alignment stehen die homologen Merkmale (hier organische Basen oder Aminosäuren) in Spalten untereinander für jede zu untersuchende Spezies. Alle Sequenzmerkmale einer Spezies stehen hierbei in einer Zeile (Abb. 1.6). In einem solchen Alignment lassen sich Mutationen ablesen, die zwischen zwei Sequenzen liegen, z.B. ist wahrscheinlich zwischen *Ochromonas danica* und den beiden anderen Spezies bei Spalte 4 irgendwann einmal ein Indel und zwischen *Guillardia theta* und den anderen eine Substitution ($C \leftrightarrow U$, bzw. $C \leftrightarrow T$ auf DNA–Ebene) in Spalte 9 aufgetreten. Die hier abzulesenden Mutationen sind allerdings nur die offensichtlichen Mutationen. Wieviele Hin– und Rückmutationen es während der Entwicklungsgeschichte gegeben hat und welche Sequenzen die Vorfahren der untersuchten Spezies hatten, können wir heute nicht mehr feststellen. Auch liegen die Sequenzen der Makromoleküle, die wir untersuchen, nach dem Sequenzieren oder der Datenbankrecherche nur als einzelne Buchstabenfolgen vor. Diese zu alignieren, ist ein anderes hervorstechendes Problem, an dem zur Zeit in der Bioinformatik gearbeitet wird. Die meisten Programme zum Alignieren von Sequenzen bringen nur sehr unbefriedigende Ergebnisse, so daß nahezu alle erhaltenen Alignments noch von Hand nachgebessert werden müssen bzw. viele Alignments von vorneherein von Hand angefertigt werden.

Die Ergebnisse der Stammbaumanalysen hängen sehr von der Qualität der benutzten Alignments ab. Daher ist es wichtig, Alignments zu benutzen, die möglichst fehlerfrei sind. Über dieses Problem wird noch später im Kapitel 1.8 über rRNA–Sequenzen gesprochen.

[7] Alignment (engl.) – wörtlich: Ausrichtung

1.5 Parallele Computerplattformen und Parallelrechnen

Eine Methode, um sehr rechenaufwendige Probleme auf Computern praktikabler lösen zu können, ist die Parallelisierung.

Hierbei wird versucht, das gesamte Problem in viele kleine Teilprobleme aufzuspalten und diese einzeln, parallel zueinander zu lösen. Die zu bewältigende Arbeit wird hierbei von einem auf viele Computer bzw. Prozessoren, im allgemeinen Knoten genannt, verteilt. Diese Methode ähnelt dem *Devide–and–Conquer–Prinzip*[8], das unter anderem in Sortierverfahren Anwendung findet (Cormen *et al.*, 1990; Ottmann und Widmayer, 1993).

Zur Parallelisierung stehen verschiedene Parallelrechnerkonzepte und Parallelrechnerplattformen zur Verfügung. In erster Linie handelt es sich dabei um drei verschiedene Plattformen. (Burkhardt, 1993)

Zum einen sind dies die shared–memory–Systeme, die aus einer unterschiedlichen Anzahl einzelner Prozessoren bestehen, die über den Zugriff auf einen gemeinsamen Speicher miteinander verbunden sind (z.B. shared–memory–Systeme von Silicon Graphics). Hierbei können die einzelnen Prozessoren auch noch zusätzlich eigenen Speicher besitzen.

Zum anderen gibt es Parallelrechner, die aus mehr oder weniger eigenständigen Recheneinheiten mit eigenem Prozessor und Speicher bestehen. Die Recheneinheiten sind durch ein schnelles Verbindungsnetzwerk mit fester oder variabler Topologie miteinander verbunden (z.B. der Parallelrechner SP2 von IBM, der in dieser Arbeit verwendet wurde).

Beim dritten Typ handelt es sich um verteilte Rechnersysteme mit LAN-Kopplung[9]. Dies sind im allgemeinen Workstations, die über Netzwerksysteme wie Ethernet, FDDI oder Tokenring miteinander verbunden sind. Solche verteilten Rechnersysteme können überall dort eingerichtet werden, wo Workstations in homogener oder heterogener Form miteinander vernetzt sind. Da dies in vielen Institutionen der Fall ist und beispielsweise Arbeitsplatzrechner nachts zumeist nicht benutzt werden, bietet sich die Möglichkeit, solche Rechenkapazitäten zu entsprechenden Zeiten zu Parallelplattformen zusammenzuschalten.

Da in den letzten Jahren der Zugriff auch auf echte parallele Computersysteme immer einfacher geworden ist, bietet es sich an, größere Probleme auf

[8] devide and conquer (engl.) – frei übersetzt: teile und herrsche
[9] LAN – Local Area Network

solche parallelen Rechnersysteme zu überführen, um die Einschränkungen durch die immensen Rechenzeiten zu mildern.

Die Methode der Parallelisierung findet auch in der Biologie, vor allem in der Molekularbiologie, immer stärkere Anwendung. So wurden z.B. das Programm BLAST zur Homologiesuche in genetischen Datenbanken (Jülich, 1995) oder Programme zur Sekundärstrukturvorhersage von RNA (Nakaya *et al.*, 1995) auf parallelen Rechnerplattformen implementiert.

Auch das Programm `fastDNAml` von Olsen *et al.* (1994a), das auf dem Programm `dnaml` Version 3.3 von Felsenstein (1981) basiert, wurde schon einmal parallelisiert (Matsuda *et al.*, 1994, unveröffentlicht). Die damals der Parallelisierung zugrunde liegende Software wird aber seit einiger Zeit nicht mehr weiterentwickelt und daher nicht mehr an neue Rechnersysteme angepaßt.

Wichtig bei solchen Parallelimplementierungen ist allerdings, die Portabilität für eine möglichst große Anzahl von Plattformen zu ermöglichen, um späteren Weiterentwicklungen der Computer ohne allzu großen Aufwand auf Softwareseite Rechnung tragen zu können und vielen Benutzern die Anwendung zu ermöglichen.

Daher soll in dieser Diplomarbeit die sequentielle Version von `fastDNAml` erneut parallelisiert werden, unter besonderer Berücksichtigung einer einfachen Portierbarkeit für unterschiedliche parallele Rechnersysteme.

1.6 Stammbaum des Lebens, Crown Group Radiation und Plastidenentstehung

Spätestens seit Carl von Linné (1707–1778) das erste umfassende System der Lebewesen schuf, beschäftigen sich Biologen mit der Frage nach den Verwandtschaftsverhältnissen zwischen den Lebewesen. Seitdem sind viele Klassifikationsschemata und Stammbäume vorgeschlagen und viele auch wieder verworfen worden (Margulis und Schwartz, 1988; Watson *et al.*, 1992).

Vergleicht man zum Beispiel den monophyletischen Stammbaum der Organismen (Abb. 1.1), wie er 1866 von Haeckel vorgeschlagen wurde, mit dem heutigen Stand der Erkenntnis, so haben tiefere Einblicke in die Strukturen der Organismen einiges verändert. Damals wurden noch alle Einzeller inklusive der Bakterien (hier Moneren) und die Schwämme (Spongidae) zu einer monophyletischen Gruppe von Protisten zusammengefaßt. Die Schwämme

werden heutzutage bei den Tieren angesiedelt. (Haeckel, 1866; Wainright *et al.*, 1993; Wehner und Gehring, 1995)

Später wurden die Organismen in zwei Überreiche[10] unterteilt, in die Prokaryoten (Bakterien; Zellen ohne Zellkern) und die Eukaryoten (Zellen mit Zellkern). Doch auch diese Einteilung mußte korrigiert werden, als Sequenzanalysen der Gene ribosomaler RNA (rRNA) zeigten, daß sich die Prokaryoten in die Archaebakterien (Archaea) und die Eubakterien aufspalten, die sich stark voneinander unterscheiden (Watson *et al.*, 1992).

Die Frage, in welcher Reihenfolge diese drei Überreiche divergierten, ist aktuelles Forschungsthema. Alle drei Möglichkeiten, d.h. Stammbäume mit Eukaryotischer, Eubakterieller oder Archaebakterieller Wurzel, werden unterstützt, je nachdem mit welchen Datensätzen man die phylogenetischen Analysen durchführt (Saccone *et al.*, 1995).

Auch die Unterteilung der Eukaryoten in die vier Reiche[11] Tiere, Pilze, Pflanzen und Protisten ist umstritten. Zwar besteht bei der Einteilung der Tiere, Pilze und Pflanzen in monophyletische Gruppen weitgehend Konsens, doch ist die Zusammengruppierung der vielen Einzeller und der von einem Einzeller ableitbaren Mehrzeller im Reich der Protisten sehr umstritten. Viele der Linien der Protisten werden heute nicht mehr als monophyletisch betrachtet. Dies macht die Einteilung in ein Reich um so problematischer.

Die Linien dieser Reiche, die die Gruppe in der Krone (Crown Group) des Stammbaums des Lebens bilden, sind wahrscheinlich in einem eng umgrenzten Zeitraum, der Crown Group Radiation, entstanden. In diesem Bereich der Crown Group Radiation gibt es viele interessante Phänomene zu untersuchen. Ein besonders interessanter Aspekt ist das Entstehen der Zellorganellen der Eukaryoten, wie Mitochondrien und Plastiden. Diese sind, wie Sequenzanalysen von Genen dieser Organellen zeigen konnten, wahrscheinlich durch Endosymbiose, nach Aufnahme anderer Organismen in die Zellen, entstanden (Endosymbiontentheorie). Die Vermutung, daß es sich bei den Plastiden um Endosymbionten handelt, wurde schon Anfang dieses Jahrhunderts aufgestellt. (Bhattacharya und Medlin, 1995; Knoll, 1992; Margulis, 1970; Mereschkowsky, 1905, 1910; Watson *et al.*, 1992)

Die Plastiden der Landpflanzen (hier im allgemeinen als Pflanzen bezeichnet) und Grünalgen, die Chloroplasten, ebenso wie die der Rotalgen (Rhodophyta), die Rhodoplasten, besitzen ein Membranenpaar. Es wird davon ausgegangen, daß diese Plastiden durch primäre Endosymbiose zwischen

[10] Superkingdoms
[11] Kingdoms

einem eukaryotischen Wirt und einen prokaryotischen Endosymbionten entstanden sind, der wahrscheinlich ein Vorfahr der heutigen Cyanobakterien war. (Douglas und Turner, 1991; Helmchen *et al.*, 1995; Bhattacharya und Medlin, 1995; Watson *et al.*, 1992)

Gleiches gilt für eine andere Art der Plastiden, die Cyanellen, die bei den Glaucocystophyta auftreten. Cyanellen haben eine wichtige Rolle bei der Untersuchung der Plastidenentwicklung gespielt, da sie im Gegensatz zu allen anderen Plastiden eine Peptidoglucan–Zellwand besitzen, ebenso wie die Cyanobakterien. Anfangs wurden die Cyanellen als eine Gruppe innerhalb der Cyanobakterien geführt, um ihre Ursprünglichkeit zu unterstreichen. Dieses mußte nach Untersuchungen des Cyanellengenoms hinsichtlich seines Umfangs und seiner Struktur revidiert werden, da hier größere Verwandtschaft zu den Plastiden als zu den freilebenden Cyanobakterien gezeigt werden konnte. (Douglas und Turner, 1991; Helmchen *et al.*, 1995; Bhattacharya *et al.*, 1995; Bhattacharya und Medlin, 1995)

Eine weitere Art von Plastiden, die vor allem bei verschiedenen Algenstämmen auftritt, sind die komplexen Plastiden. Sie besitzen statt der zwei Membranen, wie sie eine normale primäre Endosymbiose erwarten lassen würde, vier oder drei Membranen. Cryptophyten und Chlorarachniophyten haben eine Form von komplexen Plastiden, die auf den Ursprung der zusätzlichen Membranen schließen lassen. Sie besitzen innerhalb der äußeren zwei Plastidenmembranen zwei eigene Organellen, ein Plastid und einen Nucleomorph[12]. Dieser Nucleomorph enthält DNA und RNA, außerdem lassen sich in diesen Plastiden Ribosomen finden. Die Theorie, es handle sich um eine durch sekundäre Endosymbiose aufgenommene eukaryotische Zelle, konnte von Maier *et al.* (1991) untermauert werden, nachdem es ihnen gelungen war, Nuclei und Nucleomorph getrennt voneinander aus Cryptophyten zu isolieren. Sequenzanalyses ribosomaler DNA, sowohl aus dem Nucleus, als auch aus dem Nucleomorph, ergaben, daß es sich hierbei in beiden Fällen um eukaryotische Ribosomen handelte. (Bhattacharya und Medlin, 1995; Maier *et al.*, 1991; Watson *et al.*, 1992)

Maier *et al.* (1991) gaben ein Modell für die Entstehung der komplexen Plastiden an, wonach heterotrophe Eukaryoten andere, photoautotrophe Eukaryoten durch Endosymbiose aufgenommen hätten. Diese Endosymbionten verloren mit der Zeit die meisten ihrer eigenen Zellorganellen bis auf ihren zum Nucleomorph reduzierten Zellkern, der auch später noch ganz reduziert wurde. An dieser Zwischenstufe, so vermutet man, stehen die Cryptophyten und Chlorarachniophyten. (Maier *et al.*, 1991)

[12] wörtlich: kernförmig von *nucleus* (lat.) – Kern; *μορφή* (griech.) – Gestalt, Form

Weiterhin wird diskutiert, ob die einfachen Plastiden monophyletisch sind, d.h. durch eine einzelne primäre Symbiose entstanden sind, oder ob es mehrere primäre Symbiosen gegeben hat. Die Theorie, daß Plastiden polyphyletisch sind, wird vor allem dadurch gestützt, daß in verschiedenen Plastiden unterschiedliche Pigmente zur Lichtaufnahme in der Photosynthese genutzt werden (z.B. Pflanzen, Grünalgen: Chlorophyll *a* und *b*; Rhodophyta, Glaucocystophyta: Chlorophyll *a* und Phycobilline; Haptophyta, Heterokontophyta: Chlorophyll *a* und *c*; Cryptophyta: Chlorophyll *a* und *c* und Phycobilline). Die meisten phylogenetischen Untersuchungen mit Gensequenzen aus Plastiden unterstützen dagegen einen monophyletischen Urprung der Plastiden. Auch könnte der Vorfahr der Plastiden alle verschiedenen Pigmente besessen haben. (Bhattacharya und Medlin, 1995)

Ein Ziel dieser Arbeit ist es zu versuchen, mit der ML–Methode Einblicke in die Plastidenentwicklung und einige andere Aspekte der Crown Group Radiation zu gewinnen.

1.7 Mutationsraten und Molekulare Uhren

Wie schon vorher beschrieben, treten Punkt–Mutationen im Genom entweder spontan oder durch Einwirkung äußerer Einflüsse (Mutagene) auf. Untersuchungen der Mutationsraten anhand von Gensequenzen haben ergeben, daß Basensubstitutionen mit einer Rate von einer Substitution in 10^9 bis 10^{10} Basenpaaren pro Zellgeneration auftreten. Diese spontane Mutationsrate stimmt zwischen Prokaryoten und Eukaryoten weitestgehend überein. (Knippers, 1995)

Aus dieser gleichmäßigen Substitutionsrate leitet sich der Wunsch ab, bei phylogenetischen Analysen die Sequenzen als molekulare Uhren zu verwenden, um Aussagen über die Zeit zu machen, die zwischen den einzelnen Aufspaltungen vergangen ist.

Untersuchungen an Vertebraten[13] haben allerdings ergeben, daß sich die Substitutionsraten zwischen den einzelnen Linien zum Teil stark unterscheiden. Dieses Phänomen wird auf den Effekt zurückgeführt, den die unterschiedlichen Generationszeiten mehrzelliger Organismen haben können. Die für unsere Betrachungen wichtigen Mutationen finden in der Keimbahn mehrzelliger Organismen statt, da nur diese weitervererbt werden können. Haben nun zwei Organismen etwa dieselbe Anzahl an DNA–Replikationen in der

[13] Wirbeltiere

Keimbahn, so ist die mögliche Substitutionsrate pro Zeit in dem Organismus höher, bei dem die Generationsdauer kleiner ist. Das kommt daher, daß in demselben Zeitraum mehr Zellteilungen bzw. DNA–Replikationen in der Keimbahn stattfinden und damit mehr Mutationen entstehen können. Eine zusätzliche Erklärung ist, daß die unterschiedlichen Organismenlinien auch unterschiedlich gut funktionierende Reparaturmechanismen besitzen und auch daher unterschiedlich hohe Mutationsraten möglich sind. (Futuyma, 1986; Kohne, 1970; Li und Graur, 1991)

Aus diesem Grund ist es nur schwer möglich, genauere Aussagen über die zeitlichen Abstände zu machen. Denn eine Umrechnung der Kantenlängen in Zeiteinheiten ohne genaue Kenntnis der Mutationsraten pro Generation ist nicht möglich.

Obwohl also keine globale molekulare Uhr herangezogen werden kann, kann es möglich sein, lokal, d.h. bei eng verwandten Organismen, anhand molekularer Uhren Zeitabstände zu schätzen. So wird z.B. die zeitliche Entwicklung der Affen und Menschen untersucht, die mit dem Wissen um den Zeitpunkt der Divergenz einer Außengruppe aus fossilen Daten geeicht wird. (Hasegawa *et al.*, 1985; Li und Graur, 1991)

Weitere Probleme für die Benutzung molekularer Uhren liegen darin, daß es Reparationsmechanismen gibt, die bei weitem nicht verstanden sind. Dadurch kann man keine Aussagen über die zeitliche Verzerrung machen, die durch solche Mechanismen ausgelöst werden, wodurch eine Eichung der Zeitabstände sehr ungenau würde. Ein solches Problem tritt z.B. auf, wenn als Grundlage der phylogenetischen Untersuchungen Gensequenzen verwendet werden, die mit vielen Kopien im Genom (Multikopie–Gene) vorkommen. Bei diesen ist nicht bekannt, wie der Mechanismus funktioniert, mit dem die Zellen die einzelnen Sequenzen gegen den Mutationsdruck identisch halten. Es wurden zwar in der Vergangenheit mehrere Mechanismen vorgeschlagen, aber verstanden hat man diesen Vorgang nicht. (Lodish *et al.*, 1995)

Aus den oben angeführten Gründen werden molekulare Uhren selten benutzt. Auch in dieser Arbeit werden wegen dieser Unsicherheiten und der Benutzung von ribosomaler RNA, einem Multikopie–Gen, keine Aussagen über die Zeiträume der Kanten versucht.

1.8 Ribosomale RNA und Stammbaumrekonstruktion

Ein großes Problem bei der Wahl der Sequenzdaten für Stammbaumanalysen ist die Frage: Welche Sequenzen bzw. welche Gene enthalten die beste Information, um daraus einen realistischen Stammbaum zu konstruieren? Diese Frage ist schwer zu beantworten, da die Mutationsraten nicht nur zwischen unterschiedlichen Organismen, sondern auch zwischen verschiedenen Genen verschieden sind. So ist z.B. die Mutationsrate der Histone minimal, während sie bei anderen Genen besonders hoch ist. Weitere Probleme sind, daß erstens verschiedene Gene verschiedene Stammbäume unterstützen können (Saccone *et al.*, 1995) und daß zweitens bestimmte Gene nicht in der interessanten Gruppe von Organismen existieren, bei denen bestimmte Phänomene untersucht werden sollen. Auch kann es Probleme geben, ein realistisches Alignment für das zu benutzende Gen zu finden.

Gesucht werden also Sequenzen, die in möglichst vielen Organismen vorkommen und auch bekannt sind. Zusätzlich sollten sie gut untersucht und leicht verfügbar sein. Eine Sequenz, die in den letzten Jahren eine große Rolle in der Stammbaumanalyse gespielt und gute Ergebnisse erzielt hat, sind die 16S– (Prokaryoten) und 18S–Sequenzen (Eukaryoten) aus der kleinen Untereinheit der Ribosomen (16S/18S rRNA).

Ein weiterer Vorteil ist, daß es sich bei rRNA um ein Molekül handelt, das in allen bisher untersuchten Organismen vorkommt, sehr gut untersucht ist und in großer Zahl in großen rRNA–Datenbanken wie in Antwerpen oder bei RDP (Ribosomal Database Project) abrufbar ist. Aufgrund des hohen Wissens über die Sekundärstruktur der rRNA ist es möglich, diese Sequenzen in den Datenbanken in alignierter Form abzulegen. In dieses Alignment ist neben dem Wissen um die Sequenz auch die Sekundärstruktur eingeflossen. In der Datenbank in Antwerpen sind zur Zeit etwa 1500 verschiedene 16S/18S rRNA–Sequenzen aligniert. Diese stammen aus allen Arten von Organismen wie Tieren, Pflanzen, Pilzen, Protisten, Archae– und Eubakterien, aber auch aus verschiedenen Organellen wie Mitochondrien, einfachen und komplexen Plastiden, sowie auch aus Nucleomorphen. Dieses gibt uns die Möglichkeit, viele Bereiche des Stammbaumes des Lebens zu untersuchen. (Olsen *et al.*, 1992; Olsen und Woese, 1993; Van de Peer *et al.*, 1994)

Für die rRNA–Sequenzen wurde außerdem noch kein Fall von lateralem Gentranfer[14] gefunden, was für viele Gene nicht unüblich ist, da es sich beim

[14]Transfer von Genen aus dem Genom eines Organismus in das Genom eines anderen

Eukaryotengenom höchstwahrscheinlich im ein chimäres Gebilde handelt, das durch lateralen Gentransfer aus den verschiedenen Organismen Gene aufgenommen hat. So sind einige Gene, wie z.B. das einer Untereinheit einer mitochondralen RNA–Polymerase, wahrscheinlich vom Mitochondriengenom auf den Zellkern übergegangen und später im Mitochondriengenom verloren gegangen. (Lodish *et al.*, 1995)

Aufgrund der oben angeführten Vorteile, wurden auch in dieser Arbeit Sequenzen von rRNA verwendet. Da keine Aussagen über genaue Zeiträume vorgenommen werden sollten, ist davon auszugehen, daß sich die in Kap. 1.7 beschriebenen Probleme, die sich aus der Benutzung von Multikopie–Genen ergeben können, bei den in dieser Arbeit ausgeführten Untersuchungen nicht auswirken.

1.9 Entwicklung der Aktingene

Aktin ist ein Protein, das in allen Organismen ein wichtige Rolle bei Muskelbewegungen und Zellbewegungen spielt. Aktin ist das wahrscheinlich beststudierte Mitglied der Protein-Familie der ATPasen. Aktin muß aufgrund seiner Verbreitung wahrscheinlich schon in den ersten existenten Zellen vorhanden gewesen sein.

Aktin ist in Eukaryoten eines der am besten untersuchten Proteine. Aber auch bei Archae– und Eubakterien werden Aktine erforscht. Vor allem bei der Muskelbewegung von Tieren sind Aktine untersucht worden. In einigen Organismengruppen liegen Aktingene in Einzelkopie vor. In anderen wiederum gibt es mehrere Arten von Aktinen.

Im Fall der Aktine lassen sich nicht nur Verwandtschaftsverhältnisse zwischen einzelnen Organismen untersuchen. Hier lassen sich auch die Entwicklung und Entstehung von den verschiedenen Aktintypen betrachten.

Da die Aktine eine so wichtige Rolle bei der Bewegung und beim Aufbau des Cytoskeletts spielen, kann man damit rechnen, daß die phylogenetische Analyse der Aktine sowohl interessante Einblicke in die Entwicklung der Organismen, in unserem Fall der Eukaryoten, als auch in die Entwicklung von Genfamilien durch Genduplikationen im Laufe der Evolution gibt. Daher werden auch im Rahmen dieser Arbeit die Entwicklungen der Aktine anhand ihrer kodierenden Genbereiche untersucht werden.

Ein weiterer Vorteil der Aktinsequenzen ist die gut untersuchte Proteinstruktur. Aufgrund der bekannten funktionellen Domänen von Aktin ist es möglich, bessere Alignments von Aktinsequnzen zu erstellen. (Alberts *et al.*, 1994; Bhattacharya und Ehlting, 1995)

Kapitel 2

Zielsetzung

Zielsetzung dieser Arbeit ist zum einen, die Benutzung des Maximum–Likelihood–Ansatzes von Felsenstein (1981) bei phylogenetischen Untersuchungen für größere Problemstellungen zu ermöglichen. Hierzu sollte ein Computerprogramm (`pfastDNAml`) erstellt werden, das die Benutzung der Maximum–Likelihood–Methode auf parallelen Computerplattformen ermöglicht. Als Grundlage für die Parallelisierung der Maximum–Likelihood–Methode dient eine schon existente sequentielle Implementierung des Maximum–Likelihood–Ansatzes (`fastDNAml`, Olsen *et al.* 1994a), die in der Praxis erprobt ist.

Ziel der Parallelisierung ist es, den Zeitaufwand für ML–Analysen durch die Benutzung von Parallelrechnern zu verringern und damit die Analyse von größeren Datensätzen zu ermöglichen, als dies bisher in einem vertretbaren zeitlichen Rahmen möglich ist. Hierbei soll besonderer Wert auf die Portierfähigkeit auf unterschiedliche Plattformen gelegt werden. Um die Portierbarkeit zu gewährleisten, soll die Parallelisierung auf der Basis der PARMACS–Funktionsbibliothek durchgeführt werden, die für eine breite Anzahl verschiedener Parallelrechner vorhanden ist. Da es sich bei PARMACS um ein kommerzielles Produkt handelt, das auch im Bereich der Industrie verwendet wird, ist davon auszugehen, daß dieses Produkt noch geraume Zeit unterstützt und weiterentwickelt wird.

Mit dem so erhaltenen Programm `pfastDNAml` sollten anschließend phylogenetische Analysen größerer Datensätze durchgeführt werden, um Einblicke in verschiedene Details der Crown Group Radiation zu erhalten.

Hierzu sollte die Entwicklung unterschiedlicher Aktintypen bei Eukaryoten betrachtet werden. Außerdem sollte besonderes Augenmerk auf die Entstehung und Entwicklung der verschiedenen einfachen und komplexen Plastidentypen gelegt werden. Für die Analysen wurden Alignments von ribosoma-

ler 16S und 18S RNA sowie von Aktinsequenzen verwendet. An diesen sollte zudem die Anwendbarkeit einer einfachen Gewichtungsmethode zur besseren Filterung der in den Alignments enthaltenen Information überprüft werden.

Kapitel 3

Material und Geräte

3.1 Datenmaterial

Es wurden für die Untersuchungen im Bereich der Crown Group Radiation und der Plastidenentstehung unterschiedliche Datensätze verwendet. Zum einen wurden ribosomale RNA Sequenzen (rRNA) der kleinen Ribosomenuntereinheit verwendet, und zwar sowohl 18S rRNA aus dem Kern von Eukaryoten als auch 16S rRNA aus Bakterien und Organellen. Außerdem wurden Aktinsequenzen verwendet.

Die Alignments (Aktin, rRNA), die für Untersuchungen in dieser Arbeit verwendet wurden, wurden freundlicherweise von Herrn Dr. D. Bhattacharya (MPI für biophysikalische Chemie, Göttingen) zur Verfügung gestellt. Die rRNA–Alignments wurden, basierend auf bekannten Sekundärstrukturdaten, von Hand erstellt. Auch das von ihm zur Verfügung gestellte Aktin–Alignment wurde manuell anhand von bekannten Struktureigenschaften von Aktin angefertigt.

Die vorhandenen Alignments wurden vor der Benutzung noch einmal überprüft und an wenigen Stellen manuell korrigiert.

3.1.1 Datensätze zur Laufzeituntersuchung

Die zum Testen von `pfastDNAml` verwendeten Alignments stammen aus verschiedenen Quellen. Die Datensätze *rrna10* und *rrna20* sind dem Alignment des Ribosomal Database Projects (RDP, Olsen *et al.* 1992) entnommen. Der Datensatz *rrna28* stammt aus der rRNA–Datenbank in Antwerpen (Van de Peer *et al.*, 1994).

3.1.2 Aktinsequenzen

Aus dem vorhandenen Aktin Alignment wurden unterschiedliche Zusammenstellungen von Organismen ausgewählt. Aus dem gesamten Alignment wurden folgende Organismen und Gensequenzen verwendet:

1. Tiere (Metazoa)
    - *Aplysia californica*
    - *Bombyx mori* A1, A2 (Seidenspinner)
    - *Drosophila melanogaster* (Fruchtfliege)
    - *Homo sapiens* A1, A2, B2 (Mensch)
    - *Hydra attenuata*
    - *Molgula citrina*
    - *Onchocercus volvulus* 1A
    - *Strongylocentrotus purpuratus* (Purpurseeigel)
2. Pilze
    - *Absidia glauca*
    - *Cryptococcus neoformans*
    - *Kluyveromyces lactis*
    - *Puccinia graminis*
    - *Saccharomyces cerevisiae* (Bierhefe)
    - *Schizosaccharomyces pombe*
    - *Thermomyces lanuginosis*
    - *Trichoderma reesei*
3. Pflanzen (Landpflanzen)
    - *Arabidopsis thaliana* Ac1
    - *Daucus carota* Ac1, Ac2 (Karotte)
    - *Oryza sativa* Ac1, Ac2, Ac3 (Reis)
    - *Solanum tuberosum* 75, 97, 101 (Kartoffel)
    - *Glycine max* Ac1, Ac3 (Soja)
    - *Zea mays* Ac1 (Mais)

4. Protista
   (a) Haptophyta (auch Prymnesiophyta)
      - *Emiliania huxleyi*
   (b) Chlorophyta (Grünalgen)
      - *Chlamydomonas reinhardtii*
      - *Scherffelia dubia*
      - *Volvox carteri*
   (c) Glaucocystophyta
      - *Cyanophora paradoxa*
   (d) Rhodophyta (Rotalgen)
      - *Chondrus crispus*
      - *Cyanidioschyzon merolae*
   (e) Heterokontophyta
      - *Achlya bisexualis*
      - *Costaria costata*
      - *Fucus distichus*
      - *Lagenidium giganteum* Ac1, Ac2
      - *Phytophthora infestans* actA, actB
      - *Phytophthora megasperma*
      - *Pythium irregulare* Ac1, Ac2
   (f) Ciliaten
      - *Euplotes crassus*
      - *Oxytricha fallax*
      - *Tetrahymena thermophila*
   (g) andere Protisten
      - *Acanthamoeba castellanii*
      - *Dictyostelium discoideum*
      - *Entamoeba histolytica*
      - *Naegleria gruberi*
      - *Physarum polycephalum*
      - *Plasmodium falciparum*
5. Außengruppe

- *Giardia lamblia* (Protozoon ohne Mitochondrien)
- *Toxoplasma gondi*
- *Trypanosoma brucei*

3.1.3 Eukaryotische 18S rRNA–Sequenzen

Aus dem vorhandenen 18S rRNA Alignment wurden unterschiedliche Zusammenstellungen von Organismen ausgewählt. Aus dem gesamten Alignment wurden folgende Organismen verwendet:

1. Tiere (Metazoa)
   - *Anemonia sulcata*
   - *Artemia salina* (Salinenkrebs)
2. Pilze
   - *Aethelia bombacina*
   - *Aureobasidium pullulans*
   - *Saccharomyces cerevisiae* (Bierhefe)
3. Pflanzen (Landpflanzen)
   - *Zamia pumila*
4. Protista
   (a) Ciliata
      - *Oxytricha nova*
      - *Paramecium tetraurelia* (Pantoffeltierchen)
   (b) Heterokontophyta (auch Chrysophyta)
      - *Achlya bisexualis*
      - *Fucus distichus*
      - *Labyrinthuloides minuta*
      - *Ochromonas danica*
   (c) Haptophyta (auch Prymnesiophyta)
      - *Emiliania huxleyi*
      - *Phaeocystis antarctica*

- *Phaeocystis globosa*

(d) Rhodophyta (Rotalgen)

- *Glaucosphaera vacuolata*
- *Porphyra umbilicalis*
- *Porphyridium aerugineum*

(e) Filose Amöben

- *Euglypha rotunda*
- *Paulinella chromatophora*

(f) Chlorarachniophyta

- *Chlorarachnion* sp. 1
- *Chlorarachnion reptans*

(g) Cryptophyta

- *Goniomonas truncata*
- *Guillardia theta*
- *Rhodomonas mariana*
- *Rhodomonas salina*

(h) Glaucocystophyta

- *Cyanophora paradoxa*
- *Glaucocystis nostochinearum*
- *Gloeochaete wittrockiana*

(i) Chlorophyta (Grünalgen)

- *Mantoniella squamata*
- *Staurastrum M752*

(j) Sonstige Protisten

- *Acanthamoeba castellanii*
- *Hartmannella vermiformis*
- *Prorocentrum micans*
- *Sarcocystis muris*

5. Außengruppe

- *Dictyostelium discoideum*

3.1.4 16S rRNA–Sequenzen aus Plastiden/Bakterien

Aus dem vorhandenen 16S rRNA Alignment wurden unterschiedliche Zusammenstellungen von Plastiden und Bakterien ausgewählt. Aus dem gesamten Alignment wurden folgende Organismen verwendet.

