Phylogenetische Analyse: Numerische Optimierungsmethoden und Anwendung in der Virologie

Von der Fakultät für Mathematik und Physik
der Gottfried Wilhelm Leibniz Universität Hannover

zur Erlangung des Grades einer

Doktorin der Naturwissenschaften

Dr. rer. nat.

genehmigte Dissertation

von

Dipl. math. Sabrina Dreier

geboren am 14. Mai 1980 in Gehrden

(2006)

Referent: Prof. Dr. Gerhard Starke
Korreferentin: Apl. Prof. Dr. Irene Greiser-Wilke
Tag der Promotion: 24. November 2006

Danksagung

Vor allem danke ich Herrn Prof. Dr. Gerhard Starke für die Betreuung dieser Arbeit und für sein freundliches und engagiertes Interesse am Fortschritt der Arbeit. Mit seiner Bereitschaft zu fachlichen und detaillierten Diskussionen hat er viel zum Gelingen dieser Arbeit beigetragen.

Besonders danken möchte ich auch Frau Apl. Prof. Dr. Irene Greiser-Wilke für die Überlassung dieses interessanten und umfassenden Promotionsthemas, sowie für die kompetente und freundschaftliche Betreuung und für die unermüdliche Unterstützung in allen Problemen aus dem Bereich der Virologie.

Herrn Dr. Bernd Zimmermann danke ich für seine konstruktiven Vorschläge und Beantwortung vieler Fragen bezüglich der Datenbank und der Programmierung der zugehörigen Anwendungen.

Allen Mitarbeitern und den Mitdoktoranden aus dem Institut für Virologie danke ich für die freundliche Aufnahme in das Institut und die gute Zusammenarbeit.

Zuletzt bedanken möchte ich mich bei meiner Familie, meinen Freunden und insbesondere Nele Schlichenmaier, die mir während der Anfertigung der Arbeit in vielfältiger Weise zur Seite standen.

Zusammenfassung

Zur Klassifizierung von Viren wird neben klassischen Eigenschaften wie morphologische Merkmale, Genomaufbau und Wirtsspezifität zunehmend auch die genetische Typisierung anhand von Nukleinsäuresequenzen herangezogen. Nach dem *Alignment* und der Berechnung der phylogenetischen Analyse werden diese graphisch als phylogenetische Bäume dargestellt. Die Bäume zeigen die Verwandtschaft zwischen den untersuchten Organismen. Für Mitglieder der Familie *Flaviviridae*, Genus *Pestivirus*, sind weder Taxonomie noch Nomenklatur schlüssig. Bisher werden vier Spezies unterschieden: *Classical swine fever virus* (CSFV; Klassische Schweinepest), *Bovine viral diarrhea virus* 1 (BVDV-1), BVDV-2 und *Border disease virus* (BDV). Während die Klassifizierung von CSFV standardisiert ist, ist bei BVDV und BDV die weitere Einteilung in Genotypen und Subgruppen, die anhand von phylogenetischen Analysen erfolgte, nicht einheitlich. Ziel der Arbeit war, die bisherige Taxonomie auf ihre Gültigkeit zu überprüfen, und Vorschläge zur Vereinheitlichung der Nomenklatur zu erarbeiten. Hierzu wurden Datenbanken mit den verfügbaren epidemiologischen Angaben und den in GenBank verfügbaren Nukleinsäuresequenzen vervollständigt (CSFV-Datenbank) bzw. neu erstellt (BVDV/BDV-Datenbank), und für die nachfolgenden Berechnungen genutzt. Zusätzlich wurde ein Modul programmiert und in die CSFV Datenbank integriert, das eine automatisierte genetische Typisierung neuer Isolate ermöglicht. In diesem Modul wurde der Neighbor-Joining Algorithmus verwendet.
Ein weiterer Algorithmus zur Berechnung phylogenetischer Bäume ist der Maximum-Likelihood Algorithmus. Bestehend aus zwei Teilproblemen ist das Gesamtziel, einen Baum zu finden, der einen maximalen Likelihood-Wert hat. Hierfür benötigt man die optimale Baumtopologie und für diese die optimalen Astlängen. Die Berechnung der Astlängen lässt sich als kontinuierliches Optimierungsproblem mit Nebenbedingungen formulieren, das mit einer projizierten Variante des Verfahrens der konjugierten Gradienten gelöst wird. Die Suche der optimalen Baumtopologie ergibt ein diskretes Optimierungsproblem. Als Lösungsansatz wurde hier ein iterativer Algorithmus verwendet, der in jedem Schritt kleine lokale Veränderungen in der Topologie vornimmt. Durch die Einbettung des kontinuierlichen in das diskrete Optimierungsproblem ergab sich eine Lösung für das Gesamtproblem. Die Effizienz dieser Algorithmen wurde am Beispiel der *Pestivirus*-Sequenzen untersucht.
Für die Überprüfung der Klassifizierung von BVDV und BDV wurden mit den in der BVDV/BDV-Datenbank gespeicherten Sequenzen phylogenetische Bäume berechnet. Zusätzlich wurden die epidemiologischen Daten der Isolate, von denen die Sequenzen stammten, ausgewertet. Es wird vorgeschlagen, BVDV in eine Spezies mit zwei Genotypen einzuordnen. Der Genotyp BVDV-1 konnte dabei in 14 Subgruppen und der Genotyp BVDV-2 in drei Subgruppen eingeteilt werden. Für das BDV wird eine Einteilung in acht Genotypen vorgeschlagen, von denen sich lediglich einer in zwei Subgruppen unterteilen lässt.

Stichworte: Verfahren der konjugierten Gradienten, phylogenetische Analyse, *Pestiviren*

Summary

For classification of viruses, besides the classical properties like morphological features, genome structure and host specificity, genetic typing on the basis of nucleic acid sequences is increasingly being used. After alignment of the sequences and computation of the phylogenetic analysis, the results are displayed graphically as phylogenetic trees. The trees show the relationship between the organisms examined. For members of the family *Flaviviridae*, genus *Pestivirus*, nor taxonomy nor nomenclature of genotypes and subgroups are as yet conclusive. Up to date four species are distinguished: *Classical swine fever virus* (CSFV), *Bovine viral diarrhea virus* 1 (BVDV-1), BVDV-2 and *Border disease virus* (BDV). While classification of CSFV is standardised, the further allocation of BVDV and BDV isolates into genotypes and subgroups, which is performed by genetic typing, is not consistent. One aim of this work was to verify the validity of the current taxonomy, and to suggest a harmonised nomenclature for *Pestiviruses*. For this, databases were updated (CSFV database) or created (BVDV/BDV database), respectively, and used for the subsequent calculations. In the databases the nucleic acid sequences of the virus isolates published in GenBank and the available epidemiological information were stored. In addition, a module allowing the automated genetic typing of new isolates was programmed and integrated into the CSFV database. This module uses the Neighbor-Joining algorithm.

Alternatively, phylogenetic trees can be calculated using the Maximum Likelihood algorithm. Consisting of two subproblems the final aim is to find a tree with a maximal likelihood value. To this end the optimal topology of the tree is needed, and for this topology the optimal branch length has to be determined. The calculation of the branch length is a continuous optimisation problem with constraints. This problem can be solved with a projected variant of the conjugate gradient method. The search for the optimal topology is a discrete optimisation problem. For its solution an iterative algorithm was used, which does local changes in the topology in every step. The combination of the discrete and the continuous optimisation problems solved the complete problem. The efficiency of the algorithm was analysed using the *Pestivirus* sequences as examples.

For verification of the current taxonomy for BVDV and BDV, the phylogenetic trees using the sequences stored in the BVDV/BDV database were calculated, and the corresponding epidemiological data were evaluated. The results lead to the suggestion to classify BVDV as a single species with two genotypes. Further, within genotype BVDV-1 fourteen subgroups could be distinguished, and within BVDV-2, three subgroups. As for BDV, it is suggested to distinguish eight genotypes, and up to date only one of them can be further subdivided into two subgroups.

Keywords: conjugate gradient method, phylogenetic analysis, *Pestivirus*

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

Die Mathematik ist ein wichtiges Hilfsmittel für die anderen (Natur-)Wissenschaften. Die Hauptaufgabe der Angewandten Mathematik ist die Formulierung und Lösung mathematischer Modelle für Vorgänge z.B. aus der Physik, Chemie, dem Ingenieursbereich oder der Biologie. Eine wichtige Anwendung in der Biologie ist die phylogenetische Analyse, die u.a. anhand von Nukleotidsequenzen berechnet werden kann. Dabei ist der erste Schritt der Vergleich der Nukleotidsequenzen (*Alignment*). Hierbei kommen mathematische Verfahren zum Einsatz. Die Ergebnisse werden als phylogenetische Bäume dargestellt, die den Grad der Verwandtschaft graphisch zeigen. Es wurde eine Vielzahl an verschiedenen Methoden zur Berechnung phylogenetischer Bäume entwickelt, weil die möglichen Ausgangsprobleme verschiedene Voraussetzungen liefern und keine der Methoden alle Voraussetzungen erfüllt [Li97]. Die am häufigsten verwendeten Methoden sind der Neighbor-Joining Algorithmus, eine Distanzmatrix-Methode [SN87], und der Maximum-Likelihood Algorithmus [CB00].
In der Virologie wird die phylogenetische Analyse und die sich daraus ergebende genetische Typisierung inzwischen zur Optimierung der Taxonomie verwendet. Neben der taxonomischen Gliederung in Familien, Genera und Spezies werden bei Viren auch Genotypen und Subgruppen unterschieden. Da die genetische Typisierung jedoch nicht standardisiert ist, sind die Bezeichnungen der Genotypen oft nicht einheitlich. Dies wird insbesondere bei den *Pestiviren* deutlich.
Zum Genus *Pestivirus*, Familie *Flaviviridae*, gehören die Viren der Spezies *Classical swine fever virus* (CSFV)), *Bovine viral diarrhea virus* 1 (BVDV-1), BVDV-2, *Border disease virus* (BDV) und bisher noch nicht klassifizierte Spezies unterschiedlicher Herkunft (Giraffe, HoBi, Pronghorn).
In dieser Arbeit wird vor allem der mathematische Hintergrund der phylogenetischen Analyse betrachtet. Zudem wird in zwei praktischen Anwendungen die genetische Typisierung von *Pestiviren* untersucht.
Nach dem zweiten Kapitel, in dem zunächst die Grundlagen erläutert werden, wird im dritten Kapitel ein Algorithmus zur Berechnung von phylogenetischen Bäumen, der Maximum-Likelihood Algorithmus, näher betrachtet. Für eine gegebene Baumtopologie lässt sich die Berechnung der optimalen Astlängen mit dem Maximum-Likelihood Algorithmus als kontinuierliches Optimierungsproblem mit Nebenbedin-

gungen formulieren. Dieses Optimierungsproblem läßt sich in das diskrete Optimierungsproblem, das Finden der optimalen Baumtopologie, einbetten. Während für das diskrete Optimierungsproblem ein iterativer Algorithmus verwendet wurde, der auf lokalen Veränderungen der Baumtopologie basiert, ließ sich das kontinuierliche Problem mit einem projizierten Verfahren der konjugierten Gradienten lösen.
Um die Effizienz der Verfahren zu überprüfen, wurden sie an drei Beispielen aus der Virologie getestet. Hierfür wurden einmal phylogenetische Bäume mit den Sequenzen verschiedener CSFV Isolate und in den anderen beiden Beispielen mit BVDV und BDV Sequenzen berechnet. Die CSFV Sequenzen stammten aus der nicht translatierten Region am 5' Ende des Genoms (5'NTR). Für die beiden Beispiele mit BVDV und BDV wurden das eine Mal Fragmente der Genomregion, die für das E2-Hüllglykoprotein kodiert, und das andere Mal Fragmente der Genomregion, die für das Npro-Protein kodiert, verwendet.
Abschließend wurde noch untersucht, ob die Likelihood-Funktion bei den drei Beispielen in einer Umgebung der Lösung konvex ist.
Das vierte und das fünfte Kapitel beschreibt die genetische Typisierung der *Pestiviren*.
Für CSFV ist die Einteilung in drei Genotypen mit jeweils drei bis vier Subgruppen [PM00] bereits etabliert. Ein Ziel dieser Arbeit war, eine automatisierte genetische Typisierung zu ermöglichen, die einfach zu benutzen ist und schnell Ergebnisse liefert. Dies ist wichtig, weil die Klassische Schweinepest eine der bedeutsamsten Tierseuchen weltweit ist und ihre Seuchenzüge große wirtschaftliche Schäden verursachen. Die genetische Typisierung ist bei einem Primärausbruch gesetzlich vorgeschrieben. Die einzige mögliche Bekämpfungsstrategie ist eine Kombination aus schneller Diagnose und der Tötung der betroffenen Tiere. Die bereits etablierte Einteilung des CSFV in drei Genotypen mit jeweils drei bis vier Subgruppen [PM00] hat einen geographischen Bezug. Man kann daher oft anhand des Genotyps bzw. der Subgruppe den geographischen Ursprung und den Verbreitungsweg der Isolate feststellen. Diesen Zusammenhang machen sich die Epidemiologen zu Nutze, weil man so den Infektionsweg und die weitere Ausbreitung verfolgen kann.
Um die genetische Typisierung zu vereinfachen, gab es bereits eine CSFV-Datenbank [GZ00], die die Sequenzen und epidemiologischen Daten aller bekannten Isolate enthält. Durch die große Anzahl von Isolaten ist eine manuelle Auswahl der Sequenzen für die Berechnung der phylogenetischen Bäume inzwischen unmöglich. Ein Ziel dieser Arbeit war, ein Modul zu programmieren und in die Bedienoberfläche der CSFV-Datenbank einzufügen, das die genetische Typisierung neuer Isolate vollständig automatisch berechnet.
Im Gegensatz zu den CSFV ist für BVDV und BDV die Einteilung in Genotypen und Subgruppen nicht einheitlich. Auch die Frage, ob es sich bei BVDV-1 und BVDV-2 tatsächlich um zwei eigene Spezies handelt, ist noch nicht endgültig geklärt. Zum jetzigen Zeitpunkt werden BVDV-1 und BVDV-2 laut ICTV (International Committee on Taxonomy of Viruses) als eigene Spezies behandelt. BVDV-1 wurde in 11 Gruppen eingeteilt, BVDV-2 in zwei Gruppen [VP01, VD04]. Für BDV wurden insgesamt neun Gruppen beschrieben.

Ein weiteres Ziel dieser Arbeit war, die verfügbaren Sequenzen und die zu den Isolaten gehörenden epidemiologischen Daten in einer Datenbank zu sammeln. Die Daten wurden dann verwendet, um mit Hilfe phylogenetischer Bäume, Sequenzvergleichen und der Auswertungen von epidemiologischen Daten, die vorhandenen Gruppen zu überprüfen und eine einheitliche Einteilung vorzuschlagen. Die Berechnungen ergaben, dass BVDV-1 und BVDV-2 vermutlich keine eigenen Spezies sind, sondern Genotypen der Spezies BVDV. Die phylogenetischen Bäume zeigten, dass bei BVDV-1 14 Subgruppen unterschieden werden können. Drei Subgruppen wurden neu definiert. Für BVDV-2 konnten nur drei Subgruppen gefunden werden, von denen eine neu ist. Zur Spezies BDV gehörten acht Genotypen, wobei bei Genotyp BDV-1 noch zwei Subgruppen unterschieden werden konnten. Die einzelnen Gruppen stimmten mit den publizierten überein. Die Einteilung für BDV ist allerdings nicht ganz sicher, weil nur wenige Isolate vorhanden waren und von einigen Genotypen nur 5'NTR Sequenzen verfügbar waren.

Kapitel 2

Grundlagen

In diesem Kapitel werden die Grundlagen zur Berechnung eines phylogenetischen Baums erläutert, eine allgemeine Einführung zu den *Pestiviren* gegeben und die genetische Typisierung erklärt.

2.1 Pestiviren

Viren sind obligat intrazelluläre Parasiten mit einer Größe von 20-300 nm. Sie besitzen ein Genom, das entweder aus Desoxyribonukleinsäure (*Desoxyribonucleic acid*, DNA) oder aus Ribonukleinsäure (*Ribonucleic acid*, RNA) besteht und von einer Proteinhülle, dem Kapsid, umgeben ist. *Pestiviren* sind sphärische, behüllte RNA-Viren mit einem Durchmesser von 40 bis 60 nm [MP92].

2.1.1 Taxonomie

Das Genus *Pestivirus* bildet zusammen mit den Genera *Flavivirus* (z.B. Gelbfieber) und *Hepacivirus* (humanes *Hepatitis-C-Virus*) die Familie der *Flaviviridae*. Zum Genus *Pestivirus* gehören das *Classical swine fever virus*(CSFV), das *Bovine viral diarrhea virus* (BVDV) und das *Border disease virus* (BDV) [WB85, WB95], sowie die bisher noch nicht klassifizierte Spezies (Giraffe, HoBi, Pronghorn,...).

2.1.2 Genomaufbau und virale Proteine

Das Genom besteht aus einem einzelsträngigen, linearen RNA-Molekül, das plusstrangorientiert vorliegt [MW90]. Die RNA hat eine Länge von ca. 12,5 Kilobasen (kb) und besitzt einen einzigen offenen Leserahmen, der für ein Polyprotein von etwa 3900 Aminosäuren kodiert [CL88]. Der Leserahmen wird am 5' Ende und 3' Ende von nichttranslatierten Regionen (NTR) begrenzt [MH88], die insbesondere am 5' Ende hochkonserviert sind [MR89].
Das Polyprotein wird während und nach der Translation von zellulären und viralen

Proteasen prozessiert [CL88, RU93, MT96]. So entstehen die vier Strukturproteine, zu denen das *core*-Protein und die Glykoproteine Erns, E1 und E2 gehören, sowie die sieben Nichtstrukturproteine Npro, p7, NS2-3, NS4A, NS4B, NS5A, NS5B [CW91] (Abbildung 2.1).

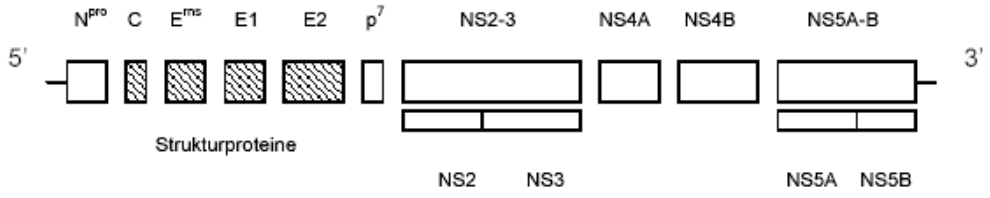


Abbildung 2.1: Genomorganisation des CSFV (modifiziert nach [MT96]).

2.1.3 Wirtsspezifität und Eigenschaften des Virus

Die Viren der Spezies CSFV, BVDV und BDV sind genetisch und antigenisch sehr eng miteinander verwandt. Unterschiede gibt es bei den natürlichen Wirten. Das CSFV kann unter natürlichen Bedingungen nur Wild- und Hausschweine infizieren [Dun70], die anderen *Pestiviren* können sowohl Paarhufer, als auch Schweine infizieren. Das BVDV kann unter natürlichen Bedingungen Rinder, Schafe, Ziegen und Schweine infizieren, aber auch Hirsche, Rehe, Giraffen, Rentiere und anderen Wildwiederkäuer [BK01].
Sowohl beim BVDV als auch bei BDV unterscheidet man aufgrund unterschiedlicher Effekte in Zellkulturen zwischen den nicht cytopatischen (ncp) und den cytopatischen (cp) Biotypen [CF05]. Der wichtigste molekulare Unterschied zwischen dem cp und dem ncp Biotyp ist die Expression des Nichtstrukturproteins NS3. Während beim ncp BVDV ein Protein, das NS2-3, exprimiert wird, werden beim cp BVDV die Proteine NS2 und NS3 einzeln exprimiert.Die Expression des NS3 ist ein molekularer Marker für cp *Pestivirus*-Isolate. Die Expression des NS3 Proteins ist u.a. auf Insertionen in der zellulären RNA zwischen dem NS2 und dem NS3 Gen, Duplikation des NS3 Gens oder Punktmutationen im NS2-3 Gen zurückzuführen. [QM06]
Im 5'NTR beträgt die Identität zwischen BDV und CSFV-Isolaten bis zu 80%, zwischen BDV und BVDV-1 Isolaten bis zu 65% [VS97]. Das Strukturprotein E2 weist die geringste Identität in der Aminosäuresequenz innerhalb des CSFV (80%) und innerhalb des Genus *Pestivirus* (60%) auf. Im Gegensatz dazu ist das Nichtstrukturprotein NS3 mit über 90% das am stärksten konservierte Protein der *Pestiviren* [MR89].

2.1.4 Virus der klassischen Schweinepest

Erstmals beschrieben wurde die Klassische Schweinepest (*Classical swine fever*) 1830 in den USA (Ohio) [KS05]. Verursacht wird sie durch das CSFV. Die Erkrankung ist anzeigepflichtig und in der Europäischen Union (EU) sind die Bekämpfungsmaßnahmen gesetzlich geregelt. Zudem ist die Impfung nur in Ausnahmefällen erlaubt [Ano01]. Das Problem bei der Impfung ist, dass man anschließend ein geimpftes Schwein nicht von einem infizierten Schwein unterscheiden kann und die Produkte von geimpften Tieren daher nicht vermarktet werden können [GM04].
Eine Infektion kann prä- oder postnatal erfolgen. Die Virusausscheidung erfolgt über Sekrete und Exkrete, wie Speichel, Konjunktivalflüssigkeit, Urin und Faezes [Oir99]. Bei einer pränatalen Infektion gelangt das Virus über das Blut akut infizierter tragender Sauen in die Plazenta und von dort in die Feten. Erfolgt die Infektion in der frühen Phase der Trächtigkeit (vor dem 41.Tag) kommt es zum embryonalen/fetalen Tod oder zur Geburt schwacher, unterentwickelter oder toter (z.T. mumifizierter) Ferkel. Bei einer Infektion zwischen dem 50. und 70. Tag der Trächtigkeit kann es zur Geburt persistent virämischer Ferkel kommen, die mit der Zeit Symptome der Spätform von Klassischer Schweinepest (*late onset*-Form) entwickeln. Die Symptome sind u.a. Kümmern, Wachstumsstörungen und Bewegungsstörungen. Diese Tiere überleben mehrere Monate und scheiden den Erreger ständig in hohen Mengen aus [MF03].
Wird das Tier erst nach dem 85.Tag infiziert, werden nicht virämische Ferkel geboren [MP92].
Bei einer postnatalen Infektion erfolgt die Übertragung des CSFV oronasal [Dun70]. Durch direkten Kontakt der Tiere, illegale Verfütterung von virushaltigen, nicht ausreichend erhitzten Speiseabfällen oder durch mechanische Vektoren (Transportfahrzeuge, Geräte, Personen) kann das Virus weiterverbreitet werden [Oir99]. Auch eine Übertragung durch den Deckakt oder die künstliche Besamung ist möglich [SB99]. Eine weitere ständige Infektionsgefahr für die Hausschweinpopulation sind infizierte Wildschweine [Moe00].
Man unterscheidet zwischen der akuten Erkrankung und der chronischen Verlaufsform.
Bei der akuten Erkrankung ist das erste Anzeichen für eine Infektion mit dem CSFV Fieber über 41°C [Oir99]. Weitere Symptome sind Apathie und Anorexie, Konjunktivitis, Erbrechen, Verstopfung gefolgt von Durchfall, vergrößerte Lymphknoten und respiratorische Symptome. Oft treten zudem neurologische Symptome auf. Die klassischen Hautveränderungen in Form von petechialen bis flächenhaften Blutungen treten in der Endphase der Erkrankung auf [Oir99]. Die Letalität schwankt zwischen 30% und 100% [Oir92]. Der Verlauf der Krankheit ist unter anderem auch vom Alter der Tiere abhängig. Junge Tiere erkranken schwerer als ältere Tiere, bei denen die Infektion oft unbemerkt bleibt [MF03]. Bei Tieren, die wieder gesund werden, lassen sich Antikörper zwei bis drei Wochen nach der Infektion nachweisen [MF03].
Von einem chronischen Verlauf spricht man bei einer Krankheitsdauer von mehr als

30 Tagen. Die Symptome sind ähnlich, aber nicht immer so stark ausgeprägt wie bei der akuten Erkrankung. Chronisch erkrankte Schweine können bis zu 100 Tage überleben, aber die Krankheit verläuft immer letal [MF03].
Es kann abwechselnd zu Phasen klinischer Besserung, gefolgt von erneuten Krankheitsschüben, kommen. Bei Nichterkennen der Infektion können diese Tiere eine wichtige Rolle bei der Weiterverbreitung der Seuche spielen, weil sie permanent Virus ausscheiden [DR96].
Die Überlebenszeit des CSFV in der Umwelt ist sehr variabel und wird von verschiedenen Faktoren beeinflusst. In Fleischprodukten kann die Infektiösität des Virus aufgrund des hohen Protein- und Feuchtigkeitsgehaltes lange erhalten bleiben. In gefrorenem Schweinefleisch konnte das Virus noch nach vier Jahren nachgewiesen werden [Edw00], in geräuchertem Fleisch war es nach 17 bis 188 Tagen noch replikationsfähig [KH78].
Ist in einem Betrieb ein Ausbruch von klassischer Schweinepest amtlich festgestellt, müssen alle Tiere des Betriebs getötet werden. Handelt es sich um einen Primärausbruch, so ist eine genetische Typisierung der Erregerisolate gesetzlich vorgeschrieben. Um eine Ausbreitung zu verhindern, wird um den Betrieb ein Sperrbezirk von mindestens drei Kilometern Radius und ein Beobachtungsgebiet gelegt, das zusammen mit dem Sperrbezirk mindestens zehn Kilometer Radius haben muss. Für alle Tiere in diesen Gebieten gilt ein Transportverbot.
Der letzte große Ausbruch 1997/1998 in den Niederlanden zeigt, wie groß die wirtschaftliche Bedeutung von klassischer Schweinepest ist. Während der Epidemie 1997/1998 wurde das Virus in 429 Herden (700.000 Schweine) nachgewiesen. Zur Bekämpfung der Epidemie und infolge des verhängten Transportverbots wurden außerdem weitere 1286 Herden gekeult. Insgesamt wurden so 12.392.000 Schweine getötet. Daraus entstand ein wirtschaftlicher Schaden von 2.3 Millionen US$ [SE00].

2.1.5 Bovine viral diarrhea virus

Bei Rindern kommt die Infektion mit dem BVDV weltweit vor [CF05] und verursacht erhebliche Verluste [FR02]. Erstmalig beschrieben wurde die Bovine Virusdiarrhoe (BVD) 1946 [OM46]. Mittlerweile unterscheidet man bei einer Infektion mit BVDV zwischen der milden akuten BVD, der schweren akuten BVD (SA BVD) und der *Mucosal Disease* (MD) [RN06]. Das Virus wird immer mit Konjunktival-, Nasen,- Uterus- und Scheidensekreten sowie mit Sperma, Harn und Kot ausgeschieden [Lie94].
Die meisten akuten Infektionen immunkompetenter naiver Tiere mit BVDV haben einen milden oder subklinischen Krankheitsverlauf (milde akute BVD). Die Symptome betreffen das Fortplanzungs-, Atmungs-, Immun- und Magen-Darm-System. Eine Infektion eines seronegativen trächtigen Tieres führt dagegen in Abhängigkeit vom Trächtigkeitsstadium zu Fruchtbarkeitsstörungen, wie Umrindern, Missbildungen, Aborten, Geburt lebensschwacher Kälber und dem auftreten persistierend infizierter (PI) Tiere [MG03].
Erfolgt die Infektion in der zweiten Hälfte der Trächtigkeit, kommt es zu Antikörper-

bildung und lebenslanger Immunität [Lie94].
PI-Tiere können bei einer Infektion zwischen dem 100. und 150. Tag der Trächtigkeit auftreten [ER02, CF05]. Sie sind immuntolerant gegenüber dem infizierenden BVDV Stamm [CF05], daher sind auch keine Antikörper nachweisbar [Lie94]. PI-Tiere sind von zentraler Bedeutung für die Verbreitung des Virus, denn sie scheiden es konstant und in hohen Mengen aus. Da das BVD Virus nicht lange außerhalb des Wirtes existieren kann, ist die Replikation in PI-Tieren wichtig für das Überleben des Virus [Bro03]. Nur die Hälfte der PI-Tiere zeigt von Geburt an klinische Symptome, die anderen bleiben unverdächtig. Obwohl 90% der PI-Tiere im ersten Lebensjahr sterben, gibt es in jeder höheren Altersklasse genug Virämiker, um das Infektionsgeschehen aufrecht zu erhalten [MG03].
Die SA BVD wird charakterisiert durch anhaltendes Fieber ($> 40°C$ für drei oder mehr Tage), Thrombozytopenie, Leukopenie, Durchfall, z.T. petchialer und ecchymotic Blutungen auf allen Flächen der Schleimhaut und verschiedenen Organen, respiratorische Beschwerden und hohe Abort und Todesraten [DJ04, Rid03].
Aus Tieren mit MD wird neben dem ncp BVDV auch cp BVDV isoliert. An MD können nur PI-Tiere erkranken [CF05]. Die Mortalität liegt bei 100% [TT94].
Eine Unterteilung von BVDV in BVDV-1 und BVDV-2 erfolgte nach den ersten Ausbrüchen von SA BVD in Nord Amerika in den späten 80ern, wo BVDV-2 das erste Mal isoliert wurde [FR02]. Danach trat BVDV-2 1993 in Quebec auf, wo 25% der Kälber, die hämorrhagische Blutungen hatten, starben [PH94]. Zunächst galt BVDV-2 daher als ein neues und hoch virulentes Virus, dann zeigte sich aber, dass es seit mindestens 20 Jahren in Nord Amerika zirkuliert und die meisten Isolate von Tieren ohne SA BVD stammen. SA BVD wurde bis jetzt immer durch eine Infektion mit BVDV-2 verursacht [Rid03]. Mittlerweile ist sicher, dass alle BVDV-2 Isolate, die zu SA BVD führen, von einem Isolat abstammen [RN06]. Die Einteilung in BVDV-1 und BVDV-2 erfolgte anhand genetischer Unterschiede [CF05]. BVDV-1 enthält alle klassischen Stämme von BVD und die gebräuchlichen Impfstoffe [DJ04]. BVDV-2 kommt neben den USA sporadisch auch in Südamerika, Europa und Japan vor [FR02].
Der Hauptanteil der wirtschaftlichen Schäden entsteht bei dem BVDV durch die pränatalen intrauterinen Infektionen [MG03]. Seit der Entdeckung des Virus 1946, sah man bis vor etwa zwei Jahrzehnten eine Bekämpfung aufgrund des unterschätzten wirtschaftlichen Schadens als überflüssig an. Erst mit zunehmenden Kenntnissen über Pathogenese und der Epidemiologie des Virus hat ein Umdenken eingesetzt. Bei Schäden von ca. 17 bis 170 Euro pro Tier, in Einzelfällen bis 400 Euro pro Kuh oder 17 Euro pro Abkalbung, stellt man sich in den meisten europäischen Ländern nicht mehr die Frage, ob bekämpft werden soll oder nicht, sondern wie bekämpft werden soll [MG03].
Die Seroprävalenz von BVDV in Deutschland liegt bei 60-90%, und 1-2% der Tiere sind PI-Tiere [BK01]. In den anderen EU Ländern schwankt sie zwischen 95% in England und Wales, 64% in Dänemark, 46% in Schweden und weniger als 1% in Finnland. Das primäre Ziel einer Bekämpfung ist die Verhinderung pränataler Infektionen und das Entfernen aller PI-Tiere aus einer Herde. Die Identifizierung

der PI-Tiere ist z.B. mit einer RT-PCR (Polymerase-Kettenreaktion nach Reverser Transkription des RNA-Genoms) möglich und relativ einfach. Eines der ersten Länder, das ein Bekämpfungsprogramm entwickelte und durchführte, war Schweden. Während 1993 nur 35% der Herden seronegativ waren, waren 2001 bereits 87% BVDV-frei. Basis des schwedischen Programms war der Verzicht auf die Impfung und eine zunächst freiwillige Teilnahme [GG03].

2.1.6 Border disease virus

Border disease (BD) ist eine virusbedingte Allgemeininfektion trächtiger Schafe und Ziegen. Erstmalig beobachtet wurde BD im Grenzbereich (*border*) zwischen England und Wales 1959. Bei Schafen tritt BD weltweit auf. Die Prevalenz variiert in den verschiedenen Ländern von 5 bis 50%. Bei Ziegen ist die Krankheit selten und durch Aborte charakterisiert [NG98].
Verursacht wird BD durch das BDV. Allerdings kann auch eine Infektion mit BVDV bei Schafen zu BD ähnlichen Symptomen führen [MG05].
Die meisten BDV Isolate kommen als ncp Biotyp vor [MP04b]. Gesunde Schafherden werden meistens durch die Zuführung persistent infizierter und immuntoleranter Dauerausscheider unter den Schafen sowie durch direkten Kontakt mit BVDV-ausscheidenden Rindern infiziert [Lie94]. Bei gesunden neugeborenen und adulten Schafen verläuft die Infektion meist symptomlos. Einige Isolate verursachen aber auch einen Krankheitsverlauf mit Symptomen wie hohem Fieber, Konjuktivitis, Durchfall und einer Mortalität von 50% bei jungen Lämmern. Das Virus wird mit Konjunktival-, Nasen,- Uterus- und Scheidensekreten sowie mit Sperma, Harn und Kot ausgeschieden [Lie94].
Bedeutender ist die Infektion von trächtigen Mutterschafen. Das Virus wird durch die Plazenta auf den Fetus übertragen. Abhängig vom Trächtigkeitstadium kommt es zum embryonalen Frühtod mit Resorption der Frucht, Aborten, Totgeburten oder zur Geburt kleiner und schwacher Lämmer [NG98].
Diese Lämmer zeigen die Symptome des *Hairy-shaker*-Syndroms [MG05], gekennzeichnet durch haarigen Vlies, niedriges Geburtsgewicht, Ataxie und unterschiedlich starkem tonisch-klonischen Tremor an Gliedmaßen, Rücken, Schwanz, Ohren und in Einzelfällen auch am ganzen Körper. Daneben sind Schwierigkeiten beim Aufstehen und Saugen zu beobachten. Ebenso können Missbildungen wie verkürzte Gliedmaßenknochen, vorgewölbte Stirnpartie und kurzer, krummer Rücken auftreten. Die Lämmer verenden größtenteils während der Saugperiode. Bei den Überlebenden verschwinden die Symptome innerhalb von 3 Monaten [NG98].
Vor allem bei einer Infektion im zweiten Drittel der Trächtigkeit werden die Lämmer oft als PI-Tiere geboren. Sie erreichen ein Alter von bis zu 5,5 Jahren und bleiben lange klinisch unauffällig [SK06]. PI Tiere bilden das wichtigste Virusreservoir zur Aufrechterhaltung der Herdeninfektion und sind besonders empfänglich gegenüber den cp Biotypen des Erregers. Wie beim Rind führt eine Infektion eines PI-Schafes mit einem cp BDV zu einer MD ähnlichen Erkrankung, die immer tödlich verläuft [MP04b].

Es gibt keine Therapie der BD. Die einzige Möglichkeit der Eindämmung ist das Entfernen aller PI-Tiere aus der Herde und die Vermeidung von Kontakt zu BVDV-infizierten Rindern. Eine Impfung der Mutterschafe gewährleistet eine höhere Schutzwirkung, vorausgesetzt, es handelt sich um einen polyvalenten Impfstoff mit Antigenen des BDV, BVDV-1 und BVDV-2 [Lie94]. Der Hauptanteil der wirtschaftlichen Schäden entsteht auch bei BD durch die pränatalen, intrauterinen Infektionen [MG03].

2.2 Genetische Typisierung von Viren

Das Ziel einer genetischen Typisierung ist die korrekte Berechnung und Darstellung der Verwandtschaftsverhältnisse für die betrachteten Organismen. Um sie durchzuführen, können unterschiedliche Merkmale der Organismen herangezogen werden. In der Virologie werden meist Nukleotidsequenzen definierter Genomabschnitte verwendet.
Die DNA ist das genetische Material aller bekannten Organismen und vieler Viren. Es gibt jedoch Viren, die RNA als genetisches Material enthalten. Das zugrundeliegende Prinzip der Natur besagt daher, dass das genetische Material immer eine Nukleinsäure ist. Eine Nukleinsäure besteht aus chemisch miteinander verbundenen Nukleotiden. Jedes Nukleotid enthält eine stickstoffhaltige Base, einen Zucker mit fünf Kohlenstoffatomen in Ringform und eine Phosphatgruppe. Von den stickstoffhaltigen Basen gibt es zwei Typen, die Pyrimidine und die Purine. Jede Nukleinsäure wird aus lediglich vier verschiedenen Basen synthetisiert. Die DNA enthält die Purine A (Adenin) und G (Guanin) und die Pyrimidine C (Cytosin) und T (Thymin). Bei der RNA ist das Thymin durch U (Uracil) ersetzt. Die DNA ist eine Doppelhelix, die durch Wasserstoffbrückenverbindungen zwischen den Basen zusammengehalten wird. Dabei kann sich G nur mit C und A nur mit T paaren (Basenpaarung). In einem bestimmten Abschnitt der DNA kodiert einer der beiden Stränge für ein Protein, daher kann man den genetischen Code als eine Folge von Basen darstellen. Eine Nukleotidsequenz ist eine Folge von mehr als vier Basen, die die Primärstruktur eines DNA bzw. RNA Moleküls repräsentieren. Der genetische Code wird in Gruppen von drei Nukleotiden abgelesen, wobei jede Gruppe eine Aminosäure bestimmt. Jedes Trinukleotid wird Codon genannt. Die zur Nukleotidsequenz gehörige Aminosäuresequenz besteht dann aus den Aminosäuren der Codons. Die Aminosäuren bauen dann das Polypeptid auf [Lew91].
Die in den DNA- bzw. RNA-Genomen enthaltene genetische Information kann durch den Austausch einzelner Nukleotide innerhalb der Nukleinsäuresequenz verändert werden (Mutationen). Die Veränderungen nur eines einzelnen Basenpaares wird Punktmutation genannt. Man unterscheidet zwischen Transitionen und Transversionen. Bei der Transition wird ein CG-Basenpaar in ein AT-Basenpaar umgewandelt oder umgekehrt. Bei der Transversion wird ein Purin durch ein Pyrimidin oder umgekehrt ausgetauscht. Ein AT-Basenpaar wird dann z.B. zu einem TA- oder CG-Basenpaar [Lew91]. Mutationen können durch molekularbiologische Methoden ge-

zielt vorgenommen werden, treten aber auch unter natürlichen Bedingungen aufgrund der Fehlerrate der RNA- bzw. DNA-Polymerasen auf. Bei RNA-Viren treten die durch fehlerhafte Synthese bedingten Mutationen besonders häufig auf, weil RNA-Polymerasen keine Korrekturfunktion besitzen. Durch die Mutation können Eigenschaften des Viruspartikels verändert werden. In vielen Fällen wirkt sich die Mutation nachteilig für das Virus aus. Manche Mutanten können aber auch Vorteile haben, vor allem, wenn es um die Anpassung an veränderte Umweltbedingungen geht.
Fortgesetzte Mutationen können im Laufe der Evolution zu so großen Unterschieden führen, so dass ursprünglich verwandte Viren in unterschiedliche Genera oder sogar Familien eingeordnet werden können [Lew91, DE01].
Für die genetische Typisierung werden Nukleotid- oder Aminosäuresequenzen definierter Genomabschnitte verwendet. Hierzu wird die Virusnukleinsäure aus infiziertem Material isoliert, die Genomabschnitte mittels PCR (Polymerase-Kettenreaktion), bei RNA-Viren mittels RT-PCR, amplifiziert und in der Regel direkt sequenziert. Die PCR ist eine Methode, um einen definierten DNA-Abschnitt zu vervielfältigen, ohne einen lebenden Organismus zu verwenden. Zur Vervielfältigung der DNA wird ein Enzym, die DNA-Polymerase, verwendet, das in allen Lebewesen vorkommt und die DNA bei der Zellteilung verdoppelt. Mittels PCR ist auch ein Nachweis von RNA möglich. Hierzu ist ein zusätzlicher Reaktionschritt vor der eigentlichen PCR erforderlich, bei dem die RNA zuvor in eine cDNA (*copy* DNA) umgeschrieben wird. Hierzu benutzt man ein Enzym, die Reverse Transkriptase, das bei Retroviren vorkommt [Mue01].
Nach der Erfindung der PCR und der automatischen Sequenzierung konnten mit relativ wenig Aufwand große Mengen an Sequenzdaten erstellt werden. Somit bot sich die genetische Typisierung zur Ergänzung und Verbesserung der Taxonomie insbesondere von Mikroorganismen an [BS01].
Die graphische Darstellung der genetischen Verwandtschaft von Organismen erfolgt in Form von phylogenetischen Bäumen (*phylogenetic trees*). Diese sind Graphen bestehend aus Knoten (*nodes*) und Ästen (*branches*). Jeder Knoten hat in einem phylogenetischen Baum genau einen Vorgänger. Als Gesamtlänge bezeichnet man die Summe aller Astlängen. Die Knoten repräsentieren die taxonomischen Einheiten (*taxonomic units*). Diese können für Arten, Populationen, Individuen oder Gene stehen. Die Äste definieren die Verwandtschaft zwischen den Einheiten. Die Länge der Äste ist proportional zur Anzahl der Sequenz-Unterschiede. Bei den Knoten unterscheidet man zwischen äußeren Knoten, den *operational taxonomic units* (OTUs) und inneren Knoten. Unter dem *Split* eines Knotens versteht man alle nachfolgenden OTUs.
Bei den Bäumen unterscheidet man zwischen Bäumen ohne Wurzel (*unrooted*) und Bäumen mit Wurzel (*rooted trees*). In einem *rooted tree* existiert eine Wurzel (*root*), ein Knoten, von dem ein eindeutiger Weg zu allen anderen Knoten führt. Die Richtung des Weges entspricht der Evolution und die Wurzel ist der gemeinsame Vorfahr aller OTUs. Ein *unrooted tree* stellt lediglich die Verwandtschaft der einzelnen OTUs dar, sagt aber nichts über Evolution oder einen gemeinsamen Vorfahren aller OTUs

aus. Ist die Wurzel bekannt, so läßt sich ein *unrooted tree* ohne weitere Berechnungen in einen *rooted tree* umwandeln. Verwandtschaftsverhältnisse und Abstand einzelner OTUs ändern sich hierdurch nicht.
Die Anzahl der möglichen *rooted* (N_R) und *unrooted* (N_U) *trees* für n OTUs berechnet sich mit den folgenden Formeln:

$$N_R := \frac{(2n-3)!}{2^{n-2}(n-2)!} \quad n \geq 2 \tag{2.1}$$

$$N_U := \frac{(2n-5)!}{2^{n-3}(n-3)!} \quad n \geq 3. \tag{2.2}$$

Man beachte, dass die Anzahl der möglichen *rooted trees* für n OTUs gleich der Anzahl der möglich *unrooted trees* für $n+1$ OTUs ist. N_R und N_U wachsen sehr schnell, für zehn OTUs gibt es über 2 Millionen mögliche *unrooted trees* und fast 35 Millionen mögliche *rooted trees*. Da nur einer die korrekte Evolution wiedergibt, ist die Auswahl des richtigen Baums für große n schwierig. Die Mehrheit der phylogenetischen Methoden stellt den *unrooted tree* da [Li97].
Häufig ist aber die Darstellung als *rooted tree* gefordert. Zum Finden der Wurzel, die hierfür nötig ist, gibt es zwei Methoden: das *outgroup* Prinzip und den Gebrauch der molekularen Uhr. Beim *outgroup* Prinzip, wird ein OTU als *outgroup* gewählt von dem bekannt ist, dass keine enge Verwandtschaft zu allen anderen OTUs besteht und auch sicher gestellt ist, dass alle anderen OTUs untereinander enger verwandt sind als mit der *outgroup*. Die Wurzel ist dann der innere Knoten, der die *outgroup* mit allen anderen OTUs verbindet. Jede andere Wahl einer Wurzel würde dazu führen, dass ein Teil der OTUs enger mit der *outgroup* als mit den übrigen OTUs verwandt wäre. Dies soll aufgrund der Wahl der *outgroup* ausgeschlossen werden.
Die zweite Möglichkeit, die Wurzel zu bestimmen, basiert auf der molekularen Uhr [Fel04].
Die molekulare Uhr, basierend auf der *molecular clock hypothesis* (MCH), wird in der Genetik von Forschern benutzt, um den Zeitpunkt zu schätzen, an dem sich zwei Arten getrennt haben. Die seit der Trennung vergangene Zeit wird aus kleinen Änderungen in deren Protein- oder DNA-Sequenzen berechnet. Sie ist nicht vergleichbar mit einer mechanischen Uhr, sondern ist eine stochastische Uhr. Die MCH setzt voraus, dass die Evolutionsrate für jedes gegebene Makromolekül (Protein oder DNA-Sequenz) während der gesamten Zeit und in allen evolutionären *lineages* nahezu konstant ist [Li97].
Der Begriff molekulare Uhr wurde als erstes von Emile Zuckerkandl und Linus Pauling verwendet [ZP62]. Sie beobachteten, dass die Anzahl der Aminosäureaustausche bei Hämoglobin linear zur Zeit steigt. Sie verallgemeinerten diese Beobachtungen und vermuteten, dass die Rate von evolutionären Veränderungen für jedes Protein konstant sei.
Diese Hypothese ist, außer für die Datierung evolutionärer Ereignisse, für die phylogenetische Rekonstruktion von Interesse, weil diese mit konstanten Raten viel einfacher zu berechnen ist. Weiterhin bietet der Variationsgrad der Raten zwischen den *lineages* einen guten Einblick in die Mechanismen der molekularen Evolution. Das

Konzept der molekularen Uhr führt immer wieder zu unterschiedlichen Meinungen. Einerseits wurde von der Existenz einer universellen Uhr für synonyme Substitutionen ausgegangen, die für alle Organismen verwendbar ist [OW87]. Andererseits existieren nicht einmal die konstanten Raten [God76, God81].
Viele Mutationen ändern zwar die DNA-Sequenzen, wirken sich aber nicht auf den Phänotyp aus (Neutrale Theorie) [Kim68]. Diese evolutionär „neutralen“ Unterschiede können zur Zeitmessung benutzt werden. Eine Methode zur Kalibrierung ist die Verwendung von Referenzarten, bei denen der Zeitpunkt ihrer Trennung durch fossile Funde bekannt ist [Li97].
Es wurden fünf Faktoren aufgelistet, die die Ganggeschwindigkeit der molekularen Uhr beeinflussen [Aya99].

