Statistical and Computational Analysis of Genetic Sequences

Tanja Stadler, Carsten Magnus, Timothy Vaughan

Joëlle Barido-Sottani, Veronika Bošková,
Jūlija Pečerska, Jana Huisman

Contents

1 Introduction

Nothing in Biology Makes Sense Except in the Light of Evolution
(Theodosius Dobzhansky [**Dobzhansky1973**])

1.1 A brief overview

In biology, we observe and measure aspects of living organisms around us and in that way obtain an understanding on how the living world functions. However, we cannot directly observe and measure every aspect of the living world. Some features are not observable because we do not have the proper technical equipment developed (yet), while other features may be regarded as unobservable *per se*. For example, many species populate this planet, but we cannot observe how they came about as this much of this process occurred millions of years before any of us were born. In other domains, we cannot observe processes due to ethical reasons. For example, while we observe human hosts who are infected by a pathogen, we cannot observe the dynamics of the pathogen jumping to a new human host and establishing an infection. (If we knew that a pathogen was about to jump to a new human host, it would be unethical to not prevent the infection.)

Statistical inference methods can help us to obtain an understanding of such unobservable processes indirectly using the available snapshot data, such as data gleaned from extant species or infected hosts. The book focuses on methods for learning about unobservable evolutionary and population dynamic processes using genotypic data (i.e. data on the genetic material carried by individuals), namely Deoxyribonucleic/Ribonucleic acid (DNA/RNA) sequence data, possibly together with some phenotypic data (i.e. data on the appearance of individuals). Throughout, we provide examples of real world data analysis results obtained using the computational and statistical tools presented. The book is structured into four main parts:

Obtaining & Organizing Sequences How do we obtain sequences from biological samples, align them, and what can data mining tell us about them? (Chapter 2-4)

Molecular Evolution How does genetic information change through time? (Chapter 5)

Phylogenetics How can we determine the relatedness of the biological samples based on their genetic information? What is their underlying phylogeny? (Chapter 6 - 8 and 11)

Phylodynamics What are the population dynamics (e.g. speciation and extinction dynamics or pathogen transmission dynamics) that give rise to the phylogeny and to the genetic information we observe? (Chapter 9-10)

A classic area of biology where statistical and computational tools are required to understand an unobservable process is macroevolution, where the biological units are species and the available data includes both genotypic and phenotypic information for some species. Phylogenetics (part 3 of the book) studies the evolutionary relationships between the biological units, e.g. the species. The central object in phylogenetics is a *phylogeny* which may be a tree or a network. Phylogenetic methods aim at reconstructing the phylogeny based on genotypic and possibly phenotypic information of the sampled biological units, e.g. the present-day species. Figure 1.1 shows a phylogenetic tree of great apes which was inferred using phylogenetic methods.

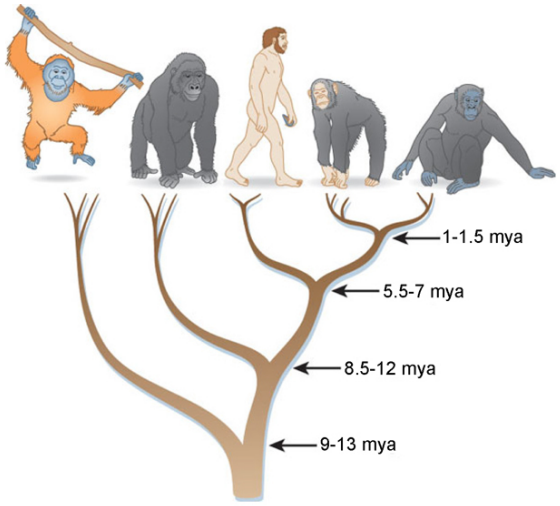


Figure 1.1: Figure adapted from [**Paabo2003**].

Going one step further in the analysis, phylodynamics (part 4 of the book) aims at describing and quantifying the population dynamic processes that gave rise to the phylogeny. In macroevolution, the relevant population dynamic process is the process of speciation and extinction, i.e. how quickly species appear and go extinct, and may further include e.g. the process of hybridization. Both phylogenetics and phylodynamics use models of molecular evolution (part 2 of the book) to capture the way the genetic code of a biological unit, e.g. a species, changes through time.

"Nothing in Biology Makes Sense Except in the Light of Evolution", the title of an essay by evolutionary biologist Theodosius Dobzhansky [**Dobzhansky1973**], pithily describes why we must acknowledge evolutionary processes when studying any area of biology. It follows then that phylogenetics and phylodynamics are crucial in order to not only understand unobserved processes in evolution but also to understand unobservable processes in a wide range of biological areas; such as:

- **Macroevolution**, where the biological unit is a species;

  - Molecular evolution describes genetic changes in the species;
  - Phylogeny displays the relationship between species, i.e. the tree of life;
  - Population dynamics describes the speciation and extinction process.

- **Microevolution**, where the biological unit is an individual (maybe a bacterial or archaeal cell, a unicellular eucaryote or a multicellular individual);
  - Molecular evolution describes genetic changes in the individuals;
  - Phylogeny displays the relationship between individuals;
  - Population dynamics describes the birth and death of individuals.

  We note here that population genetic approaches (going back to work in the 1920s by Fisher, Haldane and Wright [**fisher1930**, **wright1931**, **haldane1932**]) were developed to consider genetic changes within a population of given size. Population dynamic approaches (going back to work by Lotka and Volterra [**lotka1910**, **volterra1928**]) consider the changes in population size through time. Phylodynamic approaches discussed in this book allow to study both the population genetics and population dynamics of a population.

- **Epidemiology**, where the biological unit is an infected host;
  - Molecular evolution describes genetic changes in the pathogen population within an infected host;
  - Phylogeny displays the pathogen transmission chain;
  - Population dynamics describes the transmission of the pathogen to susceptible hosts and the recovery or death of infected hosts.

- **Immunology**, where the biological unit is an immune response cell within a host, such as a B- or a T-cell;
  - Molecular evolution describes changes in immune cells through e.g. somatic hypermutations or recombination;
  - Phylogeny displays the cells' evolutionary history;
  - Population dynamics describes the mechanisms of generation and loss of different immune cells within the host.

- **Cancer**, where the biological unit is a cell within an organism;
  - Molecular evolution describes the genetic changes of the cells;
  - Phylogeny displays the relationships of different cancer cells and healthy cells;
  - Population dynamics describes the spread and loss of cancer cell types.

- **Development**, where the biological unit is a cell within an organism;

  - Differentiation (rather than molecular evolution) describes the phenotypic changes of the cells;

  - Phylogeny displays the relationships of different cells such as stem cells and cells with different differentiation levels;

  - Population dynamics describes the mechanisms of cell differentiation.

- **Linguistics**, where the cultural unit (rather than biological unit) is a language;

  - Evolution (which is not molecular here) describes the evolution of words and/or grammatical structures within languages;

  - Phylogeny displays language evolution and differentiation through time;

  - Population dynamics describes the appearance and extinction of languages.

Genetic sequence data is the result of evolution. In this book, we delve into models and methods for molecular evolution, phylogenetics and phylodynamics to understand the genetic sequence data. We also highlight some scenarios where merely data mining approaches provide information about unobserved processes based on genetic sequences, while the evolutionary history can even be ignored. In the remainder of the introduction, we will briefly describe the main aspects of the theory of evolution, and how this theory itself went through different steps of evolution.

1.2 A brief history of evolution

For thousands of years, scientists have been trying to explain how the living world came into existence. The idea that everything was created at once, and has since existed in a fixed state, was eventually replaced by the concept of evolution and perpetual change. Extant species data as well as fossil evidence provide an overwhelming support for evolution. Initially, the concept of evolution was discarded by many, as it contradicted religious view of men being the pride of creation. Nowadays, the concept of evolution is widely accepted in the scientific community and big parts of society, while some parts of society neglect this concept in favour of creationism. A recent study [**Matzke2016**] provides a view on the evolution of anti-evolution, i.e. creationism. Evolution as a scientific theory also went through its own stages of evolution, from simple beginnings to current more elaborate concepts. The question all of the theories tried to answer is “How does evolution occur?” by assessing “Does the data support the evolutionary theory?”. In this section we introduce theories which paved the way to the modern approach. This is by no means a complete picture of the evolution of evolution, but focuses on how we came to our current understanding of evolution through some specific influential historical concepts.

1.2.1 Lamarckian evolution

In the nineteenth century, biologists and naturalists increasingly discussed the possibility of explaining species diversity via evolution. In 1809, the French biologist Jean-Baptiste Lamarck proposed that evolution occurs through use and disuse of features [**Lamarck1809**]. This means that an organism could develop a useful feature during its lifetime which would then be passed on to its offspring. This is a so-called soft inheritance, the inheritance of acquired characteristics.

Lamarck's favourite example animal was the giraffe, shown in Figure 1.2, and he explained the length of the giraffe's neck as follows: The first giraffes had short necks that made it hard for them to reach the leaves on the trees. This meant that giraffes always had to stretch their necks, which would become slightly lengthened over the course of their lives. This lengthened neck would then be passed on to their offspring, and over the course of many generations the neck length would increase to its current proportion. Thus evolution occurs via the mechanism of local adaptions to the environment.

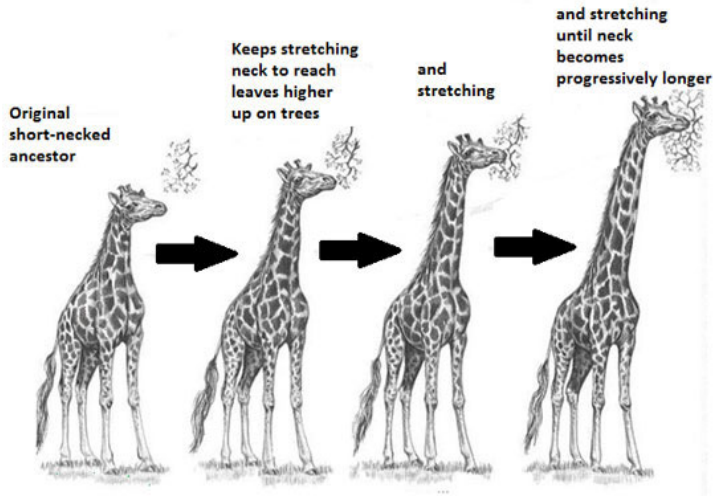


Figure 1.2: Lamarck's famous giraffe example. According to Lamarck, evolution occurs through use and disuse of features. Thus, if giraffes prefer leaves from taller branches, they will strain their necks more and more which would cause their offsprings to have longer necks. [**Lamarck1809**]

1.2.2 Darwinian evolution

In 1859 the British biologist Charles Darwin published his book "The Origin of Species by Means of Natural Selection" [**darwin1859origin**] describing a theory of evolution which aims at explaining evolution via the mechanism of natural selection. For evolution to occur via natural selection, four components are needed:

Multiplication The individuals multiply and produce offspring;

Variation There is phenotypic variation across individuals, i.e. individuals differ in some aspect of their appearance;

Heredity The phenotype is to some extent heritable from one generation to the next;

Competition There are fitness differences across phenotypes, i.e. the average number of surviving offspring depends on the phenotype of the parent individual.

While Darwin's mechanism could explain well the data he collected during his voyage on the Beagle, one important feature in the theory of natural selection was missing: The principle of how phenotypes where inherited could not be explained thoroughly.

1.2.3 Mendelian inheritance - foundations of genetics

In 1866, seven years after Darwin's influential book appeared, the Austrian monk Gregor Mendel published his observations on the possible mechanisms of heredity of the phenotypes, which he developed from his experiments with pea plants [**mendel1866**]. He observed that certain traits such as flower colours get passed on to the next generation in a predictable fashion. He described invisible factors -which we call *genes*, that have different variants -which we call *alleles*, which determine the traits -which we call *phenotypes*. The invisible factors are passed on from one generation to the next. We now introduce vocabulary that is important throughout this book:

Gene The entity encoding a phenotype (e.g. the flower colour of peas).

Allele The version of a gene (e.g. one allele of the colour gene may encode for white (y); another allele for purple (Y)).

Genotype The collection of genes of one individual.

Phenotype The collection of traits (i.e. appearance) of one individual.

Based on the experimental data, Mendel concluded that each pea plant individual has two alleles of each gene, a random one from the father and one from the mother. A *dominant allele* is the allele that determines the phenotype (e.g. if the purple allele Y is dominant, peas having the Yy allele combination would bloom purple); the other, *recessive allele*, is overruled by the dominant allele and will only have an effect if both the alleles inherited from the parents are recessive. Weismann [**weismann1893germ**] performed experiments showing that only the genes in the germ line are passed on to the next generation, and no changes in the somatic cells may be passed on.

This concept of inheritance of genes from the germ line is now widely accepted. It opposes the concept of Lamarckian evolution, where acquired phenotypes were suggested to be inherited. Gregor Mendel is nowadays acknowledged as the founder of the field of genetics, and as the person who closed the gap in Darwin's work by explaining the mechanism of inheritance. Unfortunately, Mendel's work was widely ignored for at least 30–40 years after publication before the connection to Darwin's theory was made.

1.2.4 Modern Synthesis (Neo-Darwinism)

The term "Modern Synthesis" was coined by Julian Huxley in his 1942 book "Evolution: The Modern Synthesis" [**Huxley1942**]. This theory reflects the consensus theory of evolution, which combines Darwin's theory of evolution through natural selection with Mendelian genetics providing the mechanism for inheritance. According to the Modern Synthesis, evolution acts on the phenotypes via natural selection. The phenotypes are encoded through the genotype, and the genotype is the unit which is passed on to offspring following rules of Mendelian genetics.

Many people contributed to the establishment of this theory, in particular Theodosius Dobzhansky, who demonstrated that the Modern Synthesis theory holds up if one tests it in natural populations [**Dobzhansky1937**]. George Simpson showed that paleontological data, which is evidence for the process of evolution in the past, is in accordance with the Modern Synthesis theory [**Simpson1944**]. By now, the Modern Synthesis is by and large the currently accepted mechanism for evolution.

1.2.5 Deoxyribonucleic Acid (DNA)

The next step to more fully understand the process of evolution was to understand how the genes and the genotype are encoded.

In 1871, Friedrich Miescher published work on the isolation and identification of the deoxyribonucleic acid, or DNA [**Miescher1871**], which encodes the genotype. This research paved the way for Watson and Crick to decipher the structure of DNA in 1953 [**Watson1953**]. Based on images by Rosalind Franklin they identified it to be a double-stranded DNA helix.

The double-stranded DNA helix has the same structure in all known biological entities (eucaryotes, bacteria, archaea, and viruses) on Earth, and it consists of a sugar-phosphate backbone to which nucleotides, namely purine or pyrimidine bases, are attached. There are two purines, *adenine*(A) and *guanine* (G), and two pyrimidines, *cytosine* (C) and *thymine* (T), which are being attached. The successive order of these four nucleotides determines an individual's genotype. A few viruses are RNA-based, i.e. the RNA strand is composed of the same A, C, G, but use *uracil* (U) instead of T. Again, the successive order of these four nucleotides is the *genotype* of the virus.

We call the order of nucleotides a *genetic sequence*. In the double-stranded helix consisting of two nucleotide strands, one strand is complementary to the other. The nucleotides are paired in the helix according to strict compatibility rules such that A on one strand is complemented by T on the other strand whereas G is always complemented by C (see Figure 1.3). This means that if we know the sequence of one strand we can always determine the sequence of the other strand. Thus, only one strand is reported in the genetic sequence data. The analysis of genetic sequence data is the main focus of this book.

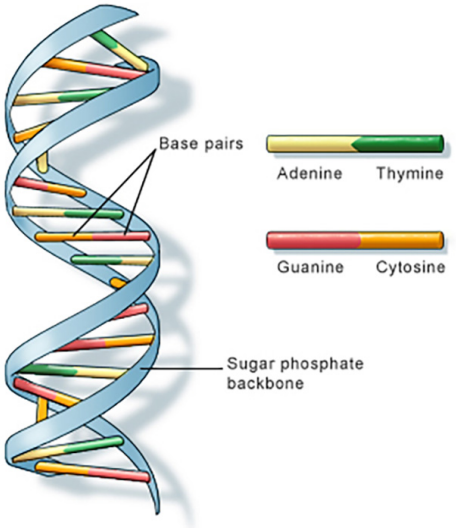


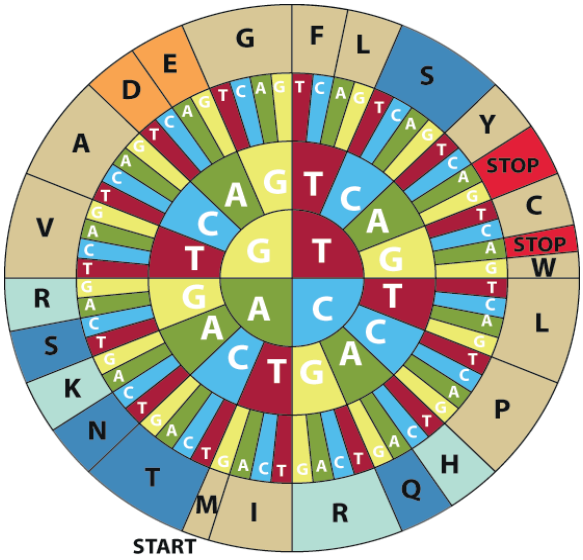
Figure 1.3: Figure adapted from [**USNat2015**]

1.2.6 The central dogma of biology

The central dogma of modern biology describes the flow of genetic information within a biological system. It states that information flows from DNA to RNA (Ribonucleic acid) through *transcription* and from RNA to proteins through *translation*, without any informational exchange flowing back. To go into some more detail, in parts of the genotype that code for proteins, each group of 3 successive nucleotides is referred to as a *codon*. Out of the $4^3 = 64$ possible codons, three codons terminate RNA translation. The remaining 61 codons translate into the 20 amino acids. Several codons can encode for the same amino acid. Amino acids are either abbreviated with one letter or with a three letter code, here we use the one letter codes only. Figure 1.4 depicts which codon translates into which amino acid. Proteins are characterised by the sequence of amino acids. Figure 1.5 illustrates the principle of information flow from DNA to RNA to proteins.

We note that only a small fraction of the genotype encodes for genes which are transcribed into RNA and then translated into proteins [**Lander2001**]. The genome also consists of non-coding regions where the nucleotides do not code for a gene. Instead, the non-coding regions may serve some regulatory function, or they may not have a function at all, or they may have some unknown function. We highlight that all these functions may play into determining the phenotype, i.e. the organism's appearance. In summary, according to the central dogma, the genotype determines the phenotype. According to Darwin, natural selection acts on that phenotype.

The famous evolutionary biologist Richard Dawkins proposed an analogy to baking [**Dawkins2009**], in which he compared the genotype to the recipe and the phenotype to the cake. This statement nicely points out a particularity of this connection – the recipe is not necessarily bad just because you failed to make the cake. In the context of cell biology, this means that even if there was an error during transcrip-



Amino acid	3-letter code	1-letter code
Alanine	Ala	A
Arginine	Arg	R
Asparagine	Asn	N
Aspartic acid	Asp	D
Cysteine	Cys	C
Glutamic acid	Glu	E
Glutamine	Gln	Q
Glycine	Gly	G
Histidine	His	H
Isoleucine	Ile	I
Leucine	Leu	L
Lysine	Lys	K
Methionine	Met	M
Phenylalanine	Phe	F
Proline	Pro	P
Serine	Ser	S
Threonine	Thr	T
Tryptophan	Trp	W
Tyrosine	Tyr	Y
Valine	Val	V

Figure 1.4: The *codon sun* shows which triplets of nucleotides encode for which amino acids. The nucleotide at the first position in the triplet is chosen from the inner most circle, the second and third are then picked from the second and the third circles from the centre, respectively. The amino acid encoded by the nucleotide triplets are displayed on the outer most circle as one-letter codes. Thus for example the triplet `TCG` would code for the amino acid Serine (S). Figure adapted from [**codonsun**]. The table shows the 1- and 3-letter code for the amino acids.

tion or translation, the next time transcription and translation might be perfectly normal again yielding to the expected protein.

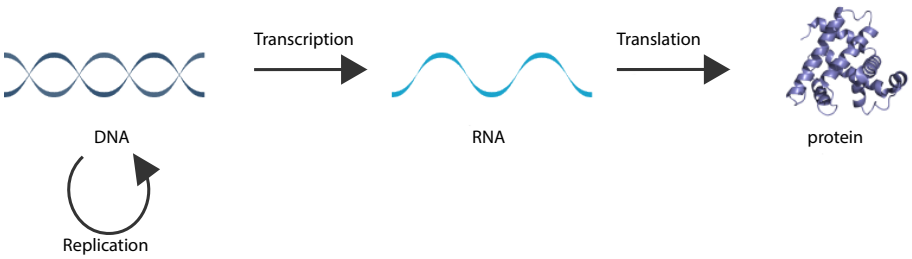


Figure 1.5: Figure adapted from [**adtbio2015**]

1.2.7 Exceptions to the central dogma

Commonly there is no rule without an exception, and this is also the case for the central dogma of biology. The actual information flow is more complicated than shown in Figure 1.5. In fact, reverse transcription of RNA to DNA is possible, and RNA may also replicate on its own. Figure 1.6 shows an updated picture of the flow.

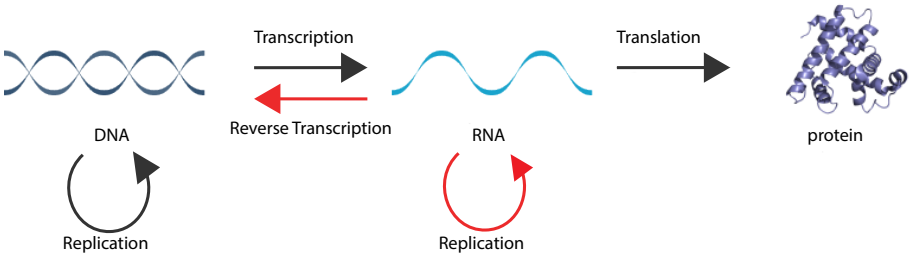


Figure 1.6: Figure adapted from [**adtbio2015**]

Reverse transcription [**Baltimore1970**] was discovered[1] in a specific class of viruses, the so-called *retroviruses*. For example, the human immunodeficiency virus (HIV) is such a retrovirus and stores all of its genetic information as RNA rather than DNA. In order for HIV to replicate, it has to transfer its genetic material into the host cell together with a reverse transcriptase enzyme, which reverse transcribes the RNA into DNA. The produced DNA is then incorporated into the host cell's genetic material by the HIV integrase enzyme, and is transcribed and replicated using standard host cell machinery.

RNA viruses may also replicate without the reverse transcription to DNA, i.e. RNA replicates on its own. In fact, this kind of replication is performed by many well-known viruses such as the influenza virus, ebolavirus, or poliovirus.

1.2.8 Errors in replication

The variation of genotypes between cells—and by extension between organisms—arises due to the error-prone replication of DNA during cell division. Errors may be introduced into the copied DNA strand during DNA replication when a cell prepares for division, due to the polymerase or due to external mutagens (such as chemicals, UV radiation, etc.). The following errors can happen during DNA replication resulting in a different sequence of nucleotides in the copied strand compared to the template strand:

- **Point mutation**: During the production of the DNA copy, a wrong nucleotide is built in with respect to the template, producing a copy where a single character is replaced by another (it has *mutated*), we refer to a point mutation as a *mutation* throughout this book;

[1] This discovery was a component of the research for which David Baltimore, Renato Dulbecco and Howard Martin Temin were awarded the 1975 Nobel Price in Physiology or Medicine.

- **Recombination**: Information is exchanged between two very similar sequences, such as chromosome pairs;
- **Insertion and deletion**: Insertion of extra information into the copy or loss of information in the copy. Areas in the genome with these alterations are called *indels*.
- **Repeats and inverted repeats**: In a repeat, a sequence of k nucleotides in the template strand is copied several times in the copied DNA strand. If the copy of length k occurs reverted in the copied DNA, we obtain an inverted repeat.

A detailed understanding of these mechanisms is not crucial for our purposes. However, it is crucial to note for this book that errors can occur during replication. The errors cause variation in genotypes, which in turn produces variation in phenotypes.

In this context, we introduce three important terms, *orthologs*, *paralogs*, and *homologs*. Two nucleotides in different sequences are orthologs if they have a shared ancestor nucleotide, with separation via speciation or the corresponding birth event when considering biological units different from species. Similarly, two nucleotides in different sequences are paralogs if they have a shared ancestor nucleotide, with separation via gene duplication. The set of paralogs and orthologs are referred to as homologs. The homologous nucleotides may differ in the two sequences due to errors in replication as pointed out above.

1.2.9 Darwin today

Put together, the molecular evolution at the genotypic level is promoted by two important elements: First, due to errors in replication, different individuals of the same population can have slightly different genotypes and thus phenotypes. Second, selection acts on this phenotype, leading to the propagation of certain variants in the population and the extinction of other variants.

To illustrate this, let us look at the interaction between the immune system of a host and a virus population, in particular how the immune system puts selective pressure on the virus population. The cells of the immune system use the proteins on the surface of the *virions* (i.e. the virus particles) entering the host to identify them as a foreign entity and attack them. The immune system tries to eliminate the virions and by doing so it exerts selective pressure on the virus population. The virions that possess a variant of the surface proteins which the host's immune system cannot recognise have an evolutionary advantage. This viral variant is linked to a mutation in the viral genotype: the virions possessing this mutation will be able to propagate while the virions without this mutation will be eliminated. After some time, the entire virus population will have the mutated genotype: evolution at the genotypic level was driven by selection at the phenotypic level.

In summary, the current understanding of evolution falls nicely into Darwin's framework. Combining our knowledge from different scientific disciplines (such as genetics

and molecular evolution) into the modern theory, and using Darwin's four components, we can summarise as follows:

Multiplication The replication of DNA leads to offspring. Replication in somatic cells produces more cells within an individual, replication in germ line cells may give rise to an offspring individual. In either case, the genotypes determine the phenotypes.

Variation Variation in the phenotype across individuals occurs due to mutations, recombination, insertions, deletions, or (inverted) repeats in the genotype at replication.

Heredity Heredity of the phenotype occurs due to the passage of DNA (or RNA in case of retroviruses) from parent to offspring via the germ line, and the phenotype of the offspring is encoded by its genotype.

Competition There are fitness differences across phenotypes, meaning the average number of surviving offspring depends on the phenotype of the parent individual.

This view completely ignores any possible impact of the environment on heredity. The last decades however saw more and more evidence for epigenetics. Epigenetics studies heritable mechanisms which change phenotypes beyond DNA modification. In particular, the activation pattern of genes and its changes was found to be such an epigenetic mechanism. Activation patterns vary due to variation in molecules binding to DNA, such as a methyl group binding to DNA (DNA methylation). The activation patterns may change throughout the lifetime of an individual due to environmental factors and can be inherited. Thus epigenetics studies mechanism that allow the inheritance of environmentally acquired phenotypes, which brings us back to Lamarck's explanation of evolution in giraffes.

One example where epigenetic effects can be seen is the phenotypic differences between identical twins. Genetically, these two individuals are identical (although there might be slight differences due to somatic mutations during development), however the two individuals might not look exactly the same due to epigenetic differences acquired throughout the twins' lifetime [**fraga2005epigenetic**]. Another example is a population of clonal bacteria – i.e. bacteria with the same genotype – in which some bacteria are more virulent than others, e.g. groups of identical salmonella bacteria that are split into those that are more adept at entering epithelial cells in the gut, those that release toxins, and those that reproduce quickly. Their DNA methylation patterns are different due to environmental effects [**casadesus2006epigenetic**].

1.3 An overview this book

The aim of this book is to provide readers coming from different backgrounds with an understanding of what kind of information is encoded in genetic sequences (Why it is a great opportunity to do genetic sequence analysis?). The reader will further

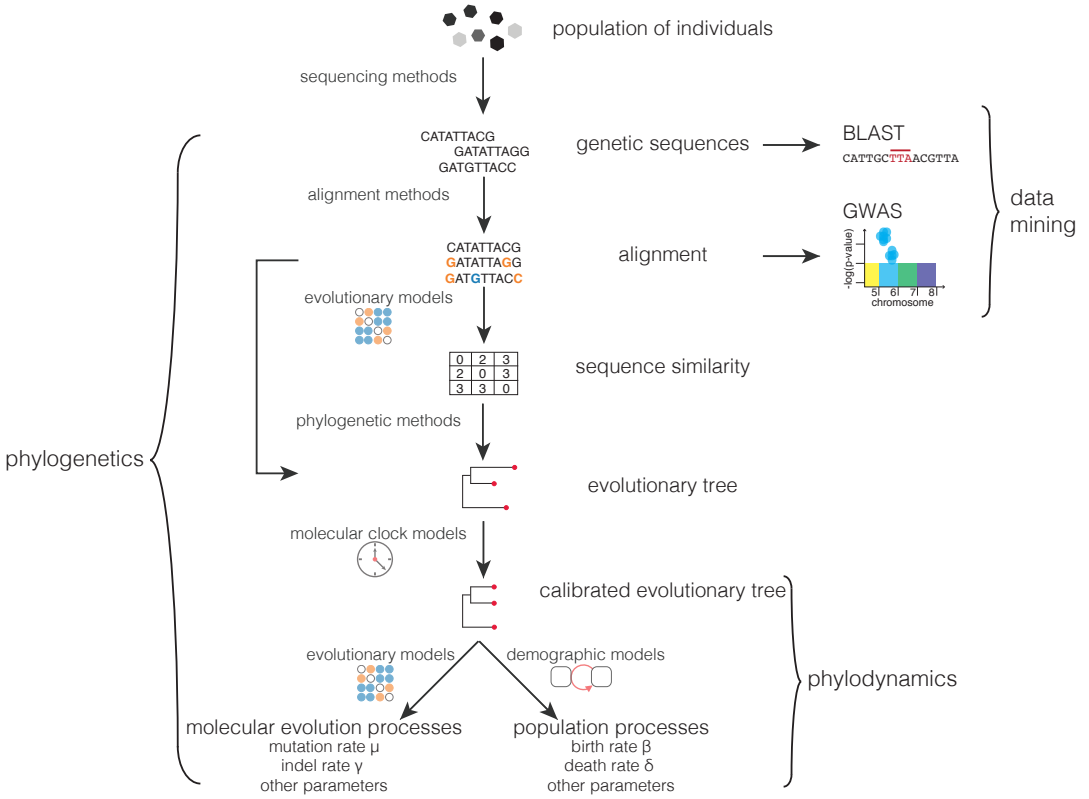


Figure 1.7: A guide through the book: from genetic sequences to statistical inference.

acquire the necessary skills to understand, plan, and perform genetic sequence analysis using data mining, phylogenetic and phylodynamic techniques (How to do an analysis?). Throughout, we provide examples of such analyses (What can be learnt from such an analysis?). We anticipate the needs of readers with different backgrounds by explaining concepts from biology and mathematics that belong to the basic education in one of these disciplines in the form of boxes. Furthermore, at the end of the introduction, we provide a short section on basic probability.

The very first step in a genetic sequence analysis is to obtain the sequence, i.e. to transform the genetic information encoded by the biological unit of interest to a format which we can use and analyse. In particular, we want to represent the DNA of the biological unit as a sequence of the letters A, C, G and T. To do so, we need to extract the DNA from cells and then use sequencing techniques to read and decipher this DNA. In the case of RNA viruses, RNA is extracted from virions, reverse transcribed into DNA, and then sequenced. The book therefore begins with an introduction to genetic sequencing technologies (Chapter 2). For the purpose of this book, these sequences fully characterise the individuals in our subject population, i.e. we do not consider epigenetic patterns.

The next step is to *align* the sequences to one another. In an alignment, different sequences are typically displayed in different rows, and their nucleotides are assigned

to columns or sites, such that the sites are orthologs. Differences in orthologous sites across individuals may determine phenotypic variation and contain information regarding evolutionary history and evolutionary and population dynamic processes. In Chapter 3, we will discuss different methods for aligning sequences in an optimal way. In addition, we will shortly explain the basic idea of *heuristic approaches* for alignments, meaning fast approaches which do not guarantee any optimality. In particular, we will introduce BLAST (Basic Local Alignment Search Tool, [**BLAST**, **Altschul1990dy**, **Altschul1997BLAST**]), with which one can find the homologs to a single sequence by comparing it against a huge library of other sequences. The BLAST algorithm is the first *data mining approach* discussed in this book, i.e. an approach aiming at finding associations within a large dataset, and is one of the most widely used bioinformatics algorithms.

Based on an alignment, we aim at extracting information it encodes, allowing us to answer biological questions. We continue with another data mining approach, namely *genome wide association studies* (*GWAS*, Chapter 4). In GWAS, the aligned genome sequences obtained from multiple individuals are considered together with certain traits of these individuals (e.g. increased risk of a certain type of cancer), to detect if certain genome variants or mutations are associated with those traits. We highlight that whole books have been written on such data mining approaches (e.g. [**aggarwal2015**]), and we only provide the main idea here.

Data mining approaches such as GWAS assume that each site in an alignment is an independent data point. This is valid for genome data stemming from e.g. different human individuals, since recombination quickly breaks up linkage between sites. However, in the absence of strong recombination, sites are linked due to a shared evolutionary history, i.e. sites evolved along the same phylogeny. As a consequence, individuals close in the phylogeny share more similarities (such as sharing the same nucleotides for a site) than distantly related individuals, and thus the sites are not independent data points. The need for acknowledging the phylogeny when analysing genotypes and their association with certain traits was explicitly spelled out in 1985 by Joseph Felsenstein [**Felsenstein1985comp**].

For the remainder of the book, we consider data where sites are non-independent due to a shared evolutionary history. Considering shared evolutionary history in fact allows us to not only assess dependencies in a statistically coherent way, but also allows us to describe the processes which gave rise to the alignment. The idea is to reconstruct phylogenies representing the evolutionary history leading to the sequences, and then to aim at understanding the evolutionary and population dynamic processes giving rise to the phylogenies.

In Figure 1.7, the left bracket shows the part in sequence analysis falling under the area of phylogenetics. Phylogenetics goes back to 1837 when Charles Darwin drew a sketch of a phylogenetic tree in his notebook, shown in Figure 1.8. However, the computational birth of this field only occurred in 1957, when Michener and Sokal published a paper on a computational tool that allows the reconstruction of a phylogenetic tree from sequencing data [**MichenerSokal1957**]. This first tool was based

on the simple principle that similarity between two individuals indicates that they are close relatives, whereas dissimilar individuals indicates a more distant relationship. Joseph Felsenstein revolutionised phylogenetic tree inference in the 1980s by introducing statistical tools allowing for the maximum likelihood and Bayesian approaches to be applied to phylogenetics [**Felsenstein1981**]. Initially, phylogenetics was developed and used in macroevolution, while now a range of other fields, as presented in Section 1.1, also employ phylogenetic concepts.

In phylogenetics, we aim at reconstructing phylogenies (Chapter 6) using molecular evolution models (Chapter 5) based on the sequencing data or morphological data. Based on the phylogenies, we can further investigate processes occurring along the phylogenetic tree. In particular, we can obtain an understanding of how genes and genotypes change along the phylogeny through time, i.e. understand molecular evolution processes (Chapter 7). Furthermore, using the phylogeny, it can be assessed how traits change along the phylogeny, with the phylogeny representing genotypic change (Chapter 8). Such analyses shed light onto phenotypic evolution processes. In particular, as in a GWAS, the relationship between the genotype and the phenotype is addressed, now with statistical tools acknowledging dependencies between sites. Importantly, so far, all molecular or phenotypic evolutionary processes are assumed to not influence the tree, instead they occur on a given tree. Furthermore, the discussed approaches all assume that we can represent the phylogeny as a tree. At the end of the book we will outline scenarios when this tree assumption is violated and networks are required.

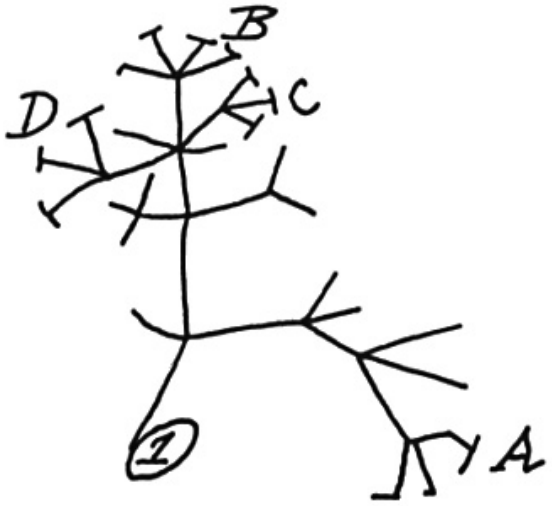


Figure 1.8: Sketch of an evolutionary tree (Darwin, from his 1837 notebook) [**darwinnotebook**].

The field of phylodynamics, denoted in Figure 1.7 by the right bracket, studies how processes give rise to and shape the phylogenies. Phylodynamic approaches fit population dynamic models (e.g. models of speciation/extinction, models of pathogen transmission, etc) to the reconstructed phylogenies. In particular, these approaches

take into account that the phenotype may influence the shape of the phylogeny. This allows e.g. to quantify fitness differences across individuals and thus to quantify selective advantages of certain phenotypes, or to assess the global migration pattern of individuals. Phylodynamic methods require a time tree – a tree with branches in units of time. Charles Darwin sketched such a time tree (Figure 1.9). The first key papers on phylodynamics of macroevolution appeared in the 1990s [**Nee1994reconstructed**, **Harvey1994**]. The field only started flourishing though after the key publication of Grenfell [**Grenfell2004**] considering the phylodynamics of pathogens. Phylodynamics is discussed in Chapters 9-10.

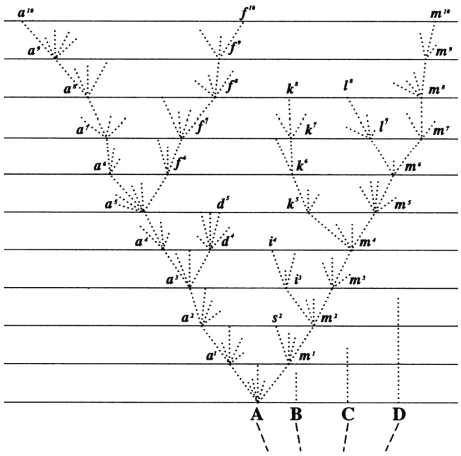


Figure 1.9: Sketch of a time tree with horizontal lines being drawn every 1000 generations (the only figure in the Origin of Species by Darwin, 1859) [**darwin1859origin**].

All approaches presented in this book so far assume that the phylogeny can be represented by a tree, i.e. no reticulation processes occurred. This means that there is complete linkage across sites of an alignment. However, there is increasing evidence that there is no "Tree of Life" but instead there is a "Network of Life" due to hybridization, horizontal gene transfer, and recombination, which cause some sites to have different evolutionary histories. In other words, some sites are not linked. We end the book by presenting very basic concepts regarding phylogenetic networks (Chapter 11) which are required in intermediate scenarios between phylogenetic trees – where sites are completely linked, and GWAS, – where sites are independent. For readers interested in more details on networks see e.g. [**huson2010phylogenetic**].

1.4 Basic definitions and concepts of probabilites

Formally, a *probability* is defined as a function, here denoted with a capital P, that maps a set to a number between 0 and 1 (including 0 and 1). The set is composed of a number of mutually exclusive alternatives: maybe the possible outcomes of an experiment, or the possible values of some unknown quantity.

A simple example is throwing a six-sided die one time. The possible outcomes are

1, 2, 3, 4, 5 or 6. Thus a result of throwing a die once can only be either 1, 2, 3, 4, 5, or 6. We call $\Omega = \{1, 2, 3, 4, 5, 6\}$ the *state space* of the die throwing example. An event is a subset of this state space. In our example the event 'Obtaining 1 when throwing a die once' is denoted as $\{1\}$. If the die is fair, i.e. each outcome is equally likely, the probability to throw a 1 is 1/6. Formally, this is denoted as:

$$P(\{1\}) = 1/6$$

Another, a bit more interesting example, is throwing the fair die twice. There are 36 different possible outcomes, the state space is $\Omega = \{(1,1), (1,2), \ldots, (1,6), (2,1), \ldots, (6,6)\}$, and each single result has the probability 1/36. If we are interested in the sum of the two numbers, we can calculate the probability of obtaining this sum by counting all possible results that lead to that number and divide it by all possible results, e.g.:

$$P(\text{sum=5}) = P(\{(1,4), (2,3), (3,2), (4,1)\}) = 4/36.$$

We call a function that is applied to the possible outcomes, such as summing the two numbers in the example above, a *random variable* and denote it with a capital letter, e.g. X. Unless otherwise noted, X maps into the real numbers. If the set of outcomes is a discrete set, then the associated random variable is said to be discrete. In our example, $X((i,j)) = i + j$ is a discrete random variable. Out of convenience one writes $P(X = x)$ as an abbreviation for "the probability of the set of outcomes for which we obtain x when we apply the function X to the possible outcomes of the random experiment". The value x is also called a realization of X. In the two-dice example this means

$$P(X = x) = P(\{(i,j) \in \{(1,1), \ldots, (6,6)\} \text{ for which } i + j = x\}).$$

The state space in the dice throwing example is discrete. This is the case for many random experiments, e.g. tossing a coin, roulette or even the set of possible hands in poker. In these experiments, it is possible to assign a probability > 0 to an outcome, such as to assign 1/6 to throwing a 1 with a fair die.

There are also situations where the state space is continuous. A classical example is a person's height. The probability that a person is exactly 1.83m tall is 0. But the probability that a person's height ranges between 1.83 and 1.84m is > 0. In the case of a continuous random experiment, one denotes its *probability density* instead of the probability of each individual outcome. To obtain the probability of a continuous random variable taking the values of a specific set, one has to integrate the probability density over this set. E.g. denote height with the continuous random variable Y, and the probability density of Y with f_Y. The probability that a person's height ranges between 1.83 and 1.84m is

$$P(1.83 < Y < 1.84) = \int_{1.83}^{1.84} f_Y(x)\mathrm{d}x.$$

Two important measures are commonly reported for every distribution. These are the *mean* and the *variance*. The mean is the average value one would expect as the outcome of a series of random experiments. For a random variable X it is denoted by EX. If the random variable is discrete, i.e. has a discrete state space $\mathcal{X}$, the mean is defined as the sum of all possible results weighted by the probability to obtain these result:

$$EX = \sum_{x \in \mathcal{X}} xP(X = x) \tag{1.1}$$

If the random variable is continuous, i.e. has a continuous state space $\mathcal{X}$, and has a distribution f_X, the mean is defined as the integral of all possible values weighted by its probability density:

$$EX = \int_{\mathcal{X}} x f_X(x) dx \tag{1.2}$$

Note that the mean is not necessarily a value that one can actually obtain as a result of the random experiment. E.g. the mean outcome when throwing a six-sided fair die is 3.5.

The variance is denoted by Var and is the average deviation from the mean, i.e. it measures how dispersed the distribution is. For a discrete random variable X with state space $\mathcal{X}$, the variance is defined as:

$$VarX = \sum_{x \in \mathcal{X}} (x - EX)^2 P(X = x) \tag{1.3}$$

Similarly, the variance of a continuous random variable is defined as:

$$VarX = E(X - EX)^2 = \int_{\mathcal{X}} (x - EX)^2 f_X(x) dx \tag{1.4}$$

Very often, instead of the variance, the standard deviation is reported. This is simply the square root of the variance:

$$sd(X) = \sqrt{VarX} \tag{1.5}$$

1.4.1 Notation and nomenclature

According to the above definitions, the probability that some random variable X takes some value x should always be written $P(X = x)$. However, when there is no ambiguity with respect to the random variable the probability is referring, it is common to write $P(x)$ instead. Similarly, when there is no ambiguity with respect to the value, $P(X)$ may also be used.

Of course, extreme care must be taken when using simplified notation, as misunderstandings can easily result. Thus, whenever possible, we will endeavour to use the formal notation in this text.

1.4.2 Mathematical areas using probability

The concept of a probability is used in different mathematical areas:

1. In *probability theory*, one studies the rules to calculate probabilities of events and random variables given the underlying distribution is known.
2. *Stochastic processes* study successively repeated random experiments and the overall behaviour of the realizations of these processes (see Markov chains in Box 17 and Brownian motion in Box 23 as examples for stochastic processes). Also here, we assume that the underlying probability distribution is known.
3. In *statistics*, one starts with a set of observations and tries to deduct information on the underlying distribution.
    - In *parameter estimation*, one uses the observations to estimate a parameter such as the probability to observe a particular outcome.
    - In *hypothesis testing*, one uses the observations to test whether a specific hypothesis –formalized as a specific stochastic model giving rise to a probability distribution– is supported by the data. Given such a hypothesis, one can calculate how likely it is to obtain the observed outcome or a more extreme outcome. This probability is called the p-value (see Box 1). This value will be important throughout the book. The p-values will be employed to test if the given data (i.e. the outcome of a probabilistic experiment) indeed evolved under the assumed stochastic model.

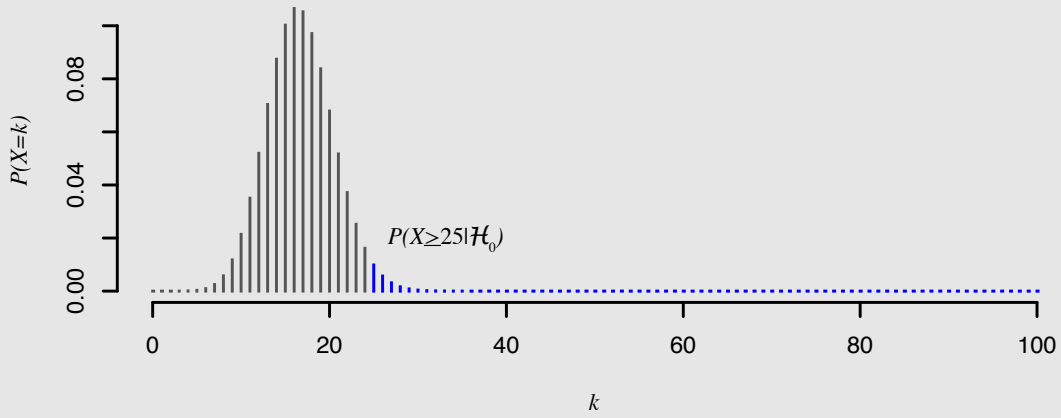
Box 1: The p-value

Definition:

As introduced in Section 1.4, we use capital X to represent a random variable and small x as its realisation. Let us assume a null hypothesis $\mathcal{H}_0$, which, in mathematical terms, corresponds to some statement about the potential probability distribution of X. The probability for the outcome being x or more extreme given the null hypothesis is true is called the *p-value* for x under the null hypothesis. If more extreme means greater than x, the p-value is defined as $P(X \geq x|\mathcal{H}_0)$. In case more extreme means smaller than x, the p-value is defined as $P(X \leq x|\mathcal{H}_0)$.

Example:

As an example, let us consider rolling a six-sided die $n = 100$ times. Let the random variable X be the number of sixes obtained. The null hypothesis $\mathcal{H}_0$ is that the six-sided die is fair, i.e. the probability that one obtains a six in a single die rolling is $1/6$. Suppose we obtained $x = 25$ as a result of our experiment, and we want to determine the p-value.
The probability to obtain k sixes out of n independent die rollings is $\binom{n}{k}\frac{1}{6}^k\frac{5}{6}^{n-k}$, i.e. X follows a *binomial distribution* with parameter $1/6$. The distribution is displayed for $n = 100$ in the graph below. The p-value for our realization ($x = 25$) is $p = P(X \geq 25|\mathcal{H}_0) = \sum_{k=25}^{100}\binom{100}{k}\frac{1}{6}^k\frac{5}{6}^{100-k} = 0.022$. The blue colored area are the events at least as extreme as our outcome ($x = 25$).



Usage:

One can pre-define a *significance level*, commonly denoted as α, which is typically set to $\alpha = 0.05$ or $\alpha = 0.01$. A test having a significance level α means that the cumulative probability of a false positive is α. If the p-value for an obtained outcome is below α, the null hypothesis is rejected at the level α and is said to differ *significantly* from the null hypothesis. One also often says that such an observation is in the *tail* of the distribution. In the dice-rolling example, we encounter a special situation. The die is loaded not only when there are too many sixes but also when there are too few. Such a case is referred to a *two sided problem*. The significance level is then divided by 2 and the hypothesis is rejected if $P(X \geq x|\mathcal{H}_0) < \alpha/2$ or $P(X \leq x|\mathcal{H}_0) < \alpha/2$. α is again the significance level, and $\alpha/2$ is referred to as *rejection threshold*. For further information on the interpretation of p-values, see e.g. [**Dorey:2010jv**]. Note that the p-value is also sometimes referred to as P value, p value, or in one of the forms without italicisation of the letter "p".

1.4.3 Conditioned probability

The concept of conditioned probability is easiest to understand for discrete random variables. Given a random experiment with discrete state space Ω and two sets of outcomes $A, B \subset \Omega$, with $P(B) \neq 0$. The conditioned probability is the probability that event A happens, given we know that B happened, and can be calculated as:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}.$$

$P(A \cap B)$ is also called the joint probability, i.e. the probability of both event A and B happening. Intuitively, we can understand that the above formula holds by considering $P(A \cap B) = P(A|B)P(B)$. Indeed, we can calculate the joint probability by first evaluating the probability of event B happening, and then evaluating the probability of event A happening when knowing that B happened.

As a toy example, we come back to our die experiment from above. Imagine you want to determine the probability of scoring at least a sum of 10 when rolling the die twice (event A) while you know that doublets have been scored (event B). We have set $A = \{(4,6),(5,5),(6,4),(5,6),(6,5),(6,6)\}$ and thus $P(A) = 6/36$. Doublets are $B = \{(1,1),(2,2),(3,3),(4,4),(5,5),(6,6)\}$ and thus $P(B) = 6/36$. One only achieves a score of at least 10 with doublets when throwing (5,5) or (6,6). This means the probability of scoring a sum of ≥ 10 given doublets is:

$$P(A|B) = 2/6 = 1/3$$

As we have $A \cap B = \{(5,5),(6,6)\}$ and thus $P(A \cap B) = \frac{2}{36}$, we obtain the same result when applying the right hand site of the conditioned probability equation above:

$$\frac{P(A \cap B)}{P(B)} = \frac{2/36}{6/36} = 1/3$$

When looking at continuous random variables, the concept of conditioned probabilities becomes more difficult. We cannot go into detail here but refer the interested reader to a textbook on probability theory such as [**Williams2001**].

1.4.4 Some common probability distributions

In Box 1, we already encountered the binomial distribution. We now introduce three more probability distributions which will be important throughout the book, namely the normal distribution (Box 2) and the χ^2-distribution (3) as well as the hypergeometric distribution (Box 4).

Box 2: The normal distribution

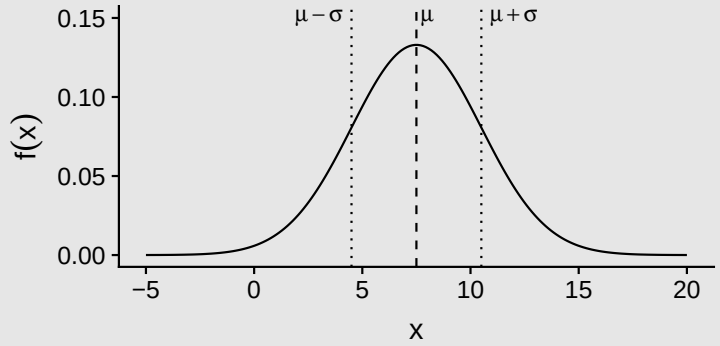
The *normal distribution*, also known as the *Gaussian distribution* or colloquially as the *bell curve*, is a density defined on the continuous values of a single real random variable X. Its probability density function is

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

It has two parameters, μ and σ, which are respectively the mean and standard deviation for this distribution, i.e:

$$\begin{aligned} EX &= \mu \\ VarX &= \sigma^2 \end{aligned}$$

The following figure displays the probability density function (PDF) for a normal distribution with $\mu = 7.5$ and $\sigma = 3$:



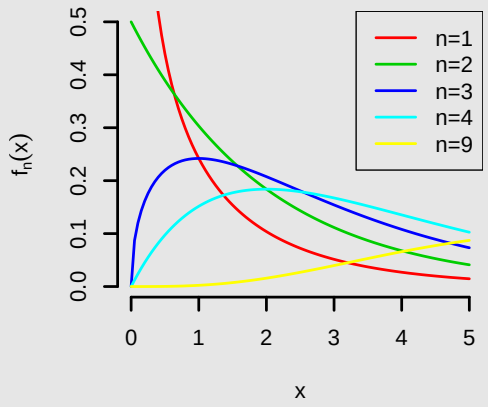
The normal distribution is highly ubiquitous in statistics, principally as the result of the *central limit theorem*. Informally, this theorem implies that if we define the random variable Z_n to be the average of n independent samples from *any* single distribution with mean m and finite variance s^2, the distribution for Z asymptotes to a normal distribution centred on m and with variance s^2/n as the sample count n becomes large.

Box 3: The χ^2-distribution

The χ^2-*distribution* (pronounced chi-square distribution) plays an important role in statistics, more precise in statistical testing. One obtains this distribution through the convolution of distributions of independent and identically distributed random variables Y_i, that are normally distributed with mean μ and variance σ^2 (see Box 2), in mathematical notation $Y_i \sim \mathcal{N}(\mu, \sigma^2)$, in the following way:

$$X_n^2 = \sum_{i=1}^{n} \frac{(Y_i - \mu)^2}{\sigma^2} \tag{1.6}$$

The distribution of X_n^2 is called χ^2-distribution with n degrees of freedom (in short χ_n^2) and takes different shapes as shown in the following figure.



Its density function is

$$f_{X_n^2}(x) = \frac{1}{2^{n/2}\Gamma(n/2)x^{\frac{n}{2}-1}e^{-\frac{x}{2}}} \tag{1.7}$$

The Γ-function is further explained in equation 5.25 in Box 20. The mean of this distribution is

$$EX_n^2 = n \tag{1.8}$$

and its variance is

$$VarX_n^2 = 2n \tag{1.9}$$

The χ_n^2 distribution is an approximation to many distributions that appear in statistical testing. Thus, given the exact distributions are not known or cannot be calculated, one can use the p-value for an output of a statistical test using a χ_n^2-distribution. The p-values of a χ^2-distribution can be found in χ^2-tables or can be directly computed with statistical programs such as R [**R:18**].
We will see applications of the χ_n^2-distribution when calculating p-values in Box 10 and more generally in Chapter 7 on Statistical Testing. More information on the χ^2-distribution can be found in [**SokalRohlf2012**].

Box 4: The hypergeometric distribution

The *hypergeometric distribution* is a discrete probability distribution that describes the probability to draw i balls of one type when in total k balls are drawn without replacement from an urn with n balls of two different types, namely r balls are colored red and s balls are colored black, with $n = r + s$. As above, we assign a capital letter for the random variable "number of red balls amongst k drawn balls", say R_k.

In the above sketch, the urn has $n = 7$ balls of which $r = 3$ balls are colored red and $s = 4$ balls are colored black; we draw $k = 3$ times, and $i = 1$ ball is colored red. What is the probability of drawing exactly i red balls? In general, we can calculate this probability by dividing the number of possibilities to obtain exactly i red balls when k balls are drawn through the number of possibilities to draw k balls out of the urn with n balls.
In order to count these possibilities, we use the *binomial coefficient* which is defined as

$$\binom{a}{b} = \frac{a!}{b!(a-b)!}$$

The expression $a!$ is called factorial and defined as

$$a! = a \times (a-1) \times (a-2) \times \ldots \times 2 \times 1$$

There are $\binom{r}{i}$ possibilities to obtain i red balls, and $\binom{s}{k-i}$ possibilities to draw $k-i$ balls from the black balls in the urn. Further, there are $\binom{n}{k}$ ways to draw k balls out of the urn without replacement. Thus, the probability to draw i red balls amongst the k drawn balls is:

$$P(R_k = i) = \frac{\binom{r}{i}\binom{s}{k-i}}{\binom{n}{k}} \tag{1.10}$$

The mean of this hypergeometric distributed random variable is

$$ER_k = \frac{kr}{n} \tag{1.11}$$

and its variance is

$$VarR_k = \frac{kr(n-r)(n-k)}{n^2(n-1)} \tag{1.12}$$

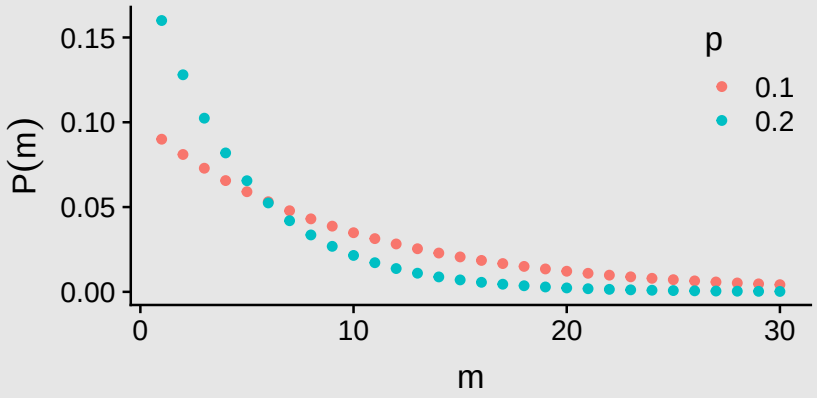
The hypergeometric distribution finds many applications, and will be used in this book in Fisher's exact test (Box 10, and Chapters 4.1.2, and 8.1) .

Box 5: The geometric distribution

Consider a repeatable experiment consisting of two outcomes: success, which occurs with a fixed probability p and failure, which occurs with a fixed probability $1-p$. Now suppose we repeat this experiment until we observe the first success. Defining X as the random variable representing the repetition on which the first success occurs, the probability of this being equal to m is given by the *geometric distribution*,

$$P(X = m) = (1-p)^{m-1}p.$$

This is the product of the probability for the first $m-1$ failures with the probability of the success in repetition m. The form of the geometric distribution is shown in the figure below, for success probabilities 0.1 and 0.2:



The mean of the distribution is simply the inverse of the success probability, i.e. $1/p$.

Note that the process behind the geometric distribution is *memoryless*. That is, the outcomes of future experiments do not depend on the observed outcomes of previous experiments. Geometric distributions arise often in the context of discrete, memoryless processes.

2 Sequencing

Genetic information serves as a blueprint for building an organism. This genetic information is contained in each cell of the organism in the form of DNA (or RNA for some viruses). The language of this blueprint is universal for all living organisms on Earth, namely each individual is characterised by a particular order of building blocks, the nucleotides, within the DNA double helix or an RNA single or double strand. A genetic sequence is a digital (human- and computer-readable) excerpt from the blueprint of an organism, representing the order of nucleotides A, C, G, T in this blueprint (U instead of T in the case of RNA) . In other words, a sequence is a string composed of letters A, C, G, T/U. The sequence may represent the whole genome of the individual, or just parts of the genome such as single genes.

This chapter is a short introduction into DNA sequencing aiming to give you an idea of the available platforms as well as their advantages and disadvantages. Sequencing of RNA requires reverse transcription of RNA into DNA, and then DNA sequencing is performed. Note that RNA is not directly sequenced due to RNA degradation, since it is not as stable as DNA.

We provide an overview of how to prepare a biological sample from an individual for sequencing, and then discuss the three generations of DNA sequencing platforms that are currently available for obtaining a genetic sequence. No platform can be labelled as the best, as each of the platforms fulfils different specific needs, with respect to e.g. the read length and coverage, and each has its own drawbacks[1].

In order to start the sequencing process, we first need to acquire the genetic material which we want to *sequence* or *read*. A group of cells is taken from the organism or a specific tissue, whereupon the genetic material is *isolated* from these cells. DNA isolation requires to break the cell membranes and molecules within the cell, and then separate the non-DNA fragments from the DNA molecules. Commercial kits employing chemical methods are available to do the breaking up step, and then a centrifuge can be used to separate the DNA molecules from the non-DNA fragments. In order to isolate RNA from virions, analogous steps are taken, however the procedure is in general more complicated as RNA is a less stable molecule and more prone to degradation. Once the RNA is isolated from the virion, it is directly reverse transcribed to complementary DNA (cDNA) to assure as little loss of information as possible due to RNA degradation.

[1] For a comparison of the advantages and disadvantages of different sequencing platforms we recommend taking a look at the extensive tables `http://www.molecularecologist.com/next-gen-table-2-2014/` and `http://www.molecularecologist.com/next-gen-table-4-2014/`.

Second, if the DNA/cDNA is too long for sequencing, we need to either selectively amplify parts of the DNA/cDNA, or *cut* the DNA/cDNA into smaller pieces. Mechanical methods, such as sonification, break up the DNA in random places, and restriction enzymes found in bacteria cut the DNA in a non-random fashion. The DNA/cDNA fragments which will be sequenced are called *templates*.

The next steps are unique to the particular sequencing technology, thus further details will be discussed when the individual technologies are presented. All technologies have in common that, when preparing our sample for sequencing, we need to physically *separate* the templates. This step is required in order to separate the signals coming from the sequencer in the sequencing step. In addition, for the first and second generation of sequencing methods, the signal from a single instance of a template would be too weak to be captured by the sequencer's detection technology. Thus, an *amplification* step is also needed, where the number of each template present is multiplied through copying using *PCR* (*polymerase chain reaction*) to intensify the signal, see Box 6 for details on PCR.

Box 6: Polymerase Chain Reaction (PCR)

The *Polymerase Chain Reaction (PCR)* is a chemical reaction using the *template molecules, the primers, free nucleotide molecules (dNTPs = deoxyribonucleoside triphosphate)* and the *DNA polymerase enzyme*.
The *primers* are short single-stranded DNA segments complementary to the parts (usually terminal ends) of the template molecules. One way to ensure complementarity is by using selected segments from the known sequence of a genome and design the primers based on these segments. Another way is to ligate (paste) adaptors (short double-stranded DNA fragments) to the ends of the templates.
The *DNA polymerase enzyme* is an enzyme which essentially occurs in all life and is central in DNA replication by synthesising the complement to the template DNA. For PCR, bacterial polymerase is used, even though it is very error-prone. Other polymerases, for example human polymerase, which have much lower error rates, cannot be used though since they denature at the high temperatures required for the reaction (around 90°C).
PCR starts by heating up the mixture to a high temperature (90 °C) in order to split the double-stranded template molecules into single-stranded templates. The mixture is then slowly cooled down to allow the primers to bind to the templates. The polymerase enzyme then binds to the place where the template overhangs the primer, i.e. where the primer ends and the single-stranded template molecule continues. Polymerase synthesises the complement to the template DNA by progressively adding free nucleotides, complementary to the template molecule, to the primer's end, until the end of the template, thereby creating a double-stranded DNA, while copying the template molecule.
This procedure of heating up, cooling down, primer extension is repeated several times until the template molecule is amplified in sufficiently high numbers.

After the templates are physically separated and amplified we can start the actual *sequencing*. Sequencing technologies provide us with sequences of the nucleotides present in the templates. Many of these technologies produce only short fragments of the full template sequences, so a bioinformatics challenge is to assemble these sequence fragments such that we get back the full sequence of the genome of our cells of interest (Section 3.2).

Table 2.1: Overview of the three generations of sequencing technologies.

Metric	Generation		
	First	Second	Third
Speed (bp[†]/hour)	10^5	3×10^{10}	3×10^8
Maximum read length (bp)	1'000	650	30'000[*]
Reads produced in parallel	96	6×10^9	10^7
Amplification step	yes	yes	no
Error rate	low	medium	high

[†] Base pairs. [*] The average size of fragments sequences by PACBIO is 30'000, however, some fragments can be 100'000 nucleotides long.

All sequencing methods have some errors, and bioinformatic tools aim to correct for these. We will not discuss these correction methods further in this book, see e.g. [**elloumi2017**, **heydari2017**] for further details.

Box 7 provides a short vocabulary useful for understanding the following descriptions of sequencing techniques and for understanding research articles concerned with this topic in general.

Box 7: Sequencing glossary

Template: a part of the DNA from an individual which will be sequenced;

Primer: a short DNA fragment complementary to (one end of) the template intended to be read. The primer is essential for starting DNA polymerisation and thus the sequencing reaction;

Library: a mixture of different templates ready for sequencing;

Sequence: the order of building blocks (nucleotides) in a template or gene or genome;

Sequencing: the act of determining the order of individual building blocks of the template;

Sequencing run: one round of operation of the sequencing machine;

Read: a single instance of output from a single sequencing run - it is a sequence representing a partial or whole template, it often still containing errors;

Sequencing error: the difference between the sequence retrieved via sequencing and the template sequence;

Assembly: the process of aligning and merging reads in an attempt to reconstruct the original sequence of the genome.

2.1 Sequencing methods

Currently, we distinguish between three generations of sequencing methods. Each of them is based on slightly different sequencing/signal detection principles. A very brief summary of their performance is given in Table 2.1.

2.1.1 First generation: Sanger sequencing

Sanger sequencing is the oldest and most reliable sequencing method so far[2] [**Sanger1977**]. Despite its age it still serves as the gold standard of sequencing. Researchers resort to this method if an observation needs to be verified (e.g. a variant of the genome needs to be confirmed as real and not as a sequencing error).

The throughput of this method goes up to only 100kbp/hour (kbp = kilo base pairs = 1000bp), the read length is up to 1000 nucleotides, and typically one can simultaneously sequence up to 96 different templates per run (limited by the number of wells on the reaction plate). One significant disadvantage of Sanger sequencing is that it requires a lot of laborious manual work, such as usage of bacteria for fragment separation and individual PCR (polymerase chain reaction) reactions for different parts of the genome.

Separation

Sanger sequencing typically uses bacteria for template separation, primarily *E. coli*. An alternative to plasmid vectors are viral or cosmid (a hybrid plasmid) libraries. The core idea remains the same though. Recombinant DNA molecules, each composed of a vector (e.g. a plasmid) plus an inserted DNA fragment (i.e. the template), is put in a solution of bacteria. The bacteria take up the recombinant DNA molecules with the inserted template. The concentration of recombinant DNA molecules in the solution is selected such that most bacteria pick up one molecule. The bacteria then multiply on a plate, each creating a single colony. When a bacterium multiplies, it passes on a copy of its genetic information to each of the daughter cells. Thus, each daughter cell gets a copy of the chromosomal DNA and in addition, it also inherits a copy of the recombinant DNA [**lodish2000molecular**]. Each of the colonies represents a clonal population stemming from a single bacterium that took up a single template. In order to keep the templates separated, the individual colonies are transferred into reaction tubes one by one. This means that the colonies are picked manually one by one from the plate and put into separate tubes.

Amplification

Next, one needs to isolate the DNA from the bacteria and amplify it further. Further amplification of the template is necessary as the sequencing method is not sensitive enough to detect the signal on a small number of template copies. For this, bacteria are lysed, i.e. the membranes are destroyed, and the proteins are denatured using high temperature. Amplification is done using a PCR step, see Box 6, employing primers complementary to the ends of the vector into which the DNA template was inserted.

[2] The Nobel prize in chemistry in 1980 was awarded to Frederick Sanger and his colleagues for their contribution to the effort to decipher the genetic code.

Sequencing

Once the amplification is finished, multiple copies of the same template are present within each tube. Now we need to read the sequence of the template, which requires a way to visibly distinguish nucleotides and their distinct positions in the sequence. The reaction for reading the templates starts from a primer, and is again based on a PCR reaction. However, this time, the PCR is carried out with a mixture of normal nucleotides (normal dNTPs) and chain-terminating inhibitors (dideoxyribonucleoside triphosphates - ddNTPs). In the first publication of this method in 1977, four different steps have to be undertaken [**Sanger:1977**]. The inhibitors are chemically modified to disallow further nucleotide attachment, and therefore act as polymerisation terminators/inhibitors. These molecules also contain radio-labelled phosphor. The content of each tube is separated into four parts, such that four sequencing reactions can be performed separately. In the first reaction, the normal nucleotides are mixed with the chain-terminating variant of adenine (ddATP). Whenever this nucleotide is built in by the polymerase, the chain is stopped. After several hours, one thus obtains a mixture of short sequences of different lengths. This reaction is repeated with chain-terminating variants of C, T, and G as well. The resulting mixtures are then processed using gel electrophoresis. Shorter sequences move farther on the gel plates. Due to the radioactive labelling, one can then observe how far the different sequences moved. This allows reading out the ending nucleotide for a sequence of any length, and thus the template sequence can be reconstructed. Figure 2.1 illustrates this process.

Fluorescent labels

To speed up the process and get rid of dangerous radio-labelling, the chain-terminating nucleotides from the original setup were later replaced by nucleotides labelled with different fluorophores. Fluorophores are molecules that emit light at a specific wavelength when excited. As the DNA polymerisation on the template begins, different nucleotides will be attached to the end of the primer. Since the tube contains a mixture of unlabelled and labelled chain-terminating nucleotides, the labelled inhibitors will terminate the primer extension at a random positions as the template is being copied, resulting in the production of many partial copies of the template with varying lengths and terminal labels.

Afterwards the newly produced partial copies of a template are separated on gel using an electric current. The shorter copies will travel further in the gel as are smaller and thus move more easily through the gel, while the longer ones are larger and do not travel as far. The different copies have discrete lengths, so fragments of the same length will end up close to one another, with shorter and longer fragments being lower, or higher on the gel, respectively.

Next, a laser is used to excite the light emitting molecules along the length of the gel. A detector then reads the corresponding labelling of the nucleotide at each particular position in the gel based on the colour (wavelength) of the light that is emitted. For each position, one can determine the most common colour and thus reconstruct the

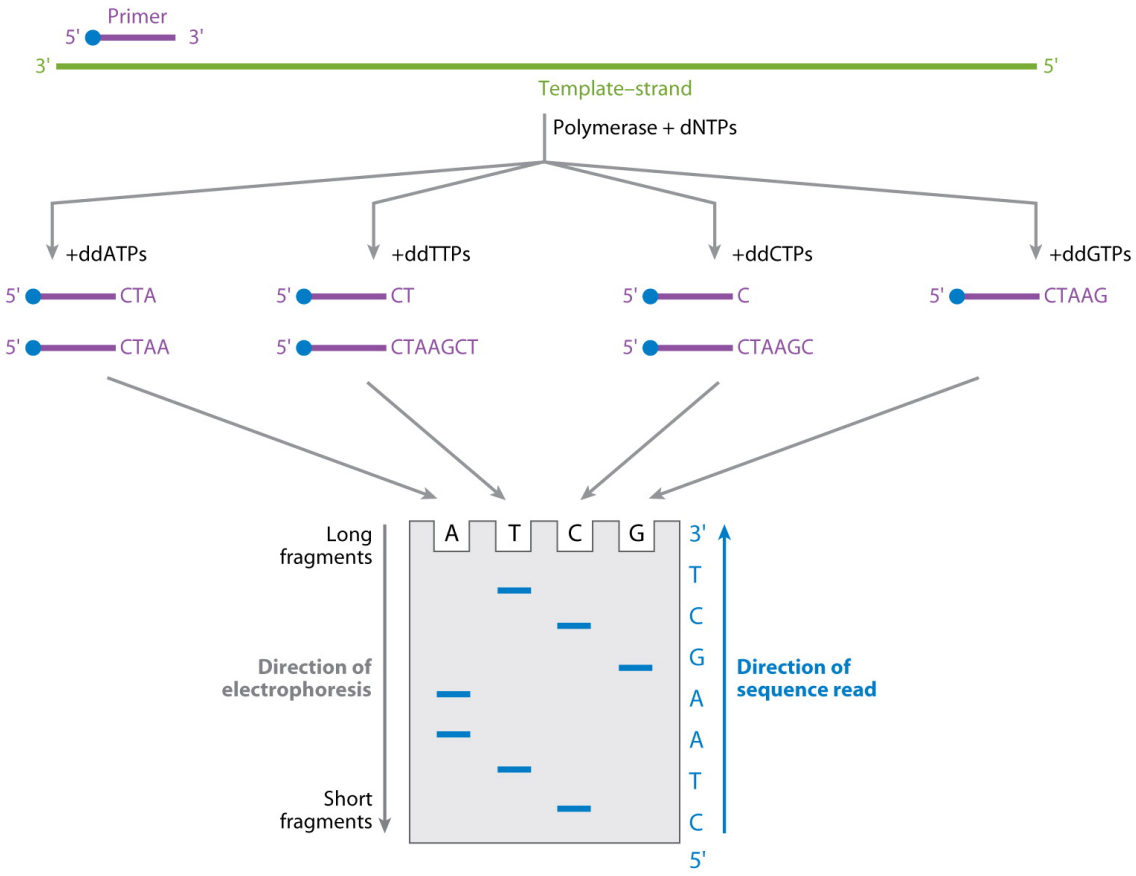


Figure 2.1: Principle of the original Sanger sequencing setup. The reaction solution is divided into four separate parts. Radio-labelled chain-terminating nucleotides of one sort per reaction are mixed with normal nucleotides. At the start of Sanger sequencing, a primer attaches to the template in question and the polymerase copies this template. DNA replication is stopped as soon as one of the chain-terminating nucleotides is built in. The differently sized sequences are then separately run in gel electrophoresis resulting in bands at the positions of the DNA where the respective nucleotides are built in. The resulting sequence can then be read out from the bands. Figure adapted from [**Mardis:2013**].

template sequence. Figure 2.2 shows a scheme of the process described here.

2.1.2 Second generation: Next generation sequencing

Next generation sequencing (NGS) was the name given to a group of new sequencing techniques developed in the mid to late 1990s. These methods are also called *high-throughput methods* as they can be parallelised to up to 6000 million reads per run, with each read of up to 650 nucleotides in length. Depending on the exact platform

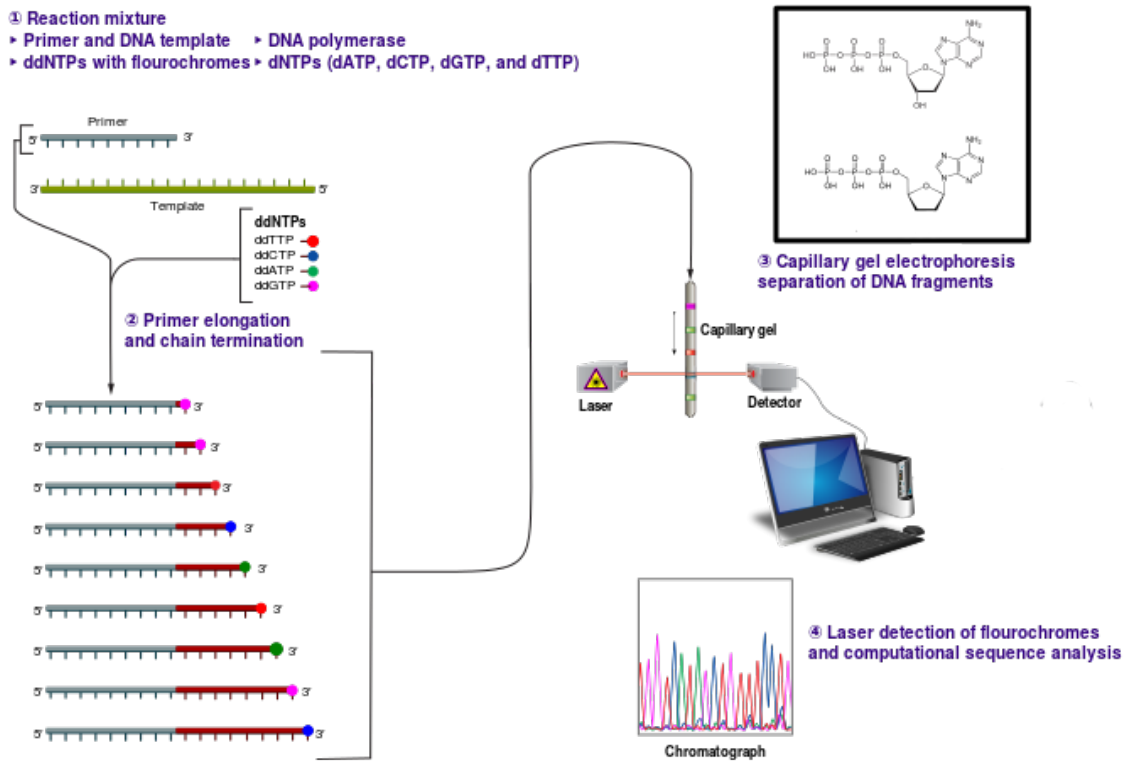


Figure 2.2: Sanger sequencing with fluorescently labelled chain-terminating nucleotides. At the start of Sanger sequencing, a primer attaches to the template in question and a PCR reaction is run with a mixture of unlabelled nucleotides and labelled inhibitors. The primer-extension reaction stops at random distances from the primer's end as the inhibitors are incorporated. This produces a set of partial template copies of different lengths, each containing a light-emitting inhibitor at the end. This set is then run on gel, with shorter copies migrating farther than the longer ones. A laser is used to excite the terminators' fluorophores. The light emitted at each position in the gel is then recorded and used to decipher the actual sequence. Figure adapted from [**SangerSequencing2015**].

and run mode selected, the speed of the sequencing can go up to 30Gbp/hour (1Gbp = 10^9bp). As with Sanger sequencing, NGS methods all have a separation, amplification, and sequencing steps, though each of these steps is done in high throughput. The high-throughput nature of NGS methods makes them much cheaper [3] (have lower per nucleotide cost) than Sanger sequencing and allows *multiplexing* of samples. In multiplexing, DNA samples coming from different individuals can be mixed during the sequencing run. To distinguish which sequence came from which individual, the DNA samples are *tagged* with a short known unique stretch of DNA before mixing.

[3] NGS sequencing is cheaper if you already own the sequencing machine. There is a tradeoff between the single-run cost and the investment to buy the instrument.

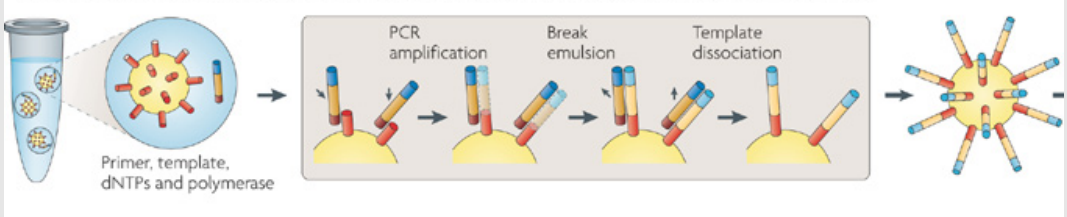
NGS employs high-throughput approaches for initial separation and amplification of the templates, and we discuss the various approaches below. By not relying on bacteria, NGS overcomes the time-consuming step of separation and amplification in Sanger sequencing. However, the NGS technologies suffer from the decrease of accuracy of the reads as the sequence gets longer. Therefore, we can only obtain reliable sequences of a few hundred base pairs. The reads are very accurate at the beginning of the sequence and the error rate grows as the read proceeds towards the end of the template.

Overall, although third generation sequencing already exists, there is still a high demand for NGS sequencers on the market as NGS a stable and easily available technology. Furthermore, post-processing methods are being developed to improve sequencing error-correction, alignment and analysis of the reads.

Four NGS platforms have been released to the market: SOLiD (formerly Applied Biosystems, now Thermo Fisher), Ion Torrent by Thermo Fisher, 454 by Roche (discontinued in 2016) and HiSeq/MySeq/... by Illumina, with Illumina being the most widespread NGS technology at the time of writing.

Box 8: Emulsion Polymerase Chain Reaction

Emulsion PCR is a PCR reaction, where each template is amplified in its own chamber. In particular, a water phase containing the DNA templates, free primers, free nucleotides, DNA polymerase enzyme, and beads (tiny sphere molecules) with primers attached to them is mixed with oil creating droplets that act as PCR reaction chambers. Then a PCR as explained in Box 6 is performed, with the single-stranded templates attaching to a primer on the bead such that the complement of the template is synthesized. The free primers ensure that the resulting single-stranded DNA extending the primer on the bead can be used for DNA synthesis, allowing for exponential increase in the number of templates. Exponential increase in the number of template molecules translates in a relatively few PCR rounds being required for each bead-attached primer to be extended by the template complement sequence. The mixture is set up such that most of the droplets contain a single bead with a single template. Since the individual chambers do not interact during the PCR, at the end of the amplification reaction each bead will be covered in many copies of the one template that paired up with it in the reaction chamber. It can happen that a single droplet contains one bead but more than one template, causing the bead to emit mixed signals during e.g. sequencing. However, since only few beads in the whole reaction contain such mixed signal, this drawback of the method is greatly overpowered by the amount of useful information which is produced. The following figure (adapted from [**Metzker2010**]) shows the principle of emulsion PCR:



SOLiD (Thermo Fisher, formerly Applied Biosystems)

SOLiD sequencing uses emulsion PCR (see Box 8) for separation and amplification of the template of interest. Once emulsion PCR is performed, each droplet should contain a single bead covered in the amplified template. The beads are then transferred to a chemically treated slide where they bind to the surface. This way physical separation of amplified templates is achieved, the step for which Sanger sequencing required bacterial cloning and colony picking.

The sequencing itself is performed by ligation, going in the opposite direction to DNA synthesis. First the primer binds to the template. The reaction uses an enzyme ligase to join the primer to 8 nucleotide long stretches of DNA (octamers). The binding octamers are complementary to the template. These stretches are fluorescently labelled based on the first two nucleotides. The last nucleotide of the stretch is modified such that no nucleotide can attach further. Schematically, this stretch can be coded as 'xynnnzzz', where x and y are the nucleotides determining the fluorescent label, 'n' are the degenerate bases that provide the indentation and 'z' are the universal bases with the last one carrying the fluorescent label and disallowing the attachment of further octamers. After the light emitted by the fluorescent markers tagging the combination of x and y is read, one step of the process is completed. The process is continued by cutting off the last three bases of the octamer (zzz) to allow the next marked octamer to attach. Once the whole template is passed through in this fashion, we have read the sequence in 5-base steps, with two bases read off and three-base unknown gaps in between.

At the completion of one such run, the newly created sequence is erased and a new primer is attached. The new primer attaches to a position in a template that is shifted forward by a single nucleotide with respect to the previous primer attachment position. The primer extension, sequence erasing and primer shifting are repeated another four times. The two-base colour-coded sequencing together with the primer shifting results in reading the sequence of the template completely while interrogating (reading) each base twice. From the combined light signals from each run, the complete sequence in question can be reconstructed. Figure 2.3 schematically shows the procedure of one primer extension run.

A big advantage of this method is that each of the bases is interrogated twice during a single run, thereby decreasing the number of potential errors. The main drawback is a low coverage in AT-rich repetitive regions, i.e. regions where a sequence rich in A and/or T nucleotide (e.g. TAT, TAAAA, TGTT, etc.) is repeated several times [**HarismendyEtAl**].

Ion Torrent (Thermo Fisher)

As in SOLiD, Ion Torrent uses emulsion PCR (Box 8) to amplify and separate the templates to be sequenced. The beads are then distributed on a plate with wells (see Figure 2.4), which can accommodate exactly one bead per well. By dropping into the wells on the plate, the beads are fixed and physically separated.

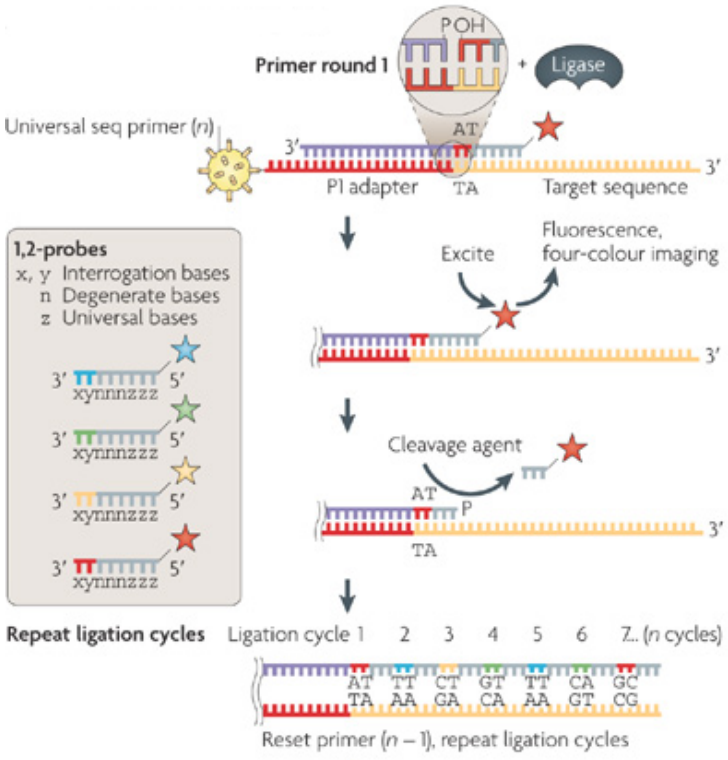


Figure 2.3: SOLiD sequencing. This figure shows the procedure for a single run-through of the sequencing in SOLiD. A nucleotide stretch of length 8 attaches to the primer, which allows the light signal from the first two marked nucleotides to be read off. Then the last three nucleotides are cut off and another 8-nucleotide stretch attaches. This way in a single run 2 bases get read and 3 get skipped. After a run is complete, the primer is shifted by one nucleotide and the procedure is repeated. Overall 5 runs are performed, which makes sure that each base is interrogated twice. Figure adapted from [**Metzker2010**].

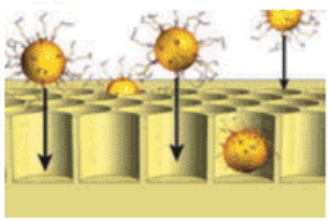


Figure 2.4: Bead are distributed on a plate with wells such that each bead ends up in one of the wells. Figure adapted from [**Rothberg2008**].