1. Plastiden aus:
   (a) Heterokontophyta (auch Chrysophyta) – komplexe Plastiden
      - *Corethron criophilum*
      - *Heterosigma akashiwo*
      - *Pylaiella littoralis*
      - *Skeletonema costatum*

   (b) Haptophyta (auch Prymnesiophyta) – komplexe Plastiden
      - *Emiliania huxleyi*
      - *Ochrosphaera neapolitana*
      - *Pavlova cf. salina*

   (c) Cryptophyta – komplexe Plastiden
      - *Guillardia theta*
      - *Rhodomonas salina*

   (d) Rhodophyta (Rotalgen) – Rhodoplasten
      - *Antithamnion* sp.
      - *Chondrus crispus*
      - *Cyanidium caldarium*
      - *Galdieria sulphuraria*
      - *Glaucosphaera vacuolata*
      - *Palmaria palmata*
      - *Porphyridium aerugineum*

   (e) Glaucocystophyta – Cyanellen
      - *Cyanophora paradoxa* (Kies)
      - *Cyanophora paradoxa* (Pringsheim)
      - *Glaucocystis nostochinearum*
      - *Gloeochaete wittrockiana*

   (f) Grünalgen – Chloroplasten
      - *Chara* sp. (Jochalge)

- *Chlorella ellipsoidea*
- *Chlorella vulgaris*
- *Closterium ehrenbergii* (Jochalge)

(g) Pflanzen – Chloroplasten

- *Glycine max* (Soja)
- *Marchantia polymorpha* (Brunnenlebermoos)
- *Zea mays* (Mais)

2. Cyanobakterien
   - *Anabaena* sp.
   - *Chamaesiphon subglosus*
   - *Chroococcidiopsis* sp.
   - *Gloeobacter violaceus*
   - *Phormidium minutum*
   - *Prochlorococcus* sp.
   - *Prochloron didemni*
   - *Synechococcus* sp.
3. Andere Bakterien
   - *Alcaligenes faecalis*
   - *Escherichia coli*
   - *Pseudomonas andropogonis*
4. Außengruppe
   - *Agrobacterium tumefaciens*

3.2 Benutzte Software

3.2.1 fastDNAml

Als Grundlage für die Parallelisierung diente das Programm `fastDNAml` von Olsen, Matsuda, Hagstrom und Overbeek (1994b). Verwendet wurde der im Internet frei verfügbare sequentielle C–Code in der Version 1.0.8.

Das Programm ist erhältlich unter folgender URL:

`ftp://rdp.life.uiuc.edu/pub/RDP/programs/fastDNAml`

3.2.2 PHYLIP

Für das Bearbeiten und Auswerten unterschiedlichster Datensätze und Ergebnisse dienten verschiedene Programme aus dem Programmpaket PHYLIP (**Phyl**ogeny **I**nference **P**ackage) in der Version 3.5 von Felsenstein (1993). Es wurden folgende Programme verwendet:

`seqboot` zum Erstellen von Bootstrap–Datensätzen.

`drawgram` zur graphischen Darstellung gewurzelter Bäume

`drawtree` zur graphischen Darstellung ungewurzelter Bäume

`retree` zum Umarrangieren von Stammbaumdarstellungen

`consense` zum Auswerten von Bootstrap–Ergebnissen

Die Handhabung der Programme kann im PHYLIP–Handbuch (Felsenstein, 1993) nachgelesen werden.

Das PHYLIP–Programmpaket findet man im Internet unter folgender URL:

`ftp://evolution.genetics.washington.edu/pub/phylip`.

3.2.3 treetool

Ebenfalls zur Darstellung von phylogenetischen Stammbäumen diente das Programm `treetool` URL:

`ftp://rdp.life.uiuc.edu/pub/RDP/programs/TreeTool`)

3.2.4 MacClade

Zum Berechnen von Gewichten für einzelne Spalten im Sequenzalignment wurde das Programm `MacClade` (Maddison und Maddison, 1992) verwendet. Dieses Programm läuft auf Apple MacIntosh.

3.2.5 PARMACS

Als Grundlage für die Parallelisierung von `fastDNAml` diente die C–Befehlsbibliothek (C library) PARMACS[1] der Firma Pallas GmbH, Brühl, in der Version 6.0 bzw. 6.1.

Es handelt sich bei PARMACS um eine Bibliothek mit Befehlen zur Parallelisierung auf Basis von *explicit message passing*. Die PARMACS–Bibliothek ist erhältlich für eine große Anzahl von Parallelen Rechnerplattformen und ermöglicht so ein großes Maß an Portabilität.

3.3 Geräte

3.3.1 Workstations und Workstationcluster

Das Programm `fastDNAml` wurde von Kerningham–Richie–C nach ANSI–C auf einem Pentium90–PC unter Linux (Version 1.2.13) portiert.

Die anschließende Parallelisierung mit PARMACS wurde auf einer SUN–Workstation unter SunOs 4.1.3 durchgeführt.

3.3.2 Parallelrechner

Die Stammbaumrekonstruktionen wurden auf einer IBM SP2 mit 34 Knoten durchgeführt. Als Scheduler zum Zuweisen von Rechenzeiten an die Benutzer lief auf dieser Maschine das Programmpaket EASY von Argonne National Laboratories.

Außerdem wurde das Programm `pfastDNAml` auf einer NEC Cenju-3 mit 64 Knoten auf diese Plattform portiert und getestet.

Auf den Computern wurde mir freundlicherweise von der GMD, Sankt Augustin, und der Firma NEC im Rahmen des Projekts PARAPHYL Rechenzeit zu Verfügung gestellt.

Da der Zugriff auf die NEC Cenju-3 erst im April 1996 möglich wurde, konnten aus Zeitgründen keine größeren Berechnungen auf diesem Computer durchgeführt werden.

[1] **Par**allel **Mac**ros

Kapitel 4

Methoden, Modelle und Algorithmen

4.1 Modellierung evolutionärer Prozesse

4.1.1 Markov–Prozesse

Um den Prozeß von Basensubstitutionen in der DNA im Laufe der Evolution beschreiben zu können, werden im allgemeinen reversible Markov–Prozesse herangezogen. Bei Markov–Prozessen handelt es sich um stochastische Prozesse, für deren Verhalten in der Zukunft lediglich die Werte in der Gegenwart, nicht aber die in der Vergangenheit eine Rolle spielen. Die Wahrscheinlichkeit, daß ein bestimmter Zustand zu einem Zeitpunkt $t > t_m$ eintritt, wenn die Zustände zu den Zeitpunkten $t_0 < t_1 < \ldots < t_m$ bekannt sind, ist bei einem Markov–Prozeß gleich groß wie die Wahrscheinlichkeit des Eintretens des Zustands zum Zeitpunkt $t > t_m$, wenn nur der Zustand zum Zeitpunkt t_m bekannt ist.

Dieses bedeutet für den Evolutionsprozeß, daß Basensubstitutionen zufällig passieren, unabhängig davon welche Mutation zuletzt aufgetreten ist.

Bei einem reversiblen Markov–Prozeß gilt noch zusätzlich, daß die Wahrscheinlichkeit $P_{s_1 s_2}(\Delta t)$, mit der ein Zustand s_1 nach einer Zeitspanne $\Delta t = t - t_m$ mit $t > t_m$ in einen Zustand s_2 übergegangen ist, gleich groß ist wie die Wahrscheinlichkeit $P_{s_2 s_1}(\Delta t)$, mit der der Zustand s_2 nach derselben Zeitspanne Δt in einen Zustand s_1 übergegangen ist.

Die Annahme eines reversiblen Markov–Prozesses ist notwendig, um mit ungewurzelten Bäumen rechnen zu können. Außerdem konstruieren wir die

Stammbäume von bekannten, rezenten Sequenzen in die Vergangenheit, was beim herkömmlichen Markov–Prozeß nicht unterstützt wird. (Felsenstein, 1981; Schneider, 1991)

4.1.2 Generalisiertes 2–Parameter–Modell

Es gibt unterschiedliche Modelle, mit denen der evolutionäre Prozeß beschrieben wird. Das hier verwendete Modell ist das generalisierte 2–Parameter–Modell von Kishino und Hasegawa (1989). Es wird im Programmpaket PHYLIP seit Version 2.6 verwendet und wurde mit in das Programm `fastDNAml`, basierend auf PHYLIP Version 3.3, übernommen. Dieses Modell findet auch unverändert in `pfastDNAml` Anwendung.

Das generalisierte 2–Parameter–Modell modelliert den Evolutionsprozeß als reversiblen Markov–Prozeß (s. Kap. 4.1.1). Das Modell liefert uns eine Wahrscheinlichkeitsmatrix

$$P(t) = e^{tR}, \tag{4.1}$$

aus der die Wahrscheinlichkeit eines Basenaustausches in einem Zeitraum t entnommen werden kann. Bei R handelt es sich um die Ratenmatrix bzw., um den wahrscheinlichkeitstheoretischen Terminus zu gebrauchen, Generatormatrix, in der die einzelnen Substitutionsraten r_{ij} von einer Base i in eine Base j enthalten sind. Dieses r_{ij} errechnet sich wie folgt:

$$r_{ij} = \begin{cases} (k/F_j + 1)\beta f_j & \text{für Transitionen} \\ \beta f_j & \text{für Transversionen .} \end{cases} \tag{4.2}$$

Dabei gibt f_j die Basenfrequenz der Base j, F_j die Frequenz des Basentyps (Purin oder Pyrimidin) an. β entspricht der Transversionsrate pro Base, während k die Rate Transition : Transversion angibt.

Die Ratenmatrix lautet also wie folgt:

$$\begin{bmatrix} -(f_C+Y_G\cdot f_G+f_T)\beta & \beta f_C & Y_G\cdot\beta f_G & \beta f_T \\ \beta f_A & -(f_A+f_G+Y_T\cdot f_T)\beta & \beta f_G & Y_T\cdot\beta f_T \\ Y_A\cdot\beta f_A & \beta f_C & -(Y_A\cdot f_A+f_C+f_T)\beta & \beta f_T \\ \beta f_A & Y_C\cdot\beta f_C & \beta f_G & -(f_A+Y_C\cdot f_C+f_G)\beta \end{bmatrix} \tag{4.3}$$

Hierbei entsprechen

$$Y_A = Y_G = \frac{k}{f_A + f_G} + 1 \tag{4.4}$$

$$Y_C = Y_T = \frac{k}{f_C + f_T} + 1 \tag{4.5}$$

Die Matrizen sind so angelegt, daß sich die Inhalte einer Zeile der Generatormatrix R zu 0 und die Inhalte der Wahrscheinlichkeitsmatrix P zu 1 (= 100%) aufaddieren.

Die Wahrscheinlichkeitsmatrix $P(t)$ wird in der nachfolgend beschriebenen Maximum–Likelihood–Methode verwendet, um die Wahrscheinlichkeit $P_{ij}(t)$ für einen Basenaustausch von i nach j in einem Zeitraum t zu berechnen.

Vorteil der Modellierung des Evolutionsprozesses mit Matrizen ist es, daß bei der Berechnung der Wahrscheinlichkeit einer Basensubstitution alle möglichen Übergangszustände automatisch mit berücksichtigt werden. D.h. es werden auch die Wahrscheinlichkeiten der verschiedenen Zwischenzustände mit berücksichtigt.

4.2 Die Maximum–Likelihood–Methode zur Rekonstruktion von Stammbäumen

Der verwendete ML–Ansatz wurde von Felsenstein (1981) eingeführt und in dem Programm `dnaml` innerhalb des PHYLIP–Programmpaketes implementiert. Änderungen an dem ursprünglichen Algorithmus sind in den entsprechenden Handbüchern (Felsenstein, 1990; Olsen *et al.*, 1994b) und Artikeln (Olsen *et al.*, 1994a) erwähnt und werden bei der folgenden Beschreibung von vornherein berücksichtigt. Weitere Informationen über die ML–Methode stammen aus Kreyszig (1975), Goldman (1990) und Swofford und Olsen (1990)

4.2.1 Berechnung des Likelihood–Wertes eines Baumes

Für die Berechnung des Likelihood–Wertes benötigen wir ein Evolutionsmodell und eine hieraus resultierende Wahrscheinlichkeitsfunktion $P_{ij}(t)$. Hier wird das in Kap. 1.4.2 beschriebene generalisierte 2–Parameter–Modell von Kishino und Hasegawa (1989) verwendet. Als Stichprobe zur Berechnung dient ein vorgegebener Datensatz in Form eines Alignments der DNA–Sequenz. Die einzelnen Stichproben sind hier die Spalten des Alignments. Zusätzlich wird ein gegebener Stammbaum als Hypothese (Goldman, 1990) in der Analyse verwendet.

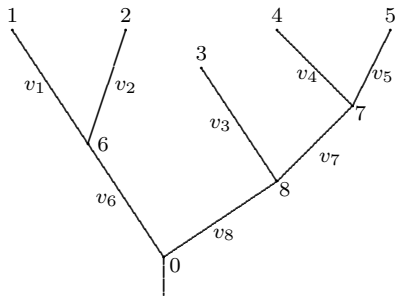


Abbildung 4.1: Beispielbaum für die Berechnung von Likelihood–Werten phylogenetischer Stammbäume

Wie schon oben erwähnt, gibt die Funktion $P_{ij}(t)$ die Wahrscheinlichkeit an, mit der eine Base aus dem Zustand i in einer Zeitspanne t in einen Zustand j übergeht. Die Zustände i und j entsprechen dabei einer der vier organischen Basen Adenin (A), Cytosin (C), Guanin (G) oder Thymin (T) bzw. Uracil (U) bei RNA–Sequenzen.

Die Berechnung des Likelihood–Wertes für eine Phylogenie soll hier anhand eines Beispielstammbaumes (Abb. 4.1) gezeigt werden.

Zuerst wird der Likelihood–Wert eines Baumes für eine Spalte x im Sequenzalignment berechnet. Dieses geschieht rekursiv; daher definieren wir für die Berechnung des Likelihood–Wertes eines Teilbaumes, der mit dem Knoten k innerhalb der Phylogenie mit dem Sequenzstatus s_k und den Nachfolgern i und j mit deren möglichen Sequenzstatus s_i und s_j beginnt, wie folgt:

$$L_{s_k}^{(k)}(x) = \left(\sum_{s_i} P_{s_k s_i}(v_i) L_{s_i}^{(i)} \right) \left(\sum_{s_j} P_{s_k s_j}(v_j) L_{s_j}^{(j)} \right). \tag{4.6}$$

Handelt es sich bei dem Knoten k um ein Blatt im Stammbaum, so ist der Likelihood–Wert, d.h. die Wahrscheinlichkeit, daß k an dieser Stelle der Sequenz die Base s_k besitzt, wie folgt definiert:

$$L_{s_k}^{(k)}(x) = \begin{cases} 1, \text{ falls } s_k \text{ im Alignment vorgegeben} \\ 0 \text{ sonst} \end{cases} \tag{4.7}$$

Mit den Gleichungen 4.6 und 4.7 wird die Wahrscheinlichkeit berechnet, daß sich an dieser Sequenzposition des Organismus k die Base s_k befindet bzw. befunden hat.

Hieraus ergibt sich für die Berechnung des Likelihood–Wertes des gesamten Stammbaumes an dieser Sequenzposition:

$$L^{(0)}(x) = \sum_{s_0} L_{s_0}^{(0)}. \tag{4.8}$$

Zur Berechnung des Likelihood–Wertes des Baumes über die gesamte Stichprobe, bestehend aus den einzelnen Proben $x_1, x_2, \ldots, x_n$, werden die Likelihood–Werte für alle n Spalten im Alignment miteinander multipliziert:

$$L = \prod_{x_n} L^{(0)}(x_n). \tag{4.9}$$

Wir verwenden anstelle des Likelihood–Wertes L dessen natürlichen Logarithmus, den sogenannten Log–Likelihood:

$$\ln L = \sum_{x_n} \ln L^{(0)}(x_n). \tag{4.10}$$

Dadurch wird das Differenzieren von Produkten aufgrund der Rechenregeln für Logarithmen

$$\begin{aligned} \ln(gh) &= \ln g + \ln h & (4.11)\\ \ln \frac{g}{h} &= \ln g - \ln h & (4.12)\\ \ln g^r &= r \ln g & (4.13)\\ \ln \sqrt[r]{g} &= \frac{1}{r} \ln g & (4.14) \end{aligned}$$

(mit $g, h \in \mathbb{R}_+^*$ und $r \in \mathbb{N}$) durch das im allgemeinen einfachere Differenzieren von Summen ersetzt.

Durch die Gleichungen 4.9 bzw. 4.10 wird die Wahrscheinlichkeit berechnet, mit der die verwendeten Sequenzdaten den vorliegenden Baum mit seinen Kantenlängen unterstützen.

Die oben beschriebene Berechnung hängt von uns unbekannten Parametern, den Kantenlängen, ab. Sinn der Maximum–Likelihood–Methode ist, solche unbekannten Parameter anhand einer Stichprobe zu schätzen. Betrachten wir nun die Stichprobe als fest, dann hängt die Likelihood-Funktion nur noch von den Kantenlängen ab. Diese sollen so approximiert werden, daß der Likelihood–Wert des vorgegebenen Baumes sein Maximum erreicht. Wie diese Parameter geschätzt werden, wird nachfolgend beschrieben.

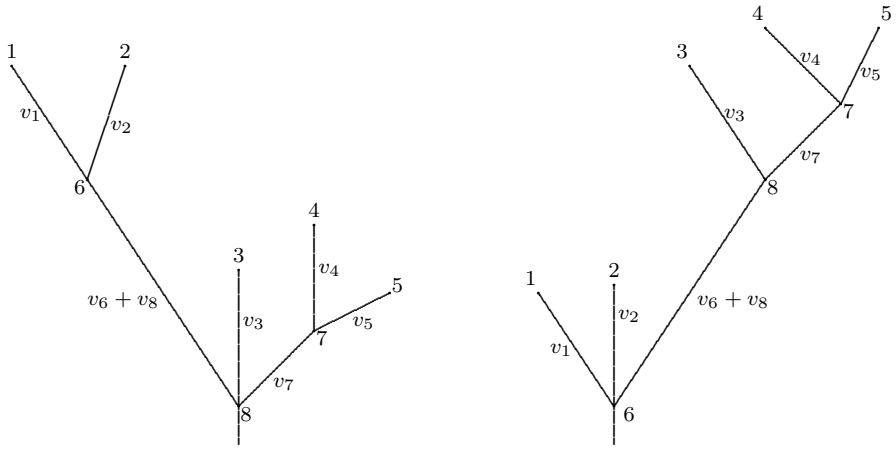


Abbildung 4.2: Beide Bäume haben, da das Pulley–Prinzip gilt, den gleichen Likelihood–Wert wie der Baum in Abb. 4.1.

4.2.2 Das Pulley–Prinzip

Aufgrund der Reversibilität der hier benutzten Markov–Prozesse (Kap. 4.1.1), gilt das sogenannte Pulley–Prinzip[1]. Das Pulley–Prinzip besagt, daß man die eine von der Wurzel des Baumes abgehende Kante verkürzen kann, wenn man die jeweils andere um den gleichen Betrag verlängert, ohne daß sich dadurch der Likelihood–Wert des Baumes verändert (Fig. 4.2). Hieraus ergibt sich letztlich, daß man die Wurzel überall auf den Kanten des Baumes verschieben kann, ohne daß der Likelihood–Wert beeinflußt wird. Diese Eigenschaft ermöglicht uns später wichtige Vereinfachungen bei der Berechnung der Kantenlängen.

Der Beweis für das Pulley–Prinzip kann im Artikel über den ML–Ansatz von Felsenstein (1981) nachgelesen werden.

4.2.3 Finden der optimalen Kantenlängen

Jetzt wird der eigentliche Teil der ML–Methode benutzt. Es wird versucht, die Kantenlängen der vorgegebenen Phylogenie so zu schätzen, daß der Likelihood–Wert sein Maximum erreicht.

[1]Pulley (engl.) – Flaschenzug

Hierzu optimiert man jede Kante v einzeln, indem man die Wurzel unmittelbar an einen der an diese Kante v angrenzenden Knoten verschiebt. Die anderen Kanten werden während dieser Optimierung festgehalten.

Betrachtet man den Likelihood–Wert als Funktion dieser Kante, so muß an der Stelle des Maximums die partielle Ableitung nach dieser Kante

$$\frac{\partial L}{\partial v} = 0 \tag{4.15}$$

ergeben.

Früher wurde zum Auffinden des Maximums der Expectation–Maximization–Algorithmus (EM–Algorithmus) verwendet. Dieser wurde jedoch aus Zeitersparnisgründen durch die Newton–Raphson–Methode ersetzt (Olsen *et al.*, 1994a).

Die Newton–Raphson–Methode (oder Newton–Methode) wird benutzt, um den Schnittpunkt einer Funktion mit der x–Achse mit Hilfe der ersten und zweiten Ableitung zu finden.

Geometrisch betrachtet, berechnet man an einem Startpunkt die Tangente zur Funktion, in unserem Fall die oben genannte partielle Ableitung. Man findet zu dieser Tangente den x–Achsenabschnitt und beginnt an diesem Punkt wieder mit der Tangente, bis die Nullstelle gefunden wurde. Die Newton–Raphson–Methode konvergiert quadratisch und ist daher ein schnelleres Verfahren. Diese Geschwindigkeit wird allerdings durch gewisse Fehlermöglichkeiten erkauft. (Press *et al.*, 1988)

Auch im PHYLIP–Programmpaket wurde inzwischen der EM–Algorithmus durch die Newton–Raphson–Methode ersetzt, was für die Akzeptanz dieser Methode spricht.

4.2.4 Suche nach dem optimalen Baum

Wie schon in der Einleitung erwähnt, müßte an sich der Maximum–Likelihood–Wert für jeden möglichen Baum berechnet werden. Der Baum mit dem höchsten Log–Likelihood–Wert ist dann derjenige, der die Entstehung des vorgegebenen Sequenzen durch Evolution unter dem benutzten Modell am besten unterstützt. Aufgrund der hohen Anzahl von möglichen Bäumen ist eine erschöpfende Suche[2] nicht praktikabel. Daher wird der heuristische Ansatz des schrittweisen Hinzufügens gewählt.

[2]Exhaustive Search

Bei diesem Ansatz wird, beginnend bei einem Baum mit drei Sequenzen, die Baumtopologie mit dem höchsten ML–Wert bestimmt. In diesen wird dann die nächste Sequenz in alle Kanten eingefügt und aus diesen Bäumen wieder der mit dem höchsten ML–Wert weiterverwendet. Dies wird so lange wiederholt, bis alle Sequenzen in den Baum eingebunden sind.

Um das Risiko zu mindern, daß man durch den heuristischen Ansatz den besten Baum gar nicht erst findet, wurden bei allen in dieser Diplomarbeit verwendeten Analysen, ab einer Baumgröße von vier Sequenzen, lokale Rearrangements über die Länge einer Kante durchgeführt; d.h. jeder Teilbaum des optimalen Baumes wird herausgeschnitten und in alle benachbarten Kanten eingefügt. Findet sich hierbei ein neuer optimaler Baum, so wird ein erneutes lokales Rearrangement ausgeführt, bis kein neues Optimum mehr gefunden wird. Erst dann wird die nächste Sequenz eingefügt.

Sind alle Sequenzen in den Baum eingefügt, wird, ausgehend vom gefundenen Optimum, ein globales Rearrangement durchgeführt. Dieses verläuft genau wie das lokale Rearrangement, allerdings mit dem Unterschied, daß die ausgeschnittenen Teilbäume in jede andere Kante eingefügt werden. Auf diese Weise wird eine große Anzahl an potentiellen Bäumen getestet, die eventuell durch die reine Heuristik übersehen worden wären.

4.3 Erstellen von gewichteten Datensätzen

Um eventuelle unerwünschte Effekte zu vermeiden, die durch hochvariable Bereiche in den Eingabesequenzen auftreten können, wurden teilweise gewichtete Datensätze erstellt. Gesucht wurden Gewichte für die einzelnen Spalten im Sequenzalignment, so daß hochvariable Spalten heruntergewichtet werden, während konserviertere Basen im Alignment ein höheres Gewicht erhalten.

Dazu wurde die Möglichkeit des Programms `MacClade` genutzt, Weightsets zu generieren (Maddison und Maddison, 1992). Dem Programm wurde ein ML–Stammbaum als Grundlage für die Gewichtung übergeben. `MacClade` generiert dann nach der Parsimony–Methode Sequenzen für die internen Knoten. Dieses geschieht so, daß möglichst wenig Mutationen für die Realisierung des Baumes nötig sind (siehe Kap. 1.4.3). Anschließend werden die notwendigen Mutationen in jeder Alignmentspalte gezählt. Die Anzahl der Mutationen S (Parsimony–Steps) ist die Grundlage zur Berechnung der Gewichte. Als Gewichtungsfunktion W wurde

$$W = \frac{1}{S} \tag{4.16}$$

verwendet. Durch Verwendung der Option *integral weights* werden die einzelnen Werte W mit der gewünschen Anzahl an Gewichten multipliziert und anschließend auf den nächsten ganzzahligen Wert gerundet. Dieser Wert entspricht dem errechneten Gewicht. Spalten mit $S = 0$, d.h. ohne Mutationen, erhalten ebenfalls das geringste Gewicht, da sie für die Stammbaumanalysen keine Aussagekraft besitzen. Es wurden im allgemeinen 10 Gewichte gewählt, die später bei der Analyse mit `pfastDNAml` jeweils einer Gewichtskategorie zugeordnet wurden.

4.4 Bootstrap–Verfahren

Das in dieser Arbeit verwendetes Verfahren zum Testen der Aussagekraft der errechneten Bäume ist das von Efron (1979) entwickelte Bootstrap–Verfahren. Das Bootstrap–Verfahren wird verwendet, um aus den vorhandenen Daten neue unabhängige Stichproben zu gewinnen (sog. Pseudo–Stichproben). Dies geschieht durch zufälliges Ziehen neuer Stichproben aus einer vorhandenen. Es wird so lange gezogen, bis wieder die Gesamtzahl an Einzelstichproben erreicht ist. Dabei werden einige Einzelstichproben aus der Originalstichprobe nicht mehr, andere dafür mehrfach in der Pseudostichprobe vorhanden sein. Diese Methode wurde von Neyman (1971) zum Testen phylogenetischer Stammbäume vorgeschlagen und von Felsenstein (1985) umgesetzt. Hierbei werden die Organismen konstant gehalten, während die Alignmentspalten neu gezogen werden.

Zum Gewinnen von Bootstrap–Stichproben wurden zwei Verfahren angewendet:

1. Erstellen von Bootstrap–Stichproben durch das PHYLIP–Programm `seqboot` (Felsenstein, 1993)

2. Nutzung der Bootstrap–Option `B` innerhalb von `pfastDNAml` und damit verbunden die Angabe eines Bootstrap–Random–Number–Seed zur Berechnung einer Reihe von Zufallszahlen.

Es wurden immer 100 unabhängige Bootstrap–Stichproben generiert. Hundert Bootstrap–Stichproben sind nach Hedges (1992) die minimale Anzahl, die nötig ist, um einen aussagekräftigen P–Wert von 95% erhalten zu können.

Zur Auswertung dieser 100 Stichproben wurde das PHYLIP–Programm `consense` verwendet (Felsenstein, 1993), das aus den errechneten Bäumen einen Konsensusbaum nach dem Majoritätsprinzip generiert und zu jedem

Teilbaum den Bootstrap–Wert angibt, mit welcher prozentualen Häufigkeit dieser Teilbaum in allen Bäumen gefunden wurde.

Kapitel 5

Parallelisierung und Laufzeituntersuchung

5.1 Ergebnisse

5.1.1 Implementierung

ANSI–C–Portierung und Strukturierung von `fastDNAml`

Grundlage für die Parallelisierung war das Programm `fastDNAml` von Olsen *et al.* (1994a,b). Dieses ist frei im Internet als Quellcode verfügbar. Startpunkt war der reine sequentielle Code, der gewonnen wurde, indem alle Teile des Quellcodes, die für den sequentiellen Ablauf nicht nötig waren, gelöscht wurden. Dies betraf in erster Linie Anpassungen im Originalcode, die von einer früheren Parallelisierung durch Matsuda *et al.* (1994) herrührten. Um diese Parallelisierung lauffähig zu machen, hätte es erstens weiterer Programmteile bedurft, die jedoch im regulären `fastDNAml`–Programmpaket nicht enthalten sind, und zweitens einer lauffähigen Installation der P4-Funktionsbibliothek, die von den Entwicklern am Argonne National Laboratory (ANL) nicht mehr weiterentwickelt und gepflegt wird.

Lediglich die Erweiterungen für Vektor–Rechner blieben zusätzlich im Quellcode erhalten.