- Generationsdauer
- Populationsgröße
- Artspezifische Unterschiede
- Funktion eines Proteins
- Änderungen der "natürlichen Selektion"

Die Interpretation der molekularen Uhr ist bis heute nicht gelöst. Es treten unterschiedliche Taktfrequenzen für verschiedene Spezies, Gene, Genome in der gleichen Zelle, Regionen im Genom und unterschiedliche Positionen in einem Molekül auf. Abschließend kann man sagen, dass die Annahme einer konstanten Uhr, bis auf wenige Ausnahmen, nicht zutrifft. Eine Ausnahme wäre z.B. eine konservierte Region, innerhalb einer abgeschlossenen Familie von Proteinen eng verwandter Spezies. Es gibt keine universelle Uhr.
Für *Pestiviren* sind die Voraussetzungen der molekularen Uhr nicht nachweisbar. Daher wird diese Methode bei *Pestiviren* nicht angewendet.
Zum besseren Verständnis werden die hier verwendeten Begriffe *Lineage*, Mutationsrate und Evolutionsrate kurz erklärt:

Lineage (auch als *clade* bezeichent): *clade* kommt vom Griechischen Wort *klados* für Ast oder Zweig und ist eine fortlaufende Abstammungslinie (Vorfahr → Nachfolger) von Populationen, Zellen oder Genen. Sie besteht daher aus Spezies, Taxons oder Individuen, die von einem gemeinsamen Vorfahren abstammen. Sie bildet eine Gruppe von Organismen, die sowohl den gemeinsamen Vorfahren aller Gruppenmitglieder, als auch alle Nachfolger dieses gemeinsamen Vorfahren enthält. Eine *Lineage* wird häufig durch eine Teilmenge (Untergruppe) eines phylogenetischen Baums dargestellt. Das Konzept der evolutionären *Lineages* stammt aus der Kladistik. *Lineages* werden oft mit Techniken der molekularen Systematik ermittelt.

Mutationsrate: Eine Mutation ist eine Veränderung im Erbgut eines Organismus durch Veränderung der Abfolge der Nukleotidsequenz. Mutationen, die keine Folgen für den Organismus haben, werden als neutrale Mutationen bezeichnet. Die Mutationsrate beschreibt die Häufigkeit, mit der sich ein Gen verändert. Die Mutationsrate ist definiert durch die Anzahl der Mutationen pro Gen und pro Zeiteinheit (z.B. Zell-Generation)

Evolutionsrate: Ist die Rate der Divergenz zwischen den taxonomischen Gruppen. Messbar ist sie als Anzahl der Aminosäure-Substitutionen pro Millionen Jahre.

2.2.1 Genetische Typisierung von CSFV

Die phylogenetischen Bäume sind ein Kriterium für die Unterteilung von Viren in verschiedene Gruppen. Diese Gruppen können Spezies, Genotypen oder Subgruppen seien. Ein universales Kriterium für die Unterteilung in Gruppen gibt es aber nicht. Neben den phylogenetischen Bäumen gibt es noch weitere, die die Einteilung in Gruppen unterstützen können. Mögliche Faktoren sind z.B. *Bootstrap*-Werte und epidemiologische Daten. Auch eine Einteilung anhand von Sequenzunterschieden, wie z.B. beim Hepatitis B Virus (mehr als 8% Unterschied im gesamten Genom entspricht einer neuen Gruppe) [AN02] ist möglich.
Zur genetischen Typisierung von *Pestiviren* werden Fragmente der Sequenz verwendet, weil diese mit ca. 12,5 kb zu lang ist. Für die CSFV werden Fragmente der für die Proteine E2 und NS5B kodierenden Gene und der 5' NTR verwendet. Für die BVDV und die BDV wird anstelle des NS5B ein Fragment des Npro Proteins verwendet. Eine genetische Typisierung mit 5'NTR zeigt aufgrund der hohen Konservierung eine geringere Differenzierung als mit den anderen Genfragmenten. Für CSFV hat sich deshalb weltweit die Typisierung mit dem E2-Fragment durchgesetzt. Das NS5B-Fragment wird meistens zur Bestätigung des Ergebnisses verwendet [WF01].
Für CSFV die Einteilung in Genotypen und Subgruppen standardisiert. Es gibt drei Genotypen mit jeweils drei bis vier Subgruppen. Ein phylogenetischer Baum mit Isolaten aller Genotypen bzw. Subgruppen wurde in [PM00] berechnet. Der Genotyp 1 beinhaltet überwiegend die alten europäischen und die meisten amerikanischen Isolate, ebenso sind die Impfstämme vom Genotyp 1. Die Isolate mit Genotyp 2 dagegen sind weltweit zu finden. Genotyp 3 beinhaltet Isolate, die aus verschiedenen Teilen Asiens stammen [JG03] (Abb. 2.2).

Dieser Zusammenhang zwischen den Genotypen, den Subgruppen und den epidemiologischen Daten läßt sich bei der genetischen Typisierung für die geographische Zuordnung neuer Isolate nutzen. Unter anderem ist bei Sekundärausbrüchen auch eine Aussage über den Infektionsweg möglich. Einige Beispiele für die Nutzung sind:

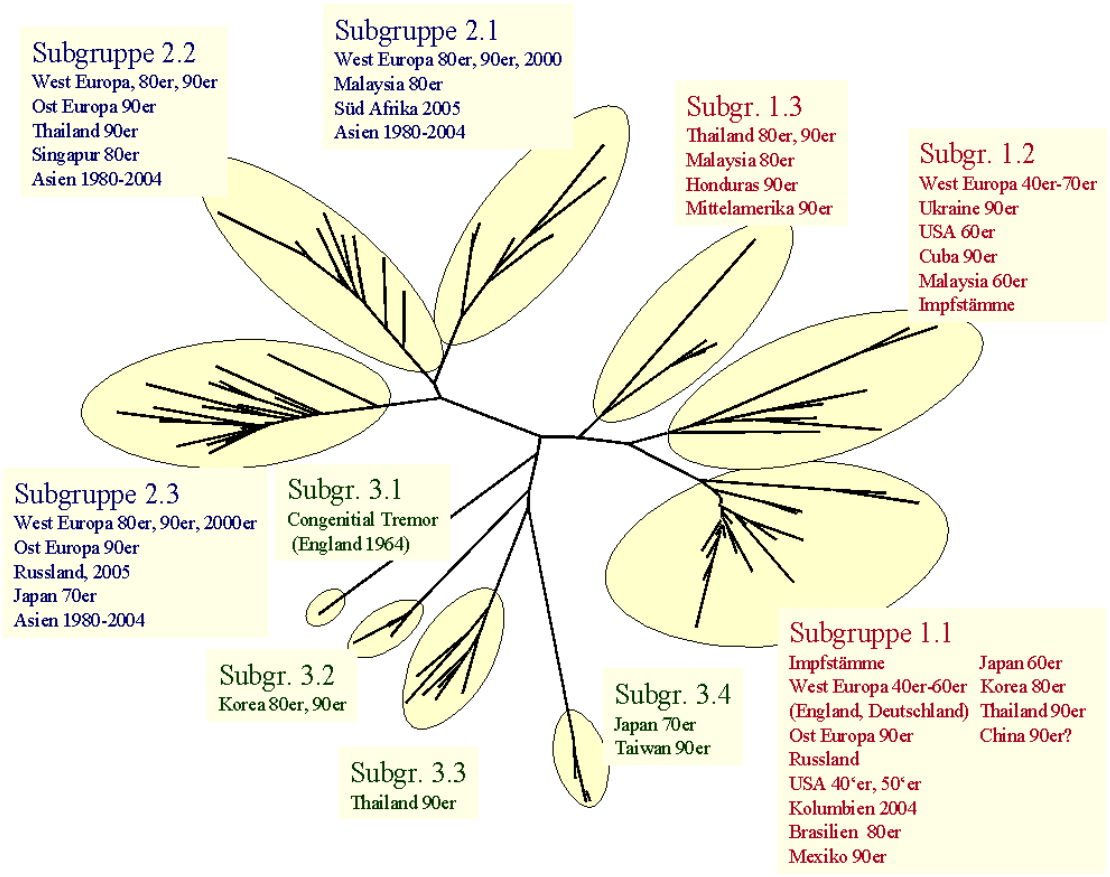


Abbildung 2.2: Übersicht der CSFV-Subgruppen und deren geographisches Vorkommen (modifiziert nach [PM00]).

- Die Genotypisierung von neun Feld-Isolaten von Haus-und Wildschweinen aus den Jahren 1997-2005, die aus mehreren Ausbrüchen von Klassischer Schweinepest in Kroatien stammten, ergab im Vergleich mit Isolaten aus anderen europäischen Staaten, eine Einordnung in die Subgruppen 2.1 und 2.3. Die Isolate aus Subgruppe 2.1 stammten aus einem sporadischen Ausbruch im Juni 1997 und sind genetisch eng verwandt. Ebenso sind sie den europäischen Isolaten aus dem gleichem Jahr sehr ähnlich. Im Gegensatz dazu ist das Isolat von Oktober 1997 in Subgruppe 2.3 und nah verwandt mit Isolaten von Ausbrüchen in den 90er Jahren aus West- und Zentraleuropa. Dieses Resultat zeigt die möglichen Quellen für Ausbrüche innerhalb eines Jahres in Kroatien [JG03].

- In der Republik Laos wurden zwischen 1997 und 1999 21 Isolate von Ausbrüchen der Klassischen Schweinepest gesammelt, sequenziert und mit anderen CSFV Isolaten verglichen. Die genetische Typisierung ergab eine Einordnung in die Subgruppen 2.1 und 2.2. Es ist eine deutliche geographische Grenze

erkennbar: Alle Isolate aus der Subgruppe 2.1 stammten aus der nördlichen Hälfte des Landes, die Isolate aus Subgruppe 2.2 dagegen aus den südlichen Regionen [BK04].

- 16 Isolate, die aus dem Ausbruch der Klassischen Schweinepest 1997/1998 in den Niederlanden stammten wurden mit anderen europäischen Isolaten und Isolaten von früheren Ausbrüchen in den Niederlanden verglichen. Die Analyse der Sequenzen ergab einen direkten Zusammenhang mit einem Isolat aus Paderborn, das 1996 auftrat [WG99].

- In der Schweiz gab es 1993 fünf Ausbrüche von Klassischer Schweinepest. Bei vier der Ausbrüche handelte es sich um Primärausbrüche ohne epidemiologischen Zusammenhang. Lediglich bei einem der Ausbrüche handelte es sich um einen Sekundärausbruch, dessen vermutliche Ursache Tierhandel war. Bei den anderen Ausbrüchen liegt die Vermutung nahe, dass die Ursache die Verfütterung von nicht ausreichend erhitzten Speiseabfällen war. Eine genetische Typisierung ergab eine nahe Verwandtschaft aller Schweizer Isolate mit Isolaten, die seit 1990 in Europa kursierten [HB98].

Um die genetische Typisierung von CSFV zu vereinfachen, wurde eine Datenbank, die die Sequenzen und epidemiologischen Daten aller bekannten Isolate enthält, aufgebaut [GZ00].

2.2.2 Genetische Typisierung von BVDV und BDV

Für BVDV wird zur Zeit die Einteilung in BVDV-1 a bis k [VP01] und BVDV-2 a bis b [VD04] verwendet. Laut ICTV (International Committee on Taxonomy of Viruses) sind BVDV-1 und BVDV-2 eigene Spezies. Diese Spezies lassen sich in Genotypen unterteilen. Teilweise wird BVDV allerdings als eigene Spezies behandelt und in die Genotypen BVDV-1 und BVDV-2 unterteilt [VD05]. Ob zusätzlich ein geographischer Zusammenhang zwischen den Genotypen und Subgruppen besteht, ist noch nicht geklärt. Einige Befunde deuten darauf hin, dass es für BVDV keinen Zusammenhang zwischen der Gruppenverteilung und der geographischen Herkunft der Isolate gibt [VD05]. Andere Ergebnisse sprechen für einen geographischen Zusammenhang. In der Schweiz wurden z.B. über 150 BVDV Isolate untersucht. Alle gehörten zum Genotyp BVDV-1 mit den Subgruppen e, h, k und b. Die Subgruppe k schien nur in der Schweiz vorzukommen [SM05]. Bei einer Untersuchung von 91 BVDV-Isolaten aus Australien wurden nur die Subgruppen BVDV-1 a und c gefunden [MM05]. In Niedersachsen ergab die genetische Typisierung von 61 Isolaten, dass überwiegend Viren der Subgruppen BVDV-1 b und d vorkommen. Die anderen Subgruppen und auch BVDV-2 wurden nur vereinzelt nachgewiesen [TF01].
Für BDV sind bisher neun Gruppen beschrieben. Allerdings gibt es weder eine einheitliche Bezeichnung der Gruppen, noch Aussagen bezüglich einer geographischen Verteilung [BA03, VN97, HG03, AF04, MG05, VA06].

Einige, von tunesischen Schafen stammende *Pestivirus* Isolate, wurden BDV zugeordnet [TL05]. Weitere phylogenetische Untersuchungen mit anderen *Pestiviren* ergaben, dass es sich bei diesen Isolaten nicht um BDV handelt [MG05]. Sie lassen sich auch nicht eindeutig einer der *Pestivirus* Spezies zuordnen. Es handelt sich nach den bisherigen Erkenntnissen um eine neue Spezies.
Ebenfalls zu den *Pestiviren* gehören die Isolate Giraffe, Pronghorn und HoBi. Sie lassen sich aber keiner der bisher beschriebenen Spezies zuordnen. Eine Übersicht aller bekannten *Pestiviren* (Abb. 2.3) ist nicht abschließend, weil für die Berechnung nur Npro-Sequenzen verwendet wurden, aber nicht von allen *Pestivirus*-Isolaten Npro-Sequenzen existieren.

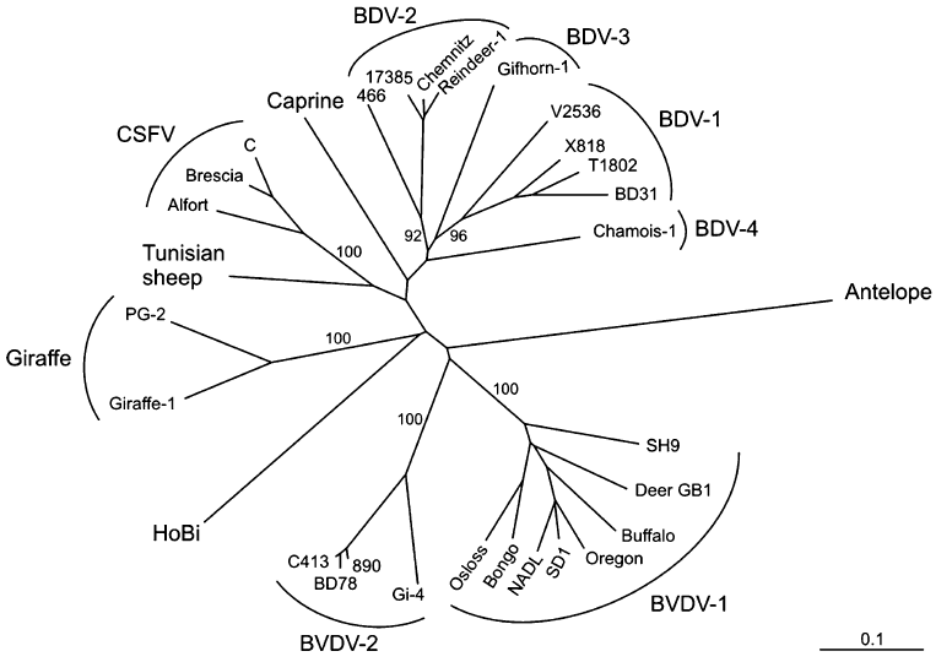


Abbildung 2.3: Phylogenetischer Baum mit den bisher bekannten *Pestivirus* Spezies und den noch nicht eingeordneten Isolaten. Berechnet wurde er mit den Npro-Sequenzen [VN06].

2.3 Methoden zur Berechnung phylogenetischer Bäume

Es gibt eine Vielzahl von Methoden zur Berechnung phylogenetischer Bäume. Die Bäume werden entweder direkt aus den Sequenzen berechnet, wie z.B. beim *Par-*

simony oder Maximum-Likelihood Algorithmus. Oder alternativ wird aus den Sequenzen zunächst eine Distanzmatrix berechnet, aus der dann die Bäume berechnet werden. Auf dieser Variante basieren z.B. das UPGMA (*Unweighted Pairwise Grouping Method using Arithmetic means*) oder der Neighbor-Joining Algorithmus. Der Grund für die Entwicklung verschiedener Methoden sind die unterschiedlichen Voraussetzungen der Ausgangsprobleme. Jede Methode benötigt andere Voraussetzungen, damit sie korrekte Ergebnisse liefert [Li97]. Nicht jede Methode ist somit für jedes Problem geeignet. Identisch bei allen Methoden ist, dass sie als Grundlage ein *Alignment* der Sequenzen benötigen.
Um einen phylogenetischen Baum zu berechnen, benötigt man zunächst eine Methode, um die einzelnen Sequenzen vergleichen zu können. Dazu ist die Berechnung eines *Alignments* notwendig. Für ein *Alignment* müssen die Sequenzen immer aus dem gleichem Genomfragment sein. Um nur die Identität zweier Sequenzen zu überprüfen, genügt ein Direktvergleich. Er beinhaltet aber keine Möglichkeit, auf Lücken in der Sequenz einzugehen. Benötigt man diese, so wird ein *pairwise Alignment* berechnet.

2.3.1 Direktvergleich

Der Direktvergleich (DV) ist ein einfacher Basenvergleich, d.h. die i-te Base von Sequenz a wird mit der i-ten Base von Sequenz b verglichen. Jede Übereinstimmung wird gezählt und am Ende prozentual ausgewertet. Wenn die Sequenzen allerdings unterschiedlich lang sind oder an verschiedenen Positionen im Genom beginnen, bekommt man bei nahezu identischen Sequenzen nur geringe Übereinstimmungen. Um dies auszugleichen, wurde zusätzlich ein Basenshift eingeführt.
Hierbei wird zunächst die Sequenz a um $n = 1$ Base verschoben und dann erst der Basenvergleich durchgeführt. Bei einer Verschiebung von Sequenz a um 1 Base bedeutet dies, dass die $(i+1)$-te Base von Sequenz b mit der i-ten Base von Sequenz a verglichen wird. Dann wird die Sequenz b um $n = 1$ Base verschoben und wieder der Basenvergleich durchgeführt. Dies wird dann genauso für alle n zwischen 2 und 100 durchgeführt. Die Verschiebung und die zugehörige Übereinstimmung, die die größte Übereinstimmung der Sequenzen ergibt, werden dann als Ergebnis ausgegeben.

2.3.2 Paarweises Alignment

Die Berechnung eines *Alignment* von zwei Sequenzen (*pairwise alignment*) dient als Grundlage zur Berechnung von Distanzen (z.B. Jukes-Cantor-Distanz) oder als Eingabe für den Maximum-Likelihood Algorithmus.
Im Folgenden sei Σ ein Alphabet, $-$ ein weiteres Zeichen, d.h. $- \notin \Sigma$. Mit $\overline{\Sigma} := \Sigma \cup \{-\}$ wird das um $-$ erweiterte Alphabet bezeichnet. Eine Sequenz a besteht aus einer Folge von Elementen aus Σ, d.h. $a := a_1a_2a_3...a_{n-2}a_{n-1}a_n$ mit $a_i \in \Sigma$ und $n \in \mathbb{N}$. Die Länge von a wird mit $|a|$ bezeichnet.
Zunächst werden die verschiedenen Möglichkeiten der Unterschiede und das *Alignment* definiert:

Definition 2.1. *Eine Edit-Operation ist ein Paar*

$$(x, y) \in \overline{\Sigma} \times \overline{\Sigma} \setminus \{-, -\}$$

Eine Edit-Operation (x,y) heißt:

- *Match, wenn* $x = y \in \Sigma$*;*
- *Substitution (oder Mismatch), wenn* $x \neq y \in \Sigma$*;*
- *Insertion, wenn* $x = -, y \in \Sigma$*;*
- *Deletion, wenn* $x \in \Sigma, y = -$*.*

Es seien $a, b \in \overline{\Sigma}$ *und* (x, y) *eine Edit-Operation mit* $x \neq y$*, so schreibt man*

$$a \rightarrow_{(x,y)} b,$$

wenn man b aus a erhält, indem man x an einer Position durch y ersetzt. Ist $S = s_1...s_r$*,* $r \in \mathbb{N}$ *eine Sequenz aus Edit-Operationen, so schreibt man*

$$a \Rightarrow_S b,$$

wenn $a = a^{(0)} \rightarrow_{s_1} a^{(1)} \rightarrow_{s_2} ... \rightarrow_{s_r} a^{(r)} = b$*, für Wörter* $a^{(0)}...a^{(r)}$*.*
Als Indel-Operation bezeichnet man eine Edit-Operation, die entweder eine Insertion oder eine Deletion ist.

In der Evolution kommen manche Mutationsarten häufiger vor als andere. Dies lässt sich in der Definition von *Edit*-Distanzen durch eine Gewichtung der *Edit*-Operationen simulieren. Diese geschieht durch eine Kostenfunktion $W(x, y)$, die den einzelnen Edit-Operationen Kosten zuweist.

Definition 2.2. *Gegeben seien eine Kostenfunktion* $w : \overline{\Sigma} \times \overline{\Sigma} \rightarrow \mathbb{R}$ *und zwei Wörter* $a, b \in \Sigma$*.* $S = s_1...s_r$*,* $r \in \mathbb{N}$ *sei eine Sequenz aus Edit-Operationen mit* $w(S) := \sum_{i=1}^{r} w(s_i)$*. Die Edit-Distanz von a und b ist dann definiert als:*

$$d_w(a, b) := min\{w(S) | a \Rightarrow_S b\}.$$

Definition 2.3. *d ist eine Metrik, wenn gilt:*

(1.) $d(x, y) = 0$*, wenn* $x = y$*;*

(2.) $d(x, y) = d(y, x)$ *(Symmetrie);*

(3.) $d(x, z) \leq d(x, y) + d(y, z)$ *(Dreiecksungleichung).*

Es gilt: Ist die Kostenfunktion w eine Metrik, so auch die *Edit*-Distanz d_w. Man kann außerdem zeigen, dass die *Edit*-Distanz mit der folgenden *Alignment*-Distanz übereinstimmt, falls die Kostenfunktion w eine Metrik ist.

Definition 2.4. *Ein Alignment ist ein Paar* $(\overline{a}, \overline{b}) \in \overline{\Sigma} \times \overline{\Sigma}$ *mit* $|\overline{a}| = |\overline{b}|$

$$\overline{a_i} = \overline{b_i} \Rightarrow \overline{a_i} \neq - \neq \overline{b_i} \quad \forall i \in [1, |\overline{a}|].$$

$(\overline{a}, \overline{b})$ *ist ein Alignment für* $a, b \in \Sigma$*, wenn gilt:*

$$\overline{a}|_\Sigma = a \text{ und } \overline{b}|_\Sigma = b.$$

Es sei $\overline{w} : \overline{\Sigma} \times \overline{\Sigma} \to \mathbb{R}_+$ *eine (Kosten-)Funktion, dann sind die Kosten eines Alignments* $(\overline{a}, \overline{b})$ *für* (a, b) *definiert als*

$$\overline{w}(\overline{a}, \overline{b}) := \sum_{i=1}^{|a|} \overline{w}(\overline{a_i}, \overline{b_i}).$$

Die Alignment-Distanz von $a, b \in \Sigma$ *ist definiert als*

$$\overline{d}_{\overline{w}}(a, b) := \min\{\overline{w}(\overline{a}, \overline{b}) | (\overline{a}, \overline{b}) \text{ ist Alignment für } a, b\}.$$

Beispiel. Für die Sequenzen $a := AGGCTG$ und
Sequenz $b := ACCGGTA$, ergibt sich als ein mögliches *Alignment*

$$\begin{array}{lccccccccc} a := & A & - & - & G & G & C & T & G \\ b := & A & C & C & G & G & - & T & A \\ & & I & I & & & D & & S \end{array}$$

Wählt man als Kostenfunktion $\overline{w} : \overline{\Sigma} \times \overline{\Sigma} \to \mathbb{R}_+$,

$$\overline{w}(\overline{a_i}, \overline{b_i}) := \begin{cases} 0 & \text{falls } \overline{a_i} = \overline{b_i} \\ 3 & \text{falls } \overline{a_i} \neq \overline{b_i} \text{ und } \overline{a_i}, \overline{b_i} \neq - \\ 2 & \text{falls } \overline{a_i} = - \text{ oder } \overline{b_i} = - \end{cases} \tag{2.3}$$

so hat dieses *Alignment* Kosten von
$\overline{w}(\overline{a}, \overline{b}) = 0 + 2 + 2 + 0 + 0 + 2 + 0 + 3 = 9.$

Mit dem Distanzmaß läßt sich der Unterschied zweier Sequenzen bewerten. Um aber auch die Ähnlichkeit bewerten zu können, wird zusätzlich noch ein Ähnlichkeitsmaß eingeführt.

Definition 2.5. *Eine Kostenfunktion* $w' : \overline{\Sigma} \times \overline{\Sigma} \to \mathbb{R}_+$ *für ein Ähnlichkeitsmaß heißt sinnvoll, wenn folgende Bedingungen erfüllt sind:*

(S1) $\forall x \in \Sigma : w'(x, x) \geq 0$*;*

(S2) $\forall x \neq y \in \overline{\Sigma} : w'(x,y) \leq 0$*;*

(S3) $\forall x, y \in \overline{\Sigma} : w'(x,y) = w'(y,x)$.

Definition 2.6. *Sei* $w' : \overline{\Sigma} \times \overline{\Sigma} \to \mathbb{R}_+$ *eine sinnvolle (Kosten-)Funktion und sei* $(\overline{a}, \overline{b})$ *ein Alignment für* $a, b \in \Sigma$*. Dann ist die Ähnlichkeit* $(\overline{a}, \overline{b})$ *definiert als*

$$w'(\overline{a}, \overline{b}) := \sum_{i=1}^{|a|} w'(\overline{a_i}, \overline{b_i}).$$

Die Ähnlichkeit von $a, b \in \Sigma$ *ist definiert als*

$$s(a,b) := \max\{w'(\overline{a}, \overline{b}) | (\overline{a}, \overline{b}) \text{ ist Alignment für } a, b\}.$$

[CB00]

Beispiel. Wählt man als Kostenfunktion $w' : \overline{\Sigma} \times \overline{\Sigma} \to \mathbb{R}$,

$$w'(\overline{a_i}, \overline{b_i}) := \begin{cases} 3 & \text{falls } \overline{a_i} = \overline{b_i} \\ -3 & \text{falls } \overline{a_i} \neq \overline{b_i} \text{ und } \overline{a_i}, \overline{b_i} \neq - \\ -2 & \text{falls } \overline{a_i} = - \text{ oder } \overline{b_i} = - \end{cases} \tag{2.4}$$

so ergeben sich für das *Alignment* aus dem vorherigen Beispiel Kosten von $w'(\overline{a}, \overline{b}) = 3$.

Um die *Alignment*-Distanz $\overline{d}_{\overline{w}}(a,b)$ oder die Ähnlichkeit $s(a,b)$ zu berechnen, muss man für alle möglichen *Alignments* die Kosten berechnen und daraus das Minimum bzw. Maximum bilden. Die folgenden beiden Algorithmen berechnen aus zwei Sequenzen das *Alignment* mit der kleinsten Distanz bzw. mit der größten Ähnlichkeit.

Needleman-Wunsch-Algorithmus

Beim Needleman-Wunsch-Algorithmus (NW) wird ein globales *Alignment* zweier Sequenzen a und b ($a, b \in \Sigma$) berechnet. Es sei $|a| = n$, $|b| = m$ und $\overline{w} : \overline{\Sigma} \times \overline{\Sigma} \to \mathbb{R}$ sei eine metrische Kostenfunktion. Global bedeutet, dass die vollständigen Sequenzen verglichen werden und auch Lücken am Anfang oder Ende einer Sequenz in die Berechnung der Distanz mit eingehen.
Um das *Alignment* zu berechnen, wird zunächst eine Matrix $D_{i,j}$ mit $0 \leq i \leq |a|$ und $0 \leq i \leq |b|$ mit der folgenden Formel rekursiv berechnet. $D(i,j)$ gibt hierbei jeweils die Distanz des optimalen *Alignments* der Teilsequenzen $a_1 ... a_i$ und $b_1 ... b_j$ an.
Die Anfangswerte seien:
$D(0,0) := 0$, $D(i,0) := \sum_{k=1}^{i} \overline{w}(a_k, -)$ und $D(0,j) := \sum_{k=1}^{j} \overline{w}(-, b_k)$.

$$D(i,j) := \min \begin{cases} D(i-1, j-1) & + & \overline{w}(a_i, b_j) \\ D(i-1, j) & + & \overline{w}(a_i, -) \\ D(i, j-1) & + & \overline{w}(-, b_j) \end{cases} \quad \forall i, j > 0. \tag{2.5}$$

Das optimale *Alignment* ergibt sich nun durch Rückverfolgen der Sieger der Minimierung. Beginnend an der Position mit (n, m) (unten rechts), solange bis die Position $(0, 0)$ (oben links) erreicht ist. Die Distanz der Sequenzen $\overline{d}_{\overline{w}}(a, b)$ läßt sich ebenfalls aus der Matrix ablesen, $\overline{d}_{\overline{w}}(a, b) := D(n, m)$. Für zwei Sequenzen a und b mit $|a| = n$ und $|b| = m$ läßt sich ein globales *Alignment* mit Zeitbedarf O(nm) und Platzbedarf O(nm) berechnen [CB00].

Beispiel. Für die Sequenzen $a := AGGCTG$ und $b := ACCGGTA$ soll mit dem Needleman-Wunsch Algorithmus das *Alignment* mit der kleinsten Distanz berechnet werden.
Die Kostenfunktion $\overline{w}$ sei identisch mit der aus Gleichung (2.3)
Daraus ergibt sich die folgende Matrix D

			b													
			$\rightarrow$													
			A		C		C		G		G		T		A	
		0		2		4		6		8		10		12		14
a	$\downarrow$	A		$\nwarrow$												
		2		**0**	$\leftarrow$	**2**	$\leftarrow$	**4**		6		8		10		12
		G						$\nwarrow$								
		4		2		3		5		**4**		6		8		10
		G								$\nwarrow$						
		6		4		5		6		5		**4**		6		8
		C										$\uparrow$				
		8		6		4		5		7		**6**		7		9
		T											$\nwarrow$			
		10		8		6		7		8		8		**6**		8
		G													$\nwarrow$	
		12		10		8		9		7		8		8		**9**

Die Elemente von D ergeben sich aus Gleichung 2.5. Zum Beispiel gilt

$$\begin{aligned} D(1,1) &= \min\{D(0,0) + \overline{w}(a_1, b_1), D(0,1) + \overline{w}(a_1, -), D(1,0) + \overline{w}(-, b_1)\} \\ &= \min\{\mathbf{0} + \mathbf{0}, 2 + 2, 2 + 2\} \\ &= 0, \end{aligned}$$

da $a_1 = b_1 = A$, also $\overline{w}(a_1, b_1) = 0$ und $\overline{w}(a_1, -) = \overline{w}(-, b_1) = 2$.

$$\begin{aligned} D(1,2) &= \min\{D(0,1) + \overline{w}(a_1, b_2), D(0,2) + \overline{w}(a_1, -), D(1,1) + \overline{w}(-, b_2)\} \\ &= \min\{2 + 3, 4 + 2, \mathbf{0} + \mathbf{2}\} \\ &= 2, \end{aligned}$$

da $a_1 \neq b_2$, also $\overline{w}(a_1, b_2) = 3$ und $\overline{w}(a_1, -) = \overline{w}(-, b_2) = 2$.

$$\begin{aligned} D(2,1) &= \min\{D(1,0) + \overline{w}(a_2, b_1), D(1,1) + \overline{w}(a_2, -), D(2,0) + \overline{w}(-, b_1)\} \\ &= \min\{2 + 3, \mathbf{0} + \mathbf{2}, 4 + 2\} \\ &= 2, \end{aligned}$$

da $a_2 \neq b_1$, also $\overline{w}(a_2, b_1) = 3$ und $\overline{w}(a_2, -) = \overline{w}(-, b_1) = 2$.

$$\begin{aligned} D(2,2) &= \min\{D(1,1)+\overline{w}(a_2,b_2), D(1,2)+\overline{w}(a_2,-), D(2,1)+\overline{w}(-,b_2)\} \\ &= \min\{\mathbf{0}+\mathbf{3}, 2+2, 2+2\} \\ &= 3, \end{aligned}$$

da $a_2 \neq b_2$, also $\overline{w}(a_2,b_2) = 3$ und $\overline{w}(a_2,-) = \overline{w}(-,b_2) = 2$.

Und

$$\begin{aligned} D(6,7) &= \min\{D(5,6)+\overline{w}(a_6,b_7), D(5,7)+\overline{w}(a_6,-), D(6,6)+\overline{w}(-,b_7)\} \\ &= \min\{\mathbf{6}+\mathbf{3}, 8+2, 8+2\} \\ &= 3, \end{aligned}$$

da $a_6 \neq b_7$, also $\overline{w}(a_6,b_7) = 3$ und $\overline{w}(a_6,-) = \overline{w}(-,b_7) = 2$.
Die fett gedruckten Elemente zeigen den Weg der Rückverfolgung.
Als optimales *Alignment* ergibt sich dann

$$\begin{array}{lcccccccc} a := & A & - & - & G & G & C & T & G \\ b := & A & C & C & G & G & - & T & A \end{array}$$

mit einer *Alignment*-Distanz $\overline{d}_{\overline{w}}(a,b) = 9$.

Smith-Waterman-Algorithmus

Beim Smith-Waterman-Algorithmus (SW) wird ein lokales *Alignment* zweier Sequenzen a und b berechnet. Es sei $|a| = n$ und $|b| = m$. Lokal bedeutet, dass in den beiden Sequenzen zwei Teilwörter gesucht werden, die möglichst ähnlich zueinander sind. Lücken am Anfang oder Ende des *Alignments* gehen in die Berechnung der Distanz nicht mit ein. Dadurch entsteht aber das Problem, dass man immer ein *Alignment* mit der Distanz Null erzeugen kann:

$$\begin{array}{lcccccccccccccccccc} a := & - & - & - & - & - & - & - & - & - & T & G & G & C & A & T & C & G & A \\ b := & A & C & C & G & G & C & T & T & A & - & - & - & - & - & - & - & - & - \end{array}$$

Um dies zu verhindern, werden hier Ähnlichkeitsmaße verwendet. w' sei eine sinnvolle Kostenfunktion für ein Ähnlichkeitsmaß. Um das *Alignment* zu berechnen, wird wie beim NW eine Matrix H rekursiv berechnet. $H_{i,j}$ gibt hierbei den Wert des optimalen lokalen *Alignments* der Teilsequenzen $a_1...a_i$ und $b_1...b_j$ an.

$$H(k,0) = H(0,l) = 0 \text{ für } 0 \leq k \leq n \ \wedge \ 0 \leq l \leq m \tag{2.6}$$

.

$$H(i,j) := \max \begin{cases} H(i-1,j-1) + w'(a_i,b_j) \\ H(i-1,j) + w'(a_i,-) \\ H(i,j-1) + w'(-,b_j) \\ 0 \end{cases} \quad \text{für } 1 \leq i \leq n \ \wedge \ 1 \leq j \leq m \tag{2.7}$$

Durch den Wert Null in der Maximierung im Fall von $(i > 0) \wedge (j > 0)$, kann das lokale *Alignment* an jeder inneren Position i bzw. j von den beiden Sequenzen beginnen.

Ebenso kann es an jeder inneren Position i bzw. j enden. Um das optimale *Alignment* zu bestimmen, muss der maximale Wert $s(a,b) := \max_{i,j} H(i,j)$ bestimmt werden. Dies ist dann die Ähnlichkeit der beiden Sequenzen. Das *Alignment* ergibt sich durch Rückverfolgen der Sieger der Maximierung. Beginnend an der Position (i,j), wo H den Wert $s(a,b)$ annimmt, solange, bis zum ersten Mal der Wert Null auftaucht. Dies ist der Anfangspunkt des *Alignments*.
Für zwei Sequenzen a und b mit $|a| = n$ und $|b| = m$ läßt sich ein lokales *Alignment* mit Zeitbedarf O(nm) und Platzbedarf O(nm) berechnen [SW81].

Beispiel. Als Beispiel sei $a := ACGATT$ und $b := GACATCG$ zwei Sequenzen auf die der SW Algorithmus anwendet wird.
Die Kostenfunktion w' sei die aus Gleichung (2.4) Daraus ergibt sich die folgende Matrix H

		b →		G	A	C	A	T	C	G
			0	**0**	0	0	0	0	0	0
a	↓	A			↖					
			0	0	**3**	1	3	1	0	0
		C				↖				
			0	0	1	**6**	4	2	4	2
		G				↑				
			0	3	1	**4**	3	1	2	7
		A					↖			
			0	1	6	4	**7**	5	3	5
		T						↖		
			0	0	4	3	5	**10**	8	6
		T								
			0	0	2	1	3	8	7	5

Die Elemente von H ergeben sich aus Gleichung 2.7. Zum Beispiel gilt

$$\begin{aligned} H(1,1) &= \max\{H(0,0) + w'(a_1,b_1), H(0,1) + w'(a_1,-), H(1,0) + w'(-,b_1), 0\} \\ &= \max\{0-3, 0-2, 0-2, \mathbf{0}\} \\ &= 0, \end{aligned}$$

da $a_1 \neq b_1$, also $w'(a_1,b_1) = -3$ und $w'(a_1,-) = w'(-,b_1) = -2$.

$$\begin{aligned} H(1,2) &= \max\{H(0,1) + w'(a_1,b_2), H(0,2) + w'(a_1,-), H(1,1) + w'(-,b_2), 0\} \\ &= \max\{\mathbf{0}+\mathbf{3}, 0-2, 0-2, 0\} \\ &= 3, \end{aligned}$$

da $a_1 = b_2 = \mathrm{A}$, also $w'(a_1,b_2) = 3$ und $w'(a_1,-) = w'(-,b_2) = -2$.

$$\begin{aligned} H(2,1) &= \max\{H(1,0) + w'(a_2,b_1), H(1,1) + w'(a_2,-), H(2,0) + w'(-,b_1), 0\} \\ &= \max\{0-3, 0-2, 0-2, \mathbf{0}\} \\ &= 0, \end{aligned}$$

da $a_2 \neq b_1$, also also $w'(a_2,b_1) = -3$ und $w'(a_2,-) = w'(-,b_1) = -2$.
Und

$$\begin{aligned} H(2,2) &= \max\{H(1,1)+w'(a_2,b_2), H(1,2)+w'(a_2,-), H(2,1)+w'(-,b_2), 0\} \\ &= \max\{0-3, \mathbf{3}-\mathbf{2}, 0-2, 0\} \\ &= 1, \end{aligned}$$

da $a_2 \neq b_2$, also also also $w'(a_2,b_2) = -3$ und $w'(a_2,-) = w'(-,b_2) = -2$.
Die fett gedruckten Elemente zeigen den Weg der Rückverfolgung, beginnend bei dem maximalen Element 10.
Als optimales lokales *Alignment* ergibt sich dann

$$\begin{array}{lc|cccc|cc} s := & - & A & C & G & A & T & T & \\ t := & G & A & C & - & A & T & C & G \end{array}$$

mit einer Ähnlichkeit von $s(a,b) = 10$. Die verbliebenen Basen der Sequenz werden vor- bzw. angehängt. Dies ist hier mit den | deutlich gemacht.

Lückenstrafen

Alignments beinhalten zum Teil sehr lange Lücken. Bei den bisher vorgestellten Verfahren wurde für eine Lücke der Länge l, Kosten für l Insertionen oder Deletionen berechnet. Da diese Lücken aber häufig durch eine Mutation entstanden sind, ist dies nicht unbedingt sinnvoll. Als Lückenstrafe wird eine Funktion $g : \mathbb{N}_0 \to \mathbb{R}_+$ verwendet. Dabei ist $g(k)$ die Bestrafung für eine Lücke der Länge k. Für Distanzmaße ist g immer nicht negativ, für Ähnlichkeitsmaße immer nicht positiv. Es gilt weiter $g(0) = 0$ und $|g| : \mathbb{N}_0 \to \mathbb{R}_+ : k \mapsto |g(k)|$ ($|g|$ ist die Betragsfunktion von g) ist eine monoton wachsende Funktion. Außerdem sei g sublinear, d.h. $g(k_1+k_2) \leq g(k_1)+g(k_2) \quad \forall k_1, k_2 \in \mathbb{N}_0$.
Es gibt verschiedene Möglichkeiten, Lückenstrafen zu definieren. Hier wurden nur die affinen Lückenstrafen verwendet, weil sich diese mit angemessener Laufzeit implementieren lassen. Affine Lückenstrafen lassen sich durch die folgende Gleichung beschreiben:

$$g : \mathbb{N} \to \mathbb{R}_+ : k \mapsto \mu \cdot k + \nu \quad \mu, \nu \in \mathbb{R}_+ \tag{2.8}$$

$g(0) := 0$ wie oben gefordert. ν sind die Kosten für die Eröffnung einer Lücke (gap open) und μ die proportionalen Kosten für die Länge einer Lücke (gap extention).
Wie beim NW -Algorithmus wird die Distanz des *Alignments* und das *Alignment* selbst rekursiv mit Hilfe einer Matrix D berechnet. Diesmal benötigt man dafür allerdings noch drei weitere Matrizen E, F und G, die wie folgt definiert sind:

- $E(i,j)$:= Distanz des optimalen *Alignments* von $a_1...a_i$ mit $b_1...b_j$, das mit einer Insertion endet.
- $F(i,j)$:= Distanz des optimalen *Alignments* von $a_1...a_i$ mit $b_1...b_j$, das mit einer Deletion endet.
- $G(i,j)$:= Distanz des optimalen *Alignments* von $a_1...a_i$ mit $b_1...b_j$, das mit einer Substitution endet.

- $D(i,j) :=$ Distanz des optimalen *Alignments* von $a_1...a_i$ mit $b_1...b_j$.