The sequencing is performed as the sequence is synthesised. The wells are sequentially flooded with unmodified A, C, G or T nucleotides (i.e. each flooding is done only with one of the four nucleotide), which are then washed away before the next nucleotide is introduced. Each time a new nucleotide is attached to the synthe-

sised sequence, a hydrogen ion (H^+) is released, which is then sensed by the semiconductor plate that is located below each of the wells. The H^+ ion is only released (and thus recorded) when the correct nucleotide binds in one of the four successive floods of different nucleotides. No light-emitting nucleotides and no optical measurements are required. Figure 2.5 shows the cross-section of the well with the bead and the Ion Torrent sequencing principle.

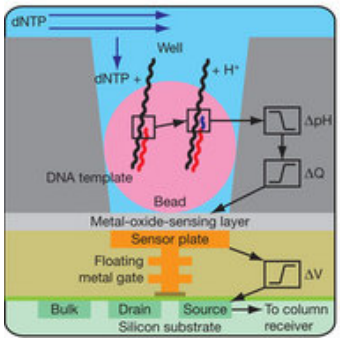


Figure 2.5: Ion Torrent sequencing. The well contains a single bead covered in template copies. During generation of the complementary sequence, H^+ ions are released as each new nucleotide is attached to the growing sequences. The ion is detected by the sensor plate beneath the bottom of the well. Figure adapted from [**Rothberg2011**].

This method may encounter problems and thus display high error rates when dealing with sequences containing nucleotide homopolymers (single nucleotide repeats), resulting in the introduction of multiple (identical) nucleotides and the consecutive release of multiple H^+ ions. Signals generated from a high repeat number are difficult to distinguish from a similar, but slightly different repeat lengths (such as 8- and 7-nucleotide repeats). The main advantages of the method are that it is quite cheap since it does not requires any fluorescent molecules, lasers, and detectors, and it is also very fast since there is no need for steps such as excitation of fluorescent molecules for the determination of the sequence.

454 (Roche, discontinued in 2016)

As in the previous methods, 454 uses emulsion PCR on beads (Box 8), where each of the beads is placed in a well on a plate (see Figure 2.4). As in Ion Torrent, the plate is washed with a single nucleotide type at a time, however, this time a light is recorded from the wells in which the particular nucleotide was incorporated. The nucleotides are then washed away and the next nucleotide type is washed over the plate. In contrast to Ion Torrent, in this technology, the double phosphate group (pyrophosphate) is detected rather than the hydrogen ions. Through a series of reactions, the pyrophosphate activates another molecule called luciferin, which emits light. This procedure, called *pyrosequencing*, is shown schematically in Figure 2.6.

This method was actually the first next generation sequencing technique to be released to the market. However, it has the same difficulty as Ion Torrent, namely

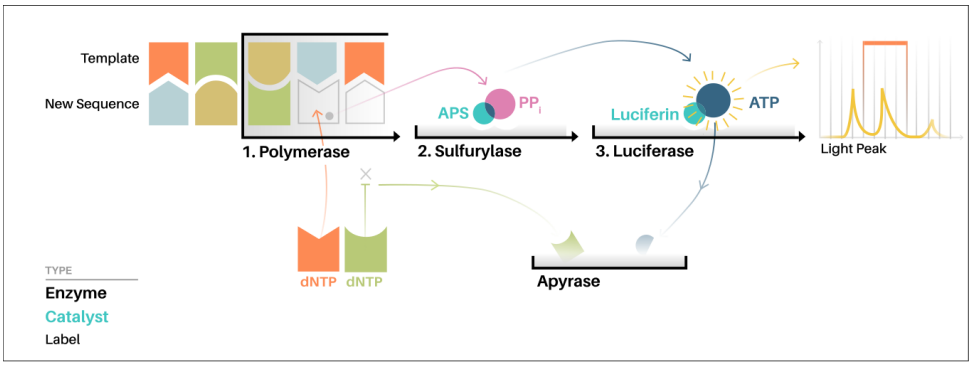


Figure 2.6: 454 sequencing. The double phosphate group gets detached as a nucleotide is attached to the generated sequence. Through a series of reactions, this activates a light-emitting molecule luciferin and the light is then recorded to indicate nucleotide incorporation. Figure adapted from [**Pyroseq2015**].

distinguishing homopolymer repeats. Furthermore, the 454 sequencer and the consumables are quite expensive. On the positive side, this method produces the longest reads among second generation sequencing methods.

Illumina sequencers

In Illumina sequencing, the templates are separated and amplified using a solid support in the form of a primer-covered slide rather than using beads. The templates are washed over the slide and attach to the primers, ideally with enough space between the attached templates. PCR is then performed to create clusters of the copies of the same template around the initial attached template. Figure 2.7 shows this process.

Sequencing is performed by synthesis using nucleotides labelled with different fluorophores corresponding to a particular nucleotide and modified to act as temporary inhibitors of synthesis[4]. As the nucleotides are distinctly labelled, we can wash a mixture of all nucleotides over the slide and then record the emitted light at each spot (i.e. cluster) of the slide.

Illumina offers a wide array of machines, all with very high throughput. Also, the company currently offers the best price per base pair. On the downside, depending on the read length and setup, some runs can be long and time consuming. As with many other methods using lasers and optical devices for signal detection, the accuracy of reads decreases towards the end of the template.

[4]The HighSeq X uses four different fluorophores, the NextSeq and MiniSeq platforms only use two. Two nucleotides are labelled with one of these fluorophores, each. The third nucleotide is labelled with both and the fourth nucleotide is not labelled at all. Thus upon incorporation, the forth nucleotide does not emit any light at all.

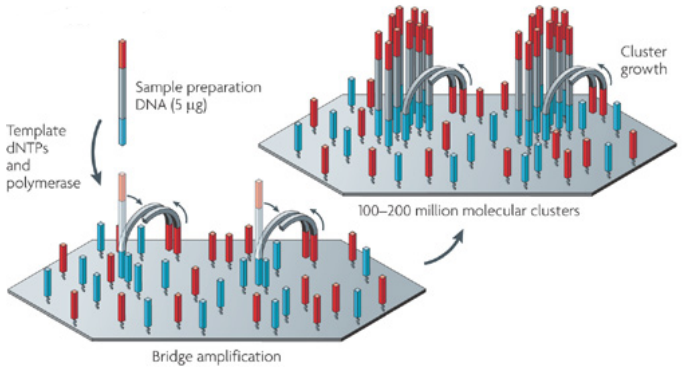


Figure 2.7: Sample separation and amplification for Illumina. Illumina sequencers use a solid support instead of beads. A primer covered slide is washed with templates, which attach and then form clusters as they are amplified. This process aims to achieve what the bacterial colonies were doing in Sanger, namely the separation of different templates. Figure adapted from [**Metzker2010**].

2.1.3 Third generation: Single-molecule sequencing

The third generation of sequencing methods is the latest wave of sequencing methodologies and allow for much longer read lengths. At each step of the NGS (second generation) sequencing processes, all copies of the template on a bead (or in a colony on a slide) receive a new nucleotide; thus, the signal comes not from one, but from many copies of the template at the same time, thereby achieving signal amplification. In contrast, third generation sequencers do not need this amplification of signal. Instead, they use technologies that allow to observe signal of a single nucleotide molecule, and are thus termed *single-molecule* sequencing methods. Thus, third generation sequencing methods have single template resolution and do not require the target template to be amplified prior to sequencing. As a consequence, the third generation methods do not require an amplification step (e.g. PCR), which is one of the main sources of errors for all of the previously mentioned methods. It is a main advantage of third-generation sequencing and a main difference compared to the second generation. Thus we can sequence e.g. a single virus without amplification, given its genome is shorter than the read length of the sequencer. This genome would be one template to be sequenced. Note however that if we want to sequence the genome of a single cell which is longer than the read length of the sequencer, the library preparation still requires an amplification step in order to have templates spanning the whole genome of a single cell being included into the library. After sequencing the templates, assembly tools lead to the full single cell genome sequence based on the individual template reads.

Third-generation sequencers employ various sequencing principles, here we will briefly sketch strand synthesis (used in PACBIO RS/PACBIO Sequel by Pacific Biosciences) and pore-based sequencing (used in MinION/PromethION/GridION by Oxford Nanopore), since these have been most widely used at the time of writ-

ing. These methods allow for long reads of about 30kbp to be produced and offer high throughput with parallelisation of up to 10 million reads per run. The disadvantage of the third generation sequencers is that they are currently still quite expensive due to the high-resolution detection systems needed to achieve single-molecule resolution, and the reads generally contain a high number of sequencing errors compared to NGS methods.

PACBIO RS (Pacific Biosciences)

No (emulsion) PCR is required for this method. Instead a plate with wells is used, with a polymerase attached at the bottom of each well. The nucleotides used for synthesis are labelled with different fluorescent markers. As the complement of the template is being synthesised, the well with the polymerase is illuminated from the bottom. The nucleotide that is being incorporated into the growing DNA strand emits light, which is immediately read by the detector. PACBIO uses a so called single-molecule real time (SMRT) sequencing technology. It is termed real-time, since both DNA synthesis and nucleotide detection occur simultaneously, enabling the observation of the incorporation of the nucleotides into the sequence in real time during synthesis. In contrast, all second generation technologies added a nucleotide but then waited for the sequencer to detect the signal, before the next addition was done, meaning synthesis was interrupted for measurement.

The PACBIO method uses proprietary technologies such as the zero-mode waveguide cells that help to guide and focus the light on the bottom of the individual wells. This allows for targeted illumination of the polymerase and the precise detection of the incorporation of a single nucleotide.

This sequencing method produces reads with many errors, as the polymerase used in PACBIO currently has a quite high error rate. However, this can be remedied by making the template circular and letting the synthesis continue for several rounds. Since we assume the polymerase makes random (and not position-dependent) errors, the errors will occur at different positions in each read. The mistakes can thus be distinguished from the true sequence by the majority rule and be excluded from the final result.

MinION/PromethION/GridION (Oxford Nanopore)

The MinION/PromethION/GridION sequencers from Oxford Nanopore all use nanopore technology to do the sequencing. Nanopores, such as pore proteins within a lipid membrane, or engineered nanopores, such as graphene, are used. An electric current passes through the porous material and as a molecule (i.e. a nucleotide) passes through the pore, the electric current changes. Oxford Nanopore claims that this technology is extremely precise, which allows it to tell exactly which particular k-mer (k being the number of nucleotides) combination of nucleotides passes through the pore at a specific time. Supposedly, the level of precision is such that it can even distinguish between methylated and non-methylated version of a particular nucleotide. The nanopore technology can also be regarded real-time as the detection

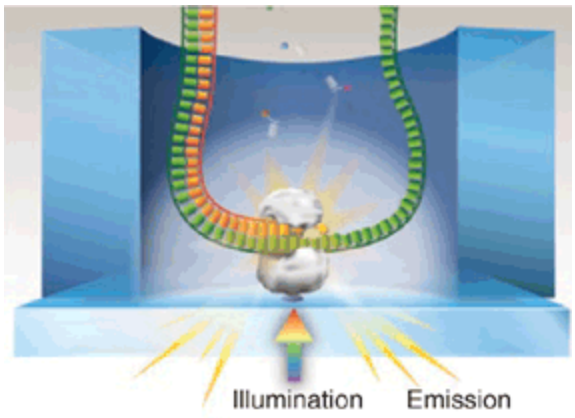


Figure 2.8: PACBIO sequencing. The reaction well with the polymerase enzyme fixed to its bottom is illuminated from underneath. The labelled nucleotide, complimentary to the template strand (the template strand depicted in green), enters the reaction site of the polymerase enzyme. The nucleotide is immobilized for a tiny fraction of time, sufficient for the dye attached to the nucleotide to be illuminated and to emit a nucleotide-specific signal. Upon the DNA synthesis reaction, the dye is cleaved off of the nucleotide and the nucleotide is attached to the growing DNA strand (orange). Figure adapted from [**Munroe2010**].

happens while the molecule travels through the pore.

In detail, there are two nanopore sequencing technologies. First, Nanopore exonuclease sequencing (Figure 2.9), in which the exonuclease enzyme linked to the pore protein cuts off a single nucleotide from the sequence at a time. The nucleotide then travels through the pore and is detected. In the second method, called strand sequencing (shown in Figure 2.10), the single stranded DNA is fed through the pore by an enzyme attached to the edge of the pore. Strand sequencing is currently the more widely used technology.

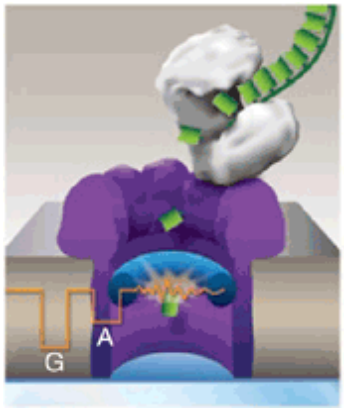


Figure 2.9: Nanopore exonuclease sequencing. The enzyme attached to the nanopore is cutting off the nucleotides from the template one by one and sends them through the pore, thereby disrupting the electric current across the membrane. Figure adapted from [**Munroe2010**].

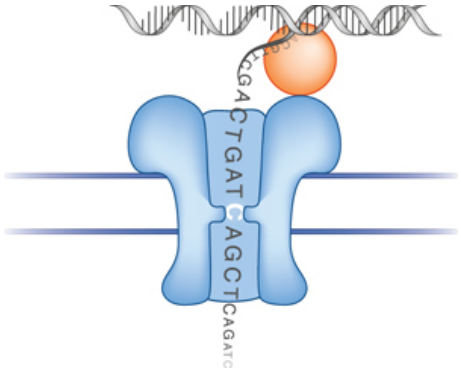


Figure 2.10: Nanopore strand sequencing. The enzyme attached to the nanopore is feeding the intact template through the pore. The identity of the nucleotides is then read by the change in the electric current occurring as a consequence of the template passing through the nanopore. Figure adapted from [**Rusk2013**].

3 Sequence alignments

After sequencing DNA from different individuals, we are interested in comparing these sequences to find similarities and differences between these individuals. But before we can do this, we need to identify the regions in the sequences that correspond to the same position in the genome for all the individual sequences. The result of such a process is called *alignment*. More precisely, alignments consist of several sequences from different individuals (on some biological unit as listed in Section 1.1) on the nucleotide or amino acid level. Sequences are called *aligned* if each character (nucleotide or amino acid) within a sequence has an assigned unique position.

In an alignment, typically different sequences are displayed in different rows, and characters with the same assigned position are displayed in the same column. Each column in the alignment is referred to be a *site* in the alignment. If a particular sequence has no character at a particular site, a hyphen ("-") is added to the alignment for that site and we call "-" a gap.

When using alignments for further analysis, we typically assume that character corresponding to the same site are orthologs (see also Section 1.2.8). Thus, when constructing an alignment, characters are *ideally* assigned to sites in the following way: Suppose we are provided with the phylogeny relating the different sequenced individuals, with its root corresponding to the most recent ancestor of all sampled individuals. Characters across sequences are assigned to the same site if they all correspond to an ancestral character in the sequence at the root of the phylogeny. The characters at a particular site may differ across biological units due to point mutations. In case of an insertion (or repeat or gene duplication) occurring for an individual, this adds a site (or many sites) to the alignment. All biological units that do not have this insertion (or repeat or gene duplication) obtain a gap for this site. In case of a deletion, the individuals with this deletion obtain a gap for the deleted character(s). In case of a recombination, each character is traced back to the root sequence via its ancestor prior to the recombination. In case of inversion of part of the sequence, the order of characters in the sequences changes when being aligned. As an example, consider characters $k, \ldots, k+m$ in sequence A which are an inversion of characters $k, \ldots, k+m$ in sequence B. In the alignment, assume the characters $k, \ldots, k+m$ of sequence B are assigned to sites $k, \ldots, k+m$. Then the characters $k, \ldots, k+m$ of the inverted sequence are assigned to sites $k+m, \ldots, k$ i.e. their order is reversed in the alignment.

In summary, in an alignment, we want to assign characters across sequences to the same site if these characters all descended from a unique character in the most recent common ancestor sequence via speciation (or the analog birth event for biological

units besides species). Thus, to build such an alignment, we need to know which positions in the sequences have changed through time and in what way: we want our alignment to represent the events that actually happened during the evolution of the sequences. However, since we do not know the true phylogeny and history of the evolution of the sequences on that phylogeny, we aim at finding an alignment that is hopefully close to the true unknown alignment.[1] To do so, we aim at finding the best alignment under a certain model or under some optimization criterion.

One distinguishes between two types of alignments: *pairwise alignments*—where only two sequences are aligned, and *multiple sequence alignments* (*MSA*)—where many sequences are aligned to each other in one go. In this chapter we will discuss exact methods for pairwise alignment as well as heuristics for pairwise alignment and MSA. Exact means that the method provides the best output under some optimality criterion. A heuristic cannot necessarily provide the optimal answer, however, it is typically much faster compared to the exact method. Heuristics for the pairwise alignment are widely used for matching one sequence against a big library through the algorithm BLAST. Heuristics for MSA are the methods of choice for obtaining an alignment for sequences of several individuals since exact methods are very slow. Besides the dot-matrix method (see below), the presented alignment tools share the property that an alignment of sequences is obtained by keeping the order of characters in each sequence, and simply adding gaps to certain positions in each sequence. In particular, inversions cannot be accounted for. While this may bias downstream results, it is still the common way of obtaining alignments.

Further, we will describe how the presented algorithms can be used to assemble the reads from a sequencing technology to obtain the genetic sequence of the corresponding individual. The obtained genetic sequences for the different individuals are then aligned as outlined in the previous paragraphs leading to an alignment of genetic sequences of different individuals from some biological unit.

Thus, after having read this chapter, you will understand the process of going from the population of individuals to the alignment as illustrated in Figure 1.7, and know how to find genetic sequences similar to a particular one you sequenced using the data mining approach BLAST.

We finish this introduction with an empirical example for a pairwise alignment.

Example: Triose-phosphate isomerase

The enzyme triose-phosphate isomerase is essential for efficient energy production in cells. Since it fulfils a very important role, it can be found in most eukaryotes. In Figure 3.1 we can see an alignment of the amino acid sequence of this protein in two very different species: rice and mosquito.

[1] There are methods capable of jointly inferring both the phylogeny (i.e. the evolutionary history) and the alignment from a set of un-aligned sequences. While this approach is statistically superior to the approaches presented in this section, it is very computationally demanding even for small data sets (e.g. [**Redelings:2005et**, **Suchard:2006iy**, **Redelings:2007eb**, **Redelings:2014bb**] and BAli-Phy [**baliphy**]).

```
NGTTDQVDKIVKILNEGQIASTDVVEVVVSPPYVFLPVVKSQLRPEIQVAAQNCW
||....!..!.|!|..|.!.:.  .||||. | .!|.:.!|||...! ||||||!
NGDKASIADLCKVLTTGPLNAD__TEVVVGCPAPYLTLARSQLPDSVCVAAQNCY
```

Figure 3.1: Alignment of partial amino acid sequences of the triose-phosphate isomerase in mosquito and rice.

(a) (b)

Figure 3.2: Schematic of a global (3.2a) and local (3.2b) alignment: aligned parts of the sequences are in dark blue, non-aligned parts of the sequences in light blue and gaps in grey.

Since this is an amino acid sequence, the alignment does not only take into account perfect matches (represented by the symbol `|`), but also positions where the two corresponding amino acids have similar chemical properties and can play similar roles in the protein (represented by the symbols `!`, `:` and `.` for strong, medium and low similarity, respectively). The two sequences are a perfect match on only 36.4% of the positions, but most positions contain amino acids that are similar, thus this is a well conserved protein.

3.1 Pairwise alignments

A pairwise alignment can be a *local alignment* or a *global alignment*, as shown in Figure 3.2. A global alignment contains both sequences aligned from start to end, whereas a local alignment only aligns sub-sequences.

Like all alignments, pairwise alignments can be between different types of sequences: Protein-protein alignments as in the triose-phosphate isomerase example, DNA-DNA, RNA-RNA and DNA- or RNA- with protein. In the last type there is no one-to-one correspondence between characters in the DNA/RNA sequence and characters in the protein sequence: one amino acid in the protein sequence corresponds to a codon (3 nucleotides) in the DNA or RNA sequence (see Section 1.2.6). This makes insertions and deletions more complicated to deal with. An additional difficulty is that multiple codons can encode for the same amino acid, a phenomenon called *codon degeneracy*.

Several strategies exist to build pairwise alignments, which we will cover in the following subsections:

- the dot-matrix method, which is a qualitative method [**Gibbs:1970jf**],
- the exhaustive method of listing all possible alignments and scoring them according to some scoring scheme (details below), and then returning the align-

ment(s) with the highest score,

- the Needleman-Wunsch algorithm for global alignments [**Needleman1970**], and its equivalent, the Smith-Waterman algorithm for local alignments [**Smith1981**], relying on dynamic programming to obtain the alignment(s) with the highest score.
-
- the algorithm BLAST as a heuristic for local alignments

Note that the exhaustive method and the Needleman-Wunsch algorithm both output the alignment(s) with the highest score. The latter is superior in speed due to dynamic programming. We nevertheless present the exhaustive method to illustrate the basic ideas behind score-based alignments.

3.1.1 Dot-matrix method

The dot-matrix method helps to visualize the similarity of two sequences [**Gibbs:1970jf**]. In this method, the two sequences are arranged in a matrix, such that the characters of one sequence are represented by the rows and the characters of the second sequence are represented by the columns of the matrix. Each position in the matrix, in which the character in the column matches the character in the row, is marked with a dot in the respective field. All other positions are left blank.

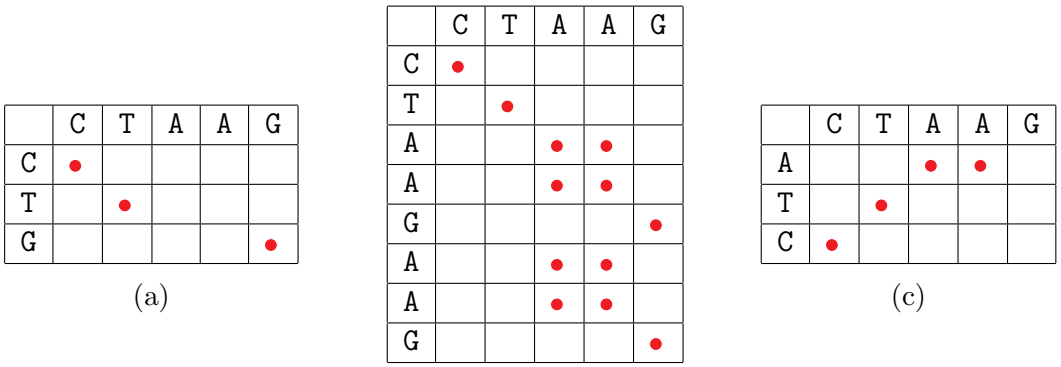


Figure 3.3: Dot-matrices illustrating important features: indels (3.3a), repeats (3.3b) and an inverted sequence (3.3c).

As shown in Figure 3.3, this method makes it easy to visually identify important features: Indels are visible as gaps in the matrix (Figure 3.3 a). Repeats are visible as repeated pattern blocks which are shifted either horizontally or vertically (in Figure 3.3 b there is a 3×3- block that is shifted vertically). Inversions are visible as reflected diagonal patterns (Figure 3.3 c). However, this method does not return an alignment, it only visually highlights areas of sequence similarity.

3.1.2 Scoring schemes

Quantitative methods rely on a scoring scheme or a stochastic model to evaluate alignments and pick the one with the highest score or likelihood. For the methods presented here, a scoring scheme is required. In Chapter 5, we will introduce stochastic models for sequence evolution, which can be extended to be used for alignment methods using statistical approaches such as maximum likelihood (the concept of maximum likelihood is explained in Box 18).

The choice of a scoring scheme will strongly affect the result, as any optimal alignment found is only optimal under the specific scheme used to evaluate it. Most scoring schemes treat all positions in the alignment as independent: the score of the entire alignment is simply the sum of the scores at each position. In the most simple scheme, a position gets one of three possible scores depending on whether there is a match, a mismatch or a gap at this position. Matches will increase the score of the alignment whereas mismatches and gaps will decrease it. Gaps, representing insertions or deletions, are biologically less likely to happen, so they generally incur a higher penalty than mismatches. In particular, the order of characters in each sequence is maintained in the alignment, and gaps are simply added between characters of a sequence. We will use this scoring scheme below, with the particular scores:

- Match score $= 3$
- Mismatch score $= -1$
- Gap score $= -2$

More complex scoring schemes can be used: one extension for instance makes the gap scores dependent on the length of the gap. The reasoning here is that opening a gap is an unlikely event, but the longer an existing gap already is, the easier it is to add another position to this gap. Another possible extension is to use substitution matrices (see Chapter 5), which imply different scores for mismatches depending on which two characters are aligned at a position.

3.1.3 Exhaustive method

The exhaustive method lists all possible alignments for two sequences, scores them, and chooses the alignment(s) with the highest score. Note that a site with a gap in each sequence will not be reported in any alignment, as such non-informative sites anyways only decrease the score. This method will always return the alignment with the highest score, but the computation time is very slow as it depends on the total number of possible alignments.

Calculation of the number of alignments for two sequences $a = a_1...a_m$ and $b = b_1...b_n$ of respective lengths m and n, $m \geq n$:

1. Assume there are $k \leq n$ gaps introduced in sequence a in the alignment ($k > n$ would lead to at least one sites at which both sequences would have a gap, which is not allowed).

2. The alignment is then of length $m + k$, which gives $\binom{m+k}{k}$ possibilities for placing the gaps in sequence a.

3. There are k' gaps introduced in sequence b in the alignment, as $m+k = n+k'$, i.e. $k' = m + k - n$.

4. There cannot be a gap in the two sequences at the same position in the alignment simultaneously. Thus, the k' gaps in sequence b need to be aligned with characters of sequence a, which gives $\binom{m}{k'} = \binom{m}{m+k-n}$ possibilities for placing those gaps.

The total number of possible alignments is: $\sum_{k=0}^{n} \binom{m+k}{k}\binom{m}{m+k-n}$.

This number grows very quickly with the length of both sequences: for short sequences of lengths $m = n = 100$, there are already 2.05×10^{75} possible alignments.

Thus, the exhaustive method is not practical for anything other than extremely short sequences.

3.1.4 Dynamic algorithms

The Needleman-Wunsch [**Needleman1970**] and Smith-Waterman [**Smith1981**] algorithms are algorithms for global and local alignments, respectively. The Needleman-Wunsch algorithm returns the same alignment(s) as the exhaustive method, i.e. the alignment(s) with the highest score. The local alignment returned by the Smith-Waterman algorithm are the aligned sub-sequences with the highest score, so the resulting local alignment cannot start or end with a gap. The local and global alignments of two sequences may be identical, but usually differ. Both algorithms score alignments faster and more efficiently compared to the exhaustive method.

To speed up the alignment process, one can utilize the fact that many possible alignments share the same start: for example, the alignments $\begin{matrix}\texttt{A-TACC}\\ \texttt{ATTG-C}\end{matrix}$ and $\begin{matrix}\texttt{A-TACC}\\ \texttt{ATT-GC}\end{matrix}$ are identical at the first three positions $\begin{matrix}\texttt{A-T}\\ \texttt{ATT}\end{matrix}$. This means that recomputing the score from scratch for each alignment would involve calculating the score of the same sub-alignment multiple times. Therefore, we can speed up the score calculations by storing the score of the sub-alignments instead of recomputing them every time.

In addition, we can even go one step further and directly determine the highest scoring sub-alignments (i.e. pairwise alignments of only sub-sequences), save their scores, and use them to obtain the highest-scoring full alignment. This technique of using a solution to a subproblem as a means to solve the full problem is called *dynamic programming*. This is a general technique which is used in many places in bioinformatics, phylogenetics and phylodynamics, as see throughout this book.

The Needleman-Wunsch and Smith-Waterman algorithms apply the concept of dynamic programming by storing the scores of best sub-alignments into a matrix that

is spanned by the two sequences. This matrix contains information in form of arrows to reconstruct the highest-scoring full alignment based on the values in the matrix. In principle the two algorithms are very similar and differ only in minor details. We will explain the general ideas by first focusing on the Needleman-Wunsch algorithm and then highlighting the differences for the Smith-Waterman algorithm.

3.1.4.1 Needleman-Wunsch algorithm

Suppose we want to align two sequences, seqA=$a_1\, a_2\, \ldots\, a_m$ and seqB=$b_1\, b_2\, \ldots\, b_n$, where a_i denotes the character at position i in sequence seqA and b_j the character at position j in sequence seqB. To determine the highest-scoring alignment, we write down a matrix, H, of dimension $(m+1) \times (n+1)$. Recall that in mathematics, the cell (i, j) in a matrix refers to the entry in the ith row and jth column. Out of convenience, we count from 0 and call the first row and column the 0th row and column, respectively. The first to mth row represent the characters of sequence seqA and the first to nth column represent the characters of seqB.

In the Needleman-Wunsch algorithm, the entry in the matrix H at position (i, j), denoted by $H(i, j)$, represents the score of the highest-scoring alignment of sequences $a_1\, a_2\, \ldots\, a_i$ and $b_1\, b_2\, \ldots\, b_j$. Note that the case $i = 0$ corresponds to the sub-sequence for seqA being an empty sequence, and $j = 0$ corresponds to the sub-sequence for seqB being an empty sequence. The initial condition is $H(0, 0) = 0$, i.e. an empty alignment has score 0.

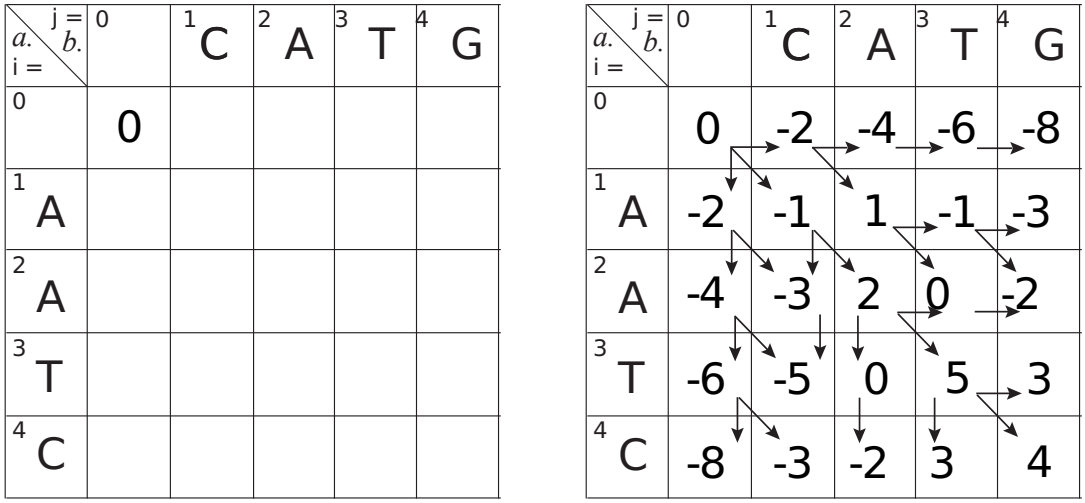
Now we iteratively fill out the matrix for all i, j, and $H(m, n)$ will provide us with the score of the highest-scoring alignment of sequences seqA and seqB. Assume we have calculated $H(k, l)$ for all $k \leq i$ and $l \leq j$ with $k + l < i + j$. Next we want to calculate $H(i, j)$. There are three cases how the alignment of sequences $a_1\, a_2\, \ldots\, a_i$ and $b_1\, b_2\, \ldots\, b_j$ may end:

1. The last site has a_i and b_j aligned. The best score of the alignment of the sequences $a_1\, a_2\, \ldots\, a_{i-1}$ and $b_1\, b_2\, \ldots\, b_{j-1}$ is $H(i-1, j-1)$; thus we need to add the score for a (mis-)match (depending on whether $a_i = b_j$ or not), $s(i, j)$, to $H(i-1, j-1)$ in order to obtain $H(i, j)$.

2. The last site has b_j aligned with a gap in seqA. The best score of the alignments of sequences $a_1\, a_2\, \ldots\, a_i$ and $b_1\, b_2\, \ldots\, b_{j-1}$ is $H(i, j-1)$; thus we need to add the gap penalty, w, to $H(i, j-1)$ in order to obtain $H(i, j)$.

3. The last site has a_i aligned with a gap in seqB. The best score of the alignments of sequences $a_1\, a_2\, \ldots\, a_{i-1}$ and $b_1\, b_2\, \ldots\, b_j$ is $H(i-1, j)$; thus we need to add the gap penalty, w, to $H(i-1, j)$ in order to obtain $H(i, j)$.

As we are looking for the highest score $H(i, j)$ at position (i, j), we need to calculate all three possibilities and decide for the sub-alignment that lead to the highest value.



(a) (b)

Figure 3.4: Needlemann-Wunsch algorithm: Example of filling out the score matrix. (a) Initial and (b) complete score matrices for the sequences seqA=`AATC` and seqB=`CATG` using the Needleman-Wunsch algorithm. We attribute a score of 3 to matches ($s(i,j) = 3$ if $a_i = b_j$), -1 for mismatches (i.e. $s(i,j) = -1$ if $a_i \neq b_j$) and penalize a gap with -2 (i.e. $w = -2$).

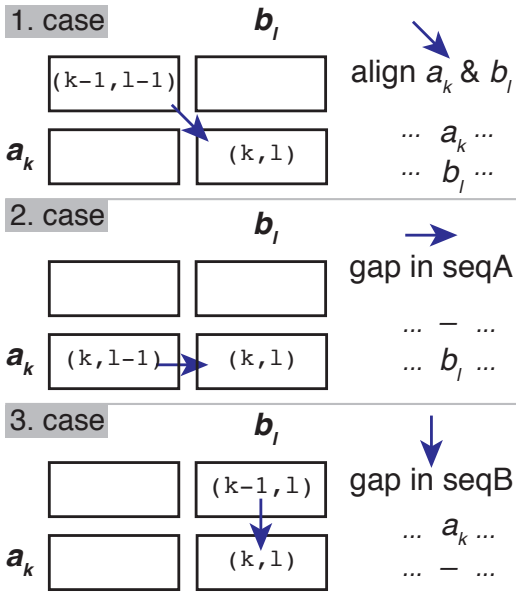
In mathematical terms, we can express these rules by:

$$H(i,j) = max \begin{cases} H(i-1,j-1) + s(i,j) & \text{(mis-)match} \quad \text{(case 1)} \\ H(i,j-1) + w & \text{gap in sequence seqA} \quad \text{(case 2)} \\ H(i-1,j) + w & \text{gap in sequence seqB} \quad \text{(case 3)} \end{cases} \tag{3.1}$$

where w is the score of a gap, and $s(i,j)$ is the score of a match if $a_i = b_j$ and the score of a mismatch otherwise.

For each value $H(i,j)$ added to the matrix, we additionally note whether the best score was achieved by case 1., 2., or 3. If the best score at position (i,j) resulted from 1., i.e. the addition of a (mis)matched character pair $a_i,\ b_j$ to the sequence represented at position $(i-1,j-1)$, a diagonal arrow from field $(i-1,j-1)$ to field (i,j) is added. If the best score was achieved by adding a gap to seqA (way 2.), we draw an arrow to the right, i.e. from field $(i,j-1)$ to (i,j). If the best score was achieved by adding a gap to seqB(way 3.), we draw a down arrow, i.e. from field $(i-1,j)$ to (i,j). Several options can give the same score, in which case all corresponding arrows are noted. An example is shown in Figure 3.4.

Once the score matrix has been filled, the score of the global alignment is found in the bottom right corner of the matrix, i.e. in position (m,n). The alignment can be reconstructed by following the arrows backwards until the top left corner of the matrix, i.e. position $(0,0)$, and reversely adding the corresponding characters to the alignment as we traverse the matrix:



(a)

a. i =	j = b.	0	1 C	2 A	3 T	4 G
0		0	-2	-4	-6	-8
1 A		-2	-1	1	-1	-3
2 A		-4	-3	2	0	-2
3 T		-6	-5	0	5	3
4 C		-8	-3	-2	3	4

(b)

Figure 3.5: Needleman-Wunsch algorithm: constructing the alignment. (a) Schematic of how to read the arrows in the score matrix. (b) Score matrix of the example in Figure 3.4 with the way through the matrix. The final global alignment is $\begin{matrix}\texttt{AATC}\\ \texttt{CATG}\end{matrix}$ and has a score of 4.

1. if the arrow pointing to position (k, l) is diagonal, then characters a_k and b_l are added to the alignment, aligned with each other;

2. if the arrow pointing to position (k, l) is a right arrow, then the character b_l is added to the alignment, aligned with a gap in sequence seqA;

3. if the arrow pointing to position (k, l) is a down arrow, then the character a_k is added to the alignment, aligned with a gap in sequence seqB.

Figure 3.5 (a) illustrates how to read the arrows and build the alignment in reverse order. In this figure, the cases correspond to the above described cases of how the intermediate alignment ends and to the cases in calculating the best score in equation 3.1. Figure 3.5 (b) demonstrates how to go backwards in the example from Figure 3.4 and obtain the global alignment. Note that if there is more than one arrow pointing towards position (k, l), each way lead to an alignment with the same (highest) score.

The number of calculations required for obtaining the best global alignment can be calculated by noting that we need the following steps: (i) For filling out each matrix entry we evaluate three products (i.e. we have three calculations) over which we take the maximum (see rule for $H(i, j)$); (ii) We have to fill out $(m + 1) \times (n + 1)$ matrix entries; (iii) We reconstruct the alignment by tracing back at most $m + n$ arrows. Thus, we have $3 \times (n + 1) \times (m + 1) + (n + m)$ calculations. Thus, for

the Needleman-Wunsch algorithm, we have a polynomial running time $\mathcal{O}(nm)$ (see Box 9). Note, for $n = m = 100$, this is of the order 10^4 steps, while the exhaustive method with an exponential running time was of the order of 10^{75} steps, illustrating the power of the dynamical programming method to make complex problems computationally tractable.

Box 9: The Landau symbol and algorithmic running times

The *Landau symbol*, or big $\mathcal{O}$ notation, is used to provide an approximation of the asymptotic behavior of an arbitrary function. The notation $\mathcal{O}$ was first used by the German mathematician Paul Bachmann and widely spread by the German mathematician Edmund Landau. Thus, this notation is also referred to Bachmann-Landau notation and goes back to the German expression "Ordnung von" (English: order of).
A function $f(x)$, where x is a real value, has the order of $g(x)$, a positive real-valued function, if there exists some constants C and x_0 such that

$$|f(x)| \leq C \times g(x) \text{ for all } x > x_0\,.$$

We denote this property by $f(x) = \mathcal{O}(g(x))$.
We use the $\mathcal{O}$-notation throughout this book for reporting asymptotic running times. A running time of the order $\mathcal{O}(g(N))$ means that there exists some constants C and N_0 such that the number of calculations required to solve the problem, f, is $f \leq C \times g(N)$ for all $N > N_0$, where N is the input data size.
We say that an algorithm has *polynomial running time*, if $g(N) = N^k$, where k is some constant (i.e. independent of N). We say that an algorithm has *exponential running time*, if $g(N) = e^{N^k}$. Since polynomials grow much slower in N than exponentials, algorithms with polynomial running time are much faster than those with exponential running time.

3.1.4.2 Smith-Waterman algorithm

The Smith-Waterman algorithm [**Smith1981**] returns the best *local* alignment of two sequences, i.e. it only aligns sub-sequences. More precisely, it finds the best alignment for sub-sequences $a_k, \ldots, a_l$ and $b_i, \ldots, b_j$, where i, j, k, l are chosen such that the score of the alignment is the maximal score. The algorithm follows a similar scheme as the Needleman-Wunsch algorithm. However, it differs in the initialization of the score matrix, the calculation of the scores, and the start/end of the reverse alignment reconstruction from the score matrix. To distinguish between the two algorithms, we now use H_{SW} and H_{NW} instead of H, where SW=Smith-Waterman and NW=Needleman-Wunsch. As above, we denote the two sequences with seqA and seqB and their length with m and n, respectively.

Initialization

The score matrix H_{SW} is of dimension $(m+1) \times (n+1)$, exactly as H_{NW}. However, the entries of row 0 and column 0 are set to 0.

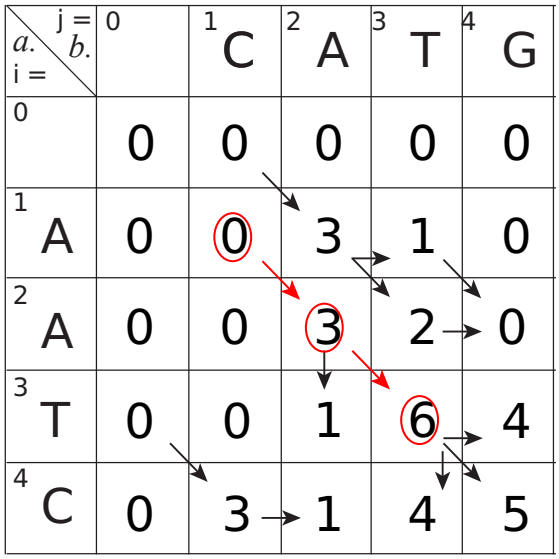


Figure 3.6: Smith-Waterman algorithm: Example of the score matrix based on the scoring scheme in equation 3.2 with alignment reconstruction for sequences seqA=AATC and seqB=CATG using the same scoring scheme as in Figure 3.4, i.e. $s(i,j) = 3$ if $a_i = b_j$, $s(i,j) = -1$ if $a_i \neq b_j$,and $w = -2$. The best local alignment is $\begin{matrix}\texttt{AT}\\ \texttt{AT}\end{matrix}$.

Score matrix

The score matrix is then successively filled based on the scoring function:

$$H_{SW}(i,j) = max \begin{cases} 0 & & \text{(stop)} \\ H_{SW}(i-1,j-1) + s(i,j) & \text{(mis-)match} & \text{(case 1)} \\ H_{SW}(i,j-1) + w & \text{gap in sequence seqA} & \text{(case 2)} \\ H_{SW}(i-1,j) + w & \text{gap in sequence seqB} & \text{(case 3)} \end{cases} \tag{3.2}$$

Building the final alignment

The alignment reconstruction starts at the position with the highest score, (l,j) (rather than the bottom right field (m,n)). This highest score is the score of the local alignment, and it ends with the aligned nucleotides a_l, b_j. The alignment reconstruction proceeds similar to the Needleman-Wunsch algorithm, and stops when a position, $(k-1, i-1)$, with a score of 0 is reached. The alignment thus starts with the aligned nucleotides a_k, b_i. Figure 3.6 shows the Smith-Waterman score matrix for the example sequences AATC and CATG (the sequences we also used in the example for demonstrating the Needleman-Wunsch algorithm). The best local alignment is $\begin{matrix}\texttt{AT}\\ \texttt{AT}\end{matrix}$ whereas the best global alignment was $\begin{matrix}\texttt{AATC}\\ \texttt{CATG}\end{matrix}$.

This procedure ensures that we find the alignment of sub-sequences with the highest score, i.e. the best local alignment. Similar to the Needleman-Wunsch algorithm, several local alignments can have the same highest score and are therefore equally good. Again, the running time is $\mathcal{O}(nm)$.

3.2 From sequencing reads to genome sequences

In the previous section we studied how to align sequences once one has them. But how do we reconstruct the sequence in the first place?

Sequencing technologies produce reads, which are incomplete snapshots of the underlying sequence of interest. By itself a read is not particularly useful, but in bulk they can be pieced together to form a correct image of the studied sequence.

There are two main avenues to reconstruct the ancestral sequence. If there is a good reference genome available, one can align against this reference. Otherwise *de novo* assembly is needed.

The choice of alignment or assembly method depends on the sequencing technology used - in particular on the length, number, and error-profile of the reads. With second and third generation sequencing techniques, the amount of data produced is so great that methods need to be optimised for speed and memory usage (typically heuristic algorithms) and there is typically a trade-off between speed and sensitivity.

3.3 Heuristic alignments: BLAST

Imagine you isolated a genetic sequence and you want to find out whether this or a similar sequence has been found before. This problem can arise in different contexts:

(i) Imagine you want to find out with which pathogen a patient is infected who suffers from symptoms that cannot be unanimously assigned to a specific disease. The only clue you have is a pathogen sequence extracted from the patient.
(ii) Or imagine that you found a gene in an organisms but you are not sure for which function this gene encodes.
(iii) In another scenario you might have sequenced the genome from a particular individual, but you do not know to which species it belongs.

These cases have in common that one obtains an unknown sequence, in the following context also referred to as *query sequence*. We want to compare the query sequence to already known sequences from a huge database – here referred to as *library*. In particular, we want to find homologs of the query sequence. Knowing characteristics such as the corresponding species or function of the homologs allows us to hypothesize regarding characteristics of the individual from which the query sequence was obtained or the characteristics of this gene. Differences of the query sequence to its homologs inform us about genotypic variation between the underlying individuals. The investigation of differences may be done either by performing GWAS-like ap-

proaches (Chapter 4) or reconstructing the evolutionary history, i.e. the phylogeny (Chapter 6).

The comparison of the obtained sequence to the sequences in the library essentially means that we calculate local pairwise alignments between our obtained sequence and each library sequence. Although the dynamic programming algorithms are much more efficient than the exhaustive approach in calculating these pairwise alignments, they are still too slow for scanning a big library of sequences.

One solution is to use the *Basic Local Alignment Search Tool* (BLAST) [**BLAST**], a heuristic word algorithm first published by Altschul in 1990 [**Altschul1990dy**]. BLAST takes advantage of the fact that two similar sequences contain local alignments of short sub-sequences with high alignment scores. Two completely different sequences do not contain local alignments of short sub-sequences with high scores and can be abandoned fairly early in the search process. Let the query sequence be of length n.

In general the algorithm is subdivided into three steps:

1. compiling a list of *words* based on the query sequence
2. scanning the database (i.e. library) for hits to these words
3. extending hits

1. step: Words of short length are determined based on the query sequence. A word is a sequence of length k. A simple option to generate such a list is to break down the query sequence into words of length k by starting at the first position and moving one letter forward for each new word. As an example let us consider the query sequence:
`NYEFILKWCL`
This sequence can be cut into the following 3-letter words:
`NYE YEF EFI FIL ILK LKW KWC WCL`
Thus, only words which occur in the original sequence are considered.

Another option – which is suggested in the original paper by Altschul et al. for amino acid sequence searches – is to consider all k-letter words that align with some part of the query sequence and have an alignment score that is bigger than a pre-defined value T. The authors used the PAM-120 matrix[2] to determine the score of the k-letter words [**Altschul1990dy**]. The default substitution matrix for the BLAST algorithm for amino acid sequence alignment in the nowadays most commonly used BLAST implementation [**BLAST**] is BLOSUM62 (BLOcks SUbstitution Matrix, shown in Figure 3.7)[3]. For the following explanation of the BLAST algorithm we

[2]The PAM (point accepted mutation) matrix describes the rate of a substitution from one amino acid to another. The PAM-1 matrix lists the substitution rates in case 1% of the sequences were different, PAM-250 lists these rates in case 20% of the amino acids were different. PAM matrices are not symmetric, i.e. a substitution from G to F for example can happen at another rate than F to G. [**Dayhoff1978**]

[3]The BLOSUM family contains symmetric amino acid substitution matrices. The alignment score of two sequences that are identical to 62% sum up to 1. These matrices were introduced in

will also use the BLOSUM62 matrix for amino acid substitutions.

Let us again consider the sequence NYEFILKWCL as an example and let us assume that we only consider words of length $k = 3$ that score at least $T = 18$ when aligned to the query sequence. The score is calculated using the BLOSUM62 matrix (Figure 3.7). For example let us calculate the score, S_{NYE}, of NYE:
NYEFILKWCL
NYE-------
This local alignment has the score $S_{NYE} = 6 + 7 + 5 = 18$. If we only include words in our list that have a score of at least $T = 18$, NYE would be added to the list. However, the substring EFI leads to a score of
NYEFILKWCL
--EFI-----
$S_{EFI} = 5 + 6 + 4 = 15$ and would not be added to the list of words, even though it is an exact substring of the query sequence. The word TWC however leads to a score of
NYEFILKWCL
------TWC-
$S_{TWC} = -1 + 11 + 9 = 19$ and would be added to the list of words even though it is not an exact substring of the query sequence. This way of generating the list of words based on similarity score is used in protein BLAST with the standard settings $k = 6$ and $T = 10$ [**BLAST**]. The authors of the original publication state that "[i]f a little care is taken in programming, the list of words can be generated in time essentially proportional to the length of the list."[**Altschul1990dy**].

When searching for a *nucleotide* sequence, exact substrings of the query word are considered (i.e. the first option of generating a list is considered). The standard settings are $k = 28$ and a match is scored +1 and a mismatch -2 [**BLAST**]. Thus, the score of the words in the word list in the case of nucleotides is equal to the word length k (and thus, all words in the list have score k). The word list contains $n - k + 1$ words.

2. step: For each entry in the library, it is checked whether a local alignment with one word of the list exceeds a pre-defined score. This score can be different from the acceptance score T in the first step. Only sequences of the library that exceed this score are further processed in step 3.

3. step: The word is successively extended to both sides by the letters from the query word and with each addition of a new letter, the score between this extended word and the word from the library is re-calculated. As soon as the score drops below a certain distance below the best score for a shorter word, the process is terminated. See Figure 3.8 for an illustration of this process. The scores and the local alignments are reported as output of the BLAST search.

Using the standard BLAST software [**BLAST**], one can search for nucleotide as well as protein sequences. The acceptance scores as well as the (mis-)match scores

[**Henikoff:1992jn**]

```
C  9
S -1  4
T -1  1  5
P -3 -1 -1  7
A  0  1  0 -1  4
G -3  0 -2 -2  0  6
N -3  1  0 -2 -2  0  6
D -3  0 -1 -1 -2 -1  1  6
E -4  0 -1 -1 -1 -2  0  2  5
Q -3  0 -1 -1 -1 -2  0  0  2  5
H -3 -1 -2 -2 -2 -2  1 -1  0  0  8
R -3 -1 -1 -2 -1 -2  0 -2  0  1  0  5
K -3  0 -1 -1 -1 -2  0 -1  1  1 -1  2  5
M -1 -1 -1 -2 -1 -3 -2 -3 -2  0 -2 -1 -1  5
I -1 -2 -1 -3 -1 -4 -3 -3 -3 -3 -3 -3 -3  1  4
L -1 -2 -1 -3 -1 -4 -3 -4 -3 -2 -3 -2 -2  2  2  4
V -1 -2  0 -2  0 -3 -3 -3 -2 -2 -3 -3 -2  1  3  1  4
F -2 -2 -2 -4 -2 -3 -3 -3 -3 -3 -1 -3 -3  0  0  0 -1  6
Y -2 -2 -2 -3 -2 -3 -2 -3 -2 -1  2 -2 -2 -1 -1 -1 -1  3  7
W -2 -3 -2 -4 -3 -2 -4 -4 -3 -2 -2 -3 -3 -1 -3 -2 -3  1  2 11
   C  S  T  P  A  G  N  D  E  Q  H  R  K  M  I  L  V  F  Y  W
```

Figure 3.7: BLOSUM62 matrix as derived in [**Henikoff:1992jn**].

		score
query sequence	NYEFILKWCL	
sequence from data base	NYEFGGTWCL	
word	TWC	$5 + 11 + 9 = 25$
expansion 1	TWCL	$5 + 11 + 9 + 4 = 29$
expansion 2	LTWCL	$-4 + 5 + 11 + 9 + 4 = 25$
expansion 3	ILTWCL	$-4 - 4 + 5 + 11 + 9 + 4 = 21$ STOP

Figure 3.8: Illustration of the word extending step of BLAST. One starts with a word from the list and expands it in accordance to the query sequence. If the word drops below a certain distance from the highest scoring the algorithm stops. In this example the algorithm stops if the score drops below 20% of the highest score ($20\% \times 29 = 5.8$, i.e. below 29-5.8=23.2 which happens in the third expansion step.

can be adapted according to the specific needs. In addition, gaps can be considered [**Altschul1997BLAST**]. The acceptance criterion in step 3 is normally also more complicated than explained in Figure 3.8 (a more detailed explanation can be found in [**Altschul1997BLAST**], but would go beyond the scope of this book). BLAST's running time depends on the library structure and can thus not be precisely estimated. However, the algorithm is extremely fast and scans huge libraries within seconds and the original BLAST algorithm sped up local alignment searches as done by FASTP by an order of magnitude [**Altschul1990dy**]. Note that despite its numerous advantages over a search where one would use the Smith-Waterman algorithm for each pair of query and library sequences, BLAST does not necessarily return the best local alignment between these two sequences [**Pertsemlidis:2001**].

3.4 Multiple sequence alignments

When more than two sequences need to be aligned, we speak of a multiple sequence alignment (MSA). This is a common situation, since MSAs form the basis of many larger comparison studies and are central to phylogenetic tree reconstruction.

MSA algorithms tend to be computationally very expensive. Several strategies exist:

- Align all sequences against a reference sequence using pairwise alignment algorithms. This strategy requires that a reference sequence is known, and that all sequences in the alignment belong to the same species or to very closely related species.
- Extend the Smith-Waterman algorithm for multiple sequences. This approach is exact but requires m^k calculation steps for k sequences of length m, and therefore is extremely slow.
- Use heuristic algorithms such as Clustal [**Clustal**] and MUSCLE [**MUSCLE**]. These algorithms are faster than the exact method but are not guaranteed to find the optimal alignment, so they require the user to check and possibly adjust the alignment.

4 Genetic associations

Once the sequences have been aligned using either local or global alignment tools, the differences between genotypes can be investigated. One direction to investigate differences is to look for *single nucleotide polymorphisms (SNPs)* and identify their phenotypic consequences.

A SNP is a variation at a single position in a DNA sequence among individuals. If more than 1% of a population does not carry the same nucleotide at a specific position in the DNA sequence, then this variation is referred to as a SNP. Connecting this expression with the early introduced terms, a gene has more than one allele if a SNP occurs within this gene. In such a case, the SNP may lead to variation in the phenotype, i.e. in the amino acid sequence. Note that SNPs can occur within coding regions as well as in noncoding regions of DNA.

From an alignment, we can easily look for SNP positions by checking each site. Note that if we know which position (i.e. site) in the alignment is a SNP and we specify which SNP position we are interested in, we can target this SNP position directly via *microarray genotyping* (these technologies have been for example reviewed in [**Distefano:2011hc**]), which allows quick screening of many individuals for specific SNPs. Sequence alignments are not necessary in this case.

A key question is whether the different alleles at a SNP position—we call them *SNP-alleles*—are linked to different phenotypes. Of highest priority is the study of the human genome. The genome of any two people is 99.9% identical [**HapMapNCBI**]. Not only do the 0.1% non-identical sites determine physical appearance but they also impact on the risk of developing genome-associated diseases such as Alzheimer's disease [**Corder:1993di**] or type II diabetes [**Altshuler:2000gh**]. The HapMap consortium was initialized in order to identify the SNPs in the human genome that might affect human health [**HapMapNCBI**]. In addition, if we know that certain SNPs are associated with a malignant trait, we can examine the parts in the DNA around these SNPs to identify the gene or genes responsible for the trait. In this chapter, we will discuss the case control setup as the simplest analysis tool for studying genetic associations with certain diseases. This analysis tool and its various extensions are commonly referred to as *genome-wide association studies (GWAS)*.

4.1 Testing for associations

4.1.1 The case control setup

To be able to identify whether a common genetic variant might be correlated with a certain common disease, the case control setup was developed. In this specific analysis, a large number of study individuals are recruited. This group is then divided into the "diseased" and "control" (healthy) groups. For each individual, the alleles for thousands of SNP positions are determined, most commonly using microarrays (rarely whole genome sequencing is performed). In what follows, it is assumed here that only two alleles occur at each SNP position, the major and the minor variant.

First, each SNP-allele is checked for its association with the disease status by calculating the *odds ratio* (OR). This is the ratio of the odds of having the disease amongst individuals with the minor variant at the SNP position over the odds of having the disease amongst individuals with the major variant at the SNP position:

$$\mathrm{OR} = \frac{\left(\frac{\text{number of } \textbf{diseased} \text{ individuals with } \textbf{minor} \text{ variant at SNP position}}{\text{number of } \textbf{healthy} \text{ individuals with } \textbf{minor} \text{ variant at SNP position}}\right)}{\left(\frac{\text{number of } \textbf{diseased} \text{ individuals with } \textbf{major} \text{ variant at SNP position}}{\text{number of } \textbf{healthy} \text{ individuals with } \textbf{major} \text{ variant at SNP position}}\right)} \tag{4.1}$$

If the odds ratio is larger than one, the minor variant is found more often in the diseased individuals than in the healthy group. Vice versa, if the ratio is smaller than one, the minor variant is present more frequently in the healthy group than in the diseased group. This gives a first hint whether a minor variant might play a role in the particular disease.

However, to make a statement about our confidence in the SNP-allele having an effect in a specific disease, we need to calculate the p-value. The p-value is defined in Box 1. In order to calculate a p-value, we need to clearly define the null hypothesis. Here, the null hypothesis is:

$\mathcal{H}_0$: *The minor variant does not have an effect on the disease. More precisely, this means that the diseased people are a random subset of the whole population and therefore independent of the allele they carry. Thus, the number of cases with the minor allele follows a hypergeometric distribution.*

4.1.2 Calculating the p-value in a GWAS

In this section, we show how to calculate the p-value using the null hypothesis $\mathcal{H}_0$ given above. We do so by using data from the first GWAS which was published in 2005. It investigated the association of genetic variants with macular degeneration, an age-related eye disease that causes loss of vision [**Klein2005**]. With the GWAS approach they could identify SNPs in complement factor H as one risk factor of developing this condition. Complement factor H is a glycoprotein which is part of

the metabolism and is involved in targeting the reaction of the immune system against pathogens.

In this study, 96 individuals suffering from age related macular degeneration (AMD) (referred to as cases) and 50 individuals not suffering from this disease (referred to as controls) were enrolled. In total, 116 204 SNP positions were tested per individual. On SNP rs380390, the common variant is a `G`. 10 cases did express a `G` on both alleles, the other cases expressed a `C`, or a mix (i.e. one `G`, one `C`). 29 controls expressed a `G` on both alleles, the other controls expressed `C`, or a mix. Could SNP rs380390 be associated with AMD? To answer this question we first calculate the odds ratio, OR, using equation 4.1:

$$\begin{aligned} OR &= \text{odds ratio} = \frac{\left(\frac{\text{number of } \textbf{diseased} \text{ individuals with } \textbf{minor} \text{ variant at SNP position}}{\text{number of } \textbf{healthy} \text{ individuals with } \textbf{minor} \text{ variant at SNP position}}\right)}{\left(\frac{\text{number of } \textbf{diseased} \text{ individuals with } \textbf{major} \text{ variant at SNP position}}{\text{number of } \textbf{healthy} \text{ individuals with } \textbf{major} \text{ variant at SNP position}}\right)} \\ &= \frac{86/21}{10/29} \\ &= 11.88 \end{aligned}$$

Thus, the odds ratio points towards an association between the minor variant and age-related macular degeneration. To calculate the p-value, we apply Pearson's χ^2-test as described in Box 12. Class 1 describes the abundance of the genetic variant: major variant being homozygous for `G`, minor variant having at least one `C`. Class 2 describes the disease status: case or control. The *contingency table* (as defined in Box 10) with the observed numbers and the row and column sums is:

Observed	case	control	row sums
minor variant	86	21	107
major variant	10	29	39
column sums	96	50	146

According to Pearson's χ^2-test, we need to calculate the expected number of cases with the minor variant based on a hypergeometric distribution, i.e. the entry of field (1,1) of the expected contingency table. We can do so, by using the fixed values of the row and column sums (Box 12):

$$E_{1,1} = 146 \times \frac{107}{146} \times \frac{96}{146} = 70.36$$

With this entry and the fixed row and column sums, we can complete the expected contingency table:

Expected	case	control	row sums
minor variant	70.36	36.64	107
major variant	25.64	13.36	39
column sums	96	50	146

We now calculate the deviance between the observed and expected numbers using Equation 4.2 (based on unrounded entries of the expected contingency table) and we obtain $S = 38.02$. As explained in Box 12, S is approximately χ^2-distributed, where the χ^2-distribution is introduced in Box 3. This allows us to calculate the p-value $= P(S \geq 38.02) = 6.99 \times 10^{-10}$. This indicates a significant association. However, since overall 116 204 SNP positions were considered, we need to correct for multiple testing in order to make a statistical statement regarding significance (see next section).

If many SNP positions are probed for their association with the disease status at the same time, the p-values of these tests are visualized to spot significant trends. One widely used method is a so-called *Manhattan plot* where the individual p-values are plotted on the y-axis and the position of the SNP in the genome (chromosome) on the x-axis. With this method, one can easily identify the chromosome and potentially the gene with the most SNPs associated with a specific disease.

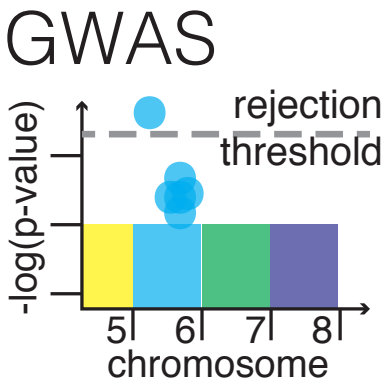


Figure 4.1: Manhattan plot of p-values resulting from a GWAS. For each SNP – ordered on the x-axis according to their position in the genome – the negative logarithm of the p-value from a test of its association with a particular disease is shown on the y-axis. The rectangular area stands for the majority of SNPs that have high p-values (and therefore have low $-\log(p\text{-values})$). The light blue dots correspond to a few SNPs with exceptionally low p-values (i.e. high negative logarithms of the p-value) of which only one SNP position lies below the rejection threshold and thus point to a possibly significant association.

Box 10: Contingency table tests

Contingency table tests are statistical tests, in which one wants to test for an association between two or more kinds of classifications with two or more characteristics each. The observations are represented in a *contingency table*. A contingency table is defined for two classes, X and Y, with two characteristics each: X_1, X_2 in class X, and Y_1, Y_2 in class Y (note that one can generalize to more than two characteristics). In total, there are n observations that can fall in any of the four categories, and the results are summarized in a contingency table:

	Y_1	Y_2	row sums
X_1	a	b	$a+b$
X_2	c	d	$c+d$
column sums	$a+c$	$b+d$	n

where $n = a + b + c + d$. The field (i, j) in the matrix describes how many observations showed X_i and Y_j.
Contingency table tests are designed to test if a characteristic within X is associated with a characteristic within Y. We will use two examples for these tests. In Chapter 4, we use *Pearson's χ^2-test* (explained in Box 12) and in Chapter 8 *Fisher's exact test* (explained in Box 11). Fisher's exact test is conceptually easier to understand and we will thus start with explaining this test.

Box 11: Fisher's exact test

Fisher's exact test

This test was developed by the British mathematician Ronald Fisher (1890-1962) in order to test the claims of a British lady who said that she was able to distinguish between two modes of preparing a British tea (which is always drunk with milk). She claimed that it was possible to taste a difference between whether one adds the tea to the cup and then milk (TIF = tea into cup first) or vice versa (MIF = milk into cup first). This example was published under the name "Mathematics of a Lady Tasting Tea" [**Fisher:1956tea**]. The two classifications in this example are how the tea was prepared (with the characteristics milk first versus tea first) and the prediction of the lady tasting the tea (with the characteristics predicted milk first and tea first). Thus, the null hypothesis is:
$\mathcal{H}_0$: *The mode of preparing the tea and the lady's predictions are independent.*
The observations of Fisher's tea example can be written in a contingency table, where X stands for the mode of preparing tea, X_1 =TIF, and X_2=MIF. Y stands for whether the lady predicts tea in cup first, Y_1, or milk first, Y_2. If the mode of preparing tea and the lady's observation was independent, this experiment is analog to the urn experiment described in Box 4. The number of red balls correspond to the cups of tea that are prepared TIF, i.e. $a + b$, and the number of black balls corresponds to MIF, i.e. $c + d$. The number of times the lady correctly predicts tea in cup first is a random variable Z. We refer to a specific realisation of this random variable with a. Then, we can specify our null hypothesis:
$\mathcal{H}_0$: *The random variable Z follows a hypergeometric distribution.*
From Equation 1.10, we write: $P(Z = a) = \frac{\binom{a+b}{a}\binom{c+d}{c}}{\binom{n}{a+c}}$. As explained in Box 1, the p-value is calculated by summing the probabilities to obtain the observed result or a more extreme result. In the tea example, a more extreme result would be obtained, if higher predictions of correct tea preparation were made, i.e. if the entry $(1, 1)$ in the table was bigger than a. Thus, p-value= $\sum_{i=a}^{a+b} \frac{\binom{a+b}{i}\binom{c+d}{a+c-i}}{\binom{n}{a+c}}$. This calculation can be done by hand for small numbers (in the tea example, only eight cups of tea were brewed, 4 with TIF and 4 with MIF). We will see an example of this procedure in Chapter 8. The summation is computationally not feasible for larger data sets. As a side note, the lady supposedly could determine all cups of tea correctly [**Salsburg:2002**].

Box 12: Pearson's χ^2-test

Another way to calculate the p-value for data in a contingency table is to use Pearson's χ^2-test [**Pearson:1900**]. This only works for reasonably large datasets. The term χ^2-test is used for all statistical tests where the distribution of interest can be approximated by a χ^2-distribution under the null hypothesis.
When doing an experiment, one fills in the contingency table entries $O_{i,j}$ according to the observations and also calculates the row and column sums as presented in Box 10. Now, we fill in the contingency table with the expected values $E_{i,j}$ under the hypergeometric distribution. The expectation for the value in field (1,1) (where we observed a) is:

$$E_{1,1} = n \times \frac{a+b}{n} \times \frac{a+c}{n}$$

Given this value, we can fill in the remaining entries of the expected table as the row and column sums are fixed.
We define the following data transformation:

$$S = \sum_{i=1}^{2} \sum_{j=1}^{2} \frac{(O_{i,j} - E_{i,j})^2}{E_{i.j}} \tag{4.2}$$

This sum describes the average deviance between the observed data and the expected data given the null hypothesis is true. One can demonstrate that S is approximately χ^2_1-distributed, i.e. has a χ^2 distribution with one degree of freedom [**Fisher:1922**, **Chernoff:1954**]. According to the definition of the p-value as the probability of obtaining the observed result, s, or a more extreme result in Box 1, we can approximate the p-value using the χ^2_1-distribution by:

$$p\text{-value} = P_{\chi^2_1}(S \geq s) \tag{4.3}$$

As explained in Box 1, we reject the null hypothesis if the p-value is less or equal to the pre-defined rejection threshold.

4.2 Correcting for multiple testing

In Box 1 we explained the usage of the p-value. If the observed outcome is too unlikely under the null hypothesis – quantified by the p-value –, the null hypothesis is rejected based on a pre-defined rejection threshold which is sometimes erroneously equated with the significance level. This concept of p-value refers to *one* statistical test. In GWAS we test hundreds of thousands of different SNP locations. If we reject with using the rejection threshold 0.05 for each of these tests, the cumulative probability of the complete study to detect a false positive might be much higher than 0.05, meaning the significance level α is > 0.05. To account for this fact, strategies for correcting for multiple testing should be used. One such strategy is the so called *Bonferroni-correction*. Assume, we test n independent SNP sites. Instead of rejecting if the p-value is < 0.05, we reject the null hypothesis if the p-value $< 0.05/n$. This rejection threshold guarantees that the significance level is $\alpha = 0.05$, i.e. that the cumulative probability of detecting a false positive is smaller than 0.05. [**vandenOord:2008ds**].

In the above described study on age-related macular degeneration, the authors used the Bonferroni-correction [**Klein2005**]. In total they looked at 116 204 SNP of which only 103 611 SNP sites were included in the data analysis, thus the null hypothesis was rejected when the $p-$value was smaller than $0.05/103\,611 = 4.8 \times 10^{-7}$.

4.3 Drawbacks and potentials

The above described case-control setup is only one way to analyse associations between genotype and phenotype. This setup has some flaws that were corrected in various extensions of this setup. We will mention some of them here.

The case-control setup makes sense in situations where one can clearly distinguish between the case and control group. However, some diseases range from expressing mild to very severe symptoms. This information on the *quantitative* trait diseases severity can be used to perform an analysis of variance (ANOVA). The null hypothesis in this case is that there is no difference between the phenotypic means of any genotype class. This allows associating risks to certain SNP-alleles [**Bush:2012fp**].

While GWAS uncovers association of SNP-alleles with phenotypes, it cannot uncover causation. This means that certain allele patterns at a SNP position may be associated with a particular disease but are not the cause. Further molecular biology experiments are therefore often needed to show if a significant SNP is indeed responsible for the disease (causation) and how it contributes to the disease status (mechanistic understanding).

By using a GWAS, we assume that variation in different SNP positions is independent of each other, meaning we assume that there is no linkage between sites. Biologically, linkage is broken up quickly between sites if there is a lot of recombination. In the next chapters, we will discuss how to deal with sequence information in case of strong linkage, and investigate associations between genotypes and phenotypes under strong linkage in Chapter 8. In Chapter 11 we will outline first advances in the field when having intermediate amount of linkage.

Many of the drawbacks of the earliest GWA studies have been addressed and the methodology has been extended further. The GWAS introduced the revolutionary idea of screening thousands of genetic variants at the same time for their association with the disease. As of July 2018, the GWAS Catalog contains 3420 publications with 62652 identified unique SNP-trait associations [**ebiGWAS**, **ethGWAS**]. The SNPedia is an attempt to collect all relevant SNPs in the human genome with the associated disease risk [**SNPedia**]. If you are interested in learning more about GWAS, please refer to [**Pearson2008**, **Bush:2012fp**].

5 Molecular evolution

In this chapter, we introduce evolutionary models describing the change of sequences through time. Evolutionary models allow us to quantify evolutionary processes acting on the sequences, e.g. to quantify rates of substitution. Further, they are an essential component of phylogenetic reconstruction methods. Evolutionary models were designed for three levels of evolution: the level of DNA or RNA sequences (i.e. genotypic level), the level of codons (i.e. the triplets of nucleotides encoding an amino acid) and the level of amino acid sequences (i.e. the phenotypic level). We will begin by defining the commonly used sequence evolution models on the DNA / RNA level, and by discussing the general properties of these models. We then extend this framework to study the evolution of sequences at the codon and the amino acid level.

The commonly used models and thus the models presented here only account for point mutations but not for insertions, deletions, inversions, or recombination. When doing downstream analysis, the rationale is to only use part of the alignment which differs due to point mutations. In practice, the used alignments often have some gaps indicating insertions/deletions, but these gaps are treated as unknown nucleotides rather than insertions or deletions, using the models below. Investigation into potential biases coming from such assumptions is ongoing.

As we discussed in the introduction, point mutations occur at the time of replication of DNA. The molecular evolution models below do not model the replication directly though. It is assumed that at any point through time, character changes happen at each site within a sequence with some rate. Thus, the probability of two characters changing at exactly the same time is 0. The rational for this model choice becomes clear when considering species. We track the sequence representing a particular species through time; each position in the sequence is associated with the character which most individuals of that species carry at that position. In this sequence, we observe a character change once the large majority (where large majority means e.g. 99 %) of individuals within the species population have acquired this change. Such a change happens if a mutation occurs in the germ line of an individual and this mutated cell gives rise to an offspring. This offspring carry the mutation in the somatic and germ cells. Eventually, by chance or selection, the mutated offspring may spread in the population until they essentially make up the large majority of the population, and we say that the mutation became *fixed*. A fixed mutation is a *substitution*. Thus, substitutions occur sequentially through time. When considering a different biological unit, e.g. one infected hosts, we observe a similar process: one pathogen entity (e.g. one bacterium or one virion) in one infected individual mutates, and

the mutation may fix within the pathogen population in this particular individual. Thus, a substitution occurred in this individual. Based on this consideration, models for sequence evolution refer to changes in the characters as *substitutions*, and the models themselves are often called *substitution models*.

We highlight that when using molecular evolution models for questions where biological units are single cells, the changes in the sequences are indeed mutations, but our models call them substitutions. We may argue that we should model this mutation process by allowing changes only at replication instead of at any time. This means that we should not use the continuous time substitution models presented here but instead some discrete time model. The rational for these substitution models still being appropriate is the following: Imagine we track one single cell through time. Upon replication, we follow one of its offspring. When this offspring replicates, we again follow one of its offspring, etc. The tracked lineage accumulates mutations through time, and if we have many replications each with few mutations, we may choose to approximate the sequence changes by a continuous time model. Once we use the alignment for further analysis, we implicitly make the assumption that we sampled few individuals from the population, ensuring we have lineages in the phylogeny with many missed replication events, and the mutations at these events occur on these lineages. If the assumption of sparse sampling is not fulfilled, there is a need to develop models with mutations only at replication.

By allowing each site to change with some rate, we assume that sites change independent from each other. It is debatable whether the assumption of independence between sites is justified, and ignoring dependence when it is present may lead to a loss in accuracy of analysis results [**Nasrallah:2010**]. There are models that define transition probabilities for the triplets of nucleotides (codon models), assuming they evolve together in a direction dictated by the properties of the corresponding amino acids ([refer here to our codon model section!]). However, only little work has been done on any dependency beyond the nucleotides within codons, as accounting for dependence is very hard (computationally): we would need to calculate the likelihood for each combination of states along a lineage for all sites simultaneously. Some work on accounting for particular site dependences is found eg. in [**Arndt:2005**, **Hoehn:2017**].

5.1 General theory on nucleotide substitution models

5.1.1 Substitution rate matrix

Each site in a sequence can express one of the four nucleotides `A`, `C`, `T`, or `G`. A site in a sequence with a particular nucleotide may change through time to a different nucleotide. The common evolutionary models assume that the change from state i into state j (with $i, j \in \{\text{T, C, A, G}\}$ and $i \neq j$) happens in an infinitesimally small time interval Δt with probability $q_{i,j}\Delta t$. $q_{i,j}$ is called *substitution rate* from i to j. Note that at any point in time, the probability of a change from i to j is the same.

"Infinitesimally" small here means that the time step is so small that only one or no event occurs with non-negligible probability, but several events (such as a change from i to k to j) have a negligible probability. When looking at the mathematical formula of multiple changes happening, one can directly see why more than one substitution is very unlikely: Let us denote the rate of any substitution with q. Two events happen with probability $(q\Delta t)^2$, three events with $(q\Delta t)^3$, and so on. In summary, the probability of more than one event is of the order of $\mathcal{O}(\Delta t^2)$. This term quickly goes to zero for small Δt. We will below derive the equations acknowledging terms of order $\mathcal{O}(\Delta t^2)$ and show how they disappear when taking the limit $\Delta t \to 0$.

The most convenient way to denote the substitution rates is in a matrix, where the rows denote the original state and the columns denote the substitution, referred to as *substitution rate matrix*:

$$\begin{array}{c} \\ T \\ C \\ A \\ G \end{array} \begin{array}{c} \begin{array}{cccc} T & C & A & G \end{array} \\ \begin{pmatrix} \cdot & a & b & c \\ d & \cdot & e & f \\ g & h & \cdot & i \\ j & k & l & \cdot \end{pmatrix} \end{array} .$$

Note that the order of the nucleotides in the nucleotide substitution rate matrix is not unambiguously defined throughout the literature. E.g. in some books you will find the nucleotides ordered alphabetically. The rates can be read off the matrix in a row-to-column way. So, if one needs the G-to-A substitution rate, one will look at the entry in the last row and the third column in our example rate matrix above, which holds the rate l.

Although the substitution rates are defined only for the off-diagonal entries, the diagonal entries are often given values so that **the sum of all entries in a given row is zero**. This is done for reasons of mathematical convenience that will be made clear below.

With this additional constraint, we can write down the complete substitution rate matrix:

$$Q = \begin{array}{c} \\ T \\ C \\ A \\ G \end{array} \begin{array}{c} \begin{array}{cccc} T & C & A & G \end{array} \\ \begin{pmatrix} -(a+b+c) & a & b & c \\ d & -(d+e+f) & e & f \\ g & h & -(g+h+i) & i \\ j & k & l & -(j+k+l) \end{pmatrix} \end{array} \tag{5.1}$$

5.1.2 Transition probability matrix

Typically we are interested in the probability of a nucleotide changing from i to j in a time interval t. This probability is, in matrix form, $Qt + \mathcal{O}(t^2)$, where the Landau symbol is added to each matrix element to take into account that the change from nucleotide i to j may happen through several changes, e.g. from i to k to j. As

explained above, if t is infinitesimally small, we may neglect the terms summarized within the Landau symbol as these terms are negligible small. For bigger t, we have to explicitly take into account these terms as we have to acknowledge that several intermediate changes may happen. In this section, we derive the probability of a nucleotide changing from i to j in any time interval t; these probabilities are summarized in the transition probability matrix.

5.1.2.1 From the rate to the exponential distribution

In order to calculate the transition probability matrix, we consider a process that generates an event E at rate α. The probability that E occurs once in an (infinitesimally) small interval of time Δt is:

$$P(E \text{ occurs once in } \Delta t) = \alpha \Delta t$$

The probability that E occurs more than once is $\mathcal{O}(\Delta t^2)$ (The probability for two events is $(\alpha \Delta t)^2$, for three events $(\alpha \Delta t)^3$, and so on). Let us denote the time until an event happens as X. We can calculate the probability of no event happening within Δt as:

$$P(X > \Delta t) = 1 - \alpha \Delta t + \mathcal{O}(\Delta t^2)$$

We now look at a bigger time interval τ and subdivide it into smaller time intervals Δt, such that $\tau = k\Delta t$, then, using the binomial theorem, we obtain

$$P(X > \tau) = (1 - \alpha \Delta t + \mathcal{O}(\Delta t^2))^k = (1 - \alpha \Delta t)^k + \mathcal{O}(\Delta t^2) = (1 - \alpha \Delta t)^{\tau / \Delta t} + \mathcal{O}(\Delta t^2) \xrightarrow[\Delta t \to 0]{} e^{-\alpha \tau}.$$

The limit for $\Delta t \to 0$ in the latter equation holds true because of the definition of the exponential function $e^x := \lim_{n \to \infty} \left(1 + \frac{x}{n}\right)^n$ (see also box 13).

Box 13: The exponential function

The exponential function can be defined in multiple ways. All of these definitions can be shown to be equivalent. In this book, we will use the following definition,

$$e^x := \sum_{n=0}^{\infty} \frac{x}{n}^n . \tag{5.2}$$

And we will now show that

$$e^x = \lim_{n\to\infty} \left(1 + \frac{x}{n}\right)^n . \tag{5.3}$$

Proof: We define $s_n := \sum_{k=0}^{n} \frac{x^k}{k!}$ and $t_n := \left(1 + \frac{x}{n}\right)^n$ for any $x \geq 0$ and an integer n. With the use of the binomial theorem, we can then develop t_n:

$$\begin{aligned} t_n &= \left(1 + \frac{x}{n}\right)^n = \sum_{k=0}^{n} \binom{n}{k} \left(\frac{x}{n}\right)^k = 1 + x + \sum_{k=2}^{n} \frac{x^k}{k!} \frac{n(n-1)\dots(n-(k-1))}{n^k} \\ &= 1 + x + \frac{x^2}{2!}\left(1 - \frac{1}{n}\right) + \dots + \frac{x^n}{n!}\left(1 - \frac{1}{n}\right)\dots\left(1 - \frac{n-1}{n}\right) \\ &\leq 1 + x + \frac{x^2}{2!} + \dots + \frac{x^n}{n!} = s_n \end{aligned}$$

If we let $n \to \infty$, we can conclude with definition 5.2 that

$$\limsup_{n\to\infty} t_n \leq \lim_{n\to\infty} s_n = e^x. \tag{5.4}$$

We need to use lim sup in equation 5.4, because we do not yet know whether the term $1 + x + \frac{x^2}{2!}\left(1 - \frac{1}{n}\right) + \dots + \frac{x^n}{n!}\left(1 - \frac{1}{n}\right)\dots\left(1 - \frac{n-1}{n}\right)$ converges.
For all $2 \leq m \leq n$ we can state that

$$1 + x + \frac{x^2}{2!}\left(1 - \frac{1}{n}\right) + \dots + \frac{x^m}{m!}\left(1 - \frac{m-1}{n}\right) \leq t_n$$

because we consider only the first $m+1$ terms of t_n on the left side of the equation. Taking the limit $n \to \infty$ on both sides of this equation, we obtain

$$s_m = 1 + x + \frac{x^2}{2!} + \dots + \frac{x^m}{m!} \leq \liminf_{n\to\infty} t_n.$$

Here, again, we need to take the lim inf because we do not know whether t_n converges. As the right side of this equation is true for all $m \leq n$, we can now take the limit $m \to \infty$ on the left side, and we obtain,

$$e^x = \lim_{m\to\infty} s_m \leq \liminf_{n\to\infty} t_n. \tag{5.5}$$

Combining equations 5.4 and 5.5, we obtain,

$$\limsup_{n\to\infty} t_n \leq e^x \leq \liminf_{n\to\infty} t_n.$$

Thus, the limit of t_n exists and is equal to e^x. This proves Equation 5.3. □
This proof is an extension of the proof for $x = 1$ in [**Rudin:1976**].

We can also calculate

$$P(0 \leq X \leq \tau) = 1 - e^{-\alpha\tau}$$

which is called the cumulative distribution function. The probability density function can then be obtained by differentiating the cumulative distribution function $f(x) = \frac{\mathrm{d}P}{\mathrm{d}t}(x) = \alpha e^{-\alpha x}$. This is the probability density function of an exponential distribution with parameter α (see Box 14). This means:

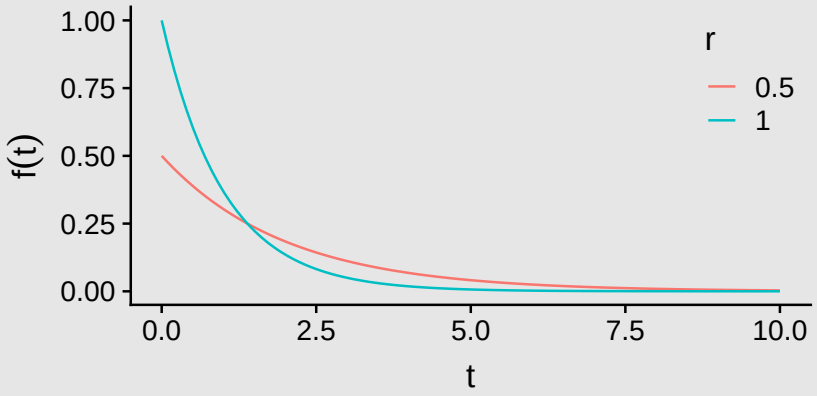
An event that occurs at rate α has exponentially distributed waiting times with parameter α.

Box 14: The exponential distribution

The exponential distribution is defined for a continuous real positive random variable T. Often this variable represents a waiting time before some event occurs. The exponential distribution takes its name from the exponential function (Box 13) appearing in its probability density function:

$$f_T(t) = e^{-rt}r.$$

As for the geometric distribution (Box 5), this can be interpreted as the product of the probability e^{-rt} that the event does not occur in the interval before time t with the probability (density) r that it occurs immediately after this interval. Its single parameter r is the *rate* of the exponential distribution. The form of the probability mass function is shown in the figure below, for rates 1 and 0.2:



The mean of this distribution is $1/r$.

The exponential distribution can be seen as the continuous analog of the geometric distribution: while the geometric distribution describes the number of discrete trials before a success with a fixed success probability p, the exponential distribution describes the amount of continuous time before an event with a fixed event rate r. Just as for the geometric distribution, the process giving rise to the exponential distribution exhibits the property of memorylessness.

Exponential distributions occur frequently in continuous time Markov processes such as birth-death processes, since the rate at which a birth or death event occurs is constant until the event occurs, meaning that the times between events are exponentially distributed.

An important property in this context is the following: if T_i are exponentially distributed random variables having rates r_i, for $i \in [1, \ldots, M]$, and we define X to be the minimum of these variables, X is itself exponentially distributed with rate $R = \sum_{i=1}^{M} r_i$.

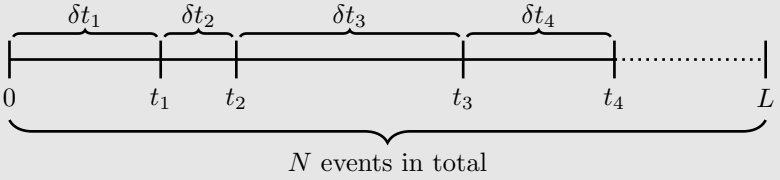
This can be seen for the two variable ($M = 2$) case by considering that

$$\begin{aligned} P(X = x) =& P(T_2 > T_1, T_1 = x) + P(T_1 > T_2, T_2 = x) \\ & P(T_2 > T_1 | T_1 = x)P(T_1 = x) + P(T_1 > T_2 | T_2 = x)P(T_2 = x) \\ =& e^{-r_2 x} e^{-r_1 x} r_1 + e^{-r_1 x} e^{-r_2 x} r_2 \\ =& e^{-(r_1 + r_2)x}(r_1 + r_2). \end{aligned}$$

The generalization to the $M > 2$ variable case is straightforward.

Box 15: The Poisson process

A Poisson process produces a sequence of events that occur at a fixed rate and yet are statistically independent of one another.