Seit 1989 existiert der ANSI–Standard für die Programmiersprache C, der ein hohes Maß an Portabilität gewährleistet, da seit der Einführung für nahezu alle Rechner– und Betriebssystemtypen ANSI–C–Compiler implementiert sind. Hierdurch ist gewährleistet, daß, wenn man sich an diesen Standard

hält, die Programme auf nahezu allen Systemen lauffähig, d.h. portabel sind. (ANSI, 1989; Harbison und Steele Jr., 1991)

`fastDNAml` ist im Original in Kernighan&Ritchie–C (K&R–C, nach Kerninghan und Ritchie 1988, die Anfang der 70er Jahre die Programmiersprache C entwickelt haben) geschrieben, das auf verschiedenen Computerplattformen immer mit anderen „Dialekten“ implementiert ist und deshalb eine portable Programmierung zunehmend erschwert.

Daher wurde anschließend der reine sequentielle Code zuerst nach ANSI–C portiert (Harbison und Steele Jr., 1991). Nach der Portierung wurde der so entstandene Code bzw. das daraus resultierende Programm daraufhin überprüft, ob es immer noch die gleichen Ergebnisse wie das Originalprogramm liefert.

Nachdem dieses gewährleistet war, wurde der Inhalt des Quellcodes auf seine Funktion hin untersucht, um den Code und dessen Funktion zu verstehen. Bei dieser Analyse wurde der Quellcode in verschiedene Module geteilt, um eine bessere Übersicht zu erhalten, welche Teile für welche Funktionen codieren.

Um das Compilieren (Übersetzen in ein dem Computer verständliches Format) der einzelnen Module zu vereinfachen, wurde ein `Makefile` erstellt. Danach kann der Compilierungsvorgang mit einem vor allem in der UNIX–Welt verbreiteten Programm namens `make` gesteuert werden, was ebenfalls der Portabilität von Software entgegenkommt, da weitreichende Anpassungen durch Ändern sogenannter Schalter innerhalb des `Makefiles` gesteuert werden können. (Oram und Talbott, 1991)

Die entstandene Modulstruktur der Parallelversion wird im Anhang A.6 beschrieben.

Da im allgemeinen auf leistungstärkeren Rechnern, genauso wie auf Parallelcomputern, das übliche Betriebsystem UNIX ist, wurden bei der Implementierung von `pfastDNAml` die Voraussetzungen von UNIX besonders berücksichtigt, soweit diese nicht vom ANSI–C–Standard abwichen.
(Curry, 1989; Gilly, 1992; ANSI, 1989; Harbison und Steele Jr., 1991)

Struktur von `fastDNAml`

Zunächst wurden nun anhand des reinen Quellcodes die Programmstruktur sowie der Aufbau der Berechnungen rekonstruiert. Bei dieser Untersuchung der ANSI–C–Version des Originalcodes mußte jede einzelne implementierte Prozedur und Routine einzeln untersucht werden, um die Funktionsweise des

Programms zu verstehen und die Möglichkeiten zum Ansatz für die Parallelisierung zu erkennen. Die Funktion jeder einzelnen Routine zu beschreiben, würde den Rahmen dieser Arbeit sprengen, denn schon der Ausdruck des Quellcodes umfaßt mehr als sechzig Seiten. Daher wird hier nur der interne Ablauf der Stammbaumrekonstruktion insoweit beschrieben, daß die Eingriffspunkte zur Parallelisierung und deren Folgen dargelegt werden können.

Die durchgeführte Analyse der Programmstruktur ergab folgenden Ablauf einer „normalen“ Stammbaumrekonstruktion:

1. Zuerst wird vom Programm die Anzahl der Spezies und Alignmentspalten sowie alle Optionen und Parameter eingelesen. Anhand dieser Daten werden die notwendigen Speicherbereiche reserviert. Wurden die Basenfrequenzen als Parameter übergeben, werden hier die verschiedenen Frequenzen der einzelnen Basen und Basentypen initialisiert. Anschließend werden noch die Sequenzdaten eingelesen.

2. Jetzt werden angegebene Kategorien, Gewichte und eventuell zu berechnende Bootstrap–Stichproben für jede Alignmentspalte in ein Gewicht umgerechnet und dieser zugeordnet. Sind die Frequenzen nicht explizit angegeben worden, werden sie jetzt anhand der Sequenzdaten bestimmt.

3. Jetzt wird die Initialisierung der Baumstrukturen abgeschlossen, soweit nicht schon geschehen.

4. Das Programm startet nun mit drei Sequenzen und konstruiert hieraus einen einfachen Stammbaum. Dann werden für diesen Stammbaum die Kantenlängen optimiert.

5. In den im vorhergehenden Schritt gefundenen optimalen ML–Baum wird nun nacheinander an jede Kante die nächste Spezies eingefügt. Jeder so entstandene Baum wird anschließend in Bezug auf seine Kanten optimiert. Am Ende jeder Berechnung wird der neu gefundene Baum mit dem jeweils letzten optimalen Baum dieses Schrittes verglichen und der bessere der beiden gespeichert.

6. Der gefundene optimale Baum wird in der `checkpoint`–Datei gespeichert. Enthält dieser Baum weniger als vier Spezies, so fährt das Programm bei Punkt 5. mit der Berechnung fort. In jedem anderen Fall fährt das Programm an Punkt 7. fort.

7. Mit dem bisher gefundenen besten Baum wird jetzt ein Rearrangement durchgeführt. D.h. nacheinander wird jeder Teilbaum aus dem Baum ausgeschnitten und testweise in alle Nachbarkanten mit der Entfernung eingefügt, die in der Eingabedatei angegeben wurde (Standard ist 1). Für alle diese Bäume werden die Kanten optimiert. Am Ende jeder Berechnung wird der neu gefundene Baum mit dem jeweils letzten optimalen Baum dieses Schrittes verglichen und der bessere der beiden gespeichert.

8. Wurde im letzten Schritt ein neuer optimaler Baum gefunden, wird dieser in die `checkpoint`–Datei geschrieben und mit diesem Baum Schritt 7. erneut ausgeführt. Wurde im letzten Schritt kein neuer optimaler Baum gefunden und sind noch nicht alle Spezies im Baum enthalten, so fährt das Programm in Punkt 5. mit den Berechnungen fort. Wurde im letzten Schritt kein neuer optimaler Baum gefunden, aber alle Spezies eingefügt, so wird im nächsten Punkt fortgefahren.

9. Der im letzten Rearrangement gefundene optimale Baum ist nun der rekonstruierte Baum. Dieser wird, falls nicht anders gewünscht, in der Ausgabedatei und in der `treefile`–Datei abgespeichert.

10. Anschließend werden alle reservierten Speicherbereiche wieder freigegeben und das Programm beendet.

Während des gesamten Ablaufs macht das Programm Ausgaben an die Standardausgabe über den Stand der Analyse, die hier nicht weiter spezifiziert wurden.

Allgemeine Struktur des Parallelprogramms

Anhand der gefundenen Programmstruktur wurden die Ansatzpunkte bestimmt, an denen eine Parallelisierung am aussichtsreichsten erschien. Diese Stellen sind in der obengenannten Struktur in den Punkten 5. und 7. zu finden. An diesen Stellen wird vom Programm in jedem Schritt eine Anzahl von Bäumen generiert, die voneinander unabhängig sind und daher auch unabhängig in Bezug auf ihre Kanten optimiert werden können.

Für die Parallelstruktur von `pfastDNAml` wurde das in Abb. 5.1 gezeigte Konzept entworfen. Ein Paralleljob startet zwei verschiedene Arten von Prozessen. Einerseits ist dies der sogenannte Master–Prozeß, der das Generieren der Bäume übernimmt, die weiteren Prozesse initialisiert und die Kommunikation mit der Außenwelt, d.h. Aus– und Eingaben, übernimmt. Andererseits

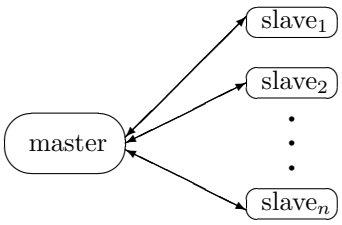


Abbildung 5.1: Parallelisierungskonzept der Parallelversion von `pfastDNAml`; die Kommunikation zwischen den Master– und Slave–Prozessen wird über Routinen aus der PARMACS–Bibliothek geregelt.

übernehmen sogenannte Rechen– bzw. Slave–Prozesse die Aufgabe, die Kanten der einzelnen generierten Stammbäume zu berechnen und zu optimieren.

Bei diesem Konzept wird die Arbeit, den Maximum–Likelihood–Wert eines Baumes zu berechnen und damit dessen Kantenlängen, von jeweils einem Slave erledigt. Während dies geschieht, können weitere Bäume, die der Master–Prozeß in der Zwischenzeit generiert hat, *gleichzeitig* von weiteren Slave–Prozessen berechnet werden, da der Master–Prozeß nicht mehr auf die Berechnung des zuletzt generierten Baumes warten muß.

Die Kommunikation zwischen den einzelnen Prozessen ist abhängig von der Rechnerplattform, auf der gerechnet werden soll. Um von dieser rechnerspezifischen Kommunikation unabhängig zu sein, wird die schon erwähnte PARMACS–Befehlsbibliothek (Kap. 3.2.5) verwendet. Diese stellt Befehle zur Verfügung, die einen standardisierten Zugriff auf die Kommunikationsmöglichkeiten der zugrundeliegenden Plattform ermöglichen. Die Implementierung der Kommunikation geschieht für den Benutzer transparent, d.h. er muß sich keine Gedanken über die Implementierung der systemspezifischen Kommunikation machen.

Ziel dieser Parallelisierung ist es, die Laufzeit des Programms durch Erhöhung der Anzahl der Slave–Prozesse auf ein erträgliches Maß zu drücken. Hierzu wurden in dieser Arbeit Laufzeituntersuchungen durchgeführt.

Programmtechnisch gesehen, handelt es sich bei den Master– und Slave–Prozessen um dasselbe ausführbare Programm, das nach seinem Start auf einem Knoten der benutzten Parallelplattform selbständig erkennt, ob es als Master– oder Slave–Prozeß aufgerufen wurde. Dieses mußte aus Gründen der besseren Portabilität sein, da es Parallelplattformen gibt, auf denen in alle reservierten Knoten nur dasselbe Programm geladen werden kann (Keller, 1995, persönliche Kommunikation).

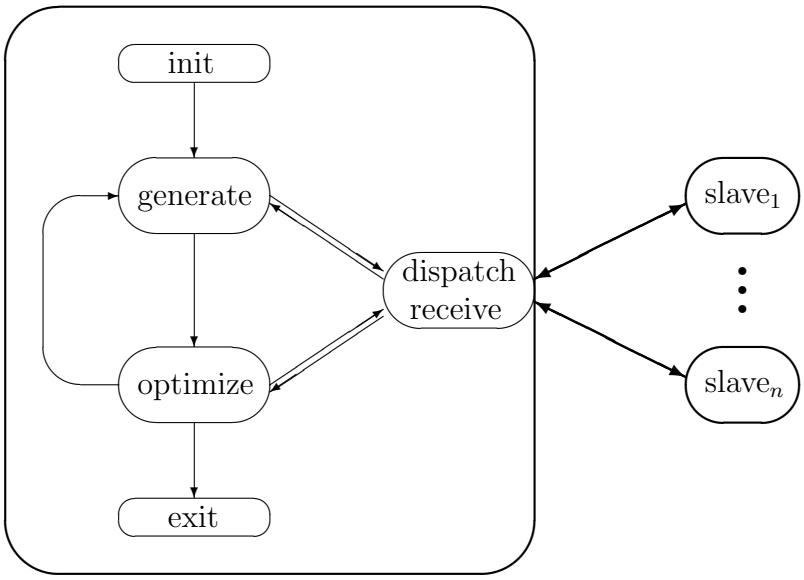


Abbildung 5.2: Struktur des Master–Prozesses bei der Parallelversion von `pfastDNAml`, inklusive der Master/Slave–Kommunikation zum Übermitteln der Bäume. Die Kommunikation während des **init**– und **exit**–Schrittes ist nicht eingezeichnet, da sie von untergeordneter Bedeutung ist.

Struktur des Master–Prozesses

Die nachfolgend beschriebene Struktur und der Ablauf innerhalb des Master–Prozesses ist auch in Abb. 5.2 in Form eines Flußdiagramms dargestellt.

init Am Anfang geschieht eine Initialisierung des Master–Prozesses genau wie in den Punkten 1. bis 3. im sequentiellen Programm. Zusätzlich werden alle angeforderten Slave–Prozesse angesprochen und ihnen der gesamte Inhalt der Eingabedatei über das Netzwerk zugeschickt. Dieses geschieht, da es Parallelrechnersysteme gibt, in denen nicht alle Knoten oder Prozessoren Zugriff auf die Festplatte haben (Keller, 1995, persönliche Kommunikation). Zusätzlich werden alle weiteren nötigen Daten, wie z.B. die Startzeit, an die Slave–Prozesse gesandt.

Anschließend wird der erste Baum aus drei Spezies generiert und die Kantenlängen vom Master–Prozeß selbst bestimmt (Punkt 4.). Hiermit ist die Initialisierung des Master–Prozesses abgeschlossen.

generate Dann beginnt der Master–Prozeß der Reihe nach neue Bäume

durch Einfügen einer neuen Spezies in den letzten gefundenen optimalen Baum zu generieren. Mit jedem neu generierten Baum wird die Routine zur Master/Slave–Kommunikation (**dispatch/receive**) aufgerufen. Es werden so alle möglichen Baumtopologien generiert und an den Kommunikationsteil übergeben.

Nach Übergabe des letzten in diesem Schritt zu generierenden Baumes verharrt der Master–Prozeß in der Kommunikationsroutine und kehrt mit dem besten der berechneten Bäume wieder an diese Stelle zurück. Der gefundene Baum wird in der `checkpoint`–Datei abgelegt, und der Master–Prozeß geht dann zum Optimieren durch Rearrangement (**optimize**) über. Sollten noch nicht mehr als vier Spezies im Baum enthalten sein, beginnt der **generate**–Schritt mit dem besten Baum und der nächsten Spezies.

optimize Mit dem besten im generate–Schritt gefundenen Baum wird nun ein Rearrangement durchgeführt, wie es in der Eingabedatei spezifiziert wurde. Hierbei werden alle Teilbäume aus dem existierenden Baum ausgeschnitten und in die Nachbarkanten eingesetzt. So entstehen Schritt für Schritt neue Bäume, mit denen wieder die **dispatch/receive**–Routine aufgerufen wird.

Auch jetzt verharrt der Master–Prozeß wieder nach Übergabe des letzten in diesem Schritt zu generierenden Baumes in der Kommunikationsroutine und kehrt mit dem besten der berechneten Bäume wieder an diese Stelle zurück. Wurde in diesem Durchlauf ein besserer Baum gefunden, wird dieser in der `checkpoint`–Datei abgelegt und der **optimize**–Schritt wiederholt, bis kein neuer optimaler Baum mehr gefunden wird.

Sind im Baum alle Spezies enthalten so geht der Master–Prozeß zum **exit**–Schritt über, anderenfalls wird mit dem diesem optimalen Baum und der nächsten Spezies wieder der **generate**–Schritt durchlaufen.

dispatch/receive Wird ein Baum an diese Routine übergeben, so wird geprüft, ob Anforderungen von Slave–Prozessen vorliegen. Falls dem so ist, wird der übergebene Baum sowie eventuell in der Warteschlange befindliche Bäume an die Slave–Prozesse geschickt, bis alle Anfragen bearbeitet oder keine Bäume mehr vorhanden sind.

Sind keine Anfragen vorhanden, wird der übergebene Baum in eine Warteschlange eingereiht.

Anschließend an das Versenden wird geprüft, ob berechnete Bäume von Slave–Prozessen zurückgesandt wurden. Solche Bäume werden in

eine mittels eines Heaps implementierte Priority–Queue (Cormen *et al.*, 1990) eingereiht, um die besten gefundenen Bäume heraussortieren zu können.

Sind im **generate**–Schritt bzw. **optimize**–Schritt noch neue Bäume zu generieren, so kehrt der Master–Prozeß zu diesem Schritt zurück.

Sind im **generate**–Schritt bzw. **optimize**–Schritt keine neuen Bäume mehr zu generieren, so verbleibt der Master–Prozeß in dieser Routine, bis alle Bäume an die Slave–Prozesse verteilt und von diesen nach der Berechnung des ML–Wertes zurückgesandt wurden.

Anschließend wird der von allen berechneten Bäumen optimale Baum aus der Priority–Queue gezogen und der Master–Prozeß springt an das Ende des **generate**– bzw. **optimize**–Schrittes.

exit Nach der Analyse wird der gefundene ML–Baum, falls nicht anders angegeben, als Ergebnis in der Ausgabedatei und der der Baum in der `treefile`–Datei ausgegeben.

Anschließend wird den Slave–Prozessen übermittelt, daß die Analyse beendet ist. Der Master–Prozeß gibt dann den belegten Speicherplatz wieder frei und endet, nachdem auch die Slave–Prozesse gestoppt haben.

Während der Analyse wird kontinuierlich jeder auszuführende Schritt und dessen Laufzeiten in der Ausgabedatei dokumentiert.

Struktur des Slave–Prozesses

Auch die Struktur und der Ablauf des im folgenden beschriebenen Slave–Prozesses ist in Form eines Flußdiagramms (Abb. 5.3) dargestellt.

init Im **init**–Schritt empfängt der Slave–Prozeß vom Master–Prozeß den Inhalt der gesamten Eingabedatei sowie alle anderen für die Initialisierung notwendigen Daten. Anschließend werden die Daten aus der Eingabedatei ausgewertet und der notwendige Speicherplatz reserviert und initialisiert, wie in der sequentiellen Version in Punkt 1. bis 3. beschrieben. Nach Abschluß der Initialisierung geht der Slave–Prozeß zum **receive/send**–Schritt über.

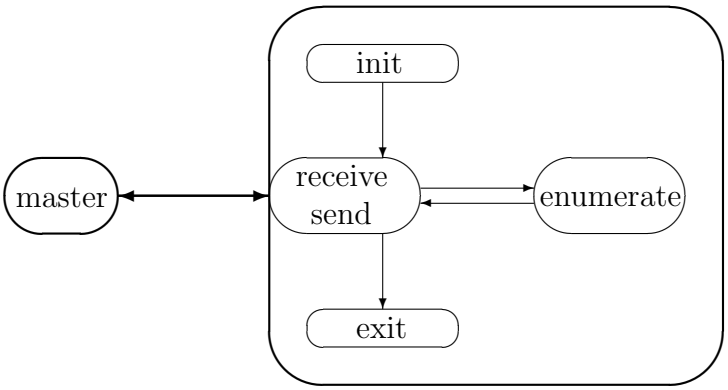


Abbildung 5.3: Struktur der Slave–Prozesse von `pfastDNAml`, inklusive der Master/Slave–Kommunikation zum Empfangen und Senden der Bäume. Die Kommunikation während des **init**– und **exit**–Schrittes ist nicht eingezeichnet, da sie von untergeordneter Bedeutung ist.

receive/send Kommt der Slave–Prozeß nach der Ausführung von **enumerate** an diese Stelle, so sendet er den berechneten Baum an den Master–Prozeß. Jetzt wird eine Anfrage nach einem zu berechnenden Baum an den Master–Prozeß gesandt.

Der Slave–Prozeß geht nun in eine Warteschleife über, bis Antwort vom Master–Prozeß übermittelt wurde. Handelt es sich bei der empfangenen Nachricht um einen Baum, so wechselt der Slave–Prozeß in die **enumerate**–Routine. Ist dagegen eine Nachricht über das Ende der Analyse eingetroffen, so wird, mit dem **exit**–Schritt fortgefahren.

enumerate Bei dem empfangenen Baum werden alle Kanten der Reihe nach so lange nach der ML–Methode (siehe Kap. 4.2) optimiert, bis der Likelihood–Wert des Baumes sein Maximum erreicht. Nach der Berechnung der Baumkanten kehrt der Slave–Prozeß mit dem berechneten Baum wieder zum **receive/send**–Teil zurück.

exit Nachdem die Analyse vom Master–Prozeß als abgeschlossen gemeldet wurde, werden alle reservierten Speicherbereiche wieder freigegeben und der Slave–Prozeß beendet.

Ist das Programm mit der `PM_DEBUG`–Option compiliert worden, so wird auch von jedem Slave–Prozeß eine Ausgabedatei angelegt und Angaben über den Programmlauf darin ausgegeben.

Hilfsprogramme und Rechnerplattformen

Das oben beschriebene Parallelisierungskonzept wurde auf einer Workstation umgesetzt und lauffähig gemacht. Danach wurde die Entwicklungsplattform gewechselt und auf einer IBM SP2 mit 34 Knoten weitergearbeitet. Die kleinen systembedingten Anpassungen gestalteten sich problemlos.

Dann wurden die durch das Programm gelieferten Ergebnisse anhand verschiedener Testdatensätze mit den Ergebnissen der sequentiellen `fastDNAml`–Version verglichen. Hierbei ergaben sich bis auf kleine Rundungsfehler in den hinteren Nachkommastellen des Likelihood–Wertes der gefundenen Bäume keine Unterschiede. Die gefundenen Fehler waren aufgrund der unterschiedlichen Rechnertypen sowie durch das Runden des Likelihood–Wertes beim Versenden des Baumes vom Slave– an den Master–Prozeß zu erwarten.

Anschließend wurde die Applikation auf eine NEC Cenju–3 mit 64 Knoten portiert. Auch diese Portierung gelang ohne größere Anpassungen im Quellcode von `pfastDNAml`. Bei der Cenju–3 handelt es sich um einen wie auf Seite 49 beschriebenen Rechnertyp, der nur ein einzelnes ausführbares Programm auf alle angeforderten Knoten verteilen kann.

Zur Vereinfachung der Handhabung des Programms auf den unterschiedlichen Rechnerplattformen wurden Skripten entwickelt, die im Anhang (Seite 102) beschrieben sind. In ersten Linie handelt es sich hierbei um Shellskripten für Bourneshell–kompatible Shells (Frisch, 1991) oder Kornshells (Rosenblatt, 1993), aber einige wurden auch in der Skriptsprache perl (Wall und Schwartz, 1991) programmiert.

5.1.2 Untersuchungen des Laufzeitverhaltens

Da mir nur die IBM SP2 dauerhaft zu Verfügung stand, wurden dort die Untersuchungen zum Laufzeitverhalten von `pfastDNAml` durchgeführt.

Hierfür wurden Alignments verschiedenen Umfangs zusammengestellt. Der erste Datensatz (*rrna10*) bestand aus 10 unterschiedlichen Bakteriensequenzen und der zweite (*rrna20*) aus 20 unterschiedliche Eukaryotensequenzen. Die Sequenzen beider Datensätze stammen aus der RDP–Datenbank. Der dritte Datensatz (*rrna28*) enthielt rRNA–Sequenzen von 28 Organismen aus allen drei Überreichen, d.h. Archaebakterien, Eubakterien (mit Mitochondrien und Plastiden) und auch Eukaryoten. Diese Sequenzen wurden der rRNA–Datenbank in Antwerpen entnommen.

Alle Testläufe wurden mit der `G`–Option durchgeführt, d.h. es wurden am Ende jeder Analyse globale Rearrangements ausgeführt, um den bis dahin

Slave–Prozesse	Laufzeit gesamt	Bäume gesamt	Laufzeit glob. Rearr.	Bäume glob. Rearr.
1	271s	306	191s	182
2	139s	306	96s	182
4	74s	306	48s	182
8	45s	306	28s	182
16	32s	306	18s	182
24	29s	306	15s	182
32	28s	306	13s	182

Tabelle 5.1: Laufzeiten des Testdatensatzes 1 (*rrna10*) mit 10 Spezies

Slave–Prozesse	Laufzeit gesamt	Bäume gesamt	Laufzeit glob. Rearr.	Bäume glob. Rearr.
1	3917s	1890	5559s	1122
2	2805s	1890	1958s	1122
4	1426s	1890	980s	1122
8	740s	1890	490s	1122
16	409s	1890	246s	1122
24	312s	1890	169s	1122
32	254s	1890	126s	1122

Tabelle 5.2: Laufzeiten des Testdatensatzes 2 (*rrna20*) mit 20 Spezies

gefundenen Baum noch zu optimieren und die Fehler der Heuristik auszugleichen.

Die unterschiedlichen Testdatensätze wurden auf der IBM SP2 mit verschiedener Anzahl von Slave–Prozessen analysiert. Die durchgeführten Berechnungen und deren Ergebnisse waren bei den unterschiedlichen Knotenzahlen identisch, wie die inspizierten Ausgabedateien ergaben. Dies war auch erwartet worden, da nur die Bäume auf eine unterschiedliche Anzahl von Slave–Prozessen zur Berechnung verteilt wurden und der Ablauf sich ansonsten nicht änderte.

Die Laufzeiten der einzelnen Analysen der drei Datensätze und die Anzahl der dabei untersuchten Bäume sind in den Tabellen 5.1, 5.2 und 5.3 dargestellt. Aufgrund der langen Laufzeiten des *rrna28*–Datensatzes mit wenigen Slave–Prozessen, wurden mit diesem Datensatz nur Laufzeiten von Konfigurationen mit 32, 16 und einem Slave–Prozeß untersucht.

Slave–Prozesse	Laufzeit gesamt	Bäume gesamt	Laufzeit glob. Rearr.	Bäume glob. Rearr.
1	106628s	6212	46954s	2450
			45651s	2450
16	7125s	6212	2946s	2450
			2864s	2450
32	3942s	6212	1484s	2450
			1446s	2450

Tabelle 5.3: Laufzeiten des Testdatensatzes 3 (*rrna28*) mit 28 Spezies

5.2 Diskussion

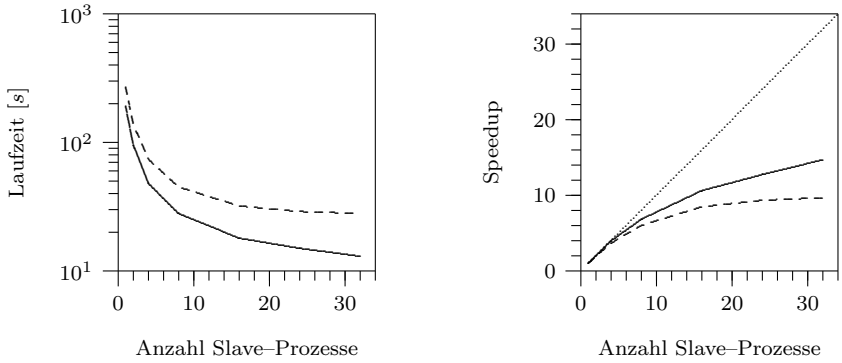
5.2.1 Speedup und Skalierbarkeit

Die Untersuchungen der verschiedenen Testdatensätze zeigen ein interessantes Ergebnis. In der graphischen Darstellung der Analyse von *rrna10* ist kein befriedigender Speedup[1] der Rechenzeit mit steigender Anzahl an Slave–Prozessen zu erkennen (Abb. 5.4). Der Speedup, der sich aus dem Laufzeitverhältnis der einzelnen Analysen zur Analyse mit einem einzelnen Slave–Prozeß berechnet (Burkhardt, 1993), weicht sehr schnell vom linearen Verlauf ab und beginnt in Sättigung überzugehen.

Bei näherem Hinsehen ist dies allerdings einfach verständlich. Da die während eines Schrittes keine große Anzahl an Bäume zur Berechnung generiert werden (im letzten Schritt generiert der Master–Prozeß durch Zufügen der zehnten Spezies 15 neue Bäume), bedeutet dies, daß bis zum globalen Rearrangement, bei dem 182 Bäume generiert werden, bei höheren Prozessorzahlen, niemals alle Prozessoren genutzt werden. Auch im globalen Rearrangement geht der Speedup einer Sättigung entgegen. Dies liegt daran, daß die Zeit, die benötigt wird, um die Kantenlängen und den Likelihood–Wert dieser relativ kleinen Bäume zu berechnen ($< 1s$), in einem schlechten Verhältnis zum entstandenen Mehraufwand durch das Verschicken der Bäume an die Slave–Prozesse steht.

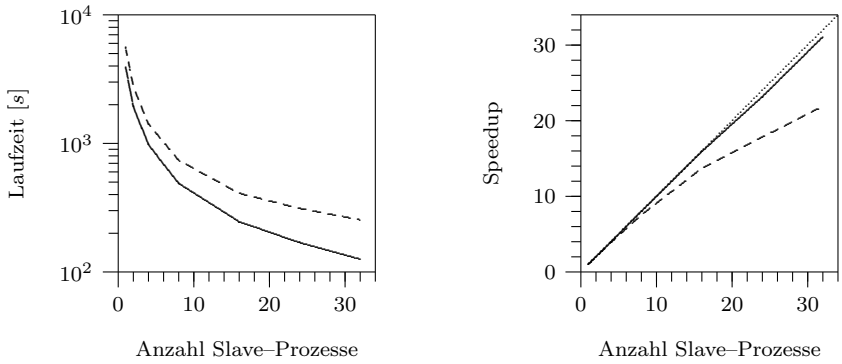
Bei der Betrachtung des Speedups der Analysenlaufzeiten bei den größeren Datensätzen (Abb. 5.5 und 5.6) zeigt sich schon ein erheblich besseres Bild. Vor allem der Speedup der globalen Rearrangements ist hier schon fast linear. Dies liegt an der exponentiell mit der Anzahl der benutzten Spezies

[1] Geschwindigkeitszuwachs



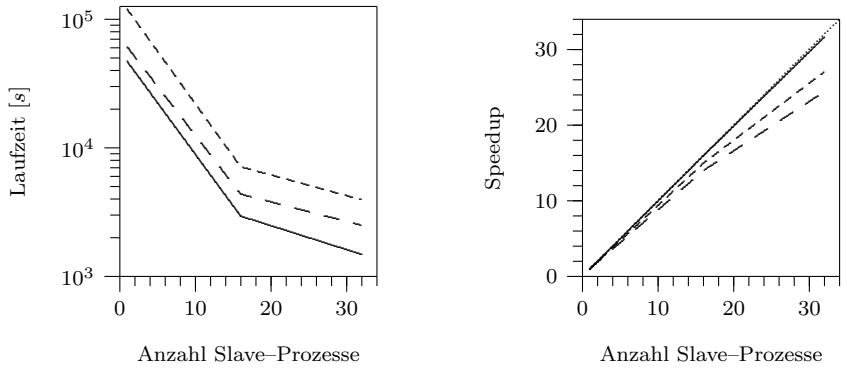
(a) Zeitverhalten (b) Skalierungsverhalten

Abbildung 5.4: Zeit– und Skalierungsverhalten beim *rrna10*-Datensatz; gestrichelte Linie – Gesamtprozeß; durchgezogene Linie – globales Rearrangement; gepunktet – linearer Speedup; (Anzahl Spezies: 10; Bäume gesamt: 306; Bäume eines globalen Rearrangements: 182; Anzahl globaler Rearrangements: 1)



(a) Zeitverhalten (b) Skalierungsverhalten

Abbildung 5.5: Zeit– und Skalierungsverhalten beim *rrna20*-Datensatz gestrichelte Linie – Gesamtprozeß; durchgezogene Linie – globales Rearrangement; gepunktet – linearer Speedup; (Anzahl Spezies: 20; Bäume gesamt: 1890; Bäume des globalen Rearrangements: 1122; Anzahl globaler Rearrangements: 1)



(a) Zeitverhalten (b) Skalierungsverhalten

Abbildung 5.6: Zeit– und Skalierungsverhalten beim *rrna28*-Datensatz kurz gestrichelte Linie – Gesamtprozeß; lang gestrichelte Linie – Gesamtprozeß abzüglich des zweiten globalen Rearrangements; durchgezogene Linie – globales Rearrangement; gepunktet – linearer Speedup; (Anzahl Spezies: 28; Bäume gesamt: 6212; Bäume des ersten globalen Rearrangements: 2450; Anzahl globaler Rearrangements: 2)

wachsenden Anzahl der in einem globalen Rearrangement generierten Bäume. Die Anzahl beträgt für n Spezies annähernd

$$4n^2 - 26n + 38$$

Bäume (Matsuda *et al.*, 1994, unveröffentlicht). Da die Bäume im globalen Rearrangement, verglichen mit den Bäumen aller anderen Schritte ($2i - 5$ Bäume beim Einfügen der i-ten Spezies und $2i - 6$ beim darauffolgenden lokalen Rearrangement über einen Ast), den bei weitem größten Anteil ausmachen, verbessert sich bei Vergrößerung des Eingabedatensatzes zusehends auch die zu beobachtende Skalierbarkeit. D.h. je mehr Spezies eingegeben werden, um so mehr kann die Berechnung durch Erhöhung der Anzahl der Slave–Prozesse beschleunigt werden, bis Sättigungseffekte auftreten.