Aufgrund der Dreiecksungleichung aus Definition 2.3 dürfen Insertion und Deletion nicht direkt aufeinander folgen. Damit ergeben sich folgende Rekursionsgleichungen:

$$E(i,j) := \min \begin{cases} G(i,j-1) + \mu + \nu & \text{davor war eine Substitution} \\ E(i,j-1) + \mu & \text{davor war eine Insertion} \end{cases} \tag{2.9}$$

$$F(i,j) := \min \begin{cases} G(i-1,j) + \mu + \nu & \text{davor war eine Substitution} \\ F(i-1,j) + \mu & \text{davor war eine Deletion} \end{cases} \tag{2.10}$$

$$G(i,j) := \min \begin{cases} G(i-1,j-1) + \overline{w}(a_i,b_j) & \text{davor war eine Substitution} \\ F(i-1,j-1) + \overline{w}(a_i,b_j) & \text{davor war eine Deletion} \\ E(i-1,j-1) + \overline{w}(a_i,b_j) & \text{davor war eine Insertion} \end{cases} \tag{2.11}$$

$$D(i,j) := \min\{E(i,j), F(i,j), G(i,j)\} \tag{2.12}$$

Die Anfangswerte seien für $i > 0$ und $j > 0$:

$$\begin{aligned} E(0,j) &:= j \cdot \mu + \nu \\ E(i,0) &:= \infty \\ E(0,0) &:= \infty \end{aligned}$$

$$\begin{aligned} F(i,0) &:= i \cdot \mu + \nu \\ F(0,j) &:= \infty \\ F(0,0) &:= \infty \end{aligned}$$

$$\begin{aligned} G(i,0) &:= \infty \\ G(0,j) &:= \infty \\ G(0,0) &:= 0 \end{aligned}$$

Aus der Matrix D berechnen sich dann Distanz und *Alignment* analog zum NW-Algorithmus. Für Ähnlichkeitsmaße und damit auch für den SW-Algorithmus sehen die Rekursionsgleichungen nahezu gleich aus. Es wird nur die Minimierung durch eine Maximierung ersetzt und aufgrund der fehlenden Dreiecksungleichung müssen bei E und F jeweils Insertion und Deletion beachtet werden [Got82].
Neben der Matrix H werden wieder die Matrizen E, F und G benötigt.

- $E(i,j) :=$ Ähnlichkeit des optimalen *Alignments* von $a_1...a_i$ mit $b_1...b_j$, das mit einer Insertion endet.
- $F(i,j) :=$ Ähnlichkeit des optimalen *Alignments* von $a_1...a_i$ mit $b_1...b_j$, das mit einer Deletion endet.

- $G(i,j) :=$ Ähnlichkeit des optimalen *Alignments* von $a_1...a_i$ mit $b_1...b_j$, das mit einer Substitution endet.
- $H(i,j) :=$ Ähnlichkeit des optimalen *Alignments* von $a_1...a_i$ mit $b_1...b_j$.

Für diese Matrizen ergeben sich die folgenden Rekursionsgleichungen.

$$E(i,j) := \max \begin{cases} G(i,j-1) - \mu - \nu & \text{davor war eine Substitution} \\ E(i,j-1) - \mu & \text{davor war eine Insertion} \\ F(i,j-1) - \mu - \nu & \text{davor war eine Deletion} \end{cases} \tag{2.13}$$

$$F(i,j) := \max \begin{cases} G(i-1,j) - \mu - \nu & \text{davor war eine Substitution} \\ F(i-1,j) - \mu & \text{davor war eine Deletion} \\ E(i-1,j) - \mu - \nu & \text{davor war eine Insertion} \end{cases} \tag{2.14}$$

$$G(i,j) := \max \begin{cases} G(i-1,j-1) + w'(a_i,b_j) & \text{davor war eine Substitution} \\ F(i-1,j-1) + w'(a_i,b_j) & \text{davor war eine Deletion} \\ E(i-1,j-1) + w'(a_i,b_j) & \text{davor war eine Insertion} \end{cases} \tag{2.15}$$

$$H(i,j) := \max\{E(i,j), F(i,j), G(i,j), 0\} \tag{2.16}$$

Die Anfangswerte seien für $i > 0$ und $j > 0$:

$$\begin{aligned} E(0,j) &:= j \cdot \mu + \nu \\ E(i,0) &:= 0 \\ E(0,0) &:= 0 \end{aligned}$$

$$\begin{aligned} F(i,0) &:= i \cdot \mu + \nu \\ F(0,j) &:= 0 \\ F(0,0) &:= 0 \end{aligned}$$

$$\begin{aligned} G(i,0) &:= 0 \\ G(0,j) &:= 0 \\ G(0,0) &:= 0 \end{aligned}$$

2.3.3 Neighbor-Joining

Die zur Berechnung von phylogenetischen Bäumen am häufigsten verwendete Methode ist das Neighbor-Joining (NJ). Die NJ-Methode wurde 1986 von Naruya Saitou und Masatoshi Nei veröffentlicht [SN87]. Es handelt sich um eine matrixorientierte Methode, d.h. aus den Sequenzen muss zunächst eine Distanzmatrix berechnet werden. Die Sequenzen selber werden danach nicht mehr verwendet. Das NJ berechnet aus der Distanzmatrix einen *unrooted tree*. Das Ziel des NJ ist, die Gesamtastlänge zu minimieren. Um die Distanz von zwei Sequenzen zu erhalten, berechnet man aus dem *pairwise Alignment* die Jukes-Cantor-Distanz.

Jukes-Cantor-Distanzen

Das 1-Parameter-Modell von Jukes und Cantor (1969) geht davon aus, dass die Wahrscheinlichkeit, dass ein Purin (A und G) zu einem anderen Purin mutiert, genauso groß ist wie die Mutation zu einem Pyrimidin (C und T). Die Wahrscheinlichkeit, dass ein Nukleotid zu einem anderen mutiert sei α. Man nimmt an, dass das Nukleotid an einer bestimmten Position x der DNA sei A zum Zeitpunkt 0. Dann ist $p_{A(0)} := 1$. Die Wahrscheinlichkeit, dass zum Zeitpunkt 1 immer noch A an Position x steht ist gegeben durch:

$$p_{A(1)} = 1 - 3\alpha. \tag{2.17}$$

Die Wahrscheinlichkeit, dass Zeitpunkt 2 ein A an der Position x steht, setzt sich aus zwei Teilen zusammen.
1. das Nukleotid ist nicht mutiert, dann ist die Wahrscheinlichkeit $(1 - 3\alpha)^2 = (1 - 3\alpha) \cdot p_{A(1)}$ oder
2. das Nukleotid ist zum Zeitpunkt 1 zu einem anderen Nukleotid mutiert. Die Wahrscheinlichkeit dafür ist $1 - p_{A(1)}$. Und ist dann zum Zeitpunkt 2 wieder, mit der Wahrscheinlichkeit α, zu A mutiert. Zusammen also $\alpha \cdot (1 - p_{A(1)})$.
Insgesamt ergibt sich dann

$$p_{A(2)} = (1 - 3\alpha)p_{A(1)} + \alpha(1 - p_{A(1)}). \tag{2.18}$$

Für einen beliebigen Zeitpunkt t berechnet sich die Wahrscheinlichkeit aus folgender Gleichung

$$p_{A(t+1)} = (1 - 3\alpha)p_{A(t)} + \alpha(1 - p_{A(t)}). \tag{2.19}$$

Aus Gleichung (2.19) ergibt sich für einen Zeitschritt

$$p_{A(t+1)} - p_{A(t)} = -3\alpha p_{A(t)} + \alpha(1 - p_{A(t)}) = -4\alpha p_{A(t)} + \alpha. \tag{2.20}$$

Gleichung (2.20) lässt sich als lineare DGL 1. Ordnung formulieren

$$\frac{dp_{A(t)}}{dt} = -4\alpha p_{A(t)} + \alpha \tag{2.21}$$

mit der Lösung

$$p_{A(t)} = \frac{1}{4} + (p_{A(0)} - \frac{1}{4})e^{-4\alpha t}. \tag{2.22}$$

Startet man mit A, dann ist $p_A(0) = 1$. Die Wahrscheinlichkeit zum Zeitpunkt t lautet dann:

$$p_{A(t)} = \frac{1}{4} + \frac{3}{4}e^{-4\alpha t}. \tag{2.23}$$

Steht zum Zeitpunkt 0 nicht A an der betrachteten Position x, so ist $p_A(0) = 0$. Die Wahrscheinlichkeit das A zum Zeitpunkt t an Position x steht lautet:

$$p_{A(t)} = \frac{1}{4} - \frac{1}{4}e^{-4\alpha t}. \tag{2.24}$$

Allgemein gilt: Die Wahrscheinlichkeit, dass zum Zeitpunkt t das Nukleotid j an einer Position steht, an der zum Zeitpunkt 0 das Nukleotid i stand, ist:

$$p_{ij(t)} = \frac{1}{4} - \frac{1}{4}e^{-4\alpha t} \qquad \text{für } i \neq j \text{ und} \tag{2.25}$$

$$p_{ii(t)} = \frac{1}{4} + \frac{3}{4}e^{-4\alpha t} \qquad \text{für } i = j. \tag{2.26}$$

Ein allgemeines Maß für die Ähnlichkeit zweier Sequenzen ist der prozentuale Anteil unterschiedlicher Nukleotide in den Sequenzen. Dieser Wert ist gleich der Wahrscheinlichkeit $I_{(t)}$, dass die beiden Nukleotide an einer Stelle der Sequenz zum Zeitpunkt t gleich sind. Angenommen an der betrachteten Stelle der Sequenzen steht zum Zeitpunkt 0 ein $i \in \{A, G, T, C\}$, dann ist

$$I_{(t)} = p^2_{iA(t)} + p^2_{iT(t)} + p^2_{iC(t)} + p^2_{iG(t)}. \tag{2.27}$$

Aus den Gleichungen (2.25) und(2.26) folgt

$$I_{(t)} = \frac{1}{4} + \frac{3}{4}e^{-8\alpha t}. \tag{2.28}$$

Die Wahrscheinlichkeit p, dass die beiden Nukleotide zum Zeitpunkt t unterschiedlich sind, ist dementsprechend:

$$p := 1 - I_{(t)} = \frac{3}{4}(1 - e^{-8\alpha t})$$

oder

$$8\alpha t = -\ln(1 - \frac{4p}{3}) \tag{2.29}$$

Die Zeit t der Divergenz ist im Allgemeinen unbekannt und obwohl man auch α nicht abschätzen kann, kann man einen Wert für K, die aktuelle Anzahl von Substitutionen pro Position seit der Divergenz der beiden Sequenzen, angeben. Für das 1-Parameter Modell ist $K := 2(3\alpha)$. K wird auch als Jukes-Cantor-Distanz (JC-Dist), bezeichnet. Aus Gleichung (2.29) ergibt sich damit für die beiden Sequenzen s_1 und s_2:

$$\text{JC-Dist}(s_1, s_2) := -\frac{3}{4} \cdot \ln(1 - \frac{4p}{3}) \tag{2.30}$$

Wobei p gleich der Hamming-Distanz $H-Dist(s_1, s_2)$ von s_1 und s_2 ist. Die Hamming-Distanz steht für den beobachteten Anteil an verschiedenen Nukleotiden der Sequenzen.

$$\text{H-Dist}(s_1, s_2) := \frac{\text{Anzahl der unterschiedlichen Nukleotide}}{\text{Länge der Sequenz}}$$

[Li97]

Neighbor-Joining Algorithmus

N sei die Anzahl der *operational taxonomic units* (OTUs). Jede Sequenz wird durch einen OTU repräsentiert. Unter Nachbarn versteht man OTUs, die nur durch einen inneren Knoten verbunden sind. D_{ij} sei die Distanz zwischen OTUs i und j und L_{ab} sei die Astlänge zwischen Knoten bzw. OTUs a und b. Für zwei Nachbarn i und j, die durch den inneren Knoten X verbunden sind, gilt:
$L_{iX} + L_{jX} = D_{ij}$
Der Verlauf des NJ ist folgendermaßen: Aus allen OTUs werden zwei Nachbarn ausgewählt, die durch einen inneren Knoten verbunden sind. Dann werden die Astlängen zwischen den beiden OTUs und ihrem inneren Knoten berechnet. Dieser innere Knoten wird zu einem neuen OTU. Jetzt werden zwei Nachbarn aus den übrig gebliebenen OTUs und dem neuen OTU ausgewählt. Dies geht solange weiter, bis nur noch drei OTUs vorhanden sind, und der Baum vollständig berechnet ist. Der erste Schritt, mit gewählten Nachbarn $i = 1$ und $j = 1$, ist in Abbildung 2.4 dargestellt.
Normalerweise ist nicht bekannt, welches Paar von OTUs Nachbarn bilden. Es soll ein Baum konstruiert werden, dessen Gesamtastlänge möglichst gering ist. Die Nachbarn sind daher so zu wählen, dass der daraus entstehende Baum die kleinstmögliche Gesamtastlänge hat. Dazu werden für alle möglichen Paarungen von OTUs die Gesamtastlängen berechnet und das Minimum gebildet.
Im Folgenden seien X und Y wie in Abbildung 2.4 (b) definiert und die gewählten Nachbarn seien mit i und j bezeichnet.

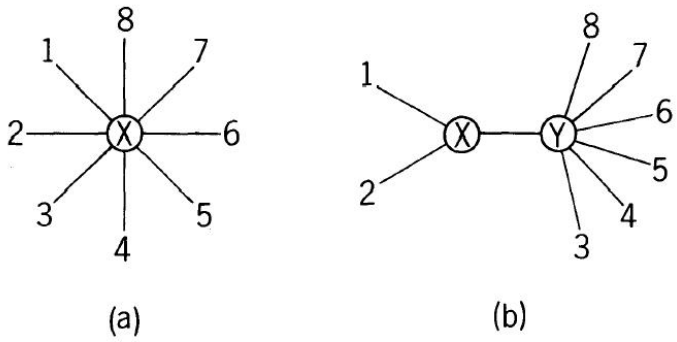


Abbildung 2.4: Graphische Darstellung des NJ für $N = 8$ [SN87]. (a) stellt den Ausgangsbaum da, (b) den Baum nach einem Schritt des Neighbor-Joinings.

Um die Gesamtastlänge zu berechnen, muss das folgende lineare Gleichungssytem gelöst werden:

$$Ax = d, \tag{2.31}$$

wobei

$$\begin{aligned} x^T := (\quad & L_{iX}, L_{jX}, L_{1Y}, ..., L_{(i-1)Y}, L_{(i+1)Y}, ...L_{(j-1)Y}, \\ & L_{(j+1)Y}, ..., L_{NY}, L_{XY}) \in \mathbb{R}^{N+1} \end{aligned} \tag{2.32}$$

der Vektor mit den Astlängen ist. Die Gesamtlänge berechnet sich dann durch

$$\sum_{k:=1}^{N+1} x_k. \tag{2.33}$$

Der Vektor

$$d^T := (D_{12}, D_{13}, ...D_{1N}, D_{23}, D_{24}, ..., D_{2N}, ..., D_{(N-1)N}) \in \mathbb{R}^{N(N-1)/2} \tag{2.34}$$

enthält die Distanzen und die Matrix sei

$$A := (a_{ij})_{1 \leq i \leq N(N-1)/2, 1 \leq j \leq N+1} \tag{2.35}$$

mit

$$a_{ij} := \begin{cases} 1 & \text{falls die } i\text{-te Distanz den } j\text{-ten Ast enthält} \\ 0 & \text{sonst} \end{cases} \tag{2.36}$$

Die Lösung eines LGS $Cx = b$ ist $x = C^{-1}b$. Um C^{-1} berechnen zu können, muss C symmetrisch sein, d.h. $C(x, y) = C(y, x) \quad \forall x, y \in [1, dim(C)]$. Die Matrix A aus (2.31) ist allerdings weder symmetrisch noch quadratisch. Um dennoch eine Lösung zu erhalten, wird das äquivalente System $A^T Ax = A^T d$ gelöst und ergibt den Lösungsvektor $x_L = B^{-1}A^T d$, wobei $B = A^T A$ ist.
B und B^{-1} sehen wie folgt aus:

$$B = \begin{bmatrix} N-1 & 1 & 1 & \dots & 1 & N-2 \\ 1 & N-1 & 1 & \dots & 1 & N-2 \\ 1 & 1 & N-1 & \dots & 1 & 2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & \dots & N-1 & 2 \\ N-2 & N-2 & 2 & \dots & 2 & 2(N-2) \end{bmatrix} \tag{2.37}$$

$$B^{-1} = \begin{bmatrix} a & b & 0 & 0 & 0 & \dots & 0 & e \\ b & a & 0 & 0 & 0 & \dots & 0 & e \\ 0 & 0 & c & d & d & \dots & d & f \\ 0 & 0 & d & c & d & \dots & d & f \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & d & d & d & \dots & c & f \\ e & e & f & f & f & \dots & f & g \end{bmatrix} \tag{2.38}$$

wobei

$$a := \frac{N}{4(N-2)} \quad b := \frac{N-4}{4(N-2)} \quad c := \frac{2N^2 - 11N + 16}{2(N-2)^2(N-3)} \quad d := -\frac{N-4}{2(N-2)^2(N-3)}$$

$$e := -\frac{1}{4} \quad f := -\frac{2}{(N-2)(N-3)} \quad \text{und} \quad g := \frac{N-1}{4(N-3)}$$

ist.
Daraus bekommt man für x_L

$$L_{iX} = \frac{1}{2}D_{ij} + \frac{1}{2(N-2)}(P-Q) \tag{2.39}$$

$$L_{jX} = \frac{1}{2}D_{ij} + \frac{1}{2(N-2)}(Q-P) \tag{2.40}$$

$$L_{kY} = \frac{1}{N-2}U_k - \frac{1}{(N-2)^2}(P+Q) - \frac{N-4}{(N-2)^2(N-3)}V \quad 1 \leq k \leq N \quad k \neq i,j \tag{2.41}$$

$$L_{XY} = \frac{1}{2(N-2)}(P+Q) - \frac{1}{2}D_{ij} - \frac{1}{(N-2)(N-3)}V \tag{2.42}$$

mit $P := \sum_{\substack{k=1 \\ k\neq i,j}}^{N} D_{ik} \quad Q := \sum_{\substack{k=1 \\ k\neq i,j}}^{N} D_{jk} \quad U_k := \sum_{\substack{n=1 \\ n\neq k}}^{N} D_{kn} \quad (k \neq i,j)$

und $\quad V := \sum_{\substack{1\leq k<n \\ k,n\neq i,j}}^{N} D_{kn}.$

Für die Gesamtastlänge S_{ij} mit i und j als Nachbarn, ergibt sich

$$\begin{aligned} S_{ij} &= L_{iX} + L_{jX} + L_{XY} + \sum_{\substack{k=1 \\ k\neq i,j}}^{N} D_{kY} \\ &= \tfrac{1}{2(N-2)}(P+Q) - \tfrac{1}{2}D_{ij} - \tfrac{1}{N-2}V \end{aligned} \tag{2.43}$$

Diese Formel kann man auch wie folgt erklären:
Für einen Baum wie aus Abbildung 2.4 (a) berechnet sich die Summe der Astlängen wie folgt:

$$S_0 = \sum_{i=1}^{N} L_{iX} := \frac{1}{N-1}\sum_{i<j} D_{ij} \tag{2.44}$$

Beispiel. Für $N = 4$ hätte der Baum vier Äste: $1X$,$2X$,$3X$ und $4X$.

$$\begin{aligned} S_0 &= \tfrac{1}{4-1}\textstyle\sum_{i<j} D_{ij} \\ &= \tfrac{1}{3}(D_{12} + D_{13} + D_{14} + D_{23} + D_{24} + D_{34}) \\ &= \tfrac{1}{3}(L_{1X} + L_{2X} + L_{1X} + L_{3X} + L_{1X} + L_{4X} + \\ &\quad L_{2X} + L_{3X} + L_{2X} + L_{4X} + L_{3X} + L_{4X}) \\ &= L_{1X} + L_{2X} + L_{3X} + L_{4X} \\ &= \textstyle\sum_{i=1}^{4} L_{iX} \end{aligned}$$

Die Astlänge zwischen den Knoten X und Y des Baumes aus Abbildung 2.4 (b) kann man mit dieser Gleichung berechnen:

$$L_{XY} = \frac{1}{2(N-2)} \underbrace{\left[\sum_{i=3}^{N}(D_{1i}+D_{2i})\right.}_{\text{I}} - \underbrace{(N-2)(L_{1X}+L_{2X})}_{\text{II}} - \underbrace{\left.2\sum_{i=3}N L_{iY}\right]}_{\text{III}} \quad (2.45)$$

I: Alle $2(N-2)$ Distanzen, die das Stück L_{XY} enthalten, werden zunächst aufaddiert.
II: Der "linke Teil" des Baumes wird nun subtrahiert, sowohl L_{1X} als auch L_{2X} wurden in (I) $(N-2)$-mal durchlaufen.
III: Zuletzt muss noch der "rechte Teil", der zuviel ist, subtrahiert werden. Jedes Stück L_{iY} wurde in (I) 2-mal durchlaufen.

Aus Gleichung 2.44 folgt:

$$\sum_{i=3}^{N} L_{iY} := \frac{1}{N-3} \sum_{3 \le i < j} D_{ij} \quad (2.46)$$

Aus allen drei Gleichungen folgt daraus für die Summe der Astlängen des Baumes aus Abbildung 2.4 (b) mit i und j als Nachbarn (in der Skizze i=1 und j=2)

$$\begin{aligned} S_{ij} &= L_{XY} + (L_{iX} + L_{jX}) + \sum_{\substack{k=1 \\ k \neq i,j}}^{N} L_{kY} \\ &= \frac{1}{2(N-2)} \sum_{\substack{k=1 \\ k \neq i,j}}^{N} (D_{ik} + D_{jk}) + \frac{1}{2} D_{ij} + \frac{1}{N-2} \sum_{\substack{k<l \\ k,l \neq i,j}} D_{kl} \end{aligned} \quad (2.47)$$

Gleichung 2.47 erlaubt uns für jede mögliche Paarung von Nachbarn die Gesamtastlänge des Baumes zu berechnen. Ziel des NJ ist es, die Gesamtastlänge des Baumes zu minimieren. Wir wählen unsere Nachbarn i und j so, das gilt: $S_{ij} = \min_{i,j} S_{ij}$. Der Beweis, dass diese beiden echte Nachbarn sind, geht mittels Induktion (siehe [SN87]).
Die OTUs i und j werden nun durch den inneren Knoten X verbunden. Die Längen von L_{iX} und L_{jX} berechnen sich mit den Gleichungen:

$$L_{iX} := (D_{ij} + D_{iZ} - D_{jZ})/2 \quad (2.48)$$

$$L_{jX} := (D_{ij} + D_{jZ} - D_{iZ})/2 \quad (2.49)$$

mit

$$D_{iZ} := (\sum_{\substack{k=1 \\ k \neq i,j}}^{N} D_{ik})/(N-2) \text{ und } D_{jZ} := (\sum_{\substack{k=1 \\ k \neq i,j}}^{N} D_{jk})/(N-2) \quad (2.50)$$

Für die weitere Rechnung werden zusätzlich die Distanzen zwischen X und allen OTUs, außer i und j, benötigt.

$$D_{Xk} = (D_{ik} - L_{iX} + D_{jk} - L_{jX})/2 \qquad \forall 1 \leq k \leq N, k \neq i, k \neq j \tag{2.51}$$

In der Distanzmatrix wird nun die zu OTU i gehörige Zeile/Spalte durch eine Zeile/Spalte mit den zu X gehörigen Daten ersetzt. Die zu OTU j ($i < j$) gehörige Zeile/Spalte wird gestrichen. Mit der so entstandenen $(N-1) \times (N-1)$-Distanzmatrix wird der selbe Algorithmus durchgeführt, solange die Dimension der Distanzmatrix größer als drei ist. Für eine 3×3-Matrix ist eine Nachbarnsuche nicht möglich. Es werden nur die Astlängen zwischen den drei OTUs und einem inneren Knoten berechnet. Der NJ-Algorithmus berechnet einen *unrooted tree*.

2.3.4 Maximum-Likelihood

1981 verwendete J. Felsenstein den Maximum-Likelihood(ML)-Algorithmus um phylogenetische Bäume aus Nukleotidsequenzen zu berechnen. Das Ziel dieser Methode ist, einen Baum zu erzeugen, der einen maximalen Likelihood-Wert hat. Ausgehend von einer gegebenen Baumtopologie wird die ML Methode verwendet, um die optimalen Astlängen zu bestimmen. Danach werden lokale Änderungen an der Topologie gemacht und die Astlängen erneut optimiert usw. [CB00].
Die Sequenzdaten D (n alignte Sequenzen der Länge m) sind das Ergebniss eines zufälligen Prozesses. Ihre Wahrscheinlichkeit hängt von einem unbekannten Baum θ ab.
ML-Prinzip: Finde den Baum $\hat{\theta}$, für den die Likelihood-Funktion

$$L_D(\theta) := \prod_{s=1}^{m} L_{D^{(s)}}(\theta) \tag{2.52}$$

maximal wird. D.h. für den Baum $\hat{\theta}$ ist das Eintreten der Sequenzdaten am wahrscheinlichsten.
Die Wahrscheinlichkeit $L_{D^{(s)}}(\theta)$ hängt ebenfalls von dem unbekannten Parameter θ ab. $D^{(s)}$ sind die Sequenzdaten an der s-ten Position.
Zwei zentrale Vorausetzungen für das ML-Prinzip sind:

1. Die Evolution in den verschiedenen Positionen ist unabhängig.
2. Die Evolution in den verschiedenen *Lineages* ist unabhängig.

Damit $L_D(\theta)$ definiert ist, benötigt man ein Evolutionsmodel, dass den Substitutionsprozess, der längs der Kanten abläuft, beschreibt. Dafür spezifiziere man die Wahrscheinlichkeit

$$P_{x \to y}(t) := p_{xy(t)}, \tag{2.53}$$

dass an einer Position, wo ein x steht, nach der Zeit t ein y steht. Bei Verwendung des oben beschriebenen Jukes-Cantor-Modell ergibt sich aus den Gleichungen 2.25

und 2.26, mit einer Basenhäufigkeit der Ursequenz von
$(\pi(A), \pi(C), \pi(G), \pi(T)) = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ (d.h. $\alpha = \frac{1}{4}$) für $P_{x \to y}(t)$:

$$P_{x \to y}(t) := \begin{cases} (1 - e^{-t}) \cdot \frac{1}{4} & \text{falls } x \neq y \\ (1 + 3e^{-t}) \cdot \frac{1}{4} & \text{falls } x = y \end{cases} \tag{2.54}$$

Für einen gegebenen Baum θ (Topologie und Astlängen) mit unbekannten inneren Knoten kann man

$$L_{D^{(s)}}(\theta)$$

wie folgt, rekursiv berechnen.
Steht am Knoten k die Base x, dann ist $W_k(x)$ die Wahrscheinlichkeit, dass die Sequenzdaten $D_k^{(s)}$ eintreten, wenn an dem Knoten k die Base x steht. Dabei ist D_k der Datensatz der äußeren Knoten, die Nachfolger vom Knoten k sind. D.h. D_k enthält nur die Sequenzen von D, die zu einem Nachfolger von k gehören. $D_k^{(s)}$ ist dann die s-te Spalte von D_k
Ist k ein äußerer Knoten an dem die Base x steht, so ist $W_k(x) = 1$ und $W_k(y) = 0$ für $y \neq x$.
Ist k ein innerer Knoten mit den direkten Nachfolgern i und j und den Astlängen l_i und l_j, dann gilt:

$$W_k(x) := \left(\sum_{y \in \{A,C,G,T\}} P_{x \to y}(l_i) \cdot W_i(y) \right) \cdot \left(\sum_{z \in \{A,C,G,T\}} P_{x \to z}(l_j) \cdot W_j(z) \right) \tag{2.55}$$

Für die Wurzel r gilt $D_r(s) = D(s)$, daher ergibt sich

$$L_{D^{(s)}}(\theta) = \sum_{x \in \{A,C,G,T\}} \pi(x) \cdot W_r(x) \tag{2.56}$$

[Fel04, Fel81].

Der Wert $L_D(\theta)$ soll maximiert werden. Da die Werte von $L_D(\theta)$ sehr klein sind, wird anstelle des direkten Wertes der natürliche Logarithmus

$$ln(L_D(\theta)) = \sum_{s=1}^{m} ln(L_{D^{(s)}}(\theta)) \tag{2.57}$$

ausgewertet.

2.3.5 Bootstrapping

Das *Bootstrapping* wurde 1979 von Bradley Efron veröffentlicht [Efr79]. *Bootstrapping* ist ein statistisches Verfahren, um die Variabilität eines Schätzers zu bestimmen. In dieser Arbeit wird das *parametric bootstrapping* verwendet [Fel04].

Als Schätzer fungiert hier der NJ-Algorithmus und als Eingabedaten dienen die n Sequenzen der Länge m. Unter Variabilität versteht man die Stabilität des Baumes, wenn die Eingabedaten sich ändern. Bei unendlich langen Sequenzen könnte man immer ein anderes Sequenzstück der Länge m nehmen und den daraus entstehenden Baum mit dem ersten vergleichen. Da die Sequenzen aber nur die Länge m haben, zieht man m neue Stichproben (mit Zurücklegen) aus den m vorhandenen Stichproben.
Die Sequenzen werden in einer $n \times m$-Datenmatrix $\mathbf{x}$ gespeichert. Aus $\mathbf{x}$ wird eine Distanzmatrix $\hat{D}$ berechnet und daraus mit NJ ein phylogenetischer Baum $\hat{\text{TREE}}$.

$$\mathbf{x} \rightarrow \hat{\mathbf{D}} \rightarrow \hat{\mathbf{TREE}}$$

Die *Bootstrap*-Datenmatrix $\mathbf{x}^*$ ergibt sich aus $\mathbf{x}$ mittels folgendem Verfahren: Es wird m-mal zufällig eine Zahl $y_i \in [1, m]$ $1 \leq i \leq m$ gewählt. Die i-te Spalte von x^* ist dann die y_i-te Spalte von x. Jede Spalte aus x kann gar nicht, einmal oder auch mehrmals vorkommen. Der Bootstrap $\hat{\text{TREE}}^*$ wird dann aus $\mathbf{x}^*$ wie $\hat{\text{TREE}}$ aus $\mathbf{x}$ berechnet.

$$\mathbf{x}^* \rightarrow \hat{\mathbf{D}}^* \rightarrow \hat{\mathbf{TREE}}^*$$

Auf diese Art werden B *Bootstrap*-Bäume berechnet. Um den *Bootstrap*-Wert $Boot_i$ eines Splits i aus dem Originalbaum $\hat{\text{TREE}}$ zu berechnen, werden zunächst alle *Bootstrap*-Bäume gezählt, in denen der Split auch vorhanden ist. Dies seien b_i Bäume. $Boot_i$ ergibt sich dann aus:

$$Boot_i := \frac{b_i}{B} * 100$$

Bei phylogenetischen Bäumen sind *Bootstrap*-Werte ein statistischer Wert für die Stabilität der Gruppen. Hohe *Bootstrap*-Werte ($> 85\%$) bedeuten eine hohe Unterstützung der Gruppierung durch die gegebenen Sequenzen. Im Gegensatz dazu bedeuten niedrige Werte nicht, dass die Gruppierung falsch ist, sondern dass sie von den gegebenen Sequenzen nicht ausreichend unterstützt wird, z.B. weil zu wenig Vergleichsdaten verfügbar sind oder die Sequenzen zu kurz oder zu ähnlich sind [Fel89].

Kapitel 3

Optimierungsmethoden für den Maximum-Likelihood Algorithmus

Der Maximum-Likelihood Algorithmus ist ein möglicher Algorithmus zur Berechnung phylogenetischer Bäume. Ziel dieser Methode ist es, Topologie und Astlänge so zu wählen, dass der resultierende Baum einen maximalen Likelihood-Wert hat. Dieser Wert wird mit der Likelihood-Funktion $\ln(L_D(\theta))$ berechnet.
Die Berechnung des maximalen Wertes ergibt eine Kombination aus zwei ineinander eingebetteten Optimierungsproblemen: dem Finden der optimalen Topologie und die Berechnung der optimalen Astlängen für diese Topologie.
Für die Berechnung der Astlängen ergibt sich ein kontinuierliches Optimierungsproblem mit Nebenbedingungen:

$$\max_{l \in \mathbb{R}^{2n-2}} \ln(L_D(\theta)) \text{ unter der Nebenbedingung}$$
$$l_i \geq 0 \; \forall \; 1 \leq i \leq 2n-2$$

wobei l_i die Astlängen sind.
Zur Lösung dieses Optimierungsproblems wird eine projizierte Variante des konjugierten Gradientenverfahrens verwendet.
Das Finden der optimalen Topologie dagegen ist ein diskretes Optimierungsproblem. Dieses diskrete Problem wird mit einem iterativen Algorithmus, der auf lokalen Änderungen in der Topologie basiert, gelöst.

3.1 Berechnung der optimalen Astlängen

Bei gegebener Topologie ist der Likelihood-Wert $\ln(L_D(\theta))$ eines Baums θ nur noch von den Astlängen $l := (l_i)_{1 \leq i \leq 2n-2}$ abhängig. Zur Vereinfachung sei der Likelihood-Wert für eine gegebene Topologie mit $\ln L_D(l)$ bezeichnet. Der Wert $\ln L_D(l)$ soll maximiert werden. Die folgende Bedingung garantiert nichtnegative Astlängen:

$$l_i \geq 0 \; \forall \; 1 \leq i \leq 2n-2$$

Daraus ergibt sich zusammen das Optimierungsproblem:

$$\max_{l \in \mathbb{R}^{2n-2}} \ln L_D(l) \text{ unter der Nebenbedingung } l_i \geq 0 \ \forall \ 1 \leq i \leq 2n-2 \tag{3.1}$$

3.1.1 Verfahren der konjugierten Gradienten

Die Grundidee der Verfahren der konjugierten Gradienten ist die sukzessive Minimierung der Zielfunktion f in eindimensionalen Unterräumen, ausgehend von der letzten Näherung. Unterschiede zwischen den Verfahren bestehen in der Wahl der Suchrichtung p. Man setzt voraus, dass $f : D \subset \mathbb{R}^n \to \mathbb{R}$ stetig differenzierbar ist. Beim klassischen Gradientenverfahren wird als Suchrichtung der steilste Abstieg gewählt, d.h. $p = -\nabla f(x)$. Ist der Winkel zwischen den Suchrichtungen sehr klein, konvergiert das Verfahren unnötig langsam. Verfahren, bei denen die Suchrichtung eine Linearkombination der Gradientenrichtung und der Suchrichtung aus dem letzten Schritt ist, konvergieren hier deutlich schneller. Die bekanntesten dieser Verfahren der konjugierten Gradienten stammen von Fletcher/Reeves (1964) und Polak/Ribiere (1969).

Algorithmus 3.1. (Fletcher-Reeves)

Startnäherung $x^{(0)}$ gegeben; $k = 0$;
$p^{(0)} = -\nabla f(x^{(0)})$;
`while` $\nabla f(x^{(k)}) \neq 0$

Bestimme α_k aus $\min_{\alpha \in \mathbb{R}} f(x^{(k)} + \alpha p^{(k)})$;

$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}; \beta_{k+1}^{FR} = \frac{\nabla f(x^{(k+1)})^T \nabla f(x^{(k+1)})}{\nabla f(x^{(k)})^T \nabla f(x^{(k)})}$

$p^{(k+1)} = -\nabla f(x^{(k+1)}) + \beta_{k+1}^{FR} p^{(k)}; k = k + 1;$

`end`

Algorithmus 3.2. (Polak-Ribiere)

Startnäherung $x^{(0)}$ gegeben; $k = 0$;
$p^{(0)} = -\nabla f(x^{(0)})$;
`while` $\nabla f(x^{(k)}) \neq 0$

Bestimme α_k aus $\min_{\alpha \in \mathbb{R}} f(x^{(k)} + \alpha p^{(k)})$;

$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}; \beta_{k+1}^{PR} = \frac{\nabla f(x^{(k+1)})^T (\nabla f(x^{(k+1)}) - \nabla f(x^{(k)}))}{\nabla f(x^{(k)})^T \nabla f(x^{(k)})}$

$p^{(k+1)} = -\nabla f(x^{(k+1)}) + \beta_{k+1}^{PR} p^{(k)}; k = k + 1;$

`end`

Um einen praktikablen Algorithmus zu erhalten, muss man in der `while`-Schleife ein Abbruchkriterium der Form $\|\nabla f(x^{(k)}\| \leq \varepsilon$ einfügen und das eindimensionale Minimierungsproblem

$$\min_{\alpha \in \mathbb{R}} f(x^{(k)} + \alpha p^{(k)}) \tag{3.2}$$

lösen.
Neben der aufwendigen exakten Lösung kann man auch einen Schritt des eindimensionalen Newton-Verfahrens durchführen. Für α_k ergibt sich dann:

$$\alpha_k = -\frac{\nabla f(x^{(k)})^T p^{(k)}}{(p^{(k)})^T(\nabla^2 f(x^{(k)}))p^{(k)}} \tag{3.3}$$

α_k muss so gewählt sein, dass eine gewisse Reduktion des Funktionals gesichert ist. Dafür fordert man, dass die 1.Wolfe-Bedingung:

$$f(x^{(k)} + \alpha_k p^{(k)}) \leq f(x^{(k)}) + c_1 \alpha_k \nabla f(x^{(k)})^T p^{(k)} \tag{3.4}$$

mit einer Konstanten $c_1 \in (0, 1)$ erfüllt ist. Erfüllt α_k diese Bedingung nicht, halbiert man α_k solange, bis die Bedingung erfüllt ist.
Hieraus ergibt sich eine weitere Möglichkeit α_k zu bestimmen, ohne das eindimensionale Minimierungsproblem zu lösen: Man wählt einen geeigneten Startwert für α_k und halbiert ihn solange, bis die 1. Wolfe Bedingung erfüllt ist.

3.1.2 Projektionsverfahren

Projektionsverfahren sind eine Verallgemeinerung der Gradientenverfahren, um Minimierungsprobleme unter Nebenbedingungen, die in der Praxis häufig auftreten, lösen zu können.
Als Minimierungsproblem ergibt sich:

$$\text{minimiere } f(x) \text{ für } x \in \mathbb{R}^n \text{ mit } x \in G \tag{3.5}$$

Vorausgesetzt wird, dass die Zielfunktion f stetig differenzierbar ist und dass die zulässige Menge G nichtleer, konvex und abgeschlossen ist.
Grundlage des Verfahrens ist die orthogonale Projektion:

Definition 3.1. *Für $x \in \mathbb{R}^n$ ist*

$$P_G(x) := arg \min_{y \in G} \|y - x\|$$

die Orthogonale Projektion von x auf G bezüglich der euklidischen Norm $\|.\|$

Satz 3.1. *(Projektionssatz)*
Seien $G \subseteq \mathbb{R}^n$ nichtleer, abgeschlossen und konvex sowie $y \in \mathbb{R}^n$ beliebig gegeben. Dann ist $z \in G$ genau dann gleich der Projektion von y auf G, wenn

$$(z - y)^T(x - z) \leq \quad \textit{für alle } x \in G$$

gilt.

Beweis siehe [GK02], Seite 29-30.

Definition 3.2. *Die zulässige Menge $G \subseteq \mathbb{R}^n$ sei konvex. Für jeden Punkt $x^* \in G$ ist der Tangentialkegel $T_G(x^*)$ definiert als kleinster abgeschlossener konvexer Kegel, der die Menge*

$$\{d = z - x^* : z \in G\}$$

enthält.

Definition 3.3. *Sei $f : \mathbb{R}^n \to \mathbb{R}$ differenzierbar. $x^* \in \mathbb{R}^n$ heißt stationärer Punkt von f, falls $\nabla f(x^*) = 0$.*

Eine Verallgemeinerung des Begriffes des stationären Punktes auf Minimierungsprobleme unter Nebenbedingungen ist:

Definition 3.4. *Der Punkt $x^* \in G$ wird als stationärer Punkt von (3.5) bezeichnet, falls*

$$\nabla f(x^*)^T d \geq 0$$

für alle $d \in T_G(x^*)$ gilt.

Zur Vereinfachung sei im Folgenden $x^{(k)}(\alpha_k) := P_G(x^{(k)} + \alpha_k p^{(k)})$.
Als Algorithmus des projizierten Gradientenverfahren ergibt sich:

Algorithmus 3.3. (Projiziertes Gradientenverfahren)
Startnäherung $x^{(0)} \in G$ gegeben; Parameter $c_1 \in (0,1)$, $\gamma > 0$; $k = 0$;
`while` $x^{(k)}$ kein stationärer Punkt
$p^{(k)} = -\nabla f(x^{(k)})$;
$x^{(k+1)} = x^{(k)}(\alpha_k)$, wobei $\alpha_k := \min\limits_{n_k \in \mathbb{N}} \gamma \cdot (\frac{1}{2})^{n_k}$ mit

$$f(x^{(k)}(\alpha_k)) \leq f(x^{(k)}) - c_1 (p^{(k)})^T (x^{(k)}(\alpha_k) - x^{(k)}) \tag{3.6}$$

`end`

Für $G = \mathbb{R}^n$ ist die Bedingung (3.6) identisch mit der 1.Wolfe-Bedingung.
Um zu zeigen, dass die Schrittweite dieses Algorithmus wohldefiniert ist, benötigt man den folgenden Satz und das danach folgende Lemma.

Satz 3.2. *Seien $f : \mathbb{R}^n \to \mathbb{R}$ stetig differenzierbar und $G \subseteq \mathbb{R}^n$ nichtleer, abgeschlossen und konvex sowie $\gamma > 0$. Dann gilt: f ist konvex und es gilt*

$$\nabla f(x^*)^T (x - x^*) \geq 0 \ \ \forall x \in G$$

genau dann, wenn $x = x^*$ der Fixpunktgleichung

$$x = P_G(x - \gamma \nabla f(x))$$

genügt.

Beweis siehe [GK02], Seite 296.

Lemma 3.1. *Sei $G \subseteq \mathbb{R}^n$ nichtleer, abgeschlossen und konvex. Seien $x \in \mathbb{R}^n$ und $d \in \mathbb{R}^n$ gegeben. Dann ist die Funktion*

$$\theta(\alpha) := \frac{\|P_G(x + \alpha d) - x\|}{\alpha}$$

für $\alpha > 0$ monoton fallend.

Beweis siehe [GK02], Seite 298-299.

Lemma 3.2. *Die Schrittweite im Algorithmus des projizierten Gradientenverfahrens ist wohldefiniert, d.h. in jeder Iteration k existiert ein $n_k \in \mathbb{N}$, so dass $\alpha_k = \gamma \cdot (\frac{1}{2})^{n_k}$ der Bedingung (3.6) genügt.*

Beweis. Sei $x^{(k)} \in G$ eine gegebene Iterierte. Aus dem Abbruchkriterium und Satz 3.3 folgt, dass

$$\|x^{(k)} - x^{(k)}(\alpha_k)\| > 0 \ \ \forall \alpha_k > 0. \tag{3.7}$$

Aufgrund des Projektionssatzes 3.1 ist

$$(x^{(k)}(\alpha_k) - x^{(k)})^T (x^{(k)} + \alpha_k p^{(k)} - x^{(k)}(\alpha_k)) > 0 \ \ \forall \alpha_k > 0. \tag{3.8}$$

Daher folgt

$$(-p^{(k)})^T (x^{(k)} - x^{(k)}(\alpha_k)) \leq \frac{\|x^{(k)} - x^{(k)}(\alpha_k)\|^2}{\alpha_k}, \tag{3.9}$$

und Lemma 3.1 impliziert somit

$$(-p^{(k)})^T (x^{(k)} - x^{(k)}(\alpha_k)) \leq \frac{\|x^{(k)} - x^{(k)}(\alpha_k)\|^2}{\alpha_k} \leq \frac{\|x^{(k)} - x^{(k)}(\gamma)\|}{\gamma} \|x^{(k)} - x^{(k)}(\alpha_k)\| \ \ \forall \alpha_k > 0. \tag{3.10}$$

Aus dem Mittelwertsatz der Differentialrechnung folgt, dass es zu jedem $\alpha \in (0, 1]$ einen Zwischenpunkt $\xi^k_{\alpha_k}$ auf der Verbindungsstrecke von $x^{(k)}$ und $x^{(k)}(\alpha_k)$ gibt mit

$f(x^{(k)}) - f(x^{(k)}(\alpha_k))$
$= \nabla f(\xi^k_{\alpha_k})^T (x^{(k)} - x^{(k)}(\alpha_k))$
$= \nabla f(x^{(k)})^T (x^{(k)} - x^{(k)}(\alpha_k)) + (\nabla f(\xi^k_{\alpha_k}) - \nabla f(x^{(k)}))^T (x^{(k)} - x^{(k)}(\alpha_k))$
$f(x^{(k)}) - f(x^{(k)}(\alpha_k)) \geq -c_1 (p^{(k)})^T (x^{(k)} - (x^{(k)}(\alpha_k))$
$\Leftrightarrow (1 - c_1) \nabla f(x^{(k)})^T (x^{(k)} - x^{(k)}(\alpha_k)) \geq (-\nabla f(x^{(k)}) - \nabla f(\xi^k_{\alpha_k}))^T (x^{(k)} - x^{(k)}(\alpha_k))$,
weil $p^{(k)} = -\nabla f(x^{(k)})$

Wegen 3.9 ist dies sicherlich erfüllt, wenn

$(1 - c_1) \|x^{(k)} - x^{(k)}(\gamma)\| \geq \frac{1}{\gamma} (-\nabla f(x^{(k)}) - \nabla f(\xi^k_{\alpha_k}))^T \frac{(x^{(k)} - x^{(k)}(\alpha_k))}{\|x^{(k)} - x^{(k)}(\alpha_k)\|}$
gilt.
Die Gültigkeit dieser Ungleichung für alle hinreichend kleinen $t > 0$ folgt aber unmittelbar aus der Tatsache, dass die linke Seite wegen 3.7 strikt positiv ist und die

rechte Seite für $t \downarrow 0$ offenbar gegen Null konvergiert.
[GK02] □

Insgesamt ergeben sich die folgenden Algorithmen zur Lösung von Minimierungsproblemen unter Nebenbedingungen mit Hilfe projizierter Verfahren der konjugierten Gradienten.