The times between successive events in a Poisson process of rate r (the δt_i intervals in the above diagram) are exponentially distributed, i.e.

$$f_{\delta t_i}(x|r) = e^{-rx}r. \tag{5.6}$$

The probability that the process will generate N events in an interval of length L (as shown in the above diagram) is given by the Poisson distribution:

$$P(N = n|rL) = e^{-rL}\frac{(rL)^n}{n!}$$

The Poisson distribution has a mean and variance both equal to the product rL.
The Poisson process is a continuous time limit of the Bernoulli process, which is simply a sequence of independent Bernoulli trials, each having a success probability p. For such a process, the probability of the next success occurring on the m^{th} trial after a previous success is given by a geometric distribution (Box 5):

$$P(m|p) = (1-p)^{m-1}p$$

Similarly, the probability that the Bernoulli process produces k successes after M total trials is given by the Binomial distribution (Box 1):

$$P(k|M,p) = \binom{M}{k}p^k(1-p)^{M-k}$$

Relating this discrete time process to the Poisson process can be done by defining r to be the average probability of success per unit time. By holding this success rate constant and taking the limit $p \to 0$ (and thus also the number of trials to infinity), the geometric distribution approaches an exponential distribution with rate r, while the Binomial distribution approaches a Poisson distribution with mean rL.
Combining the event times of two independent Poisson processes with rates r_A and r_B produces another Poisson process with rate $r_A + r_B$. That this is so can be seen by imagining superimposing the events of the two processes on a single time axis, then noting that the time interval between events is given by the minimum of two exponentially distributed random variables with rates r_A and r_B. As explained in Box (14), this random variable is also exponentially distributed with rate parameter $r_A + r_B$.
Similarly, a single Poisson process with rate r in which events are individually labeled A with probability q and B with probability $1-q$ can be regarded as the union of two independent Poisson processes with rates rq and $r(1-q)$, respectively. Decomposing Poisson processes in this fashion is known as *thinning*.

5.1.2.2 The transition probability matrix at time 0 and for small time steps

The transition probability for nucleotide i to change to j in time interval t is notated with $p_{i,j}(t)$. We summarize these probabilities in the *transition probability matrix* $P(t) = (p_{i,j}(t))_{i,j\in\{C,T,A,G\}}$.

On the way to deriving the transition probability matrix for any time step, t, we first derive the transition probability P matrix at time 0 and for infinitesimally small time steps Δt. We employ the same properties of rates and probabilities as in the previous sections, but use the matrix notation.

When no time has passed (i.e. $t = 0$), the probability for a substitution to occur is 0 and the probability to stay in the same state is 1. Thus,

$$P(0) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} =: I$$

The matrix I with 1 on the main diagonal and 0 elsewhere is called the *identity matrix*.

After an infinitesimally small time step Δt, the entry $p_{i,j}(\Delta t)$ is then

$$p_{i,j}(\Delta t) = \begin{cases} q_{i,j}\Delta t + \mathcal{O}(\Delta t^2) & \text{if } i \neq j \\ 1 - \sum_{k\neq i} q_{i,k}\Delta t + \mathcal{O}(\Delta t^2) & \text{if } i = j \end{cases}$$

That is, when j is distinct from i, $p_{i,j}(\Delta t)$ represents the probability of a state transition from i to j. In the limit of small Δt, this is simply the instantaneous transition rate $q_{i,j}$ multiplied by the time interval Δt. On the other hand, when $j = i$, $p_{i,j}(\Delta t) = p_{i,i}(\Delta t)$, which is the probability of no state change occurring in the time interval, which can be easily expressed in terms of 1 minus the probability of *any* state change occurring:

$$p_{i,i}(\Delta t) = 1 - \sum_{k\neq i} p_{i,k}(\Delta t) = 1 - \sum_{k\neq i} q_{i,k}\Delta t + \mathcal{O}(\Delta t^2).$$

We now see why it is convenient to have the rows of Q sum to 0: by defining $q_{i,i} = -\sum_{k\neq i} q_{i,k}$ we can write:

$$p_{i,i}(\Delta t) = 1 + q_{i,i}\Delta t + \mathcal{O}(\Delta t^2),$$

which in matrix notation simplifies to:

$$P(\Delta t) = I + Q\Delta t + \mathcal{O}(\Delta t^2)$$

where in the last equation, the Landau symbol is added to each matrix element. For the nucleotide substitution rate matrix defined in equation 5.1 the transition probability matrix for a small time step Δt is then:

$$P(\Delta t) = \begin{pmatrix} 1-(a+b+c)\Delta t & a\Delta t & b\Delta t & c\Delta t \\ d\Delta t & 1-(d+e+f)\Delta t & e\Delta t & f\Delta t \\ g\Delta t & h\Delta t & 1-(g+h+i)\Delta t & i\Delta t \\ j\Delta t & k\Delta t & l\Delta t & 1-(j+k+l)\Delta t \end{pmatrix} + \mathcal{O}(\Delta t^2).$$

Note that the row sum of the transition probability matrix ignoring $\mathcal{O}(\Delta t^2)$ is 1. This is quite intuitive because the probability that any of the possible events of non-negligible probability (i.e. no event or one event) happens is 1.

5.1.2.3 Transition probability matrix calculation

Now, we would like to derive the transition probability matrix P for any t, $P(t)$, i.e. for t being so large that $\mathcal{O}(\Delta t2)$ cannot be ignored. We can calculate $P_{i,j}(t+\Delta t)$ as the probability of nucleotide i changing within time interval t to nucleotide k, and nucleotide k changing in infinitesimally small time interval Δt from k to j, summed over all k. In a formula, this is:

$$P_{i,j}(t+\Delta t) = \sum_{k=1}^{4} P_{i,k}(t) P_{k,j}(\Delta t).$$

In matrix notation, this is the *Master equation* [1],

$$P(t+\Delta t) = P(t)P(\Delta t).$$

Note that the summation over k illustrates that we take into account all intermediate substitutions when calculating the transition probability from i to j.

Since $P(t)P(\Delta t) = P(t) + P(t)Q\Delta t + \mathcal{O}(\Delta t^2)$, we obtain the equation:

$$\frac{P(t+\Delta t) - P(t)}{\Delta t} = P(t)Q + \mathcal{O}(\Delta t).$$

If we take the limit $\Delta t \to 0$, we obtain the differential equation:

$$\lim_{\Delta t \to 0} \frac{P(t+\Delta t) - P(t)}{\Delta t} = \frac{\mathrm{d}P}{\mathrm{d}t}(t) = P(t)Q.$$

Now we need to solve this differential equation to state $P(t)$. By definition, $e^{Qt} := \sum_{i=0}^{\infty} \frac{(Qt)^i}{i!}$. Thus, $\frac{d}{dt}e^{Qt} = Q\sum_{i=1}^{\infty} i\frac{(Qt)^{i-1}}{i!} = Qe^{Qt}$. This means that $P(t) = e^{Qt}$ is the solution of $\frac{d}{dt}P(t) = QP(t)$ with the initial value $P(0) = I$. Thus, the substitution rate matrix Q fully defines the transition probability matrix $P(t)$.

[1] A master equation describes the time evolution of the probability of a system to occupy each one of a discrete set of states with regard to a continuous time variable t.

5.1.2.4 Evaluation of the matrix exponential

We just showed that the substitution rate matrix Q fully defines the transition probability matrix $P(t)$. However, it is not clear how the matrix exponential $P(t) = e^{Qt}$ is evaluated in practice. An important fact is that the exponential of a matrix is defined in terms of its Taylor expansion:

$$e^{Qt} = \sum_{i=0}^{\infty} \frac{(Qt)^i}{i!}$$

We could simply evaluate this sum up to a very large i. This is numerically not stable though and very slow since we require many matrix multiplications. If Q can be diagonalized (see Box 16), we can employ a *matrix diagonalization* algorithm to find U and D such that U is orthogonal, D is diagonal with the diagonal elements being the eigenvalues of Q, and,

$$Qt = UDU^{-1}.$$

Since, $(UDU^{-1})^n = UD^nU^{-1}$, substituting this into the Taylor series yields

$$\begin{aligned} e^{Qt} &= \sum_{n=0}^{\infty} \frac{UD^nU^{-1}}{n!} \\ &= U\left(\sum_{n=0}^{\infty} \frac{D^n}{n!}\right) U^{-1}. \end{aligned}$$

Furthermore, since D is diagonal, $(D^n)_{i,j} = (D_{i,j})^n$ and so

$$\begin{aligned} \left(\sum_{n=0}^{\infty} \frac{D^n}{n!}\right)_{i,j} &= \sum_{n=0}^{\infty} \frac{(D_{i,j})^n}{n!} \\ &= e^{D_{i,j}} = \left(e^D\right)_{i,j} \end{aligned}$$

The exponentiated rate matrix is then simply

$$e^{Qt} = Ue^DU^{-1} = P(t).$$

Thus, in summary, given Q is diagonalizable, we can evaluate e^{Qt} by first determining U and D (done via determining eigenvectors and eigenvalues of Q) and then taking the exponential of scalars (the diagonal elements of D) and two matrix multiplications (Ue^DU^{-1}). If Q is not diagonalizable, matrix exponentiation is very hard [**MolerEtAl1978Nineteen**, **MolerEtAl2003Nineteen**], and models with such a rate matrix Q are rarely employed.

Box 16: Diagonalizable matrices

Before we can define a diagonalizable matrix, we need to define some further terminology from linear algebra:

A quadratic matrix M of dimension $n \times n$ is called a *diagonal matrix* if it has the form:

$$M = \begin{pmatrix} m_1 & 0 & \cdots & 0 \\ 0 & m_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & m_n \end{pmatrix}$$

If all diagonal entries of a diagonal matrix are 1, the matrix is called *identity matrix* and denoted I. A quadratic matrix M of dimension $n \times n$ is called *invertible* if there exist an $n \times n$ dimensional matrix N such that

$$MN = I$$

N is also very often noted as M^{-1}.

We can now define a diagonalizable matrix: An $n \times n$ dimensional matrix D is called *diagonalizable* if there exist a diagonal matrix M and an invertible matrix N both of dimension $n \times n$ such that,

$$D = NMN^{-1}.$$

In field of linear algebra, it is shown that there are several mathematically equivalent conditions for a matrix to be diagonalizable. This means that one can check either of them when one wants to test whether a matrix is diagonalizable. (Refer to a textbook on linear algebra for further details.)

5.1.3 Markov chain model of sequence evolution

The model for sequence evolution characterized by the rate matrix Q turns out to be a *Markov chain*. Markov chains are a very common and useful stochastic process. These models have the nice property of being *memoryless*, i.e. the probability to go from one state to another only depends on the current state and not on the way to the current state. Box 17 explains the mathematical theory of Markov chains in a nutshell.

Box 17: Markov Chains

A Markov chain (or a Markov process) is a stochastic process that describes transitions ("jumps") between different states of the state space $\mathcal{S}$. A stochastic process is a series of random experiments performed through time. Time can be discrete time steps or continuous. The state space $\mathcal{S}$ of a Markov chain, i.e. the values that the random experiment can generate, is a finite or countable set. Such a stochastic process with state space $\mathcal{S}$ is a Markov Chain if the probability to jump from one state to another only depends on the current state, i.e. it is independent of the history of past states. This property is called *memorylessness* or the *Markov property.*
For the mathematically interested reader, the mathematically correct definition of a Markov chain is the following:
Given a stochastic process $(X_t)_{t\in\mathcal{T}}$ where $\mathcal{T}$ is a discrete or continuous set of times and state space $\mathcal{S}$. The process is called a Markov chain, if,

$$P(X_{t_{n+1}} = x_{t_{n+1}} | X_{t_n} = x_{t_n}, X_{t_{n-1}} = x_{t_{n-1}}, \ldots) = P(X_{t_{n+1}} = x_{t_{n+1}} | X_{t_n} = x_{t_n}) \tag{5.7}$$

for all $t_1 < t_2 < \ldots < t_n < t_{n+1}$.
This condition guarantees the memorylessness: the state at time t_{n+1} only depends on the state at time t_n, but not on state $t_1, \ldots, t_{n-1}$ meaning the process has no memory of $t_1, \ldots, t_{n-1}$. In other words, the state we are in at the moment is the only one that matters for the next step of the Markov chain.
If the probabilities on the state space do not change over time, i.e. if $P(X_{t+h} = x_1 | X_t = x_0)$ is the same for all $t > 0$, the Markov chain is called *time homogenous.*
The process is called *stationary* if $(X_{t_1}, X_{t_2}, \ldots, X_{t_n})$ and $(X_{t_1+\tau}, X_{t_2+\tau}, \ldots, X_{t_n+\tau})$ have the same distribution for all $t_1, t_2, \ldots, t_n, \tau \in \mathcal{T}$.
A Markov chain with a finite number of states in state space $\mathcal{S}$ can be uniquely defined by the transition probability matrix P, and the P matrix is directly defined by Q, the rate matrix. Consult [**Kelly:1979**, **Ross1996**] for further information on the theory of Markov chains.

We can now observe that our sequence evolution model is a time-homogeneous Markov chain:

1. The state space we use in evolutionary models is finite, e.g for the nucleotide models the state space is defined by $\mathcal{S} = \{T, C, A, G\}$.
2. The memorylessness of the process is ensured as the probability of substitution only depends on the current nucleotide i (via $q_{i,j}$), but not on the substitution history.
3. The rate matrix is identical from one time interval Δt to another; thus, we assume time-homogeneity of the process.

Since the evolutionary models assume that each position in our alignment evolves independently from the other position, each single position in our alignment is a separate Markov chain. An illustration of such a chain is shown in Figure 5.1.

Stationary distribution

The *stationary distribution* of a Markov chain is the distribution on the state space $\mathcal{S}$ which remains unchanged if the Markov chain acts on it further. In our context,

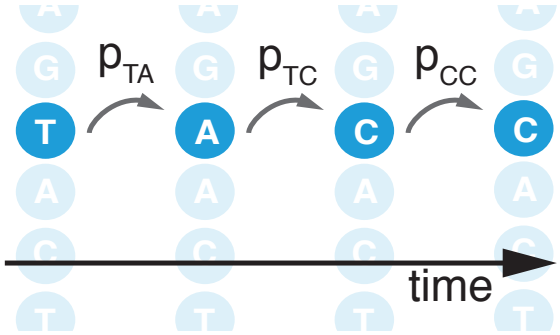


Figure 5.1: The evolution of nucleotides, codons and amino acids is modelled by a Markov chain. The vertical string of nucleotides is a sequence at a particular time. At each time point the string is in a certain state, and it transitions to another state with probabilities defined by the transition probability matrix. Here, the string at the highlighted site started in state T, and then moved to A, with probability p_{TA}. It then moved to C and stayed there for the rest of the Markov chain process.

the stationary distribution can be obtained by letting a sequence evolve for a very long time $t \to \infty$.

More formally, our substitution models are an *irreducible* and *aperiodic* Markov chain with transition probability matrix $P(t)$. Irreducible means that it is possible to go from one state to another. Mathematically this is guaranteed if for any time step $t > 0$, we have $p_{i,j}(t) > 0$ for $i \neq j$ and $i, j \in \{T, C, A, G\}$. Aperiodic means that for any time step $t > 0$, we have $p_{i,i}(t) > 0$ for $i \in \{T, C, A, G\}$. An irreducible and aperiodic Markov chain with transition probability matrix P has a unique stationary distribution. Further, $\lim_{k\to\infty} P^k$ converges to a matrix where each row is this stationary distribution. This follows from the so-called Perron-Frobenius theorem and the Perron projection (see for example Theorem 1.3 in [**Karlin:1975**]).

Biologically, the stationary distribution is interpreted as follows. Given we start with some arbitrary sequence, if we wait long enough under any substitution model with rate matrix Q, we converge to a stationary distribution of nucleotides in the evolved sequence, and this distribution is not changed further. We denote the probabilities of the four nucleotides of the stationary distribution with $\pi_T, \pi_C, \pi_A, \pi_G$. These probabilities are also called *equilibrium or stationary frequencies*, as in expectation the evolved sequence has a fraction/frequency of π_T Ts, π_C Cs, π_A As, and π_G Gs. This stationary distribution, and thus the equilibrium frequencies, is the same regardless whatever sequences the starting sequence was.

5.2 Common nucleotide substitution models

In this section, we will discuss important nucleotide substitution models and explicitly state their transition probability matrices for some, while above we only provided the toolbox to calculate the transition probability matrix via e^{Qt} for the

general model.

5.2.1 JC69 model

Substitution rate matrix under JC69

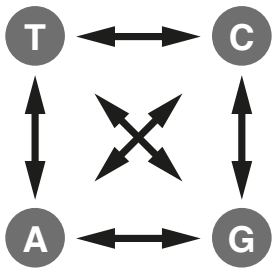


Figure 5.2: Schematic representation of the substitution rates. The width of the arrows represent the rate at which substitutions happen. We display here the JC69 model, thus all arrows have the same width.

In the Jukes-Cantor (JC69) model, shown in Figure 5.2, all the substitutions occur at the same rate λ [**Jukes1969**]. The substitution rate matrix Q_{JC69} is,

$$\mathrm{Q}_{JC69} = \begin{array}{c} \\ T \\ C \\ A \\ G \end{array} \begin{array}{c} \begin{array}{cccc} T & C & A & G \end{array} \\ \begin{pmatrix} -3\lambda & \lambda & \lambda & \lambda \\ \lambda & -3\lambda & \lambda & \lambda \\ \lambda & \lambda & -3\lambda & \lambda \\ \lambda & \lambda & \lambda & -3\lambda \end{pmatrix} \end{array}$$

Transition probability matrix under JC69

In section 5.1.2.3, we showed that we can calculate the transition probability matrix by diagonalizing the substitution rate matrix Q. We obtain $P(t) = U \,\mathrm{diag}\,(e^{\epsilon_1 t}, e^{\epsilon_2 t}, e^{\epsilon_3 t}, e^{\epsilon_4 t})\, U^{-1}$, where $\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4$ are the eigenvalues of Q_{JC69}. This expression for $P(t)$ can be further rewritten as:

$$P(t) = \begin{pmatrix} p_0(t) & p_1(t) & p_1(t) & p_1(t) \\ p_1(t) & p_0(t) & p_1(t) & p_1(t) \\ p_1(t) & p_1(t) & p_0(t) & p_1(t) \\ p_1(t) & p_1(t) & p_1(t) & p_0(t) \end{pmatrix}$$

with $p_0(t) = \frac{1}{4} + \frac{3}{4}e^{-4\lambda t}$ and $p_1(t) = \frac{1}{4} - \frac{1}{4}e^{-4\lambda t}$. We encourage the interested reader to use pen and paper to derive these formula following the above described diagonalization scheme. Note that these formulae only depend on one variable, namely λt, rather than the two variables λ and t. This makes intuitive sense: if we half the time but double the speed, i.e. the rate, we obtain the same outcome (see also chapter 5.2.4).

For example, if we consider a portion of the human genome evolving according to the JC69 model and assume the rate of substitution to be $\lambda = 2.2/3 \times 10^{-9}$ substitutions/site/year, then the probability that we start with T and end up in C after $t = 10^6$ years is $p_{TC}(10^6) = p_1(10^6) = 7.32 \times 10^{-4}$. The probability that T does not change in the same time step is $p_{TT}(10^6) = 0.9978$. Note that in this example, $\lambda t = 7.33 \times 10^{-4}$ meaning the approximation for small time steps and the exact transition probabilities are almost the same. If we are interested in $t = 10^9$ though, then $p_{TC}(10^9) = 0.237$ while $\lambda t = 0.733$, meaning the approximation is not good, and we have to use the transition probability matrix.

Stationary distribution under JC69

For JC69, the stationary distribution is $\Pi = \{\pi_T, \pi_C, \pi_A, \pi_G\} = \{0.25, 0.25, 0.25, 0.25\}$. In general, as explained above, this is derived by calculating the matrix limit $\lim_{k\to\infty} P(t)^k$ (i.e. considering infinitely many jumps), and can be directly observed for the JC69 model by recognizing that $\lim_{t\to\infty} p_0(t) = \lim_{t\to\infty} p_1(t) = 0.25$.

As an example, if we set $\lambda = 0.0015$ substitutions/site/year the change of $p_0(t)$ and $p_1(t)$ with time follows the path shown in Figure 5.3. Starting from any nucleotide, if we let the sequence evolve for a long enough time, the probability for each of the four nucleotides at that site will be 0.25 (marked with the dashed line). Note that if we set λ to a value different of 0.0015, say by a factor f different, then the Figure 5.3 looks identical up to scaling the time axis by $1/f$.

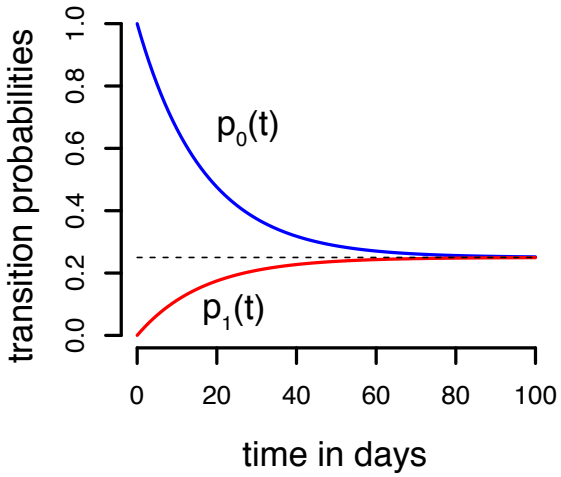


Figure 5.3: The time evolution of transition probabilities in the JC69 model. The longer the process is running for, the closer the probabilities are to 0.25. This is because the system approaches the stationary distribution after a long enough time of evolution.

Figure 5.3 shows how p_0 and p_1 change as a function of time, this specifies how $P(t)$ changes through time. In Figure 5.4 we explicitly display the matrix P for different time points. Let a sequence evolve under the JC69 model with $\lambda = 2.2/3 \times$

$10^{-9}\frac{\text{substitutions per site}}{\text{year}}$. The transition probability matrix $P(t)$ at different time steps $t = \{0, 4.5 \times 10^8, 9 \times 10^8, 1.8 \times 10^9\}$ years will look as shown in Figure 5.4.

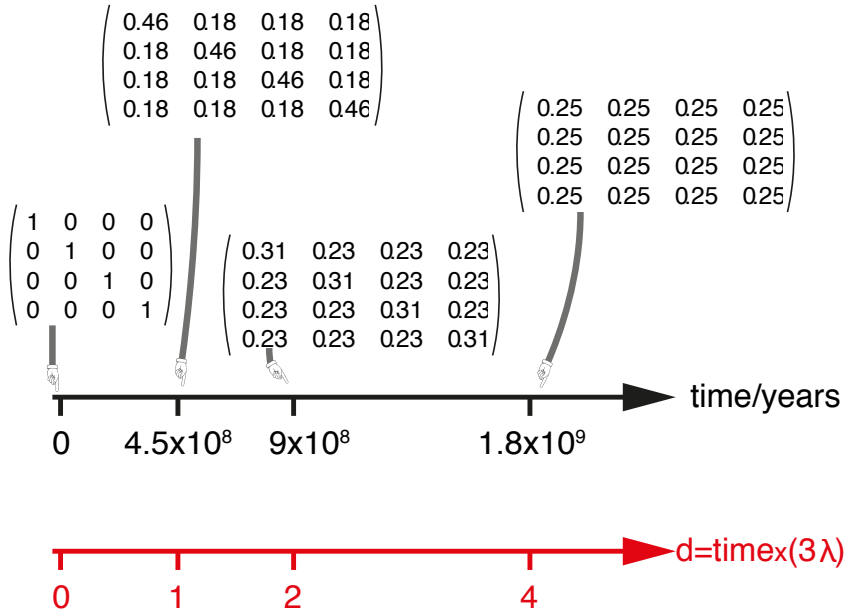


Figure 5.4: For the example substitution rate of $\lambda = 2.2/3 \times 10^{-9}\frac{\text{substitutions per site}}{\text{year}}$, the change in transition probabilities of JC69 model with increasing time is shown. At the beginning of the process $t = 0$, the sequence has not changed, i.e. all the transition probabilities away from the original state (the off-diagonal elements of the transition probability matrix) are zero. After 1.8×10^9 years all the transition probabilities are 0.25. In red, the time axis in units of substitutions is displayed, together with the corresponding transformation from units of calendar time (see Section 5.2.4).

At the last observed time point $t = 1.8 \times 10^9$ years, the chain reached the stationary phase and the final sequence is, in expectation, composed of all the nucleotides present in the frequencies given by the stationary distribution. If analyzed sequences are too divergent, such that the nucleotide content in each has reached saturation, i.e. the stationary distribution, then it is impossible to calculate the relatedness between these sequences. They would all appear completely unrelated to each other.

5.2.2 K80

Substitution rate matrix under K80

By studying substitutions in real biological samples, one found that not all substitutions occur at the same rate. Substitutions between nucleotides with similar chemical structures are more likely than between two different structures. Thymine

(T) and Cytosine (C) consist of only one pyrimidine ring-structure[2], thus they are referred to as *pyrimidines*. Adenine (A) and Guanine (G) are purine derivatives, which consists of two rings (a pyrimidine ring fused to an imidazole ring), thus they are named *purines*. Substitutions between two pyrimidines or two purines are referred to as *transitions*. Substitutions between one pyrimidine and one purine are referred to as *transversions*.

As all substitution rates in JC69 are equal, this model does not account for what was found in studies on the substitution rates. Kimura (K80) extended the basic substitution rate model by accounting for differences between transitions and transversions, i.e. the substitutions between two purines ($A \leftrightarrow G$) and between two pyrimidines ($C \leftrightarrow T$) happen more easily and more often than the transversions, i.e. the substitutions between purines and pyrimidines ($A \leftrightarrow C$, $A \leftrightarrow T$, $G \leftrightarrow C$, $G \leftrightarrow T$) [**Kimura1980**], see Figure 5.5.

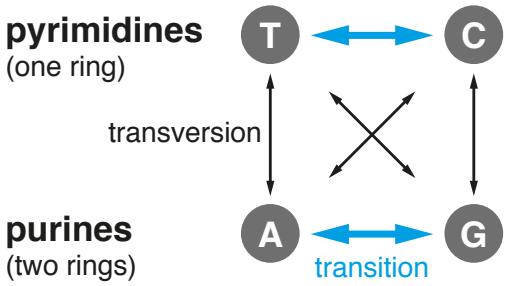


Figure 5.5: Schematic representation of the substitution rates in the K80 model. The width of the arrows represent the rate at which substitutions happen.

The substitution rate matrix Q_{K80} therefore contains two parameters, α for the transitions and β for the transversions:

$$Q_{K80} = \begin{array}{c} \\ T \\ C \\ A \\ G \end{array} \begin{array}{c} \begin{array}{cccc} T & C & A & G \end{array} \\ \begin{pmatrix} -(\alpha+2\beta) & \alpha & \beta & \beta \\ \alpha & -(\alpha+2\beta) & \beta & \beta \\ \beta & \beta & -(\alpha+2\beta) & \alpha \\ \beta & \beta & \alpha & -(\alpha+2\beta) \end{pmatrix} \end{array}.$$

Transition probability matrix under K80

Using the same diagonalization procedure as for JC69 we can calculate the transition probability matrix $P(t) = e^{Qt}$:

$$P(t) = \begin{pmatrix} p_0(t) & p_1(t) & p_2(t) & p_2(t) \\ p_1(t) & p_0(t) & p_2(t) & p_2(t) \\ p_2(t) & p_2(t) & p_0(t) & p_1(t) \\ p_2(t) & p_2(t) & p_1(t) & p_0(t) \end{pmatrix}$$

[2] A pyrimidine ring is one of the three diazines, i.e. six-membered heterocyclics with two nitrogen atoms in the ring.

where

$$\begin{aligned} p_0(t) &= \frac{1}{4} + \frac{1}{4}e^{-4\beta t} + \frac{1}{2}e^{-2(\alpha+\beta)t}, \\ p_1(t) &= \frac{1}{4} + \frac{1}{4}e^{-4\beta t} - \frac{1}{2}e^{-2(\alpha+\beta)t}, \\ p_2(t) &= \frac{1}{4} - \frac{1}{4}e^{-4\beta t}. \end{aligned}$$

As a calculation exercise, we advice the interested reader to derive these probabilities with pen and paper.

Analog to the JC69 model, these equations only depend on two instead of three (α, β, t) variables. The K80 model is thus very often parameterized in terms of the distance between two sequences separated by time t, i.e. $d = (\alpha+2\beta)t$, and the ratio between the transition and transversion rate $\kappa = \alpha/\beta$. With these definitions, the transition probabilities will transform into:

$$\begin{aligned} p_0(t) &= \frac{1}{4} + \frac{1}{4}e^{-4d/(\kappa+2)} + \frac{1}{2}e^{-2d(\kappa+1)/(\kappa+2)} \\ p_1(t) &= \frac{1}{4} + \frac{1}{4}e^{-4d/(\kappa+2)} - \frac{1}{2}e^{-2d(\kappa+1)/(\kappa+2)} \\ p_2(t) &= \frac{1}{4} - \frac{1}{4}e^{-4d/(\kappa+2)} \end{aligned}$$

As a short side note we want to make the reader aware of the order of the nucleotides appearing in the nucleotide substitution rate matrix. By using the order TCAG, transitions cluster together in the substitution rate matrix, which would not be the case for an alphabetical arrangement ACGT.

5.2.3 More general nucleotide substitution models

In the following, we provide an overview of more general models and provide their substitution rate matrices. For further properties of the models, we refer the interested reader to e.g. [**Yang2014**].

Hasegawa, Yano and Klishino extended the K80 model to account for arbitrary equilibrium frequencies of the nucleotides [**Hasegawa1984**]. Recall that equilibrium frequencies describe the expected fraction of T, C, A and G in the sequence after the stationary distribution –i.e. an evolutionary equilibrium– is reached. We denote these frequencies with π_N, $N \in \{A, C, G, T\}$. The equilibrium frequencies π_N are parameters of the model. They are either co-estimated as a parameter of the model or the empirical nucleotide frequencies extracted from the alignment are used as their estimate. The model, normally referred to as *HKY*, furthermore accounts for different transition and transversion rates and has the substitution rate matrix:

$$Q_{HKY=} \begin{array}{c} \\ T \\ C \\ A \\ G \end{array} \begin{array}{c} \begin{array}{cccc} T & C & A & G \end{array} \\ \begin{pmatrix} -(\alpha\pi_C + \beta\pi_A + \beta\pi_G) & \alpha\pi_C & \beta\pi_A & \beta\pi_G \\ \alpha\pi_T & -(\alpha\pi_T + \beta\pi_A + \beta\pi_G) & \beta\pi_A & \beta\pi_G \\ \beta\pi_T & \beta\pi_C & -(\beta\pi_T + \beta\pi_C + \alpha\pi_G) & \alpha\pi_G \\ \beta\pi_T & \beta\pi_C & \alpha\pi_A & -(\beta\pi_T + \beta\pi_C + \alpha\pi_A) \end{pmatrix} \end{array}$$

Timura and Nei [**Tamura1993**] introduced yet a more sophisticated model, called *TN93*, where the rates of transition T-to-C and C-to-T (α_1) can be different from those of A-to-G and G-to-A (α_2). The substitution rates are furthermore also dependent on the equilibrium frequencies of the nucleotides π_N, $N \in \{A, C, G, T\}$.

The substitution rate matrix under TN93 is:

$$Q_{TN93} = \begin{array}{c} \\ T \\ C \\ A \\ G \end{array} \begin{array}{c} \begin{array}{cccc} T & C & A & G \end{array} \\ \begin{pmatrix} -(\alpha_1\pi_C + \beta\pi_A + \beta\pi_G) & \alpha_1\pi_C & \beta\pi_A & \beta\pi_G \\ \alpha_1\pi_T & -(\alpha_1\pi_T + \beta\pi_A + \beta\pi_G) & \beta\pi_A & \beta\pi_G \\ \beta\pi_T & \beta\pi_C & -(\beta\pi_T + \beta\pi_C + \alpha_2\pi_) & \alpha_2\pi_G \\ \beta\pi_T & \beta\pi_C & \alpha_2\pi_A & -(\beta\pi_T + \beta\pi_C + \alpha_2\pi_A) \end{pmatrix} \end{array}$$

Note that HKY is a special case of TN93 with $\alpha_1 = \alpha_2$.

The generalised time-reversible model, *GTR*, has become very popular [**Tavare1986**, **Yang1994**, **Zharkikh1994**]:

$$Q_{GTR} = \begin{array}{c} \\ T \\ C \\ A \\ G \end{array} \begin{array}{c} \begin{array}{cccc} T & C & A & G \end{array} \\ \begin{pmatrix} -(a\pi_C + b\pi_A + c\pi_G) & a\pi_C & b\pi_A & c\pi_G \\ a\pi_T & -(a\pi_T + d\pi_A + e\pi_G) & d\pi_A & e\pi_G \\ b\pi_T & d\pi_C & -(b\pi_T + d\pi_C + f\pi_G) & f\pi_G \\ c\pi_T & e\pi_C & f\pi_A & -(c\pi_T + e\pi_C + f\pi_A) \end{pmatrix} \end{array} \tag{5.8}$$

The most general model is called *UNREST* (unrestricted model) [**Yang1994**]:

$$Q_{UNREST} = \begin{array}{c} \\ T \\ C \\ A \\ G \end{array} \begin{array}{c} \begin{array}{cccc} T & C & A & G \end{array} \\ \begin{pmatrix} -(a+b+c) & a & b & c \\ d & -(d+e+f) & e & f \\ g & h & -(g+h+i) & i \\ j & k & l & -(j+k+l) \end{pmatrix} \end{array}$$

Each substitution rate in this model can be different, and is a separate parameter of the model. All other models are special cases of this model. However, mathematical derivations for the UNREST model are very complicated and, on top of that, the model is not time reversible.

Transition probability matrices for general substitution models

There are known analytical solutions for the transition probabilities *HKY* and *TN93* substitution models, and interested readers are encouraged to refer to [**Felsenstein2004**]. There is no explicit analytical solution for the transition probabilities under the *GTR* model. However, Q_{GTR} is a symmetric matrix composed of

real values and is thus diagonalizable (see Box 16). The required probabilities can be obtained numerically very efficiently.

Obtaining the transition probabilities for the *UNREST* model is more difficult. Since this substitution model is not time-reversible, its transition rate matrix Q_{UNREST} is not in general diagonalizable.

5.2.4 Time scale: calendar time versus evolutionary time

Evolutionary processes can be measured in units of calendar time (i.e. in days, years, etc.). However one can also express time in terms of numbers of substitutions, called (expected) distance d, by a simple transformation given by $d =$ time/(expected time until any one substitution).

For the JC69 model, this would be $d = t/(3\lambda)^{-1} = 3\lambda t$, the *expected number of substitutions* in time t. The advantage of using d is that it summarizes two parameters (time and rate) into one quantity. We can estimate this quantity d, the number of substitutions which occurred when one sequence evolved into another one, by looking at the two sequences (details on that follow in the next section). To calculate the calendar time which passed while one sequence evolved into the other, we would need to know λ, which is typically unknown. The distance remains the same whether the sequences evolved at rate λ in time t or at rate 2λ in time $t/2$, etc. This means that it in many cases, it is not possible to obtain separate estimates of time and substitution rates simultaneously. Note, that the default output of many phylogenetic inference methods is in fact the distance in units of substitutions (see the red axis in Figure 5.4).

5.2.5 Time-reversibility of the nucleotide substitution models

A stochastic process $(X_t)_{t\in\mathcal{T}}$ is called *time-reversible* if it shows the same statistical behaviour forward and backward in time [**Kelly:1979**]. Intuitively, one can picture this property in the following way: Imagine you take a film of a time-reversible stochastic process through time. Then, the process will be statistically indistinguishable no matter whether you look at the film forward or backward in time.

More precisely, a stochastic process $(X_t)_{t\in\mathcal{T}}$ is called *time-reversible* if $(X_{t_1}, X_{t_2}, \ldots, X_{t_n})$ and $(X_{\tau-t_n}, X_{\tau-t_{n-1}}, \ldots, X_{\tau-t_1})$ have the same distribution for all $t_1, t_2, \ldots, t_n, \tau \in \mathcal{T}$. For a stationary Markov chain (see Box 17), this condition is equivalent to the condition,

$$\pi_i q_{i,j} = \pi_j q_{j,i}, \tag{5.9}$$

and consequently also to,

$$\pi_i p_{i,j}(t) = \pi_j p_{j,i}(t). \tag{5.10}$$

A proof of these equivalences can be found e.g. in [**Kelly:1979**]. Conditions 5.9 and 5.10 are also called *detailed balance conditions*. Intuitively, condition 5.10 can be interpreted such that the probability flux from state j to k must equal the probability flux out of state k to j.

In practice, we can even determine time reversible processes easily from the rate matrix Q.

Theorem 5.2.1. *A stationary Markov chain with rate matrix Q is time reversible if and only if the rate matrix can be decomposed into a symmetric matrix $S = (s_{i,j})_{i,j\in\{1,2,\dots,n\}}$ and the diagonal matrix Π. The equilibrium frequencies, also referred to as stationary distribution, are on the diagonals of Π, i.e.:*

$$Q = \begin{pmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,n} \\ s_{1,2} & s_{2,2} & \cdots & s_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{1,n} & s_{2,n} & \cdots & s_{n,n} \end{pmatrix} \cdot \begin{pmatrix} \pi_1 & 0 & \cdots & 0 \\ 0 & \pi_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \pi_n \end{pmatrix} \tag{5.11}$$

Proof. To proof this statement, we need to proof that equations 5.9 and 5.11 are equivalent. We do so by showing that if equation 5.9 holds, equation 5.11 can be derived and *vice versa*.
(i) Let us assume the Markov chain with rate matrix Q fulfils Equation 5.9. Equation 5.9 implies that

$$q_{i,j} \overset{(5.9)}{=} \frac{\pi_j}{\pi_i} q_{j,i} = \pi_j \underbrace{\frac{1}{\pi_i} q_{j,i}}_{=:s_{i,j}} = \pi_j s_{i,j} \tag{5.12}$$

for all $i, j \in \{1, 2, \dots, n\}$. In this equation, we defined parameters $s_{i,j}$ such that

$$s_{i,j} = \frac{1}{\pi_i} q_{j,i} \tag{5.13}$$

We can now re-write the substitution rate matrix by replacing the entries according to equation 5.12:

$$Q = \begin{pmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,n} \\ q_{2,1} & q_{2,2} & \cdots & q_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ q_{n,1} & q_{n,2} & \cdots & q_{n,n} \end{pmatrix} = \begin{pmatrix} \pi_1 s_{1,1} & \pi_2 s_{1,2} & \cdots & \pi_n s_{1,n} \\ \pi_1 s_{2,1} & \pi_2 s_{2,2} & \cdots & \pi_n s_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \pi_1 s_{n,1} & \pi_2 s_{n,2} & \cdots & \pi_n s_{n,n} \end{pmatrix}$$

And this can be decomposed into,

$$\begin{pmatrix} \pi_1 s_{1,1} & \pi_2 s_{1,2} & \cdots & \pi_n s_{1,n} \\ \pi_1 s_{2,1} & \pi_2 s_{2,2} & \cdots & \pi_n s_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \pi_1 s_{n,1} & \pi_2 s_{n,2} & \cdots & \pi_n s_{n,n} \end{pmatrix} = \begin{pmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,n} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n,1} & s_{n,2} & \cdots & s_{n,n} \end{pmatrix} \cdot \begin{pmatrix} \pi_1 & 0 & \cdots & 0 \\ 0 & \pi_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \pi_n \end{pmatrix}$$

It remains to be shown that the $(s_{i,j})$-matrix is symmetric, i.e. $s_{i,j} = s_{j,i}$. This is trivial for $i = j$, thus we assume now $i \neq j$. Then the symmetry follows out of the

definition of $s_{i,j}$ and the time-reversibility equation:

$$s_{i,j} \overset{(5.13)}{=} \frac{1}{\pi_i} q_{j,i} \overset{(5.9)}{=} \frac{1}{\pi_i}\frac{\pi_i}{\pi_j} q_{i,j} \overset{(5.13)}{=} s_{j,i}.$$

(ii) Let us assume that the Markov chain with rate matrix Q fulfils Equation 5.11. For $i = j$, Equation 5.9 is always true. Thus, we now look at the case $i \neq j$ and without loss of generality suppose $i < j$. We have $s_{i,j} = s_{j,i}$, and thus we establish Equation 5.9,

$$\pi_i q_{i,j} = \pi_i s_{j,i} \pi_j = \pi_j s_{i,j} \pi_i = \pi_j q_{j,i}$$

In summary we proved that equations 5.9 and 5.11 are equivalent. □

The most general time-reversible substitution model is the GTR model. The time-reversibility can be seen from decomposing the Q_{GTR} matrix (equation 5.8),

$$\begin{pmatrix} -(a\pi_C + b\pi_A + c\pi_G) & a\pi_C & b\pi_A & c\pi_G \\ a\pi_T & -(a\pi_T + d\pi_A + e\pi_G) & d\pi_A & e\pi_G \\ b\pi_T & d\pi_C & -(b\pi_T + d\pi_C + f\pi_G) & f\pi_G \\ c\pi_T & e\pi_C & f\pi_A & -(c\pi_T + e\pi_C + f\pi_A) \end{pmatrix}$$

$$= \begin{pmatrix} -(a+b+c) & a & b & c \\ a & -(a+e+f) & e & f \\ b & e & -(b+e+i) & i \\ c & f & i & -(c+f+i) \end{pmatrix} \cdot \begin{pmatrix} \pi_T & 0 & 0 & 0 \\ 0 & \pi_C & 0 & 0 \\ 0 & 0 & \pi_A & 0 \\ 0 & 0 & 0 & \pi_G \end{pmatrix} \quad (5.14)$$

The GTR model is the most general time-reversible substitution model. There is no other time-reversible model with nine or more parameters . JC69, K80, HKY, and TN93 are special cases of this model. For the moment, the condition of time-reversibility may seem a bit technical but we will need this property when calculating the tree likelihood, i.e. the probability to observe specific sequences given a tree and a substitution model in chapters 6.3.3.1 and 6.3.3.2.

5.2.6 Inference of phylogenies using the substitution models

We will see later how to use the transition probability matrices (i.e. e^{Qt}) directly in probabilistic approaches for phylogenetic inference based on a nucleotide sequence alignment. Probabilistic approaches estimate maximum likelihood phylogenies (Chapter 6), and Bayesian phylogenies (Chapter 10). Phenetic approaches (Chapter 6) use the estimated pairwise distances between sequences d obtained from the transition probabilities (Section 5.3). Thus, such phylogenetic reconstruction methods account for multiple substitutions at one site through time. Only the cladistic approach (Chapter 6) does not take into account multiple substitutions.

We note that when estimating the phylogenies, the parameters of the rate matrix can be estimated along with the phylogeny based on the sequence alignment. Recall from above that for a pair of sequences, we can only estimate the expected number of substitutions between them (which equals $time \times 3\lambda$ for JC69), not the actual

evolutionary time. However, when using more than two sequences for phylogenetic inference, pending on the type of data and the employed method, one can even disentangle time and substitution rate (or the overall speed of the process) in contrast to the two-sequence scenario above. Note that sequences need to be sampled through time in order to estimate the overall speed, also called clock rate .

In a phylogenetic inference, we have some sequenced individuals, while the sequences to their ancestors are typically unknown. As you will see later, unless a Bayesian method or an outgroup is used, methods return unrooted trees. In unrooted trees, it is not known on which branch the ancestor to all sequenced individuals lies. However, this does not matter as long as time-reversible methods are employed, while time-irreversible methods cannot be applied in these circumstances.

5.2.7 Overview of molecular substitution models

The overview of models mentioned in this chapter, the number of their parameters and a short description is displayed in Table 5.1.

model	parameters	description
JC69	1	all substitutions have the same rate
K80	2	accounts for transition and transversions
HKY	2+3*	distinction between transition and transversions, including equilibrium frequencies
TN93	3+3*	different rates for transitions
GTR	6+3*	general, but still time-reversible
UNREST	12	most general, not time-reversible

Table 5.1: Overview of substitution rate models and the number of the parameters. The numbers with * correspond to the number of equilibrium frequencies of nucleotides that can either be estimated from the data or co-estimated alongside the remaining parameters.

5.3 Distance estimation for nucleotide sequences

In order to reconstruct phylogenies, one approach is to calculate the distance d between all pairs of sequences, and build a phylogeny putting sequences with small distances close together (see Section 6.3.1 on phenetic approaches in the next chapter). In what follows, we will estimate d for pairs of sequences. That is, we will introduce estimators $\hat{d}$ for d. (Estimators are normally marked with ˆ.) We first explain why simply counting the number of differences between two sequences is not a good way to estimate d, and then derive an estimator for d under the JC69 model and present another estimator based on the K80 model.

5.3.1 Simple pairwise distances

The Hamming distance and p-distance are the simplest measures for calculating the distance between two sequences of equal length. The Hamming distance is simply the number of *segregating sites*, i.e. the number of sites that vary between the two sequences. The p-distance is the Hamming distance divided by the total sequence length. As an example, we look at the two sequences:

```
ATTACGAC
TCTACGAC
```

The Hamming distance between the sequences is 2, whereas the p-distance is 2/8=0.25. In the example of *triose-phosphate isomerase* from Chapter 3, Figure 3.1, the Hamming distance between mosquito and rice sequences is 35, whereas the p-distance is 0.636.

Both the Hamming distance and the p-distance are very simplistic measures which ignore that the sequences in question have evolved from a shared common ancestor and that through their evolution, some substitutions are unobserved (or hidden) when considering the observed sequences. We will go through different scenarios of successive substitutions to illustrate the cases where the Hamming and p-distance measures are biologically inadequate. We will use the tree shown in Figure 5.6 as an example. For the purpose of demonstration we assume that we know the "true" sequence at the internal node depicting the common ancestor of taxon 1 and taxon 2 [3].

A difference of the nucleotide at the 2nd position between the daughter sequences taxon 1 and taxon 2 could have resulted from a single substitution, as depicted in Figure 5.6 (red). In this case a C changed to a G on the branch leading to taxon 2. The Hamming and p-distance would correctly account for one substitution.

However, there could have also been several substitutions which are unobserved and the Hamming or p-distance cannot account for. We highlight now four such scenarios.

First, we may have *multiple substitutions* at a single position, on a branch between the common ancestor and one of the taxa. For instance, the T in position 4 of the common ancestor sequence could have been first replaced by a C and only later by an A, but all we can observe is the A in the final sequence. An example of this is shown in Figure 5.6 (blue). When our data contains only sequences for taxa 1, 2 and 3, it is impossible to distinguish between multiple substitutions and a single substitution in the evolutionary history leading to the taxa, and the Hamming or p-distance would assume there was only one substitution.

Second, we could encounter *parallel substitutions*, which occur if both descendant sequences changed from the same nucleotide into a different nucleotide and this

[3] A *taxon* (plural taxa) is a term used in the science of biological classification (which is referred to as *taxonomy*). It describes a biological unit that is defined in taxonomy and can refer to a group of one or more populations of an organism or organisms. Taxa are typically arranged in a hierarchy from kingdom to subspecies.

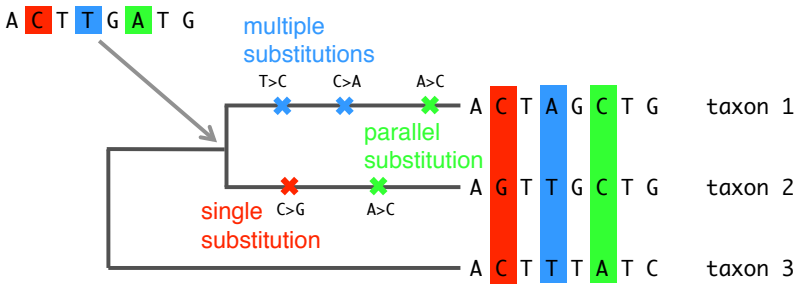


Figure 5.6: An example of a phylogenetic tree with three taxa. The arrow indicates the most recent common ancestor of taxa 1 and 2. **Red:** A single substitution happened at site 2 of the alignment on the branch leading to taxon 2. **Blue:** Two consecutive substitutions happened at site 4 of the alignment on the branch leading to taxon 1. **Green:** At site 6, both taxa 1 and 2 had each one independent substitution from adenine to cytosine.

change is the same for both lineages. In the example shown in Figure 5.6 (green), the A in position 6 changed into C in parallel for taxa 1 and 2. In real situations, we typically do not know what the common ancestor sequence was, so if we used a simple Hamming or a p-distance measure between the observed descendant sequences, we would assume there has been no character change where in fact parallel substitution could have occurred.

If the nucleotides at a certain position were different in two ancestral lineages but were substituted into the same nucleotide in the resulting two present lineages, this is called *convergent substitution*. Again, the Hamming or p-distance may underestimate the number of substitutions.

Lastly, *back substitutions* are yet another type of substitutions that cannot be observed in the collected sequences. This type of substitution occurs if a nucleotide changes and then mutates back to the original starting nucleotide. If we do not manage to obtain the sequences from the descendants before the first substitution is reverted, we will never know there was any substitution and the Hamming or p-distance will assume no substitution.

Thus, the Hamming and p-distance measure a minimal distance between sequences, since they report the minimal changes required to go from evolve from one sequence into the other. Using such minimal distances may bias downstream results, and we prefer a distance measure that represents the actual evolutionary distance acknowledging potential hidden or unobserved substitutions (such as the four cases above) given the observed sequences at present. Using the mathematical models of sequence evolution introduced above provides such distance measures.

5.3.2 Pairwise distances using a method of moments approach for JC69

Let us consider two sequences of length n each. One sequence was the starting sequence, having evolved for time t under the JC69 model, into the second sequence. We want to estimate their evolutionary distance, taking into account all possible hidden intermediate substitutions as shown in Figure 5.6. We start with defining the probability of any substitution over time t. From the transition probability matrix, one extracts the total probability that a nucleotide changes (i.e. any substitution happens):

$$p(t) = 3p_1(t) = \frac{3}{4} - \frac{3}{4}e^{-4\lambda t} \tag{5.15}$$

As explained in the previous section, we can re-write the time, t, in units of numbers of substitutions, d, as $t = \frac{d}{3\lambda}$. By plugging this expression into equation 5.15, we obtain

$$p(t) = \frac{3}{4} - \frac{3}{4}e^{-\frac{4}{3}d}. \tag{5.16}$$

We note here that $0 \leq p(t) \leq \frac{3}{4}$, since $0 \leq d \leq \infty$.

Now, we re-arrange the Equation 5.16 and obtain,

$$d = -\frac{3}{4}\log\left(1 - \frac{4}{3}p(t)\right). \tag{5.17}$$

We aim to obtain an estimator for d, we call it $\hat{d}$ (we denote an estimator of a parameter with $\hat{}$). Instead of estimating d directly, we obtain an estimate for $p(t)$, $\hat{p}$, based on the method of moments. We use the two sequences as data for obtaining $\hat{p}$. We count the number of segregating sites in the two sequences, x. The probability $p(t)$ is the probability for any site being segregating, meaning each site follows the Bernoulli distribution with parameter $p(t)$, and thus the expectation for a site to be segregating is $p(t)$. When comparing two sequences with n sites, we observe n draws from this Bernoulli distribution (one for each site), and x/n is the sample expectation for a site to be segregating. In the method of moments, the estimate for the expectation $p(t)$ is the sample expectation, i.e. $\hat{p} = \frac{x}{n}$. Thus, our estimator for d is, based on the method of moments estimator for $p(t)$,

$$\hat{d} = -\frac{3}{4}\log\left(1 - \frac{4}{3}\frac{x}{n}\right). \tag{5.18}$$

When estimating a parameter from the data, the estimated parameter does not necessarily match the "true" value and thus a parameter estimate is normally reported with a measure of uncertainty such as the variance or a confidence interval. To calculate the variance of D, one can apply the so-called *delta technique*. We refer the interested reader to [**Yang2014**] where this technique is described in detail in Appendix B. We will below explain how to obtain confidence intervals for the maximum likelihood method.

Example: For the alignment in Figure 5.7 we count $x = 2$ differences for the total length of $n = 8$ nucleotides. The estimated probability of substitution is therefore $\hat{p} = x/n = 2/8 = 0.25$. The distance according to the JC69 pairwise distance formula just derived is $\hat{d} = -\frac{3}{4}\log\left(1 - \frac{4}{3}\hat{p}\right) = -\frac{3}{4}\log\left(\frac{2}{3}\right) = 0.3$.

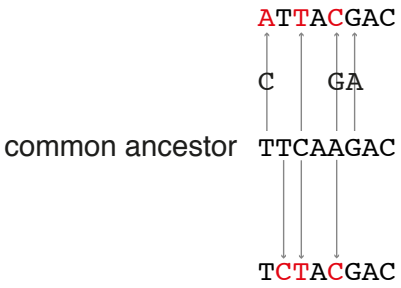


Figure 5.7: Example alignment of two sequences `ATTACGAC` at the top and `TCTACGAC` at the bottom. The middle sequence `TTCAAGAC` is the true (unknown) common ancestor sequence of the two sequences. The letters at the lines that connect the common ancestor sequence to its descendant sequences are the substitutions that occurred after the common ancestor split into the two lineages that resulted in the descendant sequences.

5.3.3 Pairwise distances using a maximum likelihood approach for JC69

In this section, we derive the *maximum likelihood estimator* for pairwise distances. Maximum likelihood estimators are explained in Box 18. In Box 19 we explain the concept of confidence intervals and how to obtain confidence intervals for maximum likelihood estimators.

Box 18: Maximum likelihood estimator

Let $P(X = data|\theta)$ be a probability density of a random variable X given parameters θ. Then the function in θ,

$$L(\theta; data) := P(X = data|\theta), \tag{5.19}$$

is called the *likelihood function* or short *likelihood*. For observed data *data*, the *maximum likelihood estimator*, short *MLE*, is,

$$\hat{\theta} = \text{argmax}_\theta \text{L}(\theta; \text{data}). \tag{5.20}$$

Very often the terms probability and likelihood are used interchangeably. However, we have just seen that these terms describe very different concepts in a strict mathematical sense.

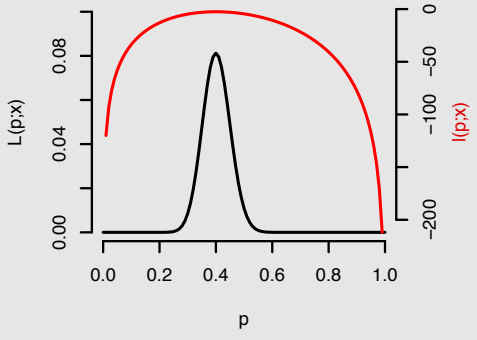
We illustrate the concept of MLE with die rolling: Based on obtaining a six $x = 40$ out of $n = 100$ independent die rolls, we want to estimate the parameter "probability to obtain a 6", denoted by p, using a MLE. Let the random variable X denote the number of sixes out of n die rollings. $x = 40$ is one realisation of this random variable, i.e. our data is $data = x$ and our parameter is $\theta = p$. The random variable X is binomially distributed, i.e. $P(X = x|p) = \binom{n}{x} p^x (1-p)^{n-x}$. Thus, the likelihood function in our experiment is,

$$L(p; x) = P(X = x|p) = \binom{100}{40} p^{40}(1-p)^{60}$$

The MLE is the value of p that best explains the observed data, i.e. that maximises Equation 5.21. A necessary condition for minima and maxima is that the first derivation equals 0. As we are only interested in the value at which the likelihood function takes its maximal value, the likelihood function can be transformed with functions that do not move the location but only the actual value of the maximum, such as taking the logarithm. This specific transformation is called the log-likelihood function, $l(p; x)$:

$$l(p; x) = \log L(p; x) \overset{\text{example}}{=} \log \binom{100}{40} + 40 \log p + 60 \log(1-p)$$

As the log-likelihood function from our die rolling exemplifies, the logarithm changes multiplications to sums, and thus facilitates obtaining an analytical solution of the maximisation problem. The following figure shows the likelihood (black) and log-likelihood (red) function for the die experiment.



To find the location of the maximum of the log-likelihood function, we calculate the first derivative of the log-likelihood function in respect to p and set the derivative to 0, which results in our example in $\hat{p} = \frac{x}{n} = \frac{40}{100} = 0.4$. Since the second derivative is < 0, we conclude that this is indeed a maximum (as already seen in the figure).

Box 19: Confidence interval

A confidence interval (CI) is a measure of uncertainty around a particular estimate. The interval is an estimate itself and depends on the observed realisation of a random experiment. Assume we repeatedly determine the interval estimate for each realisation of a random experiment whose distribution is parameterized in ϑ (e.g. in tossing a coin, $\vartheta = p$, the probability of seeing head). The interval estimate is called $(1-\alpha) \times 100\%$ *confidence interval* if the true parameter ϑ lies within the interval estimate in $(1-\alpha) \times 100\%$ of the repeated random experiments. Note that if ϑ is a vector (e.g. $\vartheta = (\mu, \sigma^2)$ in a Normal distribution), the interval is in fact a region.

The likelihood framework offers an easy estimate of the CIs. For given data, the MLE (see Box 18) is denoted with $\hat{\theta}$, and the *log likelihood ratio function* is defined as:

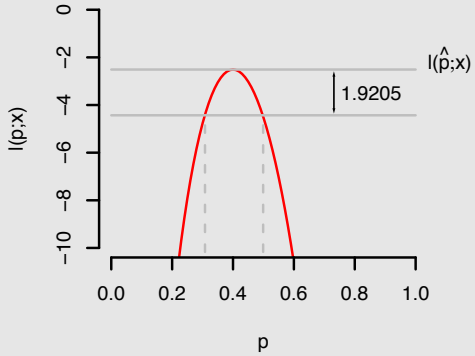
$$LR(\hat{\theta}, \vartheta) = 2(l(\hat{\theta}; data) - l(\vartheta; data)) = \log\left(\frac{L(\hat{\theta}; data)}{L(\vartheta; data)}\right)^2 \tag{5.21}$$

The right side of equation 5.21 demonstrates why this function is called log likelihood *ratio* function. When data is generated under ϑ, one can show that $LR(\hat{\theta}, \vartheta)$ (applying Wilk's theorem (see also Chapter 7.2.1)) follows approximately a χ^2_k- distribution (see Box 3):

$$LR(\hat{\theta}, \vartheta)) \sim \chi^2_k$$

The degree of freedom k corresponds in this case simply to the length of the parameter vector ϑ. The $(1-\alpha) \times 100\%$ CI is the set of parameter values θ for which the estimated $\hat{\theta}$ would not lead to rejection of the hypothesis that θ equals the true parameter at the significance level α (see Box 1). Thus, the set of θs with $LR(\hat{\theta}, \theta) < \chi^2_{k,\alpha}$ is the CI.
Each value θ in the $(1-\alpha) \times 100\%$ CI is a candidate for being the true ϑ, and in $\alpha \times 100\%$ of cases, ϑ is not contained inside the $(1-\alpha) \times 100\%$ confidence interval.

Example: In order to calculate the 95% confidence interval of the parameter $\theta = p$ for the die throwing example from Box 18, we first calculate the value of $l(\hat{\theta}; x)$. Then, we look up the value of $\chi^2_{k,5\%}$ in the χ^2-table, and calculate the values for ϑ such that $l(\vartheta; x) > l(\hat{\theta}; x) - 0.5\chi^2_{k,5\%}$:



Based on this procedure, we can determine the 95% confidence interval for the probability of throwing a 6 estimated out of a realisation of 100 times die rolling and obtaining 40 times 6, which is $[0.308, 0.499]$. Note that if the die was fair, we expected $p = 1/6$, which does not lie within the 95% CI in this particular realisation.

5.3.3.1 Maximum likelihood estimate and confidence interval of pairwise distance under the JC69 model

We can use the maximum likelihood framework just described to estimate the distance between a pair of sequences. The probability of a substitution under the JC69 model in time t is $p = 3p_1(t)$, where $p_1(t) = \frac{1}{4} - \frac{1}{4}e^{-4\lambda t}$. In a sequence of n nucleotides with x differences between two sequences, the probability of substitution is binomially distributed:

$$P(x \text{ substitutions out of } n \text{ nucleotides}) = \binom{n}{x} p^x (1-p)^{n-x} \tag{5.22}$$

After substituting for p by $3p_1(t) = \frac{3}{4} - \frac{3}{4}e^{-4\lambda t} = \frac{3}{4} - \frac{3}{4}e^{-\frac{4}{3}\lambda d}$, equation 5.22 is equal to

$$\binom{n}{x} \left(\frac{3}{4} - \frac{3}{4}e^{-\frac{4}{3}d}\right)^x \left(\frac{1}{4} + \frac{3}{4}e^{-\frac{4}{3}d}\right)^{n-x} = L(d;x)$$

which defines the likelihood function. Log-transformation of this leads to:

$$l(d;x) = \log\binom{n}{x} + x\log\left(\frac{3}{4} - \frac{3}{4}e^{-\frac{4}{3}d}\right) + (n-x)\log\left(\frac{1}{4} + \frac{3}{4}e^{-\frac{4}{3}d}\right) \tag{5.23}$$

To obtain the maximum likelihood estimate of the distance under JC69, we need to differentiate the equation 5.23 with respect to d, and set the derivative equal to 0. We then obtain $\hat{d} = -\frac{3}{4}\log\left(1 - \frac{4x}{3n}\right)$. It turns out that the maximum likelihood estimate is the same as the *method of moments* approach under JC69 (see chapter 5.3.2).

Let us calculate the maximum likelihood estimate and the confidence interval of the distance under JC69 with the two sequences shown in Figure 5.7. There are 2 differences between the 8 nucleotide long sequences, so the maximum likelihood distance estimate is $\hat{d} = -\frac{3}{4}\log\left(1 - \frac{4\times 2}{3\times 8}\right) = 0.3$. The confidence intervals are estimated to be $[0.05, 1.17]$ (grey dotted lines in the Figure 5.8). In particular, the CI is not symmetric around the MLE value. Notice that the uncertainty in the distance estimate is very large (wide CI). This is due to the small amount of information our sequences carry (i.e. short sequences). If we had more data, i.e. longer sequences, we would have more confidence in our estimate. In fact, when we repeated the calculations for a sequence of length 800 instead of 8, with 200 differences instead of 2 between the two sequences, we obtained the same MLE of $\hat{d} = -\frac{3}{4}\log\left(1 - \frac{4\times 200}{3\times 800}\right) = 0.3$ but CI= $[0.26, 0.35]$. As expected the CI is much narrower than in the short (8 nucleotide long) sequence example—see Figure 5.9.

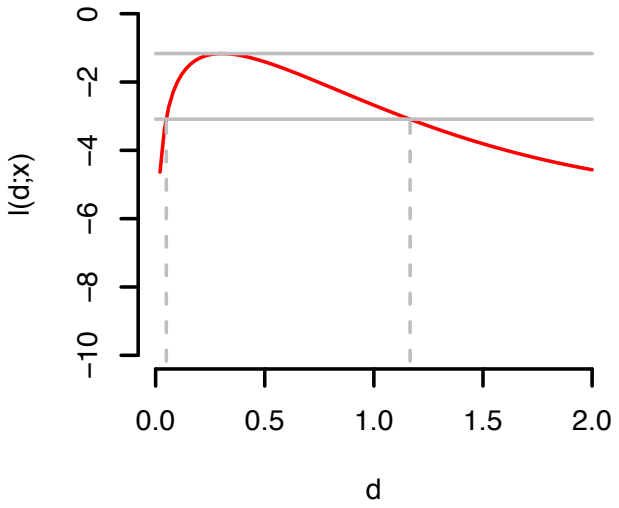


Figure 5.8: Maximul likelihood estimation for distance of two 8 nucleotides long sequences. The log likelihood $l(d;x)$, y-axis, is a function of sequence distance d, on the x-axis and the number of differences between the two sequences x. The maximum likelihood distance estimate is the point where the red curve touches the upper grey horizontal line. The 95% confidence interval is shown as with grey dashed lines.

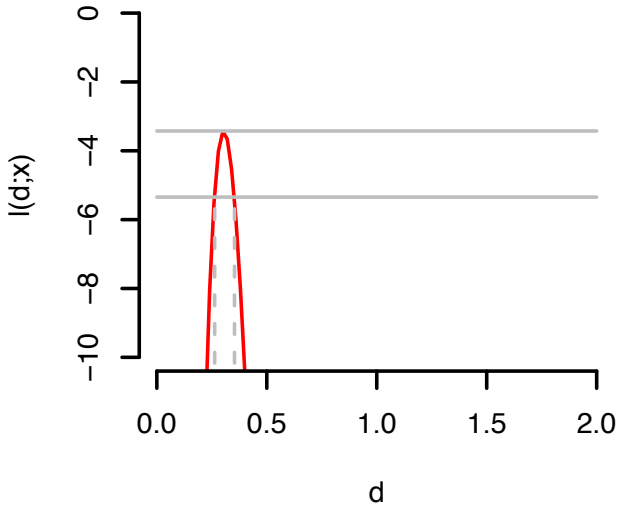


Figure 5.9: Same as Figure 5.8 but this time the calculations are done for sequences of 800 nucleotides, with a 200 nucleotide difference between the two aligned sequences. The main observation is that the function peaks more sharply, resulting in a much narrower confidence interval than before.

5.4 Allowing for rate variation across sites

So far, we only considered models where all sites evolve under the same model. However, this may not always be a reasonable assumption because the substitution rates might vary between different sites in the sequence. This variability can be due to variable mutation rates in different parts of the genome (e.g the polymerase could have different error rates across different parts of the genome), or due to variable selective pressure on different parts of the phenotype (e.g. a part of the viral sequence could be under strong selective pressure to fixate a mutation in order for the amino acids on the phenotypic level to escape the host immune system and another part of the same sequence could be under strong pressure to remain the same in order to allow the virus to enter the host cells whose receptors are conserved). Thus, we extend the substitution rate models to account for this variability.

Box 20: The Γ(Gamma)- distribution

The Γ distribution lives on $[0, \infty[$ with parameters $\alpha > 0, \beta > 0$, and has the probability density function,

$$g(x; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} e^{-\beta x} x^{\alpha-1} \tag{5.24}$$

for $x \geq 0$ and

$$\Gamma(\alpha) := \int_0^\infty e^{-t} t^{\alpha-1} \mathrm{d}t \tag{5.25}$$

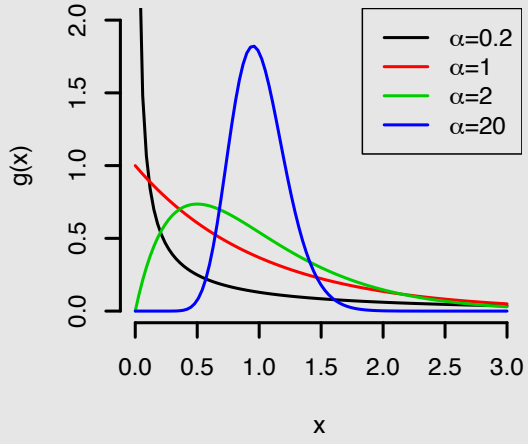
where the latter function is also called gamma-function and fulfils $\Gamma(n) = (n-1)!$ for n being a natural number $1, 2, 3, \ldots$. A gamma-distributed random variable with parameters α, β, i.e. a $\Gamma(\alpha, \beta)$-distributed random variable X, has mean

$$EX = \frac{\alpha}{\beta} \tag{5.26}$$

and variance

$$VarX = \frac{1}{\alpha} \tag{5.27}$$

In the following, we will use the Γ distribution with $\alpha = \beta$, i.e. with mean 1. The distribution is quite flexible in respect to α:



The extension of variable rates across different sites in the sequence is often modelled

by replacing constant rates by a Γ-distributed random variable (see Box 20) for each site (notation: JC69+Γ, K80+Γ, etc.). The Γ distribution is chosen such that the average substitution rate remains the same as in the original model. Thus, each site has a different substitution rate with the particular rate being Γ-distributed. In other words, when considering the rates for all sites in a sequence of length n, this empirical rate distribution corresponds to n draws from the Γ-distribution.

For the JC69 evolutionary model with Γ distributed rates, we replace the single λ parameter by λR where R is a $\Gamma(\alpha, \alpha)$-distributed random variable. According to Equation 5.26, the chosen Γ-distribution has mean 1, and thus $E(\lambda R) = \lambda$. The transition probability then becomes $p(t) = \frac{3}{4} - \frac{3}{4}e^{-4\lambda R t} = \frac{3}{4} - \frac{3}{4}e^{-\frac{4}{3}dR}$.

For a particular site, we do not know which value the random variable R takes. Thus we average over all possible values for R to obtain the expected transition probability:

$$E(p) = \int_0^\infty (\frac{3}{4} - \frac{3}{4}e^{-\frac{4}{3}dr}) g(r; \alpha, \alpha) \mathrm{d}r = \frac{3}{4} - \frac{3}{4}\left(1 + \frac{4d}{3\alpha}\right)^{-\alpha}$$

To calculate the pairwise sequence distance under this extended model we again (as in Section 5.3.2) equate the observed proportion of different sites between the two sequences $\frac{x}{n}$ (i.e. the sample expectation for a site to be segregating) to the expectation of a site to be segregating, $E(p)$, to get $\hat{d} = \frac{3}{4}\alpha\left(\left(1 - \frac{4}{3}\hat{p}\right)^{-1/\alpha} - 1\right)$. Furthermore, one can employ the maximum likelihood framework, as was done in Section 5.3.3, to derive maximum likelihood distances.

Let us again take our example alignment from Figure 5.7. The length of the two sequences is $n = 8$ each of which $x = 2$ differ. Thus $\hat{p} = \frac{x}{n} = \frac{1}{4} = 0.25$. We include the site variation into account by taking $\Gamma(2, 2)$-distributed, i.e. $\alpha = 2$, substitution rates. This results in an estimated distance of

$$\hat{d}_{\mathrm{JC69+\Gamma}} = \frac{3}{4}\alpha\left(\left(1 - \frac{4}{3}\hat{p}\right)^{-1/\alpha} - 1\right) = 0.34 > 0.3 = \hat{d}_{\mathrm{JC69}}$$

between the two sequences. The distance estimate including site variation is bigger than what we obtained when we did not consider a simple JC69 model. Therefore, ignoring the site variation—given site variation is present—leads to underestimation of the sequence distance in our example. This holds in general:

Theorem 5.4.1. *In the JC69 model, a lack of site-site rate variation leads to a smaller sequence distance estimate compared to assuming a Γ-distributed site-site rate variation.*

Proof. We derived the estimators for JC69 to be $\hat{d}_{\mathrm{JC69}} = -\frac{3}{4}\log\left(1 - \frac{4}{3}\hat{p}\right)$ and

JC69+Γ to be $\hat{d}_{\text{JC69+}\Gamma} = \frac{3}{4}\alpha\left(\left(1-\frac{4}{3}\hat{p}\right)^{-1/\alpha}-1\right)$. Thus, we need to proof that

$$-\frac{3}{4}\log\left(1-\frac{4}{3}\hat{p}\right) \leq \frac{3}{4}\alpha\left(\left(1-\frac{4}{3}\hat{p}\right)^{-1/\alpha}-1\right) \tag{5.28}$$

for all $\alpha > 0$ and $0 \leq \hat{p} < 3/4$.

By multiplying both sides of Equation 5.28 with $\frac{4}{3\alpha}$, applying $a\log x = \log x^a$, and exponentiating both sides, we see that proofing Equation 5.28 is equivalent to proving that

$$\left(1-\frac{4}{3}\hat{p}\right)^{-\frac{1}{\alpha}} \leq \exp\left(\left(1-\frac{4}{3}\hat{p}\right)^{-1/\alpha}-1\right) \tag{5.29}$$

We define $x = \left(1-\frac{4}{3}\hat{p}\right)$. As $0 \leq \hat{p} < 3/4$, x ranges between 0 and 1, i.e. $0 < x \leq 1$. We now expand the right side of Equation 5.29 using the definition 5.2 of the exponential function (Box 13):

$$\begin{aligned}
\exp\left(\left(1-\frac{4}{3}\hat{p}\right)^{-1/\alpha}-1\right) &= \exp\left(x^{-1/\alpha}-1\right) \\
&= \sum_{n=0}^{\infty}\frac{\left(x^{-1/\alpha}-1\right)^n}{n!} \\
&= 1+\left(x^{-1/\alpha}-1\right)+\sum_{n=2}^{\infty}\frac{\left(x^{-1/\alpha}-1\right)^n}{n!} \\
&= x^{-1/\alpha}+\sum_{n=2}^{\infty}\frac{\left(x^{-1/\alpha}-1\right)^n}{n!}
\end{aligned}$$

Because $0 < x \leq 1$, it follows that $x^{-1/\alpha} = \frac{1}{x^{1/\alpha}} \geq 1$ and thus $(x^{-1/\alpha}-1) \geq 0$. We can then conclude

$$\exp\left(x^{-1/\alpha}-1\right) = x^{-1/\alpha}+\underbrace{\sum_{n=2}^{\infty}\frac{\left(x^{-1/\alpha}-1\right)^n}{n!}}_{\geq 0} \geq x^{-1/\alpha}$$

Due to the definition of $x = \left(1-\frac{4}{3}\hat{p}\right)$, we obtain Equation 5.29, which proves that the JC69 distance is always less or equally big than the JC69+Γ distance. □

5.4.1 Distance estimators

Table 5.2 lists a collection of distance estimators for a number of substitution models, with and without site-to-site rate heterogeneity. Not all substitution models are present in this list (GTR and UNREST for example), as these models lack closed-form solutions for the transition probability functions from which the distance estimators in this list are derived.

model	distance estimator
JC69[**Jukes1969**]	$\hat{d}_{\text{JC69}} = -\frac{3}{4}\log\left(1-\frac{4}{3}\hat{p}\right)$ where $\hat{p} = x/n$, x number of difference, n sequence length
K80[**Kimura1980**]	$\hat{d}_{\text{K80}} = \frac{1}{2}\log(1-2S-V) - \frac{1}{4}\log(1-2V)$ where S = proportion of sites with transitional differences V = proportion of sites with transversional differences
HKY[**Hasegawa1984**]	$\hat{d}_{\text{HKY}} = \left(\frac{\pi_T\pi_C}{\pi_T+\pi_C} + \frac{\pi_A\pi_G}{\pi_A+\pi_G}\right)a - 2\left(\frac{\pi_T\pi_C(\pi_A+\pi_G)}{\pi_T+\pi_C} + \frac{\pi_A\pi_G(\pi_T+\pi_C)}{\pi_A+\pi_G} - (\pi_T+\pi_C)(\pi_A+\pi_G)\right)b$ where $a = -\log\left(1 - \frac{S}{2\left(\frac{\pi_T\pi_C}{\pi_T+\pi_C} + \frac{\pi_A\pi_G}{\pi_A+\pi_G}\right)} - \frac{\left(\frac{\pi_T\pi_C(\pi_A+\pi_G)}{\pi_T+\pi_C} + \frac{\pi_A\pi_G(\pi_T+\pi_C)}{\pi_A+\pi_G}\right)V}{2\left(\pi_T\pi_C(\pi_A+\pi_G) + \pi_A\pi_G(\pi_T+\pi_C)\right)}\right)$ $b = -\log\left(1 - \frac{V}{2(\pi_T+\pi_C)(\pi_A+\pi_G)}\right)$ and S, V as defined for K80
TN93[**Tamura1993**]	$\hat{d}_{\text{TN93}} = \frac{2\pi_T\pi_C}{\pi_Y}(a_1 - \pi_R b) + \frac{2\pi_A\pi_G}{\pi_R}(a_2 - \pi_Y b) + 2\pi_Y\pi_R b$ where $a_1 = -\log\left(1 - \frac{(\pi_T+\pi_C)S_1}{2\pi_T\pi_C} - \frac{V}{2(\pi_T+\pi_C)}\right)$ $a_2 = -\log\left(1 - \frac{(\pi_A+\pi_G)S_2}{2\pi_A\pi_G} - \frac{V}{2(\pi_A+\pi_G)}\right)$ $b = -\log\left(1 - \frac{V}{2(\pi_T+\pi_C)(\pi_A+\pi_G)}\right)$ and V as defined for K80; S_1 proportion of sites with two different pyrimidines (`T,C`) S_2 proportion of sites with two different purines (`A,G`)
JC69+Γ	$\hat{d}_{\text{JC69}+\Gamma} = \frac{3}{4}\alpha\left(\left(1-\frac{4}{3}\hat{p}\right)^{-1/\alpha} - 1\right)$ α determines the shape of the Γ distribution
K80+Γ	$\hat{d}_{\text{K80}+\Gamma} = \frac{\alpha}{2}\left((1-2S-V)^{-\frac{1}{\alpha}} - 1\right) + \frac{\alpha}{4}\left((1-2V)^{-\frac{1}{\alpha}} - 1\right)$ with α as defined for JC69+Γ, and S, V as defined for K80

Table 5.2: Overview of distance estimators for a selection of different nucleotide substitution models.

5.5 Amino acid substitution models

In the previous section we focused on the evolution of nucleotides and the models aimed at quantifying the rate of nucleotide substitution. It is however at the level of the phenotype that selection pressure takes effect. The focus of this section will be studying evolution at this level and its quantification using amino acid models.

Example: The human immunodeficiency virus (HIV) is a persistent infection and adapts to ever-changing environments quickly. The envelope protein is the only protein that sticks out of the viral membrane and is visible to the immune system. As soon as an individual becomes infected with HIV, the immune system starts producing antibodies directed against the envelope protein on the surface of the HIV virions. The immune system thus imposes selective pressure on the virus species and this leads to resistant mutations. The adaptation of the amino acid sequence of the HIV envelope protein in response to the antibody response of the host immune system is an example of evolution acting at the level of phenotype—see Figure 5.10.

The amino acid composition of the envelope protein sequences in Figure 5.10 changes over time (the weeks post infection (wpi) are displayed on the left of the figure). Especially, the three sites 276, 279 and 456 have a huge impact on the success or failure of the autologous antibody response. The pie charts show how often mutations were found at these sites in the sampled viral sequences. We observe that the fraction of viral strains bearing mutations at these sites increase in general, which can be seen as a hint of selection of viral variants growing more and more resistant against the autologous response.

This example nicely illustrates viral evolution on the phenotypic level. As evolution is the result of mutation and selection, we are interested in whether we can quantify the substitution rate of the amino acids and detect the presence of selection. For estimating substitution rates, we need to extend the substitution models presented earlier. For discovering hints of selection we need to derive further tests.

5.5.1 Definition of amino acid substitution models

Just as for the nucleotide level, we first need to define a substitution model to quantify the substitution rate on the amino acid level. This amino acid (AA) substitution model also allows us to define a measure of distance between two sequences. The process underlying the amino acid substitution is again modelled as a Markov chain, with all the properties mentioned before. The transition probability matrix is derived by applying $P(t) = e^{Qt}$. However, instead of four nucleotides there are 20 amino acids. Thus, the state space of the Markov model has dimension 20 and therefore the substitution rate matrix – and thus the transition probability matrix – have 20×20 dimensions in amino acid substitution rate models, with, in the most general case, 380 parameters (the diagonal is again chosen such that the row sum is 0).

The amino acid substitution models are more difficult to configure than the nu-

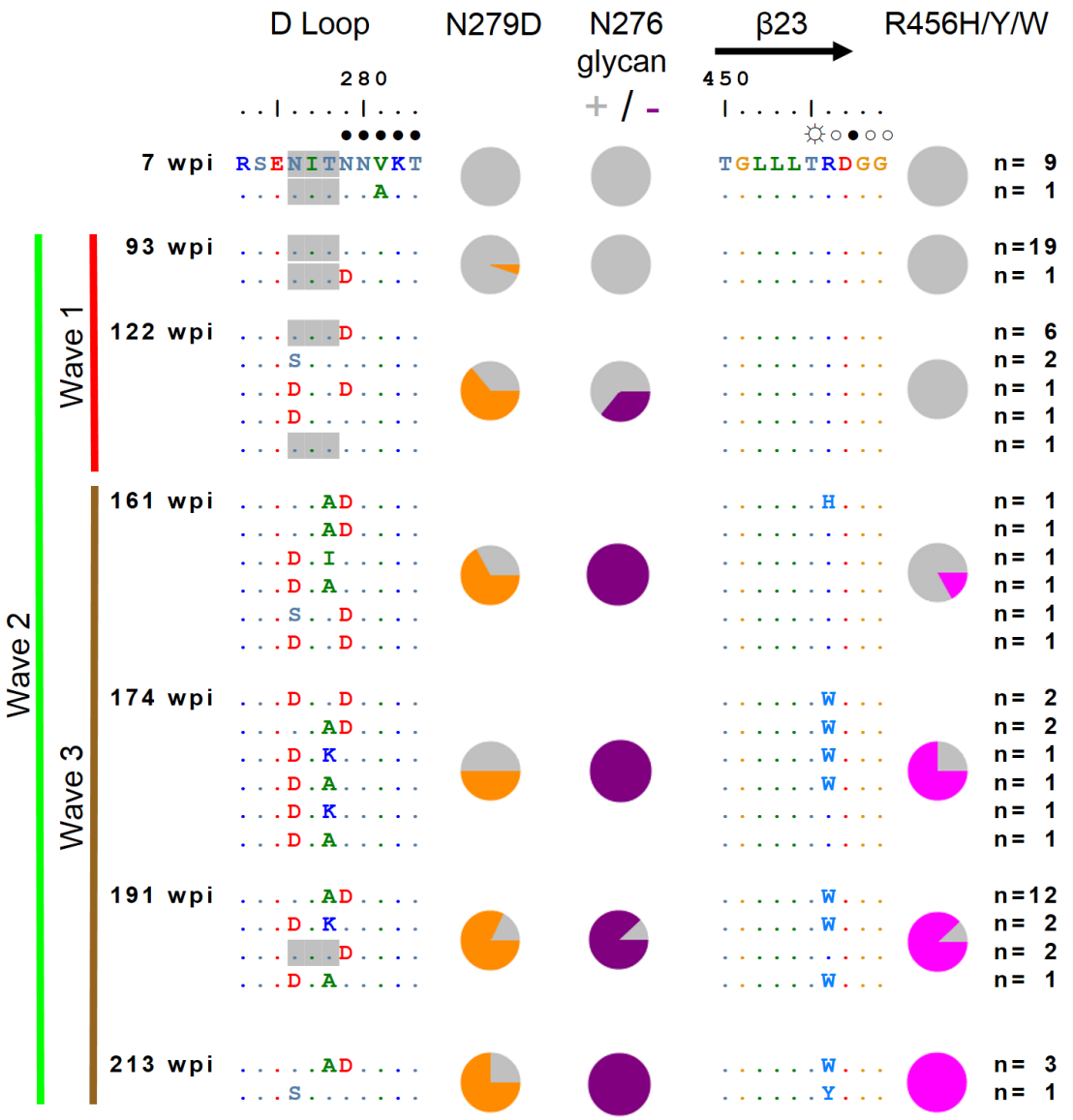


Figure 5.10: The figure shows the adaptation of the envelope protein, i.e. the protein on the surface of the HI-virus, to the immune pressure exerted by the immune system of the host over time. Viral strains were isolated over 7 weeks post infection (wpi) up to 213 wpi, of which small parts are shown in this figure, namely D loop and β23. In the patient, three waves of antibodies were identified, directed against different targets. Their timing is summarized at the left. Three specific mutations (N279D, N276 glycan, R456H/Y/W) were responsible for the failure of the antibodies. The pie charts show the presence of these mutations in the isolated viral strains at the different time points. Figure adapted from [**Wibmer2013**]

cleotide substitution models. To define the Q matrix, one can use empirical substitution rates—meaning fixed rates which were previously suggested for the particular

biological system—to fill in the matrix. Alternatively, a probabilistic model can be used, which takes into account the properties of the individual amino acids and how easy it is to change from one amino acid to another, either from a chemical or a codon point of view. In both cases however, the Q matrix should be defined such that it ensures the time-reversibility of the model as otherwise calculating the tree likelihood of trees will become difficult. Note that while the entries of the Q matrix are typically estimated based on the sequences on the nucleotide level, the entries are specified empirically or mechanistically as typical datasets do not contain information on up to 380 parameters.

5.5.2 JC69-like distance estimation for amino acid sequences

To estimate the distance between amino acid sequences with an empirical or mechanistic Q-matrix, we use the same procedure as for the nucleotide models. In the simplest, JC69-like model of AA substitution, all substitutions have the same rate λ. Thus, the mean rate of substitution is 19λ. The expected time to substitution is $\frac{1}{19\lambda}$. This translates to time $t = \frac{d}{19\lambda}$ between two AA sequences. The distance estimator between the two sequences is $\hat{d} = \frac{19}{20}\log\left(1 - \frac{20x}{19n}\right)$ (where n is the length of the sequences and x the number of substitutions).

5.6 Codon substitution models

We will now briefly discuss the construction and properties of codon substitution models. These models allow for estimating the presence of selection acting on (parts of) the sequences.

5.6.1 Definition of codon substitution models

A *codon* consists of three nucleotides and encodes for one amino acid (see Figure 1.4). As there are four nucleotides, there are $4^3 = 64$ possible codons. However, during translation, i.e. the transcription of DNA or RNA into proteins, three codons, i.e. `TAA`, `TAG` and `TGA`, stop the translation process. The codon substitution models disregard these three *stop codons* because any premature stop codons in the protein coding sequence usually cause the sequence to be translated into a non-functional protein. Thus, the codon models account for transition between 61 codons, resulting in very large substitution rate and transition probability matrices (61×61 entries). One of the codons, `ATG`, the so-called start codon, serves as the biological "barcode" signalling that the protein code starts at that position. The start codon and the remaining 60 codons each encode for an amino acid (AA). However, there are only 21 AA of which only 20 are physiologically relevant and appear in the genetic code. This means that the same AA can be encoded by one or several codons. A brief look at the "codon sun" in Figure 1.4 makes it obvious that some nucleotide substitutions do not lead to any changes at the phenotypic level and thus are less likely to be under selection.

The complexity that the codon models need to take into account is illustrated in the example of possible substitutions of the CTA codon and their effect on the amino acid the new triplet will code for, displayed in Figure 5.11. Overall, each codon can change into 9 other codons with one substitution. Recall that transversions usually occur less frequently than transitions. Interestingly, codons that result from a nucleotide transition on the third position most often translate to the same AA, and the codons that result from a transversion often produce a different AA. Nucleotide substitutions in the codon leading to the same amino acid are called *synonymous substitutions* and nucleotide substitutions in the codon leading to different amino acids are called *non-synonymous substitutions*.