Das Ziel der Parallelisierung, die Rechenzeit effektiv zu verringern, ist mit der Implementierung von `pfastDNAml` gelungen. Die Parallelversion sollte ja dafür genutzt werden, um größere Probleme in einem annehmbaren Zeitraum lösbar zu machen. Berechnungen mit bis zu 15 Spezies, die auch vorher schon mit den sequentiellen Implementierungen berechnet wurden, spielen bei den Betrachtungen nur eine untergeordnete Rolle, da sie ohnehin kein größeres zeitliches Problem darstellten.

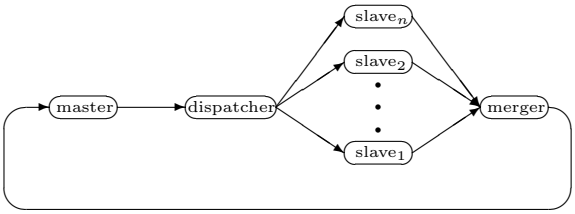


Abbildung 5.7: Parallelisierungskonzept der Parallelversion von `fastDNAml` (Matsuda *et al.*, 1994, unveröffendlicht)

5.2.2 Parallelisierungskonzept

Vergleicht man das gewählte Parallelisierungskonzept (Abb. 5.1) mit dem, das bei der früheren P4–Implementierung von `fastDNAml` verwendet wurde (Abb. 5.7, nach Matsuda *et al.* 1994, unveröffentlicht), so fällt auf, daß bei `pfastDNAml` weit weniger verschiedene Prozeßtypen implementiert wurden. In `fastDNAml` gab es neben den Master– und Slave–Prozessen noch den Dispatcher–Prozeß, der das Verteilen der Bäume an die Slave-Prozesse übernimmt, und einen Merger–Prozeß, der die fertigen Bäume empfängt und den besten an den Master–Prozeß übermittelt. Außerdem war noch ein, in der Abbildung nicht gezeigter, Host–Prozeß implementiert, der die Initialisierung des Gesamtprozesses übernahm.

Die Aufgaben des Dispatcher–, Merger– und Host–Prozesses werden bei `pfastDNAml` durch den Master–Prozeß mitübernommen. Untersuchungen bei laufender Berechnung haben ergeben, daß der Master–Prozeß von `pfastDNAml` viel Zeit damit verbringt, darauf zu warten, daß er die schon fertig generierten Bäume an die Slave–Prozesse übermitteln kann. Da der Master–Prozeß also nicht in seiner Hauptaufgabe, dem Generieren der Bäume, gestört war, bestand bei `pfastDNAml` nicht die Notwendigkeit, die Kommunikation mit den Slave–Prozessen auszulagern. Die hierdurch gesparten drei Knoten können bei `pfastDNAml`–Analysen gut durch entsprechend mehr Slave–Prozesse genutzt werden, die sich, wie oben angesprochen, bei größeren Analysen positiv auf die Laufzeit auswirken.

Kapitel 6

Untersuchung der Crown Group Radiation

6.1 Ergebnisse

Der Ergebnisteil beschränkt sich in erster Linie darauf, die beim Generieren der Bäume angefallenen Daten sowie die konstruierten Bäume vorzustellen. Außerdem wird angegeben, welche Untersuchungen und Gewichtungen an den Datensätzen vorgenommen wurden. Diese Daten werden hier nur dargestellt und erst im anschließenden Diskussionsteil diskutiert.

Alle Analysen wurden mit lokalen Rearrangements über einen Ast ausgeführt, während am Ende jeder Analyse globale Rearrangements (`G`–Option) durchgeführt wurden. Die Basenfrequenzen wurden anhand des eingegebenen Datenmaterials empirisch bestimmt. Als vorgegebene Transition/Transversions–Rate wurde der, schon in `dnaml` vorgegebene, Standardwert 2,0 verwendet (Felsenstein, 1993). Bei gewichteten Datensätzen wurden als Gewichte für die einzelnen Gewichtskategorien die mit dem Programm `MacClade` (Maddison und Maddison, 1992) berechneten Gewichte verwendet (siehe Kap. 4.3).

6.1.1 Aktinentwicklung

Es wurden aus dem zur Verfügung stehenden Alignment unterschiedliche Sequenzen ausgewählt, die die unterschiedlichsten Organismengruppen repräsentierten und gute Rückschlüsse auf die Entwicklung der Aktingene in den entsprechenden Spezies versprachen.

Analyse der ersten beiden Codon–Positionen

Da die dritte Base der Codons der Aktingene sehr variabel ist, wird diese Base bei phylogenetischen Analysen normalerweise außeracht gelassen. Der G+C–Gehalt an dieser Position bewegt sich bei unterschiedlichen Spezies zwischen 16% und 98% (z.B. *Emiliania Huxleyi* 98%, *Entamoeba histolytica* 22%, *Volvox carteri* 72%). Dagegen beträgt der G+C–Gehalt der anderen Positionen 46%–59% (erste Position) bzw. 38%–43% (zweite Position). Diese Eigenschaften hängen auch nicht mit den einzelnen Gruppen von Spezies zusammen, sondern variieren auch innerhalb der verschiedenen Gruppen. (Bhattacharya *et al.*, 1993; Bhattacharya und Ehlting, 1995)

Aus diesem Grund wurde auch hier ein Stammbaum mit nur den ersten beiden Codon–Positionen der Sequenz konstruiert. Der Datensatz bestand hier aus 51 Spezies mit einem Alignment aus 702 Basen. `pfastDNAml` benutzte hiervon 485 unterschiedliche Spalten.

Anhand der empirisch errechneten Basenfrequenzen 0,297 (A), 0,222 (C), 0,254 (G) und 0,227 (T) wurde ein Transition/Transversions–Parameter von 1,4807 bestimmt. Es wurden bei der Berechnung des Baumes 96823 verschiedene Bäume betrachtet und 10 globale Rearrangements ausgeführt. Der so gefundene Baum (Abb. 6.1) hat einen Log–Likelihood–Wert von -10585,69.

Analyse mit allen Sequenzstellen

Zur weiteren Betrachtung wurde ebenfalls ein Baum für alle Codon–Positionen (1051 Alignmentspalten, 823 benutzte Spalten) berechnet. Die gefundenen Basenfrequenzen lagen bei 0,244 (A), 0,266 (C), 0,257 (G) und 0,234 (T) und der Transition/Transversions–Parameter bei 1,5048. Nach 11 globalen Rearrangements (je 9120 Bäume) und 108543 ingesamt untersuchten Bäumen wurde ein Baum mit einem Log–Likelihood–Wert von -31275,56 gefunden.

Gewichtete Analyse mit allen Sequenzstellen

Um zu untersuchen, ob sich mit einer Gewichtung der Sequenzstellen eine bessere Auflösung der Baumtopologie bei Benutzung der dritten Codon–Position erreichen ließe, wurde eine Gewichtung mit 10 Gewichten für den Gesamtdatensatz errechnet. Die prozentuale Verteilung der Basen auf die einzelnen Gewichte ist in Abb. 6.3 aufgezeigt. Aus diesem Datensatz wurde die zweite Sequenz von *Daucus carota* entfernt, da sich das Alignment dieser Sequenz bei näherer Betrachtung als unsicher herausgestellt hatte.

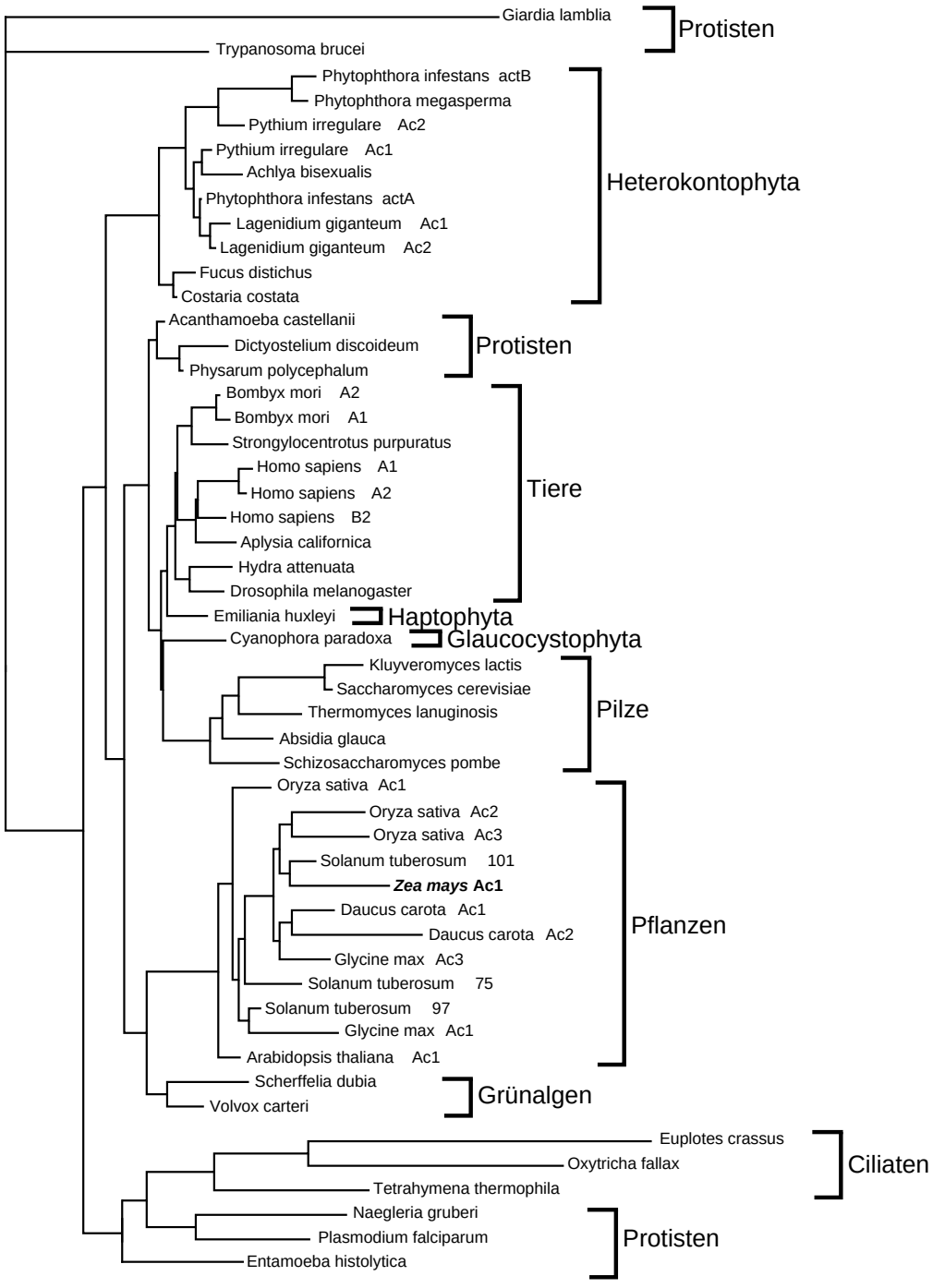


Abbildung 6.1: Phylogenetischer Stammbaum der Aktinentwicklung; berechnet mit `pfastDNAml` unter Benutzung der Codon–Positionen 1 und 2

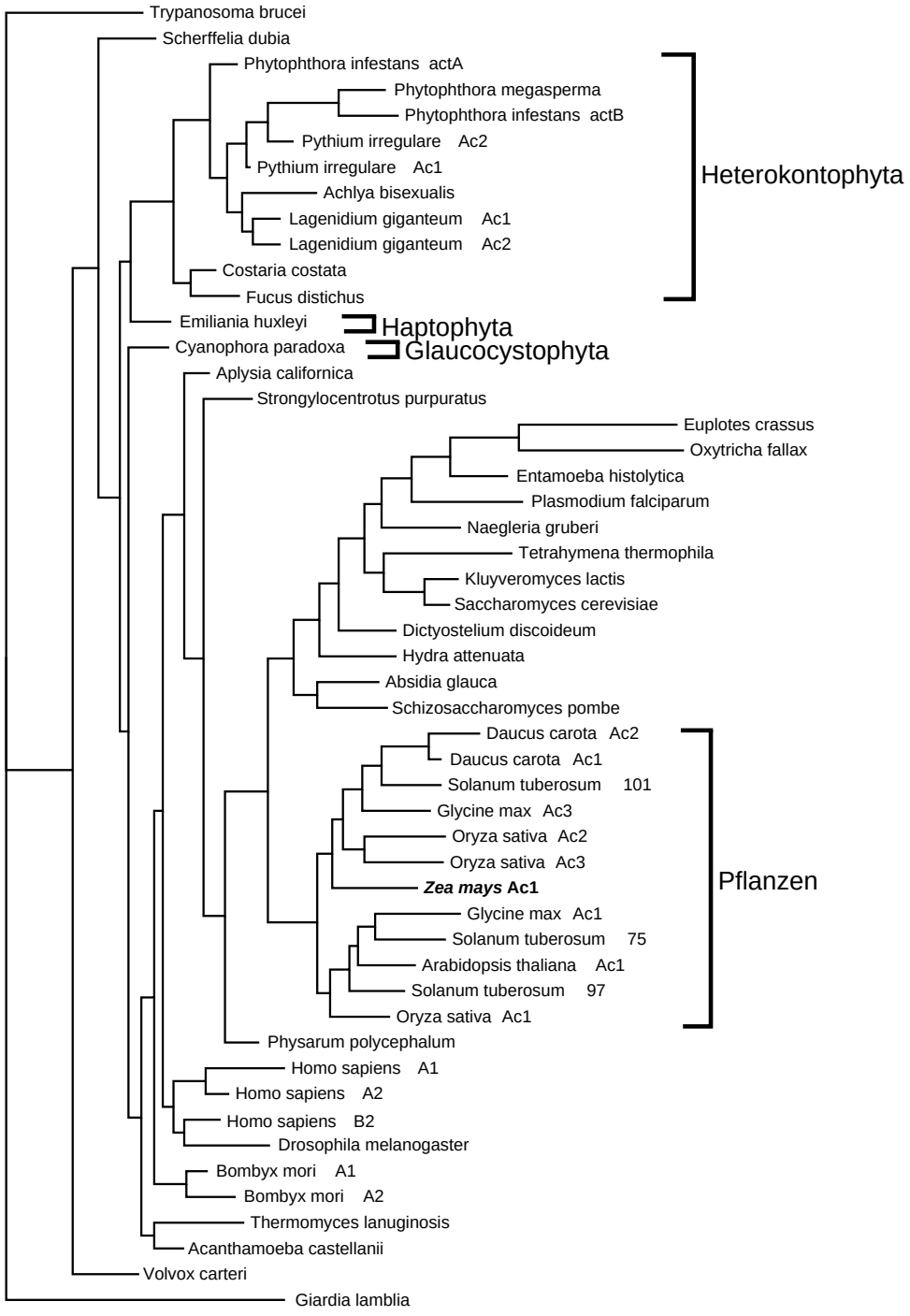


Abbildung 6.2: Phylogenetischer Stammbaum der Aktinentwicklung; berechnet mit `pfastDNAml` unter Benutzung aller Codon–Positionen

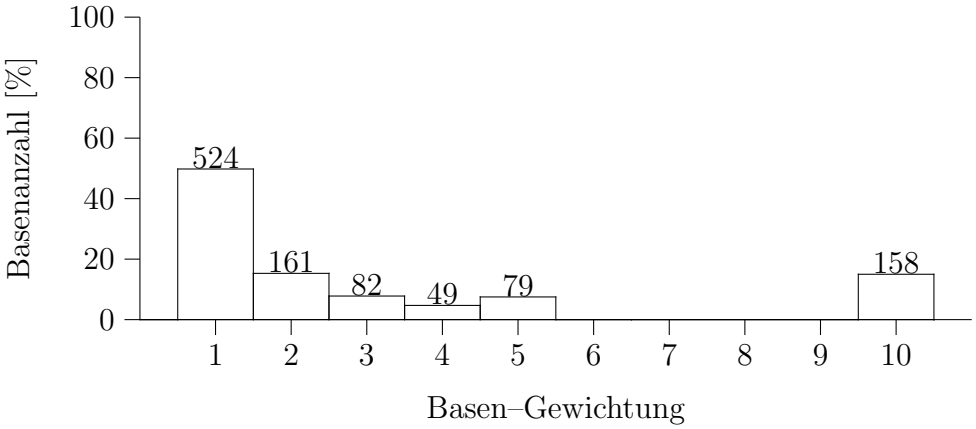


Abbildung 6.3: Verteilung der Basen des Aktindatensatzes auf die 10 Gewichtsgruppen; Gewichtungsfunktion: $1/S$; Anzahl Basen: 1053

Der anhand des gewichteten Datensatzes (50 Spezies, 1053 Spalte, davon 810 benutzt) berechnete Baum (Abb. 6.4) hat einen Log–Likelihood–Wert von -36281,56. Die ermittelten Basenfrequenzen sind 0,243 (A), 0,267 (C), 0,257 (G) und 0,233 (T) und der Transition/Transversions–Parameter 1,5054. Bei der Rekonstruktion wurden 76338 Bäume betrachtet und acht globale Rearrangement mit je 8742 Bäumen durchgeführt.

6.1.2 Plastidenentwicklung

Die Größe der Datensätze wurde so gewählt, daß die benötigte Rechenzeit für die Analyse der Datensätze, mit denen Bootstrap–Analysen geplant waren, auf der SP2 mit 34 Knoten 45 Minuten nicht überschritten. Hierdurch sollte der Aufwand für die Bootstrap–Analysen in einem erträglichen Rahmen gehalten werden.

Datensatz *plast1*

Analyse mit ungewichtetem Datensatz Der Datensatz *plast1* bestand aus einem Alignment aus 36 Spezies und 1429 Basen Länge. Hiervon wurden von `pfastDNAml` 758 unterschiedliche Spalten verwendet.

Die empirisch berechneten Basenfrequenzen lagen bei 0,268 (A), 0,220 (C), 0,302 (G) und 0,210 (T). Aus der Transition/Transversions–Rate und den Basenfrequenzen wurde 1.456 als Transversion/Transitions–Parameter berechnet.

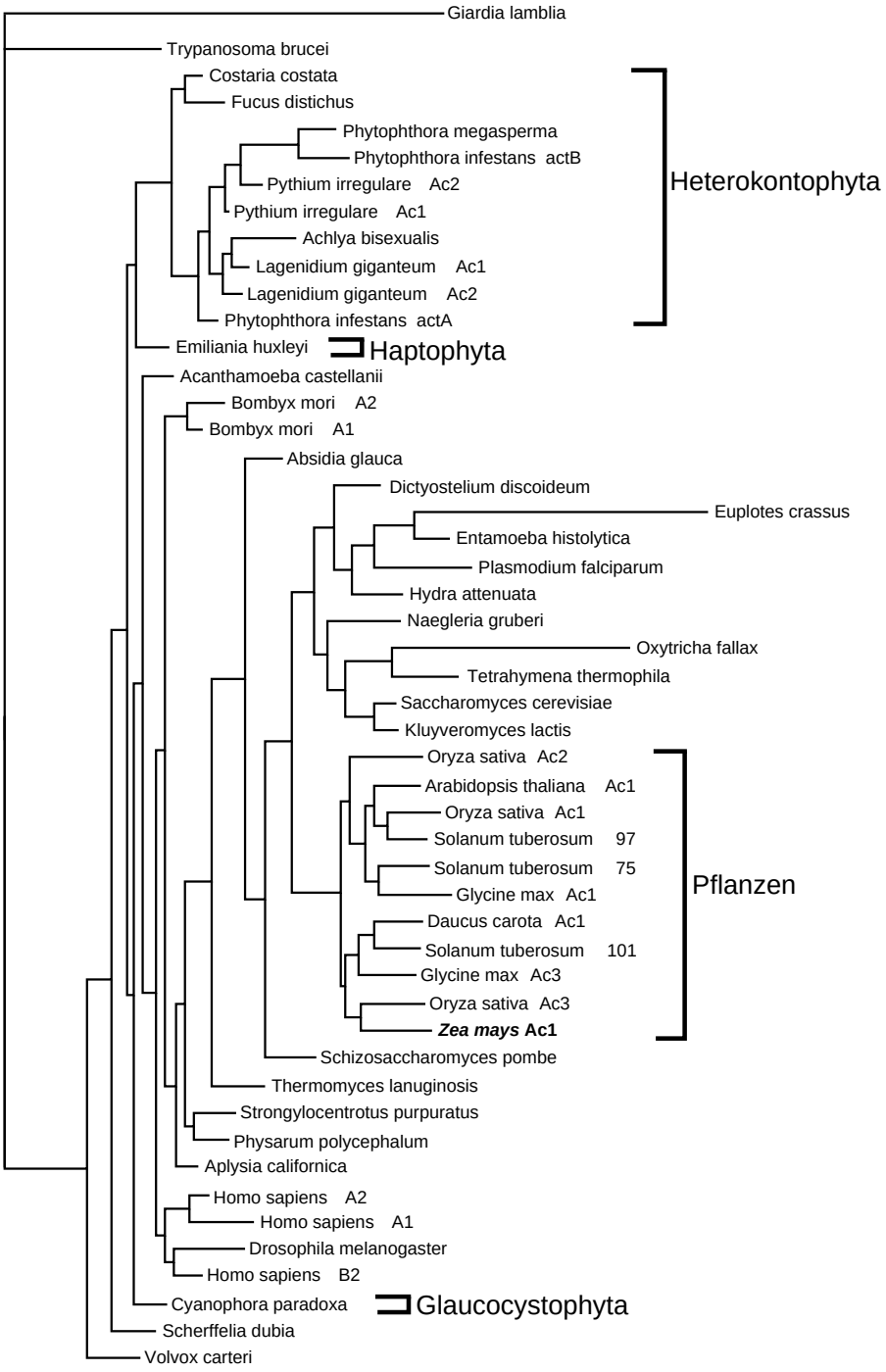


Abbildung 6.4: Phylogenetischer Stammbaum aus Aktinsequenzen unter Benutzung aller Codon–Positionen gewichtet mit 10 Gewichtsgruppen; berechnet mit `pfastDNAml`

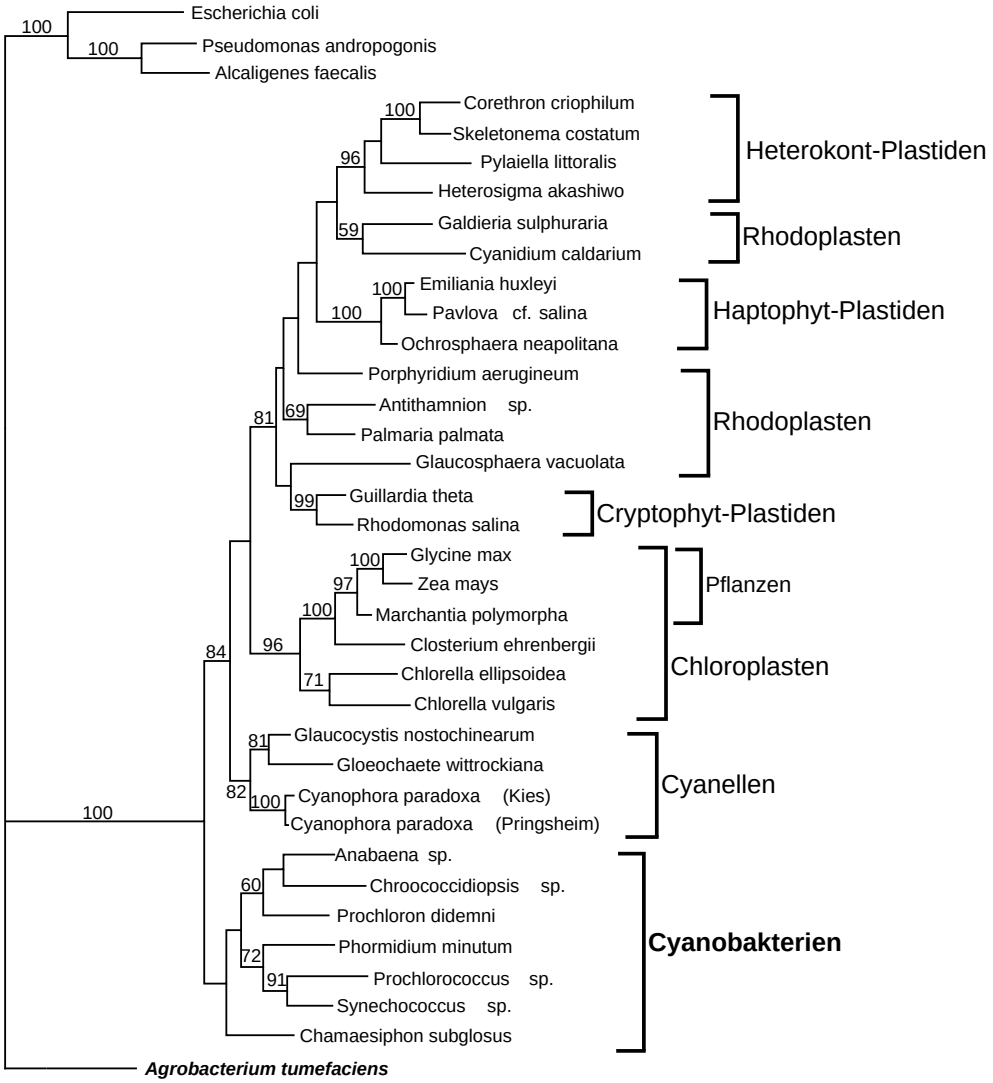


Abbildung 6.5: Phylogenetischer Stammbaum der Plastidenentwicklung; berechnet aus 16S rRNA–Sequenzen (Datensatz *plast1*) mittels `pfastDNAml`; nur Bootstrap–Werte über 60% wurden in den Baum an den entsprechenden Kanten eingefügt

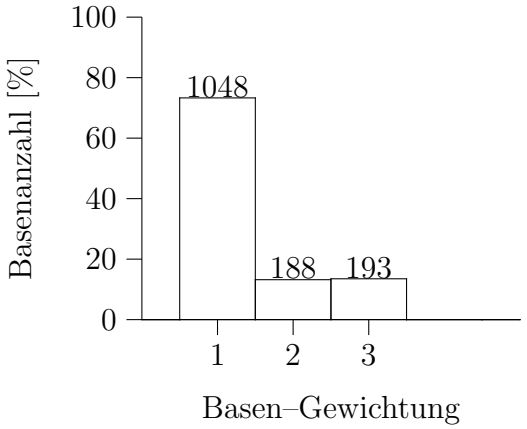


Abbildung 6.6: Verteilung der Basen des Plastidendatensatzes 2 auf 3 Gewichtsgruppen; Gewichtungsfunktion: $1/S$; Anzahl Basen: 1429

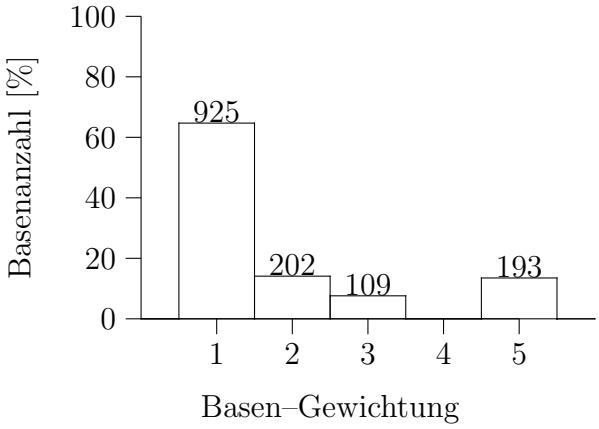


Abbildung 6.7: Verteilung der Basen des Plastidendatensatzes 1 auf 5 Gewichtsgruppen; Gewichtungsfunktion: $1/S$; Anzahl Basen: 1429

Im Laufe der Analyse wurden drei globale Rearrangements ausgeführt. D.h. zwei Rearrangements führten zu einem Baum mit einem besserem Likelihood–Wert.

Der gefundene Baum (Abb. 6.5) hatte einen Log–Likelihood–Wert von -18396,51. Insgesamt wurden während der Analyse 15532 Bäume analysiert.

Anschließend wurde eine Bootstrap–Analyse zu diesem Baum durchgeführt. Hierzu wurden aus dem Originaldatensatz 100 unabhängige Pseudostichproben gezogen und aus den 100 erhaltenen Bäumen ein Konsensusbaum erstellt. Die Bootstrap–Werte, die größer 60% waren, wurden in den Baum in Abb. 6.5 eingefügt.