Algorithmus 3.4. ($\mathrm{FR}_{\min}$)
Fletcher-Reeves (eindimensionales Minimierungsproblem)
Startnäherung $x^{(0)} \in G$ gegeben; $k = 0$;
$p^{(0)} = -\nabla f(x^{(0)})$;
`while` $x^{(k)}$ kein stationärer Punkt
Bestimme α_k aus $\min\limits_{\alpha \in \mathbb{R}} f(P_G(x^{(k)} + \alpha p^{(k)}))$;
$x^{(k+1)} = P_G(x^{(k)} + \alpha_k p^{(k)}); \beta_{k+1}^{FR} = \frac{\nabla f(x^{(k+1)})^T \nabla f(x^{(k+1)})}{\nabla f(x^{(k)})^T \nabla f(x^{(k)})}$
$p^{(k+1)} = -\nabla f(x^{(k+1)}) + \beta_{k+1}^{FR} p^{(k)}; k = k + 1;$
`end`

Algorithmus 3.5. ($\mathrm{FR}_{\mathrm{redukt}}$)
Fletcher-Reeves (Reduktionsbedingung)
Startnäherung $x^{(0)} \in G$, Parameter $\gamma > 0$, $c_1 \in (0, 1)$ und $\delta > 1$ gegeben; $k = 0$;
$p^{(0)} = -\nabla f(x^{(0)})$;
`while` $x^{(k)}$ kein stationärer Punkt
$\alpha = \gamma$
`while` $f(P_G(x^{(k)} + \alpha_k p^{(k)})) > f(x^k) - c_1 \nabla f(x^{(k)})(P_G(x^{(k)} + \alpha p^{(k)}) - x^{(k)})$
$\alpha = \alpha/\delta$;
`end`
$\alpha_k = \alpha$;
$x^{(k+1)} = P_G(x^{(k)} + \alpha_k p^{(k)}); \beta_{k+1}^{FR} = \frac{\nabla f(x^{(k+1)})^T \nabla f(x^{(k+1)})}{\nabla f(x^{(k)})^T \nabla f(x^{(k)})}$
$p^{(k+1)} = -\nabla f(x^{(k+1)}) + \beta_{k+1}^{FR} p^{(k)}; k = k + 1;$
`end`

Algorithmus 3.6. ($\mathrm{PR}_{\min}$)
Polak-Ribiere (eindimensionales Minimierungsproblem)
Startnäherung $x^{(0)} \in G$ gegeben; $k = 0$;
$p^{(0)} = -\nabla f(x^{(0)})$;
`while` $x^{(k)}$ kein stationärer Punkt
Bestimme α_k aus $\min\limits_{\alpha \in \mathbb{R}} f(P_G(x^{(k)} + \alpha p^{(k)}))$;
$x^{(k+1)} = P_G(x^{(k)} + \alpha_k p^{(k)}); \beta_{k+1}^{PR} = \frac{\nabla f(x^{(k+1)})^T (\nabla f(x^{(k+1)}) - \nabla f(x^{(k)}))}{\nabla f(x^{(k)})^T \nabla f(x^{(k)})}$
$p^{(k+1)} = -\nabla f(x^{(k+1)}) + \beta_{k+1}^{PR} p^{(k)}; k = k + 1;$

`end`

Algorithmus 3.7. ($\mathrm{PR}_{\mathrm{redukt}}$)
Polak-Ribiere (Reduktionsbedingung)
Startnäherung $x^{(0)} \in G$, Parameter $\gamma > 0$, $c_1 \in (0,1)$ und $\delta > 1$ gegeben; $k = 0$;
$p^{(0)} = -\nabla f(x^{(0)})$;
`while` $x^{(k)}$ kein stationärer Punkt

$\alpha = \gamma$
`while` $f(P_G(x^{(k)} + \alpha_k p^{(k)})) > f(x^k) - c_1 \nabla f(x^{(k)})^T (P_G(x^{(k)} + \alpha p^{(k)}) - x^{(k)})$
$\alpha = \alpha/\delta$;
`end`
$\alpha_k = \alpha$;
$x^{(k+1)} = P_G(x^{(k)} + \alpha_k p^{(k)}); \beta_{k+1}^{PR} = \frac{\nabla f(x^{(k+1)})^T (\nabla f(x^{(k+1)}) - \nabla f(x^{(k)}))}{\nabla f(x^{(k)})^T \nabla f(x^{(k)})}$
$p^{(k+1)} = -\nabla f(x^{(k+1)}) + \beta_{k+1}^{PR} p^{(k)}; k = k + 1;$

`end`

Aus dem folgenden Lemma ergibt sich als praktikables Abbruchkriterium für die `while`-Schleife: $\|\nabla(f(P_G(x^{(k)}))\| \leq \varepsilon$.

Lemma 3.3. *Die zulässige Menge $G \subseteq \mathbb{R}^n$ sei konvex. Für jedes $x^* \in G$ erfüllt die orthogonale Projektion $P_{T_{G(x^*)}}(-\nabla f(x^*))$ der Richtung des steilsten Abstiegs auf den Tangentialkegel $T_{G(x^*)}$ die folgenden Eigenschaften:*
(a) $\nabla f(x^*)^T P_{T_{G(x^*)}}(-\nabla f(x^*)) = -\|P_{T_{G(x^*)}}(-\nabla f(x^*))\|^2$;
(b) $min\{\nabla f(x^*)^T v : v \in T_{G(x^*)}, \|v\| \leq 1\} = -\|P_{T_{G(x^*)}}(-\nabla f(x^*))\|$;
(c) x^* ist stationärer Punkt des Problems genau dann, wenn $P_{T_{G(x^*)}}(-\nabla f(x^*)) = 0$.

Beweis siehe [JS04], Seite 280.

3.1.3 Konvexität und lokale Minima

Bei der Minimierung einer Zielfunktion $f : \mathbb{R}^n \to \mathbb{R}$ sucht man ein globales Minimum von f. Unter einem globalen Minimum versteht man einen Punkt $x^* \in \mathbb{R}^n$ mit $f(x^*) \leq f(x)$ für alle $x \in \mathbb{R}^n$. Meistens ist es aufgrund des Aufwands nur möglich, ein lokales Minimum von f in einer Umgebung eines Startwertes $x^{(0)}$ zu bestimmen. Dies ist ein Punkt $x^* \in \mathbb{R}^n$ mit $f(x^*) \leq f(x)$ für alle x in einer Umgebung von x^*. Gilt sogar $f(x^*) < f(x)$ für alle x in einer Umgebung von x^*, spricht man von einem strengen lokalen Minimum.

Bemerkung.

(a) Ist x^* ein lokales Minimum von f und ist die Hessematrix H_f stetig in einer Umgebung von x^*, dann gilt $\nabla f(x^*) = 0$ und $H_f(x^*)$ positiv semidefinit.

(b) Ist die Hessematrix H_f stetig in einer Umgebung von x^* mit $\nabla f(x^*) = 0$ und ist $H_f(x^*)$ positiv definit, dann ist x^* ein strenges lokales Minimum von f.

Definition 3.5. *(Konvexe Funktion)*
Es ist $D \subseteq \mathbb{R}^n$ ist eine konvexe Menge. Die Funktion $f : D \to \mathbb{R}$ heißt konvex in D, wenn

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y)$$

für alpha $\in [0,1]$ und alle $x, y \in D$ gilt.
f heißt streng konvex, wenn für alle $x \neq y \in D$ und alpha $\in [0,1]$ gilt

$$f(\alpha x + (1-\alpha)y) < \alpha f(x) + (1-\alpha)f(y).$$

Bemerkung. Es gilt das folgende Konvexitätskriterium:
Sei $D \subseteq \mathbb{R}^n$ eine offene konvexe Menge, und $f : D \to \mathbb{R}$ zweimal differenzierbar.
Dann ist f streng konvex, wenn die Hessematrix von f für alle $x \in D$ positiv definit ist.

Satz 3.3. *Seien $f : \mathbb{R}^n \to \mathbb{R}$ stetig differenzierbar und $G \subseteq \mathbb{R}^n$ nichtleer, abgeschlossen und konvex. Ist f konvex und gilt*

$$\nabla f(x^*)^T (x - x^*) \geq 0 \ \ \forall x \in G$$

so ist x^* ein (lokales = globales) Minimum von (3.5).

Beweis siehe [GK02], Seite 295.

Für den Spezialfall einer konvexen Zielfunktion folgt aus Satz 3.3, dass jeder stationärer Punkt ein globales Minimum des Optimierungsproblems ist.

3.1.4 Umsetzung für die Likelihood-Funktion

Das Optimierungsproblem (3.1) soll mit den im Abschnitt (3.1.2) beschriebenen Algorithmen (3.4) bis (3.7) gelöst werden. Damit diese praktikabel sind, muss man ein Abbruchkriterium für die `while`-Schleife bestimmen. Für die Algorithmen $\mathrm{FR}_{\mathrm{Min}}$ und $\mathrm{PR}_{\mathrm{Min}}$ benötigt man außerdem eine Möglichkeit, das eindimensionale Minimierungsproblem effizient zu lösen. Die anderen beiden benötigen nur einen geeigneten Wert für γ.
Das Abbruchkriterium wurde folgendermaßen gewählt:

$$\|\nabla(f(x^{(k)}))\| \leq \varepsilon \cdot \frac{|f(x)|}{\overline{x}}, \quad \varepsilon > 0. \tag{3.11}$$

Wobei

$$\overline{x} := \frac{1}{n} \sum_{i=1}^{n} x_i \tag{3.12}$$

ist. Zusätzlich wurde auch die Anzahl der Iterationsschritte (it) begrenzt.
Für die Lösung des Minimierungsproblems und damit für die Berechnung von α wurden zwei Methoden verwendet. Das Minimierungsproblem wurde einmal mit der in Matlab vorimplementierten Funktion `fminbnd` gelöst (α_{fm}). Bei der anderen Methode wurde ein Schritt des eindimensionalen Newton-Verfahrens durchgeführt (α_N). Hauptproblem der zweiten Variante war die deutlich längere Laufzeit, weil zusätzlich auch die Hessematrix $\mathrm{H}_f(x) = (\frac{\partial^2}{\partial x_i \partial x_j} f)$ berechnet werden muss.
Bei dem Optimierungsproblem (3.1) soll das Maximum der Funktion $\ln(L_D(l))$ bestimmt werden. Da es sich bei den Verfahren um Minimierungsmethoden handelt, wurde als Zielfunktion f für die Projektionsverfahren

$$f := -\ln(L_D(l)) \tag{3.13}$$

verwendet. Aufgrund der Nebenbedingungen, wurde $G := \mathbb{R}^+$ gewählt.
Als Ausgangstopologie und Startnähererung für l wurden die Ergebnisse des Neighbor-Joining-Algorithmus verwendet. Um die Algorithmen durchzuführen, werden neben den Werten für W_k auch die zugehörigen Gradienten $\nabla(-ln(L_D(l)))$ und für das Newton-Verfahren zusätzlich noch die Hessematrizen $H_{-ln(L_D)}(l)$ benötigt. Die Berechnung von Gradienten und Hessematrizen erfolgt, ebenso wie die Berechnung der Werte für W_k, rekursiv.
Für die $2n-1$ Knoten gilt folgende Notation: Die Knoten 1 bis n sind die äußeren, die folgenden $n-1$ Knoten sind die inneren. $B := \{\mathrm{A,C,G,T}\}$ sei die Menge der Basen.
Für $k = 1...n \quad x \in B$ gilt:

$$W_k(x) = \begin{cases} 1 & \text{falls am Knoten k die Base x steht} \\ 0 & \text{falls am Knoten k nicht die Base x steht} \end{cases}$$

außerdem sei

$$\frac{\partial}{\partial l_p} W_k(x) = 0 \quad \forall p \; 1 \le p \le 2n-2$$

und

$$\frac{\partial^2}{\partial l_p \partial l_q} W_k(x) = 0 \quad \forall p \; 1 \le p \le 2n-2 \text{ und } \forall q \; 1 \le q \le 2n-2$$

Für $k = n+1...2n-1 \quad x \in B$ gilt:

$$W_k(x) = W1(x) \cdot W2(x) \quad \text{mit}$$

$$W1(x) := P_{x \to x}(l_i) W_i(x) + P_{x \to y}(l_i) \sum_{y \in B \setminus \{x\}} (W_i(y)) \quad \text{und}$$

$$W2(x) := P_{x \to x}(l_j) W_j(x) + P_{x \to y}(l_j) \sum_{y \in B \setminus \{x\}} (W_j(y)).$$

Für den Gradienten ergibt sich

$$\begin{aligned}
\frac{\partial}{\partial l_p}W_k(x) &= \frac{\partial}{\partial l_p}W1(x)\cdot W2(x) + W1(x)\cdot\frac{\partial}{\partial l_p}W2(x) \text{ mit}\\
\frac{\partial}{\partial l_p}W1(x) &= \frac{\partial}{\partial l_p}(P_{x\to x}(l_i))W_i(x) + P_{x\to x}(l_i)\frac{\partial}{\partial l_p}W_i(x)+\\
&\quad \frac{\partial}{\partial l_p}P_{x\to y}(l_i)\sum_{y\in B\setminus\{x\}}(W_i(y)) + P_{x\to y}(l_i)\sum_{y\in B\setminus\{x\}}(\frac{\partial}{\partial l_p}W_i(y))
\end{aligned}$$

und

$$\begin{aligned}
\frac{\partial}{\partial l_p}W2(x) &= \frac{\partial}{\partial l_p}(P_{x\to x}(l_j))W_j(x) + P_{x\to x}(l_j)\frac{\partial}{\partial l_p}W_j(x)+\\
&\quad \frac{\partial}{\partial l_p}P_{x\to y}(l_j)\sum_{y\in B\setminus\{x\}}(W_j(y)) + P_{x\to y}(l_j)\sum_{y\in B\setminus\{x\}}(\frac{\partial}{\partial l_p}W_j(y)).
\end{aligned}$$

Die Hessematrix berechnet sich aus der folgenden Rekursion:

$$\begin{aligned}
\frac{\partial^2}{\partial l_p\partial l_q}W_k(x) &= \frac{\partial^2}{\partial l_p\partial l_q}W1(x)\cdot W2(x) + \frac{\partial}{\partial l_p}W1(x)\cdot\frac{\partial}{\partial l_q}W2(x)+\\
&\quad \frac{\partial}{\partial l_q}W1(x)\cdot\frac{\partial}{\partial l_p}W2(x) + W1(x)\cdot\frac{\partial^2}{\partial l_p\partial l_q}W2(x) \text{ mit}\\
\frac{\partial^2}{\partial l_p\partial l_q}W1(x) &= \frac{\partial^2}{\partial l_p\partial l_q}(P_{x\to x}(l_i))W_i(x) + \frac{\partial}{\partial l_p}(P_{x\to x}(l_i)) + \frac{\partial}{\partial l_q}W_i(x)+\\
&\quad \frac{\partial}{\partial l_q}P_{x\to x}(l_i)\frac{\partial}{\partial l_p}W_i(x) + P_{x\to x}(l_i)\frac{\partial^2}{\partial l_p\partial l_q}W_i(x)+\\
&\quad \frac{\partial^2}{\partial l_p\partial l_q}P_{x\to y}(l_i)\sum_{y\in B\setminus\{x\}}(W_i(y)) + \frac{\partial}{\partial l_p}P_{x\to y}(l_i)\sum_{y\in B\setminus\{x\}}(\frac{\partial}{\partial l_q}W_i(y))+\\
&\quad \frac{\partial}{\partial l_p}P_{x\to y}(l_i)\sum_{y\in B\setminus\{x\}}(\frac{\partial}{\partial l_p}W_i(y)) + P_{x\to y}(l_i)\sum_{y\in B\setminus\{x\}}(\frac{\partial^2}{\partial l_p\partial l_q}W_i(y))
\end{aligned}$$

und

$$\begin{aligned}
\frac{\partial^2}{\partial l_p\partial l_q}W2(x) &= \frac{\partial^2}{\partial l_p\partial l_q}(P_{x\to x}(l_j))W_j(x) + \frac{\partial}{\partial l_p}(P_{x\to x}(l_j)) + \frac{\partial}{\partial l_q}W_j(x)+\\
&\quad \frac{\partial}{\partial l_q}P_{x\to x}(l_j)\frac{\partial}{\partial l_p}W_j(x) + P_{x\to x}(l_j)\frac{\partial^2}{\partial l_p\partial l_q}W_j(x)+\\
&\quad \frac{\partial^2}{\partial l_p\partial l_q}P_{x\to y}(l_i)\sum_{y\in B\setminus\{x\}}(W_j(y)) + \frac{\partial}{\partial l_p}P_{x\to y}(l_i)\sum_{y\in B\setminus\{x\}}(\frac{\partial}{\partial l_q}W_j(y))+\\
&\quad \frac{\partial}{\partial l_p}P_{x\to y}(l_j)\sum_{y\in B\setminus\{x\}}(\frac{\partial}{\partial l_p}W_j(y)) + P_{x\to y}(l_j)\sum_{y\in B\setminus\{x\}}(\frac{\partial^2}{\partial l_p\partial l_q}W_j(y)).
\end{aligned}$$

3.1.5 Eigenschaften der Likelihood-Funktion

Eine Eigenschaft der Likelihood-Funktion ist, dass sie nur von der Summe der beiden Astlängen, die von der Wurzel ausgehen, abhängig ist. Aufgrund dieser Eigenschaft wird für die Optimierung nur eine der beiden von der Wurzel ausgehenden Äste mit einbezogen. Zur Berechnung des Maximum-Likelihood-Wertes wird der Wert der fehlenden Astlänge, dann dem der optimierten, gleichgesetzt.

Bemerkung. Sei θ ein Baum mit der Wurzel r. i und j seien die direkten Nachfolger von r. Bei dem Baum θ_Σ seien die Astlängen l_i und l_j jeweils durch $(l_i+l_j)/2$ ersetzt dann gilt: $L_{D^{(s)}}(\theta) = L_{D^{(s)}}(\theta_\Sigma)$

Denn für θ gilt:

$$\begin{aligned}
W_r(x) &= [\sum_{y\in B} P_{x\to y}(l_i)\cdot W_i(y)]\cdot[\sum_{z\in B} P_{x\to z}(l_j)\cdot W_j(z)] \\
&= [(\tfrac{1}{4}+\tfrac{3}{4}e^{-l_i})W_i(x)+(\tfrac{1}{4}-\tfrac{1}{4}e^{-l_i})(\sum_{y\in B\setminus\{x\}} W_i(x))]\cdot \\
&\quad [(\tfrac{1}{4}+\tfrac{3}{4}e^{-l_j})W_j(x)+(\tfrac{1}{4}-\tfrac{1}{4}e^{-l_j})(\sum_{y\in B\setminus\{x\}} W_j(x))] \\
&= \tfrac{1}{16}\cdot[(1+3e^{-l_i}+3e^{-l_j}+9e^{-(l_i+l_j)})W_i(x)W_j(x)+ \\
&\quad (1+3e^{-l_i}-e^{-l_j}-3e^{-(l_i+l_j)})W_i(x)(\sum_{z\in B\setminus\{x\}} W_j(z))+ \\
&\quad (1-e^{-l_i}+3e^{-l_j}-3e^{-(l_i+l_j)})W_j(x)(\sum_{y\in B\setminus\{x\}} W_i(y))+ \\
&\quad (1-e^{-l_i}-e^{-l_j}+e^{-(l_i+l_j)})(\sum_{y\in B\setminus\{x\}} W_i(y))(\sum_{z\in B\setminus\{x\}} W_j(z))].
\end{aligned}$$

Daraus folgt für $L_{D^{(s)}}(\theta)$:

$$\begin{aligned}
L_{D^{(s)}}(\theta) &= \sum_{x\in B}\tfrac{1}{4}\cdot W_r(x) \\
&= \tfrac{1}{16}\cdot[(\sum_{x\in B} W_i(x)\cdot W_j(x)(1+3e^{-(l_i+l_j)})) \\
&\quad +(\sum_{x\in B}\sum_{\substack{y\in B\\ x\neq y}} W_i(x)\cdot W_j(y)(1-e^{-(l_i+l_j)}))].
\end{aligned}$$

Für θ_Σ gilt:

$$\begin{aligned}
W_r(x) &= [\sum_{y\in B} P_{x\to y}(\tfrac{l_i+l_j}{2})\cdot W_i(y)]\cdot[\sum_{z\in B} P_{x\to z}(\tfrac{l_i+l_j}{2})\cdot W_j(z)] \\
&= [(\tfrac{1}{4}+\tfrac{3}{4}e^{-\frac{l_i+l_j}{2}})W_i(x)+(\tfrac{1}{4}-\tfrac{1}{4}e^{-\frac{l_i+l_j}{2}})(\sum_{y\in B\setminus\{x\}} W_i(x))]\cdot \\
&\quad [(\tfrac{1}{4}+\tfrac{3}{4}e^{-\frac{l_i+l_j}{2}})W_j(x)+(\tfrac{1}{4}-\tfrac{1}{4}e^{-\frac{l_i+l_j}{2}})(\sum_{y\in B\setminus\{x\}} W_j(x))] \\
&= \tfrac{1}{16}\cdot[(1+3e^{-\frac{l_i+l_j}{2}}+3e^{-\frac{l_i+l_j}{2}}+9e^{-(l_i+l_j)})W_i(x)W_j(x)+ \\
&\quad (1+3e^{\frac{-l_i+l_j}{2}}-e^{-\frac{l_i+l_j}{2}}-3e^{-(l_i+l_j)})W_i(x)(\sum_{z\in B\setminus\{x\}} W_j(z))+ \\
&\quad (1-e^{\frac{-l_i+l_j}{2}}+3e^{-\frac{l_i+l_j}{2}}-3e^{-(l_i+l_j)})W_j(x)(\sum_{y\in B\setminus\{x\}} W_i(y))+ \\
&\quad (1-e^{-\frac{l_i+l_j}{2}}-e^{-\frac{l_i+l_j}{2}}+e^{-(l_i+l_j)})(\sum_{y\in B\setminus\{x\}} W_i(y))(\sum_{z\in B\setminus\{x\}} W_j(z))].
\end{aligned}$$

Für $L_{D^{(s)}}(\theta_\Sigma)$ gilt daher:

$$\begin{aligned}
L_{D^{(s)}}(\theta_\Sigma) &= \sum_{x\in B}\tfrac{1}{4}\cdot W_r(x) \\
&= \tfrac{1}{16}\cdot[(\sum_{x\in B} W_i(x)\cdot W_j(x)(1+3e^{-(l_i+l_j)})) \\
&\quad +(\sum_{x\in B}\sum_{\substack{y\in B\\ x\neq y}} W_i(x)\cdot W_j(x)(1-e^{-(l_i+l_j)}))].
\end{aligned}$$

Insgesamt folgt also

$$L_{D^{(s)}}(\theta) = L_{D^{(s)}}(\theta_\Sigma).$$

Konvexität

Es ist nicht trivial zu zeigen, dass die Likelihood-Funktion konvex ist. Für $n = 2$ Sequenzen ergibt sich das Folgende: Zur Vereinfachung sei $B_1 := D^{(s)}(1)$ und $B_2 := D^{(s)}(2)$.

1. Fall $B_1 = B_2$

$$W_3(x) = \begin{cases} (\frac{1}{4} + \frac{3}{4}e^{-l_1})(\frac{1}{4} + \frac{3}{4}e^{-l_2}) & \text{falls } x = B_1 \\ (\frac{1}{4} - \frac{1}{4}e^{-l_1})(\frac{1}{4} - \frac{1}{4}e^{-l_2}) & \text{falls } x \neq B_1 \end{cases}$$

$$\Rightarrow L_{D^{(s)}}(\theta) = \frac{1}{16} + \frac{3}{16}e^{-(l_1+l_2)} \tag{3.14}$$

2. Fall $B_1 \neq B_2$

$$W_3(x) = \begin{cases} (\frac{1}{4} + \frac{3}{4}e^{-l_1})(\frac{1}{4} - \frac{1}{4}e^{-l_2}) & \text{falls } x = B_1 \\ (\frac{1}{4} - \frac{1}{4}e^{-l_1})(\frac{1}{4} + \frac{3}{4}e^{-l_2}) & \text{falls } x = B_2 \\ (\frac{1}{4} - \frac{1}{4}e^{-l_1})(\frac{1}{4} - \frac{1}{4}e^{-l_2}) & \text{falls } x \neq B_1, x \neq B_2 \end{cases}$$

$$\Rightarrow L_{D^{(s)}}(\theta) = \frac{1}{16} - \frac{1}{16}e^{-(l_1+l_2)} \tag{3.15}$$

Mit $x := l_1 + l_2$, $M := |I|$ mit $I := \{s|D^{(s)}(1) = D^{(s)}(2)\}$ und $N := |U|$ mit $U := \{s|D^{(s)}(1) \neq D^{(s)}(2)\}$ folgt aus Gleichung (3.14) und (3.15)

$$L_D(\theta) = (\frac{1}{16}(1 + 3e^{-x}))^M(\frac{1}{16}(1 - e^{-x}))^N \tag{3.16}$$

Wie bei den Optimierungsverfahren wird auch hier der negative Logarithmus von (3.16) betrachtet

$$-ln(L_D(\theta)) = -(M + N) \cdot ln\frac{1}{16} + M \cdot ln(1 + 3e^{-x}) + N \cdot ln(1 - e^{-x}) \tag{3.17}$$

Die erste und zweite Ableitung von (3.17) sind

$$-ln(L_D(\theta))' = M\frac{3e^{-x}}{1 + 3e^{-x}} - N\frac{e^{-x}}{1 - e^{-x}} \tag{3.18}$$

und

$$-ln(L_D(\theta))'' = -3M\frac{e^{-x}}{(1 + 3e^{-x})^2} + N\frac{e^{-x}}{(1 - e^{-x})^2} \tag{3.19}$$

Aus 3.18 folgt, dass der einzige Extremwert bei

$$x = -ln(\frac{3M - N}{3M + 3N})$$

liegt. Daher existiert nur ein Maximum, wenn $3M > N$.
$3M > N \Leftrightarrow 3M = N + y$, $y > 0$. Für die Konvexität folgt dann

$$\begin{aligned} -ln(L_D(\theta))'' > 0 \quad &\Leftrightarrow \quad N > 3\tfrac{(1-e^{-x})^2}{(1+3e^{-x})^2} M \\ &\Leftrightarrow \quad N > \tfrac{(1-e^{-x})^2}{(1+3e^{-x})^2-(1-e^{-x})^2} y \end{aligned} \tag{3.20}$$

Die Funktion

$$F(x) := \frac{(1-e^{-x})^2}{(1+3e^{-x})^2-(1-e^{-x})^2}$$

ist monoton wachsend, weil

$$F(x)' = 2e^{-x}(1-e^{-x})(1+3e^{-x})((1+3e^{-x})+3(1-e^{-x})) > 0.$$

Daher ist die Bedingung aus Gleichung 3.20 abhängig von y für genügend große x nicht mehr erfüllt. Die Funktion $-ln(L_D(\theta))$ ist also nur auf einem Teil des Definitionsbereiches konvex.
Es läßt sich vermuten, dass die Konvexität zum einen von der Beschaffenheit der Sequenzen abhängt, die Funktion dann aber auch nur auf einem Teil des Definitionsbereichs konvex ist. Um eine Konvergenz gegen ein Maximum zu erreichen, muss die Startnäherung für l in dem konvexen Teil des Definitionsbereichs liegen.
Eine Methode numerisch zu zeigen, ob eine gefundene Lösung ein lokales Maximum ist und ob zumindestens in einer Umgebung der Lösung Konvexität vorliegt, ist die Berechnung der Eigenwerte der Hessematrix im letzten Iterationsschritt. Sind alle größer als Null, die Hessematrix also positiv definit, so ist die Funktion in einer Umgebung der Lösung konvex und die Lösung ein strenges lokales Maximum. Ist die Hessematrix dagegen indefinit, ergeben sich also sowohl positive als auch negative Eigenwerte, ist die gefundene Lösung kein lokales Maximum und die Funktion ist in einer Umgebung der Lösung nicht konvex.

3.1.6 Startnäherung l der Astlängen optimieren

Mit einer verbesserten Starttopologie sollte eine schnellere Konvergenz der Verfahren erreicht werden. Die Startnäherung der Astlängen l wird durch das Lösen von Teilproblemen optimiert. Hierfür wird jedem Knoten i zunächst ein Knotenlevel ($\mathrm{KL}(i)$) zugeordnet. Für alle äußeren Knoten gilt $\mathrm{KL}(i) = 1$. Für einen inneren Knoten i mit den direkten Nachfolgern i_1 und i_2 gilt:

$$\mathrm{KL}(i) = n,\ n \in \mathbb{N}_+ \Leftrightarrow \max\{\mathrm{KL}(i_1), \mathrm{KL}(i_2)\} = n-1.$$

$l_{\mathrm{NJ}} \in \mathbb{R}^{2n-2}$ seien die Astlängen des Neighbor-Joining-Baums T. T_i sei der Teilbaum von T mit Wurzel i. D_i die zu T_i gehörenden Sequenzen. $L_D(T, l)$ sei der Likelihood-Wert der Topologie T mit den Astlängen l für die Sequenzdaten D.

Zur Vereinfachung seien $l^* = \Phi(T, l_s, D, \text{step})$ die Astlängen der Lösung des Optimierungsproblems

$$\max_{l \in \mathbb{R}^{2n-2}} \ln(L_D(T,l)) \text{ unter der Nebenbedingung}$$
$$l_i \geq 0 \ \forall \ 1 \leq i \leq 2n-2$$

mit Startnäherung l_s für die Astlängen und dem Abbruchkriterium $\|\nabla(f(x^{(k)})\| \leq \varepsilon \cdot \frac{|f(x)|}{\overline{x}}$, $\varepsilon > 0$ und $it <$ step.
Für die Optimierung ergibt sich folgender Algorithmus:

Algorithmus 3.8. (Optimierung von l)
$l = l_{\text{NJ}}$
`for` i = 1 to max(KL)
    $l^* = \Phi(T_i, l, D_i, 1)$
    $l = l^*$
`end`

Zur Lösung der Teilprobleme wird das Verfahren verwendet, mit dem auch das vollständige Problem im Anschluss gelöst wurde. Für die Verfahren mit der Reduktionsbedingung wurde für die Teilprobleme $\gamma = 10^{-3}$ gesetzt.

3.2 Berechnung der optimalen Topologie

Die Suche nach der optimalen Topologie ist ein diskretes Optimierungsproblem. Im Gegensatz zu einem kontinuierlichen Optimierungsproblem gibt es nur endlich viele Lösungsmöglichkeiten. Für n Sequenzen sind dies

$$N_U := \frac{(2n-5)!}{2^{n-3}(n-3)!} \quad n \geq 3. \tag{3.21}$$

mögliche Topologien. Da es bei dieser Anzahl nicht sinnvoll ist, alle auszuprobieren, wird zur Lösung des Optimierungsproblems ein heuristischer Ansatz verwendet. Hier bietet sich ein iterativer Algorithmus, der ausgehend von einer Starttopologie in jedem Schritt kleine lokale Veränderungen vornimmt, an. Liefert eine Änderung in der Topologie einen größeren Likelihood-Wert, so wird sie beibehalten.

Für einen Knoten k wird der folgende Knotentausch durchgeführt:

T sei die aktuelle Topologie, g der Vorgänger von k, d.h. $T(g,1) = k$ oder $T(g,2) = k$.

Die Nachfolger von k seien i und j, also $T(k,1) = i$ und $T(k,2) = j$.

Für den Fall $T(g,1) = k$ (und $T(g,2) = h$) ergeben sich als neue Topologien:

$T_{neu1} = T$ mit den Änderungen:
$T_{neu1}(g,1) = k$, $T_{neu1}(g,2) = j$, $T_{neu1}(k,1) = i$ und $T_{neu1}(k,2) = h$ und

$T_{neu2} = T$ mit den Änderungen:
$T_{neu2}(g,1) = k$, $T_{neu2}(g,2) = i$, $T_{neu2}(k,1) = h$ und $T_{neu2}(k,2) = j$.

Für den Fall $T(g,2) = k$ (und $T(g,1) = m$) ergeben sich als neue Topologien:
$T_{neu1} = T$ mit den Änderungen:
$T_{neu1}(g,1) = j$, $T_{neu1}(g,2) = k$, $T_{neu1}(k,1) = i$ und $T_{neu1}(k,2) = h$ und
$T_{neu2} = T$ mit den Änderungen:
$T_{neu2}(g,1) = i$, $T_{neu2}(g,2) = k$, $T_{neu2}(k,1) = h$ und $T_{neu2}(k,2) = j$.

Zur Berechnung des Likelihood-Wertes der geänderten Topologie benötigt man die optimalen Astlängen für die Topologie. Nur so ist eine Aussage, ob die geänderte Topologie einen größeren Likelihood-Wert hat, möglich. Die Berechnung des phylogenetischen Baums ergibt sich durch die Einbettung des kontinuierlichen Optimierungsproblems in das diskrete.
Insgesamt ergibt sich der folgenden Algorithmus zur Berechnung des maximalen Likelihood-Wertes L^*, und der zugehörigen Topologie T^* mit den Astlängen l^*.

Algorithmus 3.9.

$l_{neu} := \Phi(T_{NJ}, l_{NJ}, D, \mathrm{step}_{\mathrm{begin}})$;
$\mathrm{L}_{neu} := \ln(L_D(T, l_{neu}))$;
$\mathrm{L}_{\mathrm{alt}} := 0$; $l := l_{neu}$;
`while` $\mathrm{L}_{\mathrm{alt}} \neq \mathrm{L}_{\mathrm{neu}}$
    $\mathrm{L}_{\mathrm{alt}} = \mathrm{L}_{\mathrm{neu}}$;
    for i = n+1 to 2n-1
        Berechne durch Knotentausch am Knoten i T_{neu1} und T_{neu2};
        $l_{neu1} := \Phi(T_{\mathrm{neu1}}, l, D, \mathrm{step})$; $l_{neu2} := \Phi(T_{\mathrm{neu2}}, l, D, \mathrm{step})$;
        $\mathrm{L}_{\delta 1} := \ln(L_D(T_{\mathrm{neu1}}, l_{neu1}))$; $\mathrm{L}_{\delta 2} := \ln(L_D(T_{\mathrm{neu2}}, l_{neu}))$;
        if $\mathrm{L}_{\delta 2} \leq \mathrm{L}_{\delta 1}$ and $\mathrm{L}_{\mathrm{neu}} \leq \mathrm{L}_{\delta 1}$ then
            $\mathrm{L}_{\mathrm{neu}} := \mathrm{L}_{\delta 1}$;
            $T := T_{\mathrm{neu1}}$;
            $l := l_{\mathrm{neu1}}$;
        `end`
        if $\mathrm{L}_{\delta 1} \leq \mathrm{L}_{\delta 2}$ and $\mathrm{L}_{\mathrm{neu}} \leq \mathrm{L}_{\delta 2}$ then
            $\mathrm{L}_{\mathrm{neu}} := \mathrm{L}_{\delta 2}$;
            $T := T_{\mathrm{neu2}}$;
            $l := l_{\mathrm{neu2}}$;
        `end`
    `end`

`end`

$l^* := \Phi(T, l, D, \text{step}_{\text{final}})$; $T^* = T$;

$\text{L}^* := \ln(L_D(T^*, l^*))$;

3.3 Anwendungsbeispiele

Die Algorithmen wurden an einem Modellproblem aus fünf Sequenzen und drei Beispielproblemen aus der Virologie getestet. Im Vordergrund stand dabei eine Aussage über die Effizienz der Algorithmen bezüglich Laufzeit und Konvergenz und ob die Veränderung der Neighbor-Joining Topologie zu größeren Likelihood-Werten führt. Außerdem war die Konvexität in der Umgebung der Lösung von Interesse.

3.3.1 Mathematisches Modellproblem

Zur Veranschaulichung des Likelihood-Algorithmus wurde ein Modellproblem, das nur aus $n = 5$ Sequenzen besteht, verwendet. Für $n = 5$ existieren 15 mögliche Topologien. Jede dieser Topologien kann man in Form sieben verschiedener *rooted trees* darstellen, die alle den gleichen Likelihood-Wert und identische Astlängen haben. Bei den hier verwendeten Sequenzen D_5 waren 39,7% der Basen konserviert.
Als Topologie T_{NJ} und Startwert für die Astlängen l_{NJ} wurden die Ergebnisse des Neighbor-Joining Algorithmus verwendet. Der Likelihood-Wert für diese Ausgangswerte war $\ln(L_{D_5}(l_{NJ})) = -1.3587306126\ 10^3$.
Die Algorithmen (3.4) bis (3.7) mit und ohne Optimierung von l konvergierten alle gegen den gleichen Likelihood-Wert: $\ln(L_D(l^*) = -1.338828\ 10^3$, wobei l^* die Astlängen der Lösung des Optimierungsproblems, das mit dem jeweiligen Algorithmus gelöst wird, sind. Die Hessematrix nach dem letzten Iterationsschritt hatte nur positive Eigenwerte, die Likelihood-Funktion für das Modellproblem ist daher in der Umgebung der Lösung konvex. Die kürzeste Laufzeit hatte der Algorithmus $\text{FR}_{\text{redukt}}$ (Startwert für $\gamma = 10^{-4}$) ohne Optimierung von l.
Mit diesem Algorithmus wurde für die 15 möglichen Topologien der Likelihood-Wert berechnet. Für jede Topologie ergab sich ein anderer Likelihood-Wert. Dargestellt sind diese Topologien zusammen mit dem zugehörigen Likelihood-Wert in den Abbildungen (3.1; Seite 80) und (3.2; Seite 81). Die Topologie mit dem größten Wert war mit der Topologie identisch, die der Neighbor-Joining Algorithmus ergab.

Konvexität

Bei dem Fall $n = 2$ sind zwei Faktoren für die Konvexität entscheidend, der Startvektor und die Sequenzdaten. Um diese Faktoren für den Fall $n = 5$ zu überprüfen, wurde zum einen ein anderer Startvektor für l gewählt und zum anderen wurden die Sequenzen verändert. Um eine Aussage über die lokale Konvexität treffen zu können, wurden die Eigenwerte nach dem letzte Iterationsschritt berechnet.

$l^* = \Phi(T_{NJ}, l_s, D_5, 10000)$ seien die Astlängen nach der Optimierung mit dem Algorithmus $\mathrm{FR}_{\mathrm{redukt}}$ (Startwert für $\gamma = 10^{-4}$) mit l_s als Startnäherung für l.
Betrachtet man die oben beschriebenen Sequenzen und wählt für die Topologie T_{NJ} als Startvektor $l_s := l_{NJ}$ so ergab sich $\ln(L_D(l^*)) = -1.338828703\ 10^3$ und alle Eigenwerte der Hessematrix waren positiv. Die Funktion ist daher in einer Umgebung der Lösung konvex.
Bei einem Startvektor $l_s := l_1 = [1, 1, 1, 1, 1, 1, 1, 1]$ ergab sich ein Likelihood-Wert von $\ln(L_D(l^*)) = -1.3459\ 10^3$ ($\|\nabla(\ln(L_D(l^*))\| = 4.988\ 10^{-3}$) und die Hessematrix war ebenfalls positiv definit. Auch hier ist die Funktion in einer Umgebung der Lösung konvex.
Wählt man als Startvektor aber $l_s := l_{1.5} = [1.5, 1.5, 1.5, 1.5, 1.5, 1.5, 1.5, 1.5]$, so hat man nach 10000 Iterationsschritten ein Likelihood-Wert von nur
$\ln(L_D(l^*)) = -1.39701\ 10^3$ ($\|\nabla(\ln(L_D(l^*))\| = 1.494\ 10^{-1}$). Bei der Berechnung der Eigenwerte ergab sich neben den positiven mit -31 auch ein negativer Eigenwert. Daher ist l^* kein lokales Minimum und die lokale Konvexität ist nicht mehr vorhanden. Betrachtet man den Eigenvektor zu dem negativen Eigenwert, so ist kein Zusammenhang zwischen den Einträgen und den aktiven Nebenbedingungen erkennbar. Die Konvexität geht also nicht nur am Rand verloren. Auch der erreichte Likelihood-Wert war kleiner als der Ausgangswert.
Für die Astlängen l^* ergaben sich nach der Optimierung mit den verschiedenen Startvektoren die folgenden Werte:

$$
\begin{array}{lllllllll}
l_{NJ} & = [0.1860, & 0.3148, & 0.0946, & 0.0798, & 0.1750, & 0.0762, & 0.0713, & 0.1860\ ] \\
l^*_{l_s=l_{NJ}} & = [0.2394, & 0.4575, & 0.1414, & 0.0931, & 0.2252, & 0.1296, & 0.2157, & 0.2394\ ] \\
l^*_{l_s=l_1} & = [0.2415, & 0.4545, & 0.1434, & 0.0998, & 0.3485, & 0, & 0.2913, & 0.2415\ ] \\
l^*_{l_s=l_{1.5}} & = [0, & 0.9096, & 0.2323, & 0, & 0, & 0.4320, & 0.8167, & 0\ ]
\end{array}
$$

Für $l_s = l_{NJ}$ und $l_s = l_1$ ergab die Berechnung der Eigenwerte, dass es sich bei beiden Lösungen um ein lokales Maximum handelt. Die unterschiedlichen Likelihood-Werte zeigen aber, dass eine Veränderung des Startvektors zur Konvergenz gegen ein anderes lokales Maximum, mit einem kleineren Wert, führt. Nur für $l_s = l_{NJ}$ wurde also ein tatsächliches Maximum gefunden. Während die Astlängen für $l_s = l_{NJ}$ und $l_s = l_1$ sehr ähnlich waren, unterschieden sich die für $l_s = l_{1.5}$ von den anderen beiden.
Um zu überprüfen, welchen Einfluss die Sequenzdaten auf die Konvexität haben, wurden Sequenzen verwendet, bei denen 59,3% bzw. nur 20% der Basen konserviert waren. Als Topologie und Startwert für l wurden auch hier die Ergebnisse des Neighbor-Joinings verwendet. In beiden Fällen ergaben sich für die Hessematrix nur positive Eigenwerte. Auch diese beiden Beispiele sind demnach, zumindestens in einer Umgebung der erhaltenen Lösung, konvex.

3.3.2 Beispiele aus der Virologie

Die Algorithmen zur Berechnung phylogenetischer Bäume wurden an den drei Beispielen CSFV-NTR, BVDV-E2 und BVDV-Npro getestet. Die Beispiele beziehen

sich auf die virologischen Anwendungen in den nächsten beiden Kapiteln. Es wurden bei den Beispielen verschieden viele und verschieden lange Sequenzen ausgewählt. Als Starttopologie wurden für alle drei Beispiele Neighbor-Joining Bäume berechnet. Diese sind in den Abbildungen 3.3(Seite 82), 3.5(Seite 84) und 3.7(Seite 86) dargestellt. Die Zahlen hinter den Namen stehen für eine Einteilung in Gruppen. Bei den Gruppen kann es sich um Genotypen oder Subgruppen handeln. Wenn die Ergebnisse zeigen, dass eine veränderte Topologie zu einem größeren Likelihood-Wert führt, ist es für die Taxonomie von Interesse, ob diese Gruppen trotzdem erhalten bleiben.

Beispielproblem CSFV-NTR:

Für dieses Beispiel wurden 18 CSFV Sequenzen verwendet. Die Sequenzen sind 150 Basen lang und aus der 5'NTR. Alle Sequenzen stammen aus der in Kapitel 4 beschriebenen CSFV-Datenbank. Die Darstellung des phylogenetischen Baums erfolgte als *rooted tree* mit der *Outgroup* Kanagawa (CSF0309). Einteilen lassen sich diese Sequenzen in die Genotypen 1 und 2 mit zwei bzw. drei Subgruppen. Bei den Sequenzen waren 75,4% der Basen konserviert.