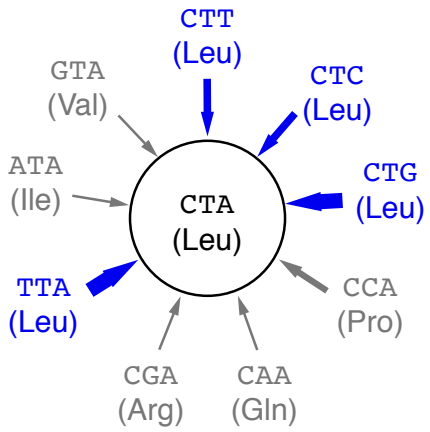


Figure 5.11: The figure shows all possible codons the CTA codon could mutate into (or from) with only one substitution. For each codon, the corresponding amino acid is shown as a three letter code in the brackets below. The non-synonymous substitutions are shown in grey. The synonymous substitutions are in blue. The bigger arrows show transitions, the smaller arrows the transversions. Figure adapted from [**Yang2014**].

Just as for nucleotide and AA models, the codon models are Markov chain models but with 61 possible states. We denote codons in the codon models with capital letters (e.g. I, J), and nucleotides with small letters (e.g. i, j). In the codon models it is assumed that the rate of change from I to J is zero if I and J differ in more than one nucleotide position, i.e. only single nucleotide changes are allowed. In general, this means that we may assume arbitrary rates for the transition of each codon into the nine codons which are one substitution away, which makes $9 \times 61 = 549$ parameters. It is typically unfeasible to estimate that many parameters. Instead, some assumptions are made in the common codon models [**Goldman:1994**]. First, common codon models assume that the ratio of synonymous transition rate for codon I and synonymous transversion rate for codon I is the same across all codons I. Similarly, the ratio of non-synonymous transition rate for codon I and non-synonymous transversion rate for codon I is often assumed to be the same across all codons I. The associated parameters are κ for the transition/transversion ratio and ω for the non-synonymous/synonymous rate ratio. Furthermore, let π_I be the equilibrium

frequency of codon I. One can assume that each codon frequency is a free parameter (with their sum being 1). In order to reduce the number of parameters in the common codon models, it is assumed that $\pi_I = \frac{1}{C}\pi_{i_1}\pi_{i_2}\pi_{i_3}$, with I consisting of nucleotides i_1,i_2,i_3 with equilibrium frequencies π_{i_1},π_{i_2},π_{i_3}. Thus, it is assumed that each codon is obtained by randomly choosing three nucleotides. Each off-diagonal entry in the substitution rate matrix is then defined as:

$$q_{IJ} = \begin{cases} 0 & \text{if I and J differ at more than 1 positions} \\ \pi_J & \text{if I and J differ by a synonymous transversion} \\ \kappa\pi_J & \text{if I and J differ by a synonymous transition} \\ \omega\pi_J & \text{if I and J differ by a nonsynonymous transversion} \\ \omega\kappa\pi_J & \text{if I and J differ by a nonsynonymous transition} \end{cases}$$

5.6.2 Detecting selection: d_N/d_S ratio

The presence of selection acting on a sequence can be revealed by comparing the amount of synonymous to that of non-synonymous changes between two sequences. The idea behind the comparison of synonymous to non-synonymous changes is that if there are significantly more (or fewer) non-synonymous than synonymous changes, the protein was likely under selective pressure to specifically adapt its amino acid composition (or to remain unchanged). If, on the other hand, there are equally many synonymous as non-synonymous substitutions, there was likely no selection at all acting on the protein.

Comparing non-synonymous and synonymous substitutions between two sequences is a challenging task. We cannot compare the number of non-synonymous and synonymous sites directly, because the probability of a random substitution leading to a non-synonymous or synonymous substitution is not the same. Thus, we have to scale the counts of these sites by the number of possible substitutions of the respective type. Furthermore, in case a non-synonymous site is the result of two substitutions, we also have to alter the counts of the number of non-synonymous / synonymous sites.

Let us define d_N as the number of non-synonymous substitutions per non-synonymous site in our two sequences. Thus, d_N is the ratio between the number of non-synonymous substitutions between the two sequences and the number of non-synonymous sites, i.e. sites leading to non-synonymous substitutions. Similarly, let us define d_S as the number of synonymous substitutions per synonymous site in our two sequences. In the upcoming section, we provide one way to count these numbers in order to estimate d_N and d_S.

Typically the d_N/d_S ratio is reported, as it contains information on the abundance of selection between two nucleotide sequences. $d_N/d_S < 1$ means that non-synonymous substitutions happen less frequently than synonymous substitutions, often referred to as purifying selection (the genome is 'purified' and substitutions are selected against; this corresponds to highly conserved phenotypes). $d_N/d_S > 1$

means that non-synonymous substitutions occur more frequently than synonymous substitutions, which leads to positive selection that accelerates the fixation of non-synonymous substitutions (this corresponds to hypermutation). $d_N/d_S = 1$ means that there is no selection, synonymous and non-synonymous substitutions happen at the same rate.

Several methods have been introduced for determining the d_N/d_S-ratio, see [**Yang2014**] for an overview. Here, we will have a look at the counting method introduced by [**NeiGojobori:1986**] in order to understand important concepts concerning the d_N/d_S-ratio.

5.7 Counting method

We introduce the counting method for determining d_N/d_S [**NeiGojobori:1986**]. We follow these three steps:

1. Count the number of non-synonymous and synonymous differences between the two sequences, referred to N_d and S_d respectively.
2. Count the number of non-synonymous and synonymous sites in the two sequences, referred to N and S respectively. These are the site at which potential mutations can lead to non-synonymous and synonymous substitutions.
3. Account for the evolutionary history by applying the pairwise distance formula (such as Equation 5.18 when assuming JC69) to the ratios N_d/N and S_d/S.

We explain the counting method by looking at the two sequences `TTTCCTCCTCCT` and `TTCCAGCCTCCT` , which each can be subdivided into 4 codons:

	codon 1	codon 2	codon 3	codon 4
sequence 1	`TTT`	`CCT`	`CCT`	`CCT`
sequence 2	`TTC`	`CAG`	`CCT`	`CCT`

From the codon sun (Figure 1.4) we can directly see that codons `TTT` and `TTC` encode for `F` (phenylalanine), `CCT` encodes for `P` (proline), and `CAG` encodes for `Q` (glutamine). Thus, sequence 1 encodes for the amino acid sequence `FPPP` and sequence 2 encodes for the amino acid sequence `FQPP`. From the amino acid sequence, we see that there is a synonymous change in codon 1 and a non-synonymous change in codon 2.

Counting the number of non-synonymous and synonymous differences.

To calculate N_d and S_d, we consider each codon position I of the two aligned sequences separately and count the number of visible non-synonymous differences N_d^I and the number of synonymous differences S_d^I. We assume here that each nucleotide changed at most once in the evolution from one sequence to the other, thus we talk about number of differences (rather than number of substitutions as we may

have hidden substitutions as highlighted for nucleotides in Section 5.3). Further, in each time step, only one nucleotide may change. For the whole sequence, we have $N_d = \sum_{I=1}^{K} N_d^I$ and $S_d = \sum_{I=1}^{K} S_d^I$ with K being the number of codons in each sequence.

If the two codons at position I are the same, we have $N_d^I = S_d^I = 0$. If the two codons only differ in one nucleotide, we have either $N_d^I = 1$ and $S_d^I = 0$, or $N_d^I = 0$ and $S_d^I = 1$. If the two codons differ in two positions, we know that $N_d^I + S_d^I = 2$. However, we have two possible orderings of how substitutions in nucleotides accumulate (i.e. two possible evolutionary pathways), and this may lead to N_d^I and S_d^I being different in each pathway. If the two codons differ in all three positions, we even have six possible pathways, and we only know that for all of them $S_d^I + N_d^I = 3$ holds. If we have more than one possible pathway, we average the resulting N_d^I and S_d^I over the possible pathways (giving each pathway equal weight).

To illustrate this procedure, we now determine N_d and S_d for our example. We start by looking at the first codon. In sequence 1 this is `TTT` and in sequence 2 it is `TTC`. Both codons encode for the same amino acid phenylalanine (`F`). Thus, we count one synonymous substitution and no non-synonymous substitution. Codon 3 and 4 are the same, meaning no synonymous and no non-synonymous substitution took place. Codon 2 differs in two sites between sequence 1 and 2. As we assumed that only one substitution can occur per time step, we have two direct pathways for the two changes to occur. We need to average over the two possible ways:

pathway	S_d^2	N_d^2
`CCT (P)` → `CAT (H)` → `CAG (Q)`	0	2
`CCT (P)` → `CCG (P)` → `CAG (Q)`	1	1
average	0.5	1.5

This means that $S_d^2 = 0.5$ and $N_d^2 = 1.5$.

We can now calculate the sum of all non-synonymous and synonymous differences between the two sequences by summing up the differences on the different codon positions:

	codon 1		codon 2		codon 3		codon 4	
sequence 1	`TTT`		`CCT`		`CCT`		`CCT`	
sequence 2	`TTC`		`CAG`		`CCT`		`CCT`	
N_d =	0	+	1.5	+	0	+	0	= 1.5
S_d =	1	+	0.5	+	0	+	0	= 1.5

Counting the number of non-synonymous and synonymous sites.

Each codon consists of three nucleotides. Each nucleotide can mutate into the three other nucleotides. Thus, for each codon of the original sequences, we list all possible single substitutions and count whether this is a synonymous or non-synonymous substitution. Note that stop codons are not considered. Figure 5.12 shows all possible one point substitution codons of the four codons contained in our two example

sequences.

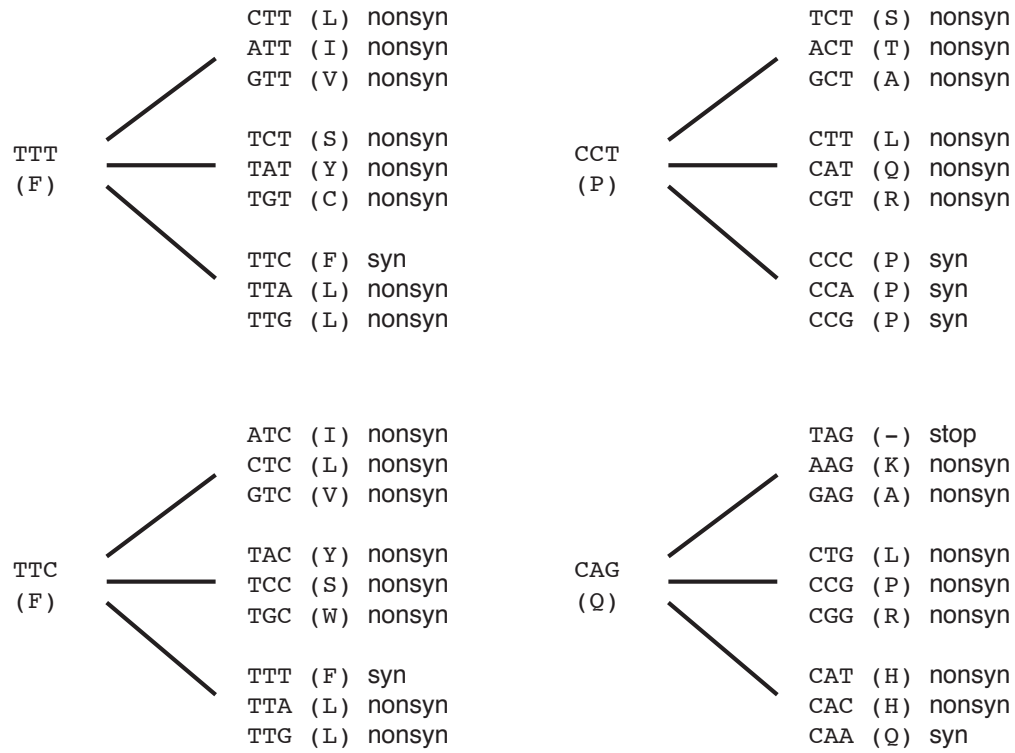


Figure 5.12: Overview of the four codons from our example sequences and all possible one point substitution codons.

The number of synonymous sites per codon I, N^I, is defined as the number of sites per codon (which is 3) times the probability to obtain a synonymous codon upon a single substitution. Likewise, the number of non-synonymous sites per codon I, S^I is defined as the number of sites per codon times the probability to obtain a non-synonymous codon upon a single substitution. Thus, $N^I + S^I = 3$. The number of non-synonymous and synonymous sites for a sequence is obtained by summing the number of non-synonymous and synonymous sites per codon over all codons. The number of non-synonymous and synonymous sites, N and S, for a pair of sequences is the average of the per-sequence numbers.

We determine the number of non-synonymous and synonymous sites, N and S, in our example. The codon `TTT` has nine possible codons it can mutate into with one single nucleotide change. Only when the `T` at the third site of the codon mutates into a `C`, the substitution is synonymous. Thus, 1 out of 9 possible substitutions are synonymous. According to the definition of number of synonymous sites per codon, we need to multiply this number with 3. In summary, codon `TTT` has $1/3$ synonymous sites. 8 out of 9 possible point substitutions lead to non-synonymous substitutions. Thus there are $3 \times 8/9 = 8/3$ non-synonymous sites in codon `TTT`.
With the same line of arguments, the codon `TTC` has $3 \times 1/9 = 1/3$ synonymous

sites and $3 \times 8/9 = 8/3$ non-synonymous sites. The codon `CCT` has $3 \times 3/9 = 1$ and $3 \times 6/9 = 2$ non-synonymous sites.
The codon `CAG` is a bit more complicated because the substitution from `C` to `T` on the first site leads to the stop codon `TAG`. Stop codons are not considered in the codon substitution models and the counting method, as they would lead to a non-complete protein which is normally not a functional protein any more. From the remaining 8 codons, only one codon is the result of a synonymous substitution. Thus, the codon `CAG` has $3 \times 1/8 = 3/8$ synonymous sites and $3 \times 7/8 = 21/8$ non-synonymous sites.

We can now calculate the number of non-synonymous and synonymous sites by averaging over the two sequences:

	codon 1		codon 2		codon 3		codon 4	
sequence 1	`TTT`		`CCT`		`CCT`		`CCT`	
sequence 2	`TTC`		`CAG`		`CCT`		`CCT`	
non-synonymous sites								
sequence 1	8/3	+	2	+	2	+	2	= 8.67
sequence 2	8/3	+	21/8	+	2	+	2	= 9.29
average								$N = 8.98$
synonymous sites								
sequence 1	1/3	+	1	+	1	+	1	= 3.33
sequence 2	1/3	+	3/8	+	1	+	1	= 2.71
average								$S = 3.02$

Accounting for evolutionary history.

We now can calculate the distance based on the ratios N_d/N and S_d/S. Similar to differences between two nucleotide sequences, the possible evolutionary steps between the two sequences are not taken into account when just looking at the raw ratios. This is why [**NeiGojobori:1986**] corrected these ratios using the distance formula based on the JC69 molecular evolution model (equation 5.18) and defined these distances as d_N and d_S respectively:

$$d_N = -\frac{3}{4}\log\left(1 - \frac{4}{3}\frac{N_d}{N}\right) \tag{5.30}$$

$$d_S = -\frac{3}{4}\log\left(1 - \frac{4}{3}\frac{S_d}{S}\right) \tag{5.31}$$

Using the values calculated above, we can directly see that

$$\frac{d_N}{d_S} = \frac{-\frac{3}{4}\log\left(1 - \frac{4}{3}\frac{N_d}{N}\right)}{-\frac{3}{4}\log\left(1 - \frac{4}{3}\frac{S_d}{S}\right)} = \frac{\log\left(1 - \frac{4}{3}\frac{1.5}{8.98}\right)}{\log\left(1 - \frac{4}{3}\frac{1.5}{3.02}\right)} = 0.23 \tag{5.32}$$

In our example, we obtain a hint of purifying selection because the d_N/d_S ratio is

less than 1.

However, the counting method includes many simplifications. The JC69 distance formula is based on the assumption that every nucleotide substitution occurs at the same rate. In particular, we do not take into account differences in transition and transversion rates and other codon biases. The counting method was therefore extended in the literature (e.g. [**Li:1993**, **Pamilo:1993**, **Comeron:1995**, **Ina:1995**, **Tzeng:2004**]). In addition, another class of models were introduced: the Maximum likelihood methods [**Goldman:1994**], in which the d_N/d_S ratio is estimated based on a Maximum-Likelihood estimator. As discussed in chapter **??** such methods have the advantage of providing a confidence interval.

6 Phylogenetic trees

Nothing in Evolution Makes Sense Except in the Light of Phylogeny
(unknown)

The next three chapters are on phylogenetics. In this chapter, we will discuss how phylogenetic trees are reconstructed based on genetic sequences. This is followed by two chapters introducing methods which allow us to understand genotypic (i.e. molecular) and phenotypic evolution processes occurring on such phylogenetic trees.

In what follows in this chapter, we will first provide examples of phylogenetic trees. Second, we will introduce mathematical notation for and properties of these phylogenetic trees. Third, we will discuss the three different approaches for reconstructing phylogenetic trees: phenetic approaches, cladistic approaches, and probabilistic approaches. The last type of methods, probabilistic methods, can be used in a maximum likelihood or a Bayesian statistical setting. The maximum likelihood approach will be discussed in detail in this chapter while the Bayesian approach will be explained in Chapter 10. We end this chapter by highlighting insights into HIV which were obtained *qualitatively* from reconstructed phylogenetic trees. In later chapters, we will discuss applying statistical methods to these reconstructed trees in order to develop *quantitative* insights into the evolutionary and population dynamic (e.g. epidemiological) processes governing the population from which the samples were taken.

6.1 Introduction to phylogenetic trees

The 1837 notebook of Charles Darwin shows a sketch of a phylogenetic tree (Figure 1.8). A phylogenetic tree displays evolutionary relationships between different individuals. The branch terminals, also called *tips*, represent *sampled individuals*, i.e. individuals of whom we have genetic (or other type of) information. An *internal node*, meaning the intercept of several branches, represents the most recent common ancestor of the individuals represented by the tips descending this node.

Phylogenetic trees were first reconstructed to display the evolutionary relationships of species. A genetic sequence of one individual per extant species is used in order to reconstruct the phylogeny, and each tip corresponds to one of these extant species. An internal node in a species tree represents a speciation event. In the simian phylogeny in Figure 6.1, we can see that humans are more closely related to chimpanzees than either humans or chimpanzees to gorillas. If lengths in calendar time units are

assigned to the branches in a tree, one can read off speciation times. E.g. based on Figure 6.2 displaying the phylogenetic tree of mammals, we can conclude that the most recent common ancestor of all mammals lived around 166 million years ago (note though there is a lot of uncertainty and controversy around this date). Sequences from different extant species can be sampled on different dates or even a few years apart. However, the evolution of species took millions of years, thus sampling of extant species within a few years time window can be interpreted as sampling at the same point in calendar time, visualized as putting all tips on the same point of a horizontal axis in Figure 6.1. Trees with all tips occur at the same time point are called *ultrametric trees*.

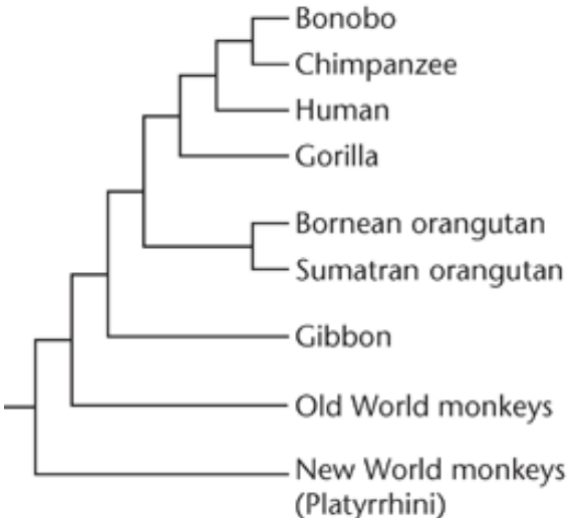


Figure 6.1: Phylogeny of simians with the subtrees of 78 old world monkeys and 53 new world monkeys being collapsed into a single tip each. Figure adapted from [**hominids**].

In a recent publication, Hinchliff et al. reconstructed the tree of life containing 2.3 million species [**Hinchliff2015**] depicted in Figure 6.3. Each tip in the tree in Figure 6.3 actually represents around 1000 extant species, meaning subtrees were collapsed into tips. This tree is thus only a first draft of the complete tree of life. In the future, additional genomic information will help to resolve the branching structure in the collapsed parts of the tree. The authors of this draft also developed a web tool (`https://tree.opentreeoflife.org/`) that allows detailed interactive exploration of the reconstructed tree of life.

Over the past decade, the phylogenetic framework has also been heavily employed in the context of viral infectious diseases. To determine how a virus spreads in the host population, the virus sequences are obtained from some infected individuals. The phylogeny of these viral sequences (one sequence per host) is then used as an approximation of the transmission tree. As an example, consider the HIV phylogenetic tree in Figure 6.4. Tips in this phylogeny represent viral samples of different patients. In Section 9.1.2, we generalize this concept, allowing samples to also correspond to internal nodes of a phylogeny (so-called sampled ancestors). An internal node repre-

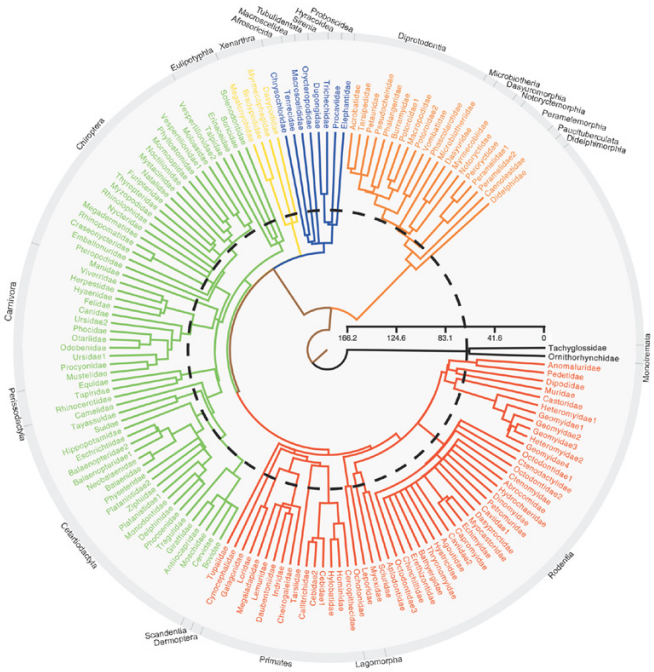


Figure 6.2: Phylogenetic tree of mammalian families. The extinction of dinosaurs (65 million years ago) is marked with a black dashed circle. Figure adapted from [**Bininda2007**].

sents a transmission of the pathogen from one host to another. The branch lengths in calendar time units can be interpreted as the time that has elapsed between the transmission events. Compared to the phylogenetic tree of species, where branches represent millions of years of evolution, the phylogeny of pathogens covers a much shorter period of time: usually months, years or decades. Thus, in viral phylogenies, typically not all tips are sampled at the same point in time, rather the tips in the phylogeny of a virus can be sampled at different points in time throughout the epidemic. We will discuss at the end of this chapter a number of insights into HIV which we can obtain using such phylogenetic trees.

Above, we assumed that we include one sequence per species, or one sequence per infected host. If we were to include several sequences per species (resp. infected host), we will obtain trees connecting the individuals within a species (resp. the different virions) nested within the species (resp. transmission) tree. This will be discussed further in Chapter **??**.

We will now introduce mathematical notation of phylogenetic trees as well as important properties, which will be required for the sections to follow on phylogenetic tree reconstruction and analysis.

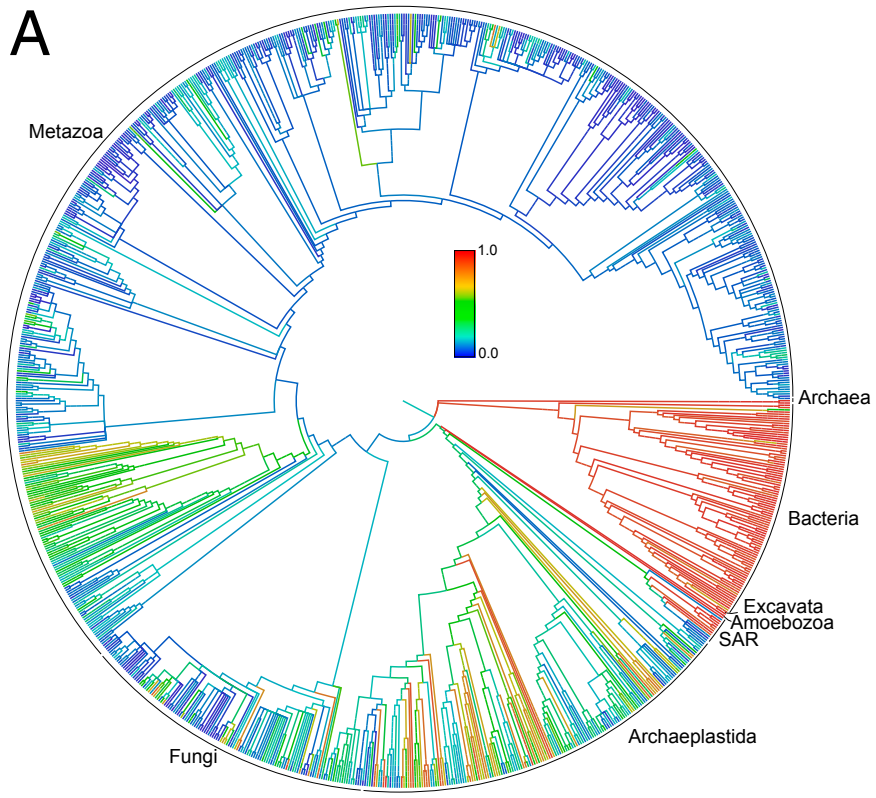


Figure 6.3: The phylogenetic tree of 2.3 million species (colors represent proportion of lineages represented in NCBI databases). Figure adapted from [**Hinchliff2015**].

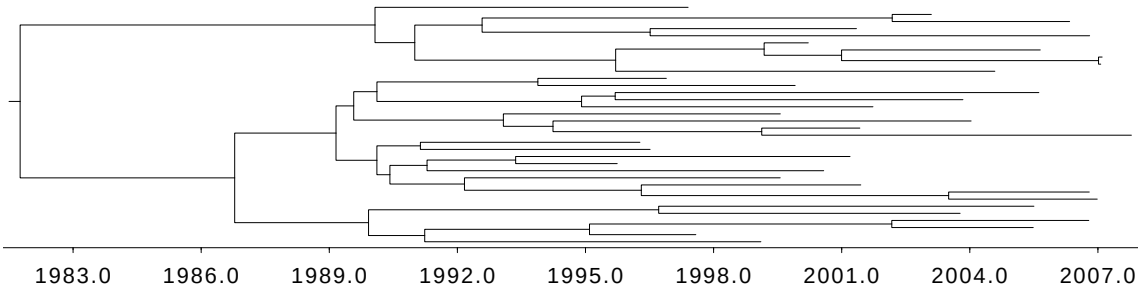


Figure 6.4: Phylogenetic tree of HIV patients [**Stadler2011R0**].

6.2 The mathematics of phylogenetic trees

6.2.1 The mathematical definition of a phylogenetic tree

A *tree* consists of nodes and branches, with branches connecting the nodes such that no cycle is formed (otherwise it would not be a tree, but rather a network). Here we consider *binary trees*[1] which may be unrooted or rooted. A node is of *degree* k if it has k branches attached. A binary *unrooted tree* (Figure 6.5, left) is defined as a

[1]In the case of virion tracking or superspreading, we require non-binary trees, see Chapter **??**.

tree with only degree-1 and degree-3 nodes. A degree-1 node is a *tip* of the tree, and a degree-3 node is an *internal node*. A binary *rooted tree* (Figure 6.5, right) is an unrooted tree with additionally having one degree-2 node, this node is called the *root* of the tree[2]. A rooted tree can be obtained from an unrooted tree by subdividing one of the branches into two by adding a new root node (in Figure 6.5, the branch leading to node B is subdivided by a root node). A *labelled tree* is a tree (rooted or unrooted) with each tip having a unique label assigned. Unless otherwise stated, a phylogenetic tree is a labelled tree (rooted or unrooted).

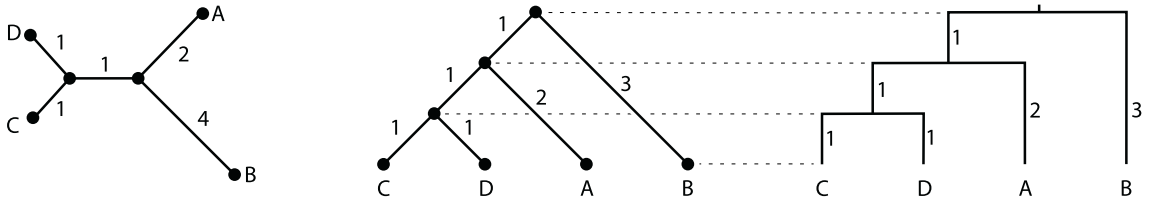


Figure 6.5: Unrooted phylogeny on 4 tips (left) and a corresponding rooted tree (middle). The right tree is equivalent to the rooted tree in the middle, but visualized as the trees in Figures 6.1-6.4.

A branch attached to a tip is often called a *pendant* branch (in Figure 6.5, both phylogenies have four pendant branches corresponding to the four tips A, B, C, and D). Two tips whose adjacent pendant branches join in the same internal node are called a *cherry* (e.g. (C, D) and (A, B) in Figure 6.5, left and (C, D) in Figure 6.5, right). A rooted tree containing a single cherry is referred to as a *caterpillar* tree (Figure 6.5, right). In a rooted tree, the set of tips descending from an internal node is also referred to as a *monophyletic group* or *clade*, e.g. tips (C, D, A) form a clade in Figure 6.5, right.

Note that in the rooted trees in Figures 6.1-6.4, branching events are not represented by a single node which has an ancester branch and two descendant branches attached. Instead, a branching event is displayed with the help of a line orthogonal to the branches from the root towards the tip. This is simply a different visualization of rooted trees: the branches from the root to the tip are proportional to time, while the orthogonal lines display relationships. We provide an example of this visualizaton in 6.5, right.

6.2.2 The Newick tree format

The drawing of a phylogenetic tree is its graphical representation. This is great to visually inspect the tree. However, for representing trees in a computer we require a more compact format. The most popular tree format is the *Newick format*. The name derives from that of the lobster restaurant in Dover (South Carolina) where it was invented [**Felsenstein2004**].

[2]Alternatively, one can define a rooted tree as an unrooted tree with one of the degree-1 nodes being the origin of the tree (rather than a tip), and its direct descendant node is the root. The advantage of that definition is that we can assign a length to the branch ancestral to the root.

The Newick Format was originally designed for rooted trees. The description of the rooted tree starts at the tips and recursively proceeds through the internal nodes until the root. First, the tips are assigned labels. Then, two tips that are connected through a cherry, say X and Y, are chosen and their most recent common ancestor node is labelled by (X : t_X, Y : t_Y), where t_X (resp. t_Y) is the length of the branch ancestral to node X (resp. Y). Note that under the Newick format, (X : t_X, Y : t_Y) and (Y : t_Y, X : t_X) is equivalent. The tips X and Y together with their pendant branches are then deleted, meaning the node labelled with (X : t_X, Y : t_Y) becomes a tip. One then proceeds recursively until the root is reached. In that way, each node is labelled with the Newick format of its descending subtree. The label of the root is the Newick format of the tree[3]. According to this definition, the Newick notation for the tree in Figure 6.5, right, is (((C : 1, D : 1) : 1, A : 2) : 1, B : 3). Note that we may also write e.g. (B : 3 , ((C : 1, D : 1) : 1, A : 2) : 1). In fact we can swap expressions for subtrees to the immediate left and right of each comma, and thus we can write our example tree in 2^3 different equivalent ways.

Unrooted trees can also be described in the Newick format by using an arbitrary internal node as the "root node", this node has three branches attached. Once the "root node" is reached, the three node labels are put together and separated by a comma, again, each ordering of the three node labels is allowed. Using these rules, the Newick Format for the tree in Figure 6.5, left, is ((C : 1, D : 1) : 1, A : 2, B : 4).

6.2.3 Counting trees

A tree reconstruction method may test all possible trees to find the best tree fitting the data. To do so, we need to know how many different rooted (or unrooted) trees with n tips exist. We will start by counting the number of branches in a tree, which will facilitate counting the number of trees.

6.2.3.1 Counting branches

We can count the number of branches in an unrooted labelled tree with n tips by listing and counting all possible branches in that tree. Listing and counting, also called enumeration, is a handy approach for small trees. However, for large trees, this is computationally very inefficient, and we do not learn general patterns. For example, we cannot conclude if all trees on n tips have the same number of branches, unless we count the branches for every single tree on n tips.

In what follows, we will derive an analytic formula for the number of branches, b_n, of a tree with n tips. Let us start with an example where $n = 2$. The two tips in an unrooted tree can only be connected with a single branch, thus $b_2 = 1$. For $n = 3$ tips, we have $b_3 = 3$ branches (see Fig. 6.6). If we increase the number of tips by one, the number of branches increases by 2. This is because to add a tip we need to break one of the existing branches into two, and add an internal node to which the

[3] If the root has an ancestral branch of length t_{root} attached, we extend the Newick format string by $:t_{root}$.

new tip with a new pendant branch attaches. Thus, for a tree with $n+1$ tips we have $b_{n+1} = b_n + 2$ branches. In general, we have,

$$b_n = b_2 + 2(n-2) = 2n - 3 \tag{6.1}$$

for $n \geq 2$. In a rooted tree, we split one branch from the unrooted tree in two by addition of the root node. This leads to

$$b_n^r = 2n - 3 + 1 = 2n - 2 \tag{6.2}$$

branches in the rooted tree for $n \geq 2$.

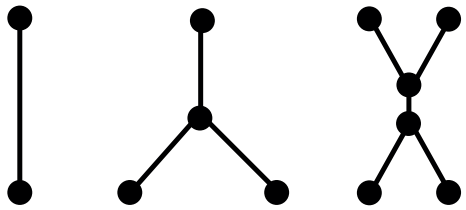


Figure 6.6: Unrooted trees with 2, 3, and 4 tips. We count the number of branches to be $b_2 = 1, b_3 = 3, b_4 = 5$.

Proof by induction

When we derived Equations (6.1) and (6.2), we employed the idea of a *proof by induction*. This technique is very common in mathematics. With this technique, one can prove a formula that depends on an integer $n \in \mathbb{N}$. The proof contains of two steps. The first step establishes the formula for some small $m \in \mathbb{N}$, commonly $m = 1$ or 2. The induction step then assumes that the formula holds for all $k < n$, and proves that, using this assumption, it also holds for n. With these two steps, it is proven that the formula holds for all $n \in \mathbb{N}$ with $n \geq m$. In the following example, we prove equation 6.1 using a proof by induction:

Hypothesis to prove: $b_n = 2n - 3$.

Base step: Check that the hypothesis holds for $n = 2$.
Yes: $b_2 = 1 = 2 \times 2 - 3$.

Induction hypothesis: We suppose the formula holds for all $k < n$.
In particular, $b_{n-1} = 2(n-1) - 3$.

Inductive step: Given the induction hypothesis, show that the formula holds for n.
This is the case because $b_n = b_{n-1} + 2 = 2(n-1) - 3 + 2 = 2n - 3$. The first equality holds, as explained above, adding a tip to a tree adds two new branches. The second equality follows from the induction hypothesis.

Thus, our formula holds for all $n \geq 2$. q.e.d.[4]

[4]Quod erat demonstrandum, meaning "what was to be demonstrated".

6.2.3.2 Counting unrooted labelled trees

We now count the number of unrooted labelled trees τ_n on n tips with labels $l_1, \ldots, l_n$. For $n = 1$ tips in a tree, just one single unrooted tree is possible: $\tau_1 = 1$. For $n = 2$ and $n = 3$, again there is just one tree, so $\tau_2 = \tau_3 = 1$. To obtain any tree with $n = 4$ tips, we start with a tree with $n = 3$ tips with labels $l_1, \ldots, l_3$, and attach a new tip with label l_4 to any of the existing 3 branches (see Fig. 6.7). Thus, $\tau_4 = 3$. Counting the number of trees with 5 tips, we get $\tau_5 = 15$. We now hypothesize that $\tau_n = 1 \times 3 \times 5 \times \ldots (2n-5)$. To simplify notation, we introduce the double-factorial as,

$$m!! := \begin{cases} 1 \cdot 3 \cdot 5 \cdot \ldots \cdot (m-2) \cdot m & \text{for uneven } m \in \mathbb{N}, \\ 2 \cdot 4 \cdot 6 \cdot \ldots \cdot (m-2) \cdot m & \text{for even } m \in \mathbb{N}, \end{cases}$$

meaning our hypothesis is $\tau_n = (2n-5)!!$. We prove this hypothesis by induction.

Hypothesis to prove: $\tau_n = (2n-5)!!$ for $n \geq 3$.

Base step: Check that the hypothesis holds for $n = 3$.
Yes, there is only one labelled tree on 3 tips, thus $\tau_3 = 1$.

Induction hypothesis: We suppose the formula holds for all $k < n$.
In particular, $\tau_{n-1} = (2(n-1)-5)!!$.

Inductive step: Given the induction hypothesis, show that the formula holds for n.
Generally, note that each tree on n tips with labels $l_1, \ldots, l_n$ can be viewed as a subtree on $n-1$ tips with labels $l_1, \ldots, l_{n-1}$ plus the nth tip with label l_n being attached to a branch of the $(n-1)$-tip tree. Importantly, starting with different $(n-1)$-tip trees or attaching the nth tip to different branches yields different trees on n tips: Trivially, if we start with two different trees on $n-1$ tips and attach the nth tip, we will always obtain different n-tip trees as the subtrees on $n-1$ tips are different. Second, if we start with the same $(n-1)$-tip subtree, but attach the nth tip to different branches, we will get different trees as l_n will cluster with different tip labels. Thus, the number of n-tip trees τ_n is the number of $(n-1)$-tip trees times b_{n-1}, $\tau_n = \tau_{n-1} \times b_{n-1}$. Using the Induction hypothesis together with the formula for b_n, we get $\tau_n = (2(n-1)-5)!! \times (2(n-1)-3) = (2n-5)!!$

Thus, our formula holds for all $n \geq 3$. q.e.d.

The number of labelled trees on n tips increases as a double-factorially with n, which roughly corresponds to exponential growth with $n \ln n$. Indeed, accoring to *Stirling's approximation*, for large n, we have $n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n = \sqrt{2\pi n} e^{n(\ln n - 1)}$. Further, $(2n-5)!! = (2n-5)(2n-6)!! = (2n-5)2^{n-3}(n-3)!$, showing that the double factorial grows exponentially with $n \ln n$. Using the Landau notation (Box 9), we say that the number of trees τ_n is on the order $\mathcal{O}(e^{n \ln n})$. In Table 6.1, we provide numerical examples for this growth.

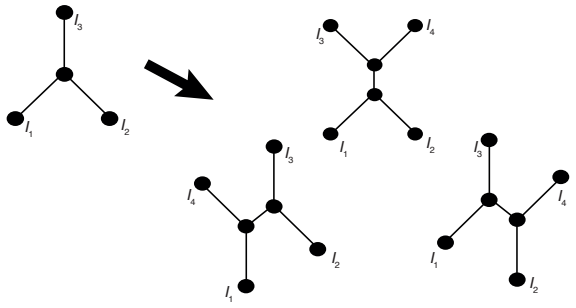


Figure 6.7: How to proceed from an unrooted labelled tree with 3 tips to all three possible trees with 4 tips, with $l_1 = A, l_2 = B, l_3 = C, l_4 = D$.

# of tips	# of unrooted trees
4	3
5	15
6	105
7	945
8	10395
9	135135
10	2027025
11	34459425
⋮	⋮
20	221643095476699771875
⋮	⋮
50	10^{74}

Table 6.1: Number of unrooted trees on n tips.

6.2.3.3 Counting rooted labelled trees

To count the number of rooted labelled trees on n tips we note that each rooted labelled tree can be obtained from an unrooted labelled tree on n tips in which we choose one branch to be subdivided such that a root node is added. Importantly, choosing different unrooted labelled trees or different branches yields different rooted labelled trees. Thus, the number of rooted trees on n tips, τ_n^r, is:

$$\tau_n^r = \tau_n \times b_n = (2n-5)!!(2n-3) = (2n-3)!! \tag{6.3}$$

which again is on the order $\mathcal{O}(e^{n \ln n})$.

Due to this large number of trees, finding the "best" tree from among all possible trees for a given dataset becomes very slow for large n. We therefore need to find smart ways to retrieve the "best" tree in reasonable time. The next section discusses the common approaches for finding the best tree.

6.3 Inferring phylogenies

In the early days of phylogenetics, observable phenotypic characteristics, i.e. morphological traits, of the species were used to reconstruct phylogenetic trees. Species with "similar" morphological traits (e.g. the ability to fly) were thought to cluster together in the species tree, i.e. be more related, than species with very different morphological traits. However, this can lead to wrong phylogenies, for example if *convergent evolution* of the phenotype occurred (i.e. evolution of the same morphological trait from different ancestors). A prominent example is flight. Not all animals that can fly are birds; there are some mammals and insects that can fly too, but are not closely related to birds and evolved flight independent from birds.

Nowadays, genetic sequencing data largely replaces the morphological trait data when inferring phylogenies (note though that for fossils we still rely on morphology as we do not have genetic sequences, for an example see Figure 10.14). Using genetic sequencing data over morphological trait data has a number of advantages:

- First, using genetic sequences is objective in the way that we use each nucleotide for analysis. On the other hand, using morphological traits relies on a subjective trait choice and its measurements.
- Second, while molecular evolution models used for phylogenetic reconstruction still make a number of simplifying assumptions, the modelling occurs where evolution happens, namely at the nucleotide, codon and amino acid level, and each site contributes towards informing the tree reconstruction. We can improve these probabilistic models of molecular evolution to better match reality (account for selection, dependence across sites, etc.), or, we can use third codon position data with simple non-selection models as these positions may be assumed to evolve close to neutral (i.e. without selection) as such mutations rarely change the codon. In contrast, morphology is where selection acts and where the consequences of molecular evolution become visible. Appropriate modelling of these selective processes and weighing the importance of the different characters when reconstructing phylogenies is far from trivial.
- Another practical advantage of using sequences is that with the new high throughput sequencing technologies, this data is much easier and cheaper to obtain compared to morphological trait data. For the latter, palaeontologists have to go on field trips for digging out fossils and then take appropriate measurements. Thus, obtaining morphological data is the opposite of high-throughput.
- Finally, when using genetic sequences, we can go beyond species and analyse e.g. pathogens or other individuals for which recording morphological traits is very hard.

Tree reconstruction methods generally take an alignment of homologous genetic sequences as an input. Alignment procedures have been described in detail in chapter 3. The rational when reconstructing phylogenies from alignments is to put "similar"

sequence 1, s_1: TCACACCT
sequence 2, s_2: ACAGACTT
sequence 3, s_3: AAAGACTT
sequence 4, s_4: ACACACCC

Table 6.2: Toy alignment on which the tree reconstruction methods are illustrated.

sequences close in the tree (accounting for little evolution having occurred), and to place distant sequences very far apart in the tree (accounting for a lot of evolution having occurred). The word "similar" is put in quotation marks because there are several approaches for defining "similar" when reconstructing phylogenetic trees. In this book, we will introduce the three main approaches. We will discuss tree-reconstruction methods designed for each approach, as well as the advantages and disadvantages of each approach:

1. **Phenetic approaches** (distance-based methods) infer the tree based on pairwise similarity of genetic sequences.
2. **Cladistic approaches** (parsimony methods) group organisms based on how many shared characteristics they have.
3. **Probabilistic approaches** (maximum likelihood and Bayesian methods) assume an explicit probabilistic model of evolution for the underlying data and group the organisms based on the likelihood.

We will illustrate the ideas behind the different tree reconstruction approaches and methods using the alignment in Table 6.2 for sequences taken from four individuals.

6.3.1 Phenetic approach: Distance-based methods

The idea of distance-based methods is to cluster sequences which are most similar to each other. Similarity is measured by the pairwise sequence distance. As described in Chapter 5, we have the choice between different pairwise distance measure, such as the Hamming distance, JC69 distance or HKY distance (see table 5.2). When reconstructing a tree, first, the distance between each pair of sequences is calculated. Then, the phylogeny is reconstructed such that a pair of sequences with a small distance is close to each other in the tree. Distances can also be defined and calculated based on morphological characters, and the distance-based methods can be applied to such distance measures equivalently. General drawback of distance-based methods is that they only use pairwise sequence distances, and no higher order correlations (shared by more than two sequences in the sample) are considered.

Here, we demonstrate the phenetic tree reconstruction using the Hamming distance measure applied to the sequence alignment shown in Table 6.2. The Hamming distance between each pair of sequences is put into a matrix called the *distance matrix*,

The matrix with all pairwise distances is symmetric, i.e. the distance between s_1 and s_2 is the same as the distance between s_2 and s_1, therefore we only report the upper triangle in the distance matrix. Further, note that the distance of a sequence

H	s_1	s_2	s_3	s_4
s_1	-	3	4	2
s_2		-	1	3
s_3			-	4
s_4				-

Table 6.3: Hamming distance matrix for the alignment in Table 6.2.

to itself, say s_1 to s_1, is 0, however, in the distance matrix, we put a $-$ on the diagonal. The "H" in the upper left corner means that all pairwise distances in that matrix were calculated using the Hamming distance measure. We denote $d(s_i, s_j)$ as the distance between sequence s_i and s_j. Based on our illustrative distance matrix, we have $d(s_1, s_2) = 3$.

A distance between two sequences is interpreted as an estimate for the amount of evolution that happened between them. Under the Hamming distance measure, two identical sequences have distance 0, suggesting that no evolution happened between them. Of course, even if two sequences are identical, sequences at intermediate steps could have been different (e.g. because back-substitutions occurred). Thus, the Hamming measure of sequence distance may be too simplistic, and more complex distance measures may be more accurate. Evolutionary models such as the Jukes-Cantor model [**Jukes1969**] take substitutions and therefore also back-substitutions into account, yielding a non-zero distance for identical sequences. As derived in Section 5.3.2, the pairwise distance formula under JC69 is $\hat{d} = -\frac{3}{4}\log(1 - \frac{4}{3}\hat{p})$ where $\hat{p} = x/n$ and x is the number of differences between two sequences (i.e. the Hamming distance), and n is the sequence length. The distance matrix under the JC69 model for the toy alignment in Table 6.2 is:

JC	s_1	s_2	s_3	s_4
s_1	$-$	0.52	0.82	0.30
s_2		$-$	0.14	0.52
s_3			$-$	0.82
s_4				$-$

There are two different classes of distance-based methods. The first class of method is referred to as *algorithmic* methods. These methods cluster sequences that are separated by a small distance according to the distance matrix in a greedy way, meaning in each step they cluster according to the best choice in that local step, with the idea that many local best choices should lead to a global good choice. In particular, the smallest pairwise distances are picked sequentially and clustered in the tree. These methods are very fast and often used if the distance matrices are large. Examples of algorithmic methods are UPGMA [**SokalMichener1958**] or neighbour-joining algorithms [**SaitouNei1987**]. The second class of methods is referred to as *optimality* methods. These methods minimise the difference of the distance matrix to the inferred *tree distance matrix* [**Fitch1967**, **cavalli1967**]. The

distance of the tree distance matrix are the sum of branch lengths on the paths between each pair of tips. These methods can be very slow as they typically have to consider all possible trees in order to find the tree that minimises the difference. We will now introduce the two classes of methods in more detail.

6.3.1.1 Algorithmic approach: UPGMA method

A simple distance-based (algorithmic) method is the UPGMA (Unweighted Pair Group Method using Arithmetic means) algorithm [**SokalMichener1958**]. The UPGMA algorithm constructs ultrametric trees, meaning it is only suited for data sampled at one point in time. Since distances are typically obtained from genetic sequence data, UPGMA makes the implicit assumption that the genetic data evolved according to a *strict molecular clock*, i.e. substitution rates were the same for all branches in the tree at all times. For many datasets a strict molecular clock cannot be assumed, and the neighbour-joining algorithm which allows for rate variation through time is used in such cases. Since the UPGMA method provides an easy and intuitive understanding of the way algorithmic distance-based methods work, we will present this method below. In what follows, we provide a step-by-step description of the UPGMA algorithm. Such a description can serve as a blueprint to implement

the algorithm in code, and is also referred to as *pseudocode*.

Input: Distance matrix for n sequences. We refer to each sequence $s_1, \ldots, s_n$ as a node.
Output: A rooted ultrametric phylogenetic tree.

begin
  Initialize the size of each node s_i as $n_i = 1$.
  Initialize the tree as the set of unconnected nodes $s_i, i = 1, \ldots, n$.
  while *the distance matrix is not empty* **do**
    Choose nodes s_i and s_j such that $d(s_i, s_j)$ is the smallest entry in the distance matrix (in case of several minima choose one uniformly at random).
    Coalesce nodes s_i and s_j of the current tree to form a new node $s_{i,j}$, with size $n_{i,j} = n_i + n_j$. The branch length between $s_i, s_{i,j}$ and between $s_j, s_{i,j}$ is chosen such that all tips descending from $s_{i,j}$ have the same distance $d(s_i, s_j)/2$ to $s_{i,j}$.
    if *the distance matrix includes only 2 nodes* **then**
      return *the tree with the root $s_{i,j}$ as the result and finish.*
    end
    Include node $s_{i,j}$ into the distance matrix, with $d(s_m, s_{i,j}) = \frac{n_i d(s_i, s_m) + n_j d(s_j, s_m)}{n_i + n_j}$ where s_m is a node in the distance matrix.
    Delete nodes s_i and s_j from the distance matrix.
  end
end

Algorithm 1: The UPGMA algorithm

Example of UPGMA with the Hamming distance matrix

Here we show how the UPGMA algorithm works for our example Hamming distance matrix, shown in Table 6.3.

Iteration 1

In step 1 of the UPGMA algorithm, we look up the minimal pairwise distance in the Hamming distance matrix. In our example distance matrix, the minimal distance is $d(s_2, s_3) = 1$, the distance between nodes s_2 and s_3. In step 2 of the UPGMA algorithm, we coalesce these two nodes and introduce a new node $s_{2,3}$, meaning the current tree is a cherry plus the unconnected nodes as shown in Figure 6.8. The distance from the new node $s_{2,3}$ to the tips s_2 and s_3 is $d(s_2, s_3)/2 = 0.5$. We further set $n_{2,3} = 2$. In step 3, the distances from the new node to the remaining nodes are calculated. For example, to calculate the distance from the new node $s_{2,3}$ to the node s_1, we calculate $d(s_1, s_{2,3}) = \frac{n_2 d(s_2, s_1) + n_3 d(s_3, s_1)}{n_2 + n_3} = \frac{1 \cdot 3 + 1 \cdot 4}{1+1} = \frac{7}{2} = 3.5$. The distances between all remaining nodes (i.e. in this case the distance between s_1 and s_4) remain

the same. The new distance matrix obtained from step 4 is shown in Table 6.4.

Figure 6.8: Intermediate UPGMA tree after 1 iteration.

	s_1	s_4	$s_{2,3}$
s_1	-	2	3.5
s_4		-	3.5
$s_{2,3}$			-

Table 6.4: Intermediate distance matrix after 1 iteration.

iteration 2

The minimal distance between a pair of nodes in the new matrix is between s_1 and s_4, $d(s_1, s_4) = 2$. We create a new node $s_{1,4}$ and set $n_{1,4} = 2$. The intermediate UPGMA tree is shown in Figure 6.9 and the new distance matrix is shown in Table 6.5.

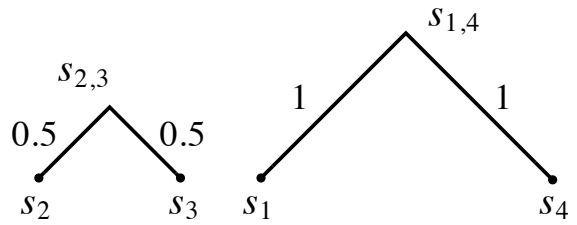


Figure 6.9: Intermediate UPGMA tree after 2 iterations.

	$s_{2,3}$	$s_{1,4}$
$s_{2,3}$	-	3.5
$s_{1,4}$		-

Table 6.5: Intermediate distance matrix after 2 iterations.

iteration 3

Now, the minimal distance is between $s_{2,3}$ and $s_{1,4}$, $d(s_{2,3}, s_{1,4}) = 3.5$. In step 2 of the algorithm we create the new node $s_{2,3,1,4}$, which is the root of the tree, with distance $\frac{3.5}{2} = 1.75$ to all tips. As only two nodes $s_{2,3}, s_{1,4}$ are in the distance matrix, nothing is done in step 3, and the distance matrix becomes empty in step 4. Therefore, the algorithm terminates, outputting the tree shown in Figure 6.10.

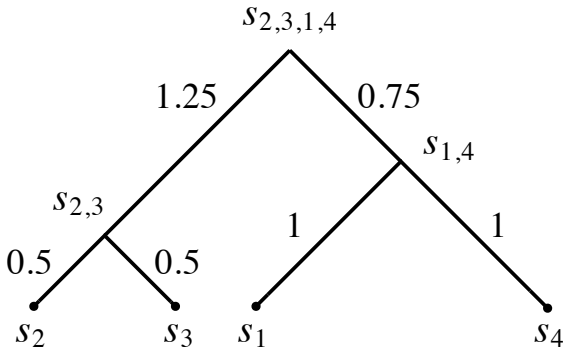


Figure 6.10: Final UPGMA tree.

Properties of the UPGMA trees

The UPGMA tree distance matrix (i.e. the distance matrix being composed of the sum of branch lengths on the path between each pair of tips in the UPGMA tree) is shown in Table 6.6.

	s_1	s_2	s_3	s_4
s_1	-	3.5	3.5	2
s_2		-	1	3.5
s_3			-	3.5
s_4				-

Table 6.6: Tree distance matrix for the reconstructed UPGMA tree.

This tree distance matrix differs slightly from the original sequence distance matrix (Table 6.3). The reason is that pairwise distances in the sequence distance matrix do not in general correspond to any ultrametric tree. The UPGMA (and other distance-based methods) construct a tree representing the distances in the matrix as well as possible. The UPGMA works as expected in the sense that if the input distance matrix equals the tree distance matrix of an ultrametric tree, then exactly this tree will be returned by UPGMA. We will prove this property of UPGMA in the following theorem.

Theorem 6.3.1. *Let D_N be a distance matrix of dimension $N \times N$ and assume there exists an ultrametric tree with a tree distance matrix T_N that is equal to D_N. Then, the UPGMA algorithm as defined in section 6.3.1.1 with input matrix D_N will return T_N.*

Proof. We prove this statement using induction as described in section 6.2.3.1.

Hypothesis to prove: If the distance matrix D_N equals the tree distance matrix of an ultrametric tree T_N, the UPGMA algorithm with input $\mathrm{D}_N^{n_1,\ldots,n_N}$, where $n_1, \ldots, n_N$ are arbitrarily chosen integers, returns the tree T_N.

Base step: Let $N = 2$, i.e. we consider an ultrametric tree T_2 with two branches of length $d(s_1, s_2)/2$. The distance matrix D_2 has only one entry $d(s_1, s_2)$, which

is of course also the smallest entry. The UPGMA algorithm (step 2) results in an ultrametric tree with two tips and branch lengths $d(s_1, s_2)/2$, i.e. we obtain the tree T_2. In particular, this does not depend on the values of n_1, n_2. Thus, the hypothesis holds true for $N = 2$.

Induction hypothesis: Suppose the hypothesis holds for all $k < N$.

Inductive step: Given the hypothesis holds for $k < N$, we need to prove the hypothesis for N. According to step 1 of the UPGMA algorithm, we choose the two nodes s_i and s_j in D_N whose entry $d(s_i, s_j)$ in the distance matrix is the smallest. This node is coalesced to a cherry C with branch lengths $d(s_i, s_j)/2$. Since D_N is the tree distance matrix for T_N, exactly this cherry also appears in T_N. Further, since D_N is a tree distance matrix for an ultrametric tree, we know that

$$d(s_i, s_m) = d(s_j, s_m) \tag{6.4}$$

for all nodes s_m, $m \neq i, j$. According to step 3 in the UPGMA algorithm, the new node $s_{i,j}$ has the following distance to the remaining nodes s_m:

$$d(s_m, s_{i,j}) = \frac{n_i d(s_i, s_m) + n_j d(s_j, s_m)}{n_i + n_j} \overset{(6.4)}{=} \frac{(n_i + n_j) d(s_i, s_m)}{n_i + n_j} = d(s_i, s_m)$$

Thus, the new matrix D_{N-1} is a distance matrix for the ultrametric tree T_N without leave j, T_{N-1}. According to the induction hypothesis, UPGMA with input D_{N-1} returns T_{N-1}. Thus, the UPGMA on D_N returns the tree T_{N-1} with tip i being replaced by cherry C, thus it returns T_N.

Thus, the hypothesis holds for all $N \geq 2$. □

The UPGMA is appropriate if all sequences are sampled at the same point in time, and the sequences evolve according to a strict molecular clock, i.e. the substitution rates are the same at all times. Then the genetic distances in the distance matrix map onto an ultrametric tree where branch lengths are in fact proportional to calendar time. If the assumption of a strict molecular clock is violated or sequences are sampled at different time points, then methods inferring non-ultrametric trees are required. An algorithmic distance-based method reconstructing non-ultrametric trees is the neighbour-joining algorithm [**SaitouNei1987**]. The main idea of such algorithms is similar to UPGMA: iteratively, pairs of nodes (corresponding to sequences or ancestor nodes obtained by joining nodes) are joined together to obtain a tree. However, the essential difference is that such methods infer unrooted trees with branch lengths in units of number of substitutions but no proportionality to calendar time.

Runtime and statistical consistency of the UPGMA method

The advantage of algorithmic distance-based methods is their speed. For reconstructing a tree with n tips, we need to perform $n - 1$ iterations of the algorithm (in each iteration, two nodes coalesce into one node). Within each iteration, a distance

matrix has to be set up and searched, which is on the order of $\mathcal{O}(n^2)$. Overall, this means that we need to perform on the order of $\mathcal{O}(n^3)$ calculations to obtain an algorithmic distance-based tree (in fact, an even faster running time can be achieved **??**). The phylogenetic reconstruction methods presented in later sections will be much slower than the polynomial-time algorithmic distance-based methods. The methods presented in this chapter have at least exponential running time, $\mathcal{O}(e^n)$, since all trees on n tips need to be checked regarding optimality. For many tens of thousands of sequences, only algorithmic distance-based methods will be fast enough to infer a tree in reasonable time.

Furthermore, UPGMA is statistically consistent (see Box 21) for ultrametric trees (e.g. [**Felsenstein2004**, **gascuel2004**]).

Box 21: Statistical consistency of phylogenetic inference tools

A phylogenetic reconstruction method is *statistically consistent* if the true tree is returned when the method is presented with an infinite amount of data, i.e. infinitely long sequences. Put more formally, a method is statistically consistent, if for all $\varepsilon > 0$, we have,

$$\lim_{n\to\infty} P(||\hat{T}_n - T|| < \varepsilon) = 1 \tag{6.5}$$

where n is the sequence length, T is the true tree, and $\hat{T}$ is the inferred tree. $||\cdot||$ is a metric measuring how different the inferred tree is from the true tree. This means that in case our tree inference method is statistically consistent, the probability that we obtain a tree which differs less than ε from the true tree tends to 1 when the input sequence length tends to infinity.

6.3.1.2 Optimality approach: a least squares method

The least squares method [**Fitch1967**, **cavalli1967**] search for a tree that minimizes the squared difference between the sequence distance matrix and the tree distance matrix. In other words, this algorithm minimizes,

$$S = \sum_{i=1}^{n} \sum_{j=i+1}^{n} w_{i,j}(D_{i,j} - d_{i,j})^2$$

where D is the sequence distance matrix, d is the tree distance matrix for the proposed tree, and $w_{i,j}$ are the weights ($w_{i,j}$ may be e.g. 1 or inversely proportional to $D_{i,j}$).

Runtime and statistical consistency of the least square method

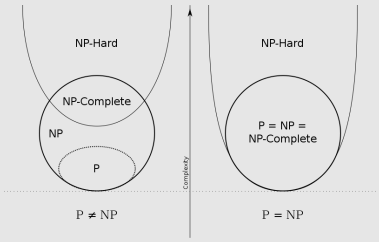
Least squares algorithms need to search the whole space of trees to find the tree that minimises the criterion meaning the running time is $\mathcal{O}(e^n)$. Furthermore, it is an *NP-hard* problem [**day1987computational**], see Box 22.

Least square methods are statistically consistent. For a proof, consider a fixed tree with tree branch lengths measured in units of substitutions, i.e. within one time unit, one substitution is expected to happen. Let sequences evolve on this tree under some model M. Let maximum likelihood distances be calculated using the model M. A maximum likelihood estimator is statistically consistent. Here, that means with increasing sequence length, the sequence induced distance matrix approaches the tree induced distance matrix, $\lim_{n\to\infty} P(||\hat{D}_n - D|| < \varepsilon) = 1$.

The mapping between a tree T and its distance matrix D is a bijection. We define the tree metric as $||\hat{T}_n - T|| := ||\hat{D}_n - D||_2 := \sum_{i=1}^{n}\sum_{j=i+1}^{n}(\hat{D}_{i,j} - D_{i,j})^2$, with the latter being the function to be minimized in the least square method. Then, $\lim_{n\to\infty} P(||\hat{T}_n - T|| < \epsilon) = \lim_{n\to\infty} P(||\hat{D}_n - D||_2 < \varepsilon) = 1$. This establishes that the least square method is statistically consistent.

Box 22: NP-completeness and NP-hardness

In computational complexity theory, P stands for polynomial time. The set of decision problems[a] P is the set of decision problems that can be solved in a polynomial amount of computation time. This means that the number of computations (time) required to solve a problem in P with input size n is on the order of $\mathcal{O}(n^k)$, where k is some fixed input-independent number (e.g. $k = 3$ for UPGMA).
NP stands for nondeterministic polynomial time. The set NP is the set of decision problems for which it is possible to verify in polynomial time whether a particular proposal is a solution. By definition, $P \subseteq NP$ (P is a subset of NP), but whether $P = NP$, i.e. whether decision problems for which a solution can be verified in polynomial time can also be solved in polynomial time, is currently unknown and one of the major conundrums in Computer Science. In the least square method above, it is easy to determine if for a given tree the least square difference of its distance matrix to the original sequence distance matrix is less than some threshold x. Thus, this decision problem is in NP.
Currently, many NP problems have no known algorithm for finding solutions in polynomial time, for instance the decision problem "Is there a tree inducing a distance matrix with a least square difference to the given sequence distance matrix of less than x?"
A decision problem X is *NP-complete* if an algorithm solving X could also be used to solve all other problems in NP, potentially using a polynomial time transformation of the algorithm. The NP-complete problems are thus the hardest problems within the NP-class of problems. As a consequence of the definition of NP-completeness, if one NP-complete problem can be solved in polynomial time, then all problems in NP can be solved in polynomial time (and thus $P = NP$). Proving or disproving $P = NP$ is one of the 7 Millennium Prize Problems announced in 2000 [**jaffe2006millennium**]. The first person solving it will be awarded 1'000'000 US Dollars. A Venn diagram can nicely display the connections between P, NP, and NP-complete:



A popular example for an NP-complete problem is the *travelling salesman problem*. The traveling salesman problem considers k cities (e.g. capitals of Europe) that a salesman has to visit. In the travelling salesman problem, we want to know whether the salesman can visit all the cities on a path shorter than length L.
A problem H is *NP-hard* if an algorithm solving it can also solve a NP-complete problem X, possibly using a polynomial time transformation to adopt the algorithm for H such that it solves X. Thus, any NP-complete problem is also NP-hard. However, an NP-hard problem does not need to be in the class NP. In particular, a solution may not be verifiable in polynomial time, or the problem may not be a decision problem. In fact, an NP-hard problem may be an optimization problem. An example is the optimization version of the travelling salesman problem, in which we want to know the shortest path for a salesman to visit all k cities. Given a decision problem is NP-complete, then the corresponding optimization problem is NP-hard, as we can answer the decision problem (i.e. if a solution $\leq L$ exists) using an algorithm solving the optimization problem.

[a] A decision problem is a question that can be posed as a yes-no question, dependent on the input values.

6.3.2 Cladistic approach: Parsimony method

We now introduce the cladistic approach, going back to [**EdwardsCavalli1964**]. It groups sequences based on how many characteristics they share. While phenetic methods group sequences based on pairwise similarity, ignoring correlations of more than 2 sequences at a time, the cladistic method implicitly accounts for an evolutionary process and higher than first order sequence correlations.

Using the cladistic approach in tree inference leads to an unrooted tree, the *maximum parsimony tree*. For a given alignment, the maximum parsimony tree is an unrooted tree on n tips with the lowest *parsimony score* among all unrooted trees on n tips. The parsimony score is defined as the minimal number of changes (e.g. nucleotide substitutions for nucleotide sequence alignments) required to explain the alignment on the tree. Thus the cladistic approach aims to determine the tree requiring the fewest number of changes.

6.3.2.1 Example for calculating the parsimony score

We illustrate the concept of parsimony on our example alignment. First we note that since our alignment contains five polymorphic sites (shown in red below), we require at least five substitutions, i.e. this is the lower bound for the parsimony score of any of the unrooted trees on four tips.

sequence 1: TCACACCT
sequence 2: ACAGACTT
sequence 3: AAAGACTT
sequence 4: ACACACCC

We will show how to calculate the parsimony score for the UPGMA tree (Figure 6.11) calculated earlier. Note that this UPGMA tree is a rooted tree. In fact, the parsimony score is calculated on rooted trees. However, as we explain below, the parsimony score for all rooted trees with the same underlying unrooted tree is the same. Thus, the parsimony method infers unrooted trees.

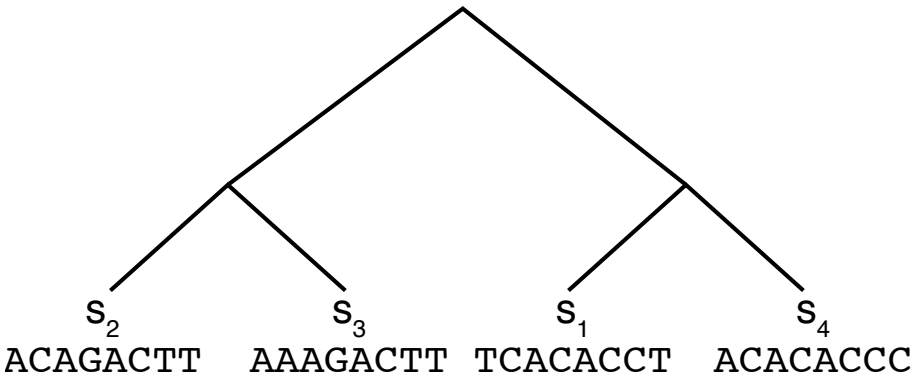


Figure 6.11: The tree induced by the UPGMA algorithm with sequences at the tips.

We start by looking at each site in turn and determine how many substitutions are required to achieve the configuration at the tips. For example, for site 1, the

sequences s_2 and s_3 have an A, so no substitution is required on the branches leading from the tips to their common ancestor. On the contrary, s_1 has a T but s_4 has an A at the first position. Thus, either there was a substitution from T to A on the branch leading to s_4 from its common ancestor with s_1, or there was a substitution from A to T on the branch leading to s_1. To decide, we look at the other sequences. Since both s_2 and s_3 have an A, it is more parsimonious that the sequence s_1 changed rather than that all the other sequences changed. We thus assign an A→T substitution to the branch between the s_1 tip and the common ancestor of s_1 and s_4. The parsimony score for site one is one, since we need one substitution to explain the alignment at the first site. We proceed in the same way for all the remaining sites, assigning and counting substitutions. The sum of the parsimony scores for each site is the parsimony score for the tree. It turns out that the parsimony score of our UPGMA tree is five (Figure 6.12).

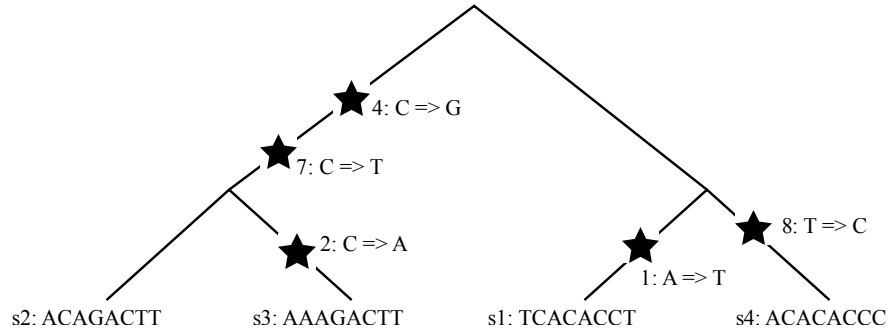


Figure 6.12: The tree induced by the UPGMA algorithm with sequences at the tips and a minimal number of substitutions assigned to the branches.

6.3.2.2 Rooted versus unrooted trees

No matter where we root the tree, we obtain the same parsimony score via the same substitutions. In other words, if we omit the root in the UPGMA tree, and then re-root by choosing another branch in the unrooted tree, the parsimony score stays the same. The reason is that we can use the same substitutions as in the original tree to explain the sequences in the re-rooted tree. In Figure 6.13, we display all possibilities for re-rooting the UPGMA tree. In particular, all these trees have the same unrooted tree.

Our example alignment has five polymorphic sites and thus the maximum parsimony tree has a parsimony score of at least five. Since the UPGMA tree has a parsimony score of five, we know that the corresponding unrooted tree is a maximum parsimony tree. Thus we found a maximum parsimony tree by just looking at one rooted tree (there may be other trees with the same parsimony score though). In general, a maximum parsimony tree has a parsimony score which is greater than the number of polymorphic sites in the alignment. This means that one has to somehow consider the parsimony score of all unrooted trees for determining the maximum parsimony tree. Figure 6.14 shows all unrooted trees on four tips.

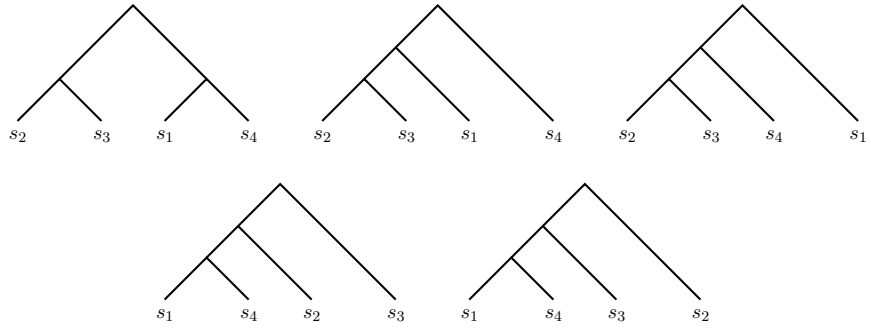


Figure 6.13: The tree induced by the UPGMA algorithm and all possible re-rootings of this tree. In particular, all such rooted trees have the same underlying unrooted tree and thus the same parsimony score.

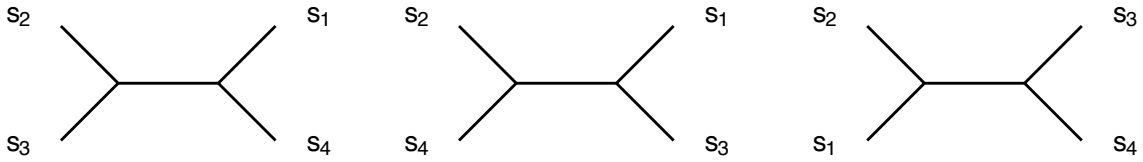


Figure 6.14: The three unrooted trees on four tips.

6.3.2.3 Fitch algorithm

In general, in order to determine the parsimony score for a tree, we can assign all possible ancestral sequences to the internal nodes, and then count the number of substitutions for each assignment of sequences. Overall, for nucleotide sequences of length m, and number of sequences n, we have a tree on n tips and $n-1$ internal nodes, and thus 4^{n-1} possible nucleotide assignments per site. Given a site assignment, for each of the $2n-2$ branches, we need to determine if the nucleotide changed or not, meaning we have to perform $2n-2$ operations. Thus, overall, we need to perform $m4^{n-1}(2n-2)$ operations to obtain the parsimony score (see also Section 6.3.2.1). Considering all these assignments is very slow. Such an approach trying all possibilities is called *brute force*.

For calculating the parsimony score for a given tree, we can instead use a smarter approach involving the concept of dynamic programming. We encountered dynamic programming already in Chapter 3 for pairwise sequence alignment. Recall that the idea behind the dynamic programming approach is to break down the problem into a collection of smaller subproblems, solve the small subproblems first, store the results, and then use them to solve the bigger problem. This strategy is superior to brute force approaches which essentially evaluate the same subproblem multiple times. In a phylogenetic context, dynamic programming translates to recursively solving the subproblem for a subtree, and combine the results on the subtrees to obtain the result for the full tree. A fast dynamic programming algorithm for computing the parsimony score is the Fitch algorithm [**Fitch1971**], the main idea of which is to recursively calculate the parsimony score on subtrees, and then combine the subtree results to obtain the parsimony score for the full tree. Here we outline the Fitch

algorithm in pseudocode form and for nucleotides. However, it works analogously for amino acids or codons.

Input: Unrooted phylogenetic tree and an alignment of n sequences of length m, corresponding to the n tips of the tree.
Output: Parsimony score of the tree, i.e. the minimal number of substitutions required to explain the sequences at the tips.

```
begin
    Root the tree at an arbitrary branch.
    k ← 0
    while the root has no sequence assigned do
        Choose a node in the tree where all the descending nodes (i.e. all
         nodes on a path down to a tip) have sequences assigned.
        Assign a sequence to the chosen node in the following way:
        for i = 1, ..., m do
            Let C_l and C_r be the sets of nucleotides being assigned to the
             two direct descendants of the chosen node for site i.
            if C_l ∩ C_r ≠ ∅ then
                Assign C_l ∩ C_r to site i of the chosen node and keep k
                 unchanged.
            else
                Assign C_l ∪ C_r to site i of the chosen node and set k ← k + 1.
            end
        end
    end
    return k as the parsimony score.
end
```

Algorithm 2: Fitch algorithm for computing the parsimony score of a tree.

We will now prove by induction over the number of tips n in the tree that the Fitch algorithm outputs the parsimony score. We provide the proof for one site, $m = 1$, since the parsimony score for many sites is the sum of parsimony scores for each single site.

Theorem 6.3.2. *For any n and for $m = 1$, the Fitch algorithm outputs the parsimony score S and all optimal root nucleotides. An optimal root nucleotide is defined to be a nucleotide at the root which allows to explain the tip nucleotides with S substitutions.*

Proof. First we perform the base step for $n = 2$. If the nucleotides are different, the parsimony score is one and an optimal root nucleotide is either of the tip nucleotides. If the nucleotides are the same, the parsimony score is zero and the optimal root nucleotide equals the tip nucleotide. The Fitch algorithm returns precisely these values (base step).

Next, we suppose that the Fitch algorithm returns the parsimony score and all optimal root nucleotides for all trees with k tips, $k < n$ (induction hypothesis).

In the inductive step, we now show that the Fitch algorithm returns the parsimony score and all optimal root nucleotides for all trees with n tips. We split the rooted tree on n tips into two rooted subtrees 1 and 2 by deleting the root and the two adjacent branches. By applying the induction hypothesis, we obtain the parsimony score S_1 and S_2, and the optimal root nucleotides using the Fitch algorithm for the two subtrees. The parsimony score of the n-tip tree is at least $S_1 + S_2$.

1. case: The intersection of the optimal root nucleotides for both subtrees is non-empty. This means that any nucleotide of the intersection can serve as a root nucleotide which leads to $S_1 + S_2$ changes. Thus, the parsimony score attains the lower bound $S_1 + S_2$, and all nucleotides in the intersection belong to the set of optimal nucleotides. Next we show that a nucleotide X not being in the intersection cannot be an optimal root nucleotide: (i) Suppose that both subtrees are assigned one of their optimal root nucleotides. Then X has to change on at least one branch adjacent to the root and the number of substitutions is at least $S_1 + S_2 + 1$. (ii) Suppose w.l.o.g. that subtree 1 has a non-optimal root nucleotide assigned. Then the number of substitutions required in that subtree is $S_1 + 1$ and thus the overall number of substitutions is at least $S_1 + S_2 + 1$ which is bigger than the parsimony score of $S_1 + S_2$.

2. case: The intersection of the optimal root nucleotides for both subtrees is empty. We show that in this case, the parsimony score is $S_1 + S_2 + 1$ and any nucleotide of the union may be an optimal root nucleotide. If the two subtrees have each assigned one of their optimal root nucleotides, then the subtrees contribute $S_1 + S_2$ substitutions, and since the optimal root nucleotides in both subtrees are all different, the joining of the two subtrees – with the root nucleotide being a nucleotide from the union – requires an additional substitution, leading to $S_1 + S_2 + 1$ substitutions. If w.l.o.g. subtree 1 is assigned a non-optimal root nucleotide X, then it contributes $S_1 + 1$ substitutions. We obtain an additional S_2 substitutions from subtree 2. No further substitution is needed if X is in the union. If both subtrees are assigned non-optimal root nucleotides, they contribute $S_1 + S_2 + 2$ substitutions (and thus require more substitutions than the two cases above). Thus, $S_1 + S_2 + 1$ is the parsimony score.

It remains to be shown that the union of the optimal subtree root nucleotides is the set of all optimal root nucleotides. Suppose nucleotide X is not in the union. (i) Suppose that both subtrees are assigned one of their optimal root nucleotides. Then X has to change on both branches adjacent to the root and the number of substitutions is $S_1 + S_2 + 2$. (ii) Suppose w.l.o.g. that subtree 1 has a non-optimal root nucleotide assigned. Then the number of substitutions required in that subtree is at least $S_1 + 1$. Now consider the two cases, (a) subtree 2 has an optimal nucleotide assigned, which requires 1 substitution from X to that optimal nucleotide plus S_2 substitutions within subtree 2, leading to at least $S_1 + S_2 + 2$ substitutions. In case (b), subtree 2 has a non-optimal nucleotide assigned, leading to at least $S_2 + 1$ substitutions in subtree 2, and overall leading to at least $S_1 + S_2 + 2$ substitutions.

Thus X is not an optimal root nucleotides. □

The Fitch algorithm is very fast as it traverses each of the $n-1$ internal nodes once, assigning a state to each site at each node, resulting in a running time of the order $\mathcal{O}(nm)$ where n is the number of sequences and m is the sequence length (in our example $n = 4, m = 8$). Thus, calculating the parsimony score on a given phylogeny is done in polynomial time.

6.3.2.4 Example of using the Fitch algorithm

We continue with our example alignment from above. We consider the three possible unrooted trees on four tips with an alignment of four sequences assigned, see Figure 6.14. We now assign sequences to internal nodes following the Fitch algorithm (Figure 6.15). We highlight that a set of nucleotides may be assigned to a particular node at each site (Step 3.b in the algorithm). For example, we write $\{A, G\}$ if both A and G are assigned. For the right and middle tree, seven substitutions are required to explain the sequences at the tips; for the left tree only five substitutions are required, which means that the left tree is the only maximum parsimony tree.

6.3.2.5 Runtime of the parsimony method

The Fitch algorithm calculates the parsimony score for a single tree, the input tree. However, we have to calculate the parsimony score for each possible tree on n tips. For $n = 4$ this is easy, as only three trees have to be considered, but in general the number of trees (see Section 6.2.3.2) – and as a consequence the running time – increase drastically with n. One may thus ask if there is a fast way avoiding to consider each unrooted tree. The answer is no, unless $P = NP$: One can show that finding the maximum parsimony tree is an NP-hard problem [**FouldsGraham1982-ParsNP**], which means that we essentially have to calculate the parsimony score for all possible trees on n tips. Afterwards the tree(s) with the lowest parsimony score is (are) the maximum parsimony tree(s).

6.3.2.6 Statistical inconsistency of the parsimony method

The main problems of the parsimony method is that it does not acknowledge possible back-substitutions, i.e. a branch with apparently no substitution may have had two substitutions, e.g. from $A \to G \to A$, but parsimony assumes no substitution at all in such a case. As a result, the parsimony method suffers from a bias called *long branch attraction*. Long branch attraction is a phenomenon that arises when two very long branches are connected through a relatively short internal branch as seen in Figure 6.16. Let the probability of a change on the long branches (leading to b and d) be p, and on the short branch be q. Now, for separating b and d via the short internal branch, we must observe a substitution on that short internal branch. This has probability q. On the other hand, seeing the same change on the long branches has probability p^2. Roughly speaking, if the probability for the same change on the long branches is bigger than the chance for a substitution

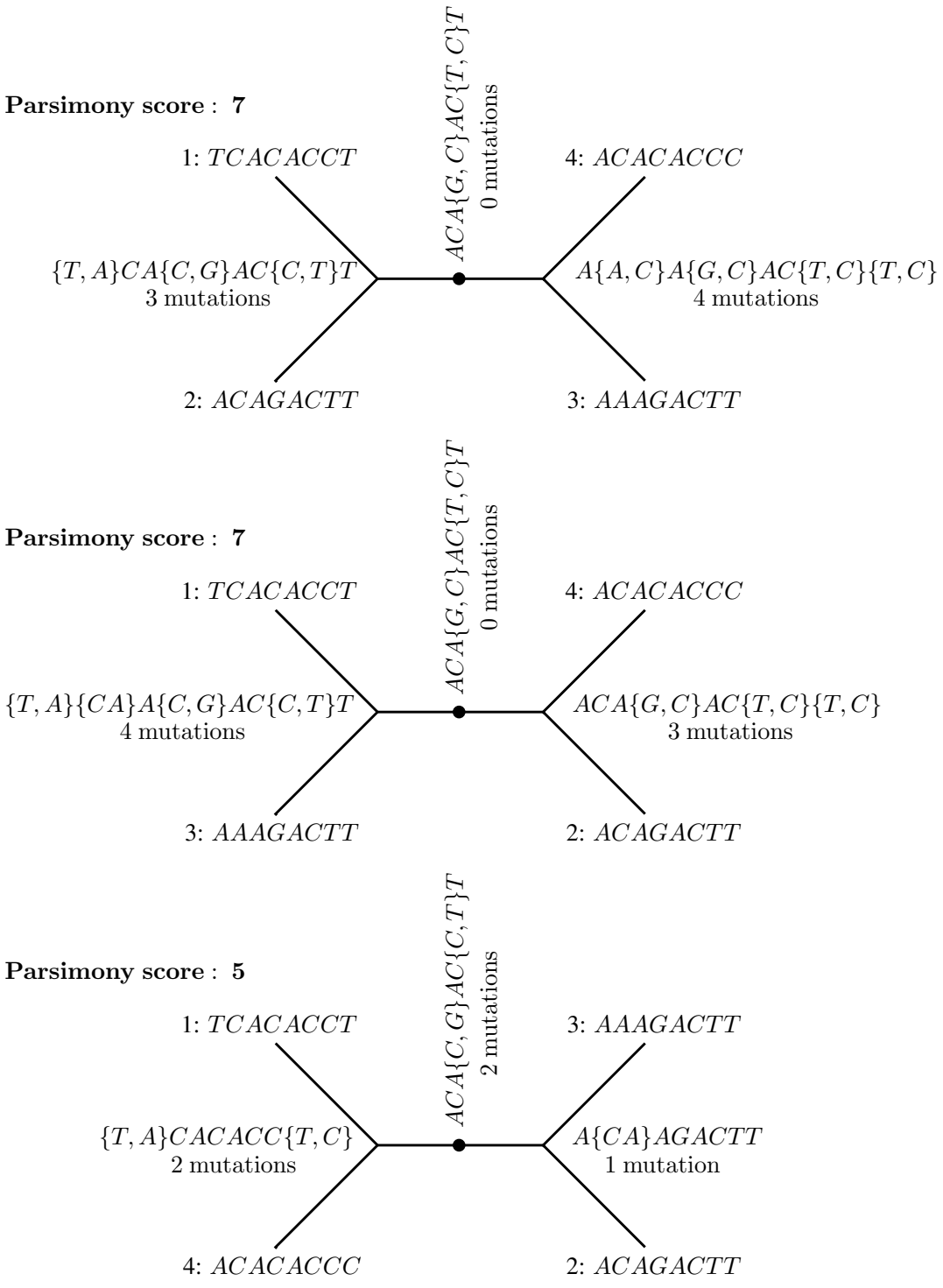


Figure 6.15: Parsimony scores for the three trees on four tips. The dot on the internal branch is the artificially added root (first step in the Fitch algorithm).

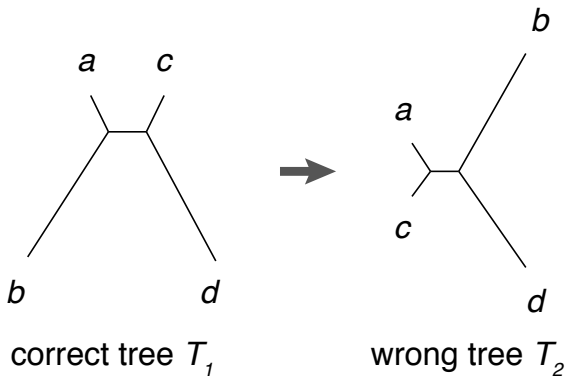


Figure 6.16: When the correct tree (T_1) has two long branches separated by a short internal branch, parsimony tends to reconstruct a wrong tree (T_2) with the two long branches grouped together. This phenomenon is called long-branch attraction. Figure adapted from [**Yang2014**].

on the short branch, parsimony favours a tree with b and d as a cherry. With the number of sites going to infinity, parsimony will thus for sure pick the tree where b and d form a cherry. As a result, parsimony methods are statistically inconsistent [**felsenstein1978parsimonyStatIncons**]. Formally, the condition for parsimony being statistically inconsistent in our example is $p^2 > q - q^2$. This is shown rigorously in [**felsenstein1978parsimonyStatIncons**] and is nicely discussed in [**Felsenstein2004**]. Due to the statistical inconsistency of parsimony, it is rarely being used nowadays. However, the cladistic community is still supportive of parsimony.