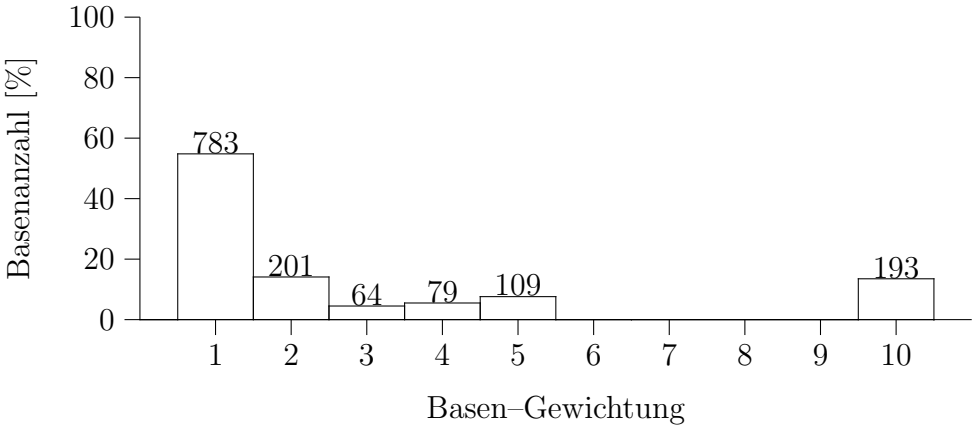


Abbildung 6.8: Verteilung der Basen im Plastidendatensatz 1 auf die 10 Gewichtsgruppen; Gewichtungsfunktion: $1/S$; Anzahl Basen: 1429

Analyse mit gewichtetem Datensatz Da die Sequenzstellen in rRNA sehr unterschiedliche Variabilität besitzen, wurde anhand des oben gefundenen Stammbaumes (Abb. 6.5) eine Gewichtung der Alignmentspalten mit 10 Gewichtsklassen vorgenommen (Abb. 6.8). Für den Datensatz mit 10 Gewichten ergab sich, nach zwei globalen Rearrangements und insgesamt 11536 untersuchten Bäumen, ein Baum mit einem Log–Likelihood–Wert von -19576,30 (Abb. 6.11). Jedes der bei den obigen Datensätzen durchgeführten Rearrangements umfaßte 4290 untersuchte Bäume.

Der so berechnete Baum in Abb. 6.11 zeigte eine interessante Veränderung zu dem ungewichteten Baum (Abb. 6.5). Die Gruppe der Cyanellen divergierte nun im Stammbaum nicht mehr vor, sondern nach den Chloroplasten.

Um die Auswirkungen von solchen Gewichtsveränderungen weiter zu untersuchen, wurde der Datensatz anhand des oben gefundenen Stammbaumes mit weiteren unterschiedlichen Gewichten (mit 3 und 5 Gewichten) belegt. Die Verteilung der Basenanzahl auf die einzelnen Gewichte sind in den Abbildungen 6.6 und 6.7 dargestellt. Zu allen gewichteten Datensätzen wurden Stammbäume berechnet.

Alle nicht explizit noch einmal aufgeführten Daten und Parameter sind identisch zu den schon oben gefundenen. Für den Datensatz mit drei Gewichten ergab sich nach zwei globalen Rearrangements und insgesamt 11484 untersuchten Bäumen ein Baum mit dem Log–Likelihood–Wert von -18980,62 (Abb. 6.9).

Für den Datensatz mit 5 Gewichten ergab sich nach sechs globalen Rearrangements und insgesamt 28668 untersuchten Bäumen ein Baum mit dem

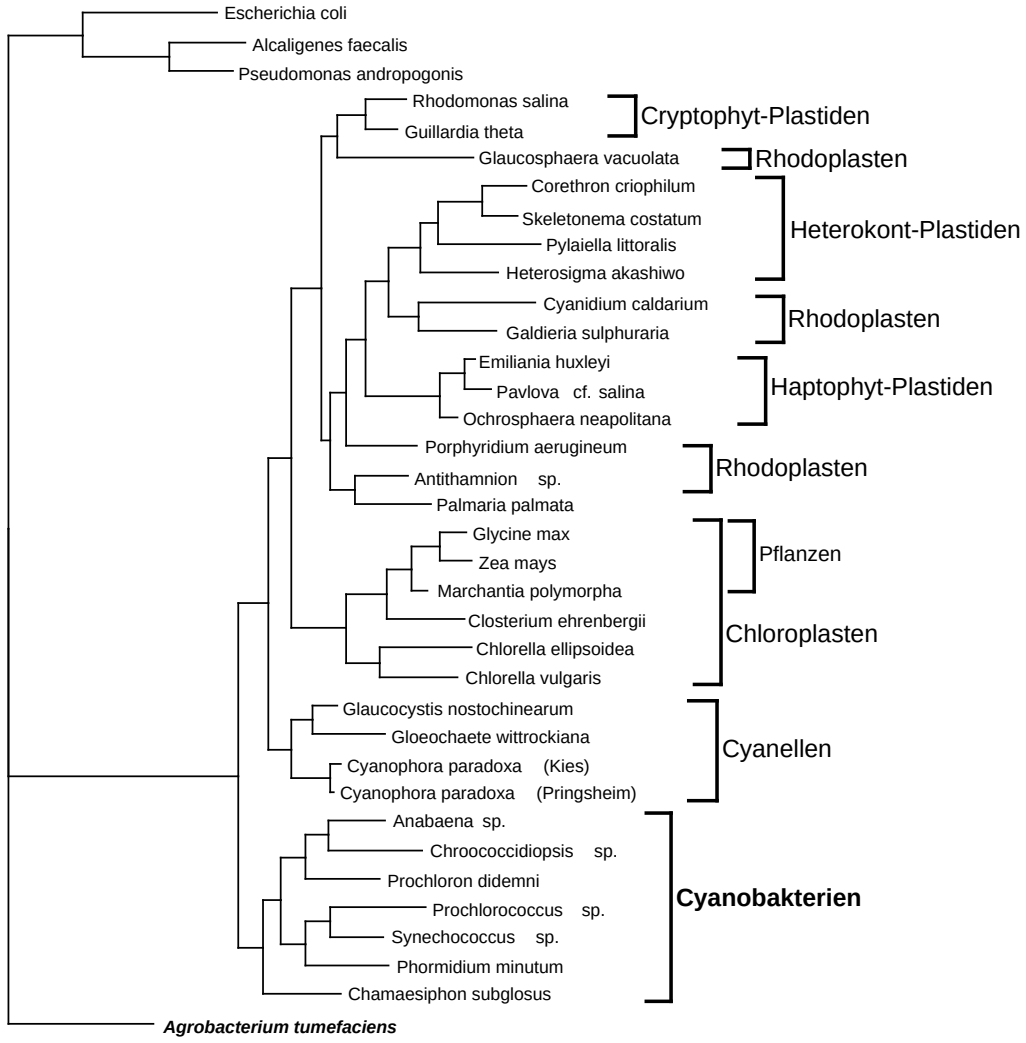


Abbildung 6.9: Phylogenetischer Stammbaum der Plastidenentwicklung; berechnet aus 16S rRNA–Sequenzen (Datensatz *plast1*) mit 3 Gewichtsgruppen mittels `pfastDNAml`

Log–Likelihood–Wert von -19276,75 (Abb. 6.10).

Anhand des letzten mit 10 Gewichten gewichteten Datensatzes wurde außerdem eine Bootstrap–Analyse mit 100 Pseudostichproben durchgeführt, ein Konsensusbaum erstellt und die Bootstrap–Werte > 60% in den Baum in Abb. 6.11 eingefügt.

Datensatz *plast2*

Analyse mit ungewichtetem Datensatz Im zweiten Plastiden–Datensatz wurden die drei Spezies *Corethron criophilum* (Heterokontophyta), *Prochlorococcus* sp. und *Prochloron didemni* (Cyanobakterien) durch andere ersetzt. Zum einen wurde die Jochalge *Chara* sp. zugefügt, die Organisationsformen aufweist, die an Pflanzen erinnern. Hierdurch sollten die Verwandtschaftsverhältnisse zwischen höherentwickelten Grünalgen und Pflanzen etwas näher beleuchtet werden. Außerdem wurde ein ursprünglicheres Cyanobakterium *Gloeobacter violacius* eingefügt, um in den Bereich der Wurzel des Plastidenastes eine bessere Auflösung zu bringen und die Auswirkungen auf die berechneten Abzweigungen von Cyanellen und Chloroplasten zu studieren. Zusätzlich wurde eine weitere Sequenz aus Rhodoplasten von von *Corethron criophilum* hinzugefügt.

Diesmal gab es 1393 alignierte Spalten im Alignment, von denen 730 unterschiedliche Spalten durch `pfastDNAml` zur Stammbaumrekonstruktion verwendet wurden. Solche Veränderungen der Anzahl der alignierten Spalten ergeben sich daher, daß die verschiedenen rRNA–Sequenzen an den Enden länger oder kürzer sind als andere.

Aus den Basenfrequenzen 0,266 (A), 0,219 (C), 0,305 (G) und 0,210 (T) wurde als Transition/Transversions–Parameter 1,455 berechnet.

Nach sechs globalen Rearrangements (je 4290 Bäume) und insgesamt 28526 untersuchten Bäumen wurde einer mit einem Log–Likelihood–Wert von -18016,00 gefunden (Abb. 6.12).

Anhand dieses Datensatzes wurde eine Bootstrap–Analyse mit 100 gezogenen Pseudostichproben durchgeführt. Die Bootstrap–Werte, die 60% überschreiten, sind auch in Abb. 6.12 dargestellt.

Analyse mit gewichtetem Datensatz Anhand des Baumes in Abb. 6.12 wurde ein gewichteter Datensatz mit 10 Gewichten erstellt. Die Verteilung der Basen auf die einzelnen Gewichte ist in Abb. 6.13 dargestellt. Der mit

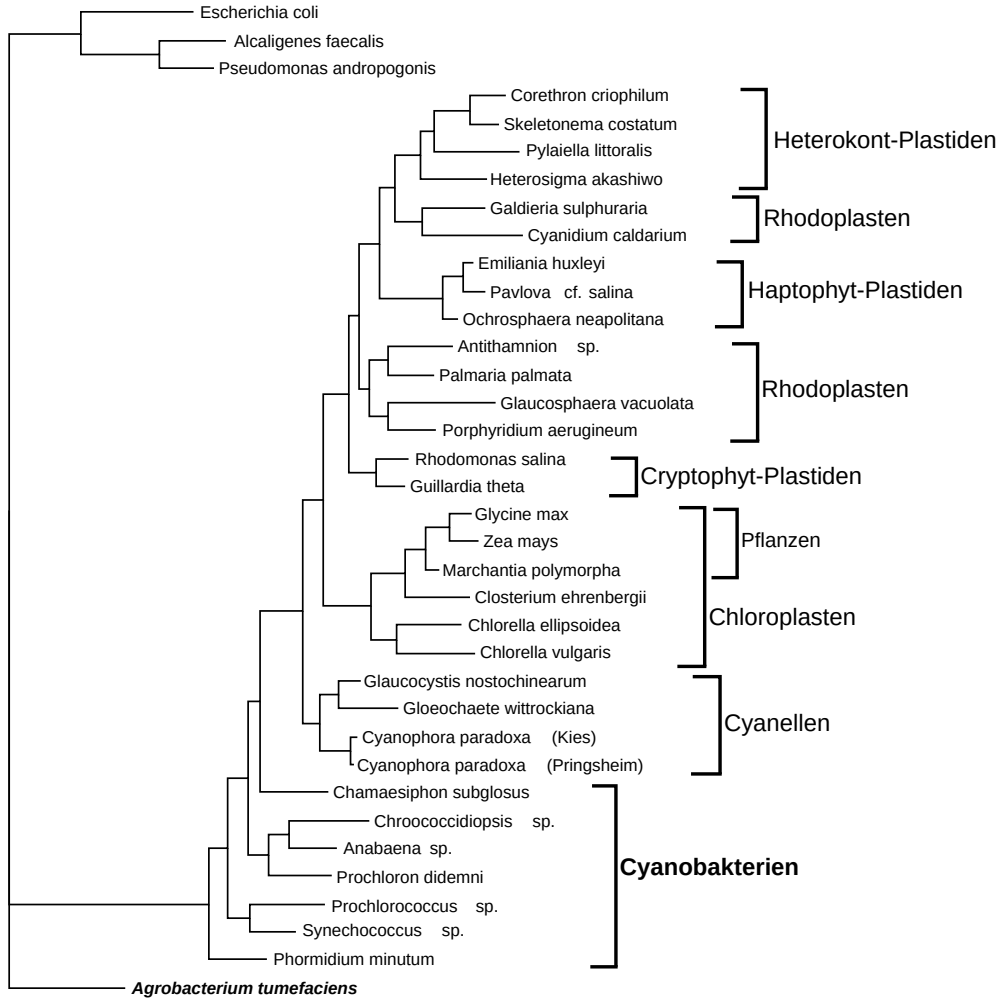


Abbildung 6.10: Phylogenetischer Stammbaum der Plastidenentwicklung; berechnet aus 16S rRNA–Sequenzen (Datensatz *plast1*) mit 5 Gewichtsgruppen mittels `pfastDNAml`

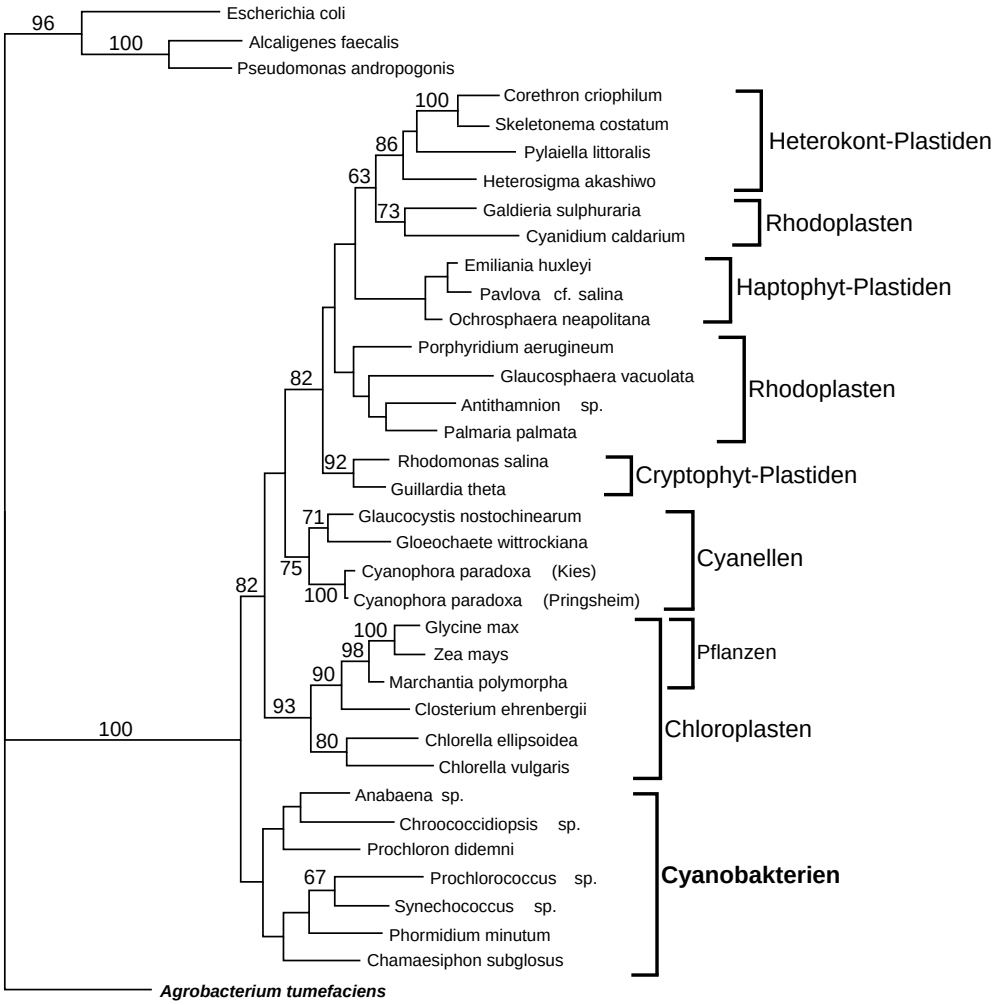


Abbildung 6.11: Phylogenetischer Stammbaum der Plastidenentwicklung; berechnet aus 16S rRNA–Sequenzen (Datensatz *plast1*) mit 10 Gewichtsgruppen mittels `pfastDNAml`; nur Bootstrap–Werte über 60% wurden an den entsprechenden Kanten eingefügt

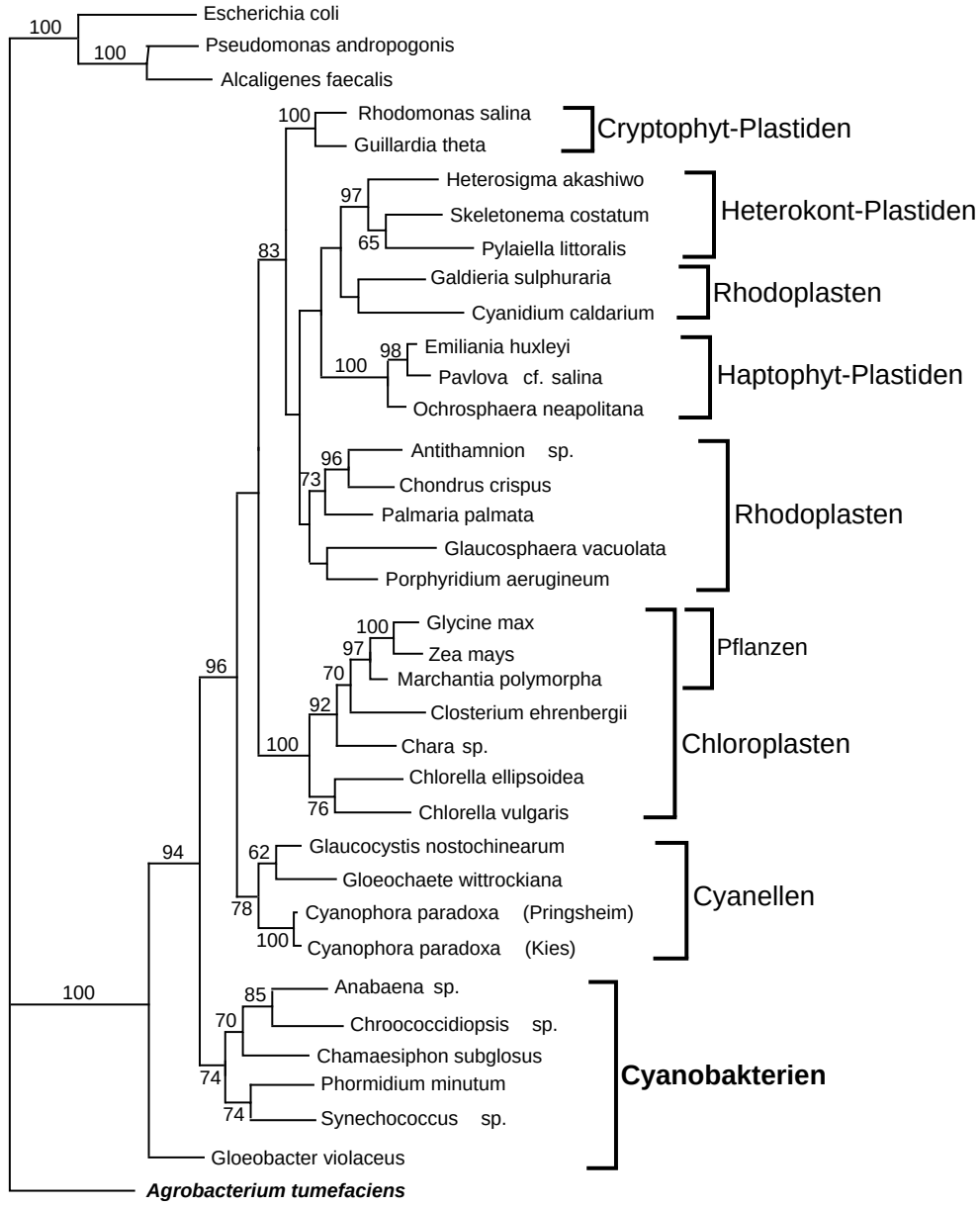


Abbildung 6.12: Phylogenetischer Stammbaum der Plastidenentwicklung; konstruiert aus 16S rRNA–Sequenzen (Datensatz *plast2*) mittels `pfastDNAml`; nur Bootstrap–Werte über 60% wurden an den entsprechenden Kanten eingefügt

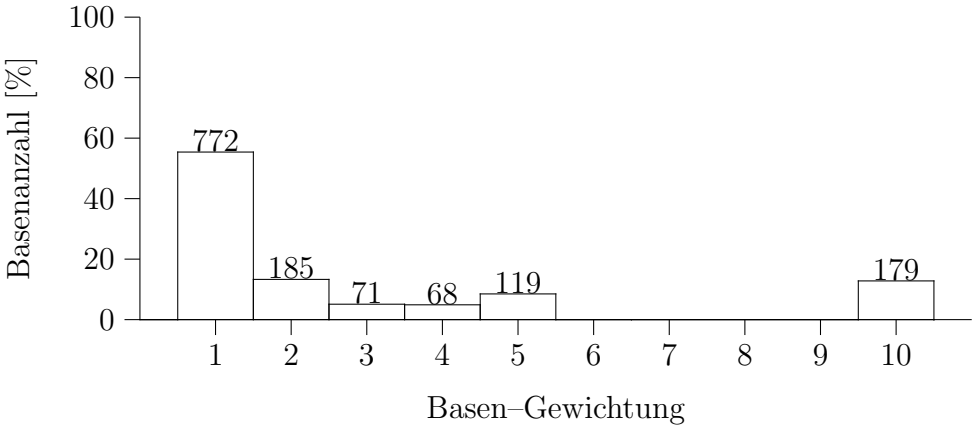


Abbildung 6.13: Verteilung der Basen des Plastidendatensatzes 2 auf die 10 Gewichtsgruppen; Gewichtungsfunktion: $1/S$; Anzahl Basen: 1393

diesem gewichteten Datensatz berechnete Baum hatte einen Log–Likelihood–Wert von -19215,50 (Abb. 6.14). Während der Berechnung des Baumes wurden sechs globale Rearrangements durchgeführt und 28542 Bäume untersucht.

Es zeigte sich diesmal keine Verschiebung der Cyanellen und Chloroplasten gegeneinander. Um den Bereich dieser Verzweigung näher zu untersuchen, wurde auch hier eine Bootstrap–Analyse mit 100 Bootstrap–Stichproben durchgeführt und die gefundenen Werte in den Baum in Abb. 6.14 eingesetzt.

6.1.3 Eukaryotenentwicklung

18S rRNA

Für die Untersuchung der Eukaryotenentwicklung wurden aus dem vorhandenen 18S rRNA–Alignment eine Teilmenge ausgewählt.

Diese Auswahl geschah so, daß alle in der Untersuchung der Plastidenentwicklung wichtigen Organismenlinien vertreten sind, die Zahl der Organismen aber dennoch eine Bootstrap–Analyse zuläßt. Der erstellte Datensatz enthält 36 Spezies mit einem Alignment über 1565 Sequenzstellen, von denen 802 zur Berechnung des Baumes genutzt wurden.

Bei der Erstellung des Baumes (Abb. 6.15) wurden 11622 Bäume untersucht und zwei globale Rearrangements mit je 4290 Bäumen durchgeführt. Es wurden die Basenfrequenzen 0,270 (A), 0,198 (C), 0,265 (G) und 0,266 (T)

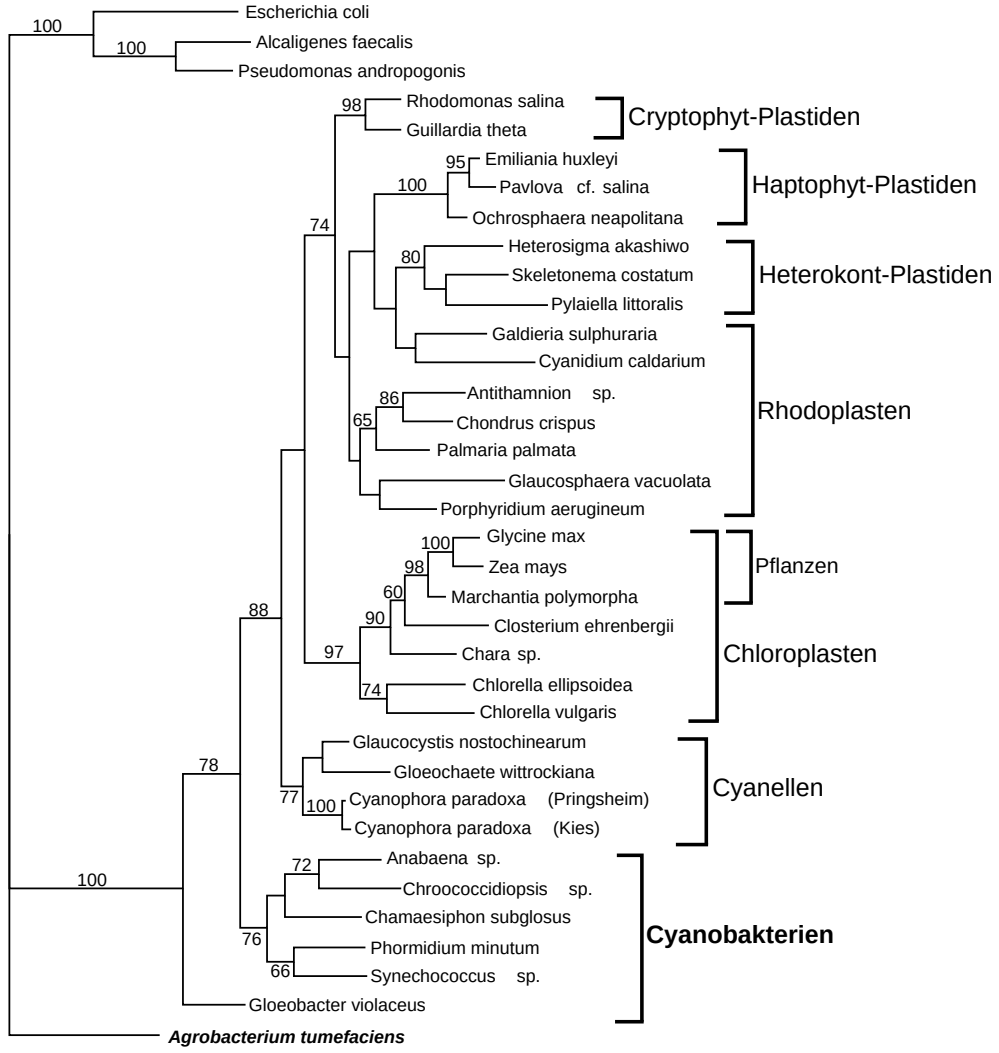


Abbildung 6.14: Phylogenetischer Stammbaum der Plastidenentwicklung; konstruiert aus 16S rRNA–Sequenzen (Datensatz *plast2*) in 10 Gewichtsgruppen mittels `pfastDNAml`; nur Bootstrap–Werte über 60% wurden an den entsprechenden Kanten eingefügt

und der Transition/Transversions–Parameter 1,5068 empirisch aus den Datenmaterial berechnet. Der Log–Likelihood–Wert des Baumes ist -21932,35.

Zu dem Baum wurde anschließend eine Bootstrap–Analyse anhand von 100 Pseudostichproben aus dem Datensatz durchgeführt. Im Konsensusbaum gefundene Bootstrap–Werte über 60% wurden in Abb. 6.15 an den entsprechenden Kanten eingefügt.

Aktin

Neben dem oben beschriebenen 18S rRNA–Datensatz wurde noch ein zweiter Datensatz aus Aktinsequenzen erstellt, um an diesem die Nutzbarkeit von Aktin zur Untersuchung der Plastidenentwicklung zu überprüfen. Hierfür wurden neben anderen Linien vor allem Vertreter aus allen bisher schon untersuchten Organismengruppen mit einfachen Plastiden (Glaucocystophyta, Rotalgen, Grünalgen und Pflanzen) ausgewählt.

Der benutzte Datensatz enthielt 46 Organismen mit einem Alignment von 744 Spalten. Auch hier wurden nur die ersten beiden Codon–Positionen der Aktinsequenzen verwendet. Von diesen 744 Spalten zog `pfastDNAml` 394 zur Analyse heran.

Die empirisch ermittelten Basenfrequenzen waren 0,294 (A), 0,221 (C), 0,258 (G) und 0,227 (T). Hieraus ergab sich ein Transition/Transversions–Parameter von 1,4786. Nach vier globalen Rearrangement mit je 7310 Bäumen und ingesamt 33764 untersuchten Bäumen wurde ein Stammbaum (Abb. 6.16) mit Log–Likelihood–Wert -9228,87 gefunden.