Beispielproblem BVDV-E2:

Hier wurden 44 Sequenzen von Virusisolaten der Spezies BVDV und BDV mit einer Länge von 270 Basen verwendet. Die Sequenzen sind aus dem E2 Gen. Alle Sequenzen stammen aus der in Kapitel 5 beschriebenen BVDV/BDV-Datenbank. Die Darstellung des phylogenetischen Baums erfolgte als *rooted tree* mit der *Outgroup* Giraffe (PES0028). Unterschieden werden die Spezies BVDV-1, BVDV-2 und BDV, die jeweils in Gruppen unterteilt werden. Bei den Sequenzen waren 23% der Basen konserviert.

Beispielproblem BVDV-Npro:

Für das letzte Beispiel wurden 53 Sequenzen, ebenfalls von Virusisolaten der Spezies BVDV und BDV, verwendet. Die Sequenzen waren 390 Basen lang und aus dem Npro Gen. Alle Sequenzen stammen ebenfalls aus der in Kapitel 5 beschriebenen BVDV/BDV-Datenbank, und die Darstellung des phylogenetischen Baums erfolgte als *rooted tree* mit der *Outgroup* Giraffe (PES0028). Auch in diesem Beispiel gilt die Einteilung in die Spezies BVDV-1, BVDV-2 und BDV und in die jeweiligen Gruppen. Bei den Sequenzen waren 36% der Basen konserviert.

Auswertung des kontinuierlichen Optimierungsproblems

Zur Lösung des kontinuierlichen Optimierungsproblems existieren mehrere Algorithmen. Um eine Aussage über die Effizienz dieser Algorithmen machen zu können, wurden alle Verfahren an den drei Beispielen CSFV-NTR, BVDV-E2 und BVDV-Npro getestet. Als Topologie diente die der Neighbor-Joining Bäume. l^* sind wieder

die Astlängen der Lösung des Optimierungsproblems mit Startnäherung l_{NJ} für die Astlängen. Als Abbruchkriterium wurde

$$it < 100 \text{ und } \|\nabla(L_D(l^*)\| \leq 10^{-6} \cdot \frac{|\ln(L_D(l_{NJ}))|}{\overline{l_{NJ}}}$$

gesetzt.
Ausgewertet wurde neben dem Likelihood-Wert $\ln(L_D(l^*))$ und dem Gradienten $\|\nabla(L_D(l^*)\|$, auch die, für die durchgeführten Iterationsschritte (it) benötigte Zeit (time) in Sekunden.
Die Ergebnisse sind in den Tabellen 3.1 bis 3.6 dargestellt. Angegeben wurden jeweils die Werte des Schrittes, nachdem das Abbruchkriterium erfüllt war.
Um nachzuweisen, ob es sich bei den Lösungen der drei Problemen um lokale Maxima handelt, wurden bei dem Verfahren mit dem größten und kleinsten Likelihood-Wert die Eigenwerte der Hessematrix im letzten Iterationschritt berechnet. Mit Hilfe dieser Eigenwerte kann auch gezeigt werden, ob die Funktion in einer Umgebung der erhaltenen Lösung konvex ist.
Zusammenfassend ergeben sich die folgenden Ergebnisse:

Beispiel CSFV-NTR: Der Ausgangsbaum des Neighbor-Joining-Algorithmus ergab einen Likelihood-Wert $\ln(L_D(l_{NJ})) = -6.028570\ 10^2$.
Der größte Likelihood-Wert nach Optimierung wurde bei den Verfahren FR_{Min} und und FR_{Min}(Newton) mit $L_D(l^*) = -5.9474396202\ 10^2$ erreicht. Die Laufzeit betrug 6.4210 beziehungsweise 567.4220 Sekunden.
Die kürzeste Laufzeit hatte das Verfahren FR_{Min} mit 3.2820 sek. Der zugehörige Likelihood-Wert ist mit $L_D(l^*) = -5.9474396207\ 10^2$ nur geringfügig kleiner als der maximal erreichte.
Der Unterschied zwischen den erreichten Likelihood-Werten ist relativ groß. Die Differenz zwischen dem kleinsten und dem größten Wert ist 7.09 (Tab 3.1 und 3.2).
Bei der Berechnung der Eigenwerte der Hessematrix, ergaben beim Verfahren FR_{Min}(Newton), das den größten Likelihood-Wert lieferte, neben fünf negative Eigenwerte zwischen -28.86 und -27.97, positive Eigenwerte zwischen 986.76 und 18595.
Die Berechnung der Eigenwerte des Verfahrens mit dem kleinsten Likelihood-Wert ergab ebenfalls fünf negative Eigenwerte zwischen -748.94 und -21.40, die übrigen Eigenwerte lagen zwischen 778.78 und $1.87\ 10^8$.
Die Eigenwerte zeigen, dass beide Lösungen keine lokalen Maxima sind und die Likelihood-Funktion in einer Umgebung der Lösung nicht konvex ist. Auch zeigen sie, wie schlecht die Hessematrix in diesem Fall konditioniert ist. Betrachtet man die Eigenvektoren (EV) zu den negativen Eigenwerten, ergibt sich beim Verfahren FR_{Min}(Newton) ein Zusammenhang zu den aktiven Nebenbedingungen: Für alle j für die $\text{EV}(j) > 0.01$ ist, ist die Nebenbedingung für j aktiv, dh. $l_j = 0$. Bei dem Verfahren FR_{Min} mit Optimierung von l ist dieser Zusammenhang nicht vorhanden.
Vergleicht man die Verfahren ohne Optimierung von l mit denen, wo l zunächst durch Lösen von Teilproblemen optimiert wurde, ergab sich Folgendes: Ein größerer Likelihood-Wert wurde siebenmal bei dem Verfahren ohne Optimierung von l und

Algorithmus	$\ln(L_D(l^*))$	it	$\|\nabla(L_D(l^*))\|$	time [sek]
FR_{Min}	-5.9474396202 10^2	26	3.594 10^{-2}	6.4210
PR_{Min}	-5.9474396207 10^2	16	3.191 10^{-2}	3.2820
FR_{Min}(Newton)	-5.9474396202 10^2	57	3.130 10^{-2}	643.6870
PR_{Min}(Newton)	-5.9474396889 10^2	100	2.402 10^{-1}	1.1302 10^3
$FR_{Redukt}(\gamma = \alpha_{fm} \cdot 10)$	-5.9474396209 10^2	38	4.088 10^{-2}	28.7810
$FR_{Redukt}(\gamma = \alpha_{fm})$	-5.9474396228 10^2	32	4.351 10^{-2}	6.1880
$FR_{Redukt}(\gamma = 10^{-3})$	-5.9474396215 10^2	29	3.981 10^{-2}	16.8590
$FR_{Redukt}(\gamma = 10^{-4})$	-5.9474396226 10^2	28	4.161 10^{-2}	6.5930
$PR_{Redukt}(\gamma = \alpha_{fm} \cdot 10)$	-5.9474396223 10^2	29	3.380 10^{-2}	29.6880
$PR_{Redukt}(\gamma = \alpha_{fm})$	-5.9474396216 10^2	59	4.143 10^{-2}	53.1720
$PR_{Redukt}(\gamma = 10^{-3})$	-5.9474396229 10^2	38	3.362 10^{-2}	68.8750
$PR_{Redukt}(\gamma = 10^{-4})$	-5.9474396224 10^2	29	4.368 10^{-2}	21.5780

Tabelle 3.1: Ergebnisse des Beispiels CSFV-NTR. Die Algorithmen wurden **ohne** die Optimierung von l durchgeführt.

Algorithmus	$\ln(L_D(l^*))$	it	$\|\nabla(L_D(l^*))\|$	time [sek]
FR_{Min}	-6.0183832396 10^2	19	4.131 10^{-2}	6.1250
PR_{Min}	-5.9474396206 10^2	18	3.040 10^{-2}	5.9220
FR_{Min}(Newton)	-5.9474396294 10^2	100	1.202 10^{-1}	1.3358 10^3
PR_{Min}(Newton)	-5.9474396497 10^2	100	1.937 10^2	856.4380
$FR_{Redukt}(\gamma = \alpha_{fm} \cdot 10)$	-5.9474396211 10^2	20	3.186 10^{-2}	24.6090
$FR_{Redukt}(\gamma = \alpha_{fm})$	-5.9474396204 10^2	25	3.821 10^{-2}	15.4840
$FR_{Redukt}(\gamma = 10^{-3})$	-5.9474396212 10^2	15	2.857 10^{-2}	13.2350
$FR_{Redukt}(\gamma = 10^{-4})$	-5.9474396240 10^2	17	3.992 10^{-2}	8.9220
$PR_{Redukt}(\gamma = \alpha_{fm} \cdot 10)$	-5.9474396211 10^2	22	3.278 10^{-2}	23.8120
$PR_{Redukt}(\gamma = \alpha_{fm})$	-5.9474396225 10^2	23	3.446 10^{-2}	13.8280
$PR_{Redukt}(\gamma = 10^{-3})$	-5.9474396212 10^2	20	3.229 10^{-2}	26.9850
$PR_{Redukt}(\gamma = 10^{-4})$	-5.9474396205 10^2	26	3.813 10^{-2}	15.5930

Tabelle 3.2: Ergebnisse des Beispiels CSFV-NTR. Die Algorithmen wurden **mit** Optimierung von l durchgeführt.

Algorithmus	$\ln(L_D(l^*))$	it	$\|\nabla(L_D(l^*))\|$	time [sek]
FR_{Min}	-6.7785372394 10^3	37	9.816 10^{-2}	106.5160
PR_{Min}	-6.7785372400 10^3	30	9.730 10^{-2}	88.5940
FR_{Min}(Newton)	-6.7522511632 10^3	100	1.866 10^1	3.0081 10^4
PR_{Min}(Newton)	-6.7515166283 10^3	100	2.215 10^2	3.0574 10^4
$FR_{Redukt}(\gamma = \alpha_{fm} \cdot 10)$	-6.7511371778 10^3	100	2.703 10^{-1}	1.7883 10^3
$FR_{Redukt}(\gamma = \alpha_{fm})$	-6.7511371817 10^3	45	1.109 10^{-1}	167.1250
$FR_{Redukt}(\gamma = 10^{-3})$	-6.7511371704 10^3	72	1.094 10^{-1}	960.1410
$FR_{Redukt}(\gamma = 10^{-4})$	-6.7511220522 10^3	71	1.085 10^{-1}	308.0790
$PR_{Redukt}(\gamma = \alpha_{fm} \cdot 10)$	-6.7511220633 10^3	63	1.107 10^{-1}	1.2938 10^3
$PR_{Redukt}(\gamma = \alpha_{fm})$	-6.7511221257 10^3	100	2.241 10^{-1}	348.7810
$PR_{Redukt}(\gamma = 10^{-3})$	-6.7511220540 10^3	69	1.000 10^{-1}	848.9850
$PR_{Redukt}(\gamma = 10^{-4})$	-6.7511227287 10^3	100	6.449 10^{-1}	201.4370

Tabelle 3.3: Ergebnisse des Beispiels BVDV-E2. Die Algorithmen wurden **ohne** die Optimierung von l durchgeführt.

Algorithmus	$\ln(L_D(l^*))$	it	$\|\nabla(L_D(l^*))\|$	time [sek]
FR_{Min}	-7.4654851736 10^3	31	8.783 10^{-2}	156.8750
PR_{Min}	-7.4654851747 10^3	26	1.041 10^{-1}	147.6560
FR_{Min}(Newton)	-6.7511457566 10^3	100	1.151 10^1	4.0221 10^4
PR_{Min}(Newton)	-6.7511220520 10^3	56	8.915 10^{-2}	2.3818 10^4
$FR_{Redukt}(\gamma = \alpha_{fm} \cdot 10)$	-6.7530175553 10^3	93	9.519 10^{-2}	2.3820 10^3
$FR_{Redukt}(\gamma = \alpha_{fm})$	-6.7529949168 10^3	60	1.072 10^{-1}	883.7960
$FR_{Redukt}(\gamma = 10^{-3})$	-6.7522116073 10^3	100	1.604 10^{-1}	1.6303 10^3
$FR_{Redukt}(\gamma = 10^{-4})$	-6.7530175652 10^3	49	9.647 10^{-2}	365.5780
$PR_{Redukt}(\gamma = \alpha_{fm} \cdot 10)$	-6.7511220621 10^3	54	9.258 10^{-2}	1.2446 10^3
$PR_{Redukt}(\gamma = \alpha_{fm})$	-6.7511220553 10^3	59	1.095 10^{-1}	904.7660
$PR_{Redukt}(\gamma = 10^{-3})$	-6.7511220536 10^3	59	7.989 10^{-2}	992.2190
$PR_{Redukt}(\gamma = 10^{-4})$	-6.7511245631 10^3	100	1.206 10^0	502.3900

Tabelle 3.4: Ergebnisse des Beispiels BVDV-E2. Die Algorithmen wurden **mit** Optimierung von l durchgeführt.

Algorithmus	$\ln(L_D(l^*))$	it	$\|\nabla(L_D(l^*))\|$	time [sek]
FR_{Min}	-9.3266135914 10^3	24	2.094 10^{-1}	165.9060
PR_{Min}	-9.3266135906 10^3	23	2.014 10^{-1}	154.2030
FR_{Min}(Newton)	-9.3256474228 10^3	40	2.107 10^{-1}	2.7600 10^4
PR_{Min}(Newton)	-9.3256474227 10^3	46	1.097 10^{-1}	3.7170 10^4
$FR_{Redukt}(\gamma = \alpha_{fm} \cdot 10)$	-9.3256474235 10^3	45	1.708 10^{-1}	1.4803 10^3
$FR_{Redukt}(\gamma = \alpha_{fm})$	-9.3256474372 10^3	38	1.918 10^{-1}	352.5620
$FR_{Redukt}(\gamma = 10^{-3})$	-9.3256474227 10^3	72	1.940 10^{-1}	2.6159 10^3
$FR_{Redukt}(\gamma = 10^{-4})$	-9.3256474291 10^3	35	2.091 10^{-1}	335.3290
$PR_{Redukt}(\gamma = \alpha_{fm} \cdot 10)$	-9.3256474335 10^3	38	2.171 10^{-1}	1.0037 10^3
$PR_{Redukt}(\gamma = \alpha_{fm})$	-9.3256474311 10^3	63	1.709 10^{-1}	1.7908 10^3
$PR_{Redukt}(\gamma = 10^{-3})$	-9.3256474216 10^3	49	8.731 10^{-2}	2.3209 10^3
$PR_{Redukt}(\gamma = 10^{-4})$	-9.3256474348 10^3	49	1.960 10^{-1}	1.9293 10^3

Tabelle 3.5: Ergebnisse des Beispiels BVDV-Npro. Die Algorithmen wurden **ohne** die Optimierung von l durchgeführt.

Algorithmus	$\ln(L_D(l^*))$	it	$\|\nabla(L_D(l^*))\|$	time [sek]
FR_{Min}	-9.3256474217 10^3	26	1.804 10^{-2}	377.2660
PR_{Min}	-9.3256474221 10^3	26	1.730 10^{-1}	390.9690
FR_{Min}(Newton)	-9.3256474247 10^3	60	2.038 10^{-1}	7.6667 10^4
PR_{Min} (Newton)	-9.3256890220 10^3	100	7.543	1.0521 10^5
$FR_{Redukt}(\gamma = \alpha_{fm} \cdot 10)$	-9.3256474253 10^3	47	1.978 10^{-1}	2.9606 10^3
$FR_{Redukt}(\gamma = \alpha_{fm})$	-9.3266135978 10^3	29	2.064 10^{-1}	1.3700 10^3
$FR_{Redukt}(\gamma = 10^{-3})$	-9.3266135947 10^3	22	1.834 10^{-1}	1.3523 10^3
$FR_{Redukt}(\gamma = 10^{-4})$	-9.3266136043 10^3	26	1.768 10^{-1}	969.8440
$PR_{Redukt}(\gamma = \alpha_{fm} \cdot 10)$	-9.3256474314 10^3	33	1.982 10^{-1}	2.3074 10^3
$PR_{Redukt}(\gamma = \alpha_{fm})$	-9.3256474200 10^3	33	1.229 10^{-1}	1.3985 10^3
$PR_{Redukt}(\gamma = 10^{-3})$	-9.3256474292 10^3	37	2.101 10^{-1}	2.2128 10^3
$PR_{Redukt}(\gamma = 10^{-4})$	-9.3256474353 10^3	29	1.956 10^{-1}	1.0074 10^3

Tabelle 3.6: Ergebnisse des Beispiels BVDV-Npro. Die Algorithmen wurden **mit** Optimierung von l durchgeführt.

fünfmal bei dem Verfahren mit Optimierung von l erreicht.
Betrachtet man die Laufzeit, so hatte viermal das Verfahren ohne Optimierung von l und achtmal das Verfahren mit Optimierung von l eine kürzere Laufzeit.
Bei der Kombination von Likelihood-Wert und Laufzeit war zweimal das Verfahren ohne Optimierung von l besser (größerer Likelihood-Wert und schnellere Laufzeit) und dreimal das Verfahren mit Optimierung von l besser (Tab. 3.1 und 3.2).

Beispiel BVDV-E2: Mit den Astlängen des Neighbor-Joining-Algorithmus ergab sich der Likelihood-Wert $\ln(L_D(l_{NJ})) = -6.941283\ 10^3$.
Der größte Likelihood-Wert mit $\ln(L_D(l^*)) = -6.7511220507\ 10^3$ wurde bei den Verfahren $\mathrm{PR}_{\mathrm{Min}}$(Newton) mit Optimierung von l erreicht, die Laufzeit bei diesem Verfahren war allerdings $2.6397\ 10^4$ Sekunden.
Die kürzeste Laufzeit hatte das Verfahren $\mathrm{PR}_{\mathrm{Min}}$ mit 88.5940 sek, dafür aber mit einem Likelihood-Wert von $\ln(L_D(l^*)) = -6.7785372400\ 10^3$.
Auch bei diesem Beispiel sind große Unterschiede zwischen den Likelihood-Werten erkennbar. Die Differenz der Likelihood-Werte liegt bei 714.36 (Tab. 3.3 und 3.4).
Bei der Berechnung der Eigenwerte ergab sich beim Verfahren $\mathrm{PR}_{\mathrm{Min}}$(Newton) mit Optimierung von l ein negativer Eigenwert (-53.63). Die übrigen und ansonsten positive Eigenwerte zwischen 173.76 und $2.58\ 10^4$. Das Verfahren $\mathrm{PR}_{\mathrm{Min}}$ mit Optimierung von l ergab zwei negative Eigenwerte und zwei positive, die mit Werten zwischen $-5.19\ 10^{-13}$ und $1.177\ 10^{-11}$ sehr nah bei Null lagen. Die anderen Eigenwerte lagen zwischen 173.8002 und $9.7739\ 10^5$.
Bei dem Problem BVDV-E2 waren beide Lösungen ebenfalls keine lokalen Maxima und die Likelihood-Funktion ist auch hier in einer Umgebung der Lösung nicht konvex.
Einen Zusammenhang zwischen den aktiven Nebenbedingungen und den Eigenvektoren zu den negativen Eigenwerten ist bei beiden Verfahren nicht vorhanden.
Beim Vergleich der Verfahren mit und ohne vorherige Optimierung von l hatte siebenmal das Verfahren ohne und fünfmal das Verfahren mit Optimierung von l der größeren Likelihood-Wert.
Die Laufzeit war zehnmal beim Verfahren ohne Optimierung von l und zweimal beim Verfahren mit Optimierung von l kürzer.
Bei der Kombination von Likelihood-Wert und Laufzeit war siebenmal das Verfahren ohne Optimierung von l besser (größerer Likelihood-Wert und schnellere Laufzeit) und zweimal das Verfahren mit Optimierung von l besser (Tab. 3.3 und 3.4).

Beispiel BVDV-Npro: In diesem Beispiel lag der Likelihood-Wert nach dem Neighbor-Joining-Algorithmus bei $\ln(L_D(l_{NJ})) = -9.589336\ 10^3$.
Der größte Likelihood-Wert nach der Optimierung war $\ln(L_D(l^*)) = -9.3256474200\ 10^3$. Er wurde bei einer Laufzeit von $1.3985\ 10^3$ sek mit dem Verfahren $\mathrm{PR}_{\mathrm{Redukt}}(\gamma = \alpha_{fm})$ mit Optimierung von l erreicht.
Die kürzeste Laufzeit hatte das Verfahren $\mathrm{PR}_{\mathrm{Min}}$ mit 154.2030 sek bei einem Likelihood-Wert von $\ln(L_D(l^*)) = -9.3266135906\ 10^3$.
Die Differenz zwischen größten und kleinsten liegt nur bei 0.96617 und ist damit im Vergleich zu den anderen beiden Beispielproblemen sehr klein (Tab 3.5 und 3.6).
Die Berechnung der Eigenwerte im letzten Iterationsschritt ergab sowohl beim Ver-

fahren $\mathrm{PR}_{\mathrm{Redukt}}(\gamma = \alpha_{fm})$ mit Optimierung von l (größter Likelihood-Wert) als auch beim Verfahren $\mathrm{FR}_{\mathrm{Redukt}}(\gamma = 10^{-4})$ mit Optimierung von l (kleinster Likelihood-Wert) nur positive Eigenwerte. Die Eigenwerte lagen zwischen 484.5875 und 3.1388 10^4, bzw. zwischen 483.7131 und 6.4052 10^5.
Daher ist dieses das einzige Beispielproblem, bei dem das Maximum tatsächlich gefunden wurde. Die lokale Konvexität zeigt sich auch in der geringen Differenz der Likelihood-Werte. Es gibt ein eindeutiges Maximum, gegen das alle Verfahren konvergieren.
Der Vergleich der Verfahren mit und ohne Optimierung von l zeigt, dass bezüglich der Laufzeit achtmal das Verfahren ohne und viermal das Verfahren mit Optimierung von l schneller war.
Der größere Likelihood-Wert wurde achtmal beim Verfahren ohne und viermal beim Verfahren mit Optimierung von l erreicht.
Bei der Kombination von Likelihood-Wert und Laufzeit war fünfmal das Verfahren ohne Optimierung von l besser (größerer Likelihood-Wert und schnellere Laufzeit) und nur einmal das Verfahren mit Optimierung von l besser (Tab 3.5 und 3.6).

Betrachtet man bei den Verfahren alle drei Probleme bezüglich Likelihood-Wert und Laufzeit, so ist das $\mathrm{FR}_{\mathrm{Redukt}}(\gamma = 10^{-4})$ ohne Optimierung von l am effizientesten.

Berechnung der optimalen Topologie durch die Kombination von diskretem und kontinuierlichem Problem

Aufgrund der Ergebnisse aus dem vorherigen Abschnitt wurde für diesen Teil der Auswertung der Fletcher-Reeves-Algorithmus verwendet. α wurde durch Erfüllen der Reduktionsbedingung bestimmt und es wurde $\gamma := 10^{-4}$ gewählt. Als Abbruchkriterium wurde $it < step$ und $\|\nabla(L_D(l^*)\| \leq 10^{-6} \cdot \frac{|\ln(L_D(l_s))|}{l_s}$ gewählt, wobei l_s der jeweilige Startvektor für die Astlängen sei.
Für *step* wurden die folgenden Werte gewählt:

1. Maximierung der gegebenen Topologie: $step_{\mathrm{begin}} = 20$
2. Knotentauschen: $step = step_{\mathrm{var}}$
3. Maximierung der Endtopologie: $step_{\mathrm{final}} = 100$

Es wurden verschiedene Werte für $step_{\mathrm{var}}$ verwendet und dann der maximale Likelihood-Wert und die Laufzeit ausgewertet. Die Ergebnisse sind in der Tabelle 3.7 dargestellt.

Bei den Beispielen CSFV-NTR und BVDV-E2 ergaben sich, abhängig von der Wahl für $step_{\mathrm{var}}$, verschiedene Topologien. Für das Beispiel CSFV-NTR wurde die Topologie mit dem größten Likelihood-Wert bei $step_{\mathrm{var}} = 15$ erreicht. Die Topologie war bei $step_{\mathrm{var}} = 10$ und $step_{\mathrm{var}} = 15$ identisch. Im Beispiel BVDV-E2 wurde der größte Wert bei $step_{\mathrm{var}} = 1$ erreicht. Identische Topologien ergaben sich bei $step_{\mathrm{var}} = 10$ und $step_{\mathrm{var}} = 15$. Bei dem Beispiel BVDV-Npro wurde dagegen für alle Werte von $step_{\mathrm{var}}$ die gleiche Topologie erreicht und auch der Likelihood-Wert war nahezu identisch

Parameter	$\ln(L_D(l^*))$	time [sek]	
$step_{\text{var}} = 1$,	$-5.9246343478e^2$	187.1410	
$step_{\text{var}} = 5$,	$-5.9084163718e^2$	275.1250	CSFV-NTR
$step_{\text{var}} = 10$,	$-5.9084163695e^2$	382.6720	
$step_{\text{var}} = 15$,	$-5.9084163691e^2$	540.9370	
$step_{\text{var}} = 1$,	$-6.7149836638e^3$	$4.6667e^3$	
$step_{\text{var}} = 5$,	$-6.7157356663e^3$	$4.7923e^4$	BVDV-E2
$step_{\text{var}} = 10$,	$-6.7149836676e^3$	$1.1642e^4$	
$step_{\text{var}} = 15$,	$-6.7149836642e^3$	$1.7336e^4$	
$step_{\text{var}} = 1$,	$-9.3106498786e^3$	$1.0946e^4$	
$step_{\text{var}} = 5$,	$-9.3106498798e^3$	$1.8765e^4$	BVDV-Npro
$step_{\text{var}} = 10$,	$-9.3106498853e^3$	$3.1926e^4$	
$step_{\text{var}} = 15$,	$-9.3106498931e^3$	$4.4632e^4$	

Tabelle 3.7: Likelihood-Wert und Laufzeit der drei Beispiele mit durchgeführtem Knotentausch, abhängig von $step_{\text{var}}$.

(Tab. 3.7). Bei allen drei Beispielen haben sich im Vergleich zur Neighbor-Joining Topologie kleinere Änderungen und größere Likelihood-Werte ergeben (Abb. 3.3 bis 3.8) .

3.3.3 Fazit

Astlängen

Insgesamt war bei diesen drei Beispielen das Verfahren Fletcher-Reeves (Reduktionsbedingung) mit $\gamma = 10^{-4}$ ohne vorherige Optimierung von l am besten geeignet. Es hatte eine kurze Laufzeit und erreichte einen Likelihood-Wert, der sich nur geringfügig von dem maximal erreichten Wert unterschied. Eine Optimierung der Startnäherung von l führte nur bei einigen Verfahren zu einem besserem Ergebnis. Ein Vergleich von Fletcher-Reeves und Polak-Ribiere bei ansonsten gleichem Verfahren ergab, dass häufiger mit Polak-Ribiere ein besserer Wert erreicht wurde. Bei den Beispielen, wo in einer Umgebung der erhaltenen Lösung keine Konvexität vorhanden war, konvergierten die einzelnen Verfahren gegen verschiedene Likelihood-Werte. Bei diesen Lösungen handelte es sich nicht um lokalen Maxima, die Verfahren haben also nicht funktioniert. Um die Konvergenz gegen ein lokales Maxima zu gewährleisten, muss die lokale Konvexität vorhanden seien.

Topologie

Nur bei dem Beispiel BVDV-Npro ist die Topologie bei allen Werten für $step_{\text{var}}$ identisch. Der Baum mit dieser Topologie ist in Abbildung 3.8(Seite 87) dargestellt.

Im Vergleich zur Neighbor-Joining Topologie ist der Likelihood-Wert größer. Die Änderungen in der Topologie sind entweder innerhalb der Gruppen oder in der Anordnung der Gruppen. Die Gruppen selber und die Unterteilung in BVDV-1, BVDV-2 und BDV sind erhalten geblieben.
Bei den anderen beiden Beispielen haben sich unterschiedliche Topologien für die verschiedenen Werte ergeben. Dies hängt vermutlich ebenfalls mit der nicht vorhandenen lokalen Konvexität zusammen, weil die einzelnen kontinuierlichen Probleme nicht gegen ein lokales Maximum konvergieren. In den Abbildungen 3.4(Seite 83) und 3.6(Seite 85) sind die Topologien dargestellt, die den größten Likelihood-Wert ergaben. Ob dies die ideale Topologie ist, kann man aufgrund der fehlenden Konvexität nicht definitiv sagen. Bei dem BVDV-E2 Beispiel waren die Topologie Änderungen wie schon beim Beispiel BVDV-Npro nur in der Anordnung der Gruppen oder innerhalb der Gruppen. Bei dem CSF Beispiel dagegen ist nur noch die Einteilung in die Genotypen 1 und 2 und die Subgruppe 2.3 vorhanden. Dies kann entweder mit den 5'NTR-Sequenzen zusammenhängen, die sich, weil die 5'NTR hochkonserviert ist, sehr ähnlich sind, oder ist ein Hinweis dafür, dass diese Topologie nicht die ideale ist.

Konvexität

Ob die Likelihood-Funktion in einer Umgebung der Lösung konvex ist, hängt von verschiedenen Faktoren ab. Die folgenden Zusammenhänge lassen sich aus den Ergebnissen vermuten. Zunächst spielen die Sequenzen eine Rolle, denn für den Fall $n = 2$ Sequenzen hing es von dem Verhältnis von gleichen und unterschiedlichen Basen ab, ob die Funktion konvex war. Auch bei den Beispielen CSFV-NTR und BVDV-E2 waren die Sequenzen wahrscheinlich der Grund für die fehlende lokale Konvexität. Nur der Anteil an konservierten Basen allein genügt aber nicht als Kriterium. In den Beispielen wurden Sequenzdaten verwendet, bei denen 20%, 23%, 36%, 39.7%, 59.3% und 75.4% der Basen konserviert waren. Die Berechnung der Eigenwerte der Hessematrix ergab keine Konvexität bei den Sequenzdaten mit 23% und 75.4%. Man kann also nicht sagen, dass ein bestimmter Prozentbereich gleichbedeutend mit Konvexität ist und alles außerhalb dieses Bereiches nicht konvex ist.
Die Ergebnisse aus dem Spezialfall $n = 2$ zeigten, dass die Funktion nicht auf dem vollständigen Definitionsbereich konvex ist. Dies zeigte sich auch beim Beispiel mit $n = 5$ Sequenzen. Bei der Optimierung mit dem Startvektor l_{NJ} wurde ein lokales Maximum gefunden, mit Startvektor $l_{1.5} = [1.5, 1.5, 1.5, 1.5, 1.5, 1.5, 1.5, 1.5]$ erhielt man dagegen eine Lösung die kein lokales Maximum ist und in deren Umgebung die Funktion auch nicht konvex ist.
Bei dem Beispiel CSFV-NTR war bei einem Verfahren ein Zusammenhang zwischen großen Einträgen in den Eigenvektoren, der negativen Eigenwerte und den aktiven Nebenbedingungen erkennbar. Dies lässt vermuten, dass die Konvexität am Rand verloren geht.
Aufgrund der Komplexität der Funktion sind die exakten Bedingungen allerdings nicht herzuleiten.

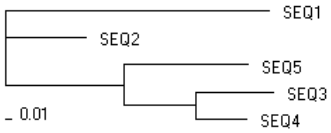
Ist die Konvexität in einer Umgebung der Lösung nicht vorhanden, ist die Lösung auch kein lokales Maximum. Das Verfahren hat in diesen Fällen versagt. Diese Problematik zeigt sich bei den Beispielen CSFV-NTR und BVDV-E2, bei denen die verschiedenen Verfahren gegen unterschiedliche Werte konvergierten.

Viorolgie

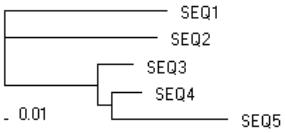
Aus virologischer Sicht kann man zwei Erkenntnisse ziehen:

1. Der hier verwendete Maximum-Likelihood Algorithmus ist in der Praxis nicht immer sinnvoll nutzbar. Um sichere Ergebnisse zu bekommen, benötigt man die lokale Konvexität, die nicht bei allen Beispielen gegeben ist. Die Voraussetzung, einen Startvektor zu benutzen, der nahe genug an der richtigen Lösung liegt, setzt voraus, dass man zunächst den NJ Algorithmus verwendet. Um nur den Genotyp zubestimmen, ist Maximum-Likelihood Algorithmus zu aufwendig. Dieses Ergebnis hat man bereits mit dem NJ Algorithmus. Die NJ-Heuristik funktionierte in allen betrachteten Beispielen, allerdings stellt die Likelihood-Funktion im Gegensatz zur NJ-Heuristik ein überzeugend begründbares Modell dar. Um im Zweifelsfall sicher zu gehen, dass die Ergebnisse des NJ korrekt sind, ist eine Absicherung mit dem Maximum-Likelihood-Algorithmus durchaus sinnvoll.

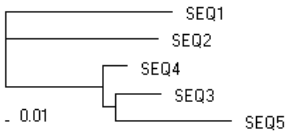
2. Das einzige konvexe Problem BVDV-Npro zeigt, dass die Gruppeneinteilung erhalten bleibt, was für die Richtigkeit dieser Einteilung spricht.



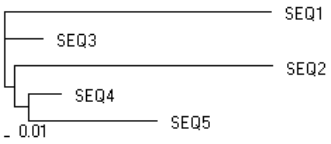
(a) NJ Topologie, $\ln L_D(l) = -1.338810^3$



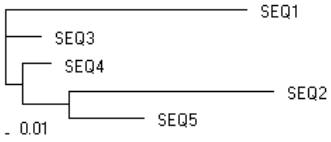
(b) Topologie 1, $\ln L_D(l) = -1.344210^3$



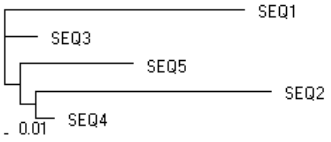
(c) Topologie 2, $\ln L_D(l) = -1.344810^3$



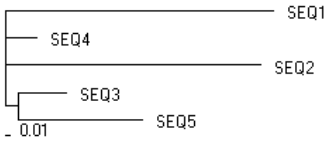
(d) Topologie 3, $\ln L_D(l) = -1.356310^3$



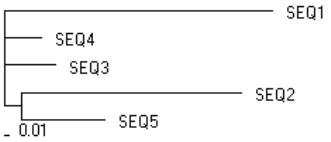
(e) Topologie 4, $\ln L_D(l) = -1.350610^3$



(f) Topologie 5, $\ln L_D(l) = -1.355810^3$

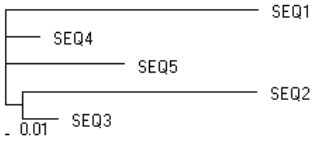


(g) Topologie 6, $\ln L_D(l) = -1.358010^3$

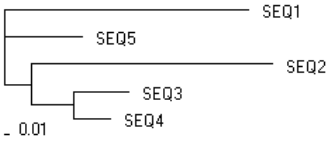


(h) Topologie 7, $\ln L_D(l) = -1.352310^3$

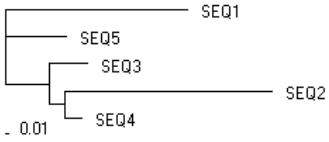
Abbildung 3.1: Mögliche Topologien mit den zugehörigen maximalen Likelihood-Werte für das Modellproblem mit $n = 5$ Sequenzen.



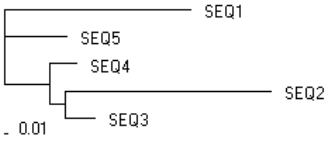
(a) Topologie 8, $\ln L_D(l) = -1.357610^3$



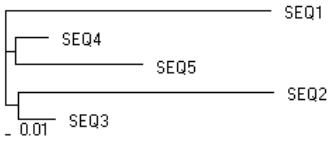
(b) Topologie 9, $\ln L_D(l) = -1.345910^3$



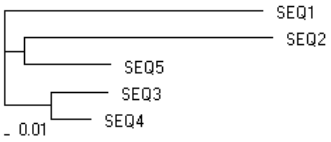
(c) Topologie 10, $\ln L_D(l) = -1.349210^3$



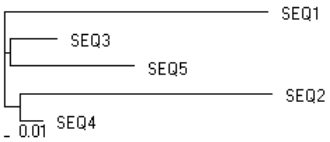
(d) Topologie 11, $\ln L_D(l) = -1.349310^3$



(e) Topologie 12, $\ln L_D(l) = -1.356310^3$



(f) Topologie 13, $\ln L_D(l) = -1.346710^3$



(g) Topologie 14, $\ln L_D(l) = -1.357110^3$

Abbildung 3.2: Fortsetzung von Abbildung 3.1.

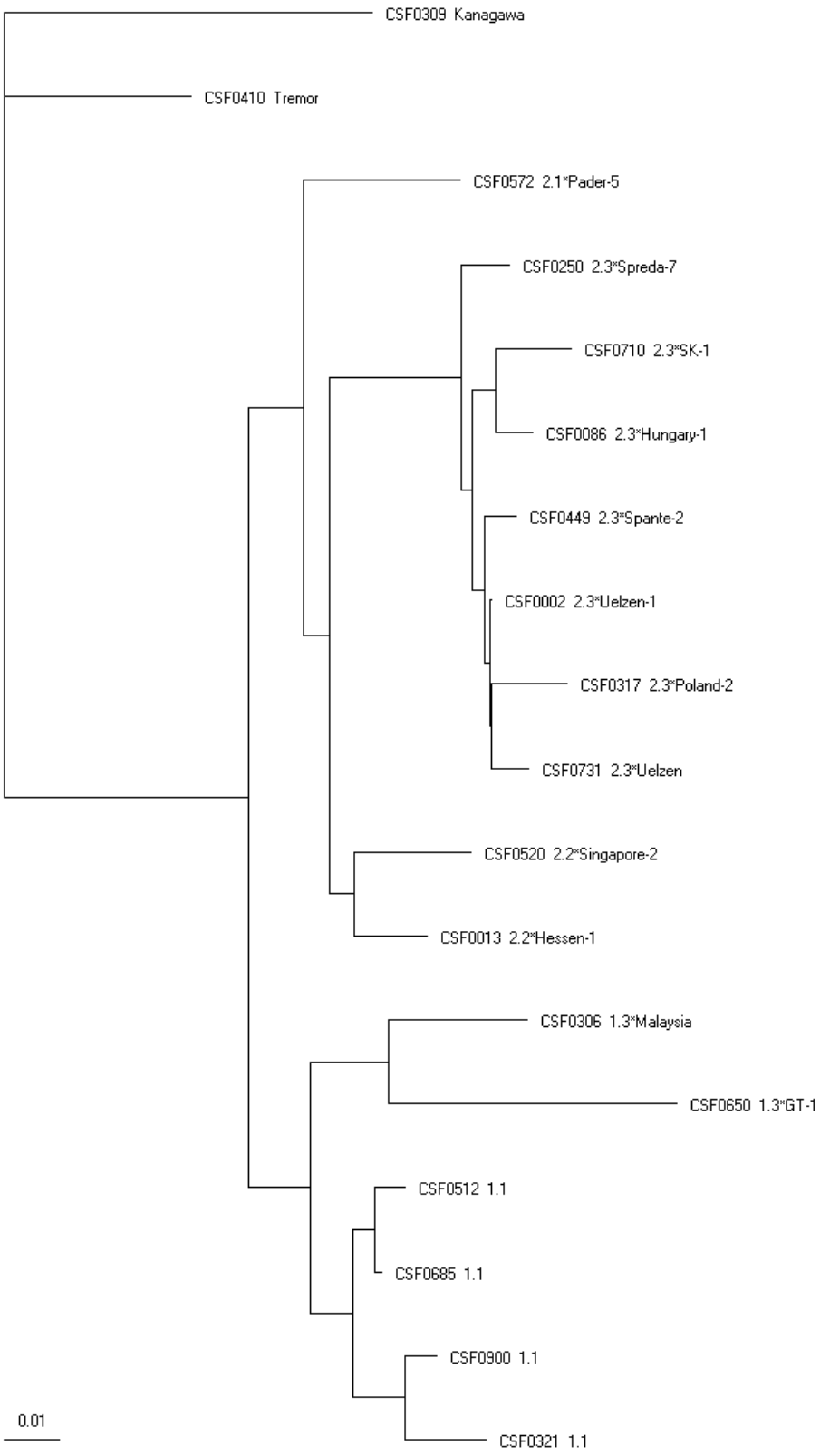


Abbildung 3.3: Neighbor-Joining Topologie für das Beispiel CSFV-NTR. Die CSF Nummern entsprechen der Nummerierung aus der CSFV-Datenbank, die folgende Bezeichnung steht für den Genotyp und die Subgruppen.

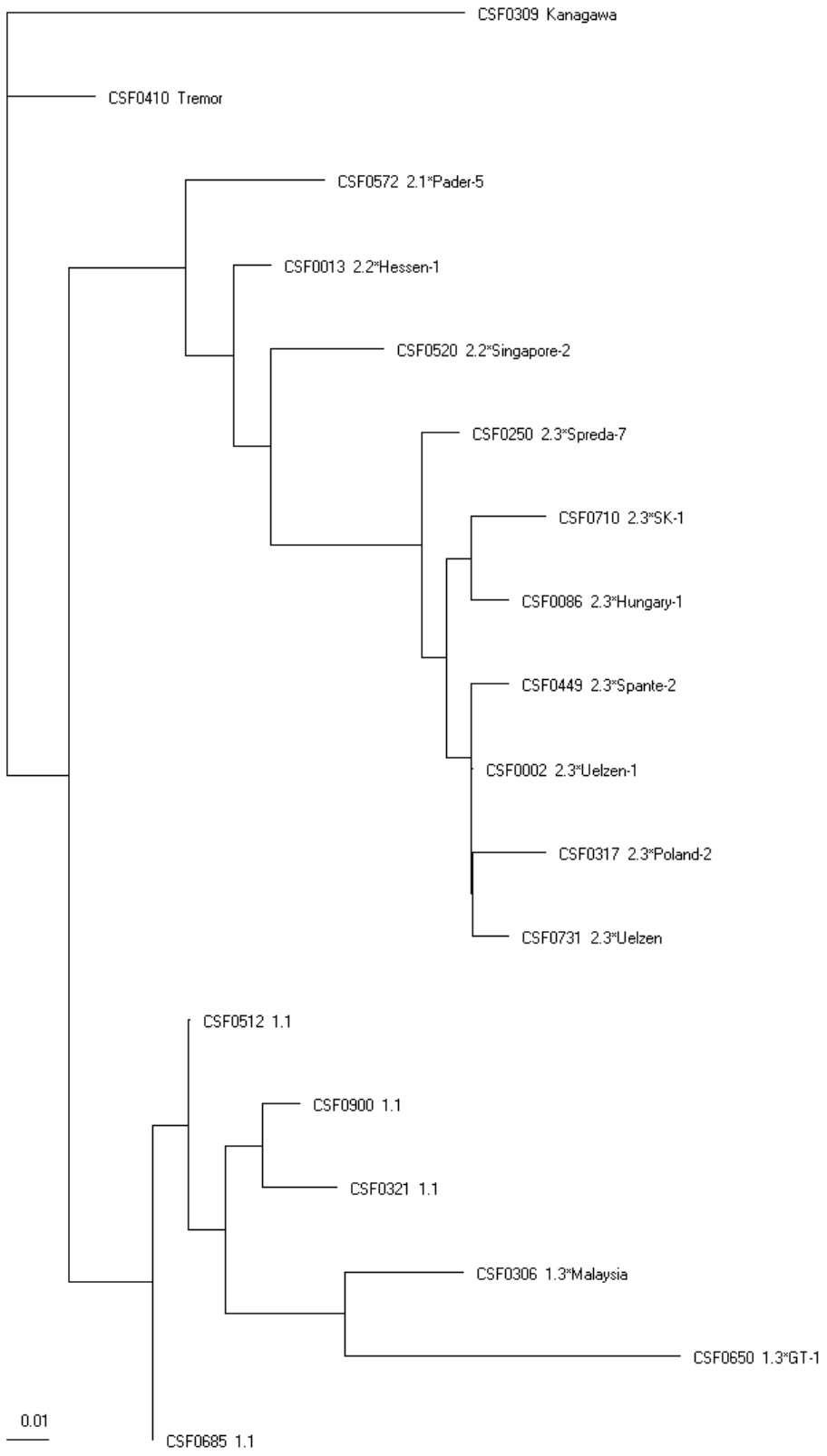


Abbildung 3.4: Veränderte Topologie für das Beispiel CSFV-NTR, nach dem durchgeführtem Knotentausch mit Wert $step_{\text{var}} = 15$. Die CSF Nummern entsprechen der Nummerierung aus der CSFV-Datenbank, die folgende Bezeichnung steht für den Genotyp und die Subgruppen.