6.3.3 Probabilistic approach: Maximum likelihood methods

The probabilistic approach (going back to [**EdwardsCavalli1964**]) assumes an explicit probabilistic model of evolution underlying the data. It evaluates the likelihood of the model parameters θ given the data D, $L(\theta; D) := P(D|\theta)$ (see also Box 18 for the likelihood function), and then compares the likelihood for different parameters. In the maximum likelihood (ML) approach, the parameters maximizing the likelihood function are estimated, and reported as maximum likelihood parameter estimates. In a Bayesian approach, the posterior distribution of parameters, which is a function of the likelihood, is estimated (see Chapter 10 for details).

In a phylogenetic context, the probabilistic model consists of two components. The first component of the model is the tree with branch lengths, $\mathcal{T}$, which describes how the organism replicated. The tree with branch length may be a parameter, or may be generated under some probabilistic tree generating model (see Chapter 9 for tree generating models). In this section, we assume that the tree is a parameter of the model. The second component is a model of sequence evolution, i.e. how nucleotides (or codons, or amino acids) change over time, with rate matrix Q. Commonly used models are e.g. JC69, HKY and GTR (see Chapter 5 for details on substitution models). The probabilistic model gives rise to a probability of the data, i.e. the

sequence alignment D, given the tree and rate matrix, $P(D|\mathcal{T}, Q)$. The likelihood of $\mathcal{T}$ and Q is thus $L(\mathcal{T}, Q; D) := P(D|\mathcal{T}, Q)$. For a given probabilistic model, the aim is to find the tree and rate matrix maximizing the likelihood $\max_{\mathcal{T},Q} L(\mathcal{T}, Q; D)$. This optimal tree and rate matrix are the maximum likelihood estimates.

We will now explain how to calculate $P(D|\mathcal{T}, Q)$, and thus the likelihood. As a very naive approach, we can simulate alignments for a given tree and rate matrix: Having a probabilistic description of the process means that given the parameters $\mathcal{T}$ and Q, we can simulate sequence data along the tree, i.e. obtain a simulated alignment of n sequences. The probability of a specific alignment D is the frequency with which the particular alignment D will be simulated along the given tree for the given rate matrix, when simulating many alignments. This simulation-based approach is very slow though for determining $P(D|\mathcal{T}, Q)$, as we have to simulate many, many alignments.

We next discuss how to analytically calculate the probability of the alignment given $\mathcal{T}$ and Q. The probability of the alignment, i.e. the sequences at the tips, is simply the joint probability of the sequences at the tips and sequences at all internal nodes of the tree, summed over all possible sequences at internal nodes. To illustrate how the likelihood is calculated based on a given tree, we will use the toy alignment and the UPGMA tree as displayed in Figure 6.17. In what follows, we will consider nucleotide sequence alignments, but the approaches are equivalent for codon or amino acid sequence alignments.

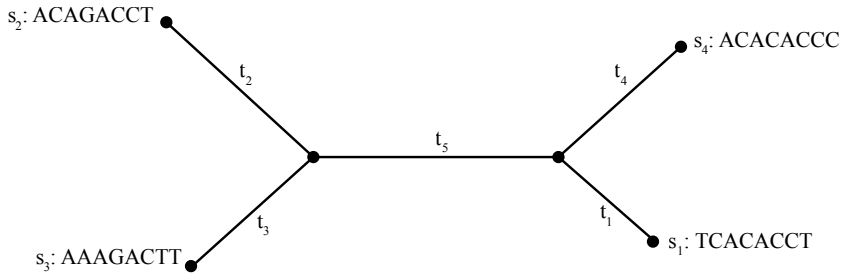


Figure 6.17: Unrooted tree with sequences for which we want to calculate the likelihood.

6.3.3.1 Calculating the likelihood using a brute force approach

We will now show how to calculate the probability of the sequences at the tips for a given tree and rate matrix, by explicitly summing over all internal node sequences. Typically one of the time-reversible models introduced in Chapter 5 is assumed, and in what follows, we will also assume one of these models. These models are a subset of the models assuming the sites in the alignment evolve independently from each other. As a consequence, we can calculate the likelihood for each site separately, and then take the product over the single site likelihoods to get the full likelihood. Put into mathematical equations, let us assume that the alignment consists of n

sequences $(s_1, \ldots, s_n)$ with m sites each, then the probability of the sequences is,

$$P(s_1, \ldots, s_n|\mathcal{T}, Q) \;=\; \prod_{j=1}^{m} P(s_{1,j}, \ldots, s_{n,j}|\mathcal{T}, Q),$$

where $s_{k,j}$ is the j-th site of sequence s_k.

Now, in order to evaluate $P(s_{1,j}, \ldots, s_{n,j}|\mathcal{T}, Q)$ for all j, we add a root to the unrooted tree at an arbitrary position, leading to tree $\mathcal{T}_r$ (node s_7 in Figure 6.18). Second we assign arbitrary sequences to all internal nodes (i.e. nodes s_5, s_6 and s_7 in Figure 6.18).

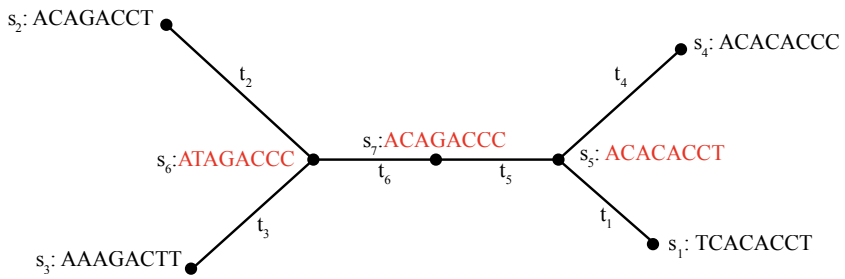


Figure 6.18: The tree with sequences for which we want to calculate the likelihood. The sequences at the internal nodes and the root node position have been assigned arbitrarily.

For n sequences, the rooted tree has $n-1$ internal nodes, with sequences $s_{n+1}, \ldots, s_{2n-1}$. Now, by summing over the possible nucleotides at the internal nodes, we obtain:

$$P(s_{1,j}, \ldots, s_{n,j}|\mathcal{T}_r, Q) = \sum_{s_{n+1,j}\in\{\mathrm{T,C,A,G}\}} \cdots \sum_{s_{2n-1,j}\in\{\mathrm{T,C,A,G}\}} P(s_{1,j}, s_{2,j}, \ldots, s_{2n-1,j}|\mathcal{T}_r, Q)\ .$$

The probability $P(s_{1,j}, s_{2,j}, \ldots, s_{2n-1,j}|\mathcal{T}_r, Q)$ is evaluated by (i) calculating the transition probability from the ancestral node to the descendant node for each branch l, with starting sequence s_{l_1}, ending sequence s_{l_2}, and branch length t_l, $p_{s_{l_1,j},s_{l_2,j}}(t_l)$, and (ii) calculating the probability of the nucleotide at the root using the equilibrium probability, $\pi(s_{2n-1,j})$. We get the overall expression,

$$P(s_{1,j}, s_{2,j}, \ldots, s_{2n-1,j}|\mathcal{T}_r, Q) = \pi(s_{2n-1,j}) \prod_{l=1}^{2n-2} p_{s_{l_1,j},s_{l_2,j}}(t_l)\ .$$

For example, for site $j = 2$ in the example of Fig. 6.18, the equation is,

$$P(s_{1,2}, s_{2,2}, \ldots, s_{2n-1,2}|\mathcal{T}_r, Q) = \pi_{\mathrm{C}} p_{\mathrm{C,C}}(t_5) p_{\mathrm{C,C}}(t_4) p_{\mathrm{C,C}}(t_1) p_{\mathrm{C,T}}(t_6) p_{\mathrm{T,C}}(t_2) p_{\mathrm{T,A}}(t_3)\ .$$

By summing over all internal sequences and then multiplying across all sites, we obtain the probability of the sequence alignment. Note that we obtain this probability for a rooted tree. However, as we assume time-reversible models, the direction of time flow along the branches is not determined by the model. This means that we obtain the same probability independent of the choice on where the root is added to the unrooted tree (see also Section 6.3.3.3 for an example). In summary,

$$P(s_1, \ldots, s_n|\mathcal{T}, Q) = \prod_{j=1}^{m} \sum_{s_{n+1,j} \in \{\mathrm{T,C,A,G}\}} \cdots \sum_{s_{2n-1,j} \in \{\mathrm{T,C,A,G}\}} \pi(s_{2n-1,j}) \prod_{l=1}^{2n-2} p_{s_{l_1,j}, s_{l_2,j}}(t_l) \ .$$

Next, we assess the running time of this approach for calculating this probability and thus the likelihood of $\mathcal{T}$ and Q given the sequence alignment. In our example, we have $4 \times 4 \times 4 = 64$ possibilities for the nucleotides at site j for a tree of four tips. Thus, the sum in our tree with three internal nodes consists of 64 terms. The sum over all nucleotides in a tree with n tips has 4^{n-1} terms, as the tree has $n-1$ internal nodes. Furthermore, for each nucleotide configuration, the approach needs to consider each branch in the tree (i.e. $2n-2$ branches). The approach also has to multiply the likelihood over all m sites in the alignment . Overall, the running time of this brute force algorithm is thus $\mathcal{O}(m4^n n)$, meaning it is exponential in n.

6.3.3.2 Calculating the likelihood using Felsenstein's pruning algorithm

The likelihood can be computed in linear time with Felsenstein's pruning algorithm [**felsenstein1973**, **Felsenstein1981**], given that the transition probabilities $p_{X,Y}(t)$ are known. Analog to the brute force algorithm, we arbitrarily root the tree and calculate the likelihood for each site in the alignment independently. However, we now sum over the possible nucleotides at the internal nodes in the tree more efficiently using dynamic programming (the concept of dynamic programming was already introduced in chapter 3). The strategy of recursively traversing the tree in Felsenstein's pruning algorithm is analog to the strategy in the Fitch algorithm (Section 6.3.2.3).

Given nucleotide $X \in \{\mathrm{T, C, A, G}\}$ at node k, let the probability of the nucleotides at the tips descending from node k be $P(D_k|X)$. This probability is central to the Felsenstein pruning algorithm. For a tip node k with nucleotide Y, we have $P(D_k|X) = 1$ if $X = Y$ and $P(D_k|X) = 0$ otherwise. In case we do not have information regarding the nucleotide of tip k (a "-" in the alignment), we initialize with $P(D_k|X) = 1$ for $X \in \{\mathrm{T, C, A, G}\}$, i.e. assuming each nucleotide was possible (note that these methods all assume that a - means that any nucleotide on that site is possible, instead of assuming a real gap). Let k being an internal node with the descending nodes l and m and branch lengths t_l, t_m, for which we already calculated $P(D_l|Y)$,$P(D_m|Z)$ for $Y, Z \in \{\mathrm{T, C, A, G}\}$. Then, the probability $P(D_k|X)$ is obtained by multiplying the probabilities of the two descendant subtrees and the transition probabilities from k to l and m,

$$P(D_k|X) = \left(\sum_{Y \in \{\mathrm{T,C,A,G}\}} p_{X,Y}(t_l) P(D_l|Y) \right) \times \left(\sum_{Z \in \{\mathrm{T,C,A,G}\}} p_{X,Z}(t_m) P(D_m|Z) \right). \quad (6.6)$$

In summary, after having the probabilities at the tips defined, we prune "cherries" recursively towards the root using the formula for $P(D_k|X)$ defined in Equation 6.6. We finish the pruning at the the root r, where we calculate $P(D_r|X)$, with $X \in \{\mathrm{T}, \mathrm{C}, \mathrm{A}, \mathrm{G}\}$. Finally, the probability of the sequences observed at tips at site j is:

$$P(s_{1,j}, \ldots, s_{n,j}|\mathcal{T}, Q) = \sum_{X \in \{\mathrm{T,C,A,G}\}} P(D_r|X)\pi_X.$$

For the example tree in Figure 6.19, where we consider a single nucleotide position, we put the values of $P(D_k|X)$ for all tip nodes, and need to calculate the probabilities for each of the nucleotides marked by "?". We can e.g. calculate $P(D_6|\mathrm{T})$ as follows:

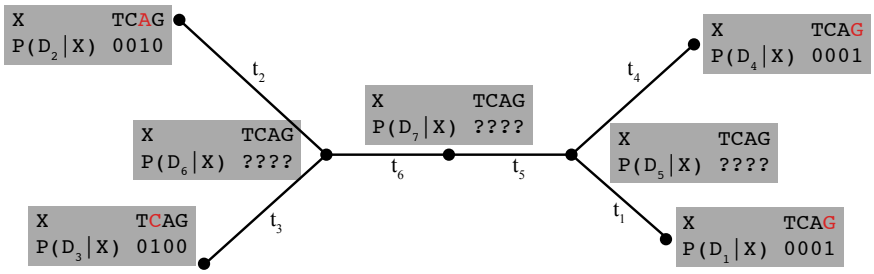


Figure 6.19: Example of Felsenstein's pruning algorithm. The red nucleotides at the tips are the data.

$$\begin{aligned} P(D_6|\mathrm{T}) &= \sum_{Y \in \{\mathrm{T,C,A,G}\}} p_{\mathrm{T},Y}(t_2)P(D_2|Y) \times \sum_{Z \in \{\mathrm{T,C,A,G}\}} p_{\mathrm{T},Z}(t_3)P(D_3|Z) \\ &= p_{\mathrm{T,A}}(t_2) \times p_{\mathrm{T,C}}(t_3)\ . \end{aligned}$$

A full example of the Felsenstein's pruning algorithm with the internal node probabilities calculated for each nucleotide is shown in Figure 6.20. For an alignment of length m on the tree shown in Figure 6.19 let us compare the brute force way of writing the likelihood,

$$\begin{aligned} P(s_1, s_2, s_3, s_4|\mathcal{T}, Q) = \prod_{j=1}^{m} \sum_{s_{7,j} \in \{\mathrm{T,C,A,G}\}} \sum_{s_{6,j} \in \{\mathrm{T,C,A,G}\}} \sum_{s_{5,j} \in \{\mathrm{T,C,A,G}\}} \pi(s_{7,j}) p_{s_{7,j},s_{6,j}}(t_6) \\ \times p_{s_{6,j},s_{3,j}}(t_3) p_{s_{6,j},s_{2,j}}(t_2) p_{s_{7,j},s_{5,j}}(t_5) p_{s_{5,j},s_{4,j}}(t_4) p_{s_{5,j},s_{1,j}}(t_1), \end{aligned}$$

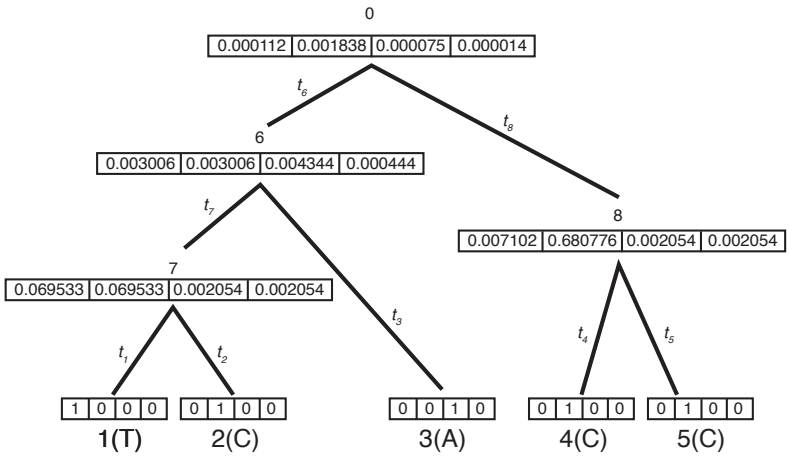


Figure 6.20: The question marks "?" in the previous figure were evaluated assuming a K80 nucleotide substitution model with $\kappa = 2$ (cf. section 5.2.2), and branch lengths $t_6, t_7, t_8 = 0.1$ and $t_1, \ldots, t_5 = 0.2$ (figure adapted—and corrected—from [**Yang2014**]).

with Felsenstein's likelihood,

$$P(s_1, s_2, s_3, s_4|\mathcal{T}, Q) = \prod_{j=1}^{m} \sum_{s_{7,j}\in\{\mathrm{T,C,A,G}\}} \pi(s_{7,j}) \times \left(\sum_{s_{6,j}\in\{\mathrm{T,C,A,G}\}} p_{s_{7,j},s_{6,j}}(t_6) p_{s_{6,j},s_{3,j}}(t_3) p_{s_{6,j},s_{2,j}}(t_2) \right) \times \left(\sum_{s_{5,j}\in\{\mathrm{T,C,A,G}\}} p_{s_{7,j},s_{5,j}}(t_5) p_{s_{5,j},s_{4,j}}(t_4) p_{s_{5,j},s_{1,j}}(t_1) \right) .$$

Notice how the summation signs moved "down the tree" when we used the Felsenstein's algorithm.

6.3.3.3 Time-reversibility implies that differently rooted tress have the same likelihood

In the calculation of the likelihood, the condition of time-reversibility implies that the likelihood remains the same no matter where one places the root of the tree. We illustrate this in a very simple example of a two-tip tree (Figure 6.21) where we consider two possible roots, D_1 and D_2.

Suppose that one site of a sequence at the tips of this tree has nucleotide s_1 and s_2 at the two tips, respectively; $s_i \in \mathcal{N} = \{T, C, A, G\}$. The likelihood of the tree starting in root D_1 is then:

$$P(D_1) = \sum_{X\in\mathcal{N}} \pi_X p_{X,s_1}(t_1) p_{X,s_2}(t_2 + t_3) \tag{6.7}$$

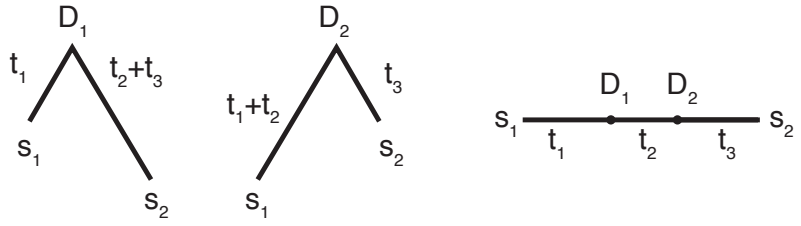


Figure 6.21: Illustration of why time reversibility plays an important role in likelihood calculations.

The transition probability of going from the internal state X to the observed nucleotide s_2 in time t_2+t_3 can be subdivided into the transition probability to another internal state (at node D_2) and then to s_2. As we do not know which state the nucleotide is in at node D_2, we have to sum over all possibilities:

$$p_{X,s_2}(t_2+t_3) = \sum_{Y\in\mathcal{N}} p_{X,Y}(t_2)p_{Y,s_2}(t_3)$$

We can now substitute this into equation 6.7 and rewrite these probabilities:

$$\begin{aligned}
P(D_1) &= \sum_{X\in\mathcal{N}} \pi_X p_{X,s_1}(t_1) \sum_{Y\in\mathcal{N}} p_{X,Y}(t_2)p_{Y,s_2}(t_3) \\
&= \sum_{X\in\mathcal{N}}\sum_{Y\in\mathcal{N}} \pi_X p_{X,s_1}(t_1)p_{X,Y}(t_2)p_{Y,s_2}(t_3) \\
&= \sum_{X\in\mathcal{N}}\sum_{Y\in\mathcal{N}} \underbrace{\pi_X p_{X,Y}(t_2)}_{\overset{(5.10)}{=}\pi_Y p_{Y,X}(t_2)} p_{X,s_1}(t_1)p_{Y,s_2}(t_3) \\
&= \sum_{X\in\mathcal{N}}\sum_{Y\in\mathcal{N}} \pi_Y p_{Y,X}(t_2)p_{X,s_1}(t_1)p_{Y,s_2}(t_3) \\
&= \sum_{Y\in\mathcal{N}} \pi_Y p_{Y,s_2}(t_3) \sum_{X\in\mathcal{N}} p_{Y,X}(t_2)p_{X,s_1}(t_1) \\
&= \sum_{Y\in\mathcal{N}} \pi_Y p_{Y,s_2}(t_3)p_{Y,s_1}(t_1+t_2) \\
&= P(D2)
\end{aligned}$$

By just re-ordering the summation and using the detailed balance condition in equation 5.10, we proved that the likelihoods for both rooted trees are the same independent of where one places the node in the tree. We can extend this procedure also to bigger trees.

6.3.3.4 Runtime of the Felsenstein's pruning algorithm

Next, we determine the running time of Felsenstein's pruning algorithm. In each pruning step, we sum over four states twice (the two descending branches and four possible nucleotides at the ends of these branches), meaning we perform a constant number of changes of operations. The number of pruning steps equals the number of

internal nodes in the rooted tree, and thus the full pruning procedure has running time $\mathcal{O}(n)$. In addition, this procedure has to be performed for each of the m sites. Thus, in total the running time of the algorithm is $\mathcal{O}(nm)$ which is linear in n (vs. $mathcalO(m4^n n)$ for the brute force approach).

6.3.3.5 Runtime and statistical consistency of the maximum likelihood tree inference method

While obtaining the likelihood of a tree for a given alignment can be done in polynomial time using Felsenstein's pruning algorithm, inferring the maximum likelihood phylogeny is NP-hard [**roch2006MLNPcomp**]. Thus, we essentially have to try out every unrooted tree on n tips for an alignment of n sequences. Good news is that maximum likelihood tree reconstruction is statistically consistent (see e.g. [**felsenstein1973**, **Felsenstein2004**]).

6.3.3.6 A connection between maximum likelihood and parsimony

Despite the fact that maximum likelihood methods have been criticised by cladists for using explicit evolutionary models (see e.g. [**Farris1983**], and [**Felsenstein2004**] for more details and more references), it has been shown that the parsimony tree is in fact equal to the maximum likelihood tree when assuming a no-common-mechanism substitution model [**TuffleySteel1997**]. The model, however, is somewhat artificial. Due to the short-comings of the distance-based methods (only pairwise distances are considered) and the parsimony method (long-branch attraction), maximum likelihood methods are, together with Bayesian phylogenetic methods (Chapter 10), the method of choice for most phylogenetic studies.

6.4 Searching the tree space

Distance-based optimality methods, parsimony, and maximum likelihood methods are NP-hard and thus essentially need to search across all possible unrooted trees. For these methods, with the exception of parsimony, additionally all possible branch lengths need to be investigated for optimality.

The search through branch lengths requires essentially to optimize a continuous function (least square function or likelihood function) over m real variables, namely the m branch lengths for a proposed tree on m branches. Such optimization can be done in a straightforward manner e.g. with hill climbing (see e.g. [**Yang2014**], Chapter 4.6).

Searching through the space of all trees without branch lengths is more complicated, as the space of trees is a huge discrete space. Checking each tree for optimality in tree space will be too slow (see again Table 6.1), thus we will check as many trees as possible. Note that this will mean though that we may miss the best tree. However, unless P=NP or computational speed will increase drastically, we cannot guarantee optimality of the output tree.

Checking random trees from all the possible trees, is inefficient, as we often check trees which will have a very bad least square, parsimony, or likelihood score. Thus chances are high that we will miss the best tree unless we check very many trees. A better approach uses a random walk to search tree space. Starting from an arbitrary starting tree, the method proceeds by iteratively modifying the "current" tree by replacing it with a similar tree having a better score. This is the same idea as hill-climbing, but applied to the discrete space of tree topologies. This procedure requires specific *tree moves* that propose new trees, three of which—NNI, SPR, TBR—we will discuss here (for more details see e.g. [**Felsenstein2004**, **Yang2014**]).

The NNI move, or ***N****earest-****N****eighbour* ***I****nterchange*, switches two neighbouring subtrees. It chooses an internal branch uniformly at random and two out of the four subtrees attached to this internal branch are then exchanged, as shown in Figure 6.22.

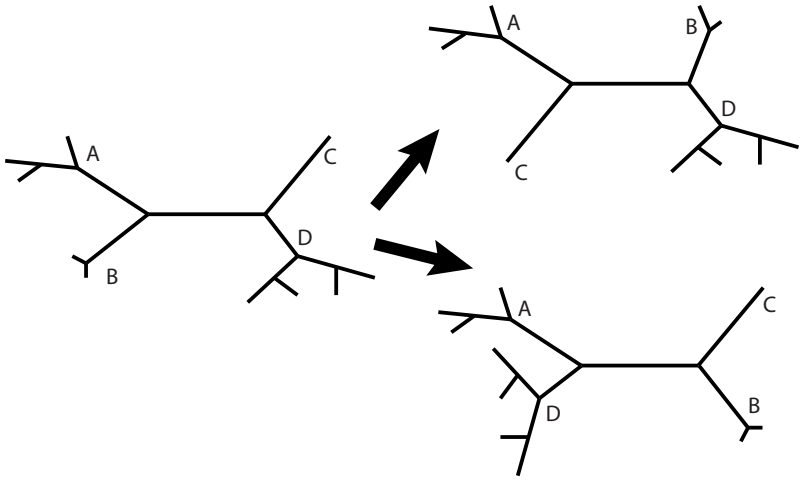


Figure 6.22: The nearest-neighbour interchange (NNI) algorithm. Each internal branch in the tree connects four subtrees or nearest neighbours (A,B,C,D). Exchanging a subtree on one side of the branch with another on the other side constitutes an NNI move. Two such rearrangements are possible for each internal branch (here $B \leftrightarrow C$ and $B \leftrightarrow D$) Figure and caption adapted from [**Yang2006**], figure 3.12.

The other two moves are SPR, ***S****ubtree* ***P****runing and* ***R****egrafting*, and TBR, ***T****ree* ***B****isection and* ***R****econnection*. In SPR, an internal branch together with its descending subtree is first chosen uniformly at random. The chosen branch and subtree is detached from the remaining tree. Then, a branch in the remaining tree is chosen uniformly at random to which the detached branch and subtree is connected (Figure 6.23, a). In TBR, a random internal branch is chosen and deleted, thereby splitting a tree into two subtrees. Two other branches, one in each subtree, are then chosen uniformly at random and merged in order to reconnect the two subtrees, as shown in Figure 6.23, b.

State-of-the-art implementations employing efficient heuristics allow us to infer maximum likelihood trees for datasets containing thousands of sequences. Some of the

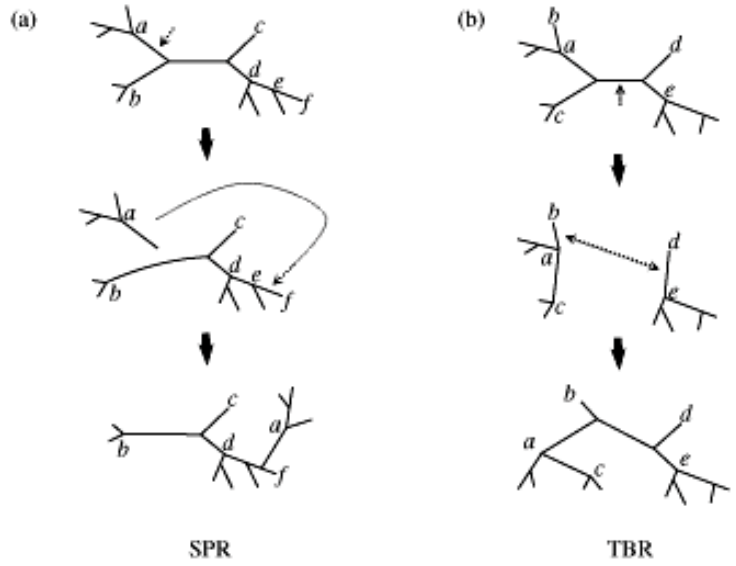


Figure 6.23: (a) Tree move by subtree pruning and regrafting (SPR). A subtree (for example, the one represented by node a) is detached, and then reattached to a different location on the tree. (b) Tree move by tree bisection and reconnection (TBR). The tree is broken into two subtrees by removing an internal branch. Two branches, one from each subtree, are then chosen uniformly at random and rejoined to form a new tree. Figure and caption adapted from [**Yang2006**], figure 3.13.

commonly used software packages are PhyML [**phyml**] and RaxML [**raxml**]. RaxML is optimized for large datasets and can infer trees on the order of 10^5 tips.

6.5 Rooting the tree

As we have seen, most presented phylogenetic inference algorithms produce an unrooted tree. This is due to the fact that the employed models of molecular evolution are time reversible, which does not allow us to distinguish between forward and backward-time evolution (see Section 5.2.5). We can *root the tree* by including an *outgroup* into the analysis. An outgroup is a group of individuals/species which are distant from the rest of the data considered in the tree. E.g. for mammals, one could use sequences from birds as an outgroup; for the transmission tree of HIV subtype B in Switzerland, one could use sequences from HIV subtype C or A. The point where the outgroup connects to the phylogeny of the species of interest is defined to be the root. Thus, one assumes *a priori* that the outgroup attaches to the root of the remaining phylogeny, because it is assumed to have diverged from the phylogeny of interest at a much earlier point in time.

6.6 Adding a calendar time scale to phylogenetic trees

All tree inference methods discussed so far estimate trees with branch lengths defined as a number of substitutions. There are many applications for which it is useful to have trees with branch lengths and internal node times corresponding to calendar time: e.g. to time speciation and extinction events on the tree of life, or to estimate the basic reproductive number of an epidemic. Such time-scaled trees are called *calendar time trees*.

Under strict conditions, i.e. when sequences evolved according to a strict molecular clock and are all sampled at the same time, the inferred UPGMA tree has branch lengths that are directly proportional to calendar time (see 6.3.1.1). For more complex scenarios, e.g. uncorrelated and correlated relaxed-clock models or serially sampled phylogenies, this is typically not the case.

A computationally efficient way to date large serially sampled phylogenies is the 'Least-Squares Dating' (LSD) method [**LSD**]. This method uses a binary phylogenetic tree with known branch lengths - inferred using a distance, parsimony or maximum likelihood based approach - and information about the sampling times of the tips, to estimate the substitution rate and the dates of all internal nodes.

Suppose that the input consists of a rooted binary tree R on n sequences, where internal nodes of R are numbered $1, \ldots, n-1$ and leaves $n, \ldots, 2n-1$. Sampling times $t_n, \ldots, t_{2n-1}$. The substitution rate is denoted by μ. The model then assumes that the substitutions along each branch (b_i) are the result of a strict-molecular clock with a constant substitution rate acting over a given calendar time interval $\Delta t = t_i - t_{a(i)}$, and a gaussian noise term $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2)$ stemming from sampling and estimation errors:

$$b_i = \mu \left(t_i - t_{a(i)}\right) + \epsilon_i$$

The estimates of the global substitution rate and of the internal node dates are then obtained by minimising the weighted least squares criterion:

$$\Phi(\mu, t_1, \ldots, t_{n-1}) = \sum_{i=2}^{2n-1} \frac{1}{\sigma_i^2} \left(b_i - \mu(t_i - t_{a(i)})\right)^2$$

Here the variance terms σ^2 are unknown, but using the Poisson nature of the substitution process one can arrive at the reasonable assumption:

$$\sigma_i^2 = \frac{\mu(t_i - t_{a(i)})}{s} \qquad \textit{and} \qquad \hat{\sigma_i^2} = \frac{b_i + c/s}{s}$$

where c is a constant, added to avoid infinite weights in the case of zero branch lengths ($b_i = 0$).

This method has been shown to be reasonably accurate also in cases with minor uncorrelated violations of the strict molecular clock, as these are absorbed into the

added gaussian noise term. With rooted input trees the time complexity of the LSD algorithm is nearly linear (i.e. $\mathcal{O}(n)$, where n is the number of tips). When the method is extended to account for unrooted trees, the time complexity is nearly quadratic.

6.7 Examples of applications of phylogenetic methods

6.7.1 The first phenetic and cladistic phylogenies

The first phenetic tree (Figure 6.24) was reconstructed based on bee data in 1957 [**MichenerSokal1957**]. The tree was built based on morphological traits only.

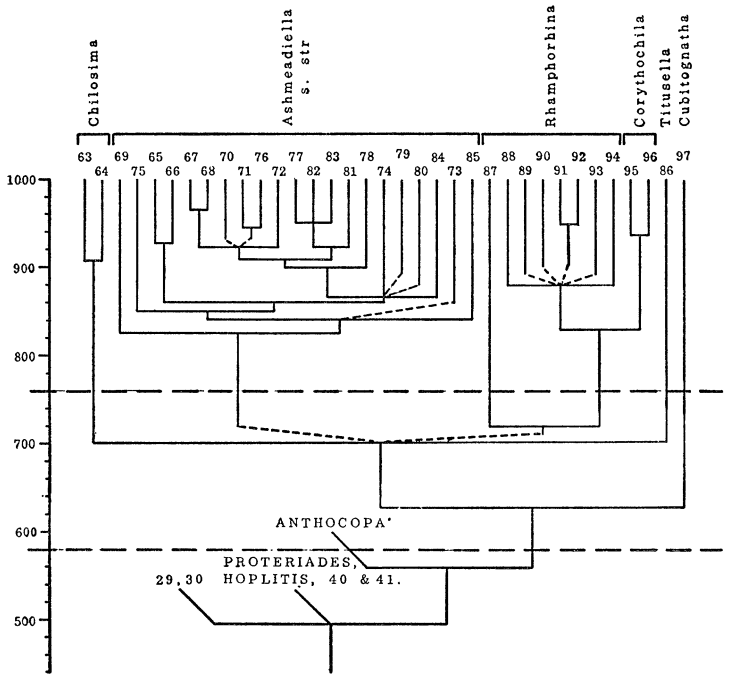


Figure 6.24: The first phenetic tree: bee phylogeny based on morphological data only. This was the first time that numerical methods entered systematics. Figure adapted from [**MichenerSokal1957**].

The first cladistic (parsimony) tree was constructed in 1964 [**EdwardsCavalli1964**]. The researchers explored the evolution of human populations using the blood groups and their frequencies. Curiously, the tree of blood type frequencies correlates with what is known about the migration of human populations (Figure 6.25).

6.7.2 Phylogenetics can reveal the origin of an emerging infectious diseases—HIV as an example

In July 1981, the New York Times reported on a rare cancer in 41 homosexual men in New York and California, 8 of whom had already died. This disease was named

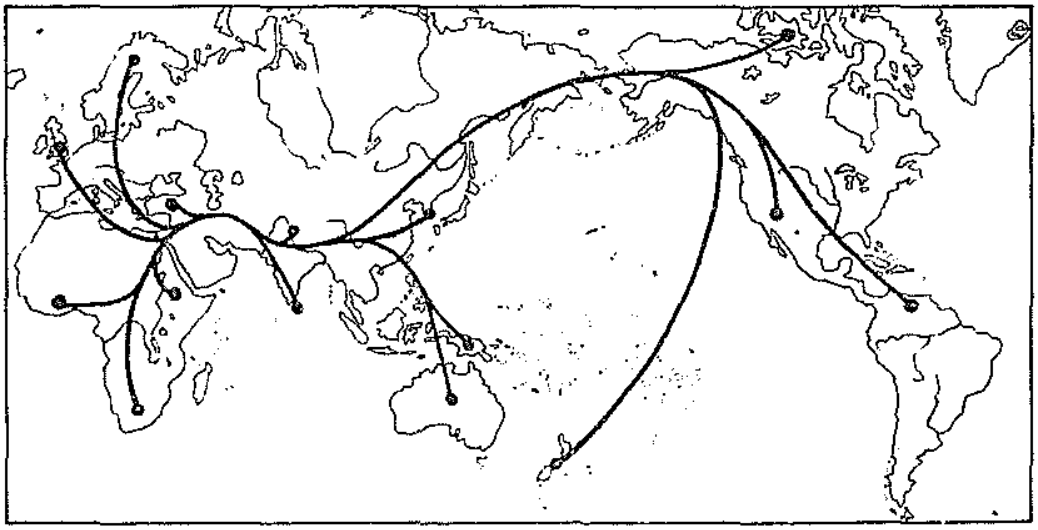


Figure 6.25: First parsimony tree: human populations based on blood group gene frequencies (same paper also introduced the maximum likelihood method, but since the pruning approach was not known, the method was computationally very slow). Figure adapted from [**EdwardsCavalli1964**].

GRID, for gay-related immunodeficiency disease.

In March 1982, the Washington Post reported about the same disease, now called Acquired Immunodeficiency Syndrome (AIDS). More than 1100 Americans, not only gay, were already diagnosed with this disease, of which more than 400 had died. The disease was spreading rapidly, more than 200 of the diagnoses were within the previous month. Half of the victims were younger than 35 years old.

In 1983, a virus was isolated from patients with AIDS in two separate laboratories [**BarreSinoussi:1983**, **Gallo:1983**]. There was a controversy whether this virus, named Human Immunodeficiency Virus (HIV) was actually the cause of AIDS [**marx1984**]. Today, it is well-established among scientists that HIV causes AIDS. While a large fraction of the general public also acknowledges the cause of AIDS being HIV, some people and even governments have denied that HIV is the cause of AIDS, e.g. until 2000 in South Africa [**durban˙declaration**].

As of today, 0.5% of the human population is infected with HIV [**WHO˙HIV˙world**]. South Africa is the country with the largest fraction of people being infected, namely around 20% [**UNAIDS2017**]. Some regions in South Africa even have a prevalence of up to 25-40% [**Shisana2012**].

Thus, it is essential that we obtain a detailed understanding of HIV dynamics, such that appropriate actions can be taken in order to fight the epidemic. In the remainder of this section, we will discuss how the origin of the HIV epidemic was determined using phylogenetic trees. In Sections 6.7.3 and 6.7.4, we will then discuss how we can investigate the spread of HIV using phylogenetic trees. As we will see, the spread may be investigated to obtain knowledge for public health, or to provide evidence

in criminal cases.

6.7.2.1 HIV phylogeny reveals the origin of the epidemic

It was clear that HIV must have evolved from some other similar virus, however, no virus similar to HIV was known within the human population. Therefore, scientists started searching for viruses similar to HIV that infect species closely related to humans. The idea is that a virus infecting a closely related species (see Figure 6.1) might easily adapt and infect humans as well (zoonotic transmission).

A similar virus, found in most simian species, is the simian immunodeficiency virus (SIV). Many simian species are a natural host to SIV, meaning SIV occurs in these species, often without causing disease. A zoonotic transmission from simians to humans was suggested.

A huge effort was made in the 1980s-1990s to collect SIV sequences from various simian species. Based on these sequences, scientists could reconstruct maximum likelihood SIV/HIV phylogenies, examples of which are shown in Figures 6.26 and 6.27. The tips of the phylogeny are labelled with the virus name but also the host species name from which the virus at the tip was isolated (e.g. cpz in the sequence name stands for chimpanzee, and sm for sooty mangabee; the remaining names are not important for our purpose).

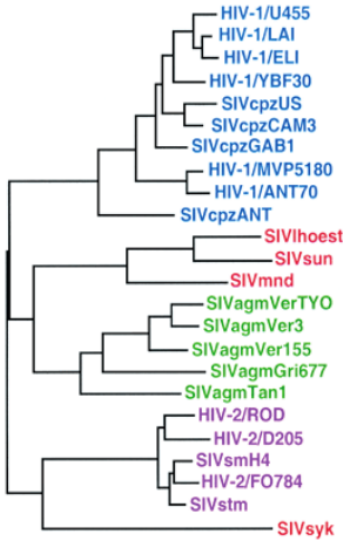


Figure 6.26: Maximum likelihood HIV/SIV phylogeny. Figure adapted from [**Hahn2000**].

Figure 6.26 shows that HIV-1 sequences are clustering with chimpanzee SIV sequences (SIVcpz) but HIV-2 sequences are clustering with sooty mangabey SIV sequences (SIVsm). These trees suggest that HIV-1 is a zoonosis from chimpanzees, or that HIV-1 jumped from humans to chimpanzees (and the analog for HIV-2 and sooty mangabeys). Phylogenies indicate the direction of transmission from chimpanzees (resp. sooty mangabees) to humans, if, given more and more sequences

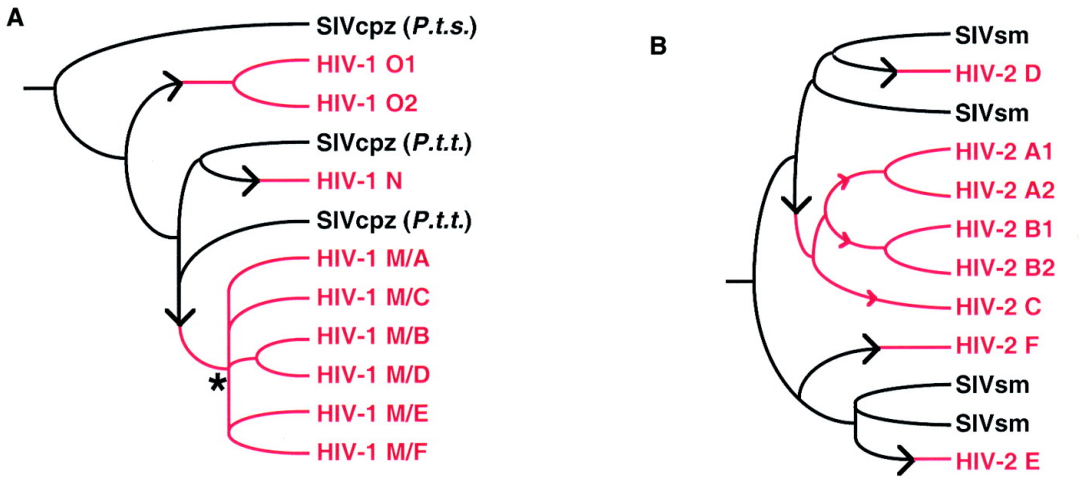


Figure 6.27: Introduction of HIV-1 (A) and HIV-2 (B) into the human population. HIV-1 is a zoonotic transmission from chimpanzees, HIV-2 a zoonotic transmission from sooty mangabeys. Figure adapted from [**Hahn2000**].

are added, the HIV sequences form few nested clades within a large tree of SIV sequences. This is in fact what we observe with adding recent sequences to the original trees. We note that the number of nested HIV clades within the SIV sequences is the number of observed zoonoses.

At the time, the direction of transmission was a controversy [**Sharp1995**]. However, non-phylogenetic evidence such as simians being a natural reservoir for SIV, their host range covering the areas where HIV appeared first, as well as results of evolutionary sequence analyses of SIV and HIV sequences all indicated that all HIV clades are results of zoonoses from simians to humans [**Sharp1995**, **Gao1999**, **Hahn2000**]. In particular, figure 6.27 (A), indicates that at least three independent introductions occurred of HIV-1 from chimpanzees into the human population. Figure 6.27 (B), indicates that at least four introductions of HIV-2 from sooty mangabeys into the human population.

6.7.2.2 How/why did HIV jump from simians to humans?

Answering this question requires considerations beyond phylogenetic analysis. Two main hypotheses were put forward to answer this question.

Hypothesis 1 (polio hypothesis): Humans caused the HIV epidemic through contaminated polio vaccines in the 1950s in Africa. This hypothesis was popularized by the journalist Edward Hooper's book "The river" [**hooper1999**]. W.D. Hamilton, one of the leading evolutionary biologists in the 20th century (kin selection, altruism; Hamilton's rule), tried to prove this hypothesis. Tragically, he died from malaria after returning from an expedition in Congo aiming to find evidence for this hypothesis. Polio vaccines found in the freezer later did not show any sign of contamination, which in the end proved the hypothesis to be wrong.

Hypothesis 2 (hunter hypothesis): HIV jumped from simians to humans in the course of hunting of simians [**sharp2001**]. Since hunters can experience blood-to-blood contact with the hunted simians, e.g. through cuts, this could indeed provide a potential transmission route. The hunter hypothesis is supported by the fact that the areas with high SIV prevalence coincide with the early HIV outbreaks and that hunting of simians occurs in these areas. Furthermore, the hunter hypothesis explains the observation that there has not been one but rather several introductions of HIV into the human population. Today, the hunter hypothesis is supported by most scientists.

6.7.3 The HIV epidemic in Switzerland

Within an epidemic, pathogen phylogenies can display the interaction between different transmission groups. The phylogenetic tree in Figure 6.28 contains 5700 HIV sequences from the Swiss epidemic [**Kouyos2010**]. The sequences were collected from patients in order to screen for drug resistant strains and to define a proper course of antiretroviral treatment. Each tip of this tree corresponds to a single (consensus) pathogen genetic sequence from an infected host. The tip color indicates the transmission group: blue - Swiss intravenous drug users, red - Swiss men having sex with men, cyan - Swiss heterosexuals, black - non-Swiss.

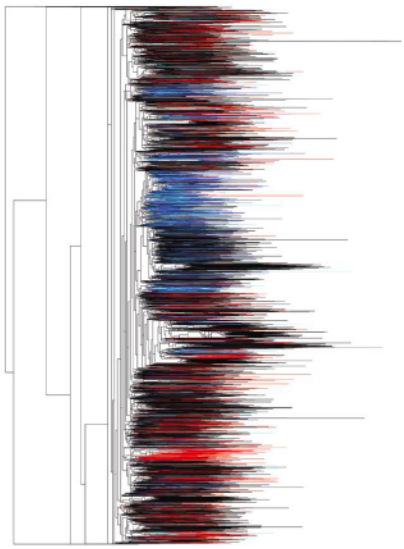


Figure 6.28: Maximum likelihood phylogeny of HIV transmission in Switzerland. Branches are colored according to the the origin and risk group: blue - Swiss intravenous drug users, red - Swiss men having sex with men, cyan - Swiss heterosexuals, black - non-Swiss. Figure adapted from [**Kouyos2010**].

Clades of colour nested within black sequences indicates transmission of HIV from abroad into Switzerland. Colour mixing indicates HIV transmission within Switzerland between different groups. No mixing means that the disease is mostly spreading

within each of the different Swiss transmission groups separately. We observe that the red sequences often form small clusters within the black "backbone". This indicates an ongoing transmissions within the group of the Swiss men having sex with men. In contrast, the Swiss heterosexuals (cyan) and intravenous drug user (blue) colours are well-mixed. This indicates that there is frequent transmission going back-and-forth between the two groups.

6.7.4 Phylogenetics as evidence in criminal investigations

6.7.4.1 HIV criminal case: Louisiana 1994

In the following, we discuss the first court case in which phylogenetic methods were used for providing forensic evidence [**Metzker2002**]. In 1994, a woman from Louisiana (USA) accused her ex-lover, a physician, of having infected her with HIV. While she was tested HIV-positive, none of the ten men she reported sexual contact with during the previous decade was HIV positive. The woman claimed that the ex-lover purposefully infected her through an injection, which he claimed was a vitamin B boost, administered during a late night visit to his practice. This visit to the physician's practice followed a fight between the victim and her ex-lover. The victim reported that her ex-lover did not want her to leave him, and if she did, he wanted to make sure she did not have sexual contact with anyone else anymore.

HIV only survives a couple of hours in vitro, however, it was observed that the physician took a blood sample earlier that day from a HIV-positive patient, but never sent the blood to the laboratory.

How can one prove the victim's claim of intentional HIV transmission on the night of the "vitamin B boost"? For the first time in a criminal trial, phylogenetic methods were used to assess the claim. HIV samples from the victim, the suspected donor (i.e. the physician's patient) and 32 other HIV infected individuals from the local area were obtained and sequenced. Based on these sequences, HIV phylogenies were reconstructed using a variety of different methods and sequences from two different genes. In every single reconstruction, the victim's sequences clustered within the suspected donor's sequences, and the remaining 32 sequences were further apart (Figure 6.29) with bootstrap support (assessing robustness of results; see Section 7.4.3) of 96%-100%, indicating that the most likely route of transmission was from the suspected donor to the victim.

Together with other evidence, the physician was found guilty of having infected the woman with HIV. Additionally, he was found guilty of having infected the woman with Hepatitis C through the "vitamin" injection. As a result of these findings, he was sentenced to 50 years of prison.

6.7.4.2 Florida dentist

In an investigation by the Centers for Disease Control and Prevention (CDC) in the 1980s, a dentist who died of HIV had been accused of infecting 6 of his patients

Figure 6.29: Phylogeny obtained using distance-based methods displaying who infected whom in the "vitamin B injection" case. Each tip is a consensus HIV genetic sequence from a different host, except for the sequences in the boxes. The sequences in the small box all belong to the victim and the sequences in the big box all belong to the suspect. The procedure of "bootstrap" is discussed in Section 7.4.3. Figure adapted from [**Metzker2002**].

with HIV in the course of his practice [**OuEtAl1992**]. Patients' viral sequences were isolated and on these a phylogeny was reconstructed. The analysis done by the CDC revealed that for most of the patients, the sequences cluster within the dentist's sequences (Figure 6.30) and thus point to the dentist as the most likely source of the infection.

This case induced a discussion on hygiene rules for health care workers. It remains unclear whether the dentist wore any kind of protective gear during the practice (already in the 1980s the dentists were supposed to wear protective gloves). Furthermore, this was an exceptional case because in no other criminal case involving HIV, a health care worker transmitted to a patient. To this day, it is not known if the transmissions by Florida's dentist occurred by chance or on purpose.

6.7.4.3 Bulgarian nurses in Libya

The last criminal case we present occurred in Libya. Five Bulgarian nurses and a Palestinian doctor who were helping in Libyan hospitals were suspected of having transmitted HIV to more than 400 children. They were sentenced to death by the Libyan government and were incarcerated for over 8 years. During this time, enough

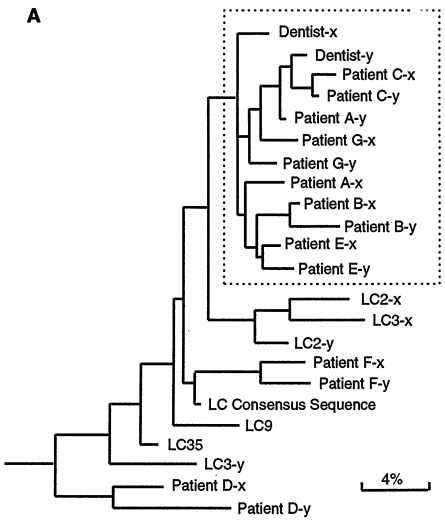


Figure 6.30: An HIV-positive Florida dentist infected 6 of his patients. More background can be obtained from the NY Times and the LA Times. Figure adapted from [**OuEtAl1992**].

data was collected and appropriate methods were developed to show that the transmission occurred before the healthcare workers arrived in the country (Figure 6.31, right). Upon political pressure, the health care workers were finally freed in 2007.

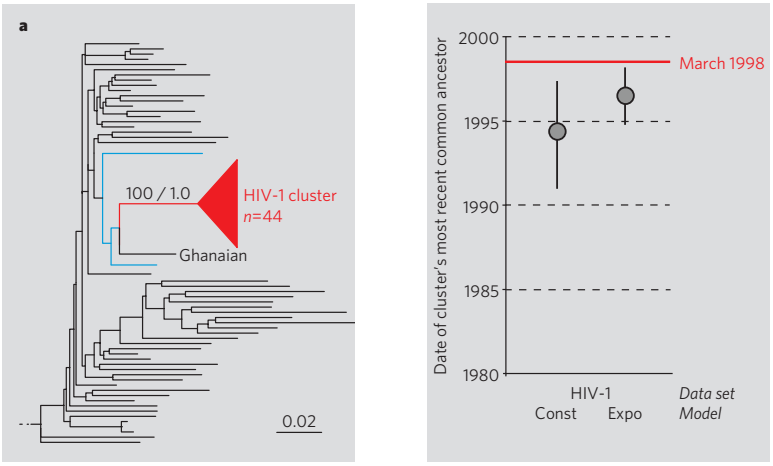


Figure 6.31: The phylogenetic tree (left) is a maximum likelihood tree with red being the subtree with the HIV sequences from the children. The transmission dates (right) were obtained from a Bayesian phylogenetics analysis with Const and Expo referring to different model assumptions (method will be described in Chapter 10). The time of arrival of the nurses in Libya is marked with the red horizontal line, highlighting that HIV was already spreading in the hospital before the nurses arrived. An informative summary of the case can be found on Wikipedia. Figure adapted from [**OliveiraEtAl2006**].

7 Statistical testing

In this chapter, we will discuss how to test whether a model fits the data well, and, given we have several different models available, we will discuss how to select the model that best fits a given dataset. Such techniques can be used to e.g. select the most appropriate substitution model for a given sequence alignment. We will also discuss how to assess confidence in parameter estimates for a given model. Further, we will assess the confidence in tree topologies using bootstrapping.

In testing theory, the word *hypothesis* is often used instead of the word model. A *null model* is a mathematical model, e.g. the Jukes Cantor model may be a null model. A *null hypothesis* is the hypothesis that the data evolved under the null model. Testing the null hypothesis means testing whether the null model is a good model for the data. If not, we reject the null hypothesis. If the null hypothesis was tested against an alternative hypothesis, rejecting the null hypothesis means that we favour the alternative hypothesis, formalized in an alternative mathematical model. As an example think of the six-sided die. The null hypothesis could be that the die is fair, i.e. obtaining a six happens with probability 1/6. The alternative hypothesis in this case would be all other probabilities to obtain a 6, i.e. $[0, 1] - \{1/6\}$. (We illustrate this example in more detail below.)

When performing *model selection*, we do not need a null model or null hypothesis, we can simply compare the different models. As throughout this section, we always directly specify a mathematical model along with a hypothesis, we use the word "model" directly and not talk about hypotheses. In what follows, we first discuss how to test for the plausibility of a null model (or null hypothesis) $\mathcal{H}_0$ (Section 7.1-7.2), and then perform model selection between a number of models $\mathcal{H}_0, \mathcal{H}_1, \ldots$ (Section 7.3). Then, given we pick a model, we assess the uncertainty in the parameter estimates (Section 7.4)

7.1 Testing for rejection of the model $\mathcal{H}_0$

In Section 4, we tested whether our data (namely a contingency table) rejects the null hypothesis of particular SNPs not being associated with a disease status (GWAS). The general idea was to determine the distribution $P(X = x|\mathcal{H}_0)$. The random variable X stands for one field in the contingency table. The null model assumes that this random variable is hypergeometrically distributed (Box 4) with parameters defined by the fixed row and column sums of the contingency table. Thus, for any possible observed data x, we can quickly evaluate $P(X = x|\mathcal{H}_0)$, and we can thus directly calculate the p-value for a particular observation (Box 1) based on the

hypergeometric distribution. If this p-value is below a given rejection threshold, we reject the null model $\mathcal{H}_0$.

7.2 Testing for rejection of the model $\mathcal{H}_0$ in favor of $\mathcal{H}_1$

Often, it is not possible to easily evaluate $P(X = x|\mathcal{H}_0)$ for all possible outcomes x. For example, when X is a random sequence alignment, and $\mathcal{H}_0$ is a particular tree with some rate matrix Q, determining the distribution $P(X|\mathcal{H}_0)$ for the particular tree and Q would mean to evaluate the probability for all sequence alignments. However, there are roughly 4^{mn} sequence alignments for n sequences of length m (we say here "roughly" as we did not consider the gaps).

In such cases, we "compare" $\mathcal{H}_0$ to other models. Such comparisons are based on likelihoods (see Box 18 for likelihoods in general and Section 6.3.3 for likelihoods in phylogenetics). We will now introduce *likelihood ratio tests* for testing if a null model $\mathcal{H}_0$ should be rejected when tested against model $\mathcal{H}_1$. Afterwards, we introduce the Akaike Information Criterion which ranks different models $\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2, \ldots$ according to their support based on likelihoods (Section 7.3).

7.2.1 Likelihood ratio tests (LRT)

Log likelihood ratios can not only be used to determine confidence intervals for maximum likelihood estimators (see Box 19) but also for model comparison. Likelihood ratio testing is an approach that lets us test a null model $\mathcal{H}_0$ against a model $\mathcal{H}_1$. For being able to use LRT, the null model needs to be a *nested model* which means that the parameters of $\mathcal{H}_0$ are a subset of the parameters of $\mathcal{H}_1$. As an example, JC69 is nested in K80, because when the two parameters of K80 have the same value, the model is a JC69 model. We determine if $\mathcal{H}_0$ should be rejected when tested against $\mathcal{H}_1$. We define the log likelihood ratio function between the two models $\mathcal{H}_0$ and $\mathcal{H}_1$ on a given dataset as,

$$LR(\mathcal{H}_1, \mathcal{H}_0) = 2 \log \left(\frac{L_1(\hat{\theta}_1)}{L_0(\hat{\theta}_0)} \right) = 2(\log L_1(\hat{\theta}_1) - \log L_0(\hat{\theta}_0)),$$

where L_1 and L_0 are the likelihood functions of models $\mathcal{H}_1$ and $\mathcal{H}_0$, and $\hat{\theta}_1$ and $\hat{\theta}_0$ are the maximum likelihood estimates under $\mathcal{H}_1$ and $\mathcal{H}_0$, respectively.

The likelihood value $L_1(\hat{\theta}_1)$ of the data under the general model will be higher or equal to the likelihood under the null model $L_0(\hat{\theta}_0)$. This is the case because the models are nested, i.e. fixing some parameters of $\mathcal{H}_1$ to certain values will lead to model $\mathcal{H}_0$. Thus, it is always possible for $\mathcal{H}_1$ to obtain at least the same likelihood as $\mathcal{H}_0$. More formally, $2(\log L_1(\hat{\theta}_1) - \log L_0(\hat{\theta}_0)) \geq 0$. However, a positive difference does not necessarily mean that $\mathcal{H}_1$ is a better choice, as $\mathcal{H}_1$ requires more parameters. Overly general models tend to have reduced explanatory power: a phenomenon

known as overfitting.

Now, we determine how big the log likelihood ratio should be for rejecting $\mathcal{H}_0$ against $\mathcal{H}_1$. Given that $\mathcal{H}_0$ indeed generated the data, Wilk's theorem [**Wilks1938**] states that if the sample size goes to ∞,

$$LR(\mathcal{H}_1, \mathcal{H}_0) = 2(\log L_1(\hat{\theta}_1) - \log L_0(\hat{\theta}_0)) \sim \chi^2_{df}$$

where *df* is the degrees of freedom of the χ^2 distribution (Box 3). *df* is calculated as the difference between the number of free parameters in $\mathcal{H}_1$ and the number of free parameters in $\mathcal{H}_0$. In case some parameters of $\mathcal{H}_0$ are at the parameter boundary of $\mathcal{H}_1$ (e.g. 0 or ∞), they typically only count for 0.5 degrees of freedom (for more details see e.g. [**SelfLiang1987**]).

In a *likelihood ratio test* (LRT), we choose a level of significance α (see also Box 1), and evaluate $LR(\mathcal{H}_1, \mathcal{H}_0)$. We will reject $\mathcal{H}_0$ if and only if the log likelihood ratio falls in the α-tail of the χ^2_{df} distribution, i.e. if $p_{\chi^2_{df}}(x > LR(\mathcal{H}_1, \mathcal{H}_0)) < \alpha$. The significance level α corresponds to the probability of falsely rejecting $\mathcal{H}_0$ when it should have been accepted. We note that the mean and variance of the χ^2_{df} distribution increase linearly with *df* (see Box 3), and the threshold for rejecting the null model is increasing for increasing *df*.

Comparing our definition of LRTs (above) and confidence regions (Box 19) reveals that a null model is rejected at the level α if and only if its parameters are not within the (1-α)-confidence region around the maximum likelihood estimates. For more details see Section 7.4.1.

Rejecting a null model $\mathcal{H}_0$ does not imply that the general model $\mathcal{H}_1$ is a good model for explaining the data. In fact, when testing $\mathcal{H}_0$ against $\mathcal{H}_1$, we can obtain very small $p-$values simply because of the fact that $\mathcal{H}_0$ is a very bad null model, but not because of the fact that $\mathcal{H}_1$ is a good model. Addressing overall fit of a model has to be done with procedures as e.g. in Section 7.1.

Example: rolling a die

We again use the simple die rolling experiment as introduced in Box 18 to illustrate the concept of likelihood ratio tests. Here, we test if we reject the null model $\mathcal{H}_0$ of the die being fair when tested against the alternative model $\mathcal{H}_1$ of a loaded die. As above, we are only considering the probability of rolling a 6. In model $\mathcal{H}_0$, this probability is known and equal to 1/6, so $\mathcal{H}_0$ has no free parameters (and the likelihood L_0 is constant for a given dataset). In model $\mathcal{H}_1$, this probability is equal to θ_1, the parameter of the model. We saw in Box 18 that, given we roll the die n times and obtain a six k times, the maximum likelihood estimate for the probability of rolling a 6 under model $\mathcal{H}_1$ is $\hat{\theta}_1 = k/n$ and that $L_1(\theta) = \binom{n}{k}\theta^k(1-\theta)^{n-k}$. Further, $L_0 = \binom{n}{k}(\frac{1}{6})^k(\frac{5}{6})^{n-k}$. We can then calculate $LR(\mathcal{H}_1, \mathcal{H}_0) = 2(\log L_1(\hat{\theta}_1 = k/n) - \log L_0)$, and check if the value is in the α-tail of the χ^2_1 distribution. For $\alpha = 0.05$, the value is in the α-tail if $LR(\mathcal{H}_1, \mathcal{H}_0) > 3.84$. Thus 3.84 is the rejection threshold. The

value being in the α-tail means that we reject the null model of our die being fair. Otherwise, we do not reject $\mathcal{H}_0$, the null model that the unknown die is fair.

First, for the die rolling experiment performed in Box 18, where the die was rolled $n = 100$ times and a six was obtaine $k = 40$ times, we obtain $LR(\mathcal{H}_1, \mathcal{H}_0) = 30.62$. The corresponding p-value under the null model, i.e. the χ^2_1 distribution, is 3×10^{-8}, which means that the LRT leads to a rejection of the null model, namely the die being fair. Recall that we reject $\mathcal{H}_0$ precisely when the confidence interval does not contain the 1/6. Thus, it also follows from Box 19 that the null model is rejected for the performed experiment.

Second, we illustrate, using a die rolling experiment, that the log likelihood ratio function is indeed well-approximated by a χ^2 distribution. We roll a fair die $n = 1000$ times and record k, the number of throws resulting in a 6. We repeat this experiment 10000 times and obtain the histogram for $LR(\mathcal{H}_1, \mathcal{H}_0)$, shown in Figure 7.1 (red). The experimental histogram corresponds very well to the $\chi^2_{df=1}$ distribution, as we can see in Figure 7.1 (black). In this figure we also plot the 95-th percentile (i.e. the boundary of the 0.05-tail) of the distribution (blue), which is the value C such that $p_{\chi^2_{df=1}}(x > C) = 0.05$ (thus only 5% of the distribution fall to the right of this blue line). Here $df = 1$, thus we have $C = 3.84$.

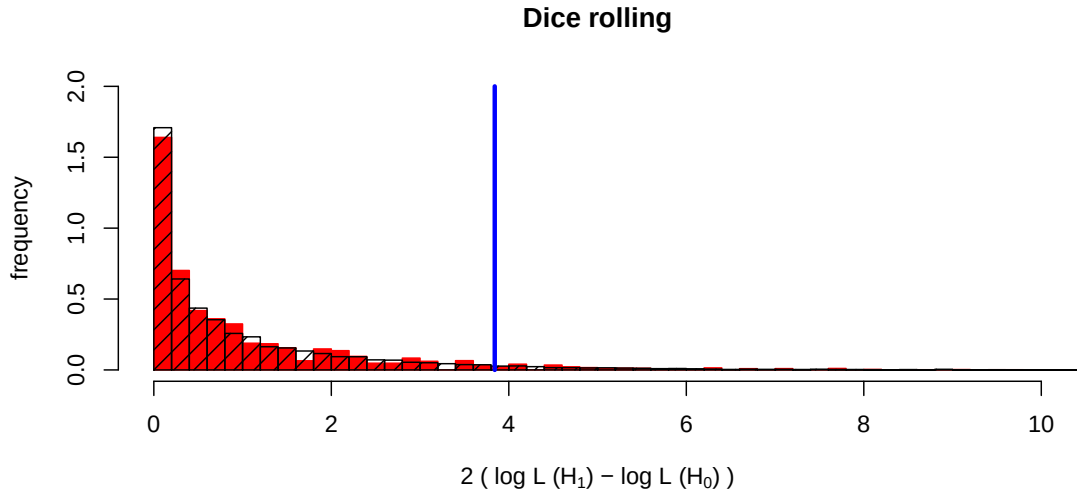


Figure 7.1: Histogram of $LR(\mathcal{H}_1, \mathcal{H}_0)$-values obtained with a fair die (red), i.e. when $\mathcal{H}_0$ is the true model. The χ^2_1 distribution is shown in black and the boundary of the 0.05-tail of the distribution is indicated in blue.

Example from phylogenetics

For a fixed tree and given sequence alignment, we can calculate the maximum likelihood estimates for different substitution models, e.g. the JC69 and the K80 model. We then determine the log likelihood ratio and reject JC69 at the 0.05 level if the

log likelihood ratio is bigger than 3.84 since $df = 1$ (see the overview of model parameters of nucleotide substitution models in Table 5.1).

The LRT only considers two models, however in many phylogenetic situations, there are many more candidate models to select from, such as JC69, K80, HKY and GTR. If we want to choose an appropriate substitution model for a given sequence alignment on a fixed tree, we may need to perform many successive LRTs. An example of a scheme for such model selection is shown in Figure 7.2. One caveat of such a model selection scheme is that it involves multiple tests. Correcting for multiple tests can be done easily only if the number of tests is known in advance. This is however not the case in the example here. Another caveat with this specific example is that some models may not be tested at all, depending on the previous tests: for instance if the very first test (JC69 vs F81 [1]) rejects $\mathcal{H}_0$, then we will proceed directly to testing F81 against HKY and the K80 model will not even be considered.

7.2.2 Errors in statistical testing

When performing likelihood ratio tests, it is important to distinguish between two types of error, shown in Table 7.1. A type I error occurs if the $\mathcal{H}_0$ model is true but we falsely reject it. A type II error occurs if $\mathcal{H}_1$ is true, but we fail to reject $\mathcal{H}_0$. The accuracy and power of a test depend on type I and type II errors, respectively. Accuracy is defined as $(1 - (\text{Type I error}))$ and power as $(1 - (\text{Type II error}))$. There is a trade-off between the two, as increasing accuracy will decrease power and vice-versa.

	$\mathcal{H}_0$ true	$\mathcal{H}_0$ false
reject $\mathcal{H}_0$	Type I error	Correct
accept $\mathcal{H}_0$	Correct	Type II error

Table 7.1: Possible errors in statistical tests.

The definition of the type I and type II error not only holds for likelihood ratio tests, but also holds for other statistical tests where we have a null model (or null hypothesis) $\mathcal{H}_0$ and a general or alternative model (or alternative hypothesis) $\mathcal{H}_1$. A general statistical test is defined through a *test statistic*, which typically is a function that transforms the data into real numbers, depending on $\mathcal{H}_0$ and $\mathcal{H}_1$. In case of likelihood ratio tests, the test statistic is the log likelihood ratio.

[1] F81 is an extension of the JC69 model in which the equilibrium frequencies are allowed to deviate from 0.25 [**Felsenstein1981**]. It has the rate matrix:

$$\mathrm{Q}_{F81} = \begin{array}{c} \\ T \\ C \\ A \\ G \end{array} \begin{array}{c} \begin{array}{cccc} T & C & A & G \end{array} \\ \begin{pmatrix} -(\pi_C + \pi_A + \pi_G) & \pi_C & \pi_A & \pi_G \\ \pi_T & -(\pi_T + \pi_A + \pi_G) & \pi_A & \pi_G \\ \pi_T & \pi_C & -(\pi_T + \pi_C + \pi_G) & \pi_G \\ \pi_T & \pi_C & \pi_A & -(\pi_T + \pi_C + \pi_A) \end{pmatrix} \end{array}$$

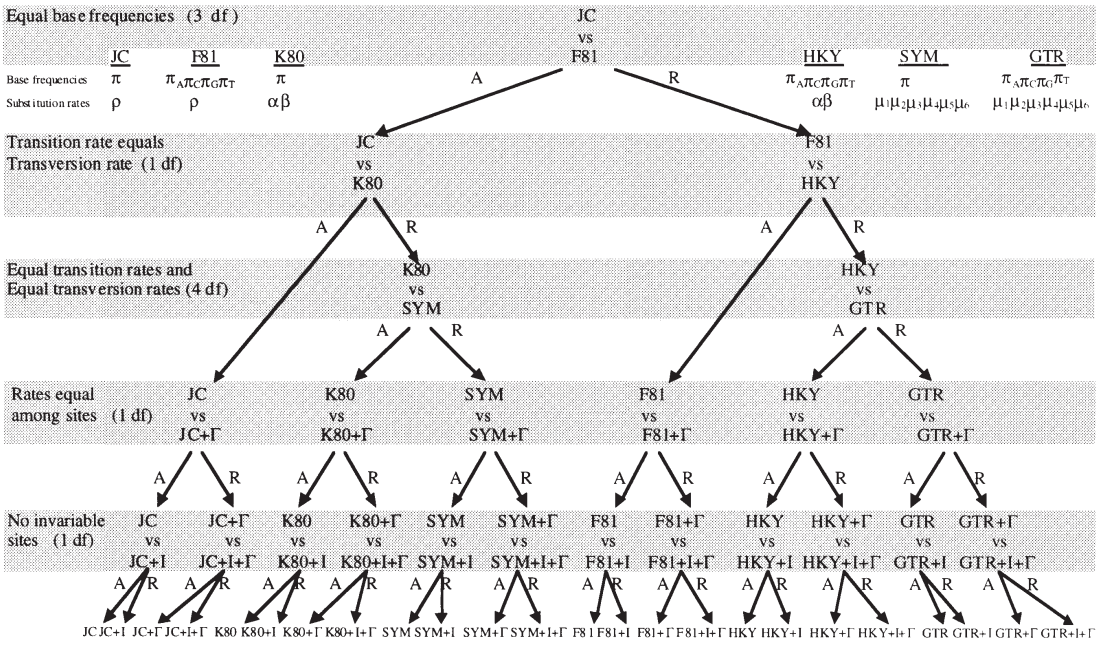


Figure 7.2: Decision tree showing the successive LRTs performed by the software ModelTest to identify an appropriate substitution model. The first test is shown at the top, and the possible outcome of each test is shown as A (accept $\mathcal{H}_0$) or R (reject $\mathcal{H}_0$). Figure adapted from [**Posada1998**].

For a given statistical test, the user decides on the level of significance as introduced in Box 1 ($\alpha = 0.05$ is often used). The significance level is equal to the type I error. Then the rejection threshold of the test is determined. Often, this calculation can be done analytically, i.e. without simulations (see e.g. the rejection threshold calculation in case of multiple testing in Section 4.2, or the rejection threshold of 3.84 for LRTs with 1 degree of freedom at $\alpha = 0.05$ as explained above). If the test statistic calculated based on the data is above (or below, pending on the test) the rejection threshold, we will reject $\mathcal{H}_0$ for the given data.

To determine the power of a test, often simulations under $\mathcal{H}_1$ are required. In general, the power increases with the difference between model $\mathcal{H}_1$ and the null model $\mathcal{H}_0$.

Example: rolling a die

We illustrate the concept of statistical errors, accuracy, and power again on a die rolling experiment. These results are based on our particular realisation of the experiment and will slightly deviate for each experiment, as the outcome of each experiment is random.

First, we experiment with a die which is actually fair. This experiment will determine the accuracy and the type I error. After performing 10000 experiments, each experiment consisting of 1000 die rolls and recording the number of times we roll a

6 (as above), we find that $\mathcal{H}_0$ is rejected in 5.1% of the experiments. This simply confirms that the significance level and the type I error we have chosen is 0.05, and thus the accuracy is 0.95.

Next we assess the power and the type II error of the test. We assume that the die is unfair meaning the null model is wrong. First, we in particular assume $\theta = 1/5$. Performing the same experiments as before, we end up correctly rejecting $\mathcal{H}_0$ in 78% of the experiments. Thus, the power is estimated to be 0.78 and the type II error to be 0.22.

Lastly, we consider another unfair die, for which $\theta = 1/2$. In all 10000 experiments we correctly reject $\mathcal{H}_0$. The power is therefore 1 and the type II error 0.

7.3 Comparing models $\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2, \ldots$: the Akaike Information Criterion

As seen in the previous section, the likelihood ratio test can only be used for two models, where one is nested within the other. In order to compare more models which are not nested, we can use the Akaike Information Criterion (AIC) [**Akaike1974**]. The AIC is a test statistic based on which the models are ranked according to how well they fit the data.

The AIC of a particular model i is:

$$AIC_i = -2 \log L_i(\hat{\theta}_i) + 2p_i \tag{7.1}$$

where p_i is the number of free parameters of the model, L_i is the likelihood function under the model, and $\hat{\theta}_i$ is the maximum likelihood estimate of the parameters of the model. The AIC needs to be calculated separately for each of the models we want to compare. The model with the lowest AIC is the model which fits the data best. The AIC is related to the expected Kullback-Leibler distance, i.e. AIC aims to measure the loss of information compared to the true (unknown) model.

When comparing different AIC values, it is important to note that the absolute number is not informative. Only the difference between the AIC values is informative. The difference represents the difference in loss of information compared to the (unknown) true model. The model with the minimum AIC value is picked. It is important to point out that again we do not know how good this best model explains the data overall, we only know that it explains the data better than any of the other considered models. Models that have an AIC within 1-2 of the minimum also have substantial support. Models with an AIC within about 4-7 of the minimum have considerably less support, while models with their AIC>10 above the minimum have essentially no support [**burnham2003model**].

Example: substitution model selection

The AIC has been used in many studies to perform substitution model selection. As an example, we use a sequence alignment of 12 plant species and use the AIC to determine which substitution model fits this alignment best. The maximum likelihood tree reconstructed using an HKY+Γ_5 model is shown in Figure 7.3 [2]. In this example, the tree topology is fixed to the topology shown in the figure, but the branch lengths are estimated separately for each of the models.

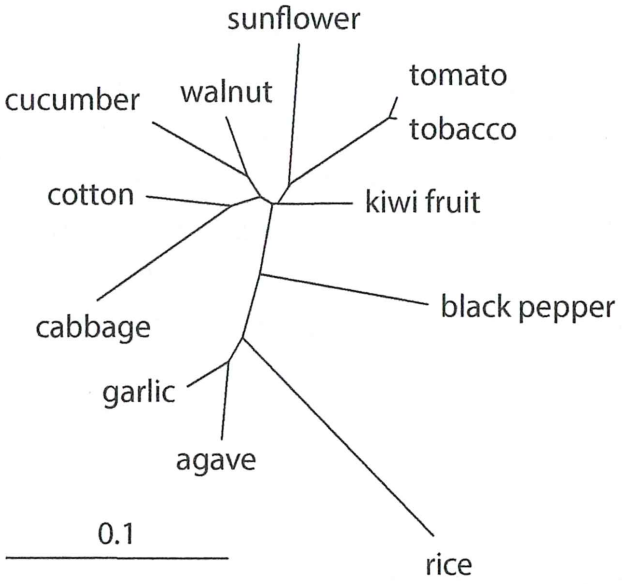


Figure 7.3: Maximum likelihood tree (plastid rbcL genes) from 12 plant species, estimated under HKY85+Γ_5. Figure adapted from [**Yang2014**].

To determine which substitution model best fits the data, we need to calculate the maximum log likelihood l of each model. The results are shown in Figure 7.4.

The number of parameters is calculated as follows: there are $2n-3$ branches in a tree with n tips (Section 6.2.3.1), so for the tree of 12 species there are $2 \times 12 - 3 = 21$ free parameters, in addition to the parameters of the substitution model. However, the substitution rate matrix and the branch lengths are correlated for trees where all tips are sampled at the same time: multiplying all rates by a factor k and dividing all branch lengths by the same factor k will give the same log likelihood value (see e.g. Section 5.2.4). Thus, we fix say the average substitution rate to 1. As a consequence, for e.g. the JC69 model, we have $p = 21$.

In this example, the tree topology was fixed. To use the LRT, both the topology and the branch lengths need to be fixed, otherwise the models are not nested. The AIC on the other hand can be used to compare models even if the tree topology is not fixed.

[2] The "Γ_5" refers to a practical improvement to the HKY substitution model that allows for substitution rate variation among sites.

Model	p	ℓ	MLEs
JC69	21	−6,262.01	
K80	22	−6,113.86	$\hat{\kappa} = 3.561$
HKY85	25	−6,101.76	$\hat{\kappa} = 3.620$
JC69 + Γ_5	22	−5,937.80	$\hat{\alpha} = 0.182$
K80 + Γ_5	23	−5,775.40	$\hat{\kappa} = 4.191, \hat{\alpha} = 0.175$
HKY85 + Γ_5	26	−5,764.26	$\hat{\kappa} = 4.296, \hat{\alpha} = 0.175$
JC69 + C	23	−5,922.76	$r_1 : \hat{r}_2 : \hat{r}_3 = 1 : 0.556 : 5.405$
K80 + C	26	−5,728.76	$\hat{\kappa}_1 = 1.584, \hat{\kappa}_2 = 0.706, \hat{\kappa}_3 = 5.651,$ $r_1 : \hat{r}_2 : \hat{r}_3 = 1 : 0.556 : 5.611$
HKY85 + C	35	−5,624.70	$\hat{\kappa}_1 = 1.454, \hat{\kappa}_2 = 0.721, \hat{\kappa}_3 = 6.845$ $r_1 : \hat{r}_2 : \hat{r}_3 = 1 : 0.555 : 5.774$

Figure 7.4: Number of parameters p, maximum log likelihood values l, and maximum likelihood estimates for different substitution models. The rates and branch lengths were optimized on a fixed topology of the 12 species displayed in Figure 7.3. Figure adapted from [**Yang2014**].

Substitution model selection using LRT and AIC has been automated in software tools such as ModelTest [**Posada1998**] and its new version jModelTest [**Posada2008**, **darriba2012jmodeltest**]. Both applications can test dozens of models on a fixed tree or when co-estimating trees.

7.4 Assessing uncertainty in estimates

After having selected the best model for a given dataset, we may also want to estimate the uncertainty associated with the maximum likelihood estimates for the parameters under this model, i.e. we want to know how wide the range of "likely" values for each parameter is. This range is called a confidence interval or confidence region. It is formally defined in Box 19.

We illustrate the confidence intervals again with our die rolls example. We perform one die roll experiment consisting of 1000 die rolls and record the number of times we roll a 6. We recorded 165 times a six, meaning the maximum likelihood estimate $\hat{\theta}$ for the probability of rolling a six is 0.165. Figure 7.5, black line, shows the likelihood curve for the probability of rolling a six, θ, together with the maximum likelihood estimate (red solid line) and the true probability of 1/6 (gray solid line).

In what follows, we provide four procedures for obtaining an estimate for the confidence interval. All procedures are illustrated with a die rolling experiment. We note that for many biological problems, we cannot freely choose between any of the four procedures, but pending on the models and the datasets, we may be limited to one particular procedure.

7.4.1 Calculating confidence intervals using the LRT

In Box 19, we described how to construct confidence intervals for maximum likelihood estimators. After having discussed the likelihood ratio tests, we note that the likelihood confidence intervals are a special case of the likelihood ratio tests. Model $\mathcal{H}_0$ corresponds to the true parameter, and the MLE for the parameter given some observed data is obtained under some more general model $\mathcal{H}_1$. The $(1-\alpha)\times 100\%$ confidence interval consists of parameters being candidates for the unknown true parameter. Assuming a candidate parameter is the true unknown parameter, model $\mathcal{H}_0$ is not rejected when performing the likelihood ratio test $LR(\mathcal{H}_1, \mathcal{H}_0)$. We show the confidence interval based on a LRT for the die rolls example in Figure 7.5 with an orange dashed line.

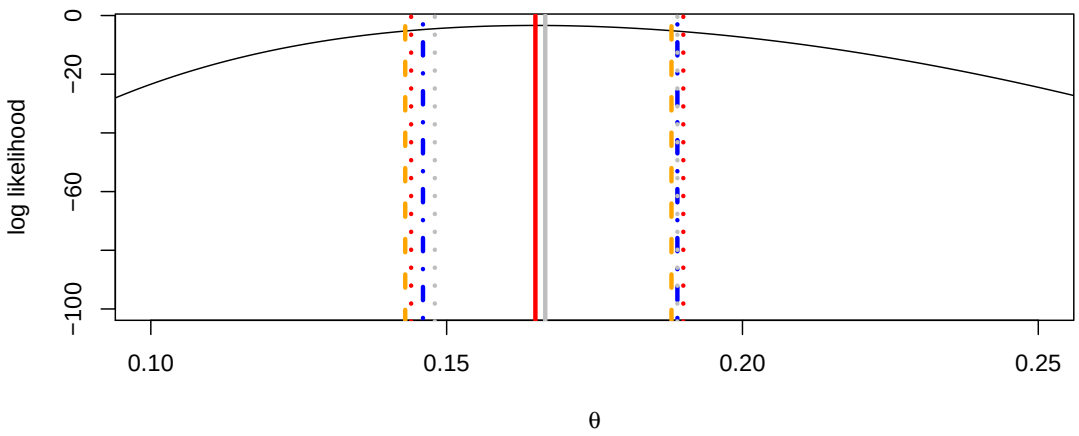


Figure 7.5: A fair die is rolled 1000 times and a six was recorded 165 times. The probability of rolling a six, θ, is estimated. The black curve is the likelihood curve for θ. The maximum likelihood estimate for θ is shown in red (solid line) and the true probability of 1/6 is shown in gray (solid line). Further, 95% confidence intervals obtained using the LRT (orange), the repetition of experiments (gray, dotted), non-parametric bootstrapping (blue), and parametric bootstrapping (red, dotted) are shown.

Most of our substitution models contain several parameters, thus we obtain a confidence region. This region is easy to characterize if it is e.g. a circle in the two-dimensional case with the maximum likelihood estimate in the center. However, often it is very different from a circle. To facilitate confidence calculations, often *profile likelihoods* are considered. For calculating the profile likelihood, all parameters except one are fixed to their maximum likelihood estimates. We can then calculate the profile confidence interval for the un-fixed parameter in the same way as in the one-parameter case. Note that for the die experiment, we have only one parameter, thus the profile confidence interval equals the confidence interval based on the log likelihood ratio.

LRT-based (profile) confidence intervals are straightforward to calculate when being provided a fixed phylogenetic tree for which a substitution model was chosen. For the given substitution model, the (profile) confidence interval for the parameter can be obtained using the LRT. In Section 5.3.3.1, we calculated LRT-based confidence intervals for pairwise distances between sequences under the JC69 model.

7.4.2 Obtaining confidence intervals by redoing experiments

If we have access to the experimental system that was used to produce the dataset, we can use this to repeat the experiment multiple times. Each experiment will produce a different maximum likelihood estimate for the parameter and, based on the definition of confidence intervals in Box 19, the 95% confidence interval of the parameter is then obtained by discarding the bottom 2.5% and the top 2.5% estimates. This procedure can for instance be used with our example of an unknown die, with a bacterial evolution experiment, etc. For the die experiment, Figure 7.5 shows the 95% confidence interval obtained via 100 repetitions of the experiment with a gray dotted line.

We note that the LRT-based and redoing experiment-based confidence intervals are not identical in Figure 7.5. This is due to the LRTs relying on the χ^2 approximation. Further, the experiments need to be re-done infinitely amount of times while we only did 100 repetitions. Thus both intervals only approximate the true 95% confidence interval.

In most biological applications, redoing experiments to obtain the confidence interval is not feasible. However, we introduce this framework here as it leads to the exact confidence intervals (if enough replicates are generated), and the non-parametric bootstrapping approach in the next section approximates the procedure of re-doing experiments.

7.4.3 Obtaining confidence intervals by non-parametric bootstrapping

In general, it is impossible to re-run real life experiments such as the evolution of mammals. However, we can mimic the repetition of experiments through *bootstrapping*. Under non-parametric bootstrapping, the procedure consists of randomly choosing samples from our dataset with replacement until we get a second dataset of the same size as the original. It is important that the sampling is carried out with replacement, as otherwise we simply recover the original dataset. Resampling the dataset multiple times and obtaining the maximum likelihood estimate of the parameters for each bootstrap dataset will allow us to construct a 95% confidence interval by ignoring the 2.5% smallest and largest maximum likelihood estimates.

For our die rolling experiment, the 95% non-parametric bootstrapping confidence interval is obtained by considering our initial experiment where we obtained 165 times a six when rolling the die 1000 times. To obtain a bootstrap dataset, we sample another 1000 die rolling results by sampling from the 1000 results of the initial

experiment with replacement. We then obtain the maximum likelihood estimate for the probability of rolling a six for this bootstrap dataset. In Figure 7.5, we show the 95% non-parametric bootstrapping confidence interval with a blue dot-dashed line based on 100 bootstrap datasets. It is straightforward to see that if the original dataset was big enough, this procedure leads to a confidence interval indistinguishable from the one we obtain if we had several independent experiments. Note though that if the dataset is small, the bootstrapping underestimates the size of the true confidence interval.

In Section 7.4.5, we will use non-parametric bootstrapping for obtaining confidence measures on tree topologies.

7.4.4 Obtaining confidence intervals by parametric bootstrapping

Finally, we can estimate confidence intervals by parametric bootstrapping. Here, additional datasets are simulated, under the model $\mathcal{H}_1$ with parameters $\hat{\theta}$, i.e. the maximum likelihood estimate for the dataset. For these additional datasets, the parametric bootstrap datasets, again maximum likelihood parameters are estimated and the 95% parametric bootstrapping confidence interval is obtained, as above, by ignoring the 2.5% smallest and largest maximum likelihood estimates.

For our die rolling experiment, we obtained 100 parametric bootstrap datasets by performing our experiment with a die where the probability of throwing a six is 0.165. The result is shown in Figure 7.5, red dotted line. For our die, the confidence interval based on parametric bootstrapping would be equivalent to the confidence interval based on redoing experiments given the maximum likelihood estimate equals the true parameter.

Parametric bootstrapping will be used for phylodynamic parameter inference in Section 9.1.6.2.

7.4.5 Tree uncertainty estimation

Estimating the uncertainty in reconstructed trees is a unique challenge, as trees are complex objects with a discrete component, namely the topology. The LRT method cannot be used as assuming different trees (with respect to branch length and/or topology) corresponds to non-nested models (see Section 7.3 for an example). Non-parametric bootstrapping however provides a very useful tool for estimating the uncertainty in a tree.