6.2 Diskussion

6.2.1 Bewertung von Bootstrap–Analysen

Die Bootstrap–Analyse ist eine bei phylogenetischen Analysen weitverbreitete Methode, um die Konsistenz eines konstruierten Stammbaumes zu überprüfen. Man findet Bootstrap–Werte in einer Vielzahl von Veröffentlichungen mit Stammbäumen. Die Bootstrap–Analyse ist eine bekannte statistische Methode, um aus einem gegebenen Datensatz neue Stichproben zu generieren, um damit Hypothesen zu testen. Dennoch gibt es keinen echten Konsens über die Bewertung und die Aussagekraft von berechneten Bootstrap–Werten beim Testen phylogenetischer Stammbäume. (Bhattacharya *et al.*, 1996)

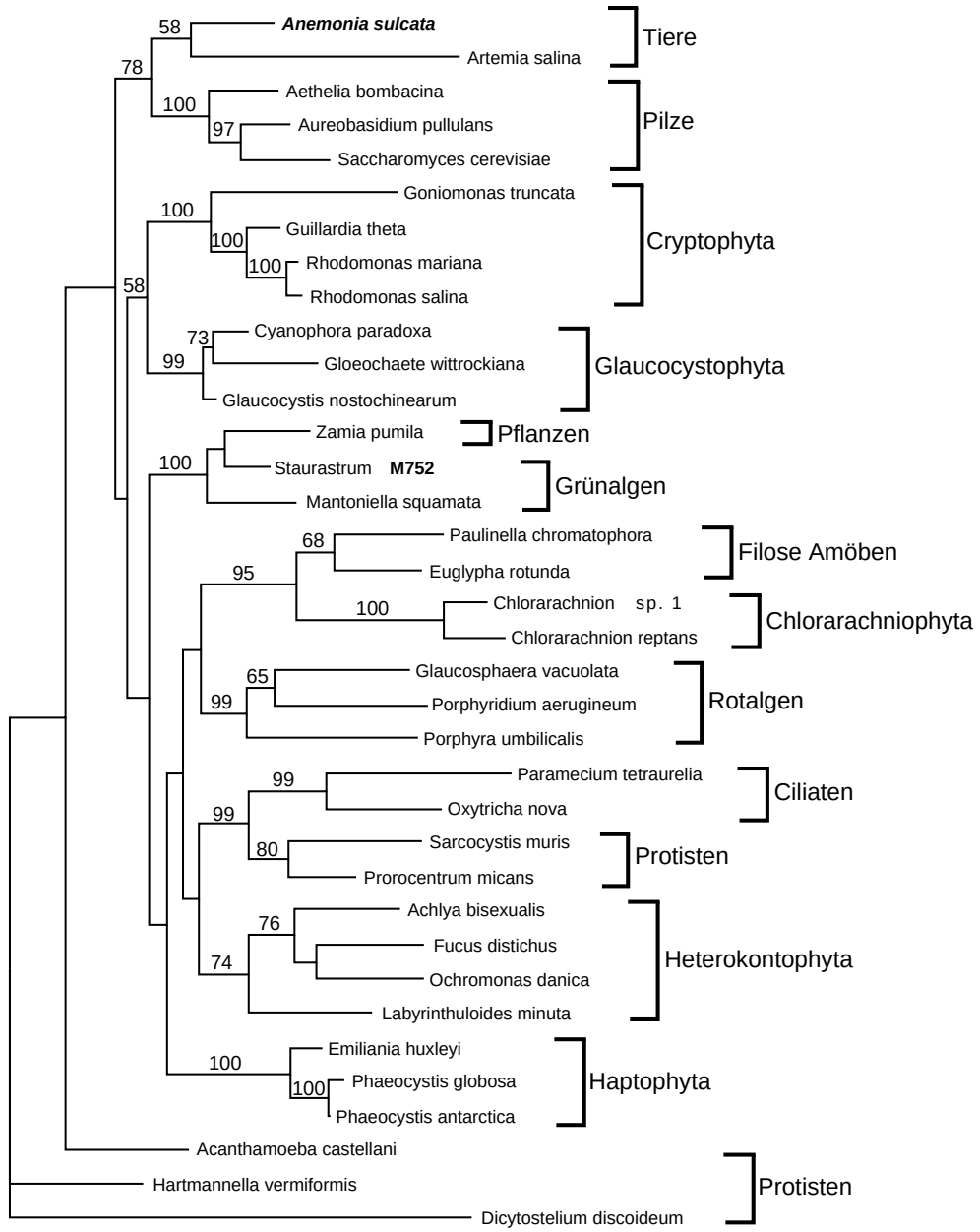


Abbildung 6.15: Phylogenetischer Stammbaum der Eukaryotenentwicklung; konstruiert aus 18S rRNA–Sequenzen mittels `pfastDNAml`; nur Bootstrap–Werte über 60% wurden an den entsprechenden Kanten eingefügt

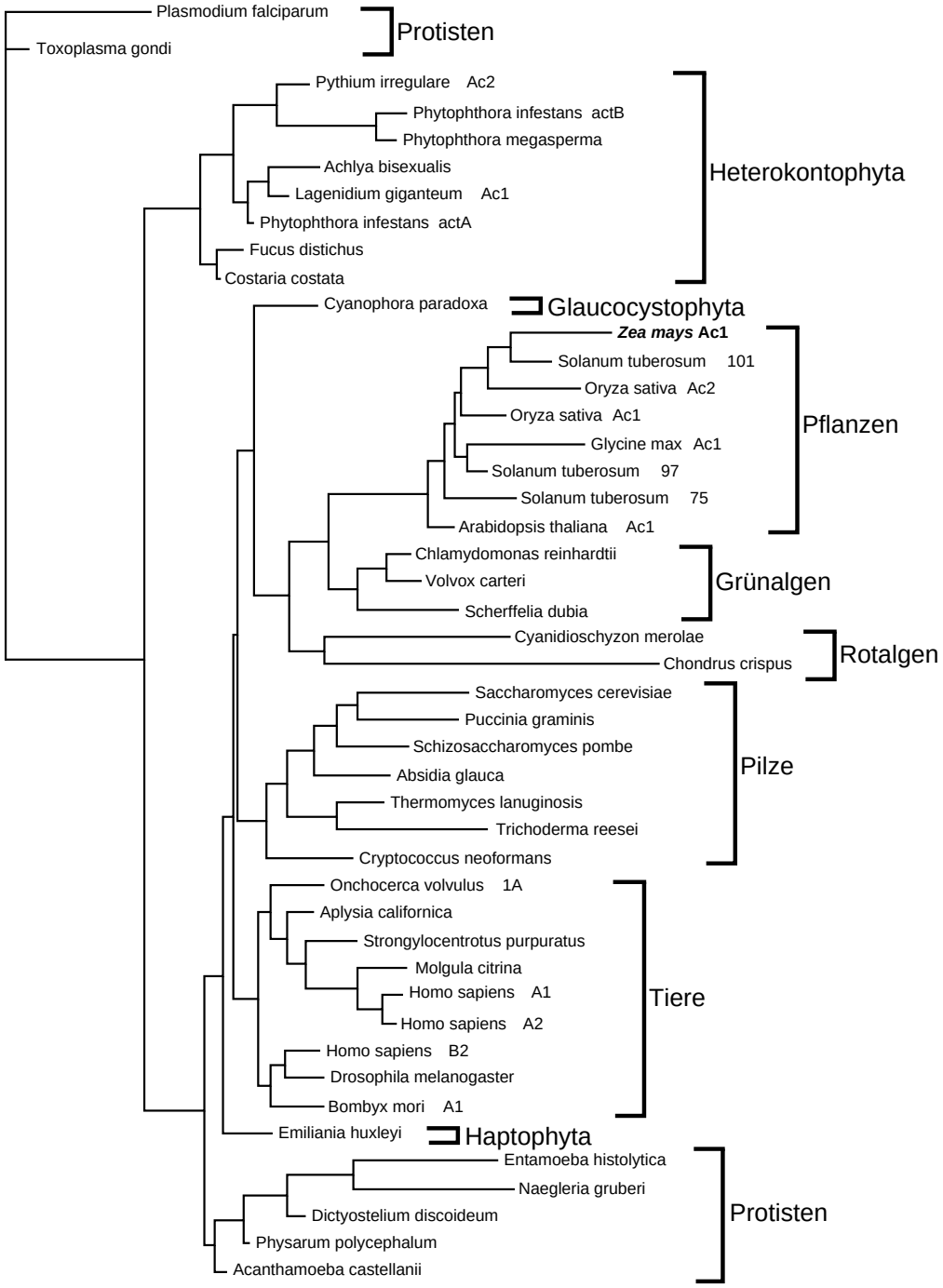


Abbildung 6.16: Phylogenetischer Stammbaum der Eukaryotenentwicklung; berechnet aus Aktinsequenzen mit `pfastDNAml` unter Benutzung der Codon–Positionen 1 und 2

Während Felsenstein (1985, 1988) vorschlug, daß Bootstrap–P–Werte $\geq$ 95% eine robuste Grundlage wären, um Gruppen als monophyletisch anzunehmen, zeigten andere Untersuchungen, daß Bootstrap–Analysen einen niedrigeren Wert als den entsprechenden statistische P–Wert liefern.

Nach Untersuchungen mit simulierten sowie mit echten Sequenzdaten fanden Hillis und Bull (1993), daß „nahezu alle internen Äste mit einen Bootstrap–Wert von mehr als 80% eine echte Klade definierten [...] und mehr als 95% der geschätzten Kladen mit Bootstrap–Konfidenz–Intervallen von über 70% korrekt waren“[1]. Sie kamen zu dem Schluß, daß Bootstrap–Analysen zwar nützliche Hilfsmittel bei der Stammbaumanalyse sind, aber nur bedingt anwendbar sind. In jedem Falle müßten weitere Untersuchungen mit natürlichen Daten durchgeführt werden, um die Bootstrap–Methode als wirkliche Klassifizierungsmethode für die Güte von Gruppierungen in phylogenetischen Bäumen zu etablieren. (Bhattacharya, 1996; Brown, 1994; Efron, 1979; Felsenstein, 1985, 1988; Hillis und Bull, 1993)

In dieser Arbeit wird die Bootstrap–Methode ebenfalls genutzt, um Kanten und die an ihnen hängenden Gruppen auf ihre Konsistenz hin zu überprüfen. Hierfür sind in den konstruierten Bäumen alle Bootstrap–Werte, die größer als 60% sind angegeben, in die Diskussion gehen aber im allgemeinen nur Bootstrap–Werte ab 70% ein.

6.2.2 Gewichtung von Alignments

Mit der Gewichtung der Spalten der benutzten Alignments sollte versucht werden, stark variable Positionen im Alignment durch niedrigere Gewichtung für die Analyse ein geringeres Gewicht zu verleihen und konstante Regionen mit wenigen Mutationen höher zu gewichten. Hierdurch sollten eventuelle Artefakte, die durch hochvariable Basen hervorgerufen werden können, möglichst umgangen werden. Ein krasser Fall solcher Artefakte ist in dem Aktinstammbaum mit allen Codon–Positionen (Abb. 6.2) zu beobachten, in dem z.B. alle tierischen Organismen, im Gegensatz zur Analyse mit nur den ersten beiden Positionen (Abb. 6.1) über den ganzen Stammbaum verteilt wurden.

Außerdem sollte untersucht werden, ob sich die Gewichtung positiv auf die Auflösung der internen Verzweigungspunkte im Baum auswirkt, wenn die Verzweigungspunkte durch kurze Kanten miteinander verbunden sind.

[1] Almost every internal branch with a bootstrap proportion of $> 80\%$ defined a true clade [...] and $> 95\%$ of the estimated clades with bootstrap confidence limits above 70% were correct (Hillis und Bull, 1993, Seite 188)

Bei der Gewichtung des Aktinalignments (Abb. 6.4) ließ sich in keiner Weise eine Verbesserung der Auflösung der internen Verzweigungspunkte feststellen. Nur monophyletische Gruppen, die schon bei der ungewichteten Analyse der ersten zwei Codon–Positionen durch lange Kanten von den anderen getrennt waren, blieben nach der Berechnung mit den gewichteten, vollständigen Sequenzdaten monophyletisch.

Hieraus läßt sich schließen, daß die $1/S$–Gewichtungsmethode nicht geeignet ist, um aus der dritten Codon–Position der Aktinsequenzen noch verwertbare Information herauszufiltern. Dies liegt wahrscheinlich an den, schon im Ergebnisteil angeführten, großen Schwankungen des G+C–Gehalts an der dritten Codon–Position der verschiedenen Organismen.

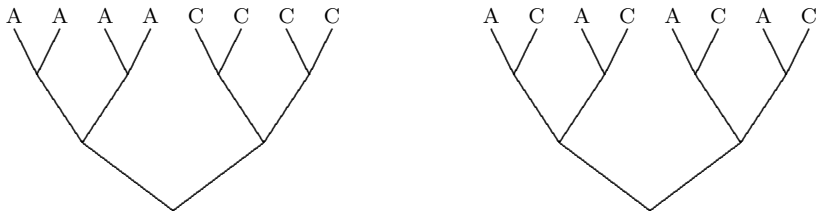
Bäume, die mit dem Plastiden–Datensatz *plast1* und unterschiedlichen Gewichtungen berechnet wurden (Abb. 6.5, 6.9, 6.10 und 6.11), zeigten ebenfalls keine erkennbaren Verbesserungen. In erster Linie werden die Teilbäume der Rhodophyten und der Cyanobakterien, je nach Wahl der Anzahl der Gewichte, umgruppiert oder aufgelöst. Die interessanteste Änderung ist der Tausch der Abzweigungspunkte von Cyanellen und Chloroplasten bei 10 Gewichtsklassen.

Dieser Fall tritt beim Datensatz *plast2* nicht auf. Hier sind die beiden Ergebnisbäume nahezu identisch (Abb. 6.12 und 6.14).

Zieht man nun die Ergebnisse der Bootstrap–Analysen hinzu, so zeigt sich, daß sich die gefundenen Bootstrap–Werte monophyletischer Gruppen in mittels der gewichteten Daten berechneten Bäumen verringern. D.h. daß diese Gruppen nicht so häufig monophyletisch gruppiert werden. Dies läßt den Schluß zu, daß auch hier die Gewichtung mit der $1/S$–Gewichtung eher zu einer Verschlechterung der Ergebnisse führt.

Weitere Probleme dieser Methode lassen sich diskutieren. Betrachtet man die Verteilung der Anzahl der Basen auf die verschiedenen Gewichte bei 10 Gewichtsklassen, so fällt auf, daß die Gewichte 2 bis 4 nicht belegt sind. Dies ist normal, da die höchste Gewichtsklasse für genau eine Mutation ($S = 1$) im MP–Baum steht, und durch den Verlauf der Gewichtungsfunktion $1/S$ werden Spalten mit $S = 2$ Mutationen auf die Gewichtsklasse 5 verteilt. So kommt es eventuell zu einer Überbewertung der Spalten im Alignment, die nur einem Mutationsschritt im MP–Baum unterworfen waren. Eventuell müßte eine mehr lineare Gewichtungsfunktion ausprobiert werden; eine solche war allerdings in den zur Verfügung stehenden Programmen nicht angeboten.

Ein anderes Problem ist die starke Abhängigkeit der Gewichtung von einem vorgegebenen Baum, der in den hier durchgeführten Analysen immer



(a) (b)

Abbildung 6.17: Zwei verschiedene phylogenetische Bäume mit einer Sequenzposition des Alignments an den Blättern. (a) wenig variabel; (b) hoch variabel

aus einer vorher durchgeführten Maximum–Likelihood–Berechnung stammte. Hier wurde also ein mit der ML–Methode berechneter Baum benutzt, um, basierend auf diesem, durch Gewichtung die Aussagekraft der Daten zu verstärken, um einen neuen besseren ML–Baum zu berechnen. Diese Strategie ist fragwürdig, selbst wenn sie häufig angewendet wird, da nicht gesagt ist, daß sich das erhaltenene Ergebnis verbessert.

Der Vorteil der Benutzung eines Baumes gegenüber der Benutzung der reinen Sequenzdaten für die Gewichtung ist einfach zu erklären. Betrachtet man die Bäume in Abb. 6.17, so läßt sich leicht erkennen, daß die im Baum dargestellte Sequenzposition in Abb. 6.17a weit weniger variabel ist, als die Sequenzposition in Abb. 6.17b. In Abb. 6.17a könnte es z.B. sein, daß die Base A hochspezifisch für die eine monophyletische Gruppe ist und C für die andere, während in Abb. 6.17b die Base möglicherweise keine Rolle für die Funktion spielt. Auf Sequenz– bzw. Alignmentebene unterscheiden sich die beiden Fälle allerdings nicht, d.h. die Spalte enthält viermal A und viermal C. In solchen Fällen müssen Methoden scheitern, die die Variabilität und damit die Gewichtung anhand der Sequenzen berechnen. Auf der anderen Seite stellt sich das Problem, woher man für eine solche Gewichtung einen Baum bekommen soll, der möglichst die wahre Information über den Verlauf der Evolution enthält, den wir aber nicht kennen, sondern nach der Gewichtung hoffen besser schätzen zu können. Dieses Dilemma ist mit den gegebenen Mitteln nicht lösbar.

Es kann der Schluß gezogen werden, daß zwar die Notwendigkeit für eine Art der Gewichtung gegeben ist, um die Artefakte möglichst klein zu halten, die aus der unterschiedlichen Variabilität der Sequenzteile entstehen können. Andererseits bietet die hier untersuchte Methode, eine Gewichtung zu erhalten, keine sehr gute Aussicht auf eine verbesserte Ausnutzung der

zugrundeliegenden Daten, da sie selbst mit Fehlern behaftet ist.

6.2.3 Maximum–Likelihood–Methode

Die verwendete Implementierung der Maximum–Likelihood–Methode birgt einige Unsicherheiten, die die Qualität der Ergebnisse beeinflussen können. Durch die Benutzung der Newton–Raphson–Methode zur Berechnung der optimalen Kantenlängen ist nicht ausgeschlossen, daß statt des globalen nur ein lokales Optimum gefunden wird. Untersuchungen von Olsen *et al.* (1994a) nach Einführung dieser Methode in `fastDNAml` zeigten allerdings keine signifikant unterschiedlichen Kantenlängen zu den mit `dnaml` (Felsenstein, 1981, 1990) gefundenen.

Auch durch die Benutzung der Heuristik zum Einschränken der Anzahl der zu untersuchenden Bäume ist nicht gewährleistet, daß wirklich der optimale Baum gefunden wird. Durch die Ausführung der Rearrangements während der Analysen liefern diese jedoch befriedigende Ergebnisse (Felsenstein, 1981). Die Verwendung der `G`–Option, d.h. das Ausführen von globalen Rearrangements am Ende jeder Analyse, ist wichtig zur Verbesserung der Ergebnisse. Dies ist ersichtlich aus der Anzahl der am Ende jeder Analyse durchgeführten globalen Rearrangements (siehe Kap. 6.1). Vor allem bei großen Datensätzen werden durch die globalen Rearrangements noch viele Verbesserungen der Stammbäume gefunden.

Obwohl es nicht gelungen ist, die Qualität der Daten durch Gewichten zu verbessern, und obwohl die oben genannten Fehlerquellen der Implementierung der ML–Methode existieren, ist dennoch zu erwarten, daß das Programm gute Ergebnisse liefert. Untersuchungen von Kuhner und Felsenstein (1994) haben ergeben, daß der Maximum–Likelihood–Ansatz von allen untersuchten Methoden in nahezu allen untersuchten Fällen mit gleichen und unterschiedlichen Substitutionsraten die besten Ergebnisse liefert. Nur bei sehr kurzen Sequenzen mit ungleichen Substitutionsraten an unterschiedlichen Positionen ergaben Distanzmethoden bessere Ergebnisse. In der genannten Untersuchung wurden viele verschiedene Methoden getestet. Das dabei benutzte Programm zur Berechnung von ML–Analysen war `fastDNAml`. Da sich die Programme `fastDNAml` und `pfastDNAml` nicht in der Art der von ihnen durchgeführten Berechnungen unterscheiden, kann dieses Ergebnis auch auf `pfastDNAml` übertragen werden.

Weitere Ungenauigkeiten, die in die Analysen miteingeflossen sein können, ergeben sich einerseits aus eventuellen Fehlern in den verwendeten Alignments und andererseits aus der Vereinfachung des Evolutionsprozesses, wie

er im verwendeten Modell vorgenommen wurde, da der eigentliche Prozeß nicht vollständig verstanden ist.

6.2.4 Entwicklung der Aktingene

Bei den Eukaryoten existieren die unterschiedlichsten Ausprägungen in Bezug auf ihre Ausstattung mit Aktingenen. Es gibt viele Organismengruppen, die nur ein einzelnes Aktingen besitzen. Dazu zählen Pilze, Grünalgen und Ciliaten, aber auch die benutzten Organismen *Cyanophora paradoxa* (Glaucocystophyta), *Giardia lamblia* (ein Protozoon, das keine Mitochondrien besitzt) und verschiedene Heterokontophyta, wie *Costaria costata*, *Fucus distychus*, *Phytophthora megasperma* und *Achlya bisexualis*. Daneben gibt es viele Eukaryoten, die verschiedene Aktintypen und –gene besitzen. Dies sind u.a. viele Heterokontophyta (außer den oben genannten), Pflanzen, Tiere, Schleimpilze und die Organismen *Emiliana huxleyi* (Haptophyta) und *Trypanosoma brucei*. (Bhattacharya und Ehlting, 1995)

Betrachtet man nun den mit der Maximum–Likelihood–Methode berechneten Stammbaum (Abb. 6.1) aus den ersten beiden Codon–Positionen des Aktingens, so zeigt sich, daß die unterschiedlichen Aktine der Tiere monophyletisch mit den Pilzen gruppiert werden, die nur einen einzelnen Aktintyp besitzen. Diese beiden Gruppen werden in diesem Baum auch als monophyletisch mit den Schleimpilzen dargestellt, die wiederum mehrere Aktingenkopien in ihrem Genom tragen. Dieses läßt den Schluß zu, daß entweder die unterschiedlichen Aktine in den Tieren und Schleimpilzen unabhängig voneinander durch Genduplikation entstanden sind oder daß ein gemeinsamer Vorfahr der Pilze diese Genduplikation sekundär wieder verloren hat. Im zweiten Fall wäre allerdings damit zu rechnen, daß nicht die einzelnen Gruppen monophyletisch gruppiert worden wären, sondern zumindest eine gewisse Durchmischung nach unterschiedlichen Aktingenen stattgefunden hätte.

Weiterhin sprechen für die Theorie, daß die unterschiedlichen Aktingene mehrfach unabhängig entstanden sind, noch andere Stellen im Eukaryotenstammbaum, an denen Gruppen mit Aktin in Einzelkopie und solche mit mehreren Aktingenen zusammen einen monophyletischen Ursprung besitzen.

Dieses tritt bei Grünalgen (Einzelkopie) und Pflanzen auf. Außerdem sind unterschiedliche Heterokontophyta (*Fucus distychus* und *Costaria costata*), die Aktin in Einzelkopie besitzen, monophyletisch gruppiert und besitzen mit den anderen Heterokontophyta, die mit wenigen Ausnahmen mehrere Aktintypen besitzen, einen gemeinsamen Vorfahren.

Dieses deckt sich auch mit den Ergebnissen, die von Bhattacharya und Ehlting (1995) mit anderen phylogenetischen Rekonstruktionsmethoden gefunden und anschließend mit der ML–Methode im kleineren Maßstab überprüft wurden.

Problematisch für die Stammbaumanalysen sind die vielen kurzen Kanten innerhalb des Stammbaumes, da diese nur sehr schwer aufzulösen sind. Vor allem lassen sich solche Kanten im Bereich der Tiere beobachten.

Um eine weitere Auflösung der Verwandtschaftsverhältnisse zu erreichen, wurde versucht, die dritte Codon–Position in die Stammbaumanalyse mit einzubeziehen (Abb. 6.2). Bei dem so berechneten Stammbaum sind nur solche Gruppen immer noch monophyletisch, die auch schon im Stammbaum aus den ersten beiden Codon–Positionen durch lange Kanten vom Restbaum getrennt waren. Die Heterokontophyta und die Pflanzen bilden immer noch zwei monophyletische Gruppen. Die anderen Organismen sind jetzt jedoch über weite Teile des Stammbaumes verteilt. Z.B. findet man Organismen aus der Tierwelt an allen möglichen Stellen im Baum und auch andere Organismengruppen sind durch den sehr unterschiedlichen G+C–Gehalt zerrissen worden.

In dieser Form ist der Datensatz daher nicht mit allen Codon–Positionen brauchbar. Es wurde versucht, durch eine $1/S$–Gewichtung mit 10 Gewichtsklassen weitere Informationen aus der dritten Position zu filtern. Doch in dem so rekonstruierten Baum in Abb. 6.4 sind immer noch die Tiere über den gesamten Baum verstreut. Selbst die Grünalgen, die in allen anderen in dieser Arbeit angestellten Analysen mit den Pflanzen eine monophyletische Gruppe bilden, sind im Baum an komplett anderen Stellen als erwartet zu finden. Auch hat sich die Auflösung im Bereich der internen kurzen Kanten nicht verbessert.

Daraus kann geschlossen werden, daß sich zumindest mit dieser Gewichtungsmethode keine weiteren Informationen aus der dritten Codon–Position herausfiltern lassen, da sich der Fehler aufgrund der großen Schwankungen im G+C–Gehalt zu sehr auf die Aussagekraft dieser Position auswirkt.

6.2.5 Plastidenentwicklung

Betrachtet man die verschiedenen berechneten Stammbäume der Plastiden (Abb. 6.5, 6.11, 6.12 und 6.14), so wird bestätigt, daß die Plastiden sich wahrscheinlich aus einem gemeinsamen Vorfahren entwickelt haben, den sie mit den Cyanobakterien teilen. Dies ist daraus ersichtlich, daß alle Plastiden eine monophyletische Gruppe innerhalb der heutigen Cyanobakterien bilden.

Auch der Bootstrap–Wert von 100% in allen Bäumen für den Ast, der die Cyanobakterien und Plastiden von den anderen in der Analyse betrachteten Bakterien trennt, läßt diesen Schluß zu.

Ebenfalls läßt sich aus den Bäumen entnehmen, daß alle einfachen Plastiden wahrscheinlich durch eine einzelne primäre Endosymbiose entstanden sind, da die Plastiden in allen vier Bäumen eine monophyletische Gruppe bilden. Auch die Bootstrap–Werte von 82% (Abb. 6.11) bis 96% (Abb. 6.12) bilden eine starke Unterstützung dieser Aussage.

Über die weitere Entwicklung in Bezug auf die Cyanellen, Chloroplasten und die anderen Plastiden läßt sich aus den Stammbäumen keine sichere Schlußfolgerung ziehen. Zwar zeigen drei der vier Bäume eine Abzweigung der Cyanellen vor der weiteren Entwicklung der anderen Plastiden, doch gibt es in keinem der Bäume eine sichere Unterstützung durch Bootstrap–Werte. Daher sollte die Entwicklung der Plastiden eher als Polytomie der drei nach der Analyse monophyletischen Gruppen Cyanellen, Chloroplasten und aller anderen Plastiden dargestellt werden. Die Ergebnisse geben die Grundlage für eine genauere Auflösung der Entwicklung nicht her.

Im Bereich der Chloroplasten zeigen die Plastiden der Pflanzen einen monophyletischen Ursprung innerhalb der Plastiden der Grünalgen, wobei sie mit den Chloroplasten der höher entwickelten Grünalgen, der Jochalgen, die schon pflanzenähnliche Strukturen bilden, eine monophyletische Gruppe innerhalb der Grünalgenplastiden bilden. In diesem Bereich der Stammbäume sind auch die Bootstrap–Werte relativ hoch, so daß nahezu jede Kante im Chloroplasten–Teilbaum einen Wert von über 70% besitzt. Dies spricht für eine nähere Verwandtschaft der Jochalgen mit den Pflanzen als mit anderen Grünalgen.

Der interessanteste Bereich ist der Teilbaum, in dem die einfachen Plastiden (Rhodoplasten) der Rhodophyten (Rotalgen) und die komplexen Plastiden der Heterokontophyta, Haptophyta und Cryptophyta eine monophyletische Gruppe bilden. Die Monophylie dieser Gruppe hat gute Bootstrap–Unterstützung (ungewichtet: $81-83\%$, gewichtet: $74-82\%$). Innerhalb dieser Gruppe befinden sich weitere monophyletische Gruppen, die innerhalb der Rotalgen beginnen. Da diese monophyletischen Gruppen von komplexen Plastiden der Cryptophyta ($92-100\%$), Haptophyta (100%) und Heterokontophyta ($80-97\%$) innerhalb der Rhodoplasten entstehen und die Gruppierung all dieser zu einer Gruppe ziemlich stark ist, liegt der Schluß nahe, daß alle diese komplexen Plastiden durch sekundäre endosymbiotische Aufnahme von Vorläufern der Rotalgen in diese Organismen entstanden sind.

Da alle diese Gruppen eigene Linien bilden, die innerhalb der Rhodopla-

sten entspringen, handelt es sich bei den komplexen Plastiden wahrscheinlich um unabhängig voneinander entstandene Plastidenlinien. Anderenfalls sollten diese Organismen einen einzelnen Ursprung innerhalb der Rhodoplasten haben.

6.2.6 Entwicklung der plastidentragenden Eukaryoten

Die Untersuchungen und die Bootstrap–Analysen des Datensatzes aus 18S rRNA–Sequenzen aus dem Zellkern verschiedener eukaryotischer Organismen (Abb. 6.15) zeigen zwar, daß sich eine Menge monophyletischer Gruppen entwickelt haben. Die Bootstrap–Werte geben allerdings auch an, daß es nicht möglich ist, eine Aussage über deren genaue Verwandtschaftsverhältnisse untereinander zu treffen. Die Tiere und Pilze werden hier als monophyletische Gruppe bestätigt. Auch die Chlorarachniophyten und die Filose Amöben werden zu einer monophyletischen Gruppe zusammengefaßt.

Den in den Plastiden–Stammbäumen gefundenen Indizien für eine einzige primäre Endosymbiose der einfachen Plastiden widerspricht hier die Ansiedlung der Chlorobionta (Grünalgen und Pflanzen), der Glaucocystophyta und der Rhodophyta in unterschiedlichen Bereichen des Stammbaumes. Da aber die internen Kanten zwischen diesen Gruppen wegen der schlechten Bootstrap–Werte eher zu einer Polytomie zusammengefaßt werden sollten, kann hieraus nur geschlossen werden, daß dieser Bereich des Stammbaumes mit 18S rRNA nicht aufgelöst werden kann. Dieses deckt sich mit Untersuchungen von Bhattacharya und Medlin (1995).

Betrachtet man allerdings den Stammbaum, der aus Aktinsequenzen zur Untersuchung der Plastidenentwicklung berechnet wurde, so zeigt dieser sehr schön eine monophyletische Gruppierung aller Organismenlinien, die einfache Plastiden enthalten. Auch in diesem Teilbaum trennen sich zuerst die cyanellentragenden Glaucocystophyta von den anderen Organismen mit einfachen Plastiden. Diese anderen Organismen spalten sich dann in solche mit Rhodoplasten (Rotalgen) und solche mit Chloroplasten (Grünalgen und Pflanzen). Grünalgen und Pflanzen bilden auch hier eine monophyletische Gruppe. Dieses Ergebnis unterstützt die in den Untersuchungen an 16S rRNA aus Plastiden gefundenen Ergebnisse, die auf einen monophyletischen Ursprung aller einfachen Plastiden hinweisen.

Das hier gefundene Ergebnis zeigt uns, daß es eventuell möglich ist, die Plastidenentwicklung auf der Ebene der Wirtszellen mit Aktinsequenzen genauer zu untersuchen. Hierfür müssen allerdings weitere Analysen mit Datensätzen durchgeführt werden, die im Bereich der photoautotrophen Wirts-

organismen mit einfachen Plastiden eine höhere Dichte an Organismen aufweisen. Zusätzlich müssen weitere Untersuchungen zur Qualität der gefundenen Abzweigungen angestellt werden, um die Aussagekraft des hier gefundenen Ergebnisses zu beleuchten. Dieses Ergebnis läßt allerdings hoffen, daß mit Aktinsequenzen eine höhere Auflösung als bei 18S rRNA–Sequenzen im Bereich der plastidentragenden Eukaryoten möglich ist.

Kapitel 7

Zusammenfassung und Ausblick

7.1 Zusammenfassung

Stammbaumanalysen sind ein wichtiges Hilfsmittel, um die Entstehung heute bestehender Phänomene zu untersuchen und diese besser zu verstehen. In dieser Arbeit ist es gelungen eine statistische Methode, basierend auf dem Maximum–Likelihood–Ansatz, zur Rekonstruktion phylogenetischer Stammbäume aus DNA–Sequenzen für die Anwendung auf größere Datensätze als solche mit 10 bis 15 Sequenzen nutzbar zu machen. Hierzu wurde das erprobte Computerprogramm `fastDNAml` für die Nutzung von Parallelcomputern parallelisiert. Das erhaltene Programm `pfastDNAml` liefert für große Datensätze einen guten Speedup bei der Nutzung einer unterschiedlichen Anzahl von Knoten auf Parallelrechnern.