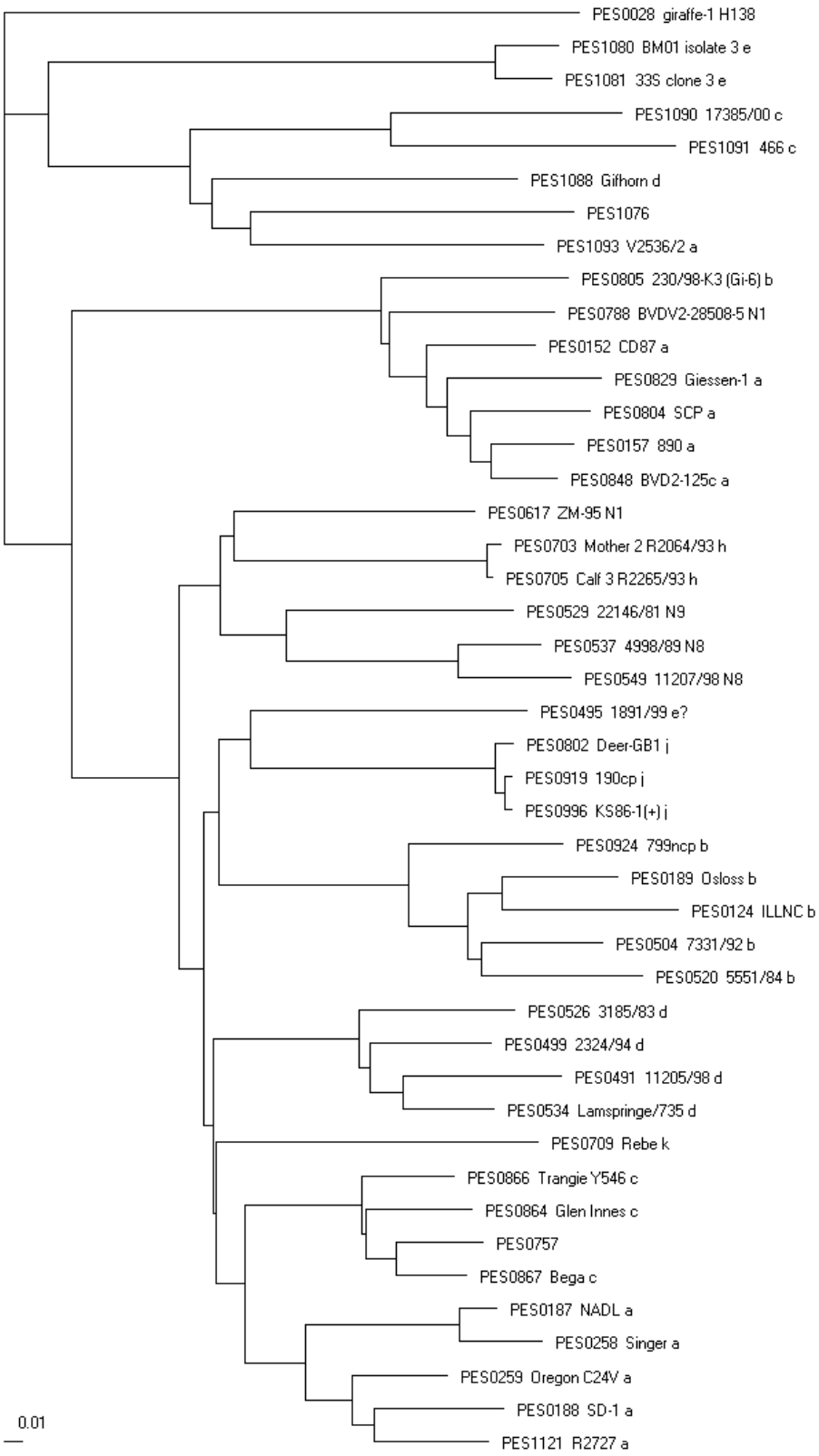


Abbildung 3.5: Neighbor-Joining Topologie für das Beispiel BVDV-E2. Die PES Nummern entsprechen der Nummerierung aus der BVDV-Datenbank, die folgende Bezeichnung steht für die Originalnamen und die Einteilung in die Gruppen (a-k, bzw. N1, N8 und N9).

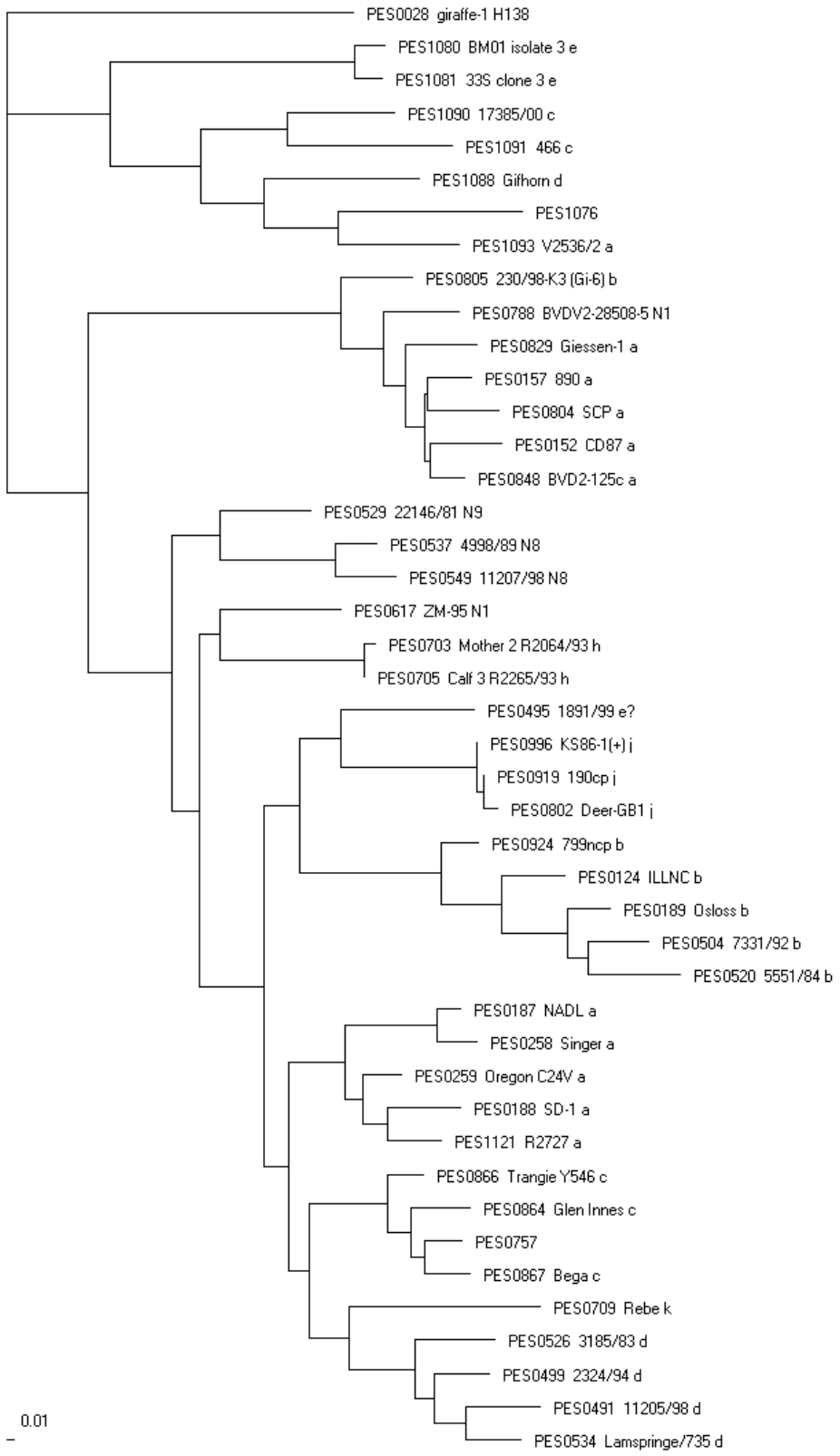


Abbildung 3.6: Veränderte Topologie für das Beispiel BVDV-E2, nach dem durchgeführtem Knotentausch Wert $step_{\text{var}} = 1$. Die PES Nummern entsprechen der Nummerierung aus der BVDV-Datenbank, die folgende Bezeichnung steht für die Originalnamen und die Einteilung in die Gruppen (a-k, bzw. N1, N8 und N9).

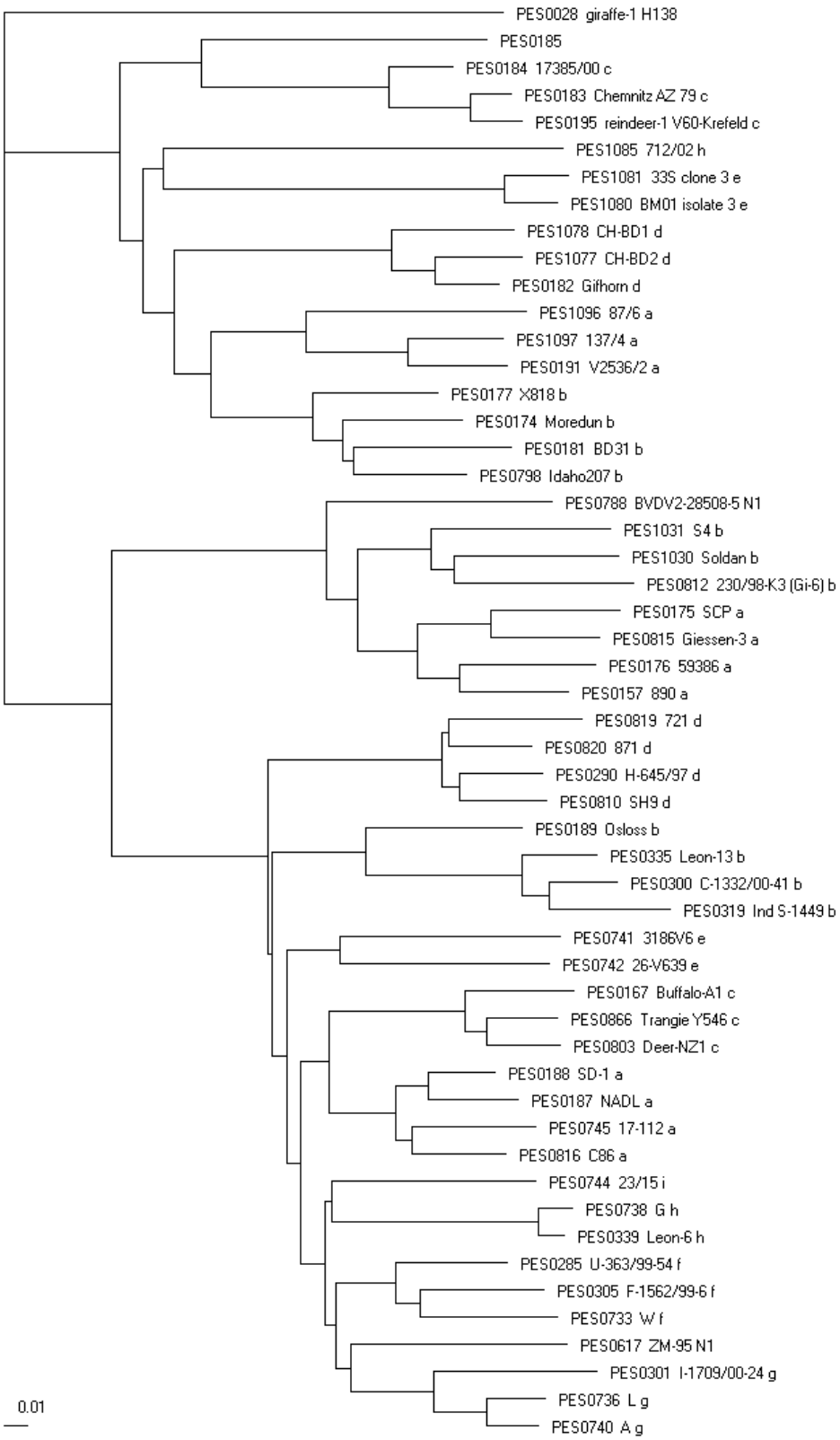


Abbildung 3.7: Neighbor-Joining Topologie für das Beispiel BVDV-Npro. Die PES Nummern entsprechen der Nummerierung aus der BVDV-Datenbank, die folgende Bezeichnung steht für die Originalnamen und die Einteilung in die Gruppen (a-k, bzw. N1, N8 und N9).

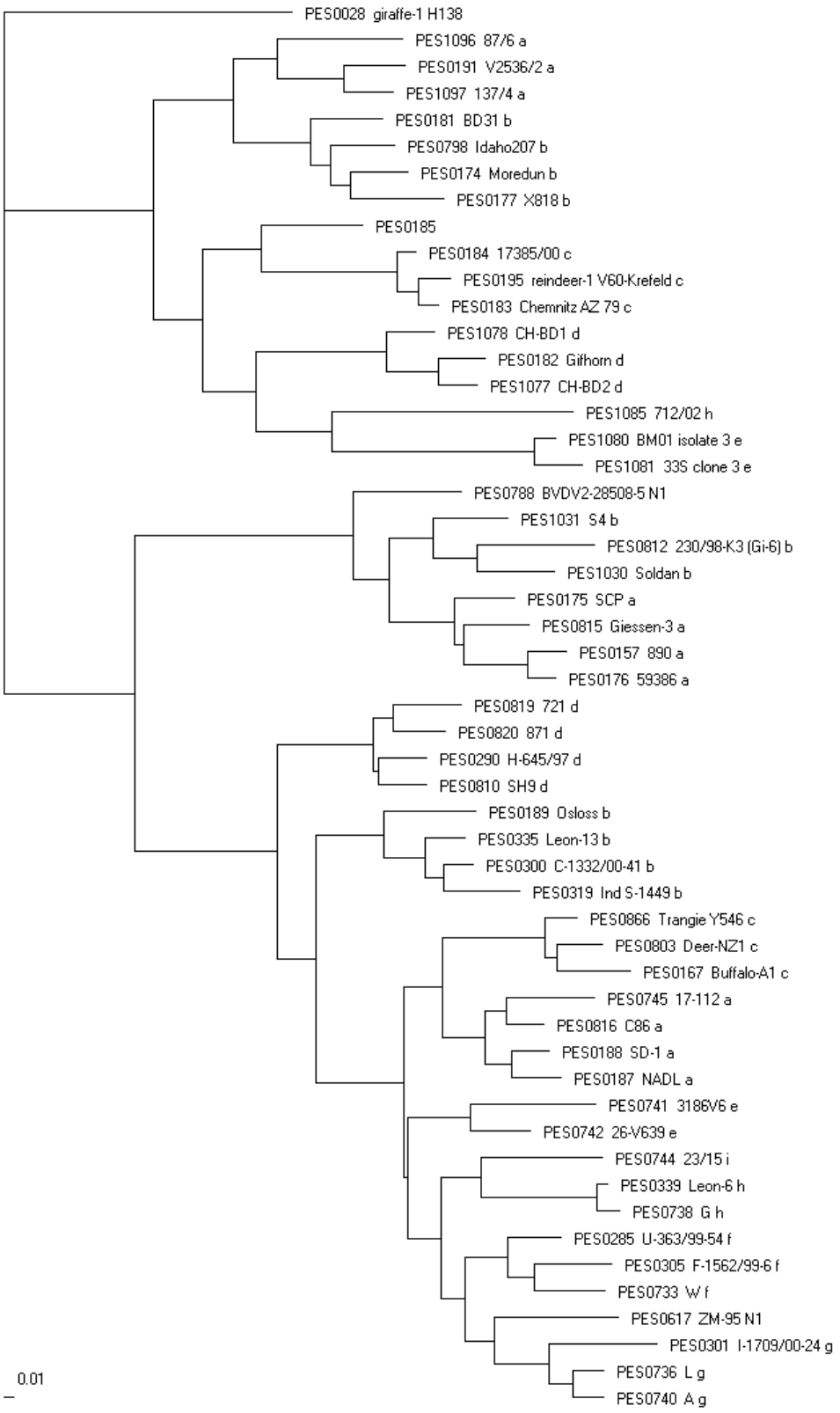


Abbildung 3.8: Veränderte Topologie für das Beispiel BVDV-Npro, nach dem durchgeführtem Knotentausch Wert $step_{\text{var}} = 1$.Die PES Nummern entsprechen der Nummerierung aus der BVDV-Datenbank, die folgende Bezeichnung steht für die Originalnamen und die Einteilung in die Gruppen (a-k, bzw. N1, N8 und N9).

Kapitel 4

Automatisierte Genotypisierung von CSF Virusisolaten

Die Klassische Schweinepest ist eine weltweit verbreitete Tierseuche, die hohe wirtschaftliche Schäden verursacht [PM00]. Daher ist es von großem Interesse, die Ausbrüche von Klassischer Schweinepest möglichst gering zu halten bzw. zu verhindern. Ein wichtiger Bestandteil der Bekämpfungsmaßnahmen ist die Genotypisierung der Erregerisolate, mit deren Hilfe man den Infektionsweg zurückverfolgen kann. Hierfür sind die Sequenzen und epidemiologischen Daten möglichst vieler bisheriger Isolate nötig. Die ständige Verfügbarkeit dieser Sequenzen war der Hintergrund der Entwicklung der CSFV-Datenbank des *EU Reference Laboratory for Classical swine fever*, Hannover [GZ00]. Mittlerweile beinhaltet die Datenbank so viele Einträge, dass eine manuelle Auswahl der Vergleichssequenzen für die Genotypisierung nicht mehr möglich ist. Um die Datenbank weiterhin effizient nutzen zu können, wurde ein Verfahren gesucht, das eine automatisierte Sequenzauswahl ermöglicht.

4.1 CSFV-Datenbank

Ausgang für die Datenbank war die vorhandene CSFV Datenbank des *EU Reference Laboratory for CSF*. Im Gegensatz zur ersten Version der Datenbank [GZ00] sind die Sequenzen und Daten in der aktuellen Version in einer SQL Datenbank (MySQL v. 4.00.24) gespeichert. Die Datenbank ist im Internet über eine Verwaltungsoberfläche unter der URL http://viro08.tiho-hannover.de/eg/csf/startCSF.cgi verfügbar
Jedem Eintrag wird eine CSF Nummer zugeordnet (CSFnnnn oder XXXnnnn). Bei den mit CSF gekennzeichneten Einträgen ist das Isolat selbst vorhanden, bei den mit XXX gekennzeichneten Einträgen ist nur die Sequenz vorhanden. Es wurden Daten wie Isolierungsjahr, Herkunftsland und Genotyp gespeichert. Von den Sequenzen wurden, soweit vorhanden, die folgenden Teilstücke der genomischen Regionen gespeichert:

- **NTR:** 201.-350. Base der Sequenz des Stammes Alfort (150 Basen)

- **E2:** 2518.-2707. Base der Sequenz des Stammes Alfort (190 Basen)
- **NS5B:** 11158.-11566. Base der Sequenz des Stammes Alfort (409 Basen)

In der Datenbank sind unter anderem alle in GenBank [Ano06b] verfügbaren Sequenzen gespeichert. Zur Zeit enthält sie 654 E2-, 100 NS5B- und 618 5'NTR-Sequenzen. Die Datenbank beinhaltet mehrere Isolate mit identischer Sequenz. Diese werden im Weiteren für die Berechnung der *Alignments* und Phylogenetischen Bäume und zur Beantwortung der Frage, ob eine Sequenz schon in der Datenbank ist, benötigt. Um diese Isolate nicht jedesmal neu berechnen zu müssen und zum Speichern weiterer Daten, die sich nicht ständig ändern, wurden in der Datenbank weitere Tabellen angelegt, die für jeden Sequenztyp die folgenden Felder enthält:

- **baseseq:** In diesem Feld sind alle Isolate aufgelistet, die eine unterschiedliche Sequenz haben.
- **duplicates:** Dieses Feld enthält alle Isolate, die eine identische Sequenz, wie das Isolat aus dem zugehörigen Feld baseseq, haben.
- **refseq:** Dieses Feld enthält die Information, ob das Isolat aus dem zugehörigen Feld baseseq eine Referenzsequenz für den Baum ist (Wert eins) oder nicht (Wert null).
- **seqalign:** Es wird ein *Alignment* der Sequenzen von Alfort und des Isolates aus dem zugehörigen Feld baseseq berechnet. Die alignte Sequenz des Isolates aus dem Feld baseseq wird auf die Länge der Alfort geschnitten und inklusive aller Lücken in diesem Feld gespeichert.

Die Referenzsequenzen wurden so gewählt, dass alle Genotypen im Baum vorkommen und der Baum übersichtlich ist. Da nicht alle drei Genfragmente für jedes Isolat vorhanden sind, gibt es für die drei Genfragmente 5'NTR, E2 und NS5B verschiedene Referenzsequenzen. Die Sequenz aus dem Feld seqalign wird für die Berechnung des Baumes verwendet. Dadurch, dass das *Alignment* aller Sequenzen mit Alfort nicht jedesmal neu berechnet wird, verringert sich die Laufzeit. Diese Tabellen werden regelmäßig neu berechnet, damit sie immer mit den aktuellen Einträgen in der Datenbank übereinstimmen.

4.2 Genotypisierung von CSF Virusisolaten

Die folgende geographische und temporäre Verteilung der in der CSFV-Datenbank vorhandenen Isolate hat gezeigt, dass bei dem CSFV die Gruppeneinteilung zusätzlich durch die epidemiologischen Daten unterstützt wird. Für jede Subgruppe wurden die Länder, aus denen Isolate dieser Subgruppe existierten, unterteilt nach Kontinenten mit den zugehörigen Jahreszahlen aufgelistet. DE (64, 90-95) bedeutet z.B., dass

Isolate dieser Subgruppe aus Deutschland aus dem Jahr 1964 und den Jahren 1990 bis 1995 existieren. Die Länderkürzel sind in Anhang B aufgelistet.

Subgruppe 1.1: Ausbreitung in Asien, Europa und Amerika

- Asien: RU (95-99), TH (88, 93, 94), JP (64), CN, IN
- Europa: DE (64, 68, 94), FR (68), SK (97), RO (01), SI, CZ , HR, GB
- Nordamerika: US (46, 54, 69), MX (91, 92, 97)
- Südamerika: CO (80, 98, 99, 02, 03), AR (78, 81), BR (87, 95, 97)
- sonstige: Alfort, Impfstämme

Subgruppe 1.2: Ausbreitung in Asien, Europa und Nordamerika

- Asien: TH (91)
- Europa: CZ (93, 96), PL (93), IT(45), RO, NL, CH, RU
- Nordamerika: CU (58, 93, 96, 97), US

Subgruppe 1.3: Ausbreitung nur in Asien und Nordamerika

- Asien: TH (88, 91, 93), MY (86)
- Nordamerika: CR (94), HN (92, 96), GT

Subgruppe 2.1: Ausbreitung in Afrika, Asien und Europa

- Afrika: ZA (05)
- Asien: MY (86), LA (98,99), TW (94, 01, 03), IN, KR, CN
- Europa: DE (89, 97), CH (93), AT (93), IT (97), BE (97), NL (92, 97), ES (97, 98, 01), HR (97),GB (00)

Subgruppe 2.2: Ausbreitung in Asien, Europa und Südamerika

- Asien: TH (96, 97), SG(87, 88), LA (97,98), TW (95)
- Europa: DE (89, 90, 97, 98), AT (90, 92, 94, 96), IT (82, 91, 97, 98), CZ (91-97, 00), CH (98), SK (96), NL
- Südamerika: CO (05), VE(04-05)

Subgruppe 2.3: Ausbreitung nur in Europa

- Europa: DE (82, 84, 89, 90, 92-03), GB (87), HR (97, 02, 03), BG (97-99, 02), BE (86, 89, 93, 94), LU (02), YU(99), FR (89, 92, 97, 02, 03), HU (92), PL (91-96), ES (01, 02), AT (94, 95, 00), CH (93), CZ (96-98), SK (93-00), IT (82, 91, 92, 94, 95, 97, 01), RO (94, 02, 04)

Subgruppe 3.1:

- sonstige: outgroup Kanagawa

Subgruppe 3.2: Ausbreitung nur in Asien

- Asien: Korea (Ende 90er)

Subgruppe 3.3: Ausbreitung nur in Asien

- Asien: TH (91-95)

Subgruppe 3.4: Ausbreitung nur in Asien

- Asien: TW (90, 96)

4.3 Implementierung des *Genetic-Typing*-Moduls

Die schnelle Genotypisierung des Erregerisolates ist im Falle eines neuen Ausbruchs von Klassischer Schweinepest zur Unterstützung der Epidemiologen bei der Bekämpfung des Ausbruchs sehr wichtig. Um die CSFV-Datenbank dafür sinnvoll nutzen zu können, wurde ein Modul implementiert, dass die Genotypisierung von neuen CSFV-Isolaten automatisiert. Die bisherige Methode mit einer manuellen Auswahl der Sequenzen und der Berechnung des phylogenetischen Baums unter Zuhilfenahme mehrerer Programme [PM00], ist aufgrund der großen Anzahl an Einträgen in der Datenbank nicht mehr möglich. Dieses neue Modul wurde in die Bedienoberfläche der Datenbank eingefügt und als *Gentic-Typing*-Modul bezeichnet. Für eine Sequenz überprüft das Programm zunächst, ob in der Datenbank bereits ein Isolat mit identischer Sequenz existiert und berechnet dann einen phylogenetischen Baum. Dieser Baum enthält zusätzlich die Information, welcher Subgruppe die verwendeten Sequenzen angehören, soweit er in der Datenbank gespeichert ist. Das Programm ermöglicht die Eingabe von bis zu fünf ungeschnittenen Sequenzen und ihrer genomischen Region (5'NTR, E2 oder NS5B). Es sind zwei Versionen entstanden. Die eine Version ist im Internet öffentlich zugänglich, die andere ist eine administrative Version, die über mehr Funktionen verfügt und auch die Möglichkeit beinhaltet, in der Datenbank Änderungen vorzunehmen, z.B. die Referenzsequenzen zu ändern. Das Modul wurde in Borland Delphi 7 Enterprise implementiert. Der Zugriff auf

die CSFV-Datenbank erfolgte mittels MySQL Data Access Components (MyDac, CoreLab Software development). Für die öffentliche Version wurden die Intraweb Komponenten von AtoZ-Software verwendet.
Im Programm wurden folgende Algorithmen implementiert:

- Beim **Direktvergleich** (DV) wird die prozentuale Basenübereinstimmung der zwei Sequenzen ausgegeben. Um Sequenzen zu berücksichtigen, die an verschiedenen Positionen des Genoms beginnen, wird eine Basenverschiebung von bis zu 100 Basen durchgeführt und der größte Wert angezeigt.

- Mit dem **Smith-Waterman-Algorithmus** (SW) wird ein lokales *Alignment* von zwei Sequenzen a und b berechnet. Zusätzlich wird die Ähnlichkeit der beiden Sequenzen mit einer Kostenfunktion

$$w'(\overline{a_i}, \overline{b_i}) := \begin{cases} 1 & \text{falls } \overline{a_i} = \overline{b_i} \text{ (Match)} \\ -0.6 & \text{falls } \overline{a_i} \neq \overline{b_i} \text{ und } \overline{a_i}, \overline{b_i} \neq - \text{ (Mismatch)} \end{cases} \tag{4.1}$$

  und den Parametern $\nu = 4.5$ (gap open) und $\mu = 0.05$ (gap extention) für die Lückenstrafe berechnet.

- Mit dem **Needleman-Wunsch-Algorithmus** (NW) wird ein globales *Alignment* von zwei Sequenzen a und b berechnet. Zusätzlich wird die Distanz der beiden Sequenzen mit einer Kostenfunktion

$$\overline{w}(\overline{a_i}, \overline{b_i}) := \begin{cases} 0 & \text{falls } \overline{a_i} = \overline{b_i} \text{ (Match)} \\ 1 & \text{falls } \overline{a_i} \neq \overline{b_i} \text{ und } \overline{a_i}, \overline{b_i} \neq - \text{ (Mismatch)} \end{cases} \tag{4.2}$$

  und den Parametern $\nu = 10$ (gap open) und $\mu = 0.2$ (gap extention) für die Lückenstrafe berechnet.

- Berechnung der **Jukes-Cantor-Distanz** (JC) von zwei Sequenzen.

- Zur Berechnung der Phylogenetischen Bäume wird der **Neighbor-Joining-Algorithmus** (NJ) verwendet, weil er unparallelisiert die kürzeste Laufzeit hat. Eine Parallelisierung war aufgrund der technischen VorausSetzungen nicht umsetzbar. Ein Zugriff auf die im Internet verfügbaren Versionen widersprach der Idee, ein Programm zu entwickeln, dass nach Eingabe einer Sequenz alle weiteren Schritte bis zur Ausgabe des phylogenetischen Baums selbständig durchführt.

- Zusätzlich wurde eine Prozedur zum **Überprüfen einer Sequenz** implementiert. Hier wird zunächst ein lokales *Alignment* mit der Referenzsequenz (bei CSF: Alfort) berechnet. Die Sequenz ist nicht korrekt, wenn das *Alignment* Lücken enthält, deren Länge größer als drei ist, der Wert der Ähnlichkeit kleiner als 30 ist (d.h. das Genfragment stimmt nicht) oder bei kodierenden Sequenzen, jeder der drei möglichen Leserahmen einen Stopcodon (TAA, TAG oder TGA) enthält.

4.3.1 Graphische Ausgabe des Baumes

Nach der Berechnung des Baumes mit dem NJ-Algorithmus sind nur die Distanzen der OTUs zueinander und die Verknüpfung untereinander bekannt. Um die graphische Ausgabe zu realisieren, muss ein Faktor bestimmt werden, der die Distanzen zwischen den Knoten auf eine geeignete Länge streckt und es müssen die vertikalen Stücke des Baumes berechnet werden. Sie sind von den Distanzen unabhängig und durch Verändern ihrer Längen verändert man lediglich die Optik des Baumes. Sind sie schlecht gewählt, kann es sein, dass Äste aufeinanderliegen, dass der Baum so klein ist, dass man nichts mehr erkennt, oder dass er zu groß wird. Die gleiche Bedeutung hat der Faktor für die Distanzen.

Die vertikalen Längen werden aus den Daten des *rooted tree* rekursiv berechnet. $L(i)$ sei die Länge des Stücks durch den Knoten i. Es wird außerdem eine Konstante A festgelegt, die den vertikalen Abstand der CSF-Nummer bestimmt. Betrachtet werden zur Berechnung von $L(i)$ die Nachfolger von i. $N(k) := (k-1) \cdot A$.
Zunächst wird eine Tabelle angelegt, die für jeden Knoten i folgende Daten speichern kann: Anzahl der Nachfolger von i ($Anz(i)$), $L(i)$, Gesamtlänge nach oben ($O(i)$) und Gesamtlänge nach unten ($U(i)$). Am Anfang sind alle Einträge gleich null. Als erstes betrachtet man alle Knoten i, die als Nachfolger zwei CSF-Nummern, also keine Knoten haben. Für diese sei $Anz(i) := 2$, $L(i) := A$, $O(i) := 0$ und $U(i) := 0$. Ab jetzt prüft eine Schleife bei allen Knoten, für die $L(i)$ noch nicht berechnet ist, ob $L(i)$ für die Knoten unter den beiden Nachfolgern schon berechnet sind. Ist dies der Fall, unterscheidet man drei Fälle:

1. Knoten i hat eine CSF-Nummer und Knoten j als Nachfolger. Für die Tabelle ergeben sich dann die folgenden Einträge:
   $Anz(i) := 1 + Anz(j)$,
   $L(i) := N(Anz(i)) - L(j)/2 - U(j)$,
   $O(i) := 0$ und $U(i) := U(j) + L(j)/2$.

2. Knoten i hat Knoten j und eine CSF-Nummer als Nachfolger. Für die Tabelle ergeben sich dann die folgenden Einträge:
   $Anz(i) := 1 + Anz(j)$,
   $L(i) := N(Anz(i)) - L(j)/2 - O(j)$,
   $O(i) := O(j) + L(j)/2$ und $U(i) := 0$.

3. Knoten i hat Knoten j und Knoten h als Nachfolger. Für die Tabelle ergeben sich dann die folgenden Einträge:
   $Anz(i) := Anz(j) + Anz(h)$,
   $L(i) := N(Anz(i)) - L(j)/2 - L(h)/2 - O(j) - U(h)$,
   $O(i) := O(j) + L(j)/2$ und $U(i) := U(h) + L(h)/2$.

Die Schleife bricht ab, wenn für alle Knoten $L(i)$ berechnet wurde.

4.3.2 Funktionen der öffentlichen Version

Im Hauptfenster wird die aktuelle Anzahl der in der Datenbank gespeicherten Daten angezeigt. Als erstes wird das Genfragment ausgewählt, für das die Berechnungen durchgeführt werden sollen. Danach kann man bis zu fünf Sequenzen entweder als CSF Nummer für Sequenzen aus der Datenbank, oder als neue Sequenz eingeben. Ein Schneiden der rohen Sequenzen ist nicht nötig. Über Checkboxen wählt man aus, mit welchen Sequenzen gerechnet werden soll.

Die öffentliche Version bietet dem Nutzer folgende Optionen:

- Die Option *test sequence* überprüft mit dem oben genannten Algorithmus, ob die ausgewählten Sequenzen korrekt sind. Für jede geprüfte Sequenz wird eine Meldung mit dem Ergebnis angezeigt

- *Show sequence* zeigt die zu den CSF Nummern gehörenden Sequenzen an.

- Die Ausgabe aller Isolate, deren Sequenz mit der ersten eingegebenen Sequenz übereinstimmt, erfolgt mit *show isolates*. Ist nur eine CSF Nummer angegeben, werden alle Isolate angezeigt, die die gleiche Sequenz haben, wie das zu der CSF Nummer gehörende Isolat. Hierfür wird zunächst ein Direktvergleich mit allen Basissequenzen (baseseq) durchgeführt. Ist eine identische Sequenz vorhanden, werden alle zu dieser Basissequenz gehörenden Isolate (*duplicates*) angezeigt.

- *Pairwise Alignment* berechnet für die erste eingegebene Sequenz ein *Alignment* mit der Sequenz des Isolats Alfort und den zehn ähnlichsten Sequenzen (Kriterium für die Auswahl sind die Jukes-Cantor-Distanzen) aus der Datenbank. Hat die Sequenz die gleiche Länge wie die Alfort-Sequenz wird das *Alignment* mit dem Needleman-Wunsch-Algorithmus berechnet. Ist die Sequenz kürzer oder länger, wird das *Alignment* mit dem Smith-Waterman Algorithmus berechnet.

- Die Berechnung eines phylogenetischen Baumes, wahlweise mit oder ohne *Bootstrap*-Werte, und die graphische Darstellung ist mit der Option *Multiple Alignment* möglich. Folgende Schritte werden dafür intern durchgeführt:

  1. Zusammenstellung aller Sequenzen, die für den Baum verwendet werden. Diese setzen sich aus den Referenzsequenzen aus der Datenbank und aus den korrekten Eingabesequenzen zusammen. Für die Eingabesequenzen, die nicht in der Datenbank sind, werden zusätzlich die zwei ähnlichsten Sequenzen (Kriterium für Auswahl sind die Jukes-Cantor-Distanzen) aus der Datenbank der Aufstellung hinzugefügt.
  2. Berechnung einer Distanzmatrix (Jukes-Cantor-Distanzen) für alle in 1. aufgeführten Sequenzen.
  3. Berechnung eines *unrooted tree* aus der Distanzmatrix mit dem Neighbor-Joining-Algorithmus.

4. Umwandlung in einen *rooted tree*. Als *outgroup* wird bei CSF das Isolat CSF0309 (Kanagawa) gewählt.
5. Berechnung der *Bootstrap*-Werte, falls sie im Baum mit angezeigt werden sollen.
6. Graphische Darstellung des *rooted tree*.

Der Baum kann als Datei im bmp-Format gespeichert werden. Ist die erste eingegebene Sequenz nicht korrekt, bricht das Programm ab. Sind eine oder mehrere der anderen Sequenzen falsch, wird eine Meldung angezeigt und die falschen Sequenzen werden nicht in die Berechnung einbezogen.

4.3.3 Zusätzliche Funktionen der administrativen Version

- Eine Möglichkeit, die Auswahl der Referenzsequenzen zu ändern, bietet die Option Referenz Sequenzen. Diese werden direkt in der Datenbank gespeichert. Die Ursprungsauswahl kann man mit dem Button Standard wieder herstellen.

- Die Option Direktvergleich berechnet für die erste eingegebene Sequenz einen Direktvergleich mit allen Sequenzen aus der Datenbank.

- Die Jukes-Cantor-Distanz der ersten eingegebene Sequenz mit allen Sequenzen aus der Datenbank berechnet die Option JC Distanzen.

- Die Funktionen SW und NW berechnen für die erste eingegebene Sequenz ein *Alignment* mit allen Sequenzen aus der Datenbank. Im Gegensatz zur öffentlichen Version muss der Benutzer der administrativen Version selber wählen, ob ein globales *Alignment* (Needleman-Wunsch-Algorithmus) oder ein lokales *Alignment* (Smith-Waterman-Algoritmus) berechnet wird.

- Bei der Option *Multiple Alignment* wird zusätzlich zu den Funktionen der öffentlichen Version eine Liste mit allen im Baum verwendeten Sequenzen angezeigt.

- *Update* berechnet die Felder *baseseq, duplicates* und *seqalign* neu.

- Alle Parameter für NW, SW, DC und *Bootstrapping* für die aktuelle Session kann man unter *parameter* ändern.

- Unter *own compare list* kann man die *Compare List* für NW, SW und DC für die aktuelle Session ändern.

- *Save* speichert den aktuellen Inhalt des Textfeldes, indem die alle Ergebnisse, bis auf den Baum, angezeigt werden, als Datei im txt-Format.

4.4 Beispiel CSF in Deutschland 2006

Nach dem letzten Ausbruch von Klassischer Schweinepest in Hausschweinen in Deutschland im Jahr 2003 wurde am 3. März 2006 in Haltern-Lavesum im Kreis Recklinghausen (Nordrhein-Westfalen) wieder ein Ausbruch der Klassischen Schweinepest festgestellt. Das Virus wurde zunächst in zwei Mastbetrieben (RE1 und RE2) nachgewiesen. In einem der beiden Betriebe waren seit Anfang Februar 72 von 329 Tiere eingegangen. Am 6. März 2006 wurde das Virus in einem dritten Betrieb (RE3) nachgewiesen. Der hohe Antikörpertiter gegen das CSF Virus in den Blutproben dieses Betriebes ließ vermuten, dass die Infektion bereits seit längerem in dem Bestand vorhanden war, und dass das Seuchengeschehen von diesem Betrieb ausging. Es handelte sich um einen Betrieb mit rund 40 Mastschweinen, eigener Hofschlachtung, Ladenlokal und Direktvermarktung. Das Virus wurde vermutlich durch Personenkontakt auf die anderen beiden Betriebe übertragen. Am 27. März 2006 wurde die Klassische Schweinepest in einem weiterem Betrieb in Haltern-Lavesum (RE4) festgestellt. Der betroffene Betrieb lag in der Nähe der anderen betroffenen Betriebe. Weitere Untersuchungen ergaben, dass es sich um keine Neuinfektion handelte, sondern dass das Virus bereits seit Ende Februar im Bestand war. Da die Infektion aber nicht die typischen Symptome zeigte, war sie bisher nicht erkannt worden. Am 31. März 2006 wurde das CSF Virus in einem fünften Betrieb in Uphusen (RE5) nachgewiesen. Dieser Betrieb lag im Sperrbezirk der anderen betroffenen Betriebe. Am 1. April 2006 wurde die Klassische Schweinepest auch in einem Bestand in Raesfeld (BOR1) im Kreis Borken festgestellt. Auch hier wurde das Virus bereits vor längerer Zeit eingeschleppt. Bei den Untersuchungen zur Aufhebung des Beobachtungsgebiets um BOR1, wurde am 5. Mai 2006 das CSF Virus in einem Betrieb in Borken-Marbeck (BOR2) nachgewiesen. Dieser Betrieb lag im drei Kilometer Sperrbezirk von BOR1. Am 9. Mai 2006 wurde das Virus in einem weiteren Betrieb in Borken-Gemenwirthe (BOR3) nachgewiesen. BOR3 lag im bisherigen Beobachtungsgebiet. Die geographische Lage der Ausbrüche ist in Abbildung 4.1 dargestellt. Dabei sind die Fälle 1 bis 4 identisch mit RE1 bis RE4, Fall 6 entspricht RE5 und die Fälle 5, 7 und 8 sind die Ausbrüche BOR1 bis BOR3.
Zur Bekämpfung der Klassischen Schweinepest wurden alle Tiere der betroffenen Betriebe, aller Betriebe im Umkreis von einem Kilometer und aller Kontaktbetriebe getötet. Um die betroffenen Betriebe wurden drei Kilometer Sperrbezirke und zehn Kilometer Beobachtungsgebiete eingerichtet. Das Beobachtungsgebiet von BOR3 reichte bis in die Niederlande. Im Beobachtungsgebiet von RE1 bis RE4 lagen ca. 340 Betriebe mit 140.000 Tieren. Zusätzlich wurden noch Pufferzonen von 20 km Radius um die Beobachtungsgebiete eingerichtet. Um die Lage dieser Gebiete zu verdeutlichen, sind die Gebiete in den Abbildungen 4.2 und 4.3 markiert. In den Sperrbezirken und den Beobachtungsgebieten galt ein absolutes "Stand Still" für alle Tiere. In ganz Nordrhein-Westfalen gab es Einschränkungen beim Transport der Tiere, zeitweise sogar ein vollständiges Transportverbot für Schweine.
Das absolute "Stand Still" hatte den Wertverlust schlachtreifer Tiere zur Folge, die zu einem späteren Zeitpunkt nur zu einem deutlich geringeren Preis vermarktet wer-

den konnten oder sogar die Tötung gesunder Tiere, weil sie zu groß für die Ställe wurden.
Nachdem es trotz dieser Maßnahmen zu weiteren Ausbrüchen kam, wurde die Tötung aller Tiere im drei Kilometer Sperrbezirk von der EU angeordnet. Der wirtschaftliche Schaden hierfür belief sich bei den rund 52.000 Schweinen in den Sperrbezirken von BOR1 und BOR2 auf rund 10 Millionen Euro.
Die acht betroffenen Betriebe hatten insgesamt einen Bestand von knapp 8000 Tieren. Zur Bekämpfung der Klassischen Schweinepest wurden im Kreis Borken ca. 92.000 Tiere aus 185 Betrieben getötet und im Kreis Recklinghausen ca. 22.000 Tiere aus 66 Betrieben [Ano06].

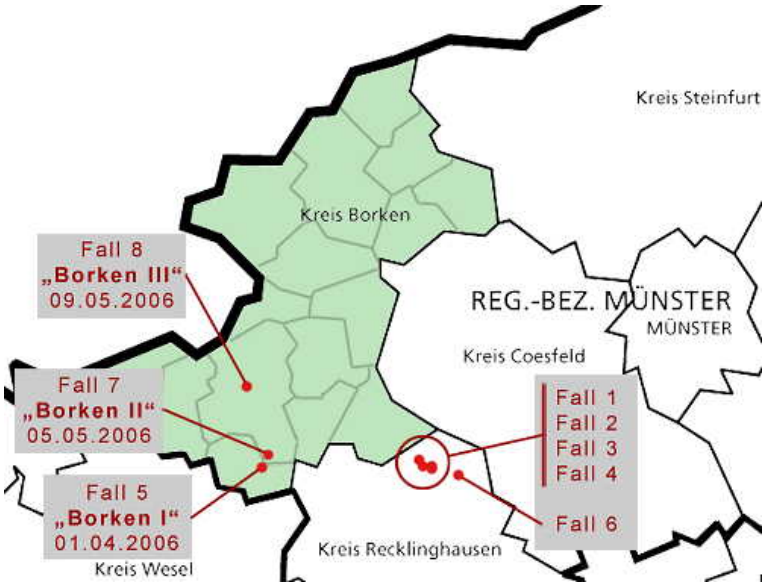


Abbildung 4.1: CSF-Ausbrüche in NRW 2006 [Ano06c].