The procedure (first introduced in [**Felsenstein1985**]) is as follows: we obtain a bootstrap alignment by sampling m sites at random with replacement from the original alignment of m sites (=columns, representing positions in the alignment), as shown in Figure 7.6. Note that the order of the rows, which represent different samples/organisms, is kept unchanged during the bootstrap procedure. From each newly created bootstrap alignment, we reconstruct a maximum likelihood bootstrap

tree. If the original alignment was long enough, the resulting set of bootstrap alignments shows approximately the same variation as would be obtained from repeating the evolution of those organisms multiple times.

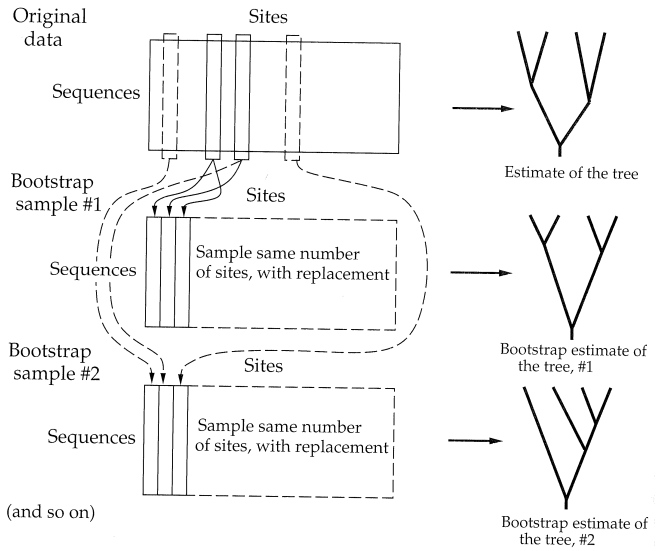


Figure 7.6: Bootstrap for phylogenies. Figure adapted from [**Felsenstein2004**].

The set of bootstrap phylogenies is compared to the original maximum likelihood tree to obtain a measure of uncertainty. A common comparison criterion is the following: for each node in the maximum likelihood tree, we count how many trees have an internal node whose set of descending tips is exactly the same as for this node. In other words, we count how many bootstrap trees contain a particular clade, for each clade in the maximum likelihood tree. Based on this count, we report the percentage of bootstrap trees which contain a clade in the maximum likelihood tree. This is called *the bootstrap support* of each particular internal node. We can see an example of the result of bootstrapping on an SIV-HIV phylogeny in Figure 7.7. Each node is marked with its bootstrap support.

7.5 Summary of maximum likelihood tree inference

In summary, the recommended procedure for reconstructing a phylogenetic tree and getting the estimates of all model parameters from a sequence alignment in the maximum likelihood framework is as follows:

- Step 1: for each substitution model, infer a maximum likelihood tree with Felsenstein's pruning algorithm and choose the tree with the topology and the branch lengths which maximize the likelihood.
- Step 2: determine the tree and substitution model with highest support using AIC.

- Step 3: determine the confidence intervals for the substitution model parameters using the likelihood ratio test.
- Step 4: determine the confidence in the maximum likelihood tree by using non-parametric bootstrapping and employing the substitution model identified in step 2 for the tree reconstruction.

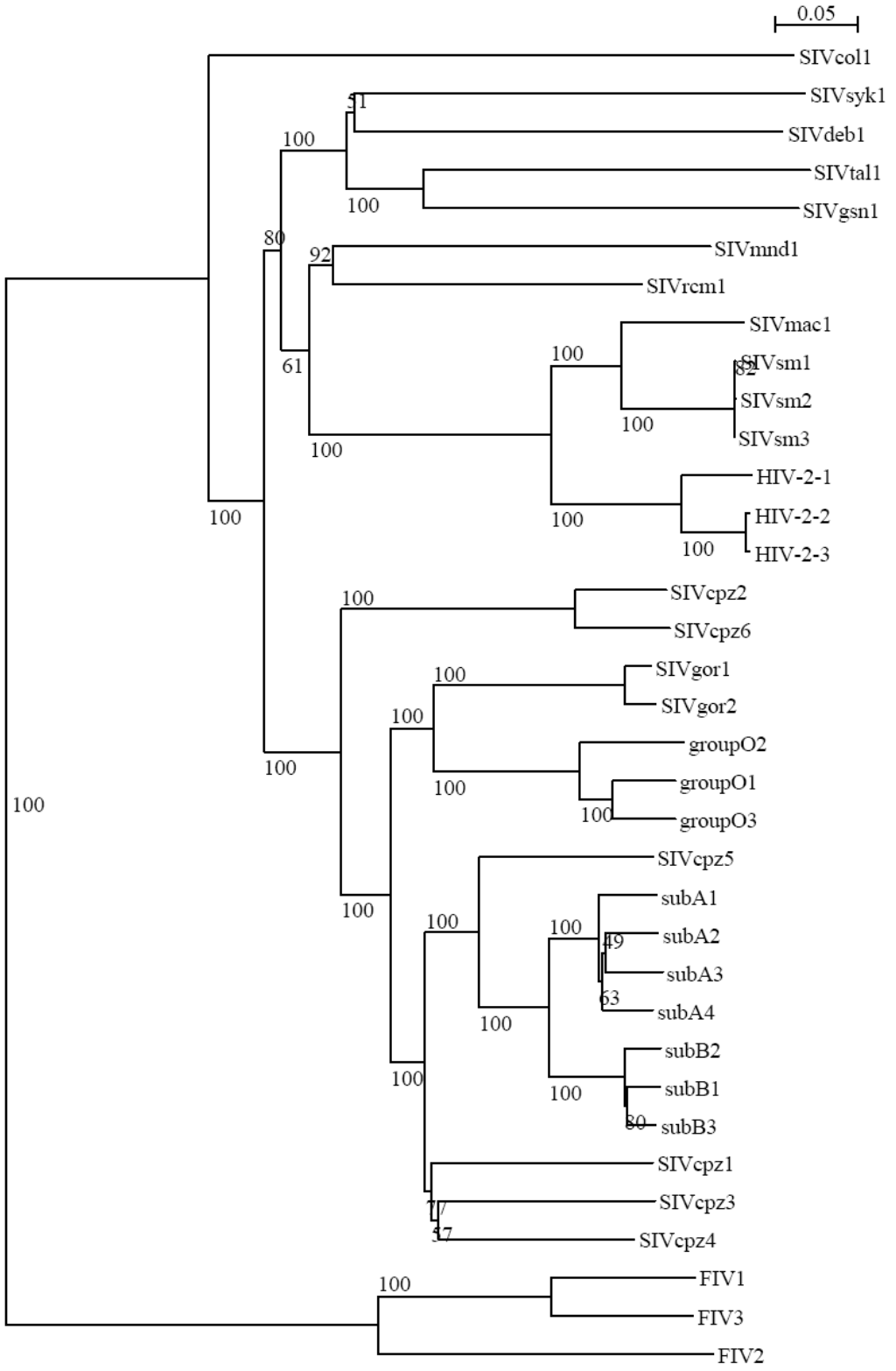


Figure 7.7: Maximum likelihood tree of SIV-HIV sequences with bootstrap support marked for each internal node.

8 Traits and comparative methods

In this chapter, we assume that we know the phylogenetic tree. We will explore the processes of phenotypic evolution occurring on this given phylogenetic tree. Such analyses require that we measure phenotypic traits for the samples in the tree. The traits are encoded in characters. Characters could either be *discrete*, e.g. spike numbers of HIV virions, leg numbers of arthropods, or presence of fur pattern in rodents, or they could be *continuous*, e.g. height, weight, virulence, or dinosaur jaw shape.

We will learn to compare the traits encoded in characters across the individuals being represented in the phylogenetic tree. Recall that in Chapter 4, we considered a trait (such as healthy / diseased) across individuals where the sites in the genome were assumed to be distant enough to be completely unlinked. As a consequence, it was appropriate to assume that each individual is an independent data point. Now we consider traits for individuals where the sites in the genome are fully linked, and thus their relationship are provided in a phylogenetic tree. This means that when we analyze the character data, we cannot assume independence of individuals any more, but must assume correlations given by the underlying phylogenetic tree. We will now describe comparative methods used for discrete and continuous trait comparison given phylogenetic trees, and illustrate them with examples.

8.1 Comparing discrete characters on a phylogeny

We want to find out whether there is a correlation between hair colour and eye colour in a group of ten individuals shown in Figure 8.1. While the sample shows only two combinations (red hair and blue eyes, black hair and brown eyes), the two other combinations (red hair and brown eyes, black hair and blue eyes) might also be possible, even though they are not present in our data sample. From looking at the data points, one is tempted to conclude that there is a strong correlation between the two characters, that is, if an individual has red hair, it will also most likely have blue eyes.

8.1.1 Assuming independence across individuals

In order to show that there is a significant correlation, we have to perform a statistical test such as the Fisher's exact test (see Box 11). Recall that before performing any type of statistical test, we need to formulate the null hypothesis. The null-hypothesis in our case is:

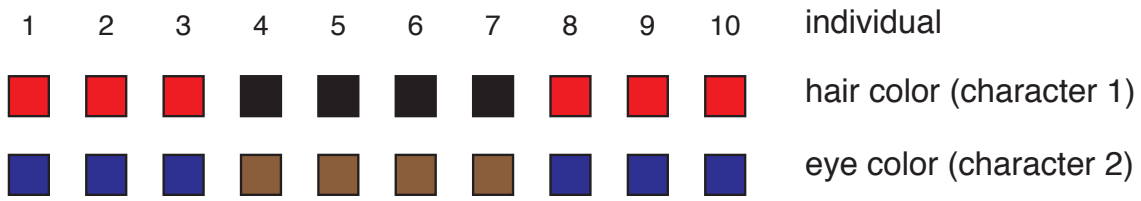


Figure 8.1: The combinations of discrete characters – hair and eye colours – for the ten sampled individuals.

$\mathcal{H}_0$: Having brown eyes is equally likely among red- and black-haired individuals.

For the Fisher's exact test, we fill out a contingency table with our observations (see Figure 8.2), and we can then formulate the null hypothesis in a mathematical correct way:

$\mathcal{H}_0$: The field in the contingency table "having brown eyes and red hair" with fixed column and row sums follows a hypergeometric distribution.

The hypergeometric distribution describes an urn experiment, and in our situation it represents the number of red balls obtained when drawing 4 balls without replacement (representing the individuals with brown eyes) from an urn with 6 red balls (representing red haired individuals) and 4 black balls (representing the black haired individuals). We denote the number of red balls among the drawn 4 balls with the random variable R. We calculate the p-value of the observed outcome, r, which is the probability to obtain the observed outcome or a more extreme outcome. Here, we define more extreme as obtaining fewer individuals with brown eyes and red hair than observed. We set the significance level to 0.05. If the obtained p-value is less than the significance level, we reject the null hypothesis, as it is very unlikely that we could get such data under the null model.

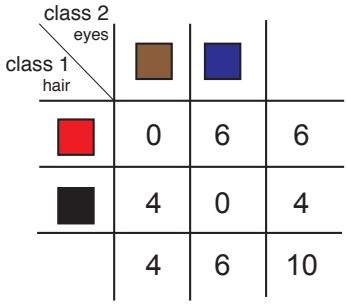


Figure 8.2: The contingency table for the hair and eye colours. No individuals in the sample have red hair and brown eyes or black hair and blue eyes, 6 individuals have blue eyes and red hair, and 4 individuals have brown eyes and black hair.

In the above example we observe 0 individuals with red hair and brown eyes. Let us now compute the probability of this event under hypothesis $\mathcal{H}_0$ by following Fisher's

exact test (Box 11):

$$P(R = 0|\mathcal{H}_0) = \frac{\binom{6}{0}\binom{4}{4}}{\binom{10}{4}} = 0.0048$$

As we cannot observe less red balls than 0, the p-value is 0.0048, and thus well below the pre-defined significance level of 0.05. We thus reject the null hypothesis of an equal distribution of brown eyes among red- and black-haired individuals. Instead, there seems to be a significant correlation between the two features.

8.1.2 Considering phylogenetic relatedness

The result from the section above seems sane. However, our analysis relies on the assumption that the individuals in our samples are independent from each other. Assume that our ten individuals are related through the evolutionary history shown in Figure 8.3. The history of trait evolution of hair color and eye color is depicted by the colours along the branches.

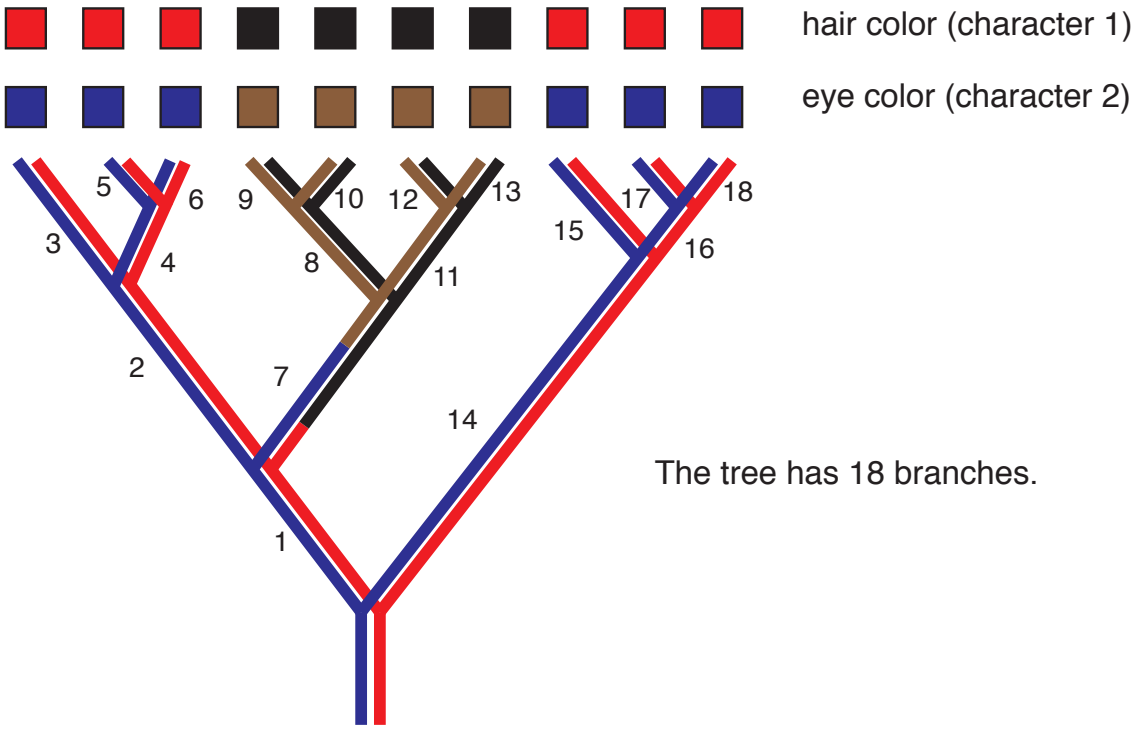


Figure 8.3: A phylogenetic tree connecting the ten sampled individuals. The tree contains 18 branches which are labelled with numbers. The two branch colours represent the character states along the tree. The individual at the root of the tree thus has blue eyes and red hair, and so do six of its descendants. In this particular example, both the change in hair and eye colour occurred only once, and both changes took place at some time along branch 7.

Now, when correcting for this relatedness, instead of looking at the traits at the tips,

we look at which changes happened on the branches of the phylogenetic tree. Suppose again we model our changes using an urn experiment. The red balls represent the branches where hair colour changes (here, 1 red ball), and the black balls represent the branches where hair colour did not change (here, 17 black balls). Now we draw k balls without replacement to determine the branch on which the change of eye colour occurred (here, $k = 1$). The number of red balls drawn follows the hypergeometric distribution (Box 4).

Thus, we change our null hypothesis to:

$\mathcal{H}_0$: The number of branches with a change in both hair color and eye color follows a hypergeometric distribution.

We summarize our example data in the contingency table shown in Figure 8.4.

eyes / hair	yes	no
yes	1	0
no	0	17

Figure 8.4: The contingency table for changes per branch. In this example there are no branches with a single change, one branch with changes in both characters, and 17 branches with no changes.

The probability of the data under the null hypothesis is:

$$\begin{aligned} P(\text{for one branch, both hair and eye color change}|\mathcal{H}_0) &= \frac{\binom{1}{1}\binom{17}{0}}{\binom{18}{1}} \\ &= 0.05555. \end{aligned}$$

The p-value is obtained by summing over the probability of the data, and the probability of any more extreme result. More extreme corresponds to more than one branch having both a hair and eye color change. As we only have one change of each trait in total, the p-value is equal to 0.05555. As the p-value is higher than the significance threshold 0.05, we do not reject the null hypothesis of the changes being equally likely on every branch and thus are not correlated.

This example shows that neglecting the phylogenetic relatedness of individuals can lead to false conclusions about correlations between characters.

8.1.3 Other methods for detecting discrete character correlations

Changes are more likely to occur on longer branches rather than on short branches. However, in the approach described in the previous section we have ignored these differences in branch lengths completely. Several methods correct for that, such as Ridley [**Ridley1983**], who used parsimony , and Pagel [**Pagel1994**], who used likelihood methods. Another key advantage of these methods is that they do not assume knowledge of the ancestral character states. For an overview of these and other approaches see [**Felsenstein2004**].

8.2 Comparing continuous characters on a phylogeny

So far, we only considered discrete characters. However, many characters are continuous, such as weight, height, transmission fitness, etc. In what follows, we aim to determine if two continuous characters are correlated. We will first explain why a linear regression on two characters that evolved on a tree is not appropriate. We then introduce the Brownian motion model to mathematically describe character evolution on a tree. Lastly, we will describe how the Brownian motion model allows us to transform the characters into variables that can be studied with a linear regression model in order to quantify correlations.

8.2.1 Assuming independence across individuals

One option to determine the significance of correlation between continuous characters is *linear regression*, which is a mathematical method to determine the dependency of a variable Y on another variable X. The discrete analog was Fisher's exact test as shown in Section 8.1.1. When using this method, it is assumed that for the observations $(x_1, y_1), \ldots, (x_n, y_n)$, the y_i depends linearly on the x_i,

$$y_i = \beta x_i + b + \epsilon \tag{8.1}$$

and the error term ϵ follows the same normal distribution for all data points, $\epsilon \sim \mathcal{N}(0, \sigma^2)$. This means, in particular, that the observations are independent from each other.

Fitting a linear regression model to the data means that we find the $\hat{\beta}$ and $\hat{b}$ which minimize the error between the measured values y_i and the predicted values $f_i := \hat{\beta} x_i + \hat{b}$. Once the model is fit to the data, we can compute the *coefficient of determination* R^2. It quantifies the amount of variance in Y which is explained by X. With the mean of the observed data $\bar{y} := \frac{1}{n} \sum_{i=1}^{n} y_i$, R^2 is defined as

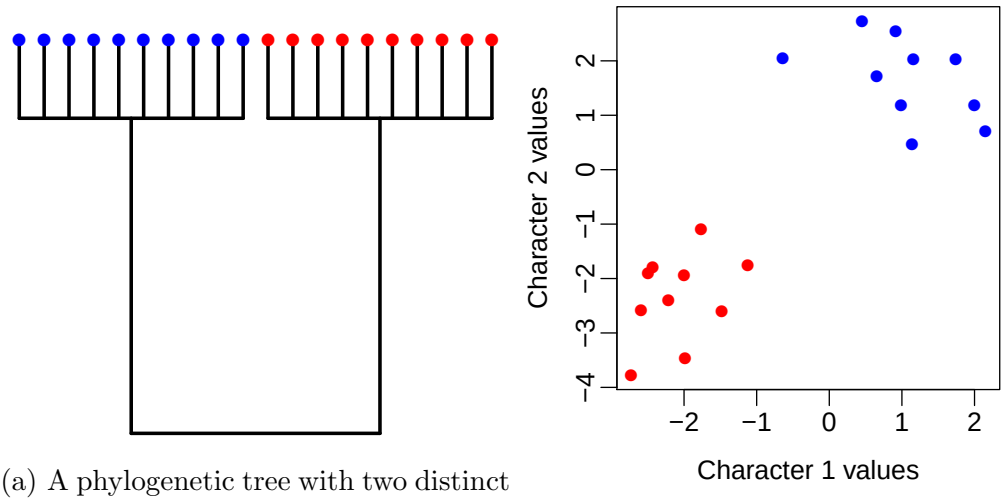
$$R^2 := 1 - \frac{\sum_{i=1}^{n} (f_i - y_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$$

Thus, a value close to 1 means that the variation in Y is perfectly explained by

the linear regression, while a value close to 0 indicates that the linear regression is unable to predict the variation in Y.

Figure 8.5b shows trait data for 20 individuals, character 1 plotted on the x-axis and character 2 on the y-axis. The R^2 for this data is 0.72, indicating a strong correlation between these two characters. The p-value is 2.4×10^{-6}, suggesting that this correlation is highly significant.

However, now assume that these individuals have a shared evolutionary history, shown in the phylogenetic tree in Figure 8.5a. The tree shows two clades (blue and red). The points in plot 8.5b are coloured according to their clade in the tree, and one sees that the red individuals seem to have a negative correlation and the blue individuals a positive correlation between the two characters. In particular, we may not perform a simple linear regression as explained above, as its assumption regarding independence of observations is not satisfied: the individuals show dependence via the phylogenetic tree. In the next section, we will define a null model for continuous trait evolution, and then we will show how to use this model for determining significant correlation between characters. For discrete characters, the analog was assuming a hypergeometric distribution, and performing a Fisher's exact test on the changes along branches.



(a) A phylogenetic tree with two distinct clades.

(b) Character trait values.

Figure 8.5: When viewed separately from the phylogenetic tree (8.5a), the character trait values (8.5b) seem to exhibit correlation, meaning the trait on the y-axis can be predicted by the trait on the x-axis. However, when one views the trait values per clade, one observes that traits can be predicted via their clade.

8.2.2 Modeling continuous trait evolution with the Brownian motion model

Box 23: Brownian motion

The Brownian motion model is named after the Scottish botanist Robert Brown, who formulated it after observing crop seed movement in water under a microscope. He noticed that the crop seeds move randomly on the water surface, due to hits from water molecules. Another way to depict this type of movement is to imagine a ball being thrown around by a crowd of people in a football stadium. The ball will be pushed around by different people, resulting in a random pattern of movement.
Brownian motion is the motion resulting from a Wiener process, which is a continuous time stochastic process. The Wiener process is called after the American mathematician Norbert Wiener (1894-1964). The Wiener process is defined as a stochastic process $(W_t)_{t\in T}$ with i.i.d random variables W_t and $T \subseteq \mathbb{R}$ that fulfils the following four conditions:

1. $W_0 = 0$: the process starts in 0;
2. W_t is almost surely continuous: $P(W_t \text{ continuous}) = 1$;
3. W_t has independent increments, which implies memorylessness: for $0 \leq s_1 \leq t_1 < s_2 \leq t_2$, $(W_{t_1} - W_{s_1})$ and $(W_{t_2} - W_{s_2})$ are independent;
4. for $0 \leq s \leq t$, the difference $W_t - W_s \sim \mathcal{N}(0, \sigma^2(t-s))$, which is a normal distribution with a variance depending on the time difference.

For further detailed mathematics on these processes please refer to [**Bertoin1998**].
We can draw an analogy between the continuous and the discrete character models that we have discussed before. While discrete processes have a probability to visit any state in the set of possible states, a continuous process has a probability density distribution on the state space. A discrete process is memoryless due to the Markov property, while a continuous process is memoryless due to property 3 of the Brownian motion model. Finally, in a discrete state process, transition probabilities scale with time, while in the case of a continuous process the variance scales with time.

A popular model to describe the evolution of continuous traits along a phylogenetic tree is the Brownian motion model (Box 23). At the root of the tree, the trait is 0. In both branches descending the root, the trait evolves according to a Brownian motion, and this evolution is independent across the two branches. At a branching event, both descending individuals inherit the trait value from the ancestor, X_a, which most likely is different from 0. Afterwards, the traits again evolve under Brownian motion, with independence across branches. Mathematically speaking, the characters, X_1, X_2, at the two child branches evolve independently and are normally distributed starting in X_a, i.e. $X_i \sim \mathcal{N}(0, \sigma^2) + X_a$. Figure 8.6a shows a phylogeny of four species. Figure 8.6b shows one possible evolution of traits on this phylogeny under the Brownian motion model. Individuals C and B as well as individuals A and B share common evolutionary history, their characters evolve together from the root until the branching in the phylogenetic tree (dotted lines in Figure 8.6b). Thus we cannot assume independence between the trait values of C and D (or A and B), and we cannot perform a standard linear regression to investigate correlations between two traits.

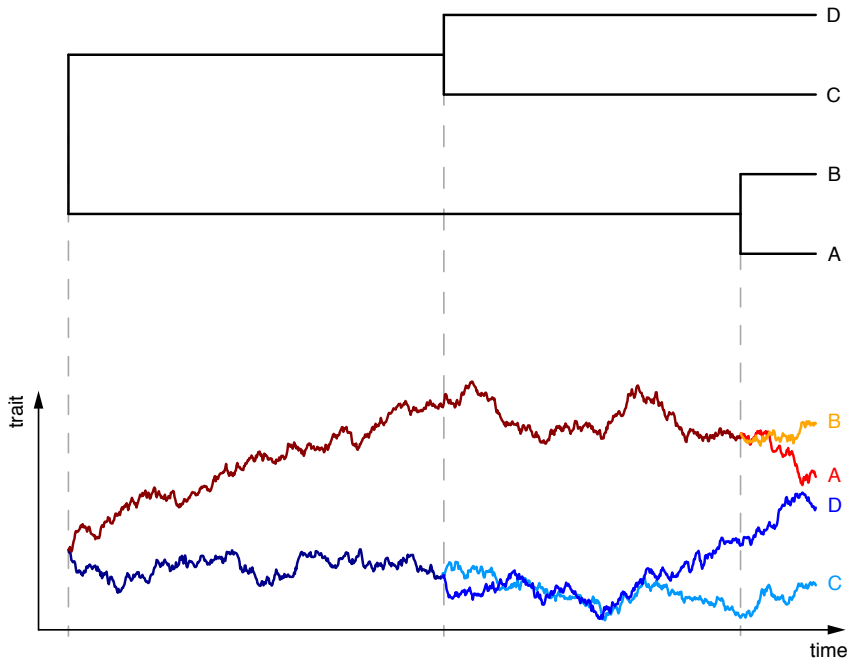


Figure 8.6: A phylogeny of four individuals (top) and one realization of trait evolution according to the Brownian motion model (bottom). As soon as a branching event happens in the phylogeny, the stochastic process modelling trait evolution splits into two separate and independent stochastic processes starting with the trait value that was reached by the ancestral species.

8.2.3 Considering phylogenetic relatedness using the contrast method

In the following, we assume that the characters on a phylogenetic tree evolve according to the Brownian motion model. One method that lets us correct for the interdependencies of evolutionary traits on trees is the *contrast method* developed by Joseph Felsenstein [**Felsenstein1985comp**]. In the contrast method, we mathematically eliminate interdependence in our data by performing linear regression not on the original data points, but on recomputed values that are independent and have identically distributed errors.The recomputed values are referred to as *contrasts*.

We will illustrate the contrast method on the example phylogeny on four tips in Figure 8.7. Imagine that for all the species at the tips, we measured n different traits (e.g. height, weight, and percentage of fat among body mass, thus $m = 3$ in this simple example). We denote the value of the observed trait j in node $k = 1, \ldots, 4$ with x_k^j for $j = 1, \ldots, m$ (here $j = 1$ could represent the height, $j = 2$ the weight, and $j = 3$ the fat percentage). Even if we cannot observe the traits in the internal nodes directly, we label these values according to the same scheme. We additionally label the branch length of the branch leading to node j with t_j. It is clear from the tree that the trait values x_1^j and x_2^j are not independent observations, as they share

evolutionary history through branches t_6 and t_5. In what follows, we use the trait values at the tips and the branch lengths to determine independent variables with identically normally distributed errors under the assumption that the traits evolved according to the Brownian motion model. To this end we will

- define independent parts of the tree, and determine transformations of the trait values at these independent parts, called *contrasts* (section 8.2.3.1)
- estimate trait values at the internal nodes based on the tip values and branch lengths (section 8.2.3.2)
- determine the variances of the contrasts (section 8.2.3.3)
- normalize the contrasts (section 8.2.3.4)

With the normalized contrasts, we can perform a linear regression while this would have not been possible with the original trait values, due to the shared genetic ancestry which leads to a violation of the assumption of independent observations with the errors that follow the same normal distribution.

As stated above, we assume that the traits evolved according to a Brownian motion model as described in chapter 8.2.2. This means, that we need to clearly distinguish between the random variables of tip values and contrasts from the observed values of these random variables. Following the notation throughout this book, we denote the random variables of traits with capital letters X_i^j and their realisations with small letters x_i^j, and the same is applied to the contrasts. In the following derivation, we will use random variables, i.e. capital letters. Only the branch lengths are considered to be fix, because the tree is independently inferred based on the sequences of the species and therefore denoted with small letters througout.

8.2.3.1 Definition of contrasts

The first aim of the contrast method is to divide the tree into independent parts. We define independent parts of the tree recursively: a cherry is an independent part as it does not have any shared evolutionary history. The cherry is replaced by a single tip, and we recursively define independent parts in the same way until the whole tree is replaced by a single tip. These independent parts are assigned each a unique colour in Figure 8.7. Each independent part is essentially a cherry, with the two tips being either a tip or internal node in the original phylogenetic tree. We characterize each independent part by the two tip nodes of the corresponding cherry. A tree with n tips will result in $n-1$ independent parts, meaning $n-1$ pairs of nodes.

The random variable of the contrast of trait j for the part between nodes k and l is defined as:

$$Z_{(k,l)}^j = X_k^j - X_l^j \tag{8.2}$$

Keep in mind that for each trait at the tips, we need to define the contrasts (above we described an example of the $m = 3$ traits height ($j = 1$), weight ($j = 2$), and fat percentage ($j = 3$)).

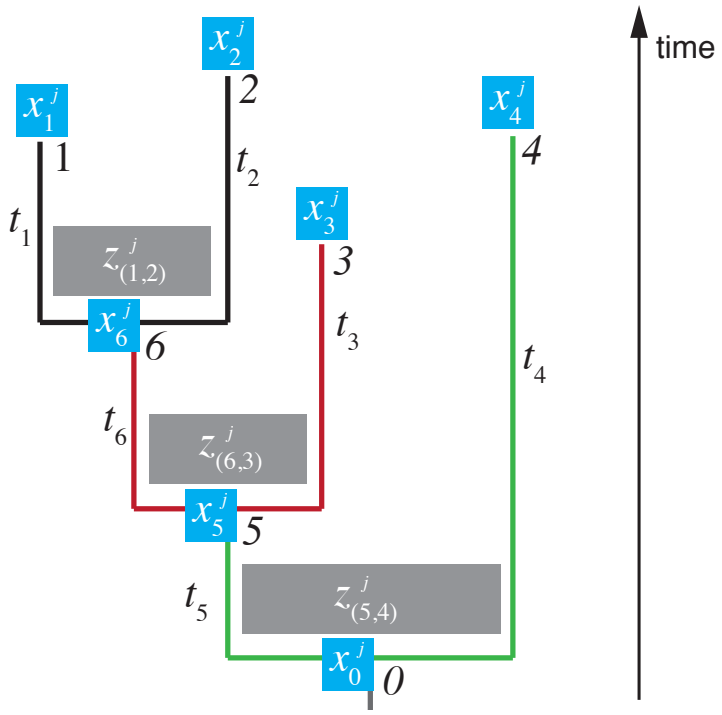


Figure 8.7: An example of a phylogenetic tree on 4 tips. The extant nodes are labelled $1, \ldots, 4$, the root 0 and the internal nodes 5, 6. The branch leading to node k is labelled t_k. The (observed) value of trait j in node k is denoted by x_k^j (blue boxes) and the contrast of trait j between nodes k and l is denoted by $z^j_{(k,l)}$ (gray boxes). The different colors of the branches correspond to independent parts of the tree.

Why are the contrasts independent? At the moment, we can intuitively see that the contrasts consider only parts in the tree that do not share common branches by definition . But we will also see a mathematical explanation below (section 8.2.3.3).

8.2.3.2 Estimating trait values at internal nodes

To estimate the trait values of internal nodes k based on the observed tip values, we will use a linear combination of the trait values at the child nodes. As the values are estimates, we denote them by $\hat{x}_k^j$. We also start at a cherry and estimate the value at the parent of this cherry. Because the values at the parent form the starting point of a Brownian motion trait evolution, child nodes that are closer to the parent will influence the trait value estimate more than more distant child nodes. Before we formally derive the estimators, we demonstrate the idea behind these estimators with an intuitive example: We aim at estimating the traits in node 6 from the traits of its child nodes 1 and 2 in Figure 8.7. As we only have information on the branch lengths and the tip values, we weigh the tip values according to:

$$\hat{x}_6^j = \frac{t_2}{t_1 + t_2} x_1^j + \frac{t_1}{t_1 + t_2} x_2^j \tag{8.3}$$

This way, x_1^j obtains a higher weight than x_2^j.

The weights in equation 8.3 arise naturally when estimating the trait values at internal nodes with a maximum likelihood estimator. To see this, we write down the probability to observe x_1^j, x_2^j given node 6 is in state x_6^j. Due to the Brownian motion model we know that the characters evolve according to a Normal distribution with mean 0 and variance $\sigma^2 \times$(branch length) starting at value x_6^j. The conditioned probability to observe these states is therefore:

$$\begin{aligned} P(X_1^j = x_1^j, X_2^j = x_2^j | X_6^j = x_6^j) &= \frac{1}{\sqrt{2\pi\sigma^2 t_1}} \exp\left(-\frac{(x_1^j - x_6^j)^2}{2\sigma^2 t_1}\right) \times \frac{1}{\sqrt{2\pi\sigma^2 t_2}} \exp\left(-\frac{(x_2^j - x_8^j)^2}{2\sigma^2 t_2}\right) \\ &= \frac{1}{2\pi\sigma^2\sqrt{t_1 t_2}} \exp\left(-\frac{1}{2\sigma^2}\left(\frac{(x_1^j - x_6^j)^2}{t_1} + \frac{(x_2^j - x_6^j)^2}{t_2}\right)\right) \end{aligned}$$

The last equal sign is just a re-write of the preceding formula. This probability can also be interpreted as a likelihood (see Box 18):

$$\begin{aligned} l(x_6^j; x_1^j, x_2^j) &= \log\left(L(x_6^j; x_1^j, x_2^j)\right) = \log\left(P(X_1^j = x_1^j, X_2^j = x_2^j | X_6^j = x_6^j)\right) \\ &= \log\left(\frac{1}{2\pi\sigma^2\sqrt{t_1 t_2}}\right) - \frac{1}{2\sigma^2}\left(\frac{(x_1^j - x_6^j)^2}{t_1} + \frac{(x_2^j - x_6^j)^2}{t_2}\right) \end{aligned}$$

To obtain the ML estimate of x_6^j we determine the maximum of the log-likelihood which requires to determine the first derivative:

$$\frac{\mathrm{d}l}{\mathrm{d}x_6^j} = \frac{1}{\sigma^2}\left(\frac{x_1^j - x_6^j}{t_1} + \frac{x_2^j - x_6^j}{t_2}\right)$$

Setting this to 0 and solving the equation in respect to x_6^j leads to equation 8.3. Thus, the weights of the tip values in equation 8.3 resulting from the maximum likelihood estimator for the node value supports our intuition of weighing the tip value higher that is closer to the internal node and has not evolved for as long as the other tip value.

The value of x_5^j is a linear combination between x_6^j and x_3^j. However, the value of x_6^j was estimated from x_1^j and x_2^j. We will see that for computing x_5^j we need to replace the actual branch length leading to node 6 has to be corrected to use the above formula. Intuitively, this captures the uncertainty in the estimated value in internal nodes.

8.2.3.3 Computing variance for contrasts

Determining variance for trait values

As explained above, the trait values at the internal nodes, are estimated based on the observations on the tips. From a model perspective, the values at all nodes (tips and internal nodes) are realisations from random variables following a Brownian

motion. Thus the estimator of the internal node 6 in Figure 8.7 can be rewritten in terms of random variables:

$$\hat{X}_6^j = \frac{t_2}{t_1+t_2}X_1^j + \frac{t_1}{t_1+t_2}X_2^j \tag{8.4}$$

Our initial goal is to calculate the contrasts according to equation 8.2 and transform them such that they all have the same variance. Only then, we can use a linear regression. Thus, in the following we calculate the variances for our contrasts, using the variances of our traits X_i^j.

Under the Brownian motion model, the variance of a character trait from the root to a tip is proportional to the branch length from the root to the tip:

$$\begin{aligned} VarX_1^j &= \sigma^2(t_5+t_6+t_1) \\ VarX_2^j &= \sigma^2(t_5+t_6+t_2) \\ VarX_3^j &= \sigma^2(t_5+t_3) \\ VarX_4^j &= \sigma^2(t_4) \end{aligned}$$

The same is also true for parts of the tree, e.g. going from node 5 to 6 in Figure 8.7, as defined in Box 23.

The calculation of the variance of trait values at internal nodes is a bit more difficult as we sum two random variables in the estimator (see Equation 8.4. In general, for two random variables A, B and two parameters α, β one can easily prove:

$$Var\left[\alpha A + \beta B\right] = \alpha^2 Var\left[A\right] + \beta^2 Var\left[B\right] + 2\alpha\beta \mathrm{Cov}[A,B] \tag{8.5}$$

Using this formula, we can compute the variance of the estimator of the jth trait value for the internal node 6:

$$\begin{aligned} Var\hat{X}_6^j &= Var\left[\frac{t_2}{t_1+t_2}X_1^j + \frac{t_1}{t_1+t_2}X_2^j\right] \\ &= \left(\frac{t_2}{t_1+t_2}\right)^2 Var\left[X_1^j\right] + \left(\frac{t_1}{t_1+t_2}\right)^2 Var\left[X_2^j\right] + \frac{2t_2t_1}{(t_1+t_2)^2}\mathrm{Cov}\left[X_1^j, X_2^j\right] \\ &= \sigma^2\left(\frac{t_1t_2}{t_1+t_2} + t_6 + t_5\right) \end{aligned}$$

Note that the covariance $Cov[X_1^j, X_2^j] = t_6 + t_5$, i.e. equals the shared branches between the two characters. As the variance of the Brownian model scales with the branch length, under this model the branch length between node 5 and 6 must be corrected to:

$$t_6' := \frac{t_1t_2}{t_1+t_2} + t_6 \tag{8.6}$$

When we go further down the tree and calculate the variance for older nodes, we need to use these corrected branch lengths to calculate the new corrected branch

lengths in order to obtain their variances. For t_5 in Figure 8.7 we obtain:

$$t'_5 := \frac{t'_6 t_3}{t'_6 + t_3} + t_5$$

We advise the reader to derive the variance $VarX_5^j$ with pen and paper. The reader will then see that this branch length correction appears naturally. In Figure 8.8, example branch lengths are assigned to the left tree, and the corrected branch lengths are displayed on the right tree.

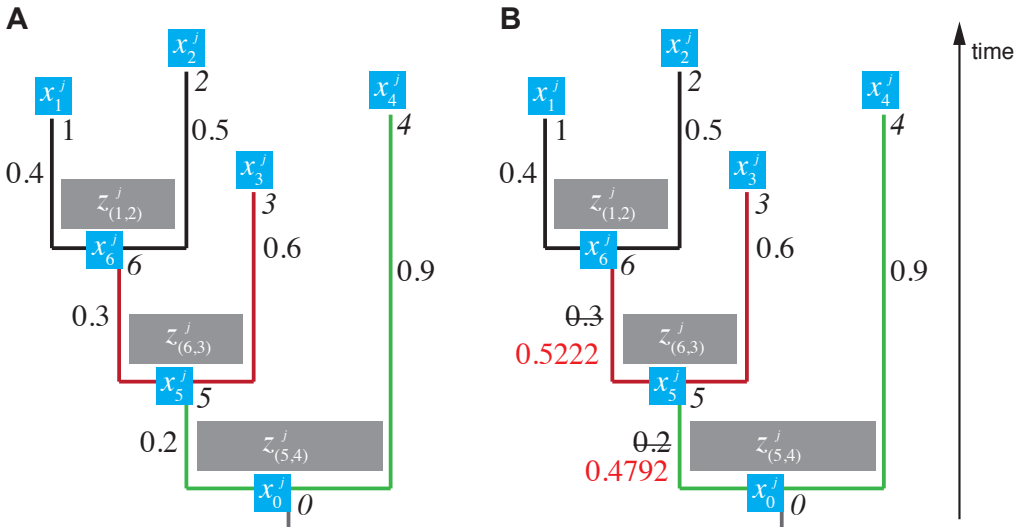


Figure 8.8: Example tree with corrected branch lengths according to equation 8.6.

For the example given in Figure 8.8, we can then calculate the contrasts – here denoted again as random variables $Z^j_{(k,l)}$:

$$\begin{aligned} Z^j_{(1,2)} &= X^j_1 - X^j_2 \\ Z^j_{(6,3)} &= \hat{X}^j_6 - X^j_3 = 5/9X^j_1 + 4/9X^j_2 - X^j_3 \\ Z^j_{(5,4)} &= \hat{X}^j_5 - X^j_4 \end{aligned}$$

Variances for contrasts

Applying equation 8.5, we can now also calculate the variance of the contrasts:

$$\begin{aligned} Var\left[Z^j_{(1,2)}\right] &= Var\left[X^j_1 - X^j_2\right] \\ &= Var\left[X^j_1\right] + Var\left[X^j_2\right] - 2\mathrm{Cov}[X^j_1, X^j_2] \\ &= \sigma^2(t_5 + t_6 + t_1 + t_5 + t_6 + t_2 - 2(t_6 + t_5)) \\ &= \sigma^2(t_1 + t_2) \end{aligned}$$

We observe that the variance of a contrast is the intrinsic variance of the Brownian motion model, σ^2, multiplied with the branch length between nodes 1 and 2. This is what we would expect intuitively from the Brownian motion model. Accordingly we obtain:

$$\begin{aligned} Var\left[Z^j_{(6,3)}\right] &= \sigma^2(t'_6 + t_3) \\ Var\left[Z^j_{(5,4)}\right] &= \sigma^2(t'_5 + t_4) \end{aligned}$$

Note that we can express all 'corrected' variances t'_i in terms of original branch lengths, by successively replacing t'_i by the original branch lengths and the correction terms.

8.2.3.4 Normalization of contrasts

Our final goal is to derive normalized contrasts. We denote the normalized contrast of trait j between nodes $k, l$by $N^j_{(k,l)}$. Knowing that $Var[\alpha X] = \alpha^2 Var[X]$, we can divide the contrasts by the square root of the factor before σ^2 to obtain identically distributed and independent variables: $N^j_{(k.l)} = Z^j_{(k.l)} / \sqrt{Var\left[Z^j_{(k,l)}\right]/\sigma^2}$. These normalized contrasts $Z^j_{(k.l)}$ can be used in linear regression assuming an error distribution $\mathcal{N}(0, \sigma^2)$ as discussed above.

The variances of the contrasts for the tree and branch lengths from Figure 8.8 are:

$$\begin{aligned} Var\left[Z^j_{(1,2)}\right] &= 0.9\sigma^2 \\ Var\left[Z^j_{(6,3)}\right] &= 1.1222\sigma^2 \\ Var\left[Z^j_{(5,4)}\right] &= 1.3792\sigma^2 \end{aligned}$$

and thus,

$$\begin{aligned} N^j_{(1,2)} &= Z^j_{(1,2)}/0.9 \\ N^j_{(6,3)} &= Z^j_{(6,3)}/1.1222 \\ N^j_{(5,4)} &= Z^j_{(5,4)}/1.3792 \end{aligned}$$

8.2.3.5 The contrast method in a nutshell

In the above part, we derived the contrast method for a given tree but the steps are the same for any tree. In a nutshell, to apply the contrast method, we need to

1. Consider a phylogenetic tree with n tips and given branch lengths;

2. Start at the cherries and estimate character values for the two characters for the parental nodes;
3. Calculate the corrected branch length leading to the parental node;
4. Repeat 2 and 3 with all cherries, and move further down in the tree until the root is reached.
5. Calculate the $n-1$ contrasts in the trees;
6. Calculate the variance of the contrasts;
7. For the m different characters calculate the independent and identically distributed normalized contrasts.
8. Finally, perform linear regression on the normalized contrasts for two out of the m different characters, j_1, j_2 according to equation 8.1,:

$$n^{j_2}_{(k,l)} = \beta n^{j_1}_{(k,l)} + b + \epsilon$$

for $k, l = 1, \ldots, n$ accordingly.

Note that even higher level regressions can be calculated.

8.2.4 Examples using the contrast method

Independent contrasts for the example tree with two clades

We will not give the detailed calculations for the first tree that we mentioned as an example for continuous traits (see Figure 8.5), but only the results. By calculating the contrasts for each of the two characters and plotting the results (see Figure 8.9), we see that the previously "obvious" correlation between the character values has vanished. The R^2 value between these two sets of contrasts is approximately 0.0003: much weaker than between the trait values, for which the R^2 was 0.72. Similarly the p-value for the rejection of the null hypothesis (no correlation between the contrasts of differente characters) is 0.94. The corrected data thus supports our early hypothesis of no correlation, based on the plotted values coloured according to their clade.

Example: Fiddler crab carapace breadth and propodus length

Our second example shows how the contrast method can be applied to real biological data. Fiddler crabs are small crabs that are characterised by extreme claw asymmetry in the males, who have one small claw and one oversized claw. The name stems from the particular movement of the small claw when males eat, which resembles the motion of someone moving a bow across a fiddle (the large claw). A fine fiddler crab specimen can be seen on Figure 8.10. The question we would like to answer for these animals is whether there is a correlation between carapace breadth and propodus length in five *Uca* fiddler crab species. The carapace is the hard upper shell of the crab's body, thus carapace breadth is basically the body width. The propodus is the

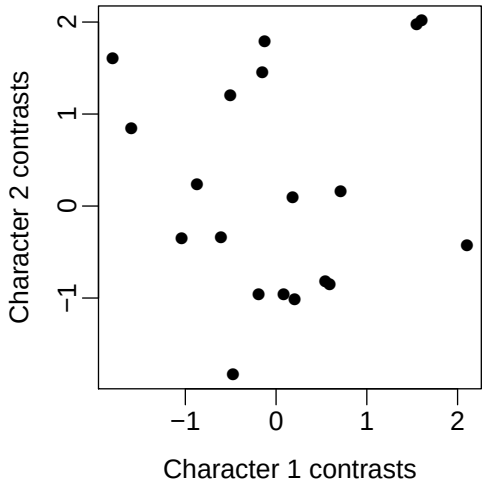


Figure 8.9: The recomputed values for the two clade phylogeny show no significant correlation.

Figure 8.10: A male fiddler crab. Figure adapted from [**fidc**].

base of the crustacean limb, where it forms the base of the claw and the fixed side of the pincer, and in this particular case we are interested in the propodus length of the massive bigger claw.

The tree in Figure 8.11 shows the phylogeny of fiddler crab species and the corresponding measurement values for both traits. Linear regression on the raw values (shown in Figure 8.12) seems to indicate that the carapace breadth is a predictor for the propodus length.

However, as we know by now, linear regression on uncorrected character values might lead to wrong results. Thus, we derive the independent contrasts by first calculating the corrected branch lengths and the character values in the nodes (see Figure 8.13 and Table 8.1. The contrast values are shown in Table 8.2. Finally, when we perform linear regression on the normalized contrast values (see Figure 8.14), we see a much weaker signal for the dependency between carapace breadth and propodus length.

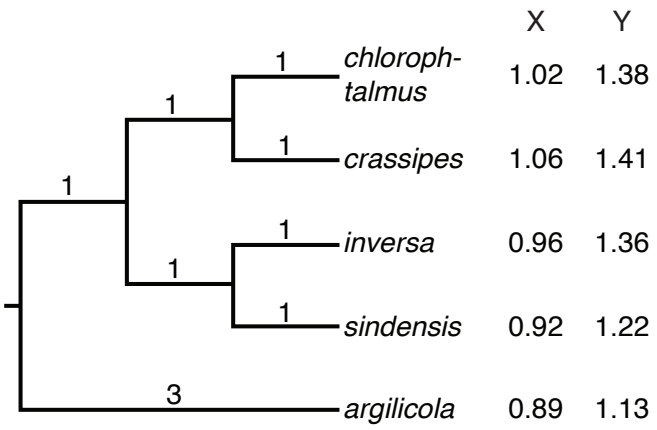


Figure 8.11: The fiddler crab species tree with the measurement values per species. X here stands for carapace breadth and Y stands for propodus length. Figure adapted from [**CompChap5**].

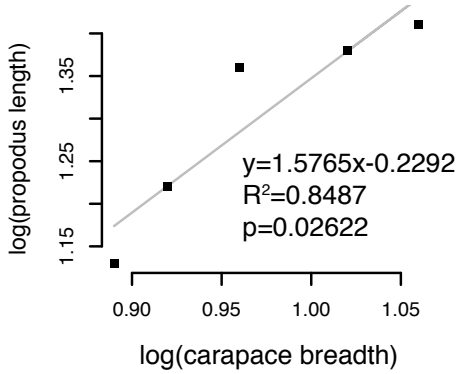


Figure 8.12: Linear regression on raw data points shows a significant ($p < 0.05$) and strong ($R^2 = 0.85$) correlation between carapace breadth and propodus length.

	X	Y
N_1	1.04	1.395
N_2	0.94	1.29
N_3	0.99	1.3425

Table 8.1: Internal node character values for the fiddler crab phylogeny.

8.3 Comparing a continuous with a discrete trait: Prediction of antelope anti-predator behaviour

One disadvantage of the presented methods is that they do not allow one to compare a discrete with a continuous character. We will now provide one example of a real-life study of antelope group size as a predictor of anti-predator behaviour [**CompChap9**]. One could hypothesize that the bigger the average ante-

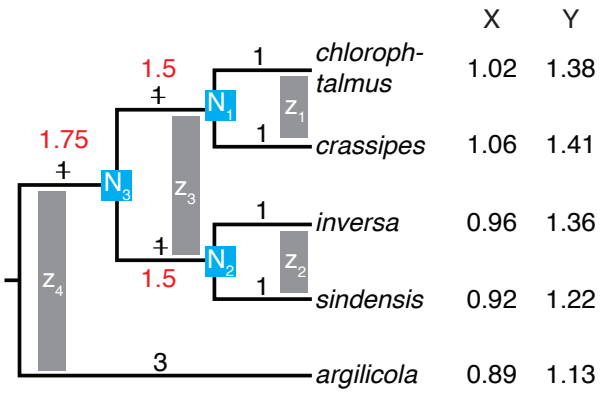


Figure 8.13: Fiddler crab phylogeny with labelled internal nodes. Here the labels for the nodes are N_1, N_2, N_3 as we calculate the node values for the two traits carapace breadth, X, and propodus length, Y. The values of the nodes are summarized in Table 8.1.

	X_{raw}	Y_{raw}	SD	X_{st}	Y_{st}
z_1	0.04	0.03	$\sqrt{2}$	0.028	0.021
z_2	0.04	0.14	$\sqrt{2}$	0.028	0.099
z_3	0.1	0.105	$\sqrt{3}$	0.058	0.061
z_4	0.1	0.2125	$\sqrt{4.75}$	0.046	0.098

Table 8.2: Contrast values for the fiddler crab phylogeny. The index "raw" indicates that the values were not corrected for the variance, "st" indicates that the "raw" values are divided by the square root of the variance (standard deviation).

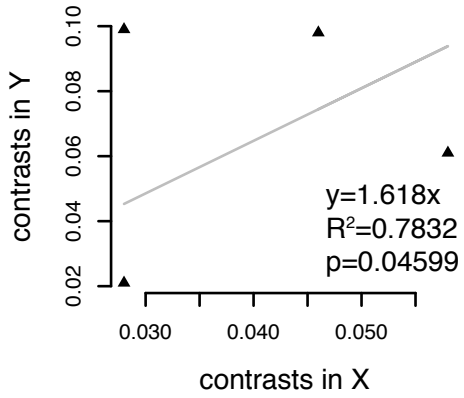


Figure 8.14: Fiddler crab contrasts show much less signal for the relation between carapace breadth and propodus length.

lope group, the more prone the population is to fight a predator instead of running away and hiding. In this example, 75 antelope species were considered. Firstly, the phylogeny of the species was established. Employing four different models, the anti-predator behaviour (which is a discrete variable) was predicted based on the

individual group size ($\log_{10}$, a continuous variable). For further information on the models used we refer the interest reader to [**CompChap9**]. The bottom rows on Figure 8.15 show these predictions. As one can see, most of the methods can quite consistently predict the anti-predator behaviour based on the group size.

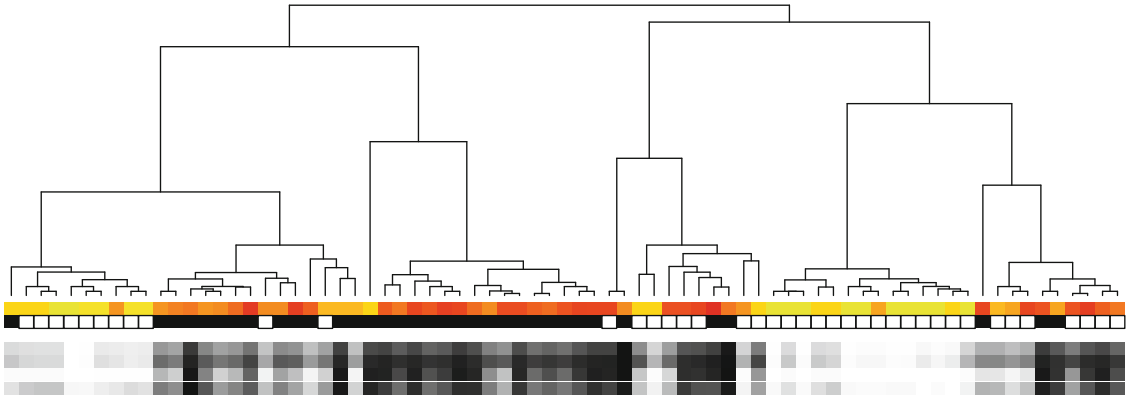


Figure 8.15: The dependency between the continuous group size ($\log_{10}$ group size in yellow-red) and the discrete antipredator behavior (hide = white, flee/fight = black) in 75 antelope species. The four bottom rows show predictions of the antipredator behaviour using different models.

8.4 Extensions

The presented methods for comparing characters that evolved on a phylogeny are only two examples of methods developed for the comparison of such characters. Many other methods have been suggested [**Felsenstein2004**, **CompMethods**] (the book by Luke Harmon [**Harmon2018**] provides a useful guide) and bear intrinsic advantages and disadvantages compared to the the two methods presented here.

We highlight two extensions. First, in this chapter we only compared discrete characters, or continuous characters. However, with the presented methods, we could not compare a discrete with a continuous character. In [**CompChap9**], it is shown how to do such comparisons.

Second, by using Brownian motion for continuous and the hypergeometric distribution for discrete characters, we assumed a process without selection. In particular, under the Brownian motion model, the variance in a trait increases linearly with time without any bounds. However, e.g. body size of animals is bounded. To address such selection constraints, the Ornstein-Uhlenbeck model has been used as a more flexible substitute for the Brownian motion model.

9 Phylodynamics

In the previous chapters, we discussed the field of phylogenetics, where we assumed a fixed phylogenetic tree, and studied genotypic and phenotypic evolution occurring along the tree. A key goal in phylogenetics is to infer the past "state" of the population of interest, i.e. the phylogenetic tree, based on the genetic sequences and the evolutionary models. In this chapter, we introduce the field of phylodynamics. Phylodynamics studies the process of how phylogenetic trees are being generated, and which factors shape that tree generation. The tree generation process is a population dynamic process of individuals representing some biological unit, and these individuals are replicating and dying. E.g. in macroevolution, trees are generated as species undergo speciation inducing new branching events. Extinction events induce the termination of branches. Analog, infected individuals induce new branching events through transmissions, and branches terminate upon recovery. For more examples see Chapter 1.1. The key goal of phylodynamics is to infer the past "process" of tree generation, i.e. quantifying the parameters determining the tree generation.

Throughout this chapter, we will assume that we have already obtained the phylogenetic tree, with branch lengths representing calendar time, e.g. days, months, or years (see Section 6.6). We will discuss the two main frameworks for generating trees, namely birth-death models and the coalescent, and how important parameters such as speciation or transmission rates can be estimated from the phylogenetic tree.

9.1 Birth-death model

An important class of models used in phylodynamic analysis are the *birth-death models*, in which two processes, birth and death, give rise to the phylogenetic trees. Phylodynamic analysis using these birth-death models aims to understand and quantify the birth and death rates in the studied population.

9.1.1 Population dynamic model

The basic model, the *constant rate birth-death model*, is the *population dynamic model* shown in Figure 9.1. In this model, the population is represented by the compartment labelled I, which stands for "Individual", and an individual may correspond to any of the biological units discussed in Chapter 1.1. The process starts at some time 0 with $I(0)$ initial individuals, which is the initial state. All individuals in this population are identical and they give rise to other individuals at birth rate β and die at death rate δ. We call the compartments within a model the "states",

and the rates quantify the "dynamics". Overall, the model is called a *compartmental model*.

In this section, we will explain phylodynamic principles based on this basic model. At the end of Section 9.1, we will mention how time dependence has been introduced into the basic model. Many applications require more complex models, in particular models with sub-populations to take into account heterogeneity across individuals. This will be discussed in Section 9.4.

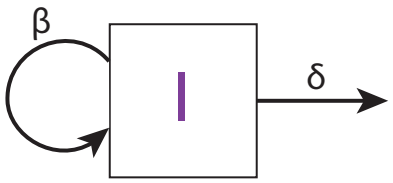


Figure 9.1: Constant rate birth-death model. Individuals from the compartment I are born at birth rate β and die at death rate δ.

The basic compartmental model can be considered as a deterministic model or as a stochastic model. In the deterministic formulation, with $I(t)$ being the number of individuals in the I compartment after time t has elapsed, the change in $I(t)$ is described via:

$$\frac{d}{dt}I(t) = (\beta - \delta)I(t).$$

Thus, since $\frac{d}{dt}I(0)e^{(\beta-\delta)t} = I(0)(\beta-\delta)e^{(\beta-\delta)t}$, we obtain $I(t) = I(0)e^{(\beta-\delta)t}$ as a solution for the differential equation. If the birth rate is equal to the death rate, then the derivative above will be equal to 0 and the size of the population will remain constant for all time. If the birth rate is bigger than the death rate, the population will grow exponentially, while if the death rate is bigger than the birth rate, the population size will tend to zero. Note that $I(t)$ will not only take integer values, but any value between zero and ∞. Thus, $I(t)$ is not the actual population size, instead the deterministic model can be regarded as tracking the average population size (*cf.* Theorem 9.1.4).

For phylodynamic applications, we do not track population averages as in the deterministic case, but we track individuals within the phylogenetic tree, where each branch in the phylogenetic tree represents an individual. A stochastic formulation of the birth-death model allows us to consider the phylogenetic tree, connecting the sampled individuals, as a stochastic outcome of the birth-death model.

In the stochastic formulation, the rate r associated to an arrow in the compartmental model states that the probability for the corresponding event to happen within a small time step, Δt, is $r\Delta t$ (see also chapter 5.1.2.1). . For the constant rate birth-death model, the probability that an individual gives birth to a new individual in a small time step is $\beta\Delta t$. The probability that an individual dies in a small time step is $\delta\Delta t$. Thus the waiting time to a birth event is exponentially distributed with

parameter β and to a death event is exponentially distributed with parameter δ (cf. Section 5.1.2 and Box 14). According to Box 15, this means that an individual gives birth through time according to a Poisson process with parameter β.

In a phylodynamic context, the birth-death model is assumed to start with one individual at time 0, and stopped after time T. The stopping time is also called present time. A simulation of such a process is shown in Figure 9.2a. Time is represented on the x-axis. Each horizontal solid black line in the graph represents the lifetime of an individual. Blue arrows in the figure represent birth events.

We will now consider the population size through time. We consider a single individual at some time point, and consider its offspring population size after some time t has elapsed, X_t. Formally, this offspring population size is a stochastic process $(X_t)_{t\in[0,T)}$. We are interested in the distribution of X_t, i.e. the probability of population size n after time t, $P(X_t = n)$, where $n = 0, 1, 2, \ldots$. We abbreviate this probability as,

$$p(n|t) := P(X_t = n)$$

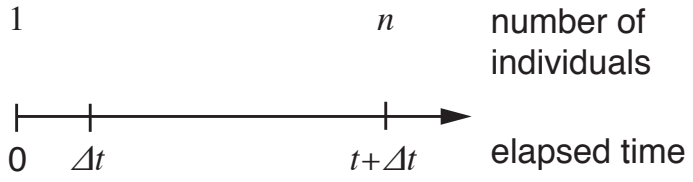
We will derive $p(n|t)$ using Master equations. These expressions were already proven through the use of generating functions in [**Kendall1948pn**]. Recall that under the deterministic model, the population size after a time interval t elapsed is $e^{(\beta-\delta)t}$. Throughout the remainder of this chapter, we assume $\beta \neq \delta$ and $\beta, \delta > 0$. For obtaining results in the cases $\beta = \delta$ or $\delta = 0$, one can take the limit $\delta \to \beta$ and $\delta \to 0$ as done e.g. in [**StadlerSteel2019swapping**].

9.1.1.1 Derivation of the probability of extinction, $p(0|t)$

The birth-death model starts with one individual. What is the probability that no individual survived after time interval t, $p(0|t)$?

First, note that for $t = 0$, we have, $p(0|t = 0) = 0$, since we consider one individual at the start of the time interval. In order to determine $p(0|t)$ for $t > 0$, we derive its Master equation. We begin by deriving $p(0|t + \Delta t)$, i.e. the probability that the process is extinct after $t + \Delta t$ time units, as a function of $p(0|t)$. The process starts with a single individual. For a single individual, during a time interval Δt, a death event happens with probability $\delta\Delta t$, and a birth event happens with probability $\beta\Delta t$.

We partition the time interval $t + \Delta t$ in the following way:



During the time Δt after the start of the process, four things can happen:

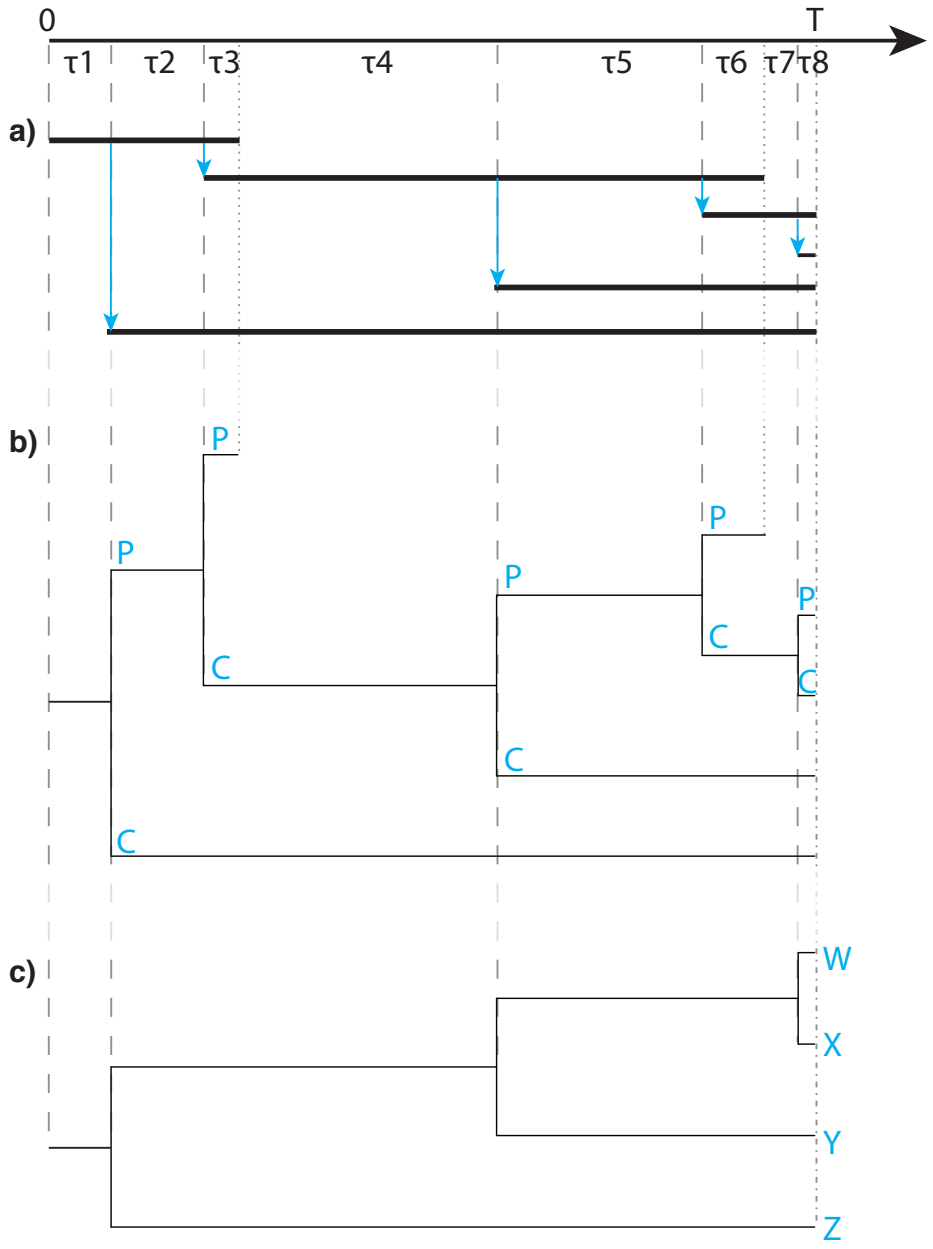


Figure 9.2: a) Visual representation of the full population dynamics of one realisation of the birth-death model. b) Complete tree obtained from the population dynamics of the top figure (P stands for 'parent', C for 'child'). c) Phylogenetic tree of sampled lineages resulting from the pruning of non-sampled lineages from the complete tree. Dashed lines indicate the time of birth events, dotted lines indicate the time of death events, and the dot-dashed line indicates the end time T of the process.

(i) Nothing happens. This event has the probability $1 - (\beta + \delta)\Delta t$. Then, the original individual has to go extinct within the remaining time t, which has

probability $p(0|t)$;

(ii) The individual dies with probability $\delta\Delta t$;

(iii) The individual gives birth to another individual with probability $\beta\Delta t$, and both individuals go extinct within time t, which has probability $p(0|t)^2$;

(iv) More than one event happens, which has probability $\mathcal{O}(\Delta t^2)$.

Thus, we obtain the Master equation,

$$p(0|t+\Delta t) = \underbrace{(1-(\beta+\delta)\Delta t)p(0|t)}_{(i)} + \underbrace{\delta\Delta t}_{(ii)} + \underbrace{\beta\Delta t p(0|t)^2}_{(iii)} + \underbrace{\mathcal{O}(\Delta t^2)}_{(iv)}.$$

Rearranging leads to,

$$\frac{p(0|t+\Delta t)-p(0|t)}{\Delta t} = -(\beta+\delta)p(0|t) + \delta + \beta p(0|t)^2 + \mathcal{O}(\Delta t).$$

Taking the limit $\Delta t \to 0$ leads to the differential equation,

$$\frac{d}{dt}p(0|t) = -(\beta+\delta)p(0|t) + \delta + \beta p(0|t)^2. \tag{9.1}$$

The solution to this differential equation with initial condition $p(0|t=0)=0$ is,

$$p(0|t) = \frac{\delta(1-e^{-(\beta-\delta)t})}{\beta-\delta e^{-(\beta-\delta)t}},$$

which can easily be verified by differentiating the expression and plugging it into the differential equation.

9.1.1.2 Derivation of $p(1|t)$

For $t=0$, we have, $p(1|t=0)=1$, since we consider one individual at the start of the time interval. For $t>0$, we again express $p(1|t+\Delta t)$ as a function of $p(1|t)$. Note that here, compared to the derivation of $p(0|t)$, a death event cannot occur during the first time step Δt, otherwise the process would die out and thus the process would not lead to one individual at present time. We obtain the following Master equation,

$$p(1|t+\Delta t) = (1-(\beta+\delta)\Delta t)p(1|t) + \beta\Delta t \times 2p(1|t)p(0|t) + \mathcal{O}(\Delta t^2).$$

The factor of 2 in this Master equation accounts for the fact that either one of the descendants of the birth event may lead to the surviving individual after time t. The Master equation can be rearranged to obtain the differential equation,

$$\frac{d}{dt}p(1|t) = -(\beta+\delta)p(1|t) + 2\beta p(1|t)p(0|t). \tag{9.2}$$

The solution to this differential equation with initial condition $p(1|t=0)=1$ is,

$$p(1|t) = (1-p(0|t))(1-\frac{\beta}{\delta}p(0|t)),$$

which again can easily be verified by differentiating the expression and plugging it into the differential equation.

9.1.1.3 Derivation of $p(n|t)$

Theorem 9.1.1. *The probability for an individual producing n ($n \in \{0,1,2,\ldots\}$) extant individuals after time t, $p(n|t)$, is,*

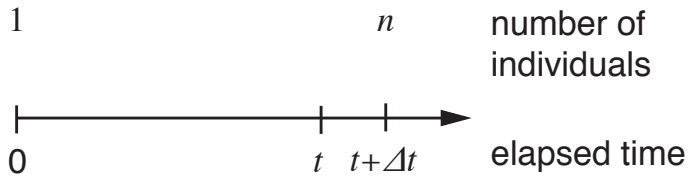
$$p(0|t) = \frac{\delta(1-e^{-(\beta-\delta)t})}{\beta-\delta e^{-(\beta-\delta)t}} \tag{9.3}$$

$$p(1|t) = (1-p(0|t))(1-\frac{\beta}{\delta}p(0|t)), \tag{9.4}$$

$$p(n|t) = p(1|t)\left(\frac{\beta}{\delta}p(0|t)\right)^{n-1} \qquad \text{for } n \geq 2. \tag{9.5}$$

Proof. The expressions for $n=0$ and $n=1$ have been derived in Sections 9.1.1.1 and 9.1.1.2. To prove the expression for $p(n|t)$, $n \geq 2$, we first note that $p(n|t=0)=0$. Indeed, Equation 9.5 is 0 for $t=0$, since $p(0|t=0)=0$.

For $t>0$, we again derive the differential equation. We now consider $p(n|t+\Delta t)$ as a function of $p(n|t)$, for all $n \geq 1$. In contrast to the derivation of $p(0|t)$ and $p(1|t)$, we split up time into an interval of length t, followed by a time interval of length Δt:



In order to arrive with n individuals after time $t+\Delta t$, we may arrive after time t with:

(i) n individuals (probability $p(n|t)$) followed by no event in the last time interval (probability $1-n(\beta+\delta)\Delta t$);

(ii) $n-1$ individuals (probability $p(n-1|t)$) followed by a birth event in the last interval (probability $(n-1)\beta\Delta t$);

(iii) $n+1$ individuals (probability $p(n+1|t)$) followed by a death event in the last interval (probability $(n+1)\delta\Delta t$);

(iv) any number > 0 of individuals, followed by more than two events in the last interval (probability on the order of $\mathcal{O}(\Delta t)^2$).

This leads to the differential equation,

$$\begin{aligned}\frac{d}{dt}p(n|t) = &- n(\beta+\delta)p(n|t) + (n-1)\beta p(n-1|t) \\ &+ (n+1)\delta p(n+1|t) \quad \text{for} \quad n \geq 1. \end{aligned} \tag{9.6}$$

We can now prove the expression for $p(n|t)$ (Equation 9.5) by induction:

- Hypothesis to prove: $p(n|t) = p(1|t)\left(\frac{\beta}{\delta}p(0|t)\right)^{n-1}$.

- Base step: Check that the hypothesis holds for $n = 2$.
  Consider Equation 9.6 for $n = 1$: $\frac{d}{dt}p(1|t) = -(\beta+\delta)p(1|t) + 2\delta p(2|t)$. Re-arranging yields to:

$$\begin{aligned} p(2|t) &= \frac{1}{2\delta}\left(\frac{d}{dt}p(1|t) + (\beta+\delta)p(1|t)\right) \\ &\overset{(9.2)}{=} \frac{1}{2\delta}\Big(-(\beta+\delta)p(1|t) + 2\beta p(1|t)p(0|t) + (\beta+\delta)p(1|t)\Big) \\ &= p(1|t)\left(\frac{\beta}{\delta}p(0|t)\right) \end{aligned}$$

- Induction hypothesis: Suppose the hypothesis holds for all $k \leq n$.

- Inductive step: Show that the formula holds for $k = n+1$. We consider Equation 9.6 in a re-arranged form:

$$(n+1)\delta p(n+1|t) = \frac{d}{dt}p(n|t) + n(\beta+\delta)p(n|t) - (n-1)\beta p(n-1|t)$$

  By using and differentiating the expression for $p(n|t)$ as stated in Equation 9.5, and combining the result with the expressions in Equation 9.1 and 9.2, we

obtain,

$$
\begin{aligned}
(n+1)\delta p(n+1|t) \overset{(9.5)}{=} & \frac{d}{dt}p(1|t)\left(\frac{\beta}{\delta}p(0|t)\right)^{n-1} + p(1|t)(n-1)\left(\frac{\beta}{\delta}p(0|t)\right)^{n-2}\frac{\beta}{\delta}\frac{d}{dt}p(0|t) \\
& +n(\beta+\delta)p(1|t)\left(\frac{\beta}{\delta}p(0|t)\right)^{n-1} - (n-1)\beta p(1|t)\left(\frac{\beta}{\delta}p(0|t)\right)^{n-2} \\
\overset{(9.1,9.2)}{=} & (-(\beta+\delta)p(1|t) + 2\beta p(1|t)p(0|t))\left(\frac{\beta}{\delta}p(0|t)\right)^{n-1} \\
& +p(1|t)(n-1)\left(\frac{\beta}{\delta}p(0|t)\right)^{n-2}\frac{\beta}{\delta}(-(\beta+\delta)p(0|t) + \delta + \beta p(0|t)^2) \\
& +n(\beta+\delta)p(1|t)\left(\frac{\beta}{\delta}p(0|t)\right)^{n-1} - (n-1)\beta p(1|t)\left(\frac{\beta}{\delta}p(0|t)\right)^{n-2} \\
= & \left(\frac{\beta}{\delta}p(0|t)\right)^{n-2}((n-1)\beta p(1|t) - (n-1)\beta p(1|t)) \\
& +\left(\frac{\beta}{\delta}p(0|t)\right)^{n-1}(-(\beta+\delta)p(1|t) + (-(\beta+\delta)p(1|t)(n-1)) + n(\beta+\delta)p(1|t)) \\
& +\left(\frac{\beta}{\delta}p(0|t)\right)^{n}(2\delta p(1|t) + \delta p(1|t)(n-1)) \\
= & \left(\frac{\beta}{\delta}p(0|t)\right)^{n}\delta(n+1)p(1|t).
\end{aligned}
$$

Thus, we obtain,

$$p(n+1|t) = p(1|t)\left(\frac{\beta}{\delta}p(0|t)\right)^{n},$$

which establishes the induction step.

□

It directly follows from Theorem 9.1.1 that,

Corollary 9.1.2. *Consider one individual at some time point. The number of extant descendants produced by this individual after time t conditioned on non-extinction of the process (which has probability function $\frac{p(n|t)}{1-p(0|t)} = (1-\frac{\beta}{\delta}p(0|t))\left(\frac{\beta}{\delta}p(0|t)\right)^{n-1}$), follows a geometric distribution with parameter $\left(1-\frac{\beta}{\delta}p(0|t)\right)$ (see Box 5) . Thus, $\frac{1}{\left(1-\frac{\beta}{\delta}p(0|t)\right)}$ is the expected number of lineages arising from a single lineage within time t, conditioned that the process survives.*

9.1.2 Phylodynamic model

The simulation displayed in Figure 9.2a does not look like a phylogenetic tree. We obtain the *complete tree* by plotting a branching event for each birth event, instead of the blue arrows, as shown in Figure 9.2b. The child-parent relationships are depicted as labels on each branching event: the lineage which already existed before the corresponding birth event is labelled (P) for parent and the lineage that just appeared is labelled (C) for child.

This complete tree contains all lineages that have ever existed during the process. Together with the P/C labels, it represents the full population dynamic history. In

most empirical datasets, however, we do not sample the full population nor do we know the P/C labels.

We now add a *sampling model* to the population dynamics model, both models together define the *phylodynamic model*. First, we define *extant tip sampling*. It models the sampling of individuals at present time. Under the simplest model, each individual at present is sampled with probability ρ.

The rate of sampling an individual prior to the present time is denoted as ψ, also referred to as *sampling through time*. Upon sampling, an individual dies with probability r and continues to live with probability $1-r$. Thus, ψr is the death rate with sampling compared to δ being the death rate without sampling. The obtained samples through time may stem from fossils, ancient DNA, or from patients throughout the time of an epidemic.

Thus, our full phylodynamic model contains the parameters [**Stadler2012**],

β: birth rate,

δ: death rate,

T: time after which the process is stopped,

ρ: extant tip sampling probability,

ψ: sampling rate of individuals prior to the presence,

r: death probability of sampled individuals prior to the present.

Under this model, we can simulate forward in time complete trees together with the sampled individuals. In order to obtain the phylogenetic tree from a complete tree, we remove the parent-child labels from the complete tree, as well as all lineages without sampled descendants. Then, each sample is labelled in the phylogenetic tree with a unique label. Thus, in line with the definitions in Chapter 6, our phylogenetic tree is a labelled tree. Note that in phylogenetic tree inference, we aim at obtaining precisely this phylogenetic tree from the sequence data. In what follows, we perform phylodynamic inference on a given phylogenetic tree.

For $\rho = 1,\ \psi = 0$, the resulting phylogenetic tree is shown in Figure 9.2c, with the samples (i.e. the four extant individuals) being labeled W, X, Y, Z. For $\rho = 0,\ \psi > 0,\ r = 1$, we observe e.g. the phylogenetic tree shown in Fig. 9.3a. For $\rho = 1,\ \psi > 0,\ r < 1$, we observe e.g. the phylogenetic tree shown in Fig. 9.3b. This latter tree has so-called *sampled ancestors* [**GavryushkinaEtAl2014**] (labelled by E and F), meaning samples which give rise to further samples.

Sampled ancestors are generated when samples are not being removed from the population (which occurs with probability $1-r$). If a descendant of such a non-removed sample is sampled, we obtain a sampled ancestor. Note that a labelled tree is now defined as a tree where a unique label is assigned to each sample (tip or sampled ancestor).

A sampled ancestor in the context of epidemiology if a patient – after being sampled – infects another patient who is also sampled. Further, fossils within species trees may be sampled ancestors, as descendants of the species representing a fossil sample may be sampled. Note that in Chapter 6 on phylogenetic inference, we did not allow for sampled ancestors. Instead all samples were tips in the tree. However, recently published Bayesian inference tools allow for the inference of sampled ancestors [**Gavryushkina2017**, **zhang2016**].

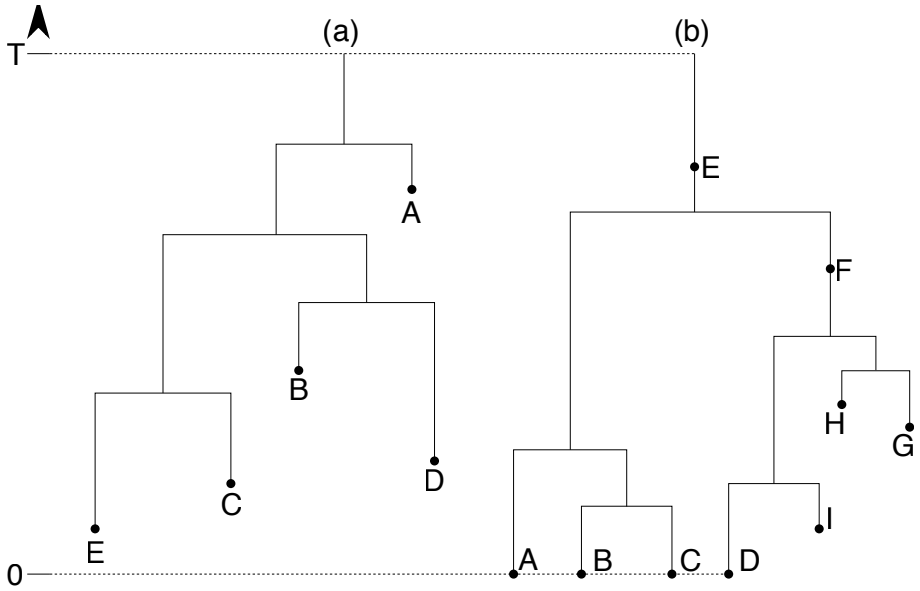


Figure 9.3: Trees generated under the birth-death phylodynamic model. (a) Tree generated with $\rho = 0$, $\psi > 0$ and $r = 1$. We obtain samples through time labeled by $A - E$, but no present-day samples. (b) Tree generated with $\rho = 1$, $\psi > 0$ and $r < 1$. We obtain both present-day samples (with labels $A - D$), sampled ancestor samples (with labels $E - F$), and sampled tips prior to the present (with labels $G - I$).

A key aim of phylodynamic analysis is to infer the birth and death rates in the underlying population based on the phylogenetic tree of the sampled individuals. In the following, we will assume that we sample the entire population at present, i.e. $\rho = 1$, but we have no samples from the past, i.e. $\psi = 0$. This setting allows us to obtain some intuitive understanding. At the end, we briefly explain how to extend the discussed inference framework to more general scenarios.

9.1.3 Ranked labelled tree topologies

First, we start by considering a phylogenetic tree where we ignore branch lengths, meaning we only consider the "discrete part" of the phylogenetic tree. We will show in this section that no information about birth- and death rates is contained in the discrete part of the phylogenetic tree.

In particular, we consider here the *ranked labelled tree topology* [**FordEtAl2009**]

generated by a constant rate birth-death model with $\rho = 1$ and $\psi = 0$. The phylogenetic tree is "ranked" by assigning a rank to each internal node. An internal node obtains the rank i if this node is the ith branching event in the tree. In particular, the root has rank 1, and the biggest rank in a tree on n tips is $n-1$ (see Figure 9.4). The term "topology" states that we ignore branch lengths. A ranked labelled tree topology (discrete part) together with the vector of branching times $x_1, x_2, \ldots$ associated with the nodes of rank $1, 2, \ldots$ (continuous part) uniquely determines the phylogenetic tree.

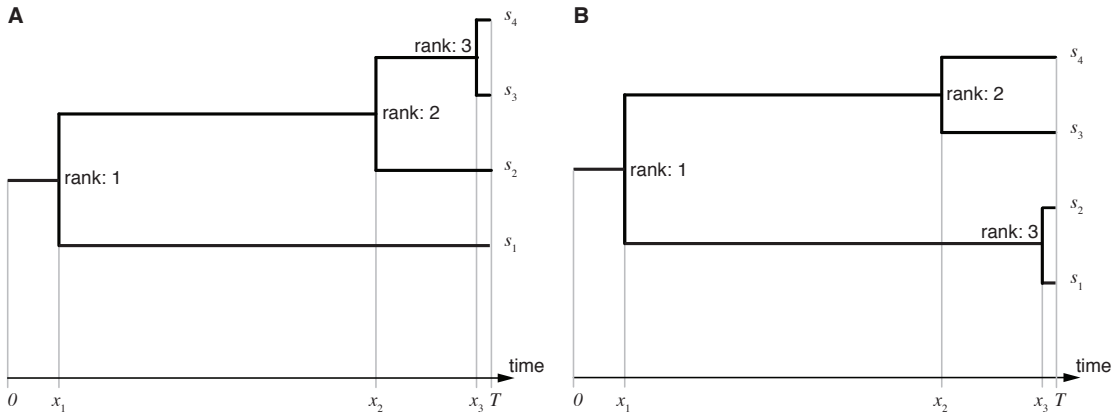


Figure 9.4: Two examples of ranked labelled trees with four tips. The branching times in the phylogenetic tree, x_1, x_2, x_3, are the same in both trees but the clustering and therefore the ranked labelled tree topologies are not.

Under the constant rate birth-death model, given an event (birth or death) happens, each individual has the same probability to be the one undergoing this event because each individual has the same birth- and death rates. This observation leads to the following theorem.

Theorem 9.1.3. *The constant rate birth-death model with $\rho = 1$ and $\psi = 0$ induces a uniform distribution on ranked labelled tree topologies. Each ranked labelled tree topology has probability $\frac{2^{n-1}}{n!(n-1)!}$.*

Proof. Consider a realization of a birth-death model leading to n extant tips, and the internal node with rank i having branching time x_i. Consider the resulting ranked labelled tree topology. In order to calculate the probability of that ranked labelled tree topology – within the set of all ranked labelled tree topologies with associated branching times $x_1, \ldots, x_{n-1}$ – we trace the individuals from the present back in time. At the time x_{n-1}, we have $\binom{n}{2}$ possibilities to coalesce two individuals, and since all individuals undergo the same dynamics (i.e. the same birth- and death rates), the probability of observing the branching event in our given tree is $1/\binom{n}{2}$. We proceed with the same reasoning until we reach the root of the tree. Overall, the probability of the ranked labelled tree topology, given any $x_1, \ldots, x_{n-1}$, is $\prod_{i=2}^{n} \frac{1}{\binom{i}{2}}$.