Mit diesem Programm ist es nun möglich, umfangreiche Untersuchungen mit Sequenz–Alignmentdaten durchzuführen, die vorher nicht im gleichen zeitlichen Rahmen möglich gewesen wären. Die Berechnung eines Stammbaumes, der mit 34 Prozessoren der SP2 30 Minuten benötigt, würde auf einem vergleichbar starken Rechner bei sequentieller Berechnung über 12 Stunden dauern.

Mittels `pfastDNAml` ist es nun auch möglich, Bootstrap–Analysen mit der ML–Methode durchzuführen, die vorher aus Zeitgründen nicht oder mit anderen Methoden durchgeführt worden wären.

Die durchgeführten Analysen stützen die Untersuchungen von Bhattacharya und Ehlting (1995) über die unabhängige Entstehung der einzelnen Aktinfamilien, die hier mit der ML–Methode untersucht wurden.

Auch konnte gezeigt werden, daß die $1/S$–Gewichtungsmethode kein brauchbares Mittel ist, um aus der dritten Codon–Position gegen die hohen Schwankungen im G+C–Gehalt noch brauchbare Information für Stammbaumanalysen herauszufiltern.

Ebenfalls konnte keine Verbesserung der Aussagekraft der benutzten 16S rRNA–Alignments durch die Benutzung dieser Gewichtungsmethode festgestellt werden. Die Möglichkeit, kurze interne Kanten aufzulösen, wurde eher schlechter.

Die Untersuchung von 18S rRNA–Sequenzen aus dem Zellkern von Eukaryoten, ließ keine sicheren Schlüsse über die Entwicklung der plastidentragenden Eukaryoten zu. Es wurden allerdings alle größeren Eukaryotenlinien als monophyletisch bestätigt.

Die wichtigsten Ergebnisse ließen sich im Bereich der Plastidenentwicklung finden. Es wurde nun auch durch die Maximum–Likelihood–Methode bestätigt, daß die Plastiden wahrscheinlich monophyletisch von frühen Cyanobakterien abstammen und wahrscheinlich durch eine einzelne primäre Symbiose entstanden sind. Ein weiteres wichtiges Ergebnis war, daß die unterschiedlichen komplexen Plastiden eine monophyletische Gruppe mit den Rhodoplasten der Rotalgen bilden; d.h. die verschiedenen hier betrachteten komplexen Plastiden sind wahrscheinlich Nachkommen von frühen Rhodophyten, die durch mehrere sekundäre Endosymbiosen in andere Eukaryoten aufgenommen und mit der Zeit zu Organellen reduziert wurden. Diese Ergebnisse werden durch gute Bootstrap–Werte erhärtet, die durch Bootstrap–Analysen mit der Maximum–Likelihood–Methode ermittelt wurden.

Der monophyletische Ursprung der einfachen Plastiden, der in den Analysen mit 16S rRNA unterstützt wurde, konnte mit den Analysen von 18S rRNA nicht auf der Ebene der eukaryotischen Wirtsorganismen bestätigt werden. Bei Untersuchungen mit Aktinsequenzen ließen sich ebenfalls Indizien für einen monophyletischen Ursprung der einfachen Plastiden finden. Hiermit konnte ein Weg aufgezeigt werden, auf dem wahrscheinlich weitere aussichtsreiche Untersuchungen in diesem Bereich möglich sind.

7.2 Ausblick

Um bessere Ergebnisse aus vorhandenen Daten erhalten zu können, sollten die vorausgesetzten Annahmen besser an die Situation der zugrunde liegenden Daten angepaßt werden. Hierzu bieten sich verschiedene Wege an. Ein Weg wäre es, die zugrundeliegenden Modelle zu verbessern. In diesem

Bereich gibt es Trends, auch unterschiedliche Mutationsgeschwindigkeiten an verschiedenen Sequenzstellen und in verschiedenen Taxa zu modellieren. Dies wird unter Zuhilfenahme von hidden–Markov–Modellen realisiert (Felsenstein, 1993). Zusätzlich gibt es ein Modell, das besonders auf die Eigenschaften von RNA eingeht, Sekundärstrukturen auszubilden. Es werden dabei die Beziehungen von Sequenzbereichen, zwischen denen sich Basenpaarungen ausbilden, besonders berücksichtigt. Hierbei werden nicht nur die üblichen Basenpaarungen zwischen Adenin und Uracil bzw. zwischen Guanin und Cytosin betrachtet, sondern auch die auftretenden Paarungen zwischen Guanin und Uracil (Tillier und Collins, 1995).

Ein anderer Weg wäre, die Gewichtungen der einzelnen Sequenzstellen besser zu berechnen. Auf diesem Gebiet haben Van de Peer *et al.* (1996) einen interessanten Ansatz durch „Substitution Rate Calibration“ aufgezeigt.

Solche Methoden müssten verglichen und auf ihre Anwendbarkeit hin überprüft werden.

Ein anderer Punkt, der in der Zukunft eine wichtige Rolle spielen sollte, ist das Eingrenzen von Artefakten durch fehlerhafte Alignments. Auf diesem Sektor gibt es Bestrebungen, die Alignmentberechnungen mit den Stammbaumrekonstruktionen zu koppeln (Vingron und von Haeseler, 1994).

All diese Vorschläge bieten interessante Ansatzpunkte, phylogenetische Analysen zu verbessern und zu verfeinern.

Auch das Programm `pfastDNAml` wird in den nächsten Monaten weiterentwickelt werden. Es wird neben der existierenden PARMACS–Implementierung auch eine Portierung nach MPI geben. MPI (Message Passing Interface) ist der aufkommende Standard im Bereich der Parallelisierungssoftware auf message–passing–Basis (Gropp *et al.*, 1994). Da ein festgeschriebener Standard große Vorteile für die Portabilität der damit entwickelten Software bedeutet und es inzwischen stabile frei verfügbare MPI–Implementierungen gibt, kann so dieses Programm einer größeren Anzahl interessierter Anwender einfacher zur Verfügung gestellt werden.

Nach Abschluß der Portierung soll das Programm `pfastDNAml` im Internet auf dem WorldWideWeb–Server der Abteilung Theoretische Bioinformatik des Deutschen Krebsforschungszentrums (DKFZ) in Heidelberg

URL: `http://www.dkfz-heidelberg.de/tbi`

Interessenten zur Verfügung gestellt werden. Dies ist für Juli 1996 geplant.

Anhang A

pfastDNAml (Bedienungsanleitung und Beschreibung)

A.1 Allgemeines

Da `pfastDNAml` auf dem Programm `fastDNAml` (Olsen *et al.*, 1994a,b) und damit auf `dnaml` (Felsenstein, 1981, 1990) basiert, bleiben die Eigenschaften bezüglich der Bedienung, vor allem aber das Eingabeformat, sehr ähnlich zu denen bei anderen PHYLIP–Programmen. In erster Linie ändert sich der Aufruf des Programms in Abhängigkeit von der zugrundeliegenden Parallelplattform und der dort benutzten Software zur Verteilung der Einzelprogramme auf die einzelnen Prozessoren.

A.2 Programmaufruf

Der bei PHYLIP–Programmen unter UNIX übliche Aufruf

```
pfastDNAml < infile > outfile
```

bleibt für die sequentielle Version von `pfastDNAml` (`sfastDNAml`) erhalten. Bei der Parallelversion hängt der Aufruf, wie oben erwähnt, von der zugrundeliegenden Plattform und der darauf verwendeten Software ab. Um diese Aufrufe zu vereinfachen, existieren für die Plattformen, die mir während meiner Arbeit zur Verfügung standen, Shellskripten, die im allgemeinen als

Parameter nur noch das Eingabefile und, falls dies notwendig ist, die Anzahl der gewünschten Prozessoren übergeben bekommen. Diese Skripten sind auf Seite 102f beschrieben und können mit wenig Aufwand für andere Systeme angepaßt werden. Weitere Instruktionen über den Aufruf von Programmen, die mit PARMACS parallelisiert wurden, findet man in den systemspezifischen Beschreibungen, die mit den PARMACS-Bibliotheken mitgeliefert werden.

Wird PARMACS Version 6.1 oder eine spätere Version verwendet, benötigt `pfastDNAml` keine weiteren Angaben über die Anzahl der gestarteten Parallelprozesse, da diese selbstständig erkannt werden. Bei Benutzung von Version 6.0 muß die Anzahl in einer der beiden Umgebungsvariablen `PFAST_NODES` oder `MP_PROCS` an das Programm übergeben werden. Bei den bereitgestellten Shellskripten wird dieses automatisch getan.

Falls dies auf dem benutzten System möglich ist, kann die Eingabedatei direkt über die Standardeingabe, wie oben beschrieben, an das Programm weitergegeben werden. Es gibt allerdings Systeme, wie z.B. die IBM SP2, bei denen die Standardeingabe hierfür nicht zur Verfügung steht, da sie beim Starten der Paralleljobs durch andere Programme verwendet wird. In diesen Fall muß der Name der Eingabedatei über die Umgebungsvariable `PFAST_INFILE` an das Programm übermittelt werden. Die Umgebungsvariable sollte den vollständigen globalen Pfad enthalten, da anderenfalls vor allem bei Batchjobs Probleme beim Auffinden der Eingabedaten auftreten können. Auch dieses Problem wird im allgemeinen durch die Benutzung der zur Verfügung gestellten Skripten umgangen, da hier verschiedene Fehlermöglichkeiten abgefangen werden.

A.3 Format der Eingabedaten

Zur Übergabe der Sequenzdaten und Parameter wird eine Eingabedatei verwendet. Die Eingabedatei entspricht dem normalerweise bei PHILIP–Programmen verwendeten Format (Felsenstein, 1993). In der einfachsten Form enthält sie nur alignierte Sequenzen verschiedener Organismen sowie deren Anzahl und die Zahl der Spalten des Alignments. Diese können in einem „interleaved“[1] oder sequentiellen Format vorliegen. Am Beginn der ersten Zeile findet man immer zwei Zahlen, die die Anzahl der benutzten Spezies und an zweiter Stelle die Anzahl der Spalten im Alignment angeben. Danach folgen die Sequenzen. Die ersten zehn Buchstaben enthalten den Namen der

[1] interleave (engl.) wörtlich – (mit Zwischenblättern) durchschießen

Spezies oder Sequenz, alle darauffolgenden Zeilen enthalten die Sequenz, bei der nur Buchstaben, Punkte, Fragezeichen und Striche verwendet werden. Ziffern und alle Arten von Leerzeichen (wie z.B. Tabstops) werden ignoriert. Im „interleaved“ Format, dem Standardformat, das der Ausgabe vieler Alignmentprogramme gleicht, wird jeweils eine Zeile für jede Spezies in Blöcken nacheinander abgelegt.

```
  5    42
Sacc.cerev   -UAUCUGGUU GAUCCUGCCA GUAGUCAUAU GCUUGUCUCA AA
Gall.gallu   -NNCCNGGUU GAUCCUGCCA GUAG-CANNN GCUNGUCUCA AA
Homo_sapie   -UACCUGGUU GAUCCUGCCA GUAG-CAUAU GCUUGUCUCA AA
Caen.elega   -UACCUGAUU GAUUCUGUCA GC-GCGAUAU GCUCAAGUAA AA
Dros.melan   -AUUCUGGUU GAUCCUGCCA GUAGUUAUAU GCUUGUCUCA AA

GUAGUCAUAU GCUUGUCUCA AA
GUAG-CANNN GCUNGUCUCA AA
GUAG-CAUAU GCUUGUCUCA AA
GC-GCGAUAU GCUCAAGUAA AA
GUAGUUAUAU GCUUGUCUCA AA
```

Zwischen den einzelnen Blöcken kann jeweils eine Leerzeile eingefügt werden.

Im sequentiellen Format folgt nach dem Namen erst die vollständige Sequenz dieser Spezies, bevor die nächste Spezies aufgelistet wird:

```
  5    42
Sacc.cerev   -UAUCUGGUU GAUCCUGCCA
GUAGUCAUAU GCUUGUCUCA AA
Gall.gallu   -NNCCNGGUU GAUCCUGCCA
GUAG-CANNN GCUNGUCUCA AA
Homo_sapie   -UACCUGGUU GAUCCUGCCA
GUAG-CAUAU GCUUGUCUCA AA
Caen.elega   -UACCUGAUU GAUUCUGUCA
GC-GCGAUAU GCUCAAGUAA AA
Dros.melan   -AUUCUGGUU GAUCCUGCCA
GUAGUUAUAU GCUUGUCUCA AA
```

Die Sequenzen dürfen die in Tabelle A.1 angegebenen Buchstaben enthalten. Die Bedeutungen dieser Buchstaben sind ebenfalls in Tab. A.1 beschrieben. Bei den Buchstaben ist hier sowohl Groß– als auch Kleinschreibung erlaubt.

A	Adenin	
G	Guanin	
C	Cytosin	
T	Thymin	
U	Uracil	
Y	Pyrin	(C, T)
R	Pyrimidin	(A, G)
W	schwache Bindung („weak“)	(A, T)
S	starke Bindung („strong“)	(G, C)
K	Keto	(T, G)
M	Amino	(C, A)
B	nicht A	(C, G, T)
D	nicht C	(A, G, T)
H	nicht G	(A, C, T)
V	nicht T	(A, C, G)
X,N,?	unbekannt	(A, C, G, T)
O,-	Deletion, Lücke	

Tabelle A.1: Die in der Eingabedatei für `pfastDNAml` berücksichtigten Buchstaben zur Bescheibung von Sequenzdaten und deren Bedeutung

Neben den Sequenzdaten können noch weitere Parameter und Optionen nach den Zahlenangaben und vor den Sequenzdaten angegeben werden. Die Optionen folgen nach den Sequenzdaten auch in der ersten Zeile. Zu den Optionen gehörige Parameter folgen in den anschließenden Zeilen.

Folgende Optionen und Parameter sind möglich:

- `1` – Sequenzdaten ausgeben

  Normalerweise werden die Sequenzdaten nicht ausgegeben. Dieses wird durch diese Option aufgehoben.

- `3` – Keinen Baum mit ausgeben

  Bei Angabe dieser Option wird kein Baum nach der Analyse ausgegeben, was normalerweise der Fall wäre.

- `4` – Baum in eine Datei schreiben

  Normalerweise werden Bäume am Ende nicht im Newick–Format in eine Baumdatei geschrieben, es sei denn, diese Option wurde angegeben.

- `B` – Bootstrap–Stichprobe ziehen

Durch diese Option wird veranlaßt, daß `pfastDNAml` eine Bootstrap–Stichprobe von den Eingabedaten generiert und mit dieser Stichprobe die Stammbaumberechnungen durchführt. Neben dieser Option muß in einer der Zeilen bis zu den Sequenzdaten ein sogenannte Random–Number–Seed zu Initialisierung des Zufallsgenerators in der Form „`B` n“ angegeben werden. Bei n handelt es sich um eine ganze Zahl. Um unterschiedliche Stichproben zu erhalten, muß jedesmal ein anderer Random–Number–Seed verwendet werden; d.h. bei gleichen Werten für n ist auch die gezogene Stichprobe dieselbe. Beispiel:

```
5  42  B
B 127
```

- `C` – Kategorien anlegen

  Mit dieser Option ist es möglich, jede Spalte im Alignment einer Kategorie zuzuordnen. Hierzu stehen bis zu 35 Kategorien (1, 2, ..., 9, A, B, ..., Y, Z) zur Verfügung. Jeder der benutzten Kategorien muß ein Gewicht zugeordnet werden, das eine Dezimalzahl sein kann. Diese Gewichte werden in einer Zeile der Form „`C` $na_1a_2 \ldots a_n$“ angegeben. n ist die Anzahl der benutzten Kategorien und a_1 bis a_n die vergebenen Gewichte (a_1 ist das Gewicht für Kategorie 1, a_2 für Kategorie 2 usw.). In *einer* separaten Zeile, die mit dem Schlüsselwort `Categories` beginnt, wird für jede Spalte im Alignment eine Kategorie vergeben. Beispiel:

  ```
  5  42  C
  C 12 0.0625 0.125 0.25 0.5 1 2 4 8 16 32 64 128
  Categories 51388923A11555238BBAAA112348973621123789AB
  ```

- `F` – Empirische Basenfrequenzen

  Diese Option bringt `pfastDNAml` dazu, die Frequenzen der organischen Basen anhand der übergebenen Sequenzdaten zu berechnen. Wird diese Option nicht gesetzt, müssen die Basenfrequenzen in einer eigenen Zeile vor den Sequenzdaten übergeben werden. Beispiel für gleichverteilte Basen:

  ```
  0.25 0.25 0.25 0.25
  ```

- `G` – Globale Rearrangements

Diese Option sorgt dafür, daß die Werte gesetzt werden können, über wieviel Äste hinweg die ausgeführten Rearrangements (siehe Seite 42) durchgeführt werden sollen.

Hierzu kann optional eine Parameterzeile „`G` n_1 n_2“ angegeben werden, in der n_2 angibt, über wieviele Äste die Rearrangements während des Einfügens gehen sollen, und n_1 die Anzahl für das Rearrangement am Ende der Analyse. Hierbei muß $n_1 \leq n_2$ gelten.

Wird n_2 nicht angegeben, wird der Standardwert 1 verwendet. Ist die Parameterzeile gar nicht gesetzt, wird am Ende ein globales Rearrangement durchgeführt.

- `I` – Sequentielles Datenformat (not interleaved)

  Diese Option gibt an, daß die Sequenzdaten im sequentiellen Format vorliegen.

- `J` – Speziesreihenfolge mischen (Jumble)

  Bei Angabe dieser Option werden die Spezies in zufälliger Reihenfolge in die bestehenden Bäume eingefügt. Zur Initialisierung des Zufallsgenerators muß auch hier, wie bei Bootstrap–Analysen, ein Random–Number–Seed übergeben werden. Beispiel:

  ```
  5 42 J
  J 1357
  ```

- `O` – Außengruppe benutzen (Outgroup)

  Über die dazugehörige Parameterzeile wird angegeben, mit welcher Spezies als Außengruppe der berechnete Baum gewurzelt werden soll. Beispiel mit Spezies 3 als Außengruppe:

  ```
  5 42 O
  O 3
  ```

- `Q` – Schnelles Einfügen (Quickadd)

  Diese Option beschleunigt die Baumkonstruktion dadurch, daß beim Einfügen neuer Spezies nicht alle, sondern nur die unmittelbar betroffenen Kanten optimiert werden.

- `R` – Neustart (Restart)

  Mit dieser Option kann `pfastDNAml`, z.B. nach einem Abbruch, erneut gestartet werden. Hierzu muß am Ende der Eingabedatei der Baum,

nach dem die vorherige Berechnung abgebrochen wurde, nach den Sequenzdaten angefügt werden. Diesen Baum findet man am Ende der `checkpoint`–Datei des abgebrochenen Jobs. Von diesem muß vor dem Neustart noch der Kommentar entfernt werden.

- `T` – Transition/Transversions–Rate eingeben

  Zusammen mit dieser Option muß in einer Parameterzeile „`T` m“ die Transition/Transversions–Rate m eingegeben werden. Die Grundeinstellung, wenn diese Option nicht angegeben wird, ist ein Wert von $2,0$.

- `W` – Gewichte eingeben

  Nach der `W`–Option kann man für jede Spalte im Alignment ein ganzzahliges Gewicht zwischen 0 und 35 angeben. Diese Gewichte geben an, wie oft eine Spalte im Alignment in der Berechnung des Likelihoodwertes berücksichtigt werden soll. Auf diese Weise können auch von außen Bootstrap–Stichproben eingegeben werden, ohne daß die Sequenzdaten neu gemischt werden müßten.

  Die Gewichte werden über eine oder mehrere aufeinanderfolgende Parameterzeilen an das Programm übergeben. Beispiel:

  ```
  5  42  W
  Weights     11111000101000011111
              11001010000111111111001
  ```

- `Y` – Baum in eine Datei schreiben

  Schreibt den entgültigen Baum in eine Baumdatei. Standardmäßig passiert dies im Newick–Format, mit der Angabe der Parameterzeile „`Y` 2“ geschieht diese im Prolog–Format.

Bei den optionalen Parameterzeilen ist zu beachten, daß sie nicht als letzte vor den Sequenzdaten stehen dürfen, da sie dann vom Programm eventuell nicht erkannt werden. Es kann mit der `T`–Option und der dann obligatorischen Parameterzeile „`T` 2.0“ oder einer anderen Rate, falls bekannt, vor den Sequenzdaten Abhilfe geschaffen werden.

A.4 Ausgabedateien

Während bei der sequentiellen Programmversion die Ausgaben über Optionen, Parameter und Tätigkeiten des Programms auf der Standardausgabe,

d.h. normalerweise dem Bildschirm, erfolgen, geschieht dies bei der Parallelversion in einer separaten Ausgabedatei mit Namen

`pfastDNAml.mmdd-hhmmss.pid`.

Der mittlere Teil setzt sich aus dem Tag, dem Monat, der Stunde, der Minute und der Sekunde der Startzeit zusammen. `pid` entspricht dem PARMACS–ID des Masterprozesses und ist im allgemeinen gleich der Anzahl der Slave–Prozesse. Bei einem Paralleljob mit 34 Prozessen, also 33 Slave–Prozessen, ist der Master–ID normalerweise ebenfalls 33. Wurde beim Compilieren des Programms im `Makefile` die Option `PM_DEBUG` gesetzt, so gibt auch jeder Slave–Prozeß eine Ausgabe–Datei aus, die sich von der des Master–Prozesses durch den ID (im Beispiel von `0` bis `32`) unterscheidet.

Der Inhalt der Ausgabedatei unterscheidet sich im allgemeinen nicht von der Ausgabe von `fastDNAml`. Zwei Dinge werden allerdings noch zusätzlich ausgegeben. Dies ist zum einen die sogenannte Wallclocktime (die in Wirklichkeit vergangene Zeit) zwischen dem Startpunkt und dem Endpunkt einzelner Berechnungsschritte sowie die für die Analyse benötigte Gesamtzeit.

Außerdem wird bei Bootstrap–Analysen ausgegeben, wie oft jede Spalte im Alignment in der Bootstrap–Pseudostichprobe verwendet wurde. Dadurch ist es dann möglich, die in die Gesamtanalyse eingegangenen Stichproben zu vergleichen.

Neben der normalen Ausgabedatei gibt es noch zwei weitere: eine die Checkpoints, abgespeicherte Zwischenschritte, enthält und eine weitere für den endgültigen Baum, soweit dieser im den Eingabeoptionen gewünscht wurde.

Die in diesen Dateien enthaltenen Bäume bestehen normalerweise aus diesem Format:

```
[Kommentar] (Teilbaum, Teilbaum, Teilbaum):Kantenlaenge;
```

`Teilbaum` besteht entweder aus

```
(Teilbaum, Teilbaum):Kantenlaenge
```

oder aus

```
Speziesname:Kantenlaenge
```

Zur Weiterverarbeitung der Bäume muß eventuell der Kommentar inclusive der eckigen Klammern entfernt werden. Hierfür ist ein Perlskript (Wall und Schwartz, 1991) namens `remrem.pl` (`rem`ove `rem`arks) vorhanden, das aus einer Eingabedatei alle Kommentare herausfiltert und die gefilterte Datei an die Standardausgabe weitergibt, die dann ohne weiteres in eine Datei umgeleitet werden kann. Das so erhaltene Format kann dann von anderen Programmen wie z.B. von den PHYLIP–Programmen `drawgram`, `drawtree` und `retree` weiterbearbeitet werden (Felsenstein, 1993).

Die in der `checkpoint`–Datei enthaltenen Bäume sind die beim Abschluß eines Schrittes, wie Einfügen oder Rearrangement, gefundenen besten Bäume. Diese können bei einer eventuellen Unterbrechung der Berechnung herangezogen werden, um die Berechnung mit der Restart–Option (`R`) neu zu starten.

Die Namen der drei möglichen Ausgabedateien sind also:

```
pfastDNAml.mmdd-hhmmss.pid
treefile.mmdd-hhmmss.pid
checkpoint.mmdd-hhmmss.pid
```

A.5 Installation

Für die Installation auf einem UNIX–System von `pfastDNAml` müssen folgende Voraussetzungen erfüllt sein. Es muß ein ANSI–C–Compiler vorhanden sein. Um die Parallelversion von `pfastDNAml` nutzen zu können, müssen außerdem die PARMACS–Bibliotheken in Version 6.0 oder später installiert sein. Es ist empfehlenswert, das `make`–Programm vom GNU–Project (`gmake`), das im Internet auf vielen FTP–Servern kostenlos erhältlich ist, installiert zu haben, da das mitgelieferte `Makefile` `gmake`spezifische Erweiterungen nutzt. Zusätzlich müssen noch die Programme `tar` und `gzip` zum Archivieren und Komprimieren von Dateien vorhanden sein.

Man packt die zum `pfastDNAml`–Paket gehörigen Dateien in einem beliebigen Verzeichnis mit dem Befehl

```
gzip -d < PFAST-TAR-ARCHIVE.gz | tar xf -
```

aus. `PFAST-TAR-ARCHIVE` entspricht dem Namen der Archivdatei, die das `pfastDNAml`–Paket enthält.

Anschließend sollte die Umgebungsvariable `PFAST_HOME` auf das neuentstandene Verzeichnis gesetzt werden. Dies sollte auch in der entsprechenden Initialisierungsdatei (z.B. `.cshrc` oder `.profile`) geschehen, um diesen Schritt nicht immer von Hand wiederholen zu müssen.

Danach wechselt man in das Verzeichnis `${PFAST_HOME}/src`. Hier befinden sich die Quellcodedateien und das `Makefile`. Das `Makefile` muß nach dem Compilieren an die benutzte Plattform durch Editieren angepaßt werden:

Version Soll eine sequentielle Version entstehen, wird `VERSION` auf `SEQUENTIAL` gesetzt, für die Parallelversion auf `PMSINGLE`.

Compiler Die Variable wird auf den zu benutzenden C–Compiler eingestellt. Für die bisher benutzten Plattformen sind Vorschläge angegeben.

Plattform/Resource Es muß angegeben werden, welche Plattform und Resource benutzt werden soll. Für weitere Informationen sollten die mit PARMACS gelieferten Unterlagen zum System konsultiert werden.

Pfade Zum Compilieren müssen die Pfade, an denen die PARMACS–Bibliotheken zu finden sind, sowie die zu benutzende PARMACS-Bibliothek (`pm6`, `pm6_v` oder `pm6_t`) gesetzt werden. Diese hängen von der PARMACS–Installation ab.

PARMACS Version 6.0 Wird diese Version benutzt, darf die Zeile

```
PM_SPECIAL_OBJS   = pm_reqnodes.o
```

nicht auskommentiert sein.

Nachdem das `Makefile` an die zugrundeliegende Plattform angepaßt wurde, kann mit dem Befehl `gmake`, oder wie das GNU–`make`–Programm auf der benutzten Plattform heißt, compiliert werden. Das so entstandene ausführbare Programm wird nun noch in das Heimatverzeichnis von `pfastDNAml` geschoben.

Anschließend können die in `${PFAST_HOME}/scripts` befindlichen Skripten zum Aufrufen des Programms dienen. Teilweise müssen diese noch an die zugrundeliegende Plattform angepaßt werden. Es ist daher unerläßlich, sich mit dem zu benutzenden System vertraut zu machen.

A.6 Quellcodestruktur

Im Folgenden sind die einzelnen Programm–Module von `pfastDNAml` und deren Inhalte kurz beschrieben.

Das `pfastDNAml`–Modul enthält die Hauptroutinen und die Hauptstruktur des Programms. In `fastDNAml.h` sind alle globalen Variablen und Variablentypen definiert, die innerhalb des gesamten Programms benötigt werden.

Das `beststr`–Modul enthält alle notwendigen Routinen, um die von den Slaves zurückgesandten ML–Bäume zu sortieren und den oder die besten gefundenen Bäume an das Hauptprogramm weiterzugeben. Die Bäume, die zwischen Master und Slaves hin– und hergeschickt werden, sind als Zeichenketten (Strings) kodiert. Daher kommt der Name `beststr`.

Das `besttree`–Modul enthält alle notwendigen Routinen, um die in der sequentiellen Version des Programms in einem Schritt berechneten ML–Bäume zu sortieren und den oder die besten gefundenen Bäume an das Hauptprogramm weiterzugeben. Die hier verwalteten Bäume sind, im Gegensatz zu den von den Slaves in der Parallelversion gelieferten, auch im Speicher als Baumstrukturen abgelegt.

Das `fileinput`–Modul enthält die Routinen, um Optionen, Parameter und Daten aus einer Datei oder der Standardeingabe einzulesen.

Das `strinput`–Modul enthält die Routinen, um Optionen, Parameter und Daten aus einer Datei bzw. der Standardeingabe oder aber einer Zeichenkette einzulesen. Dieses ist durch die gewählte Master–Slave–Übermittlung der Eingabedaten notwendig. Dieses Modul ersetzt nahezu alle Routinen aus dem `fileinput`–Modul.

Das `loadfile`–Modul liest die Eingabedatei in eine temporäre Datei ein und bestimmt deren Größe. Dies wird vor allem für die Versendung der Daten zwischen Master und Slaves benötigt.

Das `misc`–Modul beinhaltet verschiedene Routinen, die an unterschiedlichen Stellen im Programm benötigt werden, wie z.B. Umwandlungsroutinen oder solche zum Durchmustern von Zeichenketten.

Das `pmdispatch`–Modul enthält alle Routinen die zur Kommunikation zwischen Master und Slaves notwendig sind, sowie nahezu alle Aufrufe von Funktionen der PARMACS–Bibliothek. Hierdurch wird gewährleistet, daß `pfastDNAml` leicht durch Auswechseln dieses Moduls auch auf andere Parallelisierungsbibliotheken portiert werden kann.

Das `pm_reqnodes`–Modul sorgt für Kompatibilität zwischen den PARMACS–Versionen 6.1 und 6.0, in der die Funktion `pm_reqnodes` noch implementiert war. Diese Funktion ist allerdings notwendig, um zu erkennen, wie viele Slaveprozesse gestartet wurden. Dieses wird bei Version 6.0 nun durch Auslesen der Environmentvariable `MP_PROCS` (Standard bei Benutzung des Message Passing Environment `mpe` auf der IBM SP2) oder `PFAST_NODES` erreicht. Hierdurch ist `pfastDNAml` auch auf Rechnersystemen mit PARMACS 6.0 lauffähig ist.

A.7 Hilfsprogramme für verschiedene Parallelplattformen

Zur leichteren Bedienung von `pfastDNAml` auf verschiedenen Parallelsystemen wurden Shellskripten angefertigt, die das Starten von Parallel–Jobs, aber auch sequentiellen Programmläufen erleichtern.

Solche Skripten existieren für folgende Plattformen:

1. Workstation–Cluster
   - `WSi_pfastdnaml <input file> <nodes>`
     startet einen Paralleljob mit `nodes` Prozessen (Master + Slaves). Der Job benutzt hierbei `input file` als Eingabedaten. Es wird ein interaktiver Job gestartet.
   - `SEQi_pfastdnaml <input file>`
     startet einen interaktiven, aber sequentiellen Job mit `input file` als Eingabedaten.
2. IBM SP2 mit EASY als Schedulingsystem
   - `EASY_USb_pfastdnaml <input file>`
     startet einen Paralleljob auf den mittels `spsubmit` (EASY) bestellten Prozessoren mit `input file` als Eingabedaten.