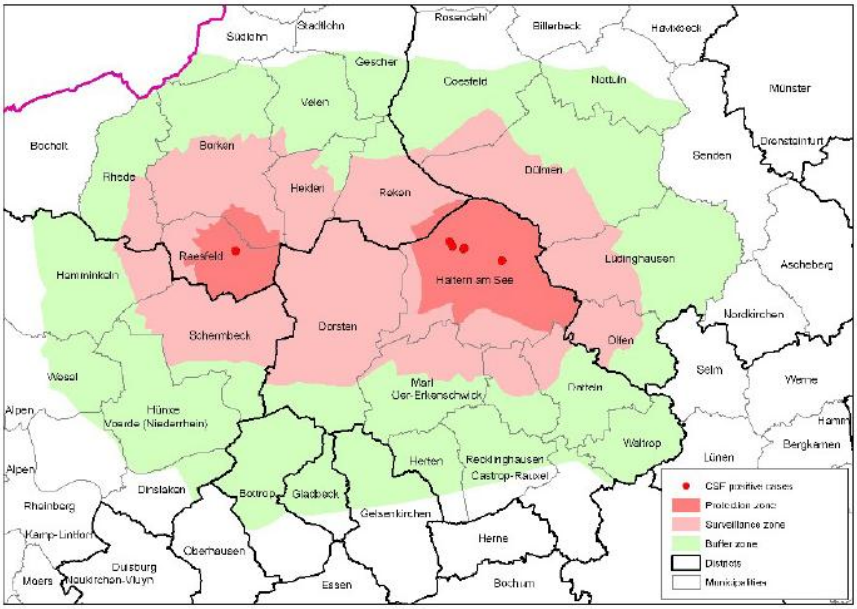


Abbildung 4.2: Sperrbezirke, Beobachtungsgebiete und Pufferzonen der Ausbrüche RE1 bis RE5 und BOR1 [Ano06c].

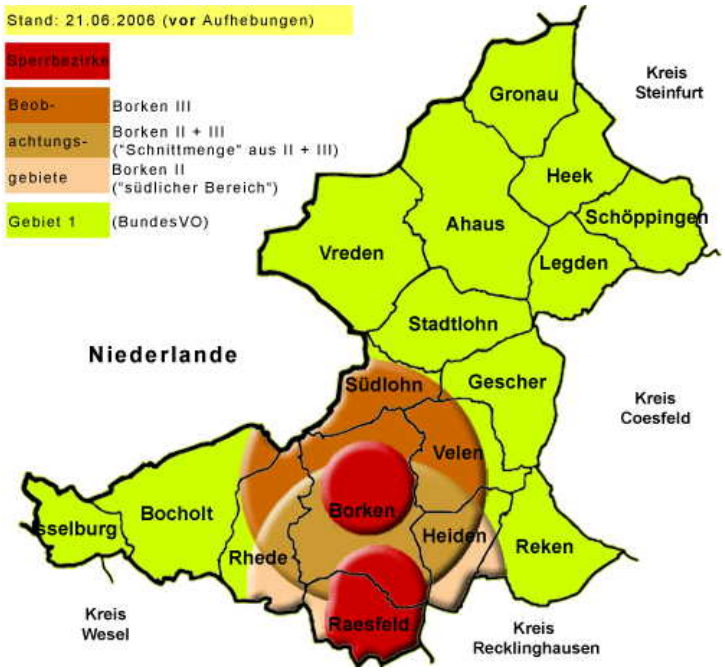


Abbildung 4.3: Sperrbezirke und Beobachtungsgebiete der CSF-Ausbrüche BOR2 und BOR3 [Ano06c].

Das Erregerisolat, bezeichnet mit 4364, wurde im 5'NTR und E2-Genfragment sequenziert. Ein Vergleich mit der CSFV-Datenbank ergab im 5'NTR eine Identität mit 26 Isolaten aus den Jahren 1994 bis 1998. Vier Isolate stammten aus Niedersachsen, vier aus Brandenburg und 18 aus Mecklenburg-Vorpommern. 12 Isolate stammten aus Wildschweinen, 14 Isolate aus Hausschweinen. Alle Isolate gehörten der Subgruppe 2.3-Guestrow an. Die E2-Sequenz war mit der Sequenz eines Isolats identisch, das 1997 in Mecklenburg-Vorpommern aus einem Wildschwein isoliert wurde. Eine phylogenetische Analyse ergab die in Abbildung (4.4) und (4.5) gezeigten Bäume, in denen man auch erkennt, dass das Isolat 4364 der Subgruppe 2.3-Guestrow angehört.

Da das Erregerisolat zu der für Osteuropa typischen Subgruppe 2.3-Guestrow gehörte, konnte man mit hoher Sicherheit ausschließen, dass der Erreger durch Wildschweine übertragen wurde [Ano06].

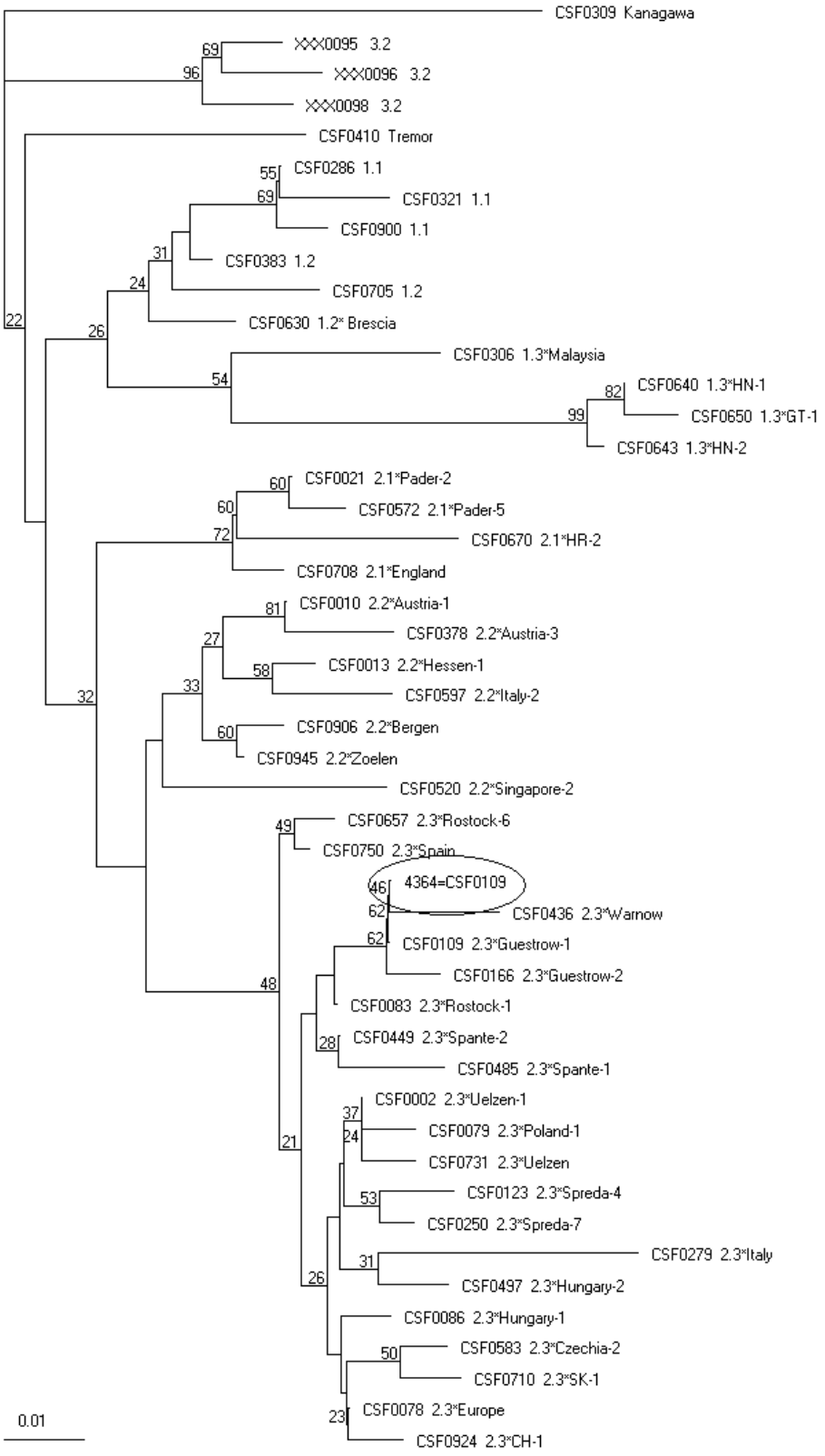


Abbildung 4.4: Phylogenetischer Baum mit dem Isolat 4364, berechnet mit den 5'NTR-Sequenzen.

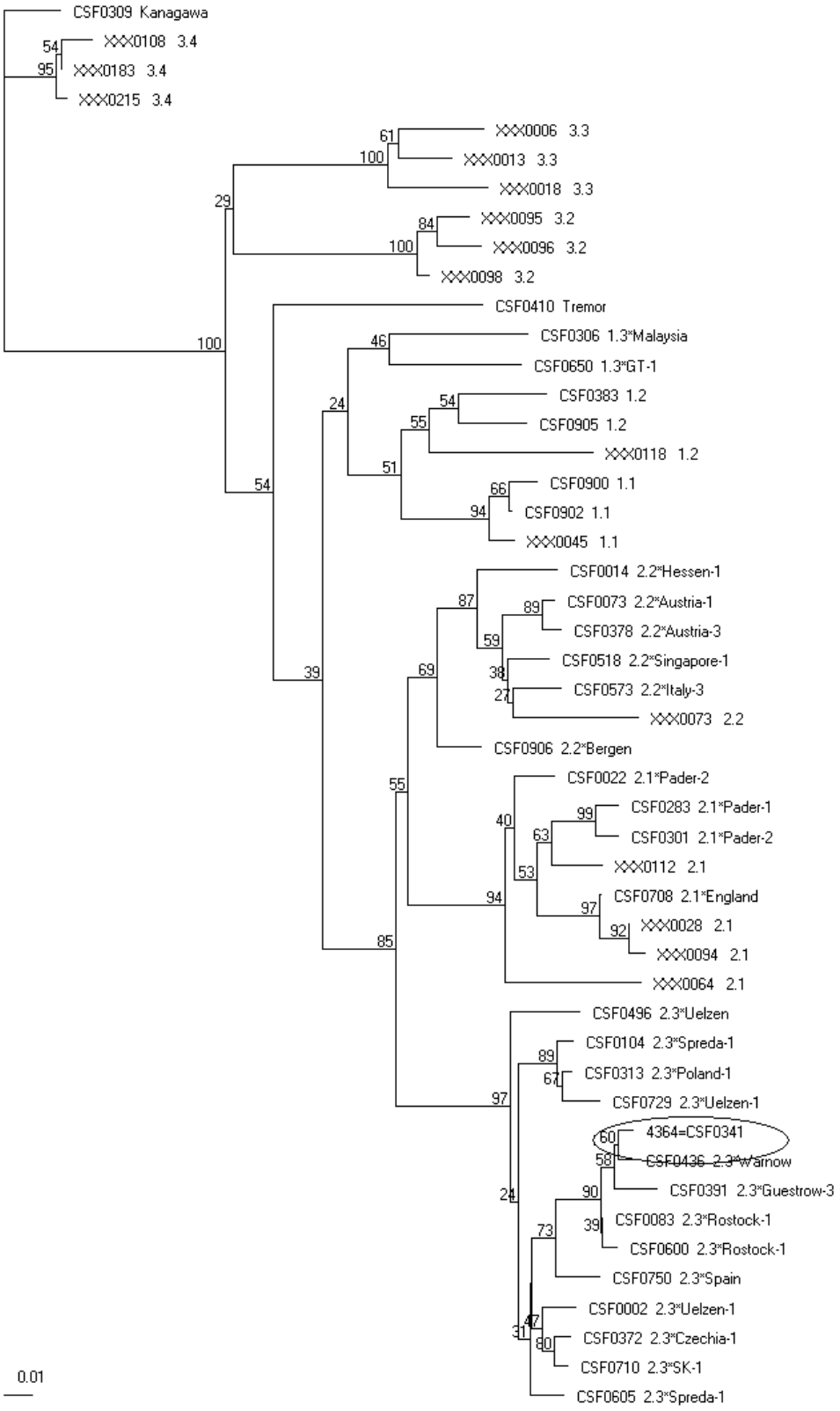


Abbildung 4.5: Phylogenetischer Baum mit dem Isolat 4364, berechnet aus E2-Genfragment-Sequenzen.

Kapitel 5

Genotypisierung von BVD und Border disease Viren

Das BVDV und das BDV wurden gemeinsam betrachtet, weil sie genetisch und antigenisch sehr eng mit einander verwandt und im Gegensatz zu dem CSFV, das ebenfalls eng verwandt ist, Paarhufer und Schweine infizieren können, während das CSFV auf natürlichem Wege nur Schweine infizieren kann. Um das Seuchengeschehen nachvollziehen zu können und so die Bekämpfung zu vereinfachen, wurde eine Datenbank und eine standardisierte Genotypisierung erstellt.

5.1 BVDV/BDV-Datenbank

Analog zur CSFV-Datenbank aus Kapitel 4 wurde auch für die BVDV und die BDV Isolate eine Datenbank erstellt. Hierzu wurden Genfragmente der 5'NTR und der E2- und Npro-Gene gewählt, weil diese am häufigsten zur Genotypisierung verwendet werden [VD05]. Als Referenzsequenz diente das Isolat NADL (NC_001461). Für die Datenbank wurden die folgenden Fragmente der genomischen Regionen verwendet:

- 5'NTR: 144.-354. Base der Sequenz des BVDV-Stammes NADL (211 Basen)
- E2: 2879.-3148. Base der Sequenz des BVDV-Stammes NADL (270 Basen)
- Npro: 386.-775. Base der Sequenz des BVDV-Stammes NADL (390 Basen)

Die Auswahl der Fragmente erfolgte mittels eines Sequenzvergleiches aller in GenBank verfügbaren Sequenzen. Das Ziel war möglichst viele Sequenzen mit einem möglichst langem Teilstück zu bekommen. Die Datenbank beinhaltet alle zum jetzigen Zeitpunkt in der GenBank veröffentlichten Sequenzen, für die mindestens eines der Teilstücke vollständig vorhanden ist. Insgesamt ergeben sich so 890 5'NTR-, 237 E2- und 232 Npro-Sequenzen.
Wie bei der CSFV-Datenbank gibt es zwei Versionen: ein öffentliche (*user* bvd) und

eine administrative (*user* bvdmaster, mit Schreibrechten). Erreichbar ist die Datenbank auf der Internetseite:
http://viro08.tiho-hannover.de/eg/bvdv/startBVDV.cgi
Die Basistabelle der Datenbank enthält die Felder:

- PES Nummer
- *Original name*: Name des Virusisolates aus der Veröffentlichung.
- *Year of isolation*: Jahr in dem das Virus isoliert wurde.
- *Country*: Land in dem das Virus isoliert wurde.
- *Region*: Region in dem das Virus isoliert wurde.
- *Host species*: Tierart bei dem das Virus isoliert wurde.
- *Species*: BVDV1, BVDV2 oder BDV.
- *Biotype*: cp oder ncp.
- EMBL-Acc: EMBL Nummer, unter der das Isolat bei GenBank gespeichert ist.
- *Reference*: Zitat der Publikation, in der die Sequenz veröffentlicht wurde.
- 5'NTR, E2, Npro: Vorhandene Sequenzen für das jeweilige Isolat.

Unter dem administrativen Zugang sind zusätzlich die folgenden Felder verfügbar:

- *Details*: enthält nähere Informationen über die Isolierung.
- *public*: mit *public* gekennzeichnete Isolate sind auch in der öffentlichen Version lesbar.
- *OnStock*: für alle hier gekennzeichneten Isolate ist das Virus im Institut vorhanden.

Außerdem besteht die Möglichkeit, nach einigen Faktoren (z.B. PES Nummer, Originalname oder EMBL-Acc.) zu suchen.
Ein Ziel der BVDV/BDV-Datenbank ist eine einheitliche Einteilung in Spezies, Genotypen und Subgruppen zu ermöglichen. Weiterhin sollte die Genotypisierung neuer Isolate automatisiert erfolgen. Aufgrund der großen Anzahl an Isolaten ergibt sich auch hier die gleiche Problematik wie bei CSFV. Eine manuelle Auswahl der Vergleichsisolate ist nicht möglich. Um diese zu lösen, wurde das Modul aus Kapitel 4 erweitert und war so auch für die BVDV/BDV-Datenbank verwendbar. Die Vergleichssequenz Alfort wurde für BVDV/BDV durch NADL ersetzt und die *Outgroup* Kanagawa durch das Isolat giraffe-1 H138 (NC_003678). Dies ist ein *Pestivirus*-Isolat, dass aus einer Giraffe aus Kenia stammt und sich weder BVDV noch BDV zuordnen

läßt [FJ05]. In die Bedienoberfläche der BVDV Datenbank soll in Zukunft ebenfalls das *Gentic Typing*-Modul eingebaut werden, das dann dem Benutzer die gleichen Anwendungen wie bei CSFV zur Verfügung stellt.

5.2 Genotypisierung von BVDV und BDV Isolaten

Hauptgrundlage für eine automatische Genotypisierung ist eine standardisierte Einteilung in Genotypen. Im Gegensatz zu CSFV ist diese für BVDV und BDV nicht einheitlich. Nach ICTV (International Committee on Taxonomy of Viruses) sind BVDV-1, BVDV-2 und BDV eigene Spezies. Diese Spezies lassen sich in Genotypen unterteilen. Teilweise wird BVDV allerdings als eigene Spezies behandelt und in die Genotypen BVDV-1 und BVDV-2 unterteilt [VD05]. Für BVDV-1 existiert eine Einteilung in die Gruppen a-k [VP01], für BVDV-2 in die Gruppen a-b [VD04]. Für BDV sind neun Gruppen beschrieben. Im einzelnen sind diese in Tabelle 5.1 aufgeführt. Da es keine einheitliche Bezeichnung für die BDV Gruppen gibt, wurden sie für die Auswertung zunächst mit BDV a bis BDV j bezeichnet.
Die bisherige Taxonomie ist nicht eindeutig. Aus den bisherigen Publikationen geht nicht eindeutig hervor, um was es sich bei den Gruppen handelt. Für BVDV-1 und BVDV-2 erfolgte die Einteilung entweder in Genotypen, mit BVDV-1 und BVDV-2 als eigene Spezies, oder in Subgruppen, wenn BVDV-1 und BVDV-2 als Genotypen der Spezies BVDV bezeichnet wurden. Für BDV kann es sich bei den Gruppen um jeweils eigene Spezies oder um Genotypen, der Spezies BDV, handeln.
Alle in der BVDV/BDV-Datenbank enthaltenen Isolate wurden mit Hilfe phylogenetischer Analysen diesen Gruppen zugeordnet. Isolate, die sich in keine dieser Gruppen einordnen ließen, wurden einer neuen Gruppe zugeordnet, wenn Sequenzen von mindestens zwei Genfragmenten vorhanden waren.
Isolate, bei denen nur die Sequenz eines der drei Genfragmente vorhanden war und die sich anhand dieser Sequenz nicht in die vorhandenen Gruppen einordnen ließen, wurden nicht weiter berücksichtigt. Durch das Fehlen der anderen Sequenzen war keine Möglichkeit vorhanden, um zu überprüfen, ob es sich wirklich um eine neue Gruppe handelt.
Insgesamt wurden drei neue Gruppen für BVDV-1 (N1, N8 und N9) und eine neue Gruppe für BVDV-2 (N1) definiert.
Die Gruppe **BVDV-1 N1** beinhaltet drei Isolate aus Japan. Es sind drei 5'NTR-, drei E2- und eine Npro-Sequenz vorhanden.
Sechs Isolate lassen sich der Gruppe **BVDV-1 N8** zuordnen. Die Isolate stammten, soweit die Herkunft bekannt ist, aus Deutschland. Es sind jeweils drei 5'NTR-, E2- und Npro-Sequenzen vorhanden. Zwei der Isolate wurden von M. Tajima zusammen in einer Gruppe mit einem Isolat aus BVDV-1 g veröffentlicht, jedoch in zwei verschiedenen Ästen [TF01].
Gruppe **BVDV-1 N9** besteht nur aus dem Isolat 22148/81 (PES0479 bzw. PES0529) von M. Tajima [TF01]. Eine Npro-Sequenz des Isolates ist nicht vorhanden.

Name	Neu: BDV	Referenz	Isolate	Bemerkungen
BDV-1	a + b	[BA03]	z.B. X818, L83/84, V2536	
BDV-2	c	[BA03]	z.B. V60, 17385	
BDV-3	d	[BA03]	z.B. Gifhorn	
BDV a	a	[VN97]	z.B. V2536	Zusammen mit BDV b identisch mit BDV-1
BDV b	b	[VN97]	z.B. X818, L83/84	Zusammen mit BDV a identisch mit BDV-1
BDV-c	f	[HG03]	z.B. 2112/99, 80582/01	Es sind nur 5'NTR Sequenzen vorhanden.
BDV-4	g	[AF04]	Chamois-1	Isolat stammt von einer Gams aus den spanischen Pyrenäen
–	h	[MG05]	712/02	keine Zuordnung zu einer bisherigen Gruppe möglich
BDV-4a	i	[VA06]	z.B. Colm24	Es sind nur 5'NTR Sequenzen vorhanden. In [VA06] beinhaltet BDV-4a auch die Gruppe BDV-c und das Isolat Chamois
BDV-4b	j	[VA06]	z.B. Rocco	Es sind nur 5'NTR Sequenzen vorhanden.

Tabelle 5.1: Übersicht der publizierten Border disease Virus Gruppen.

Der 5'NTR und der E2 Baum legten die Vermutung nahe, dass BVDV-1 N1, BVDV-1 N8 und BVDV-1 N9 eventuell eine gemeinsame Gruppe bilden. Im Npro Baum waren die Gruppen BVDV-1 N1 und BVDV-1 N8 (für BVDV-1 N9 sind keine Isolate vorhanden) allerdings nicht benachbart.
Die einzige neue BVDV-2 Gruppe **BVDV-2 N1** beinhaltet nur das Isolat BVDV2-28508-5 (AF145968). Die Herkunft des Isolates ist nicht bekannt. Es sind die E2 und die Npro-Sequenz vorhanden.
Im Gegensatz zu BVDV ließen sich alle BDV Isolate in die publizierten Gruppen einordnen.
Für alle drei Genfragmente wurden Referenzbäume erstellt. Wie bei CSFV wurden die Isolate so gewählt, dass alle Gruppen vorkommen und der Baum nicht zu groß ist. Es ergaben sich für BVDV und BDV die in Abbildung (5.1 - 5.3) dargestellten Bäume. Die drei phylogenetischen Bäume (Abb. 5.1 bis 5.3) zeigten eine eindeutige

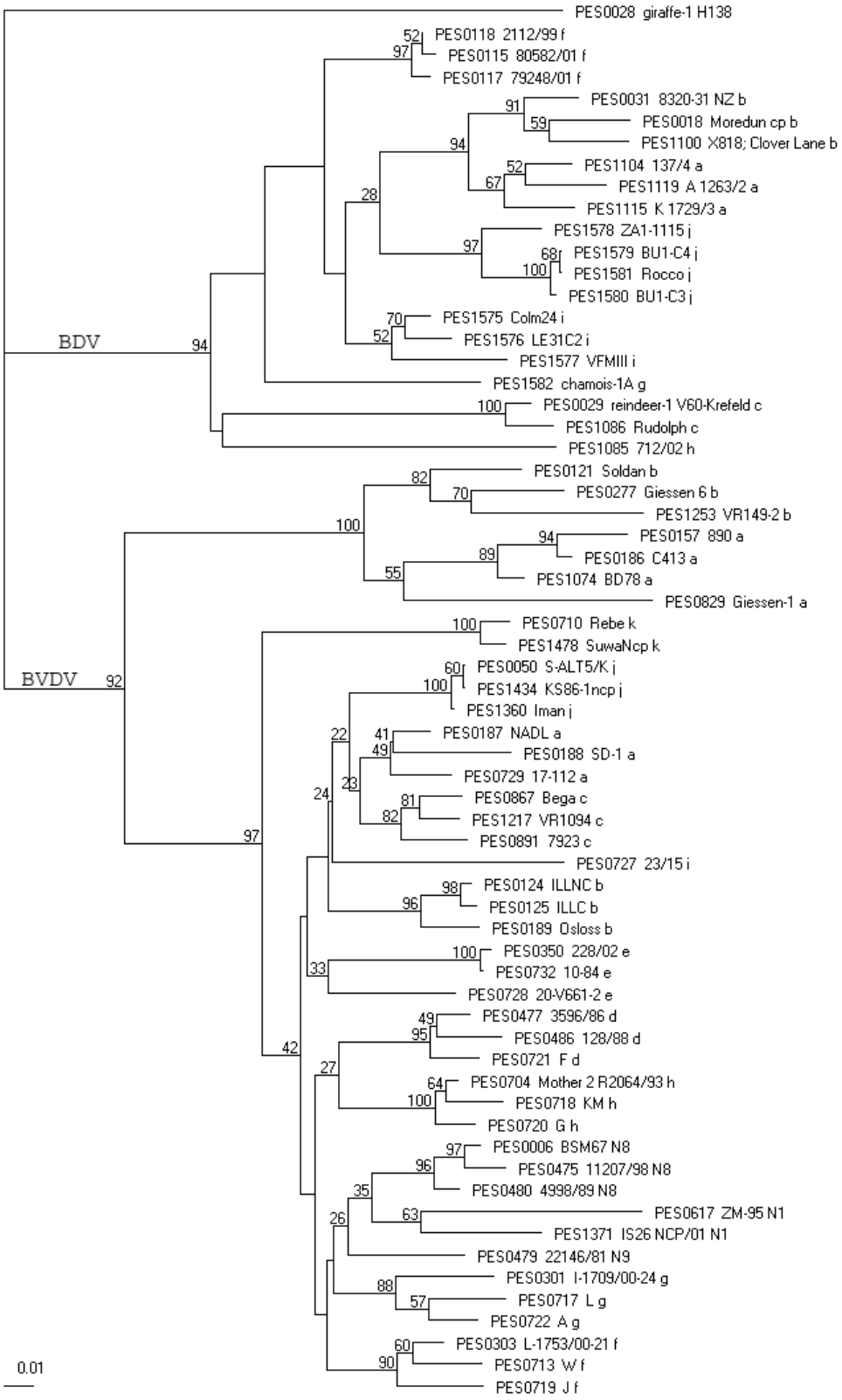


Abbildung 5.1: Referenzbäume mit der Einteilung der BVDV und BDV, berechnet aus 5'NTR-Sequenzen. Die PES Nummern entsprechen der Nummerierung aus der BVDV/BDV-Datenbank, die folgende Bezeichnung steht für die Originalnamen und die Einteilung in die Gruppen (a-k, bzw. N1, N8 und N9).

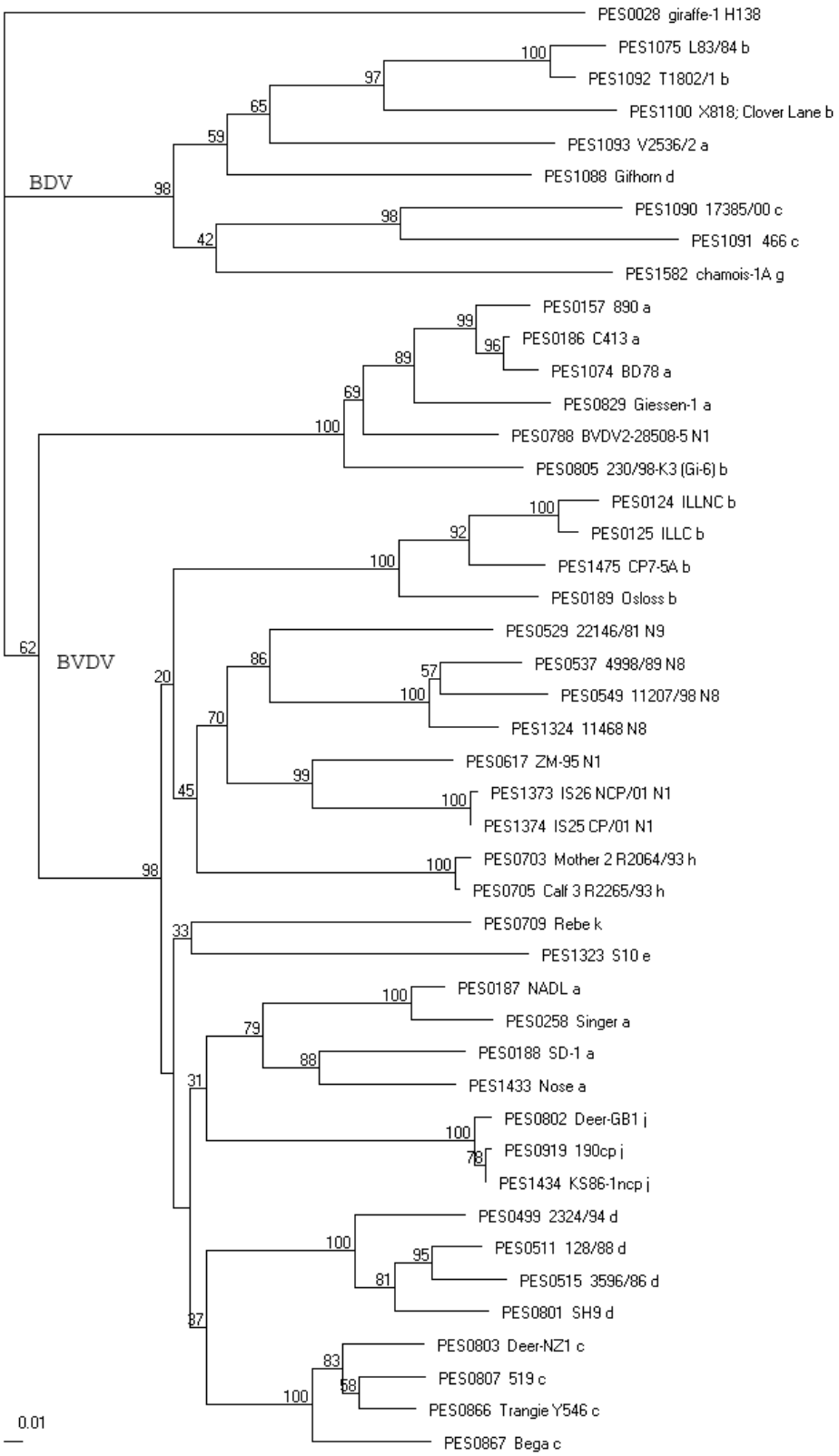


Abbildung 5.2: Referenzbäume mit der Einteilung der BVDV und BDV, berechnet aus E2-Sequenzen. Die PES Nummern entsprechen der Nummerierung aus der BVDV/BDV-Datenbank, die folgende Bezeichnung steht für die Originalnamen und die Einteilung in die Gruppen (a-k, bzw. N1, N8 und N9).

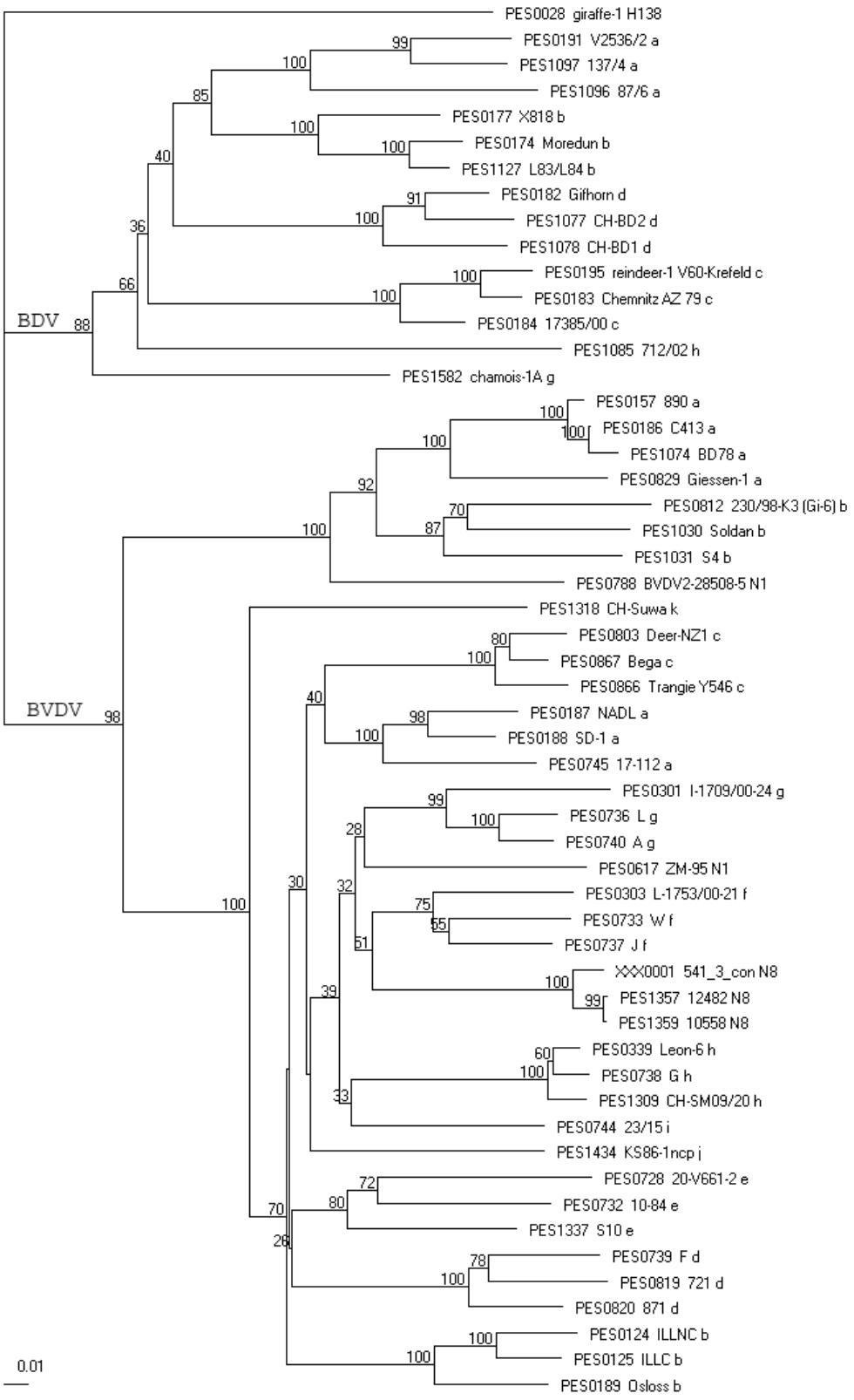


Abbildung 5.3: Referenzbäume mit der Einteilung der BVDV und BDV, berechnet aus Npro-Sequenzen. Die PES Nummern entsprechen der Nummerierung aus der BVDV/BDV-Datenbank, die folgende Bezeichnung steht für die Originalnamen und die Einteilung in die Gruppen (a-k, bzw. N1, N8 und N9).

Unterteilung von BVDV-1 in die Gruppen a-k, N1, N8 und N9, von BVDV-2 in die Gruppen a, b und N1 und von BDV in die Gruppen a-d und f-j. Es sind aber nicht alle Gruppen in allen Genfragmenten vertreten.
Die Anordnung der Gruppen in den verschiedenen Bäumen zeigt, dass ein Zusammenfassen einzelner Gruppen nur für BDV a und BDV b möglich ist. Eventuell kann man auch die die Gruppe BDV j mit BDV a und BDV b zusammenfassen. Bei allen anderen Gruppen ist dies nicht möglich, d.h. für BVDV-1 und BVDV-2 erfolgt eine Einteilung entweder in alle bisher genannten Gruppen oder in keine.

5.2.1 Bootstrap-Werte

Bei der Berechnung der drei phylogenetischen Bäume (Abb. 5.1 bis 5.3) wurden auch die *Bootstrap*-Werte berechnet. Diese sind in Tabelle 5.2 dargestellt. Die niedrigen Werte für 5'NTR zeigen, dass die Berechnung aus den 5'NTR-Sequenzen einen sehr instabilen Baum ergibt. Insbesondere die großen Gruppen wie BVDV-1 a und e haben sehr niedrige *Bootstrap*-Werte (49% bzw. 33%). Bei E2 sind nicht für alle Gruppen Werte vorhanden, weil nicht genug Isolate vorhanden sind. Die vorhandenen Werte weisen aber auf eine Unterstützung der Gruppierung hin. Auch bei Npro zeigen die *Bootstrap*-Werte, die zwischen 75% und 100% liegen, eine Unterstützung der Gruppierung an.
Die Auswertungen der *Bootstrap*-Werte ergibt, dass eine stabile Einteilung in die Gruppen nur im E2 und Npro möglich ist.

	5'NTR	E2	Npro
BVDV-1	33%-100%	79%-100%	75%-100%
BVDV-2	55%-82%	89%-100%	87%-100%
BDV	52%-100%	98%-100%	100%

Tabelle 5.2: *Bootstrap*-Werte der einzelnen Gruppen von BVDV-1, BVDV-2 und BDV in den drei verschiedenen Fragmenten. Berechnet aus den phylogenetischen Bäumen (Abb. 5.1 bis 5.3).

5.2.2 Regionale Auswertung

Wenn zwischen der geographischen Herkunft der Isolate und der Einteilung in Gruppen ein Zusammenhang besteht, kann anhand der Genotypisierung die Virusausbreitung nachvollzogen werden. Die Ergebnisse aus [VD05] deuten darauf hin, dass es für BVDV keinen Zusammenhang zwischen der Gruppenverteilung und der geographischen Herkunft der Isolate gibt. Um diese Vermutung zu bestätigen oder zu widerlegen, wurden alle Isolate, von denen zumindestens das Herkunftsland bekannt ist, in der BVDV/BDV-Datenbank ausgewertet.
Für jede Gruppe wurden die Länder, aus denen Isolate dieser Gruppe existierten,

unterteilt nach Kontinenten mit den zugehörigen Jahreszahlen aufgelistet. DE (64, 90-95) bedeutet z.B., das Isolate dieser Gruppe aus Deutschland aus dem Jahr 1964 und den Jahren 1990 bis 1995 existieren. Die Länderkürzel sind in Anhang B aufgelistet. Dies ergab die folgenden Ergebnisse:

Gruppe BVDV-1 a: (> 100 Isolate) - weltweite Ausbreitung

- Afrika: ZA (90-93, 95).
- Asien: JP (67, 74, 78, 86, 87, 89, 91, 92, 94-99,03), IR.
- Australien: AU (99), NZ (67, 75, 80, 83, 88, 89, 94-97).
- Europa: DE (82, 96), FR (94), GB (76, 77, 80, 82-86, 88-90, 93-96, 99), IE (98), PT (03), SE (02-04).
- Nordamerika: CA, US (60).
- Südamerika: AR (88-90, 93, 94, 97), BR, CL (01).

Gruppe BVDV-1 b: (> 100 Isolate) - weltweite Ausbreitung

- Afrika: ZA (92, 93, 95, 96).
- Asien: IN, JP (74, 82, 84-86, 91-93, 95, 96, 98-00).
- Australien: AU (99).
- Europa: AT (98), CH, DE (60, 80, 83-92, 95-00), DK (02,03), ES (89, 92, 93, 97-02), GB (96), IT (95-99), PT (03,04), SE (89, 90, 92, 02-04), SI (99, 00).
- Nordamerika: CA, US.
- Südamerika: AR (84, 89, 90, 99), CL (95, 96).

Gruppe BVDV-1 c: (> 100 Isolate) - weltweite Ausbreitung

- Asien: JP (97, 99).
- Australien: AU (78, 89, 94, 95, 97-00), NZ (80, 92, 97).
- Europa: DE (93), ES (89, 00), PT (03).

Gruppe BVDV-1 d: (> 100 Isolate) - Ausbreitung nur in Europa

- Europa: AT (98), DE (60, 83, 84, 86, 88, 90-94, 96, 98), DK (62,93,02,03), FR (94), IT (95, 99), PT (03), SE (02-04), SI (97-00).

Gruppe BVDV-1 e: (51 Isolate) - weltweite Ausbreitung

- Asien: JP (75).
- Europa: CH, DE (99), DK (02), ES (00-02), FR (94, 96, 98), IT (95-97, 99, 00), PT (03).
- Südamerika: AR.

Gruppe BVDV-1 f: (33 Isolate) - Ausbreitung nur in Afrika und Europa

- Afrika: ZA (94-96).
- Europa: AT (98), IT (99), SI (99-01) .

Gruppe BVDV-1 g: (8 Isolate) - Ausbreitung nur in Afrika und Europa

- Afrika: ZA (95, 96).
- Europa: AT (98), SI (00) .

Gruppe BVDV-1 h: (15 Isolate) - Ausbreitung nur in Afrika und Europa

- Afrika: ZA (95).
- Europa: AT (98), CH, DE, ES (00), IT (96, 99), SK (98).

Gruppe BVDV-1 i: (1 Isolat) - Ausbreitung nur in Europa

- Europa: GB (97).

Gruppe BVDV-1 j: (34 Isolate) - weltweite Ausbreitung

- Afrika: EG, ZA (90-93).
- Asien: JP (86, 87).
- Europa: GB (86), IT (95).
- Südamerika: AR (90, 91), BR, CL (96).

Gruppe BVDV-1 k: (5 Isolate) - weltweite Ausbreitung

- Asien: JP (75).
- Europa: CH.

Gruppe BVDV-1 N1: (3 Isolate) - Ausbreitung nur in Asien

- Asien: JP (01).

Gruppe BVDV-1 N8: (7 Isolate) - Ausbreitung nur in Europa

- Europa: DE (89, 98).

Gruppe BVDV-1 N9: (3 Isolate) - Ausbreitung nur in Europa

- Europa: DE (81).

Gruppe BVDV-2 a: (> 100 Isolate) - weltweite Ausbreitung

- Afrika: TN.
- Asien: JP (89-92, 97).
- Australien: NZ (97)
- Europa: DE (87, 93-98, 02), FR (94), GB (85, 87), PT (02).
- Nordamerika: US (60).
- Südamerika: AR (88-90, 93, 94, 97), BR, CL (95, 96, 98).

Gruppe BVDV-2 b: (21 Isolate) - weltweite Ausbreitung

- Afrika: TN.
- Australien: AU (99)
- Europa: PT (97,01,02).
- Südamerika: BR (92), PA (97).

Gruppe BDV a: (16 Isolate) - Ausbreitung nur in Europa

- Europa: GB (80, 82, 86, 87, 90, 93, 94).

Gruppe BDV b: (30 Isolate) - Ausbreitung nur in Australien und Europa

- Europa: GB (76, 80, 85, 88, 90, 92, 93).
- Australien: AU (73), NZ (90, 91, 94)

Gruppe BDV c: (6 Isolate) - Ausbreitung nur in Europa

- Europa: DE (85, 96, 99, 00).

Gruppe BDV d: (3 Isolate) - Ausbreitung nur in Europa

- Europa: CH, DE (99).

Gruppe BDV f: (4 Isolate) - Ausbreitung nur in Europa

- Europa: ES (99, 01).

Gruppe BDV g: (1 Isolat) - Ausbreitung nur in Europa

- Europa: ES (02), FR.

Gruppe BDV h: (1 Isolat) - Ausbreitung nur in Europa

- Europa: IT (02).

Gruppe BDV i: (3 Isolate) - Ausbreitung nur in Europa

- Europa: ES (01, 02).

Gruppe BDV j: (4 Isolate) - Ausbreitung nur in Europa

- Europa: ES (02).

Insgesamt ist für das Auftreten bestimmter BVDV Gruppen keine Korrelation mit der geographischen Verteilung erkennbar. Fast alle Gruppen treten nahezu zeitgleich auf verschiedenen Kontinenten auf. Die Gruppen, die sich auf eine Region beschränken, wie z.B. BVDV-1 N8, sind mit einer Anzahl von < 10 Isolaten nicht aussagekräftig.
Im Gegensatz zu BVDV scheint es bei BDV einen Zusammenhang zwischen Epidemiologie und Gruppeneinteilung zu geben, allerdings ist die Anzahl der Isolate pro Gruppe sehr gering und es sind fast nur Isolate aus Europa vorhanden. Es wird vermutet, dass eine höhere Anzahl von Isolaten zu einer Aufhebung der geographischen Zuordnung führt.
Um die hier aufgestellten Thesen zu bestätigen oder zu widerlegen, ist ein größerer Datenumfang nötig. Es müßten aus allen Gruppen genügend und ähnlich viele Isolate vorhanden sein. Weiterhin müßten aus den verschiedenen Regionen vergleichbar viele Isolate vorhanden sein, als auch aus verschiedenen Zeitabschnitten.

5.2.3 Sequenzunterschiede im gesamten Genom und im Npro Genfragment

Häufig wird die Einteilung in verschiedene Gruppen anhand von Unterschieden in den Sequenzen definiert. Beim Hepatitis B Virus z.B. bedeutet mehr als 8% Unterschied im gesamten Genom die Zuordnung in verschiedene Gruppen [AN02]. Das

ICTV definiert Isolate als verschiedene Spezies, falls u.a. im gesamten Genom der Unterschied mindestens 25% ist. Als Vertreter des Genus *Pestivirus* sind von ICTV für BDV die Isolate BD31 und X818, für BVDV-1 die Isolate NADL, Osloss, SD-1 und CP7, für BVDV-2 die Isolate *strain* 890 und C413 und für CSFV die Isolate Alfort/187, Alfort-Tübingen, Brescia und C Stamm beschrieben. Für diese Isolate wurden in Tabelle 5.3 die Übereinstimmungen im gesamten Genom berechnet und in Tabelle 5.4 wurden zum Vergleich die Übereinstimmungen in dem in der BVDV/BDV-Datenbank gespeicherten Npro-Fragment für BDV, BVDV-1 und BVDV-2 berechnet. Hier wurde nur das Npro-Fragment betrachtet, weil beim E2-Fragment zu wenig Sequenzen vorhanden waren und die 5'NTR, aufgrund der hohen Konservierung, wenig aussagt.