By using the definition of the binomial coefficient, we can simplify,

$$\begin{aligned}\prod_{i=2}^{n} \frac{1}{\binom{i}{2}} &= \prod_{i=2}^{n} \frac{2!(i-2)!}{i!} \\ &= \frac{2^{n-1}}{n!(n-1)!}\end{aligned}$$

Thus, the birth-death model induces the uniform distribution on ranked labelled tree topologies on n tips. □

This result in particular implies that the distribution of ranked labelled tree topologies is independent of the birth and death rates. In [**Aldous2001**], this result is generalized using the analog reasoning as in the proof above: in fact, any model where the birth- and death rates are the same across all individuals at every point in time – even if e.g. these rates change through time [**Stadler2011**, **Morlon2011EtAl**] or are a function of the overall number of individuals [**EtienneEtAl2011**] – induces a uniform distribution on ranked labelled tree topologies. Such models are also called *homogeneous models* as all individuals at the same time point undergo the same dynamics. We emphasize that homogeneity here refers to homogeneous individuals at one time point. In Box 17, we encountered a different type of homogeneity, namely homogenous probabilities through time.

As a consequence, when we aim at quantifying birth- and death rates under models where these rates are the same across co-existing individuals, the ranked labelled tree topology contains no information about these rates. Thus, all information about the birth- and death rates is contained in the branching times $x_1, \ldots, x_{n-1}$. We will now first provide an intuitive approach for birth-death parameter estimation based on expectations on branching times, and then a maximum likelihood approach based on the full distribution on branching times.

9.1.4 Expected population sizes and branching times

The plot with the number of lineages through time in a tree on the y-axis versus time on the x-axis is called the *lineages-through-time (LTT) plot*. An example of an LTT plot is shown in Figure 9.5, bottom. The LTT plot of the complete tree shows the population size through time, while the LTT plot of the phylogenetic tree shows the number of lineages surviving to the present through time. Note that the LTT plot only summarizes the branching times, but does not display any information regarding the ranked labelled tree topology.

The LTT plots provide one method of estimating the parameters of the birth-death model. To demonstrate how this works, we simulated a large number of realizations of the same constant rate birth-death model, i.e. we obtained a large number of trees with identical constant birth rates and identical constant death rates. In our simulation, we set $T = 50$ and $\beta > \delta$. Note that with $\beta < \delta$, the population would be on average decreasing, with most trees rapidly going extinct. We then

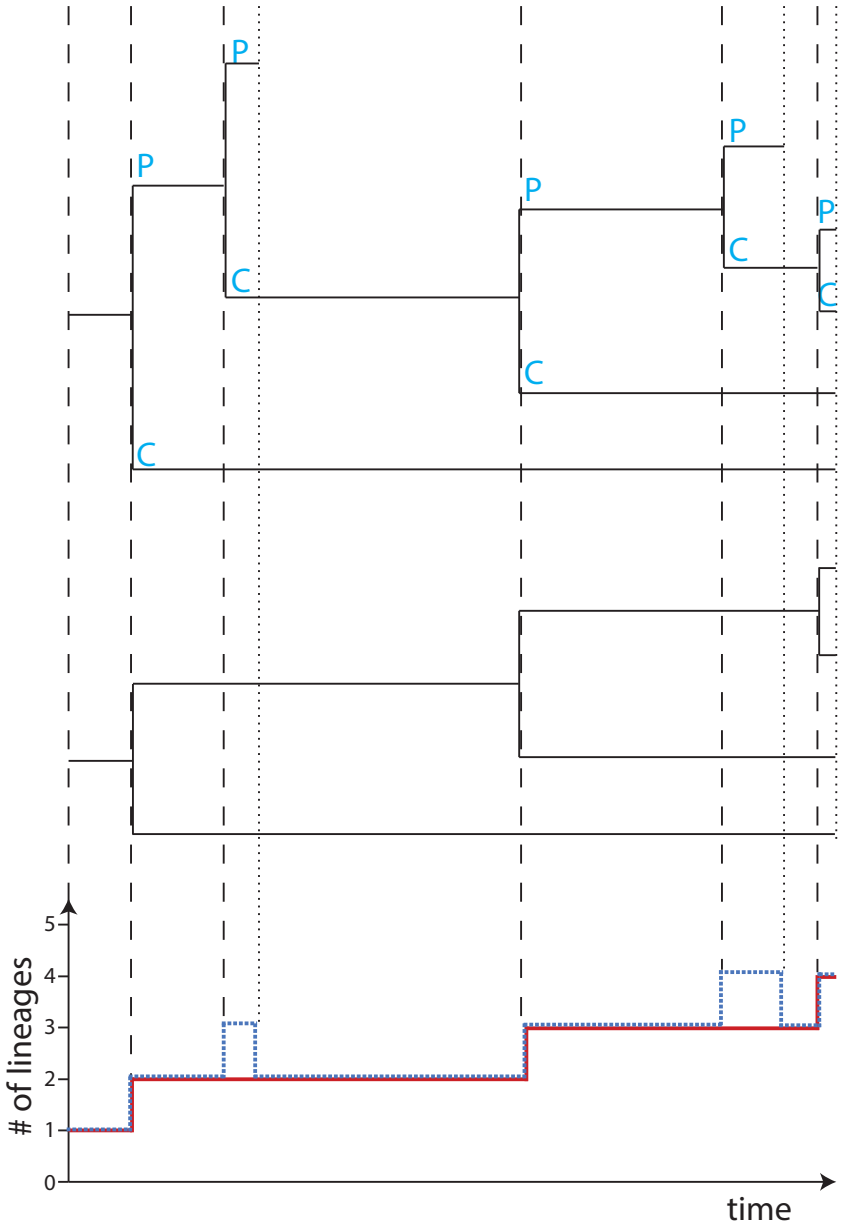


Figure 9.5: Complete tree (top), phylogenetic tree (middle), and corresponding lineages-through-time (LTT) plot (bottom), continuing the example in Figure 9.2. The LTT plot of the complete tree is shown in dotted blue, the LTT plot of the phylogenetic tree in red.

plot the average over all LTT plots for the phylogenetic trees (*cf.* Fig. 9.5, middle; phylogenetic LTT plot) in Figure 9.6, red. The average over all LTT plots for the complete trees (*cf.* Fig. 9.5, top; complete LTT plot) is shown in blue. Finally, the average total population size over all realizations, i.e. over the complete trees and over the trees which went extinct prior to time T, is shown in black. Note that the y-axis is displayed on the log-scale. In what follows, we discuss the shape of these LTT plots under the constant rate birth-death model. The resulting insights will

be employed for estimating birth- and death rates from the branching times in a phylogenetic tree.

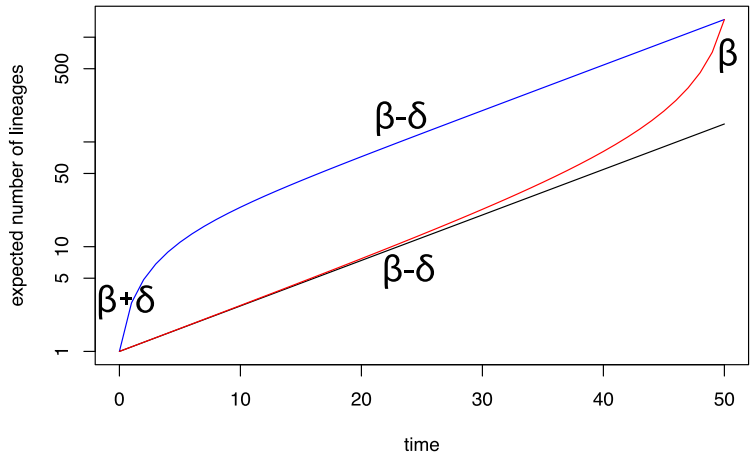


Figure 9.6: Average complete LTT plot (blue) and average phylogenetic LTT plot (red). The average total population size is shown in black. The plots display the scenario $\beta > \delta$, i.e. an on average increasing population size. The increased slope at the start of the complete LTT plot is called the "push-of-the-past"; the increased slope at the end of the phylogenetic LTT plot is called the "pull-of-the-present" [**Nee1994PullOfPresentLTT**]. The slopes associated to the lines hold in the limit $T \to \infty$. As an exception, the slope β for the red line and the slope for the black line holds for all T.

9.1.4.1 Average total population size

Based on Figure 9.6, black, we observe that the average total population size through time has a constant slope $\beta - \delta$ on the log scale, corresponding to the average total population size $e^{(\beta-\delta)t}$ at time t. We denote the average total population size at time t by $N(t)$, and note that $N(t)$ is the expectation of the random variable X_t (the offspring population size) defined in Section 9.1.1, $N(t) = E[X_t]$. We prove in the following theorem that $N(t)$ indeed grows exponentially at rate $\beta - \delta$.

Theorem 9.1.4. *The expected number of lineages at time t, $N(t)$, is,*

$$N(t) = e^{(\beta-\delta)t}.$$

Proof. First, we note that $\sum_{n=1}^{\infty} nx^{n-1} = \frac{1}{(1-x)^2}$ for $-1 < x < 1$. This follows directly from differentiating the formula for an infinite geometric series where $-1 < x < 1$, $\sum_{n=0}^{\infty} x^n = \frac{1}{1-x}$. Second, we note that $0 \leq \frac{\beta}{\delta} p(0|t)$ since β, δ and $p(0|t)$ are non-negative. Further, for $\delta > \beta$, we have $\frac{\beta}{\delta} p(0|t) \leq \frac{\beta}{\delta} < 1$. For $\delta < \beta$, $\frac{\beta}{\delta} p(0|t) =$

$\frac{\beta-\beta e^{-(\beta-\delta)t}}{\beta-\delta e^{-(\beta-\delta)t}} < 1$. Together, we have $0 \leq \frac{\beta}{\delta}p(0|t) < 1$, and we obtain,

$$\begin{aligned} N(t) &= \sum_{n=0}^{\infty} np(n|t) \\ &\overset{(9.5)}{=} \sum_{n=1}^{\infty} np(1|t)\left(\frac{\beta}{\delta}p(0|t)\right)^{n-1} \\ &\overset{(9.4)}{=} \frac{p(1|t)}{\left(1-\frac{\beta}{\delta}p(0|t)\right)^2} = \frac{1-p(0|t)}{1-\frac{\beta}{\delta}p(0|t)} = \\ &= e^{(\beta-\delta)t}. \end{aligned}$$

The last equation follows when using the equality $1-\frac{\beta}{\delta}p(0|t) = e^{-(\beta-\delta)t}(1-p(0|t))$, which can be established from Eqn. 9.3. □

Note that the average or expected population size $N(t)$ equals the population size under the deterministic model, $I(t)$.

9.1.4.2 Complete LTT plot

Based on Figure 9.6, blue, the complete LTT plot goes through a period of accelerated growth at the beginning of the process, before growing exponentially at a rate close to $\beta - \delta$. Recall that the complete LTT plot only includes populations that survive to the present. An intuitive explanation for the initial fast growth is that populations that grow slowly at the start are more likely to go extinct before the end of the process, and thus are not included in the average complete LTT plot. This phenomenon is called the *push-of-the-past* [**Nee1994PullOfPresentLTT**]. In the following, we provide the full equation for the blue line, $N_T(t)$, which has been originally stated in [**Harvey1994**].

Theorem 9.1.5. *The expected number of lineages at time $t \leq T$, conditioned on non-extinction at present time T, is denoted with $N_T(t) := E[X_t|X_T > 0]$. We have,*

$$N_T(t) = \frac{e^{(\beta-\delta)t}}{1-p(0|T)} - \frac{p(0|T)-p(0|t)}{(1-p(0|t))(1-p(0|T-t))}. \tag{9.7}$$

Proof. We have,

$$\begin{aligned} P(X_t = n|X_T > 0) &= \frac{P(X_t = n, X_T > 0)}{P(X_T > 0)} \\ &= \frac{P(X_T > 0|X_t = n)P(X_t = n)}{P(X_T > 0)} \\ &= \frac{(1-p(0|T-t)^n)p(n|t)}{1-p(0|T)}. \end{aligned}$$

Taking the expectation, we obtain,

$$\begin{aligned} N_T(t) &= \sum_{n=1}^{\infty} nP(X_t = n | X_T > 0) \\ &= \frac{\sum_{n=1}^{\infty} np(n|t)}{1 - p(0|T)} - \frac{\sum_{n=1}^{\infty} np(0|T-t)^n p(n|t)}{1 - p(0|T)}. \end{aligned}$$

Using Theorem 9.1.4 for the left expression and $\sum_{n=1}^{\infty} nx^{n-1} = \frac{1}{(1-x)^2}$ for the right expression (see also the proof of 9.1.4 for the latter), we obtain,

$$N_T(t) = \frac{e^{(\beta-\delta)t}}{1 - p(0|T)} - \frac{p(1|t)p(0|T-t)}{1 - p(0|T)} \frac{1}{(1 - \frac{\beta}{\delta} p(0|t)p(0|T-t))^2}$$

This can be simplified to Equation 9.7. □

In order to investigate the shape of the LTT plot under the constant rate birth-death model, we consider the derivative of $\log(N_T(t))$. The derivative is the slope of the LTT plot. We obtain,

$$\begin{aligned} \frac{d}{dt} \log(N_T(t=0)) &= \beta + \delta + \frac{\delta(\beta - \delta)}{\delta - \beta e^{(\beta-\delta)T}} \\ \frac{d}{dt} \log(N_T(t=T)) &= \beta - \delta + \frac{\delta(\beta - \delta)^2}{(\delta - \beta e^{(\beta-\delta)T})^2} \end{aligned}$$

For $\beta > \delta$, we now discuss the shape of this plot in the limit. First, we note that $\lim_{T\to 0} \frac{d}{dt} \log(N_T(t=0)) = \beta$, and is monotonously increasing with T, leading to $\lim_{T\to\infty} \frac{d}{dt} \log(N_T(t=0)) = \beta + \delta$. Thus, the initial slope of the complete LTT plot is between β and $\beta+\delta$. Further, $\lim_{T\to 0} \frac{d}{dt} \log(N_T(t=T)) = \beta$, and is monotonously decreasing with increasing T, leading to $\lim_{T\to\infty} \frac{d}{dt} \log(N_T(t=T)) = \beta - \delta$. Thus, the final slope of the complete LTT plot is between β and $\beta - \delta$.

In summary, for $T \to \infty$ and $\beta > \delta$, the slope of the LTT plot is at the start of the process $\beta + \delta$, and decreases with time to $\beta - \delta$. For finite T, a slope decrease with lower initial slope and higher final slope is observed, quantified in Equation (9.8). For $T \to 0$, the slope is β.

9.1.4.3 Phylogenetic LTT plot

Based on Figure 9.6, red, the phylogenetic LTT plot grows exponentially at rate $\beta-\delta$ initially, and close to the present grows exponentially at rate β. This phenomenon is called the *pull-of-the-present* [**Nee1994PullOfPresentLTT**]. An intuitive explanation for the accelerated recent growth is that lineages appearing close to the present have less time to go extinct and thus are more likely to be sampled, leading to an apparent increase in the number of lineages in the phylogenetic tree. Again, we provide the full equation for the red line, $N_{T,p}(t)$.

Theorem 9.1.6. *The expected number of lineages through time in a phylogenetic tree conditioned on non-extinction, $N_{T,p}(t)$ is,*

$$N_{T,p}(t) = \frac{e^{(\beta-\delta)t}(1-p(0|T-t))}{1-p(0|T)}$$

Proof. This result was first presented in [**Harvey1994**] and [**KuboIwasa1995**]. We prove the result analog to [**KuboIwasa1995**].

The expected number of lineages after time t, conditioned on non-extinction, is $\frac{1}{\left(1-\frac{\beta}{\delta}p(0|t)\right)}$ (see Corollary 9.1.2). Thus, $N_{T,p}(T) = \frac{1}{1-\frac{\beta}{\delta}p(0|T)}$. Further, the average number of offspring at time T produced by one individual at time t, $\frac{N_{T,p}(T)}{N_{T,p}(t)}$, is equivalent to the expected number of offspring of one individual after time $T-t$. Using again Corollary 9.1.2, it follows that $\frac{N_{T,p}(T)}{N_{T,p}(t)} = \frac{1}{1-\frac{\beta}{\delta}p(0|T-t)}$. In summary, we obtain $N_{T,p}(t) = \frac{1-\frac{\beta}{\delta}p(0|T-t)}{1-\frac{\beta}{\delta}p(0|T)}$. Since $1-\frac{\beta}{\delta}p(0|t) = e^{-(\beta-\delta)t}(1-p(0|t))$ (see proof of Thm. 9.1.4), we complete the proof. □

In order to investigate how the phylogenetic LTT plot looks under the constant rate birth-death model, we consider the derivative of $\log(N_{T,p}(t))$,

$$\frac{d}{dt}\log(N_{T,p}(t)) = \frac{\beta(\beta-\delta)}{\beta-\delta e^{(\beta-\delta)(t-T)}},$$

and thus,

$$\begin{aligned}\frac{d}{dt}\log(N_{T,p}(t=0)) &= \frac{\beta(\beta-\delta)}{\beta-\delta e^{-(\beta-\delta)T}},\\ \frac{d}{dt}\log(N_{T,p}(t=T)) &= \beta.\end{aligned}$$

For $t=0$ and $\beta>\delta$, we further note that $\lim_{T\to 0}\frac{d}{dt}\log(N_{T,p}(t=0)) = \beta$, and this function is monotonously decreasing with an increasing T, to reach $\lim_{T\to\infty}\frac{d}{dt}\log(N_{T,p}(t=0)) = \beta-\delta$. This last statement is often made without mentioning that T has to go to ∞.

In summary, for $\beta>\delta$, the phylogenetic LTT plot has a slope β at present, and decreases going into the past towards $\beta-\delta$. The slope $\beta-\delta$ is reached for $T\to\infty$.

Recall that all our derivations were for constant birth- and death rates through time. When birth- and death rates are functions of time, the work by [**Kendall1948pn**, **Nee1994reconstructed**, **KuboIwasa1995**] leads to a generalization of the expressions given in Theorem 9.1.1 and Corollary 9.1.2 as well as the expressions for the LTT plots, mostly relying on the concept of generating functions.

9.1.4.4 Parameter estimation with LTT plots

Based on the insights on LTT plots, we can determine the birth and death rates of an empirical phylogenetic tree by displaying its LTT plot, Figure 9.7: each black star corresponds to a branching event, its time is displayed on the x-axis, and the number of lineages in the tree after the branching event is displayed on the y-axis. Given that we see a phylogenetic tree, we first conclude that the population size is growing, and thus $\beta > \delta$. The initial slope of the LTT plot should thus be (in expectation) $\beta - \delta$, and its recent slope should be (in expectation) β (Section 9.1.4.3). Thus, in theory, we can estimate the birth and the death rates from the slopes of two regression lines fitted to the black stars of the empirical LTT plot, as shown in Figure 9.7, bold red lines. One regression is performed on the early part of the LTT plot in order to estimate $\beta - \delta$, and one on the late part in order to estimate β.

However, there are two problems with this method of estimating the parameters of the birth-death model. First, the variance in the timing of the next branching event (i.e. the next star in Figure 9.7) decreases with increasing population size. Thus, a classic linear regression (see Chapter 8) assuming the same variance for each data point is not valid. Second, the time of transition between the two phases of the curve is unclear. This poses a difficulty to the user in deciding where to place the cutoff between points used to fit to the first linear regression line and points used to fit the second linear regression line.

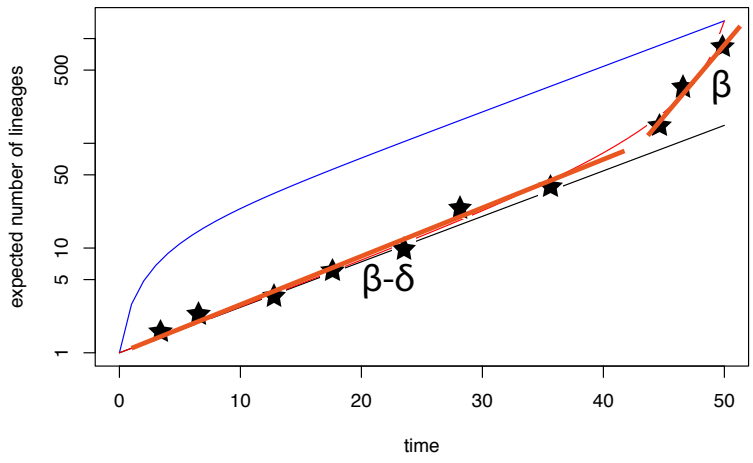


Figure 9.7: The plot shows the fit of two regression lines (bold red) to an empirical phylogenetic LTT plot (black stars).

9.1.5 Distribution on branching times

We now derive the probability density of a labelled phylogenetic tree. Given the uniform distribution on ranked labelled trees, this derivation provides the probability density of branching times, while in the last section we only considered the

expectation of branching times. Based on these derivations, we provide a maximum likelihood approach to estimate the birth- and death rates based on the branching times in a phylogenetic tree. Such an approach will overcome the problems observed when using linear regressions on LTT plots for estimating birth- and death rates.

9.1.5.1 Parameter estimation with maximum likelihood

Recall that in phylogenetics, we calculate the phylogenetic likelihood: $L(\mathcal{T}, Q; D) = P(D|\mathcal{T}, Q)$, for a phylogenetic tree $\mathcal{T}$ and substitution rate matrix Q, given a sequence alignment D. The aim of phylodynamics is to calculate the so-called *phylodynamic likelihood*: $L(\eta = (\beta, \delta, T, \rho, \psi, r); \mathcal{T}) = P(\mathcal{T}|\eta)$. We aim to determine the maximum likelihood estimate for the birth-death parameters (summarized in η), given a fixed phylogenetic tree. Such a maximum likelihood approach – ignoring sampling through time – has first been introduced in [**Nee1994reconstructed**]. In the following we derive $P(\mathcal{T}|\eta)$ based on the assumption of complete extant tip sampling, i.e. $\rho = 1$ and $\psi = 0$.

Consider a phylogenetic tree $\mathcal{T}$ on n extant tips which is obtained from a birth-death model stopped after time T. We will now measure time in reverse direction. In particular, we set present time to be 0. The $n-1$ branching events occur at time $x_1 > x_2 > \ldots > x_{n-1}$ prior to present (note that since our stochastic process is continuous in time with constant rates, the probability of two branching events to occur at exactly the same time is 0). Finally, we define the time of the start of the process as $x_0 = T$. For facilitating the mathematical derivation, we label the two descendants of each branching event with "left" and "right", while the tips are not labelled. Such trees are also called *oriented trees*, while the phylogenetic trees considered so far were labelled trees. For an example of an oriented tree with four tips, see Figure 9.8.

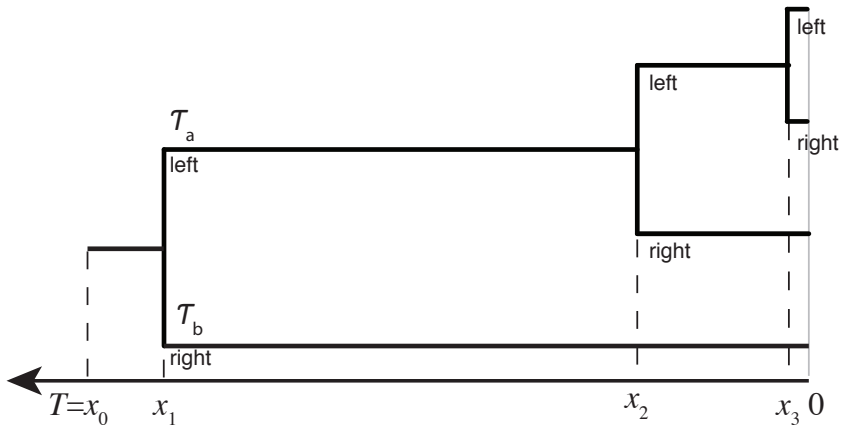


Figure 9.8: Example oriented tree $\mathcal{T}^o$ on four tips with the two subtrees $\mathcal{T}_a^o$ and $\mathcal{T}_b^o$. We reverse time such that present time is 0 and the first individual appeared at T time units in the past.

Suppose this example tree evolved under a constant rate birth-death model, without death, i.e. $\delta = 0$. Then, the probability density of this oriented tree $\mathcal{T}^o$ is a product of exponentials and rates,

$$\begin{aligned} P(\mathcal{T}^o|\beta, \delta = 0, T = x_0, \rho = 1, \psi = 0, r) &= e^{-\beta(x_0-x_1)}\beta \underbrace{e^{-\beta x_1}}_{\mathcal{T}_b^o} \underbrace{e^{-\beta(x_1-x_2)}\beta e^{-\beta x_2} e^{-\beta(x_2-x_3)}\beta e^{-2\beta x_3}}_{\mathcal{T}_a^o} \\ &= \beta^3 \prod_{i=0}^{3} e^{-\beta x_i} \end{aligned}$$

where $\mathcal{T}_a^o$ is the left and $\mathcal{T}_b^o$ is the right subtree descending branching time x_1.

For $\delta > 0$, the probability density calculation is more complicated: we need to take into account all possible unobserved events leading to extinct individuals. If we were to know all these events, we could again calculate the probability of the complete tree through a product of exponentials and rates. Since we do not know all unobserved events, we need to sum over them. This will be done below employing differential equations. Alternatively, we could sum up the probability density of all complete trees having embedded our oriented phylogenetic tree, but this will be too slow in practice as there are infinitely many such complete trees.

The probability density of the example oriented tree in Figure 9.8 can be calculated as the product of the probability density of the first branch between x_0 and x_1, the probability density of a branching event at time x_1, and the probability density of the two subtrees $\mathcal{T}_a^o$ and $\mathcal{T}_b^o$ descending from time x_1. Here, we employ the property of the birth-death model occurring independently in different parts of the tree. Generally, let $p(x_0, x_1)$ be the probability density of an individual to produce a branch of length $x_0 - x_1$ at time x_0 in the past. Then, the probability density of an oriented tree $\mathcal{T}^o$ with age $T = x_0$ is,

$$P(\mathcal{T}^o|T = x_0) = p(x_0, x_1)\beta P(\mathcal{T}_a^o|T = x_1)P(\mathcal{T}_b^o|T = x_1).$$

Note that we omit the parameters $\beta, \delta, \rho = 1, \psi = 0, r$ in the expression for the probability density of the tree in order to facilitate readability. We can continue expanding this expression recursively until we come to the tips of the tree, meaning the tree probability density is a product of probability densities of branches and birth rates.

To calculate the probability density of the branch between t and x_1, $p(t, x_1)$ (where $t \geq x_1$), we write down the Master equation using the same considerations as in the derivation of $p(1|t)$,

$$p(t + \Delta t, x_1) = (1 - (\beta + \delta)\Delta t)p(t, x_1) + 2\beta \Delta t p(t, x_1)p(0|t) + \mathcal{O}(\Delta t^2),$$

which after rearranging and taking the limit $\Delta t \to 0$ gives us the differential equation,

$$\frac{d}{dt}p(t, x_1) = -(\beta + \delta)p(t, x_1) + 2\beta p(t, x_1)p(0|t) \ .$$

This is the same differential equation as for $p(1|t)$. The initial condition is $p(x_1, x_1) = 1$, as, given we have an individual in the tree at time x_1, it induces with probability one a branch of length zero. Thus, $p(t, x_1) = \frac{p(1|t)}{p(1|x_1)}$, and in particular $p(x_0, x_1) = \frac{p(1|x_0)}{p(1|x_1)}$.

Overall, we thus obtain,

$$P(\mathcal{T}^o|T = x_0) = p(x_0, x_1)\beta P(\mathcal{T}_a|T = x_1)P(\mathcal{T}_b|T = x_1) = \beta^{n-1}\prod_{i=0}^{n-1} p(1|x_i),$$

using the observation that each internal node is once the ending and twice the starting point of a branch. As we only consider biological units from which we indeed have samples, the probability density is further conditioned on obtaining samples. For our scenario $\psi = 0, \rho = 1$, this is equivalent to non-extinction, and leads to,

Theorem 9.1.7. *Consider the constant rate birth-death model for time T with birth rate β and death rate δ. Further, consider complete extant tip sampling ($\rho = 1$) and no sampling through time ($\psi = 0$). The probability density of an oriented tree $\mathcal{T}^o$, conditioned on non-extinction ($X_T > 0$), is,*

$$P(\mathcal{T}^o|T = x_0, X_T > 0, \beta, \delta) = \frac{\beta^{n-1}}{1 - p(0|t)}\prod_{i=0}^{n-1} p(1|x_i).$$

Analogous probability densities for different types of conditioning are provided in [**Stadler2013POV**].

In order to obtain the probability density of a labelled phylogenetic tree $\mathcal{T}$ on n samples, we note that we can label the samples of an oriented tree in $n!$ ways with each labelling having the same probability. Further, we note that given a tree on $m - 1$ internal nodes, for a particular labelling, there are 2^{m-1} orientations (left or right for each of the $m - 1$ internal nodes). In the absence of sampled ancestors, we have $n = m$. Thus,

Corollary 9.1.8. *The probability density of a labelled tree $\mathcal{T}$, with $\rho = 1, \psi = 0$ and conditioned on non-extinction, is,*

$$P(\mathcal{T}|T = x_0, X_T > 0, \beta, \delta) = \frac{2^{n-1}}{n!}\frac{\beta^{n-1}}{1 - p(0|t)}\prod_{i=0}^{n-1} p(1|x_i).$$

In Theorem 9.1.3, we showed that each ranked labelled tree topology has the same probability. Thus,

Corollary 9.1.9. *The probability density of the branching times $x_1 > x_2 > \ldots > x_{n-1}$, meaning the probability density of the LTT plot, with $\rho = 1, \psi = 0$ and condi-*

tioned on non-extinction, is,

$$P(x_1, x_2, \ldots, x_{n-1}|T = x_0, X_T > 0, \beta, \delta) = (n-1)! \frac{\beta^{n-1}}{1 - p(0|t)} \prod_{i=0}^{n-1} p(1|x_i).$$

We can now perform a maximum likelihood inference on the parameters β and δ based on the branching times in the phylogenetic tree. The parameter T is typically being fixed, it is the stem age of a group of species in the macroevolutionary application, or the start of an epidemic in the epidemiological application. Alternatively, the probability of the branching times conditioned on the first branching event (x_1) can be derived (for an overview, see e.g. [**Stadler2013POV**]). The values for β and δ maximising the probability density $P(x_1, x_2, \ldots, x_{n-1}|T = x_0, X_T > 0, \beta, \delta)$ are the maximum likelihood parameter estimates of the birth-death model for the given branching times. We note that in fact the expressions in Theorem 9.1.7 and Corollaries 9.1.8 and 9.1.9 all lead to the same maximum likelihood parameter estimates, as these expressions only differ in a function which depends on the number of samples, n, but not on the tree. Recall that while this maximum likelihood approach considers the probability density of the LTT plot, the linear regression approach from the last section used properties of the expected LTT plot.

We note that in this chapter, we quantify the birth- and death rates based on a given phylogenetic tree $\mathcal{T}$. In particular, the maximum likelihood birth- and death rate estimates are $(\hat{\beta}, \hat{\delta}) = \text{argmax}_{\beta,\delta} P(\mathcal{T}|T = x_0, X_T > 0, \beta, \delta)$. The interested reader may wonder how to quantify the birth- and death rates based on sequencing data D,

$$(\hat{\beta}, \hat{\delta}) = \text{argmax}_{\beta,\delta} P(D|T = x_0, X_T > 0, \beta, \delta).$$

So far, we did not establish an expression for $P(D|T = x_0, X_T > 0, \beta, \delta)$. However, we can re-write

$$P(D|T = x_0, X_T > 0, \beta, \delta) = \int_{\mathcal{T}} P(D, \mathcal{T}|T = x_0, X_T > 0, \beta, \delta) d\mathcal{T} = \int_{\mathcal{T}} P(D|\mathcal{T}) P(\mathcal{T}|T = x_0, X_T > 0, \beta, \delta) d\mathcal{T}.$$

In the last equation, we assumed that the sequence evolution process is independent from the tree generation process. While we can calculate the two expressions within the last integral, we cannot efficiently integrate over all trees (see Section 6.2.3.3 on the size of the tree space). In Chapter 10, we will encounter algorithms to deal with this integral.

9.1.5.2 General birth-death models

As mentioned above, the probability density of a birth-death tree was initially calculated under $\psi = 0$ [**Nee1994reconstructed**], this paper further allowed time-dependence of the birth- and death parameters. Over the past few years, the probability densities for phylogenetic trees have been derived for a number of extensions of this basic birth-death model. The derivations for the generalizations rely on the ideas introduced above.

Allowing for sampling through time with $r = 0$ (recall that r is defined as being the probability for death of a sampled individual prior to the present) has been introduced in [**Stadler2010STT**]. This extension essentially requires a change of initial conditions in the differential equations. Allowing for the birth- and death parameters to change through time with $r = 1$ has been done in [**StadlerEtAl2013PNAS**]. Furthermore, this work provided the possibility to assume a sampling probability ρ_i at some time points t_i in the past. These extensions rely on changing the parameters in the differential equations through time. Using such models with any $r \in [0, 1]$ has been done in [**GavryushkinaEtAl2014**]. Allowing for competition among co-existing individuals has been accounted for in [**LeventhalEtAl2014**, **VaughanEtAl2018**].

Finally, these scenarios have been generalized to a multi-state birth-death model in [**StadlerBonhoeffer2013**, **KuehnertEtAl2016**], where we write down separate differential equations for each state. For this scenario, we do not obtain a uniform probability on ranked labelled tree topologies any more, thus the tree topology and the branching times together inform about the birth-death parameters. This model will be discussed in more detail in Section 9.4.

Under the scenario of only extant tip sampling (i.e. $\psi = 0$), a range of models have been introduced with macroevolutionary applications in mind [**MaddisonEtAl2007**, **FitzjohnEtAl2009**, **GoldbergEtAl2011**, **Morlon2011EtAl**, **EtienneEtAl2011**, **GoldbergIgic2012CLASSE**, **rabosky2014BAMM**].

9.1.6 Applications

9.1.6.1 Epidemiology - Quantification of the spread of the West African Ebola epidemic

In epidemiology, a key quantity of interest is the basic reproductive number R_0. R_0 is defined as the expected number of secondary infections caused by a single infected individual introduced into an entirely susceptible population. The value of this parameter is a strong indicator for the fate of an epidemic: if $R_0 < 1$, the epidemic will eventually die out. If $R_0 > 1$, then the infected population size will on average increase and the epidemic spreads. Furthermore, the value of R_0 indicates the amount of public health effort required to contain an epidemic already in progress. Assuming the constant rate birth-death model, the basic reproductive number can be calculated as $R_0 = \beta/(\delta + r\psi)$ ([**GavryushkinaEtAl2014**], more generally, see e.g. [**keeling2011**] for an overview on modelling epidemics).

In what follows, we calculate R_0 for the West African Ebola outbreak 2013-2016. In August 2014, 72 Ebola genomes from different patients in a Sierra Leone outbreak were published [**Gire2014**]. The publication also included the phylogenetic tree, shown in Figure 9.9. Here, we estimate R_0 based on this tree. As these genomes were sampled early in the epidemic, we assume a constant transmission rate, β, and constant becoming-uninfectious rate (i.e. rate of recovering or dying) without

sampling δ, for the entire tree. The parameters β and δ correspond to the birth and death rate in the constant rate birth-death model, respectively. No samples from the present were available, so we set the extant tip sampling probability $\rho = 0$. Using estimates from other studies, we set the sampling probability prior to the present to $\frac{\psi}{\delta+\psi} = 0.7$ (meaning $\psi = \frac{7}{3}\delta$) and $r = 1$.

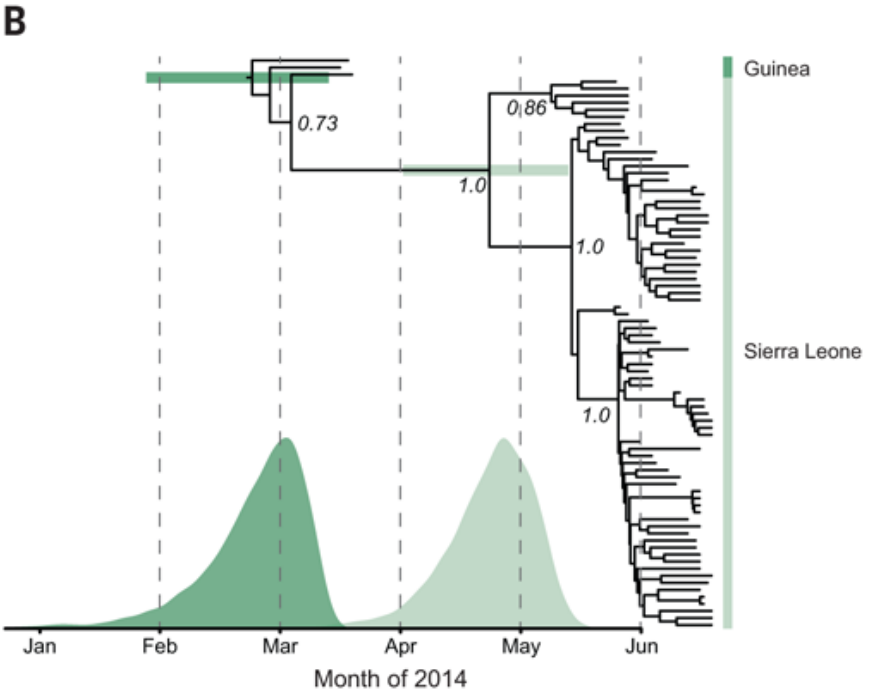


Figure 9.9: Phylogenetic tree of Ebola genomes in West Africa. Figure adapted from [**Gire2014**].

We obtain the maximum likelihood estimates $\hat{\beta}$ and $\hat{\delta}$ using the phylodynamic likelihood, and through this we calculate $R_0 = \frac{\hat{\beta}}{\frac{10}{3}\hat{\delta}} = 1.34$ with the confidence interval $CI = [1.12, 1.55]$ [**Stadler2014PLOSCur**].

In [**Stadler2014PLOSCur**], we used a fixed tree from [**Gire2014**] and therefore we ignored any uncertainty in the tree. Phylogenetic trees obtained from pathogen sequences from an epidemic outbreak often show high uncertainty in the estimated tree (see Section 7.4.5 for assessing uncertainty). In Chapter 10, we will see how Bayesian methods can provide phylodynamic parameter estimates (such as estimates for β, δ, R_0) by taking this phylogenetic uncertainty into account.

9.1.6.2 Macroevolution - Estimation of diversification rate changes through time in mammals

About 65 million years ago, a meteorite hit the planet Earth and caused a mass extinction, in particular the extinction of dinosaurs. Palaeontologists hypothesize that this event was followed by a period of increased mammalian diversification [**Archibald2001**], with diversification rate = speciation rate - extinction rate.

In what follows, we investigate if the phylogenetic tree of mammals also supports increased diversification of mammals after the extinction of dinosaurs [**Stadler2011**]. We use the phylogenetic tree of mammals from [**Bininda2007**] (shown in Figure 6.2). The birth-death model was applied to this tree, with a model extension to allow for changes in parameters through time: rates are constant until time t_1, then change to other constant rates until time t_2, etc.

The results, presented in Figure 9.10, show that the maximum likelihood diversification rate $\hat{\beta} - \hat{\delta}$ was roughly 0.05 until 35 million years ago, where there was a peak followed by a decline in diversification rate[1]. As you can see from Figure 9.10, the analysis shows no evidence for an increase or decrease in diversification rate around 65 million years ago.

The uncertainty in the estimate of the diversification rate can be assessed by constructing the confidence region (using a χ^2 distribution, as discussed in Box 19). In [**Stadler2011**], we however focused on a parametric bootstrapping approach (Section 7.4.4). Multiple trees were simulated using the estimated maximum likelihood parameters. Based on these simulated trees, birth-death parameters were re-estimated from the simulated phylogenies. Re-estimated maximum likelihood diversification rates are consistent with the original estimate, as can be seen in Figure 9.10.

Phylogenetic analysis based on a different mammalian phylogenetic tree [**Meredith2011**] did not show evidence for an increase or decrease in diversification rate around 65 million years ago either. Why does the phylogenetic inference disagree with the hypothesis put forward by the palaeontologists? It is still an open area of research, and we hope to shed light on this question by jointly analyzing fossil and phylogenetic tree data [**zhang2016**, **Gavryushkina2017**].

[1] Note that the peak observed 35 million years ago may be a flaw of the analysis. In [**Bininda2007**], the authors pulled together unresolved lineages, which potentially led to too many diversification events around 35Ma, and thus to an overestimate of diversification around that time point.

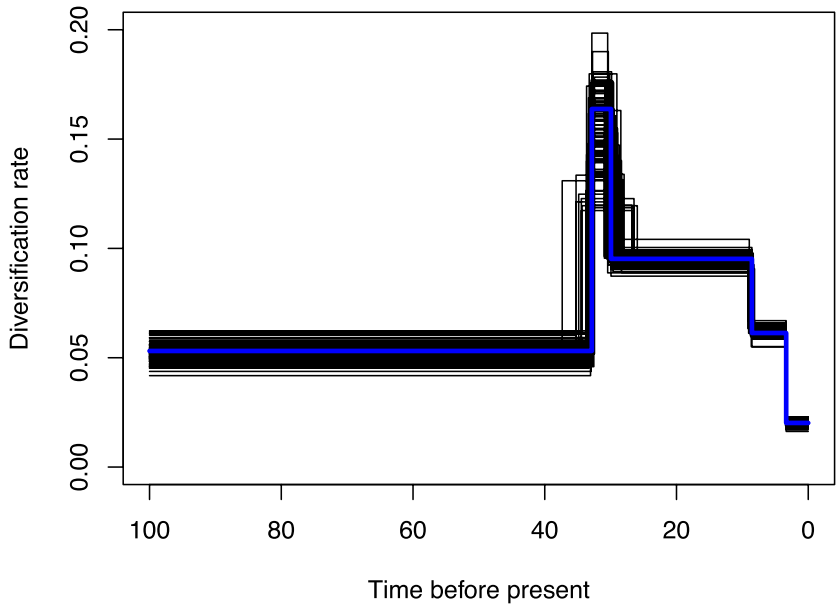


Figure 9.10: Maximum likelihood diversification rate ($\hat{\beta} - \hat{\delta}$) estimate through time (blue). The black lines indicate the parametric bootstrap interval: birth-death trees were simulated using the maximum likelihood parameters, and then the diversification rate of each tree was re-estimated and plotted in black. There is no signal for elevated diversification upon dinosaur extinction (65 Ma). Figure adapted from [**Stadler2011**].

9.2 Coalescent Theory

While the birth-death models described in the previous section allow the population size to vary stochastically as a result of the parameters of the phylogenetic model, a second class of so-called *coalescent* models instead treat the population size as a given. The original coalescent model was introduced by John Kingman in 1982 [**Kingman1982**] as a way to predict allele frequency dynamics. More recently, this model and its descendants have been widely applied to phylogenetic and phylodynamic inference. In this context, coalescent models very naturally allow the population size itself to become an explicit target of the inference.

9.2.1 The Wright-Fisher process

In order to develop a quantitative connection between population size and phylogeny, it is necessary to provide a model for the propagation of heritable traits within a population of fixed size. One such model is provided by the famous Wright-Fisher process, which has been a cornerstone of population genetics since its introduction by the founders of the field to model genetic drift.

In the classic Wright-Fisher process we assume a constant population composed of N individuals. While these individuals may differ in genetic make-up or in other ways, the model itself is completely blind to these differences, treating every individual equally.

The model assumes discrete, non-overlapping generations. Each member of a given generation is assumed to have exactly one parent in the previous generation. Thus, the model is most directly applicable to population elements undergoing asexual reproduction. That said, these elements might be genes, or other genetic elements that while belonging to sexually-reproducing organisms can be themselves treated as reproducing in an asexual fashion.

The choice of parent from the previous generation is assumed to be completely random. Importantly, this means that the choice is independent of any trait associated with the individual or its parent. It is therefore a neutral model, as any selection of parents based on fitness criteria is not allowed under the model. Correspondingly, each parent may have zero, one or many children in the next generation, but the total number of children must be that prescribed by the population size.

To illustrate these concepts, figure 9.11 represents the input to the WF model: i.e. the number of individual members of a population in each generation through time. Figures 9.12a and 9.12b represent what can be considered the output of the WF model. Figure 9.12a represents a particular realization of the WF process, which results in the assignment of children to parents between generations of the population. Figure 9.12b highlights the ancestry of an arbitrarily chosen triplet of individuals in the final generation. This is the true phylogenetic relationship between those three individuals implied by the particular outcome of the WF process in 9.12a.

As it is the sampled tree (bold lines in figure 9.12b) which produces the genetic data

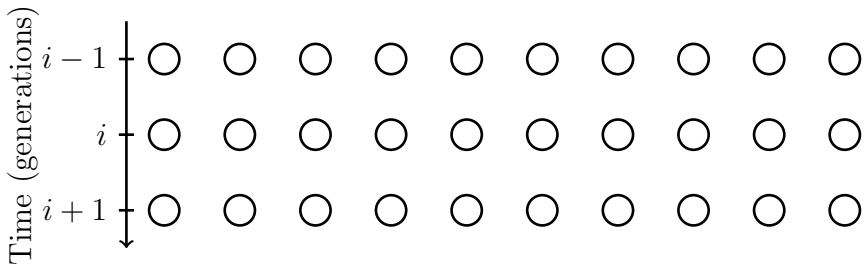


Figure 9.11: The Wright-Fisher process is a population genetic model which takes a sequence of discrete generations of a population with constant size as input. For example, the above schematic represents three generations of such a population, with each circle representing an individual member of the population, and each row of circles representing the members of a single generation.

we observe when we sample a subset of the full population, it is of central importance to determine the relationship between population size and the shape of this tree. In the simplest case of two samples drawn at random from the present population, this reduces to the following question: what is the probability for the most recent common ancestor (MRCA) of these two samples occurring at m generations before the present?

In order to answer this, consider that:

1. Since each individual picks their parent uniformly at random, the probability that two individuals in the same generation have the same parent is $\frac{1}{N}$.

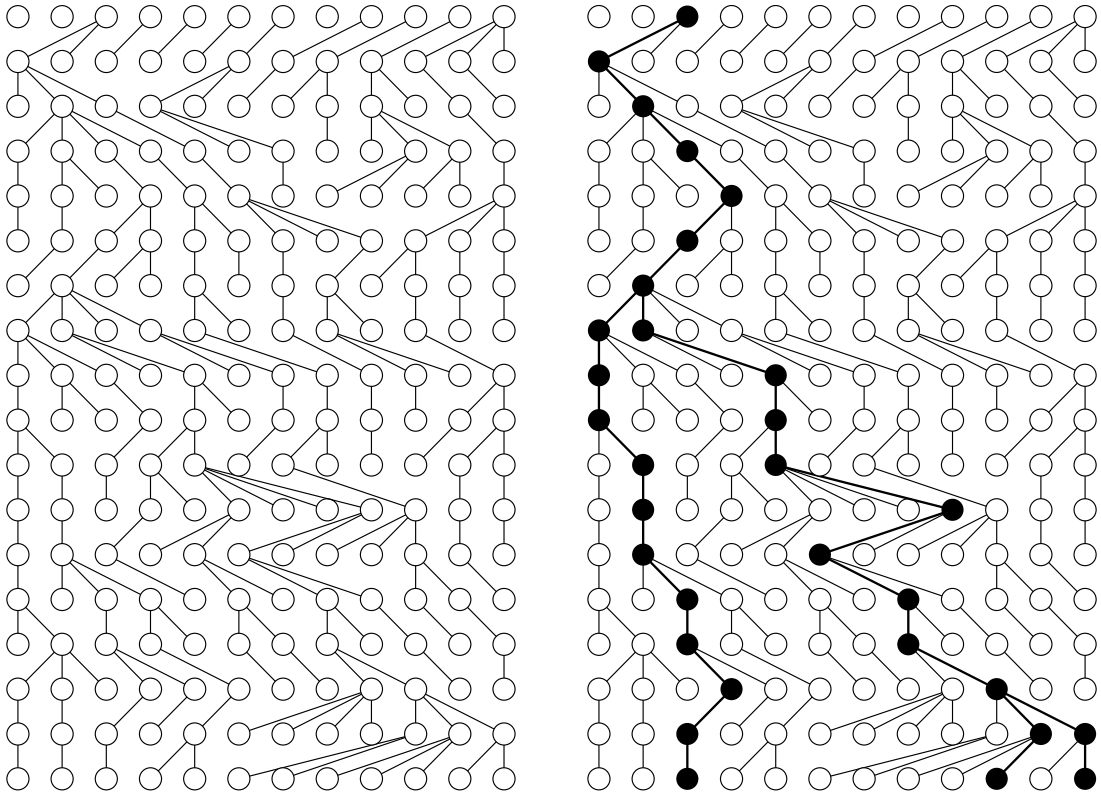
2. Thus, the probability that two individuals in the same generation do not have a common ancestor in the previous generation is $\left(1 - \frac{1}{N}\right)$.

Thus, the probability for the two sampled individuals to first share a common ancestor in the m^{th} generation before the present is the product of the probability of no common ancestor in the first $m-1$ generations and the probability of a common ancestor in the m^{th} generation before the present:

$$P_{MRCA}(m) = (1 - \frac{1}{N})^{m-1}\frac{1}{N}. \tag{9.8}$$

This is simply a geometric distribution (Box 5) with a success probability of $1/N$. Since the mean of such a distribution is equal to the inverse of the success probability, we must wait on average N generations to see a common ancestor of two samples from a population of size N.

This result can of course be extended to develop the full probability of a larger sampled tree under the discrete time WF model. In such a tree, internal nodes would occur at integer generation numbers and could involve more than two child lineages. Instead however, we will at this point leave the discrete time WF model and begin to develop an approximate model for continuous-time binary sampled



(a) (b)

Figure 9.12: Outcome of the Wright-Fisher process across multiple generations: the full inheritance tree (9.12a), where each individual chooses its parent uniformly at random from the parent generation above, and the tree reconstructed from a random sample of 3 individuals in the last generation (9.12b). Note that while the lines representing parent-child relationships may in general cross, it is always possible to reorder the members of each row in such a way that these lines do not cross—as has been done in this figure.

trees.

9.2.2 Kingman's Coalescent Process

Kingman's coalescent (also known as Kingman's n-coalescent, where the n refers to the number of sampled lineages the process begins with) is a continuous-time Markov chain (see Box 17) which produces time trees. The process runs backwards in time, building the tree through successive pairwise merging events known as "coalescence" events, or simply "coalescences". There are in fact several distinct population genetic models for which Kingman's coalescent arises as a limiting case, we focus here on developing these details using the WF model introduced in the previous section.

9.2.2.1 Coalescence rate between two lineages

To do this, we consider once again a pair of individuals sampled from a particular generation of a WF population. We saw in the previous section that, under the assumptions of the WF model, the probability of these individuals sharing a parent in the previous generation is $1/N$. In the corresponding coalescent model, we consider what happens when N is very large, meaning that the probability of a pair of lineages finding a common ancestor in any given generation becomes extremely small. As discussed in Box 15, this is the same limit in which a geometric distribution approaches an exponential distribution.

To see this explicitly, define t_c to be the exact time at which two lineages share a parent and define g to be the fixed length of time between successive generations. In terms of these variables, the exact probability for this merging event under the WF model is

$$P(t_2|N,g) = \left(1 - \frac{1}{N}\right)^{t_2/g-1} \frac{1}{N} \tag{9.9}$$

In order to find the continuous time limit, we define the pairwise coalescent rate $\theta = 1/Ng$ to be probability per unit time for a given pair of ancestral lineages to coalesce. Writing the coalescence probability above in terms of this rate yields

$$\begin{aligned} P(t_2|\theta) &= \left(1 - \frac{1}{N}\right)^{\theta t/N-1} \theta g \\ &= \left[\left(1 - \frac{1}{N}\right)^{N}\right]^{\theta t} \left(1 - \frac{1}{N}\right) \theta g \end{aligned} \tag{9.10}$$

Before taking the limit, we define the probability *density* of the coalescence occurring at time t_c as

$$f_{t_c}(t|\theta) = P(t_c = t|\theta)/g = \left[\left(1 - \frac{1}{N}\right)^{N}\right]^{\theta t} \left(1 - \frac{1}{N}\right) \theta \tag{9.11}$$

Keeping the coalescence rate θ fixed and taking the limit of large N (small g) yields

$$\begin{aligned} \lim_{N\to\infty} f_{t_c}(t|\theta) &= \left[e^{-1}\right]^{\theta t} \cdot 1 \cdot \theta \\ &= e^{-\theta t}\theta, \end{aligned} \tag{9.12}$$

where we have used the result from Box 13. Thus, in the large N limit, the time t_c for two lineages to coalesce is exponentially distributed with mean $\theta = \frac{1}{Ng}$. This probability was first calculated by John Kingman in 1982 [**Kingman1982**].

9.2.2.2 Coalescence rate between more than two lineages

After this limit has been taken, the probability per unit time for a coalescence to occur between any pair of k lineages where $k \geq 2$ follows immediately. The time

of this event is the minimum of the $\binom{k}{2}$ pairwise coalescent times, each of which is exponentially distributed with rate $1/Ng$. We can thus make use the result given in Box 14 to find that the minimum is itself exponentially distributed with rate equal to the sum of the pairwise rates, i.e.

$$f(t_k|Ng) = \exp\left[-\binom{k}{2}\frac{t_k}{Ng}\right] \cdot \binom{k}{2}\frac{1}{Ng}.$$

As usual, we can interpret this as the product between the probability of no coalescence occurring in time t_k (the exponential function) and the probability density of a coalescence occurring immediately after.

This expression is exact in the limit of infinite N. However, when treated as an approximation for the probability density of a coalescence time between k lineages belonging to a finite population under the WF model, it is clear our reasoning above omits the possibility of three or more lineages coalescing simultaneously. Such events occur with non-zero probability under the WF model, giving rise to trees in which internal nodes may have more than two child lineages. These events become occur most frequently when the number of lineages k under consideration approaches N.

For this reason, when it is treated as an approximation to the WF model, an additional assumption of the coalescent is that the number of extant lineages k in the sampled phylogeny is small compared to N at all times.

9.2.2.3 The coalescent process and the probability density of a coalescent tree

So far we have derived the probability distributions for the time taken for different numbers of sampled lineages to coalesce. This is all that is required to define the *coalescent*: a stochastic process which produces sampled coalescent trees. This process is a continuous time Markov chain (Box 17) on the lineages extant at a particular time. It proceeds from the present into the past producing a coalescent events, each of which merges a randomly chosen pair of extant lineages into a new internal tree node, decrementing the number of ancestral lineages by 1. The process terminates when only a single lineage remains.

The probability (density) of a tree generated by this process can be expressed in terms of a product between the probability for the time intervals between coalescent events and the probability densities of coalescences occurring at the times corresponding to the internal nodes. For example, the probability for the tree given

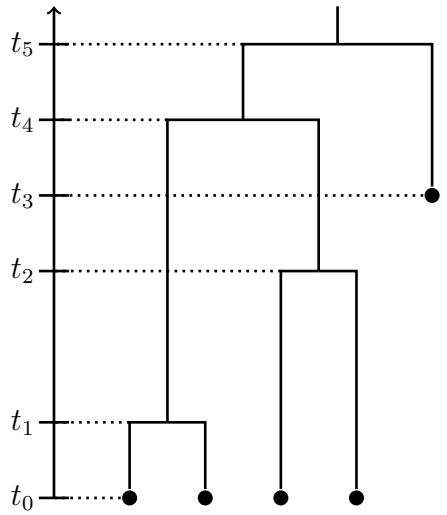


Figure 9.13: A coalescent tree with event times (both coalescent and sample times) marked.

figure 9.13 can be written

$$\begin{aligned} f(T|Ng) = &\exp\left[-(t_1-t_0)\binom{4}{2}\frac{1}{Ng}\right]\frac{1}{Ng} \\ &\times \exp\left[-(t_2-t_1)\binom{3}{2}\frac{1}{Ng}\right]\frac{1}{Ng} \\ &\times \exp\left[-(t_3-t_2)\binom{2}{2}\frac{1}{Ng}\right] \\ &\times \exp\left[-(t_4-t_3)\binom{3}{2}\frac{1}{Ng}\right]\frac{1}{Ng} \\ &\times \exp\left[-(t_5-t_4)\binom{2}{2}\frac{1}{Ng}\right]\frac{1}{Ng} \end{aligned} \tag{9.13}$$

You might notice in the above expression is that the binomial coefficients appear only in the exponential functions representing the probability of no coalescence in each interval, but do not appear in the factors to the right of the exponentials. This is because we distinguish here between coalescences involving different pairs of lineages: the probability of a *particular* pair of k lineages coalescing a factor of $\binom{k}{2}$ less than the probability of *any* pair coalescening. (A tree in which this distinction is made is known as a *labelled tree*.)

Additionally, the third line is missing the coalescent probability density term entirely. This is because, for this particular tree, t_3 corresponds to a *sample* event. The coalescent process conditions on such events explicitly, thus the event doesn't contribute any probability term, but merely increases the number of lineages by 1.

In general, the probability for a tree T can be expressed in terms of intervals between consecutive coalescent or sampling events. For a tree with n leaves and m sampling

events there are $m+n-1$ such events. Defining t_i to be time of event i, k_i to be the number of lineages immediately following this time, and ν_i to be 1 if the event i is a coalescent event and 0 otherwise, we have

$$P(T|Ng) = \prod_{i=1}^{m+n-1} e^{-(t_i - t_{i-1})\binom{k_{i-1}}{2}\frac{1}{Ng}} \left(\frac{1}{Ng}\right)^{\nu_i} \tag{9.14}$$

This probability distribution may be interpreted as the likelihood for the product Ng. Thus it allows us to infer population size based solely on a phylogenetic tree, assuming the conditions of the Wright-Fisher model are met.

It is important to emphasize that while the coalescent process for the sampled tree proceeds backwards in time, the population genetic models (such as the WF model) from which the coalescent is derived describe the evolution of the population forward in time as usual.

9.2.2.4 The maximum height of a coalescent tree

An interesting consequence of the coalescent process is that the expected time required for n lineages to coalesce into 1, i.e. the expected age of a coalescent tree with n leaves *sampled at the present* is

$$\begin{aligned}
E[t_{root}] &= \sum_{k=2}^{n} \frac{Ng}{\binom{k}{2}} \\
&= Ng \sum_{k=2}^{n} \frac{2}{k(k-1)} \\
&= 2Ng \sum_{k=2}^{n} \left(\frac{1}{k-1} - \frac{1}{k}\right) \\
&= 2Ng \left[\sum_{k=1}^{n} \frac{1}{k} - \sum_{k=2}^{n} \frac{1}{k}\right] \\
&= 2Ng(1 - \frac{1}{n})
\end{aligned}$$

which approaches the upper bound of $2Ng$ as the number of samples increases.

Of course, for leaf nodes sampled through time, the root may be arbitrarily old compared to the most recent sample.

9.2.3 Effective population size

Of course, real populations evolve in ways which are far more complex than the Wright-Fisher model. For instance, real populations usually involve non-random mating, can introduce structure into the population. For this reason, when population size is estimated using methods which assume such a model, we refer to the

result as the *effective population size*. Loosely speaking, it is the size of an idealized Wright-Fisher population model having the same genetic diversity as our actual population. It is usually proportional to the true population size, but the scaling factor is usually unknown and may even be time-dependent.

9.2.4 Population dynamics

So far we have only considered a constant population size N. However, the size of real populations usually changes through time. An obvious extension is therefore to replace this constant with a time-dependent population size function $N(t)$. For instance, we might define an exponentially growing population $N(t) = e^{-rt}N_{\text{present}}$, where r is the growth rate and N_{present} is the population size at the present. (The minus sign in the exponential is because t increases backwards in time.) These population size changes over time will impact the shape of the sampled trees.

Incorporating time dynamics into the coalescent is straight-forward. For instance, it was shown by **Griffiths1994** that in the coalescent limit, the coalescent rate between a pair of lineages in a WF model with a time-dependent population size is simply $1/N(t)g$. (Here $N(t)$ is the continuous-time large population size limit of the discrete-time WF population function.) That is, the coalescence rate increases during periods when the population is small, and decreases in periods where it is large. The probability of a labeled tree then becomes

$$P(T|N(t)g) = \prod_{i=1}^{m+n-1} \exp\left[-\binom{k_{i-1}}{2}\int_{t_{i-1}}^{t_i}\frac{dt}{N(t)g}\right]\left(\frac{1}{N(t_i)g}\right)^{\nu_i} \tag{9.15}$$

An example of the effect of population size variation on the shapes of trees is illustrated in figure 9.14. The larger the population size, the larger the waiting time to a coalescent event (since the continuous-time rate of coalescence $\frac{1}{N(t)g}$ will be smaller). Therefore, if we compare an exponentially growing population of size $N_1(t)$ with a population of constant size N_2, and both have the same present day population size $N_1(0) = N_2$, then all coalescent rates in the exponentially growing population will be larger than or equal to the coalescent rates in the population of constant size, leading to shorter trees in the exponential scenario.

Consequently, the timing of coalescent events in the phylogenies reconstructed from the sampled sequences can inform us about the total population size changes over time.

9.2.4.1 Non-parametric inference of population dynamics

What if we do not know (or do not want to assume) that the population dynamics are governed by a particular parametric model? In this case we can use so-called *non-parametric* methods, which use models in which the number of free parameters grows as the number of samples grows.

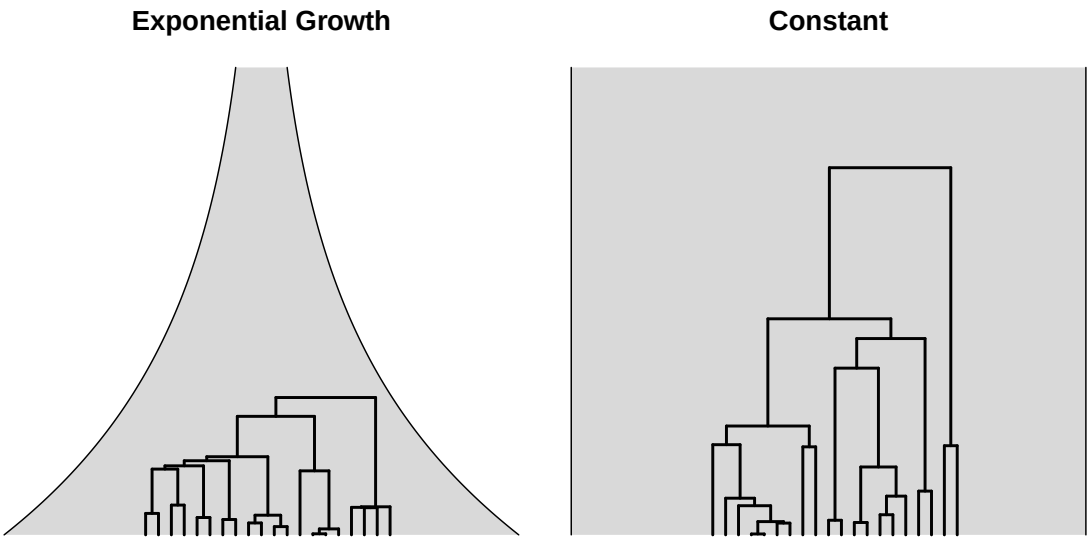


Figure 9.14: Trees generated by coalescent processes under different population dynamics. The left figure shows displays a tree generated by a coalescent assuming an exponential growth model. The right figure shows a tree generated by a coalescent assuming a constant population size model. The width of the grey background indicates the effective population size at different times.

The most well-known example of this is the Skyline Plot developed by **Pybus2000**. In this model, the probability of a tree with all leaves at the present (i.e. $m = 0$) is given by [2]

$$P(T|\vec{N}) = \prod_{i=1}^{n-1} \exp\left[-(t_i - t_{i-1})\binom{k_{i-1}}{2}\frac{1}{N_i}\right]\frac{1}{N_i} \tag{9.16}$$

where M is a vector of length $n - 1$.

Notice that, the population size here is defined by a vector $\vec{N}$ with as many elements as there are coalescent events in the tree. Thus, this model is equivalent to using a population function where the population is constant within each time interval between merging events, as illustrated in figure **??**.

As for other models, we can treat this tree probability $P(T|\vec{N})$ as the likelihood $L(\vec{N}|T)$ for the elements of $\vec{N}$. This allows us to construct a maximum likelihood estimate of the population dynamics given a tree. To see this, note that the full likelihood for $\vec{N}$ can be written as the product of the likelihoods for the individual elements:

$$L(\vec{N}|T) = \prod_{i=1}^{n-1} L(N_i) \tag{9.17}$$

[2]The original model described by **Pybus2000** was expressed slightly differently, but is equivalent to what we present here.

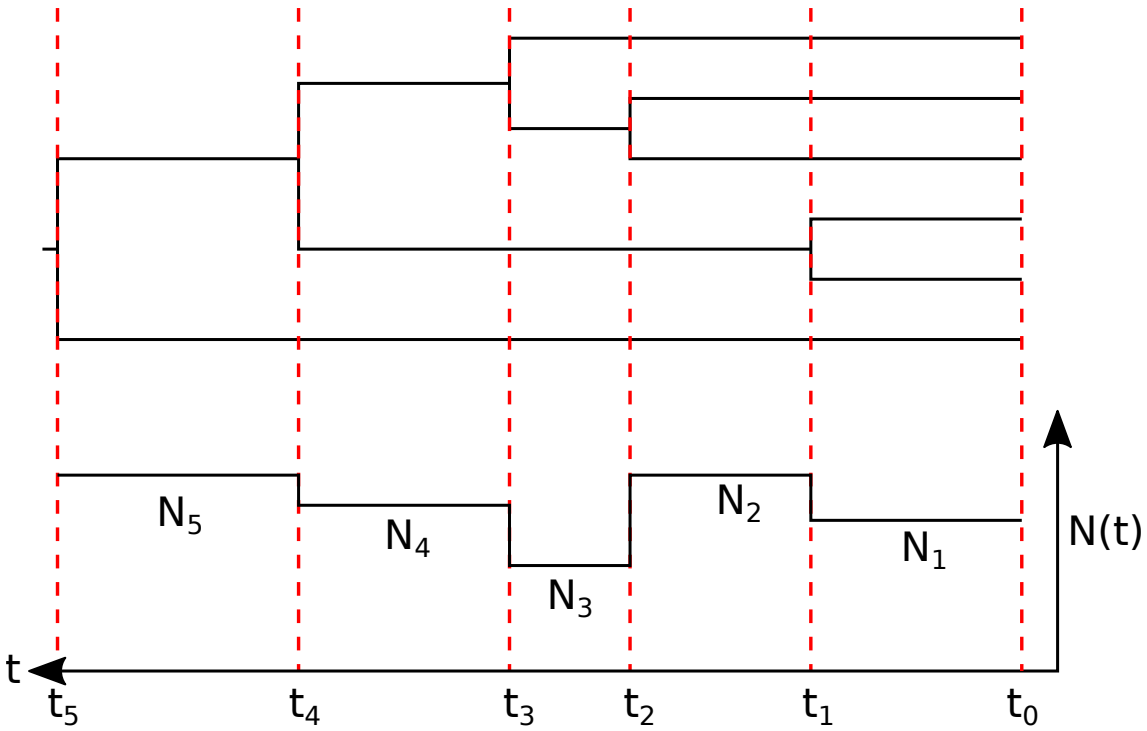


Figure 9.15: Non-parametric inference of population dynamics using the Skyline plot, which allows each interval between coalescence events to possess a distinct population size.

where

$$L(N_i) = \exp[-(t_i - t_{i-1})\binom{k_{i-1}}{2}\frac{1}{N_i}]\frac{1}{N_i}. \tag{9.18}$$

Then we define $\hat{N}_i$ to be the maximum likelihood estimate of N_i, which by definition satisfies

$$\frac{d}{dN_i}L(N_i)\Big|_{N_i=\hat{N}_i} = 0. \tag{9.19}$$

Since $\log(x)$ is a monotonically increasing function of x, we can apply the same optimality condition to $\log L(N_i)$ and optimize with respect to N_i^{-1} giving

$$\begin{aligned}
\frac{d}{dN_i^{-1}}\log L(N_i)\Big|_{N_i=\hat{N}_i} &= 0 \\
&= \frac{d}{dN_i}\left(-(t_i - t_{i-1})\binom{k_{i-1}}{2}\frac{1}{N_i} + \log(\frac{1}{N_i})\right)\Big|_{N_i=\hat{N}_i} \\
&= -(t_i - t_{i-1})\binom{k_{i-1}}{2} + N_i\Big|_{N_i=\hat{N}_i} \\
&= -(t_i - t_{i-1})\binom{k_{i-1}}{2} + \hat{N}_i
\end{aligned}$$

Thus, in each interval we have the following maximum likelihood estimate:

$$\hat{N}_i = (t_i - t_{i-1})\binom{k_{i-1}}{2}. \tag{9.20}$$

While this is the simplest case, extensions to the classical Skyline plot method involve allowing the grouping of multiple intervals together [**Strimmer2001**]. There are also now many ways of incorporating uncertainty into the results [**DrummondEtAl2005**, **Heled2008**] in a Bayesian framework (next chapter).

9.2.4.2 Coalescent approximation of birth-death models

As mentioned earlier in the chapter, coalescent theory is not intrinsically tied to the Wright-Fisher model of population dynamics. Indeed a much broader class of population genetic models possess limits in which the probability of a sampled tree is given by the coalescent process. One of the most important features of population models which possess coalescent limits is the *exchangeability* of individuals within the population, a requirement which forms the basis for the very general Cannings model [**Cannings**] of which the Moran and Wright-Fisher models are special cases. Given that this feature is also shared by the birth-death models described earlier, it is unsurprising that we can approximate such models using a coalescent distribution [**Volz2009**, **Volz2012**, **Volz2014**].

To see how such an approximation can work, consider a typical birth-death trajectory such as the one shown in figure **??**. As discussed in section 9.1.1, the birth events (represented by vertical lines in the figure) occur at a rate which, under the model, is λI, with I being the population size. While this exact rate is itself a random variable (since it depends on the outcome of the birth-death process before a given time), we can approximate its value using the deterministic limit of the birth-death process (also discussed in section 9.1.1 in which the population size at a given time is assumed to equal the expected size under the stochastic model. Doing this, we define the birth rate $B(t) = \lambda I(t) = \lambda I_0 \exp\left[-(t - t_0)(\lambda - \mu)\right]$ where I_0 is the population size at the present. (Note that, in contrast to the formulation in section 9.1.1, the time variable t here increases into the past, for consistency our discussion of coalescent theory; hence the sign change in the exponent.) This approximation is only adequate for times such that $I(t)$ is very large.

Now, consider the pair of sampled lineages extending from the right-hand side. These lineages must coalesce at a point in time corresponding to a birth event. As these ancestral lineages propagating backward in time, every birth event they encounter represents a possible coalescence time. What is the probability that a pair of lineages coalesce at a particular birth event? To answer this, consider that all members of the population are equivalent under our model. The probability that a particular pair of lineages coalesces at a given birth event is thus simply the inverse of the total number of such pairs in the population at time immediately following the birth event: $p_2(t) = 1/\binom{I(t)}{2}$. For k lineages, the probability of a coalescence occurring is simply the ratio of the number of pairs that would result in a coalescence on the

tree to the total number of pairs in the population, i.e. $p_k(t) = \binom{k}{2} / \binom{I(t)}{2}$.

By combining the per-birth-event coalescent probability $\chi_k(t)$ with the population birth rate $B(t)$ we recover an approximation for the coalescent rate at a given time:

$$\begin{aligned}\chi_k(t) = B(t)p_k(t) &= \lambda I(t)\binom{k}{2} / \binom{I(t)}{2} \\ &= \binom{k}{2}\frac{2\lambda}{I(t)-1} \\ &\simeq \binom{k}{2}\frac{2\lambda}{I(t)}. \end{aligned} \tag{9.21}$$

(We drop the -1 from the denominator on the final line since this is unimportant when $I(t)$ is large.)

This coalescent rate can be used to compute an approximate probability for a tree for a given set of birth-death parameters. This approximation is valid only when the population size remains large over the entire time-span of the tree.

It is interesting to note that this has an identical form to the coalescent rate between lineages of the WF model with a deterministically varying population size $N(t)$ we discussed in section 9.2.4. The only difference is that in the WF case the rate is proportional to $1/g$ (the inverse of the time between generations) while in the approximate BD case the rate is proportional to 2λ. To understand this better, it helps to note that it is possible to derive the WF coalescent rate under the WF model using the same considerations we applied in this section. In the WF case however, the whole-population birth rate which must be used is $N(t)/2g$, where the factor of $1/2$ appears because a parent in the WF model on average produces only 1 child (i.e. one half of a coalescing pair). This accounts for the factor of two difference in the approximate coalescent rates between the BD and WF models.

9.2.5 Application: Hepatitis C epidemic in Egypt

Hepatitis C

The hepatitis C virus (HCV) was first identified in 1989. Its genome is a single-stranded, 9.6 kilobases long RNA molecule. The virus has not been well characterized yet, since a system to culture it in tissue has been developed only recently.

More than 185 million people worldwide are infected with HCV, out of which about 80% have a chronic infection. Acute infections can resolve spontaneously, but even when they do, recovery confers no immunity against further re-infections. Chronic HCV infections damage the liver, causing liver cirrhosis and increasing the risk of some types of cancer that affect e.g. the liver and pancreas.

HCV is mostly transmitted through exposure to infected blood, although other modes such as sexual transmission and vertical (mother to child) transmission are also possible. Blood transfusions and injections with infected needles account for

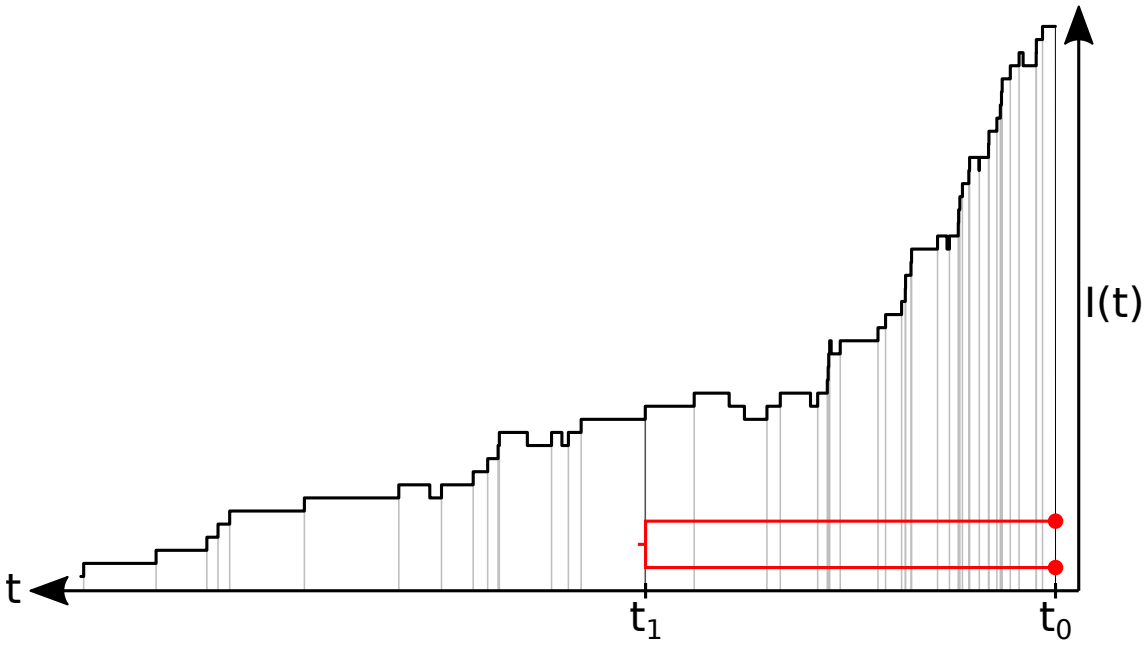


Figure 9.16: Relationship between a birth-death process trajectory and the coalescence time of a pair of sampled lineages. Each birth event time, represented by vertical grey lines, represents a possible coalescence time. This observation leads to the simple approximation of birth-death processes described in the text.

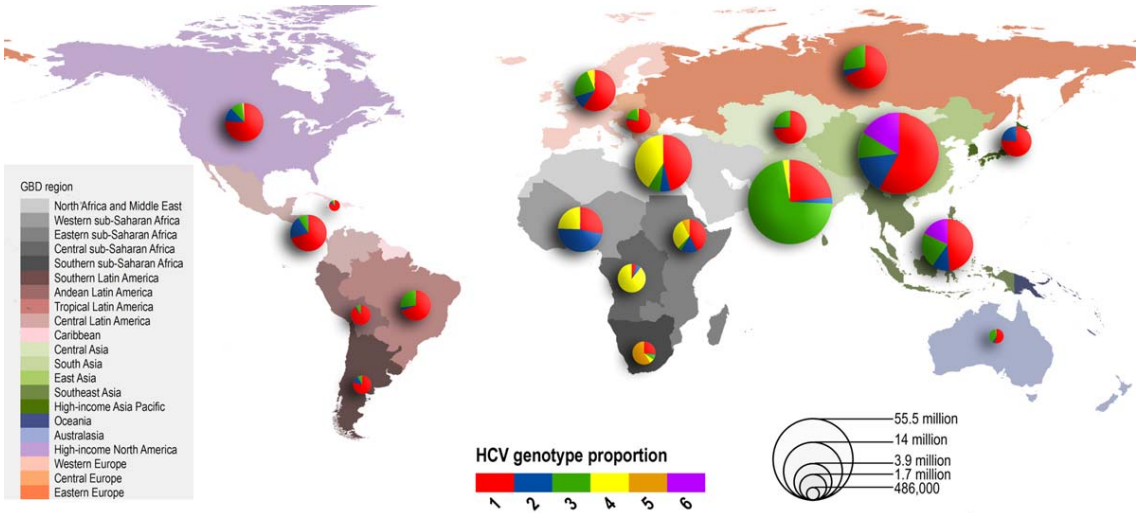


Figure 9.17: Prevalence of Hepatitis C infections in the world.

most new infections.

Hepatitis C in Egypt

The prevalence of Hepatitis C in Egypt is 10-20%, which is the highest HCV prevalence in the world. Surprisingly, the neighbouring countries have much lower HCV prevalences. Explaining this discrepancy has been an important domain of study and one of the first applications of coalescent models to phylodynamics.

The analysis presented here used a dataset of 61 HCV sequences from Egypt [**DrummondEtAl2005**]. The sequences are 411 base pairs long read-outs of the HCV gene E1. All of them were sampled at the same time (contemporaneous sampling). A phylogenetic tree of those sequences was built (see Figure 9.18) and the coalescent model was used to obtain an estimate of the infected population size through time, shown in Figure 9.19.

The results show a 100-fold increase in the epidemic spread during the first half of the 20^{th} century. This result was confirmed by another analysis performed 8 years later, using the birth-death model on the same dataset, as shown in Figure 9.20 [**StadlerEtAl2013PNAS**].

The reproductive number estimate also presents a peak at the same time period, consistent with the increase in infected population size. The agreement of the two analyses shows a strong support of the observed pattern by the data. However, this raises a new question: what caused this increase in the spread of HCV in Egypt in the first half of the 20^{th} century?

Schistosomiasis explanation

Schistosomiasis is an infection caused by a parasitic worm, the schistosoma. Until 1970s, patients in Egypt were treated for this condition with drugs injected intra-

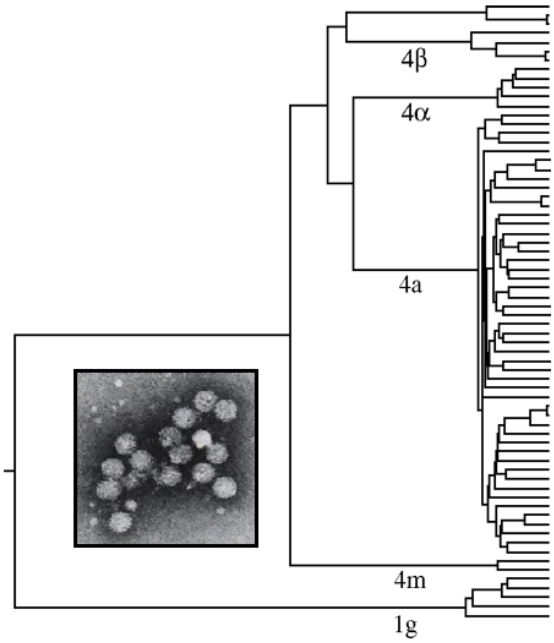


Figure 9.18: Phylogenetic tree reconstructed from the Egyptian HCV dataset.

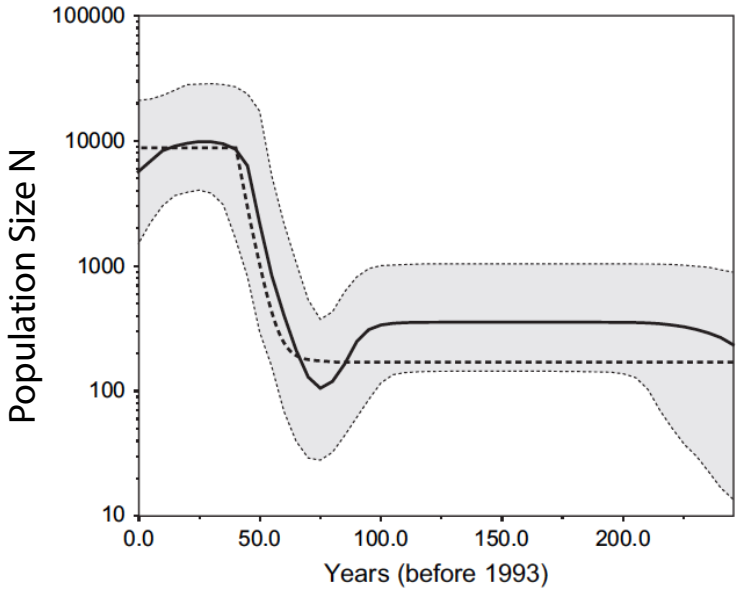


Figure 9.19: Estimate of the infected population size through time obtained from the Egyptian HCV dataset. The black line is the median estimate, the credible interval is shown in grey. Figure adapted from [**DrummondEtAl2005**].

venously. Injection therapy has since been replaced by oral therapy. Hospitals were not particularly careful about repeated needle use in the first half of the 20$^{\text{th}}$ century, so needles used for schistosomal therapy could have been contaminated. Thus,

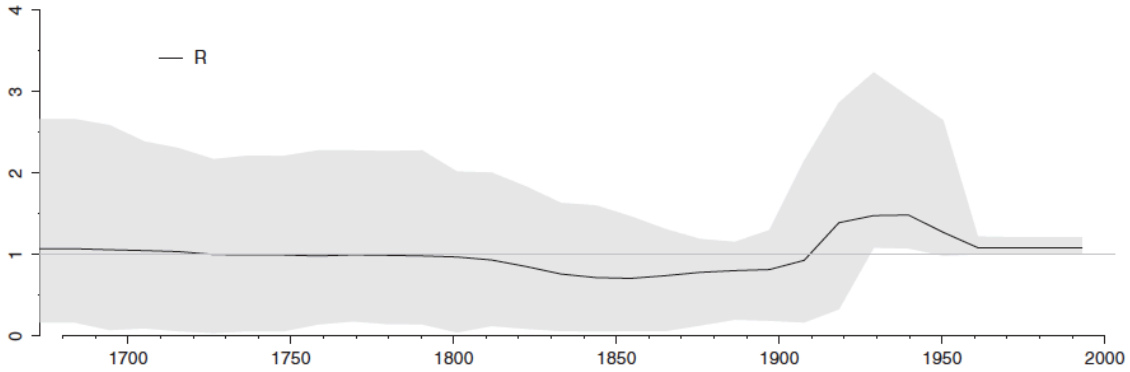


Figure 9.20: Estimate of the reproductive number through time obtained from the Egyptian HCV dataset. The black line is the median estimate, the credible interval is shown in grey. Figure adapted from [**StadlerEtAl2013PNAS**].

this treatment protocol might have contributed to the rapid spread of Hepatitis C.

The results shown previously are consistent with this hypothesis, as the use of antischistosomal injections coincides with the estimated time period of increased HCV spread. In particular, the estimated reproductive number of HCV decreases back to 1 just when the oral therapy for Schistosomiasis was introduced. Antischistosomal injections are the currently accepted explanation for the high HCV prevalence in Egypt.

9.3 Comparison of Coalescent models and Birth-Death models

Both the coalescent and the birth-death models describe $P[\mathcal{T}|\eta]$, the distribution of trees $\mathcal{T}$ given the parameters of the population model η. Coalescent models are parameterized with the population size through time, i.e. $\eta = N(t)$, whereas birth-death models are parameterized with the birth and death rates through time, i.e. $\eta = (\beta(t), \delta(t))$.

Additionally, as they are usually applied, there is a differences in the way the models deal with samples. Birth-death models generally provide an explicit model for the sampling process, while coalescent models generally condition on the number and times of samples as a given. This means that, while analyses based on coalescent methods may not be led astray by a misspecified sampling model, birth-death models can use information from the number and times at which sequences have been collected to learn about the population-level process.

Finally, coalescent models generally assume a deterministically varying population, while birth-death phylodynamic models never make this assumption. Since the relative importance of population size fluctuations can be extremely important when

the population size is small (even temporarily), this means that one should take care when applying coalescent models when the ancestral population is likely to have been small at some point along the tree. This is, for example, usually the case in populations of infected hosts belonging to epidemics.

For more information about how these models compare to each other on exponentially growing populations please refer to **BoskovaEtAl2014**.

9.4 Accounting for population structure

So far we have assumed that our populations of interest are homogeneous. In reality most populations are structured: for instance, different individuals will have different risks of catching a particular disease based on factors such as their location, their age, their social group, etc. This structure can directly affect the shape of the resulting phylogeny. For instance, consider a toy example in which samples are collected from three distinct islands, between which limited migration occurs. From the reconstructed phylogeny for these samples shown in figure 9.21, one can see that lineages ancestral to samples from the same island coalesce rapidly, while lineages ancestral to samples from distinct islands take a lot longer. While the degree to which the structure influences the phylogeny depends heavily on the details of the population (in the toy example the migration rate plays an important role), it is clear that population structure *can* affect the shape of the phylogeny to the extent that it can be possible to discern this structure from the shape of the phylogeny alone. One way of thinking about structure is that

This section will delve into the various extensions to phylodynamic models which can be used to account for underlying population structure.