- `EASY_USb_pfastdnaml_dir <input dir>`
  funktioniert wie `EASY_USb_pfastdnaml`, nutzt aber alle in `input dir` befindlichen Dateien der Reihe nach als Eingabe.
- `EASY_USb_pfastdnaml_dirmv <input dir>`
  funktioniert wie `EASY_USb_pfastdnaml_dir`, legt aber alle bearbeiteten Dateien in einem Unterverzeichniß `done` ab.
- `EASY_USi_pfastdnaml <input file>`
  startet einen Paralleljob in einer mittels `spsubmit` (EASY) bestellten interaktiven Sitzung. Wieder wird `input file` als Eingabe benutzt.
- `EASY_USb_sfastdnaml <input file>`
  funktioniert wie `EASY_USb_pfastdnaml`, startet aber ein sequentielles `sfastDNAml`.
- `EASY_USi_sfastdnaml <input file>`
  startet interaktiv ein sequentielles `sfastDNAml`. Funktioniert sonst wie `EASY_USi_pfastdnaml`.

3. NEC Cenju 3 (ohne Scheduling)

   - `CJb_pfastdnaml <input file> <nodes>`
     legt einen Batchjob für `nodes` Prozessoren in der Queue[2] ab. `input file` ist Eingabedatei.
   - `CJi_pfastdnaml <input file> <nodes>`
     legt einen interaktiven Job für `nodes` Prozessoren in der Queue ab. `input file` ist Eingabedatei.

[2] Warteschlange

Anhang B

Glossar

Zumeist nach Margulis und Schwartz (1988) und Futuyma (1986)

Aktin Eines der beiden wichtigsten Proteine im Muskel, dessen dünne Filamente es aufbaut (Tierreich); spielt auch bei der Bewegung von (einzelligen) Protisten ein Rolle

Außengruppe Ein Taxon, das von einer Gruppe anderer Taxa divergierte, bevor diese untereinander divergierten.

autotroph Ein Organismus, der selber aus energiearmen anorganischen Ausgangsstoffen unter Ausnutzung des Sonnenlichts (photoautotroph) oder chemischer Oxidationsprozesse (chemoautotroph) die Verbindungen synthetisiert, die er als Energielieferant benötigt.

Bio–, –biose von $\beta\acute{\iota}o\varsigma$ (griech.) – Leben

Chloroplast Grünes, chlorophyllhaltiges, photosynthetisch aktives Zellorganell (Plastid) der Grünalgen und der Pflanzen

Chrom– von $\chi\rho\tilde{\omega}\mu\alpha$ (griech.) – Farbe

Chromatin Grundsubstanz der Chromosomen; besteht aus Nucleinsäure und Protein und läßt sich mit der Fleugenfärbung oder mit Hilfe anderer Kernfarbstoffe nachweisen.

Chromosom Aus Chromatin bestehende Struktur innerhalb des Zellkerns, die normalerweise nur während der Kernteilung sichtbar wird; die Chromosomen sind Träger des größten Teils der zelleigenen Erbinformation, also der Gene.

Codon Auch als Basen–Triplett bezeichnete Folge von drei Nucleotiden in einem Nucleinsäure–Molekül, die für eine Aminosäure codieren.

Crown Group die Gruppe eukaryotischer Organismen, die die Krone des eukaryotischen Baumes bilden (Tiere, Pflanzen, Pilze und Protisten).

Crown Group Radiation der zeitlich relativ beschränkte Bereich, in dem die verschiedenen Linien der Crown Group divergierten.

Cyanelle Plastid der Glaucocystophyta; besitzt eine Peptidoglucan–Zellwand

Divergenz Anhäufung unähnlicher Merkmale in Organismen, die gemeinsame Vorfahren haben, aber generationenlang in grundverschiedenen Umwelten lebten.

DNA Desoxyribonucleinsäure; ein Makromolekül aus linear aufgereihten Nucleotiden, in denen die Erbinformation verschlüsselt ist; vermag sich sebsttätig (autoreplikativ) zu vermehren und dient als Matrize für die mRNA–Synthese.

Endocytose Aufnahme von festen Partikeln oder von Flüssigkeitströpfchen durch Membraneinstülpung.

Endosymbiont Organismus, der innerhalb (intrazellulär oder interzellulär) eines Individuums einer anderen biologischen Art lebt; intrazelluläre Endosymbionten werden auch als Endocytobionten bezeichnet.

Eukaryot Zelle — beziehungsweise ein Organismus, der solche Zellen besitzt — mit einem membranumhüllten Zellkern und verschiedenen Zellorganellen wie Mitochondrien und Plastiden; die im Kern enthaltenen Chromosomen sind von basischen Proteinen (Histonen) umschlossen.

–genie, –genese von *γένεσις* (griech.) – Werden, Entstehen

Genom Komplette genetische Ausstattung eines Organismus; bei Eukaryoten gewöhnlich beschränkt auf die im Zellkern enthaltenen Chromosomensätze (Kerngenom).

hetero– von *ετεροῖος* (griech.) – verschieden(artig)

heterotroph Ein Organismus, der die Verbindungen, die er als Energielieferant benötigt, nicht selbst erzeugt, sondern aus biochemischen Bausteinen aus organischer, letztlich von autotrophen Organismen produzierter, Materie gewinnt.

Histone Basische, positiv geladene Proteine, die die DNA in den Chromosomen umhüllen und reich an Lysin und Arginin sind.

homo– von *ὁμοιος* (griech.) – gleichartig

homolog Bezeichnung für Strukturen oder Merkmale, die sich aus gleichen Vorläufern entwickelt haben, auch wenn sie in Form oder Funktion nun verschieden sind.

Karyo–, –karyot von *κάρυον* (griech.) – Nuß, hier: Zellkern

Keimbahn Zellinie mehrzelliger Organismen, die durch Zellteilung von der Ei bzw. Samenzelle nach der Befruchtung zu den Keimzellen führt.

Keimbahnzelle Zelle eines mehrzelligen Organismus, die in ihrer Nachkommenschaft Keimzellen hat.

Keimzelle geschlechtlich differenzierte Fortpflanzungszelle.

Klade eine Gruppe von Arten, die von einem bestimmten Vorfahren abstammen.

Konvergenz Unabhängige Entwicklung ähnlicher Strukturen mit vergleichbaren Aufgaben oder ähnlichen Merkmalen in Organismen, die nicht näher miteinander verwandt sind.

–logie von *λόγος* (griech.) – Lehre, Wort

Messenger–RNA, mRNA RNA, die unmittelbar an der DNA synthetisiert wird und jeweils die Information für die Bildung eines oder mehrerer Funktionsproteine enthält.

Mitochondrium Zellorganell, in dem die gebundene Energie organischer Nährstoffe über den Citratzyklus und die Atmungskette auf ATP übertragen werden.

mono– von *μόνος* (griech.) – allein

monophyletisch Bezeichnung für ein Merkmal einer Organismengruppe, das sich auf einen gemeinsamen Vorfahren zurückverfolgen läßt, beziehungsweise für die Organismengruppe selbst.

Morpho–, –morph von *μορφή* (griech.) – Gestalt, Form

Morphologie Form und Struktur; Lehre und Beschreibung von Gestalt und Aufbau der Lebewesen.

Nucleoid die nicht von Membranhüllen eingeschlossene DNA enthaltende Struktur der prokaryotischen Zellen („Bakterienchromosom“).

Nucleotid Baustein der Nucleinsäuren; besteht aus einer organischen stickstoffhaltigen Base, einem Zucker (Desoxyribose oder Ribose) und einem Phosphatrest.

Nucleus Zellkern; von mindestens zwei Membranen umhülltes Organell, das den größten Teil der genetischen Information einer eukaryotischen Zelle enthält.

Photosynthese Produktion organischer Stoffe aus Kohlenwasserstoff und Wasser mit Hilfe des Sonnenlichts; dazu fähig sind grüne Pflanzen und Protisten, die mit Chlorophyll ausgestattet sind, sowie einige Prokaryoten.

Phylo– von $\varphi\tilde{\upsilon}\lambda o\nu$ (griech.) – Geschlecht

Phylogenie, Phylogenese Stammesgeschichtliche Entwicklung einer genetisch zusammenhängenden Organismengruppe; auch die schematische Wiedergabe der entsprechenden evolutionären Entfaltung und verwandtschaftlichen Beziehungen.

Plastid Spezialisiertes Zellorganell, das entweder der Photosynthese dient (Rhodoplast, Chloroplast, Cyanelle) oder Speicheraufgaben wahrnimmt (Amyloplast); pigmentierte Plastiden, die sich nicht an der Photosynthese beteiligen, heißen Chromoplasten oder Gerontoplasten.

poly– von $\pi o\lambda\acute{\upsilon}\varsigma$ (griech.) – viel

polyphyletisch Bezeichnung für Merkmale oder auch ganze Gruppen von Organismen, die sich durch konvergente Evolution von verschiedenen Vorfahren ableiten lassen.

Polytomie Eine Verzweigung im Stammbaum, an der ein gemeinsamer Vorfahr mehr als zwei nachkommende Linien hat.

Prokaryot (Einzelliger) Organismus, dessen Zelle(n) keinen Zellkern, keine Zellorganellen und keine basische Proteinhülle um ihre DNA aufweisen.

Rekombination Neukombination von Genen in der Nachkommenschaft, die in dieser Form bei den Eltern nicht kombiniert waren.

rezent In der geologischen Gegenwart lebend (Gegensatz: fossil).

Rhodoplast anderer Ausdruck für die Chloroplasten der Rotalgen (Rhodophyta).

Ribosom Kugeliger Makromolekülkomplex im Cytoplasma, am Endoplasmatischen Reticulum sowie in einigen Zellorganellen (Chloroplasten, Mitochondrien), der aus Ribonucleinsäure und Protein besteht und den Ort der Proteinbiosynthese (Translation) darstellt.

RNA Ribonucleinsäure; Molekül aus linear verknüpften Nucleotiden, das genetische Information speichern kann, in Ribosomen vorkommt (rRNA) und an der Proteinbiosynthese beteiligt ist (siehe auch Messenger–RNA).

Soma, –som von $\sigma\tilde{\omega}\mu\alpha$ (griech.) – Körper

Soma Die Teile des Körpers, die keine genetische Kontinuität aufweisen – im Unterschied zu den an die Folgegeneration weitergegebenen Gameten und ihren Bildungsstätten (Keimbahn), den Gonaden, sowie anderen Vermehrungseinrichtungen wie Sporen und Gemmen.

somatische Zelle Zelle, die sich am Aufbau der Gewebe und Organe des Soma beteiligt; beliebige Körperzelle mit Ausnahme der Keimbahnzellen.

syn–, sym– von $\sigma\acute{\upsilon}\nu$ (griech.) – zusammen

Symbiose Dauerhafte, meist sehr enge Partnerschaft zwischen artverschiedenen Lebewesen zu beiderseitigem Nutzen.

Tax– von $\tau\acute{\alpha}\xi\iota\varsigma$ (griech.) – Ordnung, Anordnung

Taxon (pl. Taxa) taxonomische Einheit (z.B. Reich, Gattung oder Art), der Individuen oder Gruppen von Arten zugeordnet werden.

Taxonomie Systematisch Einordnung der Organismen.

–tomie von $\tau o\mu\acute{\eta}$ (griech.) – Schnitt

Anhang C

Abkürzungen

A Adenin

ANL Argonne National Laboratory

ANSI American National Standard Institute

ATP Adenusintriphosphat

DNA Desoxyribonucleinsäure

C Cytosin

EASY Expandable Argonne Scheduling System

EBI European Bioinformatics Institute

EM Expectation Maximization

EMBL European Molecular Biology Laboratories

FDDI Fibre Distributed Data Interface

G Guanin

Indel Oberbegriff für Insertionen und Deletionen

LAN Local Area Network

ML Maximum Likelihood

MP Maximum Parsimony

MPI Message Passing Interface

mRNA messenger RNA

MSB Minimaler Spannbaum

PC Personal Computer

rDNA ribosomale DNA (DNA–Sequenz für ribosomale Gene)

RDP Ribosomal Database Project

RNA Ribonucleinsäure

rRNA ribosomale RNA

SSU Small Subunit (kleine Untereinheit)

T Thymin

U Uracil

URL Uniform Resouce Locator

UV Ultraviolett

Literaturverzeichnis

Ajioka, J. W., Smoller, R. W., Jones, R. W., Carulli, J. P., Vellek, A. E. C., Garza, D., Link, A. K., Duncan, I. W. and Hartl, D. L. (1991) Drosophila genome project: one–hit coverage in yeast artificial chromosomes. *Chromosoma*, **100**, 495–509. 4

Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K. and Watson, J. D. (1994) *Molecular Biology of the Cell*. Garland Publishing, New York, Third edn.. 7, 23

ANSI (1989) American national standard programming language C. Tech. Rep. X3.159–1989, American National Standard Institute. 45, 46

Ansorge, W., Voss, H., Wiemann, S., Schwager, C., Sproat, B., Zimmermann, J., Stegemann, J., Erfle, H., Hewitt, N. and Rupp, T. (1992) High–throughput automated DNA sequencing facility with fluorescent labels at the European Molecular Biology Laboratory. *Electrophoresis*, **13**, 616–619. 4

Ansorge, W., Zimmermann, J., Erfle, H., Hewitt, N., Rupp, T., Schwager, C., Sproat, B., Stegemann, J. and Voss, H. (1993) Sequencing reactions for ALF (EMBL) automated DNA sequencer. *Methods Mol. Biol.*, **23**, 317–356. 4

de Bellis, G., Consani, I., Caramenti, G., Pergallozzi, R., Debernardi, S., Invernizzi, L. and Luzzana, M. (1994) Mixed–mode fluorescent DNA sequencing. *Biotechniques*, **16**, 1112–1115. 4

Benson, D. A., Boguski, M., Lipman, D. J. and Ostell, J. (1994) GenBank. *Nucleic Acids Res.*, **22**, 3441–3444. 4

Bhattacharya, D. (1996) Analysis of the distribution of bootstrap tree lengths using the maximum parsimony method. *Mol. Phyl. Evol.*, **in Press**. 79

Bhattacharya, D., Damberger, S., Surek, B. and Melkonian, M. (1996) Primary and secondary structure analyses of the rRNA group–I introns of the zygnematales (charophyta). *Curr. Genet.*, **29**, 282–286. 76

Bhattacharya, D. and Ehlting, J. (1995) Actin coding regions: Gene family evolution and use as a phylogenetic marker. *Arch. Protistenkd.*, **145**, 155–164. 23, 61, 83, 84, 88

Bhattacharya, D., Helmchen, T., Bibeau, C. and Melkonian, M. (1995) Comparisons of nuclear-encoded small-subunit ribosomal RNAs reveal the evolutionary position of the Glaucocystophyta. *Mol. Biol. Evol.*, **12**, 415–420. 18

Bhattacharya, D. and Medlin, L. (1995) The phylogeny of plastids: A review based on comparisons of small subunit ribosomal rna coding regions. *J. Phycol.*, **31**, 489–498. 17, 18, 19, 86

Bhattacharya, D., Stickel, S. K. and Sogin, M. L. (1993) Isolation and molecular phylogenetic analysis of actin–coding region from Emiliania huxleyi, a prymnesiophyte alga by reverse transcriptase and PCR–methods. *Mol. Biol. Evol.*, **10**, 689–703. 61

Brandstädt, A. (1994) *Graphen und Algorithmen*. B. G. Teubner, Stuttgart. 8

Brown, J. K. (1994) Bootstrap hypothesis tests for evolutionary trees and other dendrograms. *Proc. Natl. Acad. Sci. USA*, **91**, 12293–12297. 79

Burkhardt, S. (1993) *Parallele Rechnersysteme: Programmierung und Anwendung*. Verlag Technik, Berlin. 15, 56

Cavalli-Sforza, L. L. and Edwards, A. W. F. (1967) Phylogenetic analysis: Models and estimation procedures. *Amer. J. Human. Genet.*, **19**, 233–257. 10

Cormen, T. H., Leiserson, C. E. and Rivest, R. (1990) *Introduction to Algorithms*. MIT Press, Cambridge, Massachusetts. 10, 12, 15, 52

Curry, D. (1989) *Using C on the UNIX System*. O'Reilly and Associates, Sebastopol. 46

Douglas, S. E. and Turner, S. (1991) Molecular evidence for the origin of plastids from a cyanobacterium–like ancestor. *J. Mol. Evol.*, **33**, 267–273. 18

Efron, B. (1979) Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, **7**, 1–26. 43, 79

Emmert, D. B., Stoehr, P. J., Stoesser, G. and Cameron, G. N. (1994) The European Bioinformatics Institute (EBI) databases. *Nucleic Acids Res.*, **22**, 3445–3449. 4

Felsenstein, J. (1978) The number of evolutionary trees. *Syst. Zool.*, **27**, 27–33. 10

Felsenstein, J. (1981) Evolutionary trees from DNA sequences: a maximum likelihood approach. *J. Mol. Evol.*, **17**, 368–376. 8, 9, 10, 12, 14, 16, 24, 36, 37, 40, 82, 91

Felsenstein, J. (1982) Numerical methods for inferring evolutionary trees. *The Quarterly Review of Biology*, **57**, 379–404. 9, 12

Felsenstein, J. (1985) Confidence limits on phylogenies: an approach using the bootstrap. *Evolution*, **39**, 783–791. 43, 79

Felsenstein, J. (1988) Phylogenies from molecular sequences: Inference and reliability. *Annu. Rev. Genet.*, **22**, 521–565. 79

Felsenstein, J. (1990) *PHYLIP manual, version 3.3*. University of Washington, Seattle. 37, 82, 91

Felsenstein, J. (1993) *PHYLIP manual, version 3.5*. University of Washington, Seattle. 33, 43, 60, 90, 92, 99

Fisher, R. A. (1912) The maximum–likelihood–method. *Messenger in Mathematics*, **41**, 155–160. 13

Fitch, W. M. and Margoliash, E. (1967) Construction of phylogenetic trees. *Science*, **155**, 279–284. 13

Frisch, Æ. (1991) *Essential System Administration*. O'Reilly and Associates, Sebastopol. 54

Futuyma, D. J. (1986) *Evolutionary Biology*. Sinnauer Associates, Sunderland, Massachusetts. 1, 3, 5, 20, 104

Gilly, D. (1992) *UNIX in a Nutshell*. O'Reilly and Associates, Sebastopol. 46

Goffeau, A. and Vassarotti, A. (1991) The european project for sequencing the yeast genome. *Res. Microbiol.*, **142**. 4

Goldman, N. (1990) Maximum likelihood inference of phylogenetic trees, with special reference to a poisson process model of DNA substitution and to parsimony analyses. *Syst. Zool.*, **39**, 345–361. 11, 13, 37

Gropp, W., Lusk, E. and Skjellum, A. (1994) *Using MPI*. MIT Press, Cambridge, Massachusetts. 90

Haeckel, E. (1866) *Generelle Morphologie der Organismen: Allgemeine Grundzüge der organischen Formen–Wissenschaft mechanisch begründet durch die von Charles Darvin reformierte Descendenz–Theorie*. Georg Riemer, Berlin. 2, 3, 16, 17

Harbison, S. P. and Steele Jr., G. L. (1991) *C – A Reference Manual*. Prentice Hall, Englewood Cliffs, Third edn.. 46

Hasegawa, M., Kishino, H. and Yano, T.-A. (1985) Dating of the human–ape splitting by a molecular clock of mitochondrial DNA. *J. Mol. Evol.*, **22**, 160–174. 12, 20

Hedges, S. B. (1992) The number of replications needed for accurate estimation of the bootstrap P value in phylogenetic studies. *Mol. Biol. Evol.*, **9**, 366–369. 43

Helmchen, T. A., Bhattacharya, D. and Melkonian, M. (1995) Analyses of ribosomal RNA sequences from glaucocystophyte cyanelles provide new insights into the evolutionary relationship of plastids. *J. Mol. Evol.*, **41**, 203–210. 18

Hillis, D. M. and Bull, J. J. (1993) An empirical test of bootstrapping as a method for assessing confidence in phylogenetic analyses. *Syst. Biol.*, **42**, 182–192. 79

Jukes, T. H. and Cantor, C. R. (1969) Evolution of protein molecules. In Munro, H. N. (ed.), *Mammalian Protein Metabolism*, Academic Press, New York. 11

Jülich, A. (1995) Implementations of BLAST for parallel computers. *CABIOS*, **11**, 3–6. 16

Keller, U. (1995) Pallas GmbH, Brühl. Personal communication. 49, 50

Kerninghan, B. W. and Ritchie, D. M. (1988) *The C Programming Language*. Prentice Hall, Englewood Cliffs, Second edn.. 46

Kimura, M. (1980) A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J. Mol. Evol.*, **16**, 111–120. 11

Kishino, H. and Hasegawa, M. (1989) Evaluation of the maximum likelihood estimate of the evolutionary tree topologies from DNA sequence data, and the branching order in Hominoidea. *J. Mol. Evol.*, **29**, 170–179. 12, 36, 37

Klaeren, H. (1991) *Vom Problem zum Programm*. B. G. Teubner, Stuttgart. 4

Knippers, R. (1995) *Molekulare Genetik*. Georg Thieme, Stuttgart, 6th edn.. 5, 6, 7, 19

Knoll, A. H. (1992) The early evolution of eukaryotes: a geological perspective. *Science*, **256**, 622–627. 17

Kohne, D. E. (1970) Evolution of higher–organism DNA. *Quart. Rev. Biophys.*, **33**, 327–375. 20

Kreyszig (1975) *Statistische Methoden und ihre Anwendung*. Vandenhoeck und Ruprecht, Göttingen, Fifth edn.. 13, 37

Kuhner, M. K. and Felsenstein, J. (1994) A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. *Mol. Biol. Evol.*, **11**, 456–468. 82

Lewin, B. (1994) *Genes V*. Oxford University Press, Oxford. 5

Li, W.-H. and Graur, D. (1991) *Fundamentals of Molecular Evolution*. Sinauer Associates, Sunderland, Massachusetts. 4, 20

Lodish, H., Baltimore, D., Berk, A., Zipursky, S. L., Matsudaira and Darnell, J. (1995) *Molecular Cell Biology*. Scientific American Books, New York, Third edn.. 7, 20, 22

Maddison, W. P. and Maddison, D. R. (1992) *MacClade, Version 3*. Sinauer Associates, Sunderland, Massachusetts. 33, 42, 60

Maier, U.-G., Hofmann, C. J. B., Eschbach, S., Wolters, J. and Igloi, G. L. (1991) Demonstration of nucleomorph–encoded eukaryotic small subunit ribosomal RNA in cryptomonads. *Mol. Gen. Genet.*, **230**, 155–160. 18

Margulis, L. (1970) *Origin of Eukaryotic Cells*. University Press, New Haven. 17

Margulis, L. and Schwartz, K. V. (1988) *Five Kingdoms*. Freeman and Company, New York. 16, 104

Matsuda, H., Hagstrom, R., Overbeek, R. and Kaneda, Y. (1994) Implementation of a parallel processing system for inference of phylogenetic trees. unpublished. 16, 45, 58, 59

Mereschkowsky, C. (1905) Über Natur und Ursprung der Chromatophoren im Pflanzenreiche. *Biol. Zentralblatt*, **25**, 593–604. 17

Mereschkowsky, C. (1910) Theorien der zwei Plasmaarten als Grundlage der Symbiogenesis, einer neuen Lehre der Entstehung der Organismen. *Biol. Zentralblatt*, **30**, 278–303. 17

Nakaya, A., Yamamoto, K. and Yonezawa, A. (1995) RNA secondary structure prediction using highly parallel computers. *CABIOS*, **11**, 685–692. 16

Neyman, J. (1971) Molecular studies of evolution: a source of novel statistical problems. In Gupta, S. S. and Yackel, J. (eds.), *Statistical Decision Theory and Related Topics*, Academic Press, New York. 43

Olsen, G. J., Matsuda, H., Hagstrom, R. and Overbeek, R. (1994a) fastDNAml: A tool for construction of phylogenetic trees of DNA sequences using maximum likelihood. *CABIOS*, **10**, 41–48. 16, 24, 37, 41, 45, 82, 91

Olsen, G. J., Matsuda, H., Hagstrom, R. and Overbeek, R. (1994b) *fastDNAml manual, version 1.0*. 32, 37, 45, 91

Olsen, G. J., Overbeek, R., Larsen, N., Marsh, T. L., McCaughey, M., Maciukenas, M. A., Kuan, W.-M., Macke, T. J., Xing, Y. and Woese, C. R. (1992) The ribosomal database project. *Nucleic Acids Res.*, **20**, 2199–2200. 21, 26

Olsen, G. J. and Woese, C. R. (1993) Ribosomal RNA: a key to phylogeny. *FASEB J.*, **7**, 113–123. 21

Oram, A. and Talbott, S. (1991) *Managing Projects with make*. O'Reilly and Associates, Sebastopol, Second edn.. 46

Ottmann, T. and Widmayer, P. (1993) *Algorithmen und Datenstrukturen*. BI Wissenschaftsverlag, Mannheim. 10, 15

Van de Peer, Y., Van der Auwera, G. and De Wachter, R. (1996) The evolution of stramenopiles and aveolates as derived by "substitution rate calibration" of small ribosomal subunit RNA. *J. Mol. Evol.*, **42**, 000–000, in press. 90

Van de Peer, Y., Van den Broeck, I. and De Wachter, R. (1994) Database on structure of small ribosomal subunit RNA sequences. *Nucleic Acids Res.*, **22**, 3488–3494. 21, 26

Press, W. H., Flannery, B. P., Teukolsky, S. A. and Vetterling, W. T. (1988) *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge. 41

Rice, C. M., Fuchs, R., Higgins, D. G., Stoehr, P. J. and Cameron, G. N. (1993) The EMBL data library. *Nucleic Acids Res.*, **21**, 2967–2971. 4

Rosenblatt, B. (1993) *Learning the Korn Shell*. O'Reilly and Associates, Sebastopol. 54

Saccone, C., Gissi, C., Lanave, C. and Pesole, G. (1995) Molecular classification of living organisms. *J. Mol. Evol.*, **40**, 273–279. 17, 21

Saitou, N. and Nei, M. (1987) The neighbor–joining method: A new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, **4**, 406–425. 13

Sanger, F., Nicklen, S. and Coulsen, A. R. (1977) DNA sequencing with chain–terminating inhibitors. *Proc. Natl. Acad. Sci. USA*, **74**, 5463–5467. 4

Schneider, H.-J. (ed.) (1991) *Lexikon der Informatik und Datenverarbeitung*. Oldenbourg Verlag, München. 36

Stryer, L. (1988) *Biochemistry*. Freeman and Company, New York. 4, 6

Swofford, D. L. and Olsen, G. J. (1990) Phylogeny reconstruction. In Hillis, D. M. and Moritz, C. (eds.), *Molecular Systematics*, Sinauer Associates, Sunderland, Massachusetts. 9, 10, 11, 12, 13, 14, 37

Tillier, E. R. M. and Collins, R. A. (1995) Neighbor joining and maximum likelihood with RNA sequences: Addressing the interdependence of sites. *Mol. Biol. Evol.*, **12**, 7–15. 90

Vingron, M. (1995) Deutsches Krebsforschungszentrum, Heidelberg. Personal communication. 10

Vingron, M. and von Haeseler, A. (1994) Towards integration of multiple alignment and phylogenetic tree construction. Arbeitspapiere der GMD 852, Gesellschaft für Mathematik und Datenverarbeitung, Sankt Augustin. 90

Wainright, P. O., Hinkle, G., Sogin, M. L. and Stickel, S. K. (1993) Monophyletic origins ot the metazoa: An evolutionary link with fungi. *Science*, **260**, 340–342. 17

Wall, L. and Schwartz, R. L. (1991) *Programming perl*. O'Reilly and Associates, Sebastopol. 54, 99

Waterman, M. S. (1995) *Introduction to Computational Biology*. Chapman and Hall, London. 8, 12, 13

Watson, J. D. (1990) The Human Genome Project: Past, present, and future. *Science*, **248**, 44–49. 4

Watson, J. D., Gilman, M., Witkowski, J. and Zoller, M. (1992) *Recombinant DNA*. Freeman and Company, New York, Second edn.. 1, 16, 17, 18

Weberling, F. and Stützel, T. (1993) *Biologische Systematik*. Wiss. Buchges., Darmstadt. viii, 1

Wehner, R. and Gehring, W. (1995) *Zoologie*. Georg Thieme Verlag, Stuttgart, 23rd edn.. 17

Weir, B. S. (ed.) (1990) *Genetic Data Analysis*. Sinauer Associates, Sunderland, Massachusetts. 12

Yang, Z., Goldman, N. and Friday, A. (1994) Comparison of models for nucleotide substitution used in maximum–likelihood phylogenetic estimation. *Mol. Biol. Evol.*, **11**, 316–324. 11

Danksagung

An dieser Stelle möchte ich für die Unterstützung bei der Durchführung dieser Diplomarbeit ganz herzlich danken:

Herrn Prof. Dr. Rainer Schrader für die Betreuung dieses interessanten Themas und die Bereitstellung eines Arbeitsplatzes am Institut für Informatik zur Durchführung meiner Arbeit.

Herrn Prof. Dr. Andreas Radbruch für die Übernahme der Betreuung dieser Arbeit für den Fachbereich Biologie der Universität zu Köln, ohne die diese interdisziplinäre Arbeit nicht möglich gewesen wäre.

Herrn Priv. Doz. Dr. Bernd Thomas für die Bereitschaft, diese Arbeit als Koreferent zu begutachten.

Herrn Dr. Debashish Bhattacharya (MPI für biophysikalische Chemie) für die Betreuung in Sachen phylogenetischer Methoden und die Zeit, die er sich nahm, wann immer fachliche Fragestellungen erörtert werden mußten.

Herrn Dr. Udo Keller (Pallas GmbH, Brühl) für die Betreuung in Sachen Parallelrechner und PARMACS sowie für die Bereitstellung der SUN–Workstation der Arbeitsgruppe Theoretische Biologie der Universität zu Köln.

Herrn Dr. Martin Vingron (DKFZ, Heidelberg) für die fruchtbaren Diskussionen über Stammbaumanalysen und Alignments, sowie für die Anbahnung der Kontakte zur GMD in Sankt Augustin.

Der GMD in Sankt Augustin und der Firma NEC, die mir im Rahmen des Projekts PARAPHYL Parallelrechnerresourcen zur Verfügung gestellt haben.

Weiterhin danke ich Günter Goetz, Heike Hennig–Schmidt, Nicole Rosenbaum, Silvia Weichl und allen anderen, die mir bei der Erstellung dieser Arbeit durch Diskussionen und Anregungen jedweder Art zur Seite gestanden haben.