	BDV	BVDV-1	BVDV-2	CSFV
BDV	89%-100%	65%-67%	66%-67%	71%-72%
BVDV-1		77%-100%	67%-69%	66%-67%
BVDV-2			94%-100%	65%-87%
CSFV				86%-100%

Tabelle 5.3: Identitäten im gesamten Genom.

	BDV	BVDV-1	BVDV-2
BDV	87%-100%	65%-71%	63%-67%
BVDV-1		82%-100%	69%-72%
BVDV-2			98%-100%

Tabelle 5.4: Identitäten im BVDV/BDV-Datenbank Npro-Fragment (390 Basen).

Die Werte aus Tabelle 5.3 zeigen deutliche Unterschiede zwischen BDV, BVDV-1, BVDV-2 und CSFV. Innerhalb der Spezies liegen die Werte zwischen 77% und 100%. Vergleicht man dagegen Isolate verschiedener Spezies, liegen die Werte zwischen 65% und 72%. Insbesondere der Vergleich zwischen BVDV-1 und BVDV-2 zeigt, dass die Sequenzen nur zu 67% bis 69% identisch sind. Damit ist BVDV-1 von BVDV-2 ähnlich weit entfernt wie von BDV oder CSFV. Das gilt analog auch für BVDV-2. Auch bei dem Vergleich der Npro-Fragmente in Tabelle 5.4 ist ein deutlicher Unterschied zwischen den Identitäten der BDV, BVDV-1 und BVDV-2 Sequenzen zu erkennen.
Auch von Interesse ist, ob man die einzelnen Gruppen anhand von Sequenzunterschieden erkennen kann. Hierfür wurden in den Tabellen 5.5 bis 5.7 die Identitäten aller in der BVDV/BDV-Datenbank vorhandenen Npro-Fragmente berechnet.

Die Tabellen 5.5 und 5.6 zeigen, dass eine Unterteilung in Gruppen nicht durch Unterschiede in den Sequenzen möglich ist. Für BDV ist eine Unterteilung anhand von Sequenzunterschieden möglich, allerdings sind auch hier die Übergänge sehr eng.

	innerhalb der Gruppe	mit den anderen BVDV-1 Gruppen
BVDV-1 a	85%-100%	75%-85%
BVDV-1 b	83%-100%	75%-86%
BVDV-1 c	88%-100%	75%-85%
BVDV-1 d	88%-100%	75%-82%
BVDV-1 e	80%-100%	75%-85%
BVDV-1 f	88%-100%	76%-87%
BVDV-1 g	89%-100%	76%-86%
BVDV-1 h	96%-100%	74%-85%
BVDV-1 i	100%	77%-86%
BVDV-1 j	100%	77%-85%
BVDV-1 k	94%-100%	74%-82%
BVDV-1 N1	100%	75%-86%
BVDV-1 N8	99%-100%	74%-87%

Tabelle 5.5: Identitäten der, in der BVDV/BDV-Datenbank gespeicherten Sequenzen der Npro-Fragmente innerhalb der BVDV-1 Gruppen und im Vergleich mit den anderen BVDV-1 Gruppen.

Bei BDV a ist innerhalb der Gruppe die Übereinstimmung zwischen 83%-100%, im Vergleich mit den anderen Gruppen beträgt sie 71%-82%. Es ist zu vermuten, dass bei einer größeren Anzahl an Isolaten (bisher 25 BDV Npro-Sequenzen) auch keine Unterscheidung mehr möglich ist. Eine weitere Möglichkeit, die Gruppen anhand von Sequenzunterschieden zu definieren, wäre ein Sequenzvergleich des gesamten Genoms. Hierfür aber sind nicht genügend Isolate vollständig sequenziert.

	innerhalb der Gruppe	mit den anderen BVDV-2 Gruppen
BVDV-2 a	86%-100%	78%-85%
BVDV-2 b	100%	78%-85%
BVDV-1 N1	84%-100%	80%-85%

Tabelle 5.6: Identitäten der, in der BVDV/BDV-Datenbank gespeicherten Sequenzen der Npro-Fragmente innerhalb der BVDV-2 Gruppen und im Vergleich mit den anderen BVDV-2 Gruppen.

	innerhalb der Gruppe	mit den anderen BDV Gruppen
BDV a	83%-100%	71%-82%
BDV b	87%-100%	70%-82%
BDV c	94%-100%	72%-78%
BDV d	90%-100%	72%-79%
BDV g	100%	73%-76%
BDV h	100%	70%-74%

Tabelle 5.7: Identitäten der, in der BVDV/BDV-Datenbank gespeicherten Sequenzen der Npro-Fragmente innerhalb der BDV Gruppen und im Vergleich mit den anderen BDV Gruppen.

5.2.4 Fazit

Nach den bisher veröffentlichten Einteilungen sind zwei Aspekte nicht eindeutig geklärt:

- Handelt es sich bei um Spezies, Genotypen oder Subgruppen?
- Ist eine weitere Unterteilung von BVDV-1 und BVDV-2 überhaupt sinnvoll?

Die Ergebnisse zeigen, dass die Einteilung von BVDV in zwei Spezies (ICTV) nicht korrekt ist. Nur die Ergebnisse des Sequenzvergleiches sprechen für die Einteilung in zwei Spezies, alle anderen Ergebnisse sprechen jedoch dagegen.
Ebenfalls für die Einteilung in Genotypen spricht, dass BVDV-1 und BVDV-2 bei einer Infektion die gleiche klinische Symptomatik erzeugen. Insgesamt kann man davon ausgehen, dass BVDV-1 und BVDV-2 zwei Genotypen der Spezies BVDV sind.
Eine Unterteilung der Genotypen BVDV-1 und BVDV-2 in Subgruppen lässt sich nur anhand der phylogenetischen Bäume definieren. Sowohl die Sequenzvergleiche als auch die Auswertung epidemiologischer Daten haben keine Unterschiede bezüglich der Subgruppen gezeigt. Aus der Anordnung der Gruppen mit den Sequenzen der Fragmente läßt sich nur sagen, dass entweder alle Subgruppen erhalten bleiben oder keine. Da aber die Subgruppen auch bei dem 5'NTR Baum, der ziemlich instabil ist, vorhanden sind, kann man vermuten, dass diese Unterteilung existiert und sinnvoll ist. Ebenfalls für die Existenz der Subgruppen sprechen die Ergebnisse aus Kapitel 3, wo die Verwendung des Maximum-Likelihood Algorithmus, zumindestens für das Npro-Fragment, eine Unterteilung in die gleichen Subgruppen ergab.
Zusammenfassend ergibt sich für BVDV eine Einteilung in zwei Genotypen BVDV-1 und BVDV-2. BVDV-1 wird in die 14 Subgruppen a-n unterteilt, wobei die Subgruppen a-k den bereits bekannten entsprechen und die Subgruppen l-n die drei neuen (N1, N8 und N9) repräsentieren. Für BVDV-2 ergeben sich die drei Subgruppen a-c, von denen c der neuen Gruppe N1 entspricht (Tab. 5.8).
Für BDV sind nur wenige Isolate vorhanden, die größtenteils auch nur aus Europa

stammen. Eine eindeutige Aussage zur Einteilung in Genotypen und/oder Subgruppen ist nur eingeschränkt möglich. Die bisherigen Ergebnisse zeigen aber größere Unterschiede zwischen den BDV Gruppen als zwischen den BVDV-1 und BVDV-2 Subgruppen. Es sind sowohl epidemiologische Unterschiede vorhanden, als auch Unterschiede im Sequenzvergleich erkennbar. Auch in den phylogenetischen Bäumen sind die Gruppen voneinander getrennt. Diese Ergebnisse sprechen für eine Unterteilung von BDV in Genotypen. Um eine analoge Bezeichnung wie für BVDV zu erhalten, sollte man die Genotypen mit BDV-1 bis BDV-8 bezeichnen. Mit den vorhandenen Sequenzen ist lediglich eine Unterteilung des Genotypen BDV-1 in Subgruppen a und b möglich (Tab. 5.8). Für BDV-6, BDV-7 und BDV-8 sind bisher nur 5'NTR-Sequenzen vorhanden, für eine endgültige Einteilung werden mindestens die Sequenzen eines weiteren Fragmentes benötigt. Eine Folge der Instabilität der 5'NTR Bäume führt dazu, dass sich abhängig von der Auswahl der Sequenzen verschiedene Einteilungen für die Gruppen ergeben. Während in Abbildung 5.1 der Genotyp BDV-8 mit BDV-1 clustert, bilden bei anderen Sequenzen die Genotypen BDV-6 bis BDV-8 ein Cluster [VA06].
Eine Übersicht aller BVDV und BDV Genotypen und Subgruppen ist in Abbildung 5.4 dargestellt. Da nur in der 5'NTR-Sequenzen von allen Genotypen und Subgruppen vorhanden sind , wurde der Baum für das 5'NTR-Fragment berechnet. In dieser Abbildung ist zuerkennen, dass die Genotypen BDV-6 bis BDV-8 ein anderes Cluster bilden als in Abbildung 5.1.

Für die Berechnungen der phylogenetischen Bäumen eigenen sich die Genfragmente E2 und Npro besser als die 5'NTR. Dies zeigt sich besonders an den *Bootstrap*-Werten. Die Berechnung aus Sequenzen der 5'NTR ergibt Werte zwischen 33% und 100%, bei E2 dagegen liegen die Werte zwischen 79% und 100% und bei Npro zwischen 75% und 100%. Diese Werte zeigen, dass der 5'NTR ziemlich instabil ist. Eine Einteilung nur anhand der 5'NTR-Sequenzen ist daher nur eingeschränkt möglich.
Die Ergebnisse haben gezeigt, dass zur definitiven Einordnung eines Isolates in eine Subgruppe mindestens die Sequenzen von zwei Fragmenten notwendig sind. Im Zweifelsfall sollte das Isolat vollständig sequenziert werden.

Genotyp	Subgruppe	bisherige Bezeichnung in dieser Arbeit
BVDV-1	a	BVDV-1 a
BVDV-1	b	BVDV-1 b
BVDV-1	c	BVDV-1 c
BVDV-1	d	BVDV-1 d
BVDV-1	e	BVDV-1 e
BVDV-1	f	BVDV-1 f
BVDV-1	g	BVDV-1 g
BVDV-1	h	BVDV-1 h
BVDV-1	i	BVDV-1 i
BVDV-1	j	BVDV-1 j
BVDV-1	k	BVDV-1 k
BVDV-1	l	BVDV-1 N1
BVDV-1	m	BVDV-1 N8
BVDV-1	n	BVDV-1 N9
BVDV-2	a	BVDV-2 a
BVDV-2	b	BVDV-2 b
BVDV-2	c	BVDV-2 N1
BDV-1	a	BDV a
BDV-1	b	BDV b
BDV-2		BDV c
BDV-3		BDV d
BDV-4		BDV g
BDV-5		BDV h
BDV-6*		BDV f
BDV-7*		BDV i
BDV-8*		BDV j

Tabelle 5.8: Aktualisierte Einteilung BVD und Border disease Viren in Genotypen und Subgruppen. Für die mit * gekennzeichneten Genotypen, existieren nur die 5'NTR-Sequenzen, die Einteilung ist daher noch nicht sicher.

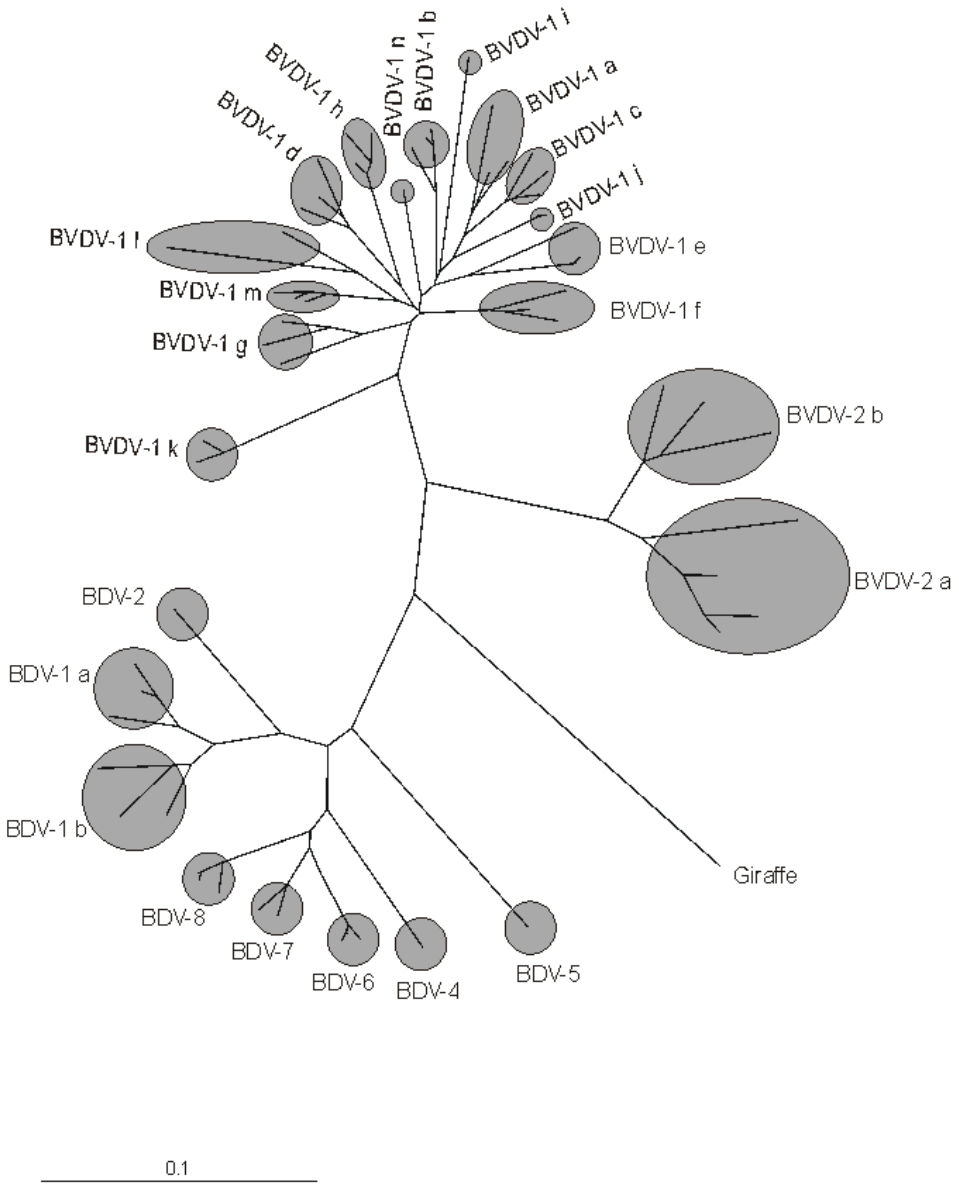


Abbildung 5.4: Einteilung der BVDV und BDV Isolate in Genotypen und Subgruppen. Der Baum wurde aus den 5'NTR-Sequenzen berechnet.

Anhang A

Abkürzungsverzeichnis

5'NTR	nichttranslatierte Region am 5' Ende des Genoms
BD	*Border disease*
BDV	*Border disease virus*
BVD	Bovine Virusdiarrhoe
BVDV	*Bovine viral diarrhea virus*
cp	cytopatisch
CSFV	*Classical swine fever virus*
DNA	*Desoxyribonucleic acid*
DV	Direktvergleich
EU	Europäische Union
ICTV	International Committee on Taxonomy of Viruses
JC-Dist	Jukes-Cantor-Distanz
kb	Kilobasen
MCH	Molekulare Uhr
MD	Schleimhautkrankheit (Mucosal Disease)
ML	Maximum-Likelihood
mRNA	messenger-RNA
nAK	neutralisierende Antikörper
ncp	nicht cytopatisch
nm	Nanometer
NJ	Neighbor-Joining
NTR	nichttranslatierte Region
NW	Needleman-Wunsch-Algorithmus
OTU	operational taxonomic unit
PCR	Polymerase-Kettenreaktion
PI	persistent infiziert
RNA	*Ribonucleic acid*

RT-PCR	Polymerase-Kettenreaktion nach Reverser Transkription des RNA–Genoms
SA BVD	schwere akute BVD
SW	Smith-Waterman-Algorithmus
UV	Ultraviolettstrahlung

Anhang B

Länderkürzel

Dieser Anhang enthält die in der Arbeit verwendeten Länderkürzel nach ISO 3166

EG	Ägypten
AR	Argentinien
AU	Australien
BE	Belgien
BR	Brasilien
BG	Bulgarien
CL	Chile
CN	China
CR	Costa Rica
DK	Dänemark
DE	Deutschland
FR	Frankreich
GB	Großbritannien
GT	Guatemala
HN	Honduras
IN	Indien
IR	Iran
IE	Irland
IT	Italien
JP	Japan
YU	Jugoslawien
CA	Kanada
CO	Kolumbien
HR	Kroatien
CU	Kuba
LA	Laos

LU	Luxemburg
MY	Malayisa
MX	Mexiko
NZ	Neuseeland
NL	Niederlande
AT	Österreich
PA	Panama
PL	Polen
PT	Portugal
RO	Rumänien
RU	Rußland
SE	Schweden
CH	Schweiz
SG	Singapur
SK	Slowakei
SI	Slowenien
ES	Spanien
KR	Süd Korea
ZA	Südafrika
TW	Taiwan
TH	Thailand
CZ	Tschechische Republik
TN	Tunesien
HU	Ungarn
VE	Venezuela
US	Vereinigte Staaten von Amerika

Anhang C

Publikationen

Teile der vorliegenden Arbeit wurden bereits vorab veröffentlicht:

GREISER-WILKE, I., DREIER, S., HAAS, L. und ZIMMERMANN, B. (2006):
Genetische Typisierung von Klassischen Schweinepest-Viren - ein Überblick.
Dtsch. tierärztl. Wschr. **113**, 133-138.

DREIER, S., ZIMMERMANN, B., MOENNIG, V. and GREISER-WILKE, I. (2006):
A sequence database allowing automated genotyping of *Classical swine fever virus* isolates.
J. Virol. Methods, accepted.

Teile der vorliegenden Arbeit wurden auf folgenden Tagungen präsentiert:

S. Dreier, B. Zimmermann, V. Moennig, I. Greiser-Wilke
Automated genotyping of Classical swine fever (CSF) virus isolates
GfV Annual Meeting, 16 -19 März 2005, Hannover

S. Dreier, B. Zimmermann, V. Moennig, I. Greiser-Wilke
BVDV Sequence Database Designed for Genotyping of Bovine Viral Diarrhea (BVDV) and Border Disease Virus (BDV) Isolates
6^{th} Pestivirus Symposium, 13 -16 September 2005, Thun, Switzerland

S. Dreier, I. Greiser-Wilke
Classification of Bovine Viral Diarrhea and Border Disease Virus Isolates: Current Status
ESVV 7^{th} International Congress of Veterinary Virology, 24 -27 September 2006, Lisboa, Portugal

Literaturverzeichnis

[AN02] P. Arauz-Ruiz, H. Norder, B.H. Robertson and L.O. Magnius *Genotype H: a new Amerindian genotype of hepatitis B virus revealed in Central America.* J. Gen. Virol. 83, 2059-2073 (2002).

[Ano01] Anonymous *Council Directive 2001/89/EC of 23 October 2001 on Community measures for the control of classical swine fever.* Official Journal L316, 5-35 (2001).

[Ano06] Anonymous *http://ticker-grosstiere.animal-health-online.de.* (2006).

[Ano06b] Anonymous *http://www.ncbi.nlm.nih.gov/.* (2006).

[Ano06c] Anonymous *http://www.kreis-borken.de/kreisregion/xgse4/gebieteundzonen.php?r=kreisregion&p=0,2.* (2006).

[AF04] M. Arnal, D. Fernadez-de-Luco, L. Riba, M. Maley, J. Gilray, K. Willoughby, S. Vilcek and P.F. Nettleton *A novel pestivirus associated with deaths in Pyrenean chamois (Rupicapra pyrenaica pyrenaica).* J. Gen. Virol. 85, 3653-3657 (2004).

[AO03] P. Arias, M. Orlich, M. Prieto, S. Cedillo Rosales, H.-J. Thiel, M. Alvarez and P. Becher *Genetic heterogeneity of bovine viral diarrhoea viruses from Spain.* Vet. Microbiol. 96 (4), 327-336 (2003).

[Aya99] F. J. Ayala *Molecular clock mirages.* Bioessays 21 (1), 71-75 (1999).

[BA03] P. Becher, R. Avalos Ramirez, M. Orlich, S. Cedillo Rosales, M. Konig, M. Schweizer, H. Stalder, H. Schirrmeier and H. J. Thiel *Genetic and antigenic characterization of novel pestivirus genotypes: implications for classification.* Virology 311, 96-104 (2003).

[BK01] P. Becher, M. König and H. J. Thiel *Bovine Virusdiarrhö und Mucosal Disease: Molekularbiologie des Erregers, Pathogenese, Labordiagnostik und Bekämpfung.* Tierärztl. Praxis 29, 266-275 (2001).

[BK04] S. D. Blacksell, S. Khounsy, D. B. Boyle, I. Greiser-Wilke, L. J. Gleeson, H. A. Westbury and J. S. Mackenzie *Phylogenetic analysis of the E2 gene of classical swine fever viruses from Lao PDR.* Virus Res. 104 , 87-92 (2004).

[BL99] H. Björklund, P. Lowings, T. Stadejek, S. Vilcek, I. Greiser-Wilke, D. Paton and S. Belak *Phylogenetic Comparsion and Modecular Epidemiology of Classical Swine Fever Virus.* Virus Genes 19:3, 189-195 (1999).

[Bro94] J. K. M. Brown *Bootstrap hypothesis tests for evolutionary trees and other dendrograms.* Proc. Natl. Acad. Sci USA 91, 12293-12297 (1994).

[Bro03] K. V. Brock *The persistence of bovine viral diarrhea virus.* Biologicals 31, 133-135 (2003).

[BS01] A. van Belkum, M. Struelens, A. de Visser, H. Verbrugh and M. Tibayrenc *Role of Genomic Typing in Taxonomy, Evolutionary Genetics, and Microbial Epidemiology.* Clin. Microbiol. Rev. Vol. 14 No. 3, 547-560 (2001).

[Bun03] C. Bunzenthal *Bestimmung der Virulenz von Virusisolaten der Klassischen Schweinepest.* Dissertation (2003).

[BV97] C. Baule, M. van Vuuren, J. P. Lowings and S. Belák *Genetic heterogeneity of bovine viral diarrhoea viruses isolated in Southern Africa.* Virus Res. 52 (2), 205-220 (1997).

[CB00] P. Clote and R. Backofen *Computational Molecular Biology An Introduction.* Wiley & Sons, Inc (2000).

[CF05] A. W. Confer, R. W. Fulton, D. L. Step, B. J. Johnson, and J. F. Ridpath *Viral Antigen Distribution in the Respiratory Tract of Cattle Persistently Infected with Bovine Viral Diarrhea Virus Subtype 2a.* Vet. Pathol. 42, 192-199 (2005).

[CL88] M. S. Colett, R. Larson, c. Gold, D. Strick, D. K. Anderson and A. F. Purchio *Molecular cloning and nucleotide sequence of the pestivirus bovine viral diarrhea virus.* Virology 165 (1), 191-199 (1988).

[CW91] M. S. Colett, M. A. Wiskerchen, E. Welniak and S. K. Belzer *Bovine viral diarrhoea virus genomic organisation.* In: Liess,B., Moennig,V., Pohlenz,J. and Trautwein,G. (Hrsg.): Ruminant pestivirus infections. Arch. Virol. (Suppl. 3), 19-27 (1991).

[DE01] N. J. Dimmock, A. J. Easton and K.N. Leppard *Introduction to Modern Virology.* Blackwell Science Ltd. (2001).

[DJ04] D. Deregt, R. M. Jacobs, P. S. Carman and S. V. Tessaro *Attenuation of a virulent type 2 bovine viral dirrhea virus*. Vet. Microbiol. 100, 151-161 (2004).

[DM96] K. R. Depner, V. Moennig and B. Liess *Epidemiologische Betrachtungen zur "typischen" und "atypischen" Schweinepest*. Amtstierärztlicher Dienst und Lebendsmittelkontrolle IV, 335-342 (1996).

[DR96] K. R. Depner, A. Rodriquez, J. Pohlenz and B. Liess *Persistent classical swine fever virus infection in pigs infected after weaning with a virus isolated during 1995 epidemic in Germany : Clinical, virological, serological and pathological findings*. Europ. J. Vet. Pathol. 2, 61-66 (1996).

[Dun70] H. W. Dunne *Hog Cholera*. In: H. W. Dunne (Hrsg.): Diseases of swine. Iowa State University Press, Ames, Iowa, U.S.A., 3rd Ed., S. 177-239 (1970).

[Edw00] S. Edwards *Survival and inactivation of classical swine fever virus*. Vet. Microbiol. 73, 175-181 (2000).

[Efr79] B. Efron *Bootstrap Metods: Another look at the Jackknife*. The Annals of Statistics Vol. 7 No. 1, 1-26 (1979).

[Efr94] B. Efron *The Jackknife, the Bootstrap and Other Resampling Plans*. SIAM (Society for Industrial and Aplied Mathematics) (1994).

[EH96] B. Efron, E. Halloran and S. Holmes *Bootstrap confidence levels for Phylogenetic trees*. Proc. Natl. Acad. Sci USA 93, 13429-13434 (1996).

[ER02] J. F. Evermann and J. F. Ridpath *Clinical and epidemiologic observations of bovine viral diarrhea virus in the northwestern United States*. Vet. Microbiol. 89, 129-139 (2002).

[FC03] E. Falcone, P. Cordioli, M. Tarantino, M. Muscillo, G. La Rosa and M. Tollis *Genetic heterogeneity of bovine viral diarrhoea virus in Italy*. Vet Res Commun. 2003 Sep;27(6):485-94.

[Fel81] J. Felsenstein *Evolutionary Trees from DNA Sequences: A Maximum Likelihood Approach*. J. Mol. Evol. 17, 368-376 (1981).

[Fel89] J. Felsenstein *Phylip: phylogeny inference package (version3.5c)*. Cladistics 5, 164-166 (1989).

[Fel04] J. Felsenstein *Inferring Phylogenies*. Sinauer Associates, Inc. (2004).

[FM99] J. Fritzemeier and V. Moennig *Bovine Virusdiarrhoe*. Nutztierpraxis. 9-11 (1999).

[FJ05] K. Frölich, S. Jung, A. Ludwig, D. Lieckfeldt, P. Gibert, D. Gauthier and J. Hars *Detection of a Newly Described Pestivirus of Pyrenean Chamois (Rupicapra pyrenaica pyrenaica) in France.* J Wildl. Dis. 41(3), 606-610 (2005).

[FR02] E. F. Flores, J. F. Ridpath, R. Weiblen, F. S. F. Vogel and L. H. V. G. Gil *Phylogenetic analysis of Brazilian bovine viral diarrhea virus type 2 (BVDV-2) isolates: evidence for a subgenotype within BVDV-2.* Virus Res. 87, 51-60 (2002).

[GG03] I. Greiser-Wilke, B. Grummer and V. Moennig *Bovine viral diarrhoea eradication and control programmes in Europe.* Biologicals 31, 113-118 (2003).

[GK02] C.Geiger and C. Kanzow *Theorie und Numerik restringierter Optimierungsaufgaben.* Springer-Verlag (2002).

[GM90] I. Greiser-Wilke, V. Moennig, C. O. Coulibaly, J. Dahle, L. Leder and B. Liess *Identification of conserved epitopes on a hog cholera virus protein.* Arch. Virol. 111, 213-225 (1990).

[GM04] I. Greiser-Wilke and V. Moennig *Vaccination against classical swine fever virus: limitations and new strategies.* Anim. Health. Res. Rev. 5(2), 223-226 (2004).

[God76] M. Goodman *Protein sequences in phylogenyVaccination against classical swine fever virus: limitations and new strategies.* In: Ayala, F. J. (ed.) Molecular Evolution. Sinauer Associates, Sunderland,MA., 141-159 (1976).

[God81] M. Goodman *Decoding the pattern of protein evolution.* Prog. Biophys. Mol. Biol. 38, 105-164 (1981).

[Got82] O. Gotoh *An Improved Algorithm for Matching Biological Sequences.* J. Mol. Biol. 162, 705-708 (1982).

[GZ00] I. Greiser-Wilke, B. Zimmermann, J. Fritzemeier, G. Floegel and V. Moennig. *Structure and presentaion of a World Wide Web database of CSF virus isolates held at EU Reference Laboratory.* Vet. Microbiol. 73, 131-136 (2000).

[HB98] M. Hofmann and S. Bossy *Klassische Schweinepest 1993 in der Schweiz: molekular-epidemiologische Charakterisierung der Virusisolate.* Schweiz. Arch. Tierklinik. 140, 365-370 (1998).

[HG03] A. Hurtado, A. L. Garcia-Perez, G. Aduriz and R. A. Juste. *Genetic diversity of ruminant pestiviruses from Spain.* Virus Res. 92 (1), 67-73 (2003).

[JG03] L. Jemersic, I. Greiser-Wilke, D. Barlic-Maganja, M. Lojkic, J. Madic, S. Terzic and J. Grom *Genetic typing of recent classical swine fever virus isolates from Croatia*. Vet. Microbiol. 96, 25-33 (2003).

[JS04] F. Jarre and J. Stoer *Optimierung*. Springer-Verlag (2004).

[Kim68] M. Kimura *Genetic variability maintained in a finite population due to mutational production of neutral and nearly neutral isoalleles*. Genet. Res. 11, 247-269 (1968).

[Kub67] G. Kubin *In vitro Merkmale des Schweinepestvirus*. Zentralbl. Veterinärmed. B 14, 543-552 (1967).

[KH78] P. D. McKercher, W. R. Hess and F. Hamdy *Residual viruses in pork products*. Appl. Environ. Microbiol. 35, 142 (1978).

[KS05] V. Kaden, H. Steyer, J. Schnabel and W. Bruer *Classical Swine Fever (CSF) in Wild Boar: the Role of the Transplacental Infection in the Perpetuation of CSF*. J. Vet. Med. B 52, 161-164 (2005)

[Lew91] W. Lewin *Gene*. VCH Verlagsgesellschaft mbH (1991).

[Li97] W. H. Li *Molecular Evolution*. Sinauer Associates, Inc. (1997).

[Lie94] B. Liess *BVD-Erreger, Krankheitsbild, wirtschaftlicher Schaden und Möglichkeiten der Vorbeuge*. Milchpraxis. 32:36-39 (1994).

[MF03] V. Moennig G. Floegel-Niesmann and I. Greiser-Wilke *Clincal Signs and Epidemiology of Classical Swine Fever: A Review of New Knowledge*. The Veterinary Journal 165, 11-20 (2003).

[MG03] V. Moennig and I. Greiser-Wilke *Perspektiven der BVD-Bekämpfung in Deutschland* . BMTW.116:222-226 (2003).

[MG05] G. M. De Mia, I. Greiser-Wilke, F. Feliziani, M. Giammarioli and A. De Giuseppe *Genetic Characterization of a Caprine Pestivirus as the First Member of a Putative Novel Pestivrus Subgroup*. J. Vet. Med. B52, 206-210 (2005).

[MH88] R. Moormann and M. M. Hulst *Hog cholera virus: identification and characterization of the virus-specific RNA synthesized in infected swine kidney cells*. Virus Res. 11, 281-291 (1988).

[ML81] H. Meyers, B. Liess, H. R. Frey, W. Hermanns and G. Trautwein *GExperimental transplancental transmission of hog cholera virus in pigs. IV. Virological and serological studies in newborn piglets*. J. Vet. Med. B 28, 659-668 (1981).

[MM05] T. J. Mahony, F. M. McCarthy, J. L. Gravel, B. Corney, P. L. Young and S. Vilcek *Gentic analysis of bovine viral dirrahoea virues from Austraila*. Vet. Microbiol. 106, 1-6 (2005).

[Moe00] V. Moennig *Introduction to classical swine fever: virus, disease and control policy*. Vet. Microbiol. 73, 93-102 (2000).

[MP69] W. L. Mengeling and R. A. Packer *Pathogenesis of chronic hog cholera: host response*. Am. J. Vet. Res. 30, 409-417 (1969).

[MP92] V. Moennig and P. G. W. Plagemann *The pestivirus*. Adv. Virus Res. 41, 53-98 (1992).

[MP04] N. Mishra, B. Pattnaik, S. Vilcek, S. S. Patil, P. Jain, N. Swamy, S. Bhatia and H. K. Pradhan *Gentic typing of bovine viral dirrahoea virus isolates from India*. Vet. Microbiol. 104, 207-212 (2004).

[MP04b] R. J. Monies, D. J. Paton and S. Vilcek *Mucosal disease-like lesions in sheep infected with Border disease virus*. Veterinary Record 155, 765-769 (2004).

[MR89] G. Meyers, T. Rümenapf and H. J. Thiel *Molecular cloning and nucleotide sequence of the genome of hog cholera virus*. Virology 171, 555-567 (1989).

[MT96] G. Meyers and H. J. Thiel *Molecular characterization of pestiviruses*. Adv. Virus Res. 47, 53-118 (1996).

[MW90] R. Moormann, P. M. Warmedam, B. von der Meer, W. M. M. Schaaperg, G. Wensvoort and M. M. Hulst *Molecular cloning and nucleotide sequence of hog cholera virus strain Brescia and mapping of the genomic region encoding envelope protein E1*. Virology 167, 184-198 (1990).

[Mue01] H.-J. Müller *PCR Polymerase Ketterreaktion*. Spektrum Akademischer Verlag Heidelberg Berlin (2001).

[NG98] P. F. Nettleton, J. A. Gilray, P. Russi and E. Dlissi *Border disase of sheep and goats*. Vet. Res. 29, 327-340 (1998).

[NH04] M. Nagai, M. Hayashi, S. Sugita, Y. Sakoda, M. Mori, T. Murakami, T. Ozawa, N. Yamada and H. Akashi *Phylogentic analysis of bovine viral dirrahea virues using five different gentic regions*. Virus Res. 99, 103-113 (2004).

[NI01] M. Nagai, T. Ito, S. Sugita, A. Genno, K. Takeuchi, T. Ozawa, Y. Sakoda, T. Nishimori, K. Takamura and H. Akashi *Genomic and serological diversity of bovine viral diarrhea virus in Japan*. Arch. Virol. 146 (4), 685-696 (2001).

[OM46] P. Olafson, A. D. Maccallum and F. H. Fox *An apparently new transmissible disease of cattle*. Cornell Vet. 36, 205-213 (1946).

[PM00] D. J. Paton, A. McGoldrick, I. Greiser-Wilke, S. Parchariyanon, J.-Y. Song, P. P. Liou, T. Stadejek, J. P. Lowings, H. Björklund and S. Belák *Gentic typing of classical swine fever virus*. Vet. Microbiol. 73, 137-157 (2000).

[PH94] C. Pellerin, J. van den Hurk, J. Lecomte and P. Tussen *Identification of a new group of bovine viral diarrhea virus strains associated with severe outbreaks and high mortalities*. Virology 203(2), 260-268 (1994).

[QM06] V. L. Quadros, S. V. Mayer, F. S. F. Vogel, R. Weiblen, M. C. S. Brum, S. Arenhart and E. F. Flores *A search for RNA insertions and NS3 gene duplication in the genome of cytopathic isolates of bovine viral diarrhea virus*. Braz. J. Med. Biol. Res. 39, 935-944 (2006).

[Rid03] J. F. Ridpath *BVDV genotypes and biotypes: practical implications for diagnosis and control*. Biologicals 31, 127-131 (2003).

[RN06] J. F. Ridpath , J. D. Neill, S. Vilcek, E. J. Dubovi and S. Carman *Multiple outbreaks of severe acute BVDV in North America occurring between 1993 and 1995 linked to the same BVDV2 strain*. Vet. Microbiol. 114, 196-204 (2006).

[RU93] T. Rümenapf, G. Unger, J. H. Strauss and H.-J. Thiel *Processing of the envelope glycoproteins of pestiviruses*. J. Virol. 67, 3288-3294 (1993).

[SB99] A. J. de Smit, A. Bouma, C. Terpstra and J. T. van Oirschot *Transmission of classical swine fever virus by artificial insemination*. Vet. Microbiol. 67, 239-249 (1999).

[SE00] A. Stegeman, A. Elbers, H. de Smith, H. Moser, J. Smak and F. Pluimers *The 1997-1998 epidemic of classical swine fever in the Netherlands*. Vet. Microbiol. 73, 183-196 (2000).

[SK06] P. Schiefer, R. Krametter-Froetscher, A. Schleiner, A. Loitsch, F. Golja, K. Möstl and W. Baumgartner *Prävalenz Pestivirus-spezifischer Antikörper in Seren von Schafen und Ziegen aus Tirol (österreich)*. Dtsch. Tierärztl. Wochenschr. 113(2), 55-58 (2006).

[SM05] H. P. Stalder, P. Meier, G. Pfaffen, D. Wageck-Canal, J. Rufenacht, P. Schaller, C. Bachofen, S. Marti, H. R. Vogt and E. Peterhans *Genetic heterogeneity of pestiviruses of ruminants in Switzerland*. Prev. Vet. Med. 72, 37-41 (2005).

[SN87] N. Saitou and M. Nei *The Neighbor-joining Method: A New Method for Reconstructing Phylogenetic Trees*. Mol. Biol. Evol. 4(4), 406-425 (1987).

[SW81] T. F. Smith and M. S. Waterman *Identifiction of common molecular subsequences*.J. Mol. Biol. 147:195-197 (1981).

[TF01] M. Tajima, H.-R. Frey, O. Yamato, Y. Maede, V. Moennig, H. Scholz and I. Greiser-Wilke *Prevalence of genotypes 1 and 2 of bovine viral dirrahea virus in Lower Saxony, Germany*. Virus Res. 76, 31-42 (2001).

[TL05] F. Thabti, C. Letellier, S. Hammami, M. Pepin, M. Ribiere, A. Mesplede, P. Kerkhofs and P. Russo *Detection of a novel border disease virus subgroup in Tunisian sheep*. Arch. Virol. 150, 215-229 (2005).

[TS04] I. Toplak, T. Sandvik, D. Barlic-Maganja, J. Grom and D. J. Paton *Genetic typing of bovine viral diarrhoea virus: most Slovenian isolates are of genotypes 1d and 1f*. Vet Microbiol. 2004 Apr 19;99(3-4):175-185.

[TT94] N. Tautz, H.-J. Thiel, E. J. Dubovi and G. Meyers *Pathogenesis of Mucosal Disease: a Cytopathogenic Pestivirus Generated by an Internal Deletion*. J. Virol. 68(5), 3289-3297 (1994).

[Oir79] J. T. van Oirschot *Experimental production of congenital persistent swine fever infections. I. Clinical, pathological and virological observations*. Vet. Microbiol. 4, 117-132 (1979).

[Oir92] J. T. van Oirschot *Hog cholera*. In: Lemann et al. (Hrsg.), Diseases of swine, 7th ed. Ames, IA, Iowa State University Press, 274-285 (1992).

[Oir99] J. T. van Oirschot *Classical swine fever (Hog cholera)*. In: W. L. Mengeling (Hrsg.): Diseases of swine. Iowa State University Press, Ames, Iowa, U.S.A., 8th Ed., 159-172 (1999).

[OT77] J. T. van Oirschot and C. Terpstra *A congential persistent swine fever infection. I. Clinical and virological observations*. Vet. Microbiol. 2, 121-132 (1977).

[OW87] H. Ochman and A. C. Wilson *Evolution in bacteria: Evidence for a universal substitution rate in cellular genoms*. J. Mol. Evol. 26, 74-86 (1987).

[VA06] Valdazo-Gonzalez, Alvarez-Martinez, M. and Greiser-Wilke,I. *Gentic typing and prevalence of Border disease virus (BDV) in small ruminant flocks in Spain*. Vet. Microbiol. Epub ahead of print (2006).

[VD04] S. Vilcek, B. Durkovic, M. Kolesarova, I. Greiser-Wilke,I. and D. Paton *Gentic diversity of international bovine viral dirrahoea virus (BVDV) iso-*

lates: identification of a new BVDV-1 gentic group. Vet. Res. 35, 609-615 (2004).

[VD05] S. Vilcek, B. Durkovic, M. Kolesarova and D. Paton *Gentic diversity of BVDV: Consequences for classification and molecular epidemiology*. Prev. Vet. Med. 72, 31-35 (2005).

[VN97] S. Vilcek, P. F. Nettleton, D. J. Paton and S. Belak *Molecular characterization of ovine pestiviruses*. J. Gen. Virol. 78 (Pt 4), 725-735 (1997).

[VN06] S. Vilcek and P. F. Nettleton *Review Pestiviruses in wild animals*. Vet. Microbiol. 116, 1-12 (2006).

[VP01] S. Vilcek, D. J. Paton, B. Durkovic, L. Strojny, G. Ibata, A. Moussa, A. Loitsch, W. Rossmanith, S. Vega, M. T. Scicluna and V. Palfi *Bovine viral diarrhoea virus genotype 1 can be separated into at least eleven genetic groups*. Arch. Virol. 146 (1), 99-115 (2001).

[VS97] S. Vilcek, T. Stadejek, I. Takacsova, L. Strojny and M. Mojzis *Genetic analysis of classical swine fever virus isolates from a small geographic area*. Dtsch. Tierarztl. Wochenschr. 104, 9-12 (1997).

[WS90] E. Weiland, R. Stark, B. Haas, T. Rümenapf, G. Meyers and H.-J. Thiel *Pestivirus glycoprotein with induces neutralizing antibodies form a part of a disulfidlinked heterodimer*. J. Virol. 64, 3563-3569 (1990).

[WB95] Wengler,G., Bradley,D.W., Collett,M.S., Heinz,F.X., Schlesinger,R.W. and Strauss,J.H. *Family Flaviviridae*. In: Murphy,F.A., Fauquet,C.M., Bishop,D.H.L., Ghabrial,S.A., Jarvis,A.W., Martelli,G.P., Mayo,M.A. and Summers,M.D. (Hrsg.): Virus Taxonomy. Sixth report of the International Committee on taxonomy of viruses. Springer Verlag, Wien, New York, S. 415-427 (1995).

[WB85] E. G. Westaway, M. A. Brinton, S. Y. A. Gaidamovich, M. C. Horzinek, A. Igarachi, L. Kääriäinen, D. K. Lvov, J. S. Poterfield, P. K. Russel and D. W. Trent *Flaviviridae*. Intervirology 24, 183-192 (1985).

[WF01] H. Wonnemann, G. Floegel-Niesmann, V. Moennig and I. Greiser-Wilke *Genetische Typisierung von deutschen Isolaten des Virus der Klassischen Schweinepest*. Dtsch. tierärztl. Wschr. 108, 252-256 (2001).

[WG99] M. N. Widjojoatmodjo, H. G. P. van Gennip, A. J. de Smit and R. J. M. Moormann *Comparative sequence analysis of classical swine fever virus isolates from the epizootic in the Netherlands in 1997-1998*. Vet. Microbiol. 66, 291-299 (1999).

[ZP62] E. Zuckerkandl and L. Pauling *Molecular disease, evolution and gentic heterogeneity*. In: Kasha, M. and B. Pullman (eds.) Horizons in Biochemistry. Academic Press, New York, 189-225 (1962).

Lebenslauf

Sabrina Dreier

14. Mai 1980	Geboren in Gehrden
1986 - 1990	Ernst-Reuter-Schule, Barsinghausen
1990 - 1992	Orientierungsstufe am Spalterhals, Barsinghausen
1992 - 1999	Ganztagsgymnasium Barsinghausen
Juni 1999	Abitur
Oktober 1999 - November 2003	Studium Mathematik mit Nebenfach BWL, Universität Hannover
November 2003	Diplom (Titel: „Ein kaskadisches Mehrgitterverfahren für nichtkonforme finite Elemente“)
März 2000 - November 2003	Wissenschaftliche Hilfskraft am Institut für Angewandte Mathematik, Universität Hannover
seit Dezember 2003	Wissenschaftliche Angestelle am Institut für Virologie, Tierärztliche Hochschule Hannover