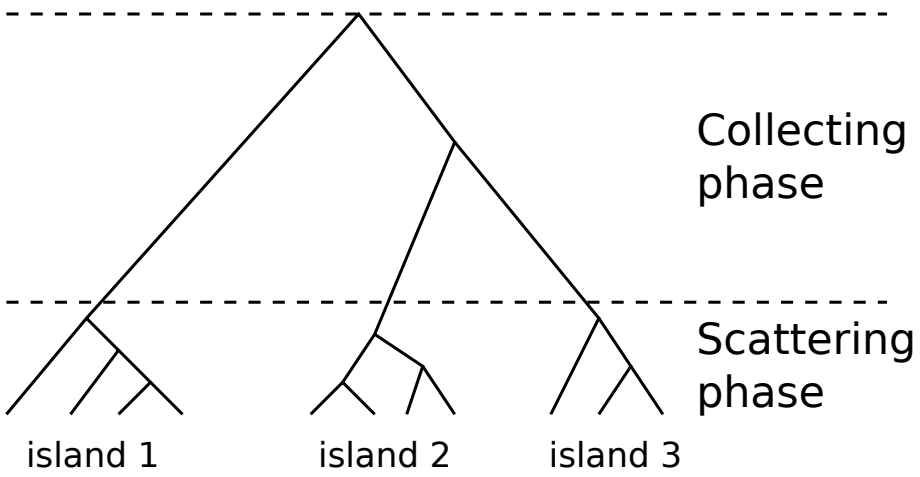


Figure 9.21: Population structure can affect the shape of the phylogeny. In this example, inspired by [**Pannel2003**], the ancestry of samples from three distinct island populations is shown. Weak migration between islands means that lineages coalesce quickly with lineages ancestral to samples from the same island ("scattering phase"), but slowly with lineages ancestral to other islands ("collecting phase").

9.4.1 Motivations for structured phylodynamics

There are two important motivations for incorporating structure into our phylodynamic models Firstly, structure which is not accounted for in the model can lead to biases in the results gleaned from phylodynamic analysis. For example, an basic coalescent analysis of the phylogeny shown in figure 9.21 might conclude that the

difference in coalescence rates is a result of a recent reduction in effective population size, while a birth-death skyline analysis might conclude that the birth rate of the population has recently increased. Incorporating structure into the phylodynamic models allows us to avoid this important source of bias.

The second motivation is that incorporating structure allows us to use phylodynamic analyses to directly address questions relating to population structure. For instance, what is the migration rate between islands? What are their respective sub-population sizes? In the epidemiological context: do infection rates depend on sub-population? When did a disease first enter a geographic region? Importantly for epidemiology, many of these population structure questions cannot be addressed by non-genetic time series data at all.

Taking the example of a pathogen with two strains, a drug-resistant strain and a drug-sensitive strain, there are two possible scenarios for the propagation of the drug-resistant strain: 1) transmitted drug resistance, where the drug resistant strain is directly transmitted from patient to patient, and 2) *de novo* drug resistance, where the drug resistant strain is never transmitted but repeatedly arises through mutation in an already infected patient. These two scenarios and their impact on the resulting phylogenetic tree are shown in Figure 9.22.

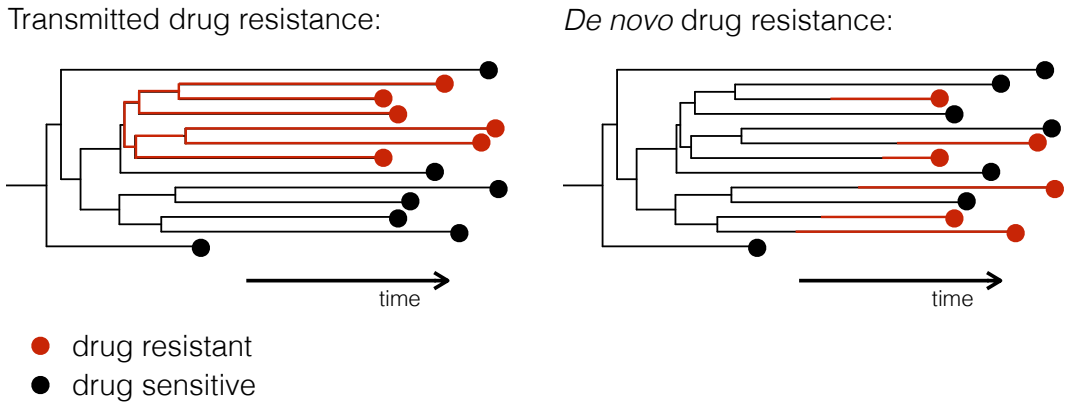


Figure 9.22: Transmission trees simulated under two different scenarios, transmitted drug resistance (left) and *de novo* drug resistance (right).

In practice, the history of the lineages is not known, only their status (drug-resistant or drug-sensitive) at the time of sampling. Reconstructing trees from this information only leads to phylogenies like the ones shown in Figure 9.23. The histories are missing in the reconstructed phylogenies. However, the phylogeny still contains information about the underlying scenario: transmitted drug resistance is more likely if drug-resistant tips are mostly clustered together, whereas *de novo* resistance leads to drug-resistant and drug-sensitive tips interspersed with one another.

Another way of thinking about the benefits of structured models is that they allow us to account, in some limited way, for non-neutral evolution. Although one usually thinks of selection as sequence-dependent fitness effects, structured models

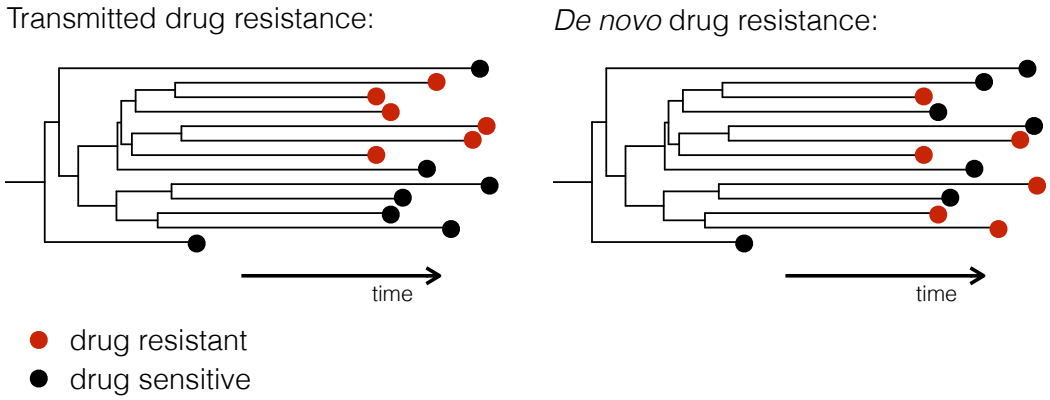


Figure 9.23: Transmission trees reconstructed from the simulated trees in figure 9.22. Notice that the only information we have besides the sequence of the pathogen is the tip label, marking the drug resistant or drug sensitive strain. The history (colors on the branches) is missing in the reconstructed phylogenies.

account for the fitness effects of any heritable trait, of which geographic location, epidemiological compartment membership, and phenotypic traits are all examples.

9.4.2 Structured birth-death phylodynamic models

The multi-type birth-death model is an extension of the birth-death model built to handle structured populations.

It contains two or more compartments, or types. In epidemiology, different compartments can represent different pathogen strains, different geographic locations, different host risk groups, or any other kind of pathogen or host population structure. Each compartment has its own birth rate β_{ii}[3], death rate δ_i and sampling rate ϕ_i. Migration rates γ_{ij} describe the rate of individuals moving from one compartment i to another j. The model with two compartments is shown in Figure 9.24.

In our previous example of drug-resistance, the two compartments represent the population infected with the drug-sensitive strain I_1 and the population infected with the drug-resistant strain I_2. β_{11} and β_{22} are the transmission rate respectively of the drug-sensitive strain and the drug-resistant strain, and γ_{12} is the rate of resistance evolution.

9.4.2.1 The multi-type birth-death likelihood

To derive the phylodynamic likelihood of this model we use similar considerations to those employed in section 9.1.1. As in that section, we again assume complete sampling in the present ($t = T$) and that no samples are collected before that time.

[3]The double index is due to the fact that in other definitions of the model there can be birth rates into other compartments, e.g. β_{ij}, $i \neq j$.

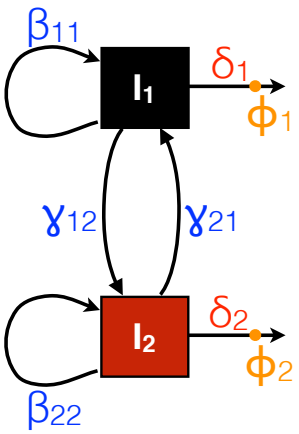


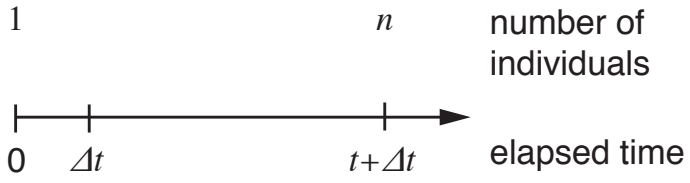
Figure 9.24: Schematic representation of a multi-type birth-death model with two compartments and all parameters associated with it.

Before delving into the details, we must point out that what can be meant by "tree" is slightly more complicated when dealing with structured models. On the one hand, we could be referring simply to a regular phylogenetic time tree, with sequences and potentially type information associated with leaves, as in figure 9.23. In the context of structured phylodynamic likelihoods we will refer to these as *sample-typed trees*. On the other hand, we could be referring to time trees in which the edges at every point are annotated with the ancestral types, as in figure 9.22. These trees are known as either *multi-type trees* or *edge-typed trees*.

Here we will present a means of computing the probability of a sample-typed tree under the multi-type birth-death model.

To do this, we firstly consider the probability $p_i(0|t)$ that the no descendants of an individual of type i survive after time t. This is equivalent to $p(0|t)$ from the unstructured case, with the sole difference being that here we also condition on the *type* of the original individual.

To compute this probability we make use of the same arguments found in section 9.1.1.1. That is, we write the probability $p_i(0|t+\Delta t)$ in terms of $p_i(0|t)$ and use this relationship to construct a master equation. We partition time in the same way as in that section:



During the time Δt after the start of the process, any of the following can happen:

1. No event occurs, with probability $1-(\delta_i+\sum_j \beta_{ij}+\gamma ij)\Delta t$. In this case, the

descendants of the original individual must go extinct within the remaining time t.

2. The individual dies with probability $\delta_i \Delta t$. If this occurs, the descendants of the individual will go extinct with probability 1.

3. The individual gives birth to another individual of type j with probability β_{ij}. In this case the probability of the descendants going extinct is the probability that the trees below individuals of types i and type j both go extinct in the remaining time t. Note that there are several possibilities here, due to the different possibilities for j.

4. The individual changes type (migrates) from i to j. in this case the probability of the descendants going extinct is the probability that the tree below the individual of type j goes extinct in the remaining time t. As in the birth case, there are several possibilities here due to the different possible destination types j.

5. More than one of these events could occur with a probability proportional to Δt^2.

Combining these possibilities allows us to write

$$\begin{aligned} p_i(0|t+\Delta t) = &\left(1 - (\delta_i + \sum_j (\beta_{ij} + \gamma_{ij}))\Delta t\right) p_i(0|t) \\ &+ \delta_i \Delta t \times 1 \\ &+ \sum_j \left(\beta_{ij} p_i(0|t) + \gamma_{ij}\right) \Delta t p_j(0|t) \\ &+ \mathcal{O}(\Delta t^2) \end{aligned} \tag{9.22}$$

Rearranging the terms and taking the limit $\Delta t \to 0$ yields the following master equation:

$$\begin{aligned} \frac{d}{dt} p_i(0|t) = &- \left(\delta_i + \sum_j (\beta_{ij} + \gamma_{ij})\right) p_i(0|t) \\ &+ \delta_i + \sum_j \left(\beta_{ij} p_i(0|t) + \gamma_{ij}\right) p_j(0|t) \end{aligned} \tag{9.23}$$

We can use a similar set of arguments to compute the probability $p_i(1|t)$ that exactly one descendant of an individual is alive after a time t. By considering the possibilities

in the interval Δt following the start of the process, we find

$$\begin{aligned} p_i(1|t+\Delta t) = &\left(1 - (\delta_i + \sum_j (\beta_{ij} + \gamma_{ij}))\Delta t\right) p_i(1|t) \\ &+ \delta_i \Delta t \times 0 \\ &+ \sum_j \beta_{ij} \Delta t \left((p_i(0|t)p_j(1|t) + p_i(1|t)p_i(0|t)\right) \\ + \sum_j \gamma_{ij} \Delta t p_j(0|t) + \mathcal{O}(\Delta t^2) & \end{aligned} \tag{9.24}$$

Again, rearranging the terms and taking the limit $\Delta t \to 0$ gives the master equation:

$$\begin{aligned} \frac{d}{dt} p_i(1|t) = &- \left(\delta_i + \sum_j (\beta_{ij} + \gamma_{ij})\right) p_i(1|t) \\ &+ \sum_j \left(\beta_{ij}(p_i(0|t)p_j(1|t) + p_i(1|t)p_j(0|t)\right) \\ &+ \sum_j \gamma_{ij} p_j(0|t) \end{aligned} \tag{9.25}$$

Unlike the master equations for $p(0|t)$ and $p(1|t)$ in the unstructured model, the master equations for $p_i(0|t)$ and $p_i(1|t)$ in the structured model do not have known analytical solutions. However, they can however be solved using standard numerical integration techniques.

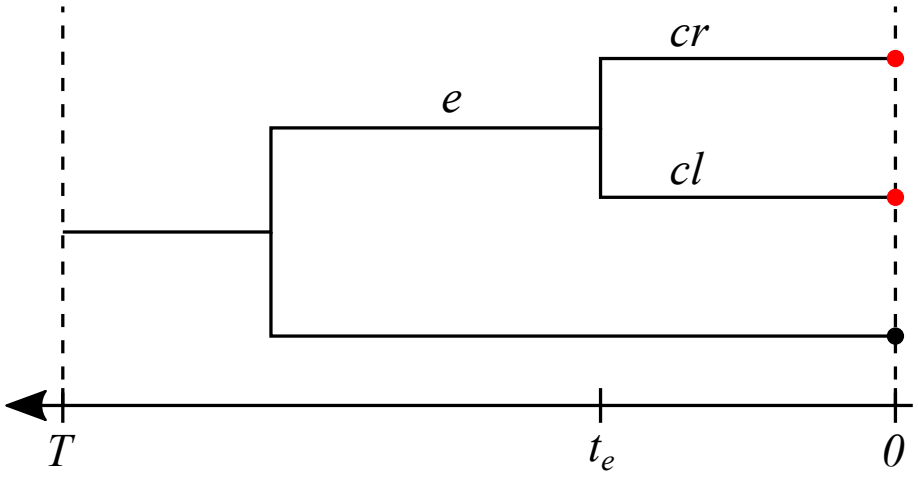


Figure 9.25: Tip-typed tree with an individual edge e marked. As this is an internal node, we denote the left and right child edges cl and cr respectively in the boundary condition for $g_i^e(t_e)$.

At this point, we consider a different probability: the probability $g_i^e(t)$ that an individual alive at time t on edge e of a tree produces the tree below it. (See the example

tree in figure 9.25.) Using the same reasoning as that used to derive the ODE for $p_i(1|t)$ above, we find that along a single edge, $g_i^e(t)$ evolves according to a master equation with exactly the same form as Eq. (9.25):

$$\begin{aligned}\frac{d}{dt}g_i^e(t) = &-\left(\delta_i + \sum_j(\beta_{ij}+\gamma_{ij})\right)g_i^e(t) \\ &+\sum_j\left(\beta_{ij}(g_i^e(t)p_j(0|t) + g_j^e(t)p_i(0|t)\right) \\ &+\sum_j \gamma_{ij}g_j^e(0|t)\end{aligned} \tag{9.26}$$

At the base of each edge e (time t_e) however, the solution is subject to the following boundary condition:

$$g_i^e(t_e) = \begin{cases} 1 & \text{if the node is a leaf} \\ \frac{1}{2}\sum_j \beta_{ij}\left(g_i^{cl}(t_e)g_j^{cr}(t_e) + g_j^{cl}(t_e)g_i^{cr}(t_e)\right) & \text{otherwise}\end{cases}$$

where we define cl and cr to be the left and right child edges in the instance that e is an internal node.

The dependence of the boundary condition for internal nodes on the solutions for the edges below them suggests that we numerically integrate Eq. (9.26) backward in time from each leaf, then successively combine these solutions until we reach the root or origin of the tree.

Doing this leaves us with $g_i^r(T)$, which is the probability of observing the tree and the leaf states given that the process began at time T before the present with an individual in state i. That is, $g_i^r(0) = P(\mathcal{T}^o, L|T, i, \beta, \delta)$. In order to convert this into the probability of the tree without conditioning on the starting state, we can introduce the initial state probabilities π_i. We can then write

$$P(\mathcal{T}^o, L|\beta, \gamma, \vec{\delta}) = \sum_i g_i^r(T)\pi_i \tag{9.27}$$

where β and $\vec{\gamma}$ are the birth and migration rate matrices, and $\vec{\delta}$ is the vector of type-specific death rates.

This is the phylodynamic likelihood under the multi-type birth-death model with complete sampling in the present and no sampling through time. It is straightforward to extend this same computation approach to handle sampling through time and incomplete sampling in the present.

It is also possible to use the same general strategy to compute the probability of an edge-typed tree, such as the one shown in figure 9.22. (The only major difference in that case is that type-change (migration) events become additional nodes in the tree, and the evolution of the probability distribution $g_i^e(t)$ is conditioned on no type changes occurring along edge e.)

Example: geographic spread of seasonal flu

The multi-type birth-death phylodynamic likelihood is the basis for performing phylodynamic inference on structured populations.

For example, to gain insight into the spread of influenza virus around the globe, flu sequences were recently analyzed [**KuehnertEtAl2016**] using a multi-type birth-death model and annotated according to the geographic location of the patient: northern hemisphere, southern hemisphere, or tropical area. The resulting phylogenetic tree, labeled with the inferred location of each lineage (seen as different colours on the branches), is shown in Figure 9.26.

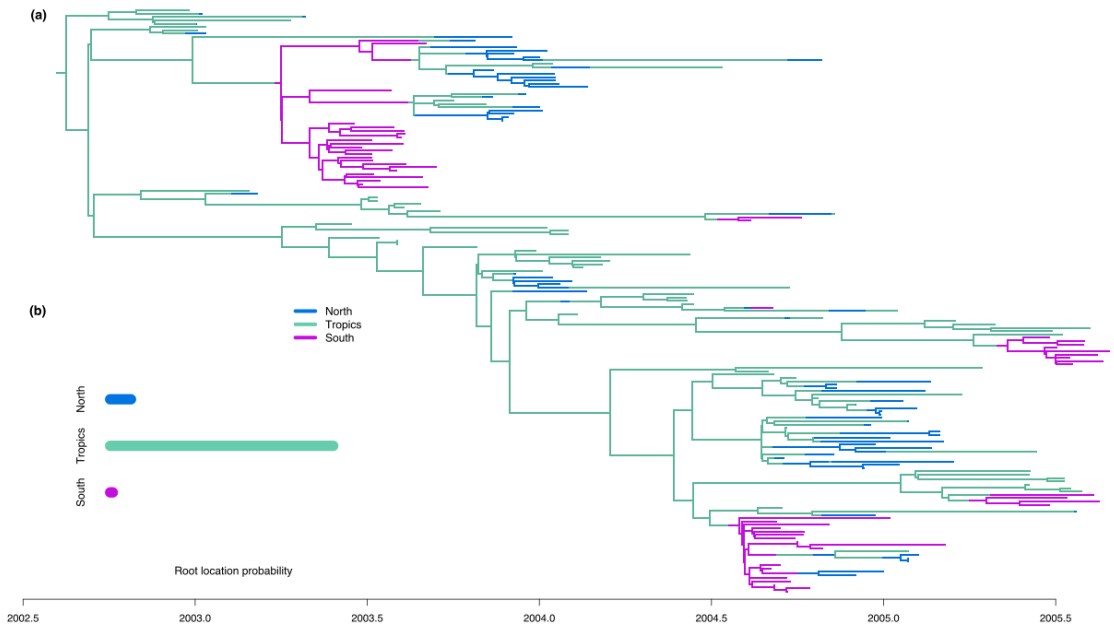


Figure 9.26: Phylogenetic tree of the geographic spread of seasonal influenza, inferred using BEAST2. The subfigure b) (bottom left) shows the posterior distribution for the root location, i.e. the estimated probabilities that the epidemic started in the north, south or tropical area. Figure adapted from [**KuehnertEtAl2016**].

The seasonality of influenza in the northern and southern hemispheres can be seen in this tree, as the tips from either hemisphere cluster together in localized epidemics. The backbone of the tree is composed almost exclusively of tropical lineages, which means that the tropical area is the reservoir for the flu virus, and tropical strains start the seasonal epidemics in other locations.

Reproductive numbers from the three different locations were also estimated in the same analysis. The results are shown in Figure 9.27. Similar to the tree, the reproductive numbers for both northern and southern hemisphere show marked seasonality, whereas the reproductive number in the tropical area is stable, confirming that influenza is endemic in this area.

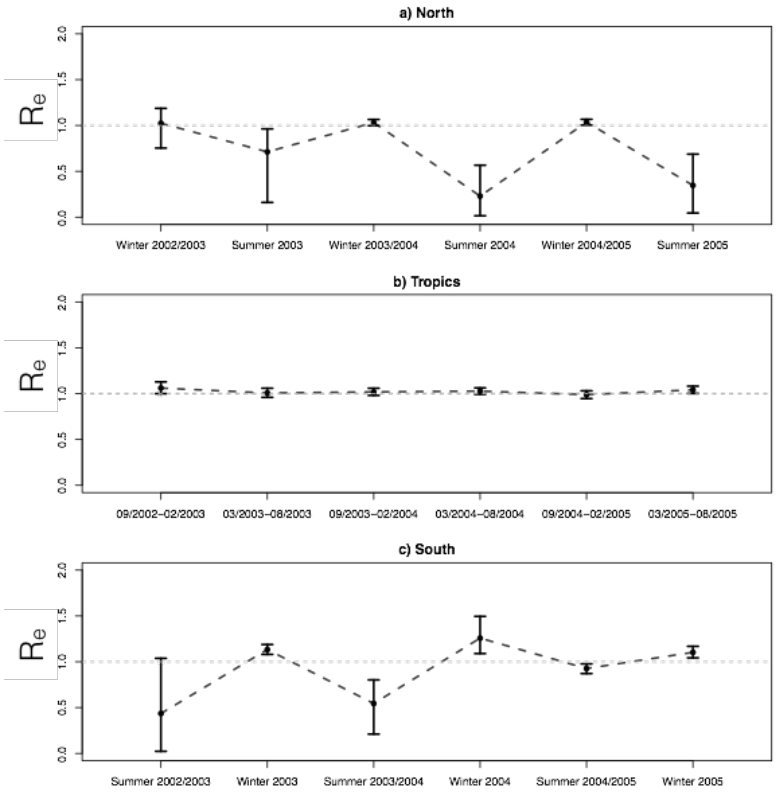


Figure 9.27: The effective reproductive number of seasonal influenza was inferred for a) northern hemisphere, b) tropical areas and c) southern hemisphere, using BEAST2 software. Figure adapted from [**KuehnertEtAl2016**].

9.4.3 Structured coalescent phylodynamic models

Like the coalescent distribution discussed in section 9.2, the structured coalescent provides a probability distribution over sampled time trees conditional on a particular demographic history. Just as in the unstructured case, the structured coalescent arises as a limiting case of a number of distinct population models, one of which is a structured extension to the Wright-Fisher (WF) model. A representation of this model is shown in Figure 9.28.

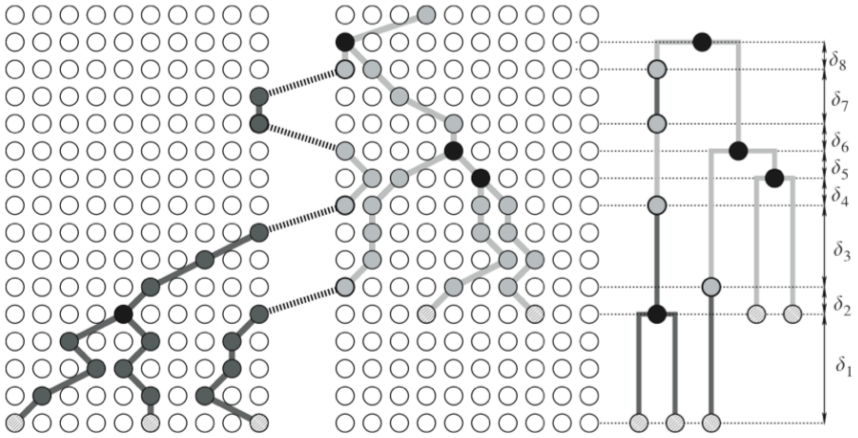


Figure 9.28: Schematic representation of a structured coalescent model with two compartments. Migration events, shown as dashed lines between subpopulations, are explicitly placed in the genealogy as bold circles (right). The δ_i label intervals between migration nodes, coalescent nodes, and leaf nodes.

9.5 Overview of phylodynamic applications

In what follows we give an overview of how evolution, phylogenetic analysis and phylodynamic analysis translate across several fields of interest.

- **Macroevolution:** lineages represent species.

  Evolution = molecular evolution, i.e. changes in genetic information and morphology through time;

  Phylogenetics = display species' ancestral relationships;

  Phylodynamics = speciation and extinction process.

  Examples: dinosaurs, penguins.

- **Epidemiology:** lineages represent infected hosts.

  Evolution = molecular evolution, i.e. changes in genetic information of pathogens through time;

  Phylogenetics = display the transmission history of the pathogen between patients;

  Phylodynamics = transmission and recovery process of patients infected by the pathogen;

  Examples: Ebola, HCV, HIV, Zika.

- **Immunology:** lineages represent B cells (cells producing antibodies).

  Evolution = molecular evolution, i.e. changes due to recombination and somatic hypermutation in response to pathogen exposure;

  Phylogenetics = display B cell evolution;

  Phylodynamics = B cell generation and loss process;

  Example: evolution of B cells in an HIV infected individual, see Figure 9.29.

- **Developmental biology:** lineages represent the cells of a multicellular organism.

  Evolution = cell differentiation from stem cells to highly specialized cells;

  Phylogenetics = display differentiation through time;

  Phylodynamics = gain and loss of cell types;

  Example: cell differentiation under different conditions, see Figure 9.30.

- **Human migration:** lineages represent human populations.

  Evolution = molecular evolution of bacteria co-evolving with humans[4];

[4] Analyzing human migration using human genome sequences would be hard since human genomes evolve slowly and recombine frequently.

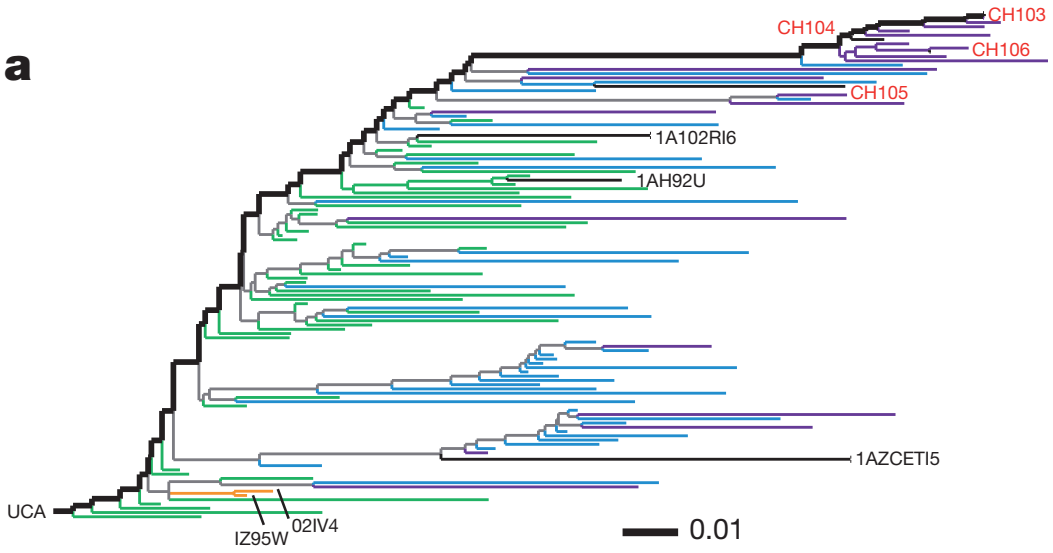


Figure 9.29: Phylogeny of B-cells from an HIV infected individual. A potential application of this work is investigating co-evolution between HIV and the B-cell response. Figure adapted from [**Liao2013**].

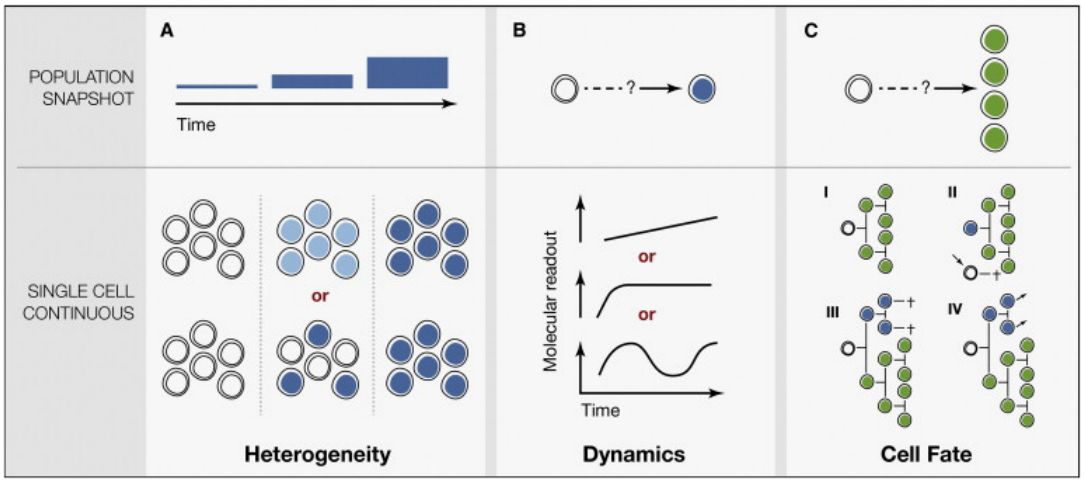


Figure 9.30: Illustration of the differentiation process and example phylogenies for that process. Figure adapted from [**EtzrodtEtAl2014**].

Phylogenetics = display movement of bacteria around the world and thus human migration;

Phylodynamics = migration process out of Africa;

Example: migration process reconstructed from *Helicobacter pylori* sequences, see Figure 9.31.

- **Language evolution:** lineages represent languages.

  Evolution = changes in letters and words through time;

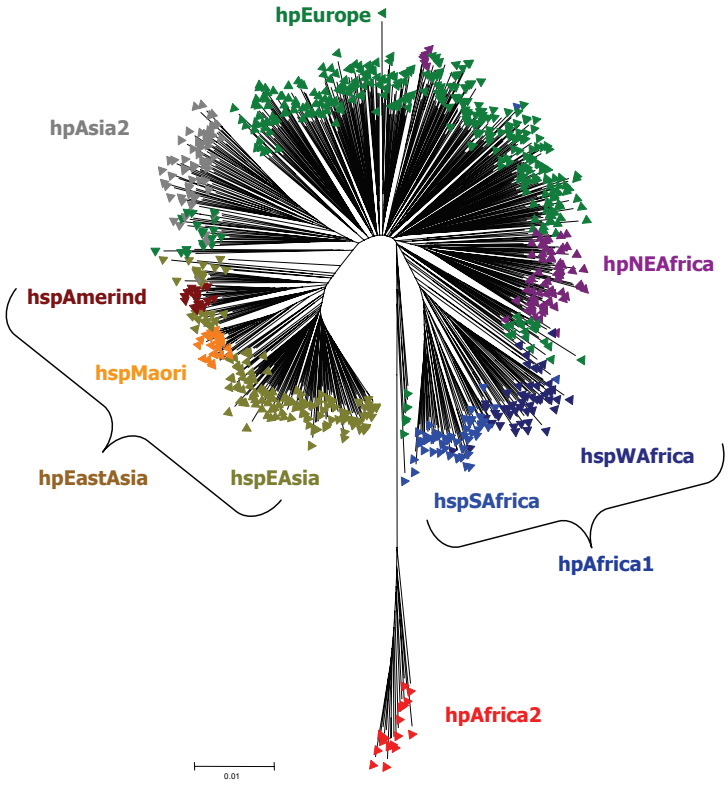


Figure 9.31: Neighbor joining tree of 769 *H.pylori* sequences labeled by area of sampling. Figure adapted from [**LinzEtAl2007**].

Phylogenetics = display language history;

Phylodynamics = gain and loss of languages;

Example: structure of language families, see Figure 9.32.

- **Cultural evolution:** only early steps have been taken in this field, potential areas of study include the evolution of religions [**BoteroEtAl2014**] and the evolution of political systems [**CurrieEtAl2010**].

9.6 Challenges

Despite their great success and widespread use, phylogenetic and phylodynamic analyses still face many challenges:

- NGS data: new generation sequencing provides an increasing amount of data. This allows us to use more detailed models and answer more complex questions, but it also poses computational challenges.
- Heterogeneity at the population level: different parts of the tree may be under different selective pressures, or show other types of heterogeneity. As seen in this chapter models able to handle population structure already exist, but

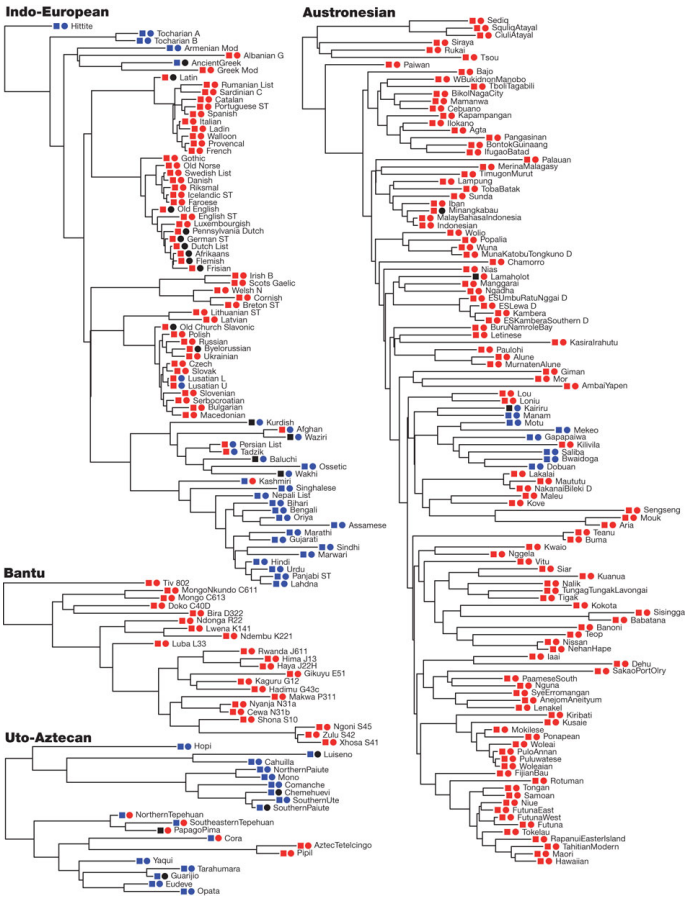


Figure 9.32: Phylogenies built from different language families. Figure adapted from [**DunnEtAl2011**].

these models are still computationally very expensive.

- Reticulate evolution: due to several phenomena such as hybridization and horizontal gene transfer, the evolutionary history can only be represented correctly by a network, rather than a tree. This presents a challenge, as most current models are only able to handle binary trees.

10 Bayesian inference

10.1 Bayesian theory

Bayesian inference is a framework that allows us to fit a model to the data while including existing knowledge. In the Bayesian framework the output of the inference is a distribution, rather than a point estimate, in contrast with the maximum likelihood framework. Bayesian probabilities express the uncertainty surrounding the possible values of the parameters rather than the frequency of a single best value. Existing knowledge is used in the form of prior probabilities assigned to the model parameters. After analyzing the data we calculate a posterior probability, which takes into account both the prior knowledge and the information provided by the data.

When applied to phylogenetics, the typical Bayesian approach allows us to jointly infer parameters of both the substitution model and the phylodynamic model together with a *rooted time tree*.

10.1.1 Bayes' formula

Bayesian inference centres around on Bayes' formula:

$$P[\sigma|D] = \frac{P[D|\sigma]P[\sigma]}{P[D]} \tag{10.1}$$

where D is the data, σ are the values of the parameters of the model, and $P[]$ can be a probability mass function or a probability density function, depending on whether D and σ are discrete or continuous, respectively.

The different components of Bayes' formula have specific names:

- $P[\sigma|D]$ is the posterior probability that we want to calculate;
- $P[D|\sigma]$ is the likelihood;
- $P[\sigma]$ is the prior probability;
- $P[D]$ is the marginal likelihood of the data.

The marginal likelihood of the data $P[D]$ is calculated by integrating out (marginalizing) the model parameters σ:

$$P[D] = \int_{\sigma} P[D|\sigma]P[\sigma]d\sigma.$$

The marginal likelihood is very often difficult to compute, especially when the dimension of σ is high (i.e. the model contains a lot of parameters) and the integral cannot be solved analytically.

Note that Bayes' formula is valid for any dimension of σ, and can be decomposed further along each dimension: for instance if σ is of dimension 2, i.e. $\sigma = (\sigma_1, \sigma_2)$, we get the following formula:

$$P[\sigma_1, \sigma_2|D] = \frac{P[D|\sigma_1, \sigma_2]P[\sigma_1, \sigma_2]}{\int_{\sigma_1}\int_{\sigma_2} P[D|\sigma_1, \sigma_2]P[\sigma_1, \sigma_2]d\sigma_1 d\sigma_2}.$$

Assuming that $P[\sigma_1|\sigma_2] = P[\sigma_1]$, the prior for the parameters can be factorized, giving

$$P[\sigma_1, \sigma_2|D] = \frac{P[D|\sigma_1, \sigma_2]P[\sigma_1]P[\sigma_2]}{\int_{\sigma_1}\int_{\sigma_2} P[D|\sigma_1, \sigma_2]P[\sigma_1]P[\sigma_2]d\sigma_1 d\sigma_2}.$$

10.1.2 Prior dependency

The choice of prior is very important in Bayesian inference as it will have an impact on the posterior. This impact will be greater if there is little information in the data or if the prior distribution excludes certain values. Due to the Bayes' formula, if a particular combination of parameter values is excluded by the prior, i.e. if its prior probability $P[\sigma] = 0$, then it will also be excluded by the posterior, i.e. its posterior probability $P[\sigma|D] = 0$. This is true regardless of any information in the data. As a consequence, it is better to pick priors that only exclude values that are biologically impossible (for instance, negative substitution rates can be excluded safely), rather than accidentally excluding any plausible values. Bayesian inference will converge to the same posterior distribution with increasing amounts of data regardless of the prior distribution, unless some likely values are excluded by the prior.

Example: pairwise alignment inference

In chapter 5 we calculated the likelihood function of the genetic distance d between two sequences under the JC69 model of molecular evolution, knowing only the sequence length and the number of substitutions between the sequences. If we have the exact sequence alignment, i.e. we know which specific substitution happened at which position, the likelihood function for the JC69 model takes a slightly different form:

$$P[D|d] = \left[\frac{1}{4} + \frac{3}{4}\exp\left(-\frac{4}{3}d\right)\right]^{L-S} \times \left[\frac{1}{4} - \frac{1}{4}\exp\left(-\frac{4}{3}d\right)\right]^{S} \tag{10.2}$$

where D is the pairwise alignment, L the length of the alignment and S the number of substitutions in the alignment.

In figure 10.1a we plot the posterior distributions obtained using the Bayes formula eq. 10.1 with the likelihood of eq. 10.2 for an alignment of length $L = 10$ and two

different prior distributions for d: a very narrow exponential prior which is heavily skewed towards values close to 0 and a flat prior which equally supports a range of values between 0 and 3. In this case the choice of prior impacts both the location of the peak of the posterior distribution and the range of values which have a strong posterior probability. Figure 10.1b shows the same plot for an alignment of length $L = 100$. In this case the two posterior distributions are almost identical.

It is possible to use the posterior probability distribution as a prior to analyze a completely new dataset. Note however that it is *wrong* to analyze data using the posterior calculated from some data as a prior for a new analysis on the same data.

10.1.3 Application to phylogenetics

To apply Bayesian inference to phylogenetics we will use the following definitions:

- Q is the substitution rate matrix constituting the parameters of the evolutionary model (e.g. substitution rate(s), stationary nucleotide frequencies, etc.);
- η represents the parameters of the tree generating model (e.g. birth and death rate(s) for birth-death models, or effective population size(s) for a coalescent model);
- τ represents the rooted time tree which gave rise to the data;
- A represents the sequence alignment (our data).

Bayesian inference aims to characterize the overall posterior distribution, which can be expressed as

$$\begin{aligned} P[\tau, Q, \eta|A] &= \frac{P[A|\tau, Q, \eta]P[\tau|\eta]P[\eta]P[Q]}{P[A]} \\ &= \frac{P[A|\tau, Q]P[\tau|\eta]P[\eta]P[Q]}{P[A]} \end{aligned}$$

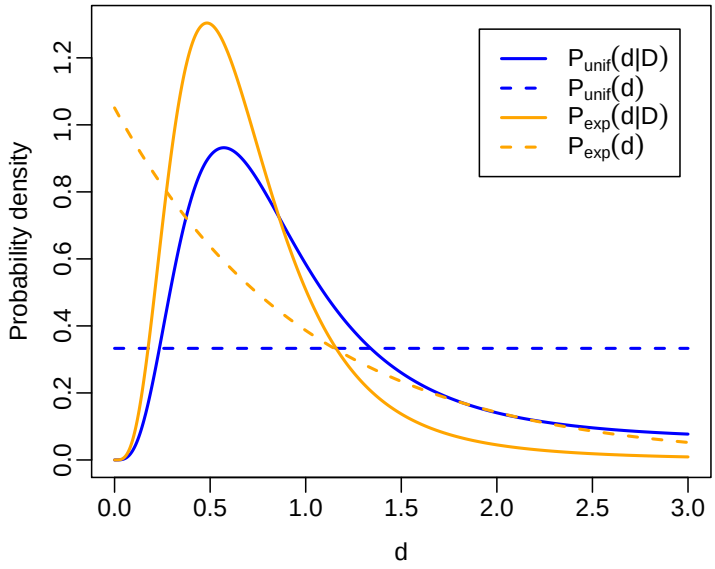
with the marginal likelihood

$$P[A] = \int_{\tau,Q,\eta} P[A|\tau, Q]P[\tau|\eta]P[\eta]P[Q]d\tau dQ d\eta.$$

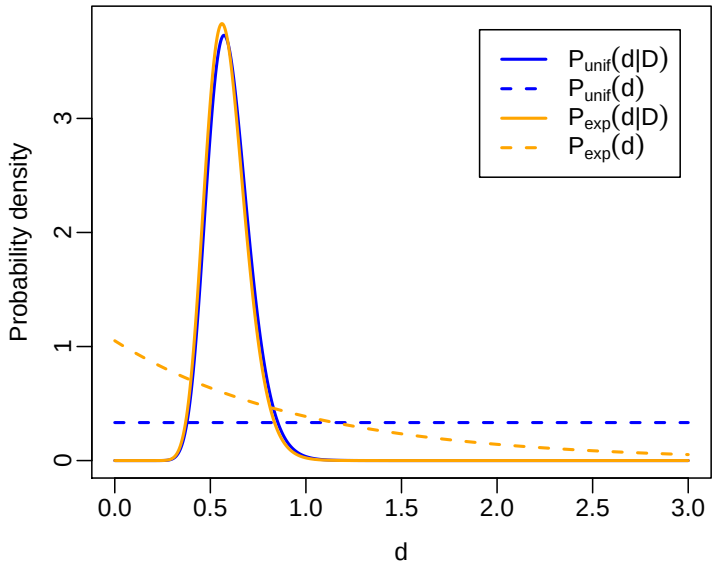
$P[A|\tau, Q]$ is the phylogenetic likelihood. $P[\tau|\eta]$ is the prior on the tree topology τ, sometimes also called the phylodynamic likelihood. $P[\eta]$ and $P[Q]$ are the prior distributions for the parameters of the evolutionary and the tree generating model, respectively.

10.2 Markov chain Monte-Carlo algorithm

The marginal likelihood is usually impossible to calculate directly. This is particularly true for phylogenetic applications which often have many parameters. Thus,



(a) $L = 10$, $S = 4$



(b) $L = 100$, $S = 40$

Figure 10.1: Posterior distributions of genetic distance d obtained with an alignment of length (a) $L = 10$ ($S = 4$) and (b) $L = 100$ ($S = 40$), for a uniform prior (in blue) or and an exponential prior (in orange).

Bayesian inference commonly uses a family of algorithms called the Markov chain Monte-Carlo (MCMC) algorithms, which only use ratios of posterior probabilities and can thus avoid calculating the marginal likelihood entirely. Several classes of MCMC algorithms exist. In this section, we will focus on the class of *random walk MCMC* algorithms, and on the particular implementation called the Metropolis-Hastings algorithm. This is the algorithm implemented in most phylogenetic inference software.

10.2.1 Random walk algorithm

The basic principle of MCMC algorithms is to construct a random walk through the parameter space. Each position in the parameter space corresponds to a specific set of parameters values. This random walk is a series of steps through this space that fulfil the properties of a Markov chain, such that each step is independent of the previous steps and only depends on the current position. To get a representation of the posterior distribution from the algorithm, we need to configure our random walk in a way that most steps will be taken in areas of the parameter space where the posterior probability is the highest. Steps will be proposed at random and accepted according to the following rules:

- if a step results in an increase in posterior probability, it will always be accepted;
- if a step results in a decrease in posterior probability, it will be accepted with a probability inversely proportional to the decrease in probability (i.e. the bigger the decrease the less likely it is that the step will be accepted).

Figure 10.2 shows a representation of this process as a robot exploring a one-dimensional landscape looking for the highest hills (areas of highest probability).

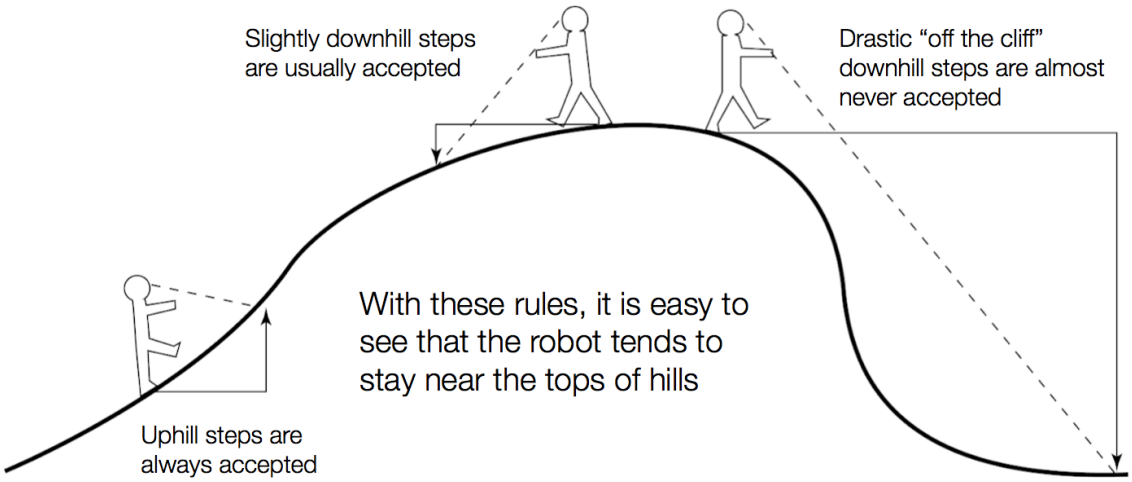


Figure 10.2: Depiction of a robot taking a step during an MCMC random walk. Figure courtesy of Paul O. Lewis.

In practice, the increase or decrease in posterior probability of a proposed step is measured by the ratio of probabilities:

$$R = \frac{P[\tau', Q', \eta'|A]}{P[\tau, Q, \eta|A]} = \frac{P[A|\tau', Q', \eta']P[\tau'|\eta']P[\eta']P[Q']}{P[A|\tau, Q, \eta]P[\tau|\eta]P[\eta]P[Q]}$$

where (τ, Q, η) is the current position in the parameter space and (τ', Q', η') is the proposed new position. The step is then evaluated by drawing a number u uniformly at random from $[0, 1]$. If $u \leq R$, the step is accepted; in particular if $R \geq 1$, the step is always accepted (since $\forall u, u \leq 1$). Figure 10.3 shows this calculation applied

to our previous robot example, where the height of the hill represents the posterior probability.

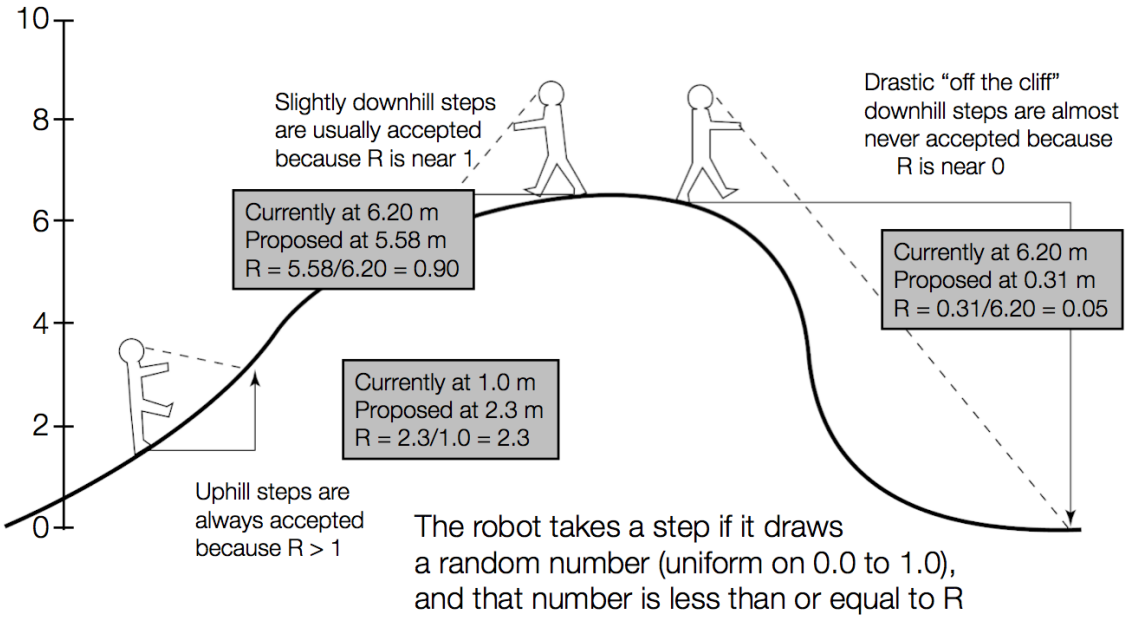


Figure 10.3: Depiction of a robot taking a step during an MCMC random walk, with calculation of height ratios R. Figure courtesy of Paul O. Lewis.

10.2.2 Metropolis-Hastings algorithm

In contrast with the previous section, the Metropolis-Hastings algorithm weighs steps not only according to the ratio of posterior probabilities, but also according to a proposal ratio. The ratio used to score steps in the Metropolis-Hastings algorithm, also called the Hastings ratio, then becomes:

$$HR = \frac{P[\sigma'|D]g(\sigma|\sigma')}{P[\sigma|D]g(\sigma'|\sigma)}$$

The proposed step here is $\sigma \rightarrow \sigma'$ and $g(a|b)$ is the probability of proposing a move to position a given the current position is b.

This proposal ratio is designed to accommodate non-symmetric proposals, i.e. when steps are not proposed uniformly at random. In the case of non-symmetric proposals, an algorithm using only the posterior ratio could otherwise be biased towards a certain type of step simply because it is proposed more often.

Example in two dimensions

The following example shows the steps taken by an MCMC Metropolis-Hastings walk on a likelihood surface in two dimensions with three peaks, represented as circles on the surface. Steps are proposed uniformly at random in all directions, with the step size drawn from a gamma distribution. Steps that otherwise would go

out of the surface are reflected at the edges. Figure 10.4 shows the result on this surface of a truly random walk, where every step is automatically accepted. It is easy to see that this walk provides no information on the underlying distribution.

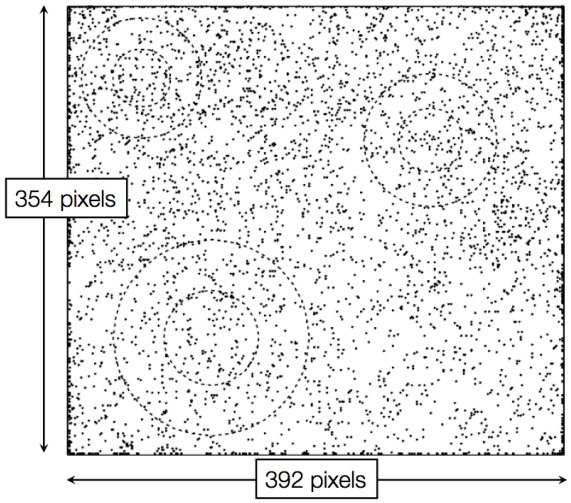


Figure 10.4: Output of a purely random walk on a likelihood surface, for $S = 5000$ steps. Circles represent peaks in the posterior distributions. Dots mark the positions explored during the random walk.

Figure 10.5 shows what happens when steps are accepted according to the MCMC criteria: the MCMC walk quickly finds one of the peaks and starts gravitating around it. We can see that since the initial position was far away from regions of high likelihood, the first few steps are not representative of the underlying distribution. These steps, called *"burn-in" steps*, should thus be discarded from the final output before drawing any conclusions about the posterior distribution.

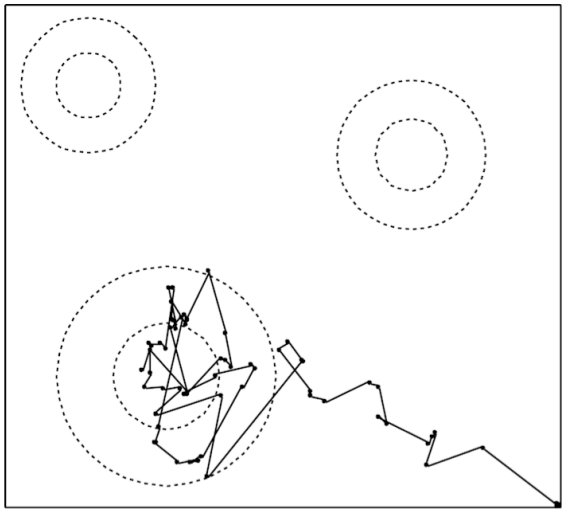


Figure 10.5: Output of an MCMC walk on the same likelihood surface, for $S = 100$ steps. The circles represent peaks in the posterior distribution and dots mark the positions explored during the MCMC walk.

Finally, figure 10.6 shows a longer run of the MCMC algorithm: all three peaks were found, and samples cluster around the peaks. In this case, 51.2% of the steps were taken in areas of the surface containing 50% of the likelihood weight, and 93.6% of the steps were taken in areas containing 95% of the likelihood weight, which represents a good approximation of the target distribution. This approximation becomes better the longer the chain runs.

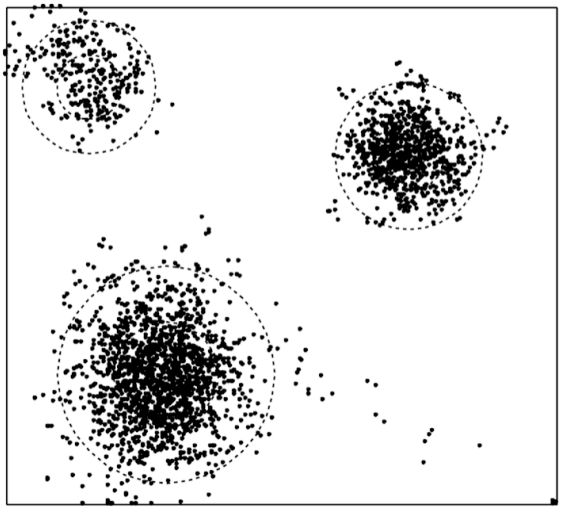


Figure 10.6: Output of an MCMC walk on the same likelihood surface, for $S = 5000$ steps. As before circles represent peaks in the posterior distributions and dots mark the positions reached by the MCMC walk.

As shown in the example, the output of a Metropolis-Hastings algorithm is a series of samples of the parameter space. If the inference has run for long enough so that the space is explored properly, the higher the number of samples from a particular area of the parameter space, the higher the posterior probability of this area. The posterior distribution for each parameter can be reconstructed from the samples by removing the burn-in samples and building a histogram of the values sampled, as shown in figure 10.7.

10.2.3 Application to phylogenetics

The MCMC algorithm relies on appropriate move proposals that allow to explore the parameter space efficiently. In phylogenetic applications, new values for the numerical parameters (i.e. Q, η) are proposed by scaling the current values, whereas new tree topologies (τ) are proposed via the moves presented in chapter 6: nearest-neighbour interchange (NNI), subtree pruning and regrafting (SPR) and tree bisection and reconnection (TBR).

The output of the MCMC algorithm is a series of samples from the parameter space (typically in the form of a log-file). Each sample contains the tree topology and associated branch lengths, as well as the values of all numerical parameters from the evolutionary and the tree generating models.

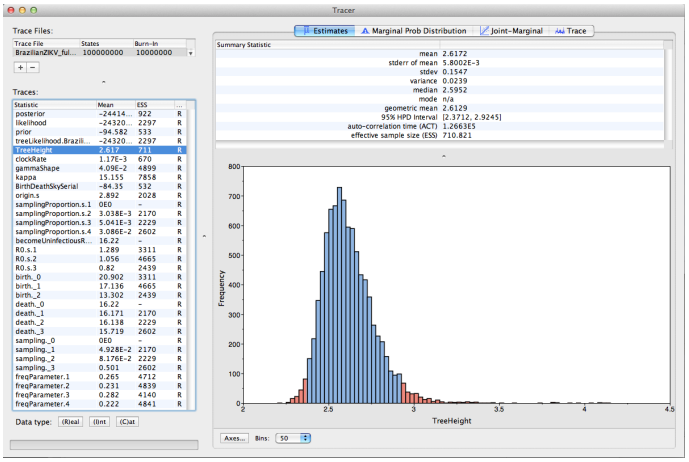


Figure 10.7: Posterior distribution of the tree height based on Zika samples from South America, visualized in the software Tracer [**Tracer**]. The tree height is the time between the first branching event and the most recent sample in the tree.

Available tools for phylogenetic applications in which the MCMC Metropolis-Hastings algorithm is implemented include:

- BEAST [**Beast**]
- BEAST2 [**Beast2**]
- MrBayes [**MrBayes**]
- RevBayes [**RevBayes**]

10.3 Comparison with Maximum Likelihood

In addition to the inclusion of priors, Bayesian inference has a few important differences in comparison to the Maximum Likelihood inference, which can be especially important for phylogenetic inferences.

10.3.1 Credible intervals

Bayesian inference does not produce confidence intervals around an estimate, but rather credible intervals which contain a range of 'credible' values, i.e. values that are supported by the posterior distribution. A 95% credible interval is defined as an interval of the posterior distribution containing 95% of the probability. Several different credible intervals can be constructed depending on which 5% of the distribution are neglected, as shown in figure 10.8.

The most commonly chosen credible interval is the smallest possible credible interval, also called the Highest Posterior Density interval (HPD, in red in figure 10.8). Credible intervals are especially useful for phylogenetic inference because they can be computed for complex objects such as tree topologies, in contrast to confidence intervals which can only be computed for numerical parameters. An example is

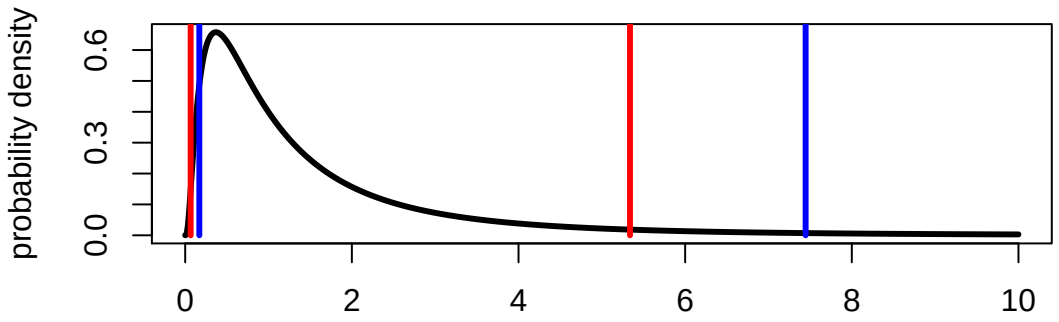


Figure 10.8: Posterior distribution with two possible credible intervals: a symmetric credible interval where 2.5% of the distribution is neglected at both ends (in blue) and the smallest possible credible interval (in red).

shown in figure 10.9, which shows the most likely topology given by the inference but also all the other credible topologies. This can help to identify the uncertainty associated with individual clades as well as with the entire tree.

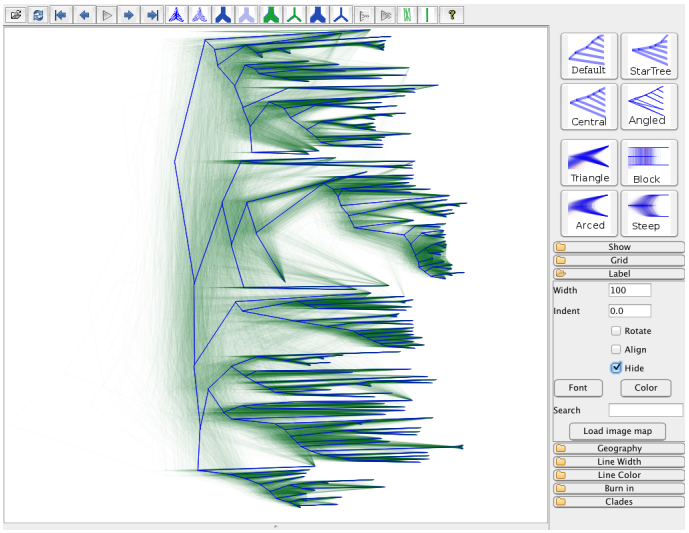


Figure 10.9: Posterior distribution of Zika trees, visualized in DensiTree [**DensiTree**]. The most likely topology is in blue, the credible trees are shown in green.

10.3.2 Chained Maximum Likelihood inference

Phylogenetic analysis aims to estimate two distinct features of the data. The tree topology and the parameters of the evolutionary model are estimated from the data using $P[A|\tau, Q]$, and the population dynamics model is estimated from the tree topology using $P[\tau|\eta]$. Bayesian inference provides a natural way of jointly inferring tree topology, evolutionary parameters and population dynamics parameters. Since every sample contains values for all the co-estimated objects, the posterior distribution for each individual parameter or topology can be obtained from the

output:

$$P[\tau, Q, \eta|A] = \frac{P[A|\tau, Q]P[\tau|\eta]P[\eta]P[Q]}{P[A]} \ .$$

Maximum likelihood inference uses one of two possible methods: a *chained* approach, or a *marginalization* approach. In the chained (also known as two-steps) approach, we obtain the ML estimates for the tree topology $\hat{\tau}$ and evolutionary model $\hat{Q}$ based on the data A, wherafter obtain the ML estimate for the population dynamics model $\hat{\eta}$ based on $\hat{\tau}$ and $\hat{Q}$. This only gives us the joint probability of the data and the topology given the evolutionary and tree generating parameters:

$$P[A, \tau|Q, \eta] = P[A|\tau, Q]P[\tau|\eta] \ .$$

It is equivalent to applying ML inference first to the $P[A|\tau, Q]$ component of the likelihood, and then to the $P[\tau|\eta]$ component. This approach is statistically problematic and completely discards the uncertainty on the estimates of τ and Q: we base the second inference on a single topology $\hat{\tau}$ whereas often several topologies are equally likely.

In the marginalization approach, the tree topology is simply integrated out using:

$$P[A|\eta, Q] = \int_{\tau} P[A, \tau|Q, \eta] d\tau \ .$$

However, this approach does not provide an estimate of τ, and requires to enumerate all possible tree topologies for the integration, which is generally not feasible.

10.3.3 Limitations of Bayesian inference

Bayesian inference tends to be much slower than Maximum Likelihood estimation, as it needs to explore a wider part of the parameter space to obtain the full posterior distribution. In particular building trees of more than a few thousand tips tends to be prohibitively slow in a Bayesian framework. The second issue with Bayesian inference is the necessary choice of priors. Priors can be difficult to choose or define, especially for little known parameters, but they have a big impact on the inference. Inappropriate priors are an important problem in the Bayesian framework, since they can bias the result in unpredictable ways.

10.4 Examples

In this section we present a few studies in which Bayesian inference was used, both for macroevolutionary and epidemiological applications.

HCV epidemic in Egypt

As presented in chapter 9, the Hepatitis C epidemic in Egypt during the last century has been widely studied. In particular because it progressed very differently from

neighbouring countries. Figure 10.10 shows the infected population size in time, using MCMC inference with a coalescent model [**DrummondEtAl2005**]. Figure 10.11 shows the reproductive number over time, estimated using birth-death models [**StadlerEtAl2013PNAS**].

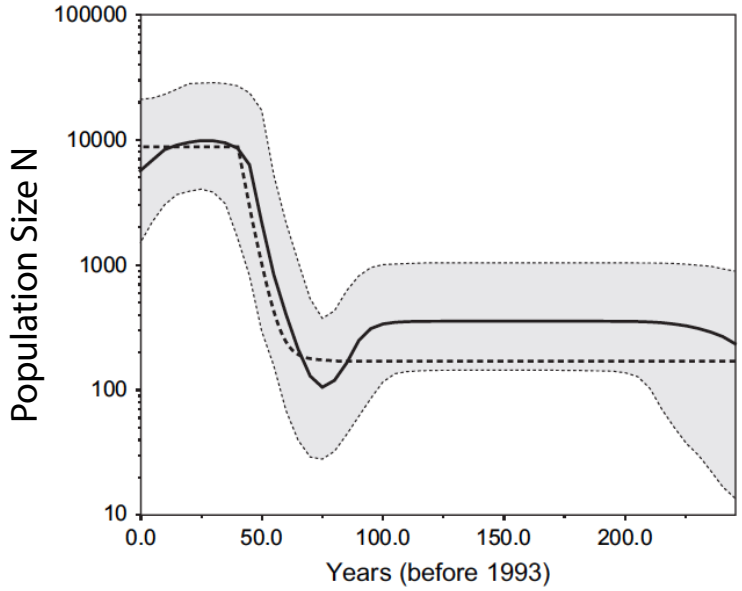


Figure 10.10: Change in size of the population infected with Hepatitis C in Egypt through time, inferred from HCV sequences using BEAST2 with a coalescent model. Figure adapted from [**DrummondEtAl2005**].

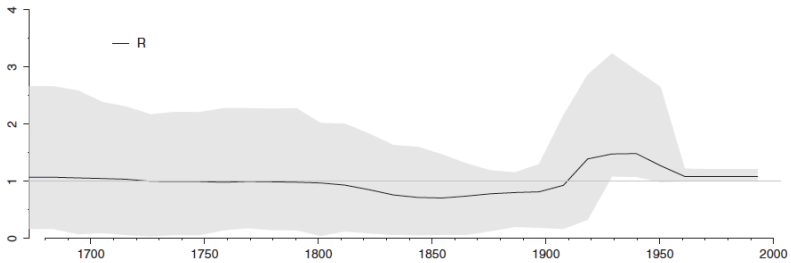


Figure 10.11: Change in reproductive number $R0$ of Hepatitis C in Egypt through time, inferred from HCV sequences using BEAST2 with a birth-death skyline model. Figure adapted from [**StadlerEtAl2013PNAS**].

Both of these analyses show that the epidemic spread rapidly between years 1920 and 1960, which is consistent with the hypothesis that the epidemic was spread by the schistosomiasis treatment program that took place during this period (see chapter 9 for more details).

Spread of Ebola

As we saw in chapter 9, the recent Ebola epidemic in West Africa was studied using phylogenetic and phylodynamic methods. Figure 10.12 shows the phylogeny

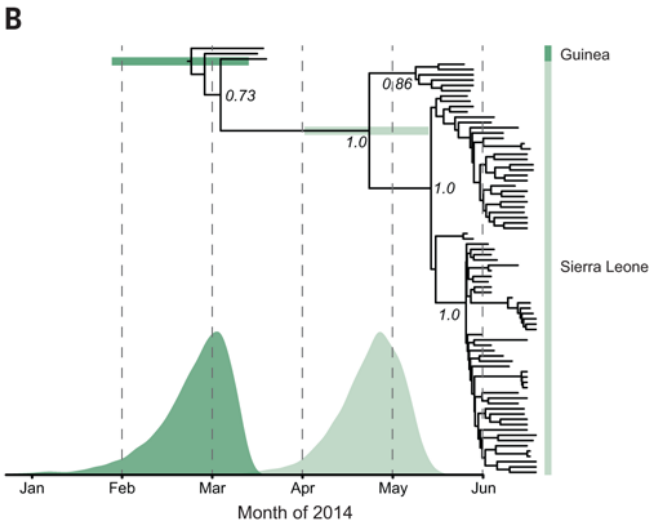


Figure 10.12: Maximum Likelihood phylogenetic tree of Ebola sequences in West Africa. Figure adapted from [**Gire2014**].

of sequences sampled in Sierra Leone, obtained using Maximum Likelihood inference. The ML estimates for the birth and death rates of the epidemic on this phylogeny were also obtained [**Stadler2014PLOSCur**], giving an estimate of the reproductive number $R_0 = 1.34$ (CI 1.12 - 1.55).

Applying Bayesian inference gives a more detailed picture of the reproductive number, taking into account the uncertainty around the phylogeny. Figure 10.13 shows the results of this analysis: we can see that the reproductive number started at a high level before going down as time passed.

Phylogeny of penguins

This example illustrates the use of Bayesian inference to study macroevolution, in this case the evolution of penguins. The analysis used the DNA sequences of extant species, but also the dates and morphological characteristics of several fossils. The phylogeny obtained is shown in figure 10.14.

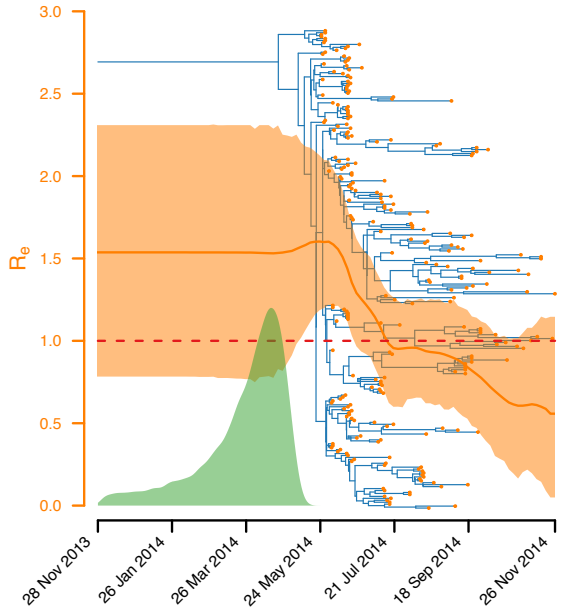


Figure 10.13: Phylogenetic tree, posterior distribution of the start of the epidemic and median with 95% HPDs of the reproductive number. Analyses were performed in BEAST2. Figure adapted from [**Stadler2014PLOSCur**].

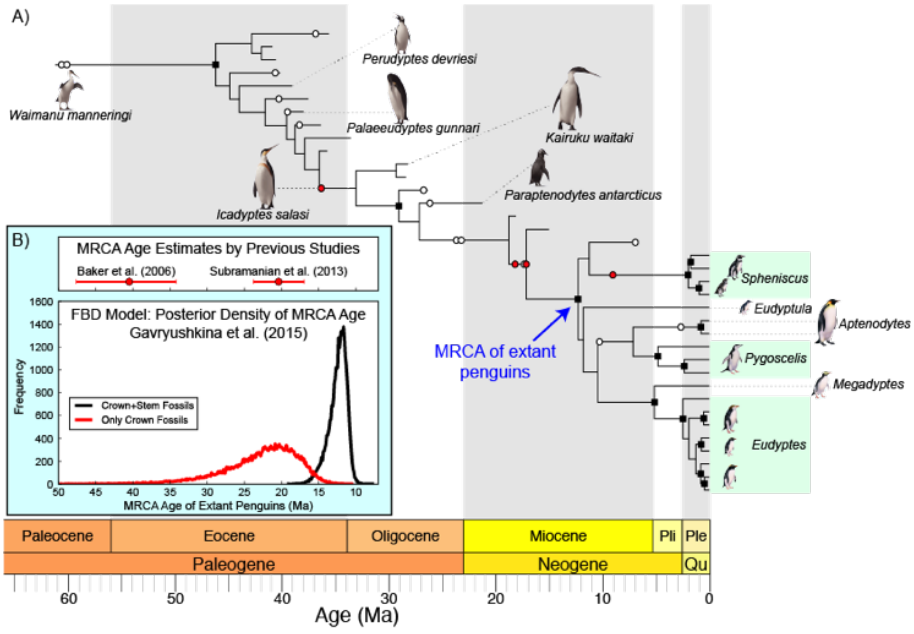


Figure 10.14: Penguin phylogeny obtained from fossil dates, morphology, and extant species sequencing data. Analysis was performed in BEAST2 using the Fossilized Birth-Death model. Figure and methods published in [**Gavryushkina2017**].

11 Phylogenetic networks

Previously we have talked about two types of methods: methods that analyse genetic sequences while ignoring their phylogenetic relationships completely, such as GWAS, and methods that assume that genetic sequences are related via a tree, which have been our main focus. In this chapter we will give an overview of further developments in phylogenetics, which mainly concern the incorporation of network structure into the inferences. While trees are good approximations of some processes, such as the clonal reproduction of bacteria, they are not structurally appropriate for representing processes such as species hybridization, virus recombination, etc. We will go through some developments in this area and discuss the different methods and the datasets they are appropriate for.

11.1 Sexual reproduction

Sexual reproduction is one of the extreme modes of organism reproduction, and also the way the human race reproduces. However, this means that human genomes cannot be properly analysed using a tree structure. For example, if a genome has two mutations, one on chromosome 3, the other on chromosome 5, we can not assume that they are linked since the chromosomes might have come from different parents. Even if both came from a single parent, that parent could have gotten the two from different ancestors. Moreover, not only do we have two possible sources for each of our chromosomes, our chromosomes are also subject to recombination, which makes the tracking of genes even harder. Basically, each of the SNPs in our genome has its own evolutionary history, which in theory can be represented as a tree, however the probability that two SNPs would share the same evolutionary tree is extremely low. In addition, none of the currently existing methods can incorporate that many trees and we are unable to gather such extensive genetic data. Thus, in this case, the mutations that we see in our genome are essentially independent and we can use methods such as GWAS to determine whether a trait is linked to a particular mutation.

11.2 Asexual reproduction

The other extreme mode of reproduction is asexual reproduction, which is typical for e.g. bacteria. In this case, all sites in an alignment are completely linked as they share the same evolutionary tree. So, if we were to ask whether a certain phenotype is explained by a certain gene, we would have to account for the ancestral structure

and could not assume that our data points (different genes) are independent. For example, we would like to determine the genotype responsible for the fast development of AIDS in HIV-infected patients. Assume this quick development phenotype is due to a mutation on site 1. However a different site, e.g. site 2, also evolved on the same tree, so mutations in site 1 are possibly correlated with mutations in site 2. If we assume that the sequences are independent and use an approach such as GWAS, we could get a false signal saying that site 2 is also responsible for the phenotype in question. To avoid such effects we need to take the ancestral structure into account using phylogenetic comparative methods or phylodynamic methods and make the assumption that all sequences share the same phylogenetic tree. Note that in the case of HIV and other recombinant viruses such an approach is only appropriate for between-host evolution, as recombination only happens in within-host evolution.

11.3 Between the extremes

11.3.1 Incomplete lineage sorting

If we look at all other possible situations, we have multiple options to consider. For example, when representing species evolution in trees, one has to bear in mind that each particular branch does not represent a single individual, but rather a whole population of individuals, and the branching events are not as obviously defined as the division of bacteria. A population might split up to give rise to two distinct species due to some physical or reproductive barrier, however it is not instantaneous and the gene pool might not be different enough for such a split to be visible for a while. Thus, we obtain two distinct types of trees, *gene trees* that follow the evolution of a specific gene and *species trees* which follow the evolution of whole populations. Since different genes may possess distinct ancestries, the gene trees corresponding to different genes may also be distinct—even when they come from the same set of species.

The gene trees may also differ from the species tree. For instance, suppose at some point in time two distinct species exist, and that each of the corresponding populations contain an ancestral gene tree lineage. If these species lineages merge (backward-in-time view) to form a single ancestral population, the gene tree lineages will not coalesce immediately but rather at some older time. This means that the divergence times in a gene tree are usually older than the corresponding nodes in the species tree. Furthermore, if there are more than two ancestral gene tree lineages extant at the time of the speciation event, the order of the coalescence events in the gene tree may differ from the order of the events in the species tree, leading to topological differences between the gene tree and species tree. This effect is known as *incomplete lineage sorting*.

Let us take a look at the example of great apes. The research community is now quite sure that humans are more closely related to chimpanzees than to gorillas. A problem would arise if we look at a gene that coalesces first between chimpanzees and gorillas, and only then coalesces with the appropriate human gene, which can

easily happen at random. If we were to build a phylogenetic tree only judging by this particular gene, we would infer a false relationship between the species. Such instances of incomplete lineage sorting were in fact the reason for long debates over the relationships between humans, chimps, and gorillas. Roughly a third of the trees clustered chimp and gorilla together, while the rest showed that humans and chimps are more closely related, thus some scientists were more inclined to believe that chimps and gorillas are closer relatives (see Figure 11.1). When taking into account incomplete lineage sorting one can compute the expected number of gene trees differing from species trees, and this can be used to explain the disparity in gene trees. After all, it turns out that humans and chimps are closer relatives than either of the species to gorillas. Something that one has to keep in mind when dealing with species and gene trees, is that for a species tree with at least 4 tips one can select such branch lengths that the most likely gene tree will be inconsistent with the species tree. The consequence of this fact is that one can not simply construct gene trees for all available genes and then pick the most common structure as the species tree, as it might be incorrect. Proper analyses of such data can be done using a software framework such as *BEAST2 [***BEAST2**] (pronounced "star Beast 2").

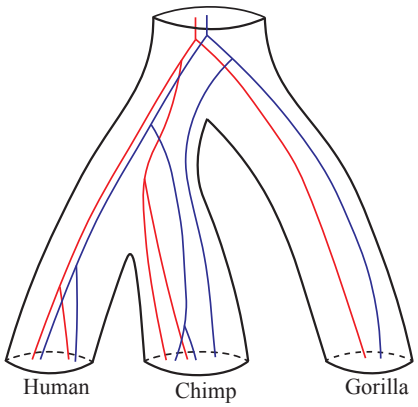


Figure 11.1: An example of two different gene trees embedded in the species tree of Humans, Chimps and Gorillas.

11.3.2 Hybridization

Incomplete lineage sorting, while complex, still only involves tree structures, even though layered on one another. True networks are necessary to describe the evolution of species that include hybridization. Some populations can merge and form a hybrid species, which is very common in plants and in some fish. If populations are close enough in terms of their genome, when they mix geographically, for example, they also mix their genomes together, thus basically forming a new species from the genetic material of the two. This is also one of the main reasons why species definition is a hard task. In this case we would need to specifically account for merging events, as trees will not represent such structures correctly. The structures representing such networks should explicitly allow hybridization nodes, nodes that have two ancestral lineages and only one descendant lineage, as shown in Figure 11.2. A method for

Bayesian inference of species networks under precisely this model has been recently been published. [**Zhang2017**].

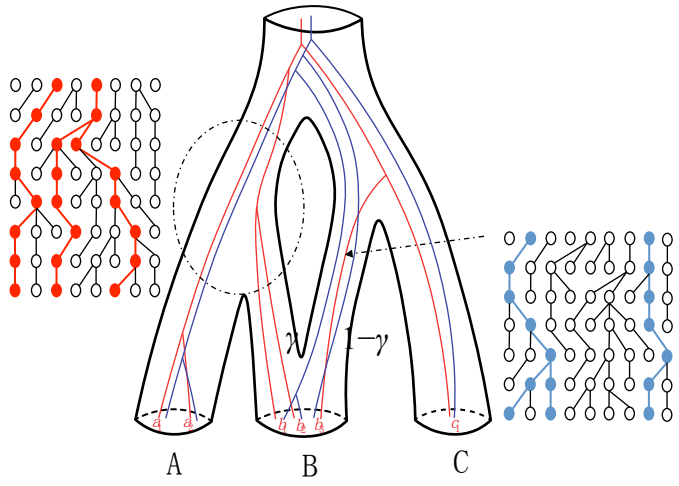


Figure 11.2: An example of a species network and its relation to embedded gene trees [**Zhang2017**].

11.3.3 Lateral gene transfer

Previously we have assumed that we know which parts of a genome have separate trees, e.g. for plasmids and the core genome in bacteria, or separate genes in plant hybridization. However as we look closer into lateral gene transfer we can see that there can be arbitrary breakpoints even within genes, which means that we have no means of clearly defining the parts of the genome that have the same tree. Such datasets are hard to analyse, as one has to account for all of the possible genome breakpoints and allow genetic fragment transfer at arbitrary lengths. An example of a possible tree with all the gene transfer events can be seen in Figure 11.3. There are many different tools that can perform inference under models which include the effects of LGT and recombination, a recent example of which is Bacter [**Vaughan2017**]. However these inferences are often limited to very small datasets due to the computational complexity involved.

11.3.4 Virus recombination

Viruses are special in the way that they can not replicate on their own and need to be within a host cell to reproduce. However, if two different virus strains infect the same cell, they can recombine within it and then the infected cell will produce new virions with the recombined genome.

11.3.5 Eukaryote recombination

Eukaryote cells can recombine in two ways, both happening during cell meiosis – the cell division that produces gametes by splitting the chromosomes. One way this

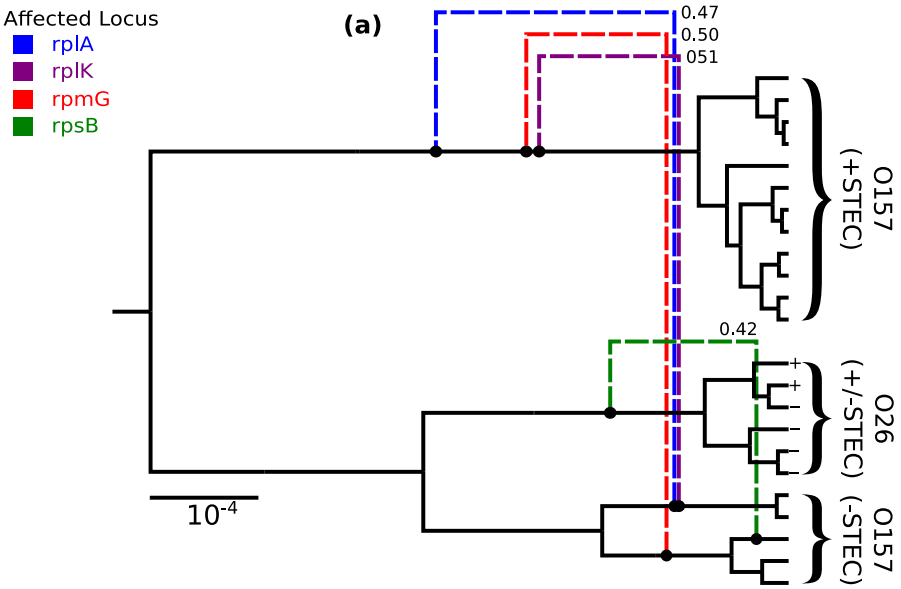


Figure 11.3: An example of lateral gene transfer inference from *E. coli* genomes[**Vaughan2017**]. In this case the tree in black is the inferred underlying phylogenetic tree resulting from the clonal reproduction process, while the coloured dashed lines show gene transfer. The numbers on lines indicate the posterior support for each transfer event, while the colours indicate the gene that is transferred.

can happen is gene conversion, a process where a part of the DNA is replaced by the homologous strand copied from another piece of DNA, which mainly happens due to base mismatch repair. The second way is through crossover, when parts of the genome are exchanged between strands. In general however, eukaryote sequences are already extremely hard to analyse using trees, so typically complete independence between ancestral trees for different SNPs is assumed and non-phylogenetic tools such as GWAS are employed.

List of Symbols

$Cov(X, Y)$	covariance of random variables X, Y $Cov(X, Y) = E\left[(X - EX)(Y - EY)\right]$
EX	mean (expectancy) of random variable X $EX = \int^{\mathcal{X}} x f_X(x) \mathrm{d}x$ if X is a continuous random variable $EX = \sum_{n \in \mathcal{X}} nP(X = n)$ if X is discrete
$f_X(\cdot)$	probability density function of random variable X
$\mathcal{H}_0$	null model or null hypothesis
$P(\cdot)$	probability of the set $\cdot$
$P(\cdot\|*)$	conditioned probability of set $\cdot$ given set $*$
$\mathcal{T}$	phylogenetic tree
$VarX$	Variance of random variable X $VarX = E(X - EX)^